

Mogelijk onvolledige datum

Auteur: Wim Bakkeren (wim.bakkeren@ictu.nl)

Datum: 25 september 2014

Versie: 1.0

Status: Definitief

Inleiding

Dit document bevat een voorstel voor een datatype voor mogelijk onvolledige datums¹. Een 'mogelijk onvolledige datum' is een datum waarvan mogelijk de dag of de maand en de dag onbekend zijn. Een voorbeeld van een volledig bekende datum is 2014-08-27. Voorbeelden van onvolledig bekende datums zijn 2014-08 of 2014. Het datatype is te gebruiken voor datums die soms wel en soms niet volledig bekend zijn, zoals de geboortedatum van personen zonder geldige persoonsdocumenten. Het basis XML-datatype voor datum (xs:date) is hiervoor niet geschikt, omdat het alleen volledige datums toestaat.² Er is geen internationaal gestandaardiseerd basistype voor de mogelijk onvolledige datum.

Het voordeel van één datatype voor de mogelijk onvolledige datum is dat alle partijen hiervoor in hun berichten hetzelfde formaat hanteren. Dat maakt de kans op interpretatieverschillen kleiner en vereenvoudigt de implementatie ervan in berichtverwerking: de verwerking van de mogelijk onvolledige datum hoeft voor berichten van verschillende partijen slechts op één wijze gerealiseerd te worden.

Consequentie van het voorstel is dat bestaande berichtschemas en berichtverwerking er in de toekomst op aangepast moeten worden. Gedurende een bepaalde periode zullen bestaande oplossingen en het hier voorgestelde nieuwe datatype naast elkaar bestaan.

Dit voorstel bevat geen datatype voor de 'mogelijk onvolledige datum en tijd'. De verwachting is dat daaraan geen behoefte bestaat. Indien die behoefte wel blijkt te bestaan, dan kan het voorstel hiervoor relatief makkelijk uitgebreid worden.

Deze versie van het voorstel beschrijft alleen een datatype voor mogelijk onvolledige datums. Eerdere versies van dit voorstel bevatten ook datatypes voor datum, datum en tijd, jaar en maand en jaar op basis van de 'primitive datatypes' van XML-schema³ en voor periode op basis van het GML-datatype TimePeriod⁴. Naar aanleiding van reviewcommentaar is besloten om die datatypes uit dit voorstel te verwijderen en op te nemen in het voorstel 'Basis datatypes', ook van het project 'Utrecht'. Dat voorstel 'Basis datatypes' bevat daarmee alle 'primitive datatypes' vanuit de W3C-recommendation voor XML-schema.

¹ Deze notitie spreekt van 'datums' in plaats van 'data' om verwarring met het begrip 'data' in de betekenis van 'gegevens' te voorkomen.

² Zie <http://www.w3.org/TR/xmlschema-2/#date>

³ Zie <http://www.w3.org/TR/xmlschema-2/#built-in-primitive-datatypes>

⁴ Zie <http://www.opengeospatial.org/standards/gml>

Dit voorstel beperkt zich, zoals gezegd, tot een datatype voor mogelijk onvolledige datums. De reden om hier een apart voorstel voor op te stellen is dat beide voorstellen ('Basis datatypen' en 'Mogelijk onvolledige datum') als aparte wijzigingsverzoeken onafhankelijk van elkaar behandeld kunnen worden.

Het voorstel beschrijft niet hoe aangegeven kan worden wat de reden is dat een datum niet volledig bekend is. Het voorstel 'Geen waarde', ook van het project 'Utrecht', beschrijft hoe in het algemeen aangegeven kan worden wat de reden is dat een gegeven geen waarde heeft, ongeacht het datatype van dat gegeven. Dat voorstel kan toegepast worden op mogelijk onvolledige datums, maar dat is niet verplicht. Beide voorstellen kunnen onafhankelijk van elkaar worden toegepast en zijn ook als onafhankelijke wijzigingsverzoeken te behandelen. Toepassing van het voorstel 'Geen waarde' maakt het overigens niet mogelijk om aan te geven wat de reden is dat van een datum een deel ontbreekt. Het is alleen geschikt om aan te geven wat de reden is dat een mogelijk onvolledige datum helemaal geen waarde heeft.

Datatype voor mogelijk onvolledige datums

Een datatype voor mogelijk onvolledige datums is op de volgende manieren te definiëren:

1. Een samengesteld type (complexType) op basis van de XML-schema 'primitive datatypes', bijvoorbeeld op basis van xs:date, xs:gYearMonth en xs:gYear.
2. Een enkelvoudige type (simpleType) op basis van het XML-schema 'primitive type' xs:string, aangevuld met een 'pattern'.

Een voordeel van de eerste manier, een complexType, is dat wordt voortgebouwd op de bestaande specificaties van de XML-schema 'primitive datatypes'. Standaard XML-tooling kent en ondersteunt deze 'primitive datatypes' 'out of the box'. Denk hierbij aan onderwerpen als aantal dagen per maand en schrikkeljaren. Een nadeel is dat een dergelijk complexType niet altijd in één oogopslag te begrijpen is en daardoor als ingewikkeld wordt ervaren. De afbeelding van het datatype in het bericht op het datatype in de registratie kan in dit alternatief ook complex zijn.

Het voordeel van de tweede manier, een simpleType op basis van xs:string en een 'pattern', is dat het er eenvoudig uitziet. De afbeelding tussen het datatype in het bericht en het datatype in de registratie kan in dit alternatief eenvoudiger zijn dan in het voorgaande alternatief. Een nadeel is dat een dergelijk simpleType geen enkele 'kennis' van kalenders, jaren, maanden en dagen bevat en ook standaard XML-tooling geen 'out of the box' functionaliteit levert voor het verwerken van datums op basis van zo'n enkelvoudig datatype.

De verwachting is dat de voordelen van het kunnen hergebruiken van de XML-schema 'primitive datatypes' opwegen tegen de nadelen van meer complexiteit in het bericht en in de afbeelding op de registratie. Het voorstel is daarom om een samengesteld type te definiëren voor de mogelijk onvolledige datum. Dit complexType ziet er als volgt uit.

Naam	DatumMogelijkOnvolledig
Definitie	De keuze van een periode in de Gregoriaanse kalender, al naar

gelang de beschikbare datumelementen, uit de onderliggende subformaten datum, jaarMaand of jaar.

Toelichting Overeenkomstig ISO8601 zijn ook beperkt nauwkeurige perioden toegestaan, dus naast een volledige datum (bijv. 2014-08-27), ook alleen jaar en maand (2014-08) of alleen jaar (2014).

Specificatie Naam: DatumMogelijkOnvolledig
Keuze (minimaal en maximaal 1) uit de XML-Schema 'primitive datatypes':

- xs:date
- xs:gYearMonth
- xs:gYear

Korte schrijfwijze:

DatumMogelijkOnvolledig

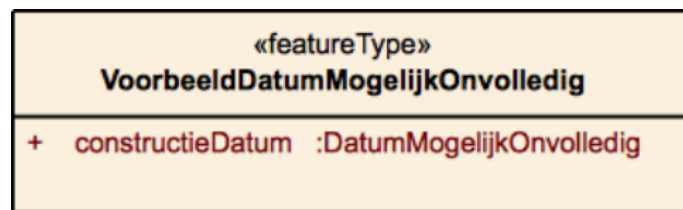
keuze (choice minOccurs=1 maxOccurs=1): Datum (date),

JaarMaand (gYearMonth), Jaar (gYear)

Voorbeelden



Afbeelding 1. UML class-diagram voor het complex type DatumMogelijkOnvolledig



Afbeelding 2. Voorbeeld van een UML class-diagram waarin het complex type wordt gebruikt.

Hieronder volgt een XSD-fragment dat hoort bij het UML class-diagram in Afbeelding 2.

```
<?xml version="1.0" encoding="UTF-8"?>
  <!-- voorbeeld xml schema implementatie van project utrecht voorstel
mogelijk onvolledige datum -->
  <schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:pu="http://www.projectutrecht.nl/test/2014/1.0"
targetNamespace="http://www.projectutrecht.nl/test/2014/1.0"
elementFormDefault="qualified" version="1.0.0beta1">
  <element name="VoorbeeldDatumMogelijkOnvolledig"
type="pu:VoorbeeldDatumMogelijkOnvolledigType" />
  <complexType name="VoorbeeldDatumMogelijkOnvolledigType">
    <sequence>
```

```

                <element name="constructieDatum"
type="pu:DatumMogelijkOnvolledigPropertyType" />
            </sequence>
        </complexType>
        <complexType name="VoorbeeldDatumMogelijkOnvolledigPropertyType">
            <sequence>
                <element ref="pu:VoorbeeldDatumMogelijkOnvolledig" />
            </sequence>
        </complexType>
        <element name="DatumMogelijkOnvolledig"
type="pu:DatumMogelijkOnvolledigType" />
        <complexType name="DatumMogelijkOnvolledigType">
            <choice>
                <element name="datum" type="date" />
                <element name="jaarMaand" type="gYearMonth" />
                <element name="jaar" type="gYear" />
            </choice>
        </complexType>
        <complexType name="DatumMogelijkOnvolledigPropertyType">
            <sequence>
                <element ref="pu:DatumMogelijkOnvolledig" />
            </sequence>
        </complexType>
    </schema>

```

Hieronder volgen voorbeelden van XML-fragmenten van respectievelijk een volledige datum, een onvolledige datum bestaande uit alleen een jaar en maand en een onvolledige datum bestaande uit alleen een jaar.

```

<?xml version="1.0" encoding="UTF-8"?>
  <!-- voorbeeld xml implementatie van project utrecht voorstel mogelijk
onvolledige datum -->
  <!-- situatie: datum is volledig bekend en opgenomen -->
  <!--Sample XML file generated by XMLSpy v2013 rel. 2 sp2
(http://www.altova.com)-->
  <pu:VoorbeeldDatumMogelijkOnvolledig
xsi:schemaLocation="http://www.projectutrecht.nl/test/2014/1.0
2OnVolDatum.xsd" xmlns:pu="http://www.projectutrecht.nl/test/2014/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <pu:constructieDatum>
        <pu:DatumMogelijkOnvolledig>
            <pu:datum>1967-08-13</pu:datum>
        </pu:DatumMogelijkOnvolledig>
    </pu:constructieDatum>
</pu:VoorbeeldDatumMogelijkOnvolledig>

```

```
<?xml version="1.0" encoding="UTF-8"?>
  <!-- voorbeeld xml implementatie van project utrecht voorstel mogelijk
onvolledige datum -->
  <!-- situatie: jaar en maand is bekend en opgenomen -->
  <!--Sample XML file generated by XMLSpy v2013 rel. 2 sp2
(http://www.altova.com)-->
  <pu:VoorbeeldDatumMogelijkOnvolledig
xsi:schemaLocation="http://www.projectutrecht.nl/test/2014/1.0
2OnVolDatum.xsd" xmlns:pu="http://www.projectutrecht.nl/test/2014/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <pu:constructieDatum>
        <pu:DatumMogelijkOnvolledig>
            <pu:jaarMaand>1967-08</pu:jaarMaand>
        </pu:DatumMogelijkOnvolledig>
    </pu:constructieDatum>
</pu:VoorbeeldDatumMogelijkOnvolledig>
```

```
<?xml version="1.0" encoding="UTF-8"?>
  <!-- voorbeeld xml implementatie van project utrecht voorstel mogelijk
onvolledige datum -->
  <!-- situatie: alleen jaartal is bekend en opgenomen -->
  <!--Sample XML file generated by XMLSpy v2013 rel. 2 sp2
(http://www.altova.com)-->
  <pu:VoorbeeldDatumMogelijkOnvolledig
xsi:schemaLocation="http://www.projectutrecht.nl/test/2014/1.0
2OnVolDatum.xsd" xmlns:pu="http://www.projectutrecht.nl/test/2014/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <pu:constructieDatum>
        <pu:DatumMogelijkOnvolledig>
            <pu:jaar>1967</pu:jaar>
        </pu:DatumMogelijkOnvolledig>
    </pu:constructieDatum>
</pu:VoorbeeldDatumMogelijkOnvolledig>
```

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.