

“Citius, altius, fortius”

Java-development met ADF en JHeadstart

Met het Application Development Framework (ADF) gecomplementeerd door JHeadstart release 10.1.2 hebben we een ongekend productieve, declaratieve en krachtige gereedschapkast. In dit artikel zullen we met ADF en JHeadstart een applicatie ontwikkelen voor het beheer van incidenten. Deze volledig functionele applicatie – ARTIS - wordt gegenereerd op basis van declaratieve specificaties. De applicatie kan worden gedownload en – eventueel na verdere aanpassing – worden gebruikt voor je eigen bug-beheer.

In veel discussies over applicatie ontwikkeling met Java/J2EE technologie ontstaat, ongemerkt bijna, enige verwarring. Het Olympisch motto ‘meedoen is belangrijker dan winnen’ lijkt plots opgang te doen – als het maar Java is, dan zal het wel goed zijn. Uiteraard is Java/J2EE een prachtige, uitdagende technologie met bijna ongekende mogelijkheden voor de ontwikkeling van geavanceerde applicaties. Maar Java is het middel, niet het doel. Als middel heeft Java-technologie enkele beperkingen, waardoor Sneller, Hoger en Sterker (Citius, Altius, Fortius) – dat andere Olympische motto – niet altijd uit de verf komt. De complexiteit, over elkaar heen buitende vernieuwingen van tools en standaarden en bovenal de veelal sterk teleurstellende productiviteit dwingen tot kritische evaluatie van de inzet van Java. Dit artikel bespreekt hoe Oracle eerlijke doping biedt voor de Java ontwikkelaar: hoe kan de productiviteit worden opgevoerd zonder concessies te doen aan de regels – de standaarden – van het spel?

Java/J2EE voor Oracle-ontwikkelaars

Het ontwikkelen van Java/J2EE applicaties kan met een groot aantal verschillende tools, frameworks en technologieën worden gedaan. Eén van de eerste uitdagingen voor een organisatie die wil overgaan op Java/J2EE is dan ook de selectie van de te gebruiken hulpmiddelen – zie over deze keuze ook het kader Criteria voor de selectie van Java/J2EE Tools, Technologie en Componenten

Zeker voor Oracle-ontwikkelaars zijn er verschillende redenen om met extra belangstelling te kijken naar de Oracle ontwikkel-stack voor Java/J2EE. Deze stack omvat naast de Oracle-database en OC4J als applicatieserver met name Oracle 10g JDeveloper inclusief diverse frameworks en ondersteuning voor de Oracle-database. Java/J2EE is niet bijzonder toegankelijk noch erg productief voor op zich ervaren ontwikkelaars in Oracle technologie.

Diverse ontwikkelconcepten – bijvoorbeeld declaratief ontwerp gevolgd door generatie - en bestaande kennis van ondermeer SQL en PL/SQL zijn in veel tools en frameworks in het Java/J2EE domein niet meer relevant. JDeveloper biedt de eenvoudigste introductie in Java/J2EE, waarbij veel van de bestaande kennis nog wel waardevol is. Zo volgt bijvoorbeeld het ADF Business Components-framework enerzijds de J2EE-standaarden en implementeert het diverse design-patterns, terwijl het aan de andere kant met eenvoudige, veelal op SQL gebaseerde wizards eenvoudig toegankelijk is voor Oracle ontwikkelaars.

JDeveloper 10g bevat ondermeer de volgende functionaliteit met grote waarde voor Oracle ontwikkelaars:

- Database Design en Development
 - (Visueel) Database Design, Design Capture en Generatie
 - SQL Worksheet
 - PL/SQL Development en Debugging
- Integratie van Java/J2EE met de database
 - Load Java Stored Procedures in de database
 - Publiceer PL/SQL program units als webservices
- OO/R-mapping – de vertaling van de relationele wereld naar het OO-model met de geïntegreerde Oracle Toplink-workbench; Toplink bestaat sinds 1994 en is in 2002 door Oracle ingelijfd; Toplink biedt een POJO (plain old Java Beans) gebaseerd OO/R Persistency Framework dat relationele data en services in object-vorm beschikbaar stelt aan Java-applicaties.
- ADF Business Components sinds november 1999 beschikbaar onder de naam BC4J (Business Components for Java); dit is een OO/R Persistency Framework dat relationele data en services in object-vorm beschikbaar stelt aan Java-applicaties.

- met specifieke ondersteuning van Sequences, Blob-kolommen, refresh after insert/update om default-waardes en waardes gezet door triggers op te pikken
- Migratie/generatie van tabellen en modules vanuit Oracle Designer naar ADF Business Components
- ADF UIX – sinds 2000 beschikbaar en door Oracle zelf veel toegepast in ondermeer de CRM en Self-Service modules van Oracle Applications, maar ook in tools als Oracle Enterprise Manager. UIX staat voor User Interface XML, en biedt een declaratief, XML-gebaseerd mechanisme om user interfaces te definiëren; de UIX rendering engine kan op basis van deze definities HTML-pagina's genereren maar bijvoorbeeld ook WML en cHTML. ADF UIX zal op korte termijn verder evolueren tot ADF Faces, Oracle's implementatie van de nog vrije nieuwe Java Server Faces standaard voor event driven User Interfaces in ondermeer HTML. ADF Faces is sinds eind 2004 beschikbaar in diverse early adopter-releases. De definitieve productieversie, volledig geïntegreerd met het ADF Binding Framework, komt in JDeveloper 10.1.3, verwacht tegen het eind van 2005.

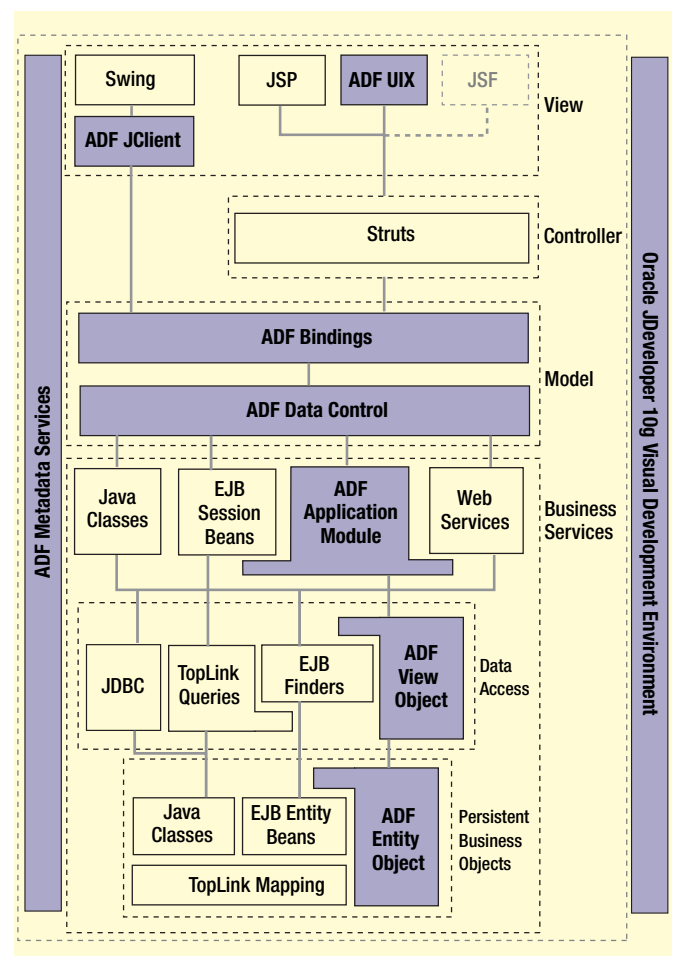
Daarnaast bevat JDeveloper visuele *what-you-see-is-what-you-get* editors voor ondermeer UIX, HTML en JSP, editors voor XML, XSLT en XSD, ondersteuning voor Java debuggen (local en remote), deployment van J2EE-applicaties, integratie van Ant en meer.

Met bovenstaande componenten heeft Oracle een krachtige geïntegreerde ontwikkelomgeving die niet onderdoet voor de beste Java IDE's in de markt. Echter, ook hiermee zijn productiviteit, toegankelijkheid voor niet-goeroes en complexiteit respectievelijk vrij laag, laag en vrij hoog. Sinds april 2004 is het Oracle ADF (Application Development Framework) in productie, als onderdeel van JDeveloper 10g. ADF biedt faciliteiten om data elementen die door het model beschikbaar worden gesteld – zoals attributen of methodes van ADF BC-Objecten, property's van JavaBeans en Methodes binnen webservices – te koppelen aan de View-pagina's, meestal JSP of UIX. Voor Oracle Designer gebruikers: dit lijkt heel erg op het definiëren van table- en column-usages in Forms-modules.

Application Development Framework

ADF biedt ontwikkelfaciliteiten om de databinding te specificeren – deels zelfs door simpel drag & drop naar de pagina. Daarnaast biedt ADF de runtime library's die de data-afhandeling rond een pagina regelen. Dit betekent dat ADF ervoor zorgt dat de data die een pagina nodig heeft bij het model wordt aangevraagd en op een voor de pagina bekende plaats wordt klaargezet. Het betekent ook dat data die door de gebruiker in de pagina is ingevoerd of gewijzigd op de juiste manier door ADF aan het model wordt teruggegeven om permanent vast te leggen in de database.

ADF laat de ontwikkelaar alle beschikbare business-services registreren als Data Controls – dit kunnen webservices zijn, Java Beans, database PL/SQL-functies en tabellen of views (de laatste twee indirect). Deze Data Controls zijn vervolgens de componenten die paginaontwikkelaars in hun pagina's kunnen toepassen, door ze er simpelweg naar toe te slepen. Uiteraard dient er dan nog het nodige aan de pagina lay-out, eventuele client-side logica en ook niet-scherm gerichte Controller logica gebouwd te worden, maar de ruggengraat van een data- en transactiegerichte applicatie kan op deze wijze vrij snel worden neergezet. Het is bijna alsof je met de hand Forms bouwt: in een lay-out editor met slepen, klikken en property palettes de basislay-out en functionaliteit bouwen. Met ADF besteedt de ontwikkelaar veel tijd aan het registreren van Data Controls en vervolgens het bouwen van individuele pagina's, waarbij vrij weinig generieke definities over pagina's heen gebruikt kunnen worden. Daarnaast kan relatief vaak gewenste functionaliteit – zoals multi-record schermen, List of Values, master-detail schermen – nog de nodige hoofdbreken opleveren en veel tijd aan handmatig ontwikkelwerk vergen.



Afbeelding 1. Het Application Development Framework

JHeadstart voor ADF

Dit is het punt waar JHeadstart zijn intrede doet. Waar Oracle Designer aan de ontwikkeling van Oracle Forms een grote mate van standaardisatie en productiviteit heeft toegevoegd door ondermeer het centraal en declaratief vastleggen van eigenschappen die door generatie op vele plaatsen in de applicatie kunnen worden toegepast, doet JHeadstart datzelfde voor ADF!

In februari 2005 zag de eerste ADF JHeadstart Release (10.1.2) het licht, nadat sinds september 2004 diverse bètaversies al in de praktijk zijn ingezet. ADF JHeadstart bouwt voort op het fundament dat sinds 2001 is opgebouwd. Eind juli publiceerde Oracle een upgradereleases voor 10.1.2.

JHeadstart 10.1.2 voegt aan JDeveloper een geïntegreerd tool toe dat op basis van simpele declaratieve definities applicaties genereert op basis van ADF en naar keuze UIX of JSP. Dit houdt concreet in dat JHeadstart Data Controls registreert, het Struts-configuratiebestand genereert en een menu, message-teksten en individuele pagina's creëert met een gemeenschappelijke look & feel.

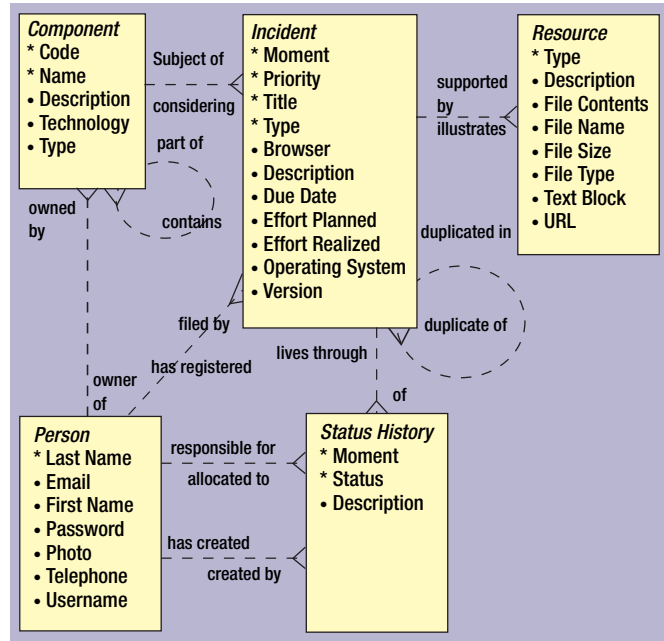
Bij de generatie ondersteunt JHeadstart geavanceerde constructies zoals:

- Item Groups
- Multi-level tabbed menus
- Master-Detail(-Detail-Detail)
- Multi-Record Insert/Update/Delete
- Tree of Navigator Style
- List of Values en Shuttle voor enkelvoudige en meervoudige selectie
- Zoek pagina's en Zoek Widgets binnen resultaat pagina's
- Sorteerbare kolommen in Tabel-layout schermen
- Table Detail-Disclosure, eventueel genest (tabel binnen tabel binnen tabel ...)
- Breadcrumbs – broodkruimelspoor langs alle pagina's die door een gebruiker zijn bezocht

Al deze elementen kunnen ook handmatig worden gebouwd, vaak wel met flinke investeringen in tijd en moeite. ADF is nog betrekkelijk nieuw en nog sterk in ontwikkeling. Eén van de grote toegevoegde waarden van JHeadstart is het feit dat het een bundeling geeft van een groot aantal 'best practices' met ADF zoals die het afgelopen anderhalf jaar door Oracle Consulting verzameld zijn. Wil je weten hoe je iets kunt bereiken met ADF dan heb je een goede kans dat je er achter komt als je de applicatie genereert met JHeadstart 10.1.2.

Ontwerp van de ARTIS demo-applicatie

De ARTIS applicatie - Application for Registration & Tracking of Incidents and Solutions – begint zijn bestaan in de Entity Relationship Diagrammer in Oracle 10g Designer (zie afbeelding 2).



Afbeelding 2. Entity Relationship Diagrammer in Oracle 10g Designer

De entiteit Incident vormt de kern van het datamodel. Voor een Incident wordt de prioriteit, een omschrijving en indien van toepassing het operatie systeem en de browser vastgelegd ondermeer vastgelegd. Daarnaast wordt een Incident geassocieerd met een Component. Componenten zijn applicatie-onderdelen, zoals schermen, rapporten en batch-functies. Iedere Component heeft een eigenaar – een Persoon – die eindverantwoordelijk is.

Incidenten worden geregistreerd door Personen en zijn vervolgens ook altijd toegewezen aan een Persoon; totdat een Incident zijn eindstatus bereikt – opgelost, duplicaat, wordt niet opgelost. Overigens zal een Incident middels de Status Historie steeds aan andere Personen toegewezen kunnen worden.

Tenslotte kunnen aan een Incident ook Resources worden gekoppeld; dit zijn bijvoorbeeld screenshots van een foutsituatie, stacktraces of voorbeelden van incorrecte rapport-uitvoer. Niet zichtbaar in het ontwerp, maar wel vastgelegd in Oracle Designer zijn alle domein definities: Component Type, Incident Prioriteit, Incident Status, Resource Type en Incident Type. Ik heb deze geassocieerd met de attributen. Deze domeinen zien we later terug als 'dropdown'-lijstjes in de schermen.

Tegelijk met het Entity Relationship Diagram hebben we ook al een aantal schetsjes gemaakt van hoe de schermen van de ARTIS-applicatie eruit komen te zien. We hebben al een redelijke voorstelling van de voornaamste modules:

- *Een scherm om nieuwe Incidenten te registreren.* Een rechttoerechtaan scherm waar naast Incident details ook in een multi-record block Resources kunnen worden toegevoegd.
- *Een scherm om bestaande Incidenten te 'tracken'.* Met behulp van een uitgebreide zoekfunctie kunnen incidenten met ver-



Wildher levert hooggekwalificeerde ICT oplossingen en beheerdiensten. In de Oracle markt is onze focus gericht op het implementeren van Oracle Collaboration Suite, Oracle Databases én onderliggende technologieën. De kracht van ons bedrijf zit in het gecombineerde vermogen van kennis en ervaring van onze professionals waarmee resultaat wordt neergezet voor en samen met de klant.

Binnen de unit ICT Diensten en Techniek is de afdeling DBA Services verantwoordelijk voor consultancy en voor de invulling van de SLA gebaseerde databasebeheercontracten. Voor advies, inrichting, migratie en optimalisatie van Oracle omgevingen en infrastructures zijn hooggekwalificeerde consultants beschikbaar. Het databasemanagement-product FOT Suite vormt een belangrijke rader in het totale palet van remote beheerdiensten van Wildher. In verband met de toenemende hoeveelheid werk zoeken we twee

Senior Oracle DBA-specialisten

Functie eisen:

- Kennis en ervaring met Oracle database versies 8i, 9i, 10g en iAS, als ook met RAC en/of Dataguard;
- Gedegen kennis van Unix en Windows;
- Kennis van het databasemanagement tool FOT Suite is een pré.

Van deze DBA verwachten wij de volgende meerwaarde:

- de drive om Oracle ontwikkelingen te onderzoeken en uit te leggen bij klanten en collega's;
- de heldere visie met betrekking tot Methodisch Beheer™, outsourcing van dienstverlening, architecturen voor high-availability, benodigde tooling voor ketenbeheer;
- de actieve rol bij de opzet en inregeling van onze beheercontracten zonder 9 tot 5 mentaliteit en vanuit een servicegerichte en communicatieve attitude.

Wat stellen wij hier tegenover:

Een aantrekkelijk salaris, afhankelijk van opleiding en ervaring. Prima secundaire voorwaarden en een prettige werkomgeving in een hectische wereld. Ten behoeve van de invulling van de functie wordt een Linux cluster ter beschikking gesteld voor research.

Ben jij die enthousiaste kenniswerker die graag wil werken in een solide bedrijf met een bijzonder servicegerichte bedrijfscultuur? Stuur je sollicitatie met uitgebreide C.V. naar:

Martin Rutten, Coördinator DBA Services, Tolakkerweg 157, 3738 JL Maartensdijk

Nadere informatie: 06-2128 4016, kijk op www.wildher.nl, of vraag informatie via martin.rutten@wildher.nl

Een psychologisch onderzoek kan deel uit maken van de sollicitatieprocedure.



De Caesar Groep is een open en eerlijke ICT-onderneming die oplossingen biedt met aantoonbaar rendement voor de klant. Wij nemen daarbij de doelstellingen van de klant als uitgangspunt en geven garantie op het behalen van rendement. Caesar beschikt over uitgebreide technologie-expertise (Oracle, Microsoft, Progress en Infrastructuur) en kennis van de markten waarin wij ons begeven. Onze diensten omvatten advies, projecten, detachering, implementatie en beheer. Caesar heeft ruim 300 medewerkers en is gevestigd in Utrecht; ons bedrijf is door Management Team verkozen tot beste IT-adviseur van 2003.

BEN JIJ TOE AAN VERDERE PROFESSIONALISERING EN RESULTAATGERICHT, DAN ZIJN WIJ OP ZOEK NAAR JOU!

RESULTAATGERICHTE ICT-ERS DIE WERKEN AAN RENDEMENT

ORACLE-SPECIALISTEN MET ÉÉN OF MEER VAN DE VOLGENDE EXPERTISES:

DBA-SPECIALIST

Je hebt kennis van de database en de applicatieserver (9ias), met betrekking tot installatie, inrichting, back-up & recovery, performance en security.

JAVA/J2EE ARCHITECT

Je hebt aantoonbare kennis van Java/J2EE. Verder ben je bekend met JDeveloper, ADF, JSP, Struts, servlets, EJB, RUP methodiek én OO analyse en design (UML).

BI-SPECIALIST

Je hebt kennis van Oracle Warehouse Builder en Discoverer (en Oracle Reports als pré). Je hebt ervaring met ontwerpen en realiseren van ETL processen.

DESIGNER/DEVELOPER SPECIALIST

Je hebt aantoonbare ervaring met minimaal één van de volgende zaken: Designer 6i, 9i of 10g en/of Developer (forms en reports) 6i, 9i, of 10g.

Je hebt 1 tot 4 jaar aantoonbare ervaring in je vakgebied, hebt een opleiding op HBO-niveau, bent ondernemingsgericht en werkt graag mee in projecten. Je salaris is afhankelijk van je ervaring en kennisniveau. Ons flexibel en individueel in te vullen arbeidsvoorwaardenpakket is zonder meer uitstekend te noemen. Bovendien besteden wij veel aandacht aan opleidingen.

INTERESSE?

Mail je reactie met CV en motivatie naar Aafke Gosses, a.gosses@caesar.nl

www.caesar.nl

schillende zoekcriteria worden opgezocht. Van een incident kan vervolgens eenvoudig de huidige stand van zaken, de volledige historie en de lijst van Resources bekeken en bijgewerkt worden.

- *De takenlijst.* Een scherm waarop een Persoon snel een overzicht heeft van de nog openstaande Incidenten die aan hem of haar zijn toegewezen.
- *Onderhoudschermen voor Personen en Componenten.* Hierbij maken we voor Componenten – een hiërarchische datastructuur – gebruik van een scherm met boomstructuur (een ‘tree’).

Met de Database Design Transformer wordt dit ontwerp getransformeerd tot een Database Design. Door de DDT worden ondermeer Surrogate Keys gecreëerd – een ID-kolom in alle tabellen – gekoppeld aan sequences. Vanuit de Server Generator in de Design Editor worden vervolgens de tabellen, sequences en ook de Table API-packages en triggers gegeneerd en in het database-schema gecreëerd. De kolommen die aan een sequence gekoppeld zijn of een default-waarde hebben, heb ik als Server Derived gedefinieerd.

Teneinde gebruik te kunnen maken van de JHeadstart Designer Generator heb ik een ARTIS_MAIN-moduledefinitie aangeemaakt. Deze module bevat module-componenten voor alle tabellen in de applicatie. Ik ga later gebruikmaken van ARTIS_MAIN om de definities van tabellen en domeinen naar JDeveloper en ADF Business Components te migreren.

Installatie en Configuratie van JHeadstart

Alvorens nu JHeadstart te kunnen gaan gebruiken moeten we het eerst installeren en configureren:

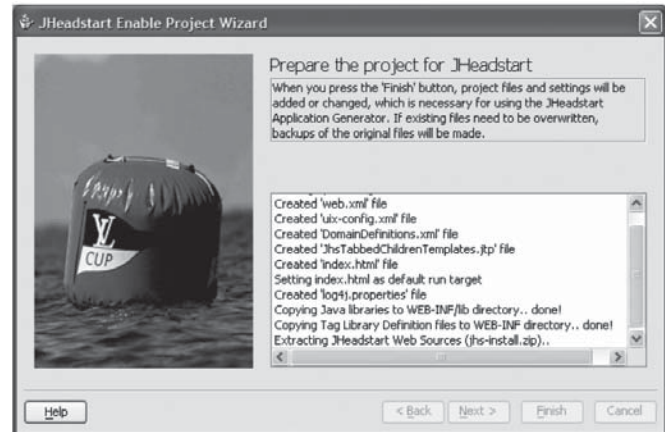
1. Download JHeadstart van OTN (<http://www.oracle.com/technology/consulting/9iservices/jheadstart.html>) NB: de upgrade release van zomer 2005 wordt niet op OTN beschikbaar gemaakt; deze versie is alleen verkrijgbaar voor licentiehouders met de zogenaamde Supplement Optie.
2. Unzip de zip-file naar een willekeurige directory, bijvoorbeeld c:\jheadstart10_1_2
3. Kopieer de files uit c:\jheadstart10_1_2\config\jdevaddins naar een nieuwe directory <JDeveloper_Home>\jdev\lib\ext\jheadstart
4. Kopieer c:\jheadstart10_1_2\config\jdevaddins\migration.xml naar <JDeveloperHome>\jdev\system10.1.2.0.0.1811
5. Start JDeveloper 10.1.2, ga naar het Tools menu, kies de optie Preferences, selecteer de categorie ‘JHeadstart Settings’ en selecteer de directory c:\jheadstart10_1_2

Op dit punt aangekomen kunnen we een nieuwe applicatie ontwerpen en genereren.

Ontwerp een JHeadstart-applicatie

Om een nieuwe applicatie te gaan ontwikkelen, starten we met een nieuwe Application Workspace met Web Application [Default]

als technology-template. Uit het rechtermuisknop-menu (RMB menu) op het ViewController project kiezen we nu ‘Enable JHeadstart on this Project’. De JHeadstart Enable Project Wizard start. Deze gaat ons JDeveloper project voorbereiden op het gebruik van JHeadstart. Standaardcomponenten en library’s worden aan het project toegevoegd, de geëigende directory-structuur wordt gecreëerd en de configuratie-files worden geïnitieerd.

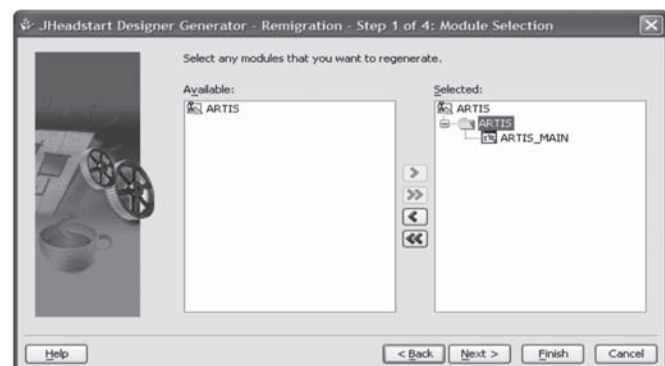


Afbeelding 3. De Enable Project Wizard gaat het JDeveloper project voorbereiden op gebruik van JHeadstart

Nu hebben we twee mogelijkheden om de Business Service op basis van ADF BC te gaan construeren in het Model-project:

- Op basis van tabellen en views in de database
- Op basis van de ARTIS_MAIN module, tabel-definities en domeindefinities in Oracle Designer

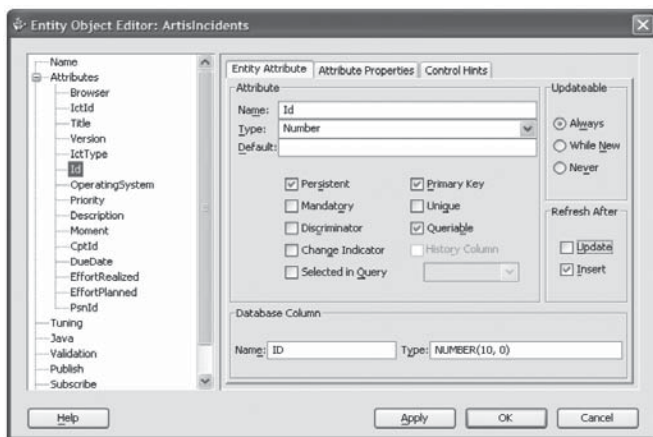
Deze twee methodes kunnen overigens ook gemengd worden. In dit geval, aangezien het databasedesign volledig vastligt in Oracle Designer, maak ik gebruik van de JHeadstart Designer Generator. Dit levert rijkere Entity Object- en View Object-definities op dan de wizard op basis van de database-objecten. De generator kan worden gestart vanuit het rechtermuisknop-menu op het Model-project.



Afbeelding 4. De Designer Generator kan worden gestart vanuit het rechtermuisknop-menu op het Model-project.

Deze wizard genereert een ADF BC Application Module, Entity Objecten voor alle tabel definities uit Oracle Designer die aan de ARTIS_MAIN module zijn gekoppeld en ook View Objecten voor alle Module Components. Daarnaast worden een ArtisApplicationStructureFile.xml en een DomainDefinitions.xml gecreëerd. De eerste bevat de definitie van de structuur van onze applicatie, de tweede bevat de definities van alle domeinen en toegestane waarden in XML formaat.

Eén van de aardige kanten van de JHeadstart Designer Generator is dat de Id-attributen, hoewel gebaseerd op verplichte database-kolommen, in het Entity Object als niet verplicht zijn gedefinieerd. Dit is de consequentie van de instelling Server Derived die we in de Designer hebben gemaakt vanwege het gebruik van de database-sequence en de TAPI-trigger. Vandaar ook dat de generator de checkbox 'Refresh After Insert?' heeft aangevinkt.



Afbeelding 5. De Id-attributen zijn als niet-verplicht gedefinieerd in het Entity Object.

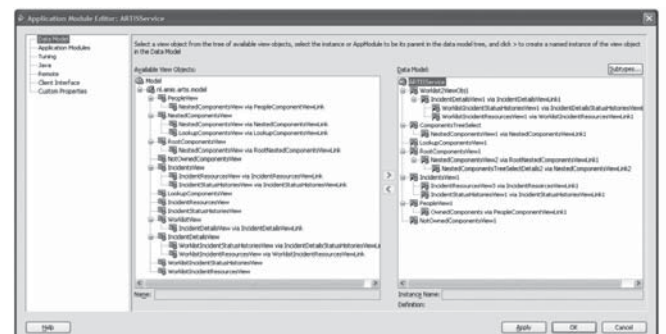
JHeadstart en ADF Business Components

JHeadstart bouwt op ADF en meer specifiek op de ondersteuning door ADF van Business Components. JHeadstart gebruikt ADF BC zowel als metadata-opslag om de applicatiegeneratie aan te sturen alsook als runtime persistency framework. Hoewel ADF op zich ook webservices, Toplink, JavaBean en POJO-gebaseerde business services ondersteunt kan JHeadstart op dit moment alleen met ADF BC overweg. JHeadstart applicatiegeneratie maakt allereerst gebruik van het Data Model van de Application Module en de definitie van de ViewObjecten. Eventueel worden eigenschappen van onderliggende Entity Objecten gebruikt. In het algemeen zal je voor iedere pagina een ViewObject maken voor ieder 'block'. Bijvoorbeeld in ARTIS willen we schermen om nieuwe incidenten te loggen, om incidenten te tracken en om de werklis van een gebruiker op te vragen – met alle momenteel aan de gebruiker toegewezen incidenten. We maken voor elk van deze drie modules een ViewObject op basis van het ArtIsIncidents

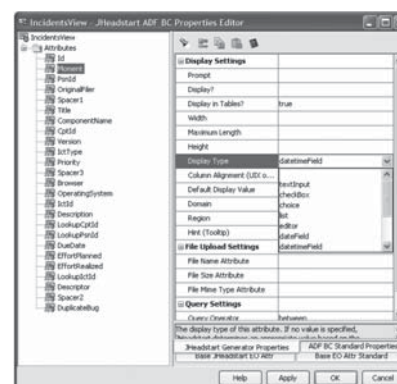
Entity Object. Al met al ziet het model van de Application Module voor ARTIS er als volgt uit. Let op de geneste structuren waar we met Master-Detail relaties tussen ViewObjecten te maken hebben; deze nesting is noodzakelijk om met JHeadstart en ADF master-detail pagina's te creëren. Dit is overigens een substantieel verschil met de vorige versie van JHeadstart.

Met een JHeadstart plugin kunnen eigenschappen van ViewObject attributen aangepast worden, vergelijkbaar met de manier waarop je in Oracle Designer in een property pallet de eigenschappen van een Module Item manipuleert. Eigenschappen die door de generator worden gebruikt zijn ondermeer Prompt, DisplayType, Domain, Region, Default Display Value en Hint.

Attributen van ViewObjecten of EntityObjecten kunnen refereren naar een Domain. De definitie van de Domains ligt vast in een XML-document. De toegestane waarden worden door JHeadstart tot select (pop-list) gegenereerd in de webapplicatie. We kunnen onze Business Service eenvoudig testen: het RMB menu op de Application Module bevat de optie Test. Deze optie start de Application Module Tester, een eenvoudige JavaClient applicatie die ons door de ViewObjecten laat browsen. We kunnen verifiëren dat we de gewenste data zien die straks in de webpagina moet worden getoond. Ook kunnen we snel zien of datamanipulatie mogelijk is.



Afbeelding 6. Het model van de Application Module voor ARTIS



Afbeelding 7. Met een JHeadstart plugin kunnen eigenschappen van ViewObject attributen aangepast worden

Ontwerp de webapplicatie

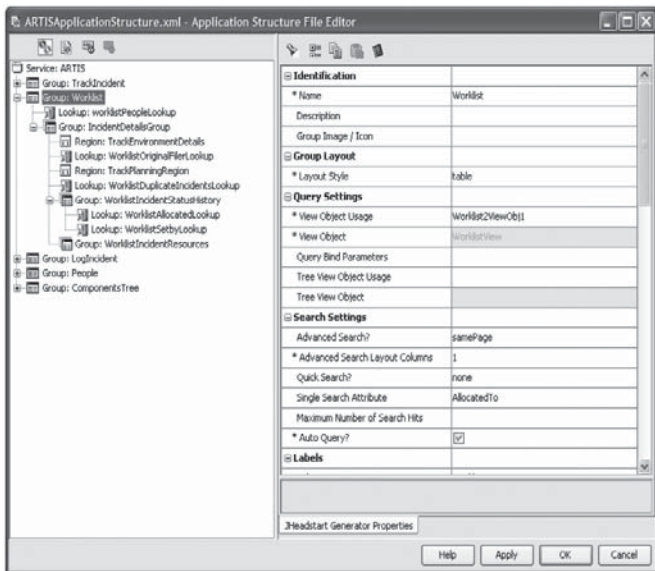
De Application Structure-file bevat de definitie van de individuele modules – clusters van pagina's. De Groups definiëren pagina's op basis van een ViewObject Usage uit de Application Module. Eén group kan een Find-, Select-, Table- en Form-page opleveren. Groups kunnen met Master-Detail constructies verbonden worden – op basis van ViewObjects die met ViewLinks verbonden zijn en in het Data Model van de Application Module ook als geneste structuur zijn opgenomen.

Een Master-Detail relatie kan op verschillende manieren tot uitdrukking worden gebracht door de generator:

- Een gewone master-block met multi-record details block
- Een tree of navigator structuur
- Een shuttle

Aan een Group kunnen een of meer Regions worden gekoppeld. Een Region is een cluster van velden, met een eigen titel en eventueel een eigen aantal kolommen. Een Region is te vergelijken met een Item Group in Oracle Designer. ViewObject attributen kunnen aan een Region gekoppeld worden.

Als een veld in een pagina via een lijst van waarden of een poplist gevuld moet kunnen worden, kan aan een de group ook een lookup worden verbonden. Een lookup is een element waarmee gespecificeerd wordt hoe op basis ViewObject een lijst van lookup-waarden kan worden gepresenteerd, als poplist of als List of Values. De geselecteerde waarde wordt in het aan-gegeven veld vastgelegd.



Afbeelding 8. De Application Structure-file bevat de definitie van de individuele modules

Genereer de webapplicatie

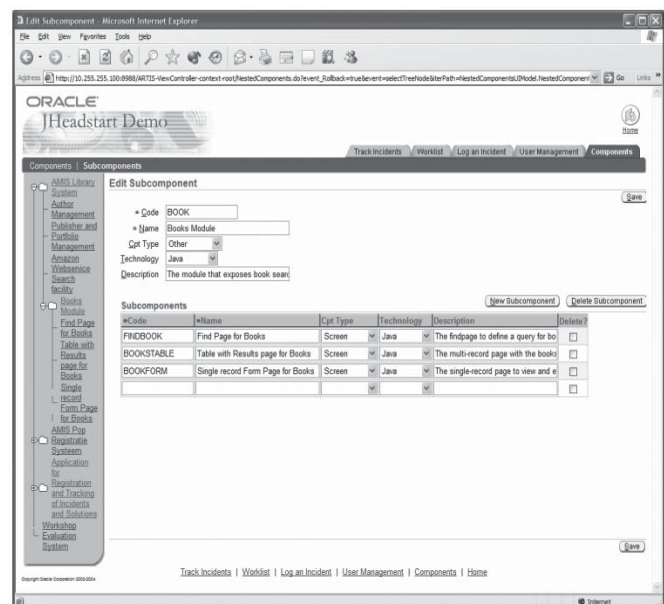
De applicatie kan op basis van de Application Structure File worden gegenereerd, met een druk op het icoontje linksboven

in de Application Structure File Editor of vanuit het rechtermuisknop-menu op de navigator-node van de Application Structure File. De JHeadstart Application Generator genereert de volgende objecten, op basis van de (configureerbare) Generator Templates, ViewObject en EntityObject definities, de Domain Definities en de Application Structure File:

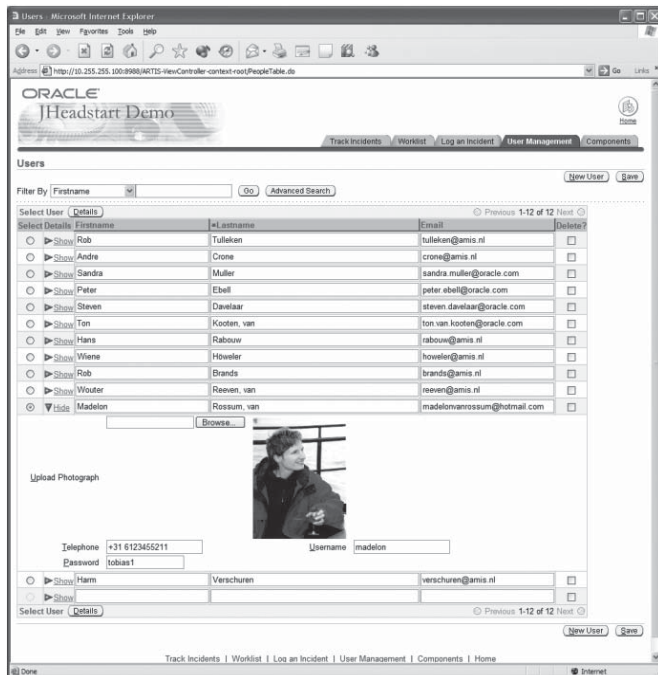
- Resource bundles met alle schermteksten zoals prompt, hint-text, buttonlabel, menulabel en informatie- en foutteksten.
- Struts-Config.xml file met ActionMappings, Forwards en FormBeans.
- UIX- of JSP-pagina's

De gegenereerde applicatie kan eenvoudig worden gerund, door de index.html file die tijdens JHeadstart-enabling van het ViewController-project is gecreëerd. Het scherm waarop we componenten beheren is een ideale kandidaat voor een tree-layout: de componenten vormen een hiërarchie, van applicatie tot individueel te buggen onderdeel. Het scherm laat ons door de componentenhiërarchie navigeren en stelt ons in staat een node te selecteren en te gaan bewerken.

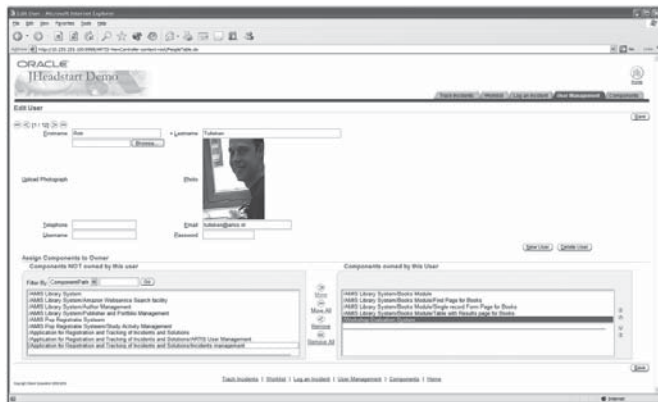
Het multi-record scherm waarop de gebruikers van ARTIS kunnen worden onderhouden, bevat het UIX-specifieke Detail Disclosure mechanisme. Hiermee kan een rij worden 'opengeklapt', zodat zonder extra navigatie naar de detail-pagina alle gegevens gezien kunnen worden. Ook wordt hier de file upload en download gedemonstreerd, om pasfoto's van de gebruikers en incident-eigenaren in het systeem op te nemen. Op de gebruiker-detailpagina kunnen met een Parent Shuttle componenten aan gebruikers worden toegekend.



Afbeelding 9. Het scherm waarop we componenten beheren is een ideale kandidaat voor een tree-layout



Afbeelding 10. Het multi-record scherm bevat het UIX-specifieke Detail Disclosure mechanisme

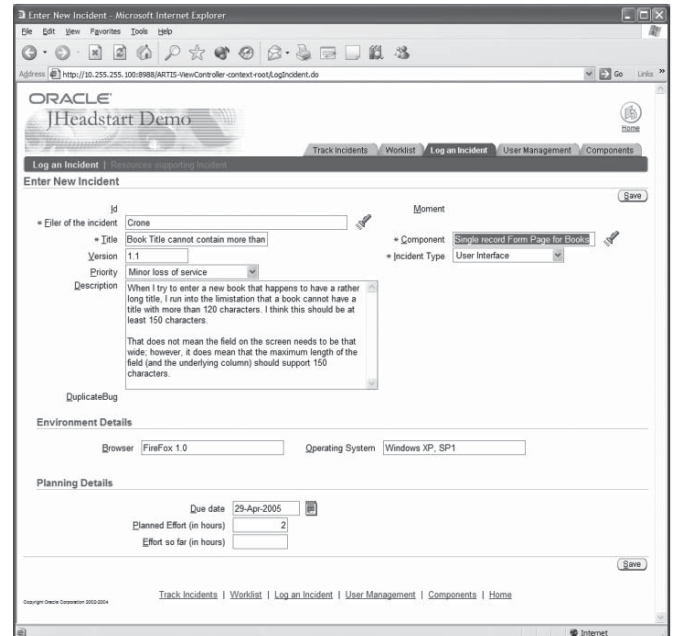


Afbeelding 11. Op de gebruiker-detailpagina kunnen met een Parent Shuttle componenten aan gebruikers worden toegelinkt

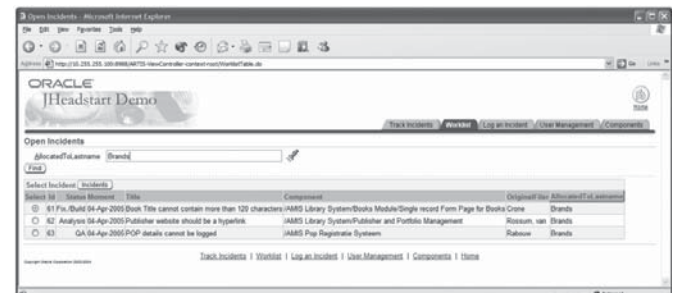
Log en track een incident

Het centrale scherm van de ARTIS-applicatie is het Log Incident scherm. Hier worden de gegevens van een incident geregistreerd. De gebruiker geeft aan voor welke (versie van welke) component wat voor soort incident is opgetreden, wat de prioriteit is en eventueel in welke browser of welk operating system het probleem zich voordoet. In het detailscherm kunnen ondersteunende materialen zoals stacktraces, memory-dumps, screenshots worden toegevoegd aan het incident. Nota bene: op het veld 'Filer of the incident' is LOV-validatie actief. Dat wil zeggen dat het invoeren van de eerste paar karakters van een naam al voldoende kan zijn om de waarde te laten aanvullen door het systeem. Als de eerste karakters niet

uniek zijn, wordt een gefilterde Lijst van Waarden getoond. UIX maakt hier gebruik van de Partial Page Refresh waarbij op het moment van verlaten van het veld met de gedeeltelijke naam een onzichtbare roundtrip naar de server wordt gemaakt waarin de invoer wordt gevalideerd. Dit is een implementatie van AJAX (Asynchronous JavaScript and XML) waarover de laatste maanden veel te doen is.



Afbeelding 12. Het volledige Log- en Incident-scherm

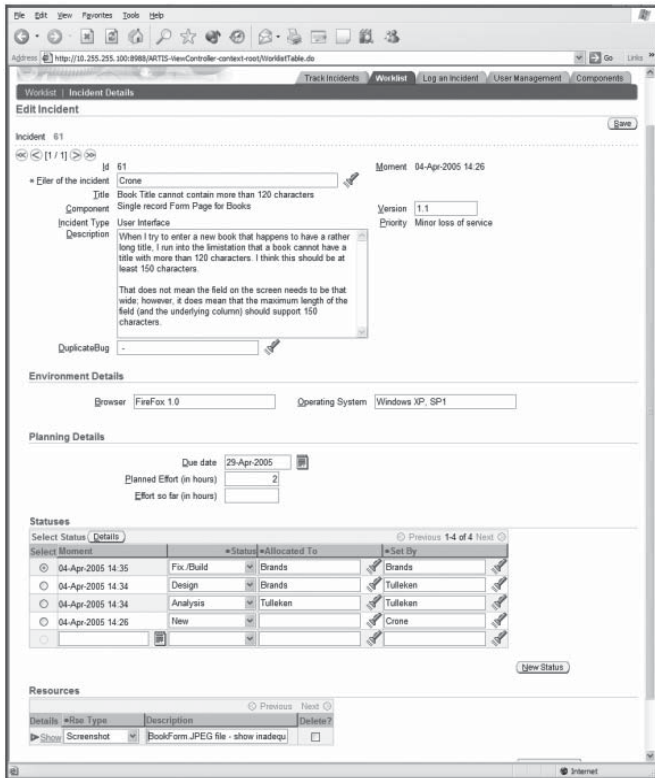


Afbeelding 13. De werklis toont een lijst van alle nog niet afgesloten incidenten die aan een medewerker zijn toegewezen

Afwerking van uiterlijk schoon

De applicatie is met generatie alleen meestal nog niet klaar. Vaak is het uiterlijk nog niet volledig naar wens, al was het maar dat standaard het Oracle-logo in de applicatie is opgenomen. De look & feel wordt grotendeels gedefinieerd buiten het generatieproces om. Op min of meer ieder willekeurig moment kan deze worden aangepast.

Voor de hand liggende aanpassingen van de look & feel liggen op de volgende gebieden:



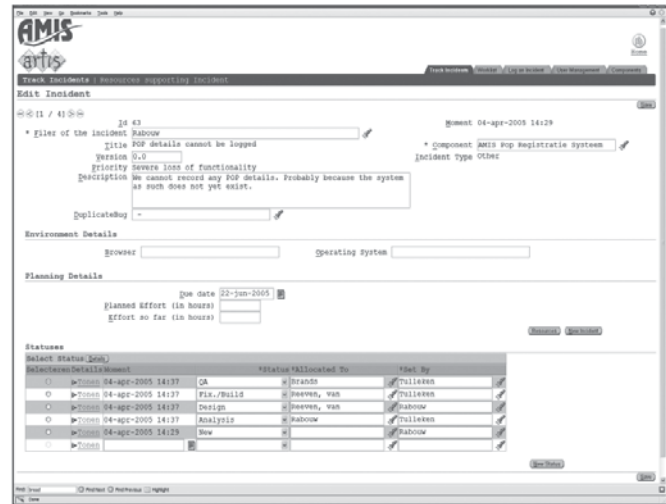
Afbeelding 14. De lijst van incidenten die nog openstaat voor medewerker 'Rob Brands'

- Schermteksten en NLS
- Look & feel: fonts, kleuren, contrasten en andere stijlkenmerken
- Vaste schermelementen als logo, kop- en voetteksten

Alle teksten die we zien in de schermen worden tijdens het uitvoeren van de applicatie vanuit door JHeadstart gegenereerde resource bundles gelezen. JHeadstart ondersteunt elf verschillende talen voor standaardelementen als Save, Details, Transaction Successfully Committed en dergelijke. Het is aan ons als applicatieontwikkelaars om in de resource bundles de vertalingen vast te leggen van ondermeer prompts, titels en domeinwaarden. Afhankelijk van de taalinstelling van de browser die de applicatie benadert zal een resource bundle voor een bepaalde taal worden gebruikt. De fonts, kleuren en andere stijlkenmerken worden bestuurd door stylesheets. Voor een JHeadstart JSP-applicatie is dat een 'normaal' CSS stylesheet, voor een op UIX gebaseerde applicatie is dat een wat meer geavanceerd XSS stylesheet. De details daarvan voeren hier te ver, maar het is betrekkelijk eenvoudig om via de stijlen in deze externe stylesheets het uiterlijk van de applicatie te beïnvloeden.

De vaste schermelementen als kop- en voetteksten, logo's en globale knoppen staan uiteraard niet in iedere pagina gedefinieerd. Voor UIX is er een standaard page lay-out template waar-

in deze elementen staan, terwijl voor JSP gebruik wordt gemaakt van een klein aantal included JSP-fragmenten. Deze kunnen eenvoudig aangepast worden.



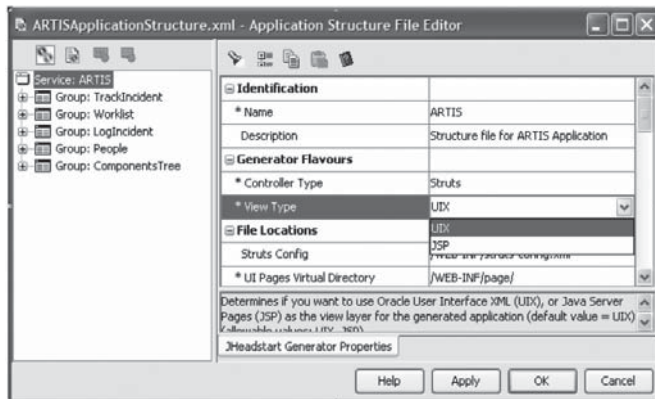
Afbeelding 15. Zonder opnieuw met JHeadstart te genereren, kan het uiterlijk van de ARTIS-applicatie worden bijgesteld.

Op zeker moment moeten we de keuze maken voor de user interface-technologie: JSP of UIX. Vanuit het JHeadstart-team komt een warme aanbeveling om waar mogelijk UIX te kiezen. Niet alleen biedt UIX nu al een rijkere user interface – hoewel waar mogelijk features ook in de JSP-generator worden ingebouwd – maar met UIX is een duidelijk migratiepad naar Java Server Faces (JSF) en meer specifiek Oracle ADF Faces ingeslagen. JSF is een J2EE-standaard die een event- en modelgebaseerde technologie beschrijft die zeer rijke HTML user interfaces kan opleveren. Oracle Applications gaat op ADF Faces standaardiseren omdat daarmee heel krachtige schermen toch binnen de webarchitectuur op basis van HTML kunnen worden ontwikkeld. Ook nu al kan met UIX declaratiever, met minder post-generatie en programmatische aanpassingen ontwikkeld worden.

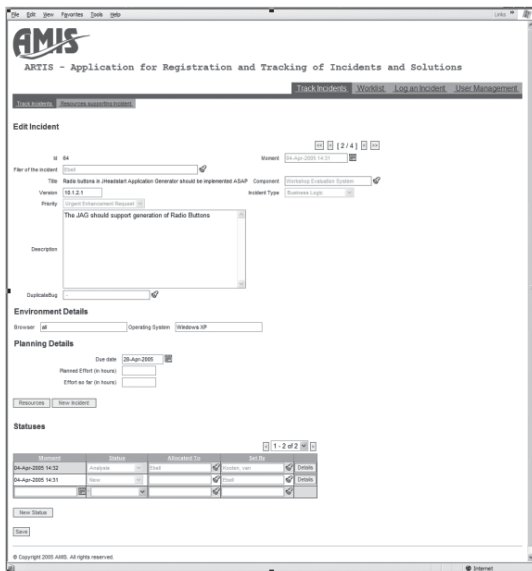
Met JDeveloper 10.1.3 – de productie versie en niet de huidige preview die geen ADF ondersteunt – zal de integratie van ADF Faces en ADF Binding Framework en JHeadstart een feit zijn. De huidige verwachting voor JDeveloper 10.1.3 is (laat) najaar 2005. Steven Davelaar, JHeadstart architect van Oracle, is desgevraagd buitengewoon enthousiast over de kwaliteit van ADF Faces; hij verwacht dat de rijke set Faces componenten, de integratie met ADF en straks de generatie door JHeadstart zo'n sterke combinatie oplevert dat de keuze voor JSP nauwelijks meer gemaakt zal worden. Steven reageert ook op een bezwaar dat vaak wordt geuit tegen UIX, namelijk dat het teveel een Oracle specifieke look & feel zou opleveren. Hij beschrijft hoe zijn team voor diverse klanten een echte 'webby' look & feel op basis van UIX heeft ontwikkeld, volledig naar

gebruikersspecificaties, met behulp van zogenaamde 'skins'. **UIX Skins** zijn een soort stylesheets die extern aan de applicatie ervoor kunnen zorgen dat de schermen een totaal ander aanzien krijgen. Zie bijvoorbeeld de volgende weblog voor details over **UIX Skins**: <http://www.orablogs.com/jjacobi/archives/000155.html>.

Het is overigens eenvoudig om naast de **UIX**-implementatie van de applicatie een **JSP**-variant te genereren. We verliezen daarbij een aantal van de geavanceerde features zoals tree, shuttle en table detail disclosure. Verder is de functionaliteit van de applicatie gelijk aan de **UIX**-applicatie.



Afbeelding 16. Om de **JSP**-variant te genereren moeten we één generatieschakelaar omzetten



Afbeelding 17. Het resultaat van de **JSP**-generatie, waarbij een **AMIS**-specifiek **CSS**-stylesheet en logo is gebruikt.

Post-generatie

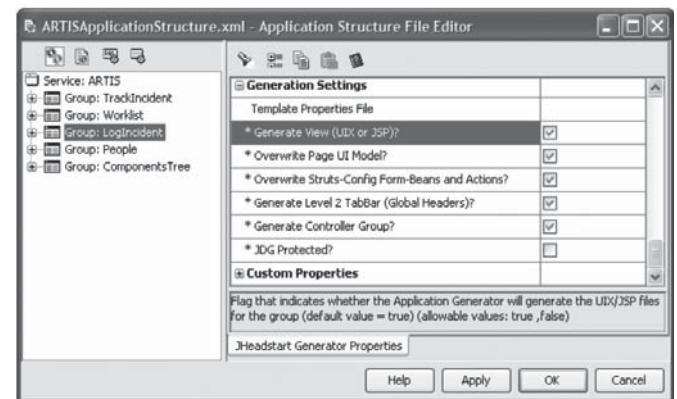
Ook al is wellicht nu de buitenkant van de applicatie zoals deze moet zijn, het komt toch ook regelmatig voor dat generatie

alleen niet voor alle schermen de volledige gewenste functionaliteit oplevert. In dat geval zou je post-generatie aanpassingen – handmatige aanpassingen in de gegenereerde objecten – kunnen overwegen. Uiteraard is dit niet ideaal: op het moment dat post-generatie zijn intrede doet in een applicatiecomponent, gaan specificatie en generator enerzijds en de werkelijke broncode anderzijds uit elkaar lopen. Dat bemoeilijkt het onderhoud en zeker een eventuele overgang naar een nieuwe versie van de generator.

Regelmatig voorkomende post-generatie aanpassingen zijn ondermeer:

- Lay-out aanpassingen (als de configureerbare template-architectuur niet toereikend is), bijvoorbeeld het opnemen van meerdere velden in een List of Values.
- JavaScript-verrijking van de client, bijvoorbeeld veldvalidatie, of het dynamisch en conditioneel tonen van schermelementen, of het toevoegen van knoppen voor navigatie of het starten van batchoperaties.
- Het uitbreiden van de **DataAction** die vanuit de **Controller** wordt aangeroepen, bijvoorbeeld om de **prepareModel()** methode uit te breiden om door database-triggers gecreëerde gegevens in te lezen.

JHeadstart biedt een systeem van generatieschakelaars, waarmee in de application structure file per group kan worden aangegeven wat er wel en niet moet worden gegenereerd.



Afbeelding 18. In de application structure file kan per group worden aangegeven wat er moet worden gegenereerd.

Met behulp daarvan kan eenvoudig geregeld worden dat handmatig aangepaste objecten – zoals **JSP**- of **UIX**-pagina's of **ActionMappings** in de **Struts Config** file - niet worden overschreven door de generator. Een meer verfijnd systeem stelt ons in staat om per scherm een specifiek template samen te stellen. In dat template kunnen we post-generatie wijzigingen opnemen. Tijdens hergeneratie maakt de **JHeadstart**-generator gebruik van dat template, en komen de post-generatie wijzigin-

Selectiecriteria voor Java/J2EE-technologie

In de vaak verhitte debatten die worden gevoerd over de keuze van Java/J2EE-technologie en criteria spelen tientallen kreten en argumenten een rol. Het lijkt zinvol deze discussie op een wat meer gestructureerde manier te voeren. Als we de gebruikte argumenten groeperen en terugbrengen tot een essentie, blijven de volgende criteria over:

- **Vereisten** – hieronder vallen met name de eisen qua functionaliteit, performance en beveiliging: kan de technologie leveren wat vereist is? En: gaan we niet investeren in mogelijkheden die helemaal niet (direct) vereist zijn? Kan de technologie eenvoudig met ons meegroeien, zowel in functionaliteit als schaalgrootte?
- **Kosten** – directe kosten licenties/beheer/omgeving, de ontwikkeling (productiviteit) en het onderhoud (NB: onderhoud vormt tot tachtig procent of meer van de totale kosten die voor een applicatie worden gemaakt!).
- **Tijd** – time-to-market (van specificatie tot implementatie) en ook: de tijd benodigd om de organisatie de technologie en tools te laten adopteren (leercurve).
- **Toekomstbestendigheid en risicogehalte** – afhankelijk van de verwachte levensduur van de applicatie, de omvang van de investering en de toekomstplannen van de organisatie kan het van groot belang zijn om een oplossing te kiezen met lage risico's en een hoge toekomstbestendigheid. Dit betekent onder meer dat afhankelijkheden kritisch beschouwd moeten worden en dat zeker afhankelijkheden van partijen waarop niet zonder meer vertrouwd kan worden zoveel mogelijk vermeden moeten worden.
 - Dat betekent ondermeer dat bij voorkeur niet gebouwd wordt op een niet bewezen technologie die slechts door een kleine speler in de markt – of de eigen organisatie – ondersteund wordt, maar ook dat technologie van een grote marktspeler die niet strategisch voor die technologie gekozen heeft vermeden moet worden. De strategische keuze van Oracle Applications om de E-Business Suite te baseren op ADF is daarmee een zeer belangrijke

garantie voor de toekomstvastheid van ADF! De beschikbaarheid van de broncode is overigens veel minder van belang dan vaak wordt gedacht: het blijkt slechts in zeer beperkte mate mogelijk laat staan wenselijk met broncode van een derde partij aan de slag te gaan.

- Andere overwegingen in deze categorie zijn stabiliteit van de applicatie en technologie, schaalbaarheid van de infrastructuur en de applicatie, geschiktheid van de applicatie voor verschillende databases of applicatieservers.
- **Beleid, ervaringen en cultuur** – hieronder valt een aantal harde randvoorwaarden zoals managementbeslissingen ten aanzien van tools of technologie (“we hebben nu eenmaal voor veel geld IBM WebSphere en een Oracle-database aangeschaft!”) en wel of geen open source (overheid en universiteiten tegen sommige conservatieve, veelal financiële instellingen), maar ook wat meer aanvechtbare overwegingen zoals persoonlijke smaak of voorkeur, wat leuk en motiverend is, en wat geavanceerd en prestigeverhogend is. In deze categorie vallen ook veel gehoorde argumenten als ‘het moet zuiver J2EE zijn’ en ‘Oracle is hartstikke proprietary’. Het lijkt er wel eens op dat een clubje ervaren Java-‘goeroes’ erop uit is zijn status overeind te houden door te hameren op criteria die veel minder hard en stellig zijn dan ze lijken.

Alle andere argumenten zijn allemaal terug te brengen tot deze vijf basiscategorieën. Als een organisatie overweegt een tools- en technologieselectie te maken, zouden allereerst deze vijf fundamentele criteria in ogenschouw moeten worden genomen.

De organisatie moet voor zichzelf, gegeven de gewenste applicatie en de verdere randvoorwaarden een gewicht aan elk van de vijf categorieën toekennen. Daarna kan iedere te beoordelen component per categorie op een schaal van 1 tot 5 of 10 gescoord worden. Gecombineerd met de wegingsfactoren levert dit een vergelijkbare eindscore op.

gen mee in het generatie resultaat. Post-generatie wijzigingen zijn daarmee vrijwel genereerbaar geworden!

Conclusies

De JHeadstart 10.1.2 release toont eens te meer de volwassenheid van het concept van declaratief ontwikkelen en genereren. Niet alleen de productiviteit maar zeker ook de structuur van het ontwikkelproces en de applicatie worden met deze wijze van ontwikkelen versterkt.

De toegankelijkheid van JHeadstart is door de verbeterde integra-

tie met ADF en binnen JDeveloper en met het vervangen van veel runtime-componenten door ADF-elementen verder toegenomen. JHeadstart is nu naadloos opgenomen binnen JDeveloper. Post-generatie wijzigingen doorvoeren is zo eenvoudig als al het ADF-ontwikkelwerk. JHeadstart neemt veel van het repetitieve, moeilijk consistent te houden of omslachtig te programmeren ADF-werk over van ontwikkelaars. Pas als het spannend wordt of nauw luistert hoeft de ontwikkelaar zijn mouwen op te stropen.

JHeadstart zelf betoont zich ook een blijvertje: sinds 2001 zien we nu de derde volledige externe releasecyclus met een steeds meer

geavanceerde set aan generatiefunctieiteit. Afhankelijk van het soort applicatie zou je kunnen zeggen dat in Olympische termen JHeadstart 10.1.2 in combinatie met ADF een echte medaillekandidaat vormt.

De toekomst

In de zomer van 2005 – vlak voor het verschijnen van dit nummer van de Optimize – publiceert Oracle een nieuwe (patch)release van JHeadstart 10.1., met daarin ondersteuning voor onder meer breadcrumbs, radiobuttons, geneste table detail disclosure, dynamisch/conditioneel aan- en uitschakelen van velden, authenticatie en autorisatie volgens J2EE en op maatwerk manier. Tegen het eind van 2005 kunnen we de 10.1.3 release verwachten. Hierin aandacht voor andere business service-technologieën naast ADF Business Components. Een belangrijke rol lijkt weggelegd voor persistentie onder de EJB 3.0-vlag met implementatie door bijvoorbeeld Oracle Toplink maar mogelijk ook Hibernate.

Lucas Jellema (jellema@amis.nl) is sinds 2002 werkzaam bij AMIS Service in Nieuwegein, als Expertise Manager Technologie en Technisch Consultant. Daarvoor werkte hij ruim

acht jaar bij Oracle, ondermeer binnen het iDevelopment Center of Excellence. Hij houdt zich onder meer bezig met Java, XML/XSLT en andere webtechnologie als ook de Oracle database en tools voor applicatie ontwikkeling.

Meer informatie

- JHeadstart homepage op OTN: www.oracle.com/technology/consulting/9iServices/JHeadstart.html
- JHeadstart Weblog: www.orablogs.com/jheadstart/
- JDeveloper en ADF Home page op OTN: www.oracle.com/technology/products/jdev/index.html
- AMIS Technology Weblog met nieuws en achtergronden bij ondermeer Oracle, Java/J2EE, ADF en JHeadstart: technology.amis.nl/blog/
- De demo-applicatie kan in zijn geheel worden gedownload van de website van Optimize (www.optimize.nl)
- Een overzicht met de voornaamste wijzigingen in de 10.1.2 versie van JHeadstart kan eveneens gedownload worden van www.optimize.nl.

NI E U W S

Eerste onderwijsinstelling over op PeopleSoft Campus Solutions 8.9

Het Regionaal Opleidingscentrum (ROC) van Twente heeft de nieuwste versie van PeopleSoft Campus Solutions in gebruik genomen. Met Campus Solutions 8.9, software die speciaal ontwikkeld is voor Nederlandse onderwijsinstellingen, beschikt het ROC over een volledig geautomatiseerde studentenadministratie die voldoet aan eisen vanuit het Ministerie van Onderwijs en de studenten van alle gemakken voorziet. Binnen het ROC van Twente is naast PeopleSoft Campus Solutions 8.9 ook PeopleSoft 8.9 HRMS geïmplementeerd. Door de inzet van deze softwarepakketten kan de onderwijsinstelling alle belanghebbers, waaronder studenten, docenten en management op elk gewenst moment direct van essentiële informatie voorzien. Gegevens over aanwezigheid, beoordelingen

en voortgang van de studenten zijn inzichtelijk, alsook de volledige administratie van het onderwijzend en facilitair personeel. Tegelijkertijd is PeopleSoft Enterprise Portal 8.9 in gebruik genomen. Binnen deze webomgeving hebben alle gebruikers direct toegang tot de applicaties die ze nodig hebben en kunnen ze zelf informatie opzoeken en contactgegevens updaten. De drie nieuwe softwaresystemen zijn binnen het bijzonder korte tijdsbestek van zes maanden in gebruik genomen. Zeker gezien het feit dat het ROC van Twente ontstaan is uit een fusie van ROC Twente Plus en ROC-Oost-Nederland, waarbij een veelheid aan informatie geïntegreerd moest worden, is de snelle implementatie geen geringe prestatie.

Release 2 van Oracle's 10g database beschikbaar

Afgelopen juli kondigde Oracle de algemene beschikbaarheid van Oracle

Database 10g Release 2 aan. Deze nieuwe versie is een verbeterde versie van Oracle Database 10g Release 1. De nieuwste versie biedt optimale beschikbaarheid, veel gebruikersgemak en extra beveiligingscapaciteiten. Klanten en partners krijgen daarmee betere ondersteuning en service tegen lagere kosten. Oracle Database 10g is de enige database die ontworpen is voor grid computing. Bèta-klanten zoals Airbus, Dell en Motorola zien de voordelen van Oracle Database 10g Release 2 ten opzichte van de eerste versie meteen. Een aantal klanten heeft direct met Oracle samengewerkt om het product te testen en feedback gegeven over Oracle Database 10g Release 2. Ook hebben meer dan honderd partners de nieuwe versie getest en zij waren tevreden over de resultaten.