

De aankondiging van JavaFX als groot nieuws tijdens de keynote van JavaOne zal de berichtgeving over de conferentie hoogstwaarschijnlijk domineren, maar er was uiteraard veel meer te horen en te zien. De thematische indeling in een negental tracks geeft iedere bezoeker de mogelijkheid zich te verdiepen in één speciaal thema, of juist naar wens overal iets uit te selecteren. In dit artikel belicht Bert Ertman een aantal toekomstige ontwikkelingen in enterprise development.

# JavaOne 2007

## Spotlight op Enterprise Development

**H**oewel er nog weinig rechtstreeks wordt gesproken over Java EE 6, de opvolger van het inmiddels één jaar oude Java EE 5, staat er toch al het een en ander in de steigers. Sommige onderdelen zijn nog puur in de ideeënfase, voor anderen zijn al een expert group en bijbehorende JSR geformeerd. In de enterprise track van de JavaOne was soms tussen de regels door en soms heel expliciet al van alles te horen over de toekomstige nieuwe enterprise variant. Naar alle waarschijnlijkheid zal de nieuwe Enterprise Java versie, die in 2008 het levenslicht zal gaan zien, een afhankelijkheid hebben met een recente versie van de Virtual Machine. De komst van steeds meer annotation gebaseerde technologie maakt dat om gebruik te kunnen maken van dergelijke oplossingen een overstap noodzakelijk is. Om die overgang zo geruisloos mogelijk te maken werden de aanwezigen op verschillende momenten tijdens JavaOne, waaronder in de technische keynotes, aangespoord om nu toch echt de overstap naar Java SE 6 te maken. Er is volgens Sun geen enkele reden om op versie 5 of nog ouder te blijven steken. Of in de praktijk gemakkelijk gehoor zal worden gegeven aan deze aanmoediging is echter nog maar de vraag.

### EJB 3.1 en JPA 2.0

Het hart van het enterprise Java platform zal worden gevormd door de nieuwste versies van de Enterprise JavaBeans (EJB) specificatie en de Java Persistence API (JPA). In tegenstelling tot voorgaande versies van de specificaties is ervoor gekozen om deze beide onderdelen nu uit elkaar te trekken en onder te brengen in aparte JSR's. Hiermee lijken de dokters erin geslaagd deze schijnbare Siamese tweeling van elkaar te schei-

den. Een ontwikkeling die reeds was te voorzien, gezien de duale toepassing van JPA zowel binnen als buiten de container.

De eerstvolgende versie van EJB oftewel de zogenaamde Simplified Components API zal versie 3.1 gaan worden. De Java Persistence API maakt een grotere sprong en zal in een versie 2.0 worden gebracht. Het is nog niet uitgesloten dat tussen door een versie 1.1 van JPA zal verschijnen, waarin een aantal issues van de huidige versie (1.0) zijn opgelost. Hoe deze er precies uit gaat zien en welke wijzigingen daar in zitten, is op dit moment nog niet duidelijk. Op het moment van schrijven zijn de JSR nummers nog niet bekend. De verwachting is dat deze omstreeks juni bekend zullen worden.

EJB 3.1 gaat verder op het pad dat is ingeslagen door EJB 3.0: een verdere versimpeling van het componentenmodel waarbij het contract tussen de runtime omgeving (container) en de ontwikkelaar steeds gunstiger uitvalt voor laatstgenoemde. Mogelijke toevoegingen die op de rol staan zijn onder andere uitbreidingen op de Timer Service, Singleton Session Beans, nieuwe lifecycle events en verbeterde ondersteuning voor asynchrone operaties. Tevens wordt er gesproken over meer discutabele features zoals ondersteuning voor stateful web services en een aanpassing van het contract voor local business interfaces.

Voor JPA versie 2.0 gelden een tweetal aandachtspunten: het terugbrengen van de gebieden waar nu nog moet worden uitgeweken naar non-portable implementaties en een heel arsenaal aan nieuwe features. Onder het eerste aandachtspunt

### Bert Ertman

is bestuurslid van de NLJUG en in zijn dagelijks werk als IT-architect werkzaam bij Info Support BV. Weblog: <http://blogs.infosupport.com/berte>

wordt het verder standaardiseren van de in JPA 1.0 opgenomen optionele features verstaan. Daar waar de huidige specificatie nog gaten laat vallen zullen deze verder waterdicht worden gemaakt. Dit is momenteel een heet hangijzer; er werd zelfs een aparte sessie gewijd aan tips en trucs voor het verminderen van afhankelijkheden met specifieke persistence providers. Uiteraard staat er een hele set aan nieuwe features op de rol. Het merendeel ervan als direct resultaat van feedback die de expert group heeft ontvangen uit de community. Ook hier geldt maar weer eens: de feedback mogelijkheden die je als Java-ontwikkelaar hebt, werken echt! Voor alle genoemde features geldt dat het nog niet zeker is of ze ook uiteindelijk allemaal geïmplementeerd gaan worden. De feature requests zijn in een aantal thema's samen te vatten: meer flexibele modellering mogelijkheden, uitbreiden van de mogelijkheden van object/relational mapping, uitbreidingen op de Query Language en nog een paar losse onderwerpen.

Meer flexibele modellering mogelijkheden zouden kunnen ontstaan door het mogelijk te maken om meerdere niveau's van embeddable objects te hebben. Momenteel is het niet mogelijk om een Embeddable object onderdeel te laten zijn van een ander Embeddable object. Denk bijvoorbeeld aan een embeddable object Postcode die onderdeel is van een embeddable object Adres, dat op zijn beurt weer onderdeel is van een entity Persoon. Andere wensen binnen dit thema zijn het ondersteunen van het persisteren van sorteer-informatie van geordende lijsten en het toestaan

van meervoudige toegangsniveaus tot entity data. Laatstgenoemde feature moet het mogelijk maken om in plaats van of field based toegang of property based toegang door middel van annotaties te kunnen specificeren dat sommige stukken van een entity field based toegang ondersteunen en andere gedeeltes van een entity property based toegang. Zeker in combinatie met embedded objects kan dit een nuttige feature zijn.

Een ander thema is het uitbreiden van de mogelijkheden van object/relational mapping. Bijvoorbeeld door het ondersteunen van foreign key mappings in unidirectionele one-to-many relaties. Nu wordt voor het oplossen van een one-to-many relatie standaard een tussentabel gebruikt. Bij bidirectionele relaties kan gebruik gemaakt worden van een foreign key. In het geval van een unidirectionele relatie zal je vanuit de database redenerend liever een foreign key gebruiken dan een tussentabel. De huidige implementatie is daarmee vrij onnatuurlijk en daarom hoogstwaarschijnlijk ook aan verandering onderhevig in de volgende versie van de specificatie. Op het gebied van entity inheritance geldt dat de nu optionele strategie single-table-per-class hoogstwaarschijnlijk een volwaardig onderdeel van de specificatie zal worden.

Ook de Java Persistence Query Language (JPQL) wordt op een aantal plaatsen uitgebreid. Gedacht wordt aan uitgebreidere ondersteuning van de SELECT clause. Daarnaast gaat het waarschijnlijk mogelijk worden om te restricteren in polymor-

**Ook de Java Persistence Query Language (JPQL) wordt op een aantal plaatsen uitgebreid**



Nasdaq is een tot de verbeelding sprekend voorbeeld van Java in the enterprise

fisme, dat wordt toegepast op query's. Hierdoor wordt het mogelijk om een query uit te voeren op specifieke objecten uit de overervingstructuur van een bepaalde entity en niet op alles wat afstamt van een bepaalde entity. Ook criteria gebaseerde query's zullen onderdeel gaan worden van de specificatie. Frameworks als Hibernate ondersteunen deze mogelijkheid al veel langer en daarmee is het een veelgevraagde feature voor JPA.

Tenslotte zijn er nog een aantal losse onderwerpen die mogelijk hun plek in de specificatie gaan vinden. Bijvoorbeeld toegang tot de unfetched toestand van detached entities, versimpeling van

Gosling bleek op de NLJUG-bijeenkomst enthousiast over Java Realtime



de application managed persistence context en integratie met web services conversaties. Tenslotte wordt er nagedacht over ondersteuning van het Bean Validation framework (JSR 303). Hiermee kunnen validaties op business logica worden gespecificeerd.

### Bean Validation API

Een logische uitbreiding op de verzameling aan annotatie gebaseerde frameworks en API's is de komst van JSR 303, de Bean Validation API. Deze specificatie beoogt een alternatief te bieden voor het bekende probleem dat code voor het valideren van constraints op input data gedupliceerd staat door de hele applicatie heen. Met de komst van annotations in het Java-platform is een trend ingezet die steeds meer declaratief programmeren mogelijk maakt. Constraints controleren door middel van annotations vormt daarmee een aanvulling op de toch al behoorlijk declaratieve programmeermodellen van simplified components en persistente entities. Het doel van deze JSR is om een standaard neer te zetten voor het gebruik van annotations voor data validatie binnen het Java platform. De API biedt een oplossing voor POJO's in het algemeen, maar vanwege de gewoonte om business logica in enterprise applicaties onder te brengen in session beans is het gebruik van EJB 3 simplified components en deze API een voor de hand liggende combinatie.

Een tweede doelstelling van de API is om tevens een oplossing te bieden voor het automatisch triggeren van validaties op objecten als ze worden gebruikt. Hiervoor moet een AOP-achtig mechanisme worden gebruikt. Met de introductie van EJB 3 is zo'n middel echter voorhanden in de vorm van de EJB Interceptors. Hiermee is een manier gevonden om zogenaamde cross-cutting acties (gedrag) te definiëren op specifieke business logica of workflow logica. Door het koppelen van de annotations en de interceptors kunnen validaties automatisch worden getriggered zonder dat daar nog code aan te pas hoeft te komen.

### Web Beans

Een ontwikkeling die al ongeveer een klein jaar bestaat, maar waar slechts mondjesmaat nieuws over naar buiten komt is Web Beans (JSR 299). De specification lead van deze JSR is Gavin King, bekend van het Hibernate persistence framework en tevens bedenker van JBoss Seam, een applicatie framework voor enterprise applicaties. Laatstgenoemde is tevens de oorsprong van deze JSR en samen met het beste uit frameworks als Guice, Struts Shale en Oracle ADF is het de bedoeling een compleet en eenvoudig programmeermodel neer te zetten voor Java EE applica-

ties, waarmee presentatie, business logica en persistence een natuurlijk geheel vormen. Een belangrijk kenmerk van Web Beans is het gebruik van EJB session beans als managed beans in een JavaServer Faces (JSF) applicatie. Daarnaast definieert het framework een aantal nieuwe context scopes, speciaal bestemd voor het modelleren van toestand in conversaties en long running business processen. Hierbij biedt het een set van annotaties voor het manipuleren van variabelen in stateful scopes van web applicaties. Tevens verzorgt het de persistentie van de toestand aanwezig in variabelen binnen deze contexten. Tenslotte, probeert Web Beans om samen de expert groups van JSF en de Common Annotations API te komen tot een geïntegreerde set van annotations voor JSF.

### Java Realtime

Bij het bouwen van complexe transactieverwerkende enterprise systemen denkt niet iedereen meteen aan Java Realtime. Interessant was dan ook om juist bij de introductie van de nieuwe Java Realtime System (Java RTS) 2.0 een bedrijf als NASDAQ de case study te zien presenteren. NASDAQ verwerkt gemiddeld zo'n 152.000 transacties per seconde gedurende een handelsdag. Bij piekmomenten kan het volume stijgen tot ongeveer het tienvoudige hiervan. We praten hier dus over een serieus complex systeem, waarvan de schaalbaarheidsuitdaging enorm is te noemen. Contingency problemen in threading of geheugen management van een dergelijke applicatie die deze een fractie van een seconde of langer onbruikbaar maken leveren daarmee potentieel zeer grote problemen op. Het gebruik van de realtime versie van Java is daarmee een vereiste om beter voorspelbare executie van logica te krijgen. Commerciële versies van Java RTS 2.0 zullen vanaf eind mei beschikbaar komen.

### EclipseLink project

Tussen al het NetBeans geweld was het soms moeilijk nog iets over Eclipse te horen of te zien te krijgen. Toch is er wel iets interessants te melden. In de keynote van Oracle kwam het EclipseLink project ter sprake. Op dit moment is EclipseLink nog slechts een projectvoorstel en is er nog slechts beperkt belangstelling voor vanuit andere bedrijven. De toegevoegde waarde is nog niet helemaal duidelijk, maar het voorstel omvat het introduceren van een universeel en uitbreidbaar framework voor data toegang. Het gaat hierbij om de interactie met relationele databases, XML en andere Enterprise Informatie Systemen (EIS). Het framework zal worden gebaseerd op bekende en gestandaardiseerde API's als de Java Persistence API, Java API for XML Binding (JAXB), Java Connector Architecture (JCA) en Service Data Objects (SDO). De bedoeling is om

samen te werken met de OSGi expert group voor het creëren van een OSGi Persistence Services framework.

### Conclusie

Hoewel op dit moment Java EE 6 nog in nevelen lijkt te zijn gehuld - de officiële JSR 313 heeft zelfs de status 'withdrawn' - staan er toch al diverse zaken in de steigers. Naar verwachting zal omstreeks JavaOne 2008 deze specificatie, of tenminste een final draft ervan, het levenslicht gaan zien. Hoewel voor de meeste features en API's besproken in dit artikel geldt dat er nog niet eens een public draft beschikbaar is, geeft het toch alvast een blik in de enterprise-keuken van Java. Met de informatie die nu bekend is, staat eigenlijk al vast dat Java EE 6 een aantal spectaculaire nieuwe eigenschappen zal hebben die de toolset van de enterprise ontwikkelaar ongetwijfeld zullen verrijken. Veelzeggend detail is wel dat pas zeer recent slechts een paar mainstream applicatieserver vendors met een Java EE 5 compliant versie op de markt zijn verschenen. Het zou dus nog wel eens even kunnen duren voordat we bovenstaande zaken in productie zullen gaan tegenkomen.

### Referenties

- Bean Validation (JSR 303) <http://jcp.org/en/jsr/detail?id=303>
- EclipseLink Project <http://www.eclipse.org/proposals/eclipse-link/>
- Enterprise JavaBeans 3.0 (JSR 220) <http://jcp.org/en/jsr/detail?id=220>
- JBoss Seam <http://www.jboss.com/products/seam>
- Realtime Java 2.0 <http://java.sun.com/javase/technologies/realtime/index.jsp>
- Web Beans (JSR 299) <http://jcp.org/en/jsr/detail?id=299>

**Met de informatie die nu bekend is, staat eigenlijk al vast dat Java EE 6 een aantal spectaculaire nieuwe eigenschappen zal hebben**