

# Java op het Oracle 9i platform

## Brede ondersteuning MVC-architectuur

*In dit artikel zal Erwin Groenendal ingaan op de ondersteuning van Java op het Oracle 9i platform. Naast de Oracle 9i database bestaat het dit platform uit de Oracle 9i Application Server en de Oracle 9i Development Suite. Het Oracle 9i platform ondersteunt de ontwikkeling, het beschikbaar stellen en het beheren van Java toepassingen. De focus in dit artikel ligt op de database, application server en JDeveloper (onderdeel van de Oracle 9i Development Suite). De auteur beschrijft de Java-ondersteuning in de database, op de application server en in JDeveloper.*

Met Oracle8i introduceerde Oracle "Java in de database". De ondersteuning van Java in de database zelf bestond aanvankelijk uit een Java Virtual Machine (JVM), Java Stored Procedures (waarvoor om verwarring met JavaServer Pages te voorkomen geen acroniem wordt gehanteerd) en J2EE ondersteuning voor o.a. Java Servlets, JavaServer Pages (JSP) en Enterprise Java Beans (EJB). In de meest recente release van de Oracle database, Oracle9i Release 2, is de ondersteuning voor Java in de database echter in belangrijke mate teruggebracht.

### OC4J

Het beschikbaar stellen van Java Servlets, JSP's en EJB's wordt niet langer ondersteund vanaf deze release. Dit is opmerkelijk te noemen omdat Oracle nog niet zo lang geleden met de kreet "300% Java" aangaf dat de database server dé plek was voor het uitvoeren van (alle) Java code. Oracle kende echter grote problemen met de performance van de JVM in de data-

base zelf. Met het in licentie nemen van de Orion application server van IronFlare beschikt Oracle inmiddels over uitstekende en snelle J2EE containers die onder de naam Oracle Containers for J2EE (OC4J) zijn geïntegreerd met de Oracle9i Application Server. Klanten worden dan ook geadviseerd om hun J2EE toepassingen in de database te migreren naar OC4J.

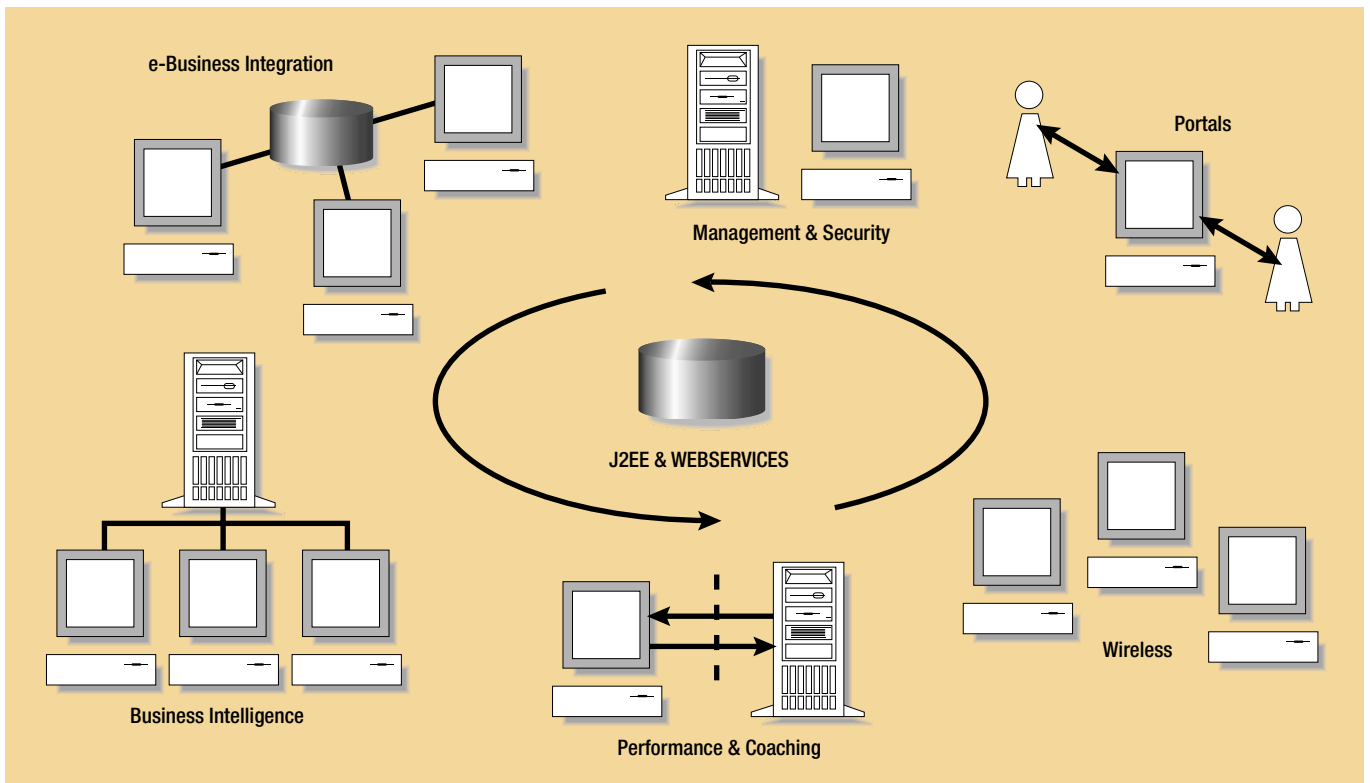
Java Stored Procedures behelst het runnen van Java code in de database. Ontwikkelaars kunnen hierbij gebruik maken van de meeste J2SE standaard classes. Eigen classes worden aangemaakt in de database en de aanroep vanuit PL/SQL is mogelijk via wrappers. Oracle ontwikkelaars maken dankbaar gebruik van de vele mogelijkheden die hiermee verkregen worden. Voor het uitvoeren van acties buiten de database waren ontwikkelaars namelijk altijd aangewezen op ingebouwde (systeem) database packages of call outs naar 3GL programmatuur (external procedures). Nu kan, door gebruik te maken van de vele J2SE library's, bijvoorbeeld eenvoudig een file worden gemanipuleerd, communicatie over het netwerk worden uitgevoerd of een e-mail worden verstuurd. Voor code waarin intensieve data manipulatie plaatsvindt verdient PL/SQL echter nog steeds de voorkeur. De PL/SQL code is sneller en minder complex door de volledige integratie van SQL in PL/SQL.

Uiteraard ondersteunt Oracle JDBC. Versie 2.0 wordt volledig ondersteund en versie 3.0 in belangrijke mate. Daarnaast implementeert Oracle Java API's voor bestaande functionaliteit in de database. Bijvoorbeeld de Java Messaging Service (JMS) API voor Oracle Advanced Queuing (AQ) en de Java Advanced Imaging (JAI) API voor Oracle InterMedia.

***Sinds 9i release 2 streeft Oracle niet meer naar volledige J2EE ondersteuning in de database***

### Java op de applicatieserver

De eerste schreden van Oracle op het gebied van applicatieservers, met de Oracle Web Application Server (OWAS) en de Oracle Application Server (OAS), waren niet succesvol. Instabiliteit, beperkte ondersteuning van standaards en slechte performance waren er debet aan dat vele Oracle database klanten overstapten op applicatieservers van BEA en Apache.



Afbeelding 1. De brede functionaliteit, J2EE 1.3 certificering en de kleine foot print pleiten voor gebruik van Oracle 9iAS

De tweede generatie applicatieservers van Oracle, de Oracle Internet Application Server (IAS), werd gebaseerd op de populaire open source software van Apache. Oracle leverde met deze versie de HTTP server van Apache en een aantal modules van Apache, waaronder de JServ module voor het runnen van Java Servlets. Naast de standaard modules van Apache bevatte IAS een aantal eigen modules, waarvan de belangrijkste de MOD\_PLSQL module is waarmee PL/SQL gebruikt kan worden voor het ontwikkelen van web applicaties.

Met de MOD\_PLSQL module kunnen PL/SQL procedures in de database gebruikt worden om op een eenvoudige en productieve manier dynamische HTML pagina's te genereren. Deze procedures worden soms wel PL/SQL Servlets genoemd, en er bestaat zelfs ook een PL/SQL equivalent van JavaServer Pages:

***Inmiddels geeft Oracle toe dat het tot twee keer toe verkeerde keuzes heeft gemaakt met betrekking tot de applicatieserver***

PL/SQL Server Pages (PSP). Voor veel Oracle ontwikkelaars was de MOD\_PLSQL module een gemakkelijke en zeer toegankelijke technologie om hun eerste webapplicaties te maken. Er zijn dan ook veel toepassingen mee ontwikkeld. In Oracle Designer zit zelfs al heel lang een generator voor dit soort toepassingen. Met deze Web Server Generator (WSG) kunnen PL/SQL web applicaties gegenereerd worden op basis van dezelfde moduledefinities die gelden voor Oracle Forms toepassingen. De vele Oracle programmeurs die op deze manier kennis maakten met het ontwikkelen van web applicaties realiseerden zich ook dat een HTML user interface (en de inherente beperkingen van de browser/server architectuur) veel beperkingen heeft ten opzichte van de grafische user interface in Oracle Forms toepassingen. Deze beperkingen komen goed tot uiting als men kijkt naar het aantal pagina's en het aantal keren dat een pagina "ververst" moet worden voor het uitvoeren van een eenvoudige functie in een applicatie zoals het zoeken en wijzigen van een aantal gegevens.

De IAS application server werd gecompleteerd met een reeks aan producten die (in verbeterde vorm) ook in de meest recente versie van de application server opgenomen zijn (zoals Oracle Portal voor het ontwikkelen van portalen). IAS was een enorme verbetering ten opzichte van OAS. Maar de ondersteuning van J2EE was nog onvoldoende. Zo ondersteunde de JServ module niet de laatste versie van de Java Servlets API.

## Nieuwe generatie

Voor de derde, en huidige, generatie van de application server nam Oracle de uitstekende J2EE containers van Orion over (en noemde deze OC4J: Oracle Containers for J2EE). De nieuwste versie van de Oracle9i Application Server (9iAS), zoals de derde generatie heet, is J2EE 1.3 gecertificeerd en voldoet dus volledig aan de meest recente J2EE standaard. De Apache HTTP server wordt nog steeds gebruikt voor web server functies zoals security en het beschikbaar stellen van statische HTML pagina's.

Net als bij de database (met de ondersteuning van JMS en JAI) is er ook bij de application server een duidelijke beweging te zien richting de ondersteuning van open Java API's. Zo is er voor Oracle's implementatie van LDAP (Lightweight Directory Access Protocol), de Oracle Internet Directory, een Java Naming Directory Interface (JNDI) API beschikbaar. Daarnaast zal in de nieuwste versie van 9iAS de functionaliteit voor het integreren van systemen (het onderdeel Oracle9iAS Integration) gebaseerd zijn op de Java Connectivity Architecture (JCA).

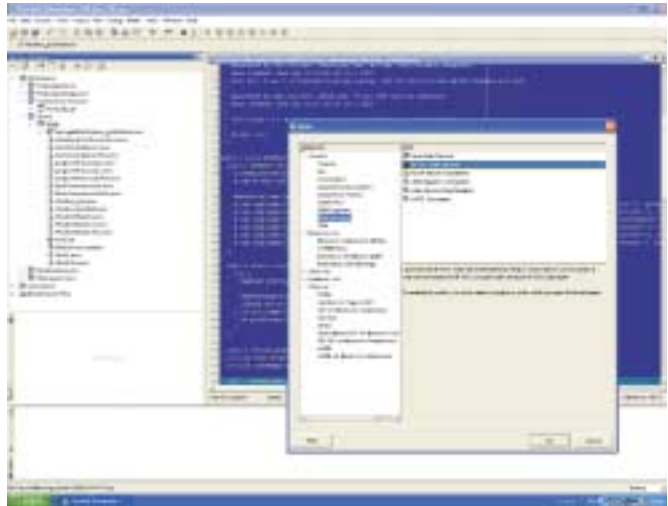
In presentaties geeft Oracle toe dat het tot twee keer toe een verkeerde keuze heeft gemaakt met betrekking tot de application server. De eerste keer door met OAS zelf een application server (proberen) te bouwen. De tweede keer door, zoals hierboven beschreven, application server functionaliteit in de database te willen stoppen. Echter met de combinatie van de Apache HTTP Server, OC4J en aanvullende producten is Oracle er in geslaagd een zeer goede application server op de markt te brengen die de concurrentie aankan met bedrijven als IBM en BEA. IBM heeft BEA inmiddels ingehaald in marktaandeel. Oracle is haar grote achterstand hard aan het inlopen en is de nummer drie. Argumenten die voor een keuze voor Oracle9iAS pleiten zijn de brede functionaliteit (zie afbeelding 1), J2EE 1.3 certificering, de prijs of de kleine foot print (OC4J is slechts enkele Mb's groot ten opzicht van IBM met meer dan 100 Mb). Voor veel Oracle gebruikers die aanvankelijk voor een andere application server kozen is 9iAS een aantrekkelijk alternatief. Maar ook voor niet-Oracle gebruikers is 9iAS een zeer serieuze optie. Het is precies deze groep die Oracle graag wil bereiken en met Oracle9iAS JDeveloper beschikt Oracle over nog een belangrijke troef.

## Java ontwikkeling met JDeveloper

Oracle heeft veel ervaring met het maken van ontwikkeltools. Producten zoals de CASE tool Oracle Designer en de 4GL ontwikkeltool Oracle Developer (bestaande uit onder andere Oracle Forms voor de ontwikkeling van schermen en Oracle Reports voor het maken van rapporten) zijn zeer succesvol. In combinatie met de bijbehorende ontwikkelmethode Custom Development Method (CDM) beschikken Oracle gebruikers over krachtige en productieve middelen voor het ontwikkelen van enterprise applicaties. In het begin waren dit terminal (server)

applicaties (SQL\*Forms 3.0), later client-server applicaties (vanaf Oracle Forms 4.0) en tenslotte (met de Oracle Forms Server, onderdeel van 9iAS) webapplicaties. Deze tools hebben een sterke aanhang in Nederland. De technologie is echter Oracle-eigen. In lijn met Oracle's strategie om open standaarden te ondersteunen, en Java en XML in het bijzonder, biedt Oracle sinds enige jaren JDeveloper (zie afbeelding 2) aan als ontwikkeltool voor Java toepassingen.

Voor de ontwikkeling van JDeveloper heeft Oracle de compiler technologie van Borland's JBuilder in licentie genomen. De oorspronkelijke Microsoft Windows user interface is inmiddels helemaal herschreven in Java waardoor JDeveloper op vele platforms, waaronder Sun en Linux, te gebruiken is. JDeveloper ondersteunt de ontwikkeling van J2EE applicaties en web services



Afbeelding 2. Het tool JDeveloper is op vele platforms -waaronder Sun en Linux- te gebruiken

en de deployment van programmatuur naar de Oracle9iAS Application Server en andere application servers. De belangrijkste J2EE standaarden voor JDeveloper zijn Java Servlets, JavaServer Pages (JSP) en Enterprise Java Beans (EJB). In JDeveloper worden ook twee Unified Modelling Language (UML) modellen ondersteund: class diagrams en activity diagrams.

## Doelgroep

Met JDeveloper richt Oracle zich op twee groepen. Enerzijds de huidige gebruikers van de CASE en 4GL ontwikkeltools die willen overstappen naar Java en anderzijds Java ontwikkelaars. De tweede groep is met name interessant voor Oracle omdat dit een grotendeels nieuwe groep voor Oracle is die nog niet de Oracle applicatieserver of Oracle database gebruiken. Voor deze groep zal Oracle zoveel mogelijk standaarden en binnen de Java community populaire raamwerken ondersteunen. Voor de eerste groep, de gebruikers van de "oude" Oracle tools,

wordt gestreefd naar aansluiting op de declaratieve manier van werken in deze tools door veelvuldig gebruik te maken van wizards en property sheets.

Veel gebruikers van de “oude tools” voelen zich gedwongen om naar Java over te stappen en Oracle Designer en Oracle Developer in te ruilen voor JDeveloper. Dit komt met name omdat Oracle onduidelijk is geweest over de toekomst van deze tools. Veel gebruikers zijn in de veronderstelling dat deze tools over een paar jaar niet meer ondersteund worden. Dit is echter geenszins het geval. Alleen de ondersteuning van het ontwikkelen van client/server applicaties zal (pas) in 2006 worden gestopt (met de mogelijkheid voor individuele organisaties om de ondersteuning te verlengen tot 2008) zoals in een officieel statement of direction is gesteld. Er is geen einde aangekondigd van de ondersteuning van deze tools voor de ontwikkeling van web applicaties, dat wil zeggen voor web deployment van Oracle Forms- en Oracle Reports toepassingen. Gebruikers hebben dus vier tot zes jaar om te migreren van client/server naar web deployment en hoeven zich geen zorgen te maken met betrekking tot het beëindigen van de ondersteuning van



Afbeelding 3. Een UIX pagina in de editor

deze tools zelf. De “lichte paniek” binnen deze groep gebruikers is dus ongegrond. Wel lijkt het verstandig om de Java trend niet te negeren en te kijken naar JDeveloper als tool voor het ontwikkelen van Java applicaties.

### Geduldige ontwikkelaars

Oracle heeft zich in haar marketing uitingen vooral gericht op de nieuwe groep Java ontwikkelaars. Ontwikkelaars zouden het Oracle kwalijk kunnen nemen dat die er blijkbaar vanuit gaat dat de Oracle ontwikkelaars “geduldiger” zijn dan hun Java-collega’s. Het lijkt er namelijk op dat het ondersteunen van populaire Java-raamwerken en standaarden belangrijker wordt geacht dan

het bieden van een gemakkelijk migratiepad van Oracle Designer en Developer naar Java. Maar de betreffende ontwikkelaars gaan met deze veronderstelling echter te gemakkelijk voorbij aan een tweetal punten, nog afgezien van het feit dat het natuurlijk cruciaal is dat een Java ontwikkeltool als JDeveloper volledige ondersteuning biedt voor de meest actuele Java standaarden.

Ten eerste is er sprake van een aantal beperkingen van de HTML user interface (en de inherente beperkingen van de browser/server architectuur) ten opzichte van de grafische user interface in Oracle Forms applicaties. Gezien deze beperkingen moet men zich afvragen of een toepassing die nu een grafische user interface heeft wel kan voldoen aan de eisen van de

***Voor de huidige generatie applicatieservers nam Oracle de uitstekende J2EE containers van Orion over***

gebruikers met een HTML user interface. Voor een deel van de bestaande toepassingen (en nieuwe toepassingen) kan dit, maar voor een belangrijk deel niet. Bij veel toepassingen zal wel een gedeelte van de applicatie voorzien kunnen worden van een HTML user interface. Voor veel intensief gebruikte en complexe functies zijn gebruikers echter beter af met een grafische user interface. Er kan dus gesteld worden dat Java applicaties met een HTML user interface simpelweg niet iedere bestaande Oracle Forms applicaties met een grafische user interface kunnen vervangen. Nu is het niet zo dat met Java alleen maar applicaties met een HTML user interface ontwikkeld kunnen worden. Met de Java Swing classes kunnen ook grafische user interfaces gemaakt worden. Dergelijke applicaties zijn echter “fat client” toepassingen (waarbij de software op de client geïnstalleerd moet worden) of maken gebruik van de logge en achterhaalde Java Applets technologie (waarbij de software telkens gedownload wordt). “Fat client” Java applicaties worden eigenlijk alleen gebruikt voor grote en (zeer) complexe programma’s zoals ontwikkeltools (bijvoorbeeld JDeveloper zelf).

Ten tweede is het zo dat het werken met de objectgeoriënteerde 3GL taal Java heel wat anders is dan de 4GL “taal” Oracle Developer. Ook bij het gebruik van raamwerken (wizards, generators en property sheets ten spijt) krijgt een ontwikkelaar vrijwel altijd te maken met programmeren in Java. Nu kunnen Oracle ontwikkelaars zeggen dat zij met het programmeren in PL/SQL ervaring hebben met het programmeren in een 3GL

taal. Maar het programmeren van sterk data-gerelateerde (aanvullende) code voor bijvoorbeeld de implementatie van business rules is heel wat anders dan het programmeren van de fundamentele werking van een applicatie. Dit laatste is te vergelijken met het programmeren van de runtime engine van Oracle Forms zelf. Hierbij zal een programmeur al snel met smart terugdenken aan alle dingen die Oracle Forms voor de ontwikkelaar regelde. Het is dus maar zeer de vraag of de gemiddelde Oracle ontwikkelaar de stap naar Java kan maken.

Het is dus helemaal zo gek nog niet om geduld te veronderstellen en te verwachten bij de gebruikers van Oracle Designer en Oracle Developer. Pas indien de technologie zich verder heeft ontwikkeld kan een goed migratiepad geboden worden. Hierbij zijn raamwerken essentieel.

### Ondersteuning van raamwerken

Ontwikkelaars van Java applicaties zijn in de door hen ontwikkelde software, patronen of terugkerende structuren (patterns) gaan ontdekken. Een belangrijk overkoepelend pattern is het Model-View-Controller (MVC) pattern. MVC wordt algemeen beschouwd als de beste applicatiearchitectuur voor J2EE toepassingen. In deze architectuur zijn drie onderdelen van de applicatie min of meer onafhankelijk van elkaar: een deel dat zorgt voor de user interfaces (View), een deel dat de interactie met de database (persistence layer) voor zijn rekening neemt (Model) en een deel dat de aansturing van de view- en model delen verzorgt (Controller). In de ideale MVC architectuur zijn de drie delen volledig onafhankelijk van elkaar en kan bijvoorbeeld het model vervangen worden zonder dat de andere twee delen hoeven worden aangepast. In veel gevallen zijn de drie delen weliswaar netjes gescheiden van elkaar maar niet (geheel) onafhankelijk. Binnen JDeveloper kunnen ontwikkelaars kiezen uit verschillende technologieën voor de drie MVC onderdelen.



Afbeelding 4. Een UIX pagina in de previewer

## Het werken met de object-georiënteerde 3GL taal Java is heel wat anders dan de 4GL "taal" Oracle Developer

### Model

Voor het model deel is JDBC de voor de hand liggende keuze. Hierbij onderschatten ontwikkelaars echter vaak de complexiteit en hoeveelheid werk die erbij komt kijken om de interactie met de (relationele) database te implementeren. In deze interactie zijn vele patterns onderkend die in de regel geïmplementeerd moeten worden. Bij het gebruik van JDBC zullen deze allemaal door de ontwikkelaar zelf moeten worden geprogrammeerd. Om de complexiteit af te schermen en het werk te verminderen biedt JDeveloper twee (elkaar aanvullende) technologieën: TopLink en Business Components for Java (BC4J).

Oracle heeft het product TopLink in juni 2002 overgenomen van WebGain (een *venture capitalist* en eigendom van BEA). Bij de acquisitie door Oracle zijn development, support en consulting afdelingen van WebGain inbegrepen. TopLink, inmiddels omgedoopt in Oracle9iAS TopLink, heeft een behoorlijk groot marktaandeel en Oracle verkrijgt hiermee toegang tot een groot aantal klanten. Onder de huidige gebruikers van TopLink zijn veel grote namen zoals Sprint, 3M en BMW. Het product TopLink zal als separaat product blijven bestaan en klanten hoeven dus niet te migreren naar een ander (Oracle) product. TopLink sluit vooral goed aan op een top-down manier van werken waarbij een zuiver objectgeoriënteerd (OO) class model wordt gemaakt van de objecten (gegevens) waarmee de applicatie werkt zonder dat hierbij al aan de opslag van deze gegevens in de database (persistence) wordt gedacht. Bij het ontwerpen en implementeren van de gegevensopslag biedt TopLink grote toegevoegde waarde met het definiëren van de (mogelijk zeer complexe) mapping tussen objecten en relationele tabellen en de generatie van Java code op basis van deze mapping. Voor ontwikkelaars die bottom-up werken en uitgaan van opslag van gegevens in een relationele database is de toegevoegde waarde van TopLink veel minder. Daarnaast implementeert TopLink niet de vele patterns in het model deel van een MVC applicatiearchitectuur. Voor deze groep ontwikkelaars biedt JDeveloper Business Components for Java (BC4J). Hoewel met BC4J ook vanuit een class model richting relationele opslag gewerkt kan worden, is BC4J vooral gericht op het verkrijgen van objecten op basis van relationele opslagstructuren. Bestaande database modellen kunnen via de BC4J componenten

## 26: Advertentie Oracle

## Oracle biedt met User Interface XML een interessant alternatief voor een HTML user interface

via Java objecten benaderd en gemanipuleerd worden. BC4J implementeert bovendien een groot aantal patterns. Dit betekent dat BC4J de aangewezen technologie is voor het model deel wanneer voor een bestaande database toepassing met Java een nieuwe user interface wordt ontwikkeld. Maar ook nieuwe database toepassingen zullen in de regel "gewoon" gebruik blijven maken van relationele opslagstructuren waarbij BC4J de beste keus is voor het model deel. Vanuit JDeveloper kunnen BC4J componenten gegenereerd worden op basis van objecten in de database. In een toekomstige versie van Oracle Designer zullen deze componenten ook gegenereerd kunnen worden op basis van het datamodel in de repository.

### View

Veel ontwikkelaars maken gebruik van JSP's voor de user interface. Een JSP is in feite een HTML pagina waarin binnen speciale tags Java code kan worden aangeroepen voor het genereren van dynamische content. Een JSP wordt door de application server omgezet in een Java Servlet. Het achterliggende idee van deze opzet is dat vormgevers de HTML ontwikkelen en Java programmeurs de code schrijven voor de dynamische content. In de praktijk bestaat de JSP echter vaak uit bijna alleen maar aanroepen van Java code. Reden hiervoor is dat door het aanroepen van dezelfde Java code voor het genereren van HTML, gemakkelijker een consistente user interface kan worden bereikt. De tol die hiervoor betaald wordt, zijn lastig te begrijpen en moeilijk te onderhouden JSP's. Oracle biedt met UIX (User Interface XML) een interessant alternatief voor het ontwikkelen van een HTML user interface. UIX is een XML taal waarmee op een hoog abstractie niveau de user interface (pagina layout) gedefinieerd wordt. Wanneer de pagina moet worden getoond, wordt op basis van de UIX pagina definitie de juiste HTML gegenereerd. Hiermee wordt een consistente user interface gegarandeerd. In de UIX definitie is tevens aangegeven welke data moeten worden opgenomen in de pagina. Het mechanisme waarmee deze data binding is aangegeven is onafhankelijk van de technologie die voor het model deel (van de MVC applicatiearchitectuur) wordt gebruikt. UIX is een technologie die door Oracle zelf is ontwikkeld en beproefd is binnen de standaard Oracle Applications. De technologie sluit echter aan op JavaServer Faces (JSF), een toekomstige J2EE standaard. UIX is bovendien volledig geïmplementeerd in

Java en past daarmee nu ook al prima binnen J2EE. UIX pagina definities worden ontwikkeld met behulp van de XML editor in JDeveloper. Hierbij wordt dankbaar gebruik gemaakt van de code completion en syntax checking features van JDeveloper. Een pagina kan in JDeveloper bekeken worden met de UIX previewer. In afbeeldingen 3 en 4 zijn voorbeelden te zien van de UIX pagina in de editor en previewer. In een toekomstige versie van JDeveloper zal een grafische editor geboden worden.

### Controller

Een populaire technologie voor het controller deel is Struts van Apache. JDeveloper ondersteunt Struts. Bij deze controller technologie wordt gebruik gemaakt van een XML file waarin staat aangegeven welke Java code class moet worden aangeroepen om bepaalde gebruikersinteractie af te handelen (waaronder aansturing van het model deel). Deze classes hebben een vaste signatuur (implementeren een bepaalde Java interface) en kunnen daardoor dynamisch door de controller worden aangeroepen. Na de aanroep van de class wordt een JSP aangeroepen om een nieuwe pagina te tonen aan de gebruiker (de aansturing van het view deel). Struts werkt eenvoudig, maar heeft daardoor ook zijn beperkingen. Herbruikbaarheid van de Java classes is gering, wat leidt tot nogal monolithische applicaties. Oracle heeft, waarschijnlijk, geïnspireerd door Struts, een veel krachtiger raamwerk ontwikkeld voor het controller deel van een MVC applicatiearchitectuur. Dit raamwerk is oorspronkelijk door de Consulting Services groep van Oracle ontwikkeld op een groot project. Het is daarna verder ontwikkeld en is nu als 'Oracle9iAS MVC framework for J2EE' onderdeel van Oracle9i Application Server. Herbruikbaarheid en productiviteit zijn de sterke punten. Voor een ontwikkelaar ligt het gebruik van dit raamwerk voor de hand. Handige Java ontwikkelaars kunnen Struts overigens prima combineren met het Oracle9iAS MVC Framework for J2EE, zoals bij JHeadstart (waarvan in Optimize nummer 6, 2001 reeds een artikel werd gewijd), en zodoende de voordelen van beide technologieën benutten.

### Conclusie

Java lijkt een definitieve en logische plaats te hebben ingenomen in de Oracle database: sterke JDBC ondersteuning, mogelijkheid om (J2SE) Java code uit te voeren en Java API's voor bepaalde database functionaliteit. De Java-ontwikkelaars wordt met name door JDeveloper zeer goed ondersteund in het ontwikkelen van J2EE applicaties en web services.

**Erwin Groenendal** is technisch directeur van Cumquat Information Technology. Cumquat richt zich op de toepassing van Oracle, Java, XML en Internet technologie en biedt oplossingen voor Internet en enterprise applicaties, web services, informatie portalen en business-to-business (B2B) en application-to-application (A2A) integratie.