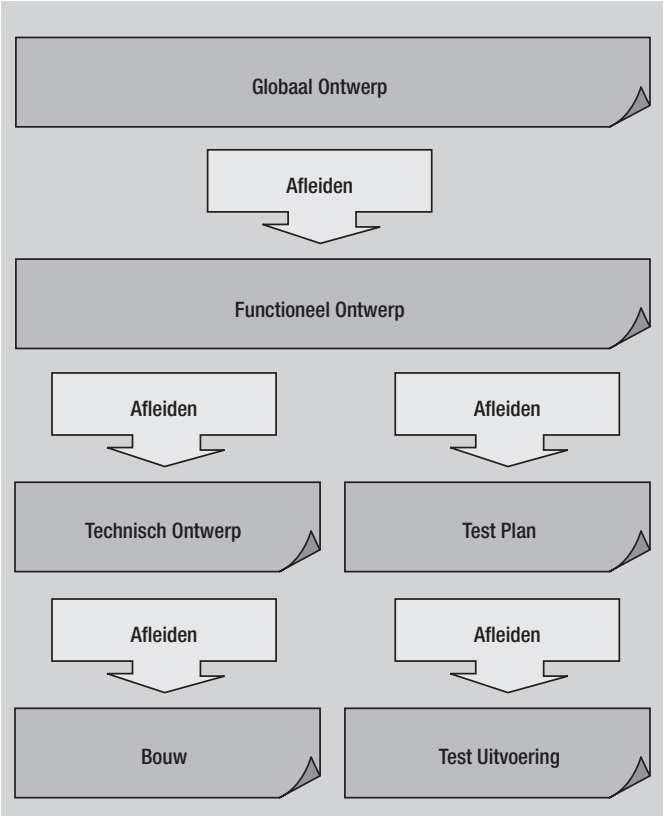
Methode voor documentatieaanpak DWH-omgeving

Documentatie op basis van metadata (1)

Burkhard Lau

Velen van ons stonden als ontwikkelaar of beheerder van een datawarehouse al eens voor de taak om de effecten van een wijziging van een bestaande informatiestroom in kaart te brengen (impact-analyse). Of wij moesten ooit als beheerder documentatie beoordelen met de bedoeling om het bijbehorende softwarepakket in beheer te nemen.

Velen van ons kennen dan ook het gevoel naar een speld in een hooiberg te moeten zoeken, waarbij het niet eens zeker is of de speld überhaupt in de hooiberg aanwezig is. Dit artikel is de eerste in een reeks van drie waarin een methode beschreven wordt die anders dan het conventionele documenteren meer op het genereren van de nodige documentatie inzet.



Afbeelding 1: Documenteren is de gebruikelijke manier om kennis over IT-processen vast te leggen.

De gebruikelijke manier om kennis over IT-processen vast te leggen, is om deze te documenteren, zie afbeelding 1.
Organisaties met een datawarehouse beginnen een documentatieprobleem te voelen, wanneer er wijzigingen op de bestaande processen plaats moeten vinden. Ondanks het feit dat er flink wat resources (inspanning, tijd, geld) in de opzet van standaards en richtlijnen voor het documenteren en het vervaardingen van documentatie zelf gestopt werd, heeft men vaak toch een gevoel van onvolledige of onbetrouwbare documentatie. Het lijkt wel een inherente eigenschap van documentatie om onvoldoende te zijn. Om toch een betrouwbare basis voor een volgend release te verkrijgen, worden dikwijls veel tijd en kosten besteed aan het *reverse engineren* van de bestaande processen; een inspanning die bij voldoende en goede documentatie overbodig zou zijn.

Soorten documentatie

Voor software bestaan twee soorten documentatie: Black Box-documentatie en White Box-documentatie.
Black Box-documentatie beschrijft het product van de buitenkant met de doelstellingen zoals vermeld in tabel 1.

Doelgroep	Document	Doelstelling	Aanpak
Gebruikers	Informatie-analyse	De gebruikers-vraag terug-koppelen	Vraag in rapportagevorm representeren
Gebruikers	Gebruiks-aanwijzing	Handleiding voor het gebruik van het product	De functionali teit van het product vertalen in de manier van gebruik
Operators	Operator-instructies	Informatie over het starten, stoppen en monitoren van het product	Foutcodes toe-lichten, batch windows specifi-ceren, herstart-acties uitleggen
DBA's	DBA-instructies	Waarborgen van beschikbaarheid van database	Opgave van nodige schijf-ruimte, backup-frequenties

Tabel 1.

Doelgroep	Document	Doelstelling	Aanpak
Applicatie bouw, beheer	Informatie-analyse, FO's en TO's	Betrouwbare basis voor verdere software releases	Door een overzichtelijke plattegrond snel de plek voor wijzigingen bepalen, en via impact-analyse het effect op andere informatiestromen in kaart hebben
Informatie-manager	Informatie-analyse, FO's en TO's	Datakwaliteit controleren	Gegevensdefinities beheren, de naleving van business-regels controleren

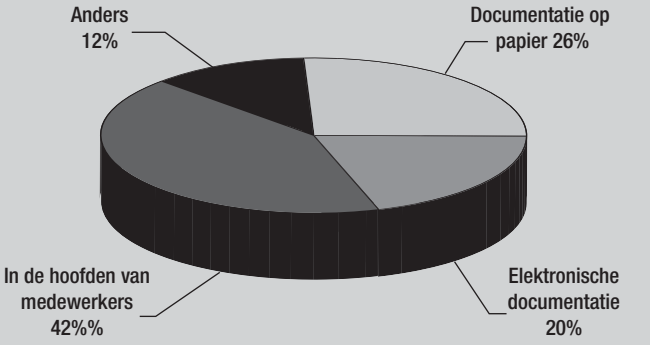
Tabel 2.

White Box-documentatie beschrijft de interne werking van het product met de doelstellingen zoals vermeld in tabel 2.
In deze artikelserie gaat het over White Box-documentatie, specifiek vanuit de beheersbaarheid van de opgeleverde processen en gegevensstructuren.

Probleemanalyse

In de praktijk bestaan twee oorzaken van gebrek aan vertrouwen in de bestaande documentatie: onvoldoende toegankelijkheid en onvoldoende betrouwbaarheid. Dan gaan we er overigens gemakshalve van uit dat men de juiste applicatie heeft om de documentatie in ieder geval te kunnen lezen. Meestal is dit in één of ander MS Office-formaat, maar wanneer deze in een ander formaat is opgesteld, zoals PDF, UML, SDW enzovoort, dan moet uiteraard wel de bijbehorende applicatie beschikbaar zijn.
Dat blijkt ook nog al eens voor problemen te zorgen.
Onvoldoende toegankelijkheid zoals in dit artikel bedoeld heeft ermee te maken dat de informatie alleen met moeite of helemaal niet ontsluitbaar is. Achter elkaar loopt men tegen de volgende drie niveaus aan:

- op welke plek moet ik het document zoeken;
- plek gevonden: welk van (de versies van) de documenten is het;
- versie gevonden: waar precies en op welke manier staat de informatie in het document?



Bron "Meta Data & Knowledge Management: *Meta Data Repository Myths*" by David Marco.

Afbeelding 2: Waar zijn volgens David Marco data gedocumenteerd?

Onvoldoende betrouwbaarheid heeft ermee te maken dat de informatie niet volledig en correct is:

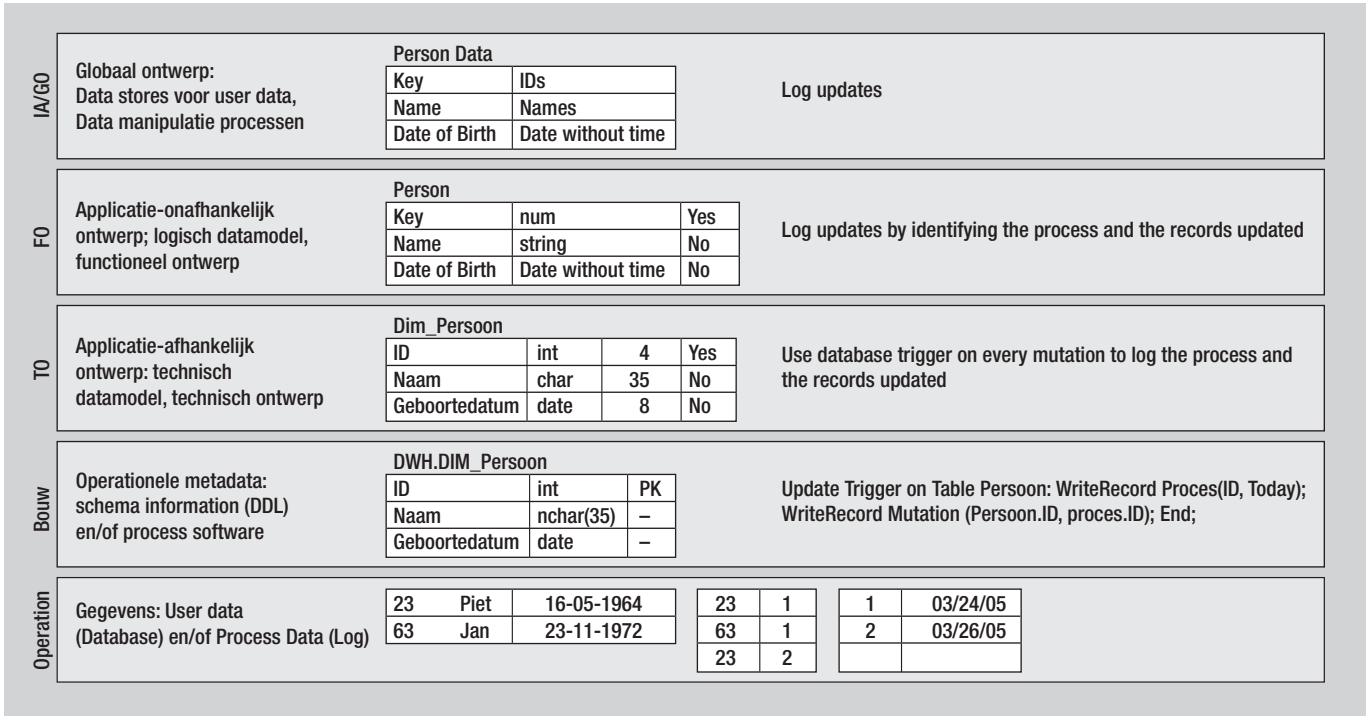
- informatie wel gevonden, maar blijkt niet (meer) overeen te komen met het proces (verkeerde documentversie);
- informatie niet gevonden, is nooit gedocumenteerd. David Marco, een internationaal erkende expert op het gebied van metadata stelt dat bijna 50 procent van alle informatie niet expliciet vastgelegd is (afbeelding 2).

Abstractieniveaus

Documentatie bestaat in drie niveaus van abstractie, alleen zijn de afbakeningen niet altijd even duidelijk. Als we veronderstellen dat een architect de functionele modules en hun interfaces van de applicaties (de grote lijnen), maar ook de te kiezen hardware en software (implementatie randvoorwaarden) heeft vastgelegd, hebben we hiermee ook de afbakeningen van de drie abstractie-niveaus vastgelegd (zie tabel 3).

Niveau	Essentie	Rol	Activiteit	Resultaat	Toelichting
Conceptueel	Wat – proces-onafhankelijke informatiestromen	Informatie Analist	Bepaling van informatiestromen	Informatie-analyse/ Bron-analyse	De informatiebehoefte van gebruikers in kaart brengen, en deze aan de bron(nen) relateren
Functioneel	Hoe – applicatie-onafhankelijk	Functioneel ontwerper	Toevoeging van procesinformatie	Functioneel ontwerp (FO) / Logisch Datamodel (LDM)	Het structureren van informatiestromen in processen
Technisch	Hoe – techniek-afhankelijk	Technisch ontwerper, Tool specialist	Vertaling van functioneel ontwerp naar de techniek	Technisch ontwerp (TO) / Technisch Datamodel (TDM)	TO tools zijn ETL tools, datamodelleringtools, en rapportagetools

Tabel 3.



Afbeelding 3: Voorbeeld.

Tussen de drie abstractieniveaus bestaan inhoudelijk relaties; ze zijn immers metadata voor hetzelfde gegeven, alleen op verscheidene abstractieniveaus voor verscheidene doeleinden. Er bestaat echter ook een relatie met de uiteindelijk gebouwde of gegenereerde applicaties en gegevensstructuren in de proceslaag, die op hun beurt weer een interpretatie aan de bedrijfs- en procesgegevens in de proceslaag geven (zie tabel 4).

Zie afbeelding 3 voor een voorbeeld van de hier geschetste vijf niveaus.

Toepassing op BI

Tot hier ging het over documentatie in algemene zin, BI heeft echter een speciaal applicatietype, waarmee we ook de documentatiebehoeftes specifieker kunnen maken. Een informatiestroom in een BI-omgeving bestaat altijd uit een aantal ETL-stappen met gegevensverzamelingen als begin en eindpunt, gevolgd door een publicatiestap om gegevens grafisch verantwoord in een afgesproken rapportagevorm aan gebruikers aan te bieden.

Hiermee hebben we meteen drie verscheidene soorten van documentatie te pakken:
1. Gegevensverzamelingen;
2. ETL-processen;
3. Rapportages.

Terwijl gegevensverzamelingen in veel meer gebieden dan alleen in BI worden toegepast, is de toepassing van ETL of rapportage betrekkelijk kenmerkend voor BI. De technieken en tools voor het vastleggen van documentatie voor gegevensverzamelingen zijn dan ook stukken verder verspreid en gevorderd dan voor ETL en rapportages het geval is.
Om een informatiestroom vanaf de bron tot een rapport te kunnen volgen, is het nodig om de bij elkaar horende documenten per abstractieniveau bij elkaar te kunnen voegen. Op technisch niveau is voor gestructureerde metadata het Common Warehouse Metamodel (CWM) bedoeld, maar voor ongestructureerde documentatie of documenten op functioneel niveau zijn geen integratiemogelijkheden bekend. Een integrale beschrijving op

Niveau	Essentie	Rol	Activiteit	Resultaat	Toelichting
Operationeel	Operationele metadata	Ontwikkelaar	Vertaling van technisch ontwerp naar een implementatie	Database schema's, ETL software, Report-definities	Alles wat TO tools niet kunnen genereren, moet gebouwd worden
Gegevens	Afspiegeling van realiteit of processen	DBA, operationeel beheer	Monitoring van processen en groei bedrijfsgegevens	Bedrijfsgegevens / Procesgegevens	Deze activiteiten hebben niets met metadata, maar met data te maken

Tabel 4.

Niveau	Gegevensopslag	ETL	Rapport
Conceptueel	Informatie model	Entiteit-operatoren, Attribuut-transformaties, Attribuut-validaties	Informatie model
Functioneel	Logisch Datamodel	Functioneel ontwerp met procesinformatie	Layout, autorisaties
Technisch	Databaseafhankelijk technisch datamodel	Technisch ontwerp met implementatie-afhankelijke details	Rapportage tool-afhankelijke realisatie

Tabel 5.

conceptueel niveau is mogelijk, wanneer een rapport als een entiteit binnen een informatiemodel wordt gezien. Dan kan namelijk de gehele informatiestroom als een volgorde van gegevenstransformaties worden gemodelleerd.

Documentatie-oplossing

White Box-documentatie is dus in een matrix te plaatsen, waarbij horizontaal informatiestromen beschreven worden, en verticaal deze beschrijving van informatiestromen op drie abstractieniveaus gebeurt (zie tabel 5).

De oplossing voor de boven genoemde problemen ligt in de relationele sfeer:

- Horizontale relaties: door de bij een informatiestroom behorende documenten aan elkaar te relateren, wordt de toegankelijkheid gewaarborgd;
- Verticale relaties: door vanuit de proceslaag via de technische en functionele documenten naar de conceptuele documenten van een bepaald item uit de proceslaag te relateren, is de documentatie verifieerbaar en hiermee dus is de betrouwbaarheid groter.

Als neveneffect zijn de geproduceerde documenten consistenter, waardoor ook ontwikkeltijden worden verkort.

Horizontale relaties: toegankelijkheid

De oplossing voor onvoldoende toegankelijkheid is om structuur aan te brengen. Zoals bestanden in een bestandssysteem door hun plek in een boomstructuur teruggevonden kunnen worden, kan ook informatie door het volgen van een structuur teruggevonden worden.

Documenten moeten zowel onderling structuur hebben (een externe structuur), als ook binnen elk afzonderlijk document dient een structuur te bestaan (interne structuur). Voor de externe documentenstructuur worden vaak afspraken gemaakt over directory-structuren en documentnaamgevingen, terwijl voor de interne documentstructuur *sjablonen* bestaan. Hiermee zou het probleem van de toegankelijkheid opgelost zijn, als de volgende punten uit de praktijk geen roet in het eten zouden gooien:

1. Zowel de externe als de interne documentenstructuren zijn niet expliciet beschreven, waardoor de betekenis onduidelijk is;
2. De externe documentenstructuur wordt niet nageleefd;
3. De sjabloon, die de interne documentenstructuur bepaalt, wordt vervormd.

Daarnaast is deze aanpak zeer gevoelig voor redundantie. Dezelfde structuur komt vaak voor in meerdere documenten, zelfs binnen één document kan dezelfde structuur gemakkelijk meermaals voorkomen. Dit is niet alleen saai voor functionele ontwerpers, maar werkt ook het ontstaan van inconsistenties in de hand.

Een andere invalshoek verkrijgen we als we de structuren van een sjabloon, maar ook de afspraken over externe documentatie-structuren, in databasestructuren vertalen. We hebben dan in plaats van Word wel een ander gebruikersinterface nodig, maar met dit uitgangspunt komen bovengenoemde punten óf te vervallen, óf wegen stukken minder:

1. De gekozen databasestructuur kan met metadata beschreven worden, de betekenis van elk item kan dus consistent op één plek worden vastgelegd;
2. De externe documentenstructuur wordt afgeleid uit de relaties in de database;
3. De interne documentenstructuur wordt afgedwongen, want de rubrieken op zich kunnen niet worden gewijzigd, alleen kan niet worden afgedwongen dat de rubriek ook inhoudelijk correct wordt ingevuld;
4. Sterkste punt is echter het ontbreken van redundantie. Structuren worden in de database slechts één keer vastgelegd, en kunnen vervolgens in diverse afgeleide documenten voorkomen, maar steeds op dezelfde manier.

Verticale relaties: betrouwbaarheid

In gebruikelijke projectmethodieken is de documentatie leidend, wat wil zeggen dat eerst de datastructuren en processen op papier worden gezet, die pas daarna gebouwd worden. Minder aandacht is er doorgaans voor het verifiëren of het technische ontwerp ook daadwerkelijk met het functionele ontwerp overeenkomt of dat het gebouwde proces ook volgens het technische ontwerp gemaakt is. Daarom wordt doorgaans ook *afgeleide* documentatie gegenereerd, dat betekent dat de bestaande datastructuren en processen worden *gere-engineerd* en de resultaten weer in documenten worden vastgelegd. Die documenten beschrijven dan het actuele beeld van het proces nauwkeurig. Deze documenten staan over het algemeen los van de projectdocumentatie, en geven eveneens naar verloop van tijd de realiteit niet meer goed weer.

De oplossing voor betrouwbaarheid is de *verifieerbaarheid* van de documentatie. Wanneer de documentatie op detailniveau koppelbaar is aan het gedocumenteerde proces of/en gegevensstructuur,

is zowel de volledigheid als correctheid controleerbaar. Dit vereist echter een identificatie van zowel documentdetails als procesdetails, en een omgeving om deze aan elkaar te kunnen koppelen. Een opslag in een database is een aangewezen manier om metadata onderwerpen aan fysieke tabellen en velden en ETL-processen te koppelen. De identificatie van gegevensstructuren en processen zal via hun technische naam gebeuren.

FCO-IM

Op het eerste gezicht lijkt de geschetste matrix op de aanpak van FCO-IM (Fully Communication Oriented Information Modeling). Beide aanpakken maken een onderscheid tussen drie lagen met onderlinge relaties. De aanpakken verschillen echter in hun doelstellingen, waardoor de uitwerking van de matrix en de relaties verschilt.

FCO-IM is een methode om gegevensstructuren op een eerder informele manier aan gebruikers te kunnen presenteren, en hiermee hun geldigheid ook te kunnen toetsen. Uit semantische gegevensmodellen kunnen logische modellen worden afgeleid en transformaties tussen gegevensmodellen kunnen ook worden beschreven. Deze relaties zijn echter niet de kern van FCO-IM en worden ook niet verder gebruikt.

De hier beschreven aanpak richt zich echter op het beschrijven van informatiestromen om deze beheersbaar te maken. Hiervoor zijn zowel beschrijvingen van gegevenstructuren als hun transformaties binnen één laag als ook de vertaling naar een lager of hoger abstractieniveau essentieel.

Het uiteindelijke doel van de informatiestromen, rapportages, zouden via FCO-IM gepresenteerd kunnen worden aan een gebruiker, terwijl de hiervoor benodigde databases alleen voor techneuten interessant zijn. Wel hebben we voor de beschrijving van informatiestromen nog enige procesinformatie nodig, die buiten de beschouwing van FCO-IM valt.

Conclusies

In dit artikel is getracht de oorzaken te identificeren van een vaak gehoorde klacht over onvoldoende documentatie, waarbij we het gebrek aan toegankelijkheid en betrouwbaarheid als oorzaken hebben gevonden. Nadat we hebben geconstateerd, dat White Box-documenten altijd via hun abstractieniveau en plek binnen een informatiestroom in een cel in een denkbeeldige matrix geplaatst kunnen worden, hebben we de introductie van horizontale en verticale relaties tussen de individuele documenten als oplossingen gevonden. In het vervolgartikel wordt besproken hoe deze horizontale en verticale relaties in de praktijk gerealiseerd kunnen worden.

Burkhard Lau

Dr. Ir. Burkhard Lau (burkhard@keper.nl) is senior BI consultant bij BI Garant BV.