

een samengesteld element worden gegroepeerd. De ontwerper van een sectormodel kan zelf specificeren of elementen binnen zo'n groep verplicht zijn of niet, bijvoorbeeld om een gegevensgroep rond huisnummer of centroidcoördinaten te definiëren. Een samengesteld element mag alleen in een bericht worden opgenomen, als het minstens één element voor een attribuut, voor een samengesteld element of voor een relatie bevat.

Voor samengestelde elementen definieert de StUF-standaard verder geen enkele functionaliteit. Door StUF gedefinieerde attributes voor de waarde van een eigenschap (StUF-element) of voor een relatie (StUF-relatie) mogen niet voorkomen op samengestelde elementen. De samengestelde elementen zijn slechts een middel om elementen te groeperen. Voorschriften in de StUF-standaard voor elementen binnen een fundamentele entiteit, een relatie-entiteit of een tabelentiteit hebben altijd uitsluitend betrekking op de elementen op het laagste niveau in een samenstelling.

3.2.4 *Het opnemen van niet in het sectormodel gedefinieerde elementen*

De in het sectormodel gedefinieerde elementen en hun tags worden door alle partijen binnen de sector ofwel ondersteund zoals ze gedefinieerd zijn ofwel niet ondersteund. Soms zal een deel van de partijen in een sector met elkaar afspraken willen maken over elementen die niet voor anderen bindend zijn.

Het kan voorkomen dat er gegevens nodig zijn die niet in de XSD-schema's van het sectormodel gedefinieerd zijn. Dit kan diverse redenen hebben:

- nieuwe ontwikkelingen in de basisregistraties of Europese wetgeving (bijv. invoering van IBAN en BIC voor bankrekeningen). Het is vaak te ingrijpend om hiervoor meteen de bestaande berichtschemata te veranderen en een nieuwe versie van het sectormodel uit te brengen.
- bij uitwisseling van zaakgegevens is er behoefte om per zaaktype specifieke gegevens op te nemen die niet van te voren zijn gedefinieerd in het sectormodel StUF-ZKN.
- een deel van de partijen in een sector heeft behoefte aan aanvullende gegevens die niet bindend zijn voor andere partijen.

Om in deze behoefte te voorzien biedt StUF twee constructies om extra elementen toe te voegen aan een bericht zonder dat de schema's van het sectormodel hoeven te worden gewijzigd. De eerste constructie bestaat al vanaf StUF 2.04 en wordt beschreven in sectie 3.2.4.1. De nieuwe constructie wordt uitgelegd in sectie 3.2.4.2. De oude constructie is vernoemd naar de elementnaam "extraElementen" (Nederlandse spelling) en de nieuwe constructie is vernoemd naar de elementnaam "extraElements" (Engelse spelling). Beide constructies hebben voor- en nadelen die in de volgende secties zullen worden besproken.

3.2.4.1 *extraElementen*

Voor deze constructie Om hierin te voorzien zijn het element <StUF:extraElementen> en het complexType <StUF:ExtraElementen> gedefinieerd. In een entiteitstype kunnen extra elementen worden opgenomen als een reference naar het element <StUF:extraElementen>. Hieronder staan de definities uit het StUF-schema [StUFXSD].

```
<element name="extraElementen" type="StUF:ExtraElementen"/>

<complexType name="ExtraElementen">
  <sequence>
    <element name="extraElement" nillable="true" maxOccurs="unbounded">
      <complexType>
        <simpleContent>
          <extension base="string">
            <attributeGroup ref="StUF:element"/>
            <attribute name="naam" type="string" use="required"/>
            <attribute ref="StUF:indOnvolledigeDatum"/>
          </extension>
        </simpleContent>
      </complexType>
    </element>
  </sequence>
</complexType>
```

<StUF:extraElementen> wordt dus gedefinieerd als één of meer elementen <StUF:extraElement> met simpleContent van het type string, waaraan het attribute naam, het attribute indOnvolledigeDatum en de attributeGroup StUF:element is toegevoegd. Als een <StUF:extraElement> een datum of tijdstip bevat, dan kan dus ook bij een <StUF:extraElement> gespecificeerd worden dat de datum onvolledig is. De elementnamen voor extra elementen worden buiten het sectormodel om gedefinieerd. Een uitgegeven elementnaam wordt geregistreerd

samen met de aanvrager en desgewenst een definitie of omschrijving. Een dergelijke elementnaam mag niet een tweede keer worden uitgegeven.

Hieronder een voorbeeld van het gebruik van de extraElementen-constructie. Het betreft de recentelijke invoering van IBAN- en BIC-nummers voor bankrekeningen via Europese wetgeving. Deze nieuwe gegevens kunnen als volgt als extra elementen worden toegevoegd aan een NPS-entiteit in StUF-BG 3.10:

```
<BG:persoon StUF:entiteittype="NPS" xsi:schemaLocation="http://www.egem.nl/StUF/sector/bg/0310/bg0310_ent_basis.xsd" xmlns:BG="http://www.egem.nl/StUF/sector/bg/0310"
xmlns:StUF="http://www.egem.nl/StUF/StUF0301" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <BG:inp.bsn>123456789</BG:inp.bsn>
  <BG:geslachtsnaam>Korver</BG:geslachtsnaam>
  <BG:voornamen>Henri Peter</BG:voornamen>
  <StUF:extraElementen>
    <StUF:extraElement naam="iban">NL06INGB0006053682</StUF:extraElement>
    <StUF:extraElement naam="bic">INGBNL2A</StUF:extraElement>
  </StUF:extraElementen>
</BG:persoon>
```

Het voordeel van deze constructie is dat extra elementen makkelijk en snel kunnen worden toegevoegd. Het nadeel is dat deze elementen niet kunnen worden gevalideerd tegen een XSD-schema. In de constructie die in de volgende sectie wordt beschreven kan dat wel.

3.2.4.2 ExtraElements

Voor deze constructie zijn in het stuf0301.xsd schema [StUFXSD] het element

```
<element name="extraElements" type="anyType"/>
```

en het attribute

```
<attribute name="schemaLocation" type="string"/>
```

gedefinieerd. Aan het gereserveerde element extraElements herkent de berichtverwerkende software waar het de extra elementen kan verwachten. Het element is van het type anyType en kan daardoor willekeurige XML-expressies bevatten. Om ervoor te zorgen dat deze expressies gevalideerd kunnen door een XSD-schema eisen we dat dat de inhoud altijd moet voldoen aan de volgende structuur:

```
<stuf:extraElements>
  <ns1:extraElements stuf:schemaLocation="..." xmlns:ns1="...">...</ns1:extraElements>
  <ns2:extraElements stuf:schemaLocation="..." xmlns:ns2="...">...</ns2:extraElements>
  ...
  <nsN:extraElements stuf:schemaLocation="..." xmlns:nsN="...">...</nsN:extraElements>
</stuf:extraElements>
```

Op het hoogste niveau binnen <stuf:extraElements> mogen alleen elementen voorkomen met de naam extraElements maar dan wel uit andere namespaces. Er moet minimaal één zo'n gelijknamig element aanwezig zijn. De namespaces van deze subelementen zijn allemaal verschillend en ongelijk aan de namespace van StUF. Het attribute xmlns: voor de namespace-declaratie is verplicht en het attribute stuf:schemaLocation is optioneel omdat de ontvangende applicatie doorgaans de schemalocaties van de betreffende namespaces al kent en op een eigen lokale plek heeft opgeslagen. In het geval dat de schema's op een centrale plek worden beheerd (bijvoorbeeld een openbare website) kan de schemalocatie worden meegegeven als een URL.

In het bovenstaande XML-fragment valideert het element <stuf:extraElements> altijd ongeacht de inhoud. De subelementen <ns1:extraElements>, <ns2:extraElements>, ..., <nsN:extraElements> kunnen in een aparte stap worden gevalideerd door in het attribute schemaLocation de prefix stuf te wijzigen in xsi (de namespace van XML). De namespace-prefix stuf in stuf:schemaLocation zorgt ervoor dat de ontvangende applicatie de extra elementen niet hoeft te valideren als die er niet in geïnteresseerd is.

Hieronder hetzelfde voorbeeld als uit de vorige sectie, maar nu met de nieuwe constructie:

```
<bg:persoon xsi:schemaLocation="http://www.egem.nl/StUF/sector/bg/0310/bg0310_ent_basis.xsd"
xmlns:bg="http://www.egem.nl/StUF/sector/bg/0310" xmlns:stuf="http://www.egem.nl/StUF/StUF0301"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <bg:inp.bsn>123456789</bg:inp.bsn>
  <bg:geslachtsnaam>Korver</bg:geslachtsnaam>
  <bg:voornamen>Henri Peter</bg:voornamen>
  <stuf:extraElements>
    <xe:extraElements
stuf:schemaLocation="http://www.egem.nl/StUF/sector/bg/0310/extraElements ..."
xmlns:xe="http://www.egem.nl/StUF/sector/bg/0310/extraElements"
xmlns:gml="http://www.opengis.net/gml">
      <xe:bic>INGBNL2A</xe:bic>
      <xe:iban>NL06INGB0006053682</xe:iban>
      <xe:mijnLocatie>
        <gml:Point srsDimension="2" srsName="urn:ogc:def:crs:EPSG:6.6:4326"
axisLabels="x y">
          <gml:pos>49.27 -123.11</gml:pos>
        </gml:Point>
      </xe:mijnLocatie>
    </xe:extraElements>
  </stuf:extraElements>
</bg:persoon>
```

BIC en IBAN zijn nu gedefinieerd als echte XSD-elementen die gevalideerd kunnen worden door een schema. Er is tevens een extra element <xe:mijnLocatie> toegevoegd om te laten zien dat de nieuwe constructie ook complexe structuren met attributen en geneste elementen zoals een GML-locatie toestaat.

In het voorbeeld zitten alle extra elementen in dezelfde namespace. Het is geen enkel probleem om nieuwe namespaces toe te voegen, bijvoorbeeld om extra elementen die specifiek zijn voor een bepaalde leverancier te onderscheiden. Of om onderscheid te maken tussen extra elementen die specifiek zijn voor een bepaald zaaktype. Het groeperen van extra elementen in namespaces is een taak voor de berichtontwerper. Eventuele richtlijnen daarvoor zullen worden beschreven in het StUF Best Practices document.

Hieronder het schema waarmee de extra elementen in bovenstaand voorbeeld gevalideerd worden:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ex="http://www.egem.nl/StUF/sector/bg/0310/extraElements"
xmlns:StUF="http://www.egem.nl/StUF/StUF0301" xmlns:GML="http://www.opengis.net/gml"
targetNamespace="http://www.egem.nl/StUF/sector/bg/0310/extraElements"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="031003">
  <import namespace="http://www.egem.nl/StUF/StUF0301" schemaLocation="stuf0301.xsd"/>
  <import namespace="http://www.opengis.net/gml" schemaLocation="geometryBasic0dld.xsd"/>
  <element name="extraElements" type="ex:ExtraElements"/>
  <complexType name="ExtraElements">
    <sequence>
      <element name="bic" type="ex:Bic-e" nillable="true" minOccurs="0"/>
      <element name="iban" type="ex:Iban-e" nillable="true" minOccurs="0"/>
      <element name="mijnLocatie" type="ex:MijnLocatie"/>
    </sequence>
  </complexType>
  <complexType name="MijnLocatie">
    <sequence>
      <element ref="GML:Point"/>
    </sequence>
  </complexType>
  <complexType name="Bic-e">
    <simpleContent>
      <extension base="ex:Bic">
        <attributeGroup ref="StUF:element"/>
      </extension>
    </simpleContent>
  </complexType>
  <complexType name="Iban-e">
    <simpleContent>
      <extension base="ex:Iban">
        <attributeGroup ref="StUF:element"/>
      </extension>
    </simpleContent>
  </complexType>
  <simpleType name="Bic">
    <restriction base="string">
      <maxLength value="11"/>
    </restriction>
  </simpleType>
```

```
</restriction>  
</simpleType>  
<simpleType name="Iban">  
  <restriction base="string">  
    <maxLength value="34"/>  
  </restriction>  
</simpleType>  
</schema>
```

3.3 Metagegevens

Een object heeft eigenschappen en de waarden van die eigenschappen worden als gegevens van dat object vastgelegd. Bij een persoon zijn de geslachtsnaam (Broek), het e-mailadres (broek009@onsnet.nu) en de geboortedatum (12-10-1958) voorbeelden van gegevens. Ook gegevens of groepen van gegevens kunnen eigenschappen hebben, bijvoorbeeld de periode waarin een bepaalde waarde geldig is (geweest), het feit dat een bepaald gegeven in onderzoek is (geweest) of het brondocument waaraan de gegevens zijn ontleend. Dit soort gegevens worden ook wel metagegevens genoemd.

Metagegevens zijn net zoals de sleutels onafhankelijk van een concreet entiteitstype als Verblijfsobject of Persoon en kunnen dus los van een concreet sectormodel gedefinieerd worden. De StUF-standaard doet dit en definieert voor een relatief brede verzameling metagegevens de betekenis en de functionaliteit. Op deze wijze is er een uniform mechanisme voor het in berichten opnemen van metagegevens. Bij het definiëren van metagegevens heeft de StUF-standaard als uitgangspunt genomen de metagegevens onderkend in de GBA en in de BAG. De StUF-standaard definieert wel van de GBA en BAG afwijkende mechanismen, omdat in StUF een zo eenvoudig en natuurlijk mogelijk gebruik van metagegevens in berichten voorop staat. Twee ontwerpcriteria liggen aan de gemaakte keuzen ten grondslag:

1. Metagegevens zijn een uitbreiding op een entiteitstype. Het toevoegen van metagegevens aan een entiteitstype mag geen consequenties hebben voor de structuur van dat entiteitstype in een bericht anders dan het toevoegen van elementen en attributes voor de metagegevens.
2. Als er voor een entiteitstype metagegevens zijn gedefinieerd, dan dienen gebruikers van berichten voor een entiteitstype met metagegevens zonder problemen ook berichten zonder metagegevens te kunnen maken c.q. dienen ze bij de verwerking de metagegevens eenvoudig te kunnen negeren. StUF heeft de ambitie om met één set objectdefinities zowel simpele als complexe functionaliteit te kunnen ondersteunen, bijvoorbeeld simpele webservices voor het opvragen van een klein setje actuele gegevens en complexe webservices voor het opvragen van een persoon inclusief zijn materiële en formele historie en inclusief brondocumenten.

De StUF-standaard onderkent de volgende groepen metagegevens:

- Metagegevens met betrekking tot historische waarden
- Metagegevens met betrekking tot brondocumenten waaraan de gegevens ontleend zijn
- Metagegevens met betrekking tot de gebeurtenis die aan (een verandering van) gegevens ten grondslag liggen
- Metagegevens met betrekking tot de status van gegevens.

Deze verschillende groepen metagegevens worden in de volgende paragrafen besproken. In de hierna volgende hoofdstukken wordt de functionaliteit voor het omgaan met deze metagegevens in de verschillende soorten berichten besproken.

StUF definieert een mechanisme voor het omgaan met metagegevens. Dit mechanisme kan gebruikt worden om binnen een sectormodel eigen metagegevens te definiëren. Bij het afhandelen van aanvragen voor voorzieningen in het kader van de Wet Maatschappelijke Ondersteuning (WMO) is het bijvoorbeeld handig om bij een aantal eigenschappen van een persoon te kunnen vastleggen in welk onderzoek deze zijn vastgesteld. Een basisregistratie ontleent zijn gegevens aan een brondocument en binnen de WMO-sector worden gegevens vaak vastgesteld in een onderzoek. Zo zijn er waarschijnlijk nog wel meer voorbeelden te geven en het is ondoenlijk om elk soort metagegeven in de StUF-standaard te voorzien. Het door StUF geïntroduceerde mechanisme kan wel binnen sectormodellen worden hergebruikt voor eigen metagegevens.

3.3.1 Metagegevens met betrekking tot historische waarden

Materiële historie van attribuutwaarden bij een object kan worden gespecificeerd met behulp van een tijdvakGeldigheid bestaande uit een beginGeldigheid en een eindGeldigheid. Formele historie van attribuutwaarden kan worden gespecificeerd met behulp van een tijdstipRegistratie. Hieronder worden deze begrippen precies gedefinieerd.