

DATAMODELLERING BASIS UML KLASSEMODEL

Inleiding

In dit whitepaper wordt de datamodelleervorm basis 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 toepassingsgebieden en kent meerdere verschijningsvormen. Vandaar een aantal whitepapers die elk een verschillend gezichtspunt van de UML klassennotatie behandelen. Dit whitepaper is het eerste en gericht op de basiselementen van UML klassediagrammen.

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

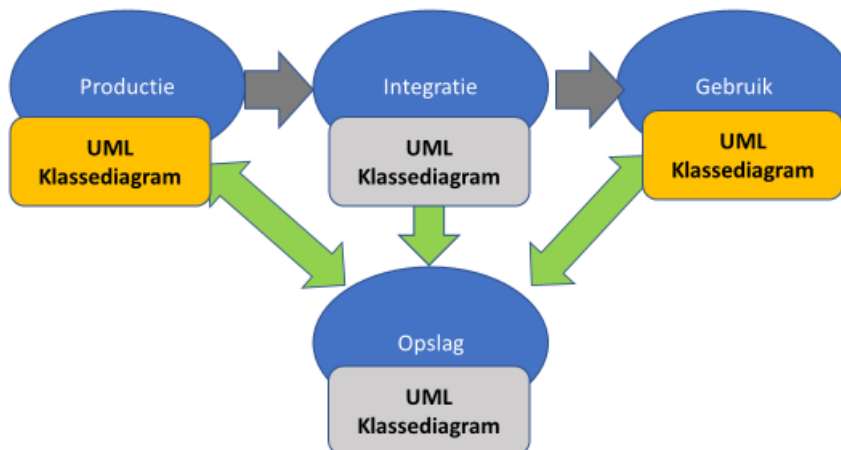
Voor het modelleren van informatie of data zijn eigenlijk alleen het klassediagram en het objectdiagram van belang. Deze worden in deze en latere whitepapers over UML in detail behandeld.

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 of getransporteerd moet worden.

Het UML klassediagram heeft een aantal eigenschappen die ervoor zorgen dat de modellen relatief eenvoudig kunnen blijven (zeker bij basis modellen) maar toch veel zeggingskracht hebben. Dat maakt dat ze geliefd zijn in veel situaties in de informatievoorziening.

Raamwerken

DATALEVENSLOOP



Binnen de data levensloop wordt het 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. Het klassediagram heeft een aantal kenmerken die goed aansluiten bij de modelleerbehoeften van datagebruik. Met name aggregaties en specialisaties zijn goed inzetbaar binnen modellen van datagebruik.

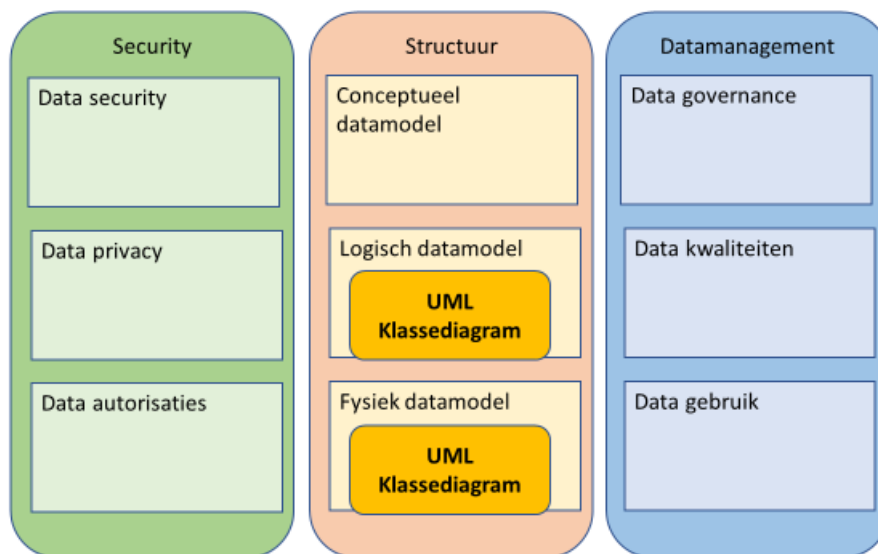
Ook bij de productie van data worden UML klassediagrammen ingezet. Vooral wanneer de data productie ondersteund wordt door software. Omdat UML dicht bij de software staat is de UML klasse notatie hiervoor geschikt. Bij andere vormen van data productie wordt nog wel eens een andere notatie gebruikt zoals 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 dataraamwerk zien we het 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 UML klasediagram is met name geschikt voor de logische data modellering. Dit vanwege de verwevenheid met software. Ook fysieke datamodellering is mogelijk met klasediagrammen. Echter dan met name bij opslag buiten relationele databases. Denk hierbij aan NoSQL (document) databases of het persistent maken van datastructuren binnen een software toepassing.

Stakeholders

Logische modellering en data gebruik, richt zich op stakeholders die zich voornamelijk op het raakvlak van binnen de business en ICT bevinden. Deze notatie dient een verband te leggen vanuit de conceptuele modellen naar de details in fysieke modellen in de implementatie. 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 op juiste wijze worden geïmplementeerd in de software.

Onderstaande opsomming geeft de belangrijkste stakeholders:

- **Gebruikers**, in kleinere domeinen, bijvoorbeeld waarbij applicaties en informatiesystemen worden ingezet kunnen gebruikers vanuit het gebruiksperspectief het logische datamodel in relatie brengen met het door hen gebruikte deel van het applicatielandschap.
- **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.

- **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:

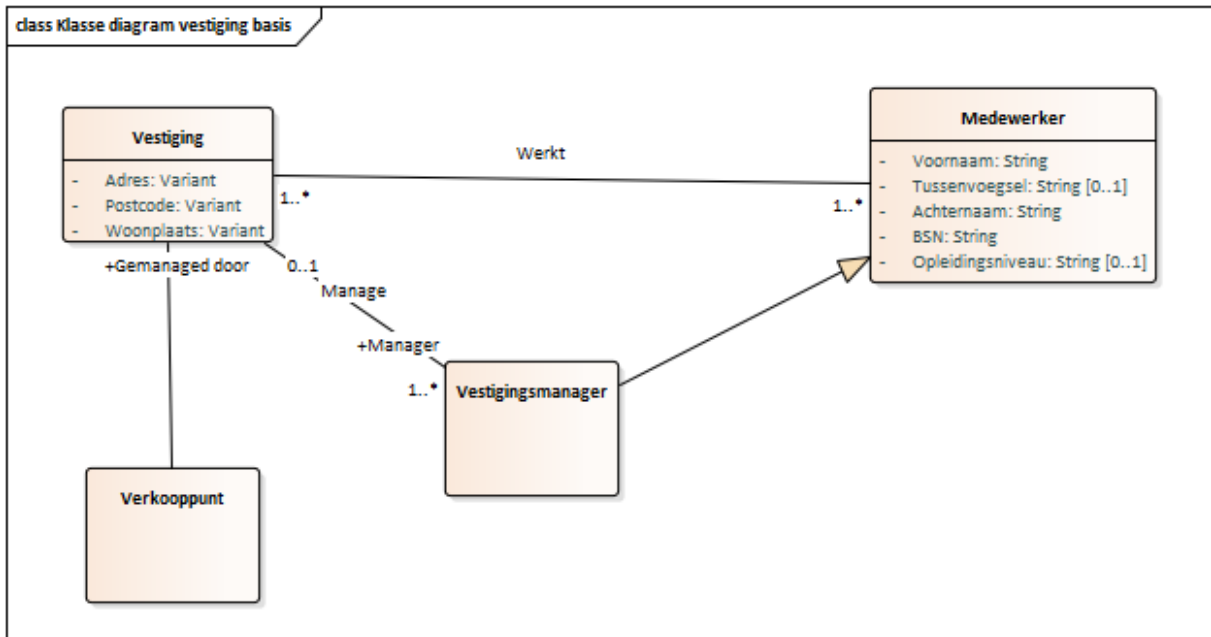
- **Klasse:** is het hoofdconcept. Een klasse kenmerkt zich in dat het is opgebouwd uit attributen en dat het een associatie kan hebben met andere klassen. Klassen zijn in natuurlijke taal meestal een zelfstandig naamwoord. Voorbeelden zijn Persoon, Medewerker of Activiteit
- **Attribuut of eigenschap**, zijn de beschrijvende onderdelen van een klasse en voegen hiermee de detaillering toe aan deze klassen. Voor attributen worden een aantal extra kenmerken beschreven, vaak zijn dat het datatype van het attribuut en of het attribuut optioneel is of niet. Een attribuut is in natuurlijke taal veelal een zelfstandig naamwoord, echter kenmerkend is dat er een overkoepelende entiteit is. Voorbeelden van attributen van een persoon zijn voornaam, achternaam of geboortedatum.
- **Associatie of relatie**, is een verbinding tussen twee klassen en geeft verbanden tussen logische entiteiten weer. Van een associatie worden net als bij eigenschappen extra kenmerken beschreven. Bijvoorbeeld de cardinaliteit en de rol die een klasse binnen de associatie vervuld.

Op basis van deze drie basis concepten zijn reeds zeer krachtige modellen op te stellen, bij de geavanceerde klasse diagrammen gaan we nog een aantal extra concepten toevoegen.

Naast het klasse diagram in de UML basis notatie is het vermelden van het object model relevant. Het object model wordt veelal gebruikt om het klasse diagram concreet te maken met voorbeelden vanuit de organisatie. Bij de notatie paragraaf wordt dit toegelicht.

Notatie

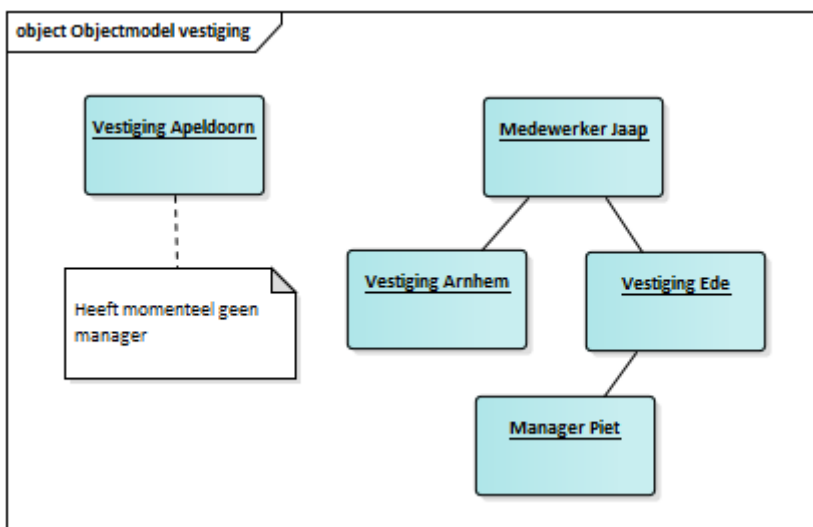
Het UML klassediagram is een verrijkte graaf waarbij met name de associaties een aantal verrijkingen kent die de taal de zeggingskracht. In onderstaande afbeelding een voorbeeld van een basis klassediagram.



In de afbeelding is te zien dat een klasse bestaat uit een naam en dat sommige klassen attributen hebben. Voor de attributen geldt dat het data type getoond worden en de optionaliteit [0..1] als het optioneel is als er niets staat is het verplicht.

Voor de associaties zijn er meerdere extra kenmerken te zien. Bijvoorbeeld welke rol de klasse vervuld in de associatie. Vestigingsmanager is manager van de vestiging etc. Ook wordt gemodelleerd wat de cardinaliteiten zijn. Dit houdt in of een klasse 0, 1 of meerdere keren kan voorkomen in de associatie. In het voorbeeld. Bij een vestiging werken een of meerdere medewerkers en een medewerker kan bij een of meerdere vestigingen werken.

Naast het klassediagram wordt in een aantal gevallen een objectmodel opgesteld. Onderstaande voorbeeld geeft een beeld van de notatie.



Het objectmodel wordt met name gebruikt om het klassediagram dat uitgewerkt is concreet te maken voor een bepaalde context (binnen het organisatiedomein). In het voorbeeld is te zien hoe dat eruit ziet in onze voorbeeldcase.

Kenmerken

UML klassennotatie heeft de volgende kenmerken:

- Een internationale en breed geaccepteerde standaard
- Rijke notatie met name voor logische datamodellering
- Interactie met leveranciers wordt eenvoudiger
- Het genereren van diverse andere output zoals XSD's
- Eenvoudig om te zetten naar documentatie zoals berichtenboeken
- 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 UML klassediagram wordt vooral toegepast binnen de logische data modellering en in een aantal gevallen binnen de fysieke modellering. Binnen dit werkveld is het een zeer krachtige en veel gebruikte modelleerwijze. Het heeft de mogelijkheid om voldoende detail toe te voegen (met name bij de attributen en de associaties).

Een extra gebruikstoepassing van het UML klassediagram is de inzet van deze notatie binnen software modellering en -ontwikkeling. Hierbij wordt het UML klassediagram veelal verrijkt met meer details en het UML klassediagram kan dan vervolgens omgezet worden tot werkende software programmatuur door middel van codegeneratie.

Houdt er rekening mee dat ondanks de eenvoud van de notatiewijze het opstellen van een UMLK basis 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 UML klassennotatie heeft met veel notatiewijzen een relatie. In onderstaande opsomming een overzicht:

- ArchiMate Datamodellering 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.

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

UML klassennotatie is een veelgebruikte notatiewijze met name voor het opstellen van logische datamodellen. Het legt daarmee een verbinding tussen de fysieke modellen en de conceptuele modellen en is daarmee een onmisbare schakel in de data modelleerketen.

Het klassediagram wordt in veel situaties toegepast, met name waar een relatie is met softwareontwikkeling. De basisnotatie biedt al een ruime hoeveelheid mogelijkheden om complexe modellen op te stellen. Dit is enerzijds de kracht van het UML klassediagram en anderzijds een zwakte omdat de modellen veelal te complex zijn voor stakeholders met minder modelleerervaring.

Voor 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