Colofon

|  |  |
| --- | --- |
| Projectnaam | Elektronisch factureren |
| Versienummer  Datum | 1.1  2010-09-15 |
| Locatie | Den Haag |
| Projectleiders | Menno van Drunen  T 06-24 51 62 44  menno.drunen@logius.nl |
| Contactpersoon | Roland Hommes  T 070-8896338  roland.hommes@logius.nl |
| Organisatie | Logius  Postbus 84011  2508 AA Den Haag  [servicecentrum@logius.nl](mailto:servicecentrum@logius.nl) |
|  |  |
| Bijlage(n) | - |
|  |  |

Inhoud

Colofon 2

Inhoud 3

1 Verantwoording 4

2 Documentatie 5

2.1 Directories 5

2.2 Leesbare documenten 6

2.3 Modelleringuitgangspunten 6

2.4 Gegenereerde documenten 7

3 Functioneel ontwerp eFactuur-NL 8

3.1 De delen van het functioneel ontwerp. 8

3.1.1 De samenvatting 8

3.1.2 De detailspecificaties 8

3.1.3 De domeinspecificaties 8

3.1.4 De rulespecificaties 8

3.2 Nadere uitleg 9

3.2.1 Onderdelen met dezelfde oorsprong 9

3.2.2 Applied rules 9

3.2.3 Formaten en domein specificaties 9

3.3 Voorbeeld 10

4 Mapping document 11

4.1 Berichttags 11

4.2 Cardinaliteit 11

4.3 xPath 11

4.4 Bijzonderheden 11

5 XML, Schematron en GeneriCode 13

5.1 Beperkte uitdrukkingskracht 13

5.2 UBL XML-schemas 13

5.3 SchemaTron bestanden 13

5.4 GeneriCode 13

5.5 Generatie 13

# Verantwoording

Deze implementatiewijzer is bedoeld als leeswijzer voor de set aan documenten behorende bij de “implementatiekit eFactuur-NL – UBL”. Uitgangspunt van deze implementatiekit is het “Functioneel Ontwerp eFactuur-NL” die te vinden is in de aansluitkit e-Factureren, eveneens te downloaden op de site van Logius. De eFactuur-NL moet gezien worden als een functionele beschrijving van de in Nederland benodigde factuurgegevens.

Deze implementatiekit kunt u gebruiken bij de implementatie van de e-Factuur op basis van de UBL standaard. Daarbij is tevens rekening gehouden met de verplichte (technische) elementen zoals beschreven worden in de “CEN WS/BII agreement en de verplichte elementen uit de UBL standaard.

Om de werking van UBL Invoice 2.0 aan te tonen hebben we een set voorbeeldberichten gemaakt, gebaseerd op de strikte XML Schema’s. Per voorbeeld bericht is de inhoud opgezet in een voorbeeld bericht conform UBL Invoice 2.0. Alle voorbeeld berichten zijn gevalideerd tegen de bijbehorende XML Schema’s. Op de voorbeeldberichten na die gekenmerkt zijn als een fout bevattend, zijn er geen fouten naar voren gekomen. De kans is hiermee klein, dat de mapping van het functionele bericht op UBL Invoice 2.0 fouten bevat zoals niet kloppende herhaalfactoren, opsplitsing van herhalende gegevensgroepen en mapping van alfanumerieke velden op numerieke velden.

# Documentatie

Deze set documenten beoogt ontwerpers en bouwers inzicht te geven in de opbouw van de berichtspecificatie van de eFactuur-NL in de syntax van UBL 2.0. Dit document geeft ook aan hoe berichten voor elektronische gegevensuitwisseling worden gespecificeerd met aanvullende documenten. De documenten zijn in te delen in de volgende groepen:

1. **Functioneel ontwerp;**

Dit omvat de functionele beschrijving van elektronisch factureren in Nederland met de Nederlandse overheid. In het functioneel ontwerp worden de noodzakelijke gegevenselementen opgesomd, hun definitie en hun onderlinge verhoudingen.

2. **Mapping document;**

Hier worden de functioneel beschreven gegevenselementen gekoppeld aan een specifieke syntax. In dit geval is die syntax UBL 2.0. Daarnaast wordt een koppeling gelegd naar de eisen die door het CEN project van de EU worden gesteld aan elektronische facturen. De gekozen syntax is in de taal XML opgesteld en dus zullen semantische begrippen gekoppeld worden aan één of meer XML elementen. In het geval de mapping niet een op een plaats kan vinden zullen de transformatieregels meegegeven worden.

3. **XML bestanden;**

Voor geautomatiseerd gebruik, zoals het valideren van berichten wordt de gekozen syntax in bestanden gegoten. Voor UBL zijn dit XML schema’s, Schematron bestanden en GeneriCode bestanden. Deze bestanden zijn een technische weergave van wat in de functionele specificaties is beschreven.

Voor de gegevenselementen wordt XML Schema 1.0 gebruikt, voor business rules ISO Schematron en voor codewaarde lijsten de OASIS standaard GeneriCode.

## Directories

De documentatie van de eFactuur wordt in de directorystructuur, zoals die voorgeschreven is door UBL, geleverd en bevat de volgende directories:

Versie

cl

gc

doc

etc

mod

common

maindoc

uml

val

xml

xsd

common

maindoc

xsdrt

common

maindoc

**Versie**

Dit is de versie van de uitlevering. De versienummering kent het formaat N.M.X.Y, waarbij N de ‘major’ versie is, M de ‘minor’ versie, X voor een a (van alfa) of b (van beta) staat en Y voor het volgnummer binnen de alfa of beta. Een productie versie zal alleen een major en minor nummer bevatten.

**cl**

Dit is de ‘codelist’ directory. Hier worden de codewaarden opgenomen die voor specifieke gegevens elementen zijn toegestaan. De belangrijkste standaard is de OASIS GeneriCode (<http://docs.oasis-open.org/codelist/cs-genericode-1.0/doc/oasis-code-list-representationgenericode.html>) Maar andere standaarden inclusief de W3C XML Schema notatie voor enumeraties zijn ook mogelijk.

**gc**

Dit is de directory voor de GeneriCode bestanden.

**doc**

Dit is de directory voor de door de mens leesbare documentatie bestanden.

**etc**

Dit is de directory voor de overige bestanden, hierin zitten o.a. de Excel mapping bestanden.

**mod**

Dit is de directory waarvan UBL 2.0 voorschrijft dat zij Excel bestanden bevatten van de UBL 2.0 gespecificeerde elementen die daadwerkelijk voor de opgenomen berichten gebruikt worden.

**uml**

Dit is de directory voor de UML Class diagrammen.

**val**

Dit is de directory voor de ISO Schematron (c040833\_ISO\_IEC\_19757-3\_2006(E)) validatie bestanden.

**xml**

Dit is de directory voor de XML voorbeeld instances.

**xsd**

Dit is de directory voor de XML Schema’s van UBL 2.0 (common) en het daadwerkelijke bericht (maindoc) waarin opgenomen de referenties naar de UN-Cefact Core Components.

**xsdrt**

Dit is de directory voor de XML Schema’s van UBL 2.0 (common) en het daadwerkelijke bericht (maindoc) zonder referenties naar UN-Cefact Core Components. Deze schema’s zijn bedoeld voor de run-time omgeving.

## Leesbare documenten

 **Implementatiewijzer eFactuur NL versie.doc**

Dit is deze Implementatiewijzer.

 **eFactuur Functioneel Ontwerp versie.doc**

Dit is het functionele model van de eFactuur. Zie hoofdstuk 3 voor nadere toelichting van dit document.

**eFacctuur NL Invoice versie.doc**

Dit is de functionele beschrijving van de eFactuur, uitgebreid met technische invulling als datatypen, business rules die van toepassing zijn op een individueel element, codelijst verwijzingen en UBL 2.0 element namen en hiërarchie.

## Modelleringuitgangspunten

Het functionele model wordt gezamenlijk met de markt vastgesteld. Het gaat daarbij om de gebruikte terminologie en de definities van de gebruikte termen. Om enige structuur in de vastgestelde gegevens aan te brengen worden de gegevens in groepen (UML: classes, ERD: entiteiten) ondergebracht. De groepen worden in een relationele verhouding tot elkaar gezet. Het geheel wordt in een Word document verwoord en in een UML class diagram inzichtelijk gemaakt.

De functionele gegevensverzameling wordt daarna uitgebreid met de door CEN verplicht geachte gegevens. Deze kunnen dus niet ‘terug’ gemapt worden op de functionele gegevensverzameling.

Met de voor CEN uitgebreide lijst van gegevenselementen wordt naast de UBL syntax gelegd. Hierdoor ontstaan eveneens (technisch) noodzakelijke uitbreidingen van de bevraagde gegevens. Ook deze uitbreiding kan niet ‘terug’ gemapt worden op zowel CEN als de functionele gegevensverzameling.

UBL maakt gebruik van XML Schema 1.0 van de W3C. Zij gebruikt echter niet de datatype restricties die soms logisch gedefinieerd zijn. Bv. Een Nederlandse postcode heeft een strak patroon van vier posities numeriek en twee posities hoofdletters. Een dergelijk controle kan met UBL niet in het schema tot uitdrukking gebracht worden en dient derhalve als een business rule opgenomen te worden. Dit leidt tot een tweedeling in de soorten business rules:

* Technische; hieronder worden datatype beperkingen en cardinaliteitseisen verstaan.
* Semantische; hieronder worden regels die over meer dan een element heen gesteld worden, verstaan.

Met de keuze van UBL zijn de gegevenselementen voorgeschreven, in het Engels. Ter ondersteuning van de implementatie is er voor gekozen alle benodigde UBL elementen tevens van een Nederlandse vertaling te voorzien. Deze zijn terug te vinden in de bericht documentatie. De keuze voor UBL stelt tevens dat met de syntax van de business rules gekozen wordt voor Schematron en met de implementatie van codelijsten voor OASIS GeneriCode. Logius heeft gekozen voor de ISO implementatie van Schematron.

UBL kent voorts nog een specifieke functie voor het omgaan met sleutelwaarden (bv. Factuurnummer of BTW nummer). Omdat UBL een internationale standaard is, moet rekening gehouden worden met alle mogelijke sleutelwaarden. Daarom wordt van elke sleutelwaarde gevraagd wie deze sleutelwaarden toekent. In deze gevallen is door Logius voorgeschreven welke instanties toegestaan zijn. Dit komt tot uitdrukking in de bericht documentatie en in de Schematron business rules.

UBL kent twee statussen van met UBL gedefinieerde berichten; conformant en compatibel. Bij conformant berichten worden uitsluitend UBL gedefinieerde elementen gebruikt, bij compatibel kunnen ook partij specifieke elementen opgenomen worden. Het spreekt voor zich dat Nederland specifieke elementen een internationale uitwisseling bemoeilijken. Daarom stelt CEN zich op dit terrein terughoudend op. Logius heeft gekozen voor de conformant variant.

## Gegenereerde documenten

Alle documentatie in de vorm van XML en de berichtspecificaties worden door een tool gegenereerd. Dat betekent aan de ene kant consistentie van de verschillende documenten maar aan de andere kant dat handmatige wijzigingen in de documenten bij een volgende versie automatisch verloren gaan. Alleen de mapping en het functionele model worden buiten de Logius tooling bijgehouden.

# Functioneel ontwerp eFactuur-NL

**Het** functioneel ontwerp beschrijft het bericht op functioneel niveau. Dat wil zeggen dat de hiërarchie in het model wordt vastgelegd in termen van (geneste) gegevensgroepen (entities, classes of objecten) en gegevenselementen (attributen, elementen), herhalingsfactoren en eventuele waardebereiken indien die ‘altijd’ geldig zijn. De hiërarchie wordt gevormd door gegevensgroepen in andere gegevensgroepen op te nemen (te nesten). Er is altijd één “top” gegevensgroep, ie. een gegevensgroep die niet in een andere groep is opgenomen. Deze groep heet **Bericht**.

## De delen van het functioneel ontwerp.

In het functioneel ontwerp zijn de volgende onderwerpen te onderkennen die in de paragrafen hieronder besproken worden.

### De samenvatting

Hierin wordt de opbouw van de hiërarchie van gegevengroepen kort weergegeven. Of een entiteit in de hiërarchie verplicht of optioneel is, wordt aangegeven met de statuscode (R = required / verplicht; O = optional/optioneel; D = dependent/afhankelijk van rule; X = wederzijds exclusief).

### De detailspecificaties

Hier worden de gegevensgroepen en de daarin opgenomen gegevenselementen uitvoerig beschreven. De volgorde waarin de groepen behandeld worden komt overeen met de volgorde in de samenvatting. Of een gegeven verplicht of optioneel is, wordt aangegeven met de statuscode (R = required / verplicht; O = optional/optioneel; D = dependent/afhankelijk van rule). Elk waardebereik van een gegevenselement wordt zeer nauwkeurig gespecificeerd als domain/domein.

### De domeinspecificaties

Elk gegevenselement heeft een bepaald waardebereik. Dit waardebereik is gebaseerd op een in het gegevensmodel vastgelegd domein. De domeinen die op het bericht van toepassing zijn worden in dit onderdeel gespecificeerd. Omdat domeinen zijn vastgelegd in het gegevenmodel, is de specificatie van een bepaald domein voor alle berichten hetzelfde. Wel kan het zijn dat aan een specifiek gegevenselement slechts een deel van het waardebereik van een domein toegewezen is. Er is dan sprake van een zogenaamde “codelijst subset”. In sectie 3.2.3 wordt uitgebreider op domeinen en hun relatie tot gegevenselementen ingegaan.

### De rulespecificaties

Hier worden eventuele nadere regels waaraan het element en met name de inhoud ervan moet voldoen weergegeven. Een voorbeeld hiervan is een afhankelijkheid tussen gegevens als in “als A gegeven is moet ook B gegeven zijn”. In de rules worden de betrokken elementen benoemd. Eisen m.b.t. optionaliteit worden niet met rules maar met een aparte rubriekeigenschap “Status” gespecificeerd.

Eisen waarbij op het niveau van gegevensmodel beperkingen worden gesteld aan de waarden worden gespecificeerd binnen de domeinen. Hierbij gaat het om ondermeer toegestane waarden uit een waardebereik. Rules worden gebruikt om aanvullende eisen op te geven die niet in domeinspecificaties, optionaliteit of cardinaliteit zijn uit te drukken.

## Nadere uitleg

Over het algemeen spreken de onderdelen voor zich. Hier onder worden alleen enkele zaken behandeld die nadere uitleg behoeven.

### Onderdelen met dezelfde oorsprong

De onderdelen (gegevensgroepen en –elementen) vormen een (semantisch) gegevensmodel. Verschillende onderdelen kunnen zich manifesteren als afgeleide van hetzelfde modelelement. Voor deze onderdelen betekent dit dat sommige eigenschappen, zoals de definitie uit het model (entity description) en het domein, altijd identiek zijn, terwijl andere, zoals de status, de beschrijving binnen het onderdeel (transaction description), het voorkomen van subelementen en de toegestane codelijst, wel verschillend kunnen zijn.

Bij het lezen van de specificatie moet hiermee rekening gehouden worden: twee onderdeel specificaties kunnen ogenschijnlijk identiek, maar op kleine details toch net verschillend zijn.

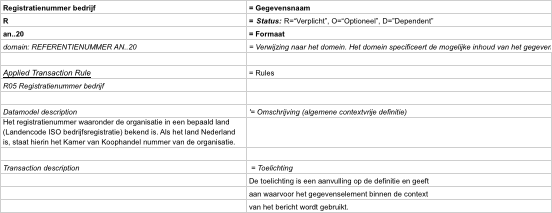
### Applied rules

In de detailspecificaties worden referenties opgenomen onder de kopjes *Applied rules* naar de verderop in het document uitgebreid gespecificeerde rules. In tegenstelling tot wat de naam *Applied* suggereert, gaat het hierbij niet om alleen die rules die op het gegeven van invloed zijn, maar om *alle* rules waarin het gegeven genoemd wordt. Dus bijvoorbeeld de rule 0001: “Als A gegeven is moet B ook gegeven worden” wordt zowel bij A als bij B genoemd.

### Formaten en domein specificaties

In de detailspecificaties wordt voor ieder gegevenselement (attribuut) een formaat gegeven en bovendien wordt aangegeven welk domein (waardebereik) van toepassing is. Ieder domein heeft een basistype (Alfanumeriek, Integer, Date, etc.) en een al dan niet variabele lengte. Daarnaast kan er een codelijst aan het domein gekoppeld zijn en kunnen er voor numerieke domeinen ook minimum en maximum waarden bepaald zijn. Verder is het nog mogelijk dat een patroon (pattern) is gegeven volgens welke de waarde van het gegeven opgebouwd moet worden. Het formaat is eigenlijk niet meer dan een korte, snel leesbare benadering van wat in het domein is vastgelegd. Er wordt ook niet gegarandeerd dat het formaat een zo nauw mogelijke representatie van het domein is. Zo kan bijv. an3 formaat gegeven worden terwijl de bijbehorende codelijst alleen numerieke codes kent (001, 002, etc.) of het patroon alleen numerieke waarden toelaat (\d{3}). Een formaat geeft kortom niet meer dan een, veelal te ruime, indicatie van het werkelijke waardebereik.

## Voorbeeld

In onderstaand voorbeeld wordt in de linkerkolom een rubriek uit de detailspecificatie van het functioneel ontwerp weergegeven en in de rechterkolom aangegeven wat de betekenis van elk onderdeel is.

# Mapping document

Het voornaamste doel van het mapping document is om de toewijzing van XML-tags aan de elementen van het “Functioneel Ontwerp eFactuur-NL” vast te leggen. De mapping is vorm gegeven in een Excel bestand door de diverse bronnen (logisch model, CEN, UBL en het Nederlandse bericht) naast elkaar te zetten. Van elk van de bronnen wordt aangegeven wat de cardinaliteit is. Aangezien CEN, UBL en het NL bericht dezelfde elementnamen gebruiken zijn deze in één kolom vertegenwoordigd. De logische elementnamen zijn eveneens apart weergegeven.

## Berichttags

De XML-tags zijn terug te vinden door de xPath expressie van de UBL elementen te volgen in het UBL Invoice schema.

## Cardinaliteit

De cardinaliteit afleiding voor het NL bericht werkt als volgt:

Een verplichting die genest is onder een optionaliteit (adres is optioneel, postcode verplicht) houdt in dat het hoogste niveau van de nesting bepaald of onderliggende verplichtingen in het bericht terecht zullen komen.

Als het logisch model stelt dat iets verplicht is heeft dit de hoogste prioriteit en is de oplossing in het bericht ook verplicht. Als het CEN stelt dat iets verplicht is, dan is dat in het bericht ook verplicht. Als UBL stelt dat iets verplicht is, dan is dat in het bericht ook verplicht. Verplichtingen uiten zich in de waarden: 1-1 of 1-N. Herhalingen uiten zich in de waarden: 0-N of 1-N. Optionaliteit uit zich in de waarden: 0-1 of 0-N. Niet toegestaan uit zich in de waarde: 0-0

## xPath

xPath is een W3C standaard om XML elementen in een nesting tot uitdrukking te brengen. Een dubbele punt wordt gehanteerd om de namespace prefix te scheiden van de elementnaam. Een slash teken (/) wordt gebruikt om een ‘kind’ relatie aan te duiden. Een @ teken wordt gebruik om een XML attribuut (i.p.v. een element) aan te geven.

Voorbeeld:

cac:AccountingSupplierParty/ID/@schemeAgencyName

Dit betekent:

cac = namespace prefix, een verwijzing naar een schema waarin het volgend element gedefinieerd is;

AccountingSupplierParty = de elementnaam

ID = een kind element van AccountingSupplierParty

schemeAgencyName = een kind attribuut van ID.

## Bijzonderheden

*Valuta*: UBL kent op elementniveau een currencyID attribuut, dit is verplicht en moet dus door Nederland gevolgd worden. Daarnaast kent UBL een element op het hoogste niveau (cbc:DocumentCurrency). In het logisch model wordt om een éénmalige vermelding van de valuta verzocht. Door het juiste UBL element op te nemen wordt het echter mogelijk tegenstrijdige meldingen te maken: het document is in EUR, terwijl een individueel bedragelement voorzien is van USD als valuta. Om deze situatie te voorkomen wordt NIET met een documentvaluta gewerkt. Indien gewenst kan er gecontroleerd worden of alle gebruikte valuta dezelfde zijn maar dit sluit specificaties in andere valuta uit.

*Huisnummer en Huisnummertoevoeging*: in de fysieke adressen kennen geen verschillende elementen in UBL. Op voordracht van de branche is gekozen voor een concatenatie van de beide waarden waarbij deze gescheiden moeten worden met een ‘-‘ afbreekstreepje. Daarvan mag er maar één in het element BuidlingNumber voorkomen.

*KvK en belastingdienst:* Voor Nederlandse bedrijven zijn de Kamer van Koophandel en de Belastingdienst de enige toegestane instanties voor de UBL velden:

[Party/PartyIdentification/ID@schemeAgencyName](mailto:Party.PartyLegalEntity.CompanyID@schemeAgencyName) (binnen AccountingCustomerParty, BuyerCustomerParty) en Party/PartyIdentification/ID@schemaAgencyName (binnen AccountingSupplierParty, SellerSupplierParty).

*Adresgegevens*: in een enkele partij MOETEN uit hetzelfde land afkomstig zijn.

# XML, Schematron en GeneriCode

## Beperkte uitdrukkingskracht

Een XML-schema is een syntax of grammatica specificatie. Het bepaalt onder andere de hiërarchische structuur van het bericht en de toegestane waarden van ieder XML element. Maar er zijn nog andere “regels” waaraan het gespecificeerde bericht moet voldoen. Voorbeelden daarvan zijn de regels die afhankelijkheden tussen meerdere elementen leggen of de beperkingen van berichtonderdelen die volgen uit beschrijvingen zoals die van de 11-proef voor een Burgerservicenummer. Dergelijke regels kunnen niet in een schema worden uitgedrukt. Hiervoor wordt ISO SchemaTron gebruikt. Daarnaast zijn er nog inhoudelijke eisen aan de gegevens die niet automatisch gecontroleerd kunnen worden.

Dat een bericht aan het schema voldoet (valideert), betekent dus niet dat het een valide bericht is. De SchemaTron validatie geeft al meer zekerheid maar de inhoudelijke eindcontrole vindt plaats bij de ontvanger.

## UBL XML-schemas

In de UBL XML-schemas zijn alle losse elementen, alle groepen en allerlei standaard

hiërarchische structuren in aparte schema’s gestandaardiseerd. Binnen het Invoice schema

worden deze standaard elementen en hiërarchische structuren aangeroepen. Door het volgen

van de referenties naar de andere UBL 2.0 schema’s kan de volledige hiërarchie van het UBL

Invoice bericht worden herleid.

## SchemaTron bestanden

Deze bevatten de “regels” die altijd gelden op een bericht. Regels die alleen van toepassing zijn voor een enkele ontvanger zullen niet aan de markt uitgeleverd worden door Logius.

De regels in de SchemaTron bestanden verwijzen middels een xPath 2.0 expressie naar de elementen waarop zij van toepassing zijn. De controle wordt tevens vermeld in de documentatie.

## GeneriCode

UBL levert standaard een aantal GeneriCode bestanden mee voor de controle van waarden in specifieke elementen die van een codelijst gebruik maken. Voorbeeld is de ISO Landencode. In sommige gevallen maakt het Nederlandse bericht gebruik van eigen (Nederlandstalige) waarden. Bijvoorbeeld voor de InvoiceCode. Deze waarden worden in een eigen GeneriCode bestand ingebracht. Deze bestanden zijn te herkennen aan de prefix NL in de bestandsnaam.

## Generatie

Alle in dit hoofdstuk genoemde bestanden worden gegenereerd door een softwarematige oplossing. De beperkingen die dit met zich meebrengt wegen niet op tegen de consistentie die de documentatie en noodzakelijke technische bestanden met zich meebrengt. Het is vanwege dit generatieproces dat wijzigingen in de technische bestanden nadrukkelijk NIET toegestaan wordt anders dan om een technische aanroep mogelijk te maken.

Gewenste wijzigingen op de bestanden moeten aangegeven worden bij Logius functioneel beheer Meta data.