Totaalconcept is adequate ontwikkelmethode

Xtreme Data Warehousing

Martin Misseyer en René Nobel

Er is veel gepubliceerd over de implementatie van Business Intelligence- en datawarehouse-omgevingen. Enerzijds omdat elke implementatie op zichzelf uniek is te noemen, anderzijds omdat er zoveel zaken mis kunnen gaan of heel anders lopen dan ze waren gepland. Op basis van ervaring op gebied van Business Intelligence hebben de auteurs aan een totaalconcept mee ontwikkeld, genaamd Xtreme Data Warehousing.

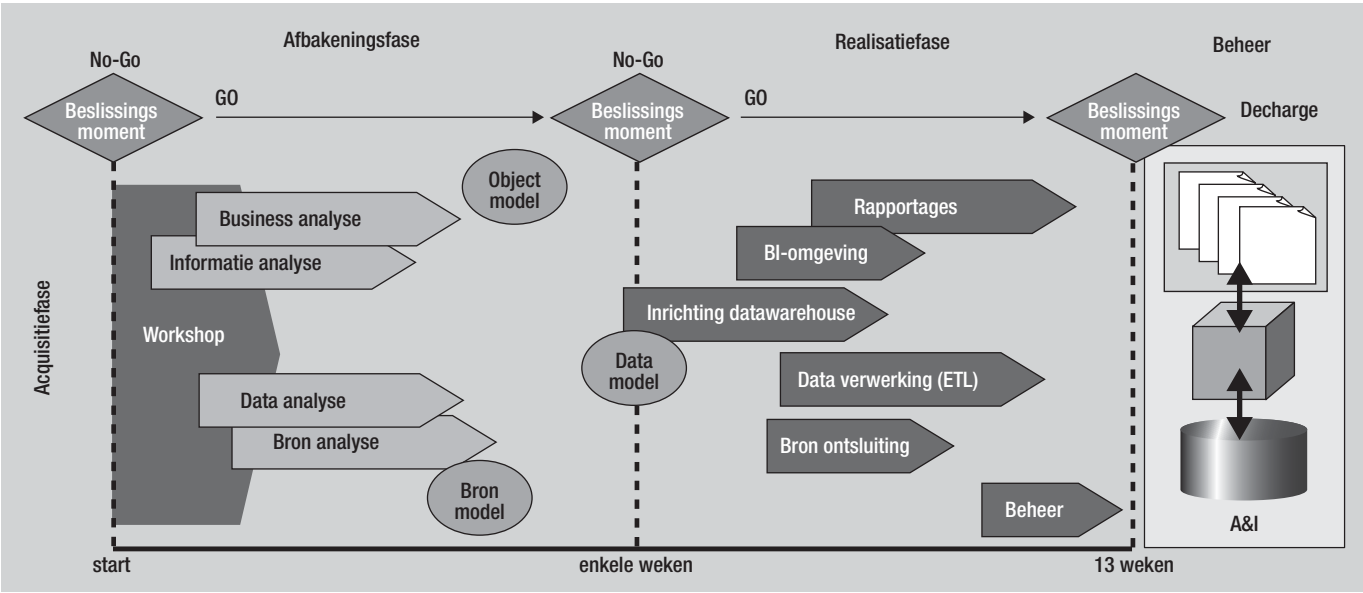
Er zijn legio hobbels, valkuilen, tips en trucs te melden aangaande de implementatie van Business Intelligence- en datawarehouse-omgevingen en verschillende analisten hebben gepubliceerd over de lage slagingskans van dit soort projecten. Belangrijke onderwerpen die steeds terugkeren bij zo'n implementatie betreffen onder andere de te hanteren aanpak, de toe te passen methoden en technieken en, niet in de laatste plaats, de keuze voor een bepaalde architectuur en de te selecteren infrastructuur.

Totaalconcept

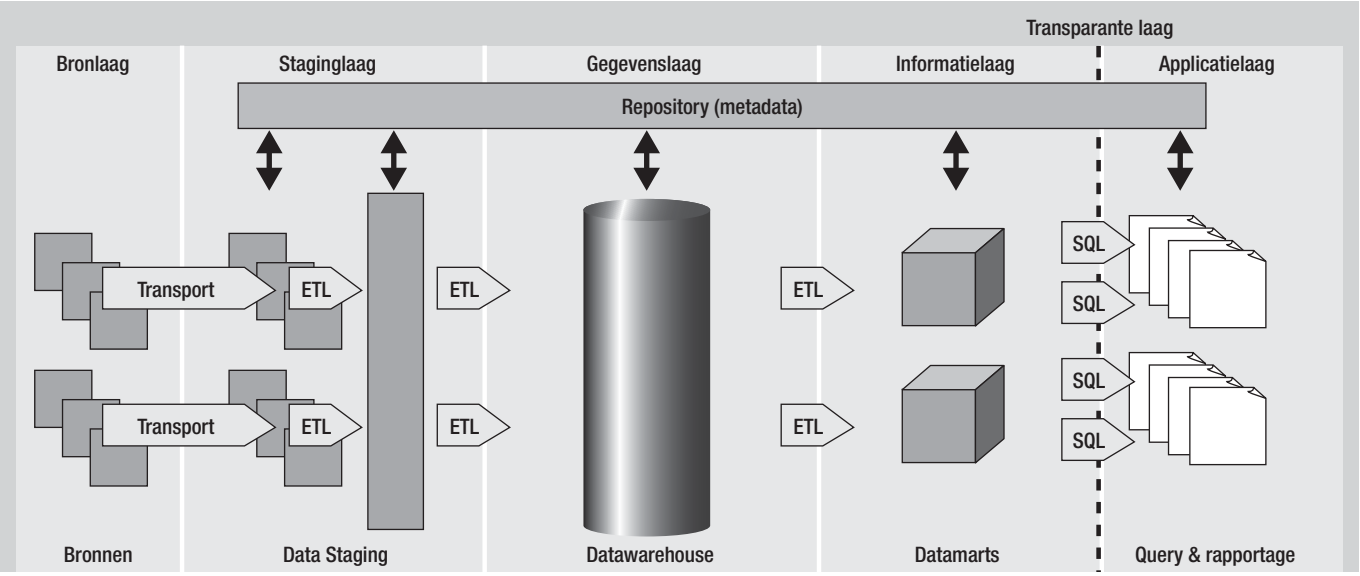
Xtreme Data Warehousing is een totaalconcept waarin 'best practices' op het gebied van aanpak, architectuur, infrastructuur en ontwikkelmethode worden samengebracht. In dit artikel staan enkele facetten van Xtreme Data Warehousing centraal.

In de BI-markt bestaat een toenemende vraag van managers en (eind)gebruikers naar kortere, goedkoper, laagdrempeliger ontwikkeltrajecten. Immers, de ervaring van de meeste managers en (eind)gebruikers is het tegenovergestelde: BI- en datawarehouse-projecten worden vaak ervaren als 'groots', duur, lang en technisch-georiënteerd, dat wil zeggen dat eerst aandacht wordt besteed aan informatietechnologie, de implementatie zelf, en daarna pas aan de gebruiker ervan.

Naar aanleiding hiervan is uitgebreid (literatuur)onderzoek gedaan naar succesvolle en minder sucesvolle BI- en DWH-projecten en de hierin gehanteerde methoden (van PRINCE2 tot en met DSDM, van RAD tot en met Kimball en Inmon) en een scala aan technieken (informatie-analyse, datamodellering, etcetera). Op basis van het onderzoek is vastgesteld dat een viertal punten



Afbeelding 1: Xtreme Data Warehousing aanpak.



Afbeelding 2: Xtreme Data Warehousing architectuur.

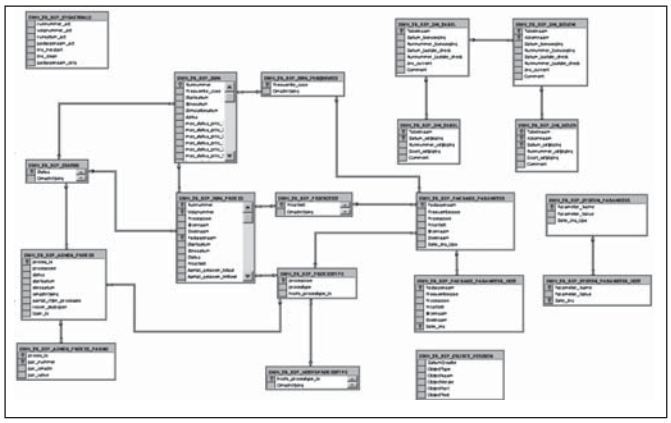
doorslaggevend is voor het al dan niet succesvol zijn van een project. Ten eerste is het essentieel om in korte tijd (een deel van) de gewenste informatievoorziening succesvol te implementeren, op basis van de eisen en wensen, die de opdrachtgever/gebruiker heeft geformuleerd. De gebruiker die (door zijn eigen organisatie) niet goed wordt bediend, gaat immers zelf knutselen.
Ten tweede, veel BI- en datawarehouse-projecten kennen een behoorlijke overhead. In de praktijk gaat veel tijd verloren aan oeverloze discussies over wat waar (beslissingen) te implementeren en vervolgens waarmee (selecties). Hoe minder discussie, des te beter voor de klant én het project.
Ten derde, de kosten van een BI- en datawarehouse-project lopen de eerste maanden snel op, omdat er veel tijd wordt besteed aan haalbaarheidsonderzoek en selectietrajecten voor architectuur en infrastructuur. Hier staan nauwelijks baten tegenover. Hoe korter deze periode, des te beter; want hoe langer deze periode duurt, des te zenuwachtiger de opdrachtgever wordt.

En ten vierde is het belangrijk om in een beginfase de technologie geen *showstopper* te laten worden. Anders gezegd, het is logisch om eenvoudige technologie en infrastructuur te gebruiken, die wijd verbreid wordt toegepast en die voldoende schaalbaar is.

Xtreme Data Warehousing

Omdat de hierboven beschreven onderwerpen onvoldoende worden geadresseerd in de eerder genoemde ontwikkelmethoden, is gezocht naar een adequaat alternatief. 'Agile' development, met name Xtreme Programming (XP), was zeer aantrekkelijk vanwege de team-gebaseerde ontwikkeling, de nadruk op communicatie, de oplevering van kleine stukjes, het uitvoerig testen (met gebruikers), en de nadruk op eenvoud. Vervolgens ontstond het idee om deze 'extreem zachte' eigenschappen te combineren met 'extreem harde' eigenschappen, in de vorm van een compleet uitgewerkte ontwikkelstraat met, op basis van best practices, uitgesproken uitgangspunten (bijvoorbeeld wijze van historie vastlegging en modellering) en principes (bijvoorbeeld procesgeneratie). Hiermee zouden alle hiervoor genoemde punten kunnen worden geadresseerd. Aldus ontstond Xtreme Data Warehousing.
Xtreme Data Warehousing, kortweg xDWH, is als totaalconcept een mix van (project)aanpak, vooraf geselecteerde methoden en technieken, een standaardarchitectuur en een eenvoudige infrastructuur. xDWH is gebaseerd op het 'time boxing' mechanisme, zoals ook DSDM dat kent. De xDWH time box is maximaal 13 weken. Onderdeel van de aanpak is tevens dat de *deliverables*, die moeten worden opgeleverd, worden gemanaged volgens het MoSCoW-principe.
Vanaf de eerste dag ligt de focus op ontwerp en implementatie. xDWH maakt gebruik van een best practice die voorziet in een standaard, vooraf bepaalde, architectuur en infrastructuur.
In xDWH zijn voor elke analyse-, modellering- en implementatie-activiteit en voor elke architectuur- en infrastructuurcomponent, standaarden gedefinieerd. Voorbeelden hiervan zijn de uitwerking van standaarden voor onder meer bronaanlevering, staging, datawarehouse- en datamart-objecten, historie-modellering, foutafhandeling, en ETL-processen. Deze standaarden behelzen niet alleen de naamgeving, maar tevens richtlijnen voor het programmeren, besturen, en metadata. Een belangrijk onderdeel in xDWH is tevens weggelegd voor generatie van objecten en processen.

Een xDWH project-team bestaat uit drie of vier ervaren all-round consultants. Dit team wordt aangevuld met enkele business- en bron-experts. Het is essentieel dat elk lid van het team beschikt over twee of meer specialismen. Dit is essentieel om tegenvallers op te kunnen vangen. Immers, zowel doorlooptijd als capaciteit zijn gefixeerd, zodat dus niet zomaar nieuwe resources zijn aan te boren. De xDWH aanpak is zo ontwikkeld dat de manager (probleemeigenaar) de opdrachtgever is voor een xDWH traject. In een xDWH project wordt direct met de business zaken gedaan.



Afbeelding 3: Deel van de xDWH metadata repository.

Aangezien tijd, doorlooptijd en capaciteit, in principe gefixeerd zijn, is het mogelijk om een xDWH project *fixed price* aan te bieden. Vanzelfsprekend kan dit alleen als aan een aantal essentiële randvoorwaarden wordt voldaan. Want in reguliere BI- en datawarehouse-projecten loopt men een groot risico, wanneer deze niet aanwezig zijn.

Aanpak

Een xDWH project bestaat uit een afbakenings- en een realisatiefase (zie afbeelding 1). De workshop wordt gehouden met het management en eindgebruikers (de 'business') en is bedoeld om vast te stellen welke informatiebehoefte (met name in de vorm van rapportages) wordt onderkend en welke prioriteit hieraan wordt verbonden.
In de afbakeningsfase wordt in detail onderzocht wat de gewenste rapportages behelzen (rapportage-analyse), uit welke data-elementen de gewenste rapportages zijn opgebouwd (informatie-analyse), welke bron-systemen de betreffende data-elementen bevatten, wat hun kwaliteit is (bronanalyse), om tenslotte een rudimentaire schets te kunnen maken van de op te leveren datawarehouse- en datamart-modellen (data-analyse). Het doel van de afbakeningsfase is om met voldoende mate van zekerheid een beslissing te kunnen nemen over het vervolg: 'go' of 'no-go'.
In de analysefase wordt voor de verschillende typen analyses gebruik gemaakt van standaard (Excel) templates waarin informatiebehoefte, rapportages, dimensies, feiten en meetwaarden gestandaardiseerd worden vastgelegd. Deze 'deliverables' vormen de documentatie op basis waarvan de oplossing wordt ontwikkeld. xDWH kent derhalve een andere (lees: extreme) documentatievorm, in vergelijking met een methode waarin standaard functionele en technische ontwerpen worden gemaakt.
De analysefase wordt in dit artikel niet verder behandeld.

De realisatiefase behelst het werkelijk implementeren van de rapportages en de bijbehorende standaardarchitectuur met behulp van de standaard infrastructuur. Allereerst worden de datamodellen gedetailleerd en geïmplementeerd in datawarehouse en datamart(s). Hiervoor moet infrastructuur, hard- en software en database worden geïnstalleerd (inrichten datawarehouse).

Op basis van de noodzakelijke data dienen afspraken te worden gemaakt met de broneigenaar voor de levering van extractiebestanden (bronontsluiting). Dataverwerkings-processen dienen te worden ontwikkeld om de bronextracties in het datawarehouse te verwerken (dataverwerking, ETL). Voor de informatievoorziening moet een BI-omgeving worden ingericht met behulp van de standaardinfrastructuur (BI-omgeving). Tenslotte kunnen de overeengekomen rapportages ook worden ontworpen en geïmplementeerd. Vanzelfsprekend wordt hierbij de input uit de afbakeningsfase als uitgangspunt genomen.
In de realisatiefase wordt ontwikkeld op basis van de specificaties uit de afbakeningsfase. Naast de eerder genoemde ingevulde templates (Excel) is de belangrijkste input voor de realisatiefase het xDWH handboek waarin is beschreven welke technische standaarden dienen te worden gevolgd (onder meer welke objecten, naamgevingsconventies). De 'deliverables' van de realisatiefase zijn de fysiek ontwikkelde objecten, datamodellen, databases, rapportages. Deze deliverables zijn exact volgens de specificaties gerealiseerd. In xDWH worden deze dan ook niet in aparte realisatiedocumentatie beschreven. Wel wordt een beheerhandleiding samengesteld, als 'deliverable' voor de overdracht naar beheer.

Architectuur

De xDWH architectuur bestaat uit vijf lagen; anders gezegd, de data kennen vijf fasen of toestanden. Deze architectuur is uitvoerig beproefd. Ten aanzien van de xDWH architectuur kan geen water bij de wijn worden gedaan; deze architectuur wordt functioneel gezien altijd op één en dezelfde wijze geïmplementeerd. Hoewel niet in afbeelding 2 getoond, is op een beperkt aantal plaatsen wel degelijk sprake van alternatieven. Elk van de architectuurlagen wordt beknopt besproken.
De *staging-laag* bestaat uit een ontvangstportaal (ontvangen bestanden), een staging area (verwerking bestanden) en een archiefportaal (archiveren bestanden). De staging area kent geen historie en is eenvoudig relationeel gemodelleerd. Hiermee wordt bedoeld dat geen constraints worden afgedwongen en alleen sprake is van een lokaal gedefinieerde technische sleutel. Dit wil zeggen dat de geïdentificeerde bronsleutel in de staging area niet als sleutel wordt gebruikt.
De *gegevenslaag* bestaat uit het datawarehouse. Het datawarehouse is relationeel gemodelleerd in 3NF. Indien dit is vereist, wordt historie bijgehouden op basis van een 'journalling mechanisme'. Dit wil zeggen dat de oude toestand van het gewijzige record wordt bewaard in een historietabel. Voor het vastleggen van gegevens wordt de originele bronsleutel gebruikt, die voor historische vastlegging wordt gecombineerd met één of meerdere datums. Evenzo worden alle niet verwerkbare rijen bewaard in een error-tabel. In de gegevenslaag worden domeinen ('coderingen') meegemodelleerd en fysiek gescheiden opgeslagen. Het voorgaande geeft aan dat de gegevenslaag, het datawarehouse, is ingericht voor efficiënte opslag van gegevens. Het datawarehouse is niet direct toegankelijk voor de (eind)gebruiker.

De *informatielaag* bestaat uit één of méér datamarts. Deze datamarts zijn dimensioneel gemodelleerd, dat wil zeggen dat de gegevens uit het datawarehouse zijn gemapt op de specifieke informatiebehoefte van de eindgebruiker in termen van dimensies en feiten. De informatielaag is geoptimaliseerd voor ontsluiting, dat wil zeggen geschikt en toegankelijk voor de (eind)gebruiker. Overigens is dit alleen mogelijk via een rapportagehulpmiddel.
Naast de besproken lagen, wordt in de architectuur ook een laagoverschrijdende component gedefinieerd, zijnde de metadata repository (zie afbeelding 3). Dit zou de metalaag kunnen worden genoemd, die haaks staat op de vijf andere lagen. De metadata repository omvat een verzameling objecten waarin metadata worden vastgelegd. Deze metadata omvatten enerzijds begrippen en definities van objecten, toestandgeoriënteerde metadata die de status van bepaalde objecten beschrijven, tot en met verwerkingsstatistieken, en logging van foutieve verwerking.
Het voordeel van de metadata repository is evident. Uit de xDWH metadata repository kunnen uiteenlopende beheerrapportages worden gegenereerd. Ook 'power' users kunnen hier baat bij hebben. Belangrijke rapportages zijn bijvoorbeeld: het datawarehouse datamodel-evolutierapport; de gedefinieerde processen en de draaistatistieken; de error-rapporten.

Verwerking

Afbeelding 4 geeft een schematische weergave van de xDWH architectuur gerelateerd aan de processen, de data en de toegepaste infrastructuur. Met betrekking tot de verwerking, worden de belangrijkste objecten voor de verwerking vermeld.

Van staging area naar datawarehouse.

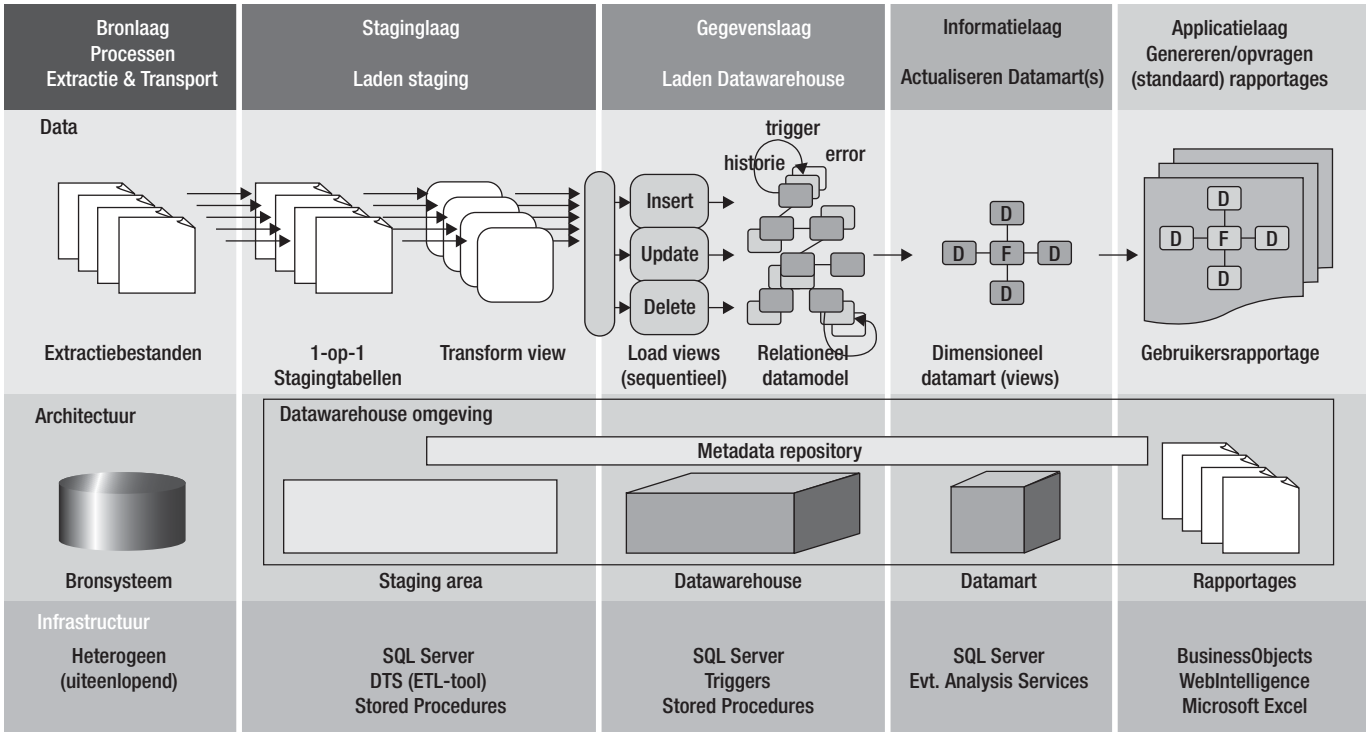
In tegenstelling tot het laden van data in de staging area, is het laden van data in het datawarehouse xDWH specifiek. Zo worden er één of meer *transformviews* gedefinieerd (zie afbeelding 5) welke data uit de staging area 'mappen' op het datawarehouse datamodel. De transformview zet de data klaar voor het datawarehouse. De tweede stap voor het verwerken van de data uit de staging area is, het bepalen op welke wijze data dienen te worden geladen in het datawarehouse. Met een loadview wordt bepaald of data moeten worden toegevoegd of worden gewijzigd (zie afbeelding 6). Input van de loadview is enerzijds de onderliggende transformview (source) en anderszijds de datawarehouse-tabel (target). Voorgaande is noodzakelijk omdat er een generiek (ETL) proces wordt gedraaid, dat volgordelijk per type laadactie de verwerking uitvoert.

Van datawarehouse naar datamart.

Een xDWH implementatie schrijft voor dat een datamart virtueel is. Dit houdt in dat de voor de gebruiker relevante datawarehouse data met behulp van views worden platgeslagen in dimensies (zie afbeelding 7) en feiten (zie afbeelding 8) views. Dit houdt in dat het laden van het datawarehouse impliceert dat virtuele datamarts automatisch worden geactualiseerd. Mochten zich, op termijn, performance-problemen voordoen, dan kan een datamart altijd nog fysiek worden gedefinieerd. Hiervoor zijn twee alternatieven beschikbaar, te weten een fysiek gemodelleerde datamart met behulp van 1. het toegepaste DBMS of 2. een specifieke kubustool (zie afbeelding 4 Analysis Services).

Hoofdverwerkingsproces.

Voor de (dagelijkse) verwerking is een hoofdverwerkingsproces gedefinieerd waarin alle subprocessen sequentieel worden aangeroepen (zie afbeelding 9). Eerst worden de generieke controles uitgevoerd, waarna vervolgens alle staging area-



Afbeelding 4: Relatie tussen processen, data, architectuur en infrastructuur.

```
create view VW_TRANSFORM_<WAREHOUSE_TABLE> as
select <transformation_expression> as
                      <WAREHOUSE_TABLE_COLUMN_1>
, ...
, <transformation_expression> as
                      <WAREHOUSE_TABLE_COLUMN_N>

from TB_STAG_<STAGING_TABLE_1>
, ...
, TB_STAG_<STAGING_TABLE_N>

where <transformation_expressions>
```

Afbeelding 5: Syntax voor een transformview.

processen worden uitgevoerd, daarna worden de datawarehouse-processen en, voor zover van toepassing, de fysieke datamart-processen uitgevoerd. De belangrijkste subprocessen als 'staglaag laad' en 'geglaag laad' impliceren per te laden tabel de aanroep van een stored procedure.

Belangrijk is de volgorde waarin het datawarehouse wordt geladen; eerst dienen stam- en domeindata te worden verwerkt, waarna er (afhankelijke) feiten worden geladen. Bij elke architectuurovergang wordt er bepaald of de verwerking mag doorgaan. Zijn er fouten geconstateerd, dan wordt het verwerkingsproces op een nette wijze beëindigd. Wanneer eenmaal de betreffende fouten zijn geanalyseerd en verholpen, dat wordt niet het 'gewone' hoofdverwerkingsproces opnieuw gestart, maar de 'herstart variant' (dit omdat er bepaalde zaken al zijn uitgevoerd die niet nogmaals dienen te worden uitgevoerd, dan wel op een nette wijze moeten worden teruggedraaid).

Infrastructuur

Momenteel is xDWH beperkt tot specifieke infrastructuur (zie afbeelding 4). Op basis van het beschrevene is onderzoek gedaan naar toegepaste infrastructuur, gebruiksgemak, en hiermee gemoeide kosten. Mocht er aanleiding toe zijn, dan kunnen de xDWH concepten worden geïmplementeerd met behulp van andere DBMS-, ETL- en/of BI-hulpmiddelen. Dit brengt met zich mee dat bepaalde functionaliteit (bijvoorbeeld templates) dienen te worden geconverteerd naar de betreffende hulpmiddelen. Behoudens kennis van xDWH en van de betreffende hulpmiddelen, en zolang de conversie 1-op-1 is (dat wil zeggen met behoud van onder meer triggers, stored procedures, transform- en loadviews, dimensie- en feitenviews), is dit geen bijzonder complexe taak. Het moge duidelijk zijn dat deze taak geen onderdeel van een xDWH-project is, maar voorafgaand ervan dient te zijn uitgevoerd.
xDWH kent vele gezichten. Hiermee wordt bedoeld dat xDWH kan worden gebruikt als 'proof of concept' om opdrachtgever en/of gebruiker het vertrouwen te geven dat de totaaloplossing werkt,

```
create view VW_LOAD_<WAREHOUSE_TABLE> as
select src.<COLUMN_1> as
                      <WAREHOUSE_TABLE_COLUMN_1>
, ...
, src.<COLUMN_N> as <WAREHOUSE_TABLE_COLUMN_N>
, case when trg.<PK_COLUMN> is null then 'I'
else 'U'
end as load_operation_type_cde

from VW_TRANSFORM_<WAREHOUSE_TABLE> src
left join TB_<WAREHOUSE_TABLE> trg
on src.<PK_COLUMN_1> = trg.<PK_COLUMN_1>
and ...
and src.<PK_COLUMN_N> = trg.<PK_COLUMN_N>

where src.<NON_PK_COLUMN_1> <>
                          trg.<NON_PK_COLUMN_1>
or ...
or src.<NON_PK_COLUMN_N> <> trg.<NON_PK_COLUMN_N>
```

Afbeelding 6: Syntax voor een datawarehouse loadview.

```
create view VW_DIM_<DIMENSION_TABLE> as
select <transformation_expression> as
                      <DIMENSION_TABLE_COLUMN_1>
, ...
, <transformation_expression> as
                      <DIMENSION_TABLE_COLUMN_N>

from TB_<WAREHOUSE_TABLE_1>
, ...
, TB_<WAREHOUSE_TABLE_N>

where <transformation_expressions>
```

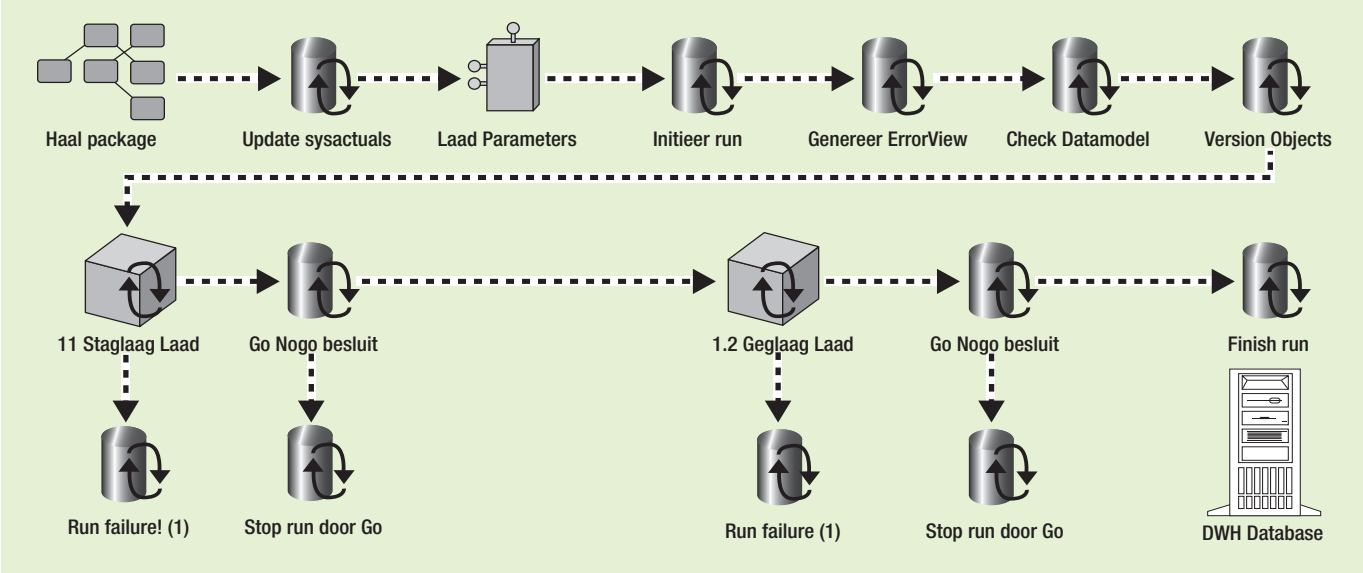
Afbeelding 7: Syntax voor een datamart dimensie(view).

```
create view VW_FACT_<FACT_TABLE> as
select <transformation_expression> as
                          <FACT_TABLE_COLUMN_1>
, ...
, <transformation_expression> as
                          <FACT_TABLE_COLUMN_N>

from TB_<WAREHOUSE_TABLE_1>
, ...
, TB_<WAREHOUSE_TABLE_N>

where <transformation_expressions>
```

Afbeelding 8: Syntax voor een datamart feit(view).



Afbeelding 9: Hoofdverwerkingproces.

als een generieke ontwikkelstraat voor een hele BI- en datawarehouse-omgeving, of als middel om aan de applicatiebeheer zijde te standaardiseren.

xDWH leent zich om zowel in architecturele (bijvoorbeeld door een datamart fysiek te implementeren) als infrastructurele (bijvoorbeeld door DTS te vervangen door een zwaardere ETL-tool) zin op te schalen naar het niveau dat een organisatie wenst, zonder dat weer helemaal opnieuw moet worden begonnen.

Martin Misseyer en **René Nobel**
Dr. Martin P. Misseyer (martin.misseyer@ordina.nl) is Profession Leader bij Ordina VisionWorks. Ir. René Nobel (rene.nobel@ordina.nl) is Senior Consultant bij Ordina VisionWorks.