

# Praatplaatjes



Dublin, september 1997: de Object Management Group (OMG) heeft UML aangenomen als nieuwe standaard op het gebied van objectgeoriënteerde analyse en ontwerp. We heffen het glas met alle betrokkenen.

Enige jaren hiervoor: ik kwam in die tijd eens bij een klant om de modellen die ze daar gemaakt hadden te reviewen. Ze lieten me trots hun mooie modellen zien.

Waarom ze zo trots waren begreep ik niet. Alle overervingrelaties klopten in het geheel niet, ze hadden er duidelijk niets van begrepen.

Daar zit je dan, met een enthousiaste klant die zijn best doet, en hoe vertel je hem nu dat hij het helemaal fout gedaan heeft? Terwijl ik hierover zat te dubben ging men verder met uitleggen van de modellen. Hé! Wat is dat voor raar symbool? Verwarring maakte zich van me meester! Kende ik mijn modelleertaal toch niet zo goed als ikzelf dacht? Al snel kwam bij mij het juiste inzicht. Mijn klant gebruikte een *andere modelleertaal*, met grotendeels dezelfde symbolen. Maar ... een aantal van die symbolen had wel een andere betekenis. Zo bleek de overerving waarover ik viel in deze taal aggregatie te betekenen. Ineens viel alles op zijn plaats en bleek dat men inderdaad goed werk had afgeleverd. Blij dat ik eerst even mijn mond had gehouden.

Software-ontwikkeling is een teamsport, waarbij communicatie van cruciaal belang is. We gebruiken hiervoor vaak een combinatie van natuurlijke taal (Nederlands, Engels, ...) en diagramtechnieken (UML, ER, DFD, SDL, ...). Voor de komst van UML was de modelleerwereld een ware toren van Babel. Er was een oerwoud aan methoden en technieken. Dit leidde, zoals uit het verhaal hierboven blijkt, tot veel verwarring en miscommunicatie. Een eerste vereiste voor communicatie is immers dat iedereen dezelfde taal spreekt. Ook was het in die tijd typerend dat projecten de eerste drie maanden besteden aan het selecteren van de te gebruiken modelleertaal. Bepaald geen efficiënt proces.

De komst van UML heeft hierin fundamenteel verandering gebracht. Iedereen begreep dat het hebben van één gezamenlijk taal communiceren eenvoudiger maakte. De urgentie was

zo groot, dat de gezamenlijke industrie binnen een jaar wist te komen tot de UML-standaard. Over UML is sindsdien wel één grote misvatting ontstaan. Er zijn mensen die denken dat een UML-model een perfecte eenduidige betekenis heeft. Niets is minder waar, UML kan van alles betekenen.

In dit opzicht lijkt UML veel op natuurlijke taal. Hoewel we iedere Nederlandse zin altijd op meerdere manieren kunnen interpreteren blijkt dat we toch in staat zijn om in het Nederlands uitstekend met elkaar te communiceren. Zo ook met UML. Voor de een is een rechthoek in een UML-klassendiagram een business-concept, voor anderen een Java of een C# klasse en voor weer anderen een tabel, een XML-message, een entity bean of nog iets anders.

UML kan op veel verschillende manieren gebruikt worden, terwijl er toch voldoende gemeenschappelijke betekenis te herkennen valt om elkaar te begrijpen. Het overweldigende succes van UML is dus te danken aan het feit dat UML een flexibele taal is waarin we over onze ideeën met elkaar kunnen communiceren. Ik noem zulke UML-modellen dan ook *praatplaatjes*. Als je dus weer eens een UML-model tegenkomt en je vraagt je af wat de schrijver bedoeld heeft, denk er dan aan dat dit juist de bedoeling is. Stap op hem af en praat erover. Daar is dat UML-plaatje voor.

Jos Warmer,  
Partner Ordina SI&D.  
E-mail:  
[jos.warmer@ordina.nl](mailto:jos.warmer@ordina.nl)