

# Java niet meer relevant?

De Object Management Group (OMG) heeft het afgelopen jaar een nieuwe missie op zich genomen. Na CORBA en UML prijkt nu MDA op de frontpage van [www.omg.org](http://www.omg.org). Model Driven Architecture zou de definitieve doorbraak moeten zijn van domein gecentreerd, model gedreven systeemontwikkeling. Wat ging er aan MDA vooraf? De *three amigos* kwamen bij elkaar en bedachten de Unified Modeling Language (UML). UML is een modelleertaal, gebaseerd op het gedachtegoed van object oriëntatie, waarmee systemen in plaatjes uitgedrukt kunnen worden. UML is op het moment de enige serieuze optie voor iedereen die met moderne technieken (lees: met object georiënteerde talen) systemen modelleert. Al vrij snel na het uitkomen van UML ontstonden er tools die in staat waren om op basis van UML diagrammen code te genereren. Die gegenereerde code bestond alleen nog maar uit een soort skelet van het te maken systeem. Systemen konden ook teruggelezen worden tot een model; reverse engineering. Ook bij reverse engineering waren de tools alleen maar in staat om een paar kernachtige aspecten van het model te genereren.

Iedereen die wel eens Java code gegenereerd heeft op basis van UML zal zich verwonderd hebben over de zeggingskracht van de modelleertaal UML en het minuscule deel van die kracht dat toegepast wordt om code te genereren. MDA is onder andere een poging om te zorgen dat alle UML, het hele model, gebruikt

kan worden voor het genereren van code. De nieuwe versie van UML is, vanwege MDA, grondig herzien om te zorgen dat alles wat je met een programmeertaal zou kunnen zeggen ook in UML gezegd kan worden. Daarmee wordt de waarde van een systeem verplaatst van de code naar het model. Aangezien UML een werkelijke standaard is, is een mogelijkheid voor volledige platformafhankelijkheid ontstaan. Op basis van een model genereer ik op dag 1 (platform specifieke) C#-code voor op een Windows machine en op dag 2 (iets minder platform specifieke) Java code voor op een Unix machine. Het maakt niet uit of de gebruikte programmeertalen platformafhankelijk zijn wanneer ik van de ene op de andere dag al mijn code opnieuw kan schrijven. Wie programmeert er straks nog Java?

Om MDA alleen maar te bestempelen als een nieuwe techniek om code te genereren is onjuist. Een belangrijk element van MDA is de doelstelling om UML business modellen te gebruiken om systemen te genereren. Het business model is de kern van de zaak, het business model heeft altijd gelijk! Als je kijkt naar de UML business modellen die in de praktijk gemaakt worden, dan zijn er een heleboel aspecten van systemen die niet, of hooguit impliciet, uitgemodelleerd worden. Ik ben nog nooit UML (business) modellen tegengekomen waarin de business-eisen aan security, transacties, logging of concurrency gemodelleerd zijn. Aan de andere kant

valt het natuurlijk wel op dat de hierboven genoemde (technische) systeemaspecten allemaal onderdeel zijn van de J2EE architectuur. J2EE heeft als doelstelling dat de Java programmeur zich alleen maar hoeft te bemoeien met de business logica, de deployer bepaalt hoe security, transacties, logging en andere technische on-interessantigheden geregeld worden. Daar hoeft de dure Java programmeur zijn uren niet meer aan te besteden. Java wordt dichtgeknepen tussen MDA en J2EE deployment!

Er resteert echter een belangrijke reden waarom we de komende tijd echt nog wel handmatig Java zullen blijven programmeren. Want hoe mooi UML modellen ook gemaakt worden, de meeste Java programmeurs snappen geen snars van de ideeën van die rare UML-modelleers. Ze verzinnen meestal ter plekke een volledig andere, volledig onontwarpbare Java implementatie en vertellen dan vol trots dat het volledig voldoet aan de modellen. De modelleur geeft zich gewonnen, hij/zij kan de code toch niet lezen en het systeem doet wat het moet doen (nou ja, grotendeels dan). Zo lang er geen tools zijn die op basis van UML dezelfde spaghetti kunnen maken als een gemiddelde Java programmeur, zal er geen enkele programmeur instemmen met het aankopen van dergelijke tools.

*J. Meermans*

is Java-kenner en te bereiken via  
[meermans@cibit.nl](mailto:meermans@cibit.nl)