

**Heb je dat ook wel eens? Een discussie over het verschil tussen twee concepten? Over de betekenis van een term? Het overkomt mij dagelijks. Neem de term *model*. Alles rond die term boeit me mateloos en ik heb regelmatig gesprekken met allerhande mensen over modellen.**

## Mograms

**M**aar wat is nu een model? Vaak wordt een model tegenover een *programma* gezet. Sommigen onder ons zijn programmeur en vinden modelleren maar niets, bij anderen is het precies andersom. Men beschouwd een model als een hoogniveau-specificatie van een systeem en een programma is iets dat op detail niveau bestaat en dat het systeem is. Programma's kunnen geëxecuteerd worden en modellen niet. Verder wordt een model vaak gezien als iets dat visueel is, en een programma als iets tekstueels. Modellen zijn globaal, programmaal gedetailleerd. Modellen zijn vaak informeel van aard, terwijl programma's precies zijn.

Enige jaren geleden hebben we de zogenaamde *Modeling Maturity Levels* (MML) bedacht. Dat zijn zes niveaus (nul tot en met vijf) waarop modellen gebruikt kunnen worden. Zonder op de details in te gaan (zie <http://www.devx.com/enterprise/Article/26664>) is er één karakteristiek van een model dat essentieel is voor deze discussie.

Modellen op MML 0 t/m MML 3 zijn *beschrijvingen* van systemen die handmatig vertaald dienen te worden naar software. Dit omvat zowel praatplaatjes als zogenaamde blueprints. Handig voor menselijke communicatie.

Op MML 4 en MML 5 zijn de modellen essentieel veel meer dan alleen een *beschrijving* van software. Een model is op dit niveau kan geheel automatisch omgezet worden naar werkende software. Op welke wijze dit gebeurd is irrelevant. Er kan code gegenereerd worden, het model kan rechtstreeks door een engine uitgevoerd worden, er kunnen configuratiebestanden voor andere tools uit gegenereerd worden enzovoorts. De essentie is dat menselijk tussenkomst niet meer nodig is.

In de CAD/CAM wereld wordt zo een model ook wel een *solid model* genoemd. Met een solid

model kan je het gemodelleerde object simuleren, maar ook de machines reeks aansturen die de onderdelen frezen. Goed bekeken is een solid model lastig te onderscheiden van een programma. Het is gedetailleerd genoeg om alle informatie te bevatten, net als een programma.

Een solid model is niet informeel, anders zou menselijk tussenkomst het model moeten interpreteren.

Van de andere kant gezien kunnen we een programma ook als een solid model beschouwen. Het voldoet aan alle eisen: het is een beschrijving van software welke automatisch omgezet kan worden naar bytecode of machinecode en daarmee executeerbaar is.

Hoewel een model vaak visueel is, kan er ook altijd een tekstuele representatie gegeven worden. Zo bestaat er bijvoorbeeld een OMG standaard HUTN: Human readable Textual Notation voor UML. Programma's kunnen ook visueel weergegeven worden. Ieder reverse engineering tool is daar een bewijs van. Dit visueel versus tekstueel is in werkelijkheid geen fundamenteel verschil, maar meer een cultuuraspect of een kwestie van smaak. Bovendien komen er meer en meer talen die meerdere representaties hebben, zowel tekstueel als visueel.

Al met al blijft er van het verschil tussen model en programma bijster weinig over. Wat mij betreft kunnen we deze termen dan ook best door elkaar kunnen gebruiken, maar mensen hebben nu eenmaal bestaande connotaties met deze termen.

In het boek *Software Language Engineering* van Anneke Kleppe (verschijnt voorjaar 2008 bij Addison-Wesley) wordt om bovenstaande reden een nieuwe term geïntroduceerd.

Vanaf heden heb ik dus een nieuwe titel op mijn visitekaartje: ik ben een *Mogramming Specialist*. Wie volgt?



**Jos Warmer**

Partner Ordina SI&D.

E-mail: [jos.warmer@ordina.nl](mailto:jos.warmer@ordina.nl).