Voorstel voor omgaan met “Geen waarde”

# Inleiding

Een element in de inhoud van een bericht kan in sommige gevallen geen waarde hebben. Het niet hebben van een waarde wil echter nog niet zeggen dat er geen betekenis gehecht kan worden aan dit element. Zo kan de betekenis van het niet hebben van waarde van het element overlijdensdatum bij een persoon betekenen dat deze persoon leeft. Het kan echter ook betekenen dat de zender de overlijdensdatum niet weet of niet mag communiceren. Binnen de drie standaarden zijn er verschillende manieren om hier mee om te gaan en wordt dit op verschillend detailniveau gestandaardiseerd.

Op basis van een analyse van de overeenkomsten en verschillen is in dit document een voorstel voor harmonisatie gegeven. De analyse van de verschillen en overeenkomsten is opgenomen in de bijlage.

# Begrippenkader

Het onderwerp dwingt tot een subtiel en nauwkeurig taalgebruik. We beginnen daarom eerst met een aantal definities. Dit voorstel gebruikt de volgende termen en begrippen:

* **veld** voor een ding in de database waar je technische waarden in kunt zetten.
* **element** voor een ding in een bericht waar je technische waarden in kunt zetten.
* **attribuut** voor een onderdeel van de functionele beschrijving van een object.
* **technische waarde** voor een rijtje characters welke je in een bepaalde techniek mag gebruiken. In een XML-bericht zijn dat UTF-8 characters.   
  Zaken als 'noValue: geenWaarde', 'xsi:nil ="true"', ‘2014-10-20’, ’00000000’ en ‘true’ zijn van die waarden en ook de concatenatie hiervan is weer zo’n waarde.
* **gangbare waarde** voor iets wat door de eindgebruiker (misschien beter de niet –IT-er) als waarde herkend wordt voor een attribuut.  
  Zo is ‘2014-10-20’ een gangbare waarde voor een datum, maar ‘0000-00-00’ niet. ‘00000000’ is een gangbare waarde voor een kilometerstand (alhoewel je die niet veel ziet). En 2014-10 is een gangbare waarde voor een onvolledige datum van type gYearMonth. Je kunt in één element verschillende gangbare waarden toestaan. Zo krijg je de gangbare waarden voor een type als ‘mogelijke onvolledige datum’.
* **fictieve waarde** voor een technische waarde die geen gangbare waarde is.  
  De technische waarde ‘geenWaarde’ is geen datum, maar deze technische waarde kan heel zinvol (functioneel) gebruikt worden in bijvoorbeeld het element geboortedatum, zoals we hieronder zullen zien. geenWaarde zien we binnen dit document als een fictieve waarde.
* **gegeven** voor een uitspraak die we als waar beschouwen en daarmee een bepaald feit uitdrukt.  
  Bijvoorbeeld: ‘Piet is overleden op 01-01-2000’, ‘Piet is overleden’, ‘De zender heeft geen gegevens over het overlijden van P’ zijn drie gegevens.

Een gegeven coderen we door in een veld of in een element een technische waarde op te nemen. Als het geen fictieve waarde is, dan is (meestal) wel duidelijk wat we bedoelen. Gebruik je fictieve waarden dan moet je (meestal) afspreken op welke manier je gegevens moet (de)coderen .

# Voorstel tot harmonisatie

De mogelijkheid tot harmonisatie is hier zeer groot. Het voorstel is om de volgende punten door te voeren:

1. Opnemen van de codelijst die StUF en NEN3610 al hanteren uitgebreid met één waarde, **waardeBestaat**. Voor een uitgebreide analyse en toelichting zie paragraaf 4.  
   Dat levert de volgende zes situaties voor ontbrekende gangbare waarden met codes en definities:

|  |  |
| --- | --- |
| **Code** | **Definitie** |
| nietOndersteund | Zender houdt in zijn registratie geen waardes voor dit element bij. |
| nietGeautoriseerd | Zender vindt dat de ontvanger niet geautoriseerd is om de waarde van dit attribuut te kennen |
| geenWaarde | Het attribuut heeft in de werkelijkheid geen waarde.  Bijvoorbeeld toepassen op element overlijdensdatum als de zender communiceert dat de persoon niet is overleden. |
| waardeBestaat | Er bestaat een gangbare waarde voor het attribuut.  Bijvoorbeeld toepassen op element overlijdensdatum als zender weet dát de persoon is overleden, maar niet weet wanneer. |
| waardeOnbekend | Of er een gangbare waarde is en zo ja, wat die waarde is voor dit attribuut, is bij de zender niet bekend.  Bijvoorbeeld toepassen op element overlijdensdatum als zender niet weet of persoon overleden is of niet. |
| vastgesteldOnbekend | Er is een gangbare waarde voor het attribuut, maar het is vastgesteld dat die waarde van het attribuut onbekend is en niet meer kan worden achterhaald  Bijvoorbeeld toepassen op element overlijdensdatum als zender weet dat persoon is overleden en vastgesteld heeft dat de overlijdensdatum niet is te bepalen. |

1. Binnen een sectormodel kan men voor bepaalde datatypen een lijst van alternatieve fictieve waarden voor de zes verschillende situaties toevoegen.
2. Omdat er geen 100% zekerheid is of dit alle mogelijke situaties zijn, is het voorstel om deze lijst als “uitbreidbare” lijst op te nemen in de standaard. Dat betekent dat als de zender iets wil communiceren over ontbrekende waarden dat deze het dan op de manier moet zoals in deze voorstellen beschreven, behalve als de zender een betekenis wilt overdragen over een situatie die hier niet in voorkomt. Uiteraard moet deze nieuwe situatie en bijbehorende code dan wel in afstemming met al de partijen die betrokken zijn bij de communicatie worden opgesteld. In dit geval is het uiteraard goed om dat ook als RFC weer aan te dragen zodat de lijst met opgenomen situaties, codes en definities zo volledig mogelijk is. (tot aan de eerstvolgende versie van de standaard, kun je dan toch al gebruik maken van deze nog ‘niet-standaard-waarde’.
3. Op één manier berichten samenstellen als er geen gangbare waarde is.  
   Voor elementen die volgens de standaard verplicht zijn en waarvoor de zender geen gangbare waarde heeft in het overeenkomende veld of geen waarde mag/wil leveren, zal óf via een XML-attribute óf via een alternatieve fictieve waarde aangegeven moeten worden wat de betekenis is van het element.
4. Voor elementen die volgens een sectormodel optioneel zijn, stellen we voor om –vanaf doorvoeren van deze RFC als best practice in nieuwe situaties – het volgende af te spreken:  
   A. In het sectormodel moet opgenomen worden wat de default betekenis is van het weglaten van een optioneel element: waardeOnbekend of geenWaarde.   
   B. Voor de uitzonderingen op die (lokale) default kan gespecificeerd worden dat het weglaten van het element de andere betekenis heeft.   
   Dus bijvoorbeeld:  
   Default: Weglaten betekent waardeOnbekend,   
   maar bij weglaten element ‘xyz’ betekent dit hetzelfde als geenWaarde  
   Ofwel waar het element ‘xyz’ is weggelaten kun je toevoegen:   
   <ns:xyz xsi:nil="true" StUF:noValue = “geenWaarde”/>  
   C. Als er niets over wordt opgenomen in het sectormodel betekent het weglaten van een element in een wijzigingsbericht, dat we te maken hebben met een situatie waar er geen nieuw gegeven is.   
     
   Dit voorkomt de miscommunicatie dat het feit dat je niets zegt over het overlijden van een persoon wordt geïnterpreteerd als een melding dat de persoon niet overleden is.   
   Door ook de mogelijkheid te bieden dat weglaten geenWaarde betekent, voorkom je veel dataverkeer om door te geven dat er niets door te geven is.  
   Door per sectormodel aan te geven wat de default is, kun je het aantal uitzonderingen beperken,  
   Bijvoorbeeld: Als iemand in zijn naam geen voorvoegsel heeft, dan kun je met behulp van de eerste twee regels aangeven dat weglaten geenWaarde betekent. Als je vervolgens voor dat element wil doorgeven dat je niet weet wat het voorvoegsel is, kun je (in StUF) gebruik maken van noValue:waardeOnbekend.

Volledige harmonisatie, precies dezelfde XML bij alle standaarden, is lastiger. GML is een internationale standaard daar is dus weinig tot geen bewegingsruimte. Daar zal allereerst dit punt moeten worden geadresseerd. Vervolgens is het afwachten of deze ook wordt doorgevoerd. In StUF is StUF:novalue geintegreerd in de StUF methodiek.

Voor de langere termijn kan er gekeken worden naar een eenduidige betekenis van de XML structuur voor de situaties waarbij geen element wordt geleverd. Dit impliceert dat voor de bestaande berichtspecificaties bepaald moet worden welke bestaande interpretatie geldt. Er zijn nu bijvoorbeeld implementaties waar geldt dat weglaten van een optioneel element overeenkomt met het gebruik van geenWaarde. Met bovenstaande best-practice is in ieder geval expliciet te beschrijven wat de betekenis is van het weglaten. Voor de bestaande koppelingen zal een migratiestrategie moeten worden bedacht. Een dergelijke harmonisatie in XML structuur kan echter beter opgepakt worden in een bredere context waarbij de structuur van meer zaken dan alleen geen waarde geharmoniseerd wordt.

# Vertaling codering naar XML

Voor de bovengenoemde codes van lege waarden is in de namespace van GAB het simpleType “NulReden” gedefinieerd.[[1]](#footnote-1)

<xs:schema

xmlns:xs="<http://www.w3.org/2001/XMLSchema>"   
 xmlns:GAB="http://www.noraonline.nl/GAB/gab0100"

targetNamespace="http://www.noraonline.nl/GAB/gab0100"  
 elementFormDefault="qualified"

attributeFormDefault="unqualified"

version="1.0">

<xs:simpleType name="NulReden">

<restriction base="string">

<xs:enumeration value="nietOndersteund"/>

<xs:enumeration value="nietGeautoriseerd"/>

<xs:enumeration value="geenWaarde"/>

<xs:enumeration value="waardeBestaat"/>

<xs:enumeration value="waardeOnbekend"/>

<xs:enumeration value="vastgesteldOnbekend"/>

</xs:restriction>

</xs:simpleType>

...

</xs:schema>

In de namespace van een andere standaard kan deze enumeratie eventueel worden uitgebreid met nieuwe codes door middel van het extension-mechanisme van XSD.

# Betekenis van de gebruikte coderingen

## Inleiding

De situaties waarin we de verschillende coderingen kunnen gebruiken laten zich opsplitsen in twee clusters:

* Situaties waarin we te maken hebben met een zendende administratie die het gevraagde element niet ondersteunt of een ontvanger die niet geautoriseerd is voor het gevraagde element.
* Situaties waarin het voorgaande niet speelt en de gegevens er niet of niet geheel zijn en dat ook al dan niet is vastgesteld.

Deze twee clusters worden hieronder uitgewerkt. Voor elk cluster zijn een aantal situaties uitgewerkt, welke zijn opgenomen in een tabel. Elke situatie is beschreven in de vorm van een gegeven. Dat is de tekst in een cel tussen haakjes.

Het probleem (het vraagstuk) is nu hoe we deze situaties aanduiden (coderen), d.w.z. welke technische waarde je in zou kunnen vullen bij een element waar deze situatie optreedt. Daarom is voor elke situatie, waar mogelijk, aangegeven met welke technische waarde je dat codeert.

De situaties gaan over overlijden, geboorte en telefoonnummer. De situaties zijn hiervoor gerangschikt naar kolom. In elke kolom is de naam opgenomen van het element dat je gebruikt om dat soort situaties te coderen. We gaan er hier vanuit dat de elementen overlijdensdatum en geboortedatum verplicht zijn in het bericht[[2]](#footnote-2), maar telefoonnummer niet. Dat wil zeggen: verplicht volgens de standaard[[3]](#footnote-3).

## Betekenis van ‘nietOndersteund’ en ‘nietGeautoriseerd’

Voor dit cluster hebben we hieronder een tabel opgenomen met 6 verschillende situaties. De situaties voor rond geboorte en overlijden zijn hier niet wezenlijk verschillend, maar omdat deze in het volgende cluster wel wat toevoegen, zijn ze hier ook opgenomen.

Voor overlijden, geboorte en telefoonnummer zijn er steeds 3 situaties die op elkaar lijken. Deze situaties zijn per **rij** opgenomen. Zo krijgen we 2 rijen.

In de eerste rij hebben we te maken met de situaties dat de zender het te leveren element totaal niet kent.

De gebruikte code is *nietOndersteund*.

In de tweede rij hebben we te maken met de situaties dat de zender vindt dat de ontvanger niet geautoriseerd is voor het te leveren element.

De gebruikte code is *nietGeautoriseerd*.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Overlijdensdatum**  **Verplicht** | **Telefoonnummer**  **Niet verplicht** | **Geboortedatum**  **Verplicht** |
| **1** | (Zender ondersteunt ‘overlijdensdatum niet)  ‘nietOndersteund’ | (Zender ondersteunt ‘overlijdensdatum niet)  *Laat bij persoon P het telefoonnummer weg* | (Zender ondersteunt ‘geboortedatum niet)  nietOndersteund’ |
| **2** | (Ontvanger is volgens zender niet geautoriseerd)  ‘nietGeautoriseerd’ | (Ontvanger is volgens zender niet geautoriseerd)  *Laat bij persoon P het telefoonnummer weg* | (Ontvanger is volgens zender niet geautoriseerd)  ‘nietGeautoriseerd’ |

Noot: Het is aan de zender om te bepalen of hij dit gebruikt. Vragen naar gegevens die de zender niet kent of waarvoor de vrager niet geautoriseerd is, kunnen ook als een fout worden bestempeld.

## Betekenis van ‘geenWaarde’, ‘waardeBestaat’, waardeOnbekend en ‘vastgesteldOnbekend’

We gaan er hier vanuit dat de ontvanger geautoriseerd is en de zender het element ondersteunt.

Voor dit cluster hebben we hieronder een vergelijkbare tabel opgenomen met 15 verschillende situaties. Situaties waarbij we van een persoon P alleen weten in welk jaar hij geboren (cq. overleden) is of alleen in welke maand van een bepaald jaar, nemen we hier niet in beschouwing.

Voor overlijden, geboorte en telefoonnummer zijn er steeds 3 situaties die op elkaar lijken. Deze situaties zijn per **rij** opgenomen. Zo krijgen we 5 rijen.

De eerste vier rijen kun je beschouwen als de meest voor de hand liggende situaties, daarna komt een minder voor de hand liggende situatie. We nemen ze hieronder door aan de hand van de overlijdenssituaties.

In rij 1 hebben we de situatie waarin je zegt dat iemand op een bepaalde dag is overleden. Alleen in de eerste rij gebruiken we een gangbare waarde om dit te coderen, d.w.z. een datum. In de volgende rijen gebruiken we een fictieve waarde.

In rij 2 zien we de situatie dat je zegt dat de persoon is overleden. Je zegt hier niets over de datum waarop dat is gebeurd. Er bestaat dus wel een datum (een gangbare waarde), maar die is niet bekend of je zegt het niet. Dit kan bijvoorbeeld het geval zijn als je niet heel actief je administratie bijhoudt, of als de exacte datum voor je administratie niet zo belangrijk is, of als je de overlijdensdatum niet wilt, mag of hoeft te communiceren.

Deze situaties duiden we aan met de code: *waardeBestaat*.

In rij 3 zien we de situatie dat je zegt dat de persoon niet is overleden. Er is dus geen datum, geen gangbare waarde.

Deze situaties duiden we aan met de code: *geenWaarde*.

Rij 2 en rij 3 bevatten wel een gangbare situatie, die we echter met een niet gangbare waarde coderen.

In rij 4 zien we de situatie dat je zegt dat je niets weet over het overlijden van de persoon. Dit wordt niet vaak gecommuniceerd, meestal zeg je dan niets. Weglaten kan echter alleen als het element niet verplicht is, maar daar moet je wel bij uitkijken. (Zie voorstel sub 5)

Deze situaties duiden we aan met de code: *waardeOnbekend*.

Vaak is het zo dat als in een administratie het veld overlijdensdatum leeg is, dit geïnterpreteerd wordt als dat de persoon nog in leven is. Dat is echter niet altijd zo. Vergelijk dit maar met deze situatie: zolang er nog geen einddatum is voor je adres, woon je er nog. De situatie dat een persoon nog in leven is, duiden we echter expliciet met *geenWaarde* (zie hiervoor bij 3.).

In rij 5 belanden we in de meer uitzonderlijke situaties. Je zegt hier net iets meer dan in rij 2.

Hier zeg je namelijk dat je weet dat de persoon is overleden en dat je niet meer kunt vaststellen (bepalen) wanneer dat is geweest.

Dit is bijvoorbeeld het geval als van iemand een overlijdensakte is gevonden, maar de overlijdensdatum die daar op staat niet meer te lezen is. Dat komt bij de GBA voor.

Deze situaties duiden we aan met de code: *vastgesteldOnbekend*.[[4]](#footnote-4)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Overlijdensdatum**  **Verplicht** | **Telefoonnummer**  **Niet verplicht** | **Geboortedatum**  **Verplicht** |
| **1** | (P is overleden op D)  D | (P heeft telefoonnummer T)  T | (P is geboren op D)  D |
| **2** | (P is overleden)  'waardeBestaat'. | (P heeft een telefoonnummer)  'waardeBestaat’ | (P is geboren)  'waardeBestaat’ |
| **3** | (P is niet overleden)  'geenWaarde' | (P heeft geen telefoon)  'geenWaarde' | (P is niet geboren)  ‘geenWaarde’ |
| **4** | (Zender heeft geen gegevens over het overlijden van P)  ‘waardeOnbekend’ | (Zender heeft geen gegevens over het telefoonnummer van P)  *Laat bij persoon P het element telefoonnummer weg* | (Zender heeft geen gegevens over de geboorte van P)  ‘waardeOnbekend’ |
| **5** | (P is overleden en zender kan niet bepalen wanneer)  'vastgesteldOnbekend' | (P heeft een telefoon en zender kan niet bepalen met welk nummer)  'vastgesteldOnbekend' | (P is geboren en zender kan niet kunnen bepalen wanneer)  'vastgesteldOnbekend' |

**Noten:**

1. In theoriekun je nog situaties onderkennen dat je niet met zekerheid kunt vaststellen (bepalen) of de persoon is overleden of niet. Je weet dus niks en dat heb je vastgesteld. Deze situatie wordt echter niet onderkend in de GBA. Theoretisch bestaat deze mogelijkheid wel, maar omdat deze situatie in de praktijk niet voorkomt, nemen we er geen codering voor op.
2. We kennen ook nog situaties bij interpretatie van onzorgvuldige administraties. Deze administraties zijn ambivalent. In dit soort administraties betekent het niet gevuld zijn van het veld overlijdensdatum bij sommige personen dat de persoon niet is overleden en bij sommige personen dat de administratie er geen gegevens over heeft, maar wat nu wanneer aan de hand is kun je aan de registratie niet meer zien. Je weet per persoon dus alleen maar dat óf het een óf het ander aan de hand is. Voor dit soort situaties moeten we afspreken dat het leeg zijn van het veld niet betekent dat de persoon niet overleden is. Registreren dat een persoon niet overleden is, is in zo’n administratie dus niet mogelijk. We nemen die hier verder niet meer op in het overzicht en coderen dit verder gewoon als in rij 4, de zender kan tenslotte niet minder melden dan dat hij niks weet.
3. Rij 5 kan alleen in administraties voorkomen waar er een meta-gegeven of fictieve waarde is om het vastgesteld zijn van het ontbreken van gegevens te registreren. Dat komt weinig voor.
4. In geval van een verplicht element waarin de gangbare waarden datums zijn, wordt (nu nog) in StUF ook nog gebruik gemaakt van codes als jaarOnbekend in combinatie met indOnvolledigeDatum. Op de hier beschreven manier werkt het echter onafhankelijk van het domein van het element. Vandaar dat we dat niet overnemen in deze RFC.
5. waardeBestaat kan ook gebruikt worden in situaties waar de zender de gangbare waarde wel heeft, maar dit niet wil communiceren vanwege bijvoorbeeld ontbrekende autorisatie van de ontvanger, terwijl het feit dat de gangbare waarde bestaat wel gecommuniceerd mag worden.
6. Rij 3 voor geboorte lijkt mogelijk wat vreemd omdat personen toch altijd geboren zijn. Voor de Jeugdzorg krijgen we echter nu ook te maken met ongeboren kinderen. Daarom kan geenWaarde ook bij geboortedatum voorkomen. In geval dat Personen per definitie geboren zijn, en dus de waarde geenWaarde niet gebruikt mag worden, zal dat expliciet in het berichtmodel moeten worden opgenomen.
7. In geval van een niet verplicht element staan we ook toe dat waardeOnbekend wordt gebruikt. De meest gebruikte praktijk is echter dat het element wordt weggelaten.
8. ‘Persoon is overleden’ en ‘Persoon is overleden op Dag’, kun je zien als twee gegevenstypen. In theorie zou je dan ook twee velden of twee elementen daarvoor kunnen opnemen in je datamodel. Uit praktische overwegingen wordt daar echter vaak vanaf gezien en wordt met fictieve waarden gewerkt. Zolang je je hier maar bewust van bent en de interpretatie van de gebruikte waarden goed uitlegt (in termen van gegevens), geeft dit geen problemen.
9. Zoals hiervoor al is gesteld zullen er weinig administraties zijn die het al dan niet vastgesteld zijn van een ontbrekend gegeven opnemen. De waarde ‘vastgesteldOnbekend’ zal dus ook weinig worden toegepast. Je zou zelfs kunnen overwegen om dit wel expliciet mee te modelleren in je informatiemodel en er dan eventueel een extra element voor op te nemen. Ook hier geldt dat we met deze praktische oplossing kunnen werken, als we de betekenis ervan goed vastleggen.

# Bijlage: Over dit voorstel van het project ‘Utrecht’

Dit voorstel is een product van het project Utrecht. De aanleiding voor het project Utrecht was de behoefte van basisregistraties aan een verzameling technische afspraken die zij kunnen toepassen bij de berichtuitwisseling met al hun afnemers. Deze verzameling afspraken heeft de naam ‘Gemeenschappelijk Afspraken Berichtstandaarden’ (GAB) gekregen. Het idee van deze GAB is dat zowel de basisregistraties als de drie sectorstandaarden StUF, SuwiML en NEN 3610 en, indien van toepassing, ook Digikoppeling zich aan de GAB gaan houden, zodat binnen de Nederlandse overheid harmonisatie van berichtuitwisseling op technisch niveau ontstaat.

Het Project Utrecht heeft vijf voorstellen opgeleverd voor elementen van de GAB. De voorstellen zijn opgesteld door een werkgroep bestaande uit vertegenwoordigers van NEN 3610, StUF, SuwiML, Digikoppeling en een aantal basisregistraties, met ondersteuning vanuit het Bureau Forum Standaardisatie (BFS) en de stelselarchitecten van het iNUP-programma.

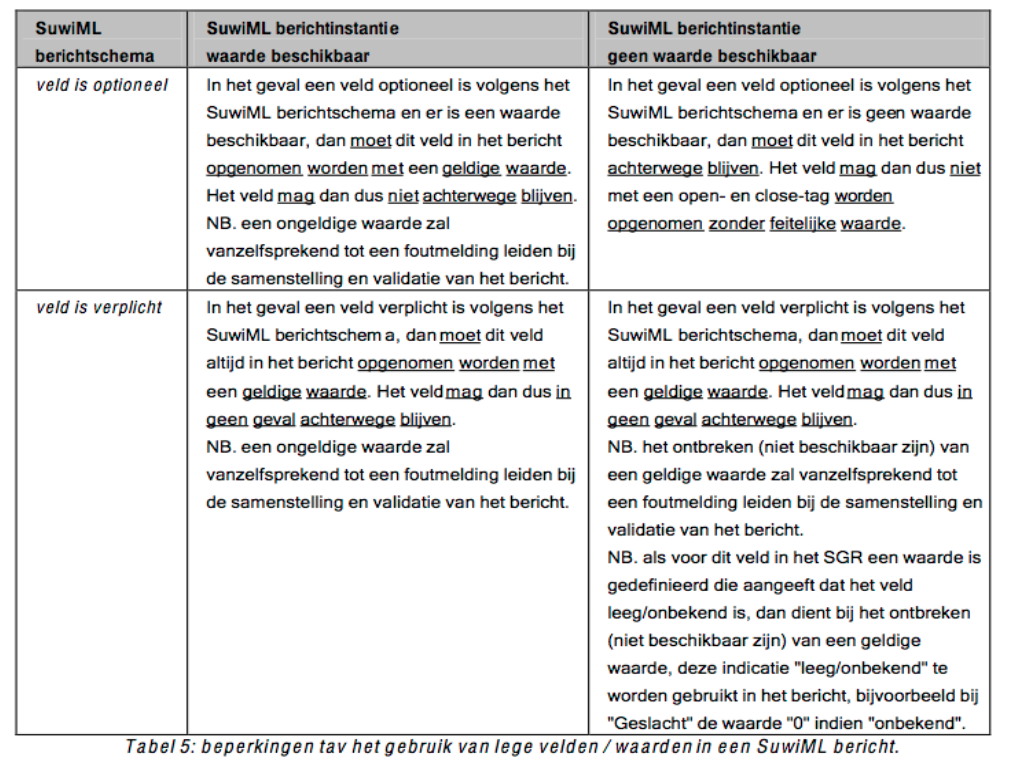
De vijf voorstellen zijn een selectie van ‘laaghangend fruit’ uit een lijst met potentiële gemeenschappelijke afspraken. De verwachting is dat deze vijf voorstellen relatief eenvoudig te adopteren zijn door de basisregistraties en de drie sectorstandaarden. De voorstellen zijn door de werkgroep van het project Utrecht breed uitgezet ter review. De definitieve versies van de voorstellen worden aangeboden aan de beheerders van de drie sectorstandaarden en bij de basisregistraties met het verzoek ze als wijzigingsverzoek in behandeling te nemen.

De sturing op het project Utrecht lag tijdens het i-NUP-programma bij de stuurgroep Digikoppeling, namens de Programmaraad Stelsel van Basisregistraties (PSB). Voor het vervolg op het project Utrecht na i-NUP wordt een Federatief Overleg opgericht waarin naar verwachting KING, BKWI, Geonovum, Logius en een aantal basisregistraties zitting zullen nemen. Dit Federatief Overleg wordt verantwoordelijk voor de sturing van het vervolg op het project Utrecht.

# Bijlage: Analyse

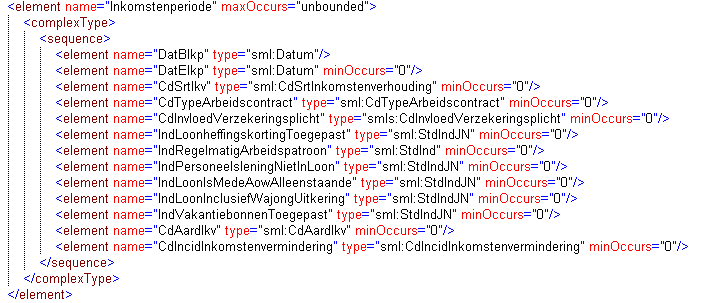
## SuwiML/SGR

SuwiML kent de simpelste vorm van standaardisatie. Deze wordt in de SuwiML standaard verwoord volgens onderstaande tabel.



Kort samengevat komt het er op neer dat als er geen waarde beschikbaar is bij de verzender en het veld optioneel is, het veld weggelaten moet worden. Is het veld verplicht dan moet in het informatiemodel(SGR) een specificatie zijn opgenomen voor een (fictieve) waarde die “geen waarde” betekent. Deze fictieve waarde wordt dan in het verplichte veld opgenomen.

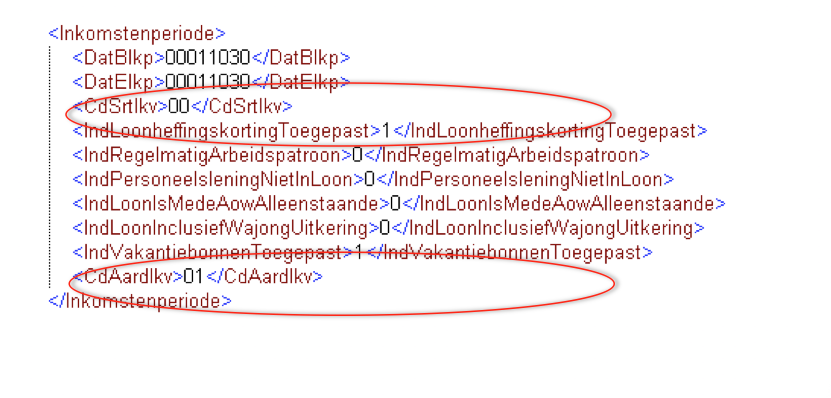
Het omgaan met geen waarde is de verantwoordelijkheid van de informatie modelleur en deze heeft per geval de vrijheid om een passende oplossing te kiezen. Hieronder volgt een voorbeeld van een schema en de uitwerking daarvan in een bericht.



Figuur 1 SuwiML Schema voorbeeld

De ellipsen geven aan waar gegevens die optioneel zijn volgens schema, maar niet geleverd worden.

Als er voor een koppelvlak gegevens zijn waarvoor een partij niet geautoriseerd is, worden deze gegevens niet opgenomen in het schema.



Figuur 2 SuwiML Bericht voorbeeld

## StUF

StUF kent een hoge mate van differentiatie van het begrip “geen waarde”. Het biedt middels het attribuut “StUF:novalue” de mogelijkheid om aan te geven waarom een element geen waarde heeft. Bijvoorbeeld:

<ns:object StUF:entiteittype="Persoon">

<ns:bsn>123456789</ns:bsn>

<ns:overlijdensdatum xsi:nil="true" StUF:noValue="geenWaarde"/>

</ns:object>

In dit geval is de persoon in kwestie nog in leven.

In het volgende geval weten we niet of de persoon wel of niet in leven is omdat de overlijdensdatum onbekend is:

<ns:object StUF:entiteittype="Persoon">

<ns:bsn>123456789</ns:bsn>

<ns:overlijdensdatum xsi:nil="true" StUF:noValue="waardeOnbekend"/>

</ns:object>

Wanneer een element is weggelaten gaat StUF er vanuit dat de waarde onbekend is. Het onderstaande StUF-fragment is equivalent aan het vorige fragment.

<ns:object StUF:entiteittype="Persoon">

<ns:bsn>123456789</ns:bsn>

</ns:object>

NB: wanneer een element “geenWaarde” heeft –bij de zender is dus bekend dat er geen waarde is voor dit attribuut- dan mag het element niet worden weggelaten en moet het element altijd worden opgenomen in het StUF-bericht.

Hieronder de definitie van het attribuut “noValue” in XML schema:

<attribute name="noValue" type="StUF:NoValue"/>

<simpleType name="NoValue">

<restriction base="string">

<enumeration value="nietOndersteund"/>

<enumeration value="nietGeautoriseerd"/>

<enumeration value="geenWaarde"/>

<enumeration value="waardeOnbekend"/>

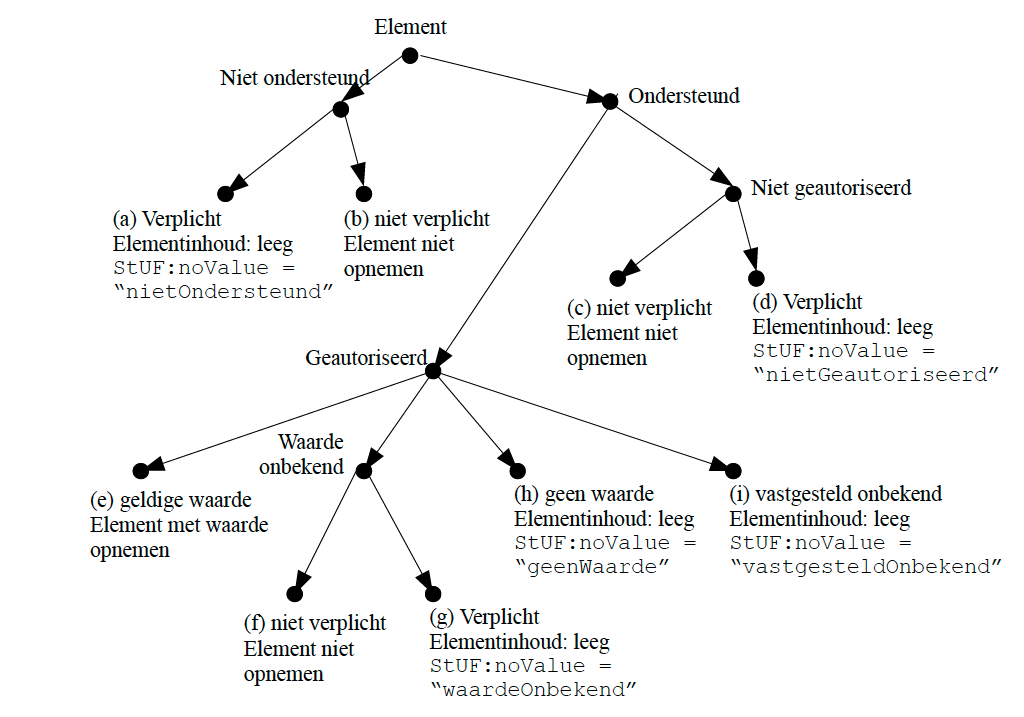
<enumeration value="vastgesteldOnbekend"/>

</restriction>

</simpleType>

StUF kent een aantal criteria op basis waarvan een andere vorm van “geen waarde“ geleverd kan worden.

Dit wordt in de standaard geïllustreerd met onderstaande boom.



En daar hoort deze tekst bij.

StUF 03.01 versie 18 pagina36/37

De eerste beslissing is of het element door de zender wordt ondersteund. Als het niet wordt ondersteund, dan wordt ge­keken of het verplicht is of niet. In de volgende gevallen moet een element verplicht in een bericht worden opgenomen:

1. Het element is een kerngegeven en het bericht is een kennisgevingbericht, waarbij het attribute StUF:sleutelOntvangend niet voorkomt in de entiteit;
2. Het element maakt deel uit van een gegevensgroep, waarvan minimaal één ander element met een waarde in het be­richt wordt opgenomen en het bericht is een kennisgevingbericht;
3. Het element is gevraagd in het vraagbericht waarop het bericht een antwoord is;
4. Het sectormodel specificeert dat het element in een kennisgevingbericht of een antwoordbericht moet voorkomen.

Als het element verplicht is, dan wordt het opgenomen met het attribute StUF:noValue=”nietOndersteund” (a) en een lege elementinhoud. Een dergelijk element kan door de ontvanger worden genegeerd. Als het niet verplicht is, dan wordt een niet-ondersteund element niet opgenomen in het bericht (b). Wanneer het element wel wordt ondersteund, wordt er gekeken of de ontvanger van het bericht geautoriseerd is voor dit element. Als het element niet verplicht is, dan wordt het niet opgenomen in het bericht (c). Als het verplicht is, dan wordt het niet-geautoriseerd zijn expliciet gemeld door het element op te nemen met het attribute StUF:noValue=”nietGeautoriseerd” (d) en een lege elementinhoud.

Indien de ontvanger van het bericht is geautoriseerd om het element te ontvangen, zijn er vier mogelijkheden voor de waarde van het element:

1. Element heeft geldige waarde
2. Het element heeft een waarde, maar de waarde is bij de zender niet bekend
3. Er is vastgesteld dat de waarde van het element onbekend is
4. Element heeft geen waarde

De derde mogelijkheid is een bijzondere vorm van het onbekend zijn van een waarde. In dat geval is vastgesteld dat de waarde 'echt' onbekend is, bijvoorbeeld omdat het brondocument voor de waarde onleesbaar is geworden of omdat het vaststellen van de waarde onmogelijk is geworden, doordat het object niet meer bestaat.

Als het element in de werkelijkheid een geldige waarde heeft, wordt het element met zijn waarde als inhoud opgenomen (e). Als er wel een waarde is, maar deze is bij de zender onbekend, dan wordt het element, als het niet verplicht is niet opgenomen (f) en als het wel verplicht opgenomen met een lege inhoud en StUF:noValue=”waardeOnbekend” (g). Een kerngegeven als de geboortedatum van een persoon waarvan het zendende systeem niet altijd de waarde kent, wordt bijvoorbeeld met StUF:noValue=”waardeOnbekend” opgenomen. Als het element geen waarde heeft, bijvoorbeeld de overlijdensdatum van een niet overleden persoon, dan wordt het element in het bericht opgenomen voorzien met een lege inhoud en StUF:noValue=”geenWaarde” (h). Als vastgesteld is dat de waarde onbekend is, dan wordt het element opgenomen met een lege elementinhoud en StUF:noValue=”vastgesteldOnbekend” (i).

Ten eerste is er de vraag of de zender die de gegevens levert het gevraagde element ondersteunt (d.w.z. opgenomen heeft in zijn administratie). Met StUF is het mogelijk een service te implementeren die niet alle verplichte elementen uit een bijbehorende XSD ondersteunt. Wordt er om zo’n element gevraagd, dan wordt “nietOndersteund“ teruggegeven.

StUF kent ook autorisaties, met name bij persoonsgegevens komen deze veel voor. Indien een vrager geen autorisatie heeft, kan hij “nietGeautoriseerd“ als waarde terugkrijgen.

Indien autorisatie en ondersteuning geen rol spelen zijn er overige gevallen voor “geen waarde“. Te beginnen bij waardeOnbekend, het systeem heeft geen waarde maar weet wel dat er een waarde moet bestaan.

Voor alle tot nu toe beschreven gevallen geldt dat indien het element niet verplicht is het niet opgenomen wordt.

Tot slot kent StUF twee gevallen waarbij een code altijd opgenomen wordt of het element optioneel is of niet. Ten eerste “geenWaarde“ het element heeft nog geen waarde en dit heeft betekenis, bijvoorbeeld het overlijdensdatum voorbeeld uit de inleiding. Tot slot “vastgesteldOnbekend“ een element zou een waarde moeten hebben maar het is vastgesteld dat deze niet te achterhalen is. Bijvoorbeeld wanneer het brondocument verloren is gegaan.

Bij deze laatste twee wordt StUF:noValue altijd opgenomen aangezien de betekenis subtiel verschilt. Bij de eerste drie houdt geen waarde in: er moet nu een waarde zijn maar door omstandigheden aan server of client kant krijg je die niet. Bij (h) geldt: er is op dit moment geen geldige waarde in te vullen anders dan geen waarde. Bij (i) geldt: het is onmogelijk om de waarde op dit moment op wat voor manier dan ook te achterhalen.

## NEN3610

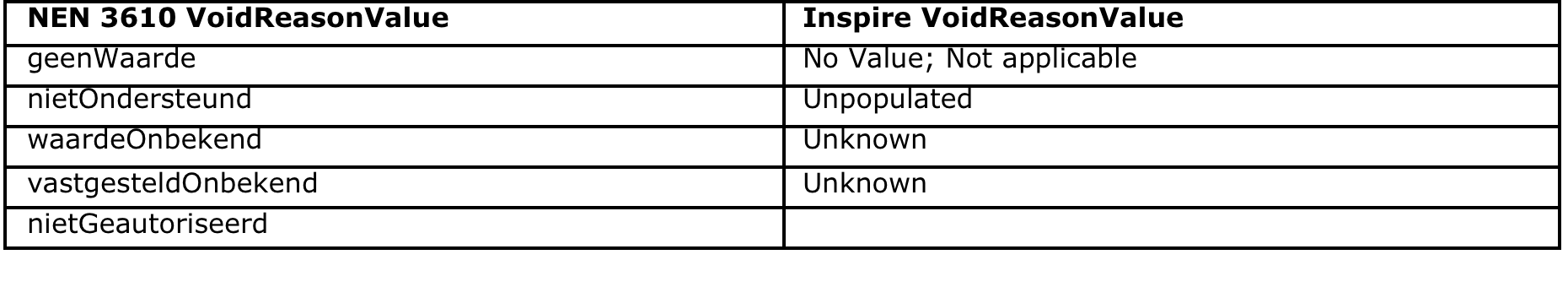
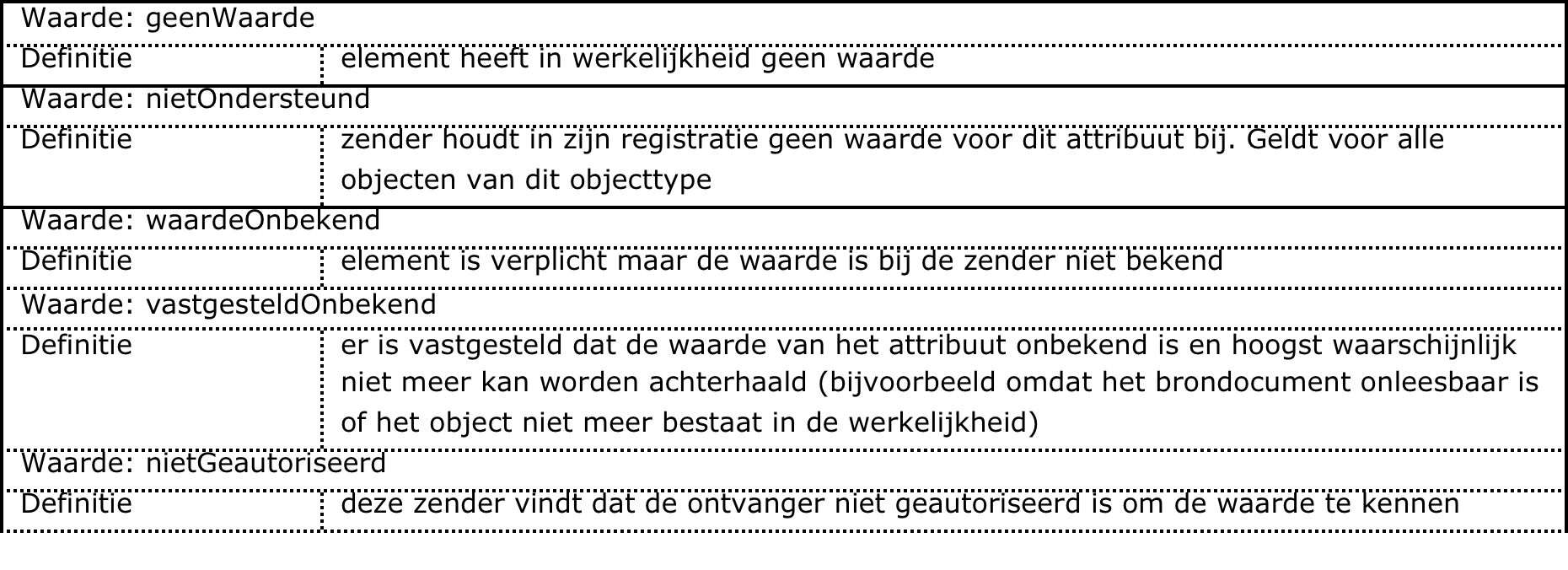
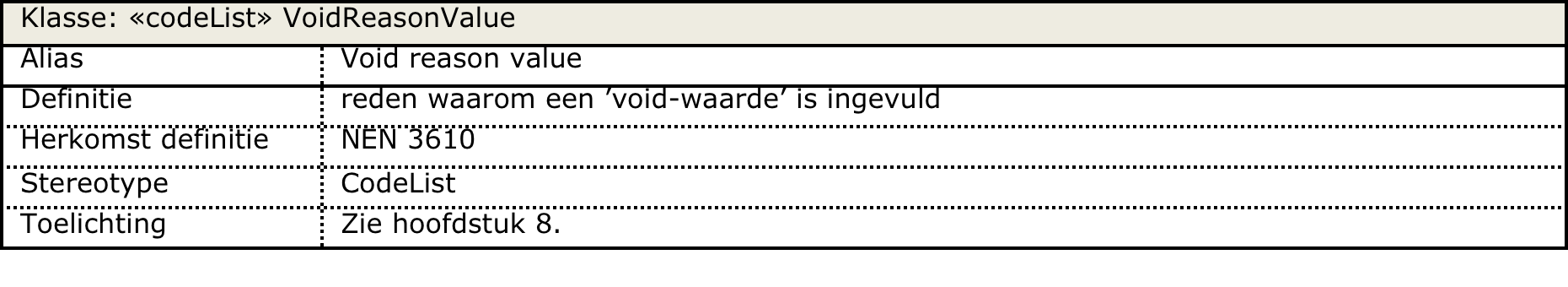
In de NEN3610 wordt “geen waarde“ in UML aangegeven met het stereotype <<voidable>> dat kan worden gebruikt op attributen en associaties. Hier een voorbeeld van een UML klasse met een attribuut ‘naam’ waar dit is toegepast:



Met daarbij een lijstje mogelijke redenen van de ontbrekende waarde, afgestemd met StUF:

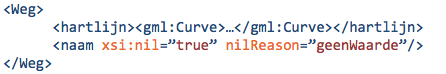


In de NEN3610[[5]](#footnote-5) is dit als volgt gedefinieerd:



In een GML application schema ziet dat er zo uit:

En in een GML data bestand:



## Overeenkomsten

In alle gevallen gaat het om situatie waarin er geen sprake is van een technische fout, maar van een functioneel herkenbare situatie,

In alle gevallen wordt de situatie onderkend dat er in de administratie een leeg veld is (technisch gezien een null value of een fictieve waarde (bijv. 99-99-9999 in een datumveld) of dat de leverende applicatie geen waarde heeft of geen waarde kan afleiden. Alleen in de StUF en NEN3610 situatie wordt daar echter bij onderkend dat dit meerdere dingen kan betekenen:

* De gangbare waarde bestaat niet (overlijdensdatum van een levend persoon)
* De gangbare waarde is niet bekend (geboortedatum van een persoon die dit zelf ook niet weet of waar we de vraag niet hebben gesteld)
* De gangbare waarde is niet bekend na goed onderzoek (geboortedatum van een immigrant die het zelf niet weet en waarbij er geen enkel formeel document is gevonden waar het op is vastgelegd).

Tussen de standaarden bestaan veel overeenkomsten. StUF en NEN3610 hebben op dit onderwerp al in het verleden geharmoniseerd en onderkennen dezelfde waardelijst voor met mogelijkheden voor geenWaarde. StUF en SuwiML hanteren een zeer vergelijkbare methode daar waar het omgang met optionele velden/elementen betreft. Binnen SuwiML vindt de definitie van een veld zonder waarde dat moet worden gecommuniceerd plaats op informatiemodel niveau in het SGR. NEN3610 onderkent ook op informatiemodel/UML niveau dat een attribuut stereotype “Voidable“ is. Bij StUF is dit niet het geval.

## Verschillen

In geval van StUF wordt een berichtmodel gespecificeerd onafhankelijk van de situatie waarin die gebruikt gaat worden. Dat leidt tot de mogelijke situaties dat:

1. Een bericht een verplicht element benoemt, welke niet door de supplier geleverd kan worden.
2. Een bericht een element benoemt, waar de consumer niet voor geautoriseerd is.

Het lijkt er op dat dit in de SuwiML context niet van toepassing is.

NEN3610 en StUF zijn al geharmoniseerd, maar er zijn nog wel verschillen.

* De NEN 3610 kent naast de StUF waarde lijst voor geen waarde ook een lijst van Inspire met andere waarden. Deze worden toegepast bij Inspire plichtige datasets.
* De XML is bij alle standaarden duidelijk anders van vorm.
* Daarnaast onderkent StUF een aantal gevallen waarin “geenWaarde” altijd wordt doorgegeven ook als het element optioneel is, terwijl bij SuwiML de regel is om bij geen waarde optionele velden altijd weg te laten

Dit leidt in de praktijk tot verschillende manieren voor het omgaan met geen waarde: in SuwiML wordt de overlijdensdatum weggelaten indien een persoon nog leeft, bij StUF kan de overlijdensdatum weggelaten worden indien niet bekend is of een persoon nog leeft. Is zeker dat een persoon nog leeft dan dient StUF:noValue = geenWaarde te worden teruggegeven. Dit kan leiden tot miscommunicatie.

1. De naam “NulReden” is de Nederlandse vertaling van de internationaal term “NilReason”. [↑](#footnote-ref-1)
2. Let op: verplicht zijn is contextgevoelig. In StUF kun je bijvoorbeeld bij het stellen van de vraag aangeven dat een element opgenomen moet zijn in het antwoord, ook als het verder in het model nergens anders verplicht is. [↑](#footnote-ref-2)
3. Voor de StUF-regels rond verplicht zijn van elementen, zie paragraaf 6.2. [↑](#footnote-ref-3)
4. In de GBA wordt dit gecodeerd met de fictieve waarde 00000000. In geval van waarden die een relatie implementeren wordt als fictieve waarde een . opgenomen. [↑](#footnote-ref-4)
5. <http://www.nen.nl/web/Normshop/Norm/NEN-36102011-nl.htm> [↑](#footnote-ref-5)