

**Het is niet de bedoeling dat de Heilige Graal ooit gevonden wordt. Zit besloten in het concept. Hoe moet dat ooit zijn gegaan? Koning Arthur zit zich samen met zijn ridders dag na dag te pletter te vervelen aan die Ronde Tafel. Geen rooftochten of oorlogen in het verschiet. Zelfs nog geen kruisvaart in de planning. De een plukt wat aan zijn langzaam wegroestende maliënkolder. De ander slijpt zijn strijdbijl nog maar eens een keer. De hofnar zit zachtjes te snikken, ten prooi aan manische depressies.**

## Heilige graal

**A**ls je dan een doorbraak weet te forceren, ben je een hele grote. En dat is Arthur, zonder meer. In een belangrijk moment van verlichting besluit hij de legende van de Heilige Graal af te stoffen. Hij stuurt zijn ridders naar alle windrichtingen om het ding te vinden. En reken maar dat ze daar lekker mee bezig zijn. Verre streken vol vuurspuwende draken, verraderlijke struikrovers en verleidelijke maagden: er moet van alles het hoofd worden geboden. Dichters en toneelschrijvers volgen het gezelschap op de voet om van de queeste getuige te doen. Het zou een ramp zijn als de schat ook echt wordt gevonden (een enkele kinderachtige geschiedschrijver beweert dat het Galahad zou zijn gelukt, maar de wens is hier overduidelijk de vader van de gedachte).

Veel is er sindsdien niet veranderd. Vooral niet in de IT, waar we niet eens koning Arthur nodig hebben om doelen te vinden om achteraan te rennen. Model Driven Development is zo'n doel. En we zijn er in het vakgebied al zo lang naar op zoek, dat we rustig van mythische proporties kunnen spreken. Vinden we de ultieme oplossing ooit? Je moet er niet aan denken, want onze eigen Ridders van de Ronde Tafel – vooral overigens agile aanhangers, geniet even van de duivelse parallellen met iteratief, cyclisch, rond – zouden zich geen raad weten.

Gelukkig moeten er nog genoeg taaie uitdagingen worden opgepakt. De notatiewijze om modellen mee vast te leggen bijvoorbeeld. Een van de meest voor de hand liggende kandidaten daarvoor is nog steeds UML. Maar dat is een modelleertaal die in zijn meest recente versie van een dusdanige complexiteit is geworden, dat je er hoogstens met raketgeleerden nog betekenisvol over kunt

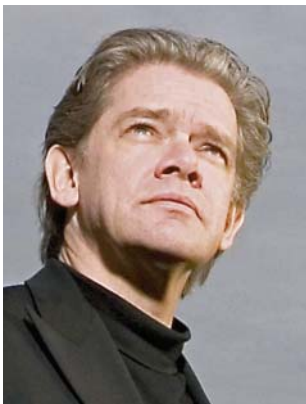
discussiëren. En dat is dan nog Jip en Janneke, vergeleken met de Object Constraint Language (OCL), die je vanzelf nodig hebt om de fijnere kneepjes van modellen vast te leggen. Het is de Wet van Behoud van Moeilijkheid: complexe, niet triviale logica zul je hoe dan ook een keer moeten vastleggen. En of je dat nu doet in machinetaal, Java of een 'hogere' specificatietaal, het blijft moeilijk.

De grootste draak op het pad van modelgedreven ontwikkeling zit hem echter in de transformatie. Vroeg of laat moeten modellen immers worden omgezet in werkende systemen. Een razend interessant gebied voor de ware liefhebber. Voor je het weet ben je alleen nog maar bezig met het zelf maken en verbeteren van codegeneratoren en metamodellen. Jazeker, een uitdagende discipline. Maar wel een die complete continenten verwijderd is van de realiteit van de bedrijfsvoering, het gebied waar het heel oorspronkelijk om was begonnen.

Misschien moeten we het daarom maar niet te gek vinden dat in dat lange pad tussen bedrijfsgeoriënteerde modellen en werkende systemen hoe dan ook ruis optreedt. En dan hebben we het niet alleen over modellen die niet goed op elkaar aansluiten.

Ik ken een wereldbepaalde softwaregoeroe die ervan overtuigd is de noot bijna te hebben gekraakt. Hij heeft een aanpak die allerlei modellen – dwars door elkaar heen – toestaat. Volstrekt briljant, maar ik kan me geen klant voorstellen die hij niet binnen 5 minuten dodelijk zal hebben beledigd. Ondertussen galoppeert hij vastbesloten en met blij gemoed door, richting horizon.

Blinkt daar in de verte niet iets? «



**Ron Tolido**  
tolido.blogspot.com