

Binnen de Java-gemeenschap wordt vaak gesproken over het opensourcen van Java. De houding van Sun in deze is niet altijd even doorzichtig. Tijd voor een gesprek met Simon Phipps, wiens werk voor een belangrijk deel bestaat uit het uitleggen van het businessmodel achter de open source-strategie van Sun.

thema

Come in we're open

De open source strategie van Sun

Intellectueel eigendom, meestal op zijn Engels IP genoemd, vertegenwoordigt een bepaalde geldwaarde, tenminste zolang er nog vraag naar die software is. Zo gezien is het open source van software kapitaalvernietiging. Of niet?

Phipps: 'Het is van belang te begrijpen dat Sun's historie opgebouwd is op het werken met partners en gemeenschappen. Sun's eerste product in 1981 was een Workstation opgebouwd uit onderdelen volgens industriestandaarden, en dat een open source-besturingssysteem gebruikte. (Berkeley, DdM). Dat is het patroon geweest in de afgelopen 24 jaar. We hebben met open technologie, en gemeenschappen gewerkt, omdat we ons realiseren dat we niet alles alleen kunnen doen. Dat is karakteristiek voor alles wat Sun doet. Waardoor verspreidde Java zich zo snel in 1995? Ik werkte toen voor IBM en maakte deel uit van het team dat het Java-platform bij IBM introduceerde. Wat het voor ons mogelijk maakte om Java te adopteren, was het feit dat het geleverd werd met de volledige source-code, die was in 1995 op internet beschikbaar. Dat was lang voordat open source in de mode kwam. Daarom verspreidde het zich zo snel, de hele source-code was op het internet beschikbaar vanaf de eerste dag en is er altijd gebleven. Je kunt nog steeds de complete source-code van Java downloaden en veranderen, zolang je je aan de licentie houdt. De licentie zegt dat wanneer je je werk wil distribueren, het compatibel moet zijn. Als je een release van Java wilt maken, is de enige voorwaarde om die gratis te distribueren dat jouw implementatie compatibel moet zijn met de specificaties. Zo is het geweest sinds 1995. Wat we wel beschermen, zijn de investeringen van onze klanten en daarom sluiten we geen compromis over de compatibiliteit op het Java platform.'

Aan de ene kant heb je ontwikkelaars die vinden dat alles gewoon open source moet zijn, aan de andere kant sommige klanten en aandeelhouders die het liefst zou zien dat alles onder zeer strikte controle van Sun blijft.

Phipps: 'Het vinden van een compromis is heel moeilijk. Toen de open source licentie ontstond, maakte het helaas geen ruimte voor het behoud van comptabiliteit.

Als dat wel het geval was geweest, zou Java nu open source geweest zijn. We werkten dus samen met Apache om te zien wat we moeten veranderen in de Java-licenties en in de JCP om het apache mogelijk te maken om een open source implementatie van het Java platform te maken. Sinds twee jaar geleden is het mogelijk om open source implementaties van Java te maken. Er zijn nu drie bedrijven die dat doen: JBoss, Jonas en Jeronimo. Alle drie zijn het open source J2EE-implementaties, en ze zijn gecommiteerd aan compatibiliteit. Dat is ideaal, het beste van alle werelden: ze zijn in staat open source te produceren en ze committeren zich aan comptabiliteit. Ik voorspel dat binnen een jaar open source J2SE-implementaties zullen ontstaan. Zo zijn de Java user groups in Brazilië samen met de Braziliaanse overheid bezig een open source-J2SE te produceren.'

Het opensourcen van Solaris was niet eenvoudig.

Phipps: 'In 1968 realiseerde niemand zich dat we ooit die rechten zouden moeten kopen. Advocaten en technici zijn vijf jaar bezig geweest om ons ervan te vergewissen dat we over alle rechten beschikten op Solaris te opensourcen. Je kunt niet met je vingers knippen en oudere software opensourcen. Wij weten hoeveel moeite het kost om op een gewetensvolle manier software te opensourcen. Nu kijk ik naar Java en bestaande applicaties en zeg: zou ik liever hebben dat mijn engineers werken om Java 5 te opensourcen, of zou ik liever zien dat zij werken aan Java 6? Onze klanten vertellen ons dat ze liever zien dat aan Java 6 werken. De enige mensen die ongelukkig zijn, zijn degene die een Java-monopolie willen creëren. Belangrijk is dat Sun op geen enkele manier tegen open source implementaties van Java is. In feite ondersteunt Sun ze zelfs door ze beurzen te geven voor compatibiliteitstesten.'

Dus op lange termijn zal Java open source worden?

Phipps: 'Op lange termijn verwacht ik dat de versie die van de expertgroep van de JCP afkomstig is, waarschijnlijk onder traditionele licentie beschikbaar wordt, terwijl

de versie daarvoor beschikbaar zal zijn van open source groepen. MySQL werkt ook zo, de commerciële versie is n, en n-1 is GPO.'

Het gevaar van open source lijkt mij dat er afscheidingen ontstaan (forking) die elkaar in de weg staan.

Phipps: 'Forking is geen probleem, de vrijheid om te forken is de kracht van open source, als ik het fout doe, ben jij altijd vrij om het goed te doen. Het probleem is niet community-forking, het probleem ontstaat wanneer een monopolist ervoor kiest om te forken uit marktverwelingen. Binnen de Java-ruimte zijn er hele grote ondernemingen die een fork zouden kunnen creëren

zonder dat de gemeenschap daartoe noodzaak ziet. Daarom moeten wij voorzichtig zijn. Jini, Juxta, open Solaris, we hebben niet eens gepraat over Sun-grid. Dat is gebaseerd op de open source grid engine, de grid engine draait bovenop open Solaris of bovenop Linux. Onze gehele business-strategie is gebaseerd op open source. Sun begrijpt open source, het gaat om het werken met community's en partnerschappen met andere slimme mensen om een win/win situatie te creëren. Sun is gecommiteerd aan partnerships en community's want het profiteert door het verkopen van de services en de apparatuur aan die gemeenschappen en partner-side die nodig hebben om te slagen, dat is ons business-model.'

Spring maakt J2EE eenvoudiger

Gesprek met Rod Johnson en Alev Arends

Spring is een open source Java-framework, dat beoogt J(2)EE programmeren eenvoudiger te maken. Het ontstond uit de code die gepubliceerd werd met het boek Expert One-on-One J2EE Design and Development geschreven door Rod Johnson. De auteur van het framework en het boek legt samen met zijn Nederlandse interface21 collega Alev Arends uit waarom Spring zo populair geworden is.

Expert One-on-One J2EE Design and Development werd gepubliceerd in oktober 2002. Het werd geleverd met een sample-applicatie die geschraagd werd door 14.000 regels code. Deze bood oplossingen voor veel voorkomende problemen die Johnson keer op keer tegenkwam in J2EE-projecten. Er was dus veel belangstelling in het opensourcen ervan en in het begin van 2003 begon het leven van Spring als open source-project. In maart 2004 werd de Spring 1.0 final release uitgegeven, in mei 2005 de 1.2 final release. Welke zijn nu de voordelen van Spring?

Johnson: 'Een belangrijk probleem in veel J2EE-projecten was dat men steeds opnieuw dezelfde functionaliteit bouwde. In de beginjaren bouwden mensen bijvoorbeeld hun eigen logging infrastructuur en geleidelijk is er een beweging ontstaan naar goede generieke frameworks die generieke oplossingen voor generieke problemen aanbieden. Natuurlijk levert dat een heel groot productiviteitsdividend op. Het betekent namelijk dat developers in ondernemingen zich concentreren op de kernactiviteiten van dat bedrijf en de business-logica die specifiek is voor dat bedrijf, in plaats van het oplossen van problemen die ieder ander heeft en die maar een keer opgelost zouden hoeven te worden op een effectieve manier. Natuurlijk krijg je, wanneer je je als community focust op het leveren van goede oplossingen voor veel voorkomende problemen, veel meer ervaring

en feedback rond die oplossing dan wanneer ieder bedrijf zijn eigen oplossing schrijft. Spring levert bedrijfsoplossingen die onafhankelijk zijn van het bedrijf of de branche in kwestie. Daardoor kunnen developers zich concentreren op kernproblemen en de infrastructuur overlaten aan het framework. Dat is waarschijnlijk het eerste belangrijke voordeel.

Het tweede is, dat Spring software-ontwikkeling in feite een stuk simpeler maakt. Spring is ontworpen om nogal wat complexiteit binnen het framework op te nemen. Traditioneel in standaard-J2EE, ben je verplicht je transactiemangement te doen, je bent verplicht het J2EE-model dat een behoorlijk niveau van complexiteit oplegt op het programmeermodel en ook het development-proces. Met Spring ligt de nadruk op het mogelijk maken de applicatie op te bouwen uit Pojo's. Spring levert die voordelen op zonder dat je mogelijkheden prijs moet geven. Het levert een complete oplossing die simpel is in zijn traditionele aanpak.'

TESTEN Johnson: 'Een derde groot voordeel is dat Spring een veelomvattende oplossing biedt om de testmogelijkheden te vergroten. Allereerst wordt door een Pojo-gebaseerd programmeermodel te leveren, unit-testing erg vergemakkelijkt, zodat het bijvoorbeeld mogelijk wordt om effectief testdriven development te doen. Spring gaat echter nog een stap verder door krachtige integratie testondersteuning te bieden. Je kunt bijvoorbeeld tests schrijven die een development staging-database gebruiken, je code uitvoeren tegen de echte enterprise services die het moeten doen, je hele Spring-configuratie testen, zonder de noodzaak op een applicatieserver te deployen. Dat kan veel sneller zijn dan een traditionele aanpak, waar een heel groot risico van incorrecte code was omdat het meestal te langzaam was om het