

1 Inleiding

De expertgroep heeft in het najaar van 2010 naar aanleiding van het koppelvlak tussen BAG en WOZ vastgesteld dat er nieuwe koppelvlakken aan een sectormodel mogen worden toegevoegd. Dit heeft consequenties voor het beheermodel, omdat het huidige beheermodel is gestoeld op de veronderstelling dat elke wijziging in een sectormodel leidt tot een nieuwe versie. De afgelopen maanden is er binnen expertgroep en regiegroep enkele malen gediscussieerd over de uitgangspunten voor het wijzigen van dit beheermodel. Daarnaast is in de expertgroep een voorstel gedaan voor een nieuwe opbouw van een sectormodel.

Dit document wil de nieuwste inzichten rond de opbouw van sectormodellen en best practices voor het maken van sectormodellen vertalen naar een beheermodel door te definiëren wat de te beheren onderdelen van een sectormodel zijn en voorstellen te doen voor het beheer van zo'n component.

2 Van informatiemodel naar berichtenmodel

De StUF-standaard gaat ervan uit dat het domein waarvoor berichten gemaakt worden beschreven is in een informatiemodel met de relevante objecttypen en hun eigenschappen. Voor de gemeentelijke basisgegevens is het RSGB zo'n model en voor de zaakgegevens het RGBZ. Voor de sector WOZ is de catalogus WOZ het informatiemodel. Zo hoort er onder elk sectormodel een informatiemodel te liggen met de definitie van de gegevens.

De StUF-standaard gaat ervan uit dat de elementen in de berichten één-op-één te herleiden zijn tot attributen en relaties van objecttypen in het informatiemodel. Voorwaarde hiervoor is dat in het informatiemodel de te gebruiken xml-tag voor een attribuut of relatie wordt gedefinieerd en dat in het informatiemodel een korte aanduiding of mnemonic voor alle onderkende objecttypen (een objecttype kan ook een relatie zijn) wordt gedefinieerd.

2.1 Van objecttypen naar (bericht)entiteiten

Idealiter worden de objecttypen uit het informatiemodel één-op-één vertaald naar entiteiten binnen de berichten. In de praktijk leidt dit niet altijd tot optimale berichten en services. Het is bijvoorbeeld niet verstandig om subtypen van Natuurlijk persoon als ingezetene, niet-ingezetene, ingeschreven persoon en ander natuurlijk persoon ook in de berichten te onderkennen, omdat niet alle systemen die berichten verzenden en ontvangen deze subtypen hoeven te kennen. Ander voorbeeld, het maakt berichten een stuk simpeler als relaties naar objecttypen als Land of Woonplaats of Openbare ruimte niet expliciet worden gelegd. Deze relaties uit het informatiemodel kunnen in de berichten worden geïmplementeerd door gegevens van het gerelateerde object op te nemen in het object vanwaaruit de relatie ligt. Dit is in bg0310 onder meer gedaan bij de implementatie van adressen. De gemaakte keuzen bij het vertalen van het informatiemodel naar een berichtenmodel dienen verantwoord te worden in een zogenaamd keuzenVerStUF-fing-document voor een sectormodel.

Het informatiemodel is een model waarin de richting van relaties tussen objecttypen veelal geen rol speelt. Vanuit het ene objecttype kunnen via relaties langs verschillende paden andere objecttypen worden bereikt. Vanuit een persoon kan ik bijvoorbeeld rechtstreeks naar het objecttype adres gaan of via de ouder van die persoon. In het eerste geval krijg je het adres van de persoon zelf en in het tweede geval het adres van één van zijn ouders. Een berichtenmodel bestaat per entiteit uit een hiërarchische structuur. Bovenaan in de hiërarchie staat de entiteit zelf met de elementen voor de

attributen van het objecttype. Daarnaast bevat de entiteit elementen voor relaties naar andere objecttypen. Het is een StUF best-practice om bij het maken van een sectormodel allereerst basis-entiteiten als complexTypes te maken, waarin een relatie en zijn gerelateerde altijd ook basis-entiteiten zijn. Een bericht met zo'n basistentiteit kan willekeurig groot worden. Ik kan voor een persoon via de heeftAlsOuders relatie bijvoorbeeld teruggaan tot Adam en Eva.

De StUF-standaard kent voor de identificatie van objecten in berichten het begrip kerngegevens. De kerngegevens zijn de gegevens waarmee een object in verreweg de meeste gevallen uniek geïdentificeerd kan worden, zelfs als niet alle kerngegevens beschikbaar zijn. Voor een natuurlijk persoon zijn de kerngegevens bijvoorbeeld het burgerservicenummer, de naamsgegevens, het adres, de geboortedatum en het geslacht. Het is een StUF best-practice om naast de complexTypes voor de basistentiteiten en ook een complexType voor de kerngegevens te definiëren. Het basis- en kerngegevens-complexType worden opgenomen in een schema met als naam sss_ent_basis.xsd met sss de code voor het sectormodel. Het complexType voor de kerngegevens is een restriction van het basis-complexType. Voor relaties wordt alleen een kerngegevens-complexType gedefinieerd, als er meer kerngegevens zijn dan de entiteit vanwaaruit de relatie ligt en de gerelateerde entiteit. Bij een huwelijk is dit bijvoorbeeld het geval, omdat een persoon meerdere keren met dezelfde partner kan trouwen. De datumSluiting is ook een kerngegeven.

StUF definieert functionaliteit voor twee veel voorkomende interactiepatronen, het doorgeven van mutaties en het opvragen van gegevens. Het is een StUF best-practice om de entiteiten die binnen de functionaliteit voor mutaties en voor vraag/antwoord worden gebruikt te definiëren als restrictions op de entiteiten in het sss_ent_basis-schema in twee schema's met als namen sss_ent_mutatie.xsd en sss_ent_vraagAntwoord.xsd. Hieronder worden de regels hiervoor nader toegelicht. Een samenvatting van de besproken complexTypes staat in onderstaande tabel. M staat hier voor de mnemonic voor een objecttype. Het kan hierbij ook gaan om een relatie. De mnemonic voor een objecttype heeft veelal als vorm XXX en de mnemonic voor een relatie XXXYYY of XXXYYYZZZ, als er meerdere relaties zijn tussen de objecttypen XXX en YYY.

Naam	Functie	Schema
M-basis	ComplexType gebruikt als basis voor de complexTypes in de berichtdefinities	sss_ent_basis.xsd
M-kerngegevens	ComplexType met de kerngegevens voor een objecttype	sss_ent_basis.xsd
M-kennisgeving	ComplexType gebruikt binnen een kennisgevingbericht	sss_ent_mutatie.xsd
M-kerngegevensKennisgeving	ComplexType gebruikt als gerelateerde binnen een kennisgevingbericht	sss_ent_mutatie.xsd
M-antwoord	ComplexType gebruikt binnen een antwoordbericht en het start element in een vraag-bericht	sss_ent_vraagAntwoord.xsd
M-historieFormeel	ComplexType voor het historieFormeel element	sss_ent_vraagAntwoord.xsd
M-historieFormeelRelatie	ComplexType voor het historieFormeelRelatie element binnen een relatie	sss_ent_vraagAntwoord.xsd
M-historieMaterieel	ComplexType voor het historieMaterieel element	sss_ent_vraagAntwoord.xsd
M-gerelateerdeAntwoord	ComplexType voor een gerelateerde in een antwoordbericht	sss_ent_vraagAntwoord.xsd
M-vraag	ComplexType voor het gelijk, vanaf, totEnMet en	sss_ent_vraagAntwoord.xsd

Naam	Functie	Schema
	scope element in een vraagbericht	
M-vraagScope	ComplexType voor het scope-element in een vraagbericht, als dit een andere inhoud heeft dan het complexType voor het gelijk, vanaf en totEnMet element	sss_ent_vraagAntwoord.xsd
M-vraagSelectie	ComplexType voor het gelijk, vanaf en totEnMet element in een vraagbericht, als dit een andere inhoud heeft dan het complexType voor het scope-element	sss_ent_vraagAntwoord.xsd
M-gerelateerdeVraag M-gerelateerdeVraagScope M-gerelateerdeVraagSelectie	ComplexTypes voor een gerelateerde in een vraagbericht.	sss_ent_vraagAntwoord.xsd

Het is best-practice om in het mutatie-schema gerelateerden met alleen de kerngegevens te gebruiken en in het vraag/antwoord schema gerelateerden met de inhoud van een kennisgeving uit het mutatie-schema. De achtergrond voor de keuze voor kerngegevens in een gerelateerde in een mutatie is dat het voor een mutatie voldoende is om de gerelateerde te kunnen identificeren c.q. te kunnen vastleggen met gegevens die de gerelateerde uniek identificeren. In een latere mutatie kunnen de gegevens van een gerelateerde altijd nog aangevuld worden. Bij het opvragen van de gegevens van een objecttype wil je regelmatig meer weten dan alleen de kerngegevens van een gerelateerde. Om die reden wordt bij vraag/antwoord een gerelateerde met de inhoud van een kennisgeving gebruikt.

In het informatiemodel hebben veel relaties geen richting c.q. is een relatie slechts in uitzonderingssituaties zonder zijn inverse relatie gedefinieerd. Bij het definiëren van de basis-entiteiten, de vraag/antwoord entiteiten en de kennisgeving-entiteiten staat de ontwerper van een sectormodel steeds voor de vraag of de relatie in beide richtingen of slechts in één richting wordt opgenomen. Het is een StUF best-practice om geen verschil te hebben in de relaties in de basisentiteiten en vraag/antwoord-entiteiten. Het is een StUF best-practice om relaties in kennisgeving-entiteiten slechts vanuit één kant te onderhouden en wel vanuit de entiteit die het minst vaak al bestaat bij het leggen van de relatie. Bij het leggen van een relatie tussen een document en zaak wordt de relatie gelegd vanuit document: de zaak bestaat vaak al voor het document, omdat de meeste documenten in het kader van een zaak worden aangemaakt.

Het is een StUF best-practice om de schema's zo scherp mogelijk te definiëren. In de kennisgeving-complexTypes is daarom bijvoorbeeld altijd het attribute StUF:verwerkingssoort verplicht en mag een attribute als StUF:scope niet voorkomen. In sss_ent_mutatie.xsd wordt om deze reden ook een kerngegevensKennisgeving-complexType gedefinieerd met alleen de kerngegevens voor gebruik als gerelateerde in het kennisgeving-complexType voor een relatie. Als een gerelateerde in een relatie meer dan alleen de kerngegevens mag bevatten dan wordt hiervoor in sss_ent_mutatie.xsd een apart complexType gedefinieerd.

In sss_ent_vraagAntwoord.xsd worden alle complexTypes gebruikt in vraag/antwoordberichten gedefinieerd. Ook hier worden de complexTypes steeds zo scherp mogelijk gedefinieerd. Het scope-attribute mag bijvoorbeeld alleen voorkomen in M-vraag complexTypes. De historie-complexTypes dienen gedefinieerd te worden op basis van de specificatie voor materiële en formele historie in het sectormodel.

De schema's `sss_ent_basis.xsd`, `sss_ent_mutatie.xsd` en `sss_ent_vraagAntwoord.xsd` vormen de kern van een sectormodel. Ze moeten altijd gedefinieerd worden, omdat de hier gedefinieerde `complexType`'s ook de basis vormen van de `complexType`'s gebruikt binnen de update en vraag of selectie elementen in vrije berichten. Het is namelijk een StUF best-practice dat een update- of een vraag- of selectie-element in een vrij bericht altijd een restriction bevat op het kennisgeving-`complexType` c.q. het vraag- of antwoord-`complexType` uit `sss_ent_mutatie.xsd` c.q. `sss_ent_vraagAntwoord.xsd`. Deze drie schema's bevatten de vertaling van het informatiemodel naar de entiteiten gebruikt binnen de berichten. Zij specificeren ook de maximale grootte van een entiteit die in een bericht kan voorkomen voor de standaard door StUF gedefinieerde functionaliteit voor kennisgevingen en vraag/antwoord. Het zijn niet de enige drie schema's nodig voor het maken van een sectormodel. Omdat de andere schema's primair een technische functie hebben en geen functionele wordt de lezer voor een nadere beschrijving ervan verwezen naar het document `Opbouw0301Sectormodellen`.

Het staat de ontwerper van een sectormodel te allen tijde vrij om af te wijken van de hier beschreven best practices. Het is dan wel gewenst dat de reden voor deze afwijkingen wordt vermeld in het `keuzenVerStUF`-document.

Omdat de berichtentiteiten de basis vormen voor alle functionaliteit in het sectormodel mogen ze niet wijzigen binnen één versie van het sectormodel. Uiteraard mogen er wel conform de errata-procedure fouten in deze schema's worden verbeterd. Functionele wijzigingen en uitbreidingen van de berichtentiteiten mogen uitsluitend worden doorgevoerd in een nieuwe versie van het sectormodel met een nieuwe namespace.

3 Van berichtentiteiten naar services

De StUF-standaard beschrijft gedetailleerd de functionaliteit voor het doorgeven van mutaties met behulp van Lk01- en Lk02-berichten, het synchroniseren van databases met de Sa- en Sh-berichten en het bevragen van gegevens met de Lv01-/La01- t/m Lv10-/La10-berichten. Vanuit het hierboven gedefinieerde `sss_ent_basis.xsd` kunnen de berichten en services voor deze functionaliteit gegenereerd worden voor de protocolbinding gebaseerd op wsdl, soap en http¹. Ook voor andere protocolbindingen kunnen services worden gedefinieerd. In de praktijk is dit alleen nog gedaan binnen het sectormodel WOZ voor de Digikoppeling protocolbinding. Er zal daarom in dit stuk niet dieper worden ingegaan op andere protocolbindingen. Het stuk `Opbouw0301Sectormodellen` beschrijft de naamgevingsconventies voor deze schema's en wsdl's voor de protocolbinding gebaseerd op wsdl, soap en http.

Vanuit het oogpunt van beheer is van belang dat de aldus gegenereerde schema's en wsdl's meer berichten en services bevatten dan systemen in de praktijk implementeren. De schema's met de gegenereerde berichten zijn dus catalogi waaruit berichten geselecteerd kunnen worden voor een bepaalde protocolbinding. De gegenereerde wsdl's behorend bij de berichtcatalogi zijn voorbeeld-wsdl's.

In het document `Opbouw0301Sectormodellen` worden voor het doorgeven van mutaties, het synchroniseren en het bevragen twee berichtcatalogi onderscheiden:

1. De mutatie-catalogus met de berichten voor het doorgeven van mutaties en het

¹ Zie het document `stuf.bindingen.030200.pdf` voor een beschrijving van deze protocolbinding.

synchroniseren van databases

2. De vraagAntwoord-catalogus met de berichten voor het bevragen van databases.

De mutatie- en vraagAntwoord-catalogus kunnen geautomatiseerd worden afgeleid uit de kern van het sectormodel. Zij vallen daarom onder hetzelfde beheerregime als deze kern. Een functionele wijziging ervan is niet toegestaan en kan slechts worden doorgevoerd in een volgende versie van het sectormodel met een nieuwe namespace.

3.1 Andere berichtcatalogi en koppelvlakken

In deze paragraaf wordt allereerst een aantal begrippen beschreven aan de hand van definities, omschrijvingen en voorbeelden. Daarna wordt aangegeven wat te beheren eenheden zijn en worden regels voor het beheer van deze eenheden voorgesteld.

Berichtcatalogus en koppelvlak

Een sectormodel zal veelal de twee standaard berichtcatalogi bevatten en kan nog meer berichtdefinities bevatten. Vanuit het oogpunt van het gebruik van deze berichtdefinities is het belangrijk om berichtcatalogi en koppelvlakken te onderscheiden.

Een *berichtcatalogus* definieert samen met zijn voorbeeld-wsdl's een verzameling services waarvan een systeem mag aangeven in hoeverre het deze implementeert in voor dat systeem specifieke wsdl's. In de systeem-wsdl's worden de voorbeeld-wsdl's teruggesneden tot het koppelvlak van dat systeem.

Een *koppelvlak* binnen een sectormodel definieert een verzameling services op basis van berichten uit één of meer berichtcatalogi die een systeem in zijn geheel moet implementeren of daadwerkelijk implementeert. Er is sprake van 'moet implementeren' en een normatief koppelvlak, als het koppelvlak een normatieve beschrijving is van functionaliteit die van systemen wordt verwacht. Er is sprake van 'daadwerkelijk implementeert' en een beschrijvend koppelvlak, als het koppelvlak een feitelijke beschrijving is van de functionaliteit die een systeem biedt of verlangt van gebruikers van zijn services. De koppelvlakken van de Landelijke Voorziening BAG en WOZ zijn voorbeelden van een beschrijvend koppelvlak. Het koppelvlak dat de Landelijke Voorziening WOZ verwacht bij een afnemer of bij de gemeente zijn voorbeelden van normatieve koppelvlakken. In geval van een normatief koppelvlak moet een systeem in de eigen wsdl de message-, portType- en binding-elementen uit de koppelvlak-wsdl integraal overnemen.

De schema's en wsdl's binnen het sectormodel bg0310 voor de koppeling van BAG en WOZ-applicaties definiëren voor deze systemen een koppelvlak, omdat zowel de BAG als de WOZ-applicaties geacht worden deze specificatie integraal te implementeren. Voor een GBA-systeem is slechts een deel van de services relevant en vormen de schema's uit het BAG/WOZ-koppelvlak een berichtencatalogus. Omdat de hierboven beschreven situatie met WOZ en GBA zich wel vaker zal voordoen, wordt in het beheermodel de hierboven gegeven definitie voor een koppelvlak naar de letter genomen. Een koppelvlak is precies de verzameling services gebaseerd op één of meer onderliggende berichtcatalogi. De onderliggende berichtcatalogi bestaan onafhankelijk van het koppelvlak en kunnen onafhankelijk van het koppelvlak beheerd worden, mits er nooit berichten verwijderd worden uit een berichtencatalogus. In het geval van de BAG wordt er dus onderscheid gemaakt tussen een BAG-berichtencatalogus en de daarop gebaseerde BAG/WOZ en BAG/GBA koppelvlakken. Wie weet komen er in de toekomst nog koppelvlakken bij tussen de BAG en andere gemeentelijke systemen. Ook deze koppelvlakken worden gebaseerd op de BAG-berichtencatalogus.

Extra functionaliteit

Met de mutatie- en vraagAntwoord-catalogus benut een sectormodel slechts een klein deel van de mogelijkheden die de StUF-standaard biedt. Er kunnen ook samengestelde kennisgevingen worden gedefinieerd of vrije berichten.

In een samengestelde kennisgeving worden twee of meer kennisgevingen samengevoegd die als één transactie verwerkt moeten worden. De exacte inhoud van deze kennisgevingen kan scherper worden gedefinieerd door voor de enkelvoudige kennisgevingen eigen complexTypes te definiëren. Het is een StUF best-practice om alleen samengestelde kennisgevingen te definiëren, als de eventuele foutafhandeling bij verwerking in de database de som is van de foutafhandeling voor de enkelvoudige kennisgevingen erbinnen. Als er meer foutafhandeling vereist is, dan dient een vrij bericht gedefinieerd te worden met daarbinnen twee of meer elementen met de update-functie.

In een vrij bericht kan de functionaliteit van een enkelvoudige kennisgeving in een element met als functie update worden opgenomen of de functionaliteit van een vraagbericht in een element met als functie vraag. Voor een vrij bericht is altijd een beschrijving van de gewenste functionaliteit nodig.

Extra berichten in een sectormodel worden altijd ondergebracht in een berichtcatalogus. Het is mogelijk dat een berichtencatalogus slechts één bericht bevat.

Er zijn twee redenen om extra berichten en services te definiëren naast de berichten uit de mutatie-, vraagAntwoord- of een andere catalogus:

1. Restrictie: een systeem wil expliciet aangeven dat het strengere eisen stelt aan binnenkomende berichten dan het bericht uit de berichtcatalogus.
2. Extra functionaliteit: een systeem wil berichten met een andere functionaliteit definiëren.

Van restrictie is sprake als een bericht gedefinieerd kan worden als een restriction op het complexType voor een bericht in de mutatie-, vraagAntwoord- of een andere berichtcatalogus. Bij restrictie mag geen extra foutafhandeling worden gedefinieerd bovenop de standaard verwerking. Als het binnenkomende bericht valide is, dan dient het net zoals elke ander bericht van dat type verwerkt te kunnen worden. Als er toch extra eisen gesteld worden, dan dient een vrij bericht gedefinieerd te worden. Deze eis wordt gesteld om te waarborgen dat elke enkelvoudige kennisgeving en elk vraag/antwoordbericht met dezelfde functionaliteit verwerkt kan worden. In alle andere gevallen is er sprake van extra functionaliteit.

Een restrictie kan op twee manieren worden gedefinieerd:

1. De restrictie wordt in een separaat document beschreven, maar niet afgedwongen via een schema. Voor deze aanpak is gekozen bij het definiëren van het BAG/GBA koppelvlak binnen het sectormodel bg0204.
2. De restrictie wordt via een schema afgedwongen.

Als de restrictie via een schema wordt afgedwongen, wordt een nieuw berichtelement gedefinieerd gebaseerd op een complexType dat een restriction is van een bericht-complexType uit een andere berichtcatalogus. Restrictions op entiteiten uit andere berichtcatalogi worden ondergebracht in een schema met als naam sss_ent_ccc.xsd en de berichtelementen en de eventuele bericht-complexTypes van een berichtcatalogus worden ondergebracht in een schema met als naam sss_msg_ccc.xsd met ccc een aanduiding van de berichtcatalogus. Eventuele wsdl's volgen ook deze naamgevingssystematiek. Het is een StUF best-practice om restricties via een schema af te dwingen.

In geval van extra functionaliteit worden de daarvoor benodigde complexTypes voor entiteiten overgenomen uit een andere berichtcatalogus of gedefinieerd als restriction op complexTypes uit

een andere berichtcatalogus. Voor entiteit-complexTypes binnen een samengestelde kennisgeving of een element met als functie update in een vrij bericht geldt als extra eis dat deze altijd een restriction moeten zijn van het kennisgeving-complexType voor die entiteit in de mutatie-berichtcatalogus. Voor entiteit-complexTypes binnen een element met als functie vraag in een vrij bericht geldt als extra eis dat deze altijd een restriction moeten zijn van de vraag-complexTypes voor die entiteit (gelijk, vanaf, totEnMet en scope-element) of het antwoord-complexType (start-element). Deze complexTypes worden ondergebracht in het schema sss_ent_ccc.xsd.

Deze voorschriften waarborgen dat een complexType voor een entiteit altijd kleiner is dan een basis-complexType en dat in samengestelde kennisgevingen en in het update element van vrije berichten nooit een entiteit zit die niet via de standaard verwerking voor een kennisgeving kan worden verwerkt. Een element met functie vraag in een vrij bericht kan ook altijd met de standaard verwerking voor een vraagbericht verwerkt worden. Deze waarborgen maken hergebruik van functionaliteit mogelijk in software die StUF-berichten verwerkt.

3.2 Het beheermodel voor de overige berichtcatalogi en koppelvlakken

Voor de mutatie- en vraagAntwoord-berichtcatalogus is het beheer al gedefinieerd in het begin van dit hoofdstuk. Het is niet toegestaan om in deze berichtcatalogi functionele wijzigingen door te voeren of ze uit te breiden, omdat deze berichtcatalogi de basis vormen voor functionaliteit in de andere berichtcatalogi.

Voor de overige berichtcatalogi is dit een te zware eis. Het is voldoende te eisen dat een eenmaal toegevoegde berichtcatalogus nooit meer verwijderd mag worden uit het sectormodel en dat uit een berichtcatalogus nooit een berichtelement verwijderd mag worden. Aan een berichtcatalogus en de bijbehorende voorbeeld-wsdl's mogen altijd nieuwe berichtelementen en de bijbehorende services worden toegevoegd. Het is niet nodig om hierbij aan versiebeheer te doen of voorschriften te geven voor het moment waarop een berichtcatalogus mag worden uitgebreid. Er is alleen versiebeheer nodig bij het doorvoeren van errata. Nieuwe berichtcatalogi mogen door willekeurige partijen op willekeurige momenten worden toegevoegd aan een sectormodel na een toets door de expertgroep op het voldoen aan de in dit document geformuleerde regels voor een berichtcatalogus. De partij die een berichtcatalogus toevoegt is ook verantwoordelijk voor het beheer ervan, denk in het bijzonder aan het doorvoeren van errata.

Beschrijvende of normatieve koppelvlakken worden geïmplementeerd door systemen en wijzigingen erin kunnen grote consequenties hebben. Beschrijvende en normatieve koppelvlakken mogen door willekeurige partijen op willekeurige momenten worden toegevoegd aan een sectormodel, nadat de expertgroep heeft vastgesteld dat het koppelvlak voldoet aan de regels in dit document. De partij die een koppelvlak toevoegt is verantwoordelijk voor het beheer ervan. Een eenmaal toegevoegd beschrijvend of normatief koppelvlak mag niet meer gewijzigd worden met uitzondering van het doorvoeren van errata. Implementerende partijen mogen met zo'n koppelvlak blijven werken, tenzij de regiegroep of een ander door die partijen overeengekomen open orgaan besluit dat het koppelvlak niet meer geldig is en zo nodig vervangen is door een ander koppelvlak.

3.3 Koppelvlakbeschrijving voor individuele systemen op basis van berichtcatalogi

Een individueel systeem geeft in eigen wsdl's aan welke koppelvlakken het ondersteunt en welke operations uit de verschillende berichtcatalogi van een sectormodel het ondersteunt. Een systeem kan de ondersteunde operations uit de voorbeeld-wsdl's overnemen. De wsdl's voor een specifiek

systeem vallen niet onder het beheermodel van StUF. Wel wordt leveranciers van systemen dringend verzocht de wsdl's van de verschillende versies van hun systemen te publiceren op de StUF-community, zodat gemeenten en andere leveranciers weten welke services een systeem ondersteunt.

In de eigen wsdl kan een systeem niet expliciet aangeven welke gegevens het ondersteunt, omdat in de voorbeeld wsdl's in het part-element binnen het message-element wordt verwezen naar een berichtelement in een berichtencatalogus. Een systeem kan in zijn wsdl alleen aangeven welke operations het ondersteunt. Het is natuurlijk wel wenselijk dat systemen expliciet kunnen aangeven welke gegevens ze ondersteunen. Het beheermodel onderkent voor dit doel de zogenaamde systeemtypebeschrijving. De systeemtypebeschrijving bevat een systeem specifieke versie van een sss_ent_ccc.xsd met ccc de naam van een (deels) ondersteunde berichtencatalogus. Deze schema's krijgen in plaats van 'ent' een aanduiding van het systeem in hun naam. In deze systeem specifieke schema's worden voor de in de systeem-wsdl's gebruikte entiteiten restrictions opgenomen op de complexTypes gedefinieerd in de 'ent'-schema's, als een systeem niet alle elementen van het complexType uit het 'ent'-schema ondersteunt. De complexTypes voor de restrictions krijgen de naam van het complexType op basis waarvan de restriction wordt gemaakt plus een suffix '-xxx' met xxx de aanduiding van het systeem. Voor vraag/antwoord-berichten worden zo nodig ook restrictions opgenomen op de historie-complexTypes om aan te geven in hoeverre een systeem historie ondersteunt.