

James Gosling is de vader van Java. Hij ontwierp de taal-structuur en implementeerde de compiler en virtual machine. Gosling werkt sinds 1984 bij Sun en is nu Vice President and Sun Fellow. Tijdens JavaPolis sprak hij meer dan een uur exclusief met Java Magazine.

James Gosling:

Wanneer niemand wil verliezen, verliest iedereen

De problemen in deze wereld ontstaan niet door stupiditeit, maar door bureaucratie

Wat doet u eigenlijk, behalve evangeliseren?

Mijn job is een rommeltje, ik doe van alles. Technisch ben ik de CTO van de client software Group. We zijn eigenaar van alle Java VM's, van smart cards tot de grote machines. Ik was ook CTO van de tools-groep, maar we hebben een deel van het mobiele gedeelte verplaatst, zodat ik heb besloten me een tijdje op iets anders te concentreren. Ik raak nu betrokken bij discussies over architectuur en business-planning. Ik ga naar veel te veel saaie bedrijfsvergaderingen, ik maak reisjes als deze en bezoek overal ontwikkelaars, en geef presentaties.

U wordt vaak de vader van Java genoemd. Nu is Java al twaalf en wanneer kinderen ouder worden, moet je langzamerhand afstand gaan nemen. Lukt dat een beetje?

Ja, ik schreef de eerste compiler en de eerste JVM, helaas is al die code allang weg. Het heeft zijn voor- en nadelen. Er was een tijd waarin ik radicale veranderingen op een middagje kon doorvoeren. Tegenwoordig raak ik min of meer gefrustreerd wanneer ik zoiets wil. Aan de andere kant heb je die grote gemeenschap die er heel veel energie inbrengt, en mensen gebruiken die energie voor de meest verbazingwekkende dingen in Java. Ik houd die dingen bij door ze uit



te proberen, want ik ben niet goed in het lezen van dingen.

U leest geen documentatie?

Ik lees het wel, maar het dringt beter tot me door wanneer ik iets bouw. Om het helemaal te begrijpen moet ik het gebruiken.

U bent vaak bezig met architectuurvragen, gebruikt u daarvoor modelleringstechnieken als UML?

Nee, nooit. We hebben een aantal van die tools, maar ik gebruik ze zelden of nooit. Ik heb het model in mijn hoofd. Ik zou er meer mee moeten doen.

U schrijft nog steeds zelf code?

Ik schrijf nog steeds behoorlijk veel, maar het heeft iets raars. Het is voor mij heel moeilijk me vast te leggen op enig delivery-schedule. Dus wat ik meestal doe, is het uitvoeren van grappige en vreemde kleine experimenten.

Gebruikt u NetBeans?

Ik gebruik NetBeans voor alles. Ik ben altijd een groot fan geweest. Ik download alle pre-release copies, ik stuur het team felicitatiebrieven en hate-mail, afhankelijk van de prestaties. Maar ik ben echt heel erg gelukkig met het team, ze doen heel goed werk.

U hebt in een ander interview eens gezegd dat sommige dingen die u zou willen veranderen in de taal te gecompliceerd zouden zijn voor de doorsnee ontwikkelaar. Is dat nog steeds een probleem?

We hebben zeker het algemene probleem dat we onze ontwerpen willen beginnen met use cases voor werkelijk fraaie en gecompliceerde oplossingen. De ontwerpen voor J2EE-cases gaan allemaal over wat grote banken of financiële instituten zouden kunnen doen in hun transactieverwerkende systemen. In de begindagen richtten we ons echt op de vraag of we zaken op grote schaal konden doen. We maakten ons geen zorgen over hoe we het ons gemakkelijk moesten maken, we concentreerden ons op het mogelijk maken van dingen. In de Java-wereld neigden we naar de situatie waarin de moeilijke dingen mogelijk zijn en de gemakkelijke dingen moeilijk. Dit in tegenstelling tot de Microsoft-manier, waar eenvoudige dingen ook gemakkelijk zijn, maar de moeilijke dingen onmogelijk. We hebben er hard aan gewerkt om de eenvoudige

dinge simpel te maken en een deel van dat werk zit in de API's, maar voor een heel groot deel in de tools.

Is het opnemen van Java Studio Creator in NetBeans daar een voorbeeld van?

Er is nu een breed spectrum aan tools in NetBeans om veel dingen gemakkelijker te maken, van het maken van webpagina's uit Ajax-componenten tot dingen om het gemakkelijk te maken telefoonspelletjes te schrijven. We hebben veel van die gespecialiseerde domeinfaciliteiten in NetBeans gericht op grote groepen van ontwikkelaars.

Dus de tendens is om de vereenvoudigen meer in NetBeans dan in de taal door te voeren?

NetBeans is zeker het platform om alle tooling in te integreren. Sommige kwesties worden geholpen door de API's, andere door de tools. Heel af en toe vind je iets dat je met aanpassingen van de taal zelf kunt oplossen, het is min of meer van alles wat. Je zou kunnen denken dat we het grootste gedeelte in tools hebben opgevangen, maar bij JEE5 maakte de specificatie het een stuk gemakkelijker. Bij de Glassfish-applicatieserver zijn de deployment en installatie stukken eenvoudiger, maar de tools maken de applicatieontwikkeling ook gemakkelijker. Veel dingen komen samen, een groot deel van de eenvoud heeft te maken met het gemak van het debuggen. NetBeans kan appservers starten, kan deployen en zet breakpoints

in applicaties. Als je een applicatie verandert en je wilt opnieuw deployen, dan krijg je veel hulp.

We hadden het net over zaken waar u niet meer zoveel invloed op heeft. Hoe zit dat met de JCP?

Ach, iedereen beweegt zo snel als mogelijk, maar verschillende gemeenschappen hebben verschillende snelheden. De financiële sector zit ergens in het midden, we hebben de webapplicatie-mensen die heel snel bewegen, en aan de ander kant van de schaal vind je mensen die embedded dingen doen. Die hebben jaren nodig om dingen door te denken. We hebben klanten die een decennium over een evaluatie doen. Sommige zijn ook echt heel erg moeilijk. Zo was er een bedrijf dat een vrij compleet prototype had gemaakt, maar een ander deel van het bedrijf zei: 'nee je kunt dat niet zo doen'. Een van de producten van dat bedrijf werd geacht dat te doen. Ze moesten dus veranderen en het ding van die andere afdelingen gebruik, en helemaal de mist in gaan. Dat duurde ongeveer drie jaar, waarna ze terug kwamen en zeiden: 'weet je dat prototype nog van drie jaar geleden, kunnen we ... Maar goed, dat is wel een industrie waar materiaal wordt gebouwd met een levensduur van 50 jaar.

Java wordt in de embedded wereld steeds meer gebruikt. Hoe komt dat?

Dat is een ontwikkeling die door een paar zaken bepaald wordt. Eén daarvan is dat



Java beter geschikt wordt voor embedded systemen. Een andere zaak is dat embedded systemen volwassen worden, en steeds meer capaciteiten nodig hebben. De standaard manier om dingen te doen schaaft gewoon niet.

Wat bedoelt u daarmee?

Niet dat de apparaten groter worden, maar ze moeten zich steeds intelligenter gedragen. Vele gebruiken standaard AI-technieken (artificiële intelligentie, DdM), en AI-software en conventionele real-time software passen niet bij elkaar. In de Java-wereld gaan die twee dingen heel goed samen. Er is echter ook een hele verzameling van dingen waarmee de traditionele embedded software niet goed kon omgaan, waarvan networking het belangrijkste voorbeeld is. De klassieke embedded systemen zaten in een gesloten doos en bouwen geen enkele verbinding op; neem medische apparaten die met papier of film werkten. Tegenwoordig is het allemaal bits. In het Braziliaanse gezondheidssysteem is nergens meer papier, ze hebben hun röntgenmachines gedigitaliseerd en de artsen kijken naar de beelden op schermen. Ze hadden ook het voordeel van de omgekeerde wet van de remmende voorsprong, er was geen sprake van gevestigde bedrijven met belangen. Wat ook echt hielp was dat de UN kwam met een set van privacyrichtlijnen betreffende gezondheidsinformatie. Er was veel oppositie, omdat men zei dat je het niet kon implementeren, het zou te rigide zijn. Op de een of ander manier

'Het grootste probleem van Corba is dat het **niet cool** is, tegenwoordig'

heeft de UN 16 miljoen dollar bij elkaar gekregen dat ze gaven aan een paar mensen in Brazilië om te zien of het gebouwd zou kunnen worden. Dat lukte. Het pakte echt goed voor hen uit.

De Google-plannen met betrekking tot mobiele telefonie doen een beetje denken aan de begindagen van Microsoft met Gates, die dankzij een licentieslordigheidje over de rug van IBM een imperium opbouwde.

Ik denk dat het echt anders is, de Google-lui zijn in wezen good guys. Ze hebben alleen ontzettend veel geld en geen enkel idee wat ze ermee moeten. Het is vaak heel moeilijk uit te maken wat ze nu in 's hemelsnaam van plan zijn. Misschien weten ze het ook zelf niet. Ze zijn in wezen een reclamebureau, daar verdienen ze hun geld mee. Ze zijn altijd op zoek naar andere manieren om geld te verdienen, maar ze hebben nooit iets gehad dat op een niet-triviale manier op dezelfde schaal succesvol was als advertenties. Het is moeilijk te zeggen wat hun plan-

nen zijn. Ze zijn zeker al jarenlang grote Javasupporters, ze gebruiken intern enorm veel Javacode. Met Android hebben ze een klein beetje code beschikbaar gesteld...

Ja, maar er is toch een echte SDK?

Ja, die is er, maar er zijn geen telefoons die er gebruik van maken, en al evenmin netwerken. Als je een van hun partners vraagt wat het partnerschap inhoudt, dan zeggen ze: 'O, we hebben ons akkoord verklaard met de persverklaring en we hoopten op die manier wat inside informatie te krijgen over wat Google doet'. Ze hebben niets, ze hebben geen commitment gedaan op welk gebied dan ook. Google doet nogal onduidelijk over hun businessplannen, als ze die al hebben. Je kunt niet zeggen of dit een plan is dat al ze jaren hadden, of dat het is iets van de laatste tijd, het is gewoon niet te zeggen.

Waarom treedt Sun niet toe tot de Open Handset Alliance?

Eh, ze hebben ons niet gevraagd. We hoorden er pas van na de persverklaring. We praten al jaren met Google over dingen die zijn gerelateerd aan mobiele telefonie. De Alliance is ook een heel losse verbinding. De dingen die ze getekend hebben, verplichten helemaal nergens toe. De mensen met wie ik gepraat heb, hebben alleen ondertekend uit nieuwsgierigheid. De leden van de Open Handset Alliance tasten net zozeer in het duister als wijzelf.

U klinkt niet echt bezorgd?

Nee, ik zou niet zeggen dat ik bezorgd ben. Het zou zorgwekkend kunnen zijn, het zou ook geweldig kunnen zijn. Er zijn gewoon geen gegevens.

Google heeft wel wat merkwaardige stappen gezet met betrekking tot het omzeilen van de Java ME-licenties, zoveel is zeker.

Ja, ze hebben rare dingen gedaan. Wat dat nu inhoudt, ik heb geen idee, dat zou je Google zelf moeten vragen.



Java is uitgegroeid tot een enorm platform. Hebt u nog steeds een beeld van alles wat er gebeurt, in ieder geval tot op een zeker niveau?

Nee, er is gewoon te veel. Zelfs bij Sun begrijp ik maar een klein deel van wat er gebeurt. Er is zoveel.

Welk deel van Java heeft uw bijzondere belangstelling, nog steeds de JVM?

Compilertechnologie, ik probeer zoveel mogelijk de JVM-technologie bij te houden. Ik ben ook een beetje een *science geek*, dus ik breng veel tijd door met klanten als Cern en ik houd van embedded.

U was onlangs nog bij Cern, nietwaar?

Ja, twee weken geleden (december). Ik heb de luxe tour door de Large Hadron Collider gekregen en ze namen me mee naar beneden door de grote tunnels, dat maakt me heel gelukkig. Ze zijn nog maar een paar maanden verwijderd van het moment waarop ze het inschakelen. Dan zal het allemaal radioactief zijn en kun je er niet meer gaan kijken. Het is allemaal zeer indrukwekkend, alleen al het beeld van al die kabels.

Welke software gebruiken ze, ik neem aan dat ze niet alleen Java doen?

Ze gebruiken van alles. Ze hebben inder tijd wat tests gedaan, en de C-optimizers waren toen sneller dan de Java-optimizers. Maar tegenwoordig wint Java van C in veel situaties, als je kijkt naar de Sysmark.

En C#?

Ik was nogal verbaasd dat C# het niet zo goed deed, het zat niet eens in de buurt. Microsoft is al een tijdje een soort raadsel voor ons.

In welk opzicht?

Wel, het lijkt alsof ze alle energie in het uitbrengen van Vista gestopt hebben. Vista is echter zo'n nachtmerrie dat ze nu al hun energie stoppen in het bestrijden van die ramp. Ze lijken echt afgeleid, alsof ze van de kaart verdwenen zijn. Ze stoppen er ook veel energie in om verticaal te zijn. Wij aan de andere kant hebben ons erop geconcentreerd om alleen een platform voor horizontale dingen te zijn, en zelfs daarin hebben we ons beperkt. We zijn tenslotte ongeveer tien keer kleiner dan

Microsoft. Maar Microsoft is veel minder actief en agressief dan ze waren.

In uw speech noemde u heel kort Corba, voor mijn gevoel een soort van SOA-dynosauriër. Komt u dat nog veel tegen?

Netwerkprotocollen zijn tegenwoordig een nachtmerrie. Er zijn er zoveel, en ongelukkigerwijze hebben ze alle hun sterke en zwakke punten. Gelet op CPU-performance en bandbreedte is XML tamelijk verschrikkelijk. Gelet op interoperabiliteit is het tamelijk geweldig. Iets als Corba is wat betreft interoperabiliteit best redelijk. Zijn capaciteit om grote hoeveelheden data heen en weer te schuiven is echt geweldig. Het grootste probleem van Corba is dat het tegenwoordig niet in de mode is. De meeste mensen die ik heb ontmoet en die Corba gebruiken, hebben een tijd nodig voordat ze het toegeven. Het is merkwaardig dat de computerindustrie zoveel lijkt op de modewereld. Dingen kunnen in en uit de mode zijn. In menig opzicht zijn veel XML-standaarden een stap terug vergeleken met de protocollen die we daarvoor gebruikten. Ze zijn lang-

'Het is grappig dat zoveel in de computerindustrie lijkt op de modewereld'

zaam aan het terugkomen op het niveau dat we hadden.

Is die teruggang alleen op het gebied van performance?

Ja, maar ook capaciteiten. Iets als een *remote object system* bestaat gewoon niet in die webservice-wereld. RMI had een soort van mobiel code-concept, je kon een object naar een remote machine sturen en de code, die alle security zou afhandelen, zou meegestuurd worden. Veel mensen gebruiken dat nog steeds, maar het is niet in de mode, dus geven ze het alleen onder druk toe. Het is nu niet meer zo erg, maar als je een paar jaar geleden wilde dat mensen in jouw bedrijf investeerden, moest je zeggen: we doen iets met XML. Ik zou willen dat de durfkapitalisten niet zo'n kuddementaliteit hadden, maar dat hebben ze helaas wel.

Hoe denkt u verder over SOA en de hype er omheen? Er zitten toch ook bedenkelijk kanten aan op het gebied van performance?

Dat is een deel van het probleem. Wat is SOA nu? Als je naar de afkorting kijkt: een geweldig idee. Het is dertig jaar lang een geweldig idee geweest! Weliswaar onder andere namen, zoals Architecting Systems As Services, gewoon een heel goed gefundeerd architectuurprincipe. Je kunt SOA via Corba doen, via RMI, over naakte sockets die bytes lezen en wegschrijven voor mijn part. Het is voornamelijk afhankelijk van de kerk waartoe je behoort. Aan de ene kant ben ik ervan overtuigd dat SOA absoluut de juiste keuze is. De huidige versie van SOA-protocollen en de zaken die mensen eromheen bouwen voor tools als NetBeans, die werken best goed. Aan de andere kant zijn er toepassingen van het concept waar ik grote vraagtekens bij zet.

Als je de durfkapitalisten nu voor je wilt interesseren, wat moet je dan doen?

Als je een iPod-killer kunt maken dan

zouden ze in je investeren. Maar ze zouden je ook uylachen, omdat je niet Steve Jobs bent. Alles wat groen genoemd kan worden, heeft de interesse van de venture capitalists. Groene technologie draait nu steeds meer om het ontwerpen van apparaten die slim genoeg zijn. Een hybride auto, zonder een krachtig computersysteem dat alle beslissingen op het gebied van de aandrijving en krachtverdeling neemt, werkt gewoon niet. Je ziet heel veel systemen over de hele wereld die steeds slimmer worden en mensen vinden manieren om er veel waarde uit te halen. Neem het Braziliaanse gezondheidssysteem, waar ik zonet over sprak. Hun recepten staan niet meer op papier, maar zijn database-entries. Daardoor werd medicijnfraude ineens vrijwel geëlimineerd. Dat alleen al leverde vijf keer meer geld op dan het IT-systeem gekost had. Cool! Het is een heel simpel idee. Ik heb er bij General Electric Medical een praatje over gehouden. Het was echt triest, want je kon zien dat de mensen in het publiek aan de ene kant boos waren en aan de andere kant wilden huilen. Het is niet zo dat de Brazilianen uitzonderlijk briljant waren, of dat ze heel revolutionaire dingen gedaan hadden, de meeste dingen lagen gewoon voor de hand. Ze deden gewoon de juiste dingen. De problemen in deze wereld ontstaan niet door stupiditeit. Ze worden opgeroepen door die combinaties van bureaucratische regeringen en bedrijfsbelangen die het onmogelijk maken de juiste dingen te doen. Het gaat er om dingen simpel te houden. Ongelukkigerwijze worden eenvoudige dingen heel erg ingewikkeld wanneer bedrijven en regeringen en organisaties met elkaar botsen. Dingen als: wat is een telefoonnummer, wat een adres? Als je met mensen praat die databaseschema's maken voor gezondheidssystemen, dan merk je dat alleen al de schema-mapping van de ene database naar de andere, iets

simpels als een adres, al bijna onmogelijk is. Voor de Brazilianen was het simpel, die konden gewoon zeggen: dit is het schema. Het gaat dan om compromissen tussen specifieke belangen. Wanneer niemand wil verliezen, verliest iedereen. Dat is echt triest. In de moderne wereld zijn er eigenlijk geen echt moeilijke technische problemen meer. Je kunt vrijwel alles betrekkelijk eenvoudig bouwen met de technologie die we hebben om systemen te bouwen. Het moeilijkste probleem is nog steeds om mensen zover te krijgen dat ze het met elkaar eens zijn, zodat er een zinvol datamodel ontstaat.

U bent op de leeftijd waarop veel mensen gaan nadenken over hun leven, soms nog een nieuwe carrière beginnen, u werkt al heel lang als ontwikkelaar.

Ik verdien mijn geld als ontwikkelaar sinds ik veertien was, en ik ben nu 52, dus reken maar uit. Het is ergens dwaas, maar ik beleef er nog steeds enorm veel lol aan.

Dus over tien jaar bent u nog steeds softwareontwikkelaar?

Zeker, zelfs wanneer het geen baan zou zijn, dan zou ik het ook doen. Ik houd ervan het te doen. Dat is een van interessante dingen. De meeste mensen die goed zijn in het ontwikkelen van software, doen het niet omdat ze een baan nodig hebben, maar omdat het leuk is. Ik heb geen enkel idee waarom het leuk is, maar het is leuk.

Maar zijn er geen andere dingen die u had kunnen doen, of zou willen doen?

Ja, allerlei dingen. Ik houd ervan dingen te bouwen, in zijn algemeenheid. Ooit was ik vrij goed in het repareren van brandblussers, en ik zou waarschijnlijk heel gelukkig hebben kunnen zijn als ik de rest van mijn leven brandblussers zou hebben gerepareerd. Ik zou de beste reparateur van brandblussers in Alberta hebben kunnen zijn. «

Dit interview werd gehouden in december 2008, tijdens JavaPolis. Het gedeelte van dit gesprek dat ging over Google, en met name Google Android, is op de website van Java Magazine voorgepubliceerd.