

DATAMODELLERING XML SCHEMA DEFINITIONS

Inleiding

In dit whitepaper wordt de datamodelleervorm XML Schema Definition (XSD) beschreven. Deze modelleervorm staat in verhouding tot een aantal andere modelleervormen. Wil je een beeld krijgen van welke modelleervormen er zijn bekijk dan het whitepaper wat een introductie geeft tot datamodelleervormen en deze serie van whitepapers zie Inleiding datamodellering.

Dit whitepaper is een onderdeel van meerdere whitepapers over modellering in de fysieke laag. XSD modellering is specifiek voor het modelleren van berichtuitwisseling op basis van gestructureerde berichten gebaseerd op XML. Daarnaast zijn modelleerwijzen rond JSON maar ook de opslag van gestructureerde data (ER) relevant. Informatie over deze modelleervormen is te vinden in deze serie van whitepapers.

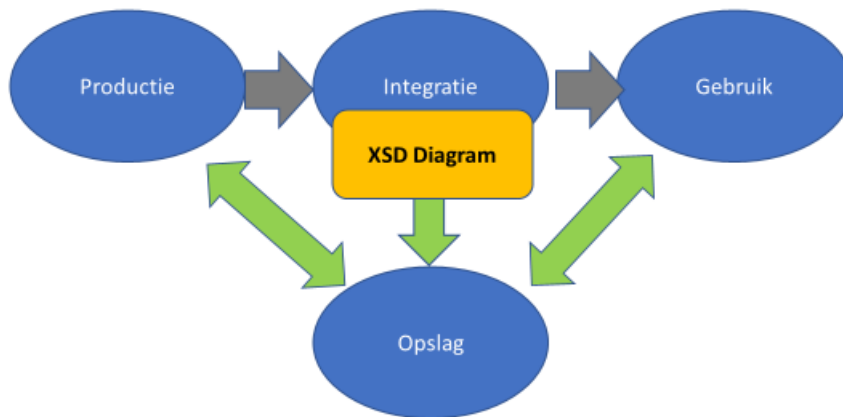
Doel

XSD diagrammen geven een gedetailleerde notatiewijze voor het modelleren van berichten. Deze berichten zijn gebaseerd op een structuur of Definition. Met behulp van deze structuur kunnen berichten die uitgewisseld worden gevalideerd worden op correctheid. Dus hoe meer structuur we kunnen toevoegen aan deze berichtdefinities hoe beter we op correctheid kunnen controleren. Dit maakt het mogelijk om in een vroeg stadium van integratie te voorkomen dat een incorrect bericht wordt uitgewisseld.

XSD diagrammen hebben tot doel om tot in detail de fysieke structuur van enerzijds één berichtdefinitie te modelleren. Anderzijds worden deze diagrammen ingezet om een canonic model te introduceren. Dit canonicke modellen zorgen voor berichtstandaardisatie en het hergebruik van schematronen in de verschillende berichten.

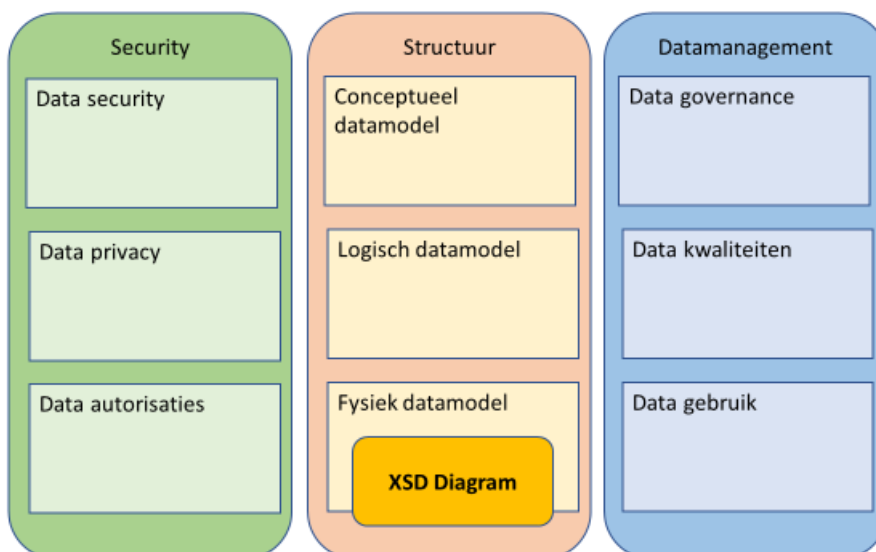
Raamwerken

DATALEVENSLOOP



Vanwege de focus op data integratie (meer specifiek voor het uitwisselen van berichten) wordt deze notatiewijze alleen op de data integratie afgebeeld in de data levensloop. XSD diagrammen zijn heel specifiek gericht op het beschrijven van de structuren en -bependingen van berichtuitwisseling op basis van XML. Sinds enige tijd is daar de JSON notatie bijgekomen welke grote overeenkomsten heeft op het gebied van de structuur van een bericht.

DATARAAMWERK



In het data raamwerk zien we het XSD diagram alleen terug in de datastructuur kolom. In het beschrijven van fysieke data structuren is het XSD diagram een zeer goed toepasbare notatiewijze die goed aansluit bij de behoefte van de stakeholders.

Het XSD diagram is met name geschikt voor de fysieke data modellering en wordt daar ook veelvuldig toegepast. Dit vanwege de mogelijkheden om de fysieke structuur van een berichtstructuur tot in detail te beschrijven in een XSD diagram. Inclusief de specifieke aspecten van XML berichten en in een aantal gevallen extra dimensies van een bepaalde berichtuitwisseling standaard (CIM/UN-CEFACT/UBL), zoals specifieke datatypes datastructuren of specifieke beperkingen zoals regular expressions, domeinen en enumeratie implementaties.

Stakeholders

Fysieke modellering met XSD richt zich op stakeholders die zich voornamelijk in het ICT werkveld bevinden, meer specifiek bij data integratie op basis van berichtuitwisseling. Vanwege de technische details is deze modelleerwijze minder geschikt voor interactie met stakeholders binnen de business.

Fysieke datamodelering dient dan ook zorg te dragen dat de implementatie in een integratie omgeving tot in detail beschreven is en desgewenst een geautomatiseerde transformatie van model naar implementatie mogelijk is.

Onderstaande opsomming geeft de belangrijkste stakeholders:

- **Systeem integratie specialist**, een groep stakeholders die diepgaande kennis heeft van (bericht) uitwisseling, integratieplatformen zoals servicebussen, gateways etc en het standaardiseren van berichtdefinities.
- **(Data) integratie- of Informatiearchitect**, zal veelal kennis hebben van het gestandaardiseerde fysieke datamodel in de huidige- en de gewenste situatie van een geïmplementeerde applicatieverandering. Ze hebben daarnaast veelal diepgaande kennis van data integratie, data patronen, data structuren en data modellen.
- **Ontwikkelaars**, software ontwikkelaars met name van front end en back end systemen dienen kennis te hebben van de gestructureerde uitwisseling van gegevens waarbij XML of JSON het uitwisselformaat is.

Concepten

Binnen XSD diagrammen worden voornamelijk fysieke datamodellen weergegeven in een verrijkte graaf. XSD diagrammen zijn opgebouwd uit de volgende concepten:

- **XSD Complextype**: is het hoofdconcept. Een complextype kenmerkt zich dat het is opgebouwd uit andere onderdelen XSD Element of XSD attribute. Daarnaast kan een XSD complextype een relatie hebben met andere XSD complextypes. Kenmerkend is dat dit altijd een boomstructuur als opbouw heeft. Er dient dus bij een berichtdefinitie altijd één root element te zijn. Voor canonieke datamodellen op basis van XSD is deze boomstructuur niet vereist
- **XSD Element**, zijn de onderdelen van een XSD complextype die ervoor zorgdragen dat de eigenschappen uit een klasse (in het logische model) binnen een bericht tabel van elkaar

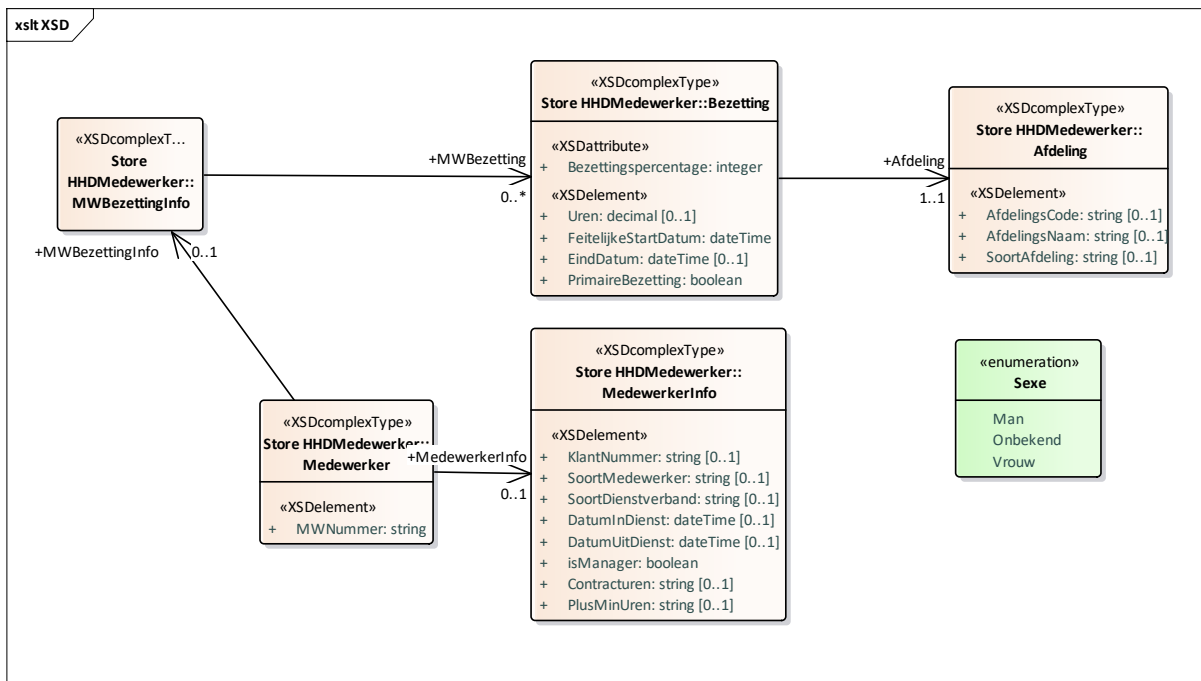
onderscheiden kunnen worden. Elementen worden gebruikt om complexe attributen te modelleren en bieden hiervoor dan ook een aantal extra validaties, types en stereotypes.

- **XSDAttribute**, is vergelijkbaar met een XSD element en zorgt voor de detaillering van een complextype. Een attribute is eenvoudiger van opzet dan een element.
- **Relaties**, tussen de verschillende elementen binnen een bericht kunnen een aantal soorten relaties gelegd worden, met name de richting van de relatie is hierbij van groot belang, ook vanwege het feit dat een bericht een boomstructuur dient te hebben
- **Restricties**, in XML schema's kunnen een groot aantal restricties ingesteld worden met name op de waarde van elementen en attributen. Een aantal voorbeelden zijn domeinen, enumeraties, regular expressions, ranges en datatypes

Op basis van deze basis concepten zijn zeer krachtige modellen op te stellen die zorgdragen voor berichtuitwisseling van data die aan bepaalde kwaliteitseisen op het vlak van correctheid, precisie en validiteit voldoen. Dit alles met als reden te voorkomen dat de berichtuitwisseling bijdraagt aan interpretatieverschillen bij data integratie.

Notatie

Het XSD diagram kent twee verschijningsvormen. Enerzijds is het een verrijkte graaf waarbij met name de relaties een aantal verrijkingen kent die de taal de zeggingskracht geeft. In onderstaande afbeelding een voorbeeld van een XSD diagram.

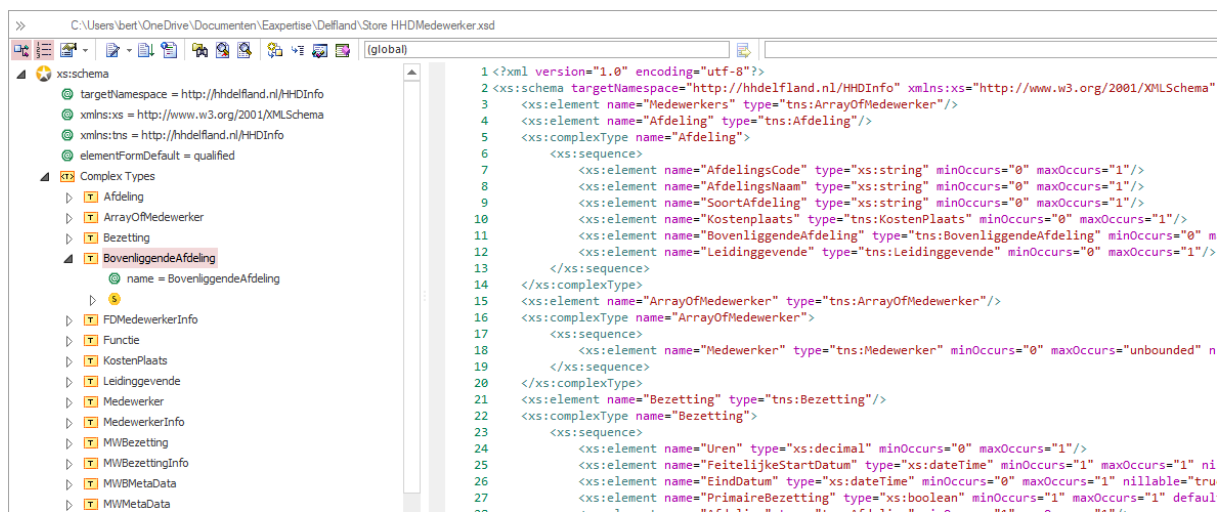


In de afbeelding is te zien dat een XSDcomplexType uit een naam en een aantal elementen en eventueel attributen. Voor de complextypes en de attributen wordt het data type getoond worden en de cardinaliteit `[0..1]` als het optioneel is of verplicht `[1..1]`, of dat een element een reeks is `[0..*]`.

Voor de relaties zijn er meerdere extra kenmerken te zien. Bijvoorbeeld welke rol het complextype vervuld in de relatie met het bovenliggende element veelal omgezet naar een datatype in de uiteindelijke berichtdefinities. Daarnaast wordt voor de relatie aangegeven wat de cardinaliteit is. Hiermee kan een bericht validatie zorgdragen voor het correct houden van de relaties tussen elementen binnen het bericht.

Als laatste zijn de verschillende beperkingen zoals enumeraties, reeksen, reguliere expressies te noemen. Dit zijn beperkingen in berichten die gemodelleerd kunnen worden in een XSD diagram. In bovenstaande afbeelding zijn een beperkt aantal deze constraints opgenomen als voorbeeld.

Naast het XSD diagram gebaseerd op een graaf zoals hierboven wordt er voor een XSD bericht ook veelvuldig gebruik gemaakt van een boomweergave. Onderstaande afbeelding geeft een voorbeeld.



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <xs:schema targetNamespace="http://hhdfland.nl/HHInfo" xmlns:xs="http://www.w3.org/2001/XMLSchema"
3 <xs:element name="Medewerkers" type="tns:ArrayOfMedewerker"/>
4 <xs:element name="Afdeling" type="tns:Afdeling"/>
5 <xs:complexType name="Afdeling">
6 <xs:sequence>
7 <xs:element name="AfdelingsCode" type="xs:string" minOccurs="0" maxOccurs="1"/>
8 <xs:element name="AfdelingsNaam" type="xs:string" minOccurs="0" maxOccurs="1"/>
9 <xs:element name="SoortAfdeling" type="xs:string" minOccurs="0" maxOccurs="1"/>
10 <xs:element name="KostenPlaats" type="tns:KostenPlaats" minOccurs="0" maxOccurs="1"/>
11 <xs:element name="BovenliggendeAfdeling" type="tns:BovenliggendeAfdeling" minOccurs="0" maxOccurs="1"/>
12 <xs:element name="Leidinggevende" type="tns:Leidinggevende" minOccurs="0" maxOccurs="1"/>
13 </xs:sequence>
14 </xs:complexType>
15 <xs:element name="ArrayOfMedewerker" type="tns:ArrayOfMedewerker"/>
16 <xs:complexType name="ArrayOfMedewerker">
17 <xs:sequence>
18 <xs:element name="Medewerker" type="tns:Medewerker" minOccurs="0" maxOccurs="unbounded" n
19 </xs:sequence>
20 </xs:complexType>
21 <xs:element name="Bezetting" type="tns:Bezetting"/>
22 <xs:complexType name="Bezetting">
23 <xs:sequence>
24 <xs:element name="Uren" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
25 <xs:element name="FeitelijkeStartDatum" type="xs:dateTime" minOccurs="1" maxOccurs="1" ni
26 <xs:element name="EindDatum" type="xs:dateTime" minOccurs="0" maxOccurs="1" nillable="tru
27 <xs:element name="PrimaireBezetting" type="xs:boolean" minOccurs="1" maxOccurs="1" defaul
28 </xs:sequence>
29 </xs:complexType>
30 </xs:schema>
```

Deze weergave staat heel dicht bij de fysieke implementatie van een bericht. Echter met name de boomweergave biedt de ontwikkelaar de mogelijkheid om zich snel een beeld te vormen van de structuren. Door delen van het bericht open – of dicht te klappen wordt analyse sterk vereenvoudigd.

Kenmerken

XSD diagram heeft de volgende kenmerken:

- Een breed geaccepteerde standaard
- Rijke notatie met name voor XML berichtmodellering
- Interactie met leveranciers wordt eenvoudiger vanwege standaardisatie
- Het genereren van diverse andere output zoals XSD definitie bestanden, berichtenboeken, eventueel WSDL's
- Notatie is rijk en modellen kunnen complex worden
- Grote mate van detaillering
- Mogelijkheid om bestaande berichtdefinities geautomatiseerd om te zetten naar een XSD modellen en omgekeerd.
- Door de opkomst van nieuwe (eenvoudiger) talen ontstaan nieuwe definities
- Het modelleren van relaties met niet relationele entiteiten is niet mogelijk in modellering
- Moeilijk te begrijpen voor Niet ICTers

Gebruikstoepassingen

Het XSD diagram wordt vooral toegepast binnen de fysieke data modellering en in een aantal gevallen binnen de logische modellering. Binnen dit werkveld is het een zeer krachtige en veel gebruikte modelleerwijze. Het heeft de mogelijkheid om voldoende bericht specifieke details toe te voegen (met name bij de relaties, de complextypes en de constraints).

Houdt er rekening mee dat ondanks de eenvoud van de notatiewijze het opstellen van een XSD diagram een complex traject kan zijn, zeker bij een complex ketenintegratie domein of binnen een organisatiecontext waar rond de informatievoorziening en logische modellering weinig volwassenheid is.

Gerelateerde notatiewijzen

Deze ER notatie heeft met veel notatiewijzen, met name in de fysieke en logische modellering een relatie. In onderstaande opsomming een overzicht:

- **ArchiMate Datamodel** of begrippenlijst en -boom, hiermee worden de verbanden gelegd vanuit het conceptuele datamodel naar het logische datamodel
- **UML klassediagram** waarmee een koppeling gelegd wordt naar het logische model in UML klassediagram vanuit een fysieke opslagstructuur in een relationele database
- **ER model** waarmee een verband wordt gelegd van het fysieke datamodel (veelal via een logisch model in UML) naar een fysieke inrichting voor relationele databases.
- **ArchiMate applicatiemodellering**, voor het modelleren welke applicatie, interface of component gebruik maakt van een bepaalde berichtdefinitie.

Tooling

Voor de ER diagrammen zijn een aantal algemene en specifieke tools aanwezig:

- Altova suite
- Sparx Enterprise Architect
- Visio met Schema extensie

Evaluatie

XSD diagram is een veelgebruikte notatiewijze met name voor het opstellen van fysieke datamodellen voor implementatie in berichtuitwisseling. Het legt daarmee een verbinding tussen de logische modellen en de fysieke implementatie in een uitwisselstandaard voor berichten. Het is daarmee een onmisbare schakel in de data modelleerketen op het vlak van data integratie.

Het XSD diagram wordt met name toegepast binnen systeem en data integratie. De basisnotatie biedt een ruime hoeveelheid mogelijkheden om complexe modellen op te stellen. Dit is enerzijds de kracht van het XSD diagram en anderzijds een zwakte omdat de modellen veelal te complex zijn voor stakeholders met minder modelleerervaring.

Voor XSD diagrammen is een veelheid aan tooling aanwezig, in dit artikel slechts een beperkte opsomming.

Over de auteur



Bert Dingemans is trainer op het vlak van data architectuur, data management en Big Data. Hij heeft een passie voor modelleren, modelleertools en het effectief inzetten van geautomatiseerde hulpmiddelen om modellen effectief in te zetten in de praktijk. Bert is te bereiken via bert@interactory.nl