

Implementatiehandleiding HL7v3 Basiscomponenten



Stichting HL7 Nederland
W. Barentszstraat 1
3902 DE Veenendaal

Telefoon : +31 (0)318 - 548869
Fax : +31 (0)318 - 548090
E-mail : info@hl7.nl

*Deze publicatie is mede tot stand gekomen
met ondersteuning van **Nictiz**.*

Versie : 2.3 NL
Datum : november 2013

Inhoudsopgave

1	Inleiding.....	5
1.1	Doel en doelgroep.....	5
1.2	Versie, status en wijzigingshistorie.....	5
1.3	Achtergrond.....	5
1.4	Reikwijdte.....	6
1.5	Structuur.....	6
1.6	Samenhang met andere documenten.....	6
1.7	Revisies.....	7
2	Uitgangspunten.....	11
2.1	Normatieve referenties.....	11
2.2	Informatieve referenties.....	11
2.3	Begrippen.....	11
3	Identificatiemechanismen.....	12
3.1	Inleiding.....	12
3.2	Object Identifiers (OID's).....	13
3.3	Uitgiftemechanisme OID's.....	14
3.4	Identificatiemethode.....	16
3.5	Identificatiesystemen OID Referentietabel.....	19
4	Vocabulaire voor berichtspecificaties.....	21
4.1	Inleiding.....	21
4.2	Vocabulaires in HL7v3.....	21
4.3	HL7 Vocabulary Domain Values.....	22
4.4	Vocabulaires OID-Referentietabel.....	26
5	OID gerelateerde implementatieaspecten.....	27
5.1	Inleiding.....	27
5.2	Gevolgen van OID's voor HL7-applicaties.....	28
6	Datatypes.....	29
6.1	Inleiding.....	30
6.2	Algemene toelichting op het gebruik van XML.....	30
6.3	XML-representatie van modelklassen en -attributen.....	31
6.4	Cardinaliteit, "mandatory" en conformance.....	34
6.5	Ontbrekende gegevens: nullFlavors.....	36
6.6	ANY (het generieke datatype).....	40
6.7	BL (Boolean – true/false).....	41
6.8	BIN (Binary data - binaire gegevens).....	42
6.9	ED (Encapsulated Data - ingekapselde gegevens).....	43
6.10	ST (String of Characters - reeks tekens).....	46
6.11	SC (String with Code - reeks tekens voorzien van code).....	47
6.12	CD (Concept Descriptor - gecodeerde gegevens).....	48
6.13	CE (Coded with Equivalents - gecodeerde gegevens, met equivalenten).....	49
6.14	CV (Coded Value - gecodeerde waarde).....	52
6.15	CO (Coded Ordinal - sorteerbare gecodeerde waarde).....	53
6.16	CS (Coded Simple - eenvoudige code).....	54
6.17	II (Instance Identifier - objectidentificatie).....	55
6.18	URL (Universal Resource Locator).....	58
6.19	TEL (Telecommunication Address - telecommunicatiecontact).....	59

6.20	AD (Postal Address - adres).....	61
6.21	EN (Entity Name - naam voor een entiteit).....	65
6.22	PN (Person Name - persoonsnaam)	66
6.23	ON (Organization Name - organisatienaam)	80
6.24	QTY (Quantity - hoeveelheid)	82
6.25	INT (Integer Number - geheel getal)	83
6.26	REAL (Real Number - getal met decimalen)	84
6.27	PQ (Physical Quantity - kwantitatieve weergave van fysieke grootheden).....	85
6.28	MO (Monetary Amount - hoeveelheid geld).....	87
6.29	TS (Time Stamp - tijdstip).....	88
6.30	BAG (Bag – verzameling met mogelijke dubbeln)	90
6.31	SET (Set – verzameling zonder dubbeln)	90
6.32	SXCM (Set Component – deelverzameling).....	90
6.33	IVL (Interval)	91
6.34	IVL_TS (Interval of Time Stamps - tijdsinterval)	93
6.35	IVL_INT (Interval of Integers – numeriek interval)	96
6.36	IVL_PQ (Interval of Physical Quantities - hoeveelheidsinterval).....	97
6.37	RTO (Ratio - onderlinge verhouding tussen twee gegevens)	99
6.38	GTS (General Timing Specification - generieke tijdspecificatie)	101
6.39	PIVL_TS (Periodic Interval of Time - periodiek herhalend tijdsinterval).....	104
7	CMET's.....	115
7.1	Inleiding	116
7.2	Algemene beschrijving van CMET varianten.....	116
7.3	R_PatientNL (patiënt)	118
7.4	E_PersonNL (persoon)	128
7.5	R_AssignedPerson (persoon in een rol).....	151
7.6	R_AssignedOrganization (organisatie)	155
7.7	R_AssignedDevice (apparaat).....	158
7.8	R_AssignedEntity (entiteit in een rol)	160
7.9	E_Organization (organisatie)	164
7.10	E_Place (plaats).....	168
7.11	R_LocationLocatedEntity (locatie).....	169
7.12	R_ResponsibleParty (verantwoordelijke partij)	171

1 Inleiding

1.1 Doel en doelgroep

De HL7 versie 3 [HL7v3] standaard beschrijft onder andere een reeks artefacten (componenten en structuren) die in vele berichten toegepast worden. CMET's en datatypen worden in elk HL7v3 bericht toegepast. De diverse HL7 versie implementatiehandleidingen bevatten tot nu toe elk een beschrijving van de daarin gebruikte datatypen en CMET's. Om herhaling te voorkomen maar ook om reden van consistentie werd in mei 2005 besloten om dit onderwerp in een aparte implementatiehandleiding onder te brengen.

Dit document heeft als doel de meestgebruikte datatypen en CMET's nader te verklaren en te specificeren voor toepassing in de Nederlandse zorgsector. Dit document dient gezamenlijk met de HL7 versie 3 standaard documentatie te worden gelezen. Dit document is opgesteld onder verantwoordelijkheid van de Technische Stuur Commissie van HL7 Nederland. Dit document is vooral bedoeld voor softwareontwikkelaars van zorgapplicaties en zorginfrastructurale applicaties, die op grond van de HL7 versie 3 communicatiestandaard en op grond van dit document hun berichtschema's en berichten willen definiëren. Deze gids is mede ontwikkeld om gebruikt te worden voor de implementatie van HL7v3 berichten binnen de landelijke basisinfrastructuur AORTA.

1.2 Versie, status en wijzigingshistorie

Dit is versie 2.3 NL van het document "Implementatiehandleiding HL7v3 Basiscomponenten". Deze versie van dit document is normatief voor alle documenten die normatief naar deze versie van dit document verwijzen. Deze versie van dit document is niet van toepassing op documenten die naar eerdere versies van dit document verwijzen.

De status van deze versie is "Definitief". Er wordt ten sterkste aanbevolen om indien mogelijk altijd gebruik te maken van de laatste, definitieve versie van dit document.

1.3 Achtergrond

Dit document bevat een aantal Nederlandse aanwijzingen bij het toepassen van HL7 versie 3. Dit document dient als toevoeging op (en niet als vervanging van) de internationale HL7 versie 3 materialen. In geval van tegenstrijdigheden tussen de internationale standaard en dit document geldt hetgeen bepaald is in deze richtlijn.

Dit document legt een aantal beperkingen op aan de vrijheden zoals deze in de internationale standaard bestaan; het is een "conformance profile". Partijen die versie 3 implementeren op basis van de internationale HL7 versie 3 standaard voldoen daarmee dus niet aan het "HL7 Nederland conformance profile". Zorgaanbieders en leveranciers die gegevens willen uitwisselen via de nationale infrastructuur moeten voldoen aan de "HL7 Nederland conformance profile".

Indien niet specifiek anders vermeld zijn in Nederland de vocabulaires (code tabellen) van toepassing zoals deze in de internationale standaard beschreven zijn. Indien u in de

Nederlandse situatie gebruikt wenst te maken van een HL7v3 datatype of CMET dat in strijd is met de Nederlandse richtlijnen, en dat niet in strijd is met de internationale HL7 versie 3 standaard, dan kunt u een voorbeeld gebruiksscenario (de use-case) aanmelden bij HL7 Nederland met een verzoek tot aanpassing van dit document.

1.4 Reikwijdte

Dit document biedt een fundament voor alle toepassingen van HL7 versie 3 binnen Nederland, met name binnen de nationale basisinfrastructuur AORTA. Het overstijgt een specifieke toepassing of project en dient te worden gezien als achtergrondinformatie, met een **bindend karakter** bij elke andere HL7 versie 3 implementatiehandleiding. Daar waar sprake is van overlap (en mogelijk daaruit voortkomende inconsistentie) tussen dit document en enige andere HL7 versie 3 implementatiehandleiding, dient dit document als leidend te worden gezien. Zodra een dergelijke inconsistentie bekend wordt, zal ernaar worden gestreefd zo snel mogelijk te komen tot harmonisatie.

1.5 Structuur

Dit document bestaat uit een beschrijving van de belangrijkste datatypen en CMET's, waarbij uitsluitend die onderdelen worden beschreven die in de Nederlandse situatie van toepassing zijn. Indien er specifieke aanwijzingen zijn hoe iets (wellicht in afwijking van de internationale standaard) in Nederland geïmplementeerd moet worden dan is dit in de tekst aangegeven.

HL7 artefacts worden in dit document aangeduid door hun officiële identificatie volgens de HL7 versie 3 ballot. De meeste van deze artefacts zijn conform ballot #7 van maart 2004, maar gaandeweg zijn ook onderdelen uit nieuwere ballots toegevoegd. In een later stadium zal waarschijnlijk integraal harmonisatie plaatsvinden met een actuele release. Enkele van de in deze implementatiegids beschreven HL7 artefacts bestaan (nog) niet in de internationale HL7 standaard. Deze artefacts worden in dit document in detail beschreven, omdat zij niet in het HL7 materiaal gedocumenteerd zijn. Als voorbereiding op een eventueel latere opname in de wereldwijde standaard zijn de namen van deze HL7 artefacts in het Engels gesteld. Aan nieuwe artefacts is een identificatie toegekend volgens de HL7v3 naam- en identificatieconventie. De (voorlopig) Nederland-specifieke artefacts zijn voorzien van een "NL" code. Alle nieuwe artefacts zullen ter beoordeling worden voorgelegd aan de internationale HL7 organisatie ter opneming in de internationale standaard. Als zij eenmaal zijn opgenomen komt de "NL" code te vervallen.

1.6 Samenhang met andere documenten

Diverse Nictiz documenten bevatten beschrijvingen voor datatypen en CMET's. Dit document is een nadere, niet context-afhankelijke, uitwerking van de in de Nictiz documentatie opgenomen aanbevelingen. Dit document is tevens aan de nieuwe inzichten aangepast.

Reeds gepubliceerde implementatiehandleidingen blijven gerelateerd aan de CMET's en datatypen, die in de implementatiehandleidingen opgenomen waren. Toekomstige publicaties zullen wel naar een versie van de implementatiehandleiding voor CMET's en datatypen gelinkt worden.

1.7 Revisies

De volgende wijzigingen zijn doorgevoerd sinds versie 2.3rc3 van 10 oktober 2011:

- Ballotcommentaren verwerkt, eerste ronde. Alle *persuasive (with mod)* commentaren zijn verwerkt. Op hoofdlijnen:
- Toelichting op vocabulaire herschreven en uitgebreid (§4.2)
- Datatype EN (§6.21) toegevoegd en in de specialisaties PN en ON de verbinding met EN gelegd.
- E_PersonNL aangepast zodat alleen meerdere namen worden toegestaan waar dat nodig is -- **NB: belangrijke wijziging!**
- E_Organization [contact] toegevoegd wegens gebruik in R_NotificationParty [contact] tevens de uitwerking van zowel R_ResponsibleParty [universal] als R_NotificationParty [contact] verbeterd
- E_Person [contact] bevat nu ook toelichting op LanguageCommunication
- Vele tekstuele verbeteringen

De volgende wijzigingen zijn doorgevoerd sinds ballotversie 2.3rc2 van 1 oktober 2011:

- Er zijn concepten toegevoegd in RoleCodeNL – zorgaanbiedertypen en de tekst onder de tabel is verduidelijkt. Het is nu duidelijker waar beheer ligt en dat de lijst een momentopname is en dat de actuele lijst bij Nictiz te verkrijgen is (§4.3)
- In de tekst onder RoleCodeNL – zorgverlenertypen is verduidelijkt dat de afgebeelde lijst een momentopname is en dat de actuele lijst in het CPS van het UZI-register staat (§4.3)

De volgende wijzigingen zijn doorgevoerd sinds ballotversie 2.3 van 10 december 2010:

- Inhoudsopgaven zijn opnieuw gegenereerd: deze waren onvolledig waardoor het leek of bepaalde paragrafen niet bestonden.
- Datatype ED (§6.9) ondersteunt nu het mediaType "vnd.oasis.opendocument.formula"
- Datatype SC (§6.11) maakt nu expliciet dat de string verplicht is en de code optioneel
- Datatype II (§6.17) heeft diverse verduidelijkingen waarin met name de verplichting op @extension is losgelaten. Gebruik van zowel @root en @extension blijft echter algemeen aanbevolen en het is mogelijk projectspecifieke @extension altijd te verplichten.
- Datatypes IVL (§6.33), IVL_TS (§6.34), IVL_INT (§6.35), en IVL_PQ (§6.36) hebben nu expliciete tekst over demotie waarmee een interval effectief gelijk wordt aan het basisdatatype (TS, INT, of PQ)
- Datatype IVL_TS (§6.34) heeft verbeterde voorbeelden waarin de tijdsaanduiding expliciet is gemaakt.
- E_PersonNL [universal] (§7.4)
 - De modelwijzigingen die leidden tot versie NL02 zijn teruggedraaid, omdat de impact ervan te groot was. Hierdoor zou er geen gebruik worden gemaakt van de nieuwe CMET-versie. De documentatie van versie NL01 welke wel overal wordt gebruikt zou alleen nog maar in versie 2.2 van dit document staan. De betreffende wijzigingen in E_Person (meerdere werknemerrollen, meerdere zorgverleners i.p.v. alleen een huisarts, doorgeven niet-preferente nummers e.a.) zullen in de grote harmonisatieslag van deze gids met de internationale ontwikkelingen terugkeren.
- R_AssignedPerson (§7.5)
 - Tekst iets generieker gemaakt zodat deze niet meer expliciet naar "zorg"medewerkers verwijst, maar toepasbaar is voor alle typen medewer-

- kers. Tevens een opmerking toegevoegd over het bekende gebruik van deze CMET in de Control Act-wrapper welke afwijkt van het doel van de CMET
- De lijst met wijzigingen onder “De volgende wijzigingen zijn doorgevoerd sinds versie 2.2 van 10 december 2009:” is aangepast zodat deze de cumulatieve stand bevat. Bij release van dit document zal alleen dat deel van deze paragraaf blijven bestaan.

De volgende wijzigingen zijn doorgevoerd sinds versie 2.2 van 10 december 2009:

- Algemeen
 - Referenties naar UCUM en IANA toegevoegd
 - RoleCodeNL – zorgaanbiedertype, nieuwe conceptcodes
 - Voorbeeld van VERWERKING toegevoegd (§6.4)
- Datatype ED (§6.9)
 - ‘encoding’ vervangen door ‘representation’ (technische correctie aangezien encoding als attribuut niet bestaat)
- Datatype SC (§6.11)
 - Het datatype AD gebruikt intussen al SC als basis voor adresdelen; dit stond hier nog als toekomstige actie
- Datatype II (§6.17)
 - Duplicaatinformatie over OID’s verwijderd en alleen verwijzing naar aparte OID-informatie laten staan
 - Uitzondering voor de verplichting op @extension toegevoegd voor het modelattribuut templateId welke meestal alleen @root heeft
 - Voorbeelden verbeterd door omschrijving toe te voegen per voorbeeld
- Datatype URL (§6.18)
 - Verwijzing naar IANA toegevoegd voor schemes
 - https in het lijstje van schemes opgenomen
- Datatype TEL (§6.19)
 - Gebruikstype (@use) ‘H’ toegevoegd voor secundaire contactadressen
- Datatype AD (§6.20)
 - Aanvullende informatie over gebruik van gecodeerde gemeenten toegevoegd
 - Aanvullende informatie over gebruik van gecodeerde postcodes toegevoegd
 - Mogelijkheid voor separaat doorgeven van huisnumertoevoegingen via buildingNumberSuffix toegevoegd
 - Richtlijn voor het doorgeven van postbusnummers en antwoordnummers toegevoegd in streetName
 - Definities van PostAddressUse codes PHYS, PST en HP verbeterd
 - Nieuwe PostAddressUse codes TMP (tijdelijke adressen) en CONF (vertrouwelijke/geheime adressen) toegevoegd
 - Implementatierichtlijn voor doorgeven van een geldigheidsperiode bij een adres via useablePeriod
 - Voorbeelden verbeterd door omschrijving toe te voegen per voorbeeld en extra voorbeelden toe te voegen
- Datatype PN (§6.22)
 - Opmerking over de status van een spatie tussen voorvoegsel en achternaam, welke géén delimiter is
 - Alle voorkomens van ‘geboortenaam’ en ‘eigenaam’ vervangen door ‘geslachtsnaam’
 - Voorbeeld toegevoegd met verschillende titels
 - Voorbeeld voor ‘Bert Kabbes, RI’ verbeterd. ‘, ’ was onterecht in de suffix opgenomen
- Datatype ON (§6.23)

- Geldigheidsperiode in het eerste, derde en vierde voorbeeld verbeterd op basis van verscherpte specificatie op IVL_TS
- Datatype IVL (§6.33)
 - Toegevoegd dat in intervallen met een onder- en bovengrens, deze niet aan elkaar gelijk mogen zijn, maar wel dezelfde precisie moeten hebben en tevens voorbeelden van correct en incorrect gebruik toegevoegd
- Datatype IVL_TS (§6.34)
 - De interpretatie van de precisie in intervallen is gewijzigd. Dit is in alle voorbeelden en teksten meegenomen. Dit heeft met name invloed op de wijze waarop een onder- en bovengrens wordt opgegeven
- Datatype IVL_PQ (§6.36)
 - Toegevoegd dat als onder- en bovengrens beiden worden opgegeven, deze dan in dezelfde eenheid moeten worden opgegeven en daar voorbeelden van toegevoegd
- Datatype PIVL_TS (§6.39)
 - Voorbeelden aangepast op basis van verscherpte interpretatie van IVL_TS
- Algemeen CMET's:
 - Alle koppen gecontroleerd en aangepast waar nodig
 - datatypen gekoppeld aan de beschrijving van dat datatype
 - teksten over opbouw van OID's verwijderd en vervangen door verwijzingen naar hoofdstuk 3
- R_Patient (§7.3.2)
 - Toegevoegd dat binnen Nederland tenminste het BSN moet worden gecommuniceerd
 - Voorbeeld van UZI-nummer toegevoegd onder AGB-Z voorbeeld
 - Tekst over niet-preferente nummers aangepast
- E_PersonNL [universal] (§7.4)
 - Toegevoegd: E_Person [contact] vanwege bekend gebruik, o.a. via R_ResponsibleParty (§7.12)
 - Gewijzigd: Een persoon kan nu in het R-MIM ook meerdere namen (0..*) hebben. Dit stond in de tekst al op 0..2, maar in het R-MIM stond het op 0..1. Intussen is er ook een use case voor meer dan twee namen geïdentificeerd.
 - Gewijzigd: verzekeringen via R_CoveredParty zijn op "niet-gebruiken" gezet. Deze was nog zonder implementatierichtlijn en kon daardoor sowieso niet eenduidig worden gebruikt. Er is momenteel geen gebruiksscenario bekend waarin communiceren van de verzekeringsgegevens wordt vereist. Verder is de huidige consensus dat verzekeringen een patiënteigenschap zijn en geen personeigenschap. Internationaal is dit dan ook voorbereid in het domein Patient Administration; het is echter nog niet normatief aanwezig in de betreffende CMET E_Person [universal] en wordt om die reden hier nog niet overgenomen.
Projecten of regio's die toch een scenario hebben waarin verzekeringsgegevens gecommuniceerd moeten worden, wordt aangeraden dicht bij de internationale ontwikkelingen te blijven bij het formuleren van implementatierichtlijnen in hun context. Meer informatie:
<http://www.hl7.org/v3ballot/html/welcome/environment/index.htm> Universal Domains > Patient Administration > Domain Message Information Model
<http://hl7book.net/index.php?title=Verzekeringen>
 - Toegevoegd: opmerking over gebruik van MaritalStatus code 'S' (never married)
- R_AssignedPerson (§7.5)
 - Tekst iets generieker gemaakt zodat deze niet meer expliciet naar "zorg" medewerkers verwijst, maar toepasbaar is voor alle typen medewerkers

kers. Tevens een opmerking toegevoegd over het bekende gebruik van deze CMET in de Control Act-wrapper welke afwijkt van het doel van de CMET

- E_Organization [universal] (§7.9.1)
 - R-MIM vervangen door de correcte versie (was [identified/confirmable])
- R_AssignedEntity (7.8)
 - Toegevoegd: R_AssignedEntity [identified] op basis van gebleken gebruik
 - Toegevoegd: R_AssignedEntity [universal] op basis van gebleken gebruik
- R_ResponsibleParty (§7.12)
 - Deze gehele paragraaf is toegevoegd op basis van gebleken gebruik
- Wijzigingen overig: tekstuele verbeteringen

2 Uitgangspunten

2.1 Normatieve referenties

De onderstaande documenten zijn beschouwd als leidend voor dit document:

Referentie	Document(en)	Bron
[HL7v3]	HL7 Version 3 Ballot 11, September 2005 Normative Edition 2005	http://www.hl7.org
[XMLSC]	World Wide Web Consortium. XML Schema.	http://www.w3.org/TR/xmlschema-0/ http://www.w3.org/TR/xmlschema-1/ http://www.w3.org/TR/xmlschema-2/
[XML]	World Wide Web Consortium. Extensible Markup Language, 1.0, 2nd Edition.	http://www.w3.org/TR/REC-xml

2.2 Informatieve referenties

De onderstaande documenten hebben gediend als bron voor dit document:

Referentie	Document(en)	Bron
[NIInfraDom]	Nictiz Implementatiehandleiding generieke berichten v3.0 (AORTA v5.0.0.0)	http://www.nictiz.nl
[NIWebSvPrf]	Nictiz Implementatiehandleiding berichttransport v2.0 (AORTA v5.0.0.0)	http://www.nictiz.nl
[NIWrappers]	Nictiz Implementatiehandleiding berichtwrappers v3.0 (AORTA v5.0.0.0)	http://www.nictiz.nl
[IANA - schemes]	Internet Assigned Numbers Association (IANA)	http://www.iana.org/assignments/uri-schemes.html
[UCUM]	Unified Codes for Unit of Measure	http://unitsofmeasure.org/

2.3 Begrippen

Overal in dit document waar de voornaamwoorden "hij", "hem" of "zijn" staan, wordt "hij of zij" respectievelijk "hem of haar" respectievelijk "zijn of haar" bedoeld.

3 Identificatiemechanismen

3.1 Inleiding

Standaardiseren van gegevensuitwisseling betekent enerzijds het afspreken van een berichtstructuur (grammatica), maar het betekent ook dat dezelfde aanduiding voor objecten en concepten gehanteerd moet worden. Het heeft immers weinig zin om precies af te spreken hoe (qua formaat en betekenis) een patiënt wordt aangeduid, als de identificatie van die patiënt vervolgens niet wordt begrepen door de ontvangende partij.

Binnen een instelling is het meestal wel duidelijk welke 'stamtabel' bedoeld wordt voor identificatie en codering. Als een opname binnen een ziekenhuis wordt uitgewisseld, dan weten alle ontvangers dat de opnamenummers degene zijn die zijn uitgegeven door het ZIS. Als een arts wordt geïdentificeerd, dan zal bekend zijn of dit gebeurt met lokale nummers óf eventueel met landelijke Vektis-nummers. Van laboratoriumbepalingen zal bijvoorbeeld bekend zijn dat de onderzoeken worden aangeduid met lokaal gedefinieerde codes.

Dit geldt zeer waarschijnlijk niet in een transmuraal geïntegreerd zorgnetwerk, waar zender en ontvanger elkaars context niet per definitie 'kennen'. In zo'n setting kunnen er namelijk geen aannames meer gedaan worden over de aard van de gebruikte identificatie over codering. Die moet dus expliciet worden doorgegeven. Een tweetal voorbeelden:

Regionale gegevensuitwisseling ziekenhuizen:

- Er worden kruistabellen bijgehouden van de nummers van een patiënt bij verschillende ziekenhuizen (MPI functie).
- Patiëntnummer 3268774 wordt (bijvoorbeeld bij onderlinge dienstverlening) verzonden van ziekenhuis A naar B.
- Bedoelt ziekenhuis A nu een eigen nummer of heeft het een nummer van ziekenhuis B gebruikt?

Landelijke aanvraag van labonderzoeken:

- Een ziekenhuis stuurt een aanvraag voor onderzoek 'B345' naar een landelijk opererend laboratorium.
- Bedoelt de aanvrager nu een eigen code, een code uit de specifieke tabel van het laboratorium of een algemene code?

Het eerste voorbeeld zal na invoering van het BSN niet (meer) spelen, omdat patiënten dan landelijk uniek identificeerbaar zijn. Maar hetzelfde probleem speelt ook bij de identificatie van bijvoorbeeld medicatieverstrekkingen of andere brongegevens in het (landelijk) EPD. Het blijft dus nodig om identifiers en codes uniek aan te duiden.

Er is overigens vaak verwarring over het onderscheid tussen de termen *identifier* (ID) en *code*. De kreten worden soms door elkaar gebruikt, terwijl hun betekenis fundamenteel verschilt:

- Een ID (identifier) duidt een **specifiek object** aan: een bepaalde patiënt, een ziekenhuis, een arts, een labaanvraag, een röntgenfoto, een medicatieverstrekking. Kortom, iets of iemand waarvan er maar één is en waar je als het ware een volgnummer aan kunt hangen.
- Een code duidt een **generiek concept** aan: een soort patiënt, een type zorgaanbieder, een artstype, een soort labonderzoek, een medicatietype. Hier gaat het niet om één 'object', maar om een categorie objecten die bepaalde kenmerken met elkaar gemeen hebben.

Merk op dat als men het dus heeft over 'de artscode' van dr. Jansen, dit feitelijk onjuist is. Het gaat dan immers om de identificatie van dr. Jansen als 'object' (oftewel individu) en 'het artsnummer' zou dan ook de juiste benaming zijn.

3.2 Object Identifiers (OID's)

Object Identifiers (OID's) is een concept dat wordt gebruikt door ISO (de wereldwijde standaardisatie-organisatie) om te komen tot unieke identificatie van een systeem waarbinnen zelf weer identifiers of codes worden uitgedeeld en beheerd.

Het achterliggende idee daarbij is dat elke identificatie en elke code onderdeel is van een systeem waarbinnen deze gedefinieerd is, een identificatie- respectievelijk codesysteem, bijvoorbeeld:

- Patiëntnummers van ziekenhuis ABC
- Zorgverleneridentificatie o.b.v. AGB-Z
- Laboratoriumbepalingen volgens LOINC

In al deze voorbeelden is sprake van een identifier (eerste twee gevallen) respectievelijk een code (laatste situatie) en van een systeem waarbinnen deze worden uitgedeeld en beheerd. In het eerste geval is het een ziekenhuis dat de nummers uitdeelt (of liever: een softwaresysteem dat door het ziekenhuis wordt gebruikt). In het tweede geval gaat het om een landelijke zorgverlenertabel (beheerd door VEKTIS) en in het derde geval zelfs om een internationaal beheerd systeem voor het coderen van laboratoriumbepalingen.

De truc is nu om dat identificatie/codesysteem zelf een unieke identificatie te geven: deze identificatie is de **OID van het identificatie- of codesysteem**. Als de OID van het systeem **globaal uniek** is en de identificatie of code is uniek *binnen* dat systeem, dan is de combinatie van die twee dus een unieke aanduiding voor de identificatie/code.

De achterliggende systematiek zorgt ervoor dat het identificatie- of codesysteem een OID heeft die gegarandeerd **wereldwijd en eeuwig** uniek is. Met andere woorden: een OID is niet alleen uniek, maar ook persistent. Er zal nooit eenzelfde OID worden toegekend aan een ander identificatie- of codesysteem.

Dit wordt later toegelicht.

Een aardige analogie is die met nummertjesautomaten, waarin bonnetjes zitten die uniek zijn voor de betreffende automaat. Als ik nu een bonnetje trek uit twee verschillende automaten, dan zou het nummer daarop best gelijk kunnen zijn. Maar stel nu dat de automaat zelf een sticker met daarop een OID bevat. De OID is dan dus een unieke identificatie van de nummertjesautomaat zelf. De combinatie van de OID van de automaat met het nummer op het bonnetje is nu gegarandeerd wereldwijd en eeuwig uniek.

In het geval van de identificatie van personen, organisaties of andere stoffelijke zaken wordt gebruik gemaakt van bestaande (HL7-externe) identificatiesysteem (zoals bijvoorbeeld Nederlands Rijbewijsnummer, SOFI-nummer, Ziekenhuisnummer binnen het St. Josef Ziekenhuis, UZI nummer van de zorgverlener). Ieder identificatiesysteem (Engels: identification scheme) wordt op zijn beurt uniek geïdentificeerd door een bepaalde OID.

In het geval van vocabulaires/terminologieën heeft een bepaalde organisatie (zoals de WHO, HL7 zelf, Z-Index, Prismant, SNOMED) één of meer vocabulaires (codesystemen/"tabellen") ontwikkeld. Iedere tabel wordt eenduidig geïdentificeerd door een bepaalde OID.

Voorbeeld: 2.16.840.1.113883.2.4.6.12 is de OID voor het identificatiesysteem van Nederlandse rijbewijzen. Het Rijbewijsnummer van Jan Janssen is 93632988. Dan wordt de persoon Jan Jansen eenduidig geïdentificeerd door de combinatie van a) de OID van het identificatiesysteem (2.16.840.1.113883.2.4.6.12) en b) de extensie daarvan (Engels: extension), een nummer binnen het identificatiesysteem (hier 93632988).

In een HL7 versie 3 bericht ziet dit er als volgt uit:

```
<id root="2.16.840.1.113883.2.4.6.12" extension="93632988"
    assigningAuthorityName="Rijksdienst voor het wegverkeer"/>
```

Voorbeeld: 2.16.840.1.113883.5.1 is de OID voor het HL7 codesysteem *AdministrativeGender*. Binnen die tabel staat de waarde "M" voor 'Mannelijk'. Dan wordt het concept 'Mannelijk' eenduidig geklassificeerd door de combinatie van a) de OID van het codesysteem (2.16.840.1.113883.5.1) en b) een code binnen dat systeem (hier "M").

In een HL7 versie 3 bericht ziet dit er als volgt uit:

```
<cd code="M" codeSystem="2.16.840.1.113883.5.1"
    codeSystemName="AdministrativeGender" displayName="Male"/>
```

3.3 Uitgiftemechanisme OID's

De opbouw van OID's is gebaseerd op het principe van **gedelegeerde verantwoordelijkheid**, d.w.z. dat de instantie die een OID beheert kan bepalen dat een andere organisatie zelf weer OID's mag uitdelen *onder* de OID die hen is toegewezen.

Alle OID's worden uitgedeeld volgens een **boomstructuur**, met **knopen (nodes) en takken**, zodat een hiërarchie ontstaat. Het allereerste niveau is ooit bepaald door ISO, dat daaronder nodes heeft toegewezen aan allerlei (categorieën van) organisaties.

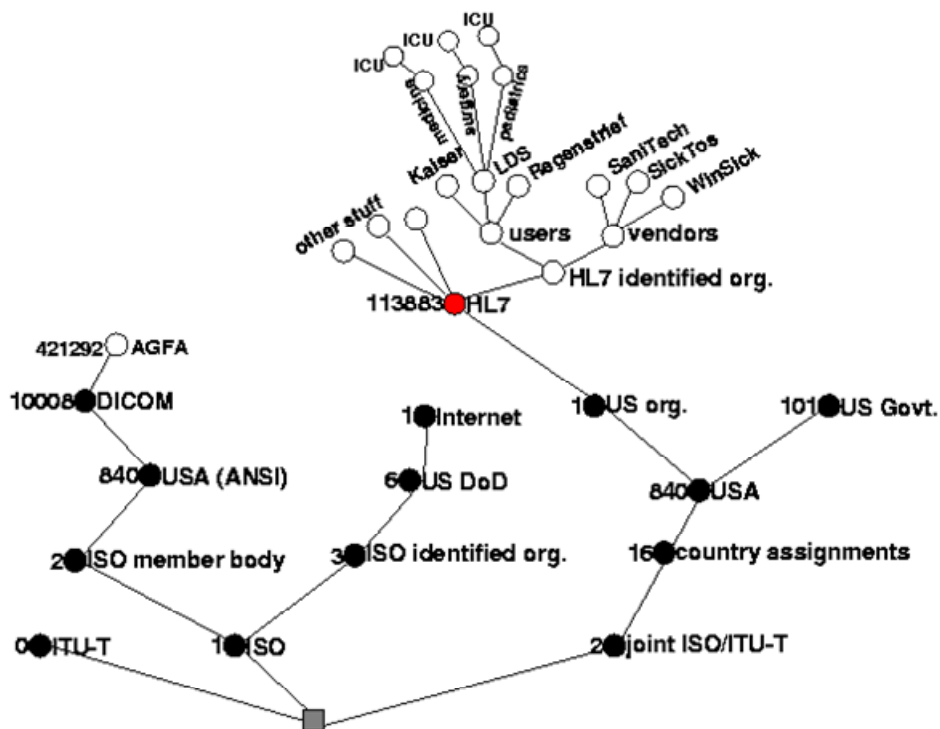
Nodes horen bij een '**assigning authority**' die een identificatie- of codesysteem beheert (bijvoorbeeld HL7 zelf, VEKTIS of een specifieke zorgaanbieder). Deze assigning authority kent sub-OID's (oftewel nieuwe takken) toe onder de 'eigen' root node.

Bij elke nieuwe tak wordt een assigning authority geacht te controleren of er al een OID bestaat voor hetzelfde identificatie- of codesysteem. Op die manier wordt zoveel mogelijk voorkomen dat voor hetzelfde systeem (dezelfde 'nummertjesautomaat') meer dan één OID gebruikt wordt. Dit aspect is echter niet 100% waterdicht, omdat er geen centrale registratie bestaat. Het gevolg daarvan is dat het zou kunnen dat er meerdere OID's voor

één identificatie- of codesysteem bestaan. Het blijft echter zo dat een OID altijd een **unieke aanduiding voor één specifiek systeem** is (en dus als basis kan dienen voor identifiers of codes die binnen dat systeem worden uitgedeeld en beheerd).

Een OID is altijd een unieke aanduiding voor één specifiek identificatie- of codesysteem.
Een OID kan nadat zij is toegewezen nooit meer aan iets anders worden toegewezen.
Een OID blijft ook geldig en bruikbaar **als de assigning authority niet meer bestaat**.
Het gaat er immers om dat de OID persistent is, dus ook als een andere assigning authority het beheer overneemt, moet de OID voor alles onveranderd blijven.

Teneinde het gebruik van OID's in HL7 berichtimplementaties te vergemakkelijken hebben de internationale HL7 organisatie en HL7 Nederland elk een OID root bij ISO verkregen. Binnen deze OID roots kan HL7 op verzoek OID's uitdelen.



Figuur 15: OID boomstructuur

Een OID bestaat uit een keten van nodes. De nodes zijn **getallen met punten ertussen**: n1.n2.n3.n4. Er is in principe geen beperking aan de lengte die een OID kan hebben (hoewel in de praktijk vaak een maximum van 100 posities wordt aangehouden).

Een OID node mag **geen voorlooppnullen** bevatten, hoewel een node wel gewoon uit het getal 0 mag bestaan. Dus 1.01.2567 is geen geldige OID, maar 1.1.2567.0 is dat wel.

Een OID is slechts bedoeld om de uniciteit te borgen en de **structuur van de OID heeft geen betekenis**. OID's mogen dus niet opgedeeld worden, bijvoorbeeld om te bepalen welke assigning authorities betrokken zijn bij het ontstaan van de keten van nodes.

Elke tak onder een root in de boomstructuur correspondeert met een domein waarbinnen een assigning authority de uitgave van OID's beheert. Indien een OID binnen deze tak

wordt uitgegeven aan een andere organisatie, dan vormt die OID de OID root van de daaraan gerelateerde organisatie (die weer een assigning authority kan zijn).

Voor meer informatie over OID's, zie de site www.oid-info.com, waar veel OID's terug te vinden zijn (op basis van actieve registratie of door koppelingen met andere registers).

3.4 Identificatiemethode

Binnen HL7 berichten is een identificatie systeem noodzakelijk waar het gaat om personen, systemen, instellingen en andere stoffelijke zaken. Daarbij wordt gebruik gemaakt van bestaande (HL7-externe) identificatiesystemen (zoals bijvoorbeeld 'Nederlands rijbewijsnummer', SOFI-nummer, Ziekenhuisnummer binnen het St. Josef Ziekenhuis, UZI nummer van de zorgverlener) die op zijn beurt uniek geïdentificeerd wordt door een bepaalde OID.

Hieronder volgt per gegevenscategorie een tekstuele toelichting van de te gebruiken identificatiemethoden, alfabetisch gesorteerd op gegevenscategorie. Na deze lijst volgt een tabel met alle aangehaalde OID's.

Berichten: Een HL7 versie 3 bericht. Berichten dienen te worden voorzien van een uniek nummer door de zendende softwareapplicatie.

Indien 2.16.840.1.113883.2.4.6.6.945 de OID van de zendende applicatie is (Zie *softwareapplicatie* in deze lijst), dan mag deze applicatie binnen de OID-tak 2.16.840.1.113883.2.4.6.6.945 zelf unieke nummers uitdelen. Als YYYY een oplopend nummer per bericht is, en de applicatie de OID tak '1' gebruikt om berichten verzonden door de applicatie te identificeren, dan vormt 2.16.840.1.113883.2.4.6.6.945.1 met extensie YYYY een unieke identificatie van het bericht.

Voorbeeld: De applicatie met OID 2.16.840.1.113883.2.4.6.6.940 staat op het punt een nieuw bericht te verzenden. Het te verzenden bericht krijgt het unieke nummer 8700333 (dit nummer mag slechts één keer worden gebruikt gedurende de levensduur van de zendende applicatie).

De applicatie kan binnen haar eigen OID-tak zelf unieke nummers uitdelen. Als 8700333 het te verzenden bericht aangeeft, en de organisatie er voor heeft gekozen OID tak 1 te gebruiken om berichten te identificeren, dan vormt 2.16.840.1.113883.2.4.6.6.940.1 met extensie 8700333 een unieke identificatie van het bericht.

```
<id root="2.16.840.1.113883.2.4.6.6.940.1" extension="8700333"
    assigningAuthorityName="ApplicatieNaam"/>
```

Softwareapplicatie: Met een softwareapplicatie wordt een applicatie bedoeld, die als 'fysieke' zender of ontvanger van berichten optreedt, eventueel namens functionele applicatiemodules binnen de softwareapplicatie. Hieronder vallen de diverse XIS-systemen.

Softwareapplicaties worden door middel van een door de verantwoordelijke organisatie uit te geven OID geïdentificeerd. Dit is mogelijk door of een OID af te leiden uit de OID van de organisatie die de applicatie beheert, of door het aanvragen van een aparte OID bij HL7 Nederland voor de applicaties binnen de

eigen instelling. De identificatie heeft in principe geen enkele relatie met het UZI nummer van het systeemcertificaat van een GBZ; een relatie kan alleen aan de hand van een koppeltabel gelegd worden.

Indien de applicatie een GBZ is (of een deel daarvan) dan wordt op moment van aansluiting van de GBZ een identificerende OID door de LSP uitgegeven en in de ZIM vastgelegd.

Voorbeeld: indien de OID 1.2.3.4.26 met extensie 22 een softwareapplicatie identificeert (zie hierboven voor een definitie van softwareapplicatie), dan mag deze softwareapplicatie binnen de OID 1.2.3.4.26.22 zelf unieke nummers uitdelen. Als YYYY een oplopend nummer per softwaremodule binnen de softwareapplicatie is, en de OID-tak 56 wordt gebruikt ter identificatie van modules in de softwareapplicatie, dan vormt 1.2.3.4.26.**22.56** met extensie YYYY een unieke identificatie van de softwaremodule.

FAQ: Gegeven dat elke softwareapplicatie een Systeempas met een certificaat voorzien van een uniek UZI nummer verkrijgt voor authenticatiedoeleinden, kan dat UZI nummer niet hergebruikt worden als unieke identificatie van de softwareapplicatie? Nee, systeemcertificaten dienen periodiek (elke drie jaar) te worden vernieuwd en verkrijgen dan een nieuw UZI nummer. Om deze reden wordt de identificatie van softwareapplicaties door middel van het UZI nummer van de Systeempas afgeraden.

FAQ: De identificerende OID van een applicatie kan vrijelijk worden gedefinieerd. Een uitzondering wordt gevormd door een GBZ en applicaties binnen een GBZ die voor communicatiedoeleinden bekend zijn bij de verwijzindex. Deze verkrijgen, na aanmelding en registratie bij de LSP, een identificerende OID die verplicht is in alle communicatie met de landelijke infrastructuur. De root van de door de LSP uitgegeven identificaties is 2.16.840.1.113883.2.4.6.6.

NOOT: Bovenstaande identificatiemethode voor applicaties is alleen van toepassing in de context van aansluiting op het Landelijk Schakelpunt in de infrastructuur van Nictiz.

Patiënt/cliënt: Persoon die zorgservices afneemt van zorgverleners of zorgverlenende instellingen. Patiënten worden landelijk uniek geïdentificeerd door middel van het burgerservicenummer (BSN).

De OID van het BSN identificatie systeem is: 2.16.840.1.113883.2.4.6.3. Naast de BSN-identificatie (die indien deze bekend is in ieder geval verzonden moet worden) kan gebruik worden gemaakt van het identificatienummer zoals bekend binnen een zorgregio of de instelling zelf.

Voorbeeld: Een patiënt heeft BSN 700434 en de locale (instellingsinterne) identificatie 830760. Er van uitgaande dat het HL7-bericht een identificerend data-element bevat wat herhalend voor mag komen, kan dit er in het bericht als onderstaand voorbeeld uitzien. Indien de OID 2.16.840.1.113883.2.4.6.1.100702 de identificatie van de organisatie is, dan mag een applicatie binnen haar OID-tak zelf unieke nummers uitdelen. Als 830760 een organisatie-intern patiëntidentificatie is, en de organisatie er voor heeft gekozen OID tak 12 te gebruiken om patiënten te identificeren, dan vormt 2.16.840.1.113883.2.4.6.1.100702.**12** met extensie **830760** een unieke identificatie van de patiënt.

```
<id root="2.16.840.1.113883.2.4.6.3" extension="000700434"
  assigningAuthorityName="Ministerie van BiZa"/>
```

```
<id root="2.16.840.1.113883.2.4.6.1.100702.12" extension="830760"
  assigningAuthorityName="Ons Ziekenhuis"/>
```

Zorgverlenende Organisaties: Organisaties (of organisatiedelen) die op enigerlei manier bij het zorgproces of daaraan verwante processen betrokken zijn. Deze organisaties dienen binnen AORTA landelijk uniek geïdentificeerd te worden door middel van:

1. het CIBG UZI-Register-Abonneenummer (URA). Dit nummer is aanwezig op de UZI pas, het gebruik ervan binnen AORTA is feitelijk verplicht.
2. indien de organisatie geen (CIBG) URA bezit: de (Vektis) AGB-Z code.

Indien van een organisatie meerdere identificaties worden verstuurd dient tenminste of de (CIBG) URA of de (Vektis) AGB-Z code aanwezig te zijn.

Het begrip organisatie is in Nederland tevens van toepassing op zorgverleners die als zelfstandig ondernemer werkzaam zijn (ongeacht of zij bij de Kamer van Koophandel ingeschreven zijn). Voorbeeld: "Huisartspraktijk Dr. T. de Vries" wordt beschouwd als een organisatie, ongeacht of dit in de formeel-juridische zin een organisatie is.

De Vektis AGB-Z zorgaanbieder identificatie OID is: 2.16.840.1.113883.2.4.6.1
De UZI-Register-Abonneenummer (URA) Zorgaanbieder identificatie OID is: 2.16.528.1.1007.3.3. Naast het primaire identificatiesysteem kunnen tevens alternatieve identificaties gebruikt worden. De Prismant (SIG) Zorgaanbieder identificatie OID is: 2.16.840.1.113883.2.4.6.2

Zorgverleners: Personen die op enigerlei manier bij het zorgproces of daaraan verwante processen betrokken zijn. Deze personen worden landelijk uniek geïdentificeerd in het UZI-register.

De OID ten bate van identificatie van personen door middel van een UZI is: 2.16.528.1.1007.3.1. Naast de UZI-identificatie (die indien zij bekend is in ieder geval verzonden moet worden) kan gebruik worden gemaakt van andere regionale identificatiesystemen. Veelal wordt het Vektis AGB-Z nummer van de zorgverlener gehanteerd. De Vektis AGB-Z Zorgverlener identificatie OID is: 2.16.840.1.113883.2.4.6.1

Voorbeeld: Een zorgverlener heeft UZI 100434, Vektis AGB-Z nummer 2032988, en een locale (organisatie interne) code "WEE". Er van uitgaande dat het HL7-bericht een identificerend data-element bevat wat herhalend voor mag komen kan dit er in het bericht als onderstaand voorbeeld uitzien. Indien 100330 het Vektis AGB-Z nummer van de organisatie is, dan mag deze organisatie binnen de OID 2.16.840.1.113883.2.4.6.1.100330 zelf unieke nummers uitdelen. Als 160 een organisatie-interne zorgverlenercode is, en de organisatie heeft besloten OID tak 6 te gebruiken om zorgverleners binnen de organisatie te identificeren, dan vormt 2.16.840.1.113883.2.4.6.1.100330.6 met extensie **WEE** een unieke identificatie van deze zorgverlener.

```
<id root="2.16.528.1.1007.3.1" extension="000100434"
  assigningAuthorityName="UZI Register"/>
```

```
<id root="2.16.840.1.113883.2.4.6.1" extension="2032988"
  assigningAuthorityName="Vektis"/>
```

```
<id root="2.16.840.1.113883.2.4.6.1.100330.6" extension="WEE"
  assigningAuthorityName="Ons Ziekenhuis"/>
```

3.5 Identificatiesystemen OID Referentietabel

De volgende tabel bevat een overzicht met de in Nederland gebruikte OID's voor gangbare identificatiesystemen. De getoonde tabel is niet volledig. Zie de website van HL7 Nederland voor een up-to-date en compleet overzicht van de uitgegeven OID's.

OID	Identificatie-systeem	Omschrijving
2.16.840.1.113883.2.4.6.1	VEKTIS AGB-Z	Dient ter identificatie van zorgverleners en zorgverlenende organisaties
2.16.840.1.113883.2.4.6.2	Prismant/SIG code	Dient ter identificatie van zorgverlenende organisaties
2.16.528.1.1007.3.1	Zorgverlener UZI	Dient ter identificatie van zorgverleners (natuurlijke personen) in de Nederlandse zorgsector. Het UZI nummer wordt uitgegeven na opname in het ZorgaanbiederRegister (ZR). Formaat: 9N, met voorloophnullen indien korter dan 9 cijfers.
2.16.528.1.1007.3.2	Systeem UZI	Dient ter identificatie van systemen (applicatiesoftware) in de Nederlandse zorgsector. Formaat: 9N, met voorloophnullen indien korter dan 9 cijfers.
2.16.840.1.113883.2.4.6.3	BSN	Landelijk Nederlands Burgerservicenummer, dient ter identificatie van patiënten of zorgcliënten. Formaat: 9N, met voorloophnullen indien korter dan 9 cijfers.
2.16.840.1.113883.2.4.6.4	UZOVI	Unieke identificatie van een zorgverzekeraar. (Zie noot 1)
2.16. 528.1.1007.3.3	UZI-Register-Abonneenummer (URA)	Dient ter identificatie van zorgorganisaties in de Nederlandse zorgsector. Het betreft zorgorganisaties (inclusief zelfstandigen) die tenminste één UZI pas bezitten. Formaat: 8N, met voorloophnullen indien korter dan 8 cijfers.
2.16.528.1.1007.5.1	BIG-ID	BIG-Register (ID van in BIG Register opgenomen entiteiten)
2.16.840.1.113883.2.4.6.6	Applicaties binnen de nationale infrastructuur	Identificeert applicaties aangesloten op de landelijke infrastructuur, waaronder de ZIM en alle GBZ applicaties.

Noot:

1. Deze OID duidde voor 15 September 2005 de Vektis ZV-tabel aan, en na die datum de UZOVI tabel. De identificatienummers in beide tabellen zijn gelijklopend.

Identificatie van ministeries

Deze tabel bevat door HL7 Nederland uitgegeven identificaties voor Nederlandse Ministeriële Departementen (Ministeries). Deze identificaties worden gebruikt als

organisatie identifiërs. Het ministerie-identificatiesysteem zelf heeft de OID 2.16.840.1.113883.2.4.6.5. Deze tabel wordt nog nader aangevuld.

Extensie	Omschrijving
1	BZK – Ministerie van Binnenlandse Zaken
2	VWS – Ministerie van VWS

4 Vocabulaire voor berichtspecificaties

4.1 Inleiding

Eenheid van taal c.q. standaardisatie van taal is een belangrijk terrein waarop zowel nationaal als internationaal veel werk dient te worden verzet. Binnen het brede begrip "eenheid van taal" is met name de standaardisatie van zorg- en behandelinhoudelijke taal en coderingen een der belangrijkste voorwaarden voor o.a. elektronische zorgdossiers en het uitwisselen van eenduidige zorg- en behandelinformatie tussen zorgprofessionals. Zowel internationaal als nationaal (Nictiz, NEN) is het belang onderkend om aandacht te besteden aan de relatie tussen taalstandaarden en HL7. De wetenschappelijke verenigingen spelen hierin een cruciale rol. Aangezien de HL7 standaard zich strikt genomen primair richt op het bieden van berichtenstructuren waarin informatie op een gestandaardiseerde wijze kan worden afgebeeld, behoren "taalstandaarden" zeker inhoudelijk niet tot het werkgebied van HL7.

Taalstandaarden vormen vanuit de HL7 standaard "externe content", vergelijkbaar met diverse andere externe coderingen en classificaties die inhoudelijk door derden worden ontwikkeld en beheerd. Met betrekking tot "standaardisatie van taal" houdt HL7 zich niet bezig met de inhoudelijke standaardisatie van taal, maar met het afbeelden en weergeven daarvan in de HL7 structuur.

Dit document bevat beschrijvingen van de vocabulaire domeinen toegevoegd (of gewijzigd) voor gebruik in de Nederlandse situatie. Indien een vocabulaire in dit document niet genoemd wordt, zijn er geen specifieke Nederlandse richtlijnen van toepassing en wordt verwezen naar de vocabulaires zoals opgenomen in de internationale HL7 standaard.

4.2 Vocabulaires in HL7v3

Aan alle HL7v3-modelattributen van een gecodeerd datatype is een "Vocabulary Domain" gekoppeld. Een vocabulairedomein (Vocabulary Domain) wordt beschreven door middel van een tabel met waarden, of een referentie naar een HL7-externe terminologie. Gecodeerde HL7v3-modelattributen hebben altijd een verwijzing (*Vocabulary Binding*) naar een vocabulairedomein of zelfs een specifieke waarde. Er kan op diverse niveau's worden gekoppeld (genoemde modelattributen dienen slechts als voorbeeld):

- **classCode***: =PSN - Het attribuut classCode heeft een vaste koppeling met de waarde "PSN".
- **classCode***: <=ORG - Het attribuut classCode heeft een koppeling met de waarde "ORG" en waarden die hier hiërarchisch onder liggen.
- **administrativeGenderCode***: CE CNE [0..1] < AdministrativeGender - Het attribuut administrativeGenderCode heeft een koppeling met waardendomein "AdministrativeGender". Dit waardendomein is niet uitbreidbaar met eigen codes (CNE=Coded No Extensibility)
- **code***: CE CWE [1..1] < ActCode - Het attribuut code heeft een koppeling met waardendomein "ActCode". Dit waardendomein is uitbreidbaar met eigen codes (CWE=Coded With Extensibility)

De verwijzing is meestal zichtbaar in de R-MIM's, maar wordt zeker in de HMD's en Message Types gespecificeerd. HL7v3-vocabulairedomeinen worden weergegeven in een tabel met een vaste opmaak welke hier onder wordt toegelicht. Externe

vocabulairedomeinen hebben een opmaak die is vast gesteld door de betreffende beherende instantie.

Structuur van een Vocabulary Domain table

MyVocabularyDomain

This table contains values related to ... etc.

Level/Type (1)	Type, Domain name and/or Mnemonic code (2)	Mnemonic (2)	Definition/ description

Noten:

1. Voorbeelden "0-A" en "2-L". Level geeft het niveau in de hiërarchie aan voor de code. Hoe hoger het nummer hoe dieper in de hiërarchie. Het type kan drie waarden aannemen:

- "A" (Abstract) - dit betreft codes die niet in berichten kunnen voorkomen. Het gaat meestal om groeperingsbenamingen van de codes die er hiërarchisch onder vallen
- "S" (Specialiseerbaar) - dit betreft codes die in berichten kunnen voorkomen, maar waar daarnaast mogelijk nog een hiërarchisch niveau onder voorkomt.
- "L" (*Leaf*) - dit betreft codes die in berichten kunnen voorkomen en waaronder geen codes kunnen bestaan.
- "D" (*Deprecated*) - dit betreft codes die nog wel in berichten kunnen voorkomen maar vooral uit historisch oogpunt in het vocabulairedomein wordt aangehouden.

2. De externe terminologienaam wordt ter informatie gegeven indien de code afkomstig is uit een bestaande HL7-externe terminologie.

4.3 HL7 Vocabulary Domain Values

Indien de waarden binnen een vocabulaire geen aanpassingen/inperkingen behoeven in de Nederlandse situatie wordt volstaan met een verwijzing naar de vocabulaire documentatie binnen de HL7 materialen (wel wordt de OID van de desbetreffende vocabulaire gedocumenteerd).

HL7 Nederland streeft ernaar zo min mogelijk vocabulaires voor Nederland te definiëren. Indien mogelijk wordt gebruik gemaakt van in de internationale standaard opgenomen vocabulaire tabellen en (indien dat niet mogelijk is) van tabellen gedefinieerd door Nederlandse standaardisatie organisaties.

RoleCodeNL - zorgverlener type (natuurlijke personen)

Dit codesysteem bevat waarden die het roltype bepaalt van een zorgverlener of andere medewerker in de gezondheidszorg. De onderstaande tabel is een invulling van het HL7v3 vocabulaire domein "RoleCode" en wel specifiek het concept "AssignedRoleType". Deze tabel is specifiek voor Nederland en niet onderhevig aan harmonisatie met HL7.

Dit codesysteem heeft de **OID 2.16.840.1.113883.2.4.15.111**.

Lvl	Type, Domain name and/or	Mnemonic	Definition/Description
-----	-----------------------------	----------	------------------------

	Mnemonic code		
1	A: AssignedRoleType		Een roltype wordt gebruikt om een entiteit verder aan te duiden, die een rol speelt met <i>AssignedEntity</i> als modelattribuut <i>RoleClass</i> .
2	S: Arts	01.000	Arts
3	L: (01.002)	01.002	Internist-Allergoloog
3	L: (01.003)	01.003	Anesthesioloog
3	L: (01.004)	01.004	Apotheekhoudende huisarts
3	L: (01.008)	01.008	Arts arbeid gezond. / bedrijfsarts
3	L: (01.010)	01.010	Cardioloog
3	L: (01.011)	01.011	Cardiothoracaal chirurg
3	L: (01.012)	01.012	Dermatoloog
3	L: (01.013)	01.013	Gastro-enteroloog
3	L: (01.014)	01.014	Chirurg
3	L: (01.015)	01.015	Huisarts
3	L: (01.016)	01.016	Internist
3	L: (01.018)	01.018	Keel- neus- en oorarts
3	L: (01.019)	01.019	Kinderarts
3	L: (01.020)	01.020	Arts klinische chemie
3	L: (01.021)	01.021	Klinisch geneticus
3	L: (01.022)	01.022	Klinisch geriater
3	L: (01.023)	01.023	Longarts
3	L: (01.024)	01.024	Arts-microbioloog
3	L: (01.025)	01.025	Neurochirurg
3	L: (01.026)	01.026	Neuroloog
3	L: (01.030)	01.030	Nucleair geneeskundige
3	L: (01.031)	01.031	Oogarts
3	L: (01.032)	01.032	Orthopedisch chirurg
3	L: (01.033)	01.033	Patholoog
3	L: (01.034)	01.034	Plastisch chirurg
3	L: (01.035)	01.035	Psychiater
3	L: (01.039)	01.039	Radioloog
3	L: (01.040)	01.040	Radiotherapeut
3	L: (01.041)	01.041	Reumatoloog
3	L: (01.042)	01.042	Revalidatiearts
3	L: (01.045)	01.045	Uroloog
3	L: (01.046)	01.046	Gynaecoloog
3	L: (01.047)	01.047	Verpleeghuisarts
3	L: (01.048)	01.048	Arts arbeid gezond. / verzekeringsarts
3	L: (01.050)	01.050	Zenuwarts
3	L: (01.055)	01.055	Arts maatschappij en gezondheid
3	L: (01.056)	01.056	Arts verstandelijk gehandicapten
2	S: (02.000)	02.000	Tandarts
3	L: (02.053)	02.053	Orthodontist
3	L: (02.054)	02.054	Kaakchirurg
2	S: (03.000)	03.000	Verloskundige
2	S: (04.000)	04.000	Fysiotherapeut
2	S: (16.000)	16.000	Psychotherapeut
2	S: (17.000)	17.000	Apotheker
3	L: (17.060)	17.060	Ziekenhuisapotheker
2	S: (25.000)	25.000	GZ-psycholoog
3	L: (25.061)	25.061	Klinisch psycholoog
2	S: (30.000)	30.000	Verpleegkundige
2	S: (83.000)	83.000	Apothekersassistent
2	S: (85.000)	85.000	Tandprotheticus
2	S: (86.000)	86.000	Verzorgenden in de individuele gezondheidszorg (VIG-ers)

2	S: (87.000)	87.000	Optometrist
2	S: (88.000)	88.000	Huidtherapeut
2	S: (89.000)	89.000	Diëtist
2	S: (90.000)	90.000	Ergotherapeut
2	S: (91.000)	91.000	Logopedist
2	S: (92.000)	92.000	Mondhygiënist
2	S: (93.000)	93.000	Oefentherapeut Mensendieck
2	S: (94.000)	94.000	Oefentherapeut Cesar
2	S: (95.000)	95.000	Orthoptist
2	S: (96.000)	96.000	Podotherapeut
2	S: (97.000)	97.000	Radiodiagnostisch laborant
2	S: (98.000)	98.000	Radiotherapeutisch laborant

Toelichting:

- Deze codes zijn gebaseerd op de Certification Practice Statement (CPS) van het UZI-register, die weer verwijst naar de tabellen voor "Beroepstitels, opleidingstitels en specialismen". Deze sluiten aan op artikel 3 en artikel 34 uit de wet BIG.
- De hier boven afgebeelde lijst is een momentopname. Voor de actuele lijst wordt u verwezen naar het Certificate Practice Statement (CPS) van het UZI-register.
- De waarde "00.000" (geen beroepstitel) komt als waarde op de UZI-pas voor, maar wordt binnen HL7 versie 3 niet gebruikt.
- De in het verleden voor testdoeleinden gebruikte codes 833 (Apotheker) en 638 (Huisarts) dienen te worden vervangen door respectievelijk de codes 17.000 en 01.015.

RoleCodeNL - zorgaanbiedertype (organisaties)

Dit codesysteem bevat waarden die het roltype bepaalt van een organisatie die als zorgaanbieder optreedt. De onderstaande tabel is een invulling van het HL7v3 vocabulaire domein "RoleCode" en wel specifiek het concept "AssignedRoleType". Deze tabel is specifiek voor Nederland en niet onderhevig aan harmonisatie met HL7.

Dit codesysteem heeft de **OID 2.16.840.1.113883.2.4.15.1060**.

Lvl	Type, Domain name and/or Mnemonic code	Mnemonic	Definition/Description
1	A: AssignedRoleType		Een roltype wordt gebruikt om een entiteit verder aan te duiden, die een rol speelt met AssignedEntity als modelattribuut RoleClass.
2	S: (V4)	V4	Ziekenhuis
3	L: (V5)	V5	Universitair Medisch Centrum
3	L: (V6)	V6	Algemeen ziekenhuis
2	S: (Z4)	Z4	Zelfstandig behandelcentrum Een Zelfstandig behandelcentrum (ZBC) is een door de overheid erkende kliniek, die verzekerde zorg levert en waarin minstens 2 specialisten samenwerken. ¹
3	L: (Z5)	Z5	Diagnostisch centrum Een diagnostisch behandelcentrum is een ZBC dat alleen diagnostiek levert.
4	A: (B1)	B1	Beeldvormende diagnostiek
5	L: (B2)	B2	Echocentrum
2	S: (X3)	X3	Verplegings- of verzorgingsinstelling

¹ Bron: <http://orde.artsennet.nl/Nieuws/Artikel/Zelfstandige-Behandelcentra-ZBCs.htm>

3	L: (R5)	R5	Verpleeghuis
3	L: (M1)	M1	Verzorgingstehuis
2	A: Apotheekinstelling		
3	L: (J8)	J8	Openbare apotheek
3	L: (K9)	K9	Zelfstandig opererende ziekenhuisapotheek
2	A: Huisartsinstelling		
3	S: (Z3)	Z3	Huisartspraktijk (zelfstandig of groepspraktijk)
4	L: (K3)	K3	Apotheekhoudende huisartspraktijk
3	L: (N6)	N6	Huisartsenpost (t.b.v. dienstwaarneming)
2	S: (L1)	L1	Laboratorium
2	L: (P4)	P4	Polikliniek (als onderdeel van een ziekenhuis)
2	L: (R8)	R8	Revalidatiecentrum
2	L: (P6)	P6	Preventieve gezondheidszorg (waaronder Bevolkingsonderzoek)
2	L: (G5)	G5	Geestelijke Gezondheidszorg
2	L: (G6)	G6	Verstandelijk gehandicaptenzorg
2	L: (T2)	T2	Thuiszorg
2	L: (JGZ)	JGZ	Jeugdgezondheidszorg
2	L: (G3)	G3	Verloskundigenpraktijk

Toelichting:

- Er is in Nederland geen complete, openbare en gekwalificeerde lijst van zorgaanbiedertypen. Deze lijst van zorgaanbiedertypen is ontstaan uit de behoefte van het zorgveld om het type zorgaanbieder te kunnen duiden om bijvoorbeeld een diagnose of een brief in context te zien.
- Dit codesysteem wordt beheerd door Nictiz en de inhoud heeft geen relatie met het CIBG.
- De hier boven afgebeelde lijst is een momentopname. Voor actuele lijst wordt u verwezen naar Nictiz via <http://www.aortarelease.nl>.
- Deze codes komen niet voor op de UZI-pas, waar wel het URA nummer van de zorgaanbieder van de pashouder wordt aangeduid, maar niet het type organisatie.

4.4 Vocabulaires OID-Referentietabel

De volgende tabel bevat een overzicht met de in Nederland gebruikte OID's voor de identificatie van vocabulaires (code tabellen), met in ieder geval de OID's van een groot aantal Nederland-specifieke vocabulaires. De onderstaande tabel is niet volledig. Zie de website www.hl7.nl voor een up-to-date en compleet overzicht van de uitgegeven OID's.

OID	Identificatiemechanisme	Omschrijving
2.16.840.1.113883.2.4.4.1	G-Standaard GPK	Medicatie. Generieke productcode (GPK, GPC). Werkzame stof (inclusief sterkte), farmaceutische vorm (doseervorm) en de toedieningsweg. Voorbeeld: Diazepam 5 mg/st, tablet, oraal. (1)
2.16.840.1.113883.2.4.4.2	Medicatie dosering WCIA Tabel 25	Medicatie dosering (precoordinated), bestaande uit een reeks 'aanelkaargeplakte' Tabel 25 codes.
2.16.840.1.113883.2.4.4.3	Tijdseenheden - WCIA Tabel 25	Toedieningstijdseenheden (numcode)
2.16.840.1.113883.2.4.4.4	Eenheden gebruiksadvies - WCIA Tabel 25	Eenheden (numcode)
2.16.840.1.113883.2.4.4.5	Tekstcategorien - WCIA Tabel 25	
2.16.840.1.113883.2.4.4.6	Aanvullende teksten - WCIA Tabel 25	Aanvullende teksten bij medicatie dosering (numcode)
2.16.840.1.113883.2.4.4.7	G-Standaard HPK	Medicatie. Handelsproductcode (HPK, TPC). Als GPK, verbijzonderd met (onder andere) fabrikantnummer. Voorbeeld: Valium 5mg tablet, Stesolid 5mg tablet als verschijningsvormen van Diazepam. (Zie noot 1)
2.16.840.1.113883.2.4.4.8	G-Standaard Artikel Code	Medicatie: Artikel Kode, Artikelnummer. Als HPK, verbijzonderd naar verpakking. Valium 5 mg tablet, 2 strip à 10 tablet, 1 verpakking à 50 EAV. (Zie noot 1)
2.16.840.1.113883.6.1	LOINC	Observation identifier, identificeert een bepaald type (laboratorium)test.
1.0.3166.1.2.2	ISO 3166-1 editie 2, 2-character codes	2-karakter country codes volgens ISO 3166-1 editie 2
2.16.840.1.113883.2.4.4.15	Nederlandse Postcodes	Lengte 6, zonder spatie
Nederlandse uitbreidingen op versie 3 vocabulaires:		
2.16.840.1.113883.2.4.15.111	HL7-NL Role Code	Zorgverlener rol typen: Identificeert de diverse rollen die een zorgverlener binnen een organisatie vervult
2.16.840.1.113883.2.4.15.4	HL7-NL Act Code (Gegevenssoort)	Gegevenssoorten, een hiërarchische value set gebruikt door de verwijzindex.

Noot:

1. In de G-Standaard kan men eigen artikelnummers gebruiken (codes 90.000.000 en hoger). Deze codes zijn leverancier- of locatiespecifiek. Bij deze codes dient gebruik gemaakt te worden van een andere OID dan de G-Standaard OID's. De G-Standaard OID's mogen dus **niet** gebruikt worden in combinatie met eigen codes.

5 OID gerelateerde implementatieaspecten

5.1 Inleiding

Nictiz heeft gekozen voor HL7 versie 3 en gebruikt daarom OID's als onderdeel van identificaties en coderingen die voorkomen in diverse onderdelen van de HL7 berichten.

Voorbeelden van identificaties die binnen een HL7 bericht in AORTA voorkomen:

- In de **transmission wrapper**:
 - Berichten (message ID, identificatie van interacties tussen applicaties)
 - Applicaties (device ID, als zender en ontvanger van de interactie)In de transmission wrapper gaat het altijd om de adressering van de interactie.
- In de **control act wrapper**:
 - Zorgaanbieders (altijd geïdentificeerd o.b.v. een URA nummer)
 - Zorgverleners (altijd geïdentificeerd o.b.v. een UZI nummer)
 - Zorgmedewerkers (ook geïdentificeerd o.b.v. een UZI nummer)
 - Zorgsystemen (geïdentificeerd o.b.v. een UZI services certificaat)In de control act wrapper gaat het altijd om een auteur of verantwoordelijke.
- In de **payload**:
 - Patiënten (altijd o.b.v. het BSN binnen de landelijke infrastructuur)
 - Zorggegevens (medicatieverstrekkingen, laboratoriumuitslagen, etc.)Daarnaast worden ook in de payload zorgverleners/zorgaanbieders aangeduid.

Voor al deze identificaties geldt dat ze **uniek en persistent** moeten zijn. Dit betekent onder andere dat ze ook bestand moeten zijn tegen migraties en fusies van informatiesystemen.

Het is onmogelijk om vanuit HL7 Nederland, of welke instantie dan ook, centraal bij te houden welke OID's gebruikt worden voor lokaal gegenereerde zorggegevens. Elke applicatie die een uniek nummer bepaalt per ingevoerde order of uitgevoerde activiteit zal daarvoor een eigen unieke en persistente OID moeten hanteren, om te zorgen dat bijvoorbeeld elke medicatieverstrekking te onderscheiden is van elke andere medicatieverstrekking.

Wat daarom nodig is, is een compromis met de volgende kenmerken:

- Forceer OID roots per applicatie, per zorgaanbieder of per softwareleverancier.
- Geef richtlijnen voor het uitdelen van nodes onder deze roots.
- Laat het beheer hiervan over aan de assigning authority (behorend bij de root).

De volgende IDs kunnen in principe als root dienen in de zorg:

- De **applicatie**, o.b.v. de door het LSP uitgedeelde applicatie ID (dit zou handig zijn, maar er is dan geen vanzelfsprekende assigning authority).
- De **zorgaanbieder**, o.b.v. het UZI Register Abonnee (URA) nummer (probleem is dat nog niet zeker is of URA 1-op-1 bij zorgaanbieder zal horen).
- Het **Goed Beheerd Zorgsysteem**, o.b.v. de daaraan uitgedeelde GBZ ID (probleem is dat de juridische definitie van een GBZ nog niet 100% zeker is).
- De **IT leverancier**, o.b.v. een nog te bepalen identificatie van de organisatie (het lijkt niet zo handig om leverancier *assigning authority* te maken, bijvoorbeeld i.v.m. conversie naar een andere leverancier).

Implementatierichtlijnen zijn nodig om duidelijk te maken welke OID structuur gebruikt moet worden in HL7 versie 3 berichten binnen AORTA. Hierover is een beslissing genomen binnen Nictiz, met deze richtlijn als uitkomst:

Ten behoeve van de wereldwijde uniciteit en persistentie van gegevensidentificatie wordt de zorgaanbieder geadviseerd haar URA (UZI Register Abonнеenummer) te gebruiken als basis voor de OID root van alle binnen de zorgaanbieder gegenereerde gegevens. Zorgaanbieders die reeds een bestaande OID hanteren, mogen deze blijven gebruiken.

Merk op dat, ondanks deze richtlijn, een applicatie nooit iets mag afleiden (o.b.v. parsing van de nodes) uit de opbouw van een ontvangen OID.

5.2 Gevolgen van OID's voor HL7-applicaties

De volledige ID (dus extensie én OID root) moet altijd reproduceerbaar zijn

- bij opslaan opgevraagde gegevens moet volledige identifier worden bewaard
- bij opnieuw opvragen van gegevens moet dezelfde identifier worden opgeleverd

De volgende praktijksituaties moeten ondersteund worden:

- Gegevens **wijzigen van bron**
 - Er **vindt** een migratie plaats van gegevens tussen softwareleveranciers, zorgverleners of locaties
 - De ID (OID root en extensie) blijft gehandhaafd; er mag dus **niet meer omgenummerd** worden
 - Gegeven wordt afgemeld voor oude bron en aangemeld als onderdeel van nieuwe bron
- Externe gegevens worden **opgenomen in eigen dossier**
 - De ID (OID root en extensie) blijft gehandhaafd; er mag dus **geen eigen nummer** meer toegekend worden.
 - Het feit dat het gegeven extern is moet ook bewaard blijven (dit is immers niet afleidbaar uit de structuur van de OID)
 - Mag echter wel worden opgeleverd als onderdeel van bredere context (bijvoorbeeld het huisartsdossier)
- Het **fuseren of splitsen** van:
 - Zorgaanbieders
 - Softwareleveranciers
 - Informatiesystemen

Momenteel wordt in implementaties vaak nog getruukt met vaste waarden voor OID's, maar juist gebruik vergt aanpassingen in minimaal de volgende applicatie-aspecten:

- **databases** (opslag root OID's)
- **configuratie** (definitie lokale roots)
- **implementatie** (uitvoering conversies)

6 Datatypen

6	Datatypen	29
6.1	Inleiding	30
6.2	Algemene toelichting op het gebruik van XML	30
6.2.1	Character Set	30
6.2.2	Speciale tekens	30
6.2.3	Algemene opbouw van een HL7 versie 3 bericht	31
6.2.4	Transport van de XML-berichten	31
6.3	XML-representatie van modelklassen en -attributen	31
6.3.1	Klassen, attributen en datatypen	31
6.3.2	Uitzonderingen met gegevens als element content	33
6.3.3	Gecombineerde datatypen	33
6.3.4	Structurele attributen	33
6.4	Cardinaliteit, "mandatory" en conformance	34
6.5	Ontbrekende gegevens: nullFlavors	36
6.6	ANY (het generieke datatype)	40
6.7	BL (Boolean – true/false)	41
6.8	BIN (Binary data - binaire gegevens)	42
6.9	ED (Encapsulated Data - ingekapselde gegevens)	43
6.10	ST (String of Characters - reeks tekens)	46
6.11	SC (String with Code - reeks tekens voorzien van code)	47
6.12	CD (Concept Descriptor - gecodeerde gegevens)	48
6.13	CE (Coded with Equivalents - gecodeerde gegevens, met equivalenten)	49
6.14	CV (Coded Value - gecodeerde waarde)	52
6.15	CO (Coded Ordinal - sorteerbare gecodeerde waarde)	53
6.16	CS (Coded Simple - eenvoudige code)	54
6.17	II (Instance Identifier - objectidentificatie)	55
6.18	URL (Universal Resource Locator)	58
6.19	TEL (Telecommunication Address - telecommunicatiecontact)	59
6.20	AD (Postal Address - adres)	61
6.21	EN (Entity Name - naam voor een entiteit)	65
6.22	PN (Person Name - persoonsnaam)	66
6.23	ON (Organization Name - organisatienaam)	80
6.24	QTY (Quantity - hoeveelheid)	82
6.25	INT (Integer Number - geheel getal)	83
6.26	REAL (Real Number - getal met decimalen)	84
6.27	PQ (Physical Quantity - kwantitatieve weergave van fysieke grootheden)	85
6.28	MO (Monetary Amount - hoeveelheid geld)	87
6.29	TS (Time Stamp - tijdstip)	88
6.30	BAG (Bag – verzameling met mogelijke dubbeln)	90
6.31	SET (Set – verzameling zonder dubbeln)	90
6.32	SXCM (Set Component – deelverzameling)	90
6.33	IVL (Interval)	91
6.34	IVL_TS (Interval of Time Stamps - tijdsinterval)	93
6.35	IVL_INT (Interval of Integers – numeriek interval)	96
6.36	IVL_PQ (Interval of Physical Quantities - hoeveelheidsinterval)	97
6.37	RTO (Ratio - onderlinge verhouding tussen twee gegevens)	99
6.38	GTS (General Timing Specification - generieke tijdspecificatie)	101
6.39	PIVL_TS (Periodic Interval of Time - periodiek herhalend tijdsinterval)	104

6.1 Inleiding

Deze sectie beschrijft de belangrijkste datatypen, waarbij uitsluitend die onderdelen worden beschreven die in de Nederlandse situatie van toepassing zijn. Indien er specifieke aanwijzingen zijn hoe iets (wellicht in afwijking van de internationale standaard) in Nederland geïmplementeerd moet worden dan is dit in de tekst aangegeven.

Alleen de binnen de modellen van deze handleiding gebruikte datatypen worden uitgelegd. Voor verdere informatie zie HL7v3 standaard/ballot materiaal, met name de hoofdstukken "abstract data types" en "XML ITS data types".

6.2 Algemene toelichting op het gebruik van XML

Het uitwisselformaat voor HL7 versie 3 berichten is (op dit moment) de Extensible Markup Language (XML, zie [XML]). Het is niet de bedoeling van deze gids, toelichting van XML in het algemeen te geven maar een paar kenmerken en vereisten worden hier wel genoemd.

6.2.1 Character Set

De encoding in de XML-proloog van een HL7 versie 3 bericht moet UTF-8 zijn.

```
<?xml version="1.0" encoding="utf-8"?>
```

Of als standaardwaarde

```
<?xml version="1.0"?>
```

6.2.2 Speciale tekens

In XML zijn bepaalde tekens in gegevens te vervangen door combinaties van tekens (character entities) omdat ze in de XML-structuur een andere en speciale betekenis hebben. Een *XML character entity* is een "&" gevolgd door een aantal tekens en afsluitend een ";". De sequentie wordt door de XML-processor vervangen door het speciale teken. De volgende tabel geeft een (niet volledige) overzicht over deze tekens. Voor verdere informatie zie de XML-specificatie ([XML]).

Speciale tekens in gegevens	Te vervangen door	Opmerkingen
&	&	
<	<	
>	>	
'	'	Een ' is binnen gegevenswaarden niet toegestaan als deze waarden ook door dit teken worden begrensd en moet dan worden vervangen.
"	"	Een " is binnen gegevenswaarden niet toegestaan als deze waarden ook door dit teken worden begrensd en moet dan worden vervangen.

XML-voorbeeld

```
<!-- XML-processor levert: Waarden > 5 mg/l zijn pathologisch -->
<text>Waarden &gt; 5 mg/l zijn pathologisch</text>

<!-- XML-processor levert: Wijkapotheek Leermans & zonen -->
<name>Wijkapotheek Leermans &amp; zonen</name>

<!-- XML-processor levert: 's-Gravenhage -->
<code ... displayName="'s-Gravenhage"/> of
<code ... displayName="&apos;s-Gravenhage"/>

<!-- XML-processor levert: "'s-Gravenhage" -->
<code ... displayName="&quot; 's-Gravenhage&quot;"/>
```

6.2.3 Algemene opbouw van een HL7 versie 3 bericht

HL7 versie 3 berichten hebben als *root*-element altijd de naam van de bijhorende interactie. Zo begint een HL7 versie 3 bericht op basis van interactie "PORX_IN123456" met een XML-root element <PORX_IN123456>. De HL7v3 interactie schema's (voor toelichting W3C schema, zie [XMLSC]) hebben in het algemeen dezelfde naam als het root-element.

```
<?xml version="1.0"? encoding="utf-8">
<PORX_IN123456 ...
  xmlns="urn:hl7-org:v3"
  xmlns:voc="urn:hl7-org:v3/voc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <!-- ... HL7 versie 3 bericht inclusief wrappers en payload →
</PORX_IN123456>
```

Een *schemaLocation* kan de zender wel toevoegen, maar de ontvanger mag de informatie in *schemaLocation* niet gebruiken. Om te voorkomen dat schemareferenties verwerkt zouden worden, wordt het zenden ervan sterk ontraden.

In het geval van een gebundelde set HL7-interacties in een zogeheten *batch* heeft het root element de naam van de batch interactie en de verschillende interacties (deelberichten) zijn daarin opgenomen. Voor verdere informatie zie [NIWrappers].

6.2.4 Transport van de XML-berichten

In deze gids wordt niets over de feitelijke transport van de berichten gezegd, zie hiervoor de Nictiz-gidsen ([NIInfraDom], [NIWebSvPrf]).

6.3 XML-representatie van modelklassen en -attributen

6.3.1 Klassen, attributen en datatypen

De attributen van klassen in de HL7-modellen worden op een bepaalde, vast voorgedefinieerde wijze omgezet naar de XML-representatie. [Merk overigens op dat de term 'attri-

buut' hier ongelukkig is, omdat verwarring kan ontstaan met *XML-attributen*. Vandaar dat buiten deze paragraaf zoveel mogelijk in XML-termen wordt gesproken.]

Voorbeeld:

```
Patient  
classCode*: <= PAT  
id*: SET<II> [1..*]  
addr*: BAG<AD> [0..*]  
telecom*: BAG<TEL> [0..*]
```

In de bijgaande klasse "Patient" zijn een aantal modelattributen opgenomen, waaronder de classCode (met als vaste waarde "PAT"), id, addr en telecom.

Elk van de modelattributen heeft onder meer een naam, bijvoorbeeld id; een datatype, bijvoorbeeld BAG<AD>; een cardinaliteit, bijvoorbeeld [0..*] en een conformiteit.

De notatieconventie voor conformiteit is een asterisk (*) voor *required* en **vet gedrukt** voor *mandatory*. Structurele attributen (met name classCode, moodCode, typeCode en determinerCode) zijn altijd *mandatory* maar deze kunnen een vaste waarde krijgen die is vastgelegd in het XML Schema, waardoor het attribuut toch niet aanwezig hoeft te zijn in de XML-berichtinstance. Zie verder §6.4 voor toelichting op conformance.

Met uitzondering van de structurele attributen (zie hieronder) worden de modelattributen weergegeven als XML-elementen. De elementen dragen de naam zoals in het model aangegeven. Bijvoorbeeld als het attribuut *id* is, dan is de naam van het XML-element <id/>.

```
<id ... />  
<addr ... />  
<telecom ... />
```

In het XML-schema is de cardinaliteit van het modelattribuut meegenomen. In bovenstaand voorbeeld is <id> [1..1], <addr> [0..1] en <telecom> [0..*] optioneel.²

De XML-elementen hebben XML-attributen (let op het andere gebruik van de term!), die door het datatype zelf worden bepaald. De hier gebruikte notatievorm is als volgt:

@naam attribuutbeschrijving / uitleg / betekenis (datatype / pattern)

N.B. een @ wordt hier als prefix voor XML-attributen gebruikt om het beter te kunnen onderscheiden van XML-elementnamen. Voor de datatypes zijn attributen gedefinieerd die de gegevens dragen. Zo heeft in het voorbeeld het *id* modelattribuut het datatype II (Instance Identifier). De bij dit datatype behorende attributen zijn bijvoorbeeld *root* en *extension* en worden als XML-attributen bij het element weergegeven. Zie hieronder:

```
<id root="" extension="" />
```

Verdere toelichtingen over datatypes en de bijbehorende XML-attributen zijn in de volgende hoofdstukken per datatype opgenomen.

² Let op: in de huidige XML-schemataal kan de cardinaliteit van een XML element vastgelegd worden maar of de "inhoud", d.w.z. de gegevens volgens de datatype specificatie aanwezig zijn of niet of de juiste combinatie hebben, is niet valideerbaar met een schema zoals ze nu uitgebracht zijn. Hier is echter een tweede validatiestap nodig als gegarandeerd moet worden dat bijvoorbeeld een II datatype altijd met een *root/extension* attribuut-paar gevuld is of een *nullFlavor* attribuut bevat. Om dit niveau van validatie mogelijk te maken zijn verdere validatiestappen nodig met zogenaamde *constraint* talen (bijvoorbeeld OCL, Schematron).

6.3.2 Uitzonderingen met gegevens als element content

Voor de meeste datatypen komen de feitelijke gegevens in de XML-attributen terecht. Er bestaan echter een aantal uitzonderingen. Voor de datatypen *Binary*, *Encapsulated Data*, *Entity Name*, *Person Name*, *Organization Name*, *Trivial Name*, *Address* en *Character String* worden de gegevens als zogenaamde 'element content' weergegeven.

Voorbeeld: de componenten van een adres worden door subelementen van het <addr> element met de gegevens als 'element content' (in het voorbeeld in rood weergegeven).

```
<addr>
  <streetName>Purmersteenweg</streetName>
  <houseNumber>42</houseNumber>
  <postalCode>1441 DM</postalCode>
  <city>Purmerend</city>
</addr>
```

6.3.3 Gecombineerde datatypen

Er bestaan een aantal gecombineerde datatypen. Zo wordt bijvoorbeeld een interval met een onder- en een bovengrens aangeduid en stelt een ratio een verhouding van twee waarden voor.

Bij gecombineerde datatypen is altijd de substructuur (bijvoorbeeld <low> en <high> bij een interval) weergegeven als subelementen van het desbetreffende datatype element.

```
<effectiveTime>
  <low value="20040507"/>
  <high value="20040909"/>
</effectiveTime>
```

Bij beschrijving van gecombineerde datatypen worden subelementen als volgt genoteerd:

<naam subelement> beschrijving / uitleg / betekenis (datatype)

6.3.4 Structurele attributen

Er is een lijst van modelattributen die niet als aparte elementen worden gerepresenteerd maar als XML-attributen van het XML-element dat hoort bij de klasse als geheel. Dit zijn:

RIM klasse	Modelattribuut
Act	moodCode
	classCode
	negationInd
	levelCode
ActRelationship	typeCode
	inversionInd
	contextControlCode
	contextConductionInd
	negationInd
Entity	classCode
	determinerCode
Participation	typeCode
	contextControlCode

RIM klasse		Modelattribuut
	Role	classCode
		negationInd
	RoleLink	typeCode

Voorbeeld: in de modelklasse *Observation* zijn twee modelattributen, *classCode* en *moodCode*, die structurele attributen zijn en dus niet als aparte XML-elementen verschijnen, maar als XML-attributen van het XML-element <Observation>, zoals hieronder:

```
<Observation classCode="OBS" moodCode="EVN"/>
```

6.4 Cardinaliteit, "mandatory" en conformance

In de HL7 versie 3 berichten kan bij de modelattributen gebruik gemaakt worden van verdere eigenschappen zoals de cardinaliteit van het betreffende modelattribuut. De volgende tabel geeft een overzicht van de betekenis van deze aanvullende eigenschappen.

Begrip	Verklaring / opmerkingen
Cardinaliteit	Specificeert het minimale en maximale aantal herhalingen van het betreffende modelonderdeel (d.w.z. modelattribuut of associatieve klasse) in de XML-instance. Bijvoorbeeld 1..* houdt in dat het XML-element minimaal 1 keer aanwezig moet zijn, en dat het XML-element een onbeperkt keer mag herhalen in de instance.
Mandatory	Een "mandatory" modelonderdeel moet altijd aanwezig zijn in de XML-instance en er is geen nullFlavor toegestaan. Zonder dit onderdeel is de XML-instance ongeldig. Voor "mandatory" modelonderdelen (en bijbehorend XML-element) is het aantal herhalingen (cardinaliteit) per definitie minimaal 1 (één).
Conformance	Hier wordt onderscheid gemaakt tussen R (<i>required</i>), NP (not permitted, niet toegestaan) en optional (optioneel). <i>R = Required</i> betekent dat zowel de zendende als de ontvangende applicatie dit modelonderdeel moeten ondersteunen. Als er gegevens beschikbaar zijn, dan moet dit onderdeel ook in de XML-instance aanwezig zijn. Als de minimale cardinaliteit 0 is en er geen gegevens beschikbaar zijn, mag het element ontbreken in de XML-instance en is het bericht toch geldig. Als de minimale cardinaliteit 1 is en er geen gegevens beschikbaar zijn, dient dit door een nullFlavor (zie paragraaf 6.5) aangeduid te worden. Als een zendende applicatie het optionele modelonderdeel altijd zou weglaten of als deze in het verplichte modelonderdeel altijd een nullFlavor zou zenden omdat deze het concept niet ondersteunt, dan is zo'n applicatie niet-conformant. Als een ontvangende applicatie het modelonderdeel indien aanwezig nooit zou verwerken, dan is zo'n applicatie niet-conformant. <i>NP = not permitted</i> (niet toegestaan) betekent dat het modelonderdeel niet in de instance mag voorkomen (en ook niet aanwezig is in het onderliggend schema). <i>Optional</i> (optioneel) betekent dat een modelonderdeel niet of wél aanwezig mag zijn en dat geen support door zendende en ontvangende applicaties verplicht is.
NullFlavor	Voor modelonderdelen met een minimale cardinaliteit van 1 dient een nullFlavor doorgegeven te worden als er geen gegevens voor dit element beschikbaar zijn in een zendende applicatie. Voorbeelden zijn "geen informatie" (NI – no information), "onbekend" (UNK – unknown) enz. Voor verdere toe-

lichting zie paragraaf 6.5 Ontbrekende gegevens: nullFlavors.

De onderstaande tabel vat de verschillende varianten samen (met **ROOD** voor verzenders en **BLAUW** voor ontvangers)

Conformance Cardinaliteit	Optioneel	Required	Mandatory
0..1	Invoer is MOGELIJK Hoeveel waarden ingevoerd? 0 → GEEN XML-element 1 → 0 of 1 elementen n → 0 of 1 elementen	Invoer MOET mogelijk zijn Hoeveel waarden ingevoerd? 0 → GEEN XML-element 1 → 1 GEVULD element n → 1 GEVULD element	
	Evt. verzonden waarde MAG verwerkt worden.	Evt. verzonden waarde MOET verwerkt worden.	
0..*	Invoer is MOGELIJK Hoeveel waarden ingevoerd? 0 → GEEN XML-element 1 → 0 of 1 elementen n → 0 tot N elementen	Invoer MOET mogelijk zijn Hoeveel waarden ingevoerd? 0 → GEEN XML-element 1 → 1 GEVULD element n → N GEVULDE elementen	
	Evt. verzonden waarde(n) MOGEN (evt. deels) verwerkt worden.	Evt. verzonden waarde(n) MOETEN ALLEMAAL verwerkt worden.	
1..1	Invoer is MOGELIJK Hoeveel waarden ingevoerd? 0 → LEEG XML-element 1 → 1 element (evt. LEEG) n → 1 element (evt. LEEG)	Invoer MOET mogelijk zijn Hoeveel waarden ingevoerd? 0 → 1 XML-element (null-Flavor) 1 → 1 GEVULD element n → 1 GEVULD element	Invoer MOET VERPLICHT zijn Hoeveel waarden ingevoerd? 0 → NIET toegestaan 1 → 1 GEVULD element n → 1 GEVULD element
	Evt. verzonden waarde MAG verwerkt worden.	Evt. verzonden waarde MOET verwerkt worden.	Verzonden waarde MOET verwerkt worden.
1..*	Invoer is MOGELIJK Hoeveel waarden ingevoerd? 0 → LEEG XML-element 1 → 1 element (evt. LEEG) n → 1 (evt. LEEG) tot N elementen	Invoer MOET mogelijk zijn Hoeveel waarden ingevoerd? 0 → 1 XML-element (null-Flavor) 1 → 1 GEVULD element n → N GEVULDE elementen	Invoer MOET VERPLICHT zijn Hoeveel waarden ingevoerd? 0 → NIET toegestaan 1 → 1 GEVULD element n → N GEVULDE elementen
	Evt. verzonden waarde(n) MOGEN (evt. deels) verwerkt worden.	Evt. verzonden waarde(n) MOETEN ALLEMAAL verwerkt worden.	Verzonden waarde(n) MOETEN ALLEMAAL verwerkt worden.

Toelichting:

	• INVOER betekent hier dat het gegeven betekenis heeft binnen de applicatie. Dat betekent meestal dat de gebruiker het invoert (terwijl die zich bewust is van de betekenis), maar kan in sommige gevallen ook betekenen dat de applicatie het au-
---	--

	tomatisch vult (als de waarde duidelijk is uit de context van een bepaalde functie). Wat NIET is toegestaan, is het standaard versturen van betekenisloze <i>dummy</i> waarden (dus zonder dat de betekenis geborgd is in de applicatie). • VERWERKING betekent ook dat het gegeven betekenis heeft binnen de applicatie. Verwerken kan betekenen: opslaan, afwijzen van specifieke waarden binnen een concept, vertalen naar interne variant op het concept, tonen aan de gebruiker, etc. Het is niet per se noodzakelijk dat het gegeven reproduceerbaar wordt opgeslagen. Voorbeeld: Een verloskundigensysteem dat geen geslacht (van wordende moeders) vastlegt omdat het altijd vrouwen zijn, maar dat wel ontvangen berichten over mannen uitfiltert en niet verder verwerkt, ondersteunt op deze manier het concept geslacht - want anders kan het systeem er ook niet op filteren. Voorbeeld: Een systeem ondersteunt wel telefoonnummers bij personen, maar geen e-mailadressen of faxnummers. • Een leeg XML-element heeft de vorm <code><element xsi:nil="true"/></code>. Een XML-element met een nullFlavor heeft de vorm <code><element nullFlavor="{type NullFlavor}"/></code>. • Het aantal ingevoerde gegevens in de zendende applicatie hoeft niet altijd overeen te komen met het aantal corresponderende, voorkomende elementen. Zo kunnen er twee adressen zijn ingevoerd waarvan er één geheim is en wordt er dus maar 1 gestuurd, ondersteunt het bericht 1 naam terwijl er 3 drie zijn ingevoerd, of is er een HIV-infectie bekend, maar wordt deze gemaskeerd en blijft het hele element achterwege op verzoek van de patiënt of op basis van ruimte voor deze situatie in de specificatie.
--	--

6.5 Ontbrekende gegevens: nullFlavors

Elk HL7v3-element dat in een bericht kan voorkomen heeft in beginsel het XML-attriboot `@nullFlavor` als mogelijkheid. Of gebruik hiervan in een bepaald bericht wordt toegestaan, wordt bepaald door de conformance van dat element voor die toepassing. Als een element 'mandatory' is, dan is een nullFlavor NIET toegestaan.

Een NullFlavor wordt daarmee dus gebruikt in gevallen waar een gegeven verplicht is in een HL7v3 instance (cardinaliteit 1..), maar waarvoor de zendende applicatie geen waarde beschikbaar heeft of het niet kan versturen (bijvoorbeeld vanwege privacy). Dit betreft gegevens waarvan de conformance hoogstens *required* is, want bij *mandatory* conformance **moet** een niet-null waarde worden meegegeven.

Een *NullFlavor* kan de volgende waarden hebben:

NullFlavor

Lvl	Type, Domain name and/or Mnemonic code	Print Name	Definition/Description
1	S: NoInformation (NI)	NoInformation	Uit het gebruik van deze nullFlavor mag geen enkele informatie worden afgeleid. Het betekent niet meer dan dat het betreffende gegeven ontbreekt, zonder reden daarvoor.
2	L: (NA)	not applicable	Er is geen waarde van toepassing binnen deze context (bijvoorbeeld datum v/d laatste menstruatie bij een mannelijke patiënt).
2	S: Unknown (UNK)	Unknown	Er is wel een waarde van toepassing, maar deze is bij de verzender niet bekend (diverse specialisaties zijn mogelijk).
3	L: (NASK)	not asked	De informatie is niet 'gezocht' (bijvoorbeeld als een bepaald gegeven niet aan de patiënt is gevraagd) en daardoor niet bekend.
3	S: AskedButUnknown (ASKU)	AskedButUnknown	De informatie is wel 'gezocht', maar niet 'gevonden' (bijvoorbeeld als het wel aan de patiënt is gevraagd, maar deze het niet wist).
4	L: (NAV)	Temporarily unavailable	De informatie is momenteel nog niet vergaard, maar de verwachting is dat dit op een later moment alsnog gebeurt.
3	L: (TRC)	Trace	Er is sprake van een hoeveelheid die groter is dan 0, maar die te klein is om te worden gekwantificeerd. Deze nullFlavor wordt alleen gebruikt bij datatype PQ (Physical Quantity).
2	S: Other (OTH)	Other	Er is geen bruikbare waarde beschikbaar binnen het domein dat voor het betreffende gegeven van toepassing is (bijvoorbeeld verplicht een vocabulairdomein voor een code).
3	L: (PINF)	positive infinity	Een numerieke waarde die (positief) oneindig is.
3	L: (NINF)	negative infinity	Een numerieke waarde die (negatief) oneindig is.
2	L: (MSK)	Masked	Er is wel informatie over dit gegeven beschikbaar, maar de zender (of een gateway) heeft deze i.v.m. beveiliging, privacy of andere redenen afgeschermd. Er kan evt. een aanvullend mechanisme zijn om de informatie te verkrijgen.

XML-voorbeelden

Omdat geen enkel gegeven in een HL7v3 bericht feitelijk het datatype ANY kan hebben, wordt hier het gebruik van het attribuut @nullFlavor toegelicht op basis van een aantal specifieke datatypes (die het attribuut @nullFlavor dus overerven van ANY). Bij de beschrijvingen in het vervolg van deze implementatiegids wordt verder uitgegaan van niet-null waarden.

1) Het aanvraag/voorschriftijdstip is verplicht aanwezig in een HL7v3 bericht, maar de zendende applicatie weet simpelweg niet wanneer de aanvraag of het voorschrift is uitgeschreven (ook al wordt het concept wel ondersteund, omdat het required is).

```
<author>
  <time nullFlavor="UNK"/>
</author>
```

2) Een applicatie kent de mogelijkheid om gegevens te registreren voor patiënten waarvan (nog) geen burgerservicenummer bekend is, maar dit later alsnog aan de geregistreerde gegevens te koppelen. In dat geval kan de persoonsidentificatie (indien verplicht aanwezig in het betreffende HL7v3 bericht), als volgt zijn weergegeven:

```
<patient>
  ...
```

```

<person>
  <id nullFlavor="NAV"/>
  ...
</person>
</patient>

```

3) Bij een bepaald gegeven is in de specificaties een vast vocabulairedomein aangegeven, zonder de mogelijkheid om extra waarden toe te voegen. De zendende applicatie kan echter geen van de waarden in het betreffende domein toepassen (een voorbeeld van deze situatie is het doorgeven van magistraal bereide medicatie).

```

<medicationKind>
  <code nullFlavor="OTH"/>
</medicationKind>

```

In een dergelijke situatie kan wel gebruik worden gemaakt van `<originalText>` bij het datatype CD (en daarvan afgeleide datatypen), zie §6.12. Dit is één van de weinige situaties waarbij `nullFlavor` kan voorkomen in combinatie met andere gegevens.

```

<medicationKind>
  <code nullFlavor="OTH">
    <originalText>Mijn magistrale receptuur</originalText>
  </code>
</medicationKind>

```

4) Bij de specificatie van de ingrediënten van een bepaald medicijn moet worden aangegeven dat er sporen ijzer in zitten, hoewel de precieze hoeveelheid niet bekend is (en niet relevant i.v.m. de geringe hoeveelheid). In dat geval zou kunnen worden aangegeven dat sprake is van een TRC (trace) hoeveelheid van onbekende grootte.

```

<ingredient>
  <quantity nullFlavor="TRC"/>
</ingredient>

```

5) Het privacy filter van een bepaalde applicatie (of een tussenliggende gateway) heeft besloten dat een bepaald gegeven niet doorgegeven mag worden. De betreffende waarde wordt vervangen door een `nullFlavor` van het type MSK (masked) om het af te schermen. Let op dat de ontvanger nog wel weet dat het gegeven beschikbaar was!

```

<subject>
  <patient>
    ...
    <addr nullFlavor="MSK"/>
    ...
  </patient>
</subject>

```



Let op dat het gebruik van het attribuut `@nullFlavor` binnen een element in een HL7v3 bericht normaal gesproken betekent dat geen enkel ander attribuut of element onder het betreffende element aanwezig mag zijn. Het attribuut `@nullFlavor` zal dus niet samen met inhoudelijke informatie voorkomen (de aanwezigheid van het attribuut `@nullFlavor` geeft immers aan dat deze informatie juist ontbreekt in het element). Een uitzondering

op deze regel vorm het gebruik van *@nullFlavor* "OTH" in combinatie met het element `<originalText>` bij datatype CD (en afgeleide datatypen).



Er is helaas binnen HL7 nog steeds onduidelijkheid over hoe *xsi:nil* moet worden gebruikt. Op zich zijn *nullFlavor="NI"* en *xsi:nil* semantisch equivalent. Sommige XML-processoren vereisen echter dat naast het attribuut *@nullFlavor* ook expliciet *xsi:nil="true"* wordt aangegeven in het betreffende XML-element.



Wanneer wordt een nullFlavor gebruikt en wanneer niet?

- De cardinaliteit van een berichtonderdeel is 0.. (d.w.z. het is optioneel aanwezig).
 - als per definitie nooit een waarde beschikbaar is:
 - niet meesturen
 - als soms een waarde beschikbaar is:
 - *@nullFlavor* sturen als er geen waarde is
 - als altijd een waarde beschikbaar is:
 - waarde sturen
- De cardinaliteit van een berichtonderdeel is 1.. (d.w.z. het is verplicht aanwezig).
 - als per definitie nooit een waarde beschikbaar is:
 - *@nullFlavor* sturen
 - als soms een waarde beschikbaar is:
 - *@nullFlavor* sturen als er geen waarde is
 - als altijd een waarde beschikbaar is:
 - waarde sturen

6.6 ANY (het generieke datatype)

Attributen:

@nullFlavor ontbrekende waarde (CS)

Elementen:

geen

Dit abstracte datatype is de basis voor alle andere datatypen. Geen enkele waarde binnen een HL7v3 bericht heeft feitelijk het datatype ANY, maar elk datatype binnen HL7v3 is wel een specialisatie van ANY. Dat wil ook zeggen dat elk ander datatype de attributen van ANY door overerving overneemt (zie "Ontbrekende gegevens: nullFlavors").

Het ANY type komt af en toe voor in de HL7 modellen, meestal als het gaat om de waarde van klinische bevindingen of bepalingen. Het ANY type komt echter in XML-berichten niet voor, ANY wordt altijd vervangen door een bepaald datatype in een bericht.

Een voorbeeld is de waarde van observaties (subelement *value* in een *observation*), omdat in het model nog niet bekend is welke type waarde er van toepassing is. Het datatype van de waarde is pas bij de totstandkoming (instantiatie) van de interactie bekend. In de interactie wordt op die plaats dan ook het betreffende datatype meegegeven met behulp van het structuurattribuut *xsi:type*. Op deze wijze beschikt de ontvanger over de benodigde informatie om de waarde te kunnen valideren en verwerken.

Attributen van een element met dit datatype zijn:

@nullFlavor ontbrekende waarde (CS)

Hieronder worden een aantal voorbeelden aangegeven, echter geldt voor elke datatype dat er een nullFlavor kan worden aangegeven (tenzij het element mandatory is).

XML-voorbeelden

1) ANY geïntantieerd als CE

```
<value xsi:type="CE" code="N11.9" codeSystem="2.16.840.1.113883.6.3"/>
```

2) ANY geïntantieerd als CD

```
<value xsi:type="CD" code="C1" codeSystem="2.16.840.1.113883.2.4.99"
  displayName="Waarschuwing: de gevonden naamgegevens
  wijken af van de naamgegevens in de vraag."/>
```

3) ANY geïntantieerd als PQ

```
<value xsi:type="PQ" value="12" unit="ml"/>
```

4) ANY geïntantieerd als ED

```
<value xsi:type="ED">dit is een tekst met de reden</value>
```


6.7 BL (Boolean – true/false)

Attributen:

@value..... waarde (true|false)

Elementen:

geen

Het datatype BL (Boolean) heeft betrekking op zogenaamde tweewaardige logica. Een gegeven van dit type kan slechts de waarden "true" of "false" bevatten, of een nullFlavor indien de conformance dit toe staat (d.w.z. als het gegeven niet mandatory is).

Elke waarde (of tweetal waarden) van het type Boolean kent de volgende bewerkingen:

Tabel voor Booleaanse logica (een nullFlavor (NULL in de tabel) betekent dat de waarde ontbreekt)											
NOT			AND	true	false	NULL		OR	true	false	NULL
true	false		true	true	false	NULL		true	true	true	true
false	true		false	false	false	false		false	true	false	NULL
NULL	NULL		NULL	NULL	false	NULL		NULL	true	NULL	NULL

Een element met datatype BL heeft (indien het niet-null is) het attribuut value. De mogelijke waarden zijn "true" of "false", wat aangeeft of het gegeven waar of onwaar is

@value..... waarde (true|false)

XML-voorbeelden

1) Een persoon (of andere 'living subject') is overleden.

```
<livingSubject>
  <!-- ... -->
  <deceasedInd value="true"/>
</livingSubject>
```

2) Een associatie is non-conductive (d.w.z.: geeft geen context door).

```
<support2 contextConductionInd="false">
  <!-- ... -->
</support2>
```

In de bovenstaande situatie is het datatype BL niet van toepassing op een XML-element, maar op een attribuut. In dat geval krijgt het betreffende attribuut direct de Booleaanse waarde (het neemt dus als het ware de rol over dat het attribuut value anders heeft).

6.8 BIN (Binary data - binaire gegevens)

BIN is het abstracte type voor multimediagegevens. Het BIN type komt niet zelfstandig in de berichten voor. Indien multimedia gegevens in een bericht moeten worden opgenomen, dient hiervoor het datatype Encapsulated Data (ED) gebruikt te worden.

6.9 ED (Encapsulated Data - ingekapselde gegevens)

Attributen:

@representation..... codering (cs)

@mediaType.....soort gegevens (cs)

Elementen:

<reference>verwijzing (TEL)

<thumbnail> verkleinde of symbolische weergave (ED)

text().....gegevens (st)

Dit is een algemeen type voor allerlei multimediegegevens. Het zal in Nederland vooralsnog gebruikt worden voor tekst, al dan niet voorzien van (eenvoudige) opmaak.

ED is een samengesteld type: het bevat XML-elementen en -attributen. De gegevens (tekst, beelden) bevinden zich binnen het ED element in de vorm die gespecificeerd is door het attribuut *@representation*.

Het ED type kent twee vormen van gebruik:

1. *Inline data*

De volledige gegevens worden in het type verzonden. Deze vorm wordt voornamelijk gebruikt voor tekst.

2. *By reference*

Een verkorte versie van de gegevens wordt gevat in een "thumbnail", daarnaast wordt voorzien in een "reference" naar de volledige gegevens. Deze vorm zal in Nederland vooralsnog niet gebruikt worden.



Indien binnen een XML-element van datatype ED een 'embedded' XML-structuur voorkomt (afgezien van de elementen van het datatype zelf), dan dient hiervoor een aparte XML-namespace aangeduid te worden. Dit omdat de extra elementen, die een buiten HL7 bepaalde syntax hebben, strikt genomen dus niet binnen de HL7 namespace gedefinieerd zijn.

@representation..... codering (cs)

Gebruik van dit attribuut is optioneel. Er zijn twee coderingen mogelijk.

Tabel representation attribuutwaarden	
code	Definitie
TXT	Voor tekstuele gegevens. Dit is het standaardtype, als geen <i>representation</i> is aangegeven wordt uitgegaan van TXT.
B64	Base 64 codering wordt gebruikt voor alle andere multimedia gegevens.

@mediaType.....soort gegevens (cs)

Dit attribuut geeft het soort gegevens aan. In Nederland worden voorlopig alleen de volgende gegevenssoorten ondersteund, zoals in de volgende tabel weergegeven:

Tabel mediaType attribuutwaarden (gegevenssoorten)			
code	naam	Status	definitie

Tabel mediaType attribuutwaarden (gegevenssoorten)			
<i>code</i>	<i>naam</i>	<i>Status</i>	<i>definitie</i>
text/plain	Plain Text	verplicht	Voor willekeurige tekst. Dit is het 'default' type. In deze vorm is het gelijk aan het ST type.
text/html	HTML Text	aanbevolen	Bedoeld voor opgemaakte tekst in HTML format. HTML is voldoende voor de meeste toepassingen waarbij opmaak gewenst is. HTML is platformafhankelijk en wordt breed gebruikt.
audio/basic	Basic Audio	verplicht	Enkelkanaals audioformaat voor spraak. Alhoewel ondersteuning van dit formaat verplicht is, zal het in Nederland nauwelijks gebruikt worden.
audio/mpeg	MPEG audio layer 3	verplicht	MPEG-1 Audio layer-3 (ook bekend als MP3) is op dit moment de standard voor gecomprimeerde audio.
image/png	PNG Image	verplicht	Portable Network Graphics (PNG) [http://www.cdrom.com/pub/png] is een verliesloze compressie voor beeldbestanden. Waarvoorheen GIF werd gebruikt, dient nu PNG te worden ondersteund.
image/jpeg	JPEG Image	verplicht	Dit formaat wordt gebruikt voor hoge resolutie foto's en andere beeldbestanden. De compressie is niet zonder verliezen waardoor dit formaat niet altijd geschikt is voor diagnostische doeleinden. JPEG zal voornamelijk worden gebruikt voor 'thumbnails' van grote (DICOM) bestanden.
video/mpeg	MPEG Video	verplicht	MPEG is een internationale standaard voor video beelden. Het is wijd gebruikt en open-source implementaties zijn beschikbaar.
text/rtf	RFT Text	facultatief	Het Rich Text Format wordt breed gebruikt om tekstverwerkdokument te delen. Echter, RTF heeft de nodige compatibiliteitsproblemen, aangezien het afhankelijk is van de tekstverwerker. Kan van belang zijn als uitgewisselde documenten te wijzigen moeten zijn door de ontvanger.
application/pdf	PDF	aanbevolen	Het Portable Document Format wordt aangeraden voor geschreven tekst die compleet is opgemaakt en read-only is. PDF is platformafhankelijk, wijd gebruikt en open. Het heeft vrij verkrijgbare tools voor creatie en weergave van bestanden.
application/dicom	DICOM	aanbevolen	De Digital Imaging and Communication in Medicine standaard wordt gebruikt voor bijvoorbeeld röntgenbeelden. http://www.rfc-editor.org/rfc/rfc3240.txt
vnd.oasis.opendocument.formula	ODF	facultatief	OASIS Open Document Format for Office Applications, is een open standaard voor het bewaren en/of uitwisselen van tekstbestanden, rekenbladen, grafieken en presentaties. De officiële standaard is NEN-ISO/IEC 26300:2007.

Toegestane *mediaTypes* in een feitelijke implementatie kunnen verder ingeperkt worden.



Er wordt nadrukkelijk geen begrenzing aangegeven voor de maximale grootte van de waarde in een element van dit datatype. Ook bij specifieke elementen van een HL7v3 bericht gebeurt dit meestal niet, omdat de maximale grootte in principe niet door de standaard wordt ingeperkt.

Afspraken hierover zullen meestal op implementatieniveau worden gemaakt. Zonder specifieke afspraken moet de ontvanger in staat zijn om Encapsulated Data (bijvoorbeeld plain text) van willekeurige grootte te verwerken.

<reference> *verwijzing (TEL)*

Dit element wordt alleen in combinatie met het 'thumbnail' element gebruikt. Het bevat de verwijzing naar de volledige gegevens in de vorm van een TEL (telecom) type.

<thumbnail> *verkleinde of symbolische weergave (ED)*

Dit element bevat een verkleinde of symbolische weergave van de volledige informatie. Het 'thumbnail' element kan bijvoorbeeld gebruikt worden om een JPEG 'cover-picture' van DICOM videostreams door te geven.

Aangezien 'thumbnail' een ED type is kunnen hier alle, eerder genoemde, gegevenssoorten in geplaatst worden.

XML-voorbeelden

Simpele vrije tekst:

```
<text xsi:type="ED" mediaType="text/plain">De patiënt heeft een lastige  
thuisituatie doordat kinderen op grote afstand wonen.</text>
```

Voorbeeld van gebruik thumbnail en reference:

```
<value xsi:type="ED" mediaType="image/png" representation="B64">  
  <reference value="http://radiology.iuuc.edu/xrays/128s.png">  
    <useablePeriod xsi:type="IVL_TS">  
      <low value="200007200845" />  
      <high value="200008200845" />  
    </useablePeriod>  
  </reference>  
  <thumbnail mediaType="image/jpeg" representation="B64">  
MNYD83jmMdomSJUEdmde9j44zmMir6edjzMMIjdMDSswdIJdksIJR3373jeu83  
6edjzMMIjdMDSswdIJdksIJR3373jeu83MNYD83jmMdomSJUEdmde9j44zmMir  
...  
omSJUEdmde9j44zmMiromSJUEdmde9j44zmMirdMDSswdIJdksIJR3373jeu83  
4zmMir6edjzMMIjdMDSswdIJdksIJR3373jeu83==  
  </thumbnail>  
</value>
```

6.10 ST (String of Characters - reeks tekens)

Attributen:

@representation..... codering ("TXT")

@mediaType..... soort gegevens ("text/plain")

Elementen:

text()..... gegevens (st)

Dit type is bedoeld voor vrije tekst in de meest eenvoudige vorm. ST is een specialisatie van het ED type, het mediaType is vastgezet op 'text/plain' en de representation op 'TXT'. De andere attributen en elementen van ED mogen niet gebruikt worden bij het ST type.

Het ST type wordt voornamelijk gebruikt in andere datatypen zoals AD en PN. In het algemeen geldt dat voor tekst beter het ED type gebruikt kan worden aangezien dit toch gereduceerd kan worden tot een ST type door de attributen op de juiste manier te vullen.

XML-voorbeeld

Dit voorbeeld is nagenoeg gelijk aan het eerste voorbeeld bij [ED](#), functioneel zijn ze identiek.

```
<text xsi:type="ST">De patiënt heeft een lastige thuissituatie doordat kinderen op grote afstand wonen.</text>
```

6.11 SC (String with Code - reeks tekens voorzien van code)

Attributen:

@code..... code (cs)

@codeSystem codesysteem (oid)

@displayNameconceptbeschrijving (st)

@codeSystemNamenaam van het codesysteem (st)

@codeSystemVersion versie van het codesysteem (st)

Elementen:

text().....gegevens (st)

Dit type is een extensie van het ST type aangevuld met de attributen van het [CV](#) type. Hierdoor is het mogelijk tekst nader te specificeren door middel van een code. Het stringgedeelte in dit datatype is verplicht; het codegedeelte is optioneel.

Het SC datatype wordt o.a. gebruikt als onderdeel van het [AD](#) (adres) datatype. Het is daardoor mogelijk om bijvoorbeeld het land, naast vrije tekst, ook van een code te voorzien.

6.12 CD (Concept Descriptor - gecodeerde gegevens)

Attributen:

@code..... code (cs)

@codeSystem..... codesysteem (oid)

@displayName.....conceptbeschrijving (st)

@codeSystemName.....naam van het codesysteem (st)

@codeSystemVersion..... versie van het codesysteem (st)

Elementen:

<qualifier>..... aanvulling op de code (LIST<CR>)

<originalText>.....oorspronkelijke tekst ([ST](#))

<translation>..... vertalingen van de code ([SET](#)<[CD](#)>)

Dit datatype specificeert een concept door middel van een code en het bijbehorende codesysteem (met uitzondering van situaties waarin geen code beschikbaar is).

Van het generieke datatype CD bestaan de nodige specialisaties, die verschillen in de eigenschappen (properties) van het datatype die al dan niet ondersteund worden:

Datatype Property	CD	CE	CV	CO	CS
@code	X	X	X	X	X
@codeSystem	X	X	X	X	
@codeSystemName	X	X	X	X	
@codeSystemVersion	X	X	X	X	
@displayName	X	X	X	X	
<originalText>	X	X	X	X	
<translation>	X	X			
<qualifier>	X				

Tabel 1 Elementen en attributen van datatype CD en specialisaties

Omdat de CE variant in de praktijk de meest gebruikte is, zal daar de algemene toelichting voor het gebruik van de meeste properties in Nederland worden gegeven. Het enige attribuut dat uniek is voor CD (*qualifier*) wordt hieronder kort toegelicht.

<qualifier>..... aanvulling op de code (CR)

Het optionele element qualifier bevat een nadere precisering van het concept zoals beschreven door het code attribuut. Het element qualifier is noodzakelijk voor bijvoorbeeld post-coördinatie van SNOMED CT concepten. In deze versie van de implementatiegids wordt dit element en het gebruikte CR datatype niet nader uitgewerkt.

6.13 CE (Coded with Equivalents - gecodeerde gegevens, met equivalenten)

Attributen:

@code..... code (cs)

@codeSystem codesysteem (oid)

@displayNameconceptbeschrijving (st)

@codeSystemNamenaam van het codesysteem (st)

@codeSystemVersion versie van het codesysteem (st)

Elementen:

<originalText>oorspronkelijke tekst ([ST](#))

<translation> vertalingen van de code ([SET](#)<[CD](#)>)

Dit datatype specificeert een concept door middel van een code en het bijbehorende codesysteem (met uitzondering van situaties waarin geen code beschikbaar is). Optioneel bevat het één of meer equivalenten met behulp van andere codesystemen.

Voorbeeld: Het concept "Mannelijk" wordt afhankelijk van het gebruikte codesysteem geïdentificeerd door bijvoorbeeld de code "M" of de code "1". De ontvanger is alleen in staat een code eenduidig te interpreteren indien naast de code eveneens het gebruikte codesysteem geïdentificeerd wordt. De combinaties ("M", HL7v3 Tabel *AdministrativeGender*) of de vertaling daarvan ("1", ABC-ZIS Systeem geslachtscodetabel) zijn eenduidig te interpreteren.

Attributen en elementen van dit datatype zijn:

@code..... code (cs)

Verplicht. Bevat de code, een klassificatie van het concept dat beschreven is in het in codeSystem aangegeven codesysteem.

@codeSystem codesysteem (oid)

Verplicht. Bevat de identificatie van het codesysteem (tabel, terminologie). Ter identificatie wordt gebruik gemaakt van een OID.

Een ISO Object Identifier (OID) is een wereldwijd unieke string die bestaat uit nummers en punten (bijvoorbeeld "2.16.840.1.113883.3.1"). Volgens de ISO definitie bestaan OID's uit paden volgens een boomstructuur, waarbij het meest linkse getal de root en het meest rechtse getal de leaf (een blad, eindpunt) representeert. Het nummer is gegarandeerd wereldwijd uniek omdat het is gebaseerd op een systeem van gedelegeerde verantwoordelijkheid. Elke tak onder een root in de boomstructuur correspondeert met een domein waarbinnen een organisatie de uitgave van OID's beheert. De Stichting HL7 Nederland publiceert een tabel met OID's. Informeer bij twijfel bij de Stichting HL7 Nederland welke (eventueel nieuw te registreren) OID gebruikt moet worden. Zie hoofdstuk 3 voor nadere informatie over codesystemen en OID's.

@displayNameconceptbeschrijving (st)

Optioneel. Een tekstuele omschrijving van het concept. Dit is de omschrijving van de code, zoals die in het zendende systeem aan de gebruikers wordt getoond. Aan de waarde van dit attribuut mag geen betekenis worden toegekend, anders dan het tonen ervan aan een gebruiker om de code te verduidelijken. Een *displayName* kan nooit voorkomen zonder bijbehorende code en moet qua betekenis bij de code aansluiten.



Er mag nooit van de ontvanger verwacht worden dat deze de *displayName* controleert op consistentie met de bijbehorende code. De ontvanger mag zelf bepalen of de eigen omschrijving (indien aanwezig) of de *displayName* van de verzender gebruikt wordt voor weergave. Bij gebruik van een *nullFlavor* mag geen *@displayName* maar wel het kindelement *originalText* gebruikt worden om een niet-gecodeerde omschrijving door te geven.

<originalText>oorspronkelijke tekst (st)

Optioneel. Een tekstuele omschrijving van het concept. Dit is de tekst op basis waarvan al dan niet een code werd toegewezen in het zendende systeem. Merk op dat dit dus andersom geredeneerd is als bij *displayName*, waar de omschrijving juist bepaald wordt door de code. Dit betekent dat *originalText* ook nadrukkelijk kan voorkomen in gevallen waarin geen code toegewezen kon worden. In dat geval is de *originalText* de manier om de tekst door te geven die blijkbaar niet te vatten was in het betreffende codesysteem.



Merk op dat *originalText* niet als attribuut van een XML-element met type CE wordt doorgegeven, maar als **subelement**. Zie ook de XML-voorbeelden.



Er mag nooit van de ontvanger verwacht worden dat deze de *originalText* controleert op consistentie met een eventueel afgeleide code. Bij gebruik van een *nullFlavor* in dit datatype kan de *originalText* uitgewisseld worden als alternatief voor een gecodeerde waarde (zie ook voorbeeld hieronder).

@codeSystemNamenaam van het codesysteem (st)

Optioneel. Een tekstuele vorm van de naam van het codesysteem dat de code bevat. Aan de waarde van dit attribuut mag geen betekenis worden toegekend, anders dan het tonen ervan aan een gebruiker om de achtergrond van een code te verduidelijken.

@codeSystemVersionversie van het codesysteem (st)

Optioneel. Een tekstuele vorm van de versie van het codesysteem dat de code bevat. Aan de waarde van dit attribuut mag geen betekenis worden toegekend, anders dan het tonen ervan aan een gebruiker om de achtergrond van een code te verduidelijken.

Verschillende versies van een codesysteem (dat wil zeggen: bij wijzigingen die niet compatibel zijn met een voorgaande versie van het codesysteem) worden geïdentificeerd door aparte OID's in het *@codeSystem* attribuut. Het ontvangende systeem mag om

deze reden nooit gebruik maken van de waarde van *codeSystemVersion* om de code te interpreteren.

<translation> vertalingen van de code (SET<CD>)

Optioneel kindelement. Nul of meer vertalingen van het concept met behulp van alternatieve codesystemen.

Alle vertalingen dienen één en hetzelfde concept aan te duiden. Hierbij is het toegestaan dat de vertalingen "minder genuanceerd/gedetailleerd" zijn dan het originele concept.

Voorbeeld: Het concept "Granny Smith" kan, indien een codesysteem niet het juiste niveau aan detail bevat, vertaald worden in het concept "Appel". Het concept "Appel" mag echter nooit worden vertaald in het meer gedetailleerde concept "Granny Smith". Het concept "Appel" mag eveneens niet worden vertaald in het concept "Groen": dit zijn wellicht gerelateerde concepten, maar qua betekenis is het geen vertaling van het originele concept.

XML-voorbeelden

```
<administrativeGenderCode code="M" codeSystem="2.16.840.1.113883.5.1"/>
```

```
<value xsi:type="CE" code="Z94.0" codeSystem="2.16.840.1.113883.6.3" displayName="Longontsteking" codeSystemName="ICD-10"/>
```

```
<code code="SF36" codeSystem="2.16.840.1.113883.2.4.4.13"/>
```

```
<code code="13715119" codeSystem="2.16.840.1.113883.2.4.4.8"
  codeSystemName="G-Standaard Artikel"
  displayName="ABSORIN ONDERLEGGERS 60X90CM 120G FLUFF 60920">
  <translation code="753696" codeSystem="2.16.840.1.113883.2.4.4.7"/>
</code>
```

```
<code nullFlavor="OTH">
  <originalText>Een zelfgemaakt medicijn</originalText>
</code>
```

6.14 CV (Coded Value - gecodeerde waarde)

Attributen:

@code..... code (cs)

@codeSystem..... codesysteem (oid)

@displayName.....conceptbeschrijving (st)

@codeSystemName.....naam van het codesysteem (st)

@codeSystemVersion..... versie van het codesysteem (st)

Elementen:

<originalText>.....oorspronkelijke tekst (ST)

Dit datatype specificeert een concept door middel van een code en het bijbehorende codesysteem (met uitzondering van situaties waarin geen code beschikbaar is). Het enige verschil tussen het CV datatype en het CE datatype is het feit dat CE vertalingen van een concept kan bevatten. Zie de beschrijving van CE, met uitzondering van het element *translation*, voor het gebruik van het CV datatype in Nederland.

XML-voorbeelden

```
<administrativeGenderCode code="M" codeSystem="2.16.840.1.113883.5.1"/>
```

```
<value xsi:type="CV" code="Z94.0" codeSystem="2.16.840.1.113883.6.3"
  displayName="Longontsteking" codeSystemName="ICD-10"/>
```

```
<code code="SF36" codeSystem="2.16.840.1.113883.2.4.4.13"/>
```

```
<code code="13715119" codeSystem="2.16.840.1.113883.2.4.4.8"
  codeSystemName="G-Standaard Artikel"
  displayName="ABSORIN ONDERLEGGER 60X90CM 120G FLUFF 60920"/>
```

```
<code nullFlavor="OTH">
  <originalText>Een zelfgemaakt medicijn</originalText>
</code>
```

6.15 CO (Coded Ordinal - sorteerbare gecodeerde waarde)

Attributen:

@code..... code (cs)

@codeSystem..... codesysteem (oid)

@displayName.....conceptbeschrijving (st)

@codeSystemName.....naam van het codesysteem (st)

@codeSystemVersion..... versie van het codesysteem (st)

Elementen:

<originalText>.....oorspronkelijke tekst ([ST](#))

Dit datatype specificeert een concept door middel van een code en het codesysteem (de tabel) waar de code uit afkomstig is.

Het enige verschil tussen het CO datatype en het [CV](#) datatype is het feit de concepten opgenomen in het CO datatype een bepaalde volgorde bezitten, ze kunnen worden gesorteerd. Zie de beschrijving van [CE](#), voor een beschrijving van het gebruik van de bovenstaande attributen en elementen in Nederland.

6.16 CS (Coded Simple - eenvoudige code)

Attributen:

@code..... code (cs)

Elementen:

geen

Dit datatype specificeert een concept door middel van een code uit een voorgedefinieerd codesysteem (de tabel) waar de code uit afkomstig is. Dit datatype wordt veelal toegepast daar waar het gaat om het gebruik van vaste, HL7 gedefinieerde, tabellen.

@code..... code (cs)

Verplicht. Bevat de code (mnemonic), een identificatie van het concept dat beschreven is in het voorgedefinieerde codesysteem.

XML-voorbeelden

```
<Observation classCode="OBS" moodCode="EVN"/>
```

```
<statusCode code="active"/>
```

6.17 II (Instance Identifier - objectidentificatie)

Attributen:

@rootidentificatiesysteem (oid)

@extension identificatie (st)

@assigningAuthorityName naam van de uitgevende organisatie (st)

Elementen:

geen

Dit datatype specificeert de identificatie van objecten. Daarbij horen bijvoorbeeld identificaties voor organisaties of personen. Een element van het II datatype bevat een wereldwijde unieke identificatie van een object.

Attributen van een element met dit datatype zijn:

@rootidentificatiesysteem (oid)

In Nederland is dit een verplicht attribuut. Root wordt voor twee verschillende doeleinden gebruikt (of het een, of het ander):

1. Als een unieke identifier (in Nederland: een OID) voor het identificatiesysteem waarbinnen de extension gegenereerd (en uniek) is. In dit geval moet @extension (zie hieronder) een waarde bevatten. Een identificatiesysteem wordt gebruikt om personen, systemen, instellingen en andere stoffelijke zaken te kunnen identificeren. Voorbeelden zijn: 'Nederlands rijbewijsnummer', Burgerservicenummer (BSN), afdelingsidentificatie binnen het St. Josef Ziekenhuis, Unieke Zorgverlener Identificatie (UZI), en het Vektis AGB-Z Organisatienummer.
2. Als een unieke identifier (in Nederland: een OID) van een object. In dit geval bevat @extension (zie hieronder) geen waarde.

Een ISO Object Identifier (OID) is een wereldwijd unieke string die bestaat uit nummers en punten (bijvoorbeeld "2.16.840.1.113883.3.1").



Het gebruik van extensies, in het datatype [II](#) komt deze in het XML-attribuut *@extension*, wordt door HL7 Nederland ten zeerste aangeraden. Door een OID toe te wijzen aan een identificatiesysteem (bijv: "BSN") wordt hergebruik van die OID bevorderd. Het alternatief is een OID toe te kennen aan ieder object (bijv: aan iedere individuele burger). De herbruikbaarheid van die uitgegeven OID's is minimaal.

Er zijn omstandigheden waarin (op welke gronden dan ook) OID's worden toegekend aan individuele objecten (wat dan weer leidt tot een II waarin alleen @root een waarde bezit). Dit geldt onder andere voor de identificatie van DICOM-objecten en van HL7 v3 Template-id's.



Merk op: in de context van AORTA gelden de volgende extra eisen:

- @root mag een maximale lengte van 128 karakters hebben.

Zie hoofdstuk 3 voor nadere informatie over identificatiesystemen en OID's.

@extension identificatie (st)

Het attribuut @extension is verplicht indien @root een identificatiesysteem identificeert. Indien @root een identificatiesysteem identificeert dan bevat @extension een unieke karakterreeks binnen de context van dat identificatiesysteem.

De lengte van de extensie-string, en het gebruik van eventuele voorloophnullen in de extensie, en de hoeveelheid daarvan, wordt bepaald door de beheerder van het identificatiesysteem.

Het attribuut @extension is niet toegestaan/mogelijk als @root een identificatie is. Zie de twee gebruiksvormen/doeleinden van @root hier boven.



Merk op: in de context van AORTA geldt de volgende extra eis:

- @extension mag een maximale lengte van 64 karakters hebben en is verplicht voor alle gebruikte Instance Identifiers, met als enige uitzondering het templateId attribuut.

@assigningAuthorityName naam van de uitgevende organisatie (st)

Optioneel attribuut. Een tekstuele vorm van de zogenaamde 'assigning authority'.

- Indien @root een identificatiesysteem identificeert (en @extension een waarde bevat) dan bevat @assigningAuthorityName de organisatie die het identificatiesysteem beheert (dat wil zeggen: de mogelijkheid heeft extensies uit te geven). Voorbeeld: "Ministerie van binnenlandse zaken" voor het BSN identificatiesysteem.
- Indien @root een object identificeert (en @extension leeg is) dan bevat @assigningAuthorityName de organisatie die de identificatie heeft bepaald (dat wil zeggen: de root OID heeft uitgegeven). Voorbeeld: "Afdeling Beeldvorming UMCG" voor OIDs uitgegeven aan aldaar aangemaakte beelden.

Aan de waarde van dit attribuut mag geen betekenis worden toegekend. De waarde van dit attribuut mag aan een gebruiker worden getoond om de achtergrond van de identificatie te verduidelijken.

FAQ Implementatieaspecten

Bij de identificatie van een bepaalde set gegevens (bijv. een recept, of een labbepaling) zullen applicaties veelal @extension van een "primary key" voorzien die in de applicatie zelf, of in haar database, gebruikt wordt om de gegevens te identificeren.

Wat dient men te doen als indien er binnen de applicatie geen "primary key" bestaat, daar waar HL7 versie 3 modellen er een vereisen?

- Als @extension geen waarde is die uit de database van het verzendende systeem kan worden gehaald, dan kan (als extensie) een GUID worden aangemaakt. Een GUID is weliswaar niet gegarandeerd wereldwijd en eeuwig uniek, maar heeft wel

andere voordelen boven het gebruik (als extensie) van een oplopende teller (bijv. de garantie dat geen dubbelingen ontstaan bij terugzetten van de backup na een crash). Uiteraard moet deze GUID vervolgens worden opgeslagen omdat hij reproduceerbaar moet zijn voor wanneer een extern systeem (opnieuw) wil refereren aan het gegeven.

XML-voorbeelden

1) Extension en OID-gebaseerde root

```
<id extension="0004453645" root="2.16.528.1.1007.3.3.99999.12.1"/>
```

2) templateId met alleen OID-gebaseerde root

```
<templateId root="2.16.840.1.113883.2.4.15.3.427.1.13234453645"/>
```

3) assigningAuthorityName, organisatie die id uitgegeven heeft

```
<id extension="100197231" root="2.16.840.1.113883.2.4.6.3"
  assigningAuthorityName="Ministerie van Binnenlandse Zaken"/>
```

4) OID-gebaseerde root en niet-numerieke extension

```
<id extension="JANS2" root="2.16.528.1.1007.3.3.99999.14"
  assigningAuthorityName="Alfa ziekenhuis"/>
```

6.18 URL (Universal Resource Locator)

Attributen:

@value..... waarde (url)

Elementen:

geen

Dit is een telecommunicatie adres gespecificeerd volgens de Internet standaard "RFC 1738 [<http://www.ietf.org/rfc/rfc1738.txt>]". De URL geeft een protocol en een contactpunt aan, die voor dit protocol gedefinieerd zijn. Bekende voorbeelden zijn telefoon- en faxnummers, een e-mailadres, hyperlinkreferenties, file transfer protocol (FTP) referenties enz.

URL's hebben een standaardrepresentatievorm als strings in het formaat *scheme:address*; de meest bekende *schemes* zijn opgenomen in de volgende tabel. De volledige lijst van *schemes* wordt bijgehouden bij IANA. Zie [IANA - schemes].

Het *address* gedeelte van een URL is een string waarvan het formaat bepaald is door de URL scheme.

Tabel Domain URLScheme:		
code	name	definition
tel	Telephone	Telefoonnummer [draft-antti-telephony-url-11.txt].
fax	Fax	Een nummer voor een faxtoestel [draft-antti-telephony-url-11.txt].
mailto	Mailto	Electronic mail address [RFC 2368].
http	HTTP	Hypertext Transfer Protocol [RFC 2068].
https	HTTPS	Hypertext Transfer Protocol over SSL/TLS [RFC 2818].
ftp	FTP	File Transfer Protocol (FTP) [RFC 1738].
mllp	HL7 Minimal Lower Layer Protocol	Het traditionele HL7 Minimal Lower Layer Protocol. De URL heeft de vorm van een IP URL, bijvoorbeeld mllp://<host>:<port>/ met <host> als IP adres of DNS hostname en <port> als port nummer waar de MLLP dienst bereikbaar is.
file	File	computer specifieke locale bestandsnamen [RFC 1738]. Deze schemes werken alleen voor lokale bestanden. Wordt nauwelijks gebruikt omdat bij een zender / ontvanger scenario de ontvanger met een lokale bestandsnaam meestal niets kan.
nfs	NFS	Network File System Protocol [RFC 2224]. Wordt voor NFS servers gebruikt om bestanden te delen.
telnet	Telnet	Referentie naar een interactieve sessie [RFC 1738].
modem	Modem	Een telefoonnummer met een modem [draft-antti-telephony-url-11.txt].
x-hl7-applicatie	Applicatie Identificatie	OID die een HL7 berichtverzendende applicatie eenduidig identificeert. Binnen de basisinfrastructuur van Nictiz worden hier de door het LSP uitgedeelde applicatie ID's voor gebruikt.

6.19 TEL (Telecommunication Address - telecommunicatiecontact)

Attributen:

@value..... waarde (url)

@usegebruiksaanwijzing (set<cs>)

Elementen:

<useablePeriod> tijdsinterval (IVL_TS)

Hoewel de naamgeving van dit datatype wellicht anders doet vermoeden bestaat er geen speciaal datatype voor telefoonnummers. Het datatype TEL wordt gebruikt alle URL's die betrekking hebben op telecommunicatiefaciliteiten.

Details over de definitie van telefoonnummers zijn gedefinieerd in de Internet "RFC 2806 [<http://www.ietf.org/rfc/rfc2806.txt>] URLs for Telephone Calls". Bijvoorbeeld is

"tel:+31(299)6754-63" een telefoonnummer, "fax:+31(20)6571412-3" een faxnummer.

De globale absolute telefoonnummers met een "+" en landcode aan te geven heeft de voorkeur. Scheidingstekens kunnen worden toegevoegd (voor de betere leesbaarheid)

maar hebben geen betekenis voor de nummers. "tel:+31299675463" en

"fax:+312065714123" zijn dus indentiek aan de bovenstaande voorbeelden.

Elementen met dit datatype hebben als attributen:

@value..... waarde (URL)

Het *value* attribuut komt **verplicht** voor en bevat een URL (zie bij dat datatype).

Dit omvat dus zowel het scheme (bijvoorbeeld tel:, fax:, mailto:) als het bijbehorende adres.

@usegebruiksaanwijzing (CS)

Met het optionele use attribuut kunnen één of meerdere codes worden aangegeven om een systeem of gebruiker een advies te geven welk telecommunicatiecontact hij kan gebruiken afhankelijk van de doeleinden.

Tabel: Domain TelecommunicationAddressUse attribuutwaarden		
Code	Naam	Definitie
H	home address (woon- / verblijfadres)	Het secundaire telecommunicatiecontact, om een persoon te bereiken, kan meerdere malen voorkomen.
HP	primary contact (woon- / verblijfadres)	Het primaire telecommunicatiecontact, om een persoon te bereiken, kan hoogstens één keer voorkomen.
HV	vacation contact (vakantie telecommunicatiecontact)	Een vakantiehuis, om een persoon te bereiken tijdens de vakantie
WP	work place (werk)	Een telecommunicatiecontact op het werk. Eerste keus voor werkgerelateerde contacten. Is voor organisaties en zorgverleners het primaire telecommunicatiecontact.
AS	Antwoorddienst	Een persoon of dienst, om berichten achter te laten.
EC	Noodgevalcontact	Een telecommunicatiecontact in een noodgeval te

		gebruiken
PG	Pager	Een pager toestel, geschikt om te vragen om terug te bellen of een kort bericht achter te laten
MC	Mobielcontact	Een mobiele telefoon of toestel dat de eigenaar altijd met zich mee neemt, mag andere use codes bevatten

<useablePeriod> tijdsinterval (IVL_TS)

Het is mogelijk om de geldigheidsduur van een telecommunicatiecontact in te perken. Het element *useablePeriod* wordt in dat geval gebruikt om deze geldigheid met een tijdsinterval aan te duiden.

XML-voorbeeld

```
<telecom value="tel:+31.10.4765342"/>
```

```
<telecom value="tel:+31.20.4637846" use="WP"/>
```

```
<telecom value="mailto:dialyse@centrum-rotterdam.nl"/>
```

```
<telecom value="x-hl7-applicatie:2.16.840.1.113883.2.4.6.6.75023"/>
```

```
<telecom use="H" value=" tel:+3120-4637846">
  <useablePeriod>
    <low value="201110020800"/>
  </useablePeriod>
</telecom>
```

6.20 AD (Postal Address - adres)

Attributen:

@use *gebruiksaanwijzing (set<cs>)*

Elementen:

<delimiter> *scheidingsteken (ST)*

<country> *land (SC)*

<county> *gemeente (SC)*

<city> *woonplaats (ST)*

<postalCode> *postcode (SC)*

<houseNumber> *huisnummer (ST)*

<buildingNumberSuffix> *huisnummertoevoeging (ST)*

<streetName> *straatnaam (ST)*

<additionalLocator> *aanvullende locatieaanwijzing (ST)*

<useablePeriod> *tijdsinterval (IVL_TS)*

Het AD datatype wordt toegepast om een adres te identificeren. Elementen van dit datatype hebben een substructuur met verschillende elementen, die in een adres kunnen voorkomen. Een adres wordt in HL7 versie 3 weergegeven als een serie van *Address Name Parts*, bijvoorbeeld straatnaam, stad enz.³ Het datatype voor alle *Address Name Parts* is coded string (SC). Op dit moment zijn *postalCode*, *county* en *country* de enige elementen waar we codes toestaan.

Tabel: Domain AddressNamePartType element namen	
Element naam	Definitie
delimiter	Begrenzers [delimiters] worden geprint zonder witte ruimte te vormen [framing]. Wanneer er geen waardecomponent wordt geleverd, verschijnt de begrenzer als een regelonderbreking [line break].
country	Het land, bijvoorbeeld "Nederland". Het datatype van <i>country</i> is coded string (SC). Als het land gecodeerd wordt dienen de 2-karakter landcodes volgens ISO 3166-1 editie 2 (OID 1.0.3166.1.2.2) gebruikt te worden.
county	In Nederland wordt dit element gebruikt om de gemeente door te geven (in andere landen kan een ander type administratieve eenheid binnen een staat/provincie gebruikt worden). De gemeente kan, maar hoeft niet, overeen te komen met de stad. Sommige gemeenten, bijvoorbeeld "Waterland", hebben een naam die geheel afwijkt van de steden die erin gele-

³ In de HL7 standaard wordt de datatype AD ook toegestaan als zogenaamde "mixed content", dat betekent dat er delen van de gegevens door een *partType* element worden omsloten, delen niet (mix van tekst en XML elementen). Dit is in Nederland niet toegestaan.

	gen zijn. In het HL7 berichtenverkeer wordt de gemeente in Nederland alleen gebruikt in het kader van wettelijke identificatie van personen. Het datatype van <i>county</i> is coded string (SC). Als de gemeente gecodeerd wordt, dan dient GBA tabel 33 (OID 2.16.840.1.113883.2.4.6.14) gebruikt te worden. Code 0000 zoals aanwezig in de GBA Tabel 33 mag onder geen enkele omstandigheid voorkomen in een HL7 versie 3 bericht. Het equivalent van 0000 (onbekende gemeente) in HL7 versie 3 is het niet opnemen van een code.
city	De naam van een stad, dorp of ander woongebied of bezorgcentrum. Merk op: dit is de woonplaats, niet de eventuele gemeente waartoe de woonplaats behoort. Voorbeeld: <i>Haren</i> , gemeente Groningen.
postalCode	Een postcode, voor Nederlandse postcodes inclusief één spatie tussen het numerieke gedeelte en de twee letters, #### AA. Het datatype van postcode is coded string (SC). Als de postcode gecodeerd dient te worden verzonden, dan dient de postcodetabel (OID 2.16.840.1.113883.2.4.4.15) gebruikt te worden. De code is dan volgens NEN Norm 5825, zonder spatie, "9999AA".
houseNumber	Het nummer van een gebouw, huis of perceel langs de straat. Ook wel "primary street number" genoemd. Dit nummert niet de straat, maar juist het gebouw. Een alfanumerieke toevoeging zoals bij "14a" wordt ook bij houseNumber doorgegeven, tenzij er een specifieke vereiste is deze gescheiden door te geven.
buildingNumberSuffix	Alfanumerieke huisnummertoevoegingen zoals 'a' in '14a'. Het houseNumber zonder deze huisnummertoevoeging staat dan in houseNumber
streetName	Straatnaam. Als het adres een postbusnummer is dan dient hier het woord "Postbus" te staan. Als het adres een antwoordnummer is, dan dient hier het woord "Antwoordnummer" te staan.
additionalLocator	Aanvullende locatie-aanduiding bij het postadres. Dit kan bijvoorbeeld een nummer van een appartement, suite of verdieping zijn. In de Nederlandse situatie wordt dit vaak gebruikt voor de verdieping, bijvoorbeeld 'III' als het gaat om een woning op 3 hoog. Dit kan ook een aanduiding zijn die de relatie met een ander adres aangeeft zoals 'to' (tegenover) of 'by' (bij). Als een woonark bijvoorbeeld tegenover nummer 14 ligt, dan wordt '14' in houseNumber gezet en 'to' (tegenover) in additionalLocator.

@use *gebruikstype(n)*
SET<CS> [0..*]

De codes voor adrestype worden gedefinieerd door het HL7 domein PostalAddressUse, aangegeven in het "use" attribuut van het <addr> moeder element (zie voorbeelden). Als er geen adrestype bekend is, kan dit attribuut worden weggelaten. Als een adres van meerdere typen is, dan kan het attribuut @use meerdere adrestypen bevatten. Voorbeelden zijn een "tijdelijk bezoekadres" (use="PHYS TMP"), of een vertrouwelijk bezoek- en postadres (use="CONF PHYS PST"). Merk op dat de volgorde van de adrestypen er niet toe doet.

Tabel: Domain PostalAddressUse attribuutwaarden		
Code	Naam	Definitie
PHYS	visit address (woon- / verblijfsadres)	Fysiek adres Een adres dat bezocht kan worden (dus bijvoorbeeld geen postbus). Kan in Nederland gebruikt worden om een adres door te geven dat afwijkt van het officiële adres
PST	postal address (postadres, postbus)	Gebruikt om post naar te sturen. Voorbeeld: postbusnummer, antwoordnummer, (voor personen) postadres dat anders is dan verblijfsadres.

Tabel: Domain PostalAddressUse attribuutwaarden		
Code	Naam	Definitie
HP	primary home (officiële adres)	Het primaire thuisadres. Het adres in de GBA wordt beschouwd als het primaire thuisadres (HP).
HV	vacation home (vakantie huis)	Een vakantiehuis, om een persoon te bereiken tijdens de vakantie
WP	work place (werk)	Een adres op het werk. Eerste keus voor werkgerelateerde contacten. Voor organisaties en zorgverleners het primaire adres.
TMP	Temporary (tijdelijk)	Een tijdelijk adres. <i>TMP</i> wordt vrijwel altijd gecombineerd met een andere use code. Voorbeeld: in combinatie met <i>PHYS</i> kan een tijdelijk bezoekadres worden opgegeven, om materiaal af te nemen zodat hierop uitslagen kunnen worden gegenereerd.
CONF	Confidential (vertrouwelijk)	Een vertrouwelijk adres; het verstrekken van de adresgegevens aan derden is onderhevig aan restricties. <i>CONF</i> wordt vrijwel altijd gecombineerd met een andere use code. Voorbeeld: adressen die vertrouwelijk moeten worden behandeld omdat de persoon in kwestie in een opvangsituatie zit of een blijf-van-mijn-lijf-huis.

Voor adresgegevens van patiënten zijn de attribuutwaarden HP, WP, PST, PHYS, CONF toegestaan, voor organisaties WP, PHYS, PST, voor artsen WP.

<useablePeriod> *tijdsinterval (IVL_TS)*

Het is mogelijk om de geldigheidsduur van een adres in te perken. Het element *useablePeriod* wordt in dat geval gebruikt om deze geldigheid met een tijdsinterval aan te duiden.

XML-voorbeelden

1) Straatnaam

```
<streetName>Dorpstraat</streetName>
```



In Nederland wordt tussen de cijfers en letters van een postcode altijd een spatie ingevoegd. Dit wijkt af van de NEN Norm 5825 (waar geen spatie is opgenomen), omdat het uitgangspunt van het datatype AD is dat alles zo wordt doorgegeven als het moet worden weergegeven.

2) Postcode

```
<addr>
  <postalCode>5099 AK</postalCode>
</addr>
```

3) Postcode (als string en als code)

```
<addr>
  <postalCode code="1200BR"
               codeSystem="2.16.840.1.113883.2.4.4.15">1200 BR</postalCode>
</addr>
```

4) Volledig adres (HP: primair adres)

```
<addr use="HP">  
  <streetName>Purmersteenweg</streetName>  
  <houseNumber>42</houseNumber>  
  <postalCode>1441 DM</postalCode>  
  <city>Purmerend</city>  
</addr>
```

5) Alleen land

```
<addr>  
  <country>Nederland</country>  
</addr>
```

6) Gemeente en land (als string en als code)

```
<addr>  
  <county code="0518" codeSystem="2.16.840.1.113883.2.4.6.14"  
    codeSystemName="GBA-33">Den Haag</county>  
  <country code="NL" codeSystem="1.0.3166.1.2.2">Nederland</country>  
</addr>
```


6.21 EN (Entity Name - naam voor een entiteit)

Attribuut:

@usegebruiksaanduiding (SET<CS>)

Elementen:

<validTime>geldigheidsinterval (IVL<TS>)

<delimiter>scheidingsteken (ENXP)

<family>familienaam (ENXP)

<given>voornaam/initialen (ENXP)

<prefix>voorvoegsel/prefix (ENXP)

<suffix>achtervoegsel (ENXP)

Entity Name is een naam voor een person, organisatie, plaats of ding. Het bestaat optioneel uit een aantal naamdelen zoals gegeven naam, achternaam, prefix/voorvoegsel, suffix/achtervoegsel en/of scheidingstekens. Voorbeelden van entiteitnamen zijn "Jan Jansen", "Stichting HL7 Nederland", "Grevelingenmeer", etc. Een entiteitnaam kan zo eenvoudig zijn als een string of bestaan uit meerdere entiteitnaamdelen zoals "Jan", "van", "Dam" en "jr.", "Stichting" en "HL7 Nederland".

Als structuur is een entiteitnaam een aantal entiteitnaamdelen met een optioneel XML-attribuut @use (gebruikstype) en optioneel een periode van geldigheid die aangeeft of en wanneer een bepaalde naam voor een bepaald gebruiksdoel geschikt is. Zo kan een naam na bijvoorbeeld een organisatiefusie, een huwelijk of een adoptie worden gewijzigd met behoud van historie worden gecommuniceerd.

Het datatype EN kent verschillende specialisaties waarvan we in Nederland de volgende gebruiken:

- [PN](#) (persoonsnamen);
- [ON](#) (organisatienamen).

Zie de datatypen PN voor toelichting op persoonsnamen en zie ON voor toelichting voor gebruik op organisatienamen.

6.22 PN (Person Name - persoonsnaam)

Attributen:

@usegebruikstype(n) (set<cs>)

Elementen:

<validTime>geldigheidsinterval (IVL<TS>)

<delimiter>scheidingsteken(s) (ENXP)

<family>familienaam (ENXP)

<given>voornaam/initialen (ENXP)

<prefix>voorvoegsel/prefix (ENXP)

<suffix>achtervoegsel (ENXP)

Structuur: Het datatype PN is een extensie op het datatype EN (*Entity Name*) en bestaat dus uit een zogenaamde 'mixed content' inhoud, waar in principe vrije tekst kan worden gecombineerd met name parts. Voor Nederland is afgesproken om bij persoonsnamen beperkingen te stellen aan het gebruik van 'mixed content'. Toegestaan zijn:

- De volledige persoonsnaam is een vrije tekst (dus er zijn geen *person name parts*), uitsluitend te gebruiken in situaties waarbij het voor de verzender niet mogelijk is om onderdelen te benoemen.
- Alle onderdelen zijn als *person name part* benoemd. Er mag in dat geval dus geen tekst voorkomen die niet omgeven is door één van de hieronder beschreven tags.



In Nederland geldt de richtlijn dat persoonsnamen **zoveel als mogelijk** moeten worden opgedeeld in de juiste naamdelen en worden getypeerd met de juiste qualifier. Dit is een verantwoordelijkheid van de verzender. Een ontvanger die niet alle onderdelen expliciet kan opslaan, heeft altijd de mogelijkheid om onderdelen samen te voegen, maar andersom kan niet.



De volgorde van de *person name parts* is relevant! De richtlijn is dat deze moeten worden doorgegeven in de 'natuurlijke' volgorde bij het gebruik van de naam (zodat de ontvanger ze achter elkaar kan weergeven). De aangegeven volgorde is met name belangrijk in de volgende gevallen:

- Voorvoegsels (prefix) moeten altijd **vóór** de naam staan waar ze bijhoren.
- Achtervoegsels (suffix) moeten altijd **ná** de naam staan waar ze bijhoren.
- Voornamen (given) moeten altijd in de officiële volgorde staan.
- Achternamen (family) en een eventuele bijbehorende delimiter ('-') moeten in de volgorde staan waarin ze officieel gevoerd worden.

Dit wordt nader toegelicht bij de verschillende *person name parts*.



Het is **niet** toegestaan om **lege naamdelen** mee te geven. Om aan te geven dat een naamdeel leeg is dient het simpelweg niet te worden meegegeven in het bericht.

XML-voorbeelden

Persoonsnaam als vrije tekst

```
<name>Jan Jansen</name>
```

De naam wordt zonder interne structuur doorgegeven

Persoonsnaam met name parts.

```
<name>
  <given>Jan</given>
  <family>Jansen</family>
</name>
```

De beide onderdelen van de naam worden benoemd en voorzien van een qualifier: "Jan" is een 'naam die bij de geboorte is gegeven', oftewel een voornaam (voluit), en "Jansen" is een 'familienaam zoals bekend bij de GBA', oftewel de eigen achternaam.

Ongeldige persoonsnaam.

```
<name>
  Jan <family>Jansen</family>
</name>
```

Dit is geen geldige persoonsnaam, omdat vrije tekst en *name part* gecombineerd zijn.

@use naamgebruikstype(n)

In principe kan van elke *Person Name* worden aangegeven in welke situatie deze gebruikt kan worden. Voor Nederland is besloten dat de volgende naamgebruikstypen voor kunnen komen:

Tabel: Domain <i>EntityNameUse</i> attribuutwaarden		
Code	Naam	Definitie
L	Reguliere naam	De naam zoals die door de persoon (entiteit) gevoerd wordt. De afkorting 'L' stond oorspronkelijk voor Legal (wettelijk), maar feit is dat hier ook componenten in voor mogen komen (zoals een roepnaam), die niet wettelijk zijn vastgelegd. Dit is het standaardtype als geen type wordt doorgegeven.
A	Pseudoniem	Een artiestennaam, 'schuilnaam' of tijdelijke naam voor een persoon (entiteit). Deze wijkt dus af van de regulier gevoerde naam en wordt bijvoorbeeld gebruikt om iemands identiteit te verbergen (in verband met privacy) of als tijdelijke naam wanneer de echte niet bekend is ('John Doe').

OR	Wettelijk geregistreerde naam	De naam met de exacte componenten zoals deze voorkomen in het bevolkingsregister van het betreffende land. Voor Nederland is dit het GBA register of ARNI voor niet-ingezetenen. Dit is de naam zoals die wordt geretourneerd indien een BSN met succes wordt geverifieerd.
----	----------------------------------	---



Merk op dat een naam ook twee waarden in het attribuut *@use* mag hebben. Dit kan bijvoorbeeld voorkomen als de wettelijke geregistreerde naam ('OR') ook als de reguliere naam gevoerd wordt (zie het voorbeeld hieronder). Let echter op dat dan geen enkele component mag voorkomen (zoals roepnaam of een partnernaam) die geen onderdeel is van de wettelijk geregistreerde naam.

XML-voorbeelden

De reguliere naam als default, dus geen use attribuut

```
<name>
  <!-- Naam, onderverdeeld in componenten -->
</name>
```

Een pseudoniem voor een patiënt:

```
<name use="A ">
  <!-- Naam, onderverdeeld in componenten -->
</name>
```

De wettelijk geregistreerde naam, zoals geretourneerd door de SBV-Z:

```
<name use="OR">
  <!-- Naam, onderverdeeld in componenten -->
</name>
```

De gevoerde naam is exact gelijk aan de wettelijk geregistreerde naam:

```
<name use="OR L">
  <!-- Naam, onderverdeeld in componenten -->
</name>
```

<validTime> *geldigheidsinterval*

Dit is een **optioneel** XML-element binnen de *Person Name* en duidt de periode aan waarin deze naam 'in gebruik'/geldig was voor de betreffende persoon. De opties zijn:

- Er is geen *validTime* element: de betreffende naam is in principe onbeperkt geldig.
- Er is een onder- en een bovengrens: de naam was geldig in de aangeduide periode.
- Er is alleen een ondergrens: de naam is geldig sinds de aangeduide datum.
- Er is alleen een bovengrens: de naam was geldig t/m de aangeduide datum.

Dit element van *Person Name* kan worden gebruikt om aan te geven dat een persoon gedurende diens leven één of meer keer van naam veranderd is. Dit gebeurt o.a. bij:

- Adoptie van een baby, waarbij het de achternaam van de adoptie-ouders verkrijgt.

- Huwelijk, waarbij de partnernaam kan worden toegevoegd aan de eigen naam.
- Scheiding, waarbij een eerder aangenomen partnernaam juist weer vervalt.
- Personen die om andere redenen hun voor- of achternaam veranderen.



In elke situatie waar één of meer *Person Names* worden doorgegeven, moet minimaal de naam worden aangeduid die op het moment van verzenden geldig/actueel is. Vervallen namen kunnen dus alleen worden doorgegeven als het betreffende berichtelement herhalend is (dus met cardinaliteit > 1).

- Merk op dat veel patiëntregistratiesystemen niet echt een historie (met ingangsdatum) bijhouden van de patiëntnaam. Wel wordt vaak een 'audit trail' (wijzigingshistorie) van de patiëntgegevens in het algemeen bijgehouden. Indien gewenst zou daaruit een historie van de persoonsnaam kunnen worden afgeleid, hoewel het natuurlijk ook mogelijk is om alleen de actuele naam door te geven (en dus geen *validTime* te gebruiken).



In tegenstelling tot de situatie bij *Organization Name* is het niet toegestaan dat de ondergrens of de bovengrens van een *validTime* aanduiding bij *Person Name* in de toekomst ligt. Er kan dus geen 'geplande' nieuwe naam of het 'gepland vervallen' van de huidige naam worden doorgegeven voor persoonsnamen.

XML-voorbeelden

De actuele naam is geldig sinds 12 juli 2005

```
<name>
  <validTime>
    <low value="20050712"/>
  </validTime>
  <!-- Naam, onderverdeeld in componenten -->
</name>
```

Bovenstaande situatie kan zich bijvoorbeeld voordoen bij een systeem dat alleen de actuele naam doorgeeft, maar wel de historie bijhoudt. Bovenstaande persoon kan bijvoorbeeld op 12 juli getrouwd zijn en daarbij de achternaam van haar partner hebben aangenomen.

Voorbeeld: oude namen plus actuele naam. Merk op dat de naam in deze voorbeelden niet op een gestructureerde wijze getoond wordt. De focus van dit voorbeeld ligt op het gebruik van de geldigheidsduur.

```

<name>
  <validTime>
    <high value="19850412"/>
  </validTime>
  <!-- "Nicole de Vries" als naam van de baby voor adoptie -->
</name>
<name>
  <validTime>
    <low value="19850412"/>
    <high value="20050824"/>
  </validTime>
  <!-- "Nicolette Scheick" als naam na adoptie, maar voor huwelijk -->
</name>
<name>
  <validTime>
    <low value="20050824"/>
  </validTime>
  <!-- "Nicolette Scheick-Jansen" als naam na huwelijk -->
</name>

```

In bovenstaand voorbeeld wordt de baby Nicole de Vries geadopteerd door de familie Scheick, waarbij ze dus van achternaam verandert. Omdat de adoptie-ouders dit een leukere naam vinden, wordt haar voornaam (of in ieder geval haar roepnaam) ook veranderd in Nicolette. Na haar huwelijk neemt ze de achternaam van haar partner (Jansen) aan. Het verzendende systeem stuurt in dit geval de hele naamhistorie mee.

<delimiter> scheidingsteken (ENXP)

Een *delimiter* heeft geen speciale betekenis als onderdeel van een *Person Name*, anders dan het doorgeven van een (stukje) letterlijke tekst dat in de geschreven naam voorkomt.

Een *delimiter* moet altijd op de plaats in de *Person Name* staan waar de tekst ook geschreven zou worden. Er zijn geen impliciete spaties, dus als er normaal gesproken een spatie voor of achter geschreven wordt, dan moet deze expliciet worden meegegeven.

Voorbeelden van *delimiters* in *Person Names* zijn:

- Het streepje ` ` tussen de eigen achternaam en de partnernaam (of andersom).
- De komma plus spatie `, ` die tussen de naam en bepaalde achtervoegsels komt.
- De tekst ` , geb. ' of ` , e.v. ` die soms gebruikt wordt bij eigen- respectievelijk partnernaam
- Merk op: de spatie tussen een voorvoegsel en de achternaam is *geen* delimiter, het behoort bij het voorvoegsel. Dit geldt tevens voor andere scheidingstekens behorende bij een voorvoegsel zoals in de naam "D'Artagnan".

Deze zouden als volgt gebruikt worden binnen een *Person Name* XML-berichtelement.

XML-voorbeelden

Scheider tussen partnernaam en geslachtsnaam

```

<name>
  <family qualifier="SP">Jansen</family>
  <delimiter>-</delimiter>
  <family qualifier="BR">Scheick</family>
</name>

```

Scheider tussen achternaam en academische titel

```
<name>
  <family qualifier="SP">Jansen</family>
  <delimiter>, </delimiter>
  <title qualifier="AC">RA</title>      <!-- registeraccountant -->
</name>
```

Zie verder de algemene voorbeelden achterin deze sectie van het document.

Een aantal voor de hand liggende vragen (en antwoorden) over *delimiters*:



Q Wordt een ontvangend systeem geacht een *delimiter* op te slaan?

A Een ontvangend systeem dat de afzonderlijke *person name parts* verwerkt, zal vrijwel nooit de *delimiters* opslaan in de eigen database.

Q Waarom zou een verzendend systeem dan ooit *delimiters* meesturen?

A Er kunnen ontvangende systemen zijn die géén of niet alle *person name parts* apart kunnen verwerken (bijvoorbeeld omdat ze maar één veld voor de achternaam hebben). Deze kunnen d.m.v. de *delimiters* een volledig uitgeschreven naam overnemen. (Een voorbeeld van zo'n eenvoudige ontvanger is een web interface die binnenkomende HL7v3 berichten omzet naar HTML formaat voor directe weergave op het beeldscherm).

<family> *familienaam (ENXP)*

Attribuut:

@qualifier *naamsoort (set<cs>)*

Een *person name part* van het type *family* heeft betrekking op een deel van de naam dat door familiebanden is verkregen, meestal **achternaam** genoemd. Normaal gesproken heeft dit dus betrekking op de eigen achternaam (verkregen via de ouders) en eventueel op een aangenomen achternaam na een huwelijk ('overgenomen' van de partner).

Enkele regels voor *person name parts* van type *family*:

- Ze moeten onderling altijd conform de officiële schrijfwijze geordend zijn (bijvoorbeeld eerst eigen achternaam en dan achternaam van de partner of juist andersom).
- Er is altijd sprake van een impliciete spatie als tussenruimte met het erop volgende *name part*, behalve als dit een *delimiter* of een *suffix* is (zie aldaar).
- De aard van de familienaam kan verder worden aangeduid door het optionele attribuut *@qualifier* te gebruiken. Zie de tabel voor daarbij toegestane waarden.
- Er mag maar één familienaam per type qualifier voorkomen in de naam!

<i>qualifiers</i> bij person name part <i>family</i>	
qualifier	toepassing
BR	(birth) Geboortenaam. Een achternaam, zoals die is geregistreerd in de GBA (daarin aangeduid als 'geslachtsnaam'). Deze is meestal bij geboorte verkregen, maar kan ook op een later moment in het leven zijn aangenomen. Hierbij is uitgesloten de naam die wordt overgenomen van de partner in een huwelijk (deze krijgt qualifier "SP"). Normaal gesproken is dit de geslachtsnaam van één de (natuurlijke) ouders, maar in sommige culturen kan dit ook een combinatie van de geboor-

qualifiers bij person name part family	
qualifier	toepassing
	tenamen van beide ouders of een afgeleide van de voornaam van één van de ouders zijn.
SP	(spouse) Partnernaam. Een achternaam die is 'overgenomen' van een partner in een huwelijksrelatie (meestal de geslachtsnaam van de partner). Er kan geen aanname worden gedaan over het geslacht van de drager, aangezien het overnemen van namen in Nederland volledig gelijkgesteld is tussen de beide partners. Ook kan geen conclusie worden getrokken over de huwelijksstaat als een partnernaam ontbreekt, aangezien iemand gehuwd kan zijn zonder de naam van de partner te voeren. Let op dat het bij het gebruik van een partnernaam niet alleen relevant is dat iemand getrouwd is, maar vooral of de betreffende persoon al dan niet met de naam van de partner wil worden aangesproken. Het toepassen van de partnernaam in de persoonsnaam staat los van het eventueel vastleggen van de partner als contactpersoon. In de huidige Nederlandse wetgeving kan na een huwelijk elke combinatie van geslachtsnaam en/of partnernaam gevoerd worden door beide partners. Vanzelfsprekend is het geslacht van de beide partners daarbij niet relevant.



Indien een *person name part* van type *family* **zonder een qualifier** wordt gebruikt, dan wordt het simpelweg geïnterpreteerd als **achternaam**. Als de ontvanger moet kiezen tussen opslag als een geslachtsnaam of als een partnernaam, dan moet in zo'n geval voor de **geslachtsnaam** gekozen worden.



Er moet nog worden bepaald hoe moet worden omgegaan met een geslachtsnaam die iemand niet meer voert na een huwelijk (omdat alleen de partnernaam wordt gevoerd). De geslachtsnaam moet dan toch opgeslagen (en doorgegeven) worden, maar er zou daarbij kunnen worden aangegeven dat de naam '**onzichtbaar**' moet zijn. Eén oplossing zou bestaan uit het implementeren van een attribuut *invisible*. Wat sowieso kan is om de naam twee keer door te geven: één met *use= 'OR'* (mét de geslachtsnaam) en één met '*L*' (zonder).



Er moet in internationaal verband nog worden uitgezocht hoe moet worden omgegaan met een achternaam die is overgenomen van de **adoptieouders**. Volgens de huidige definitie is de *qualifier* "BR" hier niet voor bedoeld, maar het is duidelijk dat de *@qualifier*-waarden "SP" en "CL" ook niet toereikend zijn. Vooralsnog is het advies om in dit geval (als überhaupt bekend is dat het om de adoptienaam gaat) geen qualifier mee te sturen. In de praktijk zal een dergelijke naam echter meestal als 'eigen achternaam' (en dus met *@qualifier* "BR") worden gebruikt.

XML-voorbeelden

Jan Jansen

```
<name>
  <given>Jan</given>
  <family qualifier="BR">Jansen</family>
</name>
```

Nicolette Jansen-Scheick

```
<name>
  <given>Nicolette</given>
  <family qualifier="SP">Jansen</family>
```



```
<delimiter>-</delimiter>
<family qualifier="BR">Scheick</family>
</name>
```

<given>..... *voornaam/initialen (ENXP)*

Attribuut:

@qualifier..... *naamsoort (set<cs>)*

Een *person name part* van het type *given* heeft betrekking op een deel van de naam dat, meestal door de ouders en meestal bij de geboorte, wordt gegeven aan een persoon. In Nederland wordt het meestal **voornaam** genoemd, hoewel het niet in alle culturen vóór de familienaam wordt geschreven. Ook naamdelen die zijn afgeleid van de voornaam, namelijk de initialen (voorletters) en de roepnaam (informele voornaam) hebben het type *given*.

Enkele regels voor *person name parts* van type *given*:

- Een persoon kan zondermeer meerdere voornamen of initialen hebben.
- Officiële voornamen en initialen moeten altijd in de juiste volgorde staan.
- Er is altijd sprake van een impliciete spatie als tussenruimte met het erop volgende *name part*, behalve als dit een *delimiter* of een *suffix* is (zie aldaar).
- De aard van de gegeven naam kan verder worden aangeduid door het optionele attribuut *qualifier* te gebruiken. Zie de tabel voor daarbij toegestane waarden.

<i>qualifiers</i> bij <i>person name part given</i>	
Qualifier	toepassing
BR	Officiële voornaam. Een voornaam die meestal bij de geboorte is gegeven (meestal door de ouders) en die officieel is vastgelegd. Deze dienen voluit en in de juiste volgorde te worden gebruikt. Als een persoon meerdere voornamen heeft, dan moet in ieder geval de eerste naam worden doorgegeven (alleen de eerste voornaam is dus toegestaan, maar alleen de tweede niet). Als er meerdere voornamen zijn, dan kunnen deze als aparte <i>given</i> elementen worden doorgegeven maar dit kan ook in één element (gescheiden door spaties). Ook komt het voor dat de eerste voornaam voluit wordt gebruikt in combinatie met de initialen.
CL	Roepnaam. Een voornaam waarmee de persoon informeel wordt aangesproken, meestal afgeleid van één van de officiële voornamen. In tegenstelling tot de officiële voornamen kan een roepnaam eenvoudig variëren tijdens iemands leven en kan een persoon zelfs meerdere roepnamen tegelijkertijd hanteren, afhankelijk van hoe mensen hem of haar kennen. In dat geval dient de meest toepasselijke roepnaam te worden verzonden.
IN	Initialen. Meestal een afkorting van een voornaam. Dit kan een enkele letter zijn of meer dan één (bijvoorbeeld 'Th.' voor Theo). Een afsluitende punt moet expliciet worden vermeld. Initialen dienen in de juiste volgorde te worden gebruikt. Als er meerdere initialen zijn, dan kunnen deze als aparte <i>given</i> elementen worden doorgegeven maar dit kan ook in één element gebeuren (gescheiden door punten). Het voordeel van deze methode is dat er geen spaties tussen de afzonderlijke initialen in de naam worden geïmpliceerd.



Indien een *person name part* van type *given* **zonder een qualifier** wordt gebruikt, dan wordt het simpelweg geïnterpreteerd als **voornaam**. Als de ontvanger moet kiezen tussen opslag als een officiële voornaam of als een roepnaam, dan moet in zo'n geval voor de **officiële voornaam** gekozen worden.



De *qualifiers* "BR" en "CL" kunnen ook gezamenlijk voorkomen, om aan te geven dat een officiële voornaam ook als roepnaam fungeert. Op deze manier kan voorkomen worden dat dezelfde naam 2 keer moet worden meegegeven. Als de roepnaam echter verschilt van de officiële voornamen, dan kunnen ze beide worden doorgegeven: eerst de officiële voornamen, vervolgens de roepnaam.



Als een combinatie van officiële voornaam en initialen wordt meegestuurd, dan is voor Nederland de afspraak dat deze niet mogen 'overlappen', d.w.z. dat de eerste voorletter (indien nodig) door het ontvangende systeem moet worden afgeleid uit de eerste letter(s) van de officiële voornaam (zie voorbeeld hierna).



Er moet in Nederland nog worden uitgezocht hoe moet worden omgegaan met een voornaam die is gegeven door **adoptieouders**. Volgens de huidige definitie is de *qualifier* "BR" hier niet voor bedoeld, maar het is niet duidelijk of de *qualifier* "CL" (roepnaam) wel toereikend is. De vraag is namelijk of een dergelijke naam al dan niet een officieel karakter krijgt of alleen als roepnaam fungeert.

XML-voorbeelden

Hans Jansen, zonder qualifier

```
<name>
  <given>Hans</given>
  <family qualifier="BR">Jansen</family>
</name>
```

Johannes Theodorus Cornelis Jansen, officiële voornamen

```
<name>
  <given qualifier="BR">Johannes Theodorus Cornelis</given>
  <family qualifier="BR">Jansen</family>
</name>
```

Johannes Theodorus Cornelis Jansen, met roepnaam Hans

```
<name>
  <given qualifier="BR">Johannes Theodorus Cornelis</given>
  <given qualifier="CL">Hans</given>
  <family qualifier="BR">Jansen</family>
</name>
```

Johannes Th.C. Jansen, zonder duplicering van voorletter 'J'

```
<name>
  <given qualifier="BR">Johannes</given>
  <given qualifier="IN">Th.C.</given>
  <family qualifier="BR">Jansen</family>
</name>
```

Kai Uwe Heitmann, Kai is zowel officiële naam als roepnaam

```
<name>
  <given qualifier="BR CL">Kai</given>
  <given qualifier="BR">Uwe</given>
  <family qualifier="BR">Heitmann</family>
</name>
```

Conclusie uit de voorbeelden is dat allerlei combinaties van officiële voornamen, roepnamen en initialen kunnen voorkomen, afhankelijk van de opslag- en communicatiemogelijkheden van het verzendende systeem. De betekenis van elk onderdeel moet echter duidelijk zijn.

<prefix>voorvoegsel (ENXP)

Attributen:

@qualifier..... naamsoort (set<cs>)

Een *person name part* van het type *prefix* heeft betrekking op een deel van de naam dat **hoort bij één of meer andere naamdelen en daar vóór wordt geschreven**. In principe zijn er twee soorten voorvoegsels, namelijk **voorvoegsels bij de achternaam en titels** (die als toevoeging in de schrijfname worden opgenomen). Het letterlijke woord 'voorvoegsel' is zelfs in de internationale HL7v3 materialen geslopen, als internationale term voor voorvoegsels bij de achternaam (al was family name prefix logischer geweest). Enkele regels voor *person name parts* van type *prefix*:

- Een *prefix* moet altijd direct vóór de naamdelen worden geplaatst waar het betrekking op heeft (d.w.z. waar het normaal gesproken wordt geschreven).
- Er is **geen impliciete spatie** als tussenruimte met het erop volgende *name part*, d.w.z. een spatie na het voorvoegsel moet expliciet worden vermeld in de tekst!
- De aard van het voorvoegsel kan verder worden aangeduid door het optionele attribuut *qualifier* te gebruiken. Zie de tabel voor daarbij toegestane waarden.

In de eerstvolgende release van de datatypen van HL7v3 wordt het mogelijk om diverse elementen die nu een 'string' zijn, optioneel ook als code aan te duiden.



Dit kan o.a. worden toegepast om titels gecodeerd te versturen, indien het verzendende systeem ze als code opslaat. Daarnaast moet echter ook altijd de tekst worden gezonden, zodat de ontvanger kan bepalen welke gebruikt wordt.

Op het moment dat deze methode beschikbaar is, moet worden bepaald welke titelcodes binnen Nederland gehanteerd zullen worden (NEN/NUFFIC/NHG?).

qualifiers bij person name part prefix	
qualifier	Toepassing
VV	Voorvoegsel bij de achternaam. Dit gaat om de in Nederland veel voorkomende naamdelen als "van ", "de ", "ten " of "el ", maar ook combinaties als "van der " of "in 't ". Een voorvoegsel hoort altijd bij het <i>person name part</i> van type <i>family</i> dat er direct achter staat (zie de XML-voorbeelden). Een voorvoegsel kan als onderdeel van de achternaam worden opgenomen, maar

qualifiers bij person name part <i>prefix</i>	
qualifier	Toepassing
	het is gebruikelijk om het apart op te slaan (en te verzenden) indien sorteren (en dus ook terugzoeken) plaatsvindt op basis van de achternaam zonder het voorvoegsel. Dit is in Nederland absoluut gebruikelijk, maar in België bijvoorbeeld niet. Merk op dat er meerdere voorvoegsels kunnen voorkomen in één naam, namelijk zowel bij de geslachtsnaam als bij een eventuele partnaam.
AC	Academische titel. Een titel (meestal in afgekorte vorm) die is verkregen door het voltooien van een studie of door het vervullen van een rol binnen een academische instelling. Als voorvoegsel kunnen bijvoorbeeld fungeren: "drs. ", "ing. ", "ir. ", "mr. ", "dr. ", "prof. ", maar ook een combinatie als "prof. dr. ".
NB	Adellijke titel. Een titel (meestal voluit geschreven) die is ontleend aan iemands aristocratische status. Voorbeelden zijn "Jonkheer ", "Graaf ", etc..
TITLE	Algemene aanhef. Dit wordt in principe gebruikt voor de aanhef bij een volledige naam (zoals gebruikt in correspondentie), bijvoorbeeld "De Wedelgeleerde Heer", "De Weledelhooggeboren Vrouwe ", maar ook simpelweg "Mevrouw ". De aard van zo'n aanhef hangt dus samen met het geslacht van de persoon en met eventuele academische en/of adellijke titels (o.b.v. afleidregels).



Indien een *person name part* van type *prefix* **zonder een *qualifier*** wordt gebruikt, dan wordt het altijd geïnterpreteerd als **titel**. Dit omdat veel verzendende systemen wel titels ondersteunen, maar geen onderscheid maken tussen academische en adellijke titels. Voorvoegsels bij de achternaam en algemene aanhef ("VV" respectievelijk "TITLE") moeten dus altijd expliciet worden doorgegeven.



Indien van een titel niet bekend is of deze voor of achter de naam geplaatst moet worden, dan moet deze worden verzonden als *prefix*. Dit zal veel voorkomen bij systemen die geen plaats in de tekst bij een titel opslaan. In dat geval zullen titels dus altijd vóór de naam binnen het datatype *Person Name* staan.



Er is geen regel voor het aantal voorvoegsels dat wordt gecombineerd in één element. D.w.z. dat "van " en "der " of "Mr. " en "Drs. " apart kunnen worden doorgegeven, maar ook gecombineerd als "van der " en "Mr. Drs. ".

XML-voorbeelden

Monique van Wijk

```
<name>
  <given>Monique</given>
  <prefix qualifier="VV">van </prefix>
  <family qualifier="BR">Wijk</family>
</name>
```

Monique van Wijk-de Boer (voert ook de naam van haar partner)

```
<name>
  <given>Monique</given>
  <prefix qualifier="VV">van </prefix>
```

```

<family qualifier="BR">Wijk</family>
<delimiter>-</delimiter>
<prefix qualifier="VV">de </prefix>
<family qualifier="SP">Boer</family>
</name>

```

Nu komen er twee <prefix> elementen voor, vóór beide delen van de achternaam.

Dr. Kai Heitmann

```

<name>
  <prefix qualifier="AC">Dr. </prefix>
  <given>Kai</given>
  <family qualifier="BR">Heitmann</family>
</name>

```

In dit geval is de regel dus dat de titel ('Dr. ') voor de gehele naam wordt geplaatst, omdat dit de normale plaats is waar het in de geschreven vorm terecht zou komen.

Berend-Jan baron van Voorst tot Voorst

```

<name>
  <given>Berend-Jan</given>
  <prefix qualifier="NB">baron </prefix>
  <prefix qualifier="VV">van </prefix>
  <family qualifier="BR">Vorst tot Voorst</family>
</name>

```

In dit (reële) voorbeeld staat de titel ('baron ') tussen de voornaam en de achternaam in, omdat dit de gebruikelijke schrijfwijze is. Daarnaast is er een 'gewoon' voorvoegsel bij de achternaam ('van '). Overigens zal veel software niet in staat zijn om een (apart opgeslagen) titel op de juiste plaats te zetten, waardoor 'baron' ook vooraan kan voorkomen.

prof. jhr. mr. A. de Jonge

```

<name use="L">
  <prefix qualifier="AC">prof. </prefix>
  <prefix qualifier="NB">jhr. </prefix>
  <prefix qualifier="AC">mr. </prefix>
  <given qualifier="IN">A.</given>
  <prefix qualifier="VV">de </prefix>
  <family>Jonge</family>
</name>

```

Weledelzeergeleerde Heer Dr. William Goossen

```

<name>
  <prefix qualifier="TITLE">Weledelzeergeleerde Heer </prefix>
  <prefix qualifier="AC">dr. </prefix>
  <given>William</given>
  <family qualifier="BR">Goossen</family>
</name>

```

Dit is een voorbeeld waarbij het verzendende systeem de volledige titel blijkbaar heeft vastgelegd (of afleidt uit de titel) en doorgeeft als deel van de 'correspondentienaam'. Maar ook zonder titel kan het verzendende systeem een algemene aanhef meesturen (zie hieronder).

Mevr. A. Jansen

```
<name>
  <prefix qualifier="TITLE">Mevr. </prefix>
  <given qualifier="IN">A.</given>
  <family qualifier="BR">Jansen</family>
</name>
```

<suffix>..... achtervoegsel (ENXP)

Attributen:

@qualifier..... naamsoort (CS)

Een *person name part* van het type *suffix* heeft betrekking op een deel van de naam dat **hoort bij één of meer andere naamdelen en daar achter wordt geschreven**. In Nederland zijn als achtervoegsel alleen **academische titels** toegestaan.

Enkele regels voor *person name parts* van type *suffix*:

- Een *suffix* moet altijd direct achter de naamdelen worden geplaatst waar het betrekking op heeft (d.w.z. waar het normaal gesproken wordt geschreven).
- Er is **geen impliciete spatie** als tussenruimte met het eraan voorafgaande *name part*, d.w.z. een spatie voor het achtervoegsel moet expliciet worden vermeld!
- De aard van het achtervoegsel kan verder worden aangeduid door het optionele attribuut *qualifier* te gebruiken. Zie de tabel voor daarbij toegestane waarden.



In de eerstvolgende release van de datatypen van HL7v3 wordt het mogelijk om diverse elementen die nu een 'string' zijn, optioneel ook als code aan te duiden.

Dit kan o.a. worden toegepast om titels gecodeerd te versturen, indien het verzendende systeem ze als code opslaat. Daarnaast moet echter ook altijd de tekst worden gezonden, zodat de ontvanger kan bepalen welke gebruikt wordt.

Op het moment dat deze methode beschikbaar is, moet worden bepaald welke titelcodes binnen Nederland gehanteerd zullen worden (NEN/NUFFIC/NHG?).

<i>qualifiers bij person name part suffix</i>	
qualifier	Toepassing
AC	Academische titel. Een titel (meestal in afgekorte vorm) die is verkregen door het voltooien van een studie of door het vervullen van een rol binnen een academische instelling. Bij achtervoegsels gaat het meestal om internationale gehanteerde aanduidingen als " MSc", " PhD" of " MD".



Een *person name part* van type *suffix* dat **zonder *qualifier*** wordt gebruikt, moet worden beschouwd als een **niet nader bepaald achtervoegsel**. Ook het gebruik van (vaak Amerikaanse) termen als *Jr.*, *Sr.* of *III* valt in deze categorie.



Er is geen regel voor het aantal achtervoegsels dat wordt gecombineerd in één element. D.w.z. dat " MSc" en " MD" apart kunnen worden doorgegeven, maar ook gecombineerd als " MSc MD".

XML-voorbeelden

Bert Kabbes, RI

```
<name>
  <given>Bert</given>
  <family qualifier="BR">Kabbes</family>
  <delimiter>,</delimiter>
  <suffix qualifier="AC">RI</suffix>
</name>
```

6.23 ON (Organization Name - organisatienaam)

Attributen:

@usegebruikstype(n) (SET<CS>), niet gebruiken in NL

Elementen:

<validTime>geldigheidsinterval (IVL<TS>)

Structuur: Het datatype ON is een extensie op het datatype EN (*Entity Name*) en bestaat dus uit een zogenaamde 'mixed content' inhoud, waar in principe *name parts* in kunnen zijn aangeduid. Voor Nederland is vooralsnog afgesproken om geen gebruik te maken van *organization name parts* en de naam altijd uit te drukken als **één tekst**. Dit betekent dat ON in Nederland in feite gelijk wordt aan het datatype TN (*Trivial Name*).

XML-voorbeelden

minimaal

```
<name>De Naam van de Organisatie</name>
```

Merk op dat de tekst van de naam zonder quotes wordt doorgegeven.

@usegebruikstype(n) (set<cs>)

In principe kan van elke *Organization Name* worden aangegeven in welke situatie deze gebruikt kan worden. Voor Nederland is echter besloten dat het onderscheid (nog) niet relevant is, zodat het advies is om het attribuut use niet te gebruiken in de XML-tag.

<validTime>geldigheidsinterval (IVL<TS>)

Dit is een **optioneel** XML-element binnen de *Organization Name* en duidt de periode aan waarin deze naam 'in gebruik'/geldig was voor de betreffende organisatie. De opties zijn:

- Er is geen *validTime* element: de betreffende naam is in principe universeel geldig.
- Er is een onder- en een bovengrens: de naam was geldig in de aangeduide periode.
- Er is alleen een ondergrens: de naam is geldig sinds de aangeduide datum.
- Er is alleen een bovengrens: de naam was geldig t/m de aangeduide datum.

XML-voorbeelden

1) De naam was geldig van 1-1-2003 t/m 31-12-2004. Onder- en bovengrens

```
<name>
  <validTime>
    <low value='20030101000000' />
    <high value='20041231235959' />
  </validTime>
  De Oude Naam van de Organisatie
</name>
```


2) De naam is geldig sinds 1-1-2005. Alleen ondergrens

```
<name>
  <validTime>
    <low value='20050101' />
  </validTime>
  De Nieuwe Naam van de Organisatie
</name>
```

3) De naam was geldig t/m 31-12-2005. Alleen bovengrens

```
<name>
  <validTime>
    <high value='20051231235959' />
  </validTime>
  De Oude Naam van de Organisatie
</name>
```

Zowel ondergrens als bovengrens van *validTime* kunnen ook in de toekomst liggen, om een geplande nieuwe naam of gepland vervallen van een bestaande naam aan te geven.



In elke situatie waar één of meer *Organization Names* worden doorgegeven, moet minimaal de naam worden aangeduid die op het moment van verzenden geldig/actueel is. Vervallen namen kunnen dus alleen worden doorgegeven als het betreffende berichtelement herhalend is (dus met max. cardinaliteit > 1).

4) Herhalende naam

```
<name>
  <validTime>
    <high value='20041231235959' />
  </validTime>
  De Oude Naam van de Organisatie
</name>
<name>
  <validTime>
    <low value='20050101000000' />
    <high value='20051231235959' />
  </validTime>
  De Huidige Naam van de Organisatie
</name>
<name>
  <validTime>
    <low value='20060101000000' />
  </validTime>
  De Nieuwe Naam van de Organisatie
</name>
```

6.24 QTY (Quantity - hoeveelheid)

Hoeveelheid is een abstract datatype. Het wordt gebruikt om basiseigenschappen van afgeleide types te definiëren. Deze abstractie is noodzakelijk om andere datatypen te kunnen definiëren zoals b.v. interval. Het kan dus niet als zodanig voorkomen in HL7 berichten.

6.25 INT (Integer Number - geheel getal)

Attributen:

@value..... waarde (int)

Elementen:

geen

Het XML-attriboot @value kan gehele getallen (integer) aannemen. Integers kunnen als zodanig in de berichten voorkomen. Voorbeelden zijn b.v. het aantal herhalingen van een voorschrift (repeatCount) en het sequenceNumber in een herhaalbare actRelationship. In het algemeen geldt dat integers gebruikt worden om te tellen, dan wel te ordenen.

XML-voorbeeld

```
<number value="24"/>
```

6.26 REAL (Real Number - getal met decimalen)

Attributen:

@value..... waarde (real)

Elementen:

geen

Het datatype REAL kan een getal met decimalen als waarde aannemen. Decimale weergave via een punt „.". Het type REAL komt als zodanig niet voor in berichten. Een REAL is altijd onderdeel van een ander type, meestal Physical Quantity (PQ).

6.27 PQ (Physical Quantity - kwantitatieve weergave van fysieke grootheden)

Attributen:

@value grootte van de hoeveelheid (real)

@unit..... meeteenheid (cs)

Elementen:

<translation> alternatieve representatie (PQR)

Elementen van dit datatype zijn fysieke hoeveelheden, d.w.z. een hoeveelheid die een meetbare (of telbare) waarde uit de fysieke wereld weergeeft. Dit is dus iets anders dan alleen een 'aantal', aangezien ook sprake is van een (natuurlijke) eenheid waarin de bepaalde waarde wordt uitgedrukt.

Voorbeelden:

- Een hoeveelheid afgenomen bloed van 20 ml (monstergrootte)
- Een snijwond met een lengte van 10 cm (resultaat observatie)
- Een dosis van 200 milligram per toediening (doseerhoeveelheid)
- Aflevering van 60 stuks van een tablet (verstreckte hoeveelheid)

Elementen van dit datatype hebben de volgende structuur.

Het *value* attribuut geeft de hoeveelheid aan, uitgedrukt in de eenheid *unit*.

@value grootte van de hoeveelheid (real)

In *unit* staat de meeteenheid, uitgedrukt conform de *Unified Code for Units of Measure (UCUM)*. Deze tabel bevat alle natuurlijke meeteenheden.

@unit..... meeteenheid (cs)

Als het attribuut *@value* "meetbaar" is (bijvoorbeeld uitgedrukt in milligram of liter), dan moet de eenheid in *@unit* worden opgenomen. Als *@value* 'telbaar' is (bijvoorbeeld tabletten) dan mag *@unit* niet worden gebruikt.

Voor een volledige beschrijving van de mogelijke eenheden, zie de *Unified Code for Units of Measure (UCUM)* specificatie. UCUM is een verzameling van atomaire eenheden en prefixes en een bijhorende grammatica om geldige combinaties daarvan op te bouwen.

Als er een *nullFlavor* attribuut aanwezig is, mogen *value* en *unit* niet aanwezig zijn; als er geen *nullFlavor* attribuut is, dan moet een *value* aanwezig zijn.

Een alternatieve representatie van dezelfde fysieke hoeveelheid, uitgedrukt in een andere eenheid, mogelijk uit een ander systeem dan UCUM en mogelijk met een andere bijbehorende waarde (*value*) kan in de *translation* worden weergegeven.

<translation> alternatieve representatie (PQR)

Omdat de meetwaarden het meest in samenhang met een algemene observatie worden weergegeven (Observation) waarbij het datatype voor *value* ANY is, moet in ieder geval het datatype voor de instantiatie door middel van de *xsi:type* aanduiding worden aangegeven.

XML-voorbeelden

1) Meetwaarde met UCUM eenheid (165 cm)

```
<value value="165" unit="cm"/>
```

2) Meetwaarde met UCUM eenheid (92,1 kg)

```
<value value="92.1" unit="kg"/>
```

3) Meetwaarde met UCUM eenheid (bloeddruk 120 mmHG)

```
<value value="120" unit="mm[Hg]"/>
```

4) Meetwaarde zonder eenheid (5)

```
<value value="5"/>
```

5) Meetwaarde met alternatieve representatie (1 [stuk], in HL7v3 altijd zonder eenheid, vertaald naar twee alternatieve eenheden [WCIA tabel 25 doseereenheid en G-Standaard Thesaurus 002 doseereenheid])

```
<value value="1">
  <translation value="1" code="245"
    codeSystem="2.16.840.1.113883.2.4.4.1.900.2"
    displayName="Stuk"/>
  <translation value="1" code="100"
    codeSystem="2.16.840.1.113883.2.4.4.1.361"
    displayName="tablet"/>
</value>
```

6.28 MO (Monetary Amount - hoeveelheid geld)

Attributen:

@value..... waarde (REAL)

@currency..... valuta (CS)

Elementen:

geen

Dit datatype wordt gebruikt om een hoeveelheid (value) geld aan te duiden in een bepaalde valuta (currency). MO vertoont veel overeenkomsten met het PQ type, er is echter een essentieel verschil: Valutakoersen zijn variabel, dit in tegenstelling tot de omrekening bij fysieke eenheden, vandaar dat geld niet in een PQ type kan worden uitgedrukt.

Valuta wordt gecodeerd volgens ISO 4217.

Tabel: codes voor het currency attribuut (belangrijkste codes)		
<i>code</i>	<i>naam</i>	<i>land</i>
EUR	Euro	Europese Unie
GBP	Brits Pond	Verenigd Koninkrijk
USD	Amerikaanse Dollar	Verenigde Staten

XML-voorbeeld

```
<value xsi:type="M0" currency="EUR" value="97.32"/>
```

6.29 TS (Time Stamp - tijdstip)

Attributen:

@value..... waarde (ST)

Elementen:

geen

De 'waarde' van een tijdstip wordt vastgelegd m.b.v. de ISO 8601 notatie die gebruikelijk is binnen HL7. Een tijdstip dient volgens het volgende formaat te worden doorgegeven:

"YYYY[MM[DD[HH[MM[SS[U[U[U[U]]]]]]][+|-ZZ[zz]]"

De betekenis van de componenten in deze syntax is als volgt:

YYYY	Het jaar volgens de Gregoriaanse kalender (waarden 0001-9999)
MM	De maand o.b.v. het volgnummer in het jaar (waarden 01-12)
DD	De dag o.b.v. het volgnummer in de maand (waarden 01-31)
HH	Het uur o.b.v. een 24-uurs notatie van de klok (waarden 00-23)
MM	De minuut binnen het uur (waarden 00-59)
SS	De seconde binnen de minuut (waarden 00-59)
UUUU	Fractie van een seconde (waarden 0-9999)
+ -	Aanduiding of de tijdzone voor of achter UTC is
ZZ	Het verschil in uren t.o.v. UTC (waarden 00-14)
zz	De bijkomende minuten verschil t.o.v. UTC (waarden 00, 30 of 45)

Noot: UTC staat voor Coordinated Universal Time. Voorheen werd dit wel aangeduid als Greenwich Mean Time (GMT). Het verschil is dat UTC niet afhangt van zomer/wintertijd, d.w.z. dat de UTC in de tijdzone van Greenwich "+01" wordt als het daar zomertijd is.



Afgezien van bovenstaande ranges voor de waarden van de componenten, geldt dat de combinatie van de componenten jaar, maand en dag een **geldige kalenderdatum** moet vormen (dus bijvoorbeeld geen 30 februari).

Als een systeem niet over voldoende informatie beschikt (bijvoorbeeld: alleen het jaar is bekend), kan van rechts naar links worden afgekapt. In principe geeft HL7 daarin volledige vrijheid, maar de interpretatie die binnen Nederland wordt gevolgd is dat het alleen toegestaan is per gehele component (bijvoorbeeld dat niet alleen het eerste cijfer van de minuten mag worden genoemd, maar alleen één minuten óf twee cijfers mogelijk is). Uitzondering op deze regel vormt de fractie van een seconde (UUUU component), die wel een willekeurig aantal cijfers mag hebben (met een maximum van 4 cijfers na de punt).



Een informatiesysteem mag een Time Stamp alleen doorgeven met een precisie die kleiner of gelijk is aan de precisie die het zelf gebruikt binnen de software. Het is dus niet toegestaan om 0'en toe te voegen voor seconden of fracties ervan. Met andere woorden: een component moet alleen gebruikt worden als het ook echt een reële waarde kan krijgen.



Merk op dat het zeker nuttig (of zelfs nodig) kan zijn om specifieke restricties voor het gebruik van TS aan te geven bij elementen in HL7 berichten. Zo zou bijvoorbeeld bij de verzendtijd van een HL7 bericht (creationTime in de transmission wrapper) kunnen worden geëist dat minimaal seconden worden doorgegeven. Zie hiervoor de diverse implementatiehandleidingen.



Bij het opstellen van *conformance profiles* door leveranciers of instellingen is het van belang om aan te geven welke precisie wordt gehanteerd voor specifieke time stamps. Zonder nadere aanduiding daarvan is de ontvanger namelijk verplicht om de volledige precisie die de verzender hanteert te kunnen verwerken, opslaan en reproduceren. Dit is vergelijkbaar met het aanduiden van de maximumlengte van strings of het aantal decimalen van reals.

Met betrekking tot de tijdzone geldt dat deze optioneel kan worden aangeduid, **indien de rest van de time stamp minimaal uren omvat**. Als een tijdzone wordt aangeduid, geldt dat deze minimaal bestaat uit een + of – en 2 cijfers (de uren van het tijdsverschil), met optioneel nog 2 cijfers (de minuten van het tijdsverschil, met “00” als defaultwaarde).



Als geen tijdzone wordt aangeduid is de aanname dat het tijdstip betrekking heeft op de tijdzone van het systeem dat de tijd heeft vastgelegd. Dit zal zelden een probleem zijn in een land als Nederland, maar kan van belang zijn i.v.m. toekomstige internationale uitwisseling.

XML-voorbeelden

<code><effectiveTime value="1967"/></code>	<code><!--1967 (alleen jaar bekend)--></code>
<code><effectiveTime value="196703"/></code>	<code><!--maart 1967 (alleen maand)--></code>
<code><effectiveTime value="19670330"/></code>	<code><!--30 maart 1967 (datum)--></code>
<code><effectiveTime value="2007050316"/></code>	<code><!--3 mei 2007, 16 tot 17 uur--></code>
<code><effectiveTime value="200705031606"/></code>	<code><!--3 mei 2007, om 16:06--></code>
<code><effectiveTime value="20070503160614"/></code>	<code><!--3 mei 2007, om 16:06:14--></code>
<code><effectiveTime value="20070503160614.0843"/></code>	<code><!--zelfde, met fractie--></code>
<code><effectiveTime value="2007050316+02"/></code>	<code><!--tijdzone A'dam zomertijd--></code>
<code><effectiveTime value="200705031606+0530"/></code>	<code><!--tijdzone in Mumbai, India--></code>
<code><effectiveTime value="20070503160614-05"/></code>	<code><!--New York in wintertijd--></code>

6.30 BAG (Bag – verzameling met mogelijke dubbelen)

Sommige modelonderdelen dragen als datatype BAG<T> met T als een discreet of continue datatype. Dit betekent een ongesorteerde verzameling van waarden, waar waarden ook meer dan eenmaal kunnen voorkomen.

In de XML-representatie wordt een BAG<T> met een cardinaliteit van n..m omgezet naar een herhaalbaar XML-element met de cardinaliteit n..*.

6.31 SET (Set – verzameling zonder dubbelen)

Sommige modelonderdelen dragen als datatype SET<T> met T als een discreet of continue datatype. Dit betekent een ongesorteerde verzameling van waarden, waar waarden maximaal een keer kunnen voorkomen.

In de XML-representatie wordt een SET<T> met een cardinaliteit van n..m omgezet naar een herhaalbaar XML-element met de cardinaliteit n..*.

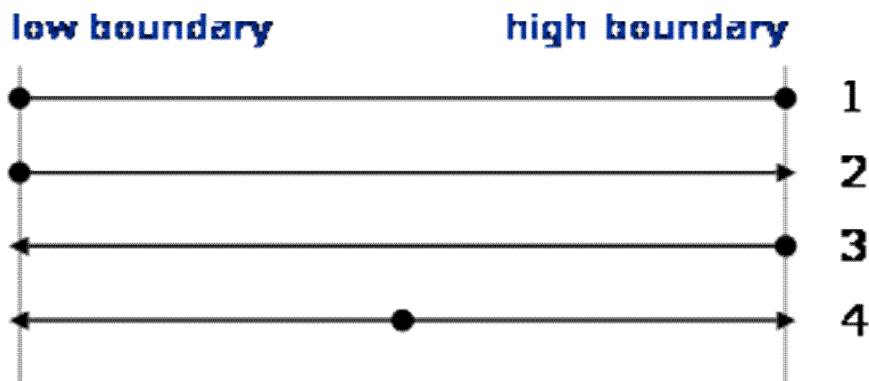
6.32 SXCM (Set Component – deelverzameling)

Er bestaat een generieke set component die een component representeert van een discreet of continue datatype en is bedoeld als onderdeel van een verzameling. Dit wordt in Nederland op dit moment alleen bij het datatype GTS gebruikt (zie GTS).

6.33 IVL (Interval)

In de Nederlandse situatie zijn tot nu toe de volgende intervaldefinities gebruikt. Intervallen zijn alleen gedefinieerd voor datatypes met een ordinaal of interval schaaltype. Er zijn verschillende mogelijkheden, een interval te bepalen (zie onderstaande figuur), zoals:

- een ondergrens <low> en een bovengrens <high> (1),
- een ondergrens <low> en een breedte <width> (2)
- een bovengrens <high> en een breedte <width> (3)
- of het midden <center> van het interval en een breedte <width> (4)



Er zijn vanzelfsprekend nog meer mogelijkheden. Ook hoeft een interval niet altijd volledig gespecificeerd te zijn (open intervallen). Zo kan bijvoorbeeld met alleen een bovengrens een maximale dosis worden aangeduid.

Het datatype kan ook worden gebruikt voor het doorgeven van een exacte waarde in het attribuut @value. Er zijn dan geen andere elementen toegestaan. Dit principe heet *run-time demotie*. Andersom kan het attribuut @value niet voorkomen als er een van de elementen <low>, <high>, <center>, <width> wordt gebruikt.

Voor het aanduiden van een interval worden alleen de volgende combinaties toegestaan:

- een ondergrens <low> en een bovengrens <high>, welke niet aan elkaar gelijk mogen zijn, en waarvan de precisie gelijk is. Bij intervallen van tijd kan echter wel verschil in precisie bestaan als bijvoorbeeld een specialist medicatie stopt op een exacte datum terwijl alleen het jaar bekend is wanneer de patiënt begonnen was met gebruik:
 - Correct zijn bijvoorbeeld:
 - een interval van 5 tot 6
 - een interval van 20090101 tot en met 20100101
 - een interval van 2009 tot 20100405
 - Incorrect zijn bijvoorbeeld:
 - een interval van 5 tot en met 5
 - een interval van 5 tot en met 6.0
- een ondergrens <low> en een breedte <width>
- alleen een ondergrens <low>, dit is functioneel gelijk aan <low> met <high>.@nullFlavor="PINF" (PINF = Positief oneindig)
- alleen een bovengrens <high>, dit is functioneel gelijk aan <high> met <low>.@nullFlavor="NINF" (NINF = Negatief oneindig)
- alleen het midden van het interval <center>
- alleen de breedte van een interval <width>

Voor de volgende datatypen zijn intervalspecificaties gedefinieerd:

- [IVL<INT>](#) (Interval van integers)
- [IVL<PQ>](#) (Interval van hoeveelheden)
- [IVL<TS>](#) (Interval van tijd)

6.34 IVL_TS (Interval of Time Stamps - tijdsinterval)

Attributen:

@value.....tijdstip (ts)

Elementen:

<low>..... ondergrens (TS)

<high> bovengrens (TS)

<center>midden van het tijdsinterval (TS)

<width>..... tijdsduur van het interval (PQ)

Elementen van dit datatype duiden tijdsintervallen aan. Daarbij wordt in de regel een boven- en een ondergrens als tijdstip aangeduid. Bij open tijdsintervallen (bijvoorbeeld vanaf tijdstip X, of geldig tot tijdstip Y) wordt slechts één van de beide elementen aangegeven.

De tijdzone op alle tijdwaarden in alle delen van het interval worden verondersteld gelijk te zijn, tenzij anders vermeld.

@value.....tijdstip (ts)

Bevat een exact tijdstip. Het datatype is dan effectief gelijk aan het datatype TS. Dit attribuut mag niet in combinatie met de elementen *<low>*, *<high>*, *<center>*, *<width>* worden gebruikt.

<low>..... ondergrens (TS)

Bevat het begintijdstip van een tijdsinterval in het XML-attribuut *@value*. In Nederland kan *low* op de volgende manieren worden toegepast: als enige element van IVL_TS (als het eindtijdstip niet bekend is) of in combinatie met *high* of *width* (als het eindtijdstip of de duur bekend is).

De waarde in *low* betekent "een begintijdstip later dan, of gelijk aan". Door daarbij expliciet aan te geven dat deze grens niet inclusief is (*inclusive="false"*), wordt de betekenis gewijzigd in "een begintijdstip later dan" (zie beschrijving *@inclusive*). Merk op dat de interpretatie **niet** afhangt van de nauwkeurigheid van het tijdstip: "na 2004" betekent "vanaf 1 Januari 2005 00:00:00.000", "na 20041201" betekent "vanaf 2 december 2004 00:00:00.000 uur", en "na 200412011200" betekent "vanaf 1 december 2004 12:01:00.000".

@inclusive..... (true|false)

De standaardwaarde is "true", dus inclusief het tijdstip in het XML-attribuut *@value*. De waarde "false" betekent dat het interval op het eerste moment na de in *@value* genoemde waarde begint.

<high> bovengrens (TS)

Bevat het eindtijdstip van een tijdsinterval in het XML-attribuut *@value*. In Nederland kan *high* op de volgende manieren worden toegepast: als enige element van IVL_TS (als

het begintijdstip niet bekend is) of in combinatie met *low of width* (als het begintijdstip of de duur bekend is).

De waarde in *high* betekent "een eindtijdstip voorafgaande aan, of gelijk aan". Door daarbij expliciet aan te geven dat deze grens niet inclusief is (*inclusive="false"*), wordt de betekenis gewijzigd in "een eindtijdstip voorafgaande aan" (zie beschrijving *@inclusive*).

Merk op dat de betekenis **niet** afhangt van de nauwkeurigheid van het tijdstip: "voorafgaande aan 2008" betekent "voor 1 januari 2008 00:00:00.000", "voorafgaande aan 20081201" betekent "voor 1 december 2008 00:00:00.000", en "voorafgaande aan 200412011200" betekent "voor 1 december 2004 12:00:00.000".

@inclusive (true|false)

De standaardwaarde is "true", dus inclusief het tijdstip in het XML-attribuut *@value*. De waarde "false" betekent dat het interval op het laatste moment voor de in *@value* genoemde waarde eindigt.

<center> *midden van het tijdsinterval (TS)*

Bevat een tijdstip dat het midden vormt van het tijdsinterval. In Nederland wordt dit uitsluitend toegepast indien men (in een element van het datatype IVL<TS>) één specifiek tijdstip wil doorgeven in plaats van een tijdsinterval. *center* wordt in Nederland nooit gebruikt in combinatie met *low*, *high*, of *width*.

Merk op dat *<center>* in tegenstelling tot *<low>* en *<high>* **geen** nadere impliciete nauwkeurigheid in zich draagt: "Op (in) 2005" betekent "in het jaar 2005", "op 20051201" betekent "op de dag 1 december 2005", en "200512011200" betekent "op 1 december 2005 om 12:00 uur".

<width> *tijdsduur van het interval (PQ)*

Bevat de tijdsduur van het interval. In Nederland wordt dit slechts in twee situaties toegepast: 1) als enige element van IVL_TS (in die gevallen waarin niets anders dan de duur van het interval bekend is) of 2) in combinatie met het *low* element, om aan te geven dat het interval een bepaalde duur heeft vanaf het beginmoment. Die laatste situatie is feitelijk synoniem met het gebruik van *low* in combinatie met *high*, maar is vaak handiger als de looptijd wordt aangeduid (en de einddatum daaruit afgeleid).



Het *<width>* element is van datatype PQ (Physical Quantity). De waarde moet natuurlijk betrekking hebben op een tijdsduur. Het is daarbij **verplicht om een tijdseenheid aan te duiden**. Hierbij zijn de eenheden 'us' (microseconde), 'ms' (miliseconde), 's' (seconde), 'min' (minuut), 'h' (uur), 'd' (dag), 'wk' (week), 'mo' (maand) en 'a' (jaar) beschikbaar.



Merk op dat ten aanzien van het gebruik van IVL_TS bij het doorgeven van doseerschema's in medicatievoorschriften (als onderdeel van het datatype GTS) aanvullende regels zijn vastgelegd binnen het Nictiz EMD project.

XML-voorbeelden

1) vanaf 7 mei 2004 00:00:00.000 tot en met 9 september 2004 00:00:00.000

```
<effectiveTime>  
  <low value="20040507"/>
```

```
<high value="20040909"/>
</effectiveTime>
```

2) vanaf 7 mei 2004 00:00:00.000

```
<effectiveTime>
  <low value="20040507"/>
</effectiveTime>
```

3) Op 3 januari 1975

```
<value>
  <center value="19750103">
</value>
```

4) In 2003

```
<value>
  <center value="2003">
</value>
```

5) vanaf 3 januari 1975 00:00:00.000, maar vóór 7 januari 1975 00:00:00.000

```
<value>
  <low value="19750103">
  <high value="19750107" inclusive="false">
</value>
```

6) gedurende 2 weken

```
<value>
  <width value="2" unit="wk"/>
</value>
```

7) gedurende 120 dagen, vanaf 1 januari 2008 00:00:00.000

```
<value>
  <low value="20080101"/>
  <width value="120" unit="d"/>
</value>
```

6.35 IVL_INT (Interval of Integers – numeriek interval)

Attributen:

@value..... integer (int)

Elementen:

<low>..... ondergrens (INT)

<high>bovengrens (INT)

Elementen van dit datatype duiden intervallen van integer (gehele) getallen aan. Daarbij wordt in de regel een boven- en een ondergrens als getal aangeduid. Bij open intervallen (bijvoorbeeld bij een uitgifte van een herhaling "hooguit 3 keer herhalen") wordt slechts een van de beide elementen aangegeven.

@value..... integer (int)

Bevat een exacte integer. Het datatype is dan effectief gelijk aan het datatype INT. Dit attribuut mag niet in combinatie met de elementen <low>, <high> worden gebruikt.

<low>..... ondergrens (INT)

@inclusive (true|false)

De standaardwaarde is "true", dus inclusief de waarde in het XML-attribuut @value. De waarde "false" betekent dat het interval bij de eerstevolgende integer na de in @value genoemde waarde begint.

<high>bovengrens (INT)

@inclusive (true|false)

De standaardwaarde is "true", dus inclusief de waarde in het XML-attribuut @value. De waarde "false" betekent dat het interval bij de hoogste integer voorafgaande de in @value genoemde waarde eindigt.

XML-voorbeeld

1) Van en met 3 tot en met 5

```
<repeatNumber>
  <low value="3"/>
  <high value="5"/>
</repeatNumber>
```

2) Tot en met 5

```
<repeatNumber>
  <high value="5"/>
</repeatNumber>
```


6.36 IVL_PQ (Interval of Physical Quantities - hoeveelheidsinterval)

Attributen:

@value.....hoeveelheid (real)

@unit.....meeteenheid (cs)

Elementen:

<low>.....ondergrens (PQ)

<high>.....bovengrens (PQ)

<center>.....exacte waarde (PQ)

Elementen van dit datatype duiden intervallen van physical quantities aan. Daarbij wordt in de regel een boven- en een ondergrens als *value-unit* paar aangeduid. Ook is het mogelijk om in plaats van <low> en <high> gebruik te maken van <center>.

Als de elementen <low> en <high> beiden gebruikt worden moeten deze in dezelfde eenheid in het attribuut @unit worden uitgedrukt.

@value.....hoeveelheid (real)

Bevat een exacte hoeveelheid. Het datatype is dan effectief gelijk aan het datatype PQ. Dit attribuut mag niet in combinatie met de elementen <low>, <high>, <center> worden gebruikt.

In *unit* staat de meeteenheid, uitgedrukt conform de *Unified Code for Units of Measure* (UCUM). Deze tabel bevat alle natuurlijke meeteenheden.

@unit.....meeteenheid (cs)

Als *value* "niet telbaar", "meetbaar" is (bijvoorbeeld uitgedrukt in milligram of liter) dan moet deze eenheid in dit attribuut worden opgenomen. Als *value* 'telbaar' is (bijvoorbeeld tabletten) dan is dit attribuut niet aanwezig.

Voor een volledige beschrijving van de mogelijke eenheden, zie de *Unified Code for Units of Measure* ([UCUM]) specificatie. UCUM is een verzameling van atomaire eenheden en prefixes en een bijhorende grammatica om geldige combinaties daarvan op te bouwen.

Als er een *nullFlavor* attribuut aanwezig is, mogen *value* en *unit* niet aanwezig zijn; als er geen *nullFlavor* attribuut is, dan moet een *value* aanwezig zijn.

Een alternatieve representatie van dezelfde fysieke hoeveelheid, uitgedrukt in een andere eenheid, mogelijk uit een ander systeem dan UCUM en mogelijk met een andere bijbehorende waarde (*value*) kan in de translation worden weergegeven.

<low>.....ondergrens (PQ)

Dit geeft de ondergrens van het interval aan (d.w.z. de minimale hoeveelheid).

<high>.....bovengrens (PQ)

Dit geeft de bovengrens van het interval aan (d.w.z. de maximale hoeveelheid).

<center>*exacte waarde (PQ)*

Dit element wordt gebruikt als geen sprake is van een interval, maar van een exacte waarde. Alternatief zou zijn om het datatype IVL_PQ in dat geval terug te brengen tot PQ, maar dit is bij verwerking van de XML lastiger, dus dit wordt in Nederland uitgesloten. Het element <center> mag niet in combinatie met <low> of <high> voorkomen.

XML-voorbeelden

1) Van 7,5 tot en met 10,0 mmol/l.

```
<referenceRange>
  <low value="7.5" unit="mmol/l"/>
  <high value="10.0" unit="mmol/l"/>
</referenceRange>
```

2) Precies 9,0 stuks.

```
<doseQuantity>
  <center value="9" unit="1"/>
</doseQuantity>
```

3) Van 7,5 milliliter tot en met 10000,0 milliliter.

```
<referenceRange>
  <low value="7.5" unit="mL"/>
  <high value="10000.0" unit="mL"/>
</referenceRange>
```

4) Van 0,0075 liter tot en met 10,0 liter.

```
<referenceRange>
  <low value="0.0075" unit="L"/>
  <high value="10.0000" unit="L"/>
</referenceRange>
```

5) < 10 mmol/L. Merk op dat <low> ook weggelaten had kunnen worden

```
<value>
  <low nullFlavor="NINF"/>
  <high value="10" unit="mmol/L"/>
</value>
```

6) > 10 mmol/L. Merk op dat <high> ook weggelaten had kunnen worden

```
<value>
  <low value="10" unit="mmol/L"/>
  <high nullFlavor="PINF"/>
</value>
```

6.37 RTO (Ratio - onderlinge verhouding tussen twee gegevens)

Attributen:

geen

Elementen:

<numerator> teller (QTY)

<denominator> noemer (QTY)

Elementen van dit datatype hebben twee subelementen, de teller (numerator) en de noemer (denominator) van een vergelijking (Ratio). Het QTY (Quantity) datatype is een abstract datatype en wordt in berichten vervangen door een specifiek datatype, veelal PQ of TS. De relevante structuurelementen zijn:

<numerator> teller (QTY)

De hoeveelheid die gedeeld wordt in de ratio. De default waarde is de integer 1.

<denominator> noemer (QTY)

De hoeveelheid waardoor de numerator gedeeld wordt in de ratio. De default waarde is de integer 1. De denominator mag niet de waarde 0 hebben. De *denominator* is veelal een tijdseenheid.

XML-voorbeelden

1) 6 (keer/stuk) per (één) dag

```
<maxDoseQuantity>
  <numerator xsi:type="PQ" value="6"/>
  <denominator xsi:type="PQ" value="1" unit="d"/>
</maxDoseQuantity>
```

2) 400 milligram per (één) dag

```
<maxDoseQuantity>
  <numerator xsi:type="PQ" value='400' unit='mg'/>
  <denominator xsi:type="PQ" value='1' unit='d'/>
</maxDoseQuantity>
```

3) 5 per (één) uur

```
<maxDoseQuantity>
  <numerator xsi:type="PQ" value="5"/>
  <denominator xsi:type="PQ" value="1" unit="h"/>
</maxDoseQuantity>
```

4) 1:10000 bijvoorbeeld bij epidemiologische gegevens of bij lab resultaten

```
<value xsi:type="RTO_QTY_QTY">  
  <numerator xsi:type="INT" value="1"/>  
  <denominator xsi:type="INT" value="10000"/>  
</value>
```

6.38 GTS (General Timing Specification - generieke tijdspecificatie)

Attributen:

@value.....tijdstip (TS)

Elementen:

<comp>tijdcomponent (SXCM_TS)

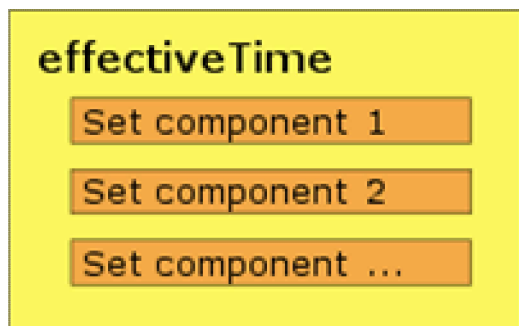
Het volgende beschrijft het GTS (General Timing Specification, generieke tijdsopgave) datatype, waarbij uitsluitend die onderdelen worden beschreven die in de Nederlandse situatie van toepassing zijn. Indien er specifieke aanwijzingen zijn hoe iets in Nederland geïmplementeerd moet worden dan is dit in de tekst aangegeven.

Het datatype GTS biedt een zeer rijke (maar ook complexe) structuur voor het aanduiden van tijdstippen, tijdsintervallen en tijdsschema's met een vrijwel onbeperkt complexe opbouw. De studie van GTS is een onderwerp op zich, maar gelukkig zijn de meeste toepassingen uit te drukken met een beperkte set GTS functies, zoals TS en IVL<TS>.

Veréénvoudigd is een GTS een SET<TS>. Deze kunnen worden gecombineerd tot sets van periodes en door set operaties is het mogelijk op deze manier ook complexe tijds-aanduidingen te specificeren. Het volgende plaatje schetst als voorbeeld een tijds-aanduiding van bijvoorbeeld een instructie voor medicatietoediening "op even dagen (begin met dag 0, de eerste dag dat de medicatie wordt toegediend) door de week ma tot en met vr" als een set van dagen. Het set van weekdays (eerste rij) wordt met een set van de dagen ma tot en met vr, gedefinieerd als een tijdsinterval doorsneden. Dit levert de derde rij. Deze wordt met een set van even dagen doorsneden en levert uiteindelijk rij vijf met het gewenste resultaat.



Er zijn verschillende XML-representaties van een element met als type GTS denkbaar, maar in Nederland heeft een volledig element van type GTS altijd het formaat:



Het XML-element `effectiveTime` wordt gedeclareerd als `SXPR_TS` (parenthetic set expression van type `TS`) en de set componenten `<comp>` als kind-elementen toegevoegd.

De tijdzone op alle tijdwaarden in alle delen van de expressie worden verondersteld gelijk te zijn, tenzij anders vermeld.

Dat betekent dat binnen het XML-element gebruik gemaakt wordt van de Set Component elementen. In XML heeft dus een GTS datatype altijd het volgende patroon.

```
<effectiveTime xsi:type="SXPR_TS">
  <comp xsi:type="...">
    ...
  </comp>
  <comp xsi:type="..." operator="...">
    ...
  </comp>
</effectiveTime/>
```

`<comp>` `tijdcomponent (SXPR_TS)`

Als set componenten zijn de datatypen `PIVL` (periodic interval of time) en `EIVL` (event related interval of time) toegestaan. In Nederland wordt op dit moment alleen `PIVL` gebruikt. Een set component wordt door de `xsi:type` aanduiding tot een datatype [PIVL TS](#), [IVL TS](#) of [TS](#) gemaakt. De set component kan echter ook zelf weer van type `SXPR_TS` zijn, zodat een geneste structuur ontstaat.

`@operator` `bewerking op verzameling (A|I|E)`

Het attribuut `operator` is **verplicht** waar het element `<comp>` niet het eerste element is van een **SXPR verzameling**. Bij tweede en verdere elementen van een `SXPR` verzameling moet immers worden aangegeven wat de relatie is met de **voorgaande** elementen (of liever gezegd met de tot dan toe opgebouwde verzameling).

De geldige operatoren bij voor componenten van een `SXPR_TS` zijn:

Domain SetOperator	
code	definition
I	Neem de vereniging met het voorgaande tijdschema.
E	Neem het verschil met het voorgaande tijdschema.
A	Neem de doorsnede met het voorgaande tijdschema.

Zie ook het datatype [PIVL TS](#) voor meer voorbeelden rondom het gebruik van operatoren.

```
<effectiveTime xsi:type="SXPR_TS">
  <comp xsi:type="IVL_TS">
    een gewone intervalaanduiding
  </comp>
```

```
<comp xsi:type="PIVL_TS" operator="A">  
  periodieke intervalaanduidingen  
  (hier met een operator tussen de twee sets)  
</comp>  
</effectiveTime>
```

Omdat GTS een superklasse van alle tijdsaanduidingen in HL7 is, kan een element van datatype GTS in een XML-instance ook tot bijvoorbeeld een interval worden gemaakt door het xsi:type attribuut te gebruiken. Stel dat effectiveTime is gedefinieerd als GTS in een model, dan zijn in de volgende XML-instances allemaal geldige tijdsaanduidingen.

```
<effectiveTime xsi:type="IVL_TS">  
  <low value="20050924"/>  
  <high value="20050925"/>  
</effectiveTime >
```

```
<effectiveTime xsi:type="SXPR_TS">  
  <comp xsi:type="IVL_TS">  
    <low value="20050901"/>  
    <width value="90" unit="d"/>  
  </comp>  
  <comp xsi:type="PIVL_TS" operator="A">  
    <period value="2" unit="d"/>  
  </comp>  
</effectiveTime>
```



Merk op dat ten aanzien van het gebruik van GTS bij het doorgeven van doseerschema's in medicatievoorschriften aanvullende richtlijnen (in de vorm van restricties) zijn vastgelegd binnen het Nictiz-project Medicatiegegevens.

6.39 PIVL_TS (Periodic Interval of Time - periodiek herhalend tijdsinterval)

Attributen:

@alignment uitgelijnd op ... (cs)

@institutionSpecified stiptheidsindicator (bl), niet gebruiken in NL

Elementen:

<phase> basisinterval (IVL<TS>)

<period> herhaalperiode PQ [time]

Structuur: Het datatype PIVL is een extensie op datatype SXCM (*Set Component*) en is dus bedoeld als onderdeel van een verzameling. In dit specifieke geval bestaat zo'n verzameling uit **tijdspecificaties** die tezamen de beschrijving geven van een bepaald tijdschema. De tijdspecificaties worden daarbij aan elkaar gerelateerd d.m.v. operatoren (set operators), zoals 'vereniging', 'verschil' en 'doorsnede'. Dit is niet eenvoudig uit te leggen zonder voorbeelden te geven, hetgeen hieronder dan ook uitgebreid zal gebeuren. Toepassing: Berichtelementen met datatype PIVL komen alleen voor als onderdeel van het (abstracte) datatype GTS. Deze *General Timing Specification* geeft een zeer algemene methode om tijdschema's weer te geven en bestaat per definitie uit een verzameling aan elkaar gerelateerde componenten. Eén van de belangrijkste soorten componenten is het datatype PIVL. Hiermee worden terugkerende patronen in de tijd weergegeven, zoals '2x per dag', of 'om de 2 weken op maandag van 10:00 tot 10:30'.

XML-voorbeelden

2x per dag

```
<comp xsi:type="PIVL_TS">
  <period value="0.5" unit="d"/>
</comp>
```

Elke 2 weken op maandag van 10:00 tot 10:30

```
<comp xsi:type="PIVL_TS" alignment="DW">
  <phase>
    <low value="200508291000"/>
    <width value="30" unit="min"/>
  </phase>
  <period value="2" unit="wk"/>
</comp>
```

Noot: in beide bovenstaande voorbeelden heeft het berichtelement de naam <comp>. Dit geeft aan dat het elementen zijn binnen het datatype SXPR (Parenthetic Set Expression), hetgeen de vorm is waarin PIVL meestal voorkomt. Meer uitleg over SXPR is te vinden bij het datatype GTS (General Timing Specification) elders in dit document.

Deze componenten duiden een tijdschema aan door aan elkaar gerelateerd te worden. Dit kan het beste worden toegelicht aan de hand van een aantal voorbeelden uit de praktijk.

Stel dat een medicatievoorschrift is uitgeschreven met een looptijd van 90 dagen vanaf 1-9-2005 (d.w.z. de beoogde toedieningsperiode loopt van 1-9-2005 t/m 29-11-2005). Het element *effectiveTime* in het gebruiksvoorschrift heeft datatype GTS en bestaat uit een verzameling van het datatype SXPR_TS (*Parenthetic Set Expression*), die begint met het interval tussen deze twee data. Het is dus een verzameling met daarin een IVL<TS>:

XML-voorbeelden

Beoogde toedieningsperiode

```
<effectiveTime xsi:type="SXPR_TS">
  <comp xsi:type="IVL_TS">
    <low value="20050901"/>
    <width value="90" unit="d"/>
  </comp>
</effectiveTime>
```

Noot: In bovenstaand voorbeeld komt op zich het datatype PIVL nog niet eens voor, maar het dient als basis om uit te leggen hoe de operator in PIVL gebruikt wordt om de verzameling te definiëren. Zie verder de beschrijving bij het algemene datatype GTS.

Stel nu dat in de betreffende periode elke 2 dagen een toediening plaats moet vinden. In feite moet dan de doorsnede worden genomen van het aangegeven interval en een periodiek herhalend 'iets', waarvan alleen bekend is dat het elke 2 dagen plaatsvindt.

Elke 2 dagen binnen het aangeduide interval

```
<effectiveTime xsi:type="SXPR_TS">
  <comp xsi:type="IVL_TS">
    <low value="20050901"/>
    <width value="90" unit="d"/>
  </comp>
  <comp xsi:type="PIVL_TS" operator="A">
    <period value="2" unit="d"/>
  </comp>
</effectiveTime>
```

De operator "A" in de hierboven toegevoegde PIVL component heeft de betekenis: 'Neem de **doorsnede** van de eerder aangeduide tijden (interval van 90 dagen vanaf 1-9-2005) met de herhaalperiode (*period*) die daarna wordt aangegeven. Het resultaat is dus 'elke 2 dagen van 1-9-2005 t/m 29-11-2005'. Dit levert dus 45 toedienmomenten op, zonder dat precies is aangegeven op welk moment van de dag deze momenten plaatsvinden.

<phase>basisinterval

Het element *phase* geeft als het ware een **voorbeeld (prototype)** van het basisinterval (of basistijdstip) dat in het PIVL datatype periodiek herhaald wordt. Zo kan bijvoorbeeld 'elke dag om 14:00' worden aangegeven in PIVL door als *phase* het tijdstip '14:00' op een willekeurige dag aan te duiden en bij *period* aan te geven dat dit dagelijks herhaalt. Het element *phase* is **niet verplicht**, aangezien het ook voorkomt dat alleen een herhaalperiode moet worden aangeduid en geen basistijdstip of -interval nodig is. Zo is bij 'elke 2 dagen' in het laatste voorbeeld hierboven geen *phase* nodig, omdat niet relevant

is op welk moment binnen elke 2 dagen de gebeurtenis plaatsvindt. Hieronder zullen enkele voorbeelden worden gegeven van situaties waarbij dat wel het geval is.

XML-voorbeelden

Elke dag om 8:00

```
<comp xsi:type="PIVL_TS">
  <phase>
    <center value="200509010800"/>
  </phase>
  <period value="1" unit="d"/>
</comp>
```

Merk op dat het feit dat hier 1 september 2005 wordt gebruikt feitelijk niet relevant is! De phase is alleen aanwezig omdat het tijdstip 08:00 moet worden doorgegeven en dient dus als een prototype van zo'n tijdstip. Daarbij moet wel een datum worden vermeld, omdat het doorgeven van een los tijdstip in HL7 versie 3 (datatype TS) niet mogelijk is! Er had dus ook `<center value='198812040800'>` kunnen staan en dan zou de verwerkende software daar exact dezelfde conclusie uit moeten trekken (elke dag om 08:00). Merk ook op dat phase altijd de vorm van een interval heeft (datatype IVL_TS). Als dus één enkel tijdstip (of één enkele dag) moet worden aangeduid, dan kan dat het beste gebeuren door het element `<center>` van IVL_TS toe te passen. Daarmee wordt het interval in feite gereduceerd tot een tijdstip, maar met behoud van de structuur van IVL.

Elke dag om 8:00, gedurende 10 minuten

```
<comp xsi:type="PIVL_TS">
  <phase>
    <low value="200509010800"/>
    <width value="10" unit="min"/>
  </phase>
  <period value="1" unit="d"/>
</comp>
```

Het enige verschil met het voorgaande voorbeeld is dat de *phase* nu een 'echt' interval is i.p.v. een enkelvoudig tijdstip. Dit wordt bijvoorbeeld gebruikt als een medicatietoediening, een fysiotherapiebehandeling of een andere activiteit gedurende een bepaalde tijd moet worden uitgevoerd. Dit kan ook voorkomen als het tijdstip feitelijk niet relevant is:

Elke dag, gedurende 10 minuten

```
<comp xsi:type="PIVL_TS">
  <phase>
    <width value="10" unit="min"/>
  </phase>
  <period value="1" unit="d"/>
</comp>
```

In dit voorbeeld is van de *phase* alleen nog maar de duur (*width*) relevant.

@alignment uitgelijnd op

Het attribuut *alignment* wordt gebruikt om aan te geven welk aspect van de *phase* (het basisinterval) relevant is voor het bepalen van het herhaalpatroon. Dat klinkt in eerste instantie zeer verwarrend, maar het geeft de mogelijkheid om vrij eenvoudig herhaalpatronen als 'elke maandag' of 'elke 14^e van de maand' of 'elke 2 mei' aan te duiden.

XML-voorbeelden

Elke maandag

```
<effectiveTime xsi:type="PIVL_TS" alignment="Dw">
  <phase>
    <center value="20050829"/>
  </phase>
  <period value="1" unit="wk"/>
</effectiveTime>
```

De *alignment* 'DW' geeft hier aan dat het relevante aspect van de aangeduide *phase* de 'day of the week' is. Aangezien 29/8/2005 een maandag was, staat hier dus niets anders dan 'elke maandag'. Zoals eerder vermeld is de precieze waarde van *phase* niet relevant.

De geldige *alignment* varianten bij PIVL_TS zijn voor Nederland beperkt tot:

Domain CalendarCycle:		
Name	code	description
day of the week	DW	elke maandag, dinsdag, woensdag, ...
day of the month	DM	elke 1e, 2e, 3e, 4e, ... van de maand
day of the year	DY	elke 1-1, 2-1, 3-1, 4-1, ... van het jaar

Merk op dat het dus alleen zin heeft om een *alignment* mee te geven als de *phase* de vorm heeft van een enkelvoudig tijdstip of van een interval dat binnen één dag valt.

Elke 15^e van de maand

```
<effectiveTime xsi:type="PIVL_TS" alignment="DM">
  <phase>
    <center value="20050915"/>
  </phase>
  <period value="1" unit="mo"/>
</effectiveTime>
```

Elke 1-3 en 1-8, van 14:00 tot 16:00

```
<effectiveTime xsi:type="SXPR_TS">
  <comp xsi:type="PIVL_TS" alignment="DY">
    <phase>
      <low value="200503011400"/>
      <width value="2" unit="h"/>
    </phase>
    <period value="1" unit="a"/>
  </comp>
  <comp xsi:type="PIVL_TS" operator="I" alignment="DY">
    <phase>
```

```

    <low value="200508011400"/>
    <width value="2" unit="h"/>
  </phase>
  <period value="1" unit="a"/>
</comp>
</effectiveTime>

```

@institutionSpecifiedstiptheidsindicator

Het attribuut *institutionSpecified* is bedoeld om aan te geven of de herhalende periode die door het PIVL datatype wordt aangeduid een 'harde' of 'zachte' tijdspecificatie is. Dat wil zeggen: mag de ontvanger bij een aanduiding van '3x per dag' zelf bepalen op welke tijdstippen de activiteit plaatsvindt of moet dit exact elke 8 uur (gelijke delen) gebeuren.

Er zijn drie redenen waarom dit attribuut vooralsnog in Nederland **niet gebruikt** wordt:

- Er bestaat nog de nodige discussie over de juiste interpretatie van dit attribuut.
- De naam is slecht gekozen, omdat hetzelfde fenomeen zich net zo goed voordoet bij bijvoorbeeld ambulante medicatievoorschriften. In dat geval betekent *institutionSpecified* 'door de patiënt met gezond verstand te bepalen'.

<period>..... herhaalperiode

Het element *period* is **verplicht** aanwezig in elk berichtelement van datatype PIVL en geeft daarin aan wat de herhaalperiode is van de activiteit waarop het betrekking heeft.

De *period* heeft zelf het datatype PQ (*Physical Quantity*), met de beperking dat alleen aanduidingen van een **tijdsduur** mogen worden gebruikt. Het formaat is dus altijd:

```
<period value=[aantal] unit=[tijdseenheid]/>
```

De volgende tijdseenheden (conform UCUM) moeten daarbij altijd ondersteund worden:

name	unit	defined as
second	s	SI-eenheid "the duration of 9,192,631,770 periods of the radiation corresponding to the transition between the two hyperfine levels of the ground state of the caesium-133 atom at zero kelvins"
minute	min	60 s
hour	h	60 min
day	d	24 h
week	wk	7 d
year	a	365.25 d (zie noot)
month	mo	1 a/12

Noot: Merk op dat de definitie van het jaar (unit 'a', afgeleid van 'annum') gebaseerd is op de zogenaamde Juliaanse kalender (met een jaar van exact 365.25 dagen). Daar waar het belangrijk is dat een activiteit altijd op dezelfde dag van de maand of van het jaar valt, zal het alignment attribuut gebruikt worden, zodat geen misverstanden ontstaan.

In het simpelste geval kan op deze manier bijvoorbeeld worden aangeduid dat een activiteit (bijvoorbeeld het toedienen van medicatie) elke 2 uur moet worden uitgevoerd. Dit gebeurt zo:

XML-voorbeelden

Elke 2 uur

```
<comp xsi:type="PIVL_TS">  
  <period value="2" unit="h"/>  
</comp>
```

Bij medicatievoorschriften is het gebruikelijk om bijvoorbeeld '3x per dag' aan te geven als frequentie voor de toediening. Dit kan binnen PIVL op twee manieren worden uitgedrukt:

3x per dag, in dagen

```
<comp xsi:type="PIVL_TS">  
  <period value="0.3333" unit="d"/>  
</comp>
```

3x per dag, in uren

```
<comp xsi:type="PIVL_TS">  
  <period value="8" unit="h"/>  
</comp>
```

In het tweede geval wordt dus omgerekend o.b.v. $1\text{ d} = 24\text{ h}$. Omdat in het PIVL datatype **geen frequentie, maar een herhaalperiode** wordt toegepast, is '3x per dag' alleen in gehele getallen uit te drukken als 'elke 8 uur'. Gevoelsmatig klinkt dat laatste strikter, maar mathematisch zijn de twee aanduidingen op afronding na equivalent.



Omdat de beide aanduidingen (o.b.v. dagen respectievelijk uren) vergelijkbaar zijn, moet elke ontvangende applicatie in staat zijn om ze beide te verwerken. Binnen de verwerkende applicatie moeten ze leiden tot exact **dezelfde interpretatie**. Dit is nog niet triviaal, omdat de weergave in dagen (met een fractie) tot afrondingsfouten kan leiden. Waarschijnlijk is het dus nodig om deze 'hard' te herkennen en om te zetten naar de interne weergave.

Een zendend systeem mag zelf bepalen welke methode de voorkeur heeft. Als richtlijn geldt wel dat bij een niet-gehele *value* (zoals 0.3333 hierboven) er **maximaal 4 relevante decimalen na de punt** worden doorgegeven.

Een vergelijkbaar fenomeen doet zich voor bij herhaalperiodes van meer dan één dag:

1x per 2 dagen

```
<comp xsi:type="PIVL_TS">  
  <period value="2" unit="d"/>  
</comp>
```

Deze situatie is eenvoudig, maar lastiger wordt het bij '3x per week' (op twee manieren):

3x per week, in weken

```
<comp xsi:type="PIVL_TS">
  <period value="0.3333" unit="wk"/>
</comp>
```

3x per week, in dagen

```
<comp xsi:type="PIVL_TS">
  <period value="2.3333" unit="d"/>
</comp>
```

Ook hier is het mogelijk om dit om te werken naar een geheel getal, maar dan moet worden gewisseld naar 'h' als eenheid. De keuze is ook hier aan het zendende systeem, maar elk ontvangend systeem moet in staat zijn om alle drie de varianten te verwerken.

3x per week, in uren

```
<comp xsi:type="PIVL_TS">
  <period value="56" unit="h"/>
</comp>
```

Het is ook mogelijk dat een frequentie wordt aangeduid als '2x per maand'. Daarbij moet worden aangetekend dat dit geen exacte aanduiding is (door het wisselende aantal dagen per kalendermaand), maar binnen PIVL kan het worden uitgedrukt op de volgende wijze:

2x per maand

```
<comp xsi:type="PIVL_TS">
  <period value="0.5" unit="mo"/>
</comp>
```

Of als de frequentie wordt aangeduid als '3x per jaar'

```
<comp xsi:type="PIVL_TS">
  <period value="0.3333" unit="a"/>
</comp>
```



Merk op dat omrekenen van maanden of jaren naar dagen of weken niet wenselijk is, omdat dit onhandige waarden zal opleveren (als de exacte omrekenfactoren worden toegepast) of zelfs tot een andere interpretatie zal leiden (als bijvoorbeeld 1 mo = 30 d of 1 mo = 4 wk als factor wordt gehanteerd).

Het is echter wel toegestaan om te rekenen tussen jaren en maanden onderling. Het voorgaande voorbeeld zou dan ook kunnen worden uitgedrukt met de volgende PIVL:

3x per jaar, in maanden

```
<comp xsi:type="PIVL_TS">
  <period value="4" unit="mo"/>
</comp>
```

Hier wordt dus gebruik gemaakt van de omrekenfactor 1 mo = 1 a/12. Feit is dat dit een minder 'harde' omrekenfactor is dan degene tussen seconden, minuten, uren en weken.

Dit levert de volgende tabel op met omrekeningen die moeten worden ondersteund:

	unit	s	min	h	d	wk	mo	a
unit								
s		X	X	+	+	+		
min		X	X	X	+	+		
h		+	X	X	X	+		
d		+	+	X	X	X		
wk		+	+	+	X	X		
mo							X	+
a							+	X

De "X" in deze tabel geven omrekeningen aan die door alle ontvangende systemen moeten worden ondersteund. De "+" geven omrekeningen aan die bij voorkeur moeten worden ondersteund. De niet gevulde cellen duiden op omrekeningen (van verzonden informatie naar opgeslagen interpretatie of andersom) die niet mogen plaatsvinden.



Vanaf de volgende release van de HL7v3 Data Type specification wordt het ook mogelijk om naast <period> een element <frequency> te gebruiken in PIVL. Op dat moment kunnen bovenstaande voorbeelden exact conform de intentie worden doorgegeven (waardoor bijvoorbeeld het verschil tussen 3x per dag en 1x per 8 uur expliciet gemaakt kan worden).

ALGEMENE VOORBEELDEN

Om het gecombineerde gebruik van *phase* en *period* toe te lichten, worden hieronder nog enige voorbeelden gegeven waarbij er sprake is van een periodieke activiteit (bijvoorbeeld een bepaalde medicatietoediening), die een vast ijkpunt en/of een vaste tijdsduur heeft.

Meestal zal dit dan voorkomen in combinatie met een SXP component die het interval aangeeft waarbinnen de activiteit plaatsvindt. De volledige *effectiveTime* wordt dan:

XML-voorbeelden

3x per dag, telkens om 06:00, 14:00 en 22:00,
gedurende 30 min., vanaf 2 sept. 2005 om 14:00

```
<effectiveTime xsi:type="SXP_TS">
  <comp xsi:type="IVL_TS">
    <low value="200509021400"/>
  </comp>
  <comp xsi:type="PIVL_TS" operator="A">
    <phase>
      <low value="200509022200"/>
      <width value="30" unit="min"/>
    </phase>
    <period value="0.3333" unit="d"/>
  </comp>
</effectiveTime>
```

De PIVL component krijgt hier de operator "A", omdat de doorsnede genomen moet worden met het interval dat daarboven al in de IVL_TS is aangeduid ('vanaf 2-9-2005'). Merk op dat het feitelijke tijdstip in de *phase* component van de PIVL niet relevant is, d.w.z. het doet alleen dienst als ijkpunt voor de *period*, maar is verder willekeurig. Er staat op deze manier '3x per dag, gedurende 30 min.', maar daarbij wordt wel aangegeven dat deze herhaling steeds op 14:00, 22:00 en 06:00 moet vallen (aangezien 22:00 als ijkpunt voor de herhaling wordt gegeven). In combinatie met het interval erboven, levert dat de volgende momenten op waarop de activiteit moet plaatsvinden:

2 sept. 2005 om 14:00 (ook al ligt dit vóór het ijkpunt)

2 sept. 2005 om 22:00 (dit is 'toevallig' het ijkpunt)

3 sept. 2005 om 06:00

.... (en zo verder om de 8 uur)

Merk op dat als de duur van de activiteit niet relevant (of niet bekend) was geweest, de component *width* van de *phase* gewoon afwezig had kunnen blijven. Als zelfs de begintijd niet relevant (of niet bekend) was geweest, dan was de hele *phase* niet nodig geweest. Dit zou de onderstaande PIVL componenten in het eerdergenoemde voorbeeld opleveren:

3x per dag, telkens om 06:00, 14:00 en 22:00

```
<comp xsi:type="PIVL_TS" operator="A">
  <phase>
    <low value="200509022200"/>
  </phase>
  <period value="0.3333" unit="d"/>
</comp>
```

3x per dag


```
<comp xsi:type="PIVL_TS" operator="A">
  <period value="0.3333" unit="d"/>
</comp>
```

Vervolgens een vergelijkbaar voorbeeld met een herhaling op twee vaste weekdays:
Elke ma en vr, om 13:00, gedurende 4 uur, tijdens sept. 2005

```
<effectiveTime xsi:type="SXPRTS">
  <comp xsi:type="IVL_TS">
    <low value="20050901000000"/>
    <high value="20050930235959"/>
  </comp>
  <comp xsi:type="SXPRTS" operator="A">
    <comp xsi:type="PIVL_TS" alignment="DW">
      <phase>
        <low value="200508291300"/>
        <width value="4" unit="h"/>
      </phase>
      <period value="1" unit="wk"/>
    </comp>
    <comp xsi:type="PIVL_TS" alignment="DW" operator="I">
      <phase>
        <low value="200508291300"/>
        <width value="4" unit="h"/>
      </phase>
      <period value="1" unit="wk"/>
    </comp>
  </comp>
</effectiveTime>
```

Merk op dat hier een interval (de maand september in 2005) wordt doorsneden (operator="A") door een geneste samenvoeging/vereniging (operator="I") van twee perioden met datatype PIVL_TS.

Merk tevens op dat het basisinterval van de eerste PIVL (voor de herhaling op maandag) niet eens binnen het interval ligt waarmee doorsneden wordt (29-8 wordt namelijk als ijkpunt gebruikt). De relevante aspecten van het ijkpunt zijn alleen de weekday (door de combinatie met de *alignment* op "DW") en het begintijdstip (vanaf 13:00). Met andere woorden: de PIVL beschrijft in feite **elke maandag tussen 13:00 en 17:00**, van het oneindige verleden tot in de oneindige toekomst. De doorsnede zorgt voor de inperking.

Al met al levert dit de volgende lijst met door deze structuur gedefinieerde momenten:

2 sept. 2005 om 13:00 (de eerste vrijdag van september 2005)
 5 sept. 2005 om 13:00 (de eerste maandag van september 2005)
 9 sept. 2005 om 13:00 (de tweede vrijdag van september 2005)

 26 sept. 2005 om 13:00 (de laatste maandag van september 2005)
 30 sept. 2005 om 13:00 (de laatste vrijdag van september 2005)

Tenslotte een voorbeeld waarin een bepaald interval periodiek wordt uitgesloten:

Pilschema: 21 dagen wel, 7 dagen niet, 1x per dag om 09:00

```
<effectiveTime xsi:type="SXPRTS">
  <comp xsi:type="IVL_TS">
```

```

    <low value="20050901000000"/>
    <high value="20051130235959"/>
  </comp>
  <comp xsi:type="SXPRTS" operator="A">
    <comp xsi:type="PIVLTS">
      <phase>
        <low value="200509010900"/>
      </phase>
      <period value="1" unit="d"/>
    </comp>
    <comp xsi:type="PIVLTS" operator="E">
      <phase>
        <low value="20050922"/>
        <width value="7" unit="d"/>
      </phase>
      <period value="28" unit="d"/>
    </comp>
  </comp>
</effectiveTime>

```

Ook hier worden twee PIVL componenten gedefinieerd, waarbij het verschil tussen de eerste en de tweede component wordt bepaald (de tweede wordt als het ware afgetrokken van de eerste). Gezamenlijk wordt deze subset weer doorsneden (operator "A") met het interval dat erboven is vastgelegd ('de maanden september t/m november').

Het effect is dat een herhalend patroon ontstaat met de volgende structuur:

3 weken met dagelijks gebruik om 09:00	01-09-2005 t/m 21-09-2005
1 week zonder gebruik	22-09-2005 t/m 28-09-2005
3 weken met dagelijks gebruik om 09:00	29-09-2005 t/m 20-10-2005
1 week zonder gebruik	21-10-2005 t/m 27-10-2005
3 weken met dagelijks gebruik om 09:00	28-10-2005 t/m 17-11-2005
1 week zonder gebruik	18-11-2005 t/m 24-11-2005
6 dagen met dagelijks gebruik om 09:00	25-11-2005 t/m 30-11-2005

De dagelijkse toediening wordt bepaald door de *period* in de eerste PIVL. De week zonder toediening wordt bepaald door de tweede PIVL, met een *phase* van 7 dagen, herhalend per *period* van 28 dagen, die wordt uitgesloten binnen het patroon van de eerste PIVL.

Merk op dat het wel degelijk relevant is welke datum als begindatum van de *phase* in de tweede PIVL wordt gekozen. Deze moet namelijk zodanig samenhangen met de **begindatum van het interval** (niet per se met die van de *phase* in de eerste PIVL!) dat de periode zonder gebruik steeds in de 4^e week valt (dus na een periode van 3 weken met gebruik).

7 CMET's

7	CMET's	115
7.1	Inleiding	116
7.2	Algemene beschrijving van CMET varianten	116
7.3	R_PatientNL (patiënt)	118
7.3.1	Functie	118
7.3.2	R_PatientNL [universal]	119
7.3.3	R_PatientNL [identified]	127
7.4	E_PersonNL (persoon)	128
7.4.1	Functie	128
7.4.2	E_PersonNL [universal]	129
7.4.3	E_Person [contact]	148
7.5	R_AssignedPerson (persoon in een rol)	151
7.5.1	R_AssignedPersonNL [universal]	151
7.5.2	R_AssignedPerson [identified/confirmable]	153
7.5.3	R_AssignedPerson [identified]	154
7.6	R_AssignedOrganization (organisatie)	155
7.6.1	R_AssignedOrganization [universal]	155
7.6.2	R_AssignedOrganization [identified/confirmable]	156
7.6.3	R_AssignedOrganization [identified]	157
7.7	R_AssignedDevice (apparaat)	158
7.7.1	R_AssignedDevice [universal]	158
7.8	R_AssignedEntity (entiteit in een rol)	160
7.8.1	R_AssignedEntity [universal]	161
7.8.2	R_AssignedEntity [identified/confirmable]	162
7.8.3	R_AssignedEntity [identified]	162
7.9	E_Organization (organisatie)	164
7.9.1	E_Organization [universal]	164
7.9.2	E_Organization [contact]	165
7.9.3	E_Organization [identified/confirmable]	166
7.9.4	E_Organization [identified]	167
7.10	E_Place (plaats)	168
7.10.1	E_Place [universal]	168
7.11	R_LocationLocatedEntity (locatie)	169
7.11.1	R_LocationLocatedEntity [universal]	169
7.11.2	R_LocationLocatedEntity [identified]	169
7.12	R_ResponsibleParty (verantwoordelijke partij)	171
7.12.1	R_ResponsibleParty [universal]	171
7.12.2	R_NotificationParty [contact]	173

7.1 Inleiding

Common Message Element Types (CMET's) zijn herbruikbare informatiemodellen (en daaruit afgeleide berichtfragmenten), die binnen HL7 versie 3 worden gebruikt om tussen interacties onderling een (interne) vorm van standaardisatie te bereiken. Door immers in alle interacties dezelfde specificaties te gebruiken voor bepaalde terugkerende onderdelen, zoals de patiënt, de zorgverlener en de zorgaanbieder, ontstaan de volgende voordelen:

- Er wordt geen tijd verspild aan het praten over steeds dezelfde zaken (efficiëntie).
- Ontwikkeling van CMET's kan aan deskundigen worden overgelaten (specialisatie).
- Software die HL7 V3 interacties genereert of verwerkt kan hiervoor steeds dezelfde basisfunctie gebruiken, ongeacht de berichtcontext waarin de CMET voorkomt.

CMET's zijn in feite R-MIM's, net zoals die voor volledige HL7 berichten gebruikt worden, maar fungeren als **bouwstenen** voor de berichtmodellen waarin ze gebruikt worden. In die zin lijken ze meer op de datatypen die elders in dit document beschreven worden, omdat ze herbruikbare, afgebakende gegevensstructuren beschrijven. Elke CMET heeft z'n eigen **XML-schema**, waaraan wordt gerefereerd vanuit de interactieschema's.

7.2 Algemene beschrijving van CMET varianten

De meeste CMET's komen in meerdere varianten voor. Deze zogenaamde 'flavors' worden toegepast afhankelijk van de informatiebehoefte in een bepaalde context. De voornaamste verschillen zitten in de 'rijkheid' aan gegevens in de CMET.

In het algemeen komen vier verschillende varianten van de meeste CMET's voor:

[Universal]

Deze variant wordt gebruikt tussen zogenaamde 'loosely coupled systems', d.w.z. systemen die geen toegang hebben tot een gemeenschappelijk of gesynchroniseerd stambestand (zoals van patiënten of zorgverleners). In deze versie van de CMET kunnen alle relevante gegevens van het betreffende object worden benoemd. Alle andere CMET-varianten zijn dan ook restricties (deelverzamelingen) op dit universele informatiemodel.

[Identified]

Deze variant wordt gebruikt tussen zogenaamde 'tightly coupled systems', die volledig blind kunnen varen op het uitwisselen van een gemeenschappelijke identificatie voor het betreffende object (op basis van de 'instance identifier'). Binnen één zorgaanbieder is dit vaak voldoende, maar bij gebruik van landelijke identificatiesystemen, zoals AGB-Z of UZI voor zorgverleners en BSN voor patiënten, geldt dit in principe ook op landelijke schaal.

[Identified/confirmable]

Deze variant is een tussenvorm van Universal en Identified, waarbij naast de identifier ook voldoende gegevens beschikbaar zijn om de identifier te kunnen verifiëren. Op deze manier kan de ontvanger de gegevens controleren met die in de eigen database. Wat er gebeurt als er verschillen zijn, hangt sterk af van de context van het bericht.

[Contact]

Deze variant wordt gebruikt in situaties waarbij het vooral belangrijk is om contact met de betreffende persoon of instantie te kunnen opnemen. Dat wil zeggen dat de nadruk ligt op adres- en telecommunicatiegegevens, inclusief die van eventuele contactpersonen.

In Nederland zullen niet alle varianten gebruikt worden. Binnen elk bericht zal worden bekeken welke informatiebehoefte in de betreffende context moet worden vervuld.

7.3 R_PatientNL (patiënt)

7.3.1 Functie

De CMET *R_PatientNL* vormt een standaardberichtelement voor het uitwisselen van **administratieve patiëntgegevens**. Deze worden vaak aangeduid als demografische of Naam Adres Woonplaats-gegevens, hoewel die laatste term niet de hele lading dekt. Ook geboortedatum, geslacht, telefoonnummers etc. maken bijvoorbeeld deel uit van *R_Patient NL*.

De CMET *R_PatientNL* komt (in één van de mogelijke varianten) voor in de meeste HL7 versie 3 berichten, doordat het de universele manier is om de patiënt (die tenslotte meestal centraal staat) te identificeren en/of te beschrijven binnen alle domeinen.

Het is belangrijk om op te merken dat een patiënt in de context van HL7 versie 3 altijd betrekking heeft op een persoon als **ontvanger van zorg**. Dit is een 'rol' die gespeeld wordt in de context van een bepaalde zorgaanbieder. Op die manier wordt er onderscheid gemaakt tussen de persoon (met zijn persoonsgegevens) en zijn rol als patiënt, met patiëntgegevens die in principe kunnen wisselen per zorgaanbieder. In de praktijk worden deze gegevens meestal gezamenlijk verzonden.



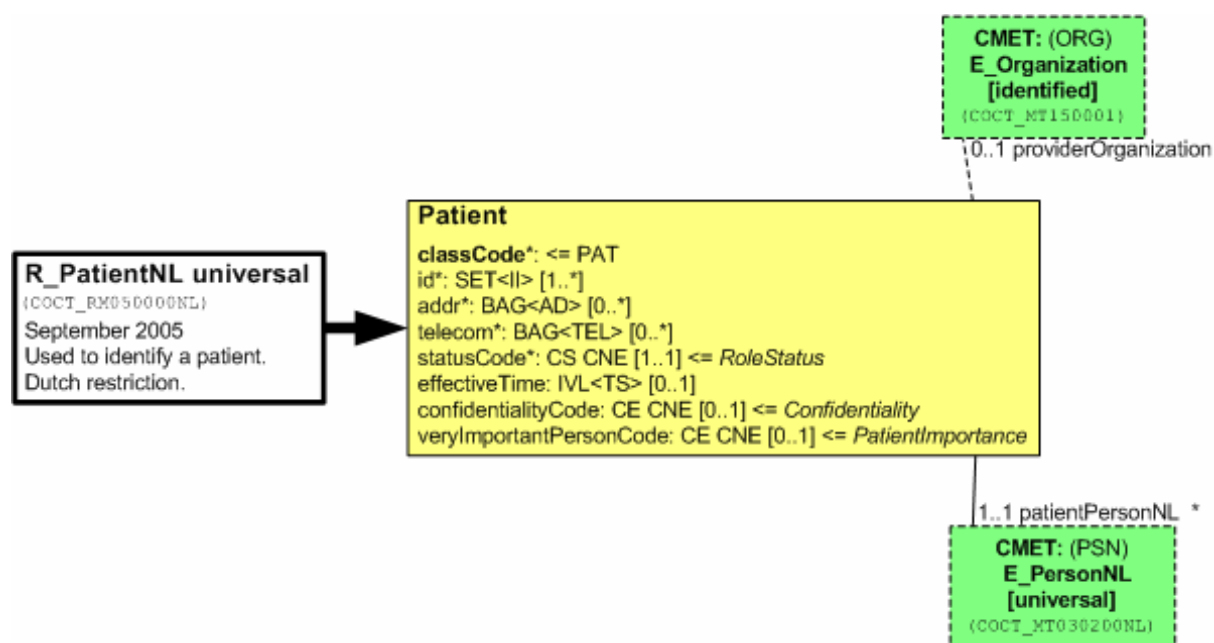
Internationaal worden ook niet-personen als ontvangers van zorg ondersteund, maar in Nederland ondersteunen we dat (nog) niet.

De verschillende 'smaken' van de CMET worden als volgt gebruikt:

- | | |
|--------------------------|--|
| • R_Patient [universal] | doorgeven van algemene patiëntgegevens |
| • R_Patient [identified] | identificatie van de patiënt |

In de volgende secties wordt een beschrijving gegeven van *R_PatientNL* [universal], waarna van de andere varianten de verschillen met de *[universal]* variant worden beschreven.

7.3.2 R_PatientNL [universal]



Figuur 1 COCT_RM050000NL R_Patient NL [universal]

De CMET *R_PatientNL [universal]* bevat de volgende gegevensklassen:

Patient	Patiëntgegevens (in de context van een zorgaanbieder)
CMET E_PersonNL [universal]	Persoonsgegevens (onafhankelijk van de zorgaanbieder) (in het NL model worden niet-personen als patiënt nog uitgesloten)
CMET E_Organization [identified]	Zorgaanbieder waarbij de persoon als patiënt bekend is (in het NL model wordt alleen het instellingsnummer gebruikt)

Belangrijk kenmerk is dus dat een **patiënt** hier altijd een **persoon** is en in de context van een bepaalde organisatie (potentieel) zorg ontvangt. Die organisatie is een zorgaanbieder, waarbij de unieke identifier (*id*) van de patiënt zowel een lokaal als een landelijk nummer (BSN) kan zijn.

Patient	Patiëntgegevens
----------------	------------------------

De klasse *Patient* heeft de volgende kenmerken:

classCode	PAT (Patient) Een persoon in de rol van patiënt (in de context van een zorgaanbieder)
id	Patiëntidentificatie(s)
addr	Woon/verblijfadres(sen)
telecom	Communicatieaanduiding(en)
statusCode	Status van de patiëntregistratie
effectiveTime	Geldigheidsinterval
confidentialityCode	Vertrouwelijkheidsaanduiding
veryImportantPersonCode	VIP aanduiding

De klasse *Patient* heeft de volgende associaties:

1..1 Person	Persoonsgegevens (onafhankelijk van de zorgaanbieder)
0..1 Organization	Zorgaanbieder waarbij de persoon als patiënt bekend is

Voor een toelichting op deze associaties wordt verwezen naar de betreffende CMET's.

Patient.id..... *Patiëntidentificatie(s)*
[SET<II> \[1..*\]](#)

Het element *<Patient><id>* is **verplicht gevuld** binnen *R_Patient* en bevat **één of meer** patiëntnummers, d.w.z. *instance identifiers* die de patiënt onderscheiden van andere (potentiële) zorgontvangers. Deze identificatie kan een nummer zijn dat lokaal binnen een zorgaanbieder is gegenereerd, maar het mag ook een regionaal of landelijk nummer zijn. De gehanteerde identificaties hangen af van de context waarin de CMET wordt toegepast.

De identificatie van een patiënt moet zodanig worden doorgegeven dat deze ondubbelzinnig is voor de ontvangende applicatie. Het datatype *II (Instance Identifier)* garandeert dit door naast de *extension* van de identifier (het feitelijke nummer) ook de *root* door te geven, die het betreffende identificatiesysteem (en dus de combinatie) uniek aanduidt.

Er zijn twee soorten patiëntidentificaties:

- **lokale nummers van de zorgaanbieder**, meestal gegenereerd door het primaire informatiesysteem (het 'XIS') van de betreffende zorgaanbieder;
- **het landelijke burgerservicenummer**, zoals deze elektronisch opvraagbaar is bij de Sectorale Berichten Voorziening voor de Zorg (SBV-Z).

Hieronder wordt voor beide situaties beschreven hoe het element *<id>* gevuld zal worden.



Binnen Nederland wordt bij gebruik tussen instellingen in ieder geval het BSN doorgegeven als patiëntidentificatie, met daarnaast eventueel het eigen (lokale) patiëntnummer.

Lokaal patiëntnummer

Dit is het nummer dat door het patiëntidentificatiesysteem de zorgaanbieder zoals een ZIS van een ziekenhuis, het HIS van een huisarts of het AIS van een apotheek is uitgegeven. Om mogelijk te maken dat elke *id* van een patiënt **universeel uniek** is, wordt in het datatype *II* gebruik gemaakt van een *OID* (*object identifier*). Het verkrijgen van een OID en het samenstellen van een geschikte root voor een identificatiesysteem, staat beschreven in paragraaf 3. Hier onder staan enkele voorbeelden beschreven:

Voorbeeld van een patiëntidentificatiesysteem in een ziekenhuis op basis van AGB-Z:

OID segment	betekenis
2.16.840.1.113883.2.4	HL7 Nederland
.6.1	AGB-Z
.6010756	AGB-Z nummer van het ziekenhuis waarbinnen het nummer is uitgedeeld (zonder de voorloopnul!)
.1	Het ZIS binnen het betreffende ziekenhuis
.1	Patiëntnr. als identificatiesysteem binnen het ZIS

Voorbeeld van een patiëntidentificatiesysteem in een huisartsenpraktijk op basis van UZI-registerabonneenummer (URA):

OID segment	betekenis
2.16.528.1.1007.3	Identificatiesystemen van het UZI-register
.3	URA; zorgaanbieder die UZI passen aanvraagt
.2222	URA van de huisartsenpraktijk (zonder de voorloopnullen!)
.12	Het HIS binnen de betreffende huisartsenpraktijk
.4	Patiëntidentificatiesysteem binnen het HIS



Als bijvoorbeeld een ziekenhuis koppelingen onderhoudt met lokale openbare apotheken, waarbij het lokale patiëntnummer, zoals gebruikt in deze apotheken, wordt bijgehouden in een kruistabel (dus gekoppeld aan de lokale patiëntidentificatie van het ziekenhuis zelf, of aan het BSN), dan moet het ziekenhuis de volledige instance identifier van het patiëntidentificatie van de apotheek gebruiken wanneer het met de apotheek communiceert. Desondanks blijft het ziekenhuis de 'scoper' van de gebruikte patiëntrol.



Met de komst van de UZI-passen en bijbehorende nummers, ontstaat daarmee ook een nieuw systeem voor het identificeren van zorgverleners én zorgaanbieders. Zorgaanbieders kunnen dan namelijk geïdentificeerd worden op basis van het UZI-abonneeregister, waarin alle instanties die UZI-passen hebben aangevraagd worden bijgehouden.

Het bovenstaande principe voor de root OID van lokale patiëntnummers (en andere lokale identificatiesystemen) blijft echter volledig intact, alleen neemt het UZI abonneenummer de functie van het AGB-Z nummer over.

De OID voor UZI abonneenummers is 2.16.528.1.1007.3.3 (dus niet onder de HL7 Nederland tak) en UZI-abonneenummers zelf zijn 8-cijferig. De OID voor lokale patiëntnummers binnen een ziekenhuis kan dan bijvoorbeeld worden: 2.16.528.1.1007.3.3.1234567.1.1 voor een ziekenhuis met als UZI-abonneenummer 01234567 (ook hier worden voorloophullenvallen verwijderd).

In feite bestaan er geen regels voor de opbouw van *root* OID's (de segmenten zijn betekenisloos), zolang maar de garantie bestaat dat de *extension* er een unieke context mee verkrijgt. Bovenstaande methode wordt echter sterk geadviseerd bij het doorgeven van lokale patiëntnummers, om te zorgen voor uniforme toepassing binnen Nederland.

Burgerservicenummer

In alle communicatie tussen een instelling en derde partijen, waaronder maar niet beperkt tot berichten die worden uitgewisseld via de landelijke infrastructuur (AORTA), zal in Nederland gebruik moeten worden gemaakt van het burgerservicenummer als primaire patiëntidentificatie. Voor het systeem van burgerservicenummers is een landelijke *OID* (*object identifier*) vastgelegd, zodat patiëntidentificaties altijd als BSN herkenbaar zijn.

De OID voor burgerservicenummers is: 2.16.840.1.113883.2.4.6.3. Aan deze *root* van de *instance identifier* is een patiëntidentificatie dus altijd herkenbaar als BSN.



Burgerservicenummers zijn altijd 9-cijferig (eventueel inclusief voorloophullenvallen) en hebben op de laatste positie een controlecijfer op basis van een modulo-11 proef. Vanzelfsprekend moet ook dit controlecijfer worden doorgegeven.

XML-voorbeeld 1

Een patiënt is binnen het ziekenhuis met AGB-Z nummer 06010756 bekend onder het intern gegenereerde nummer 120267BA501. De *Patient.id* wordt als volgt doorgegeven:

```
<id extension="120267BA501"
  root="2.16.840.1.113883.2.4.6.1.6010756.1.1"/>
```

Noot 1: Merk op dat de *extension* niet per se numeriek hoeft te zijn, maar ook **andere tekens** mag bevatten. De *extension* moet worden doorgegeven zodanig dat deze betekenisvol ook weer aan het bronstelsel kan worden aangeboden.

Noot 2: Merk op dat de AGB-Z nummerreeks voor ziekenhuizen altijd begint met '06'. De **voorloophul** vervalt altijd binnen de OID van de *root* (algemene regel bij OID gebruik).

XML-voorbeeld 2

Een patiënt is binnen de huisartsenpraktijk met AGB-Z nummer 01042461 bekend onder het intern gegenereerde nummer 018793. De *Patient.id* wordt als volgt doorgegeven:

```
<id extension="018793"
  root="2.16.840.1.113883.2.4.6.1.1042461.1.1"/>
```

Noot 1: Merk op dat eventuele voorloopnullen in de *extension* mogen, en zelfs **moeten**, blijven staan, indien deze voor betekenisvolle verwerking binnen het bronsysteem noodzakelijk is.

XML-voorbeeld 3

Een patiënt heeft burgerservicenummer 012345672.

```
<id extension="012345672"
  root="2.16.840.1.113883.2.4.6.3"/>
```

XML-voorbeeld 4

Een patiënt heeft burgerservicenummer 012345672, maar wordt binnen een ziekenhuis ook nog aangeduid met het intern (binnen het ZIS) gegenereerde nummer 0006756420.

```
<id extension="012345672"
  root="2.16.840.1.113883.2.4.6.3"/>
<id extension="0006756420"
  root="2.16.840.1.113883.2.4.6.1.6010756.1.1"/>
```

Noot 1: Merk op dat de twee herhalingen van het element *<id>* gewoon achter elkaar worden weergegeven. Het BSN is als zodanig voor de ontvanger herkenbaar door de *root* OID.

Patient.addr..... *Woon/verblijfadres(sen)*
BAG<AD> [0..]*

Het element *<Patient><addr>* is **aanwezig indien bekend** en bevat **één of meer** actuele adressen van de persoon in diens rol als patiënt. Dit gaat vrijwel altijd om een (t)huisadres, maar kan ook een vakantiewoning of ander tijdelijk adres betreffen.

Het element is *required*, hetgeen betekent dat alle applicaties die de CMET gebruiken dit element **moeten ondersteunen**. Er moet dus een waarde gestuurd kunnen worden als deze bekend is. Een ontvangend systeem moet in staat zijn de waarden te verwerken.



Men kan zich natuurlijk afvragen waarom een adres een patiëntgegeven is en geen vast persoonsgegeven. De redenatie is dat het adres afhankelijk is van de rol van de persoon als patiënt, omdat bijvoorbeeld een arts een ander adres zal hanteren in zijn rol van patiënt dan in zijn rol van zorgverlener. Ook kan een patiëntadres bijvoorbeeld van tijdelijke aard zijn i.v.m. medische behandeling.

Het gaat dus om de woon/verblijfadressen van de patiënt, zoals door de zorgaanbieder gehanteerd binnen de eigen administratie. Zie verder de beschrijving bij het datatype **'AD'**.

XML-voorbeeld

Een patiënt staat geregistreerd met (woon)adres Dorpsstraat 54, 1024 BB te Purmerend.

```
<addr use="HP">
  <streetName>Dorpsstraat</streetName>
  <houseNumber>54</houseNumber>
  <postalCode>1024 BB</postalCode>
  <city>Purmerend</city>
</addr>
```

Patient.telecom Communicatieaanduiding(en)
[BAG](#) <[TEL](#)> [0..]*

Het element <Patient><telecom> is **aanwezig indien bekend** (required) en bevat **één of meer** actuele telefoonnummers of andere communicatieaanduidingen die gebruikt kunnen worden om de patiënt te bereiken. Hieronder valt bijvoorbeeld telefoonnummer thuis, telefoonnummer op het werk, mobiel nummer of fax, maar ook een **e-mailadres** of **website**!

Het element is *required*, hetgeen betekent dat alle applicaties die de CMET gebruiken dit element moeten ondersteunen. Er moet dus een waarde gestuurd kunnen worden als deze bekend is. Een ontvangend systeem moet in staat zijn de waarden te verwerken.



Binnen het datatype 'TEL' zijn allerlei communicatieaanduidingen mogelijk. Het advies is om binnen *R_Patient* alleen de volgende typen toe te passen:

tel: voor telefoonnummers (inclusief semafoons)
fax: voor faxapparaten
mailto: voor e-mailadressen

Het gaat dus om de communicatieaanduidingen van de patiënt, zoals die door de zorgaanbieder worden gehanteerd binnen de eigen administratie. Zie verder de beschrijving bij het datatype '**TEL**'. De meest voorkomende telefoonnummertypen zijn 'HP' (telefoonnummer thuis), 'MC' (mobiel nummer) en 'WP' (telefoonnummer werk).

XML-voorbeeld

Van een patiënt worden diens telefoonnummer thuis, een mobiel nummer en een e-mailadres doorgegeven. Dit levert drie herhalingen op van het element <Patient><telecom>:

```
<telecom use="HP" value="tel:0299-955555" />
<telecom use="MC" value="tel:06-98686555" />
<telecom value="mailto:j.jansen@speednet.nl" />
```

Patient.statusCode Status van de patiëntregistratie
[CS](#) [1..1] <= RoleStatus

Het element `<Patient><statusCode>` is **verplicht gevuld**, maar dit is feitelijk alleen het geval omdat het in de internationale specificaties ook aanwezig moet zijn. Normaal gesproken zal het altijd de waarde **active** hebben, ten teken dat het een patiëntregistratie betreft die actief is binnen de administratie. Als de patiëntregistratie 'vervallen' is (dat wil zeggen niet meer gebruikt mag worden) dan wordt als `statusCode` **terminated** meegegeven.



Als binnen de patiëntregistratie van een zorgaanbieder blijkt dat dezelfde persoon meerdere keren als patiënt is ingeschreven, dan is het gebruikelijk dat deze nummers aan elkaar 'gekoppeld' (of 'gelijkgesteld') worden. Daarbij wordt één van de nummers preferent en de andere nummers vervallen.

Let op: vervallen nummers kunnen **niet in dezelfde *Patient* klasse** worden meegegeven als het prevalente nummer, omdat het element `<statusCode>` betrekking heeft op de gehele klasse en niet op afzonderlijke `<id>` elementen.

De juiste manier om nonpreferente patiëntidentificaties door te geven is d.m.v. de klasse *OtherIDs* binnen de *E_Person* CMET. Deze klasse wordt echter vooralsnog niet toegepast in de Nederlandse situatie.

XML-voorbeeld

Een actieve patiëntregistratie.

```
<statusCode code="active" />
```

Patient.effectiveTime *Geldigheidsinterval*
IVL<TS> [0..1]

Vooralsnog niet gebruiken in de Nederlandse situatie.

Patient.confidentialityCode *Vertrouwelijkheidsaanduiding*
CE CNE [0..1] <= Confidentiality

Dit **optionele** element heeft betrekking op het vertrouwelijkheidsniveau van de betreffende patiëntgegevens. Vooralsnog wordt daarbij slechts gebruik gemaakt van drie codes uit het code system Confidentiality (2.16.840.1.113883.5.25). Zie onderstaande tabel:

Confidentiality [2.16.840.1.113883.5.25]		
Concept Code	Naam	Definitie
N	normal	Normale vertrouwelijkheidsregels (in de context van de gezondheidszorg) zijn van toepassing. Dat wil zeggen dat alleen geautoriseerde personen met een legitieme medische of administratieve reden toegang mogen hebben.
R	restricted	Beperkte toegang, bijvoorbeeld alleen voor zorgverleners (of daardoor geman-dateerde medewerkers) met een actuele behandelrelatie met de patiënt.
V	very re-stricted	Zeër beperkte toegang, zoals bepaald door de patiënt zelf of diens wettelijke vertegenwoordiger.

De default is 'N', dus als het element niet wordt meegegeven is de toegang 'normaal'.

XML-voorbeeld

De toegang tot de patiëntgegevens is beperkt.

```
<confidentialityCode codeSystem="2.16.840.1.113883.5.25" code="R" />
```

Patient.veryImportantPersonCode *VIP aanduiding*
CE CNE [0..1] <= PatientImportance

Dit **optionele** element heeft betrekking op de aard van de speciale status die de patiënt voor de zorgaanbieder heeft. Daarbij wordt gebruik gemaakt van het binnen HL7 gedefi-nieerde codesysteem PatientImportance (2.16.840.1.113883.5.1075). Zie tabel:

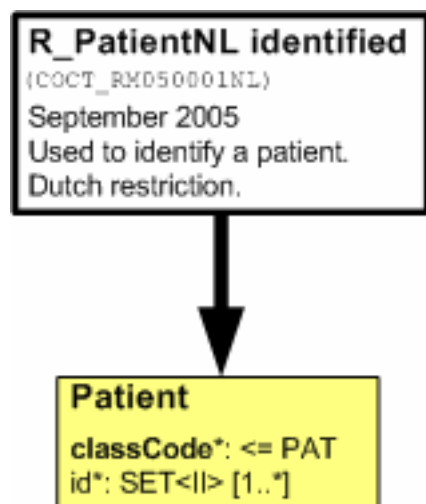
PatientImportance [2.16.840.1.113883.5.1075]		
Concept Code	Naam	Definitie
BM	Board Member	Lid van de raad van bestuur van de zorgaanbieder.
DFM	Physician Family Member	Familielid van een lid van de medische staf.
DR	Staff Physician	Lid van de medische staf.
FD	Financial Donor	Sponsor van de zorgaanbieder.
FOR	Foreign Dignitary	Buitenlandse hoogwaardigheidsbekleder.
GOVT	Government Dignitary	Hoogwaardigheidsbekleder.
SFM	Staff Family Member	Familielid van een medewerker.
STF	Staff Member	Medewerker van de zorgaanbieder.
VIP	Very Important Person	Andere VIP.

XML-voorbeeld

De patiënt is een lid van de medische staf.

```
<veryImportantPersonCode code="DR"
                           codeSystem="2.16.840.1.113883.5.1075" />
```

7.3.3 R_PatientNL [identified]



De *[identified]* variant van de CMET *R_Patient* is een restrictie op de *[universal]* variant, waarin alleen de unieke patiëntidentificatie(s) zijn opgenomen in het informatiemodel. Dit wordt alleen gebruikt om een bij de ontvanger reeds bekende patiënt te identificeren, maar niet om wijzigingen door te geven. Vandaar ook het ontbreken van de statusCode.

De CMET *R_Patient* [identified] bevat de volgende gegevensklassen:

<i>Patient</i>	<i>Patiëntidentificatie</i>
----------------	-----------------------------

De klasse *Patient* heeft de volgende kenmerken:

classCode	PAT (Patient) Een persoon in de rol van patiënt
id	Patiëntidentificatie(s)

Patient.id *Patiëntidentificatie(s)*
SET<II> [1..*]

Zie *Patient.id* bij *R_PatientNL* [universal].

7.4 E_PersonNL (persoon)

7.4.1 Functie

De CMET *E_Person* vormt een standaardberichtelement voor het uitwisselen van **persoonsgegevens** zoals naam, geboortedatum en geslacht. Deze CMET wordt vooral gebruikt in combinatie met *R_Patient* en *R_AssignedPerson*, waarbinnen *E_Person* een genest onderdeel is voor de vaste persoonsgegevens van de patiënt respectievelijk zorgverlener.



Let op dat in het HL7 versie 3 RIM (*Reference Information Model*), en dus ook in alle daarop gebaseerde informatiemodellen, **het adres en bijvoorbeeld de telefoonnummers worden beschouwd als kenmerken van de rol die een persoon vervult**, bijvoorbeeld de rol van patiënt of zorgverlener. Deze kenmerken komen dan ook niet voor in het model van de *E_Person* CMET.

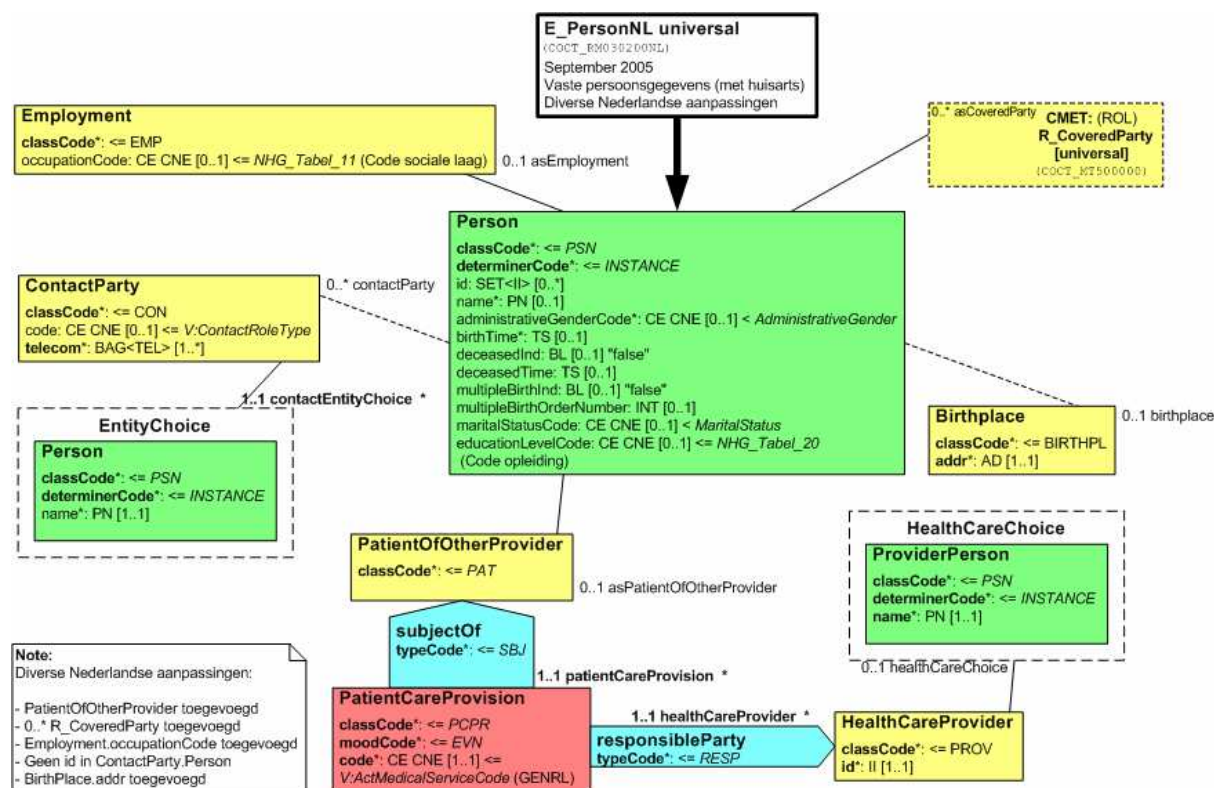
De CMET *E_Person*, althans in de *[universal]* variant (zie hieronder), bevat de meest essentiële persoonsgegevens, aangevuld met de mogelijkheid om diverse andere persoonsgebonden gegevens door te geven (zoals de huisarts).



Merk op dat het doorgeven van persoonsgegevens (d.m.v. *E_Person*) binnen een bepaald bericht (bijvoorbeeld een Voorschriftbericht voor medicatie) geen surrogaat mag zijn voor een on-line patiëntkoppeling (zoals vaak met ADT-berichten binnen ziekenhuizen wordt gebruikt).

De essentiële gegevens, te weten naam, geboortedatum en geslacht, hangen nauw samen met het landelijk BSN-register, waarin deze gegevens ook gebaseerd op de Gemeentelijke Basis Administratie (GBA) beschikbaar zijn. De CMET dwingt echter niet af dat de meegestuurde gegevens per se consistent zijn met de GBA-gegevens bij het BSN.

7.4.2 E_PersonNL [universal]



De CMET *E_PersonNL [universal]* bevat de volgende gegevensklassen:

Person	Persoon
Employment	Beroep(en)
ContactParty	Contactperso(n)en
PatientOfOtherProvider	Relatie met de huisarts
Birthplace	Geboorteplaats
CMET R_CoveredParty	Zorgverzekering(en)



Alhoewel de verzekeringen via de persoon kunnen worden doorgegeven, is de wijze waarop in Nederland nooit beschreven. Internationaal is intussen afgestemd om de verzekeringen via de patiëntrol door te geven en niet als de persoonskenmerk. De modellering voor het doorgeven van verzekeringsgegevens is ook gewijzigd.

Doorgeven van verzekeringsgegevens via de patiënt zal in in één van de volgende versies van deze gids worden meegenomen. In deze versie van de gids zal het doorgeven van verzekeringen niet worden uitgewerkt.

De meeste van de gegevens in het bovenstaande model zijn optioneel, aangezien de CMET ook gebruikt moet kunnen worden om simpelweg de naam, de geboortedatum en

eventueel het geslacht door te geven. In de uitgebreide vorm ondersteunt de CMET dus ook de huisarts, contactpersonen en zelfs beroepsaanduidingen van de persoon.

Kort samengevat kent het berichtelement de volgende kenmerken:

- Een persoon heeft een optionele identificatie (BSN als persoonsnummer) en optioneel aanvullende identificaties
- De onderdelen naam, geslacht en geboortedatum zijn optioneel, maar wel required.
- Verder zijn er optionele modelonderdelen met betrekking tot de geboorteplaats, overlijden, meerlingen, burgerlijke staat en opleiding.
- Een persoon heeft optioneel een zorgrelatie met één (vaste) huisarts.
- Een persoon heeft optioneel één of meer contactpersonen.
- Een persoon heeft optioneel één of meer beroepsmatige rollen.



Er zijn in dit uitzonderlijke geval diverse aanpassingen en uitbreidingen gemaakt op het internationale model, aangezien dit op bepaalde punten tekort schoot voor de Nederlandse situatie. Er is een proces in gang gezet om deze Nederlandse wensen in het internationale model terug te laten komen, aangezien uitgangspunt is dat altijd harmonisatie plaatsvindt.

De aanpassingen zijn:

- De klasse **PatientOfOtherProvider** is toegevoegd om de (vaste) huisarts van de persoon door te geven.
- Het element **Employment.occupationCode** is toegevoegd, omdat dit beter aansloot bij de gehanteerde beroepscodes bij huisartsen.
- Het element **Birthplace.addr** is toegevoegd, omdat anders niet het geboorteland en/of de geboorteplaats kunnen worden weergegeven.

7.4.2.1 Klasse Person (Persoon)

```

Person
classCode*: <= PSN
determinerCode*: <= INSTANCE
id: SET<II> [0..*]
name*: PN [0..1]
administrativeGenderCode*: CE CNE [0..1] < AdministrativeGender
birthTime*: TS [0..1]
deceasedInd: BL [0..1] "false"
deceasedTime: TS [0..1]
multipleBirthInd: BL [0..1] "false"
multipleBirthOrderNumber: INT [0..1]
maritalStatusCode: CE CNE [0..1] < MaritalStatus
educationLevelCode: CE CNE [0..1] <= NHG_Tabel_20
(Code opleiding)
  
```

De klasse *Person* heeft de volgende kenmerken:

classCode	PSN (Person) Een persoon (mens)
------------------	---

determinerCode	INSTANCE Een specifiek persoon (individu)
id	Persoonsnummer
name	Naam
administrativeGenderCode	Geslacht
birthTime	Geboortedatum (en evt. -tijd)
deceasedInd	Overlijdensindicatie
deceasedTime	Overlijdensdatum (en evt. -tijd)
multipleBirthInd	Meerlingindicatie
multipleBirthOrderNumber	Meerlingvolgnummer
maritalStatusCode	Burgerlijke staat
educationLevelCode	Opleidingsniveau

De klasse *Person* heeft de volgende associaties:

0..1	Employment	Beroep
0..*	ContactParty	Contactpers(o)n(en)
0..1	PatientOfOtherProvider	Relatie met de huisarts
0..1	Birthplace	Geboorteplaats

Deze associaties worden verderop in de zogenaamde 'walkthrough' nader toegelicht.

Person.id *Persoonsnummer*
SET<II> [0..]*

Het element *<id>* is binnen de CMET *E_Person [universal]* **optioneel en herhalend**.

Het kan worden gebruikt om het burgerservicenummer (BSN), dat in de Nederlandse zorg fungeert als persoonsidentificatie, uit te wisselen tussen systemen. Meestal zal het BSN dan echter ook als patiëntidentificatie gehanteerd worden (zie CMET *R_Patient*) en is het hier dus redundant. Het element is herhalend beschikbaar omdat ook andere persoonsidentificaties, zoals het paspoort- of rijbewijsnummer, gehanteerd zouden kunnen worden.

Merk op dat persoonidentificaties op deze plaats geen andere kenmerken kunnen krijgen, zoals een geldigheidsperiode voor een identiteitsbewijs.



Het BSN is een persoonsidentificatie (dat zelfs breder dan alleen in de zorg-sector wordt toegepast), maar zal in de meeste gevallen ook fungeren als patiëntidentificatie binnen zorgaanbieders.

De OID voor burgerservicenummers is: 2.16.840.1.113883.2.4.6.3. Aan de *root* van de *instance identifier* is een persoonsnummer dus altijd herkenbaar als zijnde een BSN.

XML-voorbeeld

Een persoon heeft burgerservicenummer 123456782.

```
<id extension="123456782" root="2.16.840.1.113883.2.4.6.3"/>
```

Person.name..... *Naam PN [0..1]*

Het element `<name>` is binnen de CMET *E_Person [universal]* een **optioneel** gegeven, dat echter wel de conformance-indicatie **required** heeft.

Het datatype is **PN** (*Person Name*) dat uitgebreid wordt beschreven in §6.22, waar ook voorbeelden te vinden zijn.

Op welke manier de naam wordt meegegeven binnen de CMET hangt sterk af van de context van het betreffende bericht en de mogelijkheden binnen het bronsysteem. Het zou kunnen dat het systeem slechts in staat is om één enkele string als naam door te geven, maar het is ook mogelijk dat de naam volledig opgesplitst in onderdelen beschikbaar is (inclusief eventuele adellijke, ambtelijke of academische titels en mogelijk ook een aanhef).



Let op dat het bij het gebruik van een partnernaam niet relevant is dat iemand getrouwd is, maar vooral of de betreffende persoon al dan niet met de naam van de partner wil worden aangesproken of aangeschreven.

Het toepassen van de partnernaam in de persoonsnaam staat los van het eventueel vastleggen van de partner als contactpersoon.

In de huidige Nederlandse wetgeving kan bij de huwelijksaangifte elke combinatie van eigen (geslachts)naam en/of naam van de partner gevoerd worden door beide betrokkenen. Het geslacht van de betrokkene(n) heeft op de tenaamstelling geen invloed.

Het BSN-register, zoals dit raadpleegbaar is via de SBV-Z, bevat alleen geslachtsnamen en kent geen partnernamen.

XML-voorbeeld

Voor reguliere voorbeelden, zie de beschrijving bij het datatype PN (Person Name). Indien iemand als geslachtsnaam Jansen heeft, maar sinds een huwelijk ook de naam van diens partner voert (Pietersen), ontstaat er een verschil tussen de naam in de SBV-Z en de reguliere naam waarmee de persoon wordt aangesproken (of aangeschreven). Er kan echter slechts één van beiden namen in een bericht worden meegenomen. Welke naam dit betreft zal afhangen van de afspraken voor een bepaald type communicatie.

```
<name use="L">  
  <family qualifier="SP">Pietersen</family>  
  <delimiter>-</delimiter>
```

```

    <family qualifier="BR">Jansen</family>
  </name>
  <name use="OR">
    <family qualifier="BR">Jansen</family>
  </name>

```

In bovenstaand voorbeeld zijn eventuele andere naamdelen niet opgenomen.

Person.administrativeGenderCode Geslacht
CE CNE [0..1] <= AdministrativeGender

Het element `<administrativeGenderCode>` is binnen de CMET *E_Person [universal]* een **optioneel** gegeven, dat echter wel de conformance-indicatie **required** heeft.

Voor dit gegeven **moeten** de waarden worden gebruikt uit de volgende HL7-tabel:

AdministrativeGender OID: 2.16.840.1.113883.5.1		
code	displayName HL7	Nederlandse omschrijving
F	Female	Vrouwelijk
M	Male	Mannelijk
UN	Undifferentiated	Onbepaald



De waarde "UN" moet niet gebruikt worden om aan te geven dat (nog) geen waarde voor het geslacht van een persoon bekend is. Als dat zo is, dan blijft het gegeven leeg en krijgt het een zogenaamde *nullFlavor* (zie bij het datatype ANY). De waarde "UN" wordt alleen gebruikt als het geslacht wel bekend is, maar **noch zuiver mannelijk noch zuiver vrouwelijk** is.

Merk overigens op dat het hier gaat om het "administratief geslacht", **zoals door de persoon gemeld**, welke niet noodzakelijk door een klinische geslachtbepaling tot stand is gekomen.

XML-voorbeeld

Een vrouw.

```

<code code="F"
  codeSystem="2.16.840.1.113883.5.1"
  displayName="Vrouwelijk" />

```

Noot: het attribuut `@displayName` is optioneel en mag niet door de ontvanger gebruikt worden bij verwerking.

Person.birthTime Geboortedatum (en evt. -tijd)
IS [0..1]

Het element *<birthTime>* is binnen de CMET *E_Person [universal]* een **optioneel** gegeven, dat echter wel de conformance-indicatie **required** heeft.

Het datatype **TS** staat toe om tijdstippen met verschillende nauwkeurigheidsgraden uit te wisselen, afhankelijk van de mate van precisie waarmee deze bekend is in een systeem. Normaal gesproken zal het bij het element *<Person><birthTime>* een geboortedatum (dat wil zeggen jaar, maand, en dag) betreffen, maar er mag ook een geboortetijd toegevoegd worden.



Het komt regelmatig voor dat van personen wel het geboortjaar maar geen exacte geboortedatum bekend is. Deze worden vaak geregistreerd onder de datum 1 januari van een bepaald jaar, maar binnen HL7v3 zou een datum met een lagere precisie kunnen worden gebruikt om alleen het geboortjaar door te geven (en duidelijk te maken dat de exacte datum niet bekend is).

Een persoon is geboren op 30 maart 1967.

```
<birthTime value="19670330"/>
```

Een persoon is geboren op 30 maart 1967 om 06:30.

```
<birthTime value="196703300630"/>
```

Een persoon is geboren in 1967, maar de exacte datum is onbekend.

```
<birthTime value="1967"/>
```

Person.deceasedInd *Overlijdensindicatie*
BL [0..1]

Het element *<deceasedInd>* is binnen de CMET *E_Person [universal]* **optioneel** aanwezig. Het wordt gebruikt om aan te geven dat de betreffende persoon inmiddels overleden is. Het heeft een default waarde van *false*, hetgeen inhoudt dat het afwezig zijn van het element wordt geïnterpreteerd als aanduiding dat de betreffende persoon 'in leven' is.



Als het aanverwante element *<deceasedTime>* een waarde heeft, dan moet *<deceasedInd>* expliciet als *true* worden doorgegeven.

Zie voor XML-voorbeelden bij *deceasedTime*.

Person.deceasedTime *Overlijdensdatum (en evt. -tijd)*
TS [0..1]

Het element *<deceasedTime>* is binnen de CMET *E_Person [universal]* **optioneel** aanwezig. Het wordt gebruikt om aan te geven wanneer de betreffende persoon overleden is, hetgeen zowel specifiek (inclusief tijd) als ruwweg (bijvoorbeeld alleen een jaartal) mag worden weergegeven.



Het element `<deceasedTime>` heeft alleen betekenis als het aanverwante element `<deceasedInd>` de waarde `true` heeft (d.w.z. als de persoon expliciet als overleden is aangeduid) en moet anders genegeerd worden.

De reden dat beide elementen bestaan, is dat vaak wel het feit van het overlijden bekend is, maar niet de overlijdensdatum (in dat geval blijft `deceasedTime` dus leeg of wordt expliciet een `nullFlavor` meegegeven).

XML-voorbeelden

Een persoon is overleden, maar het is niet relevant wanneer (het verzendende systeem kent het concept niet of verstuurt het nooit).

```
<deceasedInd value="true" />
```

Een persoon is overleden, maar het is niet bekend wanneer (het verzendende systeem kent het concept "overlijdensdatum" wel).

```
<deceasedInd value="true" />
<deceasedTime nullFlavor="UNK" />
```

Een persoon is overleden op 12 januari 2005.

```
<deceasedInd value="true" />
<deceasedTime value="20050112" />
```

Person.multipleBirthInd.....Meerlingindicatie
BL [0..1]

Het element `<multipleBirthInd>` is binnen de CMET *E_Person [universal]* **optioneel**. Het wordt gebruikt om aan te geven dat de betreffende persoon er één van een meerling is. Het heeft een standaardwaarde *false*, hetgeen inhoudt dat het afwezig zijn van het element wordt geïnterpreteerd als: 'de persoon is geen deel van een meerling'.

De meerwaarde van de meerlingindicatie ligt in het feit dat meerlingen een veel hogere kans hebben om als personen verwisseld te worden (door de gelijke geboortedatum, geslachtsnaam en, zeker bij kinderen, ook het adres). De meerlingindicatie werkt dan als signalering dat extra aandacht (of zelfs meer zoekcriteria) moeten worden toegepast.



Dit gegeven moet niet gebruikt worden om aan te geven dat sprake is van een meerling tijdens de zwangerschap (dus niet als kenmerk van de moeder, maar ook **niet als kenmerk van de foetus!**). Het heeft pas betekenis als **kenmerk van een pasgeborene** als sprake is geweest van meervoudige geboortes bij een bevalling (na 16 weken zwangerschap).



Als het aanverwante element `<multipleBirthOrderNumber>` een waarde heeft, dan moet `<multipleBirthInd>` als *true* worden doorgegeven.

Zie voor XML-voorbeelden bij `<multipleBirthOrderNumber>`.

Person.multipleBirthOrderNumber *Meerlingvolgnummer*
INT [0..1]

Het element `<multipleBirthOrderNumber>` is binnen de CMET *E_Person [universal]* **optioneel** aanwezig. Het wordt gebruikt om aan te geven als hoeveelste van de meerling de betreffende persoon geboren werd. De waarde is dus een geheel getal ≥ 1 .



Het element `<multipleBirthOrderNumber>` heeft alleen betekenis als het aanverwante element `<multipleBirthInd>` de waarde *true* heeft (d.w.z. als de persoon expliciet als meerling is aangeduid).

De reden dat beide elementen bestaan, is dat meestal wel bekend is dat de persoon een meerling is, maar niet in welke volgorde de geboortes plaatsvonden (in dat geval blijft *multipleBirthOrderNumber* dus leeg of wordt expliciet een *nullFlavor* meegegeven).

Het element `<multipleBirthOrderNumber>` is zeker niet bedoeld als medisch gegeven, maar kan evt. worden gebruikt als extra methode om meerlingen van elkaar te onderscheiden.



Er zijn geen harde regels voor de toegepaste volgnummers, zolang ze in relatie tot elkaar maar een volgorde aanduiden voor de meerlinggeboorte waarbij het laagste nummer de eerstgeborene voorstelt.. De gebruikte waarden mogen dus ook bijvoorbeeld 100 en 200 zijn.

XML-voorbeelden

Een persoon is (lid van een) meerling, maar de geboortevolgorde is niet relevant (het verzendende systeem kent het concept niet of verstuurt het nooit).

```
<multipleBirthInd value="true" />
```

Een persoon is (lid van een) drieling, maar de geboortevolgorde is niet bekend (het verzendende systeem kent het concept "meerlingvolgnummer" wel).

```
<multipleBirthInd value="true" />  
<multipleBirthOrderNumber nullFlavor="UNK" />
```

Een persoon is (lid van een) tweeling en is als 2e van het stel geboren.

```
<multipleBirthInd value="true" />  
<multipleBirthOrderNumber value="2" />
```


Person.maritalStatusCode..... Burgerlijke staat
CE CNE [0..1] <= MaritalStatus

Het element `<maritalStatusCode>` is binnen de CMET *E_Person [universal]* **optioneel**.

De burgerlijke staat van een persoon wordt bij de meeste zorgaanbieders niet vastgelegd (en dus ook niet uitgewisseld), maar is een gegeven dat binnen het huisartsdossier en ook in de geestelijke gezondheidszorg wel relevant is.

Voor dit gegeven **moeten** de waarden worden gebruikt uit de volgende HL7 tabel:

MaritalStatus OID: 2.16.840.1.113883.5.2		
code	displayName HL7	Nederlandse omschrijving
D	Divorced	Gescheiden (huwelijk of geregistreerd partnerschap ontbonden)
M	Married	Gehuwd
S	Never Married	Ongehuwd en geen geregistreerd partnerschap
T	Domestic partner	Geregistreerd partnerschap
W	Widowed	Weduwe/weduwenaar (zowel van toepassing op huwelijk als geregistreerd partnerschap)



Belangrijke opmerkingen:

1. Het gaat bij bovenstaande codes om de meest recente toestand (bij 'hertrouwd' gaat 'getrouwd' boven 'gescheiden' of 'weduwe'). De persoon bepaalt altijd zelf welke code van toepassing is.
2. De code 'S' is in de internationale standaard zuiver voorbehouden voor 'nooit gehuwd geweest', maar aangezien het een illusie is om aan te nemen dat een persoon een scheiding altijd zal doorgeven, is deze definitie voor Nederland iets afgezwakt tot 'niet gehuwd' in het algemeen.
3. Er is internationaal geen specifieke code voor 'geregistreerd partnerschap' (Domestic partner heeft de ruimere betekenis van 'samenwonen'). Voor Nederland is de definitie ingeperkt, omdat het feit dat iemand samenwoont los staat van de burgerlijke staat.

De tabel die momenteel meestal gebruikt wordt in de huisartssector is NHG tabel 6:

NHG tabel 6: Code burgerlijke staat NIET GEBRUIKEN IN HL7 BERICHTEN			
Code	Omschrijving	Toelichting	HL7 code
01	Gehuwd		M
02	Gescheiden		D
03	Ongehuwd		S
04	Samenwonend	bij geregistreerd partnerschap in andere gevallen	T S (of D of W)
06	Weduwstaat		W
07	LAT-relatie	bij geregistreerd partnerschap bij huwelijk in andere gevallen	T M S (of D of W)
08	Hertrouwd	zelfde als gehuwd	M
90	Onbekend		nullFlavor = "UNK"

99	Overig	geen van bovenstaande categorieën	nullFlavor = "OTH"
----	--------	-----------------------------------	--------------------

Deze tabel mag niet worden gebruikt binnen HL7v3-interacties. De codes voor samenwonen en LAT-relatie hebben geen betrekking op de burgerlijke staat, maar op het woonverband. Het onderscheid tussen hertrouwd en gehuwd is niet relevant in deze context.



Een systeem dat intern gebruik maakt van de NHG (of andere) codes (en dat zullen de meeste zijn), moet in staat zijn om een mapping te maken naar de HL7 *MaritalStatus* tabel om uitwisselbaarheid te garanderen.

XML-voorbeelden

Een persoon heeft zichzelf aangeduid als ongehuwd.

```
<maritalStatusCode code="S"
  codeSystem="2.16.840.1.113883.5.2"
  displayName="Ongehuwd" />
```

Een persoon heeft binnen een huisartssysteem de NHG tabel 6 code '08' (hertrouwd).

```
<maritalStatusCode code="M"
  codeSystem="2.16.840.1.113883.5.2"
  displayName="Gehuwd"/>
```

Een persoon heeft binnen een huisartssysteem de NHG tabel 6 code '99' (overig).

```
<maritalStatusCode nullFlavor="OTH"/>
```

Person.educationLevelCode..... Opleidingsniveau
CE CNE [0..1] <= EducationLevel + NHG_tabel_20

Het element `<educationLevelCode>` is binnen de CMET *E_Person [universal]* **optioneel**.

Het opleidingsniveau van een persoon wordt bij de meeste zorgaanbieders niet vastgelegd (en dus ook niet uitgewisseld), maar is een gegeven dat binnen het huisartsdossier en ook in de jeugd gezondheidszorg of geestelijke gezondheidszorg wel relevant is.

Voor dit gegeven kunnen twee mogelijke tabellen gebruikt worden. HL7 kent als codes:

EducationLevel OID: 2.16.840.1.113883.5.1077		
code	displayName HL7	Nederlandse omschrijving
ASSOC	Associate's or technical degree complete	
BD	College or baccalaureate degree complete	hbo voltooid
ELEM	Elementary School	basisschool (lagere school)
GD	Graduate or professional Degree complete	universiteit voltooid (is dit inclusief hogeschool?)

HS	High School or secondary school degree complete	middelbare school voltooid
PB	Some post-baccalaureate education	universiteit niet voltooid (aansluitend op hbo)
POSTG	Doctoral or post graduate education	post-doctoraal
SCOL	Some College education	hbo niet voltooid
SEC	Some secondary or high school education	middelbare school niet voltooid

Binnen Nederland is echter tabel 20 van de NHG tabellenklapper in gebruik en wordt binnen alle sectoren gebruikt.

NHG tabel 20: Code opleiding OID: 2.16.840.1.113883.2.4.4.30.20			
Code	Omschrijving	Toelichting	HL7 code
01	geen opleiding		nullFlavor = "NA"
02	lager algemeen onderwijs	basisschool (lagere school)	ELEM
03	lager beroepsonderwijs	lbo	HS
04	middelbaar algemeen onderwijs	mavo, vmbo	HS
05	middelbaar beroepsonderwijs	mbo, inclusief meao en mts	
08	hoger algemeen onderwijs	havo, vwo (?)	HS
09	hoger beroepsonderwijs	hbo, inclusief heao en hts	BD
10	wetenschappelijk onderwijs	universiteit/hogeschool	GD (hogeschool?)
90	onbekend	Niet gevraagd Wel gevraagd, maar wist het niet	nullFlavor = "UNK" nullFlavor = "ASKU"
99	overig onderwijs	geen van bovenstaande categorieën	nullFlavor = "OTH"



Merk op dat de codes 90 en 99 in HL7 als *nullFlavor* worden aangeduid. Aangezien het echter niet mogelijk is om een *nullFlavor* naast een code uit een ander codesysteem door te geven, kunnen deze *nullFlavors* alleen gebruikt worden wanneer geen NHG code aanwezig is.

Het is duidelijk dat de (internationale) HL7-tabel sterk op de Verenigde Staten is ingesteld, terwijl de Nederlandse tabel veel meer aansluit bij het systeem met verschillende niveaus van algemeen en beroepsonderwijs (mavo, havo, vwo, etc).



Er zal in internationaal verband nog het volgende worden nagegaan:

- Waarom er geen code is voor 'geen enkele (voltooide) opleiding'.
- Hoe MBO onderwijs het beste in de tabel benoemd kan worden.
- Wat de beste synoniemen zijn voor de codes ASSOC, GD en PB.

Elke combinatie van bovenstaande tabellen mag worden gebruikt bij het doorgeven van het opleidingsniveau. Hoewel de NHG tabel waarschijnlijk de voorkeur zal krijgen i.v.m. bestaande implementaties, is het aan te bevelen om waar mogelijk daarnaast de bijbeho-

rende internationale HL7-code door te geven om zo uniform mogelijk gegevens uit te wisselen.

XML-voorbeelden

Een persoon heeft als hoogste opleidingsniveau de MAVO gedaan.

```
<educationLevelCode
  code="04" codeSystem="2.16.840.1.113883.2.4.4.30.20"
  displayName="middelbaar algemeen voortgezet onderwijs">
  <translation
    code="HS" codeSystem="2.16.840.1.113883.5.1077"
    displayName="High School or secondary school degree complete"/>
</educationLevelCode>
```

Noot: het attribuut *displayName* is optioneel en mag niet gebruikt worden bij verwerking.

Een persoon heeft een opleiding gedaan die niet in de gebruikte codelijst(en) voorkomt.

```
<educationLevelCode nullFlavor="OTH">
  <originalText>Huishoudschool</originalText>
</educationLevelCode>
```

7.4.2.2 Klasse Employment (Beroep)

Employment

classCode*: <= EMP

occupationCode*: CE CNE [1..1] <= NHG_tabel_11 0..1 asEmployment

De associatie *Employment* is **optioneel** aanwezig in de CMET *E_Person*.

In het kader van de uitwisseling van huisartsdossiers komt het ook voor dat gegevens omtrent het beroep (eigenlijk de 'sociale laag') van de persoon worden vastgelegd en uitgewisseld. De beroepen worden daarbij gecodeerd aangeduid.



Er zijn een aantal zaken die internationaal nog zullen worden aangepast met betrekking tot de klasse *Employment* binnen de CMET *E_Person*:

- De naam zou eigenlijk *Employee* moeten zijn, want dat is de rol die de persoon speelt. Daarmee zou de associatie ook *asEmployee* heten.
- Het element *<occupationCode>* komt niet voor in het internationale model. Er is wel een element *<jobCode>*, maar dit is bedoeld voor de interne functie-aanduiding van een werknemer en niet van het beroep.

XML-voorbeeld

```
<person>
  ...
```

```

<asEmployment>
  ...
</asEmployment>
  ...
</person>

```

De klasse *Employment* heeft de volgende kenmerken:

classCode	EMP (Employee) De persoon als werknemer (beroepsbeoefenaar).
occupationCode	Beroepscode

Employment.occupationCode Beroepscode
CE CNE [1..1] <= NHG_tabel_11

Het element *<occupationCode>* is **verplicht gevuld** in elk voorkomen van de klasse *Employment*. Het geeft een gecodeerde aanduiding voor het beroep van de persoon. Omdat de voornaamste toepassing ligt in de huisartsensector, wordt daarbij gebruik gemaakt van de daar gebruikelijke NHG tabel 11, die de volgende codes bevat:

NHG tabel 11: Code sociale laag OID: 2.16.840.1.113883.2.4.4.30.11		
Code	Omschrijving	HL7 code
01	ongeschoolde arbeid	01
02	geschoolde arbeid	02
03	lager employée	03
04	kleine zelfstandige	04
05	middelbaar employée	05
06	hogere beroepen	06
07	zonder beroep	07
09	huishoudelijk werk (gezin)	09
90	onbekend	nullFlavor="UNK"
99	Overig	nullFlavor="OTH"



Merk op dat de codes 90 en 99 in HL7 als nullFlavor worden aangeduid. Aangezien het echter niet mogelijk is om een nullFlavor naast een code uit een ander codesysteem door te geven, kunnen deze nullFlavors alleen gebruikt worden wanneer geen NHG code aanwezig is.



NHG tabel 11 geeft feitelijk geen aanduiding voor het beroep, maar eerder van het 'niveau' waarop de persoon professioneel actief is. Het element *<occupationCode>* benadert echter het meest de toepassing in de huisartsensector.

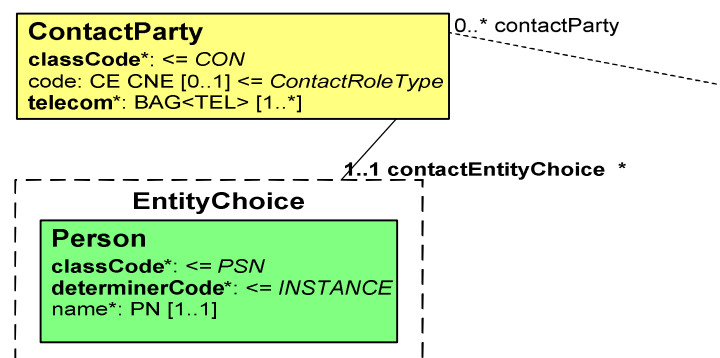
XML-voorbeeld

Een persoon heeft een functie als directeur van een middelgrote onderneming. Dit wordt door de huisarts gekwalificeerd als een 'hogere beroep', met de code '06' in NHG tabel 11.

```
<occupationCode
  code="06" codeSystem="2.16.840.1.113883.2.4.4.30.11"
  displayName="hogere beroep" />
```

Noot: het attribuut *displayName* is optioneel en mag niet gebruikt worden bij verwerking.

7.4.2.3 Klasse ContactParty (Contactpersoon)



De associatie *ContactParty* is **optioneel en herhalend** aanwezig in de CMET *E_PersonNL* [universal].

Het kan voorkomen dat bij een persoon (meestal in diens rol als patiënt) één of meer contactpersonen moeten worden doorgegeven. In sommige gevallen fungeren dergelijke contactpersonen als primair aanspreekpunt voor de persoon (bijvoorbeeld bij een kind, een doof of hardhorend persoon, een buitenlander of geestelijk gehandicapte die niet in staat zijn om zelf met de zorgverleners te communiceren). In andere gevallen betreft het personen die in geval van nood gealarmeerd kunnen worden.

XML-voorbeeld

```
<person>
  ...
  <contactParty>
    ...
    <person>
      ...
    </person>
  </contactParty>
  ...
</person>
```

De klasse *ContactParty* heeft de volgende kenmerken:

classCode

CON (Contact)

Een persoon in diens rol als contactpersoon voor een ander.

code	Contactpersoonstype
telecom	Telecommunicatie-aanduiding(en)

De klasse Person heeft de volgende kenmerken:

classCode	PSN (Persoon) Een persoon.
determinerCode	INSTANCE Een individu.
name	Naam

ContactParty.code *Contactpersoonstype*
[CE CNE \[0..1\]](#) <= *ContactRoleType*

Het element <code> is **optioneel aanwezig** in de klasse *ContactParty*. Het geeft een gecodeerde aanduiding voor het type contactpersoon (internationaal kan het ook een contactorganisatie zijn, maar deze mogelijkheid wordt in Nederland nog uitgesloten).

Hierbij wordt gebruik gemaakt van de HL7 value set *ContactRoleType* (uit het domein *RoleCode*):

RoleCode (value set ContactRoleType) OID: 2.16.840.1.113883.5.111		
code	displayName HL7	Nederlandse omschrijving
ECON	emergency contact	Een contactpersoon voor noodsituaties
NOK	next of kin	Familielid (of andere naaste)

Aangezien slechts één *ContactParty.code* kan worden meegegeven, moet de *ContactParty* rol herhaald worden als een persoon meerdere contactpersoonstypes vervult.



Er ontbreekt in feite nog een code voor een contactpersoon die als 'aanspreekpunt' fungeert voor een 'niet-communicatiebekwaam' persoon, zoals een kind, een doof of hardhorend persoon, een buitenlander of een geestelijk gehandicapte. Nu wordt er meestal automatisch van uitgegaan dat dit een naast familielid is, maar dit is zeker niet per definitie zo. Advies is om in situaties waarbij geen van bovenstaande codes relevant is, het element simpelweg niet mee te geven.



Er kan in deze versie dus wel gespecificeerd worden dat het om familie gaat, maar niet welk type (broer, zus, vader, moeder, etc.) In de nieuwste internationale materialen kan dit ook specifiek worden gemaakt. In de volgende versie van dit document zal dit worden overwogen. Dit zal dan meegaan in een grotere harmonisatie.

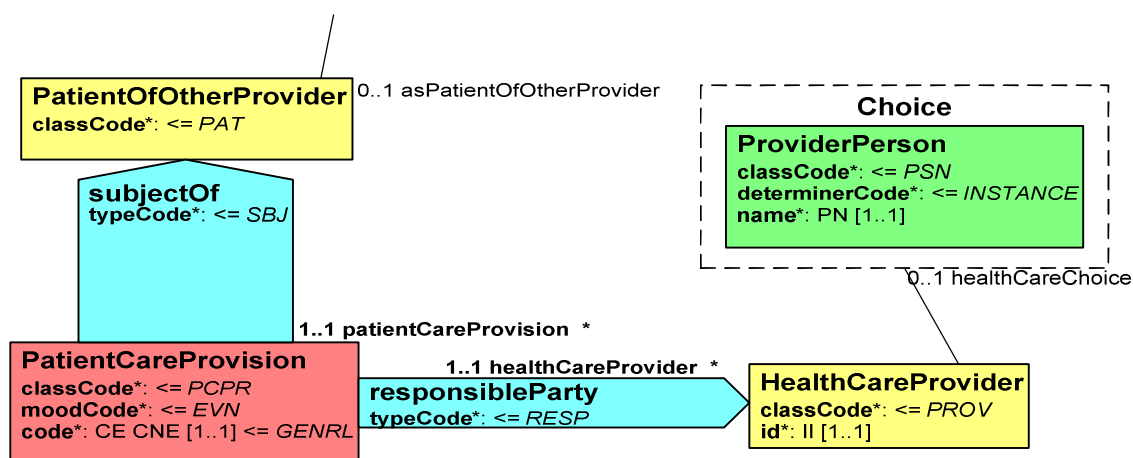
XML-voorbeeld

Een dochter fungeert als contactpersoon voor noodsituaties bij haar opgenomen moeder.

```
<code code="ECON" codeSystem="2.16.840.1.113883.5.111"
      displayName="emergency contact" />
```

Noot: het attribuut *displayName* is optioneel en mag niet gebruikt worden bij verwerking.

7.4.2.4 Klasse PatientOfOtherProvider (Relatie met huisarts/andere zorgverlener)



De relatie tussen een persoon (meestal in de rol van patiënt) en diens huisarts wordt binnen HL7 versie 3 aangeduid op een manier die misschien op het eerste gezicht nogal omslachtig lijkt. De achterliggende gedachte is dat de relatie tussen een persoon en diens huisarts niet principieel anders is dan die tussen een persoon en diens chirurg, afgezien van het feit dat er sprake is van een langdurigere relatie. Het feit dat iemand in Nederland a) verplicht is om een huisarts te hebben en b) maximaal één huisarts tegelijk mag hebben, is internationaal redelijk bijzonder. Zo ontstaat bovenstaand model.

De klasse *PatientOfOtherProvider* heeft de volgende kenmerken:

classCode	PAT (Patient) Patiënt (van een zorgverlener).
------------------	---

De klasse *subjectOf* heeft de volgende kenmerken:

typeCode	SBJ (Subject) De patiënt is het onderwerp van de zorgrelatie met de zorgverlener.
-----------------	---

De klasse *PatientCareProvision* heeft de volgende kenmerken:

classCode	PCPR (Patient Care Provision) Een zorgrelatie tussen een patiënt en een zorgverlener (ook wel een 'verantwoordelijkheidsperiode').
moodCode	EVN (Event) Een feitelijke zorgrelatie.
code	GENRL (ook andere codes zijn toegestaan, zie opmerking later in dit document)

De klasse *responsibleParty* heeft de volgende kenmerken:

typeCode	RESP (Responsible Party) De zorgverlener is de verantwoordelijke voor de zorgrelatie met de patiënt.
-----------------	--

De klasse *HealthCareProvider* heeft de volgende kenmerken:

classCode	PROV (Healthcare Provider) Zorgverlener (persoon of organisatie die gemachtigd is om zorg te verlenen).
id	Identificatie van de zorgverlener

De klasse *ProviderPerson* heeft de volgende kenmerken:

classCode	PSN (Person) De zorgverlener is een persoon.
determinerCode	INSTANCE Het betreft een specifieke zorgverlener (individu).
name	Naam van de zorgverlener

XML-voorbeeld

Een persoon heeft als huisarts de heer R. Jansen, met als AGB-Z code '01043251'.

```
<person>
...
  <asPatientOfOtherProvider>
    <subjectOf>
      <patientCareProvision>
        <code code="GENRL" codeSystem="2.16.840.1.113883.5.4" />
        <responsibleParty>
          <healthCareProvider>
            <id extension="01043251" root="2.16.840.1.113883.2.4.6.1"/>
            <providerPerson>
              <name>
                <prefix qualifier="TITLE">De heer </prefix>
                <given qualifier="IN">R.</given>
                <family>Jansen</family>
              </name>
            </providerPerson>
          </healthCareProvider>
        </responsibleParty>
      </patientCareProvision>
    </subjectOf>
  </asPatientOfOtherProvider>
...
</person>
```

PatientCareProvision.code..... *Typering zorgrelatie*
[CE \[1..1\]](#) <= *ActMedicalServiceCode*

Het element *<patientCareProvision><code>* is een typering van de aard van de zorgrelatie tussen patiënt en zorgverlener.

ActCode (value set ActMedicalServiceCode) OID: 2.16.840.1.113883.5.4		
code	displayName HL7	Internationale omschrijving
GENRL	General	General care performed by a general practitioner or family doctor as a responsible provider for a patient. (NL: Huisarts)

XML-voorbeeld

```
<code code="GENRL" codeSystem="2.16.840.1.113883.5.4" />
```

HealthCareProvider.id Identificatie van de zorgverlener of zorgaanbieder
[II](#) [1..1]

Het element `<healthCareProvider><id>` bevat de identificatie van de betreffende zorgverlener of zorgaanbieder.

Voor Nederland wordt binnen HL7v3-interacties op dit moment vooral gebruik gemaakt van de zogenaamde AGB-Z nummers voor zorgverleners en zorgaanbieders, beheerd door Vektis. Voor communicatie op de landelijke infrastructuur (AORTA) is de Unieke Zorgverleners Identificatie (UZI) verplicht, die elke zorgverlener zal aanvragen. Voor organisaties is op de AORTA het gebruik van het UZI-registerabonneenummer (URA) verplicht.

De OID voor de AGB zorgverlenernummers is **2.16.840.1.113883.2.4.6.1**.

De OID voor UZI-nummers van natuurlijke personen is **2.16.528.1.1007.3.1**.

XML-voorbeeld

Een huisarts geïdentificeerd d.m.v. zijn AGB-Z nummer 01043251

```
<id extension="01043251" root="2.16.840.1.113883.2.4.6.1" />
```

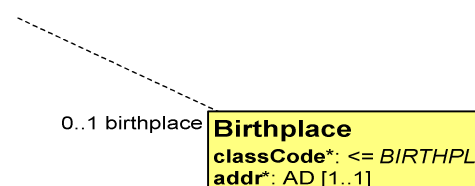
Een huisarts wordt geïdentificeerd d.m.v. zijn UZI-nummer 1234567.

```
<id extension="001234567" root="2.16.528.1.1007.3.1" />
```

ProviderPerson.name Naam van de zorgverlener (bijvoorbeeld huisartsnaam)
[PN](#) [1..1]

Het element `<providerPerson><name>` bevat de naam van de betreffende zorgverlener. De klasse *ProviderPerson* is op zich **optioneel aanwezig**, maar als deze klasse gebruikt wordt, dan is het element *name* **verplicht gevuld** (omdat anders de klasse geen functie heeft).

7.4.2.5 Klasse Birthplace (Geboorteplaats)



De klasse *Birthplace* is **optioneel** aanwezig in de CMET *E_Person [universal]* en geeft de geboorteplaats (meestal in de vorm van een land of een stad) van de persoon weer.

De klasse *Birthplace* heeft de volgende kenmerken:

classCode	BIRTHPL (Birthplace) Geboorteplaats (van een persoon).
addr	Adresaanduiding

XML-voorbeeld

```
<person>
  ...
  <birthplace>
    ...
  </birthplace>
  ...
</person>
```

Birthplace.addr..... Adresaanduiding
[AD \[1..1\]](#)

Het element *<birthplace><addr>* is **verplicht gevuld** als de klasse *Birthplace* aanwezig is en geeft een adresaanduiding van de geboorteplaats van de betreffende persoon. Zie hiervoor de gedetailleerde beschrijving van datatype AD (Address).



Merk op dat bij het vastleggen en doorgeven van geboorteplaats meestal alleen het geboorteland (bij buitenlandse personen) of de geboortestad binnen Nederland. Meer gedetailleerde adresaanduidingen zijn ook mogelijk, maar meestal niet relevant als gegeven bij de persoon.

XML-voorbeeld

Een persoon is geboren in Marokko.

```
<addr>
  <country>Marokko</country>
</addr>
```

Dezelfde aanduiding, maar nu met landcode.

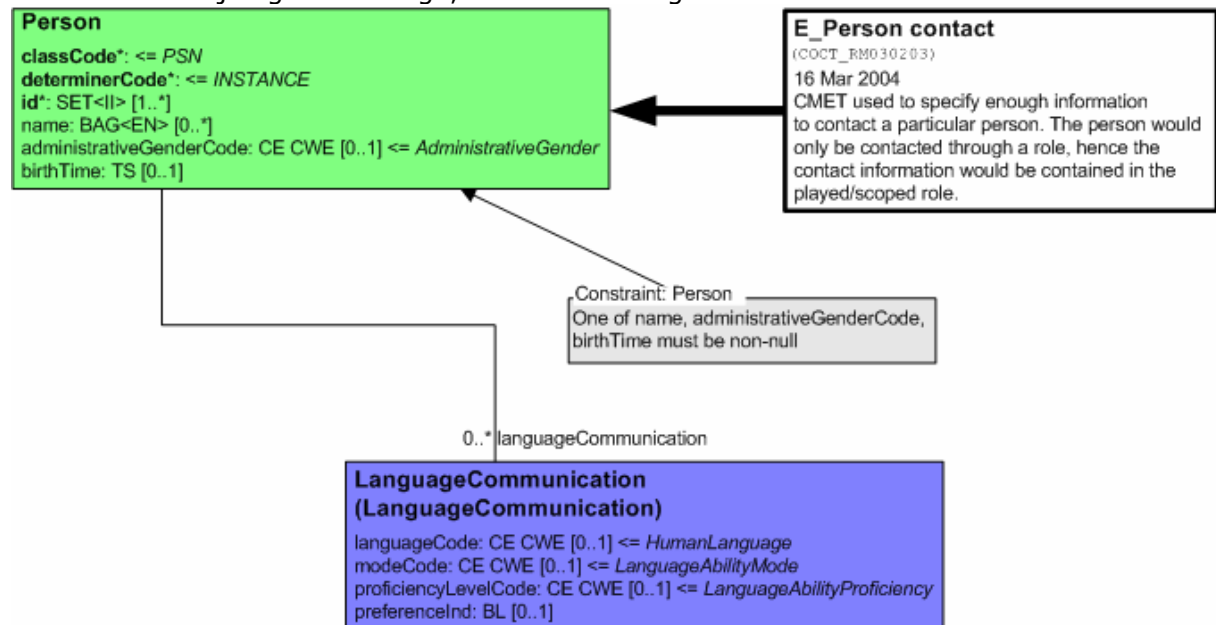
```
<addr>
  <country codeSystem="1.0.3166.1.2.2" code="MA">Marokko</country>
</addr>
```

Een persoon is geboren in Markelo.

```
<addr>
  <city>Markelo</city>
</addr>
```

7.4.3 E_Person [contact]

Deze CMET is de voor contactdoeleinden ingeperkte versie van E_PersonNL [universal]. De klasse LanguageCommunication wordt hier beschreven. Zie E_PersonNL [universal] voor de beschrijving van overige, overeenkomstige attributen.



De associatie LanguageCommunication is **optioneel en herhalend** aanwezig in de CMET E_Person [contact].

Het kan zinvol zijn, met name als de persoon een patiënt betreft, om door te geven welke taal of talen iemand machtig is, als daar iets specifiek bekend is. Dit is met name relevant als de persoon onvoldoende Nederlands beheerst en er bijvoorbeeld een slecht-nieuws-gesprek moet worden gevoerd waarvoor goed begrip noodzakelijk is. De associatie LanguageCommunication komt net zo vaak voor als er combinaties van taal en expressiemethode moeten worden gecommuniceerd.

De klasse *LanguageCommunication* heeft de volgende kenmerken:

languageCode	Taal.
modeCode	Expressiemethode zoals lezen, schrijven, spreken, verstaan
proficiencyLevelCode	Mate waarin de expressiemethode wordt beheerst.
preferenceInd	Uiting van voorkeur voor een bepaalde combinatie van taal/expressiemethode

LanguageCommunication.languageCode *Taal*
CE CWE [0..1] <= HumanLanguage

Het element `<languageCode>` is **optioneel aanwezig** in de klasse *LanguageCommunication*. Het geeft een gecodeerde aanduiding voor de betreffende taal. De code moet komen uit RFC/ISO 3066 met OID 2.16.840.1.113883.6.121. De taal wordt aangeduid conform RFC 3066 (Tags for the Identification of Languages). Het formaat om de taal aan te duiden is daarmee als volgt: ss[-CC]. ss staat voor de spraakcode, conform ISO-639-1 en de optionele CC voor landencode, conform ISO-3166 alpha-2. De aldus gemaakte taalaanduiding is **niet hoofdlettergevoelig**.

LanguageCommunication.modeCode *Expressiewijze*
CE CWE [0..1] <= LanguageAbilityMode

Het element `<modeCode>` is **optioneel aanwezig** in de klasse *LanguageCommunication*. Het geeft een gecodeerde aanduiding voor de expressiemethode van de taal indien het element `<languageCode>` een waarde heeft. Als `<languageCode>` geen waarde heeft dan betreft `<languageCommunication>` een algemene uitspraak over een bepaalde taal-expressievaardigheid. Dit is het geval bij het uitdrukken van (an)alfabetisme dat vaak niet aan enige taal is gebonden maar een algemeen persoonskenmerk betreft. Waarden komen uit de waardenset *LanguageAbilityMode* met OID 2.16.840.1.113883.6.60.

LanguageAbilityMode OID: 2.16.840.1.113883.5.60			
Lvl-Typ	Code	DisplayName	Nederlandse omschrijving
0-L	ESGN	Expressed signed	Gebruikt zelf gebaren
0-L	ESP	Expressed spoken	Spreekt zelf
0-L	EWR	Expressed written	Schrijft zelf
0-L	RSGN	Received signed	Heeft gebaren nodig
0-L	RSP	Received spoken	Kan verstaan
0-L	RWR	Received written	Kan lezen
0-L	LITRC	Literacy	Geletterdheid. Nederlandse toevoeging t.b.v. (an)alfabetisme welke EWR (schrijven) en RWR (lezen) combineert



Merk op dat de code LITRC internationaal niet bestaat maar nog zal worden ingebracht vanuit Nederland.

LanguageCommunication.proficiencyLevelCode *Mate van beheersing*
CE CWE [0..1] <= LanguageAbilityProficiency

Het element `<proficiencyLevelCode>` is **optioneel aanwezig** in de klasse *LanguageCommunication*. Het geeft een gecodeerde aanduiding voor de mate van beheersing van de `<modeCode>` genoemde expressiemethode. Waarden komen uit de waardenset *LanguageAbilityProficiency* met OID 2.16.840.1.113883.6.61.

LanguageAbilityProficiency OID: 2.16.840.1.113883.5.61			
Lvl-Typ	Code	DisplayName	Nederlandse omschrijving
0-L	E	Excellent	Uitstekend
0-L	F	Fair	Redelijk
0-L	G	Good	Goed
0-L	P	Poor	Matig

LanguageCommunication.preferenceInd *Uiting van voorkeur*
BL [0..1]

Het element `<preferenceInd>` is **optioneel aanwezig** in de klasse *LanguageCommunication*. Het geeft een voorkeur voor de in de klasse genoemde taal en/of expressiemethode. Als `<preferenceInd>` geen waarde heeft dan is er geen voorkeur bekend.

Mogelijke waarden true (heeft de voorkeur), en false (heeft niet de voorkeur). Er is geen standaardwaarde. Dit attribuut moet niet worden gecommuniceerd indien het niet bekend is.

XML-voorbeelden

1) Persoon beheerst het Nederlands in een of andere mate

```
<languageCommunication>
  <languageCode code="nl" codeSystem="2.16.840.1.113883.6.121"
    displayName="Nederlands"/>
</languageCommunication>
```

2) Persoon beheerst het Fries uitstekend

```
<languageCommunication>
  <languageCode code="fy" codeSystem="2.16.840.1.113883.6.121"
    displayName="Fries"/>
  <proficiencyLevelCode code="E" codeSystem="2.16.840.1.113883.5.60"
    displayName="Uitstekend"/>
</languageCommunication>
```

3) Persoon is analfabeet en geeft er de voorkeur om in het Engels aangesproken te worden

```
<languageCommunication>
  <modeCode code="LTRC" codeSystem="2.16.840.1.113883.5.60"
    displayName="Literacy"/>
  <proficiencyLevelCode code="P" codeSystem="2.16.840.1.113883.5.61"
    displayName="Matig"/>
</languageCommunication>
<languageCommunication>
  <languageCode code="en" codeSystem="2.16.840.1.113883.6.121"
    displayName="Engels"/>
  <modeCode code="RSP" codeSystem="2.16.840.1.113883.6.121"
    displayName="Kan verstaan"/>
  <preferenceInd value="true"/>
</languageCommunication>
```

7.5 R_AssignedPerson (persoon in een rol)

De CMET R_AssignedPerson is bedoeld voor het beschrijven van een persoon in zijn rol als medewerker, in de breedste zin van het woord. De term 'assigned' slaat op het feit dat een zorgverlener of andere medewerker in de zorg altijd gesanctioneerd is door een organisatie waar hij of zij voor werkt. De term uit de HL7v3 standaard komt wat gezocht over, maar komt dus neer op een **persoon met een toegewezen verantwoordelijkheid**. In de praktijk wordt de CMET gebruikt voor directe zorgverleners (artsen, verpleegkundigen) en eventueel voor andere medewerkers met toegang tot zorginformatie (dat komt overeen met iedereen met een persoonsgebonden UZI pas).



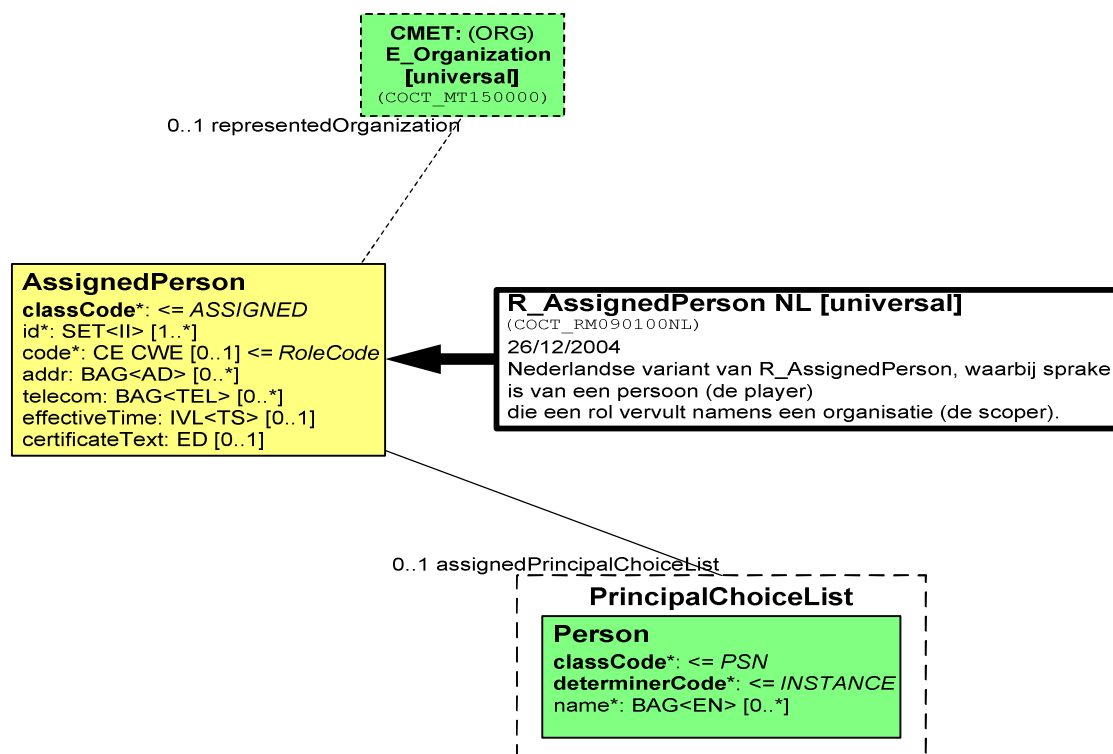
In de Control Act-wrapper van HL7v3-interacties wordt ook gebruik gemaakt van deze CMET. Wanneer een interactie wordt geïnitieerd door een patiënt, bijvoorbeeld vanuit een Personal Health Record (PHR), wordt de betekenis van deze CMET dus opgerekt aangezien de patiënt geen medewerker is van enige organisatie als hij bijvoorbeeld zijn eigen dossier raadpleegt, maar wel degelijk auteur en/of verantwoordelijke voor deze inzage. Deze situatie is internationaal aangekaart en zal in de volgende versie van de wrappers worden opgelost. Dit gebeurt waarschijnlijk door R_Patient aan de Control Act-wrapper toe te voegen en zal daarmee niet tot een wijziging leiden in R_AssignedPerson.

Van de CMET R_AssignedPerson bestaat een aantal varianten:

- R_AssignedPersonNL [universal](COCT_RM090100NL)
Volledige medewerkergegevens, onder andere geschikt om stamtabellen te synchroniseren.
- R_AssignedEntity [identified/confirmable] (COCT_RM090102)
Een unieke identificatie van de medewerker, aangevuld met de meest essentiële gegevens ter controle (naam/adres/rol).
- R_AssignedPerson [identified] (COCT_RM090101)
Niets anders dan een unieke identificatie van de medewerker.

7.5.1 R_AssignedPersonNL [universal]

Van de universele variant is een Nederlandse restrictie gemaakt. Het verschil met de *Identified/Confirmable* variant is dat de universele CMET het mogelijk maakt om gegevens over telecommunicatie, certificaten en duur van de rol vast te leggen.



Het telecom element bevat de bereikbaarheidsgegevens van de persoon, zoals telefoonnummer(s) en e-mailadres(sen). Naast telefoon- en faxnummers, kan dit element voor zorgverleners tevens de Applicatie Identificatie bevatten van de applicatie die de zorgverlener gebruikt.

XML-voorbeeld

```

<AssignedPerson>
  <id root="2.16.528.1.1007.3.1" extension="01012345"/>
  <addr use="WP">
    <streetName>Valkendael</streetName>
    <houseNumber>45</houseNumber>
    <postalCode>6245 HK</postalCode>
    <city>Maastricht</city>
  </addr>
  <telecom use="WP" value="tel:+31434073576"/>
  <telecom use="WP" value="mailto:info@amphora.nl"/>
  <telecom use="WP"
    value="x-hl7-applicatie:2.16.840.1.113883.2.4.6.6.75023"/>
  <assignedPrincipalChoiceList>
    <assignedPerson>
      <name>
        <prefix qualifier="AC">dr. </prefix>
        <given>Peter</given>
        <prefix qualifier="VV">van </prefix>
        <family qualifier="BR">Pulver</family>
      </name>
    </assignedPerson>
  </assignedPrincipalChoiceList>

```



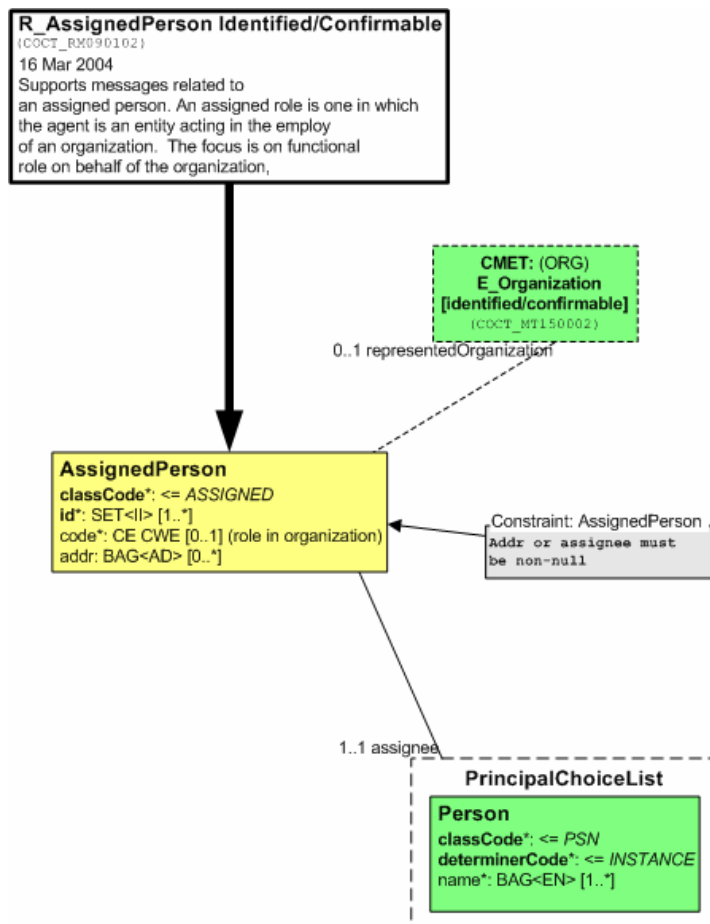
```

<Organization>
  <id nullFlavor="NI"/>
</Organization>
</AssignedPerson>

```

7.5.2 R_AssignedPerson [identified/confirmable]

Indien een id alleen niet voldoende is wordt de *Identified/Confirmable* variant gebruikt. Naam, adres en organisatie kunnen met deze CMET worden doorgegeven.



XML-voorbeeld

```

<AssignedPerson>
  <id extension="03004256" root="2.16.840.1.113883.2.4.6.1"/>
  <code codeSystem="2.16.840.1.113883.2.4.15.111" code="01.014"
    displayName="Chirurg"/>
  <addr use="WP">
    <streetName>Valkendael</streetName>
    <houseNumber>45</houseNumber>
    <postalCode>6245 HK</postalCode>
    <city>Maastricht</city>
  </addr>
  <assignedPrincipalChoiceList>

```

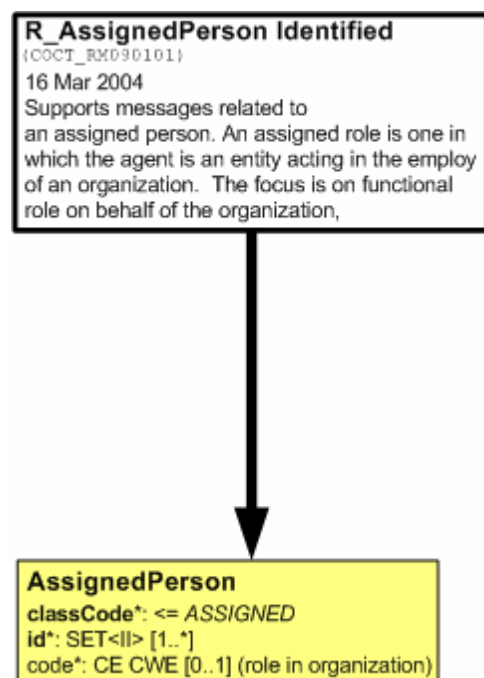
```

<assignedPerson>
  <name>
    <prefix qualifier="AC">Dr. </prefix>
    <given>Peter</given>
    <prefix qualifier="VV">van </prefix>
    <family qualifier="BR">Pulver</family>
  </name>
</assignedPerson>
</assignedPrincipalChoiceList>
<Organization>
  <id nullFlavor="NI"/>
</Organization>
</AssignedPerson>

```

7.5.3 R_AssignedPerson [identified]

Dit is de meest eenvoudige variant om personen, die betrokken zijn bij het zorgproces, te identificeren. Van deze CMET wordt in Nederland alleen het *id* element gebruikt.



XML-voorbeeld

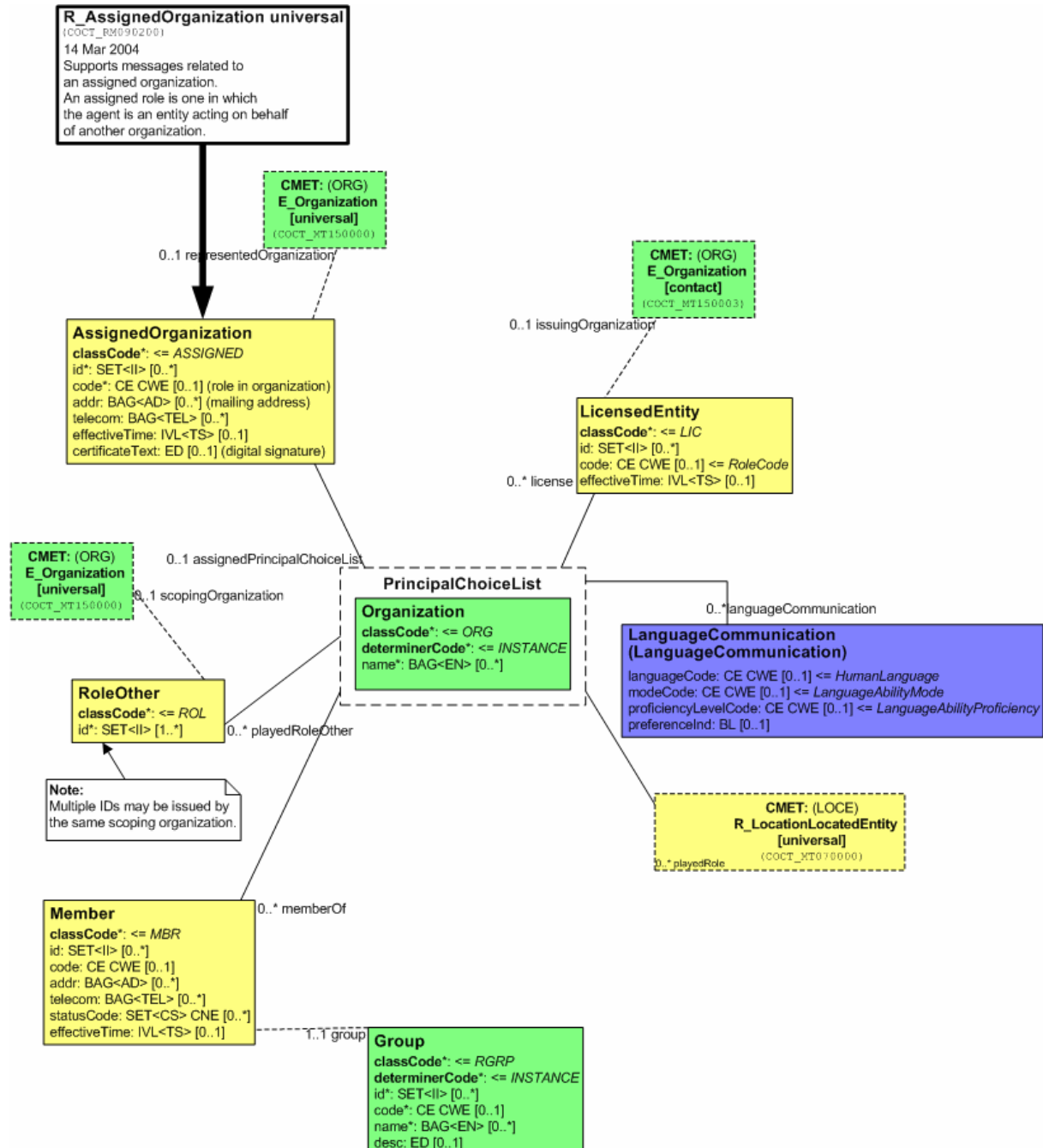
```

<AssignedPerson>
  <id extension="03004256" root="2.16.840.1.113883.2.4.6.1"/>
</AssignedPerson>

```

7.6 R_AssignedOrganization (organisatie)

7.6.1 R_AssignedOrganization [universal]



XML-voorbeeld

```
<AssignedOrganization>
```

```

<id root="2.16.840.1.113883.2.4.6.1" extension="06003652"/>
<id extension="00501534" root="2.16.528.1.1007.3.3"/>
<addr use="WP">
  <streetName>Valkendael</streetName>
  <houseNumber>45</houseNumber>
  <postalCode>6245 HK</postalCode>
  <city>Maastricht</city>
</addr>
<telecom use="WP" value="tel:+31434073576"/>
<assignedPrincipalChoiceList>
  <assignedOrganization>
    <name>Gezondheidscentrum Aesculaap</name>
  </assignedOrganization>
</assignedPrincipalChoiceList>
</AssignedOrganization>

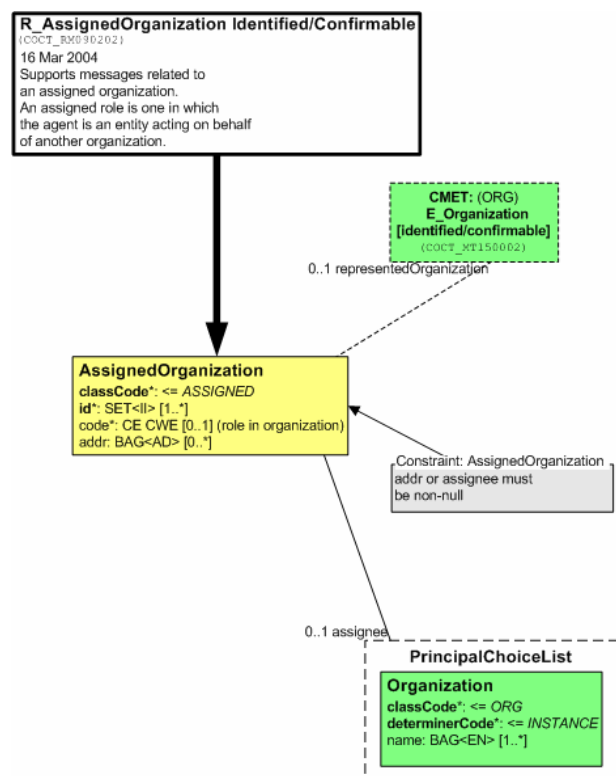
```

7.6.2 R_AssignedOrganization [identified/confirmable]

Van deze CMET worden in Nederland de volgende onderdelen gebruikt:

Id	identificatie van de organisatie
addr	adres van de organisatie
name	naam van de organisatie (in Organization)

Er dient minimaal één id plus adres of naam te worden vastgelegd.



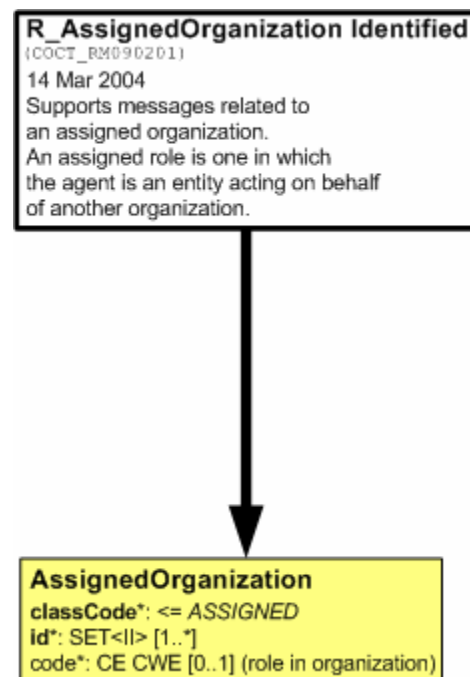
XML-voorbeeld

Een organisatie met UZI-registerabonneenummer 00002222:

```
<AssignedOrganization>
  <id root="2.16.528.1.1007.3.3" extension="00002222"/>
  <assignee>
    <assigneeOrganization>
      <name>St. Jozefziekenhuis</name>
    </assigneeOrganization>
  </assignee>
</AssignedOrganization>
```

7.6.3 R_AssignedOrganization [identified]

Voor het identificeren van een organisatie middels één of meerdere id's volstaat deze CMET. Het *code* element wordt in Nederland niet gebruikt.



XML-voorbeeld

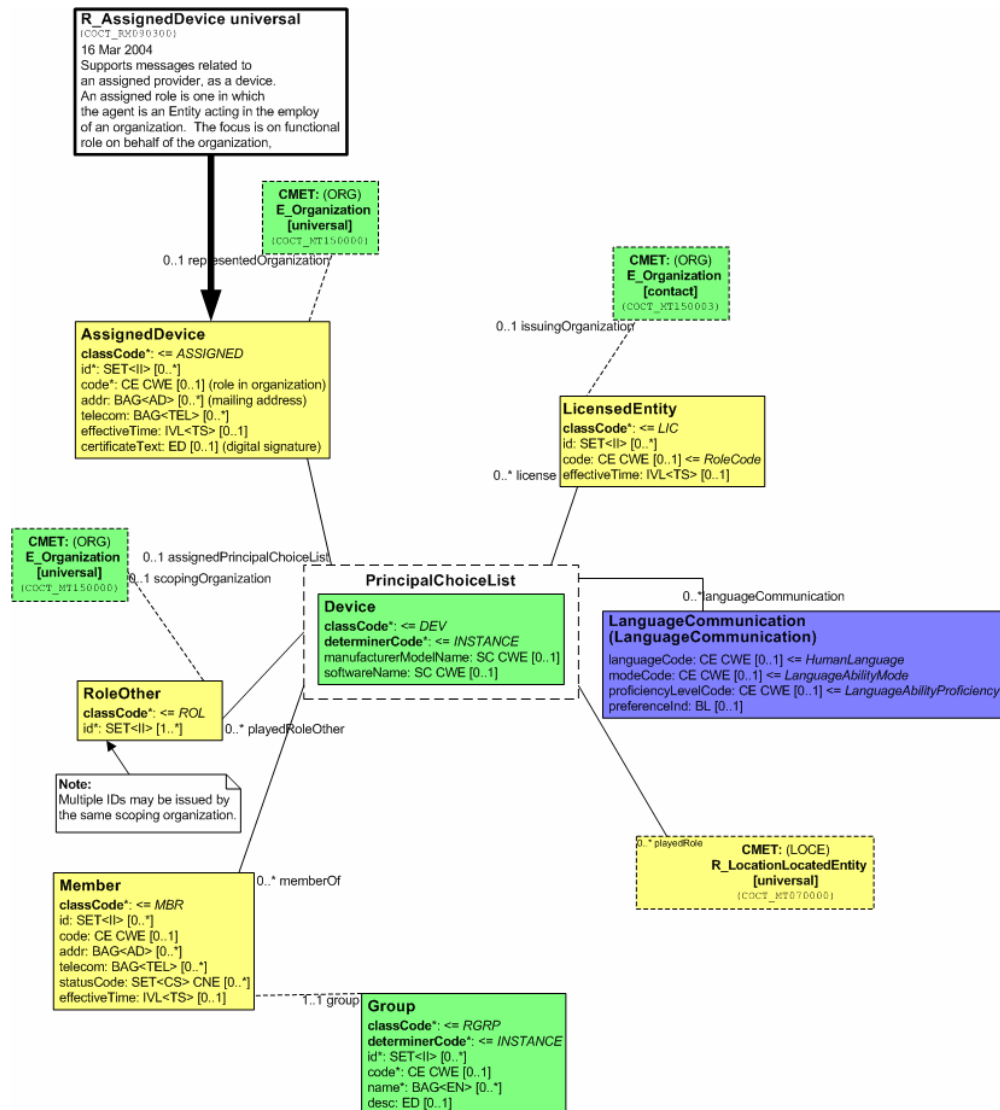
Een organisatie met UZI-registerabonneenummer 00002222:

```
<AssignedOrganization>
  <id root="2.16.528.1.1007.3.3" extension="00002222"/>
</AssignedOrganization>
```

7.7 R_AssignedDevice (apparaat)

Deze CMET wordt gebruikt om gegevens rond apparatuur vast te leggen.

7.7.1 R_AssignedDevice [universal]



XML-voorbeeld

```

<AssignedDevice>
  <id root="2.16.528.1.1007.3.3.12345.1" extension="006456"/>
  <telecom value="http://www.Aesculaap.nl"/>
  <assignedPrincipalChoiceList>
    <assignedDevice>

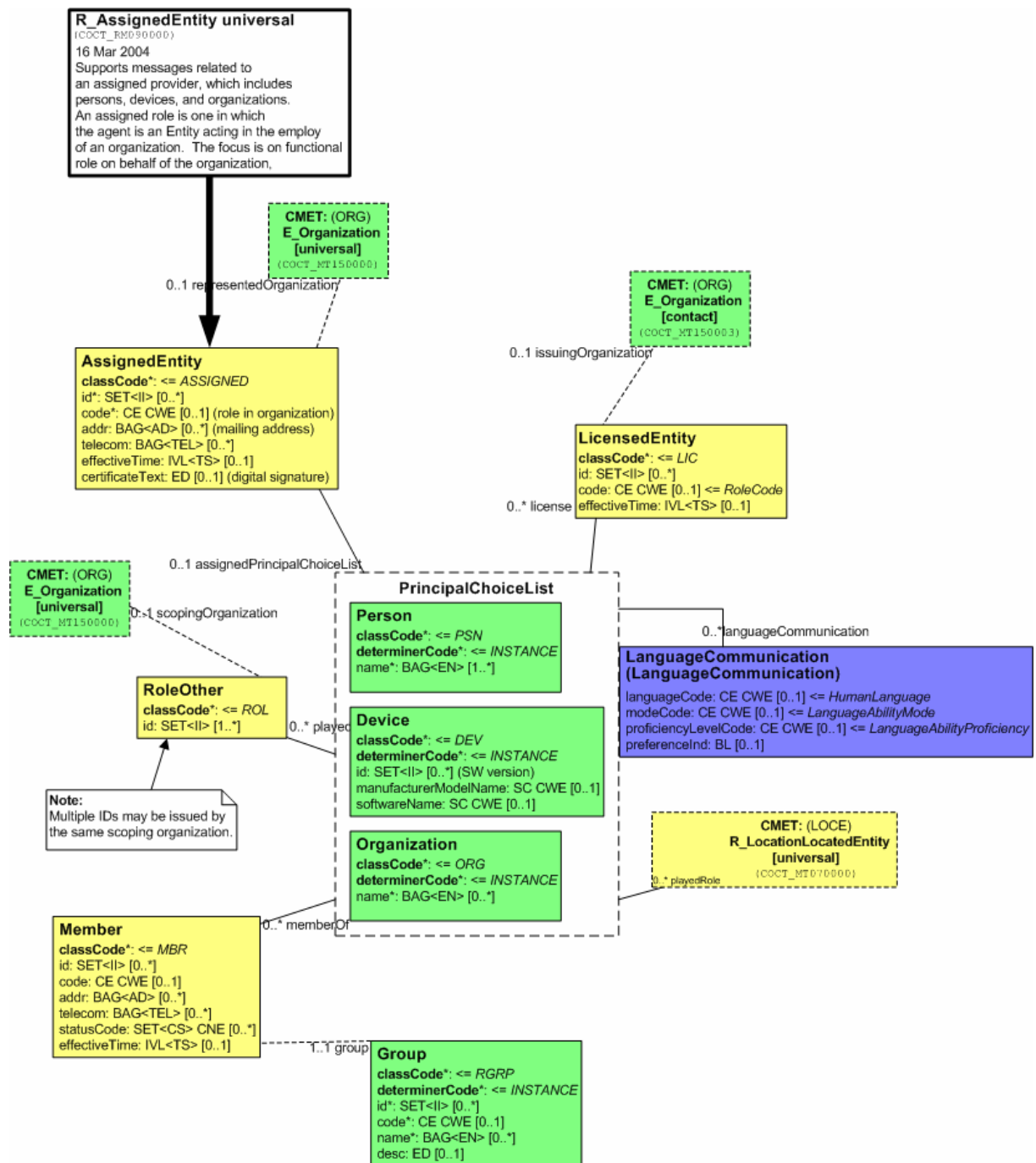
```

```
    <softwareName>Apache web server</softwareName>
  </assignedDevice>
</assignedPrincipalChoiceList>
<Organization>
  <id root="2.16.528.1.1007.3.3" extension="00012345"/>
  <name>Gezondheidscentrum Aesculaap</name>
</Organization>
</AssignedDevice>
```

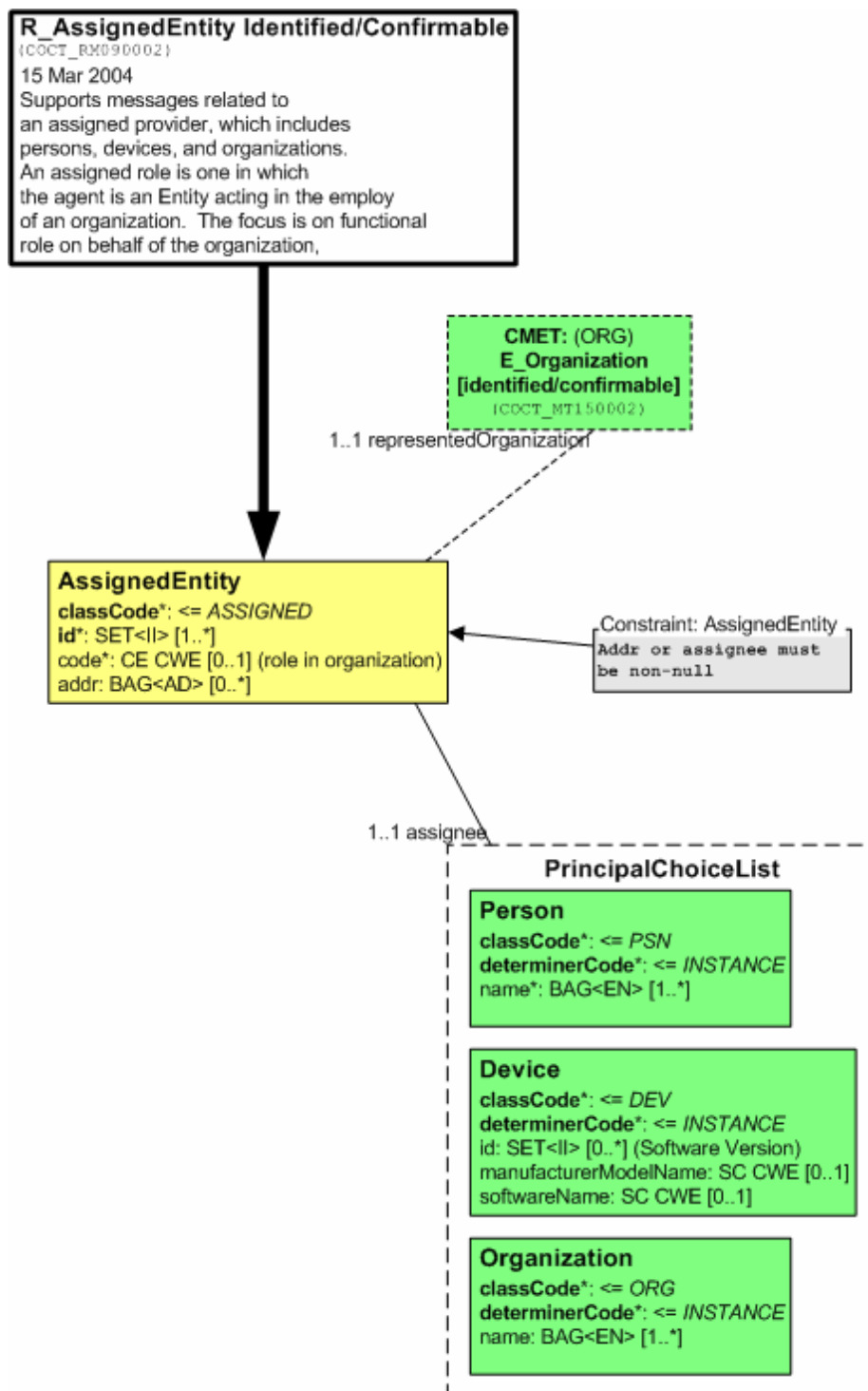
7.8 R_AssignedEntity (entiteit in een rol)

De R_AssignedEntity CMET bestaat uit een algemene rol die de "verantwoordelijke" aan-geeft. De verantwoordelijke wordt gespeeld door of een instantiatie van een Person, of een Organization, NonPersonLivingSubject of een Device klasse. In feite is deze CMET een algemene versie van of de R_AssignedOrganization CMET, de R_AssignedDevice CMET en de R_AssignedPerson CMET. Zie de beschrijving van die CMET's voor details rondom gebruik van de Rol klasse en de entiteit die die rol speelt.

7.8.1 R_AssignedEntity [universal]



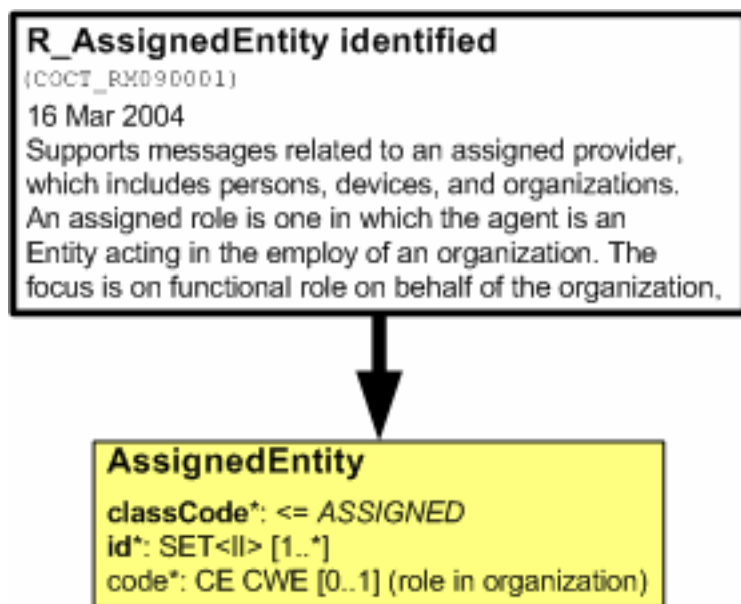
7.8.2 R_AssignedEntity [identified/confirmable]



7.8.3 R_AssignedEntity [identified]

De *[identified]* variant van de CMET *R_AssignedEntity* is een restrictie op de *[universal]* variant, waarin alleen de unieke identificatie(s) en eventueel een (rol)code zijn opgeno-

men in het informatiemodel. Dit wordt alleen gebruikt om een bij de ontvanger reeds bekende entiteit in deze rol te identificeren, maar niet om wijzigingen door te geven.



XML-voorbeeld

Een organisatie met UZI-registerabonneenummer 000002222:

```
<AssignedEntity>
  <id root="2.16.528.1.1007.3.3" extension="000002222"/>
</AssignedEntity>
```

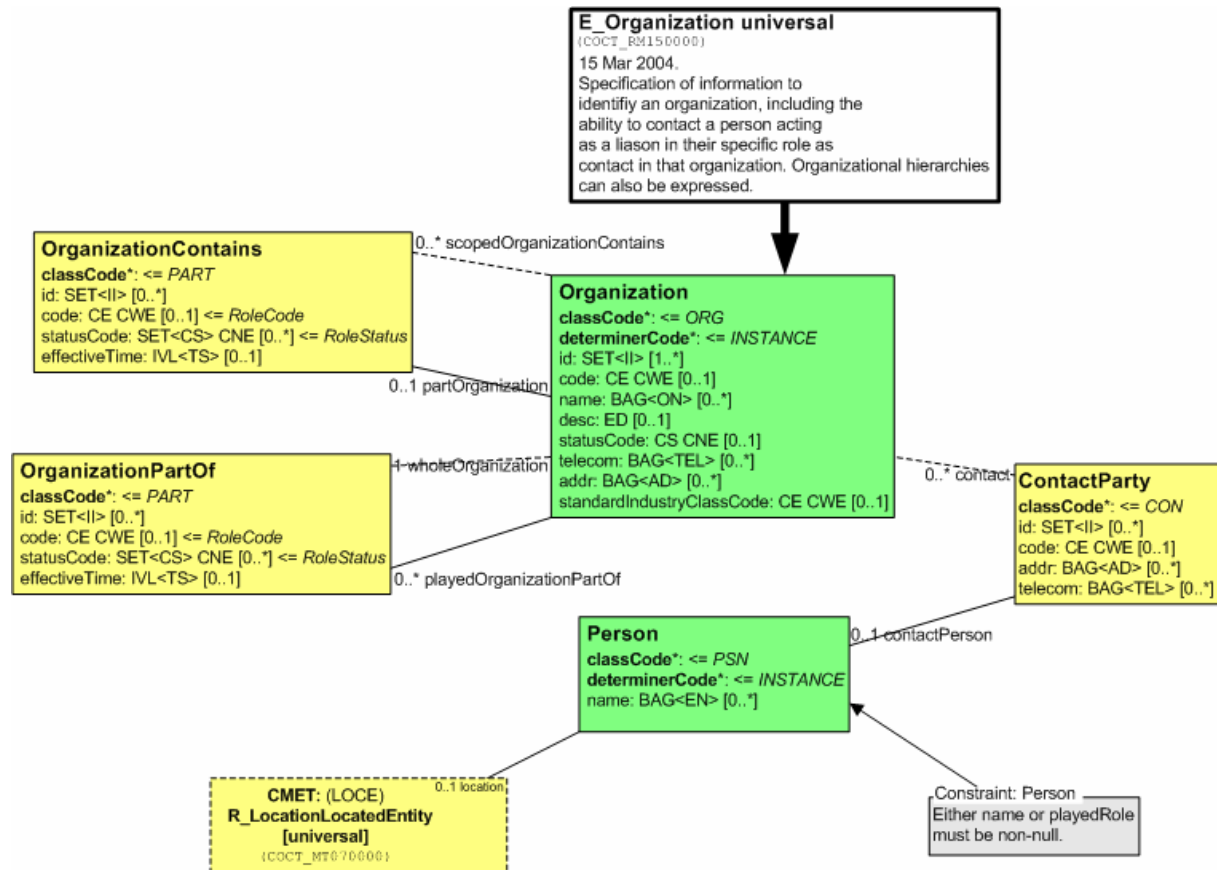
Een huisarts met UZI-nummer 12344321:

```
<AssignedEntity>
  <id root="2.16.528.1.1007.3.1" extension="12344321"/>
  <code code="01.015" codeSystem="2.16.840.1.113883.2.4.15.111"/>
</AssignedEntity>
```

7.9 E_Organization (organisatie)

Deze CMET wordt zelden los gebruikt, maar meestal als onderdeel van andere CMET's.

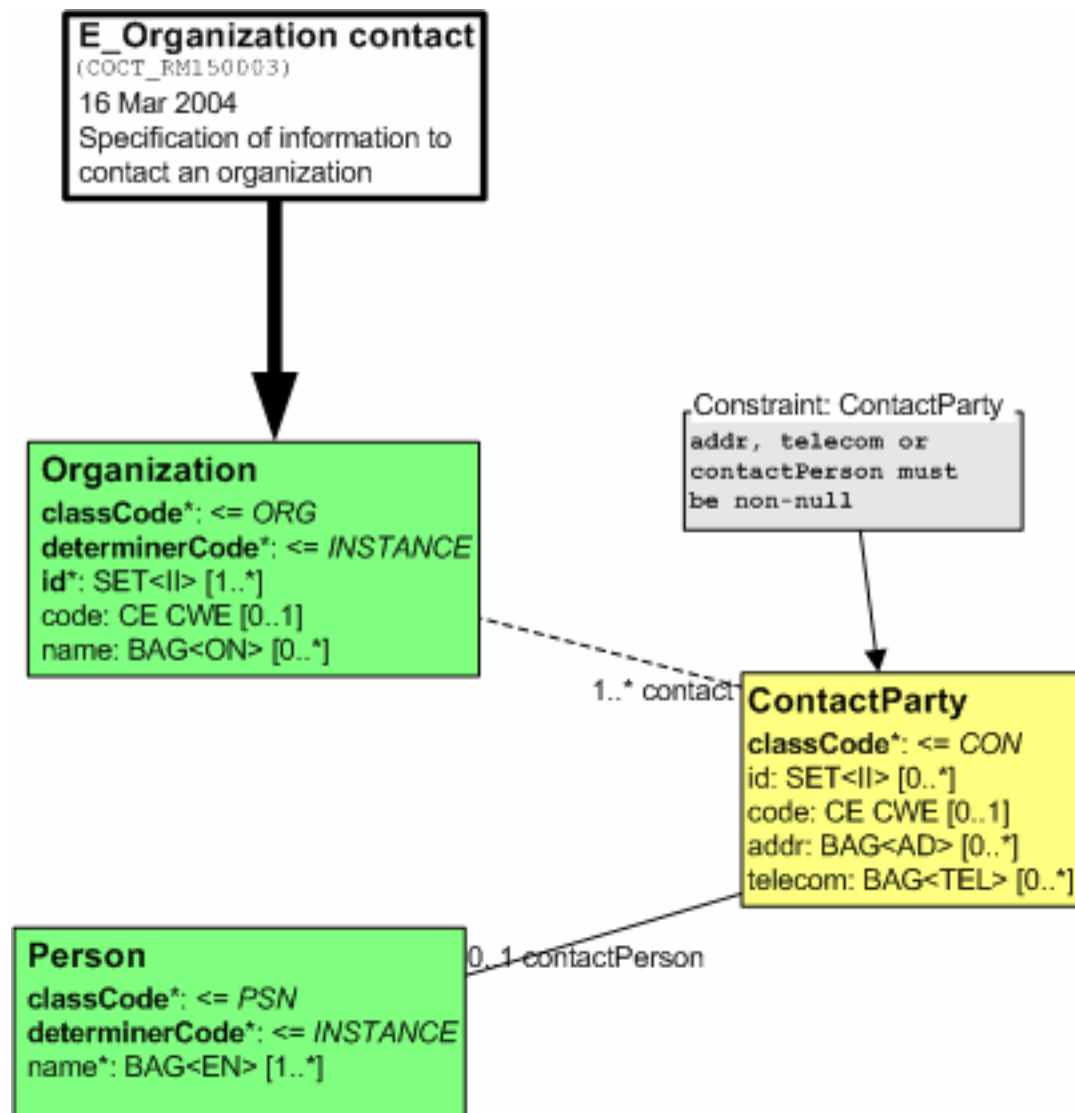
7.9.1 E_Organization [universal]



XML-voorbeeld

```
<Organization>
  <id root="2.16.840.1.113883.2.4.6.1" extension="06004213"/>
  <name>Gezondheidscentrum Aesculaap</name>
  <telecom use="WP" value="tel:+31434073576"/>
  <addr use="WP">
    <streetName>Valkendael</streetName>
    <houseNumber>45</houseNumber>
    <postalCode>6245 HK</postalCode>
    <city>Maastricht</city>
  </addr>
</Organization>
```

7.9.2 E_Organization [contact]



Van een organisatie die contactpartij is moet tenminste een naam van de contactpersoon in `<person>.<name>` bekend zijn of een (telecommunicatie)adres van de organisatie `<contact>.<addr>` of `<contact>.<telecom>`.

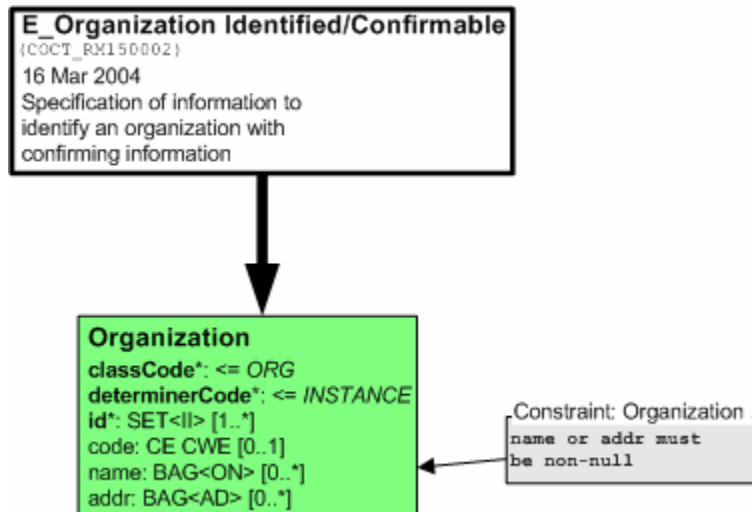
```

<Organization>
  <id extension="11223344" root="2.16.528.1.1007.3.3"/>
  <contact>
    <telecom value="tel:010-1234567"/>
    <contactPerson>
      <name>Gert Kamermans</name>
    </contactPerson>
  </contact>
</Organization>

```

7.9.3 E_Organization [identified/confirmable]

Naast id's kan deze CMET ook adresgegevens en de naam van de organisatie bevatten. Het code element wordt wederom niet gebruikt. Deze CMET zal vooral gebruikt worden om de naam van een organisatie door te geven.

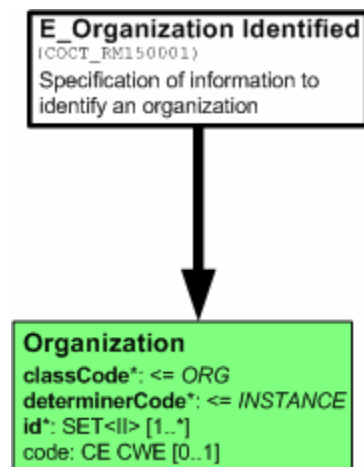


Tenminste één van de elementen <name>, <addr> moet een waarde hebben zonder het attribuut @nullFlavor.

XML-voorbeeld

```
<Organization>
  <id root="2.16.840.1.113883.2.4.6.1" extension="06004213"/>
  <name>Gezondheidscentrum Aesculaap</name>
  <addr>
    <streetName>Valkendael</streetName>
    <houseNumber>45</houseNumber>
    <postalCode>6245 HK</postalCode>
    <city>Maastricht</city>
  </addr>
</Organization>
```

7.9.4 E_Organization [identified]



XML-voorbeeld

```
<Organization>  
  <id extension="1" root="2.16.840.1.113883.2.4.6.5"/>  
</Organization>
```

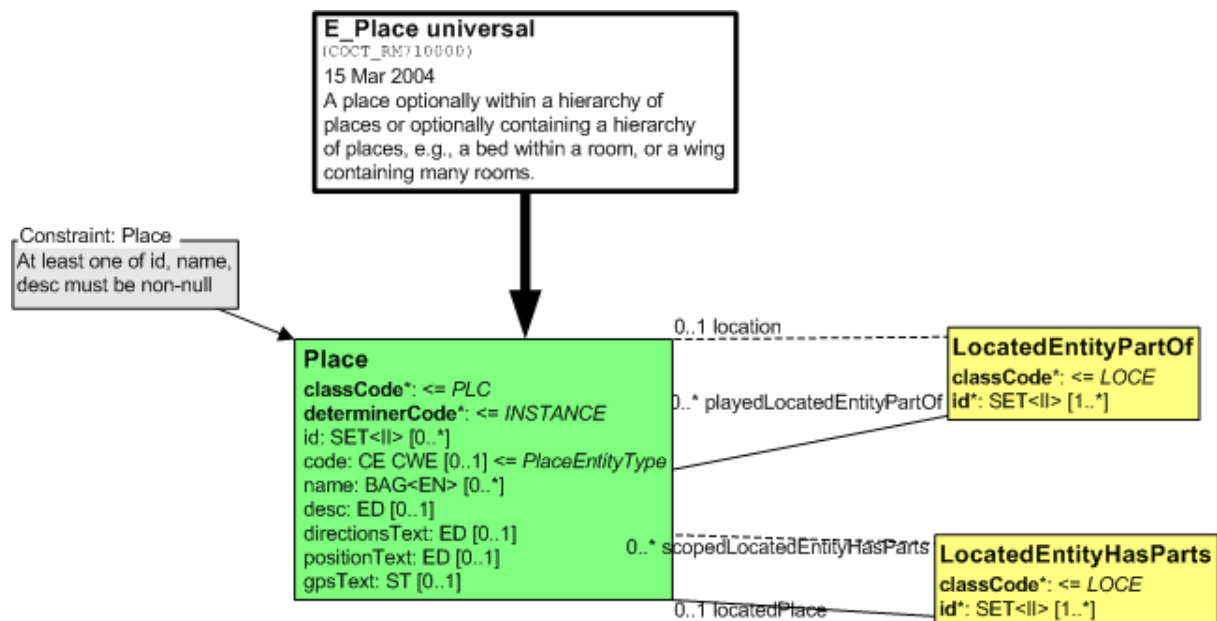
7.10 E_Place (plaats)

Deze CMET wordt gebruikt om gegevens over een plaats vast te leggen. In Nederland zal slechts een aantal van de onderdelen gebruikt worden:

id	identificatie van de plaats
name	naam van de plaats
desc	omschrijving van de plaats

De relaties worden in Nederland nog niet gebruikt. Tenminste één van *<id>*, *<name>*, of *<desc>* moet een waarde hebben zonder het attribuut *@nullFlavor*.

7.10.1 E_Place [universal]



XML-voorbeeld

```
<Place>
  <name>Naam van een plaats, afdeling, kast of anders</name>
  <desc>Omschrijving van de plaats, dit is een ED type en
    kan ook een foto bevatten.</desc>
  <directionsText>Beschrijving van hoe de plaats te bereiken.
    ED type, kan dus ook een routekaart zijn.</directionsText>
</Place>
```


7.11 R_LocationLocatedEntity (locatie)

Deze CMET wordt gebruikt om een bepaalde locatie aan te duiden, bijvoorbeeld een bed of een afdeling in een ziekenhuis, een verdieping of een gebouw in een zorgaanbieder. Dit wordt meestal gezien in een associatie met andere klassen en kan ook hiërarchisch worden gebruikt, om een locatie binnen en locatie aan te geven.

7.11.1 R_LocationLocatedEntity [universal]

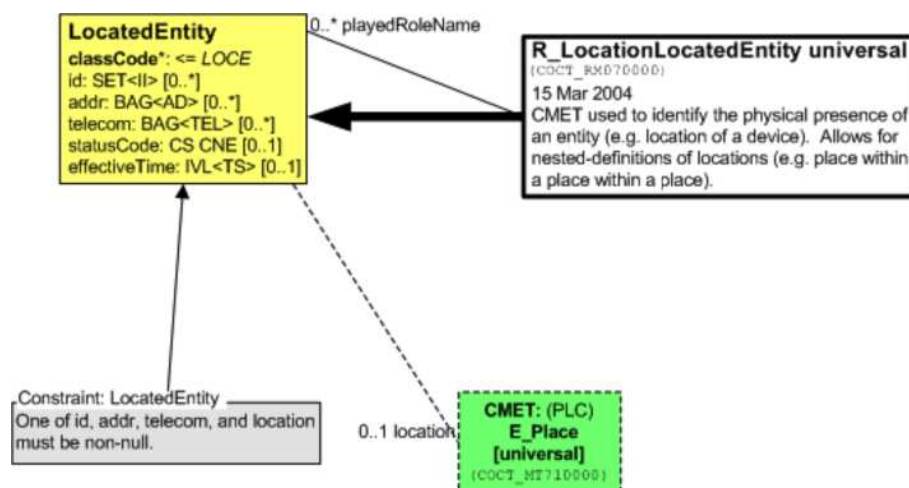
Van deze CMET worden in Nederland de volgende onderdelen gebruikt:

id	identificatie van de locatie
addr	adres van de locatie
telecom	bijhorende telecommunicatiecontacten
statusCode	status van de locatie
effectiveTime	geldigheidstermijn van de locatie

Van deze CMET wordt in Nederland de volgende relatie gebruikt:

Location	Naam enz van de locatie (in E_Place)
-----------------	--------------------------------------

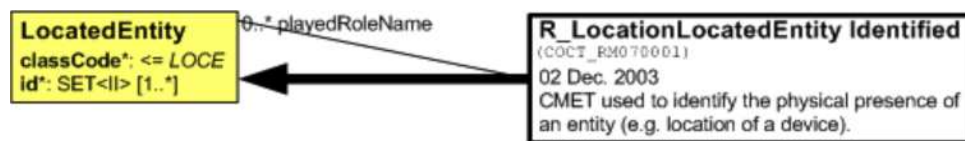
Er dient minimaal één id, adres, telecom of de associatie location te worden vastgelegd.



7.11.2 R_LocationLocatedEntity [identified]

Als een identificatie van een locatie bekend is kan de identified variant van de CMET gebruikt worden.

id	Verplichte identificatie van de locatie
-----------	---



XML-voorbeeld

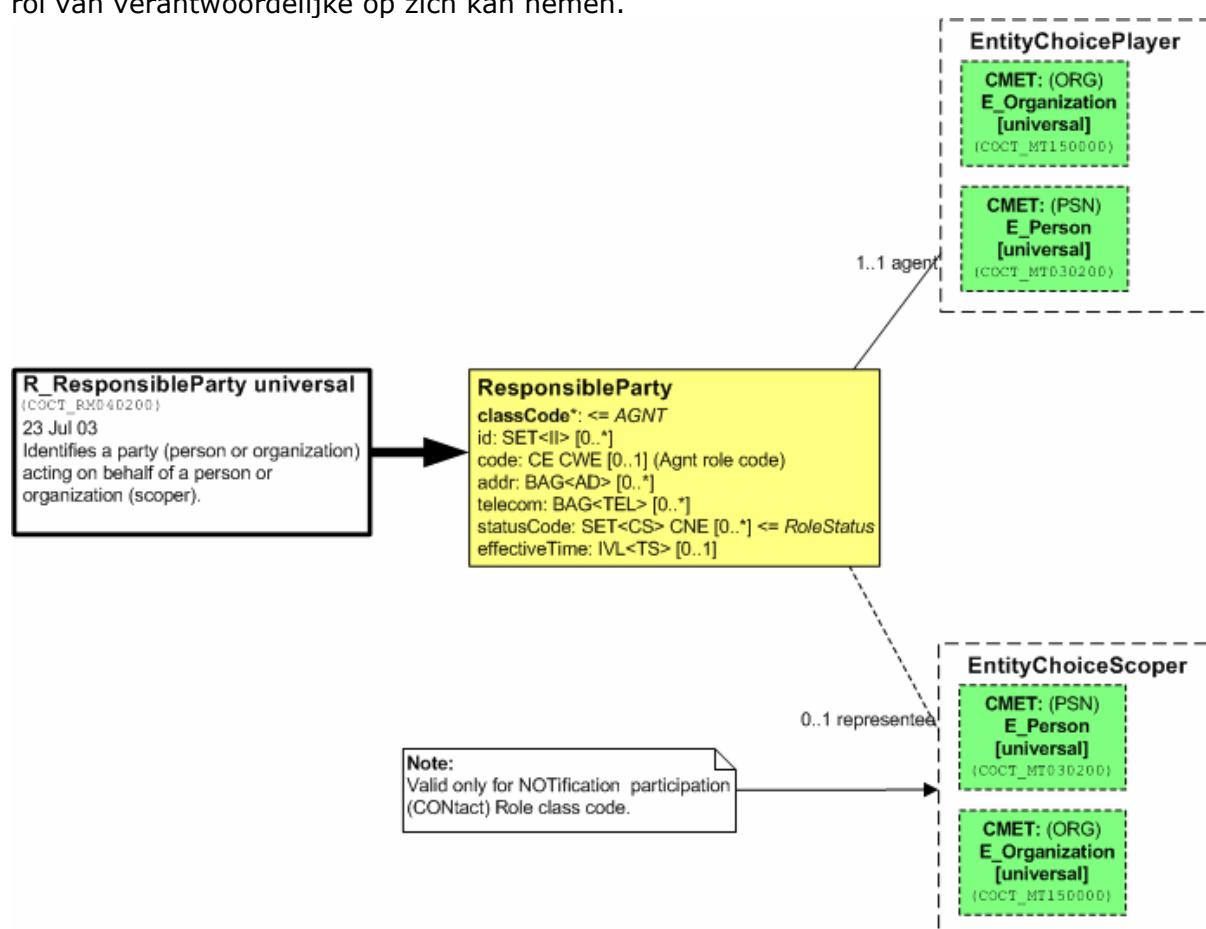
```
<LocatedEntity>
  <id extension="012" root="2.16.528.1.1007.3.3.12345.23.1"/>
</LocatedEntity>
```

7.12 R_ResponsibleParty (verantwoordelijke partij)

Deze CMET's worden gebruikt om een partij te identificeren die optreedt als verantwoordelijke ten behoeve van een persoon of organisatie. Van R_ResponsibleParty is ook een ingeperkte versie beschikbaar R_NotificationParty die dus in tegenstelling tot normale inperkingen een andere naam heeft gekregen.

7.12.1 R_ResponsibleParty [universal]

Deze CMET is een ingeperkte versie van R_Responsible (waarvan in Nederland nog geen gebruik bekend is), maar waar naast een organisatie of persoon ook nog een apparaat de rol van verantwoordelijke op zich kan nemen.



ResponsibleParty	Contactgegevens
-------------------------	------------------------

Van de klasse **ResponsibleParty** moet naast de required/mandatory elementen tenminste een adres, of een communicatieaanduiding worden doorgegeven. De klasse **ResponsibleParty** heeft de volgende kenmerken:

classCode	AGNT (Contactpartij)
------------------	-----------------------------

	Een persoon of organisatie in de rol van contactpartij
id	Contactpartij-identificatie(s)
code	Rolcode van de contactpartij
addr	Woon/verblijfadres(sen)
telecom	Communicatieaanduiding(en)
statusCode	Status van de contactpartij
effectiveTime	Geldigheidsperiode van de contactpartij

De klasse *ResponsibleParty* heeft de volgende associaties:

1..1 agentOrganization of agentPerson	Gegevens van de contactpersoon of contactorganisatie
	CMET E_PersonNL [universal] Persoonsgegevens
	CMET E_Organization [universal] Organisatiegegevens
1..1 representedOrganization of representedPerson	Gegevens van de verantwoordelijke persoon of organisatie voor de contactpartij
	CMET E_PersonNL [universal] Persoonsgegevens
	CMET E_Organization [universal] Organisatiegegevens

Voor een toelichting op deze associaties wordt verwezen naar de betreffende CMET's.

ResponsibleParty.id *Contactpartij-identificatie(s)*
[SET<II> \[1..*\]](#)

Het element *<ResponsibleParty><id>* is **verplicht gevuld** binnen *R_ResponsibleParty* en bevat **één of meer** contactpartij-identificatie(s), d.w.z. *instance identifiers* die de contactpartij onderscheiden van andere (potentiële) contactpartijen. De gehanteerde identificaties hangen af van de context waarin de CMET wordt toegepast.

ResponsibleParty.code *Rolcode van contactpartij*
[CE CWE \[0..1\]](#)

Het element *<ResponsibleParty><code>* is **aanwezig indien bekend** en bevat een geldige code uit het zeer uitgebreide waardendomein RoleCode met OID 2.16.840.1.113883.5.111. Hierin zijn bijvoorbeeld codes voor familiërelaties te vinden.

ResponsibleParty.addr *Adres(sen)*
[BAG<AD> \[0..*\]](#)

Het element *<ResponsibleParty><addr>* is **aanwezig indien bekend** en bevat **één of meer** actuele adressen van de contactpartij in diens rol.

ResponsibleParty.telecom Communicatieaanduiding(en)
[BAG<TEL> \[0..\]](#)*

Het element `<ResponsibleParty><telecom>` is **aanwezig indien bekend** en bevat **één of meer** actuele telefoonnummers of andere communicatieaanduidingen die gebruikt kunnen worden om de contactpartij te bereiken. Hieronder valt bijvoorbeeld telefoonnummer thuis, telefoonnummer op het werk, mobiel nummer of fax, maar ook een **e-mailadres** of **website**.

ResponsibleParty.statusCode Status van de contactpartij
[CS CNE \[1..1\] <= RoleStatus](#)

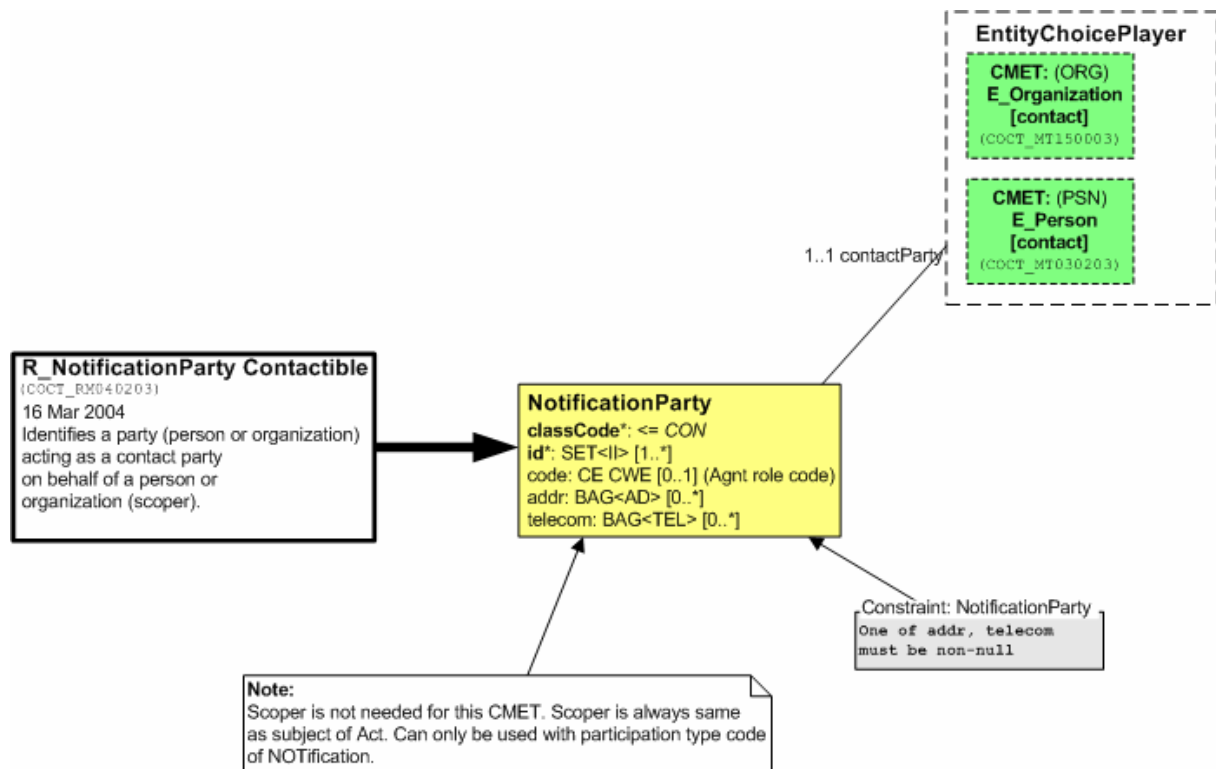
Het element `<ResponsibleParty><statusCode>` mag **optioneel één maal worden gevuld**. Normaal gesproken zal het element altijd de waarde **active** hebben, ten teken dat het een actieve contactpartij betreft. Als de contactpartij 'vervallen' is (dat wil zeggen niet meer gebruikt mag worden) dan wordt als *statusCode* **terminated** meegegeven.

ResponsibleParty.effectiveTime Geldigheidsinterval
[IVL<TS> \[0..1\]](#)

Dit element is bedoeld om door te geven vanaf en/of tot wanneer de contactpartij actief is. Vooralsnog niet gebruiken in de Nederlandse situatie.

7.12.2 R_NotificationParty [contact]

Deze CMET is de ingeperkte versie van de CMET R_ResponsibleParty [universal] voor contactdoeleinden, met als aanvullende inperking ten opzichte van R_ResponsibleParty [contact] dat de partij in kwestie alleen een contact (@classCode="CON") kan zijn. Van R_ResponsibleParty [contact] is in Nederland nog geen gebruik bekend, maar R_NotificationParty [contact] wordt in elk geval aangeroepen in de HL7v3-interactiewrappers.



NotificationParty	Contactgegevens
-------------------	-----------------

Van de klasse NotificationParty moet naast de required/mandatory elementen tenminste een adres, of een communicatieaanduiding worden doorgegeven. De klasse NotificationParty heeft de volgende kenmerken:

classCode	CON (Contact) Een persoon of organisatie in de rol van contactpartij
id	Contactpartij-identificatie(s)
code	Rolcode van de contactpartij
addr	Woon/verblijfadres(sen)
telecom	Communicatieaanduiding(en)

De klasse NotificationParty heeft de volgende associaties:

1..1 contactParty	Gegevens van de persoon of organisatie
	CMET E_Person [contact] Persoonsgegevens
	CMET E_Organization [contact] Organisatiegegevens

Voor een toelichting op deze associaties wordt verwezen naar de betreffende CMET's.

NotificationParty.id Contactpartij-identificatie(s)
[SET<II>](#) [1..*]

Het element `<NotificationParty><id>` is **verplicht gevuld** binnen *R_NotificationParty* en bevat **één of meer** contactpartij-identificatie(s), d.w.z. *instance identifiers* die de contactpartij onderscheiden van andere (potentiële) contactpartijen. De gehanteerde identificaties hangen af van de context waarin de CMET wordt toegepast.

NotificationParty.code Rolcode van contactpartij
[CE CWE](#) [0..1]

Het element `<NotificationParty><code>` is **aanwezig indien bekend** en bevat een geldige code uit het zeer uitgebreide waardendomein RoleCode met OID 2.16.840.1.113883.5.111. Hierin zijn bijvoorbeeld codes voor familiërelaties te vinden.

NotificationParty.addr Adres(sen)
[BAG<AD>](#) [0..*]

Het element `<NotificationParty><addr>` is **aanwezig indien bekend** en bevat **één of meer** actuele adressen van de contactpartij in diens rol.

NotificationParty.telecom Communicatieaanduiding(en)
[BAG<TEL>](#) [0..*]

Het element `<NotificationParty><telecom>` is **aanwezig indien bekend** en bevat **één of meer** actuele telefoonnummers of andere communicatieaanduidingen die gebruikt kunnen worden om de contactpartij te bereiken. Hieronder valt bijvoorbeeld telefoonnummer thuis, telefoonnummer op het werk, mobiel nummer of fax, maar ook een **e-mailadres** of **website**.

XML-voorbeeld

```
<NotificationParty>
  <contactParty>
    <Organization>
      <id extension="11223344" root="2.16.528.1.1007.3.3"/>
      <contact>
        <telecom value="tel:010-1234567"/>
        <contactPerson>
          <name>Gert Kamermans</name>
        </contactPerson>
      </contact>
    </Organization>
  </contactParty>
</NotificationParty>
```