



Telematica
Instituut

Over het service-ge- oriënteerde gehalte van StUF 3.0



Colofon

Datum : 15 oktober 2007
Versie : 1.0
Verandering : —
Project referentie: SOA-gehalte StUF 3.0
TI referentie : —
Bedrijfsreferentie : —
URL : —
Toegangsrechten : —
Status : Definitief
Redacteur : Paul Oude Luttighuis
Bedrijf : Telematica Instituut
Auteur(s) : Paul Oude Luttighuis

Inhoudsopgave

1	Situatieschets, vraagstelling en aanpak	1
1.1	Situatieschets	1
1.2	Vraagstelling	1
1.2.1	Wat is StUF 3.0?	2
1.2.2	Wat is SOA?	3
1.2.3	Wat is het SOA-gehalte van een standaard?	4
1.3	Aanpak	4
2	StUF als gereedschapskist	5
2.1	De gereedschappen	5
2.2	Toepassingsvormen	6
2.3	Analysekader	7
2.4	Model-gedreven berichtenstandaarden	8
2.5	Functie-specifiek of functie-generiek?	9
3	Analyse van de StUF-gereedschappen	10
3.1	Modellerwijze en syntactische afbeelding	10
3.1.1	Kracht van de modellerwijze	10
3.1.2	Van grafen naar bomen	10
3.1.3	Opnemen van entiteiten in berichtenschema's	11
3.1.4	Tabelentiteitstypen	12
3.2	Functionele semantiek: StUF-kennisgevingen en –bevragingen	12
3.3	Objectidentificatie	13
3.4	Gegevenshistorie	13
3.5	Groepering	13
3.6	Speciale waarden	13
3.7	Logistiek en foutafhandeling	14
3.8	Over hergebruik binnen vrije berichten op functioneel niveau	14
3.9	Overzicht	14
4	Conclusie	16
4.1	Service-oriëntatie als een wig	16
4.2	De wig verdrijven?	16
4.3	En nu?	17
4.4	Ten slotte	17

1 Situatieschets, vraagstelling en aanpak

1.1 Situatieschets¹

StUF is een generieke standaard voor berichtenuitwisseling in vooral het gemeentelijke veld, maar met een overheids-brede ambitie op het gebied van basisgegevens. Het ICTU-programma EGEM biedt onderdak aan het beheer en de doorontwikkeling van StUF.

De ontwikkeling van STUF is in 1996 gestart. De eerste versie van StUF was gebaseerd op het GBA-berichtformaat. Om in te spelen op technologische ontwikkelingen rondom XML en Internet, heeft StUF in 2004/2005 een versie gekregen die gebaseerd is op XML-technologie. Deze versie wordt wel StUF XML genoemd, maar de officiële naam is StUF 2.04.

Recent heeft StUF bovendien moeten reageren op een andere majeure ontwikkeling in architectuurland: die naar de service-georiënteerde architectuur². Daarvoor is StUF 2.04 verrijkt met het zogenaamde vrije bericht, die het toe moet staan om vrijer dan voorheen berichten samen te stellen. De nieuwe versie heet StUF 3.0.

StUF 3.0 heeft momenteel de status van Kandidaat Aanbeveling. Toch is er debat over mate waarin deze specificatie het verdraagt toegepast te worden in SOA-omgevingen. In dat debat is de Gemeente Den Haag uitgesproken kritisch over StUF 3.0. Dit debat is voor EGEM aanleiding geweest om mij te vragen om een onderzoek naar het SOA-gehalte van StUF 3.0.

De notitie in uw handen is het resultaat van dat onderzoek.

1.2 Vraagstelling

De centrale vraag in dit onderzoek is dus wat het SOA-gehalte van StUF 3.0 is. Op drie punten vraagt deze vraagstelling een nadere beschouwing.

- Wat is StUF 3.0?
- Wat is SOA?
- Wat is het SOA-gehalte van een standaard?

¹ Als bron voor deze sectie geldt grotendeels de inhoud van <http://www.egem.nl/kennisbank/informatievoorziening/uitwisseling/stuf> op 5 oktober 2007.

² Verder afgekort tot SOA.

1.2.1 Wat is StUF 3.0?

Hoewel duidelijk is welke documenten de naam StUF 3.0 dragen^{3,4}, is uit het onderzoek en de daarin gevoerde discussies gebleken dat het normatieve karakter van StUF 3.0 niet geheel duidelijk is. Dat wil zeggen dat het, op grond van die documenten, niet voldoende scherp wordt wanneer van een willekeurige uitwisselingspecificatie (een koppelvlak, berichtenschema of, in StUF-termen, een sectormodel) nu wel of niet aan de standaard StUF 3.0 voldoet.

Daarbij speelt ook mee bij dat StUF 3.0 haar eigen denkmodel niet expliciet maakt, in de vorm van bijvoorbeeld een metamodel. In plaats daarvan moet dit denkmodel door de lezer zelf samengesteld worden uit niet altijd even scherpe be- en omschrijvingen en soms ook tussen de regels door.

Voor een belangrijk deel — maar zeker niet geheel — lijkt StUF eerder een berichtenontwikkelmethode, met interpretatievrijheden, dan een toetsbare standaard. Dat is allereerst voor de standaard zelf een beperking. Een standaard voor interoperabiliteit dient namelijk na te streven dat, indien twee volledig onafhankelijk ontwikkelaars ermee aan de slag gaan, zij zo interoperabel mogelijk toepassingen realiseren, op grond van de specificatie alleen. Daarbij moeten zij niet afhankelijk zijn van nader onderling overleg of van bronnen (documenten of mensen) buiten de specificatie.

Meer in het bijzonder echter heeft deze omstandigheid het onderhavige onderzoek zelf bemoeilijkt omdat een uitspraak over het SOA-gehalte van StUF 3.0 alleen te grondvesten is op een scherp beeld van de inhoudelijke keuzes (en dus beperkingen) die StUF 3.0 oplegt aan de berichtenontwikkelaar.

En, wat het denkmodel betreft: de essentie van SOA is op de allereerste plaats het SOA-denkenmodel, waarin de service en het contract de centrale rol spelen. Een analyse van het SOA-gehalte van StUF 3.0 zou dus allereerst een vergelijking tussen de denkmodellen moeten zijn. Zoals gezegd echter is die van StUF niet expliciet.

Toch is er van StUF 3.0 wel degelijk het een en ander duidelijk. In de kern zou StUF 3.0 gekwalificeerd kunnen worden als een domein-generiek berichtensjabloon, gerepresenteerd in XML-technologie. Deze staat gespecificeerd in één van de twee StUF 3.0-documenten: het XML Schema voor StUF 3.0, kortweg het schemadocument.

Een sjabloon is echter niet meer dan een syntactische constructie, hoe nodig ook. Daarom staat in het andere document wat de verschillende onderdelen van dit schema betekenen, hoe zij kunnen en mogen worden toegepast en welke beperkingen daarbij worden gesteld. Dit onderzoek is vooral gericht op het andere document, dat we verder kortweg als het StUF 3.0-document zullen aanduiden. Het SOA-gehalte is immers niet af te lezen aan een schema, maar wel aan het denkmodel eraan en de beperkingen eraan. In het onderzoek is vooral veel discussie geweest met de opdrachtgever over de interpretatie

³ Egem, *Standaard Uitwisseling Formaat voor applicaties — StUF 03.00: Kandidaat Aanbeveling*. Versie 02, 20 juni 2007.

<http://www.egem.nl/kennisbank/informatievoorziening/uitwisseling/stuf/stuf30stufsoa/stuf30egemkandidaataanbeveling>

⁴ Egem, *StUF 3.0: Template schema*.

<http://www.egem.nl/kennisbank/informatievoorziening/uitwisseling/stuf/stuf30stufsoa/stuf3.0berichtdefinities>

van het StUF 3.0-document. Hoewel deze discussie veel heeft verhelderd en de kwaliteit van dit onderzoek heeft verbeterd, kan dit onderzoek niet anders dan zich maximaal baseren op de letter van het StUF 3.0-document zelf. Dat is immers wat de berichtenontwikkelaar ook moet doen.

In overleg met EGEM is besloten tegen StUF 3.0 aan te kijken als een gereedschapskist voor berichtenontwerpers. Het onderzoek zelf heeft deze gereedschapskist ontleend in een aantal gereedschappen en een aantal toepassingsvormen van de gereedschapskist onderkend. De analyse van het SOA-gehalte van StUF 3.0 vindt plaats op de kruispunten van gereedschappen en toepassingsvormen. Dat wil zeggen, het onderzoek analyseert of het gebruik van één van de gereedschappen in één van toepassingsvormen op bezwaren of beperkingen stuit in een SOA-omgeving.

1.2.2 Wat is SOA?

Er bestaat geen eenduidige definitie van SOA, laat staan dat daarvoor een toetskader voorhanden is. De aard van de vraagstelling ontstijgt gelukkigerwijze echter het niveau van het technologische detail, zodat het onderzoek zich dat ook kan veroorloven. Dit onderzoek gaat daarom vooral over de geest van service-oriëntatie en niet over de exacte letter. Daarvoor heeft het onderzoek zich bediend van een bijlage van de NORA⁵. De NORA kiest voor service-oriëntatie en werkt deze keuze uit in het zogenaamde “dossier SGA”, dat recent is gepubliceerd⁶.

Een keuze voor een specifieke definitie van SOA herbergt natuurlijk het gevaar dat de uitkomst van dit onderzoek alleen geldig is bij deze specifieke opvatting van SOA. Dat gevaar is in dit geval echter miniem, omdat bij het onderzoek gebleken is dat het zich kon beperken tot het gebruik van kerneigenschappen van SOA, die als een rode draad door vrijwel elke SOA-definitie heenlopen. Nergens hoefde afgedaald te worden tot een detailniveau waarop SOA-definities verschillen.

Zo'n kerneigenschap van SOA is dat het in het ontwerp de service en het bijbehorende contract centraal stelt. Daarbij zijn service en contract zakelijke grootheden, geen technische. Het ontwerp van software, van bedrijfsprocessen en ook van berichten is geheel dienend aan de service waarvoor zij worden ingericht. Op dat terrein onderscheidt het zich van werkstroom-georiënteerde of gegevens-georiënteerde benaderingen.

Historisch kent StUF een gegevens-georiënteerde benadering. Dat is verklaarbaar uit haar oorspronkelijke positionering als gegevens-synchronisatiestandaard. Dit onderzoek zal laten zien dat StUF 3.0 nog steeds dit gegevens-georiënteerde denkmodel ademt. Hoewel de conclusie over het SOA-gehalte van StUF genunanceerd zal zijn, kan zonder veel reserve gezegd worden dat StUF 3.0 vanuit SOA-perspectief wezenlijke beperkingen kent. De oorzaak hiervoor ligt niet in details, het ligt in de paradigmatische afstand tussen het service-denken en het gegevens-denken.

⁵ Nederlandse Overheids Referentie-Architectuur,
<http://www.elektronischeoverheid.nl/atlas/referentiearchitectuur/>

⁶ <http://www.elektronischeoverheid.nl/data/files/architectuur/ServiceGerichteArchitectuur.pdf>

1.2.3 Wat is het SOA-gehalte van een standaard?

Men zou kunnen zeggen dat de SOA-ontwikkelingen van de laatste jaren een voor StUF nieuwe, externe kracht zijn, die de StUF-community plaatst voor de vraag hoe men hier toe zou moeten verhouden. Uiteindelijk zal daarbij de tactische vraag aan de orde zijn of de StUF-community zich hierbij defensief of offensief opstelt.

Bij een *defensieve* keus zal de inhoudelijke vraag naar het SOA-gehalte van StUF 3.0 zich vertalen naar de vraag of StUF 3.0 staande kan blijven onder de SOA-ontwikkelingen, door StUF 3.0 — of onderdelen daarvan — te koppelen zijn aan het SOA-denken. Bij een *offensieve* keus zal de inhoudelijke vraag naar het SOA-gehalte van StUF 3.0 zich vertalen naar de vraag of StUF 3.0 de SOA-ontwikkelingen kan omarmen en dus of het StUF 3.0-denkmodel kan convergeren naar een SOA-denkmodel.

Het zij duidelijk dat de offensieve invulling van de onderzoeksvraag strenger zal zijn voor StUF 3.0 dan de defensieve. Toch is het van belang ook aan de offensieve optie aandacht te besteden. Mochten de SOA-ontwikkelingen zich doorzetten en mocht de internationale standaardisatiewereld daarvoor een dekkende familie van standaarden gaan afleveren, zal een nationale variant het daartegen gaan afleggen. In dat geval geldt: *You can't beat them, so join hem*. Bovendien kent een defensieve benadering het gevaar dat StUF gaandeweg een rafelige standaard wordt, waarin uitzonderingen en addenda ervoor moeten zorgen dat de aanhaking bij SOA in stand blijft. Eigenlijk is het vrije bericht daarvan al een voorbeeld. In de offensieve variant zou StUF “native SOA” worden en dus korter, duidelijker en simpeler.

1.3 Aanpak

Het onderzoek is uitgevoerd in de volgende vier stappen:

1. **Eerste analyse van het materiaal.** Daarbij behoort natuurlijk de StUF 3.0-document zelf⁷, maar ook de schriftelijke neerslag van een discussie tussen de Gemeente Den Haag en EGEM over het thema van het onderzoek.
2. **Interviews.** Op basis van deze eerste analyse zijn gesprekken gevoerd met de Gemeente Den Haag en EGEM. Het gesprek met de Gemeente Den Haag was op 15 augustus 2007. De Gemeente werd daarbij vertegenwoordigd door Chris Batist van de Gemeente zelf en Cathy Dingemanse en Melvin Lee, beiden van Comites B.V. Daarop volgde op 20 augustus 2007 een gesprek met EGEM, vertegenwoordigd door Henri Korver van EGEM zelf en Maarten van den Broek van messagedesign.nl.
3. **Analyse.** In de derde stap is de kernanalyse gemaakt. Op basis van de eerste resultaten daarvan heeft op 12 september 2007 een tweede gesprek plaatsgevonden met Henri Korver en Maarten van den Broek. Daarin is afgesproken StUF in het onderzoek te beschouwen als een gereedschapskist, zoals hierboven al is besproken.
4. **Verslaglegging.** In deze stap is op 5 oktober een conceptrapportage opgeleverd. Hierop is door Egem nog eenmaal per e-mail gereageerd. Deze reactie is gebruikt voor het definitieve rapport, dat voor u ligt.

7

<http://www.egem.nl/kennisbank/informatievoorziening/uitwisseling/stuf/stuf30stufsoa/stuf30egemkandidaataanbeveling>

2 StUF als gereedheidskist

2.1 De gereedheidskisten

StUF kan worden gezien als een gereedheidskist voor berichtenontwerpers. In deze gereedheidskist heeft het onderzoek de volgende gereedheidskisten geïdentificeerd:

- (Eisen aan) een **modellerwijze**⁸. StUF neemt als uitgangspunt dat berichten worden ontworpen op basis van een technologie-onafhankelijk gegevensmodel. Als modellerwijze vereist StUF het gebruik van concepten uit (een variant op) zogenaamde ERD's⁹.
- Een **syntactische afbeelding**¹⁰. Een berichtenontwerper vertaalt (onderdelen van) ERD's naar XML-syntax, naar een berichtenschema. StUF bevat bepalingen over hoe deze vertaling plaats moet vinden.
- Een tweetal typen standaard-**functies**¹¹, namelijk kennisgevingen en vraag-antwoord-functies. Voor deze functies wordt de semantiek vastgelegd en worden eisen gesteld aan de berichtendialoog en de berichtenschema's.
- Mechanismen voor **objectidentificatie**¹². Communicerende partijen moeten onderling overeenstemming kunnen hebben over het precieze object in de werkelijke wereld waaraan in hun communicatie wordt gerefereerd.
- Mechanismen voor **gegevenshistorie**¹³. Communicerende partijen moeten ook kunnen communiceren over situaties in het verleden, over "oude gegevens".
- Mechanismen op gegevens te **groeperen**¹⁴, voor uiteenlopende doeleinden.
- **Speciale gegevenswaarden**¹⁵, voor het geval communicerende partijen de waarde van een zeker gegeven niet kennen, niet ondersteunen of niet mogen kennen.
- Mechanismen voor **berichtenlogistiek**¹⁶, sturing en foutafhandeling.

In beginsel staat het de berichtenontwerper vrij om deze gereedheidskisten al dan niet toe te passen. Geheel onafhankelijk zijn de gereedheidskisten echter niet: zonder het gebruik van de geboden modellerwijze is het niet mogelijk gebruik te maken van de syntactische afbeelding, de geboden standaard-functies en de mechanismen voor object-identificatie.

⁸ Secties 3.1.1 en 3.1.2 van het StUF 3.0-document.

⁹ Entity-Relationship Diagram

¹⁰ Sectie 3.1.2 en 3.2 van het StUF 3.0-document.

¹¹ Hoofdstukken 5 en 6 van het StUF 3.0-document.

¹² Sectie 3.1.3 van het StUF 3.0-document.

¹³ Sectie 3.1.5 van het StUF 3.0-document.

¹⁴ Sectie 3.1.4 van het StUF 3.0-document.

¹⁵ Sectie 3.3 van het StUF 3.0-document.

¹⁶ Hoofdstuk 4 van het StUF 3.0-document.

2.2 Toepassingsvormen

In een SOA worden berichten uitgewisseld in het kader van een dienstverleningsrelatie tussen de communicerende partijen. Berichten worden dus ontworpen voor zo'n specifieke dienst¹⁷. De berichtenontwerper die daarbij van StUF gebruik wil maken, kan daarbij kiezen uit:

- het gebruik van *StUF-kennisgevingen*, indien het om een kennisgevingsdienst gaat
- het gebruik van *StUF-bevragingen*, indien het om een bevragsdienst gaat
- het gebruik van de zogenaamde *vrije berichten* van StUF.

In dat laatste geval is de berichtenontwerper vrijwel geheel vrij in het inrichten van het bericht, maar ondervindt daarbij dus ook geen ondersteuning van de StUF-specificatie. Daarom wordt de mogelijkheid geboden om op onderdelen hergebruik te maken van wat StUF biedt. Dat kan op twee niveaus:

- op *functioneel niveau*. In dat geval maakt de berichtontwerper, binnen een vrij bericht, gebruik van de semantiek van StUF-kennisgevingen en/of –bevragingen om daarmee een deel van de dienst af te dekken.
- op *entiteitenniveau*. In dat geval maakt de berichtenontwerper, binnen een vrij bericht, gebruik van de mechanismen voor syntactische afbeelding van objecten, wanneer een dergelijk object een rol speelt in de dienst.

Samenvattend identificeert dit onderzoek vijf toepassingsvormen van de StUF-specificatie:

- het gebruik van *StUF-kennisgevingen*
- het gebruik van *StUF-bevragingen*
- het gebruik van *vrije berichten met hergebruik op functioneel niveau*
- het gebruik van *vrije berichten met hergebruik op entiteitenniveau*
- het zo vrij mogelijk gebruik van *vrije berichten*, dat wil zeggen, *zonder hergebruik* op functioneel of entiteitenniveau.

¹⁷ Dit is geen pleidooi voor top-down ontwerp, maar het principieel centraal stellen van de business service in het ontwerp van een koppelvlak, zoals SOA dat doet. Elk bericht hoort bij een service, niet bij een entiteit. Natuurlijk kunnen berichten ook over services heen op onderdelen hetzelfde zijn of eenzelfde patroon gebruiken. Dus, verschillende services hergebruiken onderling niet bij voorbaat hele berichten, maar alleen berichtenonderdelen of berichtenpatronen.

2.3 Analysekader

Deze beschouwing levert een analysekader in de vorm van een matrix, als hieronder. Voor elk van de cellen in deze matrix heeft het onderzoek geanalyseerd of er bezwaren kleven aan gebruik van het betreffende StUF-gereedschap in de betreffende toepassingsvorm, in een SOA-omgeving.

Daarbij zijn niet alle cellen van toepassing. In het gebruik van het vrije bericht in zijn meest vrije vorm worden vrijwel alle gereedschappen achterwege gelaten. En bij enkel hergebruik op entiteitenniveau is geen sprake van het gebruik van de semantiek van StUF-kennisgevingen en –bevragingen.

Voor gebruik van het vrije bericht met hergebruik van StUF-kennisgevingen of –bevragingen kent de matrix geen aparte kolom. Immers, mogelijke bezwaren tegen het hergebruik van StUF-kennisgevingen of –bevragingen in vrije berichten in SOA-omgevingen zijn dezelfde als die tegen het gebruik van StUF-kennisgevingen en –bevragingen op zichzelf.

StUF-toepassingsvormen	vrije bericht			
	kennisgevingsberichten	vraag/antwoordberichten	hergebruik van entiteiten	hergebruik zonder entiteiten
metamodel				
vertaling naar schema				
functionele semantiek				
objectidentificatie				
gegevenshistorie				
groepering				
speciale waarden				
logistiek en foutafhandeling				

2.4 Model-gedreven berichtenstandaarden

StUF 3.0 hanteert een model-gedreven benadering voor het ontwerpen van berichten. Dat betekent dat berichtenschema's niet direct — in bijvoorbeeld XML — worden ontworpen, maar worden gebaseerd op een model, waarin betekenis wordt uitgedrukt van de begrippen in het domein waarover het berichtenverkeer gaat (en hun relaties).

Domeinmodellen drukken dus betekenis uit van het door de betrokken partijen gedeelde interessedomein of “gespreksonderwerp”. Een domeinmodel is liefst geheel technologie- en syntax-onafhankelijk. Het is een “conceptueel” model. In dit advies zullen we model-gedrevenheid op zichzelf niet ter discussie stellen. Of een berichtenstandaard nu service-georiënteerd is of niet, model-gedrevenheid wordt gezien als een grote kwaliteit die de toepasbaarheid, begrijpelijkheid, onderhoudbaarheid en kwaliteit van berichtenstandaarden ten goede komt.

In model-gedreven berichtenontwikkeling worden vervolgens bepalingen en/of richtlijnen gebruikt die zeggen op welke manier de begrippen uit het domeinmodel terugkomen in berichtenschema's. Daarbij gaat het dus om een vertaling van het domeinmodel naar een syntactisch uitdrukingsmiddel. In moderne berichtenstandaarden is dat typisch XML, maar er is op voorhand geen reden daarbij andere syntactische uitdrukingsmiddelen uit te sluiten. Sterker nog, als het domeinmodel syntax-onafhankelijk is, kan vanuit hetzelfde domeinmodel desgewenst naar meerdere verschillende syntaxen worden vertaald, zoals XML én EDI én CSV.

Welke begrippen in welk berichtenschema terechtkomen (als veld of element) is principieel afhankelijk van de functie van het bericht, dat wil zeggen, van de bedoeling die de betrokken partijen hebben met de uitwisseling ervan. Daar komt bij dat voor het uitvoeren van die bedoeling vaak één enkele berichtenuitwisseling niet volstaat. Regelmatig moeten er meerdere berichten in een gestructureerde dialoog worden uitgewisseld om de bedoelde functie tussen de partijen af te handelen. Een overbekende dialoog is bijvoorbeeld de vraag-antwoord-dialoog.

Vuistregels voor model-gedreven berichtenontwikkeling zijn:

- Alle inhoudelijke velden in alle berichten zijn gerelateerd aan het domeinmodel. Anders zijn ze betekenisloos.
- Een berichtenschema wordt ontworpen vanuit een specifieke (soort) functie en in de context van de berichtendialoog die daarbij hoort.
- Domeinmodellen zijn functie- en syntax-onafhankelijk.
- Functies worden gedefinieerd in termen van het domeinmodel.
- Berichtenlogistieke keuzes (en dus berichtenenveloppen) zijn domein- en functie-onafhankelijk.

Model-gedrevenheid kan nog op vele manieren plaatsvinden. Eenieder kan zijn eigen uitdrukingsmiddelen kiezen, voor domeinmodellen, voor de syntactische taal en, vooral, voor de vertaling daartussen. Om interoperabiliteit te bevorderen worden daarom in model-gedreven berichtenstandaarden vaak tenminste die uitdrukingsmiddelen (voor

domeinmodellen, voor functies, voor dialogen en voor berichtenschema's) en de vertaling daartussen gestandaardiseerd. Tot op zekere hoogte doet StUF 3.0 dat ook.

2.5 Functie-specifiek of functie-generiek?

Maar met de talen alleen is nog lang geen interoperabiliteit gerealiseerd. Uiteindelijk moeten de betrokken partijen ook de specifieke berichten, in het kader van specifieke functies en dialogen, kunnen begrijpen en passen kunnen verwerken. Pas dan is er interoperabiliteit. Op dit punt ontstaat er een belangrijke scheiding tussen twee typen (model-gedreven) berichtenontwikkeling:

- functie-specifieke berichtenontwikkeling
- functie-generieke berichtenontwikkeling

De functie-specifieke benadering is integraal toegesneden op een specifiek type functie, bijvoorbeeld bevestigingsfuncties of indieningsfuncties of welke type functies dan ook. Keuzes voor uitdrukkingsmiddelen (talen) en vertalingen daartussen worden dan specifiek toegesneden op dat type functie. Standaarden van het tweede soort hebben dat niet. Zij kennen een expliciet en generiek functiebegrip en geven aan hoe, gegeven een domeinmodel en gegeven een willekeurige functie, de dialoog en de berichten daarbinnen vorm krijgen.

Dit onderscheid speelt een hoofdrol in de discussie over het SOA-gehalte van StUF. Een service-georiënteerde berichtenstandaard is functie-generiek. Dat is belangrijk, omdat daarmee de functie van het berichtenverkeer (de service) onder expliciete controle wordt gebracht van de business en niet bij voorbaat beperkt wordt door de berichtenstandaard zelf. Die functie wordt tussen de betrokken partijen vastgelegd in een op hun situatie toegesneden contract.

Uiteindelijk kan ook in een functie-generieke (service-georiënteerde) benadering worden gewerkt aan de standaardisatie of harmonisatie van specifieke functies en de bijbehorende dialogen en berichten. Maar, dat is **bovenop** een functie-generieke basis. Die basis bestaat uit functie-generieke keuzes voor de uitdrukkingsmiddelen (talen voor domeinmodellen, functies, dialogen en berichtenschema's). Zo wordt de business de baas en niet de standaard.

3 Analyse van de StUF-gereedschappen

Dit hoofdstuk analyseert de verschillende StUF-gereedschappen per stuk op mogelijke bezwaren bij toepassing in een SOA-omgeving.

3.1 Modellerwijze en syntactische afbeelding

3.1.1 Kracht van de modellerwijze

Om in de geest van StUF 3.0 vanuit een domeinmodel tot berichten te komen dient het domeinmodel noties van entiteit, relatie en attribuut te kennen, zoals we die uit ERD's kennen. StUF 3.0 kan niet omgaan met noties van generalisatie en specialisatie (zoals die in bijvoorbeeld UML voorkomen) en ook niet met constraints. In een SOA is het bovendien aan te bevelen om in het domeinmodel ook met noties van bestaansafhankelijkheid tussen objecten te werken, maar vooral ook met gebeurtenissen en betrokkenheid van objecten bij die gebeurtenissen.

De keuze voor een ERD-variant als modellerwijze verraadt de gegevens-oriëntatie van StUF 3.0. ERD's zijn bedoeld voor database-ontwerpomgevingen en zijn te arm voor generieke service-ontwerpomgevingen. Toch vloeit hieruit geen SOA-bezwaar voort, als de onderzoeksvraag in defensieve zin wordt opgevat. Dat wil zeggen, het gebruik van ERD's als modellerwijze maakt de uiteindelijke koppelvlakproducten (sectormodellen) nog niet ongeschikt voor een SOA. In de offensieve opvatting echter is er wel sprake van een bezwaar: bij omarming van het SOA-denkmiddel zou ook StUF haar modellerwijze moeten verrijken met bovengenoemde concepten.

3.1.2 Van grafen naar bomen

Een belangrijke uitdaging bij het maken van een XML-berichtschemata is het omvormen van de graaf-structuur van een domeinmodel naar de boomstructuur van een XML-schemata. Daarvoor moet zo hier en daar hiërarchie worden aangebracht in het schema, waar deze in het domeinmodel niet aanwezig was. Belangrijk is dat, in een SOA, deze hiërarchie alleen mag worden aangebracht in het licht van een zeker (soort) (business)functie (service). Er mag niet a priori hiërarchie worden aangebracht in het domeinmodel om zo te anticiperen op het berichtschemata. StUF 3.0 doet dit echter wel.

Het duidelijkst is dit bij relaties. Relaties in StUF 3.0 hebben namelijk een richting en zijn daardoor bovendien binair¹⁸. In generieke ERD's zoals we ze kennen uit de theorie¹⁹ kunnen relaties ook tussen drie of meer objecten worden gelegd. Echter, als relaties altijd binair en gericht zijn, dient zich een simpele vertaling naar hiërarchische schemata's aan, waarin de van-naar-richting zich één-op-één laat vertalen in een ouder-kind-relatie in het schema. Waarschijnlijk legt StUF hierom deze beperking op.

¹⁸ Zie pagina 14 van het StUF-document: “... Een relatie-entiteittype definieert een verband tussen het linker object en het rechter object uitgaande van het linker object (de richting van de relatie is dus relevant). ...”

¹⁹ Peter Pin-Shan Chen, *The Entity-Relationship Model — Toward a Unified View of Data*, ACM Transactions on Database Systems, Vol 1, No.1, pp. 9–36.

StUF kan deze beperking alleen aanbrengen als het bij voorbaat een beperking legt op de aard van de (business)functies die worden ondersteund. Daarmee is StUF 3.0 op dit punt niet (business)functie-generiek. Om dat wel te zijn, zou het moeten aangeven hoe ook ongerichte en zelfs meervoudige relaties zich, in het licht van een zeker type (business)-functie, vertalen naar een hiërarchisch berichtenschema. Onder verwijzing naar figuur 1 in het StUF 3.0-document: het moet met hetzelfde domeinmodel mogelijk zijn om, voor (business)functie X, een bericht te definiëren waarin een kind bij de ouder is ondergebracht en, voor (business)functie Y, een bericht te definiëren waarin een ouder bij het kind is ondergebracht. Beide gevallen zijn syntactische voorkomens van dezelfde ouder-kind-relatie in het semantische model, die ongericht is.

3.1.3 Opnemen van entiteiten in berichtenschema's

StUF 3.0 stelt, op het moment dat de ERD vast staat, beperkingen op aan de berichtenschema's die daarop kunnen worden gebaseerd. Daarbij associeert het allereerst berichten met entiteitstypen uit het ERD. In een SOA worden berichten geassocieerd aan services, niet aan entiteitstypen.

Voorbeelden van de beperkingen die StUF 3.0 oplegt zijn:

- Voor relatie-entiteitstypen worden geen berichten gedefinieerd²⁰. Toch zijn er zeker services denkbaar die vanuit de relatie redeneren. Denk bijvoorbeeld aan een service die gegevens vraagt over alle recent gesloten huwelijken. StUF 3.0 vereist daarvoor dat het bericht ook gegevens bevatten over de entiteiten die gerelateerd zijn. Dit is echter in veel services overbodig.
- Voor attributen van een entiteit geldt iets dergelijks²¹.

Samengevat kan gezegd worden dat, bij gebruik van model-gedrevenheid vanuit een ERD, relaties en attributen alleen in StUF 3.0-berichten kunnen voorkomen in de context van een “bovenliggende” entiteit. Zoals al gezegd echter is — in een functie-generieke standaard — bij dit soort keuzes de service leidend en niet de berichtenstandaard. Bijvoorbeeld, in een service waarin alleen de oogkleur van een persoon gevraagd wordt, moet het niet verplicht zijn een gehele persoons-entiteit terug te geven, maar moet het mogelijk zijn alleen de oogkleur te retourneren.

Waarschijnlijk doet StUF 3.0 dit om een zo gestandaardiseerd mogelijk berichtensjabloon te kunnen aanbieden, dat wil zeggen, om de berichten in sectormodellen zoveel mogelijk op elkaar te laten lijken. Vanuit de specifieke business service gezien echter, levert dit in veel gevallen “te zware” berichten op, met niet ter zake doende velden.

Dit bezwaar is reeds aan de orde in de defensieve opvatting van de vraagstelling en dus ook in de offensieve. Hoewel het denkbaar is dat StUF 3.0 op deze punten wordt aangepast, is hiervan in het huidige StUF 3.0 geen sprake.

Zie ook sectie 3.6.

²⁰ Dit is een letterlijk citaat uit pagina 14 van het StUF 3.0-document.

²¹ Zie sectie 7.2.2 van het StUF 3.0-dokument.

3.1.4 Tabelentiteitstypen

Tabelentiteitstypen komen oorspronkelijk niet voor in ERD's. Zij zijn waarschijnlijk vooral in StUF opgenomen om te kunnen omgaan met types die te allen tijde een eindig aantal instanties kennen, waarbij echter die lijst van instanties niet op modelleertijd bekend is. Dit is echter op semantisch niveau (in het domeinmodel dus) van geen belang. Dat databases met dergelijke typen omgaan door een tabel met op enig moment geldige instanties bij te houden, is prima, maar een implementatie-issue.

Dat tabelentiteiten minder reëel zouden zijn dan fundamentele entiteiten, zoals het StUF 3.0-document stelt, is dus niet aan de orde. Er bestaan relevante, maar zeer abstracte concepten die zich niet in een tabel laten bijhouden; omgekeerd bestaan er behoorlijk concrete concepten die zich wel in een tabel laten bijhouden, zoals de planeten in ons zonnestelsel. Tabellen zijn een issue in database-omgevingen, geen semantisch issue in een SOA.

Naar het zich laat aanzien echter, kan het gebruik van tabelentiteiten geheel worden voorkomen. Elk tabelentiteitstype laat zich ook als fundamenteel entiteitstype modelleren. In de defensieve opvatting van de onderzoeksvraag ligt hier dus geen SOA-bezwaar. In de offensieve echter wel: in dat geval zouden tabelentiteiten niet in het metamodel moeten worden opgenomen.

3.2 Functionele semantiek: StUF-kennisgevingen en –bevragingen

Met name aan StUF-kennisgevingen kleven belangrijke bezwaren voor toepassing in een SOA-omgeving.

Zo wordt bij een kennisgeving van de zendende partij geëist dat hij de betreffende gegevens ook zelf registreert, terwijl dat voor de ontvangende partij veelal onbelangrijk is. Scheiding tussen interface en implementatie is een kernwaarde van service-georiënteerde architecturen.

Het relevant worden van een object voor de zendende partij geldt als aanleiding voor kennisgevingen, terwijl dat in het algemeen niet belangrijk is voor de ontvangende partij. Die ontvangt liefst informatie in reactie op een voor hem belangrijke gebeurtenis.

Ten slotte wordt met de zogenaamde “indicatorOvername” geregeld of de ontvangende partij de verzonden gegevens dient over te nemen in de eigen registratie. Eén verplichte overname is echter geen kennisgeving meer, maar impliceert een omdraaiing van de dienstrelatie. Waar bij “informatieve” StUF-kennisgevingen de verzender de dienstleverancier is en de ontvanger de afnemer (typisch in publish-subscribe-verkeer), is bij verplichte overname de verzender ineens de afnemer van een registratiedienst van de ontvanger geworden.

Naar de letter van het StUF 3.0-document kleeft er ook aan StUF-bevragingen een SOA-bezwaar. Voor vraag/antwoord-berichten wordt namelijk gesteld dat de bron van het antwoord de database van het bevraagde systeem is. Voor de service zelf is dat niet belangrijk: de bevraagde partij mag het antwoord ook van elders (bij een onderaannemer) halen. Anders gezegd, als StUF 3.0-bevragingen zouden worden gesteld aan een portaal, zou StUF eisen dat het portaal de gegevens per se uit een eigen database haalt. Deze bepaling is echter eenvoudig aan te passen en levert dus geen fundamenteel SOA-bezwaar op.

De SOA-bezwaren van StUF 3.0-kennisgevingen, echter, spelen reeds in de defensieve opvatting van de onderzoeksvraag en daarmee ook in de offensieve. Dit zal overigens geen verrassing zijn, omdat het vrije bericht niet voor niets in StUF 3.0 is opgenomen.

3.3 Objectidentificatie

Hieraan kleven geen specifieke bezwaren vanuit SOA-perspectief.

3.4 Gegevenshistorie

StUF 3.0 biedt functionaliteit oor het omgaan met historische gegevens, door geldigheidstermijnen bij een waarde op te nemen. Met kennisgevingsberichten kan de geldigheidstermijn van gegevens direct worden gemanipuleerd. Daarbij blijft onbetwist dat statusinformatie die in databases wordt ontsloten feitelijk afgeleide informatie is. De bron van historische ontwikkeling van de gegevens ligt niet in de statusinformatie maar in opgetreden gebeurtenissen. De statusinformatie is daarvan afgeleid om bevestigingen snel te kunnen afhandelen.

Aanpassingen van gegevenshistorie dient in een SOA-context daarom plaats te vinden door aanpassingen van de historische gebeurtenissen die eraan ten grondslag liggen. Ook binnen de overheid zijn registraties de afgeleide van een bronregister, waar door middel van bijvoorbeeld aktes een gebeurtenissenoverzicht wordt bijgehouden.

Als de geldigheidstermijn aangepast moet worden, is het allereerst zaak niet direct de oude waarde van de geldigheidstermijn te overschrijven, maar de tijdsaanduiding aan te passen van de historische gebeurtenissen die de waardeveranderingen veroorzaakten. Als een historische waarde aangepast moet worden, moet de inhoudelijke gegevens van de historische gebeurtenis worden aangepast, die tot deze waarde heeft geleid.

Dit SOA-bezwaar treedt allen op bij kennisgevingen en eventuele services waarin historie aangepast moet worden, niet bij bevestigingen. In dat geval is dit een SOA-bezwaar in zowel de defensieve als offensieve opvatting van de onderzoeksvraag.

3.5 Groepering

Hieraan kleven geen specifieke bezwaren vanuit SOA-perspectief.

3.6 Speciale waarden

StUF definieert een aantal speciale waarden voor velden in een bericht.

Zo is er een waarde “nietGeautoriseerd”, voor als de ontvanger het betreffende gegeven niet mag kennen. Echter, in een SOA is een partij al dan niet geautoriseerd om een dienst af te nemen. Berichten worden geheel ontworpen in het kader van de dienst. Een veld opnemen waartoe een zekere afnemer niet geautoriseerd zou zijn, is dus geheel overbodig. Dat komt omdat berichten in een SOA maatwerk zijn voor het contract. StUF probeert teveel om berichtenschema's te standaardiseren voor alle diensten en contracten tegelijk. Dat zorgt ervoor dat deze speciale waarde nodig is. Hetzelfde geldt voor de waarde “nietOndersteund”.

Dit SOA-bezwaar is dus een verschijningsvorm van het SOA-bezwaar dat al in sectie 3.1.3 werd genoemd. De lezer zij daarom daarnaar verwezen.

3.7 Logistiek en foutafhandeling

Hieraan kleven geen specifieke bezwaren vanuit SOA-perspectief.

3.8 Over hergebruik binnen vrije berichten op functioneel niveau

De SOA-bezwaren die optreden als de berichtenontwerper rechtstreeks kiest voor StUF-kennisgevingen of –bevragingen, treden ook op als de berichtenontwerper de semantiek hiervan hergebruikt binnen vrije berichten (hergebruik op functioneel niveau). Echter, aan dit hergebruik kleeft op zichzelf nog een additioneel bezwaar.

Immers, als de berichtenontwerper zijn dienst functioneel moet decomponeren om hergebruik te kunnen maken van een StUF-kennisgeving of –bevraging, is hij bezig met een implementatie-aspect van zijn dienst, terwijl hij alleen met het interface bezig zou moeten zijn.

Dit bezwaar is aan de orde in de defensieve en dus ook de offensieve opvatting van de onderzoeksvraag.

3.9 Overzicht

Deze analyses leiden tot de volgende vulling van de tabel. De rode kleur is gebruikt waar een belangrijk SOA-bezwaar kleeft aan de toepassing van het betreffende StUF-gereedschap in de betreffende toepassingsvorm. Bij groen is een dergelijk bezwaar niet gevonden. De gele kleur wordt gebruikt als het betreffende gereedschap in de betreffende toepassingsvorm niet van toepassing is.

StUF-toepassingsvormen	vrije bericht			
StUF-gereedschappen	kennisgevings-berichten	vraag/antwoord-berichten	hergebruik van entiteiten	Zonder hergebruik
<i>metamodel</i>	rood	rood	rood	geel
<i>vertaling naar schema</i>	rood	rood	rood	geel
<i>functionele semantiek</i>	rood	groen	geel	geel
<i>objectidentificatie</i>	groen	groen	groen	geel
<i>gegevenshistorie</i>	rood	groen	rood	geel
<i>groepering</i>	groen	groen	groen	geel
<i>speciale waarden</i>	rood	rood	rood	geel
<i>logistiek en foutafhandeling</i>	groen	groen	groen	groen

In alle rode cellen geldt dat er niet alleen SOA-bezwaren zijn, maar ook specifiek in de defensieve opvatting van de onderzoeksvraag. Wel is in de defensieve opvatting het aantal SOA-bezwaren minder dan in de offensieve opvatting, in de bovenste twee rijen. Ook zijn de SOA-bezwaren in de “speciale waarden”-rij feitelijk dezelfde als één van de bezwaren in de tweede rij.

4 Conclusie

4.1 Service-oriëntatie als een wig

Een blik op de overzichtstabel in het vorige hoofdstuk leert dat er belangrijke bezwaren kleven aan het rechtstreekse gebruik van StUF-kennisgevingen en –bevragingen in SOA-omgevingen. Deze bezwaren treden op in het fundament van StUF: de modelleerwijze en de vertaling naar berichtenschema's. Deze bezwaren worden meegeorven naar het vrije bericht, indien daarin hergebruik gemaakt wordt van deze kennisgevingen en/of bevragingen. Zij spelen bovendien al in de defensieve opvatting van de onderzoeksvraag.

Maar ook bij hergebruik op entiteitsniveau worden dergelijke kernbezwaren vanuit SOA-perspectief georven. Daarom is de kans groot dat, indien StUF 3.0 in SOA-omgevingen zou worden toegepast, het vrije bericht hiervoor alleen in zijn meest vrije vorm zou worden toegepast.

Dus, service-oriëntatie drijft een wig tussen het vrije bericht en de rest van de specificatie. De reden hiervoor is dat SOA-bezwaren verweven zijn in StUF 3.0.

Moet een service-georiënteerde berichtontwerper dan gewoon maar StUF 3.0 grotendeels negeren en geheel met vrije berichten aan de slag gaan? Daarvoor lijkt geen aanleiding, want vrije berichten, in hun meest vrije vorm, bieden de ontwerper zeer beperkte ondersteuning. De enige ondersteuning die de berichtenontwerper nog wel krijgt in dat geval is die van de onderste laag: de berichtenlogistiek. Hiervoor zijn echter internationale open standaarden voorhanden en kan beter geen nationaal alternatief worden gebruikt.

Toch vraagt de interoperabiliteit in service-georiënteerde omgevingen ook om afspraken over domeinmodellen, functies, dialogen en berichtenschema's, op een model-gedreven wijze. StUF 3.0 kan dat nu nog niet bieden.

4.2 De wig verdrijven?

Een belangrijke vraag is welke veranderingen nodig zouden zijn om de wigwerking op zijn minst gedeeltelijk teniet te doen. Pogingen daartoe zouden zich allereerst moeten richten op het hergebruik van entiteiten in vrije berichten. Voorgaande analyse laat zien dat, in de defensieve opvatting van de onderzoeksvraag, daartoe ten minst twee wijzigingen moeten worden aangebracht.

- In de domeinmodellen moeten relaties ongericht zijn en liefst ook meer dan twee entiteiten kunnen relateren.
- In de vertaling naar berichtenschema's moet de eis vervallen dat in berichten relatie-gegevens en attribuut-gegevens alleen mogen voorkomen in de hiërarchische context van entiteiten.

Hoewel dit onderzoek niet heeft gekeken naar de impact van dergelijke veranderingen, lijken zij zonder al te veel impact te kunnen worden doorgevoerd.

4.3 En nu?

Het was geen deel van de onderzoeksvraag om ook te adviseren over de toekomst van StUF. Toch wil dit rapport enkele scenario's schetsen voor de toekomst van StUF, daartoe geïnspireerd door het onderzoek.

Voor een belangrijk deel is de actualiteit van de onderzoeksvraag het gevolg van de brede ambitie van StUF: een breed toepasbare berichtenstandaard, ook voor service-georiënteerde omgevingen. Mocht de StUF-community zich kunnen vinden in de analyse in dit rapport, zou één van de reacties kunnen zijn om de ambitie van StUF in die zin in te perken, dat het alleen bedoeld is voor gegevenssynchronisatie-toepassingen op de gegevenslaag. Dit past veel beter bij het paradigma en de geest waarin het is ontworpen. Bij een dergelijke keuze zou StUF echter een marginalere rol gaan spelen in een wereld die zich richting service-oriëntatie ontwikkelt.

Een tweede scenario zou kunnen zijn om door te gaan op de huidige weg. De analyse in dit rapport onderbouwt deze keuze niet, noch in de defensieve, noch in de offensieve opvatting van de onderzoeksvraag.

Een derde scenario gaat uit van een defensieve houding van StUF tegenover SOA. Aanpassingen en toevoegingen aan StUF 3.0 zouden de wigwerking kunnen beperken en in elk geval het hergebruik van entiteiten in vrije berichten mogelijk maken. Dat zou een belangrijke stap zijn, want daarmee blijft de model-gedrevenheid van StUF in belangrijke mate op peil. Voor de nadelen hiervan zij verwezen naar sectie 1.2.3.

Een vierde scenario, ten slotte, gaat uit van de offensieve variant en betreft een fundamentele herziening van het denkmodel achter StUF. In dat scenario behoudt StUF zijn inhoudelijke positionering, namelijk als generieke benadering voor het model-gedreven ontwerpen van berichten. Echter, het zou gefundeerd moeten worden in een nieuw service-gericht denkmodel, dat minder registratie-gericht is, expliciet aandacht besteedt aan gebeurtenissen en aan functionele modellen en berichtenschema's vormgeeft in de situationele context van de dienst en haar contract. De klassieke kernfunctionaliteit van het huidige StUF (kennisgevingen en bevestigingen) kunnen vervolgens op dat nieuwe fundament opnieuw worden gebouwd.

De inhoud van de bestaande GFO's is in een dergelijk scenario waarschijnlijk voor het overgrote deel te hergebruiken. Verder zou in dit scenario vanaf het begin gebouwd moeten worden op internationale beschikbare open standaarden, voor logistiek, syntax en beschrijving, in plaats van deze steeds achteraf in te bedden of met specifieke bindingen te bedienen.

4.4 Ten slotte

Ten slotte komt dit rapport terug op het gestelde in sectie 1.2.1. en wil het twee adviezen geven die aan de orde zijn onafhankelijk van de keuze voor het toekomstscenario van StUF. Die adviezen zijn:

- Expliciteer het denkmodel achter StUF in een metamodel.
- Ontwikkel een StUF-compliancetoets, waarmee een onafhankelijke berichtenontwerper zelf kan toetsen of een koppelvlak of sectormodel StUF-compliant is.