

Laatst was ik bij een softwarehuis dat het woeste plan had opgevat om de volgende versie van een van hun producten te ontwikkelen in Java. Omdat ze nog weinig ervaring met Java hadden werd mij gevraagd te helpen brainstormen: "The Guru is In" noemen wij dat met de nodige zelfoverschatting.

## Puur of melk?

De "Guru" kwam dus "In", overzag de probleemstelling en werd onmiddellijk vrolijk. Het Javaplatform zit immers boordevol features en componenten die het zeer geschikt maken om allerhande soorten informatiesystemen mee te bouwen. In combinatie met aanpalende technologieën als CORBA kunnen complexe state-of-the-art oplossingen worden gerealiseerd: de droom van iedere IT-architect. Enthousiast ging ik van start: Java Swing voor de GUI, Java Beans voor componenten, Java Spaces voor communicatie, CORBA voor remote objecten, HTML en een Java web server voor de on-line documentatie en Enterprise Java Beans voor complexe transactiegeoriënteerde server interacties.

Toen kwam echter de "Reality Check": de primaire markt van het product was, is en blijft het Windows platform. Niks platformonafhankelijkheid, geen Java Network Computers, geen Unix servers. Gewoon, Windows NT servers en Windows 9x clients. Toen ik over de eerste schok heen was begon ik ook de voordelen hiervan in te zien. Microsoft heeft intussen allerlei interessante uitbreidingen aan Java toegevoegd die de integratie van Java-applicaties met het Windows besturingssysteem aanzienlijk verbeteren. Door de platformkeuze voor Windows kunnen we bij die klant al die extra technologie

inzetten. Met alle voordelen van dien: sneller ontwikkelen, betere performance, Windows standaard userinterface, goede aansluiting op andere Microsoft BackOffice producten. Dan maar geen "100% Pure Java" applicaties. Trouwens, Windows draait op meer dan 90% van alle computers ter wereld. "Who needs platform independence anyway?"

Bij een andere klant heb ik voortdurend discussie over het wel of niet gebruiken van zogenaamde "native code compilers". Dit zijn Java-compilers die in plaats van de gebruikelijke platform- en architectuurneutrale tussencode meteen machinetaal genereren voor het doelplatform. De voordelen zijn evident: de performance is beter en het is moeilijker om "reverse engineering" op de applicatie los te laten. Er zitten echter ook nadelen aan vast. De applicatie is niet langer platform- onafhankelijk en sommige Java-features worden niet ondersteund. Een gruwel dus in de ogen van de echte Java-puristen. Maar ja, de klant wil snelheid en de eerlijkheid gebiedt te vermelden dat veel "100% Pure Java"-applicaties niet vooruit zijn te branden.

De meeste leveranciers zijn niet geïnteresseerd in Java pur sang. Ze willen klanten bedienen, oplossingen bieden, en, daardoor, verkopen en dus omzet en winst maken. Om een reeks van redenen passen leveranciers Java

aan of breiden het uit. Microsoft is het meest sprekende voorbeeld. Ter gunste van de klant, want die wil snelheid en integratie met Visual Basic en DCOM. Droevig maar waar. "100% Pure Java" is Sun's inspanning om te voorkomen dat leveranciers Java verdunnen of aanlengen met features die het leven van de klant veraangemen. Alleen applicaties die voldoen aan Sun's richtlijnen kunnen dit kwaliteitsmerk krijgen. Software ontwikkelaars worden aangespoord om "100% Pure Java" te schrijven en klanten wordt aangepraat dat ze "100% Pure Java"-applicaties moeten hebben om hun kostbare investeringen veilig te stellen. Het merk zou moeten betekenen dat een applicatie werkt met alle compatibele Java-implementaties. Om er echter zeker van te zijn dat de applicatie echt loopt leveren de meeste leveranciers van in Java geschreven applicaties maar de complete Sun-Java-omgeving mee. Hierdoor lopen ze niet het risico dat de klant toevallig de "verkeerde" Java implementeert, want: "Compatible is not Identical".

Het is met Java eigenlijk net als met chocolade. De liefhebbers propageren puur, maar de meeste mensen prefereren melk.

*Jos Visser*

is werkzaam als goeroe bij de Open Solution Providers. Zijn e-mailadres is [josv@osp.nl](mailto:josv@osp.nl).