

Het recyclen van Oracle Forms

Hergebruik in nieuwe webapplicaties mogelijk

*De historie van Oracle Forms, voorheen SQL*Forms, is bijna net zo lang als die van de Oracle database zelf. Veel organisaties die vandaag de dag nog gebruik maken van Oracle Forms, hebben al 10 of meer jaar ervaring met het product. Blijkbaar heeft Forms de tand des tijds dus prima doorstaan. Er zijn niet veel IT producten die een dergelijk lang leven hebben.*

De toekomst van Oracle Forms lijkt ook veiliggesteld. Er komt in ieder geval nog een versie 11 aan en Oracle heeft aangegeven Forms tot minimaal 2017, maar waarschijnlijk nog langer, te ondersteunen. Toch is er onrust bij sommige Forms gebruikers. Het product mag dan nog een lange toekomst voor de boeg hebben, veel van de moderne ontwikkelingen vinden plaats op andere vlakken. Oracle heeft duidelijk ingezet op JDeveloper en ADF voor applicatieontwikkeling en Forms naar een zijspoor verwezen.

Wat uiteindelijk de reden ook is, veel organisaties zijn op zoek naar een andere technologie voor hun toekomstige ontwikkelingen. Eén van de vragen is dan wat te doen met de bestaande Oracle Forms applicatie. Voor veel organisaties is dit zelfs een reden geweest nog niet te beginnen aan een overstap naar een andere technologie. Het is vaak onwenselijk of zelfs onacceptabel om gebruikers een deel van hun werk in de bestaande Forms applicatie te laten doen en een ander deel in een applicatie die gebouwd is in een andere technologie, zeker wanneer een gebruiker veelvuldig moet wisselen tussen deze twee delen.

Gevangen in Oracle Forms

Op korte of lange termijn zullen veel organisaties wel willen of moeten afstappen van Oracle Forms. Het overgaan naar een andere technologie is echter geen eenvoudige opgave. Het leren omgaan met de nieuwe technologie is vaak al een uitdaging op zich. Als hierbij ook nog het bestaande Forms systeem moet worden vervangen of gemigreerd naar de nieuwe technologie, dan haken veel organisaties af.

Een volledige migratie van een groot Forms systeem kost vaak

te veel tijd en geld en de risico's zijn onacceptabel groot. Toch is het jammer wanneer organisaties zich gedwongen voelen gebruik te blijven maken van Oracle Forms, ook voor nieuwe ontwikkelingen. Er is dan sprake van een technologie lock-in, die niet vanzelf zal verbeteren. Sterker nog, door Oracle Forms ook te blijven gebruiken voor nieuwe ontwikkelingen wordt dit probleem voor de toekomst alleen maar groter.

Wat zijn de opties?

In principe hebben Oracle Forms gebruikers een aantal opties voor de toekomst van hun systemen:

- Simpelweg niet overstappen. Dit klinkt misschien als een vreemde optie, maar niets doen kan een valide optie zijn. Zeker wanneer een systeem nog maar een beperkte levensduur heeft en er geen strategische redenen zijn om Forms te verlaten. Oracle zal Forms nog jaren ondersteunen, al zal het vinden van gekwalificeerd personeel wel steeds moeilijker worden.
- Volledige herbouw in een nieuwe technologie. Dit is wellicht de meest uitdagende optie voor ontwikkelaars, maar de applicatie moet wel erg slecht aansluiten bij de huidige wensen van de organisatie om deze radicale stap te rechtvaardigen.
- Migreren of converteren van de bestaande Forms applicatie naar de nieuw gekozen technologie. Dit kan (deels) automatisch gebeuren en lijkt daarmee een aantrekkelijke optie.
- Integreeren van de bestaande Forms applicatie met de nieuwe technologie. Er ontstaat een hybride applicatie die deels uit oude en deels uit nieuwe technologie bestaat. Voordeel is dat er minimale investeringen nodig zijn en dat er direct gebruik gemaakt kan worden van nieuwe technologie voor het verder uitbouwen van het systeem.

Converteren als ei van Columbus?

Het automatisch converteren van een Forms applicatie naar een andere technologie lijkt de ideale oplossing. Er zijn tools beschikbaar die een (vrijwel 100%) automatisch conversie beloven naar Microsoft .NET, open source Java, of Oracle ADF/Java.

Een volledig automatische conversie klinkt bijna te mooi om waar te zijn en dat is het eerlijk gezegd ook.

Bij een conversie is het van groot belang dat de uiteindelijke applicatie een goede architectuur heeft die aansluit bij de technologie waarnaar geconverteerd is. Dit maakt het mogelijk de geconverteerde applicatie ook in de toekomst te blijven onderhouden en verbeteren.

Vrijwel iedere moderne webapplicatie maakt gebruik van een geïntegreerde architectuur gebaseerd op componenten met een gescheiden verantwoordelijkheid en zo min mogelijk onderlinge afhankelijkheden. Het meest bekende patroon is het MVC (Model View Controller) model, waar duidelijk gescheiden delen van de applicatie verantwoordelijk zijn voor de opslag van gegevens, de presentatie en de besturing. Oracle Forms kent een dergelijke scheiding niet. In Forms wordt dit alles in één enkel Form gebouwd en het is dan ook lastig om twee zulke verschillende architecturen in elkaar te vertalen.

Een Oracle Forms applicatie bestaat allereerst uit alle metadata in de vorm van objecten en hun eigenschappen, zoals Data Blocks, Items, ListOfValues, etc. Voor deze gegevens is een automatische conversie in principe mogelijk. Doordat alles is vastgelegd in metadata is de intentie duidelijk en kan gekozen worden voor een goede implementatie van dezelfde functionaliteit in de nieuwe technologie.

Echter, over het algemeen bevat een Forms applicatie ook veel PL/SQL code. Dit is het deel wat voor de grote problemen kan zorgen. De PL/SQL code heeft veel meer vrijheden en variaties dan de vastgelegde metadata. Het is extreem moeilijk, zo niet onmogelijk, om de PL/SQL code geautomatiseerd te interpreteren om daaruit af te leiden wat de gewenste functionaliteit is. Doordat de intentie van een stuk PL/SQL code niet duidelijk is, kan ook niet gekozen worden voor een optimale implementatie in de nieuwe technologie. Het enige wat dan overblijft is te kiezen om PL/SQL niet te converteren of om te kiezen voor een rechttoe rechtaan conversie.

De meeste tools kiezen voor deze laatste aanpak, waarbij PL/SQL één op één wordt overgezet. Het voordeel voor de leverancier is dat deze een (bijna 100%) automatische conversie kan beloven, maar het is de vraag of dit voor de klant ook de beste optie is. Het leidt tot een applicatie met een zeer afwijkende architectuur waarin nog duidelijk de Forms oorsprong is te herkennen. Zo ontstaan er vaak Java methoden genaamd `whenNewFormInstance`, `goBlock` of `getItemProperty`. De uiteindelijke applicatie wijkt zo sterk af van wat gebruikelijk is in de Java of .NET wereld, dat er nog steeds veel Forms kennis nodig is om de applicaties te onderhouden en dat moeilijk gebruikt kan worden van de in de markt aanwezige .NET of Java kennis. Verder wordt er op deze manier niet of nauwelijks gebruik gemaakt van de mogelijkheden die de nieuwe technologie biedt. In feite is de hele oude Forms runtime engine in .NET of Java

code omgezet. Als dat een acceptabele oplossing is, dan is de vraag wat er mis was met het gewoon blijven gebruiken van Oracle Forms.

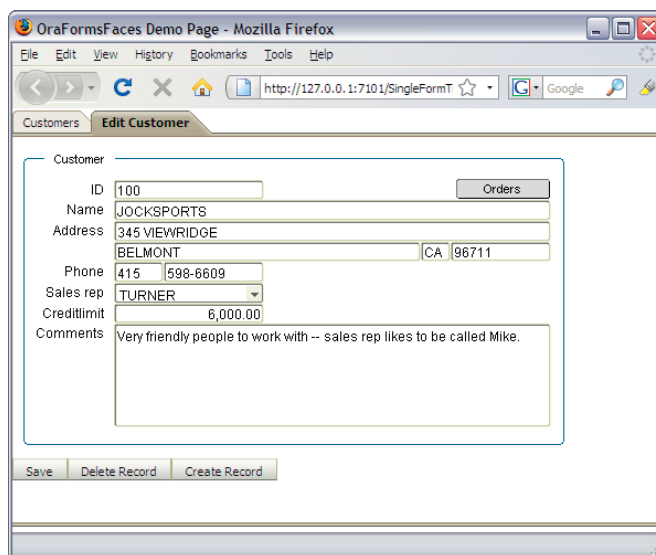
Integreren als oplossing

De enige oplossing die we nog niet besproken hebben is het naadloos integreren van de bestaande Forms in een JSF, ADF, .NET, Application Express of andere web applicatie. Dit zou veel voordelen opleveren en een aantal problemen uit de weg gaan. Het obstakel hier is dat Oracle Forms weliswaar al jaren als een applet in een webbrowser draait, maar in feite nog steeds een vrij gesloten client/server applicatie is. Idealiter zou je een Form willen opnemen als volwaardige component in een web applicatie, waarbij Forms en de gekozen webtechnologie eenvoudig gegevens en events kunnen uitwisselen. Je zou dan één hybride webapplicatie kunnen maken waarbij een gebruiker naadloos wisselt tussen de twee technologieën en in principe niet eens hoeft te merken wanneer welke technologie gebruikt wordt.

Dit klinkt als een ideale oplossing, maar is helaas niet mogelijk met het standaard Oracle Forms product. Toch zijn wij een aantal jaren geleden de uitdaging aangegaan om dit zelf mogelijk te maken. Daar is uiteindelijk het product OraFormsFaces uit voortgekomen, wat dit allemaal wel mogelijk maakt.

OraFormsFaces

Met OraFormsFaces is het mogelijk om een bestaand Form zonder of met slechts geringe aanpassingen te hergebruiken als volwaardig component in een webpagina. Omdat veel Forms gebruikers de overstap maken naar JSF of het daarop gebaseerde Oracle Application Development Framework (ADF), is er voor gekozen OraFormsFaces als een aantal JSF componenten te ontwikkelen. Met wat extra inspanning kan dezelfde technologie echter ook worden ingezet in een .NET, Application Express, of normale JSP omgeving.



De meest opvallende component is het Form component die het mogelijk maakt om de Forms applet op te nemen in een webpagina. Er zijn echter wel een aantal belangrijke verbeteringen. Zo is de Forms applet aangepast zodat dezelfde instantie van de applet op alle webpagina's kan worden hergebruikt. Dit is een belangrijk voordeel, omdat anders voor iedere webpagina een nieuwe applet moet worden gestart met de bekende vertraging van het opstarten van een Forms applicatie. Dat zou onacceptabel zijn voor de gebruiker.

Verder is er een JavaScript API toegevoegd aan Oracle Forms die het mogelijk maakt om vanuit de PL/SQL code in Forms JavaScript uit te voeren binnen de webpagina. Andersom is het mogelijk om vanuit JavaScript op de pagina een aanroep te doen naar een PL/SQL procedure in Oracle Forms. Deze JavaScript API is cruciaal voor de naadloze samenwerking tussen de Forms applet en de webapplicatie.

De Form component zorgt ook voor het formaat en de positie van de applet op de pagina en kan er daarbij voor zorgen dat bepaalde delen van de applet onzichtbaar zijn voor de gebruiker. In feite kan er een uitsnede van de applet worden gemaakt waarbij zaken als de statusbalk, het menu en/of de knoppenbalk kunnen worden verwijderd. Dit maakt het mogelijk om het Form visueel naadloos te integreren met de webpagina, zoals te zien op het screenshot. Voor de gebruiker vervaagt hierdoor het verschil tussen Forms en de rest van de webapplicatie. Voor ervaren gebruikers kan gekozen worden zaken als de knoppenbalk of andere delen toch te blijven gebruiken.

Parameters doorgeven

Om het Form daadwerkelijk een onderdeel uit te laten maken van de webapplicatie is het niet genoeg om alleen de applet op te nemen in een webpagina. Er is daarom een ondersteunend component voor het doorgeven van parameters tussen de webapplicatie en het Form.

Deze component vormt de verbinding tussen PL/SQL variabelen in de Forms server en variabelen op de webapplicatie-server. Beide programmeeromgevingen kunnen vanaf hun eigen server de waarden van deze component uitlezen of wijzigen. Dit maakt het mogelijk om zaken als het geselecteerde klantnummer van de webapplicatie door te sturen naar de Forms applicatie (en andersom). Hieronder een simpel voorbeeld waarbij het attribute "id" van een JSF bean is gekoppeld aan het FormParameter component.

```
<off:formParameter value="{bean.id}" id="param"/>
```

Vervolgens kan de waarde eenvoudig worden uitgelezen of gewijzigd vanuit Forms PL/SQL code:

```
v_param := offParams.getParamValue('param');
offParams.getParamValue('param', 1234);
```

Events uitwisselen

Naast het doorgeven van gegevens is het ook noodzakelijk dat de twee technologieën events kunnen uitwisselen. Als simpel voorbeeld kan er een normale button of link op een pagina staan die het Form zijn wijzigingen moet laten opslaan.

Andersom kan er een button in het Form zitten dat voorheen een ander Form aanriep, maar nu navigatie naar een andere webpagina moet initiëren.

Er is een tweede ondersteunend component beschikbaar om dit mogelijk te maken. Dit kan worden opgenomen in de JSF pagina en kan verwijzen naar normale JSF Actions of ActionListeners:

```
<off:formCommand action="goHome" id="home"/>
```

Vervolgens kan deze actie eenvoudig vanuit de Forms PL/SQL code worden geactiveerd. Hierdoor zal de webpagina worden gesubmit naar de server waarna afhandeling op de webserver plaatsvindt:

```
offInterface.execJSPCommand('home');
```

Andersom is het ook mogelijk om vanuit andere componenten op de pagina met JavaScript een event te sturen naar de Forms applet. Vervolgens wordt dit in PL/SQL code afgehandeld en kan de gewenste actie worden ondernomen:

```
<input type="submit"
onclick="sendEvent('frm','do_key','commit_form');"
/>
```

JHeadstart Application Generator

Met OraFormsFaces is de weg open om bestaande Forms met minimale, of zelfs zonder, aanpassingen te integreren in een (nieuwe) webapplicatie. Dit betekent dat men niet langer gevangen zit in Oracle Forms en dat voor uitbreidingen op het systeem gebruik gemaakt kan worden van de nieuwe technologie.

Voor veel voormalig Forms gebruikers ligt een keuze voor ADF als web technologie voor de hand. Als voor ADF gekozen wordt, dan zou het gebruik van JHeadstart ook serieus overwogen moeten worden. JHeadstart is een uitbreiding op Oracle JDeveloper waarin jaren ervaring en best practices van Oracle Consulting is verwerkt.

JHeadstart maakt het mogelijk met metadata en properties declaratief vast te leggen wat een applicatie moet doen.

Vervolgens kan de JHeadstart Application Generator een geavanceerde applicatie genereren op basis van ADF Faces en ADF Business Components.

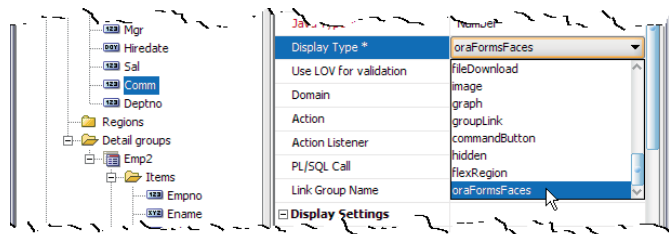
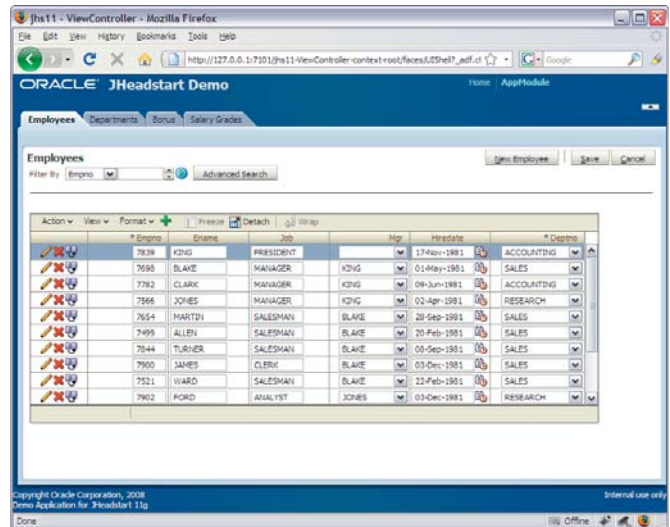
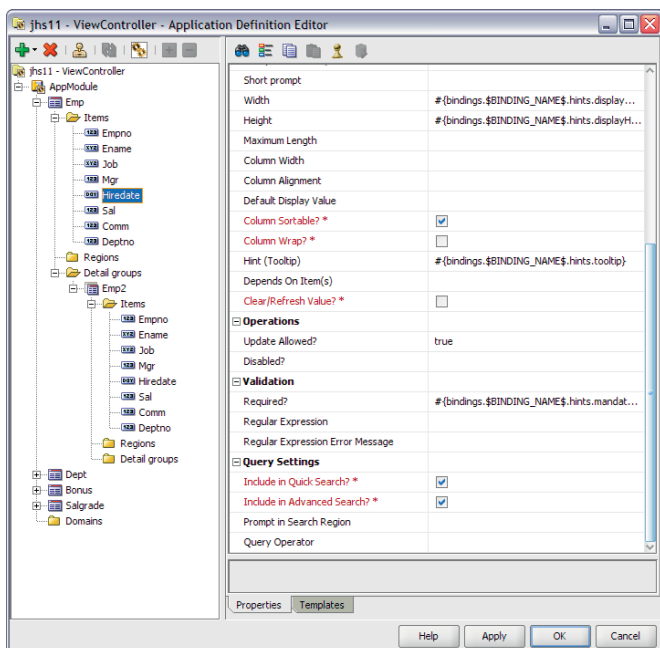
De JHeadstart Application Generator is te vergelijken met de Forms Generator zoals we die vroeger hadden in Oracle Designer, al is JHeadstart wel een stuk flexibeler dan de Designer Forms Generator.

JHeadstart genereert geen Java code. Alles wat gegenereerd wordt is XML en is opgebouwd door verschillende stukjes template te combineren. Hierbij wordt gebruik gemaakt van de open source Apache Velocity template engine. Als ontwikkelaar kun je al deze templates globaal of specifiek voor een pagina of item aanpassen. Hiermee geeft JHeadstart je letterlijk een grote voorsprong. De meegeleverde templates leveren al een solide applicatie gebaseerd op jaren ADF ervaring, maar het is toch mogelijk om op ieder klein onderdeel af te kijken als dat gewenst is.

JHeadstart en OraFormsFaces

De JHeadstart generator kan prima gebruikt worden voor het bouwen van (delen van) nieuwe applicaties. In combinatie met OraFormsFaces geeft dit de mogelijkheid om de bestaande Forms op te nemen als JSF componenten en kan de applicatie op termijn verder worden uitgebreid of vernieuwd met behulp van JHeadstart.

Het JHeadstart team ziet hier ook de meerwaarde van en het standaard JHeadstart product heeft dan ook al OraFormsFaces ondersteuning out-of-the-box. Het is mogelijk om pagina's te laten genereren die met behulp van OraFormsFaces een bestaand Form hergebruiken, waarbij zaken als deeplinken naar het Form, ondersteunende buttons en security verzorgd worden.



JHeadstart Forms2ADF Generator

Terwijl OraFormsFaces gebruikt kan worden om bestaande Forms te hergebruiken en de JHeadstart Application Generator om nieuwe delen van de applicatie te bouwen, komt er een moment dat sommige Forms vervangen moeten worden door volledige ADF pagina's. Sinds kort bevat JHeadstart een nieuwe generator, de Forms2ADF Generator, die hierbij helpt. De Forms2ADF Generator neemt bestaande Forms FMB bestanden en analyseert de metadata daarin. Op basis van deze informatie worden de ADF Business Components en de JHeadstart metadata aangemaakt. Eenmaal op dit punt aangekomen kunnen de Velocity templates en de JHeadstart Application Generator worden gebruikt om een applicatie te genereren waarbij de data structuren, user interface elementen en pagina layout grote overeenkomsten vertonen met het originele Oracle Form.

Het JHeadstart team heeft bewuste gekozen om te streven naar een solide ADF applicatie met een goede architectuur. Om deze reden wordt de PL/SQL code uit het Form niet geconverteerd naar Java en ADF. Er is dus nog steeds werk te doen voor een ontwikkelaar, maar het is allicht een stuk efficiënter dan de Forms geheel handmatig te herbouwen.

Gefaseerde Migratie Strategie

Met OraFormsFaces is het mogelijk om gefaseerd te migreren van Oracle Forms naar een andere web technologie. Simpel

gezegd kan een bestaande applicatie van 400 Forms worden omgezet naar een web applicatie van 400 pagina's waarbij iedere pagina een bestaand Form bevat.

Op termijn kunnen dan individuele Forms of een aantal Forms worden herbouwd in de nieuwe technologie. Dit geeft de mogelijkheid om de hele migratie te spreiden over een lange periode, wat de kosten en risico's een stuk beter beheersbaar maken. Wellicht dat een aantal Forms op deze manier nog vele jaren meekunnen.

Wanneer gekozen wordt voor een faseerde migratie naar ADF dan kan JHeadstart nog het nodige toevoegen. JHeadstart kan de initiële applicatie genereren die gebruik maakt van OraFormsFaces om de bestaande Forms te hergebruiken. JHeadstart kan ook gelijk gebruikt worden om de applicatie verder uit te bouwen. En wanneer het moment komt dat het logisch is om een aantal Forms om te zetten naar ADF, dan kan de JHeadstart Forms2ADF Generator worden gebruikt om de herbouw van het Form in ADF een stuk sneller te maken. Al met al biedt OraFormsFaces, zeker in combinatie met

JHeadstart, een uitweg voor die organisaties die zich gevangen voelen in de Forms technologie en niet veel voelen voor een big-bang migratie naar een andere technologie.



Wilfred van der Deijl is werkzaam als senior systeemarchitect bij Eurotransplant, heeft 15 jaar ervaring met diverse Oracle tools en is vorig jaar benoemd tot Oracle ACE.

(Advertentie)

Open Source Business Intelligence

met Jos van Dongen

10 november 2009 – Hotel Lapershoek Hilversum



Business Intelligence is inmiddels door verschillende ontwikkelingen, zoals het standaard toevoegen van BI-functionaliteit aan databases of office-pakketten, toegankelijk voor elke organisatie en elke gebruiker. De belangrijkste ontwikkeling is echter het ontstaan van een verzameling Open Source BI-hulpmiddelen én diverse professionele organisaties die een complete Open Source BI-stack leveren en ondersteunen. Waar we de afgelopen jaren een groeiend aandeel zagen van Linux, Open Office en MySQL, zou dat ook wel eens kunnen gaan gebeuren met bijvoorbeeld Pentaho en JasperSoft.

Tijdens het seminar worden de volgende vragen beantwoord:

- Wat is Open Source BI en welke tools en leveranciers zijn er beschikbaar?
- Is Open Source BI volwassen genoeg voor bedrijfsbrede inzet?
- Hoe kosteneffectief is Open Source BI nu echt en bestaan er al concurrerende 'closed source' alternatieven?
- Hoe richt u een Open of Mixed Source BI-stack in?
- Kunt u een 100% gratis BI-architectuur inrichten?

Bestemd voor ú

Bent u betrokken bij het evalueren, selecteren, implementeren, ondersteunen of gebruiken van Business Intelligence in een organisatie? Dan mag u dit seminar met Jos van Dongen niet missen!

Array Seminars: specialist in IT-vakkennis!

Kijk snel op www.arrayseminars.nl voor het complete programma!

Array SEMINARS

in samenwerking met **COMPUTABLE**