**Deelproduct 6d**

**Onderbouwde analyse**

**Beroepsproduct 3**

*Blok 3*

*Adam Oubelkas (2150912)*

Avans Academie Associate degree Informatica

**Inleiding van de analyse**

Om een applicatie en/of database te ontwikkelen, moet er vooraf al bepaald worden hoe deze structuren van pas komen en welke gegevensstromingen er plaats vindt. Dit wordt gedaan door ontwerpen te maken, op basis van de klant en ontwikkelaar overeenkomende Requirements (Swart, 2010). Deze ontwerpen dienen als blauwdrukken voor de applicatie en/of database. Echter moet er nog een vertaalslag worden gemaakt van de Requirements naar de ontwerpen. Deze vertaalslag zorgt ervoor dat alle Requirements transparant en zorgvuldig worden verwerkt op één plek, zodat hieruit een door de klant en ontwikkelaar gewenste applicatie en/of database (het eindproduct) ontstaan. De vertaalslag bestaat in deze context uit een CogNIAM-analyse (Nijssen, 2009) die in dit verslag grondig wordt behandeld.

*Verwoording van de analyse*

De eerste stap tot het omzetten van Requirements in ontwerpen is het normaliseren van Requirements. Het begon bij het opzoeken van zelfstandige naamwoorden uit relevante contexten van de Requirements. Deze naamwoorden tonen specifieke entiteiten aan.

Vervolgens heeft de ontwikkelaar samen met de klant overwogen welke van de opgezochte entiteiten ontworpen moeten zijn. Uit de opgezochte entiteiten is ieder door de klant gewenste entiteit apart geformuleerd in de CogNIAM-analyse zodat deze entiteiten transparant en overzichtelijk zijn voor het ontwerpen van de applicatie en/of database. Deze entiteiten zijn: plaats, activiteit, IoT-apparaat, burger en advies.

Daarna zijn er, in overeenstemming met de klant en ontwikkelaar, eigenschappen toegevoegd aan ieder door de klant gewenste entiteit in de analyse.

Verder zijn er in de analyse van ieder gewenste entiteit drie populaties gemaakt die bestaan uit fictieve gegevens. Deze gegevens zijn bedoeld voor het testen van gegevensstromingen van ieder entiteit. Daarom wordt de zinsbouw herhaald in alle populatie van ieder entiteit, waarvan alléén de zinsdelen moeten veranderen die een specifieke eigenschap opnoemt van de entiteit.

Door, in overeenstemming met de klant en ontwikkelaar, entiteiten met toebehorende eigenschappen en populaties in kennis grammaticale zinnen te formuleren in de analyse, is er alvast een eerste opzet gemaakt voor de structuur van zowel de applicatie als de database.

*Classificaties van de analyse*

Nádat de ontwikkelaar samen met de klant alle kennis grammaticale zinnen zijn opgebouwd, zijn de populaties van deze zinnen kenbaar gemaakt. Dit is gedaan door in de zinnen herhalende zinsdelen weg te halen. Het vervolg hiervan is dat de gegevensstromingen binnen en tussen entiteiten transparanter zijn waardoor het nog makkelijker is om de structuur voor de applicatie en/of database te ontwerpen.

*Kwalificaties van de analyse*

Door de gegevensstromingen in de analyse bekend en duidelijk te maken, is ieder gegeven van ieder populatie gemarkeerd als doorgetrokken gestreepte kader. Hierdoor zijn alle gegevens van ieder entiteit makkelijk te identificeren binnen de gegevensstromingen van de applicatie en/of database.

Daarnaast moet ieder eigenschap van een entiteit blauwgedrukt zijn zodat deze eigenschappen makkelijker te vinden zijn tijdens het ontwerpen van de applicatie en/of database.

*Begripsdefinities van de analyse*

Als gevolg van het markeren van eigenschappen in de analyse, zijn deze zichtbaar maar niet begrijpelijk voor de klant. Om alle eigenschappen begrijpelijk te maken, is er aan ieder eigenschap een definitie verleend. Deze definities zijn belangrijk om zowel de eigenschappen als entiteiten altijd te kunnen begrijpen tijdens het hiervan implementeren in de applicatie en/of database.

*Identificaties van de analyse*

In ieder eigenschap van een entiteit zitten gegevens die geïdentificeerd zijn op de inhoud van de (unieke) waarde. Dit is gedaan door telkens in iedere entiteit de zinsbouw ‘Binnen de verzameling van alle \*gegevens\* identificeert eigenschap \*gegeven\* een specifieke eigenschap’ te hebben toegepast. Hierdoor is ieder gegeven makkelijker te identificeren in zowel de applicatie als de database.

*Sjablonen van de analyse*

Van de kennis grammaticale zinnen uit de identificaties en kwalificaties zijn sjablonen gemaakt door ieder zinsdeel waar de populatie bevindt te hebben vervangen met een afkorting van de toebehorende eigenschap.

Deze sjablonen zijn gebruikt in gebruikersschermen van de applicatie, om de entiteiten met bijbehorende eigenschappen visueel/tekstueel te tonen.

*Diagrammen van de analyse*

Met iedere sjabloon uit de analyse hebben de eigenschap(en) en populatie(s) samen vorm gekregen als diagrammen.

Ieder diagram heeft een unieke naam, één unieke sjabloon en is verdeeld in geïdentificeerde of gekwalificeerde. Eén geïdentificeerd diagram bestaat altijd uit slechts één eigenschap en één populatie, terwijl één gekwalificeerde altijd meerdere eigenschappen en populaties heeft.

Bovendien gaat een geïdentificeerd diagram altijd over één eigenschap en een gekwalificeerd diagram altijd over één entiteit met daarin meerdere eigenschappen. Zo is het duidelijk hoe de achtergrond van de applicatie en database er uit ziet.

*Uniciteit in de analyse*

In overeenstemming met de klant is bepaald of een eigenschap wel of geen unieke waardes binnen de bijhorende populatie mag hebben. Dit wordt ook wel de uniciteit genoemd.

Boven op iedere diagram is een pijltje als symbool voor de uniciteit aangebracht. Op iedere uniciteitsymbool staat de tekst ‘pk\*nummer\*’ waarvan ‘pk’ afkorting is voor primaire sleutel en \*nummer\* het rangtelnummer van het diagram is.

Op deze manier is in de applicatie en database bepaald van welke eigenschappen per entiteit de waarde wel of niet uniek zijn.

*Hoofd- en deelverzamelingen van de analyse*

Iedere geïdentificeerde diagram is een hoofdverzameling op zich doordat deze als eigenschap zijnde onafhankelijk functioneert van een gekwalificeerd diagram oftewel entiteit. Een gekwalificeerd diagram (entiteit) is afhankelijke van geïdentificeerde diagrammen (eigenschappen) omdat een entiteit eigenschappen moet hebben om zichzelf te identificeren en kwalificeren.

Dus er zijn door hoofd- en deelverzamelingen vaste relaties bepaald tussen geïdentificeerde diagrammen en gekwalificeerde diagrammen.

In zowel de applicatie als database zijn deze relaties als referenties gebruikt voor het bepalen van functies voor iedere entiteit.

*Gelijkheden in de analyse*

Aan de hand van overeenkomstige eigenschappen tussen geïdentificeerde diagrammen en gekwalificeerde diagrammen, zijn verbanden getrokken.

Bij iedere verband hebben twee diagrammen één overeenkomstig verzamelingsnummer en één overeenkomstige uniciteit. Hier is dan het symbool ‘=:300x’ aangebracht, waarvan ‘x’ staat voor het rangtelnummer van het verband oftewel gelijkheid. Dit symbool staat dus voor de gelijkheid tussen twee overeenkomstige diagrammen, waarvan altijd één geïdentificeerde en één gekwalificeerde.

In zowel de applicatie als database zijn deze gelijkheden, samen met de uitsluitingen, als referenties gebruikt voor het bepalen van wel of niet samenvoegen van entiteiten.

*Uitsluitingen in de analyse*

Bij iedere gelijkheid tussen één geïdentificeerd diagram en gekwalificeerde diagram, is er gekeken naar de twee populaties waar het gelijkheidssymbool bovenop staat. Als deze populaties van elkaar verschillen, dan is er sprake van een uitsluiting in de gelijkheid tussen de twee diagrammen. Dit betekent dat het geïdentificeerde diagram niet samengevoegd mag zijn met het gekwalificeerde diagram.

Deze uitsluiting is dus bepalend voor het wel of niet samenvoegen van één geïdentificeerd diagram met een gekwalificeerd diagram. Deze bepalingen zijn toegepast in de applicatie en database.

*NietLeeg\_Optioneelregel in de analyse*

Bepaalde eigenschappen hebben in hun gegevens een verplichting tot het wel of geen waardes hebben, afhankelijk van de context en functionaliteit van een applicatie en/of database die hier gebruik van maken. In overeenstemming met de klant en ontwikkelaar is dan de tekst ‘nl’ (niet leeg) of ‘opt’ (optioneel) aangegeven bij iedere eigenschap waar geen uniciteitsymbool boven hangt.

*Mini- en maximalen in de analyse*

Het aantal waardes binnen de populatie van ieder eigenschap moet bepaald zijn. Dit wil dus zeggen dat zowel de applicatie als database moeten weten wat het minimaal aantal waardes en maximaal aantal waardes zijn per eigenschap van een entiteit.

*Formaatwaarderegel in de analyse*

Populatie van ieder eigenschap bevat gegevens die één eigen soort data hebben, omdat zowel de applicatie als database dit vereist voor het verwerken van deze gegevens in bepaalde functionaliteiten. Het gaan hier om functionaliteiten die moeten weten of ze een tekst moeten ontleden of numerieke waardes moeten berekenen, bij wijze van spreken.

*Waarderegel in de analyse*

Het kan zijn dat gegevens binnen een populatie op voorhand bepaald is en op lange termijn niet zal veranderen. Er is hier dan sprake van een beperkte populatie die bestaat uit een vast aantal waardes. Dit betekent dat deze populatie alléén in de applicatie zal bevinden, omdat de populatie (voorlopig) niet relevant is voor gebruik in de database.

*Samenvoegingen in de analyse*

Als er tussen één geïdentificeerd diagram en één gekwalificeerd diagram gelijkheid is waarin geen uitsluiting bevindt, dan pas mag de geïdentificeerde met de gekwalificeerde samengevoegd zijn. Dit is nodig om redundanties in entiteiten te voorkomen binnen zowel de applicatie als database.

**Samenvatting van de analyse**

De eerste stap tot het omzetten van de Requirements in ontwerpen is het normaliseren, oftewel verwoorden, van de Requirements. Nádat, in overeenstemming met de klant en ontwikkelaar, alle kennis grammaticale zinnen zijn opgebouwd, zijn de populaties van deze zinnen kenbaar gemaakt.

Van de kennis grammaticale zinnen uit de identificaties en kwalificaties van de analyse zijn sjablonen gemaakt. In de analyse zijn met iedere sjabloon de bijbehorende eigenschap(en) en populatie(s) samen vorm gaan krijgen als diagrammen.

Tussen geïdentificeerde diagrammen en gekwalificeerde diagrammen zijn er, door de diagrammen verwante hoofd- en deelverzamelingen, vaste relaties bepaald. Aan de hand van overeenkomstige eigenschappen tussen geïdentificeerde diagrammen en gekwalificeerde diagrammen, zijn verbanden oftewel gelijkheden getrokken. Bij iedere gelijkheid is er gekeken of er verschil(len) in de twee vergeleken populaties zijn.

Populatie van ieder eigenschap bevat gegevens die één eigen soort data hebben, omdat zowel de applicatie als database dit vereist voor het verwerken van deze gegevens in bepaalde functionaliteiten. Een populatie kan op voorhand bepaald zijn en op lange termijn niet veranderd worden waardoor deze populatie alleen toegepast moet worden in de applicatie.

Uiteindelijk is het samenvoegen van entiteiten, indien mogelijk, nodig om redundanties in entiteiten te voorkomen binnen zowel de applicatie als database.

# Literatuurlijst

Swart, N. (2010). Handboek Requirements. Brug tussen Business en ICT. Uitgeverij: Eburon Business.

Nijssen, S. Van Leeuwen, K. Hoppenbrouwers, J. Bouwmeester, H. (2009). OntoRule. Ontologies meets Businessrules. Conceptual (Enterprise) Modeling. Uitgeverij: OntoRule Consortium.