

# Oracle Workflow 10g (2)

## Werkstroombesturing volgens Oracle

**Oracle Workflow is voor de meeste Oracle klanten gratis. Het is flexibel, begrijpelijk en last but not least, het kan met een beperkte inspanning gebruikt worden om bestaande applicaties efficiënter te laten werken. Reden genoeg om uitgebreid stil te staan bij Oracle Workflow. In deze tweedelige artikelenserie, waarvan hier laatste deel, legt Harold Gerritsen uit wanneer workflow zinvol ingezet kan worden, hoe het er voor de eindgebruiker uitziet, hoe het werkt en hoe je het er mee ontwikkelt. Tot slot werpt hij een blik in de nabije toekomst van Oracle Workflow.**

In het hierna volgende zullen we de concepten van het modelleren met Oracle Workflow in de vorm van een korte tutorial uiteen zetten. Ook zal worden besproken hoe een procesbesturingslaag, tussen de user interface en de business logica kan worden toegevoegd voor het *workflow enablen* van bestaande applicaties. Tenslotte zult u een vooruitblik vinden naar een nieuw workflow-product van Oracle, dat naar verwachting eind december beschikbaar zal komen.

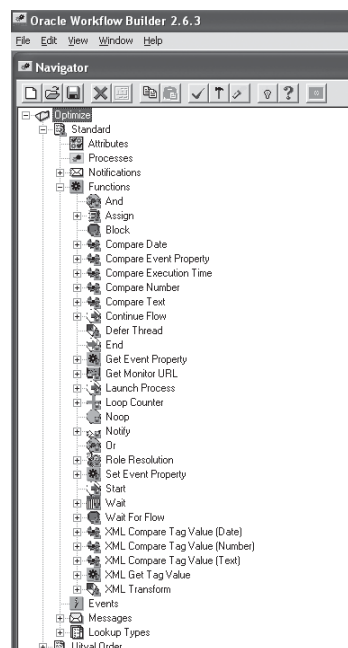
### Ontwikkelen met Oracle Workflow

Kent u dat? Als er net een nieuwe aankoop is gedaan en het product of speeltje in gebruik genomen mag gaan worden blijken er twee soorten mensen te zijn. De ene soort leest eerst alle handleidingen en gaat daarna pas aan de slag met een nieuwe aanschaf, de andere soort gaat zonder enig voorbehoud de nieuwe aanwinst te lijf. Voor alle workflow-geïnteresseerden van de eerste soort heb ik goed nieuws. Er is genoeg documentatie over Oracle Workflow om de tanden in te zetten. En 'genoeg' is hierbij een extreem understatement. De echte die-hards die eerst alles willen lezen kunnen beter een sabbatical opnemen. Het basiswerk, zeg maar de workflow-bijbel, is de Oracle Workflow Guide. Deze bestaat uit twee volumes, samengepakt in één pdf van welgeteld 1192 pagina's. Als je deze hebt gelezen en begrijpt, heeft Oracle Workflow nog maar weinig geheimen. Hoewel ik zelf van de eerste soort ben

(eerst lezen, dan doen), heb ik jaren geleden, toen ik me ging verdiepen in Oracle Workflow, gemerkt dat dat hier niet helemaal opging. Het beste is om eerst de globale concepten van Oracle Workflow te doorgronden, er dan een tijdje mee te modelleren, de gemodelleerde flows vervolgens te deployen in de runtime engine, en er dan eigen applicatielogica mee aan te roepen. Alleen zó worden de vragen opgeroepen die je nodig hebt als context om de complete handleiding te begrijpen. Daarna leest de Workflow Guide als een roman (nou ja...). Aan de hand van een korte tutorial zal ik de concepten van het modelleren met Oracle Workflow in de vorm van een korte tutorial uiteen zetten.

### Tutorial Workflow Builder

We beginnen met het starten van Workflow Builder. Daarna openen we de meegeleverde workflow-file met standaarddefinities (WFSTD.wft). Deze is te vinden in de WF\DATA\US directory van de 10g clientproducts homedirectory.



Afbeelding 1.  
Standaard functions  
in Workflow Builder

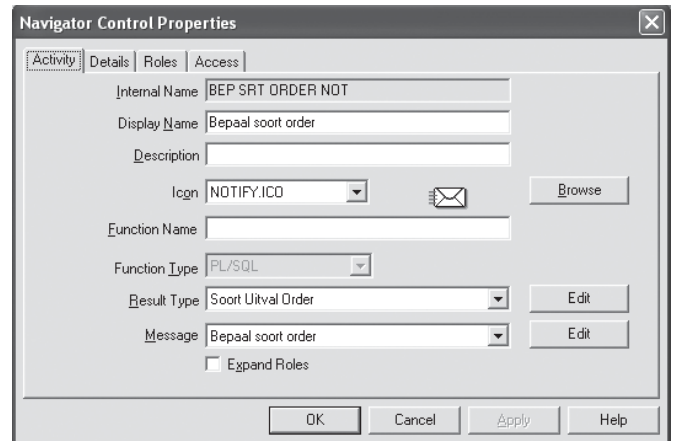
Zodra deze is geopend, kan hij direct worden opgeslagen onder een eigen naam, bijvoorbeeld *Optimize.wft*. Dit hernoemde bestand zullen we vervolgens uitbreiden met eigen workflow definities. Als we in het navigator window kijken van Builder, dan zien we (zie afbeelding 1) dat direct onder de filenaam ('Optimize') een icoon staat met de naam 'Standard'. Dat is het zogenaamde Item Type. Het Item Type is het soort object waarvan je de werkstroom wilt modelleren. Het Item Type 'Standard' is hierop een uitzondering. Dit Item Type gebruik je vanwege de voorgedefinieerde objecten die het bevat. Deze gebruik je als standaardobjecten in je eigen workflows.

Stel nu dat we een knelpunt in een order afhandelingsysteem willen oplossen met behulp van workflow. In dit ordersysteem worden orders afgehandeld waarvan sommige niet volledig geautomatiseerd kunnen worden verwerkt. Deze worden door het systeem als 'Uitval Orders' gemarkeerd. De teamleider orderverwerking moet deze uitvalorders beoordelen, het 'soort order' toekennen en eventueel de klantenservice inschakelen om de soort 'handmatig' af te handelen. Als de soort niet handmatig is, dan is het een backorder, die verder automatisch wordt verwerkt, of een procesgestuurde uitvalorder. Deze laatste soort doorloopt nog een aantal (hier niet nader gespecificeerde) stappen alvorens te worden afgewikkeld.

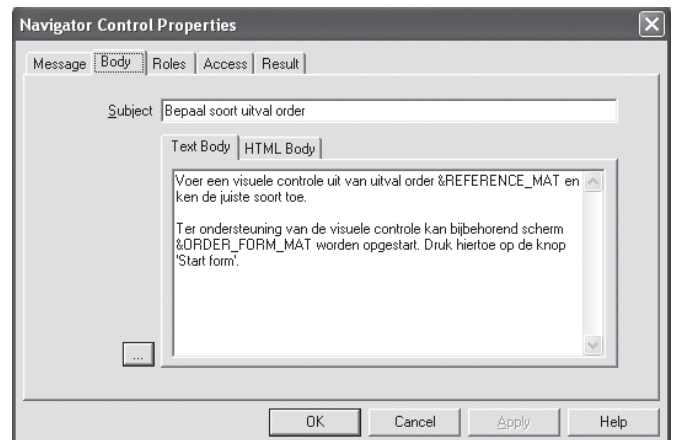
## Asynchroon

Als we dit modelleren in Builder ziet dat eruit zoals in afbeelding 1 uit het vorige artikel (zie *Optimize 2004* nr. 4, pagina 21). Hierin zagen we in de Navigator naast Item Type 'Standard' het Item Type 'Uitval Order'. We zien dat een Item Type bestaat uit een vast aantal soorten workflowelementen: Attributes, Processes, Notifications, Functions, Events, Messages en Lookup Types. Voorts zien we dat het proces 'Verwerk uitvalorder' is geselecteerd en gevisualiseerd in het weergegeven procesdiagram. Simpel gesteld kan een proces bestaan uit een viertal verschillende soorten processtappen:

- een *subproces* (zoals hier 'Verwerk complexe uitvalorder'). Een dergelijk proces is in feite een proces als alle anderen, maar is een detailproces van een hoger gelegen proces. Als een subproces wordt afgewikkeld ligt het parentproces stil, en wordt hervat als het subproces gereed is.
- een *notificatie* (zoals hier 'Bepaal soort order' en 'Signaleer klantenservice'). Een notificatie meldt aan de gebruiker dat hij een bepaalde actie moet ondernemen. Dit uit zich veelal in een taak op een takenlijst. De gebruiker moet hierop actie ondernemen en de taak afmelden. De tekst die in de notificatie wordt getoond, is in Builder uitgemodelleerd in het workflowelement Message. Zo kan dezelfde Message in verschillende Notificaties worden weergegeven (zie afbeelding 2



Afbeelding 2. Eigenschappen van workflowelement 'Notification'



Afbeelding 3. Eigenschappen van workflowelement 'Message'

en 3 voor de eigenschappen van de workflow elementen Notificatie en Message). Bij het afmelden van de taak door de gebruiker moet eventueel een resultaat opgegeven worden. De List of Values voor dit resultaat is in Builder vastgelegd als Lookup Type. Zo moet bij notificatie 'Bepaal soort order' als resultaat 'Procesgestuurd', 'Handmatig' of 'Backorder' worden opgegeven. Pas dan kan de taak worden afgemeld. In afbeelding 4 is weergegeven hoe dit eruit ziet met de Oracle Forms user interface.

- een *function* (ook wel: function activity, zoals hier 'Maak backorder'). Een function is een stuk applicatielogica welke door de workflow engine automatisch wordt uitgevoerd. Omdat de aanroep door de engine is gestandaardiseerd, moet de applicatielogica een aantal vaste parameters hebben. Deze staan uitgebreid beschreven in de *Workflow Guide*. Een function is van het type PL/SQL, external Java, of external overig. External geeft aan dat de workflow engine hier standaard niet in voorziet, en dat er bij installatie een zogenaamde 'Agent' in de lucht moet worden gebracht om dergelijke aanroepen af te handelen. Een agent is een proces dat 'luistert' of er een

aanroep wordt gedaan, en dan de juiste logica aanroept. Voor aanroepen van Java wordt er een agent meegeleverd, voor overige externe functions (bijvoorbeeld shellscripts voor het besturingssysteem waarop de database draait) moet er zelf een worden geschreven (zie afbeelding 5 voor de eigenschappen van het workflow element Function).

- een *event* (die wordt geraised, dan wel waarop moet worden gewacht, zoals hier 'Receive uitval order'). Een event is een gebeurtenis welke is geregistreerd in het Business Event System van Oracle Workflow. Vanuit een workflow proces (door het eenvoudig definiëren van een Event processtap) of vanuit de eigen applicatie met behulp van een aanroep naar één van de Workflow API's, kan een Event worden geraised. Door in een ander workflow proces een receive Event processtap te definiëren, kunnen de respectievelijke flows dus asynchroon gekoppeld worden (zie afbeelding 6 voor de eigenschappen van het workflow element Event).

Het moge duidelijk zijn dat de geschetste case uitermate simpel is en een simpel procesdiagram te zien geeft. In de praktijk zal een proces vele malen ingewikkelder zijn. In die gevallen zullen er bijvoorbeeld loops te zien zijn, en knooppunten waar verschillende flows samenkomen. In sommige gevallen zal pas worden vervolgd als alle inkomende flows zijn aangekomen (AND) in andere gevallen zal de eerstkomende volstaan (OR). In het 'Standard' Item Type zijn de standaard functions te vinden waarmee dergelijke functionaliteit kan worden gemodelleerd.

## Procesbesturingslaag

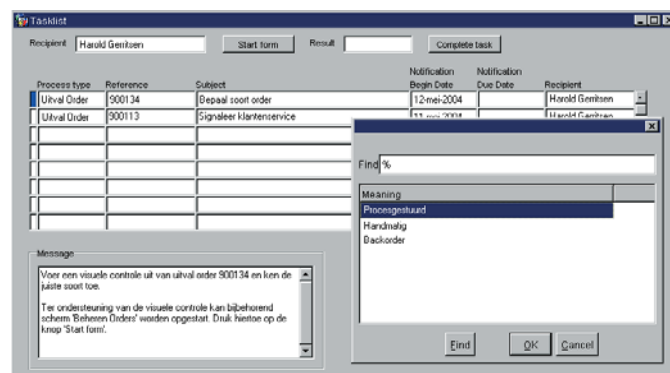
Als we reguliere applicaties beschouwen volgens het drielagen model (user interface, business logica en data) dan behelst de aanpassing het introduceren van een vierde laag, de procesbesturingslaag, tussen de user interface en de business logica. Zoals we hebben gezien bevatten de gedefinieerde procesflows processtappen van het type 'Function'. De uit deze processtappen aangeroepen logica heeft een vaste aanroep, specifiek voor Oracle Workflow. Daarom willen we deze logica in een separate laag (in de vorm van naamgeving, database schema et cetera). De logica in de procesbesturingslaag maakt vanzelfsprekend weer gebruik van logica in business logic layer. Door deze gelaagdheid wordt het eenvoudig mogelijk de Workflow toepassing te verwijderen indien nodig, of bijvoorbeeld te vervangen door een ander Workflow product. Het aanroepen vanuit de ene laag naar de andere wordt niet alleen vanuit de Workflow Engine (het uitvoeren van 'Function' processtappen) geïnitieerd. Soms willen we het andersom; dat de applicatie de workflow processen triggert. In dat geval is het onmogelijk om de bestaande applicatie onaangetaast te laten. De applicatie zal moeten worden uitgebreid met triggers (of, bij gebruik van het

CDM RuleFrame, met Change Event Rules) die met behulp van een Workflow API Events raisen. Ook hierbij is het van belang eigenlijk geen logica in de triggers of Change Event Rules op te nemen, en de daadwerkelijke functionaliteit wederom in de procesbesturingslaag te definiëren. Door een centraal punt te definiëren welke door alle applicatietriggers wordt aangeroepen, ontstaat er een duidelijk koppelpunt tussen de twee lagen. Dit fungeert dan als event handler van alle relevante events in de bestaande applicatie.

## Contextgevoelig

Indien gewenst kan ook de user interface laag worden aangepast aan het gebruik van workflow. De noodzakelijke aanpassing is uiteraard het opnemen van de takenlijst in het menu. Daarnaast kunnen er echter ook schermen worden ontwikkeld die specifiek zijn afgestemd op de uit te voeren taken. Stel dat er in de bestaande applicatie al een scherm 'Beheren Orders' bestaat waarmee alle eigenschappen van orders kunnen worden onderhouden, inclusief de eigenschap 'ontvangstdatum orderformulier'. Stel vervolgens dat er met behulp van workflow een notificatie 'Registreer ontvangstdatum Orderformulier' wordt gedefinieerd. Dan zou er voor gekozen kunnen worden om een specifiek scherm te ontwikkelen voor het vastleggen van alleen de ontvangstdatum van de order. Vanuit de notificatie kan dit scherm dan worden aangeroepen. Dergelijke taakgerichte schermen zijn bijzonder snel te ontwikkelen maar doen door de beperkte toepassing (doorgaans klein van afmeting, met weinig velden) wat amateuristisch aan. Niettemin geldt hier dat het uitermate wordt gewaardeerd door onervaren krachten, die minder bekend zijn met de exacte taak.

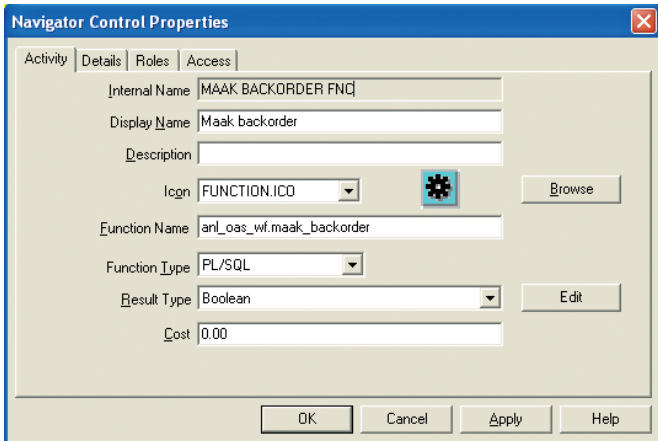
Een tussenvorm wordt wél veel toegepast. De user interface van de bestaande applicatie wordt dusdanig aangepast dat de bestaande schermen zich contextgevoelig gaan gedragen. Zodra ze worden aangeroepen vanuit de takenlijst, dan worden de voor de taak relevante velden (die in het vorige voorbeeld in een apart scherm waren opgenomen) geaccentueerd. Dit kan



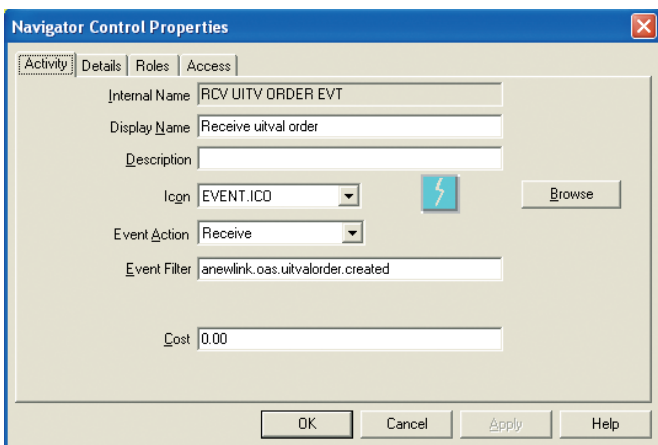
Afbeelding 4. Takenlijst met (maatwerk) webforms interface en List of Values voor de resultaatwaarde

---

# **Adv. Masterclass**



Afbeelding 5. Eigenschappen van workflowelement 'Function'



Afbeelding 6. Eigenschappen van workflowelement 'Event'

bij het gebruik van Oracle Forms bijvoorbeeld door de achtergrondkleur van de relevante velden te veranderen. Worden dezelfde schermen vanuit het menu aangeroepen, dan gedragen ze zich als vanouds. Ook bij deze aanpassingen (de user interface van de bestaande applicatie) is het verstandig aandacht te besteden aan een zuivere architectuur. De kennis die de schermen moeten hebben van welke velden in welke taak relevant zijn, is in feite workflowkennis. Deze kennis moet in de procesbesturingslaag worden gedefinieerd om de schermen algemeen bruikbaar te houden.

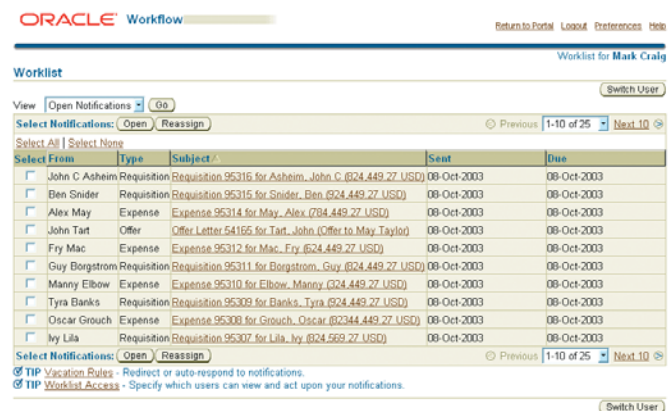
## Oracle WorkFlow for Java

Zoals in de inleiding is genoemd, is Oracle Workflow precies tien jaar op de markt. Al die tijd is het niet ingrijpend aangepast. Workflow Builder is uitsluitend beschikbaar op het Windows platform en de workflow engine is gebaseerd op PL/SQL en draait in de Oracle database. Hierdoor is het aanroepen van PL/SQL vanuit de engine eigenlijk een native call en is het aanroepen van Java een externe aanroep. Aangezien Oracle al enkele jaren een stevige Java strategie voert rond al haar producten is dat eigenlijk niet meer te verkopen. Om die reden komt Oracle naar verwachting eind december met een nieuw

product: Oracle Workflow for Java (OW4J). Dit bestaat eveneens uit een clienttool en een runtime component. Het clienttool zal in twee smaken beschikbaar zijn: als een add-in voor JDeveloper (zodat de modeler op alle Java-enabled platforms kan draaien) en als een HTML variant welke met behulp van een browser kan worden gedraaid. Oracle mikt hiermee op ontwikkelaars respectievelijk procesanalisten.

De runtime component is in feite een doorontwikkelde versie van de Java Agent die in de huidige versie 10g (Workflow 2.6.3) wordt meegeleverd. De nieuwe versie is echter in staat om naadloos Java classes, alsmede Enterprise Java Beans en Webservices uit te voeren, en JMS messages te versturen en te ontvangen. Theoretisch is dat met de huidige Java Agent ook mogelijk, alleen moet je dat allemaal zelf doen in de aangeroepen 'external Java Functions'. Met OW4J kun je al deze varianten voortaan expliciet modelleren en onderneemt de engine zelf de specifieke aanroep-actie. Een ander verschil is dat OW4J niet een Agent is van de workflow engine in de database, maar dat het zelf een engine is en draait in de applicatieserver. Dat betekent dat je voortaan de keuze hebt uit twee alternatieven voor het uitvoeren van workflowprocessen.

Als je met de modeler hebt gemodelleerd kun je het deployen in de database engine of in de OW4J engine (hiertoe wordt het model bewaard als XML-file die in OW4J kan worden ingelezen). Klanten die voornamelijk in Java ontwikkelen zullen OW4J toejuichen, klanten met PL/SQL gebaseerde systemen zullen waarschijnlijk de databasevariant blijven gebruiken. Naast verschil voor de ontwikkelaars is er ook verschil voor de eindgebruikers van de workflow enabled applicaties. Het belangrijkste verschil is dat de HTML schermen rond de workflow functionaliteit, zoals de takenlijst, er bijzonder fraai uitzien. Voor de kenners: alles heeft de Oracle Browser Look And Feel (BLAF) en is gerealiseerd met behulp van UIX (zie voor een indruk afbeelding 7). Bovendien kan in de takenlijst op allerlei eigenschappen worden gesorteerd en geselecteerd. Voor geïnteresseerden is er inmiddels ruime informatie over OW4J



Afbeelding 7. OW4J takenlijst met UIX interface

beschikbaar op OTN, inclusief een Flash demo van de modeler (zie de referenties aan het eind van dit artikel).

## Resumerend

Een aanzienlijk aantal bedrijven in Nederland is recent begonnen met Oracle Workflow implementaties. In het merendeel van de gevallen betreft het hier een vernieuwingsslag om bestaande applicaties workflow enabled te maken. Hoewel workflow-gebaseerd werken op de eindgebruikers een aanzienlijke impact kan hebben, is de drempel vanuit het aandachtsgebied software-ontwikkeling niet zo hoog. De licentiekosten zijn voor de bestaande klanten meestal reeds inbegrepen bij de gebruikte suites, en de ontwikkelinspanning voor de workflow besturing is relatief beperkt. Wel is het zo dat workflow besturing zich niet leent voor elk bedrijfsproces en elke bedrijfssituatie. In de in dit artikel genoemde situaties, is toepassing van workflow echter een uitstekende keuze. Voor hen die geïnteresseerd zijn in het fenomeen Oracle Workflow bevat dit artikel de nodige informatie om zich daar verder in te verdiepen.

## Referenties

Meer informatie over OW4J en andere versies van Oracle Workflow:

*Oracle Technology Network*

<http://otn.oracle.com/products/ias/workflow/index.html>

Totale productdocumentatie Oracle Workflow via hoofdindex:  
*MetaLink*

<http://metalink.oracle.com/>

vervolgens Advanced Site Search op Document ID '67183.1'

## Harold Gerritsen

is Principle Consultant bij A New Link bv ([www.anewlink.nl](http://www.anewlink.nl)). Hij heeft meer dan veertien jaar ervaring in het adviseren over effectief inzetten van Oracle technologie (e-mail: [h.gerritsen@anewlink.nl](mailto:h.gerritsen@anewlink.nl)).