

DATAMODELLERING GEAVANCEERD UML KLASSEMODEL

Inleiding

In dit whitepaper wordt de datamodelleervorm geavanceerd UML klassemodel 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 UML klassemodellen. Het klassemodel wordt in veel domeinen toegepast en kent meerdere verschijningsvormen. Vandaar een aantal whitepapers die elk een verschillend gezichtspunt van de UML klassenotatie behandelen. Dit whitepaper is het tweede en gericht op geavanceerd notatievormen van UML klassediagrammen. Het sluit aan op het whitepaper over basis UML diagrammen.

Doel

UML is een modelleertaal die voornamelijk in de softwareontwikkeling gebruikt wordt. UML kent veel verschillende diagramtechnieken die in meer of mindere mate met elkaar in verband staan. Een aantal veel toegepaste diagrammen uit UML zijn:

- **Klassediagram**
- **Objectdiagram**
- Sequence diagram
- Component diagram
- Use case diagram

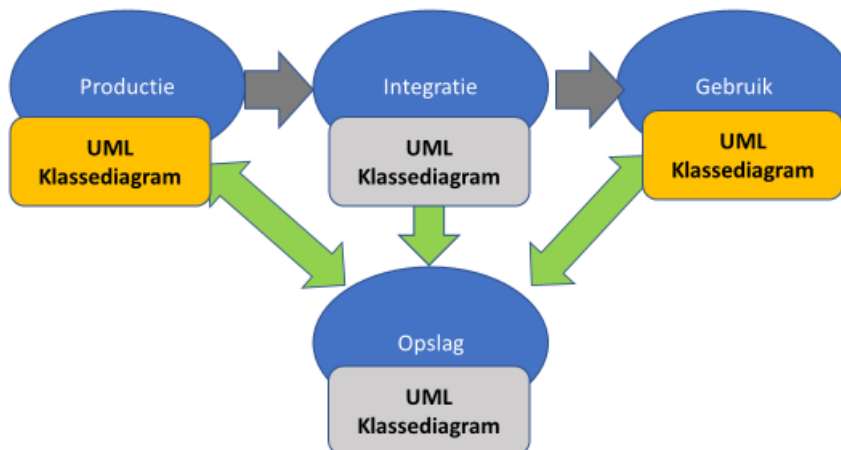
Voor het modelleren van informatie of data zijn eigenlijk alleen het klassediagram en het objectdiagram van belang. Deze worden in deze en eerdere whitepapers over UML in detail behandeld.

Het UML klassediagram heeft tot doel om de structuur van gegevens te beschrijven. Hierbij is het van belang dat het uitgangspunt is dat het de structuur van gegevens beschrijft binnen software en dus minder rekening houdt met hoe de data opgeslagen moet worden.

De geavanceerde UML klassenotaties zijn met name gericht op het toevoegen van extra kenmerken aan de basis UML klassenotatie. Het doel hiervan is om de diagrammen die ontstaan rijker te maken aan notatie, meer zeggingskracht te geven en zorg te dragen voor hergebruik en standaardisatie.

Raamwerken

DATALEVENSLLOOP



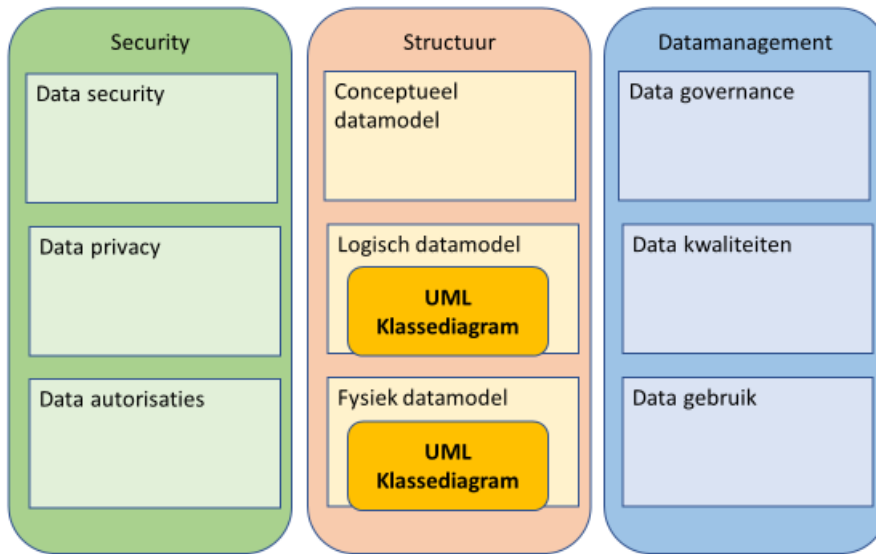
Binnen de data levensloop wordt het geavanceerde UML klassediagram met name ingezet voor het modelleren van de data binnen het datagebruik. Hier wordt een logisch model ingezet voor het beschrijven van de structuur van de data binnen het toepassingsgebied en zorg te dragen voor het hergebruik van elementen waar dit relevant is. Het klassediagram heeft een aantal kenmerken die goed aansluiten bij de modellerbehoefte van datagebruik. Met name aggregaties en specialisaties zijn goed inzetbaar binnen modellen van datagebruik.

Ook bij de productie van data worden UML klassediagrammen ingezet. Vooral wanneer de data productie ondersteund wordt door software. Omdat UML dicht bij de software staat is de UML klasse notatie hiervoor geschikt. Bij andere vormen van data productie wordt nog wel eens een andere notatie gebruikt zoals de DFD notatie. Echter tegenwoordig wordt vrijwel elke vorm van data productie op enigerlei wijze ondersteund door software en zal het UML klassediagram dus veelvuldig worden toegepast.

Voor data integratie en data opslag kan gebruik gemaakt worden van UML klassediagrammen. Bijvoorbeeld als bij integratie gegevens in een bepaalde structuur (gestructureerde berichten of bestanden) worden uitgewisseld. Echter in de afbeelding is de weergave grijs gemaakt omdat er andere specifieke notaties zijn die dit krachtiger ondersteunen zoals het XSD diagram.

Bij opslag wordt voor relationele databases gebruik gemaakt van een andere notatie (ER diagrammen). Vandaar dat deze grijs is in de afbeelding. Bijvoorbeeld bij NoSQL databanken kan een UML klassediagram in bepaalde situaties zeer goed worden ingezet.

DATARAAMWERK



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

Het geavanceerde UML klassediagram is voornamelijk gericht op logische data modellering en wordt daar ook veelvuldig toegepast. Veel informatiemodellen van open standaarden maken gebruik van de concepten in de geavanceerde UML klassediagrammen. Dit omdat deze informatiemodellen veelal complex van opzet zijn en met de geavanceerde concepten met name een vereenvoudiging in de modellen realiseren.

Stakeholders

Geavanceerde logische modellering en data gebruik, richt zich op stakeholders die zich voornamelijk op het raakvlak van de business en ICT bevinden. Logische datamodellering dient dan ook zorg te dragen dat beide werelden met elkaar in gesprek gaan en een gemeenschappelijk model ontwikkelen op basis waarvan de behoeften van de business stakeholders worden geïmplementeerd in de software.

Onderstaande opsomming geeft de belangrijkste stakeholders:

- **Functioneel (Applicatie) Beheerders**, een groep stakeholders die diepgaande kennis heeft van (standaard) informatiesystemen en het onderliggende logische datamodel inclusief de door de leverancier gebruikte datastructuren en eventueel opslagmechanismen
- **Applicatie- of Informatiearchitect**, zal veelal kennis hebben van het logische datamodel in de huidige- en de gewenste situatie van een verandering. Ze hebben daarnaast veelal diepgaande kennis van data patronen, data structuren en logische data modellen.

- **Open standaarden**, ontwikkelen van logische informatiemodellen voor open standaarden worden veelal uitgewerkt in geavanceerde modellen
- **Ketenintegraties**, binnen ketenintegratie is een overeenstemming rond de data die uitgewisseld wordt van groot belang. Vaak wordt begonnen met een conceptueel model, echter voor de detaillering van dit model wordt een UML klassediagram ingezet.

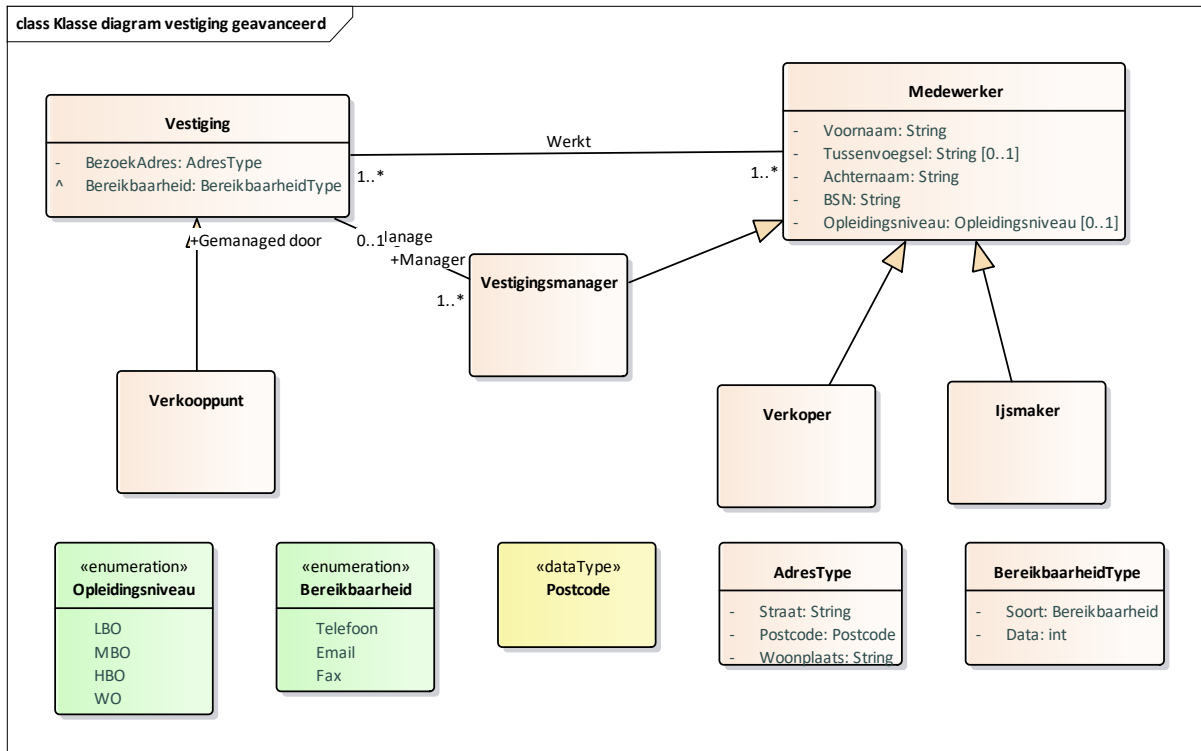
Concepten

Binnen UML klasse diagrammen worden voornamelijk logische datamodellen worden weergegeven in een verrijkte graaf. UML klasse modellen zijn opgebouwd uit de volgende concepten:

- **Klasse** (zie basis UML klassediagram)
- **Attribuut of eigenschap** (zie basis UML klassediagram)
- **Associatie of relatie** (basis UML klassediagram)
 - **Specialisatie of overerving**, door een specialisatie associatie kun je het klassemodel vereenvoudigen omdat eigenschappen en associaties die generieker zijn in één (abstracte) klasse geplaatst worden, waar andere klassen van overerven en daarmee een specialisatie worden. Het introduceert een **Is_een** associatie. Bijvoorbeeld Medewerker is een Persoon
 - **Aggregatie of compositie**, geeft aan dat er een klasse een deel is van een andere klasse. Met name bij het omzetten van een klassemodel naar programmatuur is dit om te zetten naar collecties. Het introduceert een **heeft** associatie. Bijvoorbeeld Klant heeft Bestellingen.
- **Enumeratie**, maakt het mogelijk om voor attributen of eigenschappen extra beperkingen te geven met een domein van waarden
- **Complex datatype**, hiermee kun je een attribuut of eigenschap uit een samengesteld datatype laten bestaan. Dit samengestelde datatype heeft bijvoorbeeld eigen attributen of extra beperkende regels of bedrijfsregels

Notatie

Het geavanceerde UML klassediagram is net als het basisdiagram een verrijkte graaf waarbij met name de associaties en de attributen een aantal verrijkingen kent die de taal de extra zeggingskracht geeft. In onderstaande afbeelding een voorbeeld van een geavanceerd klassediagram



In de afbeelding is te zien dat het model bestaat uit klassen en een aantal extra elementen zoals enumeraties, datatypes en complex types.

De enumeraties laten zien dat er een generiek domein gedefinieerd kan worden. Deze generieke domeinen worden vervolgens aan één of meer attributen in het UML klassediagram. Bijvoorbeeld de opleidingsniveau enumeratie is gekoppeld aan een attribuut bij de persoon.

De complex datatypes bieden de mogelijkheid om het model enerzijds te vereenvoudigen (de klassen worden minder omvangrijk) en anderzijds hergebruik te introduceren. Bijvoorbeeld het Adrestype is een complex datatype dat is opgebouwd uit meerdere attributen die een adresgegevens beschrijven. Dit wordt vervolgens gebruikt als type van een attribuut. Zie bijvoorbeeld de vestiging. Voordeel is dat het eenvoudiger is dergelijke modellen te beheren. Echter de transformatie van model naar implementatie wordt complexer.

Voor de associaties zijn er twee extra typen associaties specialisatie/generalisatie en aggregatie/compositie. In het voorbeelddiagram zie je een uitwerking van de specialisatie. Een manager is een specialisatie van een medewerker. De manager ontvangt hierdoor alle eigenschappen van de medewerker dus een naam etc. Er blijft een mogelijkheid om de manager eigen associaties te geven naar klassen en daarmee extra beperkingen e.d. te maken, zoals in het diagram van Manager naar Vestiging.

Kenmerken

UML klassennotatie heeft de volgende kenmerken:

- Een internationale en breed geaccepteerde standaard
- Rijke notatie met name voor logische datamodelering
- Interactie met leveranciers wordt eenvoudiger

- Mogelijkheid om overervingsstructuren te introduceren
- Eenvoudiger beheer en aanpassing van de modellen
- Moeilijk te converteren naar een (niet object georiënteerde) implementatie
- Notatie is rijk en modellen worden snel complex
- Kans op te grote mate van detaillering
- Het uitwerken van de richtlijnen voor viewpoints is noodzakelijk
- Het modelleren van relaties met services en processen is niet mogelijk in modellering
- Moeilijk te begrijpen voor Niet ICTers

Gebruikstoepassingen

Het geavanceerde UML klassediagram wordt vooral toegepast binnen complexe- en omvangrijke logische data modellering. Binnen dit werkveld is het een zeer krachtige en veel gebruikte modelleerwijze. Het heeft de mogelijkheid om geavanceerde details toe te voegen (met name bij de attributen en de associaties).

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

Gerelateerde notatiewijzen

Deze geavanceerde UML klassennotatie heeft met veel notatiewijzen een relatie. In onderstaande opsomming een overzicht:

- ArchiMate Datamodeltering of begrippenlijst en -boom, hiermee worden de verbanden gelegd vanuit het conceptuele datamodel naar het logische datamodel
- ER notatie waarmee een koppeling gelegd wordt van het logische model in UML naar de fysieke opslagstructuur in een relationele database
- XSD schema waarmee een verband wordt gelegd van het logische model in UML naar een fysieke inrichting voor gestandaardiseerde berichtuitwisseling.
- Basis UML klassediagram, geavanceerde datamodeltering is een nadere detaillering van het basisdiagram

Tooling

Voor de UML klassennotatie zijn een aantal specifieke tools aanwezig:

- Visio met UML class template
- Visual Paradigm tool
- Lucidchart
- Sparx Enterprise Architect
- BizDesign
- ADO-IT
- ARIS

Evaluatie

Geavanceerde UML klassennotatie is een veelgebruikte notatiewijze met name voor het opstellen van logische datamodellen voor bijvoorbeeld open standaarden. Het geeft een detaillering van de UML basisdiagrammen en introduceert met name hergebruik.

Voor geavanceerde UML klassennotatie is een veelheid aan tooling aanwezig, in dit artikel slechts een beperkte opsomming. Wil je het klassediagram gaan inzetten voor het genereren van programmatuur dan is de tooling keuze minder breed maar nog steeds een ruim voldoende.

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