

Genereer functioneel ontwerp vanuit gestructureerd conceptueel niveau

# Documentatie op basis van metadata (2)

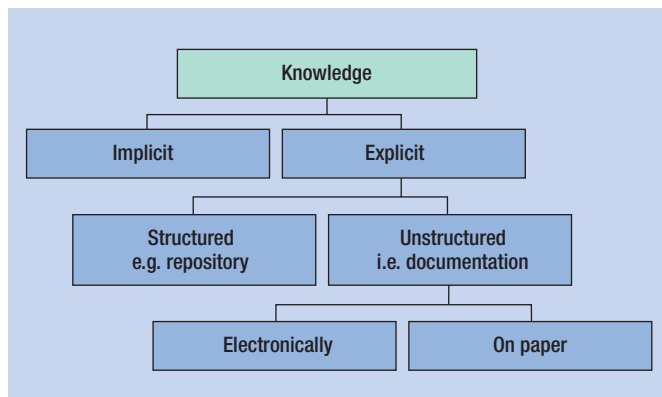
Burkhard Lau

**In het eerste deel is geconstateerd, dat de toegankelijkheid en betrouwbaarheid van White Box documentatie het beste te bereiken is als we de documenten in een matrixstructuur plaatsen en aan de horizontale en verticale burens relateren.**

De matrixstructuur is het resultaat van gangbare projectmethodieken, maar vanwege het gebrek aan relaties tussen de documenten wordt documentatie doorgaans als onvoldoende ervaren. Dit tweede artikel in het drieluik gaat over een manier om vanuit de tegenwoordig gehanteerde praktijk van documenteren, tot een aanpak te komen die wel relaties tussen documenten legt.

## De huidige situatie: vier kennisvormen

Documentatie is een vorm van kennis, verscheidene vormen van kennis worden getoond in afbeelding 1. Impliciete kennis bestaat alleen als kennis en ervaringen in de hoofden van mensen. Om deze kennis te waarborgen, worden de betreffende mensen geacht hun werkplekken niet te verlaten; maar als het dan toch door vakantie, vertrek of in het ergste geval door overlijden gebeurt, wordt de noodzaak voor een expliciete vorm van kennis duidelijk. Expliciete kennis kan gestructureerd of ongestructureerd zijn. Is een tekstdocument, gebaseerd op een sjabloon, gestructureerd? Onder gestructureerde kennis wordt in het kader van dit artikel een opslag in structuren verstaan met de volgende eigenschappen: aan de structuurelementen valt een betekenis toe te kennen; de kennis is in geautomatiseerde processen opvraagbaar.



Afbeelding 1: Impliciete en expliciete kennis.

Een database voldoet aan deze eisen, een sjabloon gebaseerd tekstdocument voldoet alleen onder de volgende voorwaarden:

- het moet gewaarborgd zijn dat de structuur tijdens het invullen niet gewijzigd kan worden;
- als de sjabloonelementen beschreven zijn;
- als een geautomatiseerde toegang via Word Automation of een ander interface mogelijk is.

Ongestructureerde kennis tenslotte, kan elektronisch beschikbaar zijn, of op papier in het archief. Alhoewel deze twee vormen naar elkaar om te zetten zijn, gebeurt het scannen van oude beschrijvingen niet vaak. De verdeling tussen deze vier vormen volgens metadata-expert David Marco (zie afbeelding 2 van het eerste deel in DB/M 1) toont een zorgwekkende situatie, dat bijna de helft van alle kennis niet ergens is vastgelegd maar slechts in hoofden van medewerkers bestaat.

## De gehanteerde structuur zorgt voor uniforme beschrijvingen

Kennis kan dus in allerlei vormen op verscheidene niveaus voor bepaalde doelstellingen bestaan. Sterker nog: zelfs voor één bepaald niveau, één doelstelling en één vorm kan en is de kennis niet homogeen, maar uiteenlopend beschreven. Zo is de functionele beschrijving van een ETL-proces tussen een bron en de staging area vaak afwijkend van het ETL-proces tussen de staging area en het Enterprise datawarehouse, en deze beschrijving wijkt structureel weer af van het ETL-proces dat aggregaten in een datamart vormt.

## Gewenste situatie: alles gestructureerd en gerelateerd

We hebben eigenlijk met drie problemen te maken: gebrek aan relaties tussen documenten; onvolledige documentatie; geen uniforme manier van beschrijvingen.

Vanwege deze problemen hangt rond documentatie een geur van overbodigheid, ontoegankelijkheid en onbetrouwbaarheid.

Niemand houd ervan om hetzelfde in meervoud te doen (op

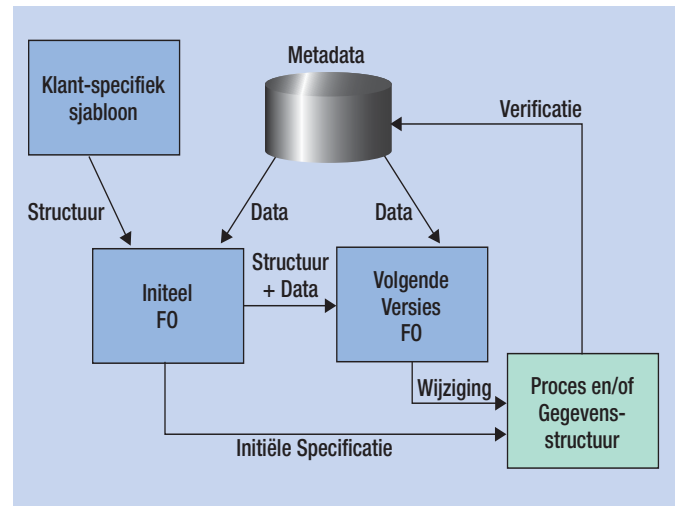
verscheidene niveaus documenteren en bouwen), geproduceerde documentatie blijkt vaak onvindbaar of onbegrijpelijk, en bovendien onvolledig en incorrect. Het is tijd om even terug te leunen en na te denken, waarvoor we de documentatie ook weer nodig hadden.

Bij datawarehouse-omgevingen gaat het om het doorgeven van informatie. Vanuit bronsystemen vloeit informatie via diverse tussenstations, zoals export files, een staging area, een datawarehouse, data marts, etcetera naar rapporten voor de eindgebruikers. Tijdens dit proces ondergaat een gegevenselement verschillende transformaties, en komt het in diverse gedaantes voor: een veldwaarde in een tabelkolom, een attribuutwaarde in een bericht, een aantal posities in een regel van een tekstbestand, een meetwaarde in een kubus, of als gegeven in een report. De documentatie moet dus informatiestromen inzichtelijk maken.

## Bij wijzigingsbeheer is documentatie essentieel

Documentatie speelt een rol tijdens het specificatieproces en tijdens het wijzigingsbeheer. In het specificatieproces wordt bij voorkeur niet de 'code & fix' aanpak gevolgd, maar beginnend met informatie-analyse van boven naar beneden (van conceptueel, functioneel naar technisch) gewerkt, waarbij elk onderliggend abstractieniveau uit het bovenliggend niveau afgeleid wordt. Voor een garantie van de betrouwbaarheid is de terugkoppeling van elk onderliggend niveau naar het bovenliggend niveau noodzakelijk. De relaties tussen de niveaus dienen dus ook vastgelegd te worden.

Bij wijzigingsbeheer is documentatie essentieel om enerzijds de plek op te zoeken waar de gewenste wijziging plaats moet gaan vinden, maar anderzijds moeten via een impactanalyse met behulp van documentatie ook de neveneffecten in kaart worden gebracht. Hiervoor is de toegankelijkheid van documentatie essentieel, informatiestromen moeten gevolgd kunnen worden vanaf de bron tot en met de report. Dit vereist een koppeling van



**Afbeelding 2:** Genereren van functionele ontwerpen vanuit een gestructureerd conceptueel niveau.

alle betrokken documenten, die dan ook volledig en begrijpelijk (dus zo uniform mogelijk) dienen te zijn. Deze eisen zijn alleen met gestructureerde documentatie op alle niveaus te realiseren. Dit laat nog wel de mogelijkheid open om documenten, diagrammen of overzichten uit deze gestructureerde kennis te genereren, maar idealiter ziet een documentatie-architectuur er uit als in tabel 1.

Omdat individuele structuren identificeerbaar zijn, zijn ze ook onderling horizontaal en verticaal relateerbaar, tot en met de operationele metadata in de proceslaag. De gehanteerde structuur zorgt voor uniforme beschrijvingen, en door de bijbehorende specificatie- en verificatieprocessen te implementeren, wordt ook de volledigheid en consistentie van de documentatie gewaarborgd.

### Migratie van huidige naar gewenste situatie

Met de visie van migratie van de huidige naar de gewenste situatie in het achterhoofd is een bepaling van eerste mijlpalen mogelijk. Als eerste mijlpaal gaan we ons richten op het genereren van functionele ontwerpen vanuit een gestructureerd

Niveau	Gegevensverzameling	ETL Proces	Rapportage
Conceptueel	Gestructureerd (entiteitsgroepen, entiteiten, attribuutdomeinen)	Gestructureerd (relationele operatoren, transformaties, validaties)	Reporten worden ook als gegevensverzameling gemodelleerd
Logisch	Gestructureerd (schema indeling, entiteiten, attributen, logische gegevenstypen), maar gepresenteerd in LDM tools	Gestructureerd als boven, maar nu worden de transformaties in acties samengevat, presentatie in documentvorm	Gestructureerd als boven, met layout technische toevoegingen
Technisch	Gestructureerd, in een TDM tool	Gestructureerd, in een ETL tool	Gestructureerd, in een report tool

**Tabel 1.**

Niveau	Gebied	Enkele voorbeelden ter illustratie zonder intentie op volledigheid
Conceptueel ontwerp	Algemene modellering	Rational Rose, Select
	Meta DWH	MetaStage, SuperGlue, MetaCenter, Adaptive Repository, Advantage Repository, MetaBase, InfoLibrarian, EMR, Logidex
Functioneel Ontwerp	Logische Data Modellering	Oracle Designer, ERWin, PowerDesigner, CaseTalk
	Proces Modellering	Data Integrator
Technisch Ontwerp	Data Modellering	Oracle Designer, ERWin, PowerDesigner
	ETL Proces Modellering	ETL Studio, Oracle Warehouse Builder, Data Integrator, DataStage

Tabel 2.

conceptueel niveau. In principe is het beschreven proces echter ook bruikbaar voor andere soorten van documentatie zoals functiepuntnalyses (FPA), testplannen, logische en technische gegevensontwerpen etcetera. In grote lijnen zal het proces er uit zien als in afbeelding 2.

Initieel:

1. Leg de metadata vast;
2. Genereer de eerste versie van het functioneel ontwerp, en laat een ontwerper de gegevens aanvullen;
3. Wanneer het proces gebouwd is, koppel het met de metadata.

Iteratief:

4. Voor wijzigingen: muteer de metadata;
5. Genereer een volgende versie door een kopie te maken van de

oude versie, de gewijzigde metadata bij te werken, en laat een ontwerper de gegevens aanvullen;

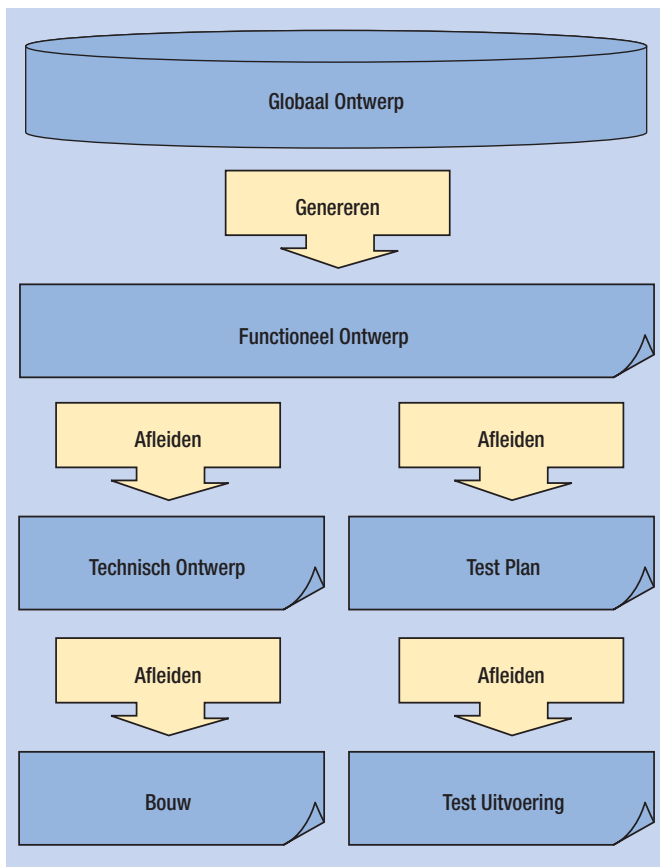
6. Wanneer het proces gewijzigd is, koppel het weer met de metadata.

Het conceptuele niveau zal altijd de actuele stand weergeven, terwijl elke versie van een functioneel ontwerp als Word-document bewaard blijft. De wijzigingen tussen twee opeenvolgende of willekeurige versies kunnen dan worden bepaald, door de twee documenten met elkaar te vergelijken. Als alternatief kan er ook voor gekozen worden om de wijzigingen van elk object op het conceptuele niveau in de tijd bij te houden. Hiermee zou de specificatiegeschiedenis van informatiestromen kunnen worden vastgelegd. Dit houdt echter een behoorlijke verhoging van de complexiteit van de structuur van het conceptuele niveau in, die onvoldoende baten heeft. Dit alternatief wordt dan ook niet gevolgd.

### Implementatie-aspecten

Hier is ook de vaak voorkomende discussie op zijn plaats om tussen zelfbouw en een standaard applicatie te kiezen. Op elk niveau bestaan ontwerptools, zie tabel 2.

Idealiter is het gekozen ontwerptool in staat om op alle niveaus zowel gegevensstructuren en rapporten als (ETL) processen op



Afbeelding 3: Generatie functioneel ontwerp.

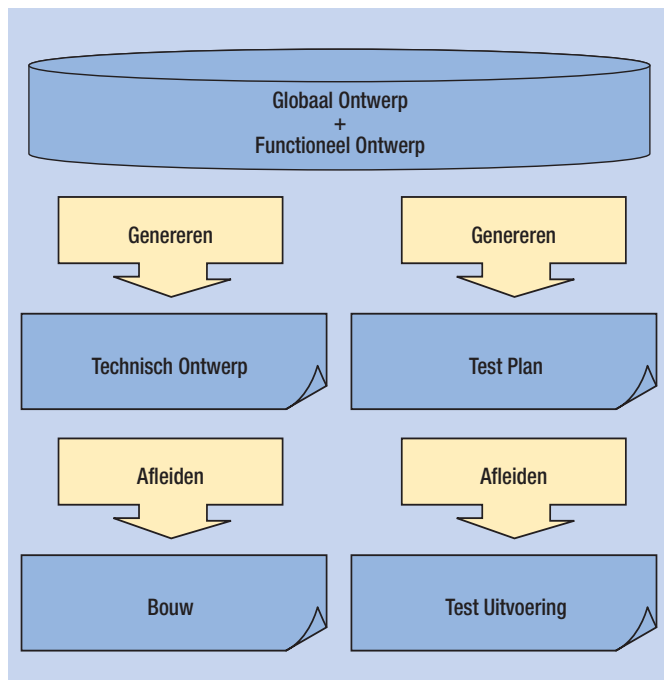
**logicaCMG**

**GURU4Pro**  
**Dr. Harm van der Lek**  
 'Meta Driven  
 Data Warehouses'

Woensdag 19 april  
 in Amstelveen

**GURU  
 4PRO**

Geïnteresseerd in deze of andere Guru4Pro's?  
 Kijk op [www.logicacmg.nl/guru4pro](http://www.logicacmg.nl/guru4pro).



**Afbeelding 4:** Functioneel ontwerp als onderdeel van de metadatabase.

een uniforme manier te modelleren en onderling te relateren. Hierdoor zou zo'n tool ook de volledigheid en correctheid van de documentatie garanderen.

In afhankelijkheid van de documentatiedoelstelling en de reeds aanwezige tools in een organisatie zal een keuze gemaakt moeten worden om bestaande tools, aan te schaffen tools en wellicht zelfbouw, in een architectuur te combineren, die aan de beschreven doelstellingen voldoet.

## Aanbevolen wordt om met een aantal nieuw te specificeren processen te beginnen

De praktische realisatie van een gestructureerd conceptueel niveau in een zelf ontwikkeld tool genaamd MetaInfo, leverde de volgende inzichten op:

- de specificatie van metagegevens voor de initiële versie van een functioneel ontwerp gaat niet significant sneller dan het conventionele schrijven van een ontwerp vanuit een sjabloon;
- de review cycle en het specificatieproces worden wel significant versneld, omdat de inhoud van de gegevens consistent is, en de interpretatie van sjabloonaspecten veel minder consultant-afhankelijk is;
- de meeste winst werd bereikt bij nieuwe releases; de inspanning voor het reverse engineeren van het bestaande proces is nihil, de nodige impactanalyse wordt snel en omvattend uitgevoerd, en aanpassingen zijn eveneens erg snel gedaan.

De hieruit getrokken conclusie was dat de kosten voor het opzetten van een gestructureerd conceptueel niveau zich snel terugverdienen. Voorwaarde is echter wel dat de relevante projectleden zich van de veranderde werkwijze bewust zijn, en deze ook praktiseren. Voor grotere projecten zal de rol van een kwaliteitsmanager nodig zijn om erop toe te zien dat eerst de metadata en daarna de documentatie aangepast worden.

Wanneer het de bedoeling is om met tools samen te werken, die functionele en/of technische metadata bevatten voor een bepaalde toepassing (bijvoorbeeld ERWin of Oracle Designer voor data-modellering, Data Integrator of ETL studio voor ETL-ontwerp), zal de gekozen oplossing aan de CWM (Common Warehouse Metamodel) specificaties moeten voldoen voor een soepele integratie.

## Stappenplan

De lezer zal zich wellicht afvragen, of de in het begin beschreven conventionele manier van documenteren makkelijk te vervangen is door een op metadata gebaseerde aanpak. In de praktijk valt dit inderdaad te doen door wel groot te denken, maar klein te beginnen.

Aanbevolen wordt om met een aantal, nieuw te specificeren processen te beginnen. Het *reverse engineeren* van bestaande kennis en documentatie in een metadatabase zal een lagere prioriteit hebben. Verder wordt aanbevolen om in een eerste implementatiestap het globale ontwerp in de metadatabase op te nemen, en de FO's (en optioneel ook een globaal ontwerp) uit de database te genereren, zie afbeelding 3.

Als verder in de tijd duidelijk wordt, wat over het algemeen nog handmatig aan de gegenereerde FO's toegevoegd wordt, kan deze informatie ook als metadata aan de structuur worden toegevoegd, en zullen de functionele ontwerpen ook onderdeel zijn van de metadatabase, zie afbeelding 4. In de ultieme situatie is er geen documentatie meer nodig zoals men nu gewend is. Wel zullen specificaties met tool repository's uitgewisseld worden, en zullen deze tools de gewenste processen en datastructuren genereren. Hiervoor is echter nog een lange weg te gaan, maar elke reis begint met een eerste stap.

## Samenvatting

Uitgaand van de huidige manier van documenteren en een visie op geheel gestructureerde en gerelateerde documentatie is de realisatie van een eerste mijlpaal uitgewerkt. De noodzaak van een gestructureerd conceptueel niveau is duidelijk gemaakt, en het proces van generatie van functioneel ontwerpen is beschreven. In het derde en laatste deel van deze reeks gaan we nader in op de nodige structuren om metadata op conceptueel niveau op te slaan.

### Burkhard Lau

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