

Oracle Map Viewer: gegevens in kaart

Visualisatie van geografische informatie

Het opnemen van een geografische kaart in een administratief systeem kan een belangrijke toegevoegde waarde bieden voor de eindgebruiker. Recentelijk heeft Oracle de technologie op de markt gebracht om dit beter te kunnen realiseren: Oracle Application Server MapViewer. In dit artikel schetst Jeroen van Schaijk de historie, positionering en grondslag van Oracle MapViewer. Aan de hand van een praktijkvoorbeeld geeft hij een 'routebeschrijving' voor het opnemen van een geografische kaart in Oracle webform, waarbij gebruik is gemaakt van de meest recente (10g-) versie van MapViewer.

Iedereen heeft wel eens gebruik gemaakt van een routeplanner op Internet: een routebeschrijving in combinatie met een geografische kaart. De toegevoegde waarde van een geografische kaart behoeft hier geen uitleg. Maar ook een geografische kaart in combinatie met een administratief systeem kan toegevoegde waarde bieden voor de eindgebruiker. Net als voor een routeplanning zijn we met een geografische kaart namelijk in staat het ruimtelijk inzicht over de administratieve gegevens te verbeteren om sneller en beter een goed begrip van de werkelijkheid te krijgen. En met een sneller en beter begrip van de

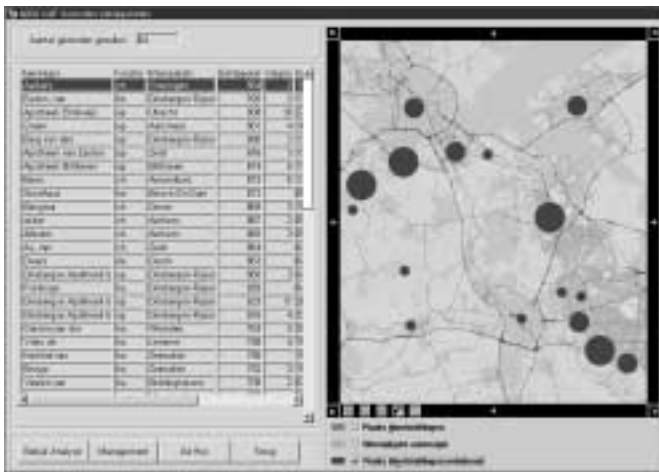
werkelijkheid kunnen we efficiënter en nauwkeuriger werken, niet onbelangrijk, communiceren. Recentelijk heeft Oracle de technologie op de markt gebracht om dit te kunnen realiseren; Oracle Application Server MapViewer.

Ruimtelijke gegevens

Voordat we administratieve objecten kunnen opnemen in een geografische kaart, zal het object naast inhoudelijke kenmerken ook ruimtelijke kenmerken moeten hebben. Hiermee zijn we in staat het object te relateren aan het aardoppervlak. Een dergelijk kenmerk of locatiebeschrijving zal in de meeste registraties al aanwezig zijn in de vorm van een adresbeschrijving (straat, woonplaats en postcode). Het vervolgens omzetten van een dergelijke adresbeschrijving naar een geografische locatie wordt geocoderen genoemd. Hiervoor zijn, afhankelijk van de wensen, verschillende conversiebestanden in omloop. Niets lijkt ons dus nog in de weg te staan bestaande registraties in kaart te brengen.

Voor het kunnen verwerken en analyseren van ruimtelijke gegevens heeft Oracle een aantal jaar geleden haar databaserver uitgereikt met domeinspecifieke opslagstructuren en functionaliteiten voor Geografische Informatie Systemen (GIS). Tot en met versie 8i was deze uitbreiding alleen beschikbaar als optie (Oracle Spatial) op de enterprise editie. Vanaf versie 9i is een groot gedeelte van deze uitbreiding ook als feature (Oracle Locator) beschikbaar in de standaard editie. Tot voor kort was voor het visualiseren van deze ruimtelijke gegevens, die opgeslagen zijn in een Oracle database, (kostbare) software benodigd van een derde partij. Maar hierin is verandering gekomen.

Recentelijk heeft Oracle een component, Oracle AS MapViewer, opgenomen in de applicatieserver om deze ruimtelijke gegevens in de database ook te kunnen visualiseren in de vorm van een geografische kaart. Zo'n component wordt een kaartserver genoemd. Vanwege de open architectuur van MapViewer kan een geografische kaart zonder veel ontwikkelinspanning eenvoudig worden opgenomen in een (bestaande) web- of desktop-



Afbeelding 1. Een geografische kaart binnen Oracle Forms

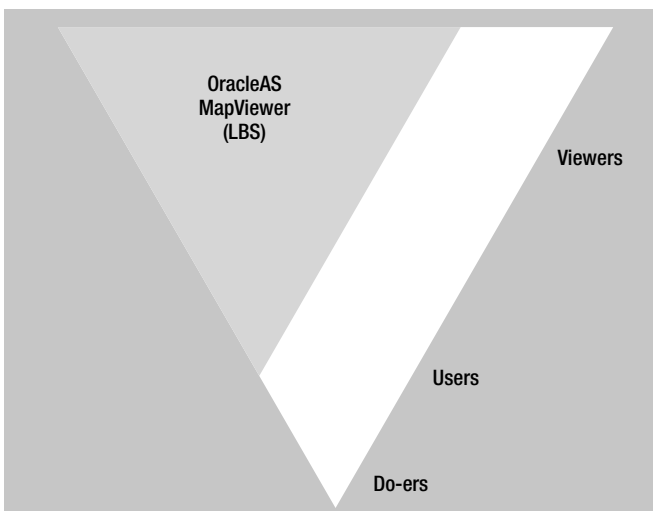
applicatie. Onlangs heeft IT-eye dit gedaan voor een Oracle webformsapplicatie.

Historie

Oracle Application Server MapViewer is het resultaat van een succesvolle ontwikkelingspanning voor de module Field Service (FS) uit de Oracle E-Business Suite (voorheen Oracle Applications). Voor het opnemen van een geografische kaart in FS heeft Oracle haar generieke TCA-datamodel uitgebreid met opslagstructuren van Oracle Spatial. Voor het visualiseren van de business objecten in een geografische kaart is in eerste instantie gebruik gemaakt van een kaartserver van een bekende softwareleverancier van GIS-technologie. Vanwege performanceproblemen heeft Oracle besloten deze kaartserver te vervangen. Hiervoor is in twee weken tijd een eigen kaartserver ontwikkeld, die succesvol is opgenomen in FS en kort daarop in de modules Transportation en Logistics, Product Exchange, Human Resources en Real Estate. Deze kaartserver is vervolgens doorontwikkeld en vanaf Oracle 9iAS release II beschikbaar gekomen in alle edities van de applicatieserver.

Positionering

Voor de positionering van Oracle MapViewer binnen de ruimtelijke informatievoorziening (RIV) delen we de gebruikersgroep van geografische toepassingen op naar de do-ers, de users en de viewers. De do-ers zijn de GIS-specialisten, de tekenaars. Zij creëren en onderhouden de gegevens met high-end GIS-applicaties. De users zijn de kenniswerkers. Zij integreren de gegevens en voeren complexe, ad-hoc query's uit op de ruimtelijke gegevens en zetten de gegevens klaar voor de viewers. De viewers vormen verreweg de grootste groep (meer dan tachtig procent). Deze groep omvat de "gewone" eindgebruikers. Zij consumeren de ruimtelijke informatie voor basisbewustzijn en eenvoudige analyses.



Afbeelding 2. Positionering Oracle MapViewer

Oracle MapViewer is een tool dat ingezet kan worden voor deze eindgebruikers. Oracle positioneert MapViewer zelf als bouwsteen voor het implementeren van Location Based Services (LBS). Dit zijn toepassingen die als vertrekpunt uitgaan van een geografische locatie en vallen binnen het toepassingsgebied van de viewers. De meest gebruikte toepassingen voor MapViewer is de plaats- en afstandbepaling ('waar ligt...?'), de geografische verdeling ('waar liggen de meeste...?') en de oriëntatie ('wat ligt er in de buurt van...?').

Grondslag

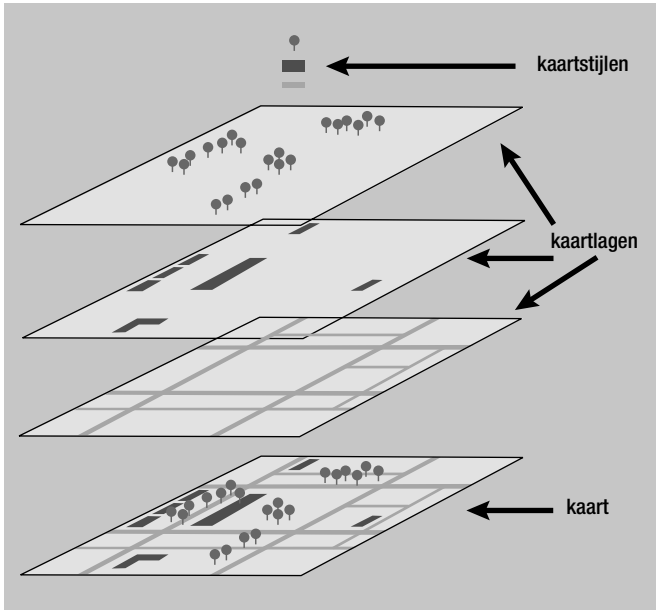
Oracle MapViewer is een eenvoudige en snelle kaartserver, die binnen een netwerk beschikbaar is als een service. De communicatie tussen een client en MapViewer vindt plaats via het vraag/antwoord-model op basis van XML over HTTP. In zekere zin kun je MapViewer beschouwen als een (document-style) web service.



Afbeelding 3. Een XML-kaartaanvraag

Dit betekent dat Oracle MapViewer ook kan worden (her)gebruikt in applicaties die draaien op een ander platform, zoals Microsoft dotNet. Alleen kan er dan geen gebruik worden gemaakt van de MapViewer Software Development Kit (SDK). De toolkit is namelijk alleen geschikt voor het Java-platform.

De kerntaak van MapViewer is het genereren van geografische kaarten op basis van een (kaart)aanvraag om vervolgens het resultaat in de vorm van een rasterkaart beschikbaar te stellen aan de aanvrager voor verdere verwerking. De hoeksteen voor het formuleren van een kaartaanvraag en het vervolgens genereren van een kaart is de metadata-repository. De MapViewer metadata-repository bestaat uit een drietal views waarin kaart-, thema- en stijldefinities kunnen worden vastgelegd.

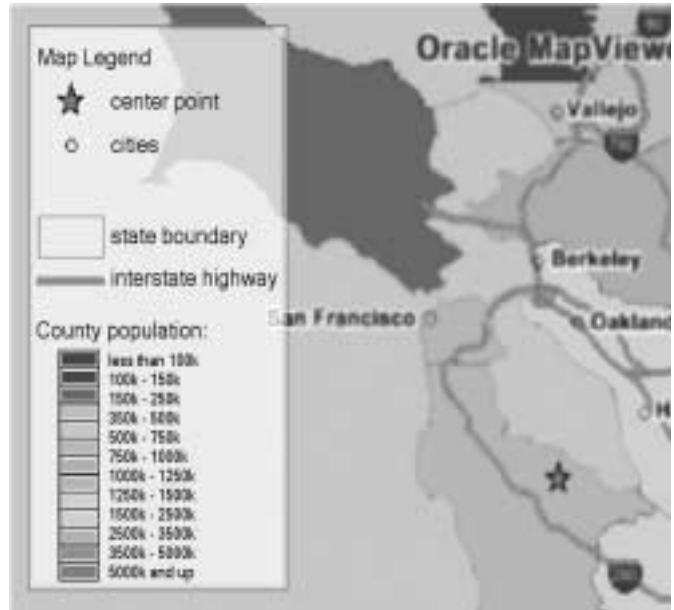


Afbeelding 4. Kaart, kaartlagen en kaartstijlen

- Een *kaartdefinitie* beschrijft uit welke thema's (kaartlagen) de kaart is opgebouwd en bij welke schaal zo'n thema moet worden getoond.
- Een *thema-definitie* beschrijft op welke data een thema is gebaseerd en hoe het opgemaakt moet worden door te verwijzen naar stijldefinities.
- De *stijldefinities* beschrijven de opmaak van een thema (zie afbeelding 4). Hierbij wordt onderscheid gemaakt tussen eenvoudige en complexe stijldefinities. Eenvoudige stijldefinities hebben betrekking op de symbologie (stijl, dikte en kleur), de labelkenmerken (font, grootte, regelafstand, et cetera) en de te gebruiken markers (symbolen) voor een thema. In het voorbeeld van een XML-aanvraag is bijvoorbeeld gebruik gemaakt van een ster-symbool als markering van een (bekende) locatie.

Complexe stijldefinities hebben betrekking op de visualisatie van een administratief 'gewicht', bijvoorbeeld de hoeveelheid neerslag, het gemiddelde inkomen, het aantal inwoners et cetera. Zo'n gewicht kan vervolgens worden gekoppeld aan eenvoudige stijldefinities, waardoor bijvoorbeeld woonplaatsen met een groot aantal inwoners door een rode stip kunnen worden weergegeven en woonplaatsen met een klein aantal inwoners door een zwarte stip.

Het is ook mogelijk een gewicht eerst te vertalen naar een visueel model, zoals een taartdiagram, een staafdiagram of een in grootte variërende marker, en vervolgens zo'n model af te beelden op de kaart. Door bijvoorbeeld gebruik te maken van een in grootte variërende marker, kan een woonplaats, afhankelijk van het aantal inwoners, door een grotere of kleinere stip worden weergegeven.



Afbeelding 5. Oracle MapViewer legenda

De betekenis van de stijldefinities die op de kaart van toepassing zijn, zoals kleuren, symbolen en lijnen, kunnen opgenomen worden in een legenda. Het toepassen van een legenda is optioneel maar kan de eindgebruiker op een snelle manier ondersteunen bij het interpreteren van het kaartbeeld. De

Pluggable Java Components (PJC)

Vanaf versie 6i is de interface van een Oracle Forms applicatie helemaal opgebouwd uit Java-componenten. De interface is zodanig opgezet dat er naast de bestaande interfacecomponenten (jargon: widgets) ook zelfgemaakte UI-componenten kunnen worden opgenomen.

Het is bijvoorbeeld mogelijk zelfgemaakte kalender- of GIS-componenten op te nemen in een bestaande form, of het gedrag van de standaard UI-componenten, zoals tekstvelden en buttons, te veranderen of uit te breiden (zie Oracle Forms demo op OTN voor code-voorbeelden). Ook is het mogelijk niet-visuele UI-componenten op te nemen voor een specifieke functie, bijvoorbeeld een Microsoft Office Word koppeling.

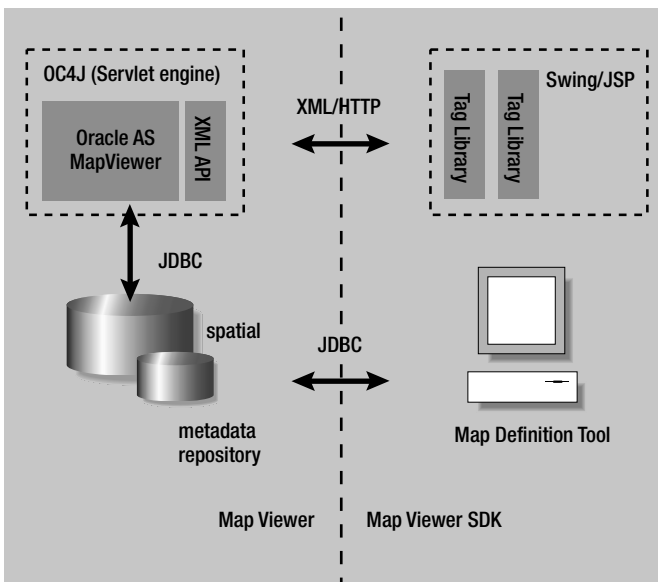
Voor het kunnen opnemen van een zelfgemaakte UI-component is Oracle Forms voorzien van een PJC-raamwerk. Voordat je een zelfgemaakte UI-component kunt opnemen in een form, ontwikkeld in Java, zal de code wel aan een aantal voorwaarden moeten voldoen. Dit zou je het PJC-programmeermodel kunnen noemen, dat goed beschreven is terug te vinden op OTN. Door deze (Java-)uitbreiding is Oracle Forms een open, uitbreidbaar en krachtige (ontwikkel)omgeving geworden.

inhoud, opmaak en indeling van een legenda maakt onderdeel uit van de kaartaanvraag. Door een paar extra regels toe te voegen aan de kaartaanvraag, kan een legenda zoals in afbeelding 5 eenvoudig worden toegevoegd aan het kaartbeeld.

MapViewer componenten

Oracle AS MapViewer bestaat uit een kaartserver, een metadata-repository en een SDK. De kaartserver-component is een Java-applicatie (servlet), die met behulp van de Oracle Enterprise Manager, of handmatig in een Oracle container (OC4J), ten uitvoer kan worden gebracht.

De metadata-repository is vanaf versie 9.2 een standaard onderdeel van een Oracle Locator/Spatial installatie. Voor vorige versies zal de metadata-repository handmatig moeten worden aangemaakt door een meegeleverd script uit te voeren. Alhoewel de kaartserver en de metadata-repository voldoende zijn om de kaartserver te kunnen gebruiken, kan de productiviteit aanzienlijk worden verhoogd door gebruik te maken van de SDK.



Afbeelding 6. Oracle MapViewer componenten

MapViewer SDK

De SDK bestaat uit een JavaBean, een JSP Tag Library en een Map Definition Tool (Java desktopapplicatie). Voor een ander platform dan Java is de SDK dus niet geschikt. Overigens plaatst Oracle deze technologieën niet onder de noemer SDK, terwijl deze technologieën wel de eigenschappen van een SDK hebben.

De JavaBean is een client-component, die rechtstreeks communiceert met de kaartserver. De JavaBean is een façade, waarmee alle kaartserver-functionaliteiten aan de ontwikkelaar beschikbaar worden gesteld. Door gebruik te maken van de

JavaBean is het opstellen van een XML-aanvraag door de gebruiker overbodig geworden en kan de aanvraag programma-tisch worden geformuleerd via het aanroepen van methoden en het zetten van eigenschappen van de bean. De JavaBean zal vervolgens de aanvraag (in XML) uitwisselen met de kaartserver. Het codefragment in afbeelding 7 resulteert in dezelfde kaart als de eerder getoonde XML-aanvraag in afbeelding 3.

```
// Oracle client MapViewer JavaBean
MapViewer = new MapViewer( "MY_ENDPOINT" );
// Define map content
_mapper.setMapTitle( "OracleAS MapViewer 10g" );
_mapper.setDataSourceName( "m" );
_mapper.setBaseMapName( "m" );
_mapper.setCenterAndSize( 122120, 442060, 2000 );
_mapper.addPointFeatures( 122120, 442060, "M.VYCAR" );
// Finally, request map
_mapper.run();
```

Afbeelding 7. JavaBean voorbeeld

De voordelen van het gebruik van de JavaBean-interface ten opzichte van de XML-interface, zijn de eenvoud, het onderhoud, de productiviteit en de ongevoeligheid voor protocolwijzigingen. Daarbij biedt de JavaBean voor de 'fat' client (Java Swing) nog extra functionaliteiten, die op het XML-niveau niet beschikbaar zijn, zoals "live features". Met live features kan een 'intelligent' kaartbeeld worden geconstrueerd. Dit is een kaartbeeld waarin (vector)objecten zijn opgenomen die direct door de eindgebruiker zijn te selecteren. Een 'intelligent' kaartbeeld is vaak wenselijk wanneer er meer interactiviteit met de kaart gewenst is.

Degenen die een geografische kaart willen opnemen in een HTML-applicatie, een 'thin' client, kunnen gebruik maken van de MapViewer JSP Tag Library. De Tag Library is een 'JSP-schil' om de JavaBean heen, waardoor de functionaliteiten van de JavaBean met de JSP-taal zijn aan te roepen. Tevens is er een uitbreiding voor JDeveloper beschikbaar om tijdens de ontwikkeling van de applicatie gebruik te kunnen maken van de MapViewer tags via de component palette. Net als de JavaBean zijn de genoemde voordelen ook van toepassing op de JSP Tag Library.

Tot slot, de Map Definition Tool. De Map Definition Tool is een desktop Java-applicatie, waarmee de gebruiker via een eenvoudige grafische interface de metadata-definities in de repository kan vastleggen, bekijken en onderhouden. Het voordeel van het gebruik van de definitietool boven het gebruik van SQL*Plus is, naast het visuele voordeel, de waarborging van de integriteit. De metadefinities worden namelijk opgeslagen als ongestructureerde data, waardoor bijvoorbeeld verwijzingen naar stijldefinities in themadefinities buiten het integriteitsmodel van de Oracle database vallen. De definitietool wordt overigens (nog) niet door Oracle ondersteund en is alleen beschikbaar via OTN.

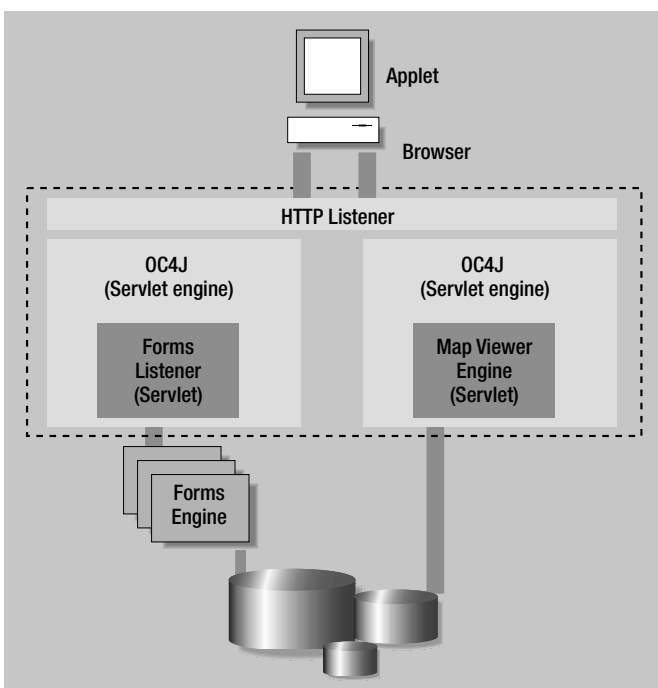
Uit het voorgaande mag duidelijk zijn dat Oracle MapViewer geen zelfstandige applicatie is. Het is een bouwsteen die zijn toegevoegde waarde pas heeft wanneer het binnen een applicatie wordt opgenomen.

Oracle Forms integratie

Hoewel een kaartbeeld op basis van Oracle MapViewer opgenomen kan worden in verschillende interfaces van Oracle applicaties, zoals Oracle Portal, Oracle Discoverer maar ook UIX, lijkt Oracle Forms vanwege de grote installed base een aantrekkelijke kandidaat. In dit laatste gedeelte van het artikel wil ik de stappen en randvoorwaarden presenteren die nodig zijn om deze integratie te realiseren. Deze stappen zal ik vanuit de (logische) architectuur belichten.

Zowel Oracle Forms als Oracle MapViewer hebben qua architectuur veel overeenkomsten. Beiden zijn binnen een netwerk als service (over HTTP) beschikbaar. Vandaar dat de runtime-omgeving van Oracle Forms, Forms Services wordt genoemd.

De uitvoering van een service wordt verzorgd door een engine, in dit geval een 'forms engine' en een 'map engine'. Vanuit de engines wordt een directe verbinding opgezet met een database. De client maakt dus (gelukkig) nooit rechtstreeks een verbinding met een database. De databases van beide engines kunnen verschillend zijn, maar meestal zullen de administratieve en de ruimtelijke gegevens uit dezelfde database ontsloten worden. Beide engines draaien binnen een webcontainer van een applicatieserver. Het is vanuit de licentiekosten aannemelijk,



Afbeelding 8. Integratie Oracle MapViewer en Oracle Forms

doch technisch niet vereist, dat beide engines binnen dezelfde applicatieserver draaien. Hierdoor vallen beide services ook binnen dezelfde beheeromgeving.

De daadwerkelijke integratie vindt pas plaats op de client. Hiervoor zal een UI-component ontwikkeld moeten worden volgens het PJC-programmeermodel (zie tekstkader). De UI-component is een schil om de MapViewer JavaBean heen, gebaseerd op de JFC/Swing componenten-architectuur. Hierbij kan het beste gebruik worden gemaakt van een Applet-panel vanwege de standaard ondersteuning van Applets in Oracle Forms. Het PJC-raamwerk past namelijk introspectie toe op applets, waardoor methodes en property's zonder ontwikkelinspanning benaderd kunnen worden vanuit Forms PL/SQL. Maar ook vanuit de herbruikbaarheid is een Applet-panel erg geschikt.

Tot slot zal voor de geografische kaart gebruik gemaakt worden van geografische data zoals bijvoorbeeld topografische kaarten, gemeente-, postcode- en wegennetkaarten. Hiervoor is een grote diversiteit aan 'kant-en-klare' bestanden beschikbaar. Maar let op: afhankelijk van het detailniveau kunnen de kosten wel behoorlijk oplopen!

Samenvatting

Oracle heeft met Oracle Application Server MapViewer een (commercieel) interessante uitbreiding aan de applicatieserver toegevoegd. Tot voor kort was men voor het visualiseren van ruimtelijke gegevens in een Oracle database aangewezen op een 'prijzige' GIS-leverancier. Met de introductie van Oracle MapViewer is hier verandering in gekomen. Oracle MapViewer wordt zonder meerprijs meegeleverd in alle edities van de applicatieserver. Degenen die reeds gebruik maken van Oracle Portal of Oracle Forms Services hebben Oracle MapViewer dus al in huis. Dat Oracle het serieus meent met Oracle MapViewer is evident. De lijst van nieuwe features in 10g, maar ook de integratie van MapViewer in de E-Business Suite, zijn daar duidelijke voorbeelden van. Oracle positioneert MapViewer als een bouwsteen voor het realiseren van Location Based Services (LBS). Dit betekent dat MapViewer vooral gericht is op de visualisatie van ruimtelijke gegevens en minder geschikt is voor interactieve en analytische kaarttoepassingen. Wat we op het gebied van kaartinteractiviteit nog wel kunnen verwachten is redlining; een techniek om op de kaart aantekeningen te kunnen maken. Kortom, Oracle MapViewer is een aantrekkelijk hulpmiddel om ons ruimtelijk inzicht (over onze administratieve gegevens) aanzienlijk te verbeteren.

ir. **Jeroen van Schaijk** is senior consultant bij IT-eye (e-mail: jeroen.van.schaijk@it-eye.nl).