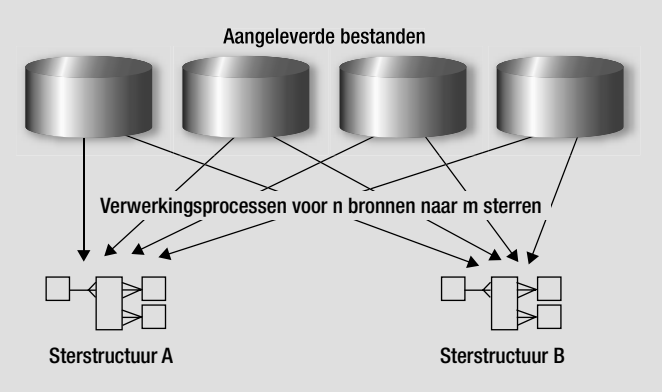
Parallel laadproces als tussenformaat

Het laden van toestanden: het fundament

Paul Hulst en Fons Rooijers

In het eerste artikel is beschreven dat het datawarehouse een hiërarchie van sterren kent, waardoor een ster een of meer verwijzingen naar andere sterren bevat, bijvoorbeeld de ster 'dienstverbanden' verwijst naar 'personen' en 'werkgevers'. Tevens is gemeld dat de gegevens van zo'n hogere ster (bijvoorbeeld 'personen' in het geval van dienstverbanden) ook in de ster 'dienstverbanden' worden opgenomen om het onderzoeken van dienstverbanden naar persoonskenmerken gemakkelijk en sneller te maken.

Het datawarehouse van het UWV zou dus omschreven kunnen worden als een stelsel van sterren met onderlinge verbanden. Er zijn veel bronsystemen[1] die gegevens leveren aan het datawarehouse. Als er gekozen wordt voor een sequentieel laadproces waarbij de gegevens uit een bepaalde bron in één slag verwerkt worden in een ster, dan ontstaat er een zeer ingewikkeld proces voor het verwerken van die gegevens. Er zijn namelijk bronnen die in meerdere sterren verwerkt moeten worden en er zijn ook gegevens die alleen verwerkt mogen worden als er in een andere bron juist of juist geen bijbehorende gegevens gevonden worden. Daarnaast moet er ook goed rekening gehouden worden met de volgorde van verwerking. Een sequentieel proces is geïllustreerd in Afbeelding 1.

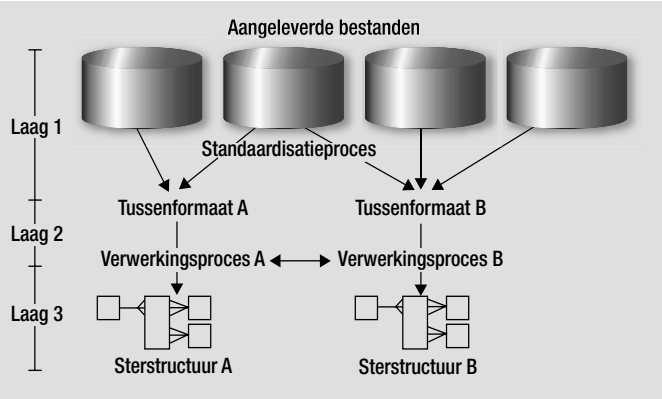


Afbeelding 1. Sequentieel proces.

Zo'n sequentieel laadproces is zeer complex qua structuur en daardoor lastig om te ontwerpen, te bouwen en te onderhouden. Daarom is voor het UWV gezocht naar een andere opzet voor het laadproces. Die andere opzet is een parallel laadproces geworden, waarbij soortgelijke processtappen tegelijk uitgevoerd worden. Om te garanderen dat de juiste volgorde van uitvoeren wordt gevolgd, zijn die processtappen in een de drie eerder genoemde lagen gegroepeerd, zie Afbeelding 2. Die lagen vallen vervolgens weer in sublagen uiteen. Een proces uit een specifieke (sub-)laag mag pas starten als alle processen uit de vorige (sub-)laag beëindigd zijn.
De processtappen in een specifieke laag zijn genummerd om de volgorde van verwerking aan te geven. Een proces voor een bepaald bestand of ster mag niet starten voordat zijn voorganger (voor hetzelfde bestand of ster) is beëindigd. Nu kunnen er gelijktijdig wel meerdere processen uit één laag lopen als die geen onderlinge afhankelijkheid hebben. Bij iedere laag zal aangegeven worden of er sprake is van onderlinge afhankelijkheid tussen de processtappen.

Het laden van toestanden (2)

In het artikel 'Het laden van toestanden' in DB/M 5-2003 beschrijven de auteurs de informatiebehoefte van de Uitvoeringsinstelling Werknemers Verzekeringen UWV en welke wensen er zijn voor het ontwerp van het datawarehouse en de bijbehorende laadprogrammatuur.
In het eerste artikel wordt een laadproces geschetst dat uit drie lagen bestaat:
Eerste laag, standaardisatie van aangeleverde bestanden;
Tweede laag, verwerking van die gegevens waarvan geen historie wordt bijgehouden;
Derde laag, verwerking van die gegevens waarvan de historische waarden wel relevant zijn.
In dit artikel zullen de eerste twee lagen worden besproken, waarbij ingegaan wordt op de processtappen in die lagen en waarom ze uitgevoerd worden. De derde laag wordt in het derde en laatste artikel beschreven dat in DB/M 7-2003 zal verschijnen; in dat artikel wordt ook ingegaan op de verschillende controle-aspecten bij het laden van de gegevens in het datawarehouse.



Afbeelding 2. Parallel laadproces.

Het tussenformaat

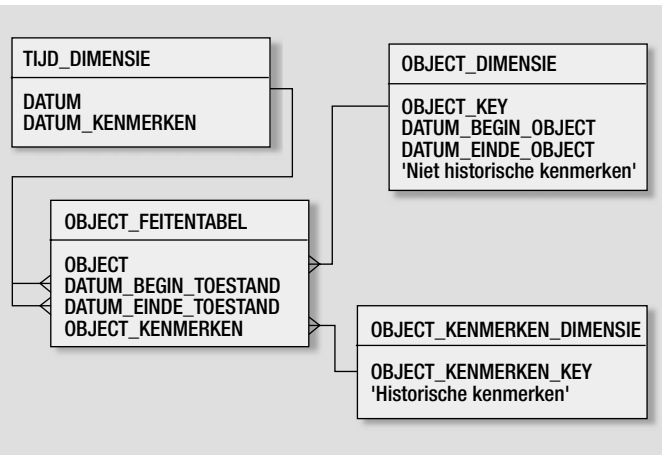
Een cruciaal onderdeel van het laadproces is de zogenaamde tussenformaat-tabel. Het is de scheiding tussen laag 1 enerzijds en lagen 2 en 3 anderzijds. In laag 1 worden de processtappen opgenomen die afhankelijk zijn van de structuur van de bronnen. De lagen 2 en 3 bevatten de processtappen die afhankelijk zijn van de sterren.

Voor iedere ster is er één tussenformaat. De structuur van de ster bepaalt de structuur van de tussenformaat-tabel. Het bevat namelijk alle kenmerken die in de ster voorkomen plus gegevens over de periode waarin die kenmerken geldig zijn. Het is dus het ideale mutatiebestand voor een bepaalde sterstructuur[2].

In het eerste artikel is de structuur van een toestandsgeoriënteerde ster besproken en weergegeven als in Afbeelding 3.

Afbeelding 3 toont:

- Een feitentabel waarin de toestanden met datum_begin_toestand en datum_einde_toestand worden vastgelegd. In het tijdvak tussen die twee data zijn de historische kenmerken ongewijzigd.
- De primaire dimensie. Hierin worden de niet-historische kenmerken worden vastgelegd. In deze dimensie wordt tevens de periode vastgelegd waarin het object 'bestaat'. Vanuit de feitentabel wordt verwezen naar de primaire sleutel van deze dimensie.



Afbeelding 3. Basisopzet sterstructuren.

- Een of meer objectkenmerken dimensies waarin de historisch vast te leggen kenmerken worden vastgelegd. In de feitentabel is een verwijzing opgenomen naar deze tabel. Die verwijzing komt in plaats van de kenmerken zelf. Deze tabellen worden minidimensies[3] genoemd.
- Een tijddimensie die twee keer aan de feitentabel is gekoppeld, voor de datum_begin_toestand en voor de datum_einde_toestand van de periode.

In het tussenformaat dienen de volgende gegevens opgenomen te worden:

1. De logische sleutel van het object;
2. De datum waarop de kenmerken van het object de specifieke waarde krijgen;
3. De datum waarop de kenmerken van het object niet meer geldig zijn (optioneel);
4. De kenmerken van het object uit de primaire dimensie, dit zijn de kenmerken waarvoor geen historie hoeft te worden bijgehouden;
5. De kenmerken van het object uit de minidimensies, waarvan dus wel historie wordt bijgehouden;
6. Een verwijzing naar de bron waaruit de gegevens komen, dit om herkomstanalyses mogelijk te maken. Daarnaast ook het moment waarop de kenmerken zijn vastgelegd in het bronsysteem zelf. Indien er meerdere uitspraken zijn die met elkaar in tegenspraak zijn (bijvoorbeeld twee verschillende geslachten voor een persoon), 'wint' de recentste uitspraak.

Bij de transformatie van de aangeleverde bestanden wordt de kennis die we hebben van het aanleverende systeem gebruikt. Bijvoorbeeld: in veel systemen wordt een mutatiecode gebruikt voor het weergeven wat voor mutatie in het aanleverende systeem is uitgevoerd. Gedacht kan worden aan het toevoegen van een nieuw object, verwijdering van een object uit het aanleverende systeem of een specifieke mutatie op het object. Die kennis wordt gebruikt bij het verwerken van een mutatie in het tussenformaat. Sommige mutatiecodes zullen anders worden verwerkt dan in het aanleverende systeem gebeurde. De mutatiecode die aangeeft dat het object verwijderd is uit het aanleverende systeem leidt niet tot verwijdering uit het datawarehouse, maar het markeren van het object als 'verwijderd uit bron'.

Het kan ook voorkomen dat één rij uit een aangeleverd bestand leidt tot meerdere rijen in het tussenformaat. Bijvoorbeeld: indien in het aangeleverde bestand twee velden voorkomen waarvan in hetzelfde record ook een begindatum voor elk van die velden is opgenomen, dan moeten we twee rijen aanmaken in het tussenformaat (voor elk veld één, met de bij dat veld behorende begindatum.)

De eerste laag: standaardisatie

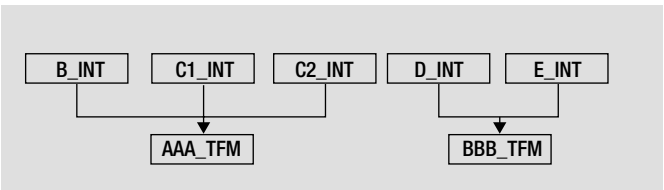
Doel van deze laag is drieledig: inlezen, controleren en standaardiseren. Door die standaardisatie is de verwerking van de bestanden onafhankelijk gemaakt van de toevallige structuur

waarin ze aangeleverd zijn.
De processtappen in laag 1 voor een specifiek bestand zijn niet afhankelijk van processtappen voor een ander bestand, ze kunnen daar parallel aan elkaar uitgevoerd worden.

Sublaag 1.A: controleren en inlezen van aangeleverde bestanden (oude opzet laag 1)
Op de gegevens in de bestanden worden allerlei technische controles uitgevoerd, gedacht moet worden aan juiste regellengte, correct gevulde datumvelden en zo verder. De resultaten van die controles worden in een hulptabel opgeslagen. Daarna worden de regels van het bestand geladen in een tabel.

Sublaag 1.B: fiattering aangeleverde bestanden door beheerder (oude opzet laag 2)
De beheerder onderzoekt de aangeleverde bestanden. Als hij overtuigd is dat het bestand een goede weergave is van de gegevens in het bronsysteem, zal hij het bestand laten verwerken en anders laten verwijderen.
Deze beslissing zal hij baseren op de resultaten van de technische controles uitgevoerd in sublaag 1.A en op inhoudelijke bestudering van de gegevens in het bestand. Voor deze sublaag is geen laadprogrammatuur nodig.

Sublaag 1.C: transformeren geaccepteerde bestanden naar tussenformaat (oude opzet Aa)
In deze sublaag wordt de informatie uit de aangeleverde bestanden getransformeerd naar een standaardstructuur. Het grote voordeel van deze transformatie is dat verdere verwerking voor alle bestanden dezelfde is. Natuurlijk worden alleen die bestanden opgepakt die aangemerkt zijn voor verwerking.



Afbeelding 4. Sublaag 1.C.

Voorbeeld:
Stel er is een ster gemaakt waarin gegevens worden vastgelegd over personen. De logische sleutel van een persoon is het sofinummer; de niet-historische kenmerken zijn het geslacht en de geboortedatum. Van zijn burgerlijke staat en de woonplaats wordt wel historie bijgehouden.

Stel er is een bestand met persoonsgegevens:

sofinr	dting	dtgeb	geslacht	dting_burgst	burgst	wnplts	dtreg	volgnr
123456789	1-2-02	1-1-49	M	1-3-02	Gehuwd	Hilversum	15-3-02	12
123456789	1-3-02	15-1-49		1-3-02	Ongehuwd		20-3-02	13

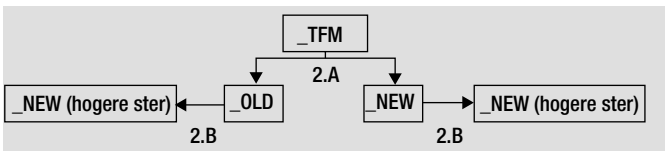
Het tussenformaat zou er dan als volgt uit zien:

Sofi-nummer	123456789	123456789	123456789
datum_begin_toestand	01-02-2002	01-03-2002	01-03-2002
datum_einde_toestand			
datum_geboorte	01-01-1949		15-01-1949
geslacht	M		
burgerlijke staat		Gehuwd	Ongehuwd
woonplaats Hilversum			
datum_registratie	15-03-2002	15-03-2002	20-03-2002
Volgnummer	12	12	13

Alle informatie in het bestand wordt vertaald naar deze structuur. Uit één regel in het aangeleverde bestand kunnen meerdere regels in het tussenformaat ontstaan. In het voorbeeld hierboven ontstaan uit de eerste regel in het bronbestand (volgnummer 12) de twee rijen in het tussenformaat die ook volgnummer 12 hebben.
Het zou natuurlijk zo kunnen zijn dat verschillende regels elkaar tegenspreken, zoals de tweede rij met volgnummer 12 en de rij met volgnummer 13. In dat geval zal het laadproces aan de hand van de registratiedatum bepalen welke informatie de juiste is.
Het tussenformaat is een tabel waarvan de kolommen hetzelfde datatype hebben als de overeenkomende kolom in de ster. In deze processtap zullen dus allerlei omzettingen van datatype plaatsvinden. Zo zal de string in het bronbestand die de geboortedatum bevat omgezet moeten worden in een date.
Dit is het enige deel van het laadproces waarin kennis over de betekenis van de velden in de bestanden gebruikt wordt om het datawarehouse te vullen. Dit is natuurlijk erg praktisch voor het onderhouden van de programmatuur.

Tweede laag: niet-historische kenmerken

Na de eerste laag beschikken we over een gevulde tussenformaat tabel voor alle sterstructuren die we van gegevens willen voorzien. Deze tussenformaten zijn de ideale mutatiebestanden voor de toestands- en transactie feitentabellen.
Het verwerken van de tussenformaten gebeurt vervolgens in een aantal stappen, zoals te zien in Afbeelding 5.



Afbeelding 5. Verwerken van tussenformaten.

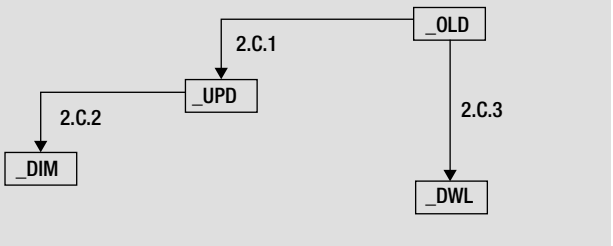
Sublaag 2.A: splitsing van tussenformaten in old en new
In deze sublaag worden de mutaties in het tussenformaat in twee groepen verdeeld en wel in mutaties die betrekking hebben op al bekende (_OLD geheten) en op nog onbekende objecten (_NEW). De reden hiervoor is dat de verdere verwerking iets anders is.

Er zijn geen afhankelijkheden tussen de verschillende tussenformaten, ze zouden dus allemaal tegelijkertijd verdeeld kunnen worden.

Sublaag 2.B: onderkennen nieuwe in hogere ster
Het datawarehouse bestaat uit een aantal sterren die gezamenlijk een hiërarchie vormen. Een ster kan dus een verwijzing naar een andere (hogere) ster bevatten, bijvoorbeeld dienstverbanden verwijzen naar werkgevers en personen. In het tussenformaat zitten daarom ook velden die gebaseerd zijn op die hogere ster, onder andere de logische sleutel. Nu kan het natuurlijk voorkomen dat een mutatie op een dienstverband verwijzingen bevat naar werkgever of persoon die nog niet bekend is. Die nog onbekende personen en/of werkgevers moeten ook aangemaakt worden in het datawarehouse om de referentiële integriteit te bewaren.
Die nog onbekende objecten in hogere sterren worden in deze sublaag opgespoord. Hiervoor worden alle rijen in het tussenformaat bekeken. Gecontroleerd wordt of de logische sleutel van de hogere ster al voorkomt in het datawarehouse. Daarnaast wordt ook gecontroleerd of de logische sleutel al voorkomt in het tussenformaat van de eigen ster. Als er immers al andere mutaties bekend zijn die dit object aan gaan maken, is het niet nodig er extra mutaties voor aan te leveren.

Deze processtappen kennen een grote onderlinge afhankelijkheid, ze moeten in omgekeerde volgorde van de hiërarchie uitgevoerd worden. Stel er is een sterrenstelsel waarin een uitkering verwijst naar een dienstverband en de dienstverbanden-ster verwijst weer naar personen. In dat sterrenstelsel moeten allereerst uit de uitkeringen de nieuwe dienstverbanden gehaald worden alvorens die dienstverbanden weer eventueel nieuwe personen kunnen leveren.

Sublaag 2.C: Bestaande objecten.
Doel van deze sublaag is het bijwerken van de gegevens waarvan alleen de actuele waarde relevant wordt geacht. Die gegevens worden vastgelegd in de 'primaire dimensie' van het object.
Het gaat in deze sublaag alleen om de objecten die al bekend zijn in het datawarehouse, zie Afbeelding 6.



Afbeelding 6. Primaire dimensie.

Er zijn geen afhankelijkheden tussen de processtappen voor verschillende sterren, ze kunnen dus gelijktijdig uitgevoerd worden.

Natuurlijk moet 2.C.1 afgerond zijn alvorens 2.C.2 voor dezelfde ster kan worden gestart.

Processtap 2.C.1 Dweilen mutaties
Er kunnen in de _OLD tabel meerdere mutaties op hetzelfde object zitten. In deze processtap worden die mutaties gecombineerd om te bepalen wat de actuele waarde van de verschillende kenmerken is. Een voorbeeld:
Als de persoon met sofi-nummer 123456789 uit het voorbeeld in 1.C al bekend is in het datawarehouse bevat de _OLD tabel de volgende rijen:

Sofi-nummer	123456789	123456789	123456789
Mutatiedatum	15-03-2002	15-03-2002	20-03-2002
Datum_begin_toestand	01-02-2002	01-03-2002	01-02-2002
Datum_einde_toestand			
Datum_geboorte	01-01-1949		15-01-1949
Geslacht	M		
Rijnummer	12	12	13

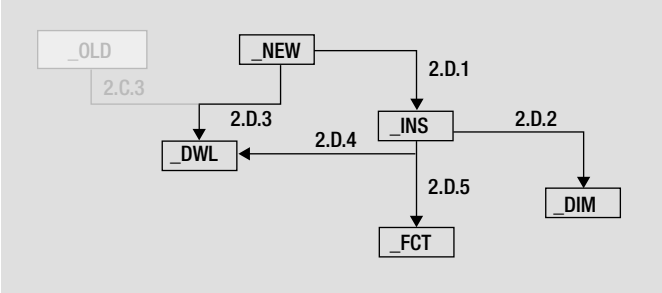
Het actuele beeld van die persoon is dat het een man is, geboren op 15-1-1949. Dat beeld wordt opgebouwd uit de twee afzonderlijke mutaties. Het geslacht komt uit de mutatie met rijnummer 12. De geboortedatum uit die rij wordt echter 'overruled' door de geboortedatum uit de rij met nummer 13. Die is namelijk dominant omdat hij later is ingevoerd, wat blijkt uit de mutatiedatum.
Per persoon wordt die éne blik in een aparte tabel gezet, geheten UPD (voor updatetabel). In het voorbeeld zal die tabel het volgende bevatten:

Sofi-nummer	123456789
Datum_geboorte	15-1-1949
Geslacht	M

Processtap 2.C.2 Wijzigen dimensie
In deze processtap wordt de dimensie gewijzigd aan de hand van de kenmerken in de tabel _UPD.

Processtap 2.C.3 Overzetten mutaties naar dwl-tabel
In de processtappen 2.C.1 en 2.C.2 zijn de kenmerken waarvan alleen de actuele waarde relevant is bijgewerkt voor de objecten die al bekend waren in het datawarehouse. In de aangeleverde bestanden zitten ook gegevens die betrekking hebben op kenmerken waarvan de veranderingen wel vastgelegd worden (in de feittabel). Voor het verzamelen van die mutaties wordt een aparte tabel gebruikt, genoemd _DWL (=dweil). In deze laatste processtap 2.C.3 worden de gegevens uit de _OLD –tabel toegevoegd aan die _DWL-tabel.
De _DWL-tabel is de centrale tabel waarmee in laag 3 de historische kenmerken zullen worden gewijzigd.

Sublaag 2.D: Nieuwe objecten
De sublaag 2.D is vergelijkbaar met de sublaag 2.C, het onderscheid is dat 2.C betrekking heeft op objecten die al bekend zijn in het datawarehouse en 2.D op objecten die dat nog niet zijn. Zie Afbeelding 7.



Afbeelding 7. Sublaag Nieuwe objecten.

Processtap 2.D.1 Dweilen mutaties
Deze processtap is functioneel gelijk aan processtap 2.C.1. Hij wordt alleen uitgevoerd op mutaties op objecten die nog niet bekend zijn in het datawarehouse. Het actuele beeld wordt in een andere tabel gezet, genoemd _INS.

Processtap 2.D.2 Toevoegen objecten aan dimensie
De rijen in de _INS tabel worden toegevoegd aan de primaire dimensie. Hierbij wordt tevens de technische sleutel voor dat object bepaald. Deze technische sleutel wordt ook in de volgende processtap gebruikt voor het vullen van de dweiltabel.

Processtap 2.D.3 Overzetten mutaties aan dwl-tabel
In deze processtap worden de mutaties in de _NEW tabel toegevoegd aan de _DWL tabel.

Processtap 2.D.4 Ophalen historie hogere sterren
In het eerste is artikel is het overerven van historie beschreven: de historie van een hogere ster (bijvoorbeeld personen) wordt ook opgeslagen bij een ster die naar de hogere ster verwijst (bijvoorbeeld dienstverbanden). Voor nieuwe objecten wordt die historie in deze processtap opgehaald en toegevoegd aan de _DWL tabel. Het gaat immers om kenmerken waarvan alle veranderingen bewaard worden.
Het kan natuurlijk voorkomen dat er een nieuw dienstverband wordt aangemaakt en er dus historie van de betreffende persoon wordt opgehaald. Dat is dan wel de historie die voor de start van het laadproces aanwezig was! Tegelijkertijd kan de historie van die persoon gewijzigd worden, bijvoorbeeld een verhuizing. Het laadproces zal ervoor zorgen dat die mutatie ook doorgevoerd wordt op het nieuwe dienstverband. Hoe dat gebeurt, zal in het derde artikel worden toegelicht.
Processtap 2.D.5 Toevoegen initiële rij aan feitentabel
In het datawarehouse bestaat een referentiële integriteit tussen feitentabellen en dimensietabellen. Voor de primaire dimensies geldt dat voor elke rij in de dimensie ten minste één rij aanwezig moet zijn in de feitentabel (is een toestand). Deze rij wordt aangemaakt wanneer een object wordt toegevoegd aan het datawarehouse.
De historische kenmerken van het object worden in deze feitrij op onbekend gezet. Op dit moment in het laadproces kennen we de historische kenmerken van het object namelijk nog niet. In laag 3 zullen die toegevoegd worden.

Resultaat van de eerste en tweede laag

De processtappen in de eerste laag hebben de aangeleverde bestanden gecontroleerd en ingelezen. Vervolgens zijn ze gestandaardiseerd naar een formaat dat gebaseerd is op de sterstructuur waarvoor ze als invoer dienen.
In de tweede laag zijn nieuwe objecten toegevoegd aan de primaire dimensietabellen en is zijn niet-historische kenmerken bijgewerkt aan de hand van de actuele kennis. In de _DWL-tabel zijn alle mutaties verzameld die iets zeggen over de kenmerken waarvan de veranderingen wel belangrijk worden geacht.
In de derde laag van het laadproces zullen die mutaties doorgevoerd worden op al aanwezige historie en zal beschreven worden hoe het gebruik van minidimensies het ruimtebeslag beperkt en de snelheid van uitvragen vergroot.
Deze laag zal worden beschreven in het volgende artikel. Daarin zal ook een uitgewerkt voorbeeld opgenomen worden van een simpele sterstructuur waarin een hiërarchie aanwezig is.

Fons Rooijers en **Paul Hulst** zijn beide betrokken bij het datawarehouse project van het UWV. Fons Rooijers (a.rooyers@chello.nl) werkt bij het UWV, Paul Hulst (phulst@deloitte.nl) bij Deloitte & Touche.

Literatuur
1 *Het UWV is ontstaan uit de fusie van 6 partijen. De integratie van de verschillende bronsystemen is nog lang niet afgerond.*
2 *Deze structuur wordt door Harm van der Lek 'One Attribute Set Interface' genoemd. Zie zijn artikel 'Het pletten van een ster' in DB/M 6-2000.*
3 *R. Kimball: The Data Warehouse Toolkit, 1996*