

# Ontwikkelaars verbeteren de wereld

Ik heb er al een heleboel keren over lopen drammen: Java is een objectgeoriënteerde taal. Helaas is dat niet voor alle ontwikkelaars even evident. Sterker nog, ik vind de meeste met Java gebouwde systemen nauwelijks objectgeoriënteerd. Natuurlijk zijn ze dat wel. Met Java kun je niets anders dan objecten bouwen, dus Java systemen zijn altijd objectgeoriënteerd. Toch herken ik er zelden in wat de kern van objectoriëntatie zou moeten zijn: Dynamische 'business-' objecten met hun eigen gedrag en state. Het nogal behoudende, traditionele ontwikkelmodel van J2EE heeft hier helemaal een eind aan gemaakt. J2EE objecten zijn domme data-objecten geworden óf zijn processe-rijge sessie-achtige spaghetti. De reguliere J2EE architectuur staat echte objectoriëntatie nauwelijks toe.

Misschien moeten we maar concluderen dat ons verhaal niet goed genoegd was toen wij al die ontwikkelaars vertelden wat OO was. Toen wij probeerden uit te leggen waarom objectoriëntatie beter is dan die eeuwige starre onbeheersbare tweedeling tussen de database en de procedurele programmatuur, hebben wij blijkbaar geen goed punt kunnen maken. Misschien hebben de OO docenten er eigenlijk geen snars van gesnapt, dachten ze dat ze gewoon hun oude verhaal konden blijven vertellen onder de noemer van een nieuwe hype.

Eén van de dingen die we altijd verteld hebben is dat OO zo'n goed concept is omdat je er zo mooi de werkelijke wereld mee kunt modelleren. Iedereen is dus objecten aan het maken die de werkelijke wereld nadoen. Dat is dus mooi mis. Met andere technieken (vooral datamodel-leren) kun je precies zo goed de werkelijkheid modelleren als met objecten. We moeten daarvan af. Objecten hebben namelijk een hele andere mogelijkheid die met geen enkel ander ontwikkelmodel te benaderen is: met objecten kun je de wereld verbeteren! Echt waar, je kunt die saaie, inefficiënte, onhandige werkelijke wereld naar je eigen hand zetten. Da's mooi, want van die echte wereld hebben we allang genoeg. Wanneer we een nieuw systeem maken, *moeten* we de wereld verbeteren: een klein stukje software toevoegen aan de werkelijke wereld zodat die wereld een beetje mooier wordt.

Waarom kan ik de wereld niet verbeteren met andere dan OO concepten? In een ontwikkelmodel waar data en proces gescheiden zijn, moet je twee dingen tegelijk verbeteren. Hoe kan ik de wereld verbeteren door de data te veranderen? Wat voor effecten heeft die verandering op de processen? Wat gebeurt er wanneer ik de processen verbeter? Wat heeft dat dan weer voor consequenties voor het datamodel? Je leeft met twee modellen die samen één werkelijke wereld modelleren. Waar moet je beginnen met het ver-

beteren van de werkelijkheid? Dat maakt het gewoonweg onmogelijk om de wereld te verbeteren. Processen kún je bovendien helemaal niet verbeteren! Wat je aan een kant van het proces verandert, heeft consequenties voor de andere kant van het proces.

Een voorbeeld van wereldverbeteren met objecten: het object 'vloer' in een schoonmaakmanagementsysteem (galgewoord!), zal vanuit traditioneel oogpunt vooral weet-verantwoordelijkheden krijgen. 'Geef je oppervlakte' en 'welk schoonmaakmiddel heb je nodig' zijn dingen die in de werkelijkheid ook bij een vloer passen. Een wereldverbeterende optie is om de vloer de verantwoordelijkheid 'maak jezelf schoon!' te geven. Dat is niet bepaald natuurgetroouw maar wel zeer effectief. De vloer weet als de beste hoe hij schoongemaakt moet worden. Die vloer moet dat wel delegeren aan de schoonmaker en daarbij vertellen wat de schoonmaker precies moet doen. Dat is beter voor de vloer én voor de schoonmaker. De schoonmaker hoeft niet aangepast te worden, zodra er een nieuw soort vloer geïntroduceerd wordt.

Tip: probeer eens los te komen van dat saaie inefficiënte wereldbeeld. Probeer de wereld te verbeteren!

*Daan Kalmeijer is docent consultant bij  
CIBIT adviseurs | opleiders  
(e-mail: daan@cibit.nl).*