

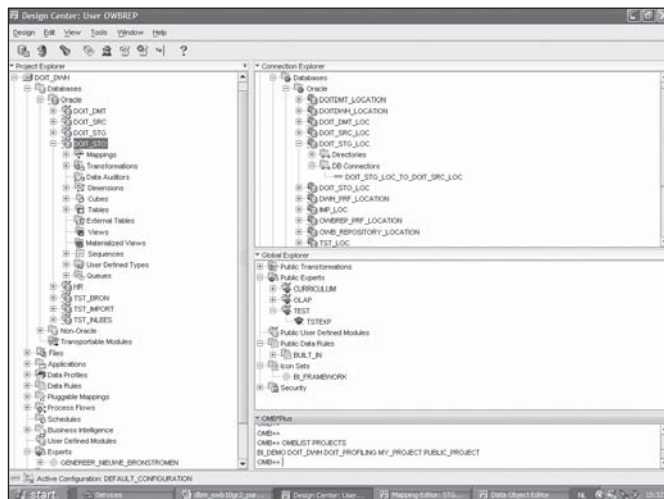
Oracle Warehouse Builder 10g release 2 getest in referentie business case

Data-integratiegereedschap krijgt meer functionaliteit

Bertram Hof

De nieuwe versie van Oracle Warehouse Builder (codenaam Paris) heeft lang op zich laten wachten. De laatst uitgebrachte versie draagt weliswaar de naam 10g maar bevat geen extra functionaliteit ten opzichte van de Oracle 9i release, en was dan ook voornamelijk bedoeld voor de ondersteuning van Oracle 10g. Sinds juni 2005 is de OWB 10g release 2 (Paris) voor bèta-testers beschikbaar. De productie-release wordt voor de zomer verwacht.

Met deze nieuwe versie positioneert Oracle het product niet meer alleen als een codegenerator voor ETL-achtige processen, maar meer als data-integratiegereedschap met metadata repository-potentie. Het is dan ook niet voor niets dat het product (als onderdeel van de Oracle BI-suite) in het vierde kwadrant van Gartner terecht is gekomen. Naast een volledig andere user interface is ook de functionaliteit zowel in de breedte als in de diepte enorm toegenomen. Het is in dit artikel niet mogelijk om alle aspecten van de nieuwe versie te behandelen, dit zou het product geen recht doen. Dit artikel beperkt zich dan ook tot de meest opvallende veranderingen en de in het oog springende functionele toevoegingen, zoals men deze bij het realiseren van BI-oplossingen op basis van datawarehouses in de praktijk kan tegenkomen.



Afbeelding 1: Design Center.

Design Center

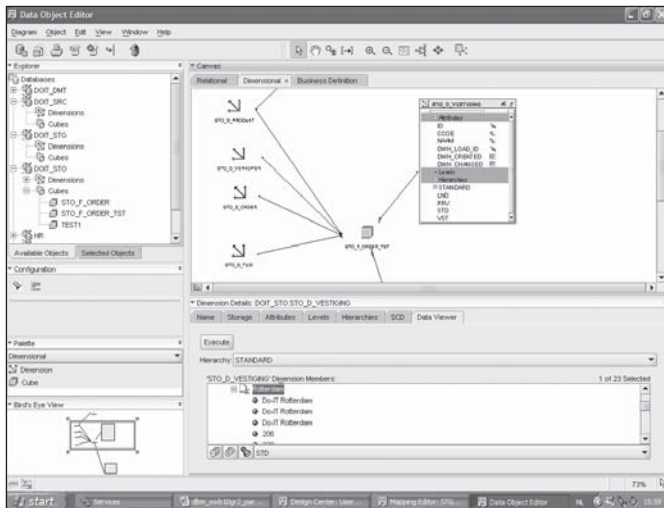
Menigeen die met de huidige versie van OWB werkt zal kunnen beamen dat de 'look and feel' niet vergelijkbaar is met een Window-applicatie, de Java-basis van het product is hier debet aan. In de Paris-versie echter heeft men dit aanzienlijk kunnen verbeteren. Niet alleen heeft men de applicatie kunnen voorzien van een beter ogende interface met een betere grafische kwaliteit, maar ook heeft men de consistentie in de gebruikersfuncties aanzienlijk verbeterd. In plaats van de huidige OWB Client-applicatie, waarbij de gebruiker voor nagenoeg iedere functie een apart popup-scherm krijgt om gegevens in te vullen, wordt er nu gebruik gemaakt van een Design Center met daar waar nodig goed ingerichte wizards voor snelle definities van objecten (zie afbeelding 1).

Op basis van de profileerresultaten kunnen Data Rules worden gedefinieerd

Als een echte IDE (Integrated Development Environment) voorziet het product de gebruiker van een overzichtelijke eenduidige interface, waarbij geassocieerde informatie goed, integraal toegankelijk is en flexibel door de gebruiker valt in te richten. Zo worden de binnen het huidige project beschikbare objecten overzichtelijk gepresenteerd in de Project Explorer, locaties in de Connection Explorer en algemene functionaliteiten in de Global Explorer. Zelfs werken met de scriptingtaal OMB (standaard onderdeel van OWB) is integraal aanwezig in het Design Center, wat het gebruik zeker zal bevorderen. Vanuit het Design Center worden de te behandelen objecten geselecteerd en geopend. Afhankelijk van het soort object worden de betreffende object-editoren dan aangeroepen. De twee belangrijkste hiervan worden verder bekeken.

Data Object Editor.

Deze editor heeft als taak om alle voorkomende data-objecten te kunnen onderhouden (denk hierbij aan tabellen, views, dimensies, cubes, etcetera). Door een efficiënte indeling van deze editor en het gebruik van tabbladen wordt de gebruiker niet overspoeld met



Afbeelding 2: Data Object Editor.

popups en dialoogboxen. In het begin is het wel even wennen aan 'wat staat waar', maar de voordelen worden snel zichtbaar. De editor zelf bestaat uit een tekengedeelte en een attributen-gedeelte. Ook hier kunnen weer vanuit een explorer objecten worden geselecteerd. Door gebruik van tabbladen kan informatie overzichtelijk en contextafhankelijk worden gepresenteerd. Doordat data-objecten onderling gerelateerd kunnen zijn door foreign key- of dimension/cube-associaties, zijn deze op het tekenvlak beschikbaar en selecteerbaar. Het geeft hierdoor een rustiger omgeving dan bij de huidige versie, waar de object editor iedere keer weer gesloten en geopend wordt en men snel de context uit het oog verliest.

Vooral gedurende de beheerfase van een datawarehouse is het van belang om een gevoel te krijgen bij de inhoudelijke gegevens die door data-objecten worden gedefinieerd. Het Data Viewer tabblad geeft hier de gelegenheid om query's op de database uit te voeren om direct een zicht te krijgen op de gegevens. Als er sprake is van een Analytic Workspace (Oracle OLAP) dat wordt de OLAP viewer (BI-bean) gebruikt. Een ander heeft natuurlijk alleen zin als objecten zijn *gedeployed* en gegevens zijn geladen (zie afbeelding 2).

Het instellen van de specifieke configuratie van data-objecten zoals indexen en storage clauses, vindt plaats via het Configuration-venster als onderdeel van de Data Object Editor. Dit gebeurde vroeger via het option menu configuration vanuit de Project Explorer.

Mapping Editor.

Het eigenlijke ontwerpscherm voor een ETL-proces is toch wel de Mapping Editor. Hier worden de transformaties aangegeven tussen de diverse data-objecten. Wat opvalt is uiteraard de gewijzigde interface. Werkend aan een mapping kunnen nu, naast selectie van objecten uit het Palette, direct vanuit de Project Explorer objecten worden geplaatst. De beperkte ruimte voor het

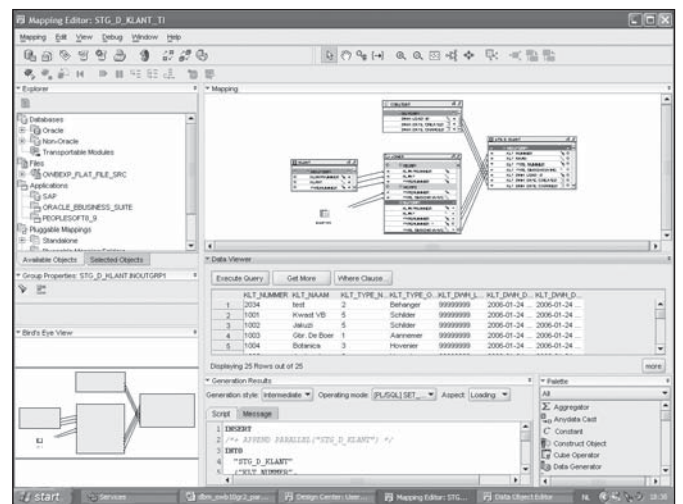
eigenlijke diagram (mappings, proces-flows, impact- en lineage-diagrammen) op het scherm wordt ruimschoots gecompenseerd door de vele view-manipulaties die de grafische editor biedt. Zo zijn er bijvoorbeeld een *bird's eye view* voor eenvoudige navigatie, een full-screen modus, en diverse layout- en zoom-functies aanwezig. Ook een functie om alle objecten als *icon* weer te geven ontbreekt niet. Deze functies zijn overigens bij alle grafische editors standaard aanwezig (zie afbeelding 3).

Het definiëren van expressies vindt weliswaar nog plaats via de Expression Editor, maar wordt wel integraal aangeroepen vanuit de Property Inspector en gelukkig kan men de ruimte die er voor een expressie beschikbaar is variabel instellen. Dit was vaak een nadeel bij de wat grotere join-condities. Het resultaat van de mapping, en hierdoor ontstaat ook vaak inzicht in de logica van de laadstream, kan ieder moment weer worden getoond door de generatie van code en is uiteraard integraal aanwezig in de Mapping Editor (intermediate result).

Ook kunnen aanwezige gegevens worden ingezien door de data viewers tijdens het ontwerpen van mappings, iets wat vaak een eerste stap is tot beter inzicht van het bouwproces. Een ander opvallende wijziging is de naamswijziging van reconciling naar synchronizing, om de repository en de toegepaste objecten in diverse editors synchronoos te houden. Dit draagt bij tot meer helderheid.

Control Center

De vroegere Deployment Manager heeft ook een ander uiterlijk gekregen en geeft nu ook toegang tot de Oracle 10g schedule-functionaliteit. Hierdoor is het uitvoeren van periodieke laadprocessen vanuit één repository en omgeving te beheren. Het Control Center geeft overzichtelijk weer welke taken met welk resultaat zijn uitgevoerd en geeft voldoende mogelijkheden om historisch uitgevoerde taken op detailniveau te bekijken. Vanuit het verleden zijn ETL-engines en ETL-generatoren groot geworden door het goed kunnen oplossen van datatransformatie-



Afbeelding 3: Mapping Editor.

processen. Als onderdeel van een volledige BI-architectuur blijkt echter ook het opschonen, verrijken en beheer van gegevens (waaronder ook metagegevens), en dus kwaliteit van de gegevens, essentieel voor het verkrijgen van meerwaarde van de BI-oplossing. Denk hierbij bijvoorbeeld aan de voorwaarden die worden gesteld aan Corporate Data Master Management. Oracle zelf spreekt dan ook van "informatie = kwaliteit (data + meta-data)". In de tool OWB komt dit tot uiting door onder andere de nieuwe data profiling-module.

Data profiling

Voor juiste informatieverstrekking naar eindgebruikers en beheerders is het van belang dat men kennis krijgt over de gegevens. In nagenoeg ieder BI-project is er een activiteit 'bron-onderzoek' gedefinieerd. Hierbij wordt niet alleen gekeken naar de gedefinieerde structuur van de gegevens (metadata) maar juist naar de daadwerkelijk aanwezige gegevens en de mate van compliance hiervan met de bedoelde datastructuur. Data profiling geeft inzicht in de daadwerkelijk aanwezige structuren van de te laden gegevens. Dit geeft op zich weer inzicht in de kwaliteit van de gegevens en kan aanleiding zijn tot het eerst oppoetsen van gegevens (correctie-mappingen), alvorens deze worden doorgeladen naar het uiteindelijke datawarehouse. Ook kunnen de resultaten van een profiling-exercitie worden gebruikt om bewuste uitvalstromen in te richten, die dan weer terug gaan naar de gegevensbeheerder om, gecorrigeerd, opnieuw te worden aangeboden. Allemaal functionaliteiten die een normale BI-oplossing dient te bevatten om correcte informatie aan de business te kunnen leveren.

Door de profiling een onderdeel van de totale toolset te maken en integraal in de IDE op te nemen, wordt het een stuk eenvoudiger om deze functionaliteit te realiseren. Men begint hierbij met het profileren van een beperkte hoeveelheid gegevens waarvan men de kwaliteit wil onderzoeken. Er worden drie typen profielen binnen de applicatie onderkend, zie onderstaande tabel.

Type	Omschrijving
Attribuut	Domein, unique key en patroon detectie.
Functionele afhankelijkheid	Hierbij worden functioneel gerelateerde gegevens binnen een entiteit gedetecteerd. Denk hierbij bijvoorbeeld aan omschrijving en code-combinaties.
Referentiële Integriteit	Potentiële foreign key-relaties tussen een of meerdere kolommen van verschillende entiteiten worden worden hierbij gedetecteerd.

Het is zinvol om selectief om te gaan met het uitvoeren van profileren. Door de vele mogelijke combinaties en structuren die tijdens een profileer proces worden doorlopen kan de doorlooptijd behoorlijk oplopen. Meestal neemt men een aantal attributen waarvan men inzicht wil krijgen over de kwaliteit van de gegevens – denk hierbij bijvoorbeeld aan klantgegevens (zie afbeelding 4).

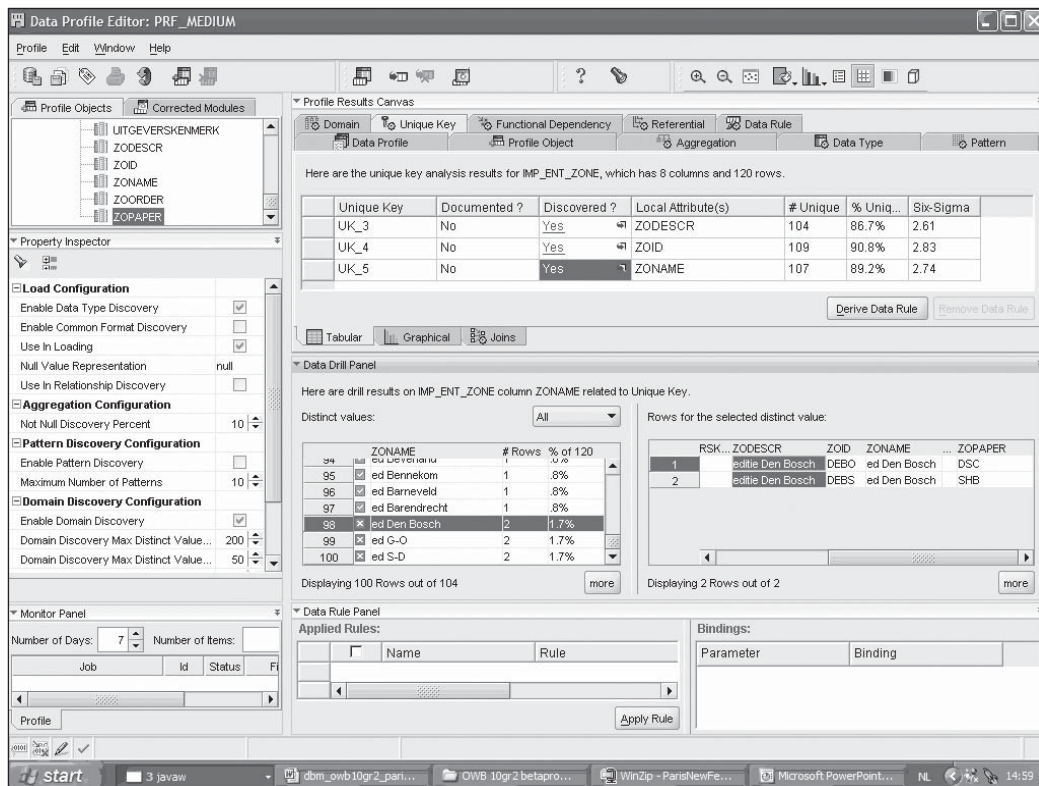
Door gebruik van tabbladen wordt informatie contextafhankelijk gepresenteerd

Via de Property Inspector kan men zeer individueel instellen op welke wijze er geprofileerd wordt. Zo kan men domeindetectie doen op grond van een maximaal aantal voorkomende distincte waarden of een bepaald percentage van de totale hoeveelheid records. Bij feitgegevens bijvoorbeeld is het zinloos om een domein te detecteren op meetattributen. Dan is het dan ook verstandig om deze niet mee te laten profileren. Zoals gezegd is het belangrijk om niet de gehele bron te profileren. De resultaten worden via de Data Profile Editor zeer gebruikersvriendelijk gepresenteerd. Door de vele tabbladen en deelschermen is het even zoeken, maar men merkt dat er een visie achter zit. Ieder tabblad geeft inzicht in de resultaten voor een bepaald type profiling.

De kwaliteit van gegevens, bijvoorbeeld de mate van uniekheid, wordt zowel procentueel als via Six-Sigma uitgedrukt. Six-Sigma is een eenheid voor het uitdrukken van het aantal afwijkingen per volume eenheid (het aantal fouten per totale hoeveelheid aangeboden records) en wordt veelvuldig gebruikt bij het aangeven van de kwaliteit in processen. Resultaten kunnen ook grafisch worden weergegeven, wat de kwaliteit snel inzichtelijk maakt. Op basis van de profiling-resultaten, ervaring en business-kennis, kan nu eenvoudig worden geanalyseerd of de gegevens aan een bepaalde verwachting voldoen. In het voorbeeld in afbeelding 4 is te zien dat er een uniekheid is gedetecteerd voor 89,2 procent van het ZONAME attribuut (een editiebenaming voor een kranttitel), en dat de bijbehorende ZOID uniek is voor 90,8 procent. De verwachting was dat er een 100 procent uniekheid zou zijn en dat zowel de ZOID als de ZONAME uniek zouden zijn. In het 'Data Drill Panel' kunnen we direct zien welke records niet uniek zijn. Hieruit is dan op te maken of het echt foute records zijn of dat er bij het laadproces, en dus de transformaties, rekening moet worden gehouden met deze uitzonderlijke situaties.

Data Rules

Op basis van de profileerresultaten kunnen er nu zogenaamde 'Data Rules' worden gedefinieerd en/of afgeleid (derived). De



Afbeelding 4: Data Profile Editor.

rules kunnen dan vervolgens worden gebruikt om correcte datadefinities in het datawarehouse te implementeren (aangepaste datatypes toepassen van constraints etcetera) en eventuele correctie-mappingen te genereren. Zo kan men bijvoorbeeld bij een e-mail adres van een klant via patroonherkenning tijdens het laden, laten testen of deze aan de juiste structuur voldoet. In de Oracle 10g database wordt dit dan in een functie via zogenaamde Regular Expression-syntax (conform de POSIX-standaard) geïmplementeerd.

Voorbeeld van een regular expression voor een e-mail structuur in Oracle 10g:

```
(^[[:space:]]*[A-Za-z0-9]+\.[A-Za-z0-9]+\@[A-Za-z0-9]+\.[A-Za-z]{2,3}[[:space:]]*$)
```

Zoals gesteld kunnen op basis van toegekende data rules verbeterde datastructuren en benodigde correctie-mappingen worden gegenereerd. Deze gebruiken dan de standaard gedefinieerde rules (geïmplementeerd door voorgedefinieerde of maatwerkfuncties) om de gegevens die worden aangeboden te verladen naar een gecorrigeerde/gevalideerde omgeving. Hierbij kan worden gekozen of de niet conforme gegevens worden overgeslagen, gecorrigeerd of alleen gerapporteerd in de audit-tabellen.

Via wizards worden de benodigde correctie-datastructuren en -mappingen gegenereerd. Logische volgende stap is dan om de correctie-mappingen op te nemen in de laadprocessen. De door Oracle gegenereerde mappingen hebben wel een afwijkende structuur en zijn daarom niet door de mapping editor te onderhouden. Veranderingen in de profiling en hiervan afgeleide rules

leiden dan ook tot regeneratie van correctie-mappingen.

Eveneens op basis van de aangelegde rules kan er preventief worden geaudit tijdens het laadproces. Hiervoor worden dan speciale Data Auditors gegenereerd. Deze mappingen kunnen worden ingezet om de gegevens op kwaliteit te monitoren tijdens het laadproces op basis van ingestelde kwaliteitsnormen. Stel dat er gedefinieerd is dat er een bepaald percentage uitval geaccepteerd mag worden, alvorens een laadproces af te breken of een gegevenslevering af te keuren. Een aangemaakte Data Auditor wordt dan in de proces-flow opgenomen en de uitgang van het proces levert criteria (percentage-uitval/Six-Sigma) op basis waarvan het verdere verloop van het proces kan worden gestuurd.

Verbeterde integratie met BI-rapportages

Niet alleen aan het begin van het datawarehouse biedt OWB extra functionaliteit door profiling, maar ook een vergaande integratie naar de rapportagekant van een BI-oplossing wordt door OWB gefaciliteerd. Door het opnemen van business definitions en business presentations in de OWB repository, kunnen objecten voor discoveren en BI-beans worden 'gedeployed' en ontstaat er een end-to-end metadata-model waarmee impact- en lineage-rapportages mogelijk worden. Uiteraard gaat Oracle hiermee het verst voor de eigen producten maar door het toevoegen van user defined objects en icon's kunnen ook andere rapportage-objecten en zelfs SQL scripts als rapportage-objecten hieraan worden toegevoegd; hetgeen de functionaliteit van de impact- en lineage-rapportages en de voortaan hieraan gerelateerde change manager propagation aanzienlijk waardevoller maakt.

Deze laatste biedt in de nieuwe versie de mogelijkheid om vanuit een impact-analyse een activiteitenlijst van uit te voeren wijzigingen aan te leggen. Vanuit deze lijst kunnen dan de diverse objecten direct in de benodigde editors worden aangeroepen en wordt het doorvoeren van een wijziging aanzienlijk eenvoudiger. Naast de diverse toevoegingen aan het werkterrein ter ondersteuning van de nieuwe positionering van OWB, is er ook veel veranderd en toegevoegd aan het product ter verbetering van de oorspronkelijke kernfunctionaliteit, het uitvoeren van ETL-taken.

Definitie Dimensionaal Model

De meest in het oog springende verandering heeft plaatsgevonden in de functionaliteit om een dimensionaal model te definiëren en te onderhouden. Hierbij gaat het om het aanmaken van dimensies en cubes. Er is nu volledige ondersteuning voor historie vanuit de tool. Hierbij worden de typen 1, 2 en 3 van de Kimball-theorie ondersteund. Men heeft hiervoor de first class-objecten 'dimension' en 'cube' op een hoger abstractieniveau gemodelleerd. Voor een dimensie bijvoorbeeld wordt niet alleen vastgelegd uit welke attributen, niveaus en hiërarchieën deze bestaat, maar ook op welke wijze historie moet worden bijgehouden.

De attributen voor de geldigheid (effective/expiration date) zijn instelbaar en/of worden automatisch gegenereerd. Door informatie op deze manier toe te voegen aan de objecten in de repository kan bij het maken van een laad-mapping voor een dimensie worden volstaan met het aangeven waar de informatie vandaan geladen wordt en welke transformaties deze dient de ondergaan. De eigenlijke verwerking van de gegevens in de dimensie wordt dan door de mapping-generator opgelost.

Bij een type 2 wordt dan automatisch een nieuw record aangemaakt en de oude situatie afgesloten. De code die hiervoor nodig is moet in de oude versie volledig met de hand worden ingebracht met splitters, joins en set-operatoren. Door de dimensie operator nu als een pluggable mapping (nieuwe mogelijkheid om een submapping, mapplet te definiëren), blijft de eigenlijke complexiteit voor de gebruiker verborgen. Achter de schermen wordt

natuurlijk wel gelijksoortige SQL gegenereerd als voorheen, maar de mapping wordt veel doorzichtelijker en onderhoud is op het niveau van het dimensie-object makkelijker uit te voeren. Ook een cube (feitentabel) wordt op deze wijze gedefinieerd. Hierbij worden de relaties met dimensies aangegeven en worden measures gedefinieerd. Als nu een laadstroom voor een kubus wordt gemaakt dan is Oracle op basis van de repository-informatie van de kubus en hieraan gerelateerde dimensies, in staat om correcte source code te genereren. De eigenlijk gegenereerde structuur is inzichtelijk te maken door de pluggable mapping als onderdeel van de mapping te bezoeken (zie afbeelding 5).

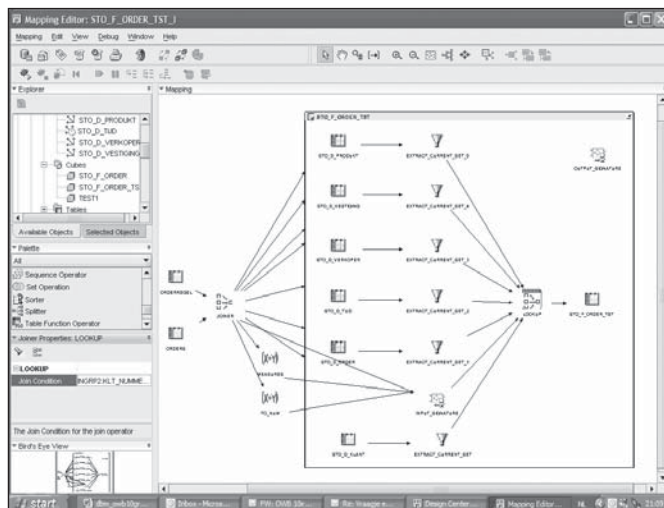
Implementatie Dimensionaal model

Niet alleen de definitie van dimensionale objecten is op een hoger niveau gebracht, maar ook de implementatiemogelijkheden van deze objecten zijn aanzienlijk verbeterd. Zoals bekend beschikt Oracle ook over een OLAP engine als onderdeel van de database.

Voor dimensies kan worden gekozen voor volledige sterschema- of snowflake-versies

Deze is afkomstig van het oude Express maar is aanzienlijk in functionaliteit opgepoetst en dient nu ter invulling van MOLAP door middel van zogenaamde analytische workspaces. Oracle Warehouse Builder gebruikers kunnen nu voor de implementatie van een dimensioneel model kiezen voor implementatie als ROLAP (relationeel) of als MOLAP (Analytic Workspace in combinatie met de OLAP catalog). Voor dimensies kan dan ook nog worden gekozen voor volledige sterschema- of snowflake-versies, waarbij per gedefinieerd hiërarchisch niveau een aparte tabel wordt geïmplementeerd. De diverse wizards helpen bij het definiëren van de specifieke OLAP-configuratie zoals pre-aggregatie en parsity.

Doordat er vanuit één repository zowel een relationeel als een OLAP-model kan worden geïmplementeerd, geeft dit de mogelijkheid om datgene te implementeren wat optimale performance biedt voor de eindgebruiker. In de referentie-case is besloten om het datawarehouse als relationeel stermodel en een hiervan afgeleid aggregaat als datamart in OLAP te implementeren. Men moet zich wel realiseren dat de dimensies wel in beide contexten moeten worden gedefinieerd. Dit is ook logisch, omdat een OLAP-implementatie bijvoorbeeld niet goed om kan gaan met een slowly changing dimensie, zonder hier extra aandacht aan te besteden. Vaak wordt in de praktijk de datamart voorzien van de *as is* situatie voor de dimensies. Ook moet men opletten bij het laden van OLAP-kubussen, omdat standaard de laatste waarde van een feit voor een bepaalde situatie van dimensies wordt geladen. Terwijl men bijvoorbeeld de aggregatie van alle losse feiten voor een dimensie-situatie wil hebben.



Afbeelding 5: Voorbeeld van cube laadstroom.

Experts

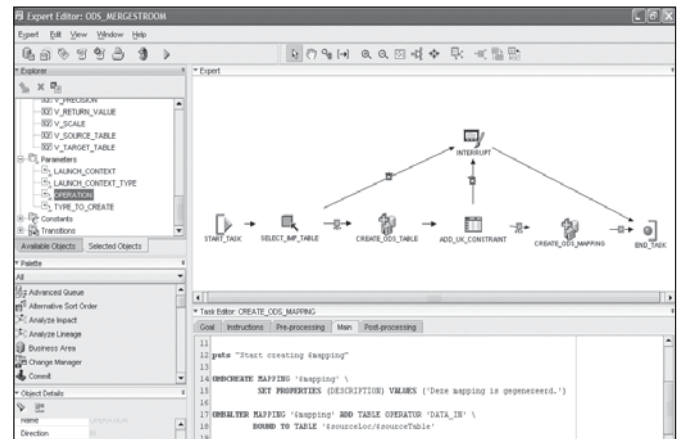
Tijdens de ontwikkeling van ETL-processen worden er vaak gelijksoortige oplossingen bedacht. Toch blijkt in de praktijk dat men vaak dezelfde functionaliteit (lees transformaties) in meerdere mappings uitvoert en modelleert. Aan de ene kant kan men nu in OWB gebruik maken van zogenaamde pluggable mappings. Hierin wordt via een input/output signature een herbruikbare mapping-component gerealiseerd. Deze is dan vanuit meerdere mappings oproepbaar en levert een beter alternatief voor functies/procedures zoals dit bij de huidige versie vaak wordt opgelost.

Aan de andere kant is het nu mogelijk om gebruik te maken van zogenaamde Experts. Dit zijn programma's gebaseerd op de OMB scripting-taal. Ten opzichte van de huidige versie van OMB is het nu mogelijk om alle componenten van de GUI aan te roepen wat een interactieve dialog tijdens het uitvoeren van een expert mogelijk maakt. Door ook nog eens de programma-flow inzichtelijk te maken via een grafische schil wordt het wel heel intuïtief om een expert te ontwerpen en uit te voeren. Op een gelijksoortige wijze als de proces-flow editor wordt een expert in elkaar gezet. Men definieert processtappen die via variabelen en parameters informatie aan elkaar doorgeven en/of waarbij GUI-componenten worden aangeroepen. Iedere zich in de flow bevindende component kan worden voorzien van een pre- en/of post-procedure (in OMB). Het uitvoeren van een expert wordt zelfs via een grafische presentatie gevolgd, zodat de eindgebruiker weet waar de expert zich bevindt tijdens het uitvoeren van zijn opdracht.

In de praktijk blijkt dat men vaak dezelfde functionaliteit in meerdere mappings modelleert

Experts kunnen met en zonder visuele ondersteuning worden opgestart en toegevoegd aan de contextgevoelige menu's op meerdere niveaus van OWB. Ook kunnen ze apart van OWB worden opgestart hetgeen mogelijkheden geeft om functionaliteit aan eindgebruikers beschikbaar te stellen. Binnen het voor de test gebruikte BI framework is een aantal productiviteitsboosters gedefinieerd, waarmee ontwikkelaars van OWB-gebaseerde oplossingen sneller eenvoudige mappings kunnen realiseren. Als voorbeeld kan men in afbeelding 6 zien hoe een expert eruit ziet, die als doel heeft om vanuit een bestaande importtabel een ODS tabel af te leiden te creëren en de benodigde laadstroom te realiseren.

Bij het starten van de Expert (via een contextmenu binnen de ODS module) wordt er aan de gebruiker gevraagd van welke tabel er een ODS mapping moet worden gemaakt. Hierbij wordt er een grafische component van OWB opgeroepen (OBJECT SELECTOR). Hierin wordt een bestaande importtabel geselecte-



Afbeelding 6: Voorbeeld Expert voor creatie van ODS laadstroom.

teerd en de Expert creëert de metadata voor een nieuwe ODS tabel, voegt daar enige attributen aan toe en creëert vervolgens een volledige onderhouds-mapping (MERGE) voor de nieuwe ODS tabel. Het voert hier te ver om alle mogelijke toepassingen op te sommen, maar een aantal willen we u niet onthouden. Flatfile import; Expert voor het samplen en inlezen van een extern bestand en integreren hiervan in de datawarehouse-omgeving. De gebruiker is hierbij zelf in staat om externe gegevens toe te voegen aan het datawarehouse zonder tussenkomst van de IT-afdeling.

Kubusgeneratie; vanuit bestaande OLAP-kubussen genereren van afgeleide kubussen voor speciale doeleinden. Denk hierbij aan tijdelijke kubussen voor bijvoorbeeld simulaties/forecasting. Deze kunnen dan gebaseerd zijn op al bestaande kubussen waar de dimensies geladen zijn. De Expert genereert dan de benodigde structuur, dan een deployment naar de OLAP catalog en genereert source code voor het uitvoeren van de benodigde OLAP dml (aparte taal voor Analytic Workspace).

Conclusie

Naast volledig veranderde GUI en de applicatiestructuur, is er door Oracle enorm veel functionaliteit aan het product toegevoegd. In dit artikel zijn niet alle veranderingen aan bod gekomen maar zijn alleen de in het oog springende zaken behandeld. Zo zijn er mogelijkheden gekomen om niet alleen naar Oracle zelf te schrijven maar ook naar andere databases.

In ieder geval getuigt Oracle van een visie over data-integratie en komt dit tot uiting in de ingezette ontwikkeling van de tool. Een aantal oorspronkelijk functionaliteiten is gedurende de bèta-fase geschrapt, maar zal bij volgende releases snel worden opgenomen in de basisfunctionaliteit van het product. Een voorbeeld hiervan is bijvoorbeeld Datamining.

Bertram Hof

Ing. A.B. Hof (bertram.hof@logicacmg.com) is werkzaam bij LogicaCMG als Senior BI-consultant en lid van de klankbordgroep BICC (Business Intelligence Competence Center).