XML Handboek

Versie
20160301.1

INHOUDSOPGAVE

VERSIEBEHEER

Versie	Datum	Auteur	Status
20090701.1	1 juli 2009	SIVI	Vervallen
20120401.1	1 april 2012	SIVI	Vervallen
20141101	1 nov 2014	SIVI	Vervallen
20160301	7 maart 2016	SIVI	Definitief

Wijzigingen ten opzichte van de vorige versie

Hoofdstuk/ Paragraaf	Aanpassing
Titelblad	Nieuw SIVI logo

1. Inleiding

1.1 Doelstelling

Deze handleiding beschrijft het gebruik van XML in samenhang met de SIVI-standaarden.

1.2 Doelgroep

Doelgroep	Upper Management	Midden / Lijn Management	Consultant, Ontwikkelaar e.d.
Verzekeraars[1]	Nee	Nee	Ja
Intermediairs	Nee	Nee	Ja
Leveranciers	Nee	Nee	Ja

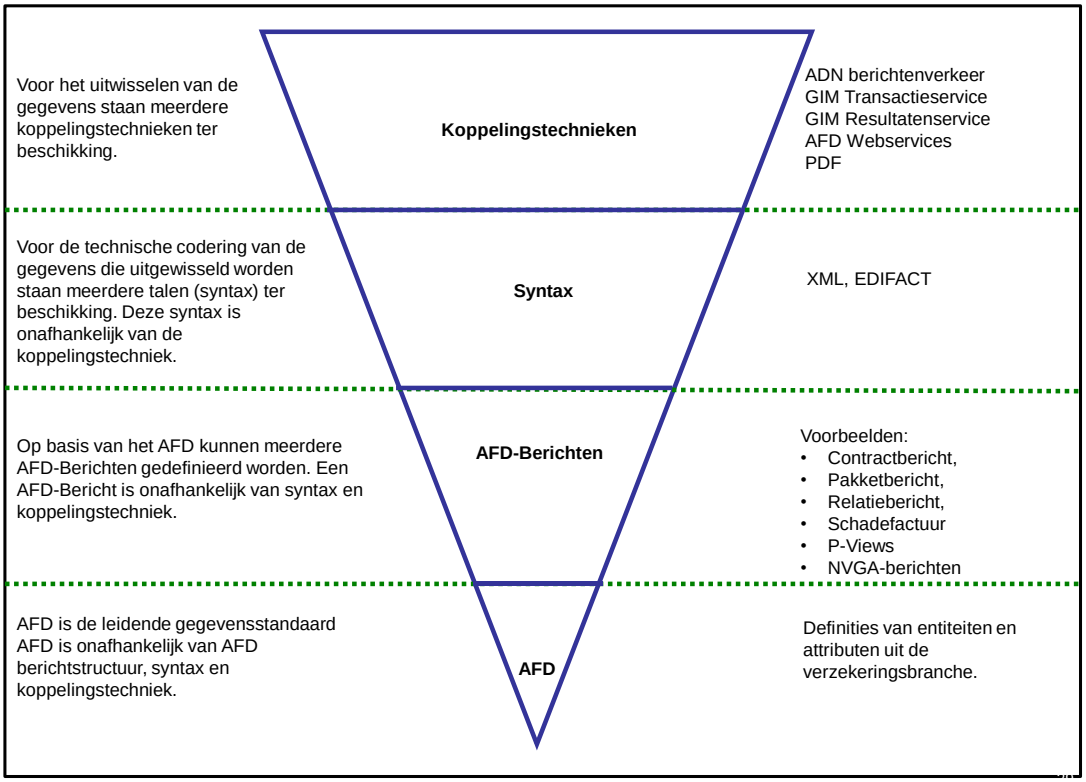
1.3 Opzet

Na deze inleiding wordt in hoofdstuk 2 een aantal basisconcepten behandeld. Hoofdstuk 3 bespreekt hoe XML in AFD-Berichten wordt toegepast. De hoofdstukken 4 t/m 6 gaan in op de toepassing van een aantal op XML gebaseerde technieken: XML-Schema, WSDL en SOAP.

[1] Verzekeraars en volmachten. Volmachten worden in het vervolg niet apart genoemd.

2. Belangrijke concepten

2.1 SIVI-piramide van standaarden



Het AFD en de onderkende AFD-berichten zijn belangrijke SIVI standaarden en zondermeer toekomstvast. De terreinen koppelingstechniek en in mindere mate syntax zijn in de tijd gezien meer aan verandering onderhevig.

Het ontkoppelen van de standaarden in meerdere lagen heeft als voordeel dat het AFD via meerdere berichtstructuren, via meerdere syntaxen en via meer koppelingstechnieken kan worden toegepast. Afspraken rond de berichtstructuur, syntax of koppelingstechniek hebben geen impact op het AFD.

2.2 XML

XML is de afkorting van Extensible Markup Language. XML is een computertaal (syntax) om gegevens op een gestructureerde manier elektronisch te communiceren.

Kenmerken:

- XML wordt onderhouden door de W3C.
- XML is breed geaccepteerd in de internetwereld als taal om applicaties met elkaar te laten communiceren en berichten elektronisch te versturen.
- XML is platform onafhankelijk.
- Op basis van XML zijn specifieke industriestandaarden ontwikkeld zoals SOAP, XML-Schema, XPath en WSDL.

3. AFD-Berichten in XML

3.1 Inleiding

Elk element binnen het AFD (entiteit of attribuut) bezit een eigen uniek ‘label’. Deze labels worden in XML gebruikt als XML-tags.

Entiteiten
Voorbeelden van entiteiten zijn Polis, Verzekeringnemer, Casco-dekking, Motorrijtuig. Iedere entiteit heeft een uniek label, bestaande uit 2 letters (bijvoorbeeld PP, VP, CA, OB).
Een entiteit heeft een of meer attributen.

Attributen
Binnen een entiteit bevinden zich de attributen. Voorbeelden van attributen zijn Contractnummer, Achternaam, Bedrag eigen risico, Soort brandstof. Ieder attribuut heeft een uniek label bestaande uit het label van de entiteit waar het attribuut toe behoort, gevolgd door een underscore (_) en gevolgd door een attribuutlabel bestaande uit maximaal 7 tekens (letters en/of cijfers).Voorbeelden zijn PP_NUMMER, VP_ANAAM.

Voorbeeld

```
<PP>
     <PP_NUMMER>123455667</PP_NUMMER>
</PP>
```

Dit voorbeeld geeft weer hoe de AFD-entiteit PP (Polis) en het AFD-attribuut PP_NUMMER (Contractnummer van een polis) in een XML-bericht worden opgenomen.

3.2 Label/waarde methode

Verwerkingscodes
Iedere entiteit in een AFD-Bericht bevat het attribuut ‘entiteitsverwerkingscode’(`VRWRKCD`).
De ontvanger kan op basis van deze verwerkingscode de entiteit correct verwerken.
De waarde van dit attribuut kan zijn:
0 : Nieuwe entiteit
1: Wijziging op bestaande entiteit
2: Verwijdering van een bestaande entiteit
3: De entiteit wordt ter identificatie gecommuniceerd
4: De entiteit wordt ter informatie gecommuniceerd

Voorbeeld in XML:

```
<PP>
     <PP_VRWRKCD>1</PP_VRWRKCD>
     <PP_NUMMER>123455667</PP_NUMMER>
</PP>
```

Waarde leeg maken
Om een waarde leeg te maken worden wordt een attribuut meegegeven zonder waarde.

Voorbeeld in XML:

```
<PP>
     <PP_VRWRKCD>1</PP_VRWRKCD>
     <PP_NUMMER>123455667</PP_NUMMER>
     <PP_BIJZBET></PP_BIJZBET>
</PP>
```

De “Omschrijving bijzondere betaling” (AFD label PP_BIJZBET) wordt hiermee ‘leeg’ gemaakt.

Niet opnemen van een attribuut binnen een gewijzigde entiteit
Wanneer een attribuut niet is opgenomen binnen een entiteit die wijzigt, blijft de waarde van dit attribuut aan de ontvangende kant ongewijzigd.

Voorbeeld in XML:

```
<PP>
     <PP_VRWRKCD>1</PP_VRWRKCD>
     <PP_NUMMER>123455667</PP_NUMMER>
</PP>
```

De “Omschrijving bijzondere betaling” (AFD label PP_BIJZBET) wordt niet gecommuniceerd en is dus niet veranderd aan de zendende kant.

3.3 Nesten van entiteiten

Entiteiten kunnen ‘genest’ voorkomen in een XML-bericht. Wanneer een entiteit is genest dan wordt deze omsloten door het open- en sluitgat van de bovenliggende entiteit.

Voorbeeld:
Een Polis (PP) met een Verzekeringnemer (VP) en een CASCO dekking (CA) wordt als volgt gecommuniceerd:

```
<PP>
     <VP>
     ......
     </VP>
     <CA>
     ......
     </CA>
</PP>
```

3.4 Volgorde van entiteiten

De volgorde van entiteiten in een XML-bericht is vrij. De enige uitzondering is dat binnen AFD-berichten de entiteit ‘Bericht Algemeen (AL)’ altijd als eerste entiteit moet worden opgenomen.

Er dient echter rekening gehouden te worden met het volgende:

XML-Schema
Een XML-Schema beschrijft de volgorde en nesting van entiteiten. Wanneer het XML-Schema de basis is voor een XML-bericht moet de volgorde van de entiteiten voldoen aan de definitie binnen het XML-Schema.

WSDL
Ook een WSDL beschrijft de volgorde en nesting van entiteiten. Wanneer de WSDL de basis is voor een XML-bericht moet de volgorde van de entiteiten voldoen aan de definitie binnen de WSDL.

3.5 Volgorde van attributen

De volgorde van attributen binnen een entiteit is in principe vrij te kiezen.

Er dient echter rekening gehouden te worden met het volgende:

XML-Schema
Een XML-Schema beschrijft de volgorde van attributen binnen de entiteiten. Wanneer het XML-Schema de basis is voor een XML-bericht moet de volgorde van de attributen voldoen aan de definitie binnen het XML-Schema.

WSDL
Ook een WSDL beschrijft de volgorde van attributen binnen entiteiten. Wanneer de WSDL de basis is voor een XML-bericht moet de volgorde van de attributen voldoen aan de definitie binnen de WSDL.

3.6 Tekensets

In XML-berichten wordt de UTF-8 tekenset gebruikt. Ook voor XML-berichten gebaseerd op AFD is er geen beperkte tekenset meer van toepassing.

3.7 Verpakken van AFD-berichten

AFD-berichten in XML-formaat worden voor het verzenden omsloten door XML-tags, die de berichtsoort aangeven. De volgende berichtsoorten zijn onder meer mogelijk:

- Los contract
- Een pakketpolis
- Een groep / deelnemerspakket
- Een relatie
- Een schade
- Batch-berichten

Voorbeeld: los contract

```
<Contractdocument>
    .....AFD Data.....
</Contractdocument>
```

Voorbeeld pakketpolis

```
<Pakket>
    <Mantel>.....AFD Data.....</Mantel>
    <Onderdeel>..... AFD Data.....</Onderdeel>
    <Onderdeel>..... AFD Data.....</Onderdeel>
</Pakket>
```

3.8 Bijlage in bericht

Het is mogelijk 1 of meerdere bijlagen binnen een bericht mee te sturen in de entiteit Bijlage (BY).

De entiteit Bijlage (<BY>) bevat minimaal de volgende attributen:

AFD label	Omschrijving	Formaat	Max.lengte
BY_VRWRKCD	Entiteitsverwerkingscode (waarde altijd 0)	string	1
BY_VOLGNUM	Volgnummer	decimal	3
BY_BYLSRT	Soort bijlage, code	string	2
BY_BYLOMS	Bijlage omschrijving	string	70
BY_EXT	Extensienaam van het bestand (Bijv. “PDF”, “DOC”, etc.).	string	70
BY_DATA	De bijlage als CodeBase64 datastring	string	Onbeperkt

Voorbeeld:

```
<BY>
    <BY_VRWRKCD>0</BY_VRWRKCD>
    <BY_VOLGNUM>1</BY_VOLGNUM>
    <BY_BYLSRT>11</BY_BYLSRT>
    <BY_BYLID>12345678</BY_BYLID>
    <BY_BYLOMS>Offerte Arbeidsongeschiktheidsverzekering dhr. Jansen
    </BY_BYLOMS>
    <BY_FILNM>Offerte12345678</BY_FILNM>
```

```
    <BY_EXT>PDF</BY_EXT>
    <BY_DATA>FSO:Hwo12S2jASHDasaKASJIgtaQtyiHzbsqq1lpfqsn........
    </BY_DATA>
</BY>
```

3.9 Namespaces

XML Namespace is een methode om een verzameling van elementen in een XML-bericht uniek te maken. Een namespace is in XML een ‘prefix’ die voor de XML-tag wordt geplaatst, waardoor de XML-tag ‘uniek’ wordt gemaakt binnen eenzelfde bericht.

Namespace is een W3C standaard, zie: http://www.w3.org/TR/REC-xml-names/

Er moeten namespaces gebruikt worden in AFD-berichten in de volgende gevallen:

- Bij pakketpolissen om ieder onderdeel uniek te maken.
- Bij entiteiten in een bericht als een entiteit meerdere keren met dezelfde naam (tag) en met verschillende inhoud kan voorkomen in een bericht.

De volgende regels zijn van toepassing:

- Namespaces worden door de maatschappij bepaald. De naamconventie van W3C dient te worden gehanteerd voor de namespaces.
- Het AFD-bericht mag alleen namespace(s) bevatten indien de maatschappij dit nodig acht. Als een maatschappij geen namespaces nodig heeft voor de verwerking, moet er ook geen namespace gebruikt worden.
- Een door het systeemhuispakket gevuld bericht bevat de door de maatschappij aangeleverde namespaces.

3.10 XPath

XPath is een techniek voor het adresseren van delen van een XML-bericht.
Zie voor de complete beschrijving van XPath: http://www.w3.org/TR/xpath/

In AFD-berichten wordt XPath gebruikt om in geval van een bijlage deze te koppelen aan een entiteit in het bericht waartoe de bijlage behoort. In geval van een foutmelding kan met XPath de melding gekoppeld worden aan het juiste element in het bericht.

Voorbeeld van XPATH in een bijlage entiteit:

```
<BY>
    <BY_VRWRKCD>0</BY_VRWRKCD>
    <BY_VOLGNUM>1</BY_VOLGNUM>
    <BY_BYLSRT>11</BY_BYLSRT>
    <BY_XPATH>Contractdocument/PP/VZ[1]</BY_XPATH>
    <BY_BYLID>12345678</BY_BYLID>
    <BY_BYLOMS>Offerte Arbeidsongeschiktheidsverzekering dhr. Jansen
    </BY_BYLOMS>
    <BY_FILNM>Offerte12345678</BY_FILNM>
    <BY_EXT>PDF</BY_EXT>
    <BY_DATA>FSO:Hwo12S2jASHDasaKASJIgtaQtyiHzbsq1lpfqsn
    </BY_DATA>
</BY>
```

4. XML-Schema

4.1 Inleiding

XML-Schema is een techniek om de specificatie van een bericht op een exacte manier vast te leggen. In een XML-Schema wordt o.a. de structuur van entiteiten en attributen vastgelegd.

Voorbeelden van toepassingen:
1) Verzekeraars kunnen XML-Schemas gebruiken bij de ontwikkeling van applicaties en webservices.
2) Leveranciers van intermediairsoftware kunnen XML-Schemas gebruiken om berichten te valideren voordat zij verstuurd worden of gebruikt worden om bijvoorbeeld een webservice aan te roepen.

Dit hoofdstuk beschrijft de richtlijnen waaraan een XML-Schema moet voldoen wanneer hiermee een AFD-Bericht wordt gedefinieerd.

SIVI levert geen XML-Schemas. Maatschappijen kunnen zelf schema's maken of dit uitbesteden aan een leverancier.

De informatie op basis waarvan XML-Schemas gemaakt kunnen worden ligt vast in de volgende AFD tabellen:
- Datacat.afm
- Codelijst.afm
- Hierarchie.afm

De regels waarin AFD-berichten moeten voldoen liggen vast in het handboek AFD-berichten.

4.2 Roottag van een AFD-bericht

Voorbeeld van de structuur van een AFD-bericht:

```
<Contractdocument>
     -----  AFD Data ------
</Contractdocument>
```

Tussen de zogenaamde roottags bevinden zich de AFD gegevenselementen (entiteiten en attributen).

4.3 XML-Schema tags

Een XML-Schema is op zich een XML-document en mag alleen de door de W3C voorgeschreven tags bevatten. In deze paragraaf volgt een beschrijving van deze tags.

Schema
De roottag voor van een XML-Schema is "schema". Alle tags worden altijd voorzien van een namespace, in het algemeen 'xsd'. Dit wordt ook in de roottag gedefinieerd.

Het XML attribuut "xmnls" definieert de namespace "xsd".

```
<xsd:schema xmlns:xsd=http://www.w3.org/2001/XMLSchema/>
```

De end-slash (/) betekent dat een sluittag niet volgt en ook niet nodig is.

Element
Om een AFD-gegevenselement te definiëren wordt de tag “element” gebruikt.

Een element kan zijn ComplexType of SimpleType.

ComplexType
Een entiteit uit het AFD wordt gedefinieerd als complex type. Een complex type kan zelf weer elementen bevatten (attributen of geneste entiteiten). Een complex type heeft eigenschappen (XML attributes):

- Het attribuut “name" definieert de tagnaam van het element.
- Het attribuut “minOccurs" definieert de minimale herhaling (minimum occurance). Deze heeft waarde 0 als een entiteit facultatief is en 1 als een entiteit verplicht is.
- Het attribuut "maxOccurs" definieert de maximale herhaling (maximum occurance).

```
<xsd:complexType>
<xsd:element name="……"  minOccurs="…."  maxOccurs="…..">
```

SimpleType
Een attribuut uit het AFD wordt gedefinieerd als simple type. Een simple type kan geen elementen in zich hebben. Een simple type heeft wel eigenschappen (XML attributes):

- Het attribuut “name" definieert de tagnaam van het element.
- Het attribuut “minOccurs" definieert het minimale herhaling (minimum occurance). Deze heeft waarde 0 als een attribuut facultatief is en 1 als een attribuut verplicht is.
- Het attribuut “maxOccurs" definieert de maximale herhaling (maximum occurance). Attributen kunnen niet worden herhaald, de waarde is daarom altijd 1.

```
<xsd:complexType>
<xsd:element name="……"  minOccurs="…."  maxOccurs="…..">
```

Sequence
Binnen een XML-Schema is het noodzakelijk om de AFD-elementen binnen de tag ‘sequence’ te plaatsen. Hiermee wordt de volgorde van de elementen bepaald. Men dient er dan ook voor te zorgen dat de elementen in een XML-bericht in de volgorde liggen zoals dat in het XML-Schema is bepaald.

```
<xsd:sequence>
```

Enumeration
Met de tag enumeration wordt een toegestane waarde van het AFD-attribuut gedefinieerd. Wanneer een AFD-attribuut een codelijst heeft kunnen de toegestane geldige codes met meerdere enumeration tags worden gedefinieerd.
Ook als er bij een attribuut slechts enkele bedragwaarden zijn toegestaan (bijvoorbeeld eigen risico bedrag 100, 200, 500) wordt dit met enumeration gedefinieerd.
Als bijvoorbeeld de waarden A, B or C, zijn toegestaan dan worden 3 enemuration tags opgenomen in het schema, ieder met één van deze waarden.

De tag enumeration kent het XML-attribuut value waarmee de mogelijke waarde wordt vastgelegd.

```
<xsd:enumeration value="…"/>
```

Attribute
Het begrip XML “Attribute" wordt niet gebruikt binnen AFD-berichten.

Annotation
De tag “annotation” wordt gebruikt om in een XML-Schema specifieke toelichting toe te voegen.
Het is toegestaan deze tag te gebruiken.

```
<xsd:annotation>
```

Documentation
De tag “documentation” wordt gebruikt om algemene toelichtende tekst toe te voegen.
Het is toegestaan deze tag te gebruiken.

```
<xsd:documentation>
```

Restriction
Ieder element is gebaseerd op een algemeen XML datatype zoals decimal, date, string etc.
Met de tag Restriction worden datatype en formaat van het attribuut gedefinieerd.
Met het XML attribuut ‘base’ wordt het datatype genoemd waarop het element is gebaseerd.
In navolgende tags worden de beperkingen gedefinieerd (bijvoorbeeld de lengte).

```
<xsd:restriction base="…..">
```

Mogelijke waarden voor het XML-attribuut 'base':
- String
- Decimal

MaxLength
Met de tag maxLength wordt het maximum aantal tekens gedefinieerd.

Het XML-attribuut 'value' geeft de maximale lengte.

```
<xsd:maxLength value=”..”/>
```

Voorbeeld
an..70 wordt <xsd:maxLength value=”70”/>

Length
Met de tag Length wordt het vaste aantal tekens gedefinieerd. Het attribuut heeft dan geen variabele lengte.

Het XML-attribuut 'value' geeft de lengte en het XML-attribuut 'fixed' geeft aan of het element een vaste lengte heeft.

```
<xsd:length value="…" fixed="true‟ />
```

Voorbeeld
n8 wordt <xsd:length value="8" fixed="true“ />

totalDigits
De tag totalDigits definieert het maximum aantal cijfers van een bedrag of aantal.
Dit is zonder minteken of punt.

Het XML-attribuut 'value' geeft het maximum aantal cijfers.

```
<xsd:totalDigits value="…."/>
```

fractionDigits
De tag fractionDigits definieert het maximaal aantal cijfers achter de decimale punt.
Het XML-attribuut 'value' geeft het aantal cijfers achter de decimale punt.

```
<xsd:fractionDigits value="…"/>
```

Pattern
De tag Pattern definieert het ‘patroon’ van een element.

```
<xsd:pattern value="…."/>
```

Voorbeeld:
<xsd:pattern value="[0-9]{1}"/>

Dit betekent dat het element bestaat uit 1 positie met de mogelijke waarde 0 t/m 9.

4.4 Van AFD datatype en formaat naar XML.

XML datatypes

In XML-berichten die van AFD zijn afgeleid mogen slechts een beperkt aantal datatypes worden gebruikt, namelijk alleen datatypes die ook binnen AFD bekend zijn.

Dat betekent dat de alleen volgende XML-datatypes toegestaan zijn:
- Type = "xsd:string" : voor alfanumerieke velden.
- Type = "xsd:decimal" : voor datum, tijd, bedragen, aantallen, percentages.
- Type = ”xsd:gYear” : voor jaartal aanduiding

Dat betekent ook dat o.a. de volgende XML-datatypes NIET zijn toegestaan:
- date
- time
- gYearMonth
- gMonthDay
- float
- double
- Boolean
- Notation
- Duration
- dateTime
- hexBinary
- qName
- base64Binary
- anyURI

AFD Datatypes

AFD kent een aantal datatypes. Het default type voor elk attribuut is “string”, dit kan worden overruled door een type op te geven, zie onderstaande tabel.
In de AFD tabel “datacat.afm” staat het type op positie 20-21.

Mapping van AFD datatype naar XML datatype:

AFD Datatype	XML Datatype	Omschrijving
Pn	<xsd:restriction base="xsd:decimal"> <xsd:totalDigits value="x"/> <xsd:fractionDigits value="n"/> </xsd:restriction>	Percentage met maximaal n decimalen. b.v. P5 = Percentage met maximaal 5 decimalen
Bn	<xsd:restriction base="xsd:decimal"> <xsd:totalDigits value="x"/> <xsd:fractionDigits value="n"/> </xsd:restriction>	Bedrag met maximaal n decimalen. b.v. B0 = Bedrag zonder decimalen
An	<xsd:restriction base="xsd:decimal"> <xsd:totalDigits value="x"/> <xsd:fractionDigits value="n"/> </xsd:restriction>	Aantal met maximaal n decimalen. b.v. A1 = Aantal met maximaal 1 decimaal
D1	<xsd:restriction base="xsd:string"> <xsd:length value="8"/> <xsd:pattern value= "(((\d{4})(0[13578]|10|12)(0[1-9]|[12][0-9]|3[01]))|((\d{4})(0[469]|11)([0][1-9]|[12][0-	AFD-Datum formaat EEJJMMDD Geen xsd:Date hanteren omdat deze scheidingstekens bevat.

AFD Datatype	XML Datatype	Omschrijving
	9]|30))|((\d{4})(02)(0[1-9]|1[0-9]|2[0-8]))|(([02468][048]00)(02)(29))|(([13579][26]00)(02)(29))|(([0-9][0-9][0][48])(02)(29))|(([0-9][0-9][2468][048])(02)(29))|(([0-9][0-9][13579][26])(02)(29)))?"/> </xsd:restriction>	
D3	<xsd:restriction base="xsd:gYear"> </xsd:restriction>	AFD-Datum formaat EEJJ
D5	<xsd:restriction base="xsd:string"> <xsd:length value="4"/> <xsd:pattern value= "(0[1-9]|1[012])(0[1-9]|[12][0-9]|3[01])"/> </xsd:restriction>	AFD-Datum formaat MMDD
D6	<xsd:restriction base="xsd:string"> <xsd:length value="6"/> <xsd:pattern value="[0-9]{4}((0[1-9])|(1[0-2]))"/> </xsd:restriction>	AFD-Datum formaat EEJJMM
T1	<xsd:restriction base="xsd:string"> <xsd:length value="4"/> <xsd:pattern value="([01]\d|2[0123])([0-5]\d)"/> </xsd:restriction>	AFD-Tijd formaat UUMM Geen xsd:Time hanteren omdat deze een scheidingsteken bevat.
JN	<xsd:restriction base="xsd:string"> <xsd:length value="1" fixed="true"/> <xsd:enumeration value="J"/> <xsd:enumeration value="N"/> </xsd:restriction>	Logische waarde J of N XML-datatype Boolean is iets heel anders en dus verboden.
RK	<xsd:restriction base="xsd:decimal"> <xsd:totalDigits value="10"/> </xsd:restriction>	11-proef voor bankrekeningnummer
--	<xsd:restriction base="xsd:string"> <xsd:maxLength value="x"/> </xsd:restriction>	Tekstveld zonder codelijst of waarde lijst
--	<xsd:restriction base="xsd:string"> <xsd:length value="x" fixed="true"/> <xsd:enumeration value="A"/> <xsd:enumeration value="B"/> <xsd:enumeration value="C"/> ..... </xsd:restriction>	Element met (subset) codelijst, of waardenlijst.

n = maximaal aantal cijfers achter de decimale punt.
x = (maximale) lengte van het hele veld.

Datatypes Pn en Bn
AFD attributen van het datatype Pn of Bn worden in een XML-Schema als volgt beschreven:

'n' in Pn and Bn betekent maximum aantal cijfers achter de decimale punt.
In een XML-Schema wordt dit gedefinieerd met de tag 'fractionDigits'
Een element van n..15 met het datatype B2 wordt:

```
<restriction base='decimal'>
    <totalDigits value='15'/>
    <fractionDigits value="2"/>
</restriction>
```

Formats/Length/MaxLength/MinLength
In AFD heeft elk attribuut een formaat. Deze formaten worden volgens onderstaande tabel omgezet naar XML schema restricties.
In de AFD tabel “datacat.afm” staat het formaat op positie 13-20.

Mapping van AFD formaten naar XML restrictions

Formaat	Voorbeeld	In XML
a	a..3	<restriction base="string">
an	an3	<restriction base="string">
n	n..15	<restriction base="decimal">
x	anx (bijv. an3)	<length value="3" fixed="true"/>
..x	an..x (bijv. an..15)	<maxLength value="15"/>
x	nx (bijv. n3)	<totalDigits value="3"/>
..x	n..x (bijv.. n..15)	< totalDigits value="15"/>

Voorbeelden
- an3 wordt <restriction base='string'><length value='3' fixed="true"/ </restriction>
- an..70 wordt <restriction base='string'><maxLength value='70'/></restriction>
- n..15 wordt <restriction base='decimal'><totalDigits value='15'/></restriction>

XML-Schema definitie
Om het gebruik van de AFD-datatypes binnen een XML-Schema te vereenvoudigen kunnen onderstaande type definities in het schema worden opgenomen. Een schema met dit type definities kan dan weer gebruikt worden bij het maken van bijvoorbeeld een webservice.

afdDate (voor AFD type D1)
```
<xsd:simpleType name="afdDate">
    <xsd:restriction base="xsd:string">
        <xsd:length value="8"/>
        <xsd:pattern value="(((\d{4})(0[13578]|10|12)(0[1-9]|[12][0-9]|3[01]))|
        ((\d{4})(0[469]|11)([0][1-9]|[12][0-9]|30))|((\d{4})(02)(0[1-9]|1[0-9]|
        2[0-8]))|(([02468][048]00)(02)(29))|(([13579][26]00) (02)(29))|(([0-9]
        [0-9][0][48])(02)(29))|(([0-9][0-9][2468][048])(02)(29))|(([0-9][0-9][13579]
        [26])(02)(29)))?"/>
        </xsd:restriction>
</xsd:simpleType>
```

afdYearMonth (voor AFD type D6)
```
<xsd:simpleType name="afdYearMonth">
    <xsd:restriction base="xsd:string">
        <xsd:length value="6"/>
        <xsd:pattern value="[0-9]{4}((0[1-9])|(1[0-2]))"/>
    </xsd:restriction>
</xsd:simpleType>
```

afdMonthDay (voor AFD type D5)
```
<xsd:simpleType name="afdMonthDay">
    <xsd:restriction base="xsd:string">
         <xsd:length value="4"/>
         <xsd:pattern value="(0[1-9]|1[012])(0[1-9]|[12][0-9]|3[01])"/>
    </xsd:restriction>
</xsd:simpleType>
```

afdTime (voor AFD type T1)
```
<xsd:simpleType name="afdTime">
    <xsd:restriction base="xsd:string">
        <xsd:length value="4"/>
        <xsd:pattern value="([01]\d|2[0123])([0-5]\d)"/>
    </xsd:restriction>
</xsd:simpleType>
```

In een XML-Schema wordt aan deze type definities gerefereerd als volgt (voorbeeld):
```
<xsd:element name="PP_INGDTW" type="afdDate" minOccurs="0"/>
```

Toestaan van 'leeg-waarden'
Teneinde het mogelijk te maken om lege numerieke waarden te kunnen communiceren, wordt het attribuut 'nillable' toegestaan.

Voorbeeld:

```
<xsd:element name="PP_CDUUMND" minOccurs="0"  nillable="true">
     <xsd:simpleType>
          <xsd:restriction base="xsd:decimal">
               <xsd:totalDigits value="3"/>
          </xsd:restriction>
     </xsd:simpleType>
</xsd:element>
```

In een bericht dient het weglaten van de waarde als volgt te worden doorgegeven:

```
<PP_CDUUMND xsi:nil="true"/>
```

MinOccurs/MaxOccurs
MinOccurs geeft het minimaal aantal herhalingen aan van een element.
MaxOccurs geeft het maximum aantal herhalingen aan van een element.

```
MinOccurs="0"  maxOccurs="2"
```

Als een element verplicht is, dan wordt minOccurs="1", maar omdat dit default is kan de tag worden weggelaten.

Als een element facultatief is, dan wordt minOccurs="0". Deze moet dan aanwezig zijn.

4.5 Entiteiten structuur

De vastlegging van de structuur van een AFD-bericht gaat op basis van de entiteiten die achter elkaar of onder elkaar (genest) worden gedefinieerd.

In een XML-Schema gaat dat als volgt:

Vast begin van een XML-schema:

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd=http://www.w3.org/2001/XMLSchema">
     <xsd:annotation>
          <xsd:documentation>    </xsd:documentation>
     </xsd:annotation>
```

Definitie van de roottag:

```
<xsd:element name="Contractdocument">
<xsd:complexType>
```

Definitie van een verplichte entiteit
Als een entiteit met entiteitlabel "XX" ("XX" is bijv.CA, WA) verplicht is, wordt dit als volgt beschreven:

```
<xsd:element name="XX">
     <xsd:complexType>
          <xsd:sequence>
          ……
          </xsd:sequence>
     </xsd:complexType>
</xsd:element>
```

Definitie van een facultatieve entiteit

```
<xsd:element name="XX" minOccurs ="0">
     <xsd:complexType>
          <xsd:sequence>
          ……
          </xsd:sequence>
     </xsd:complexType>
```

```
</xsd:element>
```

Definitie van een facultatieve entiteit die meerdere keren voor kan komen

```
<xsd:element name="XX" minOccurs="0" maxOccurs="n">
     <xsd:complexType>
          ……
     </xsd:complexType>
</xsd:element>
```

Definitie van een verplichte entiteit die meerdere keren voor kan komen

```
<xsd:element name="XX" maxOccurs="n">
     <xsd:complexType>
     ……
     </xsd:complexType>
</xsd:element>
```

De entiteit moet minimaal 1x voorkomen.

Definitie van een AFD attribuut met label XX_YYYYYYY

Definitie vindt plaats tussen de complexType tags.

```
<xsd:element name="XX_YYYYYY">
     <xsd:simpleType>
     </xsd:simpleType>
</xsd:element>
```

Definitie van een AFD attribuut met beperkte set waarden

De waarden zijn een subset van een aan het attribuut gekoppelde codelijst of reeks aan toegestane bedragwaarden (bv eigen risico bedragen).

```
<xsd:element name="XX_YYYYYYY">
     <xsd:simpleType>
          <xsd:restriction base="xsd:string"
          <xsd:enumeration value="XXX"/>
          <xsd:enumeration value="YYY"/>
          etc…..
          </xsd:restriction>
     </xsd:simpleType>
</xsd:element>
```

Definitie van een facultatief AFD attribuut

```
<xsd:element name="XX_YYYYYYY" minOccurs="0">
     <xsd:simpleType>
     …..
     </xsd:simpleType>
</xsd:element>
```

Definitie van een verplicht AFD attribuut.

```
<xsd:element name="XX_YYYYYYY">
     <xsd:simpleType>
     …..
     </xsd:simpleType>
</xsd:element>
```

minOccurs wordt weggelaten, default waarde is 1.

4.6 Regels XML-Schema & AFD-bericht

De regels om een XML-Schema op te zetten waarmee een AFD-bericht wordt gedefinieerd zijn als volgt:

1) Iedere entiteit uit het AFD wordt een ComplexType
2) Ieder attribuut uit het AFD wordt een SimpleType
3) Als een (child) entiteit genest wordt onder een andere (parent) entiteit wordt dit een XML element van het type ComplexType binnen deze (parent) entiteit.
4) Als een attribuut gekoppeld is aan een codelijst in AFD, dan worden de codes uit de codelijst opgenomen als enumerations. Uitzondering: Als het gaat om een zogenoemde Externe codelijst, worden de codes niet opgenomen in het XML-Schema (bijvoorbeeld ISO-Currencycode, ISO-Countrycodes)
5) Als een attribuut gekoppeld is aan een subset van een codelijst, dan worden alleen de subset waarden opgenomen in het XML-Schema.
6) Als bij een attribuut slechts enkele waarden zijn toegestaan, dan worden deze waarden opgenomen als enumerations.
7) Als een entiteit facultatief is krijgt het minOccurs="0".
8) Als een attribuut facultatief is krijgt het minOccurs="0".
9) Als een entiteit meerdere keren kan voorkomen krijgt maxOccurs het maximale aantal keren dat de desbetreffende entiteit herhaald kan worden.
10) Een attribuut kan niet herhalen en heeft dus altijd maximaal aantal herhalingen = 1.
11) Als een entiteit of een attribuut facultatief is, dan moet minOccurs=’0’ aanwezig zijn.

5. WSDL

5.1 Inleiding

Web Service Description Language (WSDL) is een taal waarmee de eigenschappen van webservices worden beschreven.

Bij een webservice hoort een WSDL. Een WSDL is een XML-bestand dat opgehaald kan worden vanaf een bepaalde locatie op het internet. Applicaties kunnen de WSDL inlezen en op basis daarvan de 'webservice class' genereren en daarmee de aanroep van de webservice zoveel mogelijk voorbereiden. De informatie hiervoor bevindt zich immers in het WSDL bestand. De applicatie kan vervolgens met de juiste gegevens de webservice class aanroepen. De webservice class roept op haar beurt de webservice aan over het internet en voert de gevraagde functie uit

WS-I (Web Services Interoperability) Basic Profile Version 1.0 schrijft het gebruik van WSDL op een bepaalde manier voor. SIVI raadt aan dit te volgen.
Zie voor WS-I Basic Profile: http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html

Voor bijna elke programmeertaal zijn hulpmiddelen beschikbaar waarmee WSDL-documenten op eenvoudige wijze gegenereerd kunnen worden, om ze zodoende via SOAP aanroepbaar te maken.
Zie voor een volledige beschrijving van WSDL: http://www.w3.org/TR/wsdl

5.2 WSDL

Hieronder volgt een overzicht van de belangrijkste mogelijkheden van WSDL.

1) Definitie Input/output elementen (types)
   In een WSDL worden de te gebruiken input- en output elementen, los van hoe ze in een bericht voorkomen, gedefinieerd.
2) Definitie Input/output berichten (message)
   In een WSDL worden de input- en output berichten beschreven op basis van de binnen de types sectie beschreven gegevenselementen.
3) Definitie Functies (operation)
   In een WSDL worden de mogelijk functies benoemd en wordt verwezen naar het te gebruiken inputbericht en het te gebruiken outputbericht.
   Binnen 1 webservice kunnen meerdere functies benoemd worden, ieder met een eigen inputbericht en outputbericht
   Meerdere functies kunnen gebruik maken van hetzelfde inputbericht en/of outputbericht.
4) Definitie Service locatie
   De service locatie geeft aan op welke internet lokatie (URL) de service te vinden is.
5) Definitie SOAP bericht (binding)
   In de WSDL is ook beschreven hoe het SOAP bericht eruit moet zien als de webservice op SOAP is gebaseerd. Dit is standaard voor de SIVI-protocollen.

5.3 Gebruik

AFD-Webservice en WSDL
Het gebruik van WSDL is voorgeschreven binnen de door SIVI beschreven standaard AFD Webservice protocol.

WSDL voor GIM Transactieservice en GIM Resultatenservice
Voor GIM Transactieservice en GIM Resultatenservice is een voorbeeld WSDL opgezet. Deze kan gebruikt worden bij de ontwikkeling van de in deze protocollen gedefinieerde (web)services. De WSDL is verkrijgbaar bij SIVI.

6. SOAP

6.1 Inleiding

Simple Object Access Protocol (SOAP) is een koppelingstechniek die wordt gebruikt voor communicatie tussen verschillende applicaties. SOAP wordt voornamelijk gebruikt bij het aanroepen van webservices. SOAP schrijft voor hoe de input en output er uit ziet (in XML) voor het aanroepen van een webservice.

SOAP is een protocol dat XML-berichten stuurt, meestal over HTTP, maar ook over SMTP, HTTPS of FTP.

SOAP is een W3C standaard, zie:
http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

Voor het gebruik van SOAP bestaat een WS-I binding profile:
http://www.ws-i.org/profiles/simplesoapbindingprofile-1.0.html

SOAP wordt in de volgende - door SIVI gestandaardiseerde protocollen - gebruikt:
1) GIM Resultaten Service
2) GIM Transactie Service
3) AFD Webservice

6.2 Structuur SOAP-bericht

SOAP schrijft voor de gegevens door middel van een SOAP envelop te communiceren. Een SOAP envelop bestaat uit een Header en een Body.

Header
- De Header bevat gegevens die nodig zijn voor de verwerking van de envelop.
- De SIVI protocollen waarin SOAP wordt gebruikt stellen (nog) geen eisen aan de Header. De Header kan dan ook weggelaten worden.

Body
- De Body bevat het bericht dat nodig is om de aangevraagde functie uit te kunnen voeren. De eerste XML-tag binnen de Body geeft de naam van de functie weer (bijvoorbeeld Premieberekening). In een WSDL noemt men dit ‘operation’, binnen een computerprogramma wordt dit vaak ‘method’ genoemd.
- Wanneer de gegevens correct verwerkt zijn, wordt het resultaat ook in de Body geretourneerd. De eerste tag is gelijk aan de originele functienaam met de toevoeging “Response”. (bijvoorbeeld PremieberekeningResponse).
- Wanneer een foutsituatie ontstaat, wordt er een foutmelding in de Body geretourneerd (Fault).

Algemene envelop structuur:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Header>

    </soap:Header>
    <soap:Body>
        <"Functienaam" xmlns="http://www.sivi.org/">
        ---  AFD-bericht   ----
        </"Functienaam" >
    </soap:Body>
</soap:Envelope>
```

Voorbeelden:
In onderstaande voorbeelden wordt een input-, output- en foutbericht weergegeven.

Input:

```
POST /SimpelServer/Service.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.sivi.org/Premieberekening"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
     <soap:Body>
          <Premieberekening xmlns="http://www.sivi.org/">
              <Contractdocument>
                   <AL>
                        <AL_FUNCTIE>08</AL_FUNCTIE>
                        <AL_BRON>AAC</AL_BRON>
                   </AL>
                   <PP>
                        <PP_NUMMER>1234567</PP_NUMMER>
                        <PP_INGDAT>20070401</PP_INGDAT>
                        <PP_BTP></PP_BTP>
                        <VP>
                        <VP_ANAAM>Vrolijk</VP_ANAAM>
                        <VP_GEBDAT>19640213</VP_GEBDAT>
                        </VP>
                   </PP>
              </Contractdocument>
          </Premieberekening>
     </soap:Body>
</soap:Envelope>
```

Output:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
     <soap:Body>
          <PremieberekeningResponse xmlns="http://www.sivi.org/">
               <Contractdocument>
                   <AL>
                        <AL_FUNCTIE>08</AL_FUNCTIE>
                        <AL_BRON>AAC</AL_BRON>
                   </AL>
                   <PP>
                        <PP_NUMMER>1234567</PP_NUMMER>
                        <PP_INGDAT>20070401</PP_INGDAT>
                        <PP_BTP>100</PP_BTP>
                        <VP>
                              <VP_ANAAM>Vrolijk</VP_ANAAM>
                              <VP_GEBDAT>19640213</VP_GEBDAT>
                        </VP>
                   </PP>
              </Contractdocument>
          </PremieberekeningResponse>
     </soap:Body>
</soap:Envelope>
```

Foutmelding

Een webservice geeft fouten door in een zogenaamd “Fault” element.

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
    <SOAP-ENV:Body>
        <SOAP-ENV:Fault>
            <faultactor/>
            <faultcode>SOAP-ENV:Server</faultcode>
            <faultstring>Service not available</faultstring>
        </SOAP-ENV:Fault>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```