

Volgens Wikipedia geeft de uitdrukking *dertien in een dozijn* aan dat iets heel gewoon is. Als het niet eens opvalt dat men dertien exemplaren van iets aanziet voor twaalf, dan zal het wel niets bijzonders zijn.

JavaOne 2008 – dertien in een dozijn?

Java 8, geen fratsen

Het getal dertien wordt in onze cultuur door velen gezien als een ongeluksgetal. Opvallend is dat het woord ongeluksgetal zelf ook dertien letters heeft, maar dat terzijde. In veel vliegtuigen wordt in de nummering van de zitplaatsen rij dertien overgeslagen. In de Formule 1 start niemand met dit nummer. Er is zelfs een officiële naam voor angst voor het getal dertien, namelijk *triskaidekafobie*. Maar wat heeft het getal dertien eigenlijk met JavaOne?

In-between

JavaOne 2008 betekent de dertiende verjaardag van Java als programmeertaal en platform. Hoewel de echte ontstaansgeschiedenis van Java nog iets verder teruggaat, wordt 1995 met de JDK 1.0 bèta 2 release over het algemeen gezien als de feitelijke ‘geboorte’ van Java. Het jaar 2008 is voor wat betreft grote nieuwe releases van het Java-platform niet heel bijzonder te noemen. Wat dat betreft zou je kunnen spreken van een ‘in-between’ jaar. Volgend jaar immers staan er nieuwe releases gepland van zowel Java SE 7 en Java EE 6. Eigenlijk hadden we vorig jaar eenzelfde situatie, alleen dacht Sun toen de wereld te moeten verbazen met de aankondiging van JavaFX.

JavaFX op herhaling

In het afgelopen jaar hebben we bijzonder weinig gezien van JavaFX en dat heeft dan ook onvermijdelijk tot heftige kritiek geleid. Demo's waarbij een vierkantje in een cirkel werd veranderd, tel ik voor het gemak even niet mee. De keynotes van dit jaar waren dus duidelijk een rebound van afgelopen jaar. De grote vraag is waarom Sun eigenlijk opnieuw al die moeite doet om de aandacht van de 15.000 aanwezigen op JavaFX te vestigen? Het mag duidelij-

lijk zijn dat de loze beloften van afgelopen jaar behoorlijk afgestraft zijn. Het met groot tromgeroffel aankondigen van een revolutionaire nieuwe technologie, maar vervolgens iedereen ruim een jaar laten wachten op een fatsoenlijke bèta kan echt niet door de beugel. Het sarcastische commentaar uit de community is bij Sun blijkbaar luid en duidelijk overgekomen. Er moest dus duidelijk iets goed gemaakt worden. Ook zijn er de afgelopen maanden diverse discussies geweest waar Sun werd gevraagd naar de meerwaarde van JavaFX ten opzichte van concurrerende technologieën van Microsoft en Adobe. ‘Performance’ was één van de steevaste antwoorden. Dat moest dus duidelijk even worden aangetoond. Vooral de demo waarin ongeveer 200 HD-video's op een ronddraaiende bol werden gerenderd (op een Macbook Pro!) was erg indrukwekkend. Dit soort demonstraties tonen zeker de power en performance aan waartoe JavaFX in staat is. Ook het uit de webpagina kunnen pulken van een JavaFX-applet en deze vervolgens naar de desktop slepen, terwijl de applicatie blijft draaien en de browser wordt gesloten, is een eigenschap waarop de concurrentie voorlopig nog het antwoord schuldig moet blijven. Een andere interessante ontwikkeling was het tonen van een JavaFX-applicatie draaiend in een door Sun geschreven emulator van Google Android. Voor alle demo's geldt natuurlijk wel dat het volkomen onduidelijk was welke versie van de JVM hieronder draaide en hoe de API's er precies uit zien.

Ondanks een serie indrukwekkende demo's blijft de vraag dan ook wanneer JavaFX daadwerkelijk beschikbaar komt. Helaas is het nog wachten tot ver in de herfst van dit jaar voordat de officiële bèta

SDK beschikbaar komt. Om die datum ook daadwerkelijk te halen zal er volgens goed ingewijde bronnen nog hard moeten worden doorgewerkt. Sun doet er wel alles aan, getuige de interne overplaatsing van een aantal kopstukken op compilergebied. De keynotes waren zodanig met JavaFX doordrenkt dat je bijna zou vergeten welke andere zaken er dit jaar de revue hebben gepasseerd.

OSGi

Eén van de opvallende zaken die in het technische deel van de keynotes naar voren kwam, was dat OSGi dan toch eindelijk zijn officiële intrede in het Java-landschap gaat maken. Tot nu toe werd het eigenlijk alleen in het domein van de rebel frameworks en veelal niet-JCP geleide initiatieven toegepast, maar nu lijkt daar verandering in te gaan komen. De nieuwste versie van Glassfish is voor wat betreft zijn packaging en dependency-structuur helemaal op OSGi gebaseerd. Het Java Module System (JSR 277) - dat tot nu toe eigenlijk eerder een concurrent voor OSGi was - lijkt te worden omgetoverd tot een soort van packaging-abstractie waarvan OSGi één van de mogelijke implementaties is.

Sensors

Net zoals mobile Java een aantal jaren nodig heeft gehad om echt het mainstream ontwikkelaarspubliek te bereiken is er nu een blijvende trend op het gebied van sensor-based computing waar te nemen. De afgelopen jaren was er al de SunSPOT en dit jaar was een sensor development kit van Sentilla één van de show-devices die door de aanwezigen voor een zacht prijsje kon worden aangeschaft. Green computing en CO2 Neutral zijn in Californië momenteel populaire thema's. Om het sensor-



Foto: Die de Man

verhaal wat kracht bij te zetten waren het complete conferentiecentrum en de nabije omgeving op strategische plaatsen voorzien van sensors waarmee energieverplaatsing en CO2 werden gemeten. In de keynote van James Gosling op vrijdagochtend kwamen nog meer voorbeelden van real-life oplossingen met sensortechnologie aan de orde.

Opensolaris

Tussen de bedrijven door was er ook de nodige (sluik)reclame voor OpenSolaris. Qua marketingpower, aantal ontwikkelaars en innovatie is dit product duidelijk één van de speerpunten in de huidige strategie van Sun. De recente acquisitie van MySQL kwam eigenlijk maar nauwelijks aan bod. Tijdens keynotes kwam het vrijwel niet ter sprake en je moest

goed zoeken naar technische sessies over dit onderwerp. De overige keynotes van medehoofdsponsors Intel, AMD en Oracle waren niet veel meer dan groepsknuffelen en benadrukken hoe fantastisch de samenwerking met Sun wel niet is. In het geval van Oracle betrof het een ouderwetse vendor-pitch. Hieruit viel weinig nieuws te halen. Jammer genoeg gaf Oracle verder geen openheid van zaken voor wat betreft de plannen over het opslokken van BEA en welke gevolgen dit zal hebben voor de productstacks van beide bedrijven.

Thema

Het thema van de conferentie "Java + You" was duidelijk een voortborduursel op thema's uit voorgaande jaren waarbij telkens de boodschap terugkomt dat Java

overal is en je in het alledaagse leven eigenlijk niet meer zonder Java kunt. Dit thema werd kracht bijgezet door het tonen van een gelikte video waarin alledaagse activiteiten werden gelinkt aan Java. Sun wil zich duidelijk hip voordoen en af en toe kun je je niet helemaal aan de indruk onttrekken dat Sun zich daarbij vooral laat inspireren door wat Apple doet. Bij het bestuderen van de commentaren van analisten en bloggers lijkt er wat onvrede te zijn over het feit dat deze conferentie niet bulkt van de nieuwe aankondigingen. Maar is dat eigenlijk wel zo slecht als iedereen doet geloven? Mag je van een technologie die dertien jaar oud is en waarmee bijzonder veel business-critical systemen worden gebouwd verwachten dat jaar na jaar de ene na de andere vernieuwing nog over elkaar heen moet rollen? Is het eigenlijk niet gewoon heel erg gezond dat er dit jaar niet weer 'the next best thing' is gelanceerd? Geeft dat niet gewoon de volwassenheid van het Java-platform aan? Natuurlijk is er op allerlei gebieden volop innovatie aan de gang, maar laten we die vooral de tijd geven om tot rijping te komen.

Waardevol

Misschien is het zinvoller te kijken naar de waarde die uit de JavaOne van dit jaar is mee te nemen. Een duidelijke verbetering was waar te nemen in de kwaliteit van de technische sessies. Het feit dat Sun voor de beoordeling en selectie van de ingezonden papers heeft gebruikgemaakt van een flink arsenaal aan experts uit de industrie was overduidelijk te merken aan het kwaliteitsniveau van het overgrote deel van de sessies. Dit wil niet zeggen dat daarmee alles inhoudelijk ook even goed was, maar over het geheel genomen lag het niveau duidelijk hoger dan in voorgaande jaren.

Sexy

Een veelgehoorde uitspraak tijdens discussies op en rond de conferentie was "Java als taal is niet meer sexy". Als argumentatie hoorde je dan: "talen als Groovy, Ruby, Python en Scala hebben deze coole feature en Java heeft dat niet, dus is het niet sexy meer". "Java moet deze feature ook krijgen, anders overleeft het niet". Grappig, een aantal jaren geleden toen de strijd tussen Microsoft .NET en J2EE los barstte, ging het over



Foto: Drié de Man

single platform, multi-language (.NET) versus multiplatform, single language (Java). Hebben we toen niet met ons allen hartelijk gelachen om Microsoft dat met een handvol programmeertalen op de Common Language Runtime wel even dacht het marktaandeel van Java weg te snoepen. We sneerden dat multi-language een prachtig idee was, maar dat in de praktijk waarschijnlijk iedereen wel gewoon C# zou gaan programmeren en dat .NET in essentie dus eigenlijk ook gewoon een single language platform was. Nu, jaren later, blijkt dat nog voor een groot deel waar te zijn ook. Gaan we met de huidige explosie aan dynamische talen op het Java-platform niet ook gewoon die kant weer op?

Puberen

Natuurlijk zijn er met dynamische talen in combinatie met krachtige frameworks en scaffolding enorme productiviteitswinsten te behalen, maar je kunt ook doorslaan. Je kunt eigenlijk alleen maar hoofdschuddend luisteren naar gerenommeerde sprekers die serieus verkondigen dat het

nu ook mogelijk is om enterprise applicaties volledig in JavaScript te schrijven. Of zou dit nu juist kenmerkend zijn voor een dertienjarig kind dat al puberend zijn eigen onnavolgbare keuzes aan het maken is? Er tegenin gaan heeft dan eigenlijk alleen maar een averechts effect. Hoewel, in plaats van ons heel begrijpend op te stellen is het misschien toch leuk om als tegenbeweging een volgende versie van het Java-platform te betitelen als: *Java 8, geen fratsen*.

Conclusie

Achter de schermen is ondertussen de discussie gestart hoe het verder moet met de toekomst van de JavaOne-conferentie. Duidelijk mag zijn dat het met een publiek van 15.000 ontwikkelaars met een eigen mening het nooit helemaal goed kan zijn. Er gaan stemmen op om de BoF's – de zogenaamde Birds of a Feather-sessies waarbij je in een informele setting in contact kunt komen met de geestelijk vaders van een bepaalde technologie of API – een prominentere plaats te geven. Deze zijn nu alleen later op de avond ingepland.

De meeste JavaOne-bezoekers hebben er dan al een intensieve dag opzitten en moeten deze waardevolle sessies vaak verstek laten gaan. Een andere verbetering zou zijn om meer hands-on labs te doen. Hiervoor is nu slechts een beperkt aantal plaatsen beschikbaar en deze sessies zijn vaak al ver voor aanvang van de conferentie volledig volgeboekt. De huidige koers voor het selecteren van de papers is in ieder geval een blijvertje. Ook volgende keer zal een panel van internationale experts zich buigen over het programma.

Ten slotte, als we dan toch een eindoordeel moeten vellen over de afgelopen conferentie, dan zou je eigenlijk kunnen zeggen: “*JavaOne 13 was gewoon de betere versie van JavaOne 12. Alleen dan een jaar te laat*”. «

Ing. Bert Ertman

is werkzaam als Technology Manager van het Competence Center Java bij Info Support BV. Naast zijn dagelijkse werkzaamheden is hij bestuurslid van de NLJUG en recentelijk verkozen tot Sun Java Champion. Reacties? Mail naar berte@infosupport.com.