

Verzekering tegen risicovol en kostbaar onderhoud goede keuze

# Een extra tussenlaag of niet?

Karien Verhagen

**Een terloopse opmerking in de laatste alinea van mijn vorige artikel luidde: “Een tussenlaag weglaten is meestal onverstandig”. Dat vonden enkele collega's wat vrijblijvend, een te boude stelling, te gemakkelijk geponeerd.**

Wanneer zet u in de architectuur een tussenlaag, wanneer is dat niet efficiënt en kostbaar? Er zijn voor en tegens voor elke tussenlaag. In dit artikel staan wat overwegingen die u mogelijk-erwijs verder helpen met die beslissing.

Grootste nadelen van een tussenlaag zijn de opslag- en beheer-kosten en de vertraagde verwerking van bron richting manage-mentinformatie. Het getuigt van Hollandse gepaste zuinigheid die tussenlagen dan ook tot een minimum te beperken. Maar goedkoop kan duurkoop zijn.

De route van bron naar managementinformatie gaat via een aan-tal tussenstops. Van operatie naar managementinfo leggen data een reeks bewerkingen af, die je functies op de dataverzameling zou kunnen noemen.

#### 1. Data naar real-time stuurinformatie:

- De data worden via een message broker of een trigger naar een rapportageomgeving gestuurd en gevalideerd;
- De data worden geïntegreerd in een al of niet virtuele ODS, als het real-time ontsluiten van méér bronsystemen dat noodzakelijk maakt;
- De data worden in het vraagformaat gezet;
- De data worden passief beschikbaar gesteld voor ad libitum gebruik of actief naar de gebruiker gestuurd als alert bij-voorbeeld naar een mobile device.

In mijn vorige artikel (DB/M 4, juni 2011) is er al op gewezen dat operationele stuurinformatie niet meer nodig heeft dan een rap-portagetool op de bron of eventueel een ODS. Pas wanneer de behoefte ontstaat aan corporate brede strategische en tactische stuurinformatie met een wijdere horizon, ontstaat ook de behoef-te aan een Enterprise Datawarehouse (EDW). De bewerkingen die de data op hun weg naar corporate stuurinformatie onder-gaan worden dan ook wat talrijker en gecompliceerder.

#### 2. Data naar corporate brede geconsolideerde stuurinformatie:

- De data worden geladen en op formaat gevalideerd;
- De data worden op verwerkingsvolgorde gelegd;
- De data worden gelinkt aan de data van andere bronnen en 'intersysteem' relaties worden gelegd;
- De data worden gezuiverd volgens beschreven correcties en validatiefuncties en een errorlog vangt de uitgevallen records op;
- Er wordt waar nodig historie opgebouwd, het proces dat Harm van der Lek zo mooi indweilen noemt;
- Er worden begrippen gemaakt die systeemoverschrijdend zijn;
- Er worden begrippen gemaakt waar de gebruiker in zijn eigen belevingswereld om gevraagd heeft. De data worden dan als vraagmodel gemodelleerd;
- Van het vraagmodel wordt een rapport- of analyseomgeving gemaakt.

Deze verwerkingsfunctie zou u als proces kunnen modelleren bijvoorbeeld als Steady State model, zie afbeelding 1.

In extremo zou je al die functies in aparte lagen kunnen onder-brengen. Het doet recht aan een aloude wet die de klassieke programmeur herkent als de wet van de functiescheiding.

Een groot aantal tussenlagen bouwen heeft voordelen. Het data-warehouse is in een lagenarchitectuur heel transparant en goed onderhoudbaar. Alle bewerkingen zijn logisch gescheiden, het resultaat wordt in lagen ondergebracht. Een nieuwe BICC-medewerker kan de plek van eventuele aanpassingen eenvoudig vinden. De structuur is overzichtelijk, de inleertijd kort.

Bovendien is de kans op fouten in een tussenlagenarchitectuur kleiner. Aanpassingen zijn sneller gemaakt, de structuur is min-der foutgevoelig.

Terugkomen op eerdere besluiten ten aanzien van definities hoeft de eerste twee en mogelijk drie lagen niet aan te tasten,

ook de voorkant kan blijven werken Omgekeerd gesteld is het aanpassen van een alles-in-één datamodel of ontsluitingsalgoritme zonder tussenlaag tijdens de rit heel gecompliceerd, riskant en soms zelfs onmogelijk. Tijdens de rit kan immers inhouden dat er aan de voorkant duizenden of miljoenen gebruikers werken die het EDW voor allerlei verschillende doeleinden gebruiken. Tot slot geldt dat tenminste één harde tussenlaag een must is, wanneer het datawarehouse de enige plaats is waar historische waarden worden opgeslagen.

## Een voorbeeld

Een vertragende factor is een extra tussenlaag ook lang niet altijd. Ter illustratie volgt een praktijkvoorbeeld.

Personeelskosten is zo'n begrip dat kenmerkend wordt opgebouwd uit de bewerking van vele velden afkomstig uit verschillende bronnen. De post 'personeelskosten' staat misschien als kostensoort in het grootboek, maar dan is zij al geaggregeerd. Zo was er in dit praktijkvoorbeeld een bron die de personeelskosten per uur registreerde en een ander bronsysteem waarbij we de personeelskosten tot hetzelfde detailniveau moesten herleiden uit een bedrag per maand, gedeeld door het aantal potentiële werknemers van de (in de maand wisselende) bezetting. Het mag duidelijk zijn dat aan deze berekening een intensieve discussie over de definitie vooraf is gegaan. Een definitie die door aanvullende inzichten ongetwijfeld aangescherpt of gecorrigeerd zal worden. Tot die tijd deden we het maar met deze. Het berekende detailveld 'personeelskosten per medewerker per uur' moet u in dit geval ergens een vaste plaats geven, in een databank, al of niet in de Cloud. Het stante pede berekenen is om begrijpelijke redenen niet verstandig. Als u een aparte laag hebt voor berekende velden is dat de plek. Wanneer de definitie wijzigt is die plek ook eenvoudig te vinden en aan te passen, veel eenvoudiger dan wanneer de berekening ergens onderweg van bron naar rapport in een algoritme zit verstopt dat ook valideert, corrigeert, integreert, historie opbouwt en formatteert. Zo is een extra tussenlaag ook een verzekering tegen het risico van voortdurend wisselende inzichten of lastige communicatie. Bij de opzet van

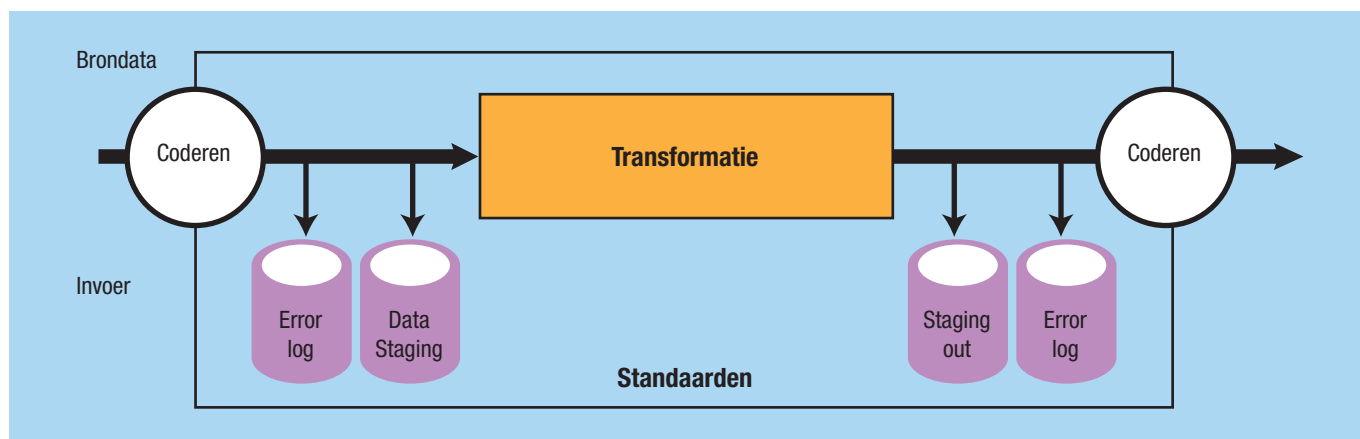
een Data Governance programma zullen in het begin de definities vaak aangescherpt worden.

Wat voor de begripssamenstelling geldt aan de voorkant, geldt ook voor het laadgedeelte aan de achterkant. Wanneer de bronindeling voortdurend wijzigt of de betekenis van de velden niet meteen helder is en onderhavig aan steeds nieuwe inzichten, ook dan is het aan te bevelen het laadstuk van de bewerking te scheiden. Een Data Vault model kan dan goede diensten bewijzen. Dat respecteert immers een wisselende bronmodellering en scheidt de herleiding van bronelementen naar corporate brede stuurinformatie op een heldere manier in de architectuur.

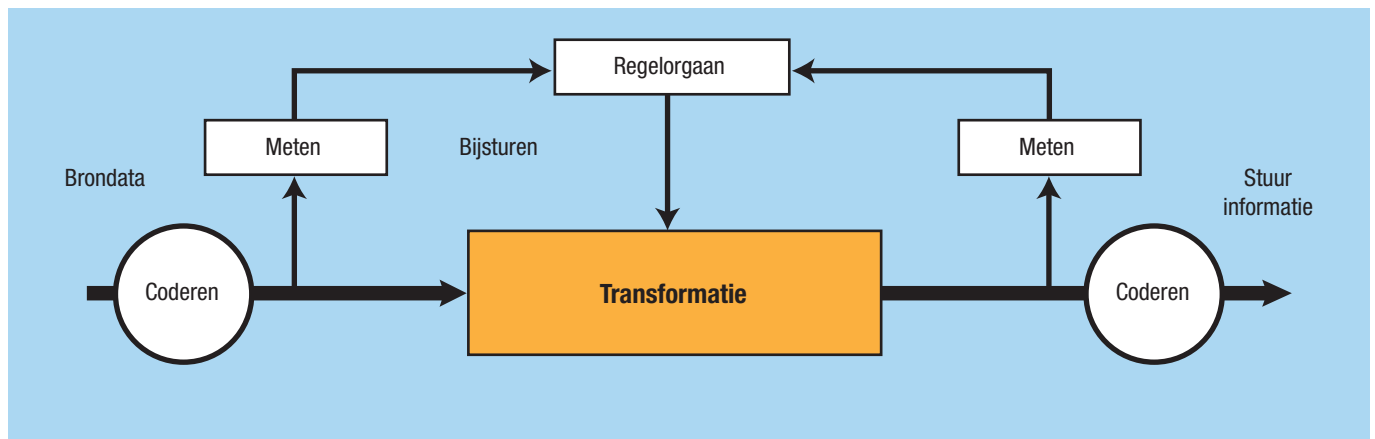
## Verwante oplossingen

Natuurlijk zijn er metadata driven designoplossingen en business rules engines die de algoritmes om te komen van brondata naar corporate brede begrippen helder in beeld brengen, zodat een tussenlaag daarvoor althans niet nodig is. De praktijk is echter dat niet veel bedrijven over die tools beschikken. Veel bestaande architecturen zijn opgezet met het Kimball model als uitgangspunt. Die architecturen botsen nu op de grenzen van dit model. Het model dreigt aan zijn populariteit ten onder te gaan. Van bron naar multidimensionaal vraagmodel is in de Kimball architectuur slechts één stap. Door de jaren heen is het een brei geworden van functies als begripssamenstelling, historieopbouw, validatie, integratie en synchronisatie, voorwaardelijke correcties en andere bewerkingen waar geen programmeur meer uit kan komen. U kunt dan één of twee lagen voor de datamart-laag zetten om bovengenoemde functies te scheiden. De datamarts blijven dan intact. Tijdens de operatie een laag ertussen bouwen is een relatief voordelige oplossing. U kunt dan parallel draaien via de oude weg en de nieuwe route met de ingevoegde tussenstops. Tijdens de migratie staat de functionele ontwikkeling natuurlijk wel stil. Het is een *relatief voordelige* maar toch een absoluut kostbare en complexe architectuurcorrectie die u vandaag de dag vóór kunt zijn.

Wanneer u de architectuur wijzigt moet u natuurlijk ook conform de andere bedrijfsprocessen het DWH-proces meetbaar en



**Afbeelding 1:** De transformatie omvat de opslag, de historie-opbouw, de creatie van corporate brede begrippen en de vertaling naar het vraagmodel, de codering omvat de vertaling van invoerformaat naar respectievelijk transformatie- of uitvoerformaat.



**Afbeelding 2:** Ook het DWH-proces is een bedrijfsproces waarvan het succes gemeten en gerapporteerd kan worden.

bestuurbaar maken. Hoe u dat doet en wat u dan meet is wellicht aanleiding voor een volgend artikel. Marjolein van den Heuvel heeft ooit een scriptie geschreven over de kritische succesfactoren van een BI-project die ik u met haar goedvinden graag toestuur. Het is toch raar dat we allerlei bedrijfsprocessen meten maar ons 'eigen' BI-proces niet, zie afbeelding 2.

## Conclusie

Een extra tussenlaag genereert kwantitatief meer opslag en beheer. Kwalitatief zal het onderhoud sneller en beter zijn wanneer u de functies van het informatieontsluitingproces in de DWH architectuur scheidt in verschillende lagen. Of dat ook effi-

ciënter en dus goedkoper is hangt samen met de complexiteit van de omgeving, de vluchtigheid van definities, vraagmodellen en bronnen en de beschikbaarheid van metadata driven modelleertools en business rules engines. Wanneer de architect weinig veranderende inzichten voorziet kan het aantal tussenlagen worden geminimaliseerd. Tussenlagen kunnen ook virtueel zijn. Wanneer veel roering wordt verwacht aan de voorkant, aan de achter- of de zijkant, dan is een verzekering tegen risicovol en kostbaar onderhoud in de vorm van een extra tussenlaag de beste keuze.

**Drs. C. Verhagen** is BI architect voor 4BIS scholing en advies.