
# Scriptie

## Opportuniteit als norm in een data warehouse

**Student**

| | |
|---|---|
| Naam: | ing. Bas Tonissen |
| Studentnr: | 0249440 |
| E-mail: | bas@tonissen.com |
| Scriptienr: | 28 IK |

**Radboud Universiteit Nijmegen**

| | |
|---|---|
| Begeleider: | prof. dr. ir. Theo van der Weide |
| Referent: | dr. Patrick van Bommel |

**Info Support B.V.**

| | |
|---|---|
| Afstudeercoördinator: | bc. Pascalle Hijl |
| Opdrachtgever: | ing. Bas Stemerdink |
| Technisch begeleider: | Bertjan Kaan |

# Abstract

Van de gegevens die een bedrijf produceert, moet snel en inzichtelijk, informatie worden verkregen. Er blijkt behoefte te zijn aan real-time informatie uit het data warehouse. Echter blijkt uit de literatuur dat dit vrijwel onmogelijk is. Er wordt daarom aangenomen dat informatie op het juiste moment inzichtelijk gemaakt dient te worden. Een gebruiker heeft behoefte aan bijgewerkte en betrouwbare informatie. Er is aangenomen dat dit afhankelijk is van de kwaliteit van de gegevens. Om deze reden is er onderzocht welke kwaliteitsdimensies van gegevens in de literatuur bekend zijn. Hierbij blijkt de kwaliteitsdimensie opportuniteit het best aan te sluiten bij de beschreven behoefte van een gebruiker.

In het onderzoek wordt er een model, het informatie vervaardigingsysteem, voor een data warehouse voorgesteld waarin opportuniteit meetbaar gemaakt kan worden. Het informatie vervaardigingsysteem is met behulp van FCO-IM beschreven. Hiermee is een formele definitie van het informatie vervaardigingsysteem opgesteld. Met behulp van een voorbeeld in LISA-D wordt aangetoond dat het model een correcte basis van redeneren over de werkelijkheid biedt.

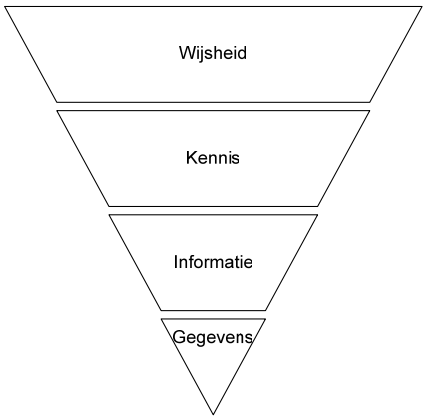
# Voorwoord

Deze scriptie is het resultaat van het afstudeeronderzoek dat ik heb uitgevoerd als afsluiting van de opleiding Informatiekunde aan het Nijmeegs Instituut voor Informatica en Informatiekunde van de Radboud Universiteit Nijmegen. Dit onderzoek heb ik verricht bij Info Support B.V. te Veenendaal.

Om te beginnen wil ik Info Support bedanken dat ze mij de mogelijkheid hebben geboden om mijn onderzoek bij hen te volbrengen. Ze hebben mij een enorme vrijheid gegeven in het formuleren van de onderzoeksopdracht welke aanzienlijk is afgeweken van de oorspronkelijke formulering. De begeleiding die geboden wordt is uitzonderlijk en onderscheidt zich in positieve zin van andere bedrijven. Ik wil iedereen van Info Support die mij heeft geholpen met mijn onderzoek bedanken. In het bijzonder wil ik graag Bas Stemerdink bedanken die altijd tijd vrij kon maken voor mijn vragen en hiermee verder ging dan enkel de rol van opdrachtgever. Verder wil ik Pascalle Hijl en Bertjan Kaan bedanken die mij respectievelijk procesmatige en technische ondersteuning hebben geboden. Tot slot wil ik Tom Langerak bedanken die tijd heeft kunnen vinden in zijn drukke agenda om mijn scriptie door te lezen en te voorzien van waardevolle feedback.

Vanuit de universiteit heeft Theo van der Weide mijn onderzoek begeleid. Ik heb erg veel aan zijn begeleiding gehad. De gesprekken die wij hebben gevoerd hebben me altijd erg gemotiveerd. Ik wil hem bedanken voor al zijn inzichten, tips en leermomenten.

Tot slot wil ik mijn familie en vrienden bedanken voor de opbeurende woorden en ondersteuning en die zij mij hebben geboden tijdens het hele traject. Hierbij wil ik David Fremeijer in het bijzonder bedanken voor zijn bijdrage aan de afronding van deze scriptie.




De bovenstaande omgedraaide piramide sluit naar mijn mening mooi aan bij het gedachtegoed van een data warehouse. Hierin is duidelijk te zien hoe wijsheid wordt gevormd uit gegevens. Gegevens worden geaggregeerd en geanalyseerd om informatie te bieden. Kennis is informatie die voldoende vertrouwd kan worden om op beslissingen te baseren of acties op te ondernemen. Wijsheid is kennis die selectief is toegepast. Een data warehouse wordt dus ontworpen om de gebruikers wijsheid te bieden die is gebaseerd op gegevens uit het verleden.

Vaak wordt gezegd dat in het verleden resultaten geen garantie bieden voor de toekomst. Ze zijn echter wel een goede graadmeter. Ik wil dit voorwoord dan ook besluiten met de volgende wijsheid van Clarence Darrow (1857–1938), een Amerikaanse advocaat:

*“History repeats itself; that's one of the things that's wrong with history”.*

# Inhoudsopgave

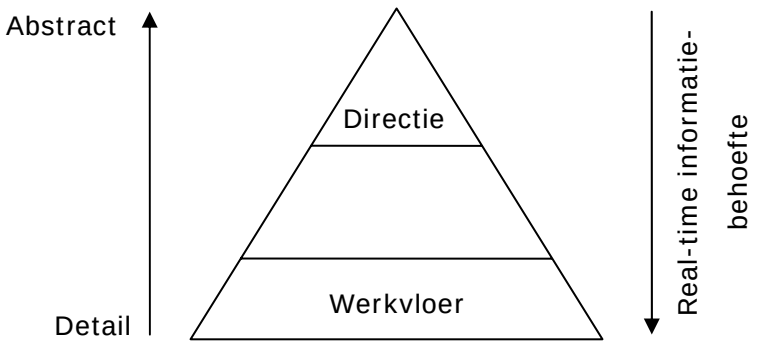
# 1 Introductie

## *1.1 Context*

De informatiebehoefte van een organisatie is vandaag de dag groot en aan veel wisselingen onderhevig. Uit de gegevens die een bedrijf produceert, moet snel en inzichtelijk informatie worden verkregen. Een data warehouse voorziet in deze behoefte. Met een data warehouse is het mogelijk om verschillende bronnen van informatie te ontsluiten. Een data warehouse integreert de gegevens uit gegevensbronnen om zo één centrale plek te bieden waar deze kunnen worden geraadpleegd. Het is ontworpen om grote hoeveelheden data te bevragen waarmee snel (complexe) rapportages kunnen worden gemaakt. Hierdoor is het mogelijk om bijvoorbeeld knelpunten sneller te signaleren, trends waar te nemen of andere analyses uit te voeren.

In het algemeen is het voldoende om de bronsystemen op vaste tijden te ontsluiten: één keer per dag, per week of per maand. Dit is afhankelijk van het soort informatie: hoe snel gegevens veranderen en het abstractieniveau van de informatie.

Met de huidige technische ontwikkelingen en het feit dat informatie steeds sneller benodigd is, wordt het interessant om te kijken naar een real-time oplossing. Bij een real-time oplossing worden veranderingen in de bronsystemen direct doorgevoerd in het data warehouse. Rapportages uit dit data warehouse laten de veranderingen dan ook onmiddellijk tot uitdrukking komen in de gepresenteerde informatie. In figuur 1.1 staat grafisch weergegeven hoe het abstractieniveau en de real-time informatiebehoefte zich binnen de hiërarchie van een organisatie profileert.




**Figuur 1.1 Real-time behoefte**

## 1.2 Probleemstelling

Volgens Dey et al. maken zowel de grote volumes gegevens als de redundantie in de gegevens die zijn opgeslagen in een data warehouse het onpraktisch het data warehouse real-time te synchroniseren [16]. Zij gaan er hierbij vanuit dat wanneer de synchronisatie plaatsvindt het data warehouse systeem niet beschikbaar is. Zhang et al. geven aan dat het mogelijk is de updates te optimaliseren en zo de gegevens in het data warehouse incrementeel in plaats van volledig te vernieuwen [33]. Doordat het data warehouse in delen wordt gesynchroniseerd blijft het beschikbaar tijdens het doorvoeren van de updates.

Van het data warehouse wordt een hoge beschikbaarheid gewenst, maar er is tijd nodig om de nieuwe gegevens te synchroniseren. Uit deze tegenstrijdigheid blijkt dat het vernieuwen van de gegevens in een data warehouse niet kan zonder hiervoor een prijs te betalen.

De term real-time informatiebehoefte uit paragraaf 1.1 kan worden genuanceerd tot de behoefte aan *informatie op het juiste moment*. Een gebruiker heeft behoefte aan bijgewerkte en betrouwbare informatie. De snelheid waarmee de informatie in het data warehouse wordt vernieuwd hoeft niet per definitie real-time te zijn. De gebruiker wil van de informatie die is opgevraagd weten hoe betrouwbaar deze is met betrekking tot de meest recente vernieuwing in de gegevensbronnen.

Het uitgangspunt is dat een geschikte combinatie van snelheid van vernieuwingen en betrouwbaarheid een belangrijke kwaliteitsdimensie is van een data warehouse. Om het besef van betrouwbaarheid en snelheid van gegevens te bereiken dient van een data warehouse in kaart te worden gebracht hoe de gegevensverwerking van gegevensbron tot gebruiker verloopt. Zo mogelijk dient er een norm gevonden te worden die het voor de gebruiker inzichtelijk maakt of de gegevens op het juiste moment gepresenteerd worden.

Dit leidt tot de volgende onderzoeksvraag:

*Kan de kwaliteit van een data warehouse door middel van een raamwerk van de gegevensverwerking inzichtelijk gemaakt worden voor een gebruiker?*

Om inzicht te krijgen in een data warehouse dient de structuur hiervan in kaart te worden gebracht. Door middel van deze structuur kan inzicht verkregen worden in de positie van updates in het data warehouse. Hierdoor kan het update probleem inzich-

telijk worden gemaakt. Concepten en relaties dienen geïsoleerd te worden ten behoeve van de theorievorming. Hierbij dient er een model gemaakt te worden zodat onafhankelijk van de toepassingen uitspraken gedaan kunnen worden.

Verder dient er onderzocht te worden welke dimensies kwaliteit kent en welke hiervan te maken hebben met gegevens op het juiste moment aan de gebruiker presenteren.

Dit leidt tot de volgende deelvragen:

*"Hoe kan de kwaliteit van een data warehouse gedefinieerd worden?"*

*"Is er een raamwerk op te stellen waarmee de architectuur van de gegevensverwerking in een data warehouse in kaart kan worden gebracht?"*

*"Hoe is kwaliteit van een data warehouse voor de gebruiker inzichtelijk te maken?"*

## 1.3 State-of-the-art

Snijders en van Reijswoud geven aan dat real-time data warehousing een doodlopend spoor is [30]. Zij beschrijven een alternatieve architectuur om dit anders op te lossen. Barnes Nelson licht toe wanneer er wel of niet naar een real-time data warehouse gestreefd dient te worden [9]. Bruckner et al. beschrijven een J2EE (Java 2 Platform Enterprise Edition) architectuur voor een ETL omgeving die het volgens hen mogelijk maakt een gedistribueerde, schaalbare, nagenoeg real-time ETL omgeving te implementeren [11]. Er worden in de literatuur verder nog andere architecturen beschreven waarmee real-time data warehousing mogelijk zou moeten zijn [2][23].

Bruckner et al. beschrijven een conceptueel model voor de consistentie van tijd [12]. Ze presenteren een model dat de gebruikers te kennen geeft waarom een systeem gereageerd heeft op een bepaalde toestand van de gegevens in de bronsystemen. Dey et al. reiken methode aan om de optimale tijd die tussen de synchronisaties van het data warehouse zit te berekenen [16].

## 1.4 Opbouw

Deze scriptie in opgebouwd uit zes hoofdstukken. Na het huidige hoofdstuk wordt in hoofdstuk twee een definitie gegeven van wat een data warehouse is en wat een

data warehouse precies inhoudt. Er wordt tevens aandacht besteed aan de synchronisatie van een data warehouse.

Hoofdstuk drie gaat in op de kwaliteit in een data warehouse. Er wordt een definitie geschetst van kwaliteit in een data warehouse. Verder wordt er licht geworpen op de kwaliteitsdimensies van gegevens en wordt de kwaliteitsdimensie opportuniteit inzichtelijk gemaakt.

In hoofdstuk vier wordt er een model van een data warehouse gepresenteerd. Hiermee is behalve de architectuur van een data warehouse tevens in kaart te brengen hoe de opportuniteit van een gegeven meetbaar gemaakt kan worden. Dit wordt verduidelijkt door het gebruik van een casus en formeel gespecificeerd door middel van FCO-IM.

Hoofdstuk vijf zal ingaan op de conclusies een aanbevelingen die gedaan kunnen worden aan de hand van het beschreven onderzoek. De gestelde onderzoeksvraag en deelvragen kunnen aan de hand van dit hoofdstuk beantwoord worden.

Tot slot geeft hoofdstuk zes een weergave van de literatuur die is gebruikt in het onderzoek.

# 2 Data warehouse

## 2.1 Inleiding

Het begrip data warehouse is een begrip dat veel wordt gebruikt. Wat een data warehouse nu precies inhoudt, daar zijn de meningen over verdeeld. In dit hoofdstuk wordt vastgesteld uit welke onderdelen een data warehouse bestaat en wat binnen het onderzoek onder een data warehouse wordt verstaan.

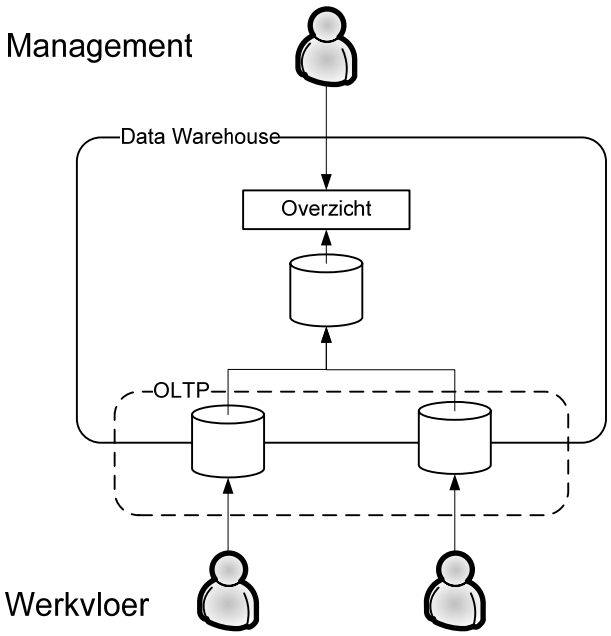
## 2.2 Wat is een data warehouse

Volgens Kimball is een van de belangrijkste dingen die een onderneming bezit haar informatie. Deze informatie wordt over het algemeen op twee manieren bewaard: in de operationele systemen en in het data warehouse [24]. Wanneer de organisatie vergeleken wordt met een schip, dan zorgen de gebruikers van de operationele systemen ervoor dat het schip blijft varen en de gebruikers van het data warehouse bepalen de koers van het schip.

De operationele systemen ondersteunen de dagelijkse operationele aspecten van een organisatie. Denk hierbij aan bijvoorbeeld order verwerking, voorraad bijhouden, betalingen ontvangen, etc. Dit soort systemen, waarin alle dagelijkse transacties worden vastgelegd, worden veelal aangeduid als ‘online transaction processing’ (OLTP) systemen. De gebruikers van een OLTP systeem zijn verantwoordelijk voor de invoer van de gegevens en herhalen veelal dezelfde operationele taak. Ze hebben over het algemeen maar te maken met één enkel gegeven per transactie.

Ofschoon OLTP systemen goed werken bij het ondersteunen van de operationele kant van de onderneming, zijn ze over het algemeen niet bedoeld om het management ondersteuning te bieden bij hun beslissingen en strategische inzichten. Het management wil informatie dat een overzicht van het geheel biedt. Denk hierbij aan bijvoorbeeld inzicht in het orderverloop, vergelijking maandelijkse omzetten, etc. Dit soort inzichten vereisen geaggregeerde en samengevatte gegevens geëxtraheerd uit grote hoeveelheden informatie. Dit betekent dat de gebruikers nooit te maken hebben met slechts één gegeven per transactie, maar een heleboel gegevens samengesmolten tot één antwoord in enkele of meerdere transacties. Een data warehouse voorziet gebruikers in deze behoefte. Deze kan gezien worden als een database die

gevuld wordt met operationele gegevens uit verschillende bronnen binnen de organisatie.




**Figuur 2.1 Globaal overzicht data warehouse**

In figuur 2.1 staat een globale weergave van een data warehouse. Op deze weergave is goed te zien dat de (OLTP) bronsystemen samen komen in één bron. Uit deze bron kan door het management een overzicht worden gegenereerd. Uit de illustratie blijkt tevens dat de bronsystemen een onderdeel zijn van het data warehouse. In paragraaf 2.3 zal verder op de onderdelen van het data warehouse worden ingegaan.

Kimball omschrijft een data warehouse als volgt:

*"The conglomeration of an organization's data warehouse staging and presentation areas, where operational data is specifically structured for query and analysis performance and ease-of-use [24]."*

Deze definitie omschrijft een data warehouse vrij globaal. Vrij vertaald omschrijft Kimball een data warehouse van een organisatie als de verzameling van de gegevens verzamellaag en de gegevens presentatielaag (zie respectievelijk paragraaf 2.3.2 en 2.3.3) waar operationele gegevens in een speciale structuur zijn opgeslagen. Op

deze structuur kunnen queries en analyses worden uitgevoerd die goed presteren en makkelijk zijn in het gebruik.

Inmon heeft een specifiekere definitie van een data warehouse:

*"A data warehouse is a subject-oriented, integrated, nonvolatile, and time-variant collection of data in support of management's decisions [20]."*

Hieruit blijkt dat een data warehouse managers een collectie gegevens aanreikt die hen ondersteuning biedt bij het maken van beslissingen. Inmon stelt verder dat deze collectie onderwerpgericht, geïntegreerd, onvergankelijk en tijdsgebonden is.

Onderwerpgericht betekent dat het data warehouse ingericht moet worden op de manier zoals de managers hun organisatie zien. Elk bedrijf heeft zijn eigen verzameling van onderwerpen. De onderwerpen van een supermarkt zijn bijvoorbeeld verkoop, winkel en product.

Het volgende en belangrijkste kenmerk van een data warehouse is integratie. Gegevens vanuit verschillende informatiebronnen worden in het data warehouse opgeslagen. Tijdens dit opslaan worden de gegevens geconverteerd, opgemaakt, gesorteerd, samengevat, etc.

Onvergankelijk houdt in dat er gegevens in het data warehouse worden geladen, maar (in principe) niet worden bijgewerkt. Het data warehouse is een statische afspiegeling van de situatie op dat moment. Wanneer er veranderingen plaatsvinden wordt er een nieuwe afspiegeling gemaakt. Hierdoor is het mogelijk geschiedenis van de gegevens te handhaven.

Het laatste kenmerk van een data warehouse is dat het tijdsgebonden is. Dit houdt in dat elk gegeven correct is vanaf een bepaald moment. Op een gesteld punt in de tijd heeft een bepaald feit plaats gevonden. Dit dient op een bepaalde manier gemarkeerd (bijvoorbeeld met een timestamp of datum van transactie) te worden om aan te geven vanaf welk moment het gegeven correct is.

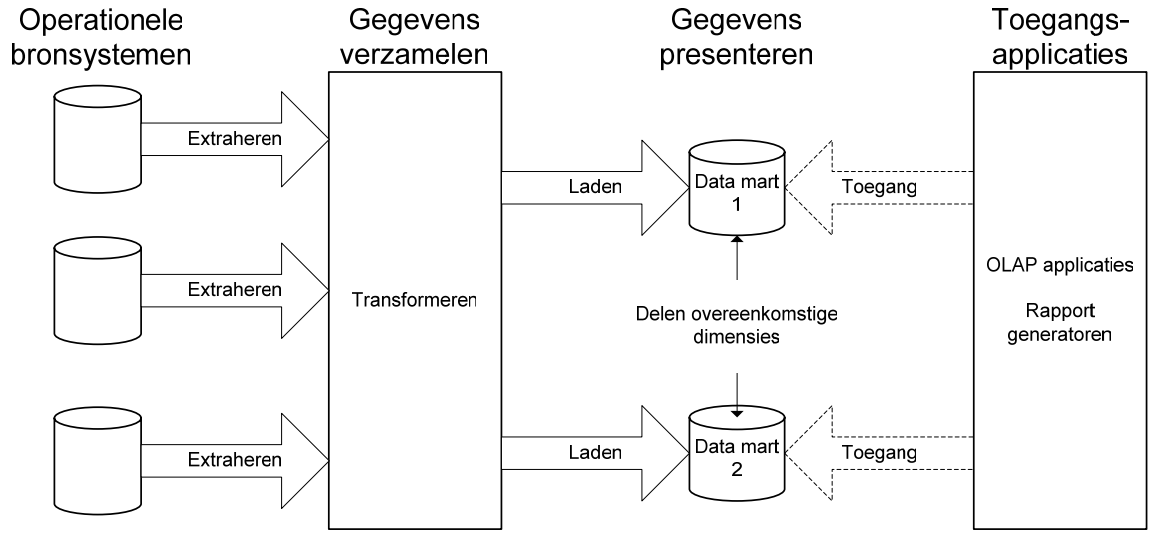
In dit onderzoek wordt uitgegaan van de volgende definitie van een data warehouse:

*"Een data warehouse beschrijft een complex afhankelijkheidsnetwerk van gegevensbronnen en processen dat ondersteuning biedt bij het maken van analyses en beter gefundeerde beslissingen."*

Deze definitie is meer gericht op de structuur van de architectuur van het data warehouse dan op de kenmerken zoals deze zijn beschreven in de gegeven definitie van Inmon.

## 2.3 Onderdelen van een data warehouse

Kimball schrijft dat een data warehouse uit vier onderdelen of lagen bestaat: operationele bronsystemen, gegevens verzamelen, gegevens presenteren en toegangsapplicaties (zie figuur 2.2) [24].




**Figuur 2.2 Onderdelen data warehouse**

### 2.3.1 Operationele bronsystemen

De operationele bronsystemen leggen de transacties van het bedrijf vast. Omdat ze zich buiten het data warehouse bevinden is er weinig tot geen invloed op uit te oefenen. De voornaamste prioriteiten van de bronsystemen zijn beschikbaarheid en verwerksnelheid. Bij een goed data warehouse dienen er weinig historische gegevens in de operationele bronsystemen opgeslagen te worden en kunnen de bronsystemen grotendeels ontzien worden van de weergave van het verleden.

### 2.3.2 Gegevens verzamelen

Deze laag van het data warehouse is verantwoordelijk voor zowel de opslag als de processen die verantwoordelijk zijn voor het proces van extraheren, transformeren en laden (ETL). Bouzeghoub schrijft dat de verversing (het laden van gegevens) van een data warehouse een belangrijk proces is dat de bruikbaarheid van de verzamelde

en geaggregeerde brongegevens bepaalt [10]. In deze laag worden de gegevens klaargemaakt om aan de gebruikers getoond te worden. Echter zullen de gebruikers nooit rechtstreeks met deze laag te maken hebben. Er mogen hierop geen queries uitgevoerd worden. Zodra dit wel gebeurt is het onderdeel van de laag die de gegevens presenteert.

Het extraheren is de eerste stap om de gegevens in het data warehouse systeem te krijgen. Dit betekent dat er uit de bronsystemen de benodigde gegevens gekopieerd worden naar het data warehouse om hier verder bewerkt te kunnen worden.

Zodra de gegevens in het data warehouse staan worden ze onderworpen aan tal van transformaties zoals het zuiveren van de gegevens (spellingscorrecties, ontbrekende elementen afhandelen, standaard indelingen hanteren), meerdere gegevensbronnen combineren, toekennen van warehouse sleutels (identificerende sleutels die geen relatie hebben met het bronsysteem) en verwijderen van dubbele gegevens. Behalve deze globale transformaties kunnen er tevens specifieke bedrijfsregels worden toegepast. Al deze transformaties ter bevordering van de kwaliteit gaan vooraf aan het laden van de gegevens in de presentatielaag.

De laatste stap in het ETL proces is het laden van de data. Dit gebeurt meestal in de vorm van het aanbieden van de gegevens aan de data marts in gegevens presentatielaag.

### 2.3.3 Gegevens presenteren

In deze laag zijn de gegevens toegankelijk voor de gebruikers. Hier worden alle gegevens opgeslagen die de gebruikers de mogelijkheid biedt tot het maken van queries (zie paragraaf 2.3.4). Omdat de voorgaande lagen niet zichtbaar zijn voor de gebruikers is de presentatielaag in hun beleving het complete data warehouse. Dit gedeelte is het enige dat ze zien en gebruiken via applicaties.

Hoe de gegevens presentatielaag er precies uitziet is afhankelijk van de organisatie waarvoor het data warehouse is opgezet. Het kan worden gezien als een verzameling data marts. Kimball beschrijft een data mart als een flexibele set gegevens, die gepresenteerd wordt in een dimensionaal model (zie paragraaf 2.4) en idealiter gebaseerd is op gegevens met de hoogst mogelijke fijnkorreligheid die uit de operationele bron te extraheren zijn [24]. Een data mart is een stuk uit het hele te presenteren gebied van de organisatie. De meest simpele data mart presenteert één enkel bedrijfsproces. Hierbij staat het proces los van de grenzen binnen de organisatiestructuur. Soms is er echter een centrale data warehouse database waarin alle gege-

vens worden geladen. Deze kan bevraagd worden door gebruikers en hieruit kunnen eventueel weer losse data marts gevuld worden.

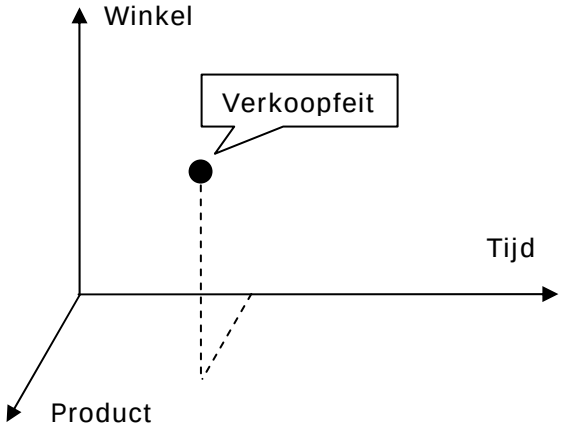
Een data mart kan relationeel van aard zijn, maar het is tevens mogelijk dat deze gebaseerd is op een multidimensionale database (ook wel aangeduid als een multi-dimensionale online analytic processing database of kortweg een OLAP database). In het laatste geval wordt het een kubus genoemd. Volgens Inmon voorziet een multi-dimensionale database management systeem (MDBMS) een organisatie van een structuur die hen in staat stelt flexibele toegang tot gegevens te krijgen, hierop op elke mogelijke manier in en uit- te zoomen en dynamisch de relaties tussen de samengevatte en detail gegevens te ontdekken [20]. Een kubus kan worden gevuld vanuit de relationele data mart, maar ook rechtstreeks uit de gegevensbron(nen).

### 2.3.4 Toegangsapplicaties

Het laatste onderdeel van een data warehouse zijn de toegangsapplicaties. Deze laag omvat alle mogelijke applicaties die de gegevens presentatielaag bevragen om de gebruikers te ondersteunen bij beslissingen en analytische inzichten. Denk hierbij aan online analytic processing (OLAP) applicaties, simpele ad hoc query applicaties, rapportage applicaties, maar ook data mining applicaties.

## *2.4 Ontwerpmodel van een data warehouse*

Er wordt wel eens gezegd dat managers een druk schema en weinig tijd hebben. Over deze uitspraak kan worden gediscussieerd, echter zodra zij hun data warehouse applicaties gebruiken willen zij zo snel mogelijk antwoord op hun vragen. Verder moeten managers in staat zijn om op een intuïtieve manier hun vragen in de toegangsapplicaties samen te stellen. Het dimensionale ontwerpmodel sluit aan bij deze eisen.




**Figuur 2.3 Dimensies**

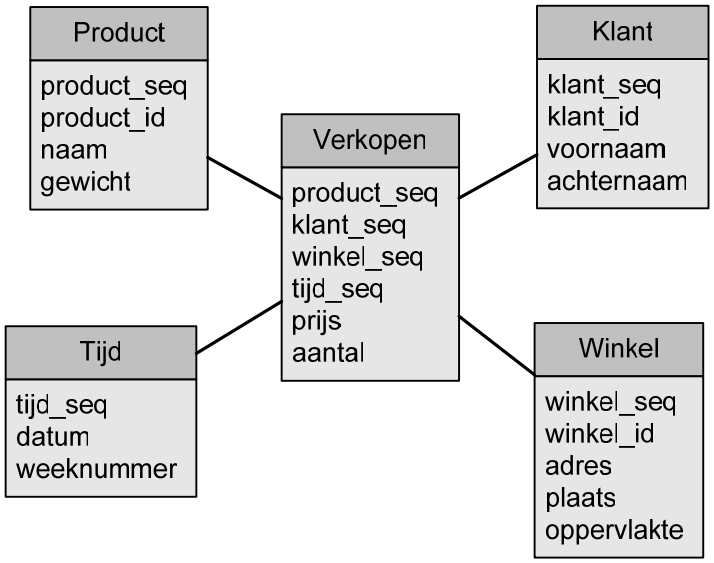
Uiteraard zal de manier waarop een manager intuïtief vragen samen kan stellen in de toegangsapplicaties sterk afhangen van de toegangsapplicatie zelf. Echter kan het dimensionaal model wel hulp bieden om aansluiting te vinden bij de visie en denkwijze van een manager omdat het aansluit bij hoe zij hun bedrijfsprocessen zien. Het beschrijft een domein door middel van dimensies. Merk op dat dit vergelijkbaar is met onderwerpgericht; een van de door Inmon gedefinieerde kenmerken van een data warehouse (zie paragraaf 2.2) [20]. Een dimensie kan worden gezien als een ondeelbare groep van gegevens. Het is voor gebruikers intuïtief om dingen onder te verdelen in dimensies. Een supermarktketen kan omschreven worden als een bedrijf dat producten verkoopt in verschillende winkels. Hierbij komen drie dimensies tevoorschijn, te weten tijd, winkel en product. Tijd is een dimensie die altijd aanwezig is, temeer omdat een data warehouse draait om het bijhouden van historische gegevens. Het verband met tijd is hierbij onmiskenbaar. De supermarktketen zal geïnteresseerd zijn in vragen als waar en wanneer verkopen plaats hebben gevonden en wat er verkocht is. De dimensies winkel en product geven hierop respectievelijk een antwoord. Een dimensie kan worden gezien als een tabel waarin, in het geval van de winkel, gegevens over de betreffende winkels van de supermarktketen worden opgeslagen.

De gevonden dimensies zijn uit te zetten in een driedimensionale grafiek waarbij er op elke dimensie in- en uitgezoomd kan worden. De oorsprong van de grafiek is weergegeven in figuur 2.3. De punten in deze grafiek zijn de opgeslagen metingen

van wanneer, welk product, waar verkocht is. Door doorsneden van deze grafiek te maken zijn gebruikers in staat antwoord op hun vragen te vinden.

In het beschreven voorbeeld van de supermarktketen is in de dimensies zelf geen ruimte is om de eigenlijke verkopen op te slaan. Hiervoor wordt een ander soort tabel geïntroduceerd: de feitentabel. De feitentabel geeft aan dat een feit heeft plaatsgevonden. In het geval van de supermarktketen legt het bijvoorbeeld vast dat product X door een klant Y in winkel Z is gekocht op tijdstip T. In figuur 2.3 staat een weergave van een punt van zo'n verkoopfeit. Hier is bijvoorbeeld af te lezen welk product, waar en wanneer is gekocht.

In de feitentabel worden ook meetwaarden vastgelegd, zoals het aantal producten dat is aangeschaft. Tezamen met de dimensies heet dit model in de relationele database wereld een sterschema. In figuur 2.4 staat een sterschema afgebeeld. Hierin is te zien dat er een dimensie klant is toegevoegd om bijvoorbeeld op te slaan wie een product heeft aangeschaft. Van der Lek beschrijft een sterschema als een relationeel ontwerp waarbij vanuit één centrale tabel, de zogenaamde feitentabel, verwijzingen zijn naar één of meer andere tabellen [25]. Die andere tabellen zijn de reeds besproken dimensies. Het sterschema is in een aantal smaken te modelleren. Het beschreven schema, wat tevens gevisualiseerd is in figuur 2.4, staat bekend als een simpel sterschema. Behalve het simpele sterschema zijn er nog andere soorten dimensionale ontwerpmodellen zoals een sneeuwvlok, hybride en optimale sneeuwvlok. Deze zijn elk een uitbreiding van het simpele sterschema. Redenen om een ander type model te gebruiken zijn flexibiliteit (herbruikbaarheid van dimensies) of een simpeler laadproces.




**Figuur 2.4 Simpel sterschema**

In paragraaf 2.2 is reeds beschreven dat OLTP systemen ontworpen zijn voor individuele transacties en ondersteuning bieden aan de dagelijkse operationele aspecten. Deze systemen zijn gebaseerd op een transactioneel ontwerpmodel dat is bedoeld om gegevens in één transactie op te halen, in te voeren of te wijzigen. Op het model van OLTP systemen zijn de door Codd geïnitieerde normalisatieregels toegepast [13]. Dit zorgt ervoor dat er geen redundantie in de opgeslagen gegevens ontstaat. Wat tot gevolg heeft dat wanneer er een feit bijgewerkt of toegevoegd wordt dit slechts op één plek gedaan hoeft te worden. Hierdoor wordt redundantie voorkomen, kunnen gegevens snel worden bijgewerkt en ontstaan er geen update-anomalieën.

Een dimensionaal model is ontworpen voor het goed presteren wanneer gegevens worden opgevraagd. Het stelt gebruikers in staat vragen te beantwoorden waarvoor aggregatie van grote hoeveelheden gegevens voor nodig is. Traditioneel wordt dit gedaan door een enkele laadtransactie op gezette tijden die mogelijkerwijs miljoenen records laadt. Om de gebruiker voldoende snelheid te bieden bij het opvragen van gegevens wordt het database schema gedenormaliseerd. Dit gaat echter ten koste van de snelheid van het bijwerken van de gegevens. Een simpel laadproces en een goede performance naar de gebruiker toe gaan dan ook moeilijk samen.

Wanneer een feitentabel te groot wordt kan dit invloed hebben op de snelheid van bepaalde queries. Het dimensionaal model biedt de mogelijkheid aggregaat tabellen af te leiden van een basis feitentabel. De aggregaat tabel bevat dezelfde meetwaar-

den als de basis feitentabel, maar heeft een lagere fijnkorreligheid. Denk bij een aggregaat tabel bijvoorbeeld aan een tabel waarin de maandelijkse verkoopfeiten worden opgeslagen in tegenstelling tot de dagelijkse verkoopfeiten van de basis feitentabel. Aggregaat tabellen worden uitsluitend toegevoegd om de prestatie van queries te bevorderen. In het genoemde voorbeeld zijn de maandelijkse verkopen reeds voorberekend. Hierdoor kan een aanzienlijke snelheidswinst geboekt worden. Het hangt volledig af van het gebruik en de eisen van een gebruiker waar en of een aggregaat tabel noodzakelijk is.

## 2.5 Synchronisatie

In paragraaf 2.3 zijn de lagen van een data warehouse beschreven. In deze paragraaf zal verder ingegaan worden op de eerste drie lagen. Gebruikers kunnen via toegangsapplicaties pas bij de gegevens in de presentatielaag als deze via het ETL proces uit de bronsystemen zijn ontsloten. De verzamellaag is verantwoordelijk voor het extraheren, transformeren en het laden in de presentatielaag. Dit proces waarbij de gegevens van de eerste naar de laatste laag worden gepropageerd wordt laden of synchroniseren genoemd. Bouzeghoub et al. beschrijven het laden en synchroniseren van een data warehouse als een workflow proces [10]. Zij definiëren de levenscyclus van een data warehouse in drie fasen:

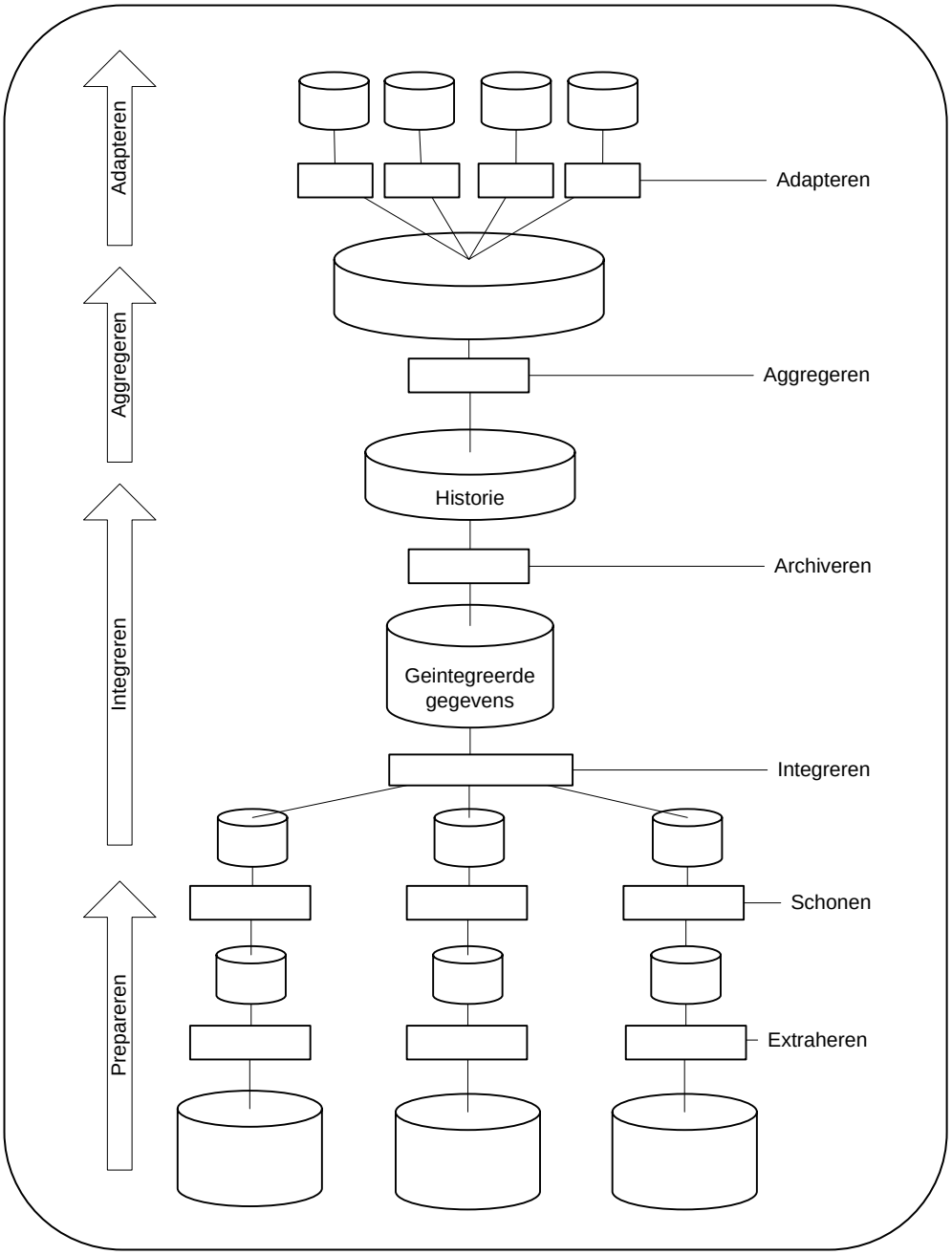
- Ontwerpen
- Laden
- Synchroniseren

In de ontwerpfase wordt er een weergave van het domein gemaakt. Dit wordt vertaald naar de architectuur van het data warehouse. Hier hoort onder andere het ontwerp van het dimensionale model bij (zie paragraaf 2.4).

Het laden van het data warehouse is het initiële vullen van het data warehouse met gegevens. Bouzeghoub et al. omschrijven dit als een proces dat plaatsvindt in een viertal stappen: prepareren, integreren, aggregeren, adapteren [10]. De eerste stap wordt voor elke gegevensbron gedaan en bestaat uit het extraheren en schonen van de gegevens. Eventueel worden tijdens deze processen de gegevens nog opgeslagen. Stap twee bestaat uit de integratie van de gegevens. Dit zijn gegevens uit verschillende gegevensbronnen die anders gerepresenteerd worden, maar hetzelfde betekenen. Denk bijvoorbeeld aan de representatiemogelijkheden van het geslacht.

Dit kan worden aangeduid als man en vrouw, ‘m’ en ‘v’, 0 en 1, etc. Deze gegevens worden samengebracht in de gegevens verzamellaag van het data warehouse. Stap drie bestaat uit het berekenen van geaggregeerde gegevens. De gegevens uit de gegevensbronnen en uit de gegevens verzamellaag hebben over het algemeen een hoge fijnkorreligheid. In deze stap worden die gegevens in een nieuwe gegevensbron gezet (soms corporate data warehouse [10] of global data warehouse [20] genoemd). Vanuit hier worden de data marts gevuld. Het vullen van de data marts is dan ook meteen de vierde en laatste stap. De gegevens uit de vorige stap worden geadapteerd naar de data marts die meer gericht zijn op specifieke bedrijfsprocessen (zie paragraaf 2.3.3).

De fase van synchroniseren heeft gelijke stappen aan die van het laden met het verschil dat het synchronisatieproces de opgetreden verschillen in de gegevensbronnen in het data warehouse laadt. De preparatiestap extraheert de veranderingen die hebben plaats gevonden sinds de vorige extractie uit elke gegevensbron. De integratie vindt op dezelfde manier plaats als bij de laadfase. Ook de aggregatiestap is gelijk, met het verschil dat de aggregaties herberekend worden met de nieuwe gegevens. Tot slot zullen in de laatste stap de gegevens naar de data marts worden verspreid.




**Figuur 2.5 Fasen voor laden en synchroniseren**

In figuur 2.5 staat een weergave van de fasen zoals beschreven door Bouzeghoub et al. [10]. Dit is slechts een mogelijke weergave van de opbouw van een data warehouse. Zij geven verder aan dat er nog een aantal verschillen zijn tussen het laden en synchroniseren. Ten eerste hoeven bij het laden de verschillende stappen (prepareren, integreren, aggregeren en adapteren) niet sequentieel uitgevoerd te worden,

maar kan dit asynchroon plaatsvinden. Ten tweede kan er bij het prepareren veel parallel uitgevoerd worden, omdat elke gegevensbron op verschillende momenten beschikbaar is en tevens een specifieke methode van extractie heeft. De synchronisatie wordt bewerkstelligd door de integratie stap. Een ander verschil zit in de tijd van beschikbaarheid van de gegevensbron. Het laden heeft een lange beschikbaarheid nodig omdat er ineens een grote hoeveelheid gegevens geladen dient te worden. Dit is een eenmalige actie en zal daarom de bronsystemen slechts eenmaal belasten. Bij het synchroniseren zal echter telkens het verschil tussen nu en de laatste keer van synchroniseren worden gepropageerd. Dit betekent dat de bronsystemen op terugkomende tijdstippen worden geraadpleegd. Het is van belang dat de operationele applicaties geen last hebben van het synchronisatieproces. Dit kan worden opgelost door elke gegevenbron slechts op bepaalde tijdstippen een bepaalde tijd te raadplegen. Tot slot zijn er bij het synchroniseren meer beperkingen met betrekking tot de responstijd dan bij het laden. Voor de gebruikers bestaat het data warehouse niet voor het initiële laden. Na het laden worden de gegevens zichtbaar en dient het te voldoen aan de eisen van de gebruikers in de zin van de beschikbaarheid en versheid van de gegevens.

Tenzij anders vermeld, zal in het vervolg synchronisatie worden gebruikt om zowel het proces van synchroniseren als dat van het laden aan te duiden.

Het synchroniseren van het data warehouse is een belangrijk proces dat de bruikbaarheid van de verzamelde gegevens bepaald. De kwaliteit van de verzamelde gegevens, die aan de personen verantwoordelijk voor het nemen van beslissingen wordt aangeboden, is afhankelijk van het vermogen van het data warehouse om veranderingen in de gegevensbronnen zo snel mogelijk door te voeren.

Bouzeghoub et al. menen dat bij een efficiënte synchronisatiestrategie rekening gehouden dient te worden met het volgende [10]:

- Applicatie eisen (bijvoorbeeld versheid van gegevens, nauwkeurigheid van gegevens, berekentijd van queries)
- Bron beperkingen (bijvoorbeeld beschikbaarheid, aantal veranderingen)
- Data warehouse systeem beperkingen (bijvoorbeeld opslagruimte)

De meeste van deze parameters zullen tijdens de levensduur van het data warehouse ontwikkelen. Dit zal leiden tot aanpassingen aan de data warehouse architectuur en synchronisatiemethoden.

Behalve de beschreven aandachtspunten dient er bij de synchronisatie tevens rekening gehouden te worden met de eisen van de gebruikers. Deze beoordelen, zal zoals beschreven in hoofdstuk 3, de kwaliteit van de gegevens. Bouzeghoub et al. beschrijven consistentie, volledigheid, nauwkeurigheid en versheid als kwaliteitsdimensies die het synchronisatieproces het best omschrijven [10]. In hoofdstuk 3 zal nader op de kwaliteit worden ingegaan.

## 2.6 Synchronisatie strategieën

In de literatuur wordt een data warehouse ook wel aangeduid als een set van “materialized views” [1][26]. Dit zijn views die werkelijk opgeslagen (gematerialiseerd) worden. Een view in een database kan worden gezien als het resultaat van een query dat wordt afgebeeld in een virtuele tabel. Wanneer er updates plaatsvinden in de gegevenbronnen zijn de materialized views (hierna aangeduid als views) inconsistent en dienen ze gesynchroniseerd te worden.

Wanneer een gegevensbron wordt bijgewerkt is het zaak dat deze verandering wordt doorgevoerd naar het data warehouse. In paragraaf 2.5 is reeds beschreven hoe deze synchronisatie dient te gebeuren. Echter wanneer het data warehouse gesynchroniseerd dient te worden is niet behandeld. Engström et al. schrijven dat bij de synchronisatie de twee termen *wanneer* en *hoe* centraal staan [17]. Het *hoe* deel bestaat uit keuzes als incrementeel of volledig verversen, on-line of off-line, etc. Het *wanneer* heeft betrekking op het tijdstip waarop een view ververst dient te worden. Er zijn verschillende strategieën van timing synchronisatie mogelijk:

- Periodieke synchronisatie
- Onmiddellijke synchronisatie
- Vertraagde synchronisatie
- Hybride synchronisatie
- Willekeurige synchronisatie

Periodieke synchronisatie is de meest voorkomende synchronisatie strategie. Hierbij wordt het data warehouse op een regelmatige interval gesynchroniseerd met

de gegevensbronnen (één keer per week, uur, etc). Een view kan inconsistent worden met de oorspronkelijke gegevensbron en geeft een momentopname van de gegevenbronnen weer. Updates in de gegevensbronnen worden pas zichtbaar bij de volgende synchronisatie.

Wanneer er een update plaatsvindt en er onmiddellijke actie tot synchronisatie wordt ondernomen is er sprake van onmiddellijke synchronisatie. Als er globale transacties beschikbaar zijn zou het tevens mogelijk zijn om de synchronisatie als onderdeel van de update transactie uit te voeren. Dit is echter niet waarschijnlijk omdat het in een data warehouse omgeving veelal niet mogelijk is de gegevensbronnen te manipuleren. Een variant op deze strategie is om het data warehouse te synchroniseren na een vastgesteld aantal updates groter dan één.

Er is sprake van vertraagde synchronisatie wanneer het data warehouse wordt bijgewerkt zodra de gebruiker een query uitvoert. Dit betekent dat van een gegeven vertraagd wordt geverifieerd of deze bijgewerkt is, namelijk pas als de gebruiker het gegeven opvraagt. Dit kan overhead voorkomen wanneer een view weinig wordt gebruikt. Ook deze strategie heeft een variant waarbij het aantal door de gebruiker uitgevoerde queries als drempel dient voor een synchronisatie.

Hybride synchronisatie is een samenstelling van meerdere strategieën. Segev et al. beschrijven een strategie die de periodieke synchronisatie combineert met de onmiddellijke synchronisatie [28].

Bij willekeurige synchronisatie wordt er op willekeurige tijdstippen gesynchroniseerd. Een mogelijk voordeel hiervan is dat de belasting van de gegevensbronnen verdeeld is. Een voorbeeld van willekeurige synchronisatie is dat er manueel wordt bepaald wanneer er een synchronisatie gestart wordt.

Bouzeghoub et al. beschouwen de veranderingen in de gegevensbron als een parameter die invloed heeft op de synchronisatie strategie van een data warehouse [10]. De veranderingen worden gemeten door het aantal update transacties in de gegevensbronnen. Een andere parameter die door hen wordt genoemd is het aantal queries dat wordt uitgevoerd door de gebruikers. Dey et al. gebruiken deze parameters bijvoorbeeld om de optimale synchronisatietijd van het data warehouse te berekenen [16].

Volgens Bouzeghoub et al. is de synchronisatie van een data warehouse is een belangrijk proces dat de kwaliteit van de verzamelde en geaggregeerde gegevens van de bronnen bepaalt [10]. De gegevensbronnen zijn hierbij de sturende factor, omdat

hier de updates plaatsvinden. Agrawal et al. verdelen updates die plaatsvinden in de gegevensbron in drie categorieën [1]:

1. Enkele update transacties waarbij elke update bij één gegevensbron plaatsvindt.
2. Lokale transactie op de gegevensbron waar een reeks van updates plaatsvindt als een enkele transactie. Echter zijn alle updates wel gericht aan één enkele gegevensbron.
3. Globale transacties waar de updates meerdere gegevensbronnen betreffen.

De update is echter pas het begin van het synchronisatieproces. Nadat de update heeft plaatsgevonden dienen, volgens Bouzeghoub et al., een aantal problemen opgelost te worden [10]:

- Detectie van veranderingen in de gegevensbronnen
- Berekenen en extraheren van deze veranderingen
- Vastleggen van de veranderingen

Deze punten zijn sterk afhankelijk van de functionaliteit en beschikbaarheid van de gegevensbron. Bouzeghoub et al. verdelen gegevenbronnen dan ook in verschillende groepen [10], waarbij een gegevensbron slechts in één groep kan vallen. Engström et al. onderscheiden karakteristieken voor gegevensbronnen: veranderingsbewust, veranderingsdetectie, deltabewust [17]. Dit maakt het mogelijk een gegevenbron onder meerdere karakteristieken in te delen.

Wanneer een gegevenbron veranderingsbewust is heeft deze het vermogen om, wanneer dit gevraagd wordt, aan te geven wanneer de laatste verandering heeft plaatsgevonden. Een voorbeeld van een veranderingsbewuste gegevenbron is een fileserver. Deze kan op verzoek het tijdstip van de laatste wijziging doorgeven.

Onder veranderingsdetectie wordt het vermogen om automatisch veranderingen te detecteren en rapporteren verstaan. Denk hierbij bijvoorbeeld aan triggers van een relationele database. Hierbij is het mogelijk op de hoogte gesteld te worden van een verandering in de database.

De laatste karakteristiek is deltabewust. Dit betreft het vermogen om automatisch of op verzoek de gemaakte veranderingen door te geven. Triggers kunnen behalve

veranderingsdetectie tevens van het type deltabewust zijn. Ze melden dan behalve dat er iets veranderd is tevens wat er veranderd is.

# 3 Kwaliteit

## 3.1 Inleiding

Kwaliteit is afgeleid van het Griekse woord voor hoedanig. Aristoteles (384 v. Chr. - 322 v. Chr.) beschreef kwaliteit als een onderdeel van het 'zijn' [3]. Kwaliteit bestaat volgens hem niet op zich, maar in dingen. Ter illustratie, in de samenstelling 'zwart paard' duidt zwart aan dat het een kwaliteit is van paard. Juran omschrijft kwaliteit als geschiktheid voor gebruik, waarbij de geschiktheid wordt beoordeeld door de gebruiker [22]. Crosby stelt dat kwaliteit wordt bereikt wanneer wordt voldaan aan de specificaties [14].

Men poogt al sinds voor onze huidige jaartelling een goede definitie te vinden voor kwaliteit. Behalve de genoemde definities zijn er nog vele anderen en zullen er wellicht nog vele volgen. Reeves et al. concluderen dan ook dat kwaliteit zo breed is en zoveel componenten omvat dat het onmogelijk is deze allemaal in kaart te brengen [27]. Dit hoofdstuk werpt licht op het begrip kwaliteit in een data warehouse. Er worden dimensies genoemd waaraan kwaliteit meetbaar is en hiervan wordt er één gekozen die het best aansluit bij het begrip 'op het juiste moment'.

## 3.2 Kwaliteit in een data warehouse

In een data warehouse omgeving zijn verschillende rollen te onderscheiden. Elke rol impliceert een andere visie op het begrip kwaliteit. Een gebruiker van het data warehouse, bijvoorbeeld een manager die beslissingen neemt op basis van de gegevens in het data warehouse, zal ander eisen stellen aan de kwaliteit dan een data warehouse administrator. Binnen dit onderzoek zal enkel worden ingegaan op de gebruiker van de gegevens in het data warehouse.

Een data warehouse biedt ondersteuning bij het maken van analyses en beter gefundeerde beslissingen (zie hoofdstuk 2). Het is voor de gebruiker van belang dat de gegevens waarop beslissingen zijn gebaseerd van voldoende kwaliteit zijn. De kwaliteit van de gegevens in het data warehouse heeft invloed op de kwaliteit van de analyses en beslissingen. Hierbij wordt aangenomen dat het uiteindelijk de gebruiker van de gegevens is die beslist of de kwaliteit afdoende is.

Jarke et al. geven aan dat de kwaliteit van gegevens in een data warehouse beïnvloed wordt door drie factoren [21]:

- Ontwerp van het data warehouse model
- Kwaliteit van de gegeven die in het data warehouse worden geladen
- Manipulatie van de gegevens in het data warehouse

Het data warehouse model is verantwoordelijk voor de (semantisch) correcte, complete en nuttige integratie van gegevensbronnen. Wanneer er bij het ontwerpen gefaald wordt om alle informatie in het model van het data warehouse te krijgen, is de kans groot dat de gegevens niet compleet of dubbelzinnig zijn. Als de semantiek van de bronnen verkeerd wordt geïnterpreteerd of als de verschillende bronnen niet goed worden geïntegreerd zal het data warehouse incorrecte gegevens bevatten. Verder is het van belang dat de integriteitregels gehandhaafd worden zodat het data warehouse geen incorrecte of betekenisloze gegeven bevat.

Logischerwijs zijn de gegevens in het data warehouse afhankelijk van de kwaliteit van de gegevens die gebruikt worden om het data warehouse te synchroniseren. Incorrecte gegevens die zijn opgeslagen in de gegevensbronnen kunnen in het data warehouse terecht komen. De gegevens worden in het data warehouse gebracht via een synchronisatieproces (zie paragraaf 2.5), welke een invloed op de kwaliteit kan hebben. Dit proces zorgt ervoor dat de gegevenbronnen correct worden geïntegreerd en filtert op gegevens die niet voldoen aan de integriteitregels. Dit proces kan tevens gebruikt worden om de correctheid van de gegevens te verifiëren en de kwaliteit hiervan te verbeteren.

De gegevens in een data warehouse (verzamellaag en presentatielaag, zie paragraaf 2.3) worden over het algemeen afgehandeld door een database management systeem (DBMS) en kunnen niet worden bijgewerkt door de gebruikers. De meest voorkomende manipulaties zijn aggregaties en reorganisaties van gegevens over verschillende dimensies, beide worden door het DBMS afgehandeld. Dit betekent dat de kwaliteit van de gegevens behouden wordt binnen het data warehouse en amper wordt beïnvloed door de beschreven manipulaties.

In dit onderzoek zal de nadruk gelegd worden op de kwaliteit van de gegevens die in het data warehouse worden geladen. De gebruiker van het data warehouse zal

deze gegevens beoordelen op kwaliteit. In de volgende paragraaf zal worden aangegeven op welke punten kwaliteit meetbaar te maken is.

## 3.3 Kwaliteitsdimensies

In het begin van dit hoofdstuk is reeds beschreven dat het onmogelijk is een algemene definitie van kwaliteit te geven. Volgens Wang et al. is de definitie van Juran (kwaliteit is geschiktheid voor gebruik) inmiddels algemeen geaccepteerd in de literatuur over kwaliteit [32]. Zij hebben de definitie van gegevenskwaliteit aangescherpt tot “gegevens die geschikt zijn voor gebruik door gegevensconsumenten”. In dit onderzoek zal deze definitie worden gehanteerd. Merk op dat gebruikers in de definitie als gegevensconsument aangeduid worden. De gegevens die aan de gebruiker van het data warehouse getoond worden zijn eigenlijk een product van het data warehouse (denk hierbij aan bijvoorbeeld een rapport). De gebruiker kan daarom worden gezien als de consument van dit product. Het data warehouse kan op zijn beurt beschouwd worden als de fabriek die het product vervaardigt. Dit principe wordt verder beschreven in hoofdstuk 4.

Davis schrijft reeds in 1974 dat gegevenskwaliteit eerder een relatieve dan een absolute term is die het best kan worden omschreven in de context van het eindgebruik [15]. Echter is de vraag hoe deze context omschreven kan worden. Wang et al. beschrijven dat er in de literatuur drie methoden worden gebruikt om gegevenskwaliteit te bestuderen: een intuïtieve, een theoretische en een empirische aanpak.

De intuïtieve aanpak wordt gekozen wanneer de onderzoekers zelf uit ervaring of intuïtie bepalen welke dimensies belangrijk zijn. Bij de theoretische aanpak wordt de nadruk gelegd op hoe gegevens ontoereikend worden gedurende de vervaardiging hiervan. Denk hierbij bijvoorbeeld aan een ontologische benadering van gegevenskwaliteit [31]. Tot slot is er de empirische aanpak. Bij deze aanpak wordt de gebruiker of consument van de gegevens betrokken bij het onderzoek. Er wordt onderzocht welke karakteristieken zij gebruiken om te bepalen of gegevens geschikt zijn voor gebruik.

In 1985 werd er door Ballou et al. reeds onderscheid gemaakt tussen een viertal intuïtieve dimensies van gegevenskwaliteit [7]. Zij behandelen binnen hun model de volgende dimensies:

- Nauwkeurigheid
  *De opgeslagen waarde is gelijk aan de werkelijke waarde*
- Volledigheid
  *Alle waarden van een bepaalde variabele zijn opgeslagen*
- Consistentie
  *De representatie van de waarde is in alle gevallen hetzelfde*
- Opportuniteit
  *De opgeslagen waarde is niet achterhaald*

Wang et al. hebben door middel van empirisch onderzoek vijftien dimensies gevonden, onderverdeeld in vier categorieën, die gebruikers belangrijk vinden [32]:

1. Intrinsieke gegevenskwaliteit
   - Nauwkeurigheid
     *De mate waarin gegevens correct, betrouwbaar en gegarandeerd foutloos zijn*
   - Objectiviteit
     *De mate waarin gegevens onbevooroordeeld en onpartijdig zijn*
   - Geloofwaardigheid
     *De mate waarin gegevens worden geaccepteerd als waar, echt en overtuigend*
   - Reputatie
     *De mate waarin gegevens worden vertrouwd of beschouwd in termen van hun bron of content*

2. Contextuele gegevenskwaliteit
   - Toegevoegde waarde
     *De mate waarin gegevens nuttig zijn en voordelen bieden door gebruik*
   - Relevantie
     *De mate waarin gegevens toepasbaar en bruikbaar zijn voor de taak voorhanden*
   - Opportuniteit
     *De mate waarin de leeftijd van de gegevens geschikt is voor de taak voorhanden*
   - Volledigheid
     *De mate waarin gegevens voldoende breedte, diepte en bereik hebben voor de taak voorhanden*
   - Juiste hoeveelheid gegevens
     *De mate waarin de kwantiteit of het volume van de beschikbare gegevens voldoende is*
3. Representatieve gegevenskwaliteit
   - Interpreteerbaarheid
     *De mate waarin gegevens in geschikte taal en eenheid worden gegeven en de definities van de gegevens duidelijk zijn*
   - Begrijpbaarheid
     *De mate waarin gegevens zonder ambiguïteit duidelijk en gemakkelijk te bevatten zijn*
   - Representatieve consistentie
     *De mate waarin gegevens altijd gepresenteerd worden in hetzelfde indeling en opmaak en compatible zijn met de vorige gegevens*
   - Bondige representatie
     *De mate waarin gegevens compact gerepresenteerd worden zonder overweldigend over te komen*
4. Toegankelijke gegevenskwaliteit
   - Toegankelijkheid
     *De mate waarin gegevens beschikbaar of makkelijk en snel te achterhalen zijn*
   - Toegangsbeveiliging
     *De mate waarin toegang tot gegevens kan worden beperkt en daarom veilig wordt gehouden*

De dimensies in de categorie intrinsieke gegevenskwaliteit zijn dimensies die eigenschappen zijn van de gegevens zelf. Ze beschrijven de waarde die de gegevens zelf hebben. De categorie contextuele gegevenskwaliteit heeft betrekking op de context waarin iets wordt uitgevoerd. De dimensies in deze categorie dienen te worden gezien in relatie tot de taak van de gebruiker. Representatieve gegevenskwaliteit staat voor de opmaak en de betekenis van de gegevens. Tot slot staat toegankelijke gegevenskwaliteit voor de toegankelijkheid van de gegevens zelf. Hierbij wordt gedoeld op hoe gemakkelijk de gebruiker zichzelf toegang kan verschaffen tot de gegevens. Dit heeft temeer betrekking op het gebied van informatiebeveiliging.

De intuïtieve dimensies die door Ballou et al. [7] gevonden waren sluiten aan bij de door middel van empirisch onderzoek gevonden dimensies van Wang et al. [32]. Het is reeds aangetoond dat de beschreven intuïtieve dimensies tegenhangers zijn van elkaar. Volledigheid is de tegenhanger van consistentie [6] en nauwkeurigheid die van opportuniteit [5][18].

De dimensie opportuniteit komt zowel in de beschrijving van de intuïtieve dimensies als bij de empirische dimensies voor. De omschrijving van opportuniteit in de intuïtieve dimensies (de opgeslagen waarde is niet achterhaald) sluit aan bij de bewering dat een gebruiker behoefte heeft aan bijgewerkte en betrouwbare informatie (zie paragraaf 1.2). De omschrijving van de empirische dimensie heeft betrekking op de leeftijd van de gegevens en in hoeverre de leeftijd van de gegevens invloed heeft op de taak voorhanden. Ook deze omschrijving sluit aan bij de genoemde bewering. In het vervolg van dit onderzoek zal verder daarom de dimensie opportuniteit verder worden uitgediept. In paragraaf 2.5 wordt gesproken over de versheid van gegevens, hiermee wordt eigenlijk gedoeld op de opportuniteit.

## *3.4 Opportuniteit*

Het woord opportuun betekent volgens de Van Dale ‘geschikt doordat het juiste moment gekozen is’ en opportuniteit betekent ‘geschiktheid van tijd of gelegenheid’. De kwaliteitsdimensie opportuniteit van gegevens geeft aan of de opgeslagen waarde niet achterhaald is. Anders gezegd bepaald deze dimensie of een gegeven op een geschikt moment aan de gebruiker wordt getoond.

Ballou et al. schrijven dat de opportuniteit van gegevens beïnvloed wordt door twee factoren, te weten actualiteit en vergankelijkheid [8]. Actualiteit heeft betrekking op de leeftijd van een gegeven dat wordt gebruikt om een informatieproduct te maken. Een informatieproduct is een gegeven dat voor de gebruiker waarde heeft. Met de tweede factor vergankelijkheid wordt er gedoeld op hoe lang een gegeven geldig is.

Bij sommige gegevens doet het er niet toe wat de leeftijd (actualiteit) van een bepaald gegeven is. Het feit dat op 1 februari 1953 de watersnoodramp in Nederland plaatsvond zal waar blijven ongeacht wanneer dat feit in het systeem terecht komt. Echter de waarde van een aandeel van gisteren kan drastisch verschillen van de waarde van vandaag.

De actualiteit is een eigenschap van het opnamemoment van de gegevens en is in geen geval een intrinsieke eigenschap. De vergankelijkheid daarentegen is wel intrinsiek en is daarom onafhankelijk van bijvoorbeeld het opnamemoment. Een pak melk heeft bijvoorbeeld een houdbaarheidsdatum (vergankelijkheid). Deze zal niet veranderen ongeacht hoe lang de melk al in de koelkast staat.

Om het begrip opportuniteit meetbaar te kunnen maken, dient eerst bepaald te worden hoe de twee factoren actualiteit en vergankelijkheid zich tot de opportuniteit verhouden. Uiteraard dienen ze beide in dezelfde eenheid gemeten te worden.

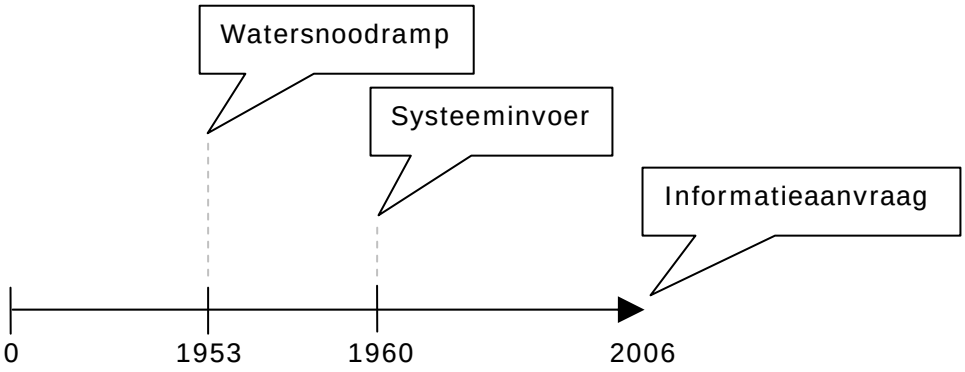
Het is gebruikelijk om actualiteit te meten door middel van timestamps. Met dit hulpmiddel is te bepalen wat de actualiteit van een gegeven is. Ballou et al. geven aan dat de actualiteit een functie is van verschillende factoren: wanneer het informatieproduct bij de consument afgeleverd is (aflevertijd); wanneer het gegeven is ingevoerd (invoertijd); en hoe oud het gegeven was toen het werd ingevoerd (leeftijd) [8]. Hiervoor hebben ze de volgende definitie van actualiteit opgesteld:

*Actualiteit = (Aflevertijd – Invoertijd) + Leeftijd*

Merk op dat het resultaat van het verschil tussen de haakjes aangeeft hoe lang het gegeven in het systeem staat. De leeftijd staat voor het verschil tussen wanneer de gebeurtenis werkelijk plaatsvond en wanneer deze in het systeem is ingevoerd.

In de nacht van 31 januari op 1 februari in het jaar 1953 braken in Zeeland en Zuid-Holland vele dijken door en kwamen vele stukken land onder water te staan.

Deze tragische gebeurtenis, waarbij veel mensen de dood vonden, staat nu bekend als de watersnoodramp. In figuur 3.1 staat een tijdlijn van een systeem. Hierin is te zien dat de watersnoodramp in 1953 plaats vond en dat deze is ingevoerd in het systeem in 1960. Een gebruiker van het systeem heeft in 2006 het gegeven aangevraagd. Er wordt hier aangenomen dat het moment van aanvraag identiek is aan het moment dat de persoon in kwestie het gegeven onder ogen krijgt: de aflevertijd. Het gegeven is ingevoerd in 1960 en uit het verschil van de invoertijd en de tijd dat de ramp plaatsvond $(1960-1953)$ volgt de leeftijd van 7 jaar. De actualiteit van het gegeven is nu als volgt te berekenen $(2006-1960)+7=53$. Het is nu tevens af te leiden dat het gegeven zich 46 jaar in het systeem bevind. Dit is af te leiden door het gedeelte tussen de haakjes op te lossen: $2006-1960=46$.




**Figuur 3.1 Tijdlijn watersnoodramp**

De vergankelijkheid van een gegeven is te vergelijken met de houdbaarheidsperiode van een product. Beperkt houdbare producten zoals voedsel worden alleen tijdens een bepaalde tijd voor de volledige prijs verkocht. De kans dat het product in die tijd bederft wordt klein geacht. Op deze manier kunnen leveranciers van ruwe gegevens bepalen hoe lang de betreffende gegevens geldig blijven. Dit getal, de houdbaarheidsperiode, is volgens Ballou et al. de manier om vergankelijkheid te meten [8]. De houdbaarheidsperiode van gegevens zoals aandelenstanden zijn zeer kort. Daarentegen is de houdbaarheidsperiode van gegevens zoals de datum van de watersnoodramp oneindig.

Zoals beschreven is vergankelijkheid een intrinsieke eigenschap van een gegeven. Dit betekent dat de houdbaarheidsperiode verschilt per product. De verantwoordelijkheid om de houdbaarheidsperiode te bepalen ligt bij de consumenten van het informatieproduct en/of de domeinspecialist.

Uit de definities van actualiteit blijkt dat opportuniteit afhankelijk is van wanneer het informatieproduct bij de consument wordt afgeleverd. Hieruit volgt de belangrijke aanname dat opportuniteit onbekend is tot de aflevering.

Ballou et al. hebben aan de hand van de genoemde factoren van opportuniteit een definitie opgesteld om opportuniteit meetbaar te maken. Ze hebben ervoor gekozen om opportuniteit uit te drukken in een absolute waarde tussen de 0 en 1. Dit maakt het mogelijk opportuniteitswaarden met elkaar te vergelijken. Hierbij geldt de waarde 1 voor gegevens die voldoen aan opportuniteitstandaard en de waarde 0 voor gegevens die helemaal niet voldoen. Anders gezegd zijn de gegevens bij een waarde van 0 compleet achterhaald en bij een waarde van 1 maximaal recent. De actualiteit of totale leeftijd van gegevens is goed of slecht afhankelijk van de vergankelijkheid (geldigheidsperiode). Een grote waarde voor de actualiteit is niet belangrijk als de vergankelijkheid onbeperkt is. Echter, een kleine waarde voor de actualiteit kan desastreus voor de kwaliteit zijn als de geldigheidsperiode erg kort is. Dit duidt er op dat opportuniteit een functie is van de verhouding tussen actualiteit en vergankelijkheid. De volgende functie om de opportuniteit te berekenen wordt gegeven:

$$opportuniteit = \begin{cases} 1 & als\ vergankelijkheid = \infty \\ (1 - \frac{actualiteit}{vergankelijkheid})^s & als\ actualiteit \leq vergankelijkheid \\ 0 & anders \end{cases}$$

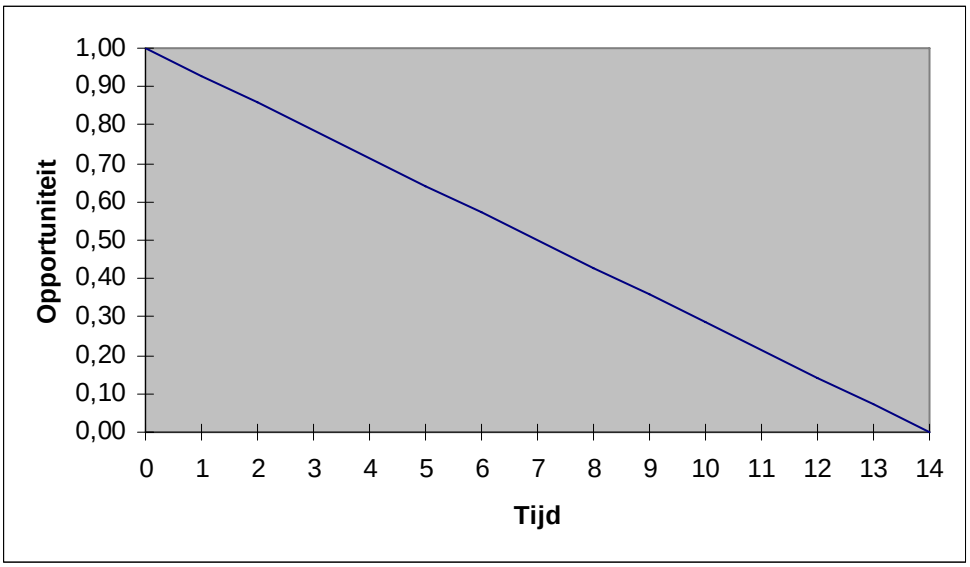
Deze functie is gebaseerd op de functie voor opportuniteit van Ballou et al. [8]. Behalve de verandering in notatie is er gekozen om een oneindige vergankelijkheid bij de functie te betrekken. Zoals reeds beschreven bestaan er gegevens die oneindig geldig blijven. Dit betekent dat wanneer die gegevens geraadpleegd worden, ze altijd een maximale opportuniteit hebben. Dit blijkt uit de eerste regel uit de functie. De tweede regel uit de functie is de belangrijkste omdat deze de opportuniteitswaarde berekent die tussen de 0 en 1 ligt. Wanneer de actualiteit kleiner of gelijk is aan de vergankelijkheid wordt deze regel gebruikt. Zodra de actualiteit groter is dan de vergankelijkheid betekent dit dat het gegeven over de geldigheidsduur is. Uit regel drie van de functie blijkt terecht dat dit een opportuniteit van 0 tot gevolg heeft.

De exponent s is een parameter die ervoor zorgt dat de gevoeligheid van opportuniteit ten opzichte van de verhouding tussen actualiteit en vergankelijkheid vastge-

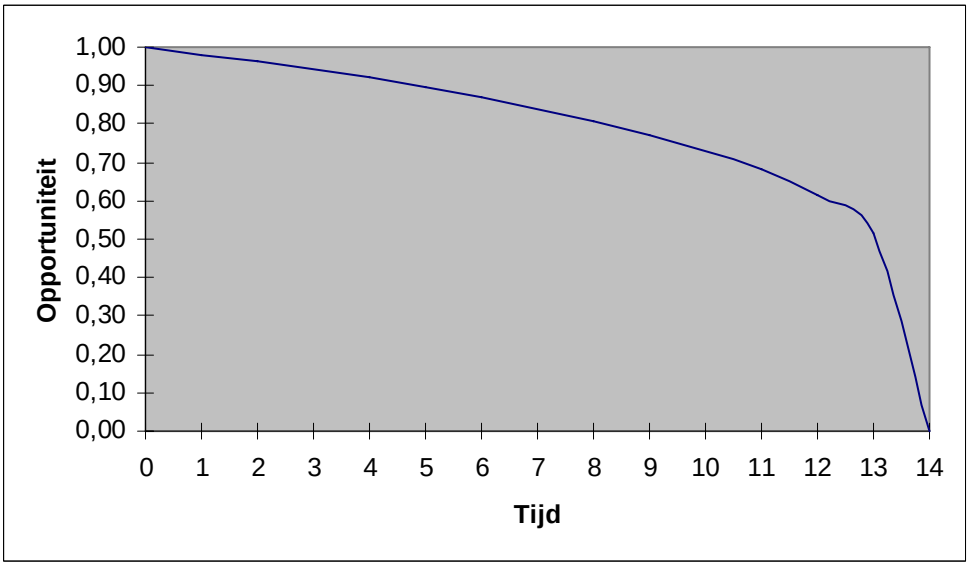
steld kan worden. Bij een hoge vergankelijkheid (korte houdbaarheid) is de uitkomst van de verhouding groot, maar bij een lage vergankelijkheid (lange houdbaarheid) is de uitkomst laag. De waarde van deze exponent (lees gevoeligheidsfactor) is afhankelijk van wat er gemeten wordt en is daarom een inschatting van de domeinspecialist. Bijvoorbeeld, stel dat de opportuniteit zonder de exponent (hetzelfde als een exponent met de waarde 1) gelijk aan 0,73 is. Wanneer de exponent de waarde 2 krijgt, wordt de opportuniteit 0,53. Oftewel een hoge exponent de gegevens sneller achterhaald raken. Bij een exponent van 0,5 is de opportuniteit 0,85. Anders gezegd, bij een lage exponent zullen de gegevens minder snel achterhaald zijn.

In figuur 3.2 staat een mogelijke grafiek afgebeeld van een willekeurig gegeven. Hierbij is de tijd (x-as) uitgezet tegen de opportuniteit (y-as). Op tijdstip 0 is te zien dat de opportuniteit nog 1 (maximaal) is, terwijl deze op tijdstip 14 gedaald is tot 0. Hieruit blijkt dat het gegeven een vergankelijkheid heeft van 14. Het is goed te zien dat de opportuniteit lineair afneemt over de tijd. Dit hoeft echter niet altijd het geval te zijn. In figuur 3.3 is hetzelfde gegeven met een exponent van ¼ gegeven. Hieruit valt op te maken dat de opportuniteit geleidelijker afneemt.

De exponent kan inzichtelijker worden gemaakt door middel van een voorbeeld. Neem een willekeurig een willekeurig apparaat uit de consumentenelektronica, bijvoorbeeld een mp3-speler. Wanneer hiervan een model op de markt wordt gebracht geniet deze de maximale populariteit. Deze populariteit zal geleidelijk afnemen, maar wanneer er een nieuw model geïntroduceerd wordt zal het nieuwe model het oude aftroeven. In figuur 3.3 kan de tijd nu worden geïnterpreteerd als maanden. Wanneer de mp3-speler wordt geïntroduceerd (maand 0) geniet deze nog een maximale populariteit. Na omstreeks 13 maanden is het nieuwe model op de markt gebracht en vrijwel onmiddellijk neemt de populariteit van het oude model af.




**Figuur 3.2 Geen exponent**




**Figuur 3.3 Exponent ¼**

# 4 Model data warehouse

## 4.1 Inleiding

Om formeel over een data warehouse te kunnen redeneren dient er een model gevonden te worden waarin de architectuur de gegevensverwerking van een data warehouse wordt beschreven. Dit hoofdstuk zal zo'n model beschrijven. Er wordt tevens een mogelijkheid beschreven om opportuniteit binnen het model te berekenen.

## 4.2 Informatie vervaardigingsysteem

Ballou et al. hebben een model ontwikkeld dat een informatie vervaardigingsysteem beschrijft [8]. Zij definiëren een informatie vervaardigingsysteem als een informatiesysteem dat voorgedefinieerde informatieproducten vervaardigd. Een informatieproduct gebruiken zij om aan te geven dat de uitvoer van informatie waarde heeft en wordt doorgegeven aan de consument.

Een data warehouse kan worden beschouwd als een informatie vervaardigingsysteem. Bij het synchroniseren van een data warehouse wordt er uit één of meer gegevensbronnen via een proces nieuwe informatie gemaakt. Dit proces herhaald zich één of meerdere keren. In figuur 2.5 zijn deze processen afgebeeld. Voorbeelden hiervan zijn adapteren, aggregeren en extraheren. Deze informatie zal uiteindelijk bij de gebruiker (consument) arriveren in de vorm van bijvoorbeeld een rapport of grafiek (informatie product). De toegangsapplicatie neemt de taak op zich om het informatieproduct op aanvraag aan de gebruiker te tonen. Dit opvragen van de informatie uit de data marts en het tonen aan de gebruiker kan worden beschouwd als een proces dat een informatieproduct oplevert.

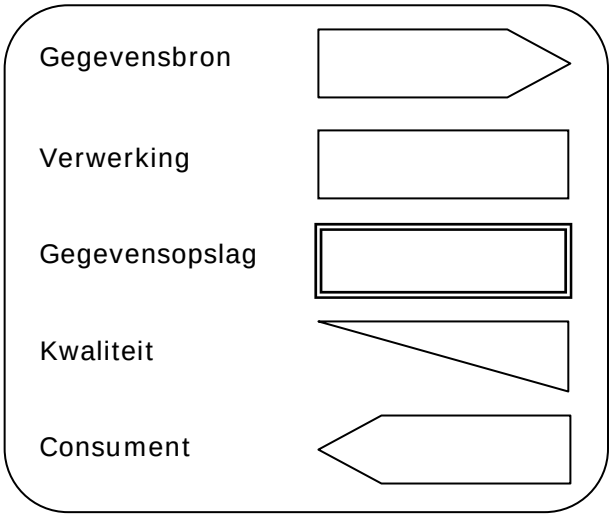
Een informatie vervaardigingsysteem heeft veel overeenkomsten met de fabricage van een fysiek product. Door hier een verband tussen te leggen kunnen volgens Ballou et al. veel van de concepten en procedures van product kwaliteitscontrole toegepast worden op het produceren van betere informatieproducten [8]. Volgens Shankaranarayan et al. zijn ruwe materialen, opslag, verwerking, inspectie, bewerking en verpakking (opmaak) net zoals bij een fysiek product toepasbaar op een informatieproduct [29]. Informatieproducten zijn volgens Ballou et al. dan ook standaard producten die kunnen worden geassembleerd in een productielijn [8]. Verder

denken zij dat meerdere informatieproducten een subset aan processen en informatie invoer kunnen delen en aldus gecreëerd kunnen worden in een enkele productielijn met minimale variaties waardoor de informatieproducten zich kunnen onderscheiden. Hierbij wordt nogmaals de overeenkomst met een data warehouses benadrukt. Een data warehouse wordt telkens gesynchroniseerd via standaard stappen. Elk informatieproduct dat de consument opvraagt wordt opgebouwd uit de informatie die aanwezig is in de data marts.

Hoewel het erg nuttig is om de overeenkomsten tussen de fabricage van een fysiek product en van een informatieproduct te belichten zijn er ook verschillen. Een significant verschil is dat gegevens niet geconsumeerd worden. Een fysiek product kan over het algemeen op een of andere manier uitgeput worden. Bijvoorbeeld als het onderdeel is van de inventaris wordt het na een bepaalde tijd afgeschreven. Bij gegevens is dit niet het geval. Deze worden wel gebruikt, maar niet echt geconsumeerd.

## 4.3 Het model

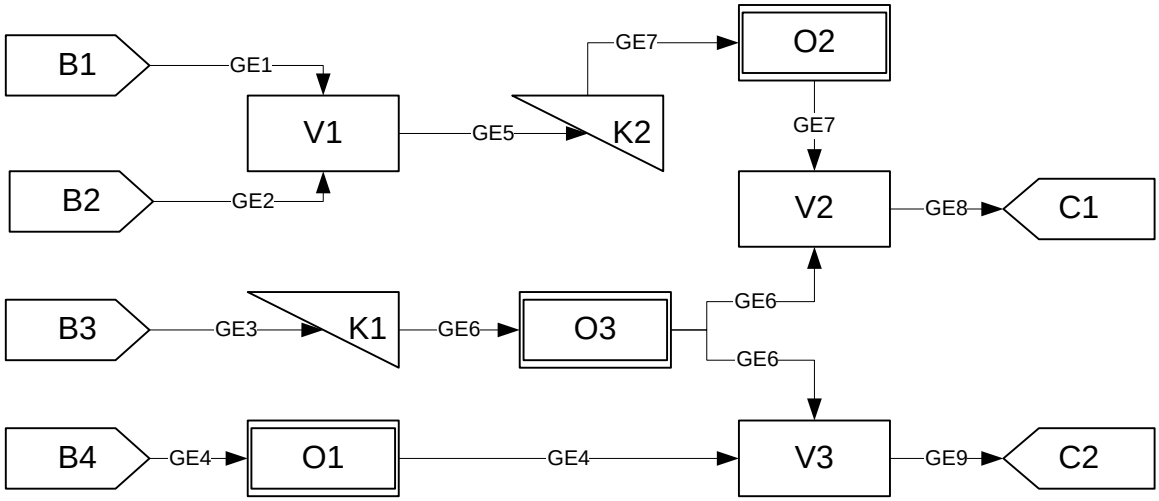
Volgens Shankaranarayan et al. is het bij goed gegevenskwaliteit management noodzakelijk de persoon verantwoordelijk voor de beslissingen te informeren over de kwaliteit van de gegevens die gebruikt worden en/of deze de mogelijkheid te geven dit te meten [29]. Een belangrijke stap in het meetbaar maken van de kwaliteit is inzicht te verkrijgen in het proces dat plaatsvindt zodra de gegevens in het data warehouse worden gesynchroniseerd en hoe deze gegevens bij de consument terecht komen. Ballou et al. hebben een model, het informatie vervaardigingsysteem, ontworpen dat beschrijft hoe vanuit de gegevensbron(nen) een informatieproduct wordt samengesteld [8]. Dit model, gebaseerd op het data flow diagram (DFD), is erg geschikt om ditzelfde proces te beschrijven in een data warehouse. In figuur 4.1 staan de componenten van dit model weergegeven.




**Figuur 4.1 Onderdelen informatie vervaardigingssysteem**

Het gegevensbron blok staat voor alle mogelijke gegevensbronnen van ruwe gegevens invoer. Elk gegevensbron blok is een gespecialiseerd verwerkingsblok dat geen voorganger heeft. Een gegevensbron kan dan ook verschillende typen ruwe gegevens voortbrengen. Het verwerking blok voegt waarde toe aan de ruwe gegevens door ze te manipuleren of combineren. De rol van de gegevensopslag is om gegevens op te slaan om ze wanneer dit noodzakelijk is beschikbaar te stellen voor verdere verwerking. Het kwaliteit blok verhoogt de kwaliteit van gegevens. De gegevenseenheid die eruit komt heeft een gelijke of hogere kwaliteit dan degene die erin ging. Tot slot staat het consument blok voor de uitvoer of het informatieproduct van het systeem. De consument is het einde van de gegevensstroom en kent geen opvolger. De invoer van de consument is de laatste gegevenseenheid van het systeem: het informatieproduct.

Tussen de blokken worden pijlen gebruikt om de bron en bestemming van een gegevenseenheid aan te geven. Bij een gegevenseenheid kan gedacht worden aan bijvoorbeeld een nummer of een record. Het is een ondeelbare eenheid met dezelfde kwaliteitsdimensies. Een gegevensbron zal altijd een ruwe gegevenseenheid als uitvoer hebben en een consument een informatieproduct als invoer. Een ruwe gegevenseenheid start in het systeem vanuit de gegevensbron en zal na verschillende verwerkingen eindigen als onderdeel van een informatieproduct.




**Figuur 4.2 Informatie vervaardigingsysteem**

In figuur 4.2 staat een voorbeeld van een model van een informatie vervaardigingsysteem. In dit voorbeeld staan vier ruwe gegevenseenheden (GE1, GE2, GE3, GE4) die hun oorsprong hebben in de gegevensbronnen (B1, B2, B3, B4). Er zijn twee gegevenseenheden (GE6, GE7) die worden gevormd nadat ze één van de twee kwaliteitsblokken (K1, K2) zijn gepasseerd. In het figuur staan drie verwerkingsblokken (V1, V2, V3) die een gelijk aantal gegevenseenheden opleveren (GE5, GE8, GE9). Er zijn drie gegevensopslagblokken (O1, O2, O3). Deze hebben als enige blok met uitvoer geen nieuwe gegevenseenheid als uitvoer. De invoer is identiek aan de uitvoer. Tot slot zijn er nog twee consumenten (C1, C2).

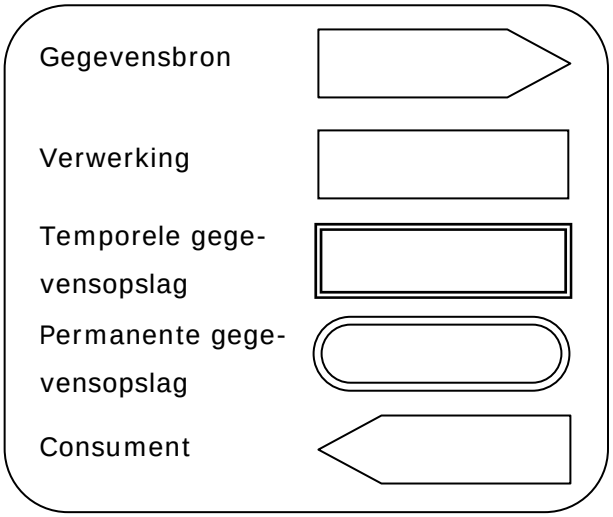
Verwerking V1 vormt de nieuwe gegevenseenheid GE5 uit de twee ruwe gegevenseenheden GE1 en GE2. De overige verwerkingsblokken combineren tevens twee gegevenseenheden tot een enkele. Het kan echter ook een heel ander soort verwerking zijn waarbij er één gegevenseenheid als invoer is en één gegevenseenheid als uitvoer. Zoals te zien is bij gegevensopslagblok O3 kunnen er meerdere gegevenseenheden uit een opslagblok komen. Verder valt hierbij op dat er bij een opslagblok, in tegenstelling tot een kwaliteitsblok of verwerkingsblok, geen nieuwe gegevenseenheid wordt gecreëerd. Dit is logisch omdat bij opslag er geen wijzigingen worden aangebracht in de gegevenseenheid.

## 4.4 Modificatie model

In het model van Ballou et al. wordt er geen onderscheid gemaakt tussen soorten gegevensopslag [8]. Er is in een data warehouse echter een verschil tussen een temporele gegevensopslag en een permanente gegevensopslag. Dit verschil blijkt uit de onderdelen van een data warehouse beschreven in paragraaf 2.3. In de verzamellaag worden de gegevens uit de bronnen verzameld en vanaf daar gepropageerd naar de presentatielaag. Hierbij worden gegevens tijdelijk opgeslagen. Wanneer deze worden gepropageerd, verdwijnen ze tevens uit de gegevensopslag. In de presentatielaag worden gegevens opgeslagen die continu geraadpleegd kunnen worden. Wanneer de gegevens gepropageerd worden (bijvoorbeeld door middel van een query) blijven ze bestaan in de gegevensopslag en zijn klaar om nogmaals geraadpleegd te worden. Uit figuur 2.5 blijkt tevens een onderscheid tussen twee soorten gegevensopslag. Zo is de gegevensopslag na het extraheren, of die na het integreren een temporele gegevensopslag. De gegevensopslag na het adapteren is een permanente gegevensopslag.

Om het onderscheid te maken tussen de temporele gegevensopslag en de permanente gegevensopslag wordt het model uitgebreid met een nieuw blok, te weten de permanente gegevensopslag. Het huidige blok voor gegevensopslag krijgt een andere betekenis. Dit staat nu voor een temporele gegevensopslag.

Er is gekozen om behalve een blok toe te voegen er tevens één te verwijderen. Er is in paragraaf 3.4 reeds aangegeven dat de opportuniteit van een informatieproduct pas bekend is zodra het is afgeleverd. Dit betekent dat het pas aan het eind te meten is. Een kwaliteitsblok kan hierom niets veranderen aan de opportuniteit. Bovendien is het mogelijk een kwaliteitsblok te modelleren met een verwerkingsblok. Deze verwerking kan als taak de verbetering van de kwaliteit van de gegevenseenheid hebben. Door het weglaten van het blok voor de kwaliteit verliest het model niets aan kracht om het data warehouse te modelleren. Figuur 4.3 geeft een weergave van de gewijzigde onderdelen van het model.




**Figuur 4.3 Gewijzigde onderdelen informatie vervaardigingssysteem**

## 4.5 Casus model

Om het model uit de vorige paragraaf wat te verduidelijken is er een casus opgesteld. Hierbij is met opzet gekozen voor een simpele weergave van een data warehouse.

Het bedrijf Konzultant verhuurd zijn medewerkers aan andere bedrijven om daar projecten te doen. De sales afdeling is verantwoordelijk voor het werven van nieuwe projecten bij (nieuwe) klanten. Zodra ze een nieuw project hebben geregeld wordt ingeschat hoeveel uren nodig zijn om het project succesvol af te ronden. Het project, de klant en het uurtarief dat de klant in rekening wordt gebracht worden ingevoerd in hun eigen sales database. De medewerkers houden zelf in een webapplicatie bij hoeveel uren ze voor welk project bij welke klant hebben gewerkt. Deze webapplicatie maakt gebruik van een eigen database. Tot slot heeft Konzultant nog een HRM afdeling die bijhouden wat de kosten van het personeel zijn. Zo staat hier het jaarsalaris in opgeslagen.

In figuur 4.4 staat een weergave van de verschillende gegevensbronnen die Konzultant heeft. Deze bronnen zijn reeds individueel benaderbaar, maar door het combineren van de gegevens in de verschillende bronnen kunnen er nieuwe vragen worden beantwoord die eerst niet mogelijk waren. Het is dan bijvoorbeeld mogelijk

in kaart te brengen wat de winst is, hoe goed projecten worden ingeschat, hoeveel projecten er per klant zijn, wat de opbrengst per medewerker is, etc.

| Urenregistratie |
|---|
| Medewerker |
| Project |
| Uren |
| Klant |

| HRM |
|---|
| Medewerker |
| Jaarsalaris |
| |
| |

| Sales |
|---|
| Project |
| Klant |
| Uurtarief |
| |

**Figuur 4.4 Gegevensbronnen Konzultant**

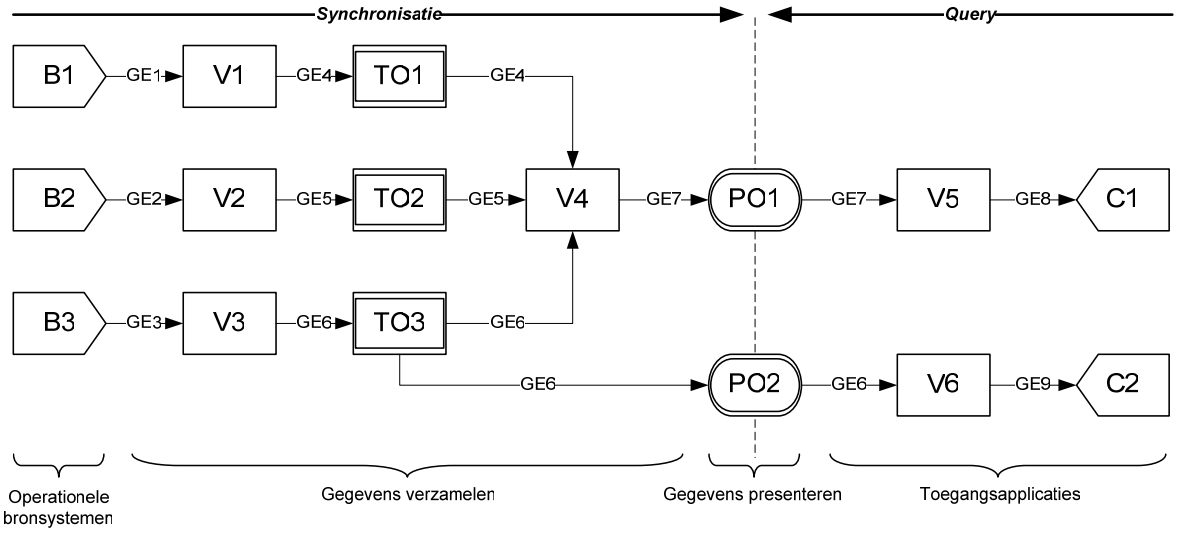
Konzultant heeft een data warehouse laten ontwerpen dat eruit ziet zoals afgebeeld in figuur 4.5. B1, B2 en B3 staan respectievelijk voor de gegevensbronnen van de urenregistratie, HRM en sales. De verwerkingen V1, V2 en V3 zijn het extraheren van de gegevens uit de bronnen. Deze worden vervolgens letterlijk gekopieerd naar de temporele gegevensopslag TO1 t/m TO3. Verwerking V4 integreert de drie bronnen en propageert dit naar de permanente gegevensopslag PO1 (data mart). B3 wordt vanuit TO3 behalve naar V4 tevens naar PO2 gepropageerd. Er zijn binnen het model van het data warehouse dan ook twee data marts beschikbaar, PO1 en PO2. Hier is tevens de scheidingslijn (stippellijn) te zien in het figuur. Deze lijn geeft aan waar de synchronisatie eindigt en de door de gebruiker geïnitieerde query begint. V5 en V6 staan hier voor het verwerkingsproces van de door de consument geïnitieerde queries. Consument C1 staat voor de directeur van Konzultant die wilt weten wat de winst van zijn bedrijf is. C2 staat voor een consument van de afdeling sales die wil weten welke projecten er op dit moment lopen.

De pijlen tussen de blokken visualiseren de gegevenseenheden die de blokken als invoer of uitvoer hebben. Zo is gegevenseenheid GE7 de uitvoer van verwerking V4 en de invoer van permanente gegevensopslag PO1. Het data warehouse kent drie ruwe gegevenseenheden:GE1, GE2 en GE3. Uit deze ruwe gegevenseenheden worden uiteindelijk de twee informatieproducten van het data warehouse gecreëerd: GE8 en GE9.

Als aangenomen wordt dat GE4, GE5 en GE6 respectievelijk de tabellen van de Urenregistratie, HRM en Sales bevat, dan kan verwerking deze samenvoegen tot één tabel. Deze tabel wordt als gegevenseenheid GE7 gepropageerd naar permanente gegevensopslag PO1. Consument C1 (de directeur) wilt weten wat de winst van zijn

bedrijf is en doet een verzoek in zijn toegangsapplicatie. Die voert een query (verwerking V5) uit. Verwerking V5 heeft de gehele tabel uit PO1 als invoer en selecteert hieruit de winst van Konzultant door bijvoorbeeld (*uren* * *uurtarief*) – *jaarsalaris* uit te voeren. Deze wordt als gegevenseenheid GE8 getoond aan de directeur. GE8 is daarom een informatieproduct.

Onderaan figuur 4.5 staan de onderdelen uit paragraaf 2.3 weergegeven. Hierbij wordt nogmaals duidelijk dat door middel van het model het volledige data warehouse in kaart kan worden gebracht.




**Figuur 4.5 Data warehouse Konzultant**

## 4.6 Opportuniteit in het model

Nu bekend is hoe opportuniteit berekend kan worden (zie paragraaf 3.4) en hoe het model van een data warehouse eruit ziet (zie paragraaf 4.4) kunnen deze twee gecombineerd worden.

Aan elke informatieproduct uit het model kan een opportuniteitswaarde gekoppeld worden. Het informatieproduct is een uitvoer van een bepaalde verwerking met verschillende invoeren. Deze invoeren kunnen op hun beurt weer ontstaan zijn uit een andere verwerking. Kort gezegd is elk informatieproduct afhankelijk van verschillende fasen van verwerking en een bron van ruwe gegevenseenheden.

Aan elke uitvoer van informatie in het model kan een opportuniteitswaarde gekoppeld worden. Hierbij wordt er gedoeld op de uitvoer van de blokken voor verwerking, temporele gegevensopslag en permanente gegevensopslag.

Een verwerkingsblok kan één of meerdere gegevenseenheden als invoer hebben. Wanneer er meerdere invoeren zijn zal elke invoer een eigen opportuniteitswaarde hebben. Zodra deze invoeren door het verwerkingsblok worden gecombineerd is er slechts een enkele gegevenseenheid als uitvoer. De vraag is echter wat de opportuniteit van deze uitvoer is. Het antwoord op deze vraag is afhankelijk is van het soort verwerking en de gegevenseenheden. Er is dan ook geen correcte allesomvattende oplossing. Echter er kunnen wel wat uitspraken over worden gedaan.

Wanneer uitvoer $y$ het verschil is tussen de invoerwaarden $x_1$ en $x_2$, bijvoorbeeld $y = x_1 - x_2$, kan de opportuniteit niet zomaar als het verschil tussen de opportuniteit van $x_1$ en $x_2$ worden genomen. Oftewel het is niet aan te raden om de verwerking die wordt toegepast op de invoeren uit te voeren op de opportuniteit van de invoeren om zo de opportuniteit van de uitvoer te bepalen. Als voorbeeld wordt aangenomen dat $x_1 = 10$ en $x_2 = 100$. Zo zal bijvoorbeeld $y = 10 + 100$ een andere opportuniteit geven dan $y = 10 \cdot 100$. In het voorbeeld wordt duidelijk dat het verschilt in hoeverre $x_1$ invloed heeft op de opportuniteit.

Verder is het logisch om aan te nemen dat wanneer bij de invoer elke gegevenseenheid een opportuniteit van 1 (uitmuntend) heeft, de opportuniteit van de uitvoer tevens 1 is. Ditzelfde geldt eveneens wanneer de gegevenseenheden opportuniteit 0 hebben. De verwachting is dan dat de uitvoer tevens een waarde van 0 zal hebben. Volgens deze aanname hebben Ballou et al. de volgende formule opgesteld die de opportuniteit bepaald van de uitvoer van een verwerkingsblok met meerdere invoeren [5]:

$$O(y) = \frac{\sum_{i=1}^{n} w_i \cdot T(x_i)}{\sum_{i=1}^{n} w_i} \quad \textit{waarbij} \quad w_i = \left|\frac{\partial f}{\partial x_i}\right| \cdot \left|x_i\right|$$

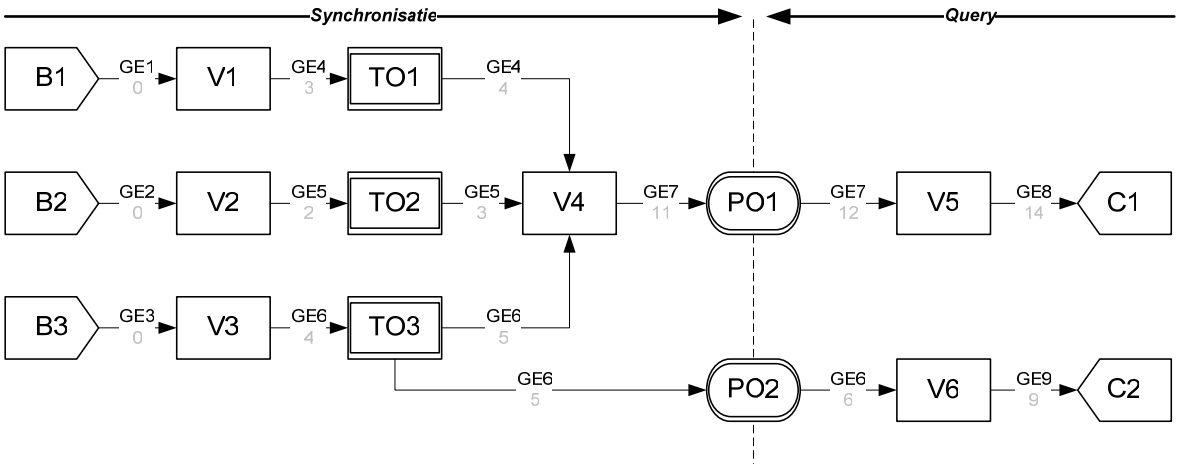
De bovenstaande vergelijking is een gewogen gemiddelde van de $O(x_i)$ (opportuniteit van de invoer $x_i$). Hierbij wordt de opportuniteit van uitvoer $y$ bepaald door alle invoeren $x$ te nemen.

Een verwerkingsblok hoeft niet noodzakelijk meerdere gegevenseenheden als invoer te hebben. Het is tevens mogelijk dat er een enkele gegevenseenheid in gaat. Denk hierbij bijvoorbeeld aan het sorteren van een recordset. Hierbij zal de uitvoer de opportuniteit van de invoer overnemen. Het sorteren van een recordset kost wel tijd, maar doordat de opportuniteit pas wordt bepaald zodra het product bij de consument arriveert zal dit geen invloed hebben.

De opportuniteit van de invoer van een permanente gegevensopslag of een temporele gegevensopslag is, net zoals een verwerking met één invoer, gelijk aan die van de uitvoer. Uiteraard kost het tijd om de kwaliteit van een gegeven te verbeteren of om een gegeven op te slaan. Echter komt deze tijd tot uiting op het moment dat het product bij de consument arriveert. Hoewel de overige kwaliteitsdimensies uit paragraaf 3.3 hier niet worden behandeld, kan worden gemeld dat een permanente gegevensopslag of een temporele gegevensopslag tevens geen invloed hebben op de kwaliteit van de gegevenseenheid (aannemend dat er geen externe invloeden zijn die deze beïnvloeden).

## 4.7 Casus opportuniteit

Gegevenseenheden GE8 en GE9 zijn de twee informatieproducten van het data warehouse van het in paragraaf 4.5 beschreven fictieve bedrijf Konzultant. Om de opportuniteit van deze producten te berekenen dient er meer bekend te zijn over de ruwe gegevenseenheden en de individuele blokken van het data warehouse. In figuur 4.6 staat nogmaals het data warehouse van Konzultant afgebeeld. Hierbij staat onder elke gegevenseenheid wat de aflevertijd is op dat moment.




**Figuur 4.6 Aflevertijden Konzultant**

Uit de bronnen B1, B2, B3 komen respectievelijk de ruwe gegevenseenheden GE1, GE2, GE3. Om van de informatieproducten de actualiteit te kunnen berekenen zijn behalve de aflevertijd tevens de invoertijd en leeftijd noodzakelijk. Om de functie voor opportuniteit verder in te kunnen vullen zal tevens de vergankelijkheid en exponent bekend dienen te zijn. Tabel 4.1 geeft een overzicht van de noodzakelijke eigenschappen van de ruwe gegevenseenheden.

| Invoer | Bron | Invoertijd | Leeftijd | Vergankelijkheid | Exponent |
|---|---|---|---|---|---|
| GE1 | B1 | 1 | 3 | 60 | 1 |
| GE2 | B2 | 0 | 5 | 40 | 2 |
| GE3 | B3 | 4 | 7 | 20 | ½ |

**Tabel 4.1 Ruwe gegevenseenheden**

Elke verwerking, permanente gegevensopslag en temporele gegevensopslag neemt tijd in beslag om de activiteit uit te voeren. In tabel 4.2, tabel 4.3 en tabel 4.4 staat hoeveel tijdseenheden elke activiteit kost. Net zoals bij de verwerking gaat het er bij de permanente gegevensopslag en temporele gegevensopslag om hoeveel tijd het kost iets op te slaan en niet hoe lang iets wordt opgeslagen.

In het model van Ballou et. al wordt nog een vertraging vermeld [8]. Dit aspect is hier achterwege gelaten, omdat enige vertraging bij de tijd die het kost om de activiteit uit te voeren kan worden opgeteld.

| Verwerking | Tijd |
|---|---|
| V1 | 3 |
| V2 | 2 |
| V3 | 4 |
| V4 | 6 |
| V5 | 2 |
| V6 | 3 |

**Tabel 4.2 Verwerkingstijden**

| Permanente gegevensopslag | Tijd |
|---|---|
| TO1 | 1 |
| TO2 | 1 |
| TO3 | 1 |

**Tabel 4.3 Tijden temporele gegevensopslagtijden**

| Temporele gegevensopslag | Tijd |
|---|---|
| PO1 | 1 |
| PO2 | 1 |

**Tabel 4.4 Tijden permanente gegevensopslagtijden**

Nu bekend is hoeveel tijdseenheden elke activiteit kost en wat de invoertijd, leeftijd, vergankelijkheid en exponent van de ruwe gegevenseenheden is, kan de opportuniteit van de informatieproducten berekend worden. In tabel 4.5 staan de tijdseenheden vermeld wanneer een gegevenseenheid bij een activiteit of de gegevensconsument aankomt; de aflevertijden. Ter verduidelijking zijn deze tijden tevens in figuur 4.6 in het grijs onder de gegevenseenheden vermeld.

Zoals valt op te maken uit tabel 4.5 heeft gegevenseenheid GE7 een aflevertijd van 11 tijdseenheden. Deze komt tot stand doordat GE1 op tijdstip 0 bij V1 aankomt. Deze verwerking duurt 3 tijdseenheden en komt dan aan bij temporele gegevensopslag TO1. TO1 heeft 1 tijdseenheid nodig voor het opslaan. GE4 komt daarom na 4 tijdseenheden aan bij V4. Verwerking V4 kan echter pas beginnen zodra alle drie de invoeren binnen zijn. Hiervan heeft GE6 de langste aflevertijd en komt pas binnen na

5 tijdseenheden. V4 kan dus pas beginnen na 5 tijdseenheden. Verwerking V4 duurt 6 tijdseenheden wat tot gevolg heeft dat GE7 na 11 tijdseenheden aankomt bij PO1.

| | V1 | V2 | V3 | V4 | V5 | V6 | TO1 | TO2 | TO3 | PO1 | PO2 | C1 | C2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **GE1** | 0 | | | | | | | | | | | | |
| **GE2** | | 0 | | | | | | | | | | | |
| **GE3** | | | 0 | | | | | | | | | | |
| **GE4** | | | | 4 | | | 3 | | | | | | |
| **GE5** | | | | 3 | | | | 2 | | | | | |
| **GE6** | | | | 5 | | 6 | | | 4 | | 5 | | |
| **GE7** | | | | | 12 | | | | | 11 | | | |
| **GE8** | | | | | | | | | | | | 14 | |
| **GE9** | | | | | | | | | | | | | 9 |

**Tabel 4.5 Tijd gereedheid gegevenseenheid**

Om de opportuniteit van de twee informatieproducten te berekenen dient er een tweede maal door het systeem gegaan te worden. De eerste maal is om de aflevertijd te berekenen. GE9 wordt bijvoorbeeld op tijdstip 9 bij de consument afgeleverd. Om de opportuniteit te berekenen moet er behalve de aflevertijd, tevens een invoertijd, leeftijd en vergankelijkheid bekend zijn. De oorspronkelijke gegevensbron van GE9 is te herleiden naar de ruwe gegevenseenheid GE3. Hieruit kan daarom de actualiteit worden berekend door de waarden in de formule in te vullen: $(9-4)+7=12$ . De opportuniteit van GE3 kan nu bepaald worden door $(1-\frac{12}{20})^2 = 0{,}16$ . Omdat er slechts één invoer is wordt 0,16 tevens de opportuniteit voor GE9. GE8 heeft echter zowel GE1, GE2, als GE3 als ruwe gegevensinvoer. De opportuniteit is hierbij afhankelijk van meerdere invoeren en wordt nu berekend via het gewogen gemiddelde. Voor het gemak zullen er hier gelijke gewichten worden toegepast. Dit kan normaliter uiteraard naar gelang de situatie worden aangepast. De opportuniteit van GE1, GE2 en GE3 worden respectievelijk berekend als $(1-\frac{16}{60})^1 = 0{,}73$ , $(1-\frac{19}{40})^2 = 0{,}28$ , $(1-\frac{25}{20})^{0,5} = 0$ . De opportuniteit van GE8 is daarom $\frac{0{,}73+0.28+0}{3} = 0{,}34$ .

## 4.8 FCOIM

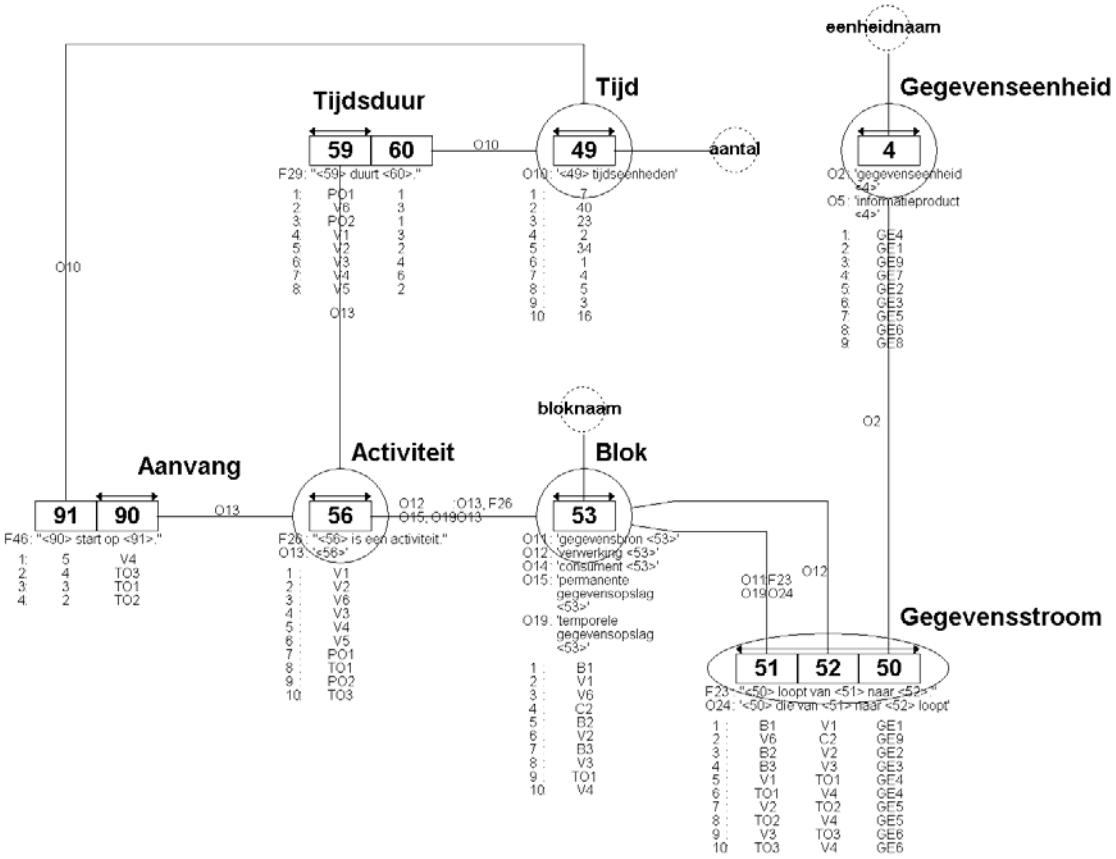
Om het informatie vervaardigingsysteem eenduidig te beschrijven is er gekozen om het met behulp van Volledig Communicatiegeoriënteerde Informatiemodellering (Engels: Fully Communication Oriented Information Modelling, afgekort FCO-IM) te modelleren [4]. FCO-IM modelleert alle conceptuele aspecten van de communicatie. Met behulp van elementaire feitexpressies is het informatie vervaardigingsysteem omschreven. Met behulp van deze feiten is vervolgens een informatiegrammaticadiagram (IGD) van het systeem vervaardigd. De mogelijk in te vullen populaties van dit IGD zijn de mogelijke structuren van elk data warehouse. Zo kan de er een populatie gemaakt worden die de structuur van de casus afgebeeld in figuur 4.5

In de volgende subparagrafen wordt het gehele IGD van het informatie vervaardigingsysteem beschreven. Het is voor de overzichtelijkheid onderverdeeld in kleine IGD's die tezamen een geheel vormen. In de diagrammen staan geen totaliteitsregels vermeld. Er kan worden aangenomen dat voor elk genominaliseerd feittype een totaliteitsregel geldt.

### 4.8.1 Blok

In figuur 4.7 staat het IGD afgebeeld van een blok in het informatie vervaardigingsysteem. Onder een blok worden alle onderdelen weergegeven in figuur 4.3 verstaan: gegevensbron, verwerking, temporele gegevensopslag, permanente gegevensopslag, consument. De blokken worden aan elkaar verbonden door middel van de gegevenseenheden. Elke gegevenseenheid heeft een blok als bron en een blok als bestemming. Dit wordt in het FCO-IM model aangeduid als een *Gegevensstroom*.

De blokken voor verwerking, temporele gegevensopslag en permanente gegevensopslag zijn bijzondere blokken. Deze blokken hebben elk een tijdsduur nodig om hun betreffende activiteit uit te voeren. In het model worden ze daarom aangeduid als een *Activiteit*, waarbij iedere activiteit een blok is. Behalve een tijdsduur heeft een activiteit een tijd waarop wordt gestart met het uitvoeren van de activiteit. Voor elke activiteit behalve de verwerking is dit de tijd waarop de gegevenseenheid arriveert. Bij een verwerking met meerdere gegevenseenheden als invoer start de verwerking pas zodra alle invoeren gearriveerd zijn. Logischerwijs start een verwerking met één invoer meteen zodra deze is gearriveerd.



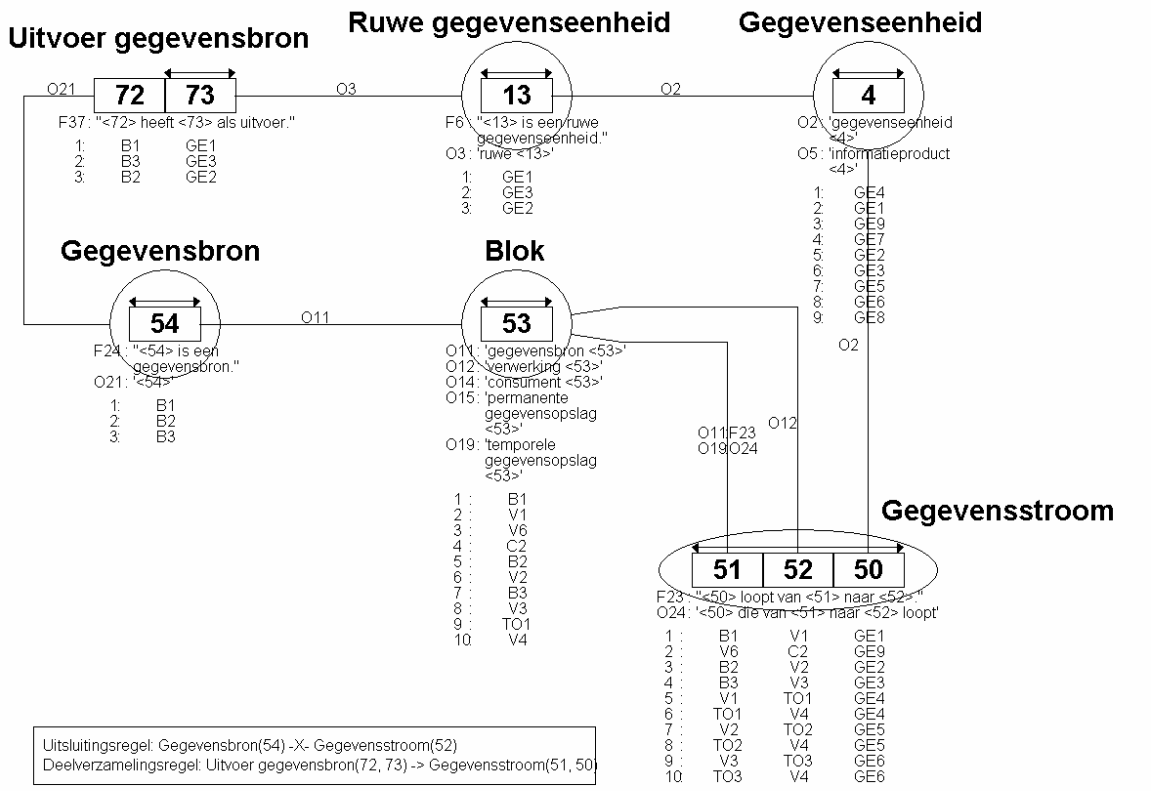

**Figuur 4.7 Blok**

## 4.8.2 Gegevensbron

Het informatie vervaardigingsysteem start altijd met een gegevensbron. Dit betekent dat een gegevensbron nooit een invoer heeft, maar alleen maar één of meerdere uitvoeren. In figuur 4.8 staat het IGD van een gegevensbron afgebeeld. De uitsluitingsregel die hierin is opgenomen zorgt ervoor dat het niet mogelijk is een gegevenseenheid op te nemen die een gegevensbron als bestemming heeft. De deelverzamelingsregel geeft aan dat elke uitvoer van een gegevensbron tevens de invoer is van een ander blok. Uit het IGD is tevens af te leiden dat een gegevensbron enkel een ruwe gegevenseenheid als uitvoer kan hebben.

De uniciteitsregel op rol 73 van het feittype *Uitvoer gegevensbron* zorgt ervoor dat elke gegevensbron één of meerdere unieke ruwe gegevenseenheden als uitvoer heeft. Er blijkt hieruit tevens dat één bron meerdere unieke gegevenseenheden als uitvoer kan hebben.

In eerste instantie lijkt het wellicht onnodig om de uitvoer van een gegevensbron apart te benoemen aangezien elk blok een uitvoer heeft. Behalve dat de gegevensbron in tegenstelling tot de andere blokken enkel een invoer heeft, blijkt tevens dat elk blok andere eisen stelt aan de invoer en uitvoer. Deze eisen kunnen worden afgedwongen met behulp van beperkingsregels zoals uniciteitsregels, uitsluitingsregels en deelverzamelingsregels. Om andere beperkingsregels toe te kunnen passen op verschillende blokken is het noodzakelijk de invoer en uitvoer van elke blok apart te modelleren.



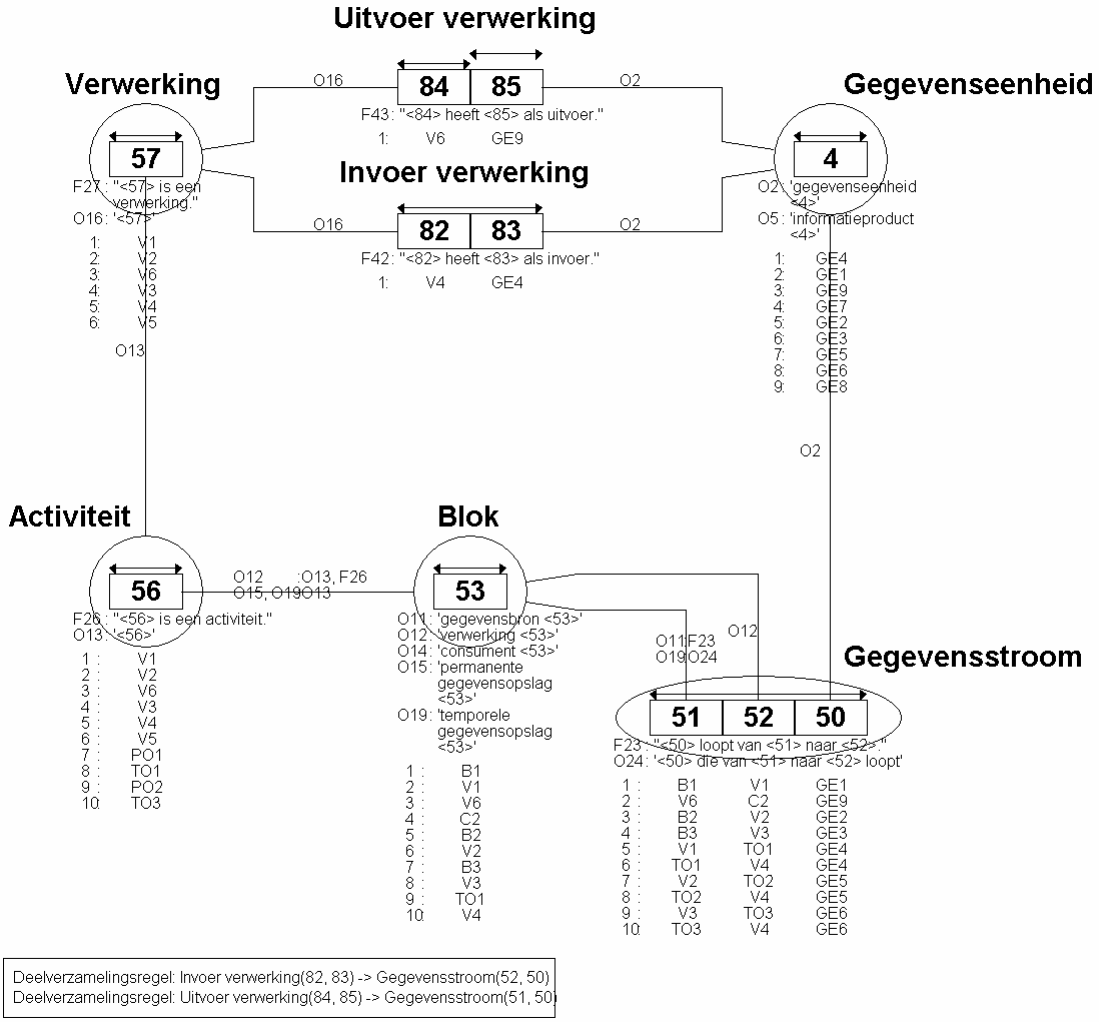

**Figuur 4.8 Gegevensbron**

## 4.8.3 Verwerking

Uit het IGD afgebeeld in figuur 4.9 is af te leiden dat, zoals reeds eerder vermeld, een verwerking een activiteit is. Verder is te zien dat een verwerking zowel een uitvoer als een invoer heeft. Echter kan de verwerking één of meerdere invoeren hebben, maar slechts één uitvoer. Door de uniciteitsregel over beide rollen van het feittype *Invoer verwerking* worden meerdere invoeren toegestaan. Door de twee uniciteitsregels op de enkele rollen in het feittype *Uitvoer verwerking*, wordt afge-

dwongen dat er geen twee verwerkingen kunnen zijn met dezelfde gegevenseenheid als uitvoer en dat een verwerking slecht één gegevenseenheid als uitvoer kan hebben. Een verwerking kan echter wel twee identieke gegevenseenheden als uitvoer hebben. Hiermee wordt gedoeld op het feittype *Gegevensstroom*. Een gegevenseenheid kan een verwerking als bron hebben, maar twee andere blokken als bestemming. Dit betekent dat er twee pijlen met dezelfde gegevenseenheid vanuit de verwerking naar verschillende blokken gaan.

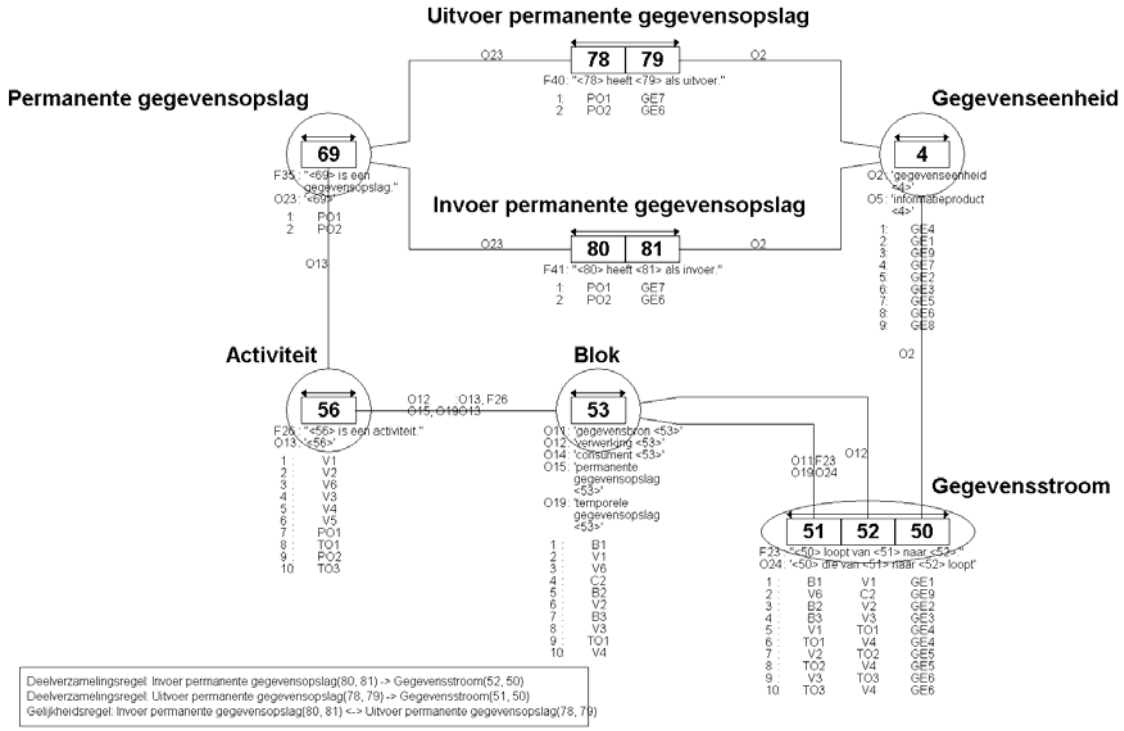
De deelverzamelingsregels betekenen respectievelijk dat de invoer van een verwerking de bestemming van een gegevenseenheid is en dat de uitvoer van een verwerking de bron van een gegevenseenheid is.




**Figuur 4.9 Verwerking**

### 4.8.4 Permanente gegevensopslag

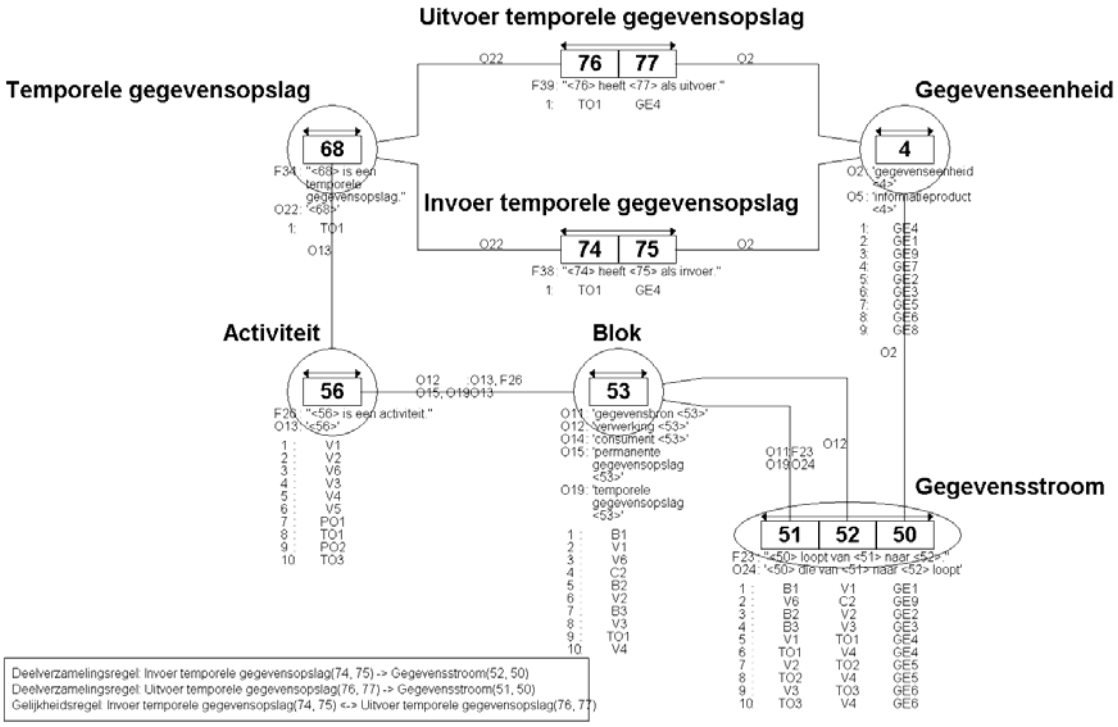
In figuur 4.10 staat het IGD van een permanente gegevensopslag afgebeeld. Hierbij geldt dat een permanente gegevensopslag veel gelijkheden heeft met een verwerking. Zo is het net zoals een verwerking een activiteit en heeft het een invoer en uitvoer. In tegenstelling de verwerking is bij de permanente gegevensopslag de invoer gelijk aan de uitvoer. Dit betekent wanneer een gegevenseenheid wordt opgeslagen (invoer) dat dezelfde gegevenseenheid weer wordt opgevraagd (uitvoer). Dit wordt afgedwongen door de gelijkheidsregel.




**Figuur 4.10 Permanente gegevensopslag**

### 4.8.5 Temporele gegevensopslag

De temporele gegevensopslag verschilt alleen wat betreft functionaliteit van de permanente gegevensopslag. De gegevens uit de temporele gegevensopslag verdwijnen zodra deze worden opgevraagd, terwijl de gegevens in de permanente gegevensopslag niet vluchtig zijn en permanent blijven bestaan (zie paragraaf 4.4). Het IGD van de temporele gegevensopslag staat in figuur 4.11 afgebeeld.




**Figuur 4.11 Temporele gegevensopslag**

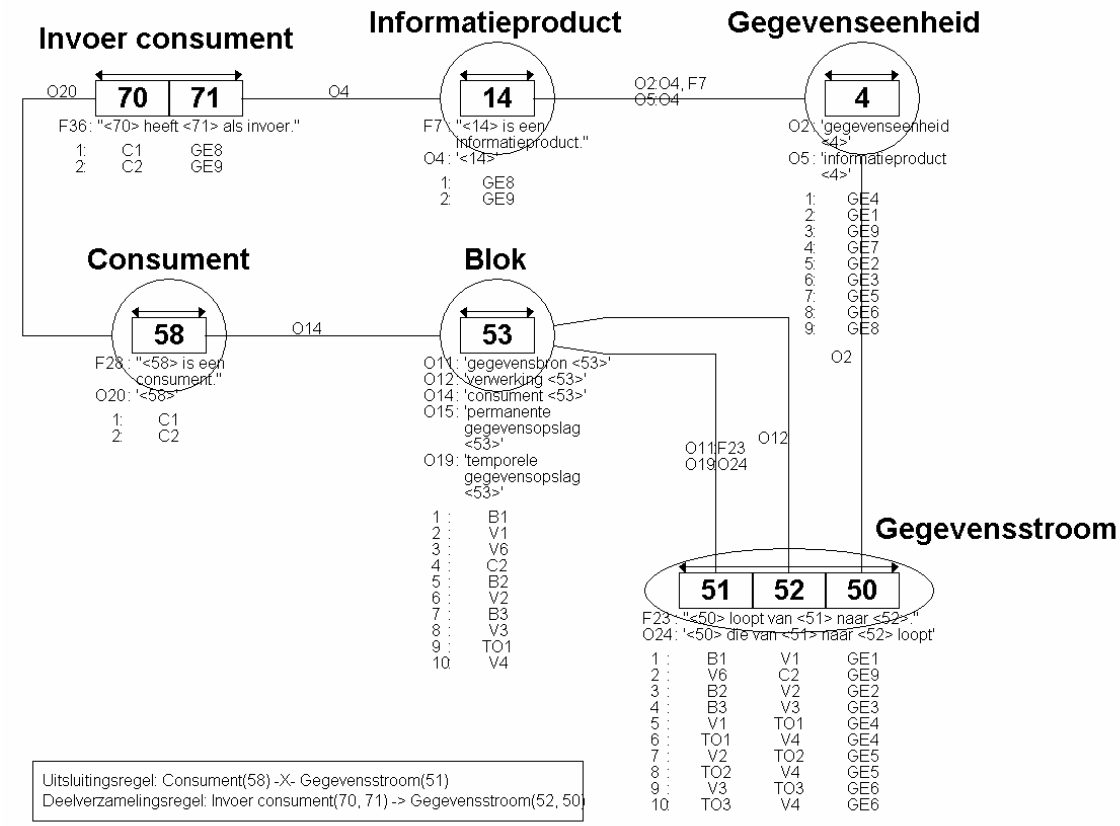
## 4.8.6 Consument

In figuur 4.12 is het IGD van de consument weergegeven. De consument is altijd het einde van het informatie vervaardigingsysteem. Dit leidt ertoe dat een consument nooit een uitvoer heeft, maar enkel één of meerdere invoeren. Door de uitsluitingsregel wordt afgedwongen dat het voor een consument onmogelijk is een uitvoer te hebben.

Uit het IGD is duidelijk af te leiden dat een consument altijd een *Informatieproduct* als invoer heeft. Tevens is te zien dat een *Informatieproduct* een soort *Gegevenseenheid* is.

De uniciteitsregel over de rollen van het feittype *Invoer consument* zorgt ervoor dat een consument slechts eenmaal hetzelfde informatieproduct als invoer kan hebben. Het is echter wel mogelijk dat één consument meerdere informatieproducten als invoer heeft en dat verschillende consumenten hetzelfde informatieproduct als invoer hebben.

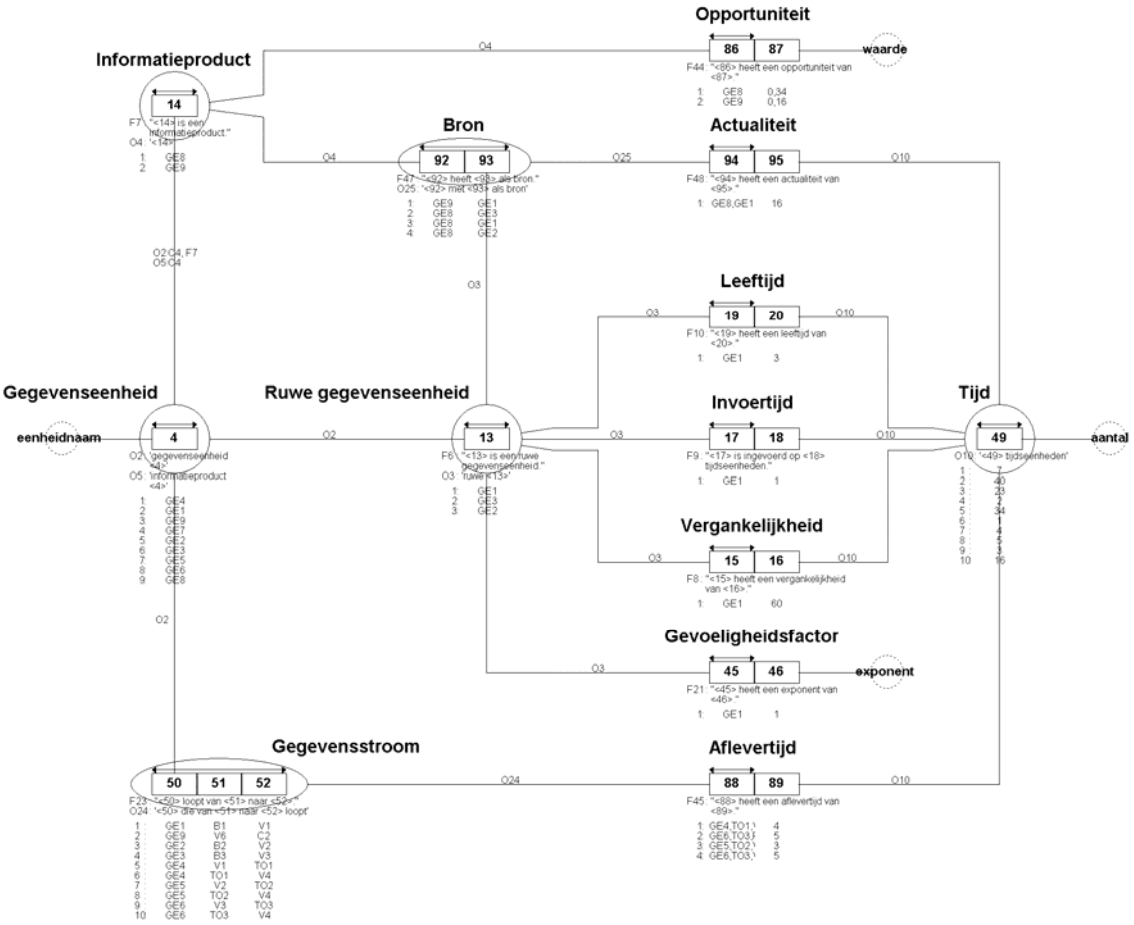
De deelverzamelingsregel benadrukt dat een consument altijd een gegevenseenheid als invoer heeft die afkomstig is van een ander blok.




**Figuur 4.12 Consument**

## 4.8.7 Opportuniteit

In het IGD in figuur 4.13 zijn alle variabelen die nodig zijn om de opportuniteit te berekenen opgenomen. Hierbij is duidelijk afgebeeld dat de opportuniteit berekend wordt van een informatieproduct. Verder is te zien dat de gevoeligheidsfactor, leeftijd, invoertijd en vergankelijkheid eigenschappen zijn van een ruwe gegevenseenheid. Elke gegevensstroom heeft een aflevertijd. Een gegevensstroom is een gegevenseenheid met een bron en bestemming en een informatieproduct is een gegevenseenheid. Hierdoor is van elk informatieproduct tevens een aflevertijd bekend. De actualiteit van een informatieproduct is afhankelijk van de leeftijd en invoertijd van de ruwe gegevenseenheden waaruit het informatieproduct is ontstaan. Het objecttype *Bron* geeft aan uit welke ruwe gegevenseenheden een informatieproduct is opgebouwd. Van elke bron van een informatieproduct kan daardoor een actualiteit worden vastgelegd.




**Figuur 4.13 Opportuniteit**

## 4.9 Grammatica

De FCO-IM modellen uit paragraaf 4.8 dienen als een formeel model voor het informatie vervaardigingsysteem. Dit model is een beschrijving van een basis grammatica waarmee over het systeem geredeneerd kan worden. De grammatica dient als basis voor een taal die kan worden gebruikt om uitspraken over het informatiesysteem te doen. LISA-D (Language for Information Structure and Access Descriptions) is een voorbeeld van zo'n taal [19]. Met behulp van LISA-D kunnen in het model voor het informatie vervaardigingsysteem extra regels opgenomen worden. In figuur 4.14 staan een aantal voorbeelden van de basis grammatica van het informatie vervaardigingsysteem.

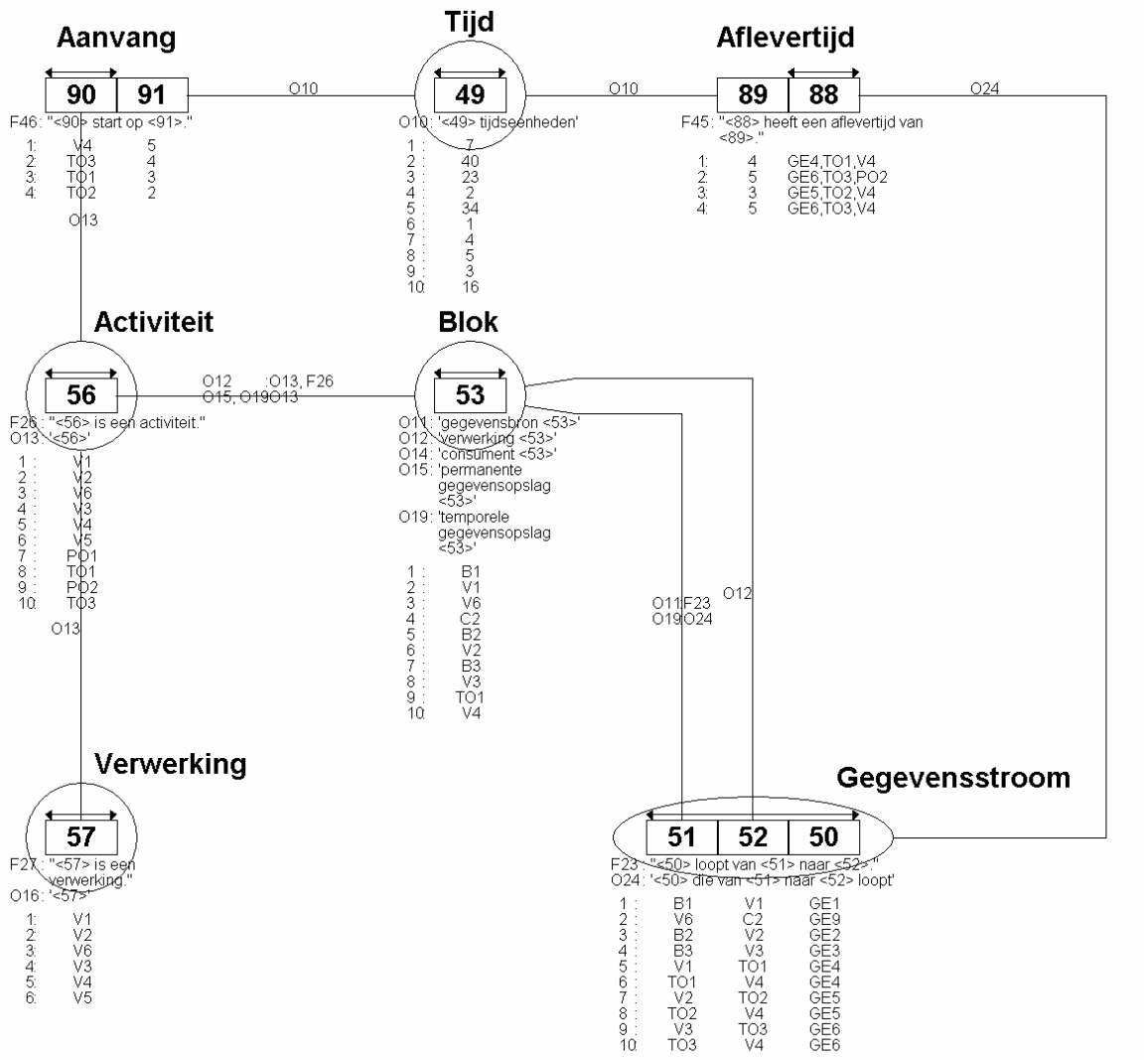
*INFORMATIEPRODUCT “GE9” wordt afgeleverd op TIJD “9”*
*VERWERKING “V4” heeft als uitvoer GEGEVENSEENHEID “GE7”*
*GEGEVENSEENHEID “GE7” is de uitvoer van VERWERKING “V4”*
*GEGEVENSBRON “B1” is een BLOK*
*TIJD “60” is de vergankelijkheid van RUWE GEGEVENSEENHEID “GE1”*

**Figuur 4.14 Basis grammatica**

Een verwerking start pas zodra alle invoeren zijn gearriveerd. Dit is echter niet op te maken uit de opgestelde FCO-IM modellen. De variabelen om dit vast te leggen zijn wel aanwezig in het systeem. Van een verwerking is vastgelegd welke invoeren deze heeft, wanneer deze invoeren arriveren en wanneer een verwerking start. Met behulp van LISA-D kan de regel dat een verwerking pas begint met zijn taak zodra alle invoeren zijn gearriveerd worden vastgelegd.

Een ander voorbeeld is de formule voor opportuniteit uit paragraaf 3.4. Er is uit het model af te leiden wat de opportuniteit van een informatieproduct is (zie figuur 4.13). Hoe deze wordt berekend ligt echter niet vast in het model. De benodigde variabelen om de opportuniteit te berekenen (*Actualiteit*, *Vergankelijkheid*, G*evoeligheidsfactor*) echter wel. In LISA-D kan dan ook worden gespecificeerd hoe de opportuniteit wordt berekend.

In dit onderzoek wordt niet verder ingegaan op de taal die kan worden ontwikkeld op basis van de opgestelde FCO-IM modellen. Om aan te tonen dat het model een correcte basis van redeneren over de werkelijkheid biedt zal er één LISA-D expressie worden uitgewerkt.

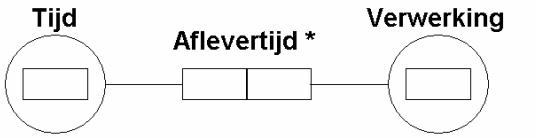



**Figuur 4.15 FCO-IM model als basis voor Lisa-D expressie**

De reeds eerder genoemde stelling dat een verwerking pas start zodra alle invoeren zijn gearriveerd zal worden aangetoond met behulp van LISA-D. In figuur 4.15 staat het FCO-IM model van een deel van het informatie vervaardigingsysteem afgebeeld waarin alle benodigde onderdelen om dit te formuleren aanwezig zijn. In het feittype *Aanvang* is vastgelegd wanneer een activiteit en dus een verwerking start. Het feittype *Aflevertijd* legt vast wanneer een gegevensstroom afgeleverd wordt bij een blok. Dit is hetzelfde als de tijd waarop een bepaalde gegevenseenheid arriveert bij een blok.

Het pad dat vanaf *Tijd* via *Aflevertijd* naar *Verwerking* loopt beschrijft de aflevertijd van een bepaalde gegevenseenheid bij een verwerking. Dit pad ontstaat door vanaf

het objecttype *Tijd* de rollen 89, 88, 52, 53 en 56 te volgen naar *Verwerking* en kan omschreven worden door de volgende LISA-D expressie: TIJD als aflevertijd van GEGEVENSSTROOM naar VERWERKING. In deze expressie wordt omschreven dat een gegevensstroom naar een verwerking loopt. Dit kan zo worden gesteld, omdat een verwerking een soort activiteit is en een activiteit een blok is. In figuur 4.16 staat een weergave van het af te leiden pad dat de LISA-D expressie beschrijft.




**Figuur 4.16 LISA-D pad**

Op dezelfde manier als de aflevertijd kan de aanvang van de verwerking met LISA-D worden beschreven: TIJD als aanvang van VERWERKING. Deze expressie omschrijft het pad van *Tijd* naar *Verwerking* via de rollen 91, 90 en 56.
Door de beschreven expressies te combineren ontstaat een LISA-D expressie die omschrijft dat een verwerking pas start zodra alle invoeren zijn gearriveerd:

*TIJD als aanvang van VERWERKING = max(TIJD als aflevertijd van GEGEVENS-STROOM naar VERWERKING)*

Het voordeel van deze expressie is dat deze zeer leesbaar is. Het lijkt sterk op natuurlijke taal. Hierdoor is een domeindeskundige die geen verstand heeft van FCO-IM of een andere methode om een domein te modelleren in staat het te lezen. Ondanks het feit dat de expressie veel op natuurlijke taal lijkt staat er eenduidig dat een verwerking pas start zodra alle invoeren zijn gearriveerd.

# 5 Conclusies en aanbevelingen

## 5.1 Inleiding

De onderzoeksvraag voor dit onderzoek luidt:

*"Kan de kwaliteit van een data warehouse door middel van een raamwerk van de gegevensverwerking inzichtelijk gemaakt worden voor een gebruiker?"*

In de voorgaande hoofdstukken is er onderzocht in hoeverre deze onderzoeksvraag beantwoord kan worden. Dit hoofdstuk zal een conclusie uit de voorgaande hoofdstukken trekken. Verder zullen er nog een aantal aanbevelingen worden gedaan voor toekomstig onderzoek op dit gebied.

## 5.2 Conclusies

Een data warehouse beschrijft een complex afhankelijkheidsnetwerk van gegevensbronnen en processen dat ondersteuning biedt bij het maken van analyses en beter gefundeerde beslissingen. In een data warehouse zijn een viertal lagen te onderscheiden:

1. Operationele bronsystemen
2. Gegevens verzamelen
3. Gegevens presenteren
4. Toegangsapplicaties

De eerste drie lagen hebben betrekking op het synchronisatieproces. Dit is een belangrijk proces dat de bruikbaarheid van de gegevens in het data warehouse bepaald. De gegevens worden gesynchroniseerd via een vast patroon. Dit start bij de gegevensbronnen en vanaf hier worden de gegevens via een reeks van processen en bronnen voor gegevensopslag gepropageerd naar de data marts.

De laatste laag van een data warehouse betreft de toegangsapplicaties die de informatie die opgeslagen is in de data marts als informatieproduct aanbieden aan de consument (gebruiker). Een informatieproduct is een gegeven dat voor de gebruiker waarde heeft, bijvoorbeeld een rapport of grafiek. De toegangsapplicaties voeren een

query uit om de informatie aan de consument te tonen. Deze query wordt tevens beschouwd als een proces dat gegevens verwerkt.

Het informatie vervaardigingsysteem is in staat het gehele data warehouse in kaart te brengen van gegevensbron tot aan de consument. Dit model dient als raamwerk om de architectuur van elk data warehouse in kaart te brengen.

De kwaliteit in een data warehouse wordt beïnvloed door meerdere factoren. In dit onderzoek is enkel de kwaliteit van de gegevens die in het data warehouse worden geladen in acht genomen. Hierbij is gegevenskwaliteit gedefinieerd als gegevens die geschikt zijn voor gebruik door gegevensconsumenten.

Het blijkt dat gegevenskwaliteit te onderscheiden is in meerdere dimensies. Hiervan is de dimensie opportuniteit gekozen om verder uit te diepen, omdat deze goed aansluit bij de bewering dat een gebruiker behoefte heeft aan bijgewerkte en betrouwbare informatie.

Opportuniteit staat voor geschiktheid van tijd of gelegenheid. Of, anders gezegd, deze dimensie bepaalt of een gegeven op een geschikt moment aan de gebruiker wordt getoond. Het blijkt dat opportuniteit een verhouding is tussen de actualiteit en de vergankelijkheid. Hierbij heeft actualiteit betrekking op de leeftijd van het gegeven dat wordt gebruikt om een informatieproduct te vervaardigen. De vergankelijkheid heeft betrekking op de geldigheidsduur van zo'n gegeven. Doordat de aflevertijd van een informatieproduct onderdeel is van de actualiteit blijft de opportuniteit onbekend tot de aflevering van een informatieproduct.

De opportuniteit is een waarde van 0 tot en met 1, waarbij 0 een slechte opportuniteit aanduidt en 1 een goede. Doordat de waarde altijd op een vaste schaal wordt aangeduid is de gebruiker in staat verschillende waarden met elkaar te vergelijken en kan er een inschatting worden gemaakt van hoe goed of slecht deze is.

Het model van het informatie vervaardigingsysteem kan bijdragen aan een verbetering van de opportuniteit. Doordat de architectuur van de gegevensverwerking van het data warehouse in kaart gebracht is, kan worden aangetoond waardoor de opportuniteitswaarde wordt beïnvloed. Hierdoor kunnen aanpassingen worden gedaan aan het model en dus aan het data warehouse om zo een betere opportuniteit voor de gebruiker te realiseren.

Om het informatie vervaardigingsysteem eenduidig te beschrijven is er gekozen om het met behulp van FCO-IM te modelleren. FCO-IM modelleert alle conceptuele aspecten van de communicatie. De populatie van het opgestelde IGD is de architectuur van een data warehouse. Op deze manier kan elk data warehouse in het IGD beschreven worden.

Het FCO-IM model is een beschrijving van een basis grammatica waarmee over het systeem geredeneerd kan worden. De grammatica dient als basis voor een taal die kan worden gebruikt om uitspraken over het informatiesysteem te doen. Hiermee kunnen in het model voor het informatie vervaardigingsysteem extra regels opgenomen worden.

## 5.3 Aanbevelingen

Behalve opportuniteit zijn er vele andere kwaliteitsdimensies. In het huidige model ontbreekt het blok dat betrekking heeft op kwaliteit. Hoewel dit is af te vangen door een verwerking als kwaliteitsverbeteraar te modelleren kan er een nieuw blok worden opgenomen dat dient voor het verbeteren van de kwaliteit. In het huidige informatie vervaardigingsysteem is dit blok achterwege gelaten.

Ballou et. al hebben reeds gebruik gemaakt van een blok voor verbetering van kwaliteit [8]. Zij geven aan dat de meeste kwaliteitsdimensies hiermee gemodelleerd kunnen worden.

Er dient nog een uitbreiding gemaakt te worden op de taal die is gebaseerd op de grammatica die in kaart is gebracht met behulp van FCO-IM. Hierdoor kan er eenduidig over opportuniteit en het informatie vervaardigingsysteem worden geredeneerd. Bovendien kunnen er verbanden die niet in het model zijn opgenomen worden vastgelegd.

De opportuniteit is in dit onderzoek enkel theoretisch uitgewerkt. Een test in de praktijk met een data warehouse en de gebruikers dient uitgevoerd te worden om te valideren of het werkelijk een meerwaarde is voor de gebruiker. Er kan bijvoorbeeld gedacht worden aan een meter die aangeeft wat de opportuniteit van de informatie is. Zo kan er simpelweg een getal getoond worden, maar het is tevens mogelijk een kleurgradatie (bijvoorbeeld rood is slecht en groen is goed) te tonen. Op deze manier kan de gebruiker aan de kleur zien wat de opportuniteit is.

# 6 Literatuur


[1] Agrawal, D., El Abbadi, A., Singh, A., Yurek, T., “Efficient View Maintenance at Data Warehouses”, ACM, 1997.

[2] Araque, F., “Real-time Data Warehousing with temporal requirements”, Decision Systems Engineering Workshop (DSE'03), 2003.

[3] Aristoteles, Metafysica.

[4] Bakema, G., Zwart, J.P., van der Lek, H., “Volledig Communicatiegeoriënteerde Informatiemodellering (FCOIM)”, ten Hagen & Stam, 2000.

[5] Ballou, D.P., Pazer, H.L., “Designing Information Systems to Optimize the Accuracy-Timeliness Tradeoff”, Information Systems Research, vol. 6, no. 1, pp. 51-72, 1995.

[6] Ballou, D.P., Pazer, H.L., “Modeling Completeness versus Consistency Tradeoffs in Information Decision Contexts”, IEEE Transactions on Knowledge and Data Engineering, vol. 15, no. 1, January/February 2003.

[7] Ballou, D.P., Pazer, H.L., “Modeling Data and Process Quality in Multi-Input, Multi-Output Information Systems”, Management Science, Vol. 31, No.2 (Feb., 1985), 150-162.

[8] Ballou, D.P., Wang, R., Pazer, H., Kumar.Tayi, G, “Modeling Information Manufacturing Systems to Determine Information Product Quality”, Management Science, Vol. 44, No. 4, April 1998.

[9] Barnes Nelson, G., “Real Time Decision Support: Creating a Flexible Architecture for Real Time Analytics”, SUGI 29, Data Warehousing, Management and Quality, 2004.

[10] Bouzeghoub, M. Fabret, F. Galhardas, H. Matulovic-Broqué, M., Pereira, J. en Simon, E., “Fundamentals of Data Warehouses”, hoofdstuk 4, “Data Warehouse Refreshment”, Springer, 2000.

[11] Bruckner, R.M., List, B., Schiefer, J., “Striving towards Near Real-Time Data Integration for Data Warehouses”, DaWaK 2002, LNCS 2454, 2002.

[12] Bruckner, R.M., Tjoa, A.M., “Managing Time Consistency for Active Data Warehouse Environments”, DaWaK 2001, LNCS 2114, 2001.

[13] Codd, E.F., “A Relational Model of Data for Large Shared Data Banks Communications of the ACM”, Vol. 13, No. 6, June 1970.

[14] Crosby, P.B., “Quality is Free”, Signet (Re-issue edition), 1980.

[15] Davis, G.D., “Management Information Systems: Conceptual Foundations, Structure and Development”, McGraw Hill, New York, 1974.

[16] Dey, D., Zhongju, Z., Prabuddha, D., “Optimal Synchronization Policies for Data Warehouses”, November 12, 2002.

[17] Engström, H., Chakravarthy, S., Lings, B., “A Holistic Approach to the Evaluation of Data Warehouse Maintenance Policies”, Technical Report, HS-IDA-TR-00-001, 2000.

[18] Han, Q. Venkatasubramanian, N., “Addressing Timeliness/Accuracy/Cost Tradeoffs in information Collection for Dynamic Environments”, IEEE International Real-Time Systems Symposium (RTSS’03), 2003.

[19] Hofstede, A.H.M. ter, Proper, H.A., Weide, Th.P. van der, “Formal definition of a conceptual language for the description and manipulation of information models”, Information Systems, 18(7):489-523, October 1993.

[20] Inmon, W. H., “Building The Data Warehouse”, Third Edition, Wiley Computer Publishing, 2002.

[21] Jarke, M., Jeusfeld, M.A., Quix, C. Sellis, T., Vassiliadis, P., “Fundamentals of Data Warehouses”, hoofdstuk 7, “Data Warehouse Refreshment”, Springer, 2000.

[22] Juran, J.M., Godfrey, A.B., “Juran's Quality Handbook”, Fifth Edition, McGraw-Hill, 1998.

[23] K.A., Delic, Douillet, L., Dayal, U., “Towards an Architecture for Real-Time Decision Support Systems: Chalanges and Solutions”, IDEAS’01, IEEE, 2001.

[24] Kimball, R. en Ross, M., “The Data Warehouse Toolkit”, Second Edition, Wiley Computer Publishing, 2002.

[25] Lek, van der H., Habers, F. en Schmitz, M., “Sterren en Dimensies”, “Ontwerp en onderhoud van datawarehouses”, Array Publications b.v., 2000.

[26] Quix, C., Jarke, M., “Fundamentals of Data Warehouses”, hoofdstuk 1, “Data Warehouse Practice: An Overview”, Springer, 2000.

[27] Reeves, C.A., Bednar, D.A., “Defining Quality: Alternatives and Implications”, The Acadamy of Management Review vol 19, No. 3, Special Issue: Total Quality (Jul., 1994), pp 419-445.

[28] Segev, A., Fang, W., “Optimal Update Policies For Distributed Materialized Views, Management Sciences”, Vol. 37, No. 7, July 1991.

[29] Shankaranarayan, G., Ziad, M., Wang, R.Y., “Managing Data Quality in Dynamic Decision Environments: An Information Product Approach”, Journal of Database Management, 14(4), 14-32, Oct-Dec, 2003.

[30] Snijders, R., Reijswoud, van V., “Realtime data warehousing is een doodlopend spoor”, “Hoedt u voor het telefoonstrijkijzer!”, Database Magazine, Nummer 5, september 2000.

[31] Wand, Y., Wang, R.Y., “Anchoring Data Quality Dimensions in Ontological Foundations”, Communications of the ACM, November 1996/Vol.39, No. 11.

[32] Wang, R.Y., Strong, D.M., “Beyond Accuracy: What Data Quality Means to Data Consumers, Journal of Management Information Systems”, Spring 1996;12, 4, ABI/INFORM Global pg. 5.

[33] Zhang, X., Ding, L., Rundensteiner, E.A., “PVM: Parallel View Maintenance under Concurrent Data Updates of Distributed Sources”, DaWaK 2001, LNCS 2114, pp. 230-239, 2001.