



Hebt u wel eens een project meegemaakt waarin gebruik werd gemaakt van de modelleertechnieken van de Unified Modeling Language? Hebt u zelf wel eens use cases opgesteld? Of hebt u wel eens op traditionele wijze een functioneel en technisch ontwerp gemaakt, en dat vervolg naar Java vertaald? De kans is groot dat u dan tegen dezelfde problemen bent aangelopen als vele andere vakgenoten.

In theorie zijn praktijk en theorie hetzelfde



Alle prachtige literatuur en cursussen ten spijt, is het in de praktijk van alledag niet eenvoudig om van requirements via het ontwerp van de functionaliteit naar een geaccepteerde applicatie te komen. Onderweg vindt u heel wat beren op de weg. De Unified Modeling Language (UML) biedt een verzameling prima modelleertechnieken. Helaas biedt UML niet die vingerwijzingen die het toepassen van de individuele modelleertechnieken makkelijk maakt. De meeste literatuur op dit gebied doet dat ook niet, en komt niet veel verder dan het uitleggen van de bekende technieken. En dan nog, ook als u expert bent in het opstellen van use cases, of van klasdiagrammen, ontbreekt een goed gedefinieerde samenhang tussen de modelleertechnieken. Die zult u dus zelf moeten invullen.

Auteur Doug Rosenberg verwoordt het goed in zijn, al wat oudere boek "Use Case Driven Object Modeling with UML. A practical approach". De meeste methodologen opereren op 3000 voet boven de aarde, zegt Rosenberg. In theorie zijn praktijk en theorie vanaf deze hoogte hetzelfde, vervolgt de auteur. Hijzelf probeert daarentegen op zo'n 1000 voet

mensen in alledaagse projecten te ondersteunen met objectgeoriënteerde ideeën. Hij beschrijft daarom in zijn boek een praktische aanpak voor dit type systeemontwikkelprojecten. Deze aanpak biedt een mix van de individuele methoden van de three amigo's Booch, Rumbaugh (OMT) en Jacobson (vooral Objectory). Rosenberg heeft de verschillende uitgangspunten en ideeën van deze methoden gecombineerd tot wat hij noemt: driving object models from use cases.

De aanpak van Rosenberg omvat hiertoe een vijftal hoofdactiviteiten, te weten domain modeling, use case modeling, robustness analysis, interaction modeling en implementation. In het boek komen ieder van deze vijf hoofdactiviteiten elk in een hoofdstuk aan bod. Daarbij schuwt Rosenberg niet om de modelleertechnieken van UML uit te breiden daar waar hij dat nodig acht. Zo gebruikt hij bijvoorbeeld bij voorkeur niet de stereotypen <<includes>> en <<extends>> wanneer hij use case diagrammen tekent. Rosenberg hanteert hier liever <<preceeds>> en <<invokes>>. Een erg aardig aspect aan het boek is dat de auteur regelmatig aangeeft waar in de verschillende activiteiten ana-

lysis paralysis voorkomt en hoe deze verlamingsverschijnselen kunnen worden voorkomen. Aan het eind van ieder hoofdstuk geeft hij bovendien een top tien van misverstanden.

Het boek lijkt dus een oplossing te bieden voor de vragen waarmee in vrijwel ieder project dat gebruik maakt van objectgeoriënteerde technieken geworsteld wordt. "Use Case Driven Object Modeling with UML. A practical approach" telt slechts 148 pagina's en blijft zo beperkt van opzet. Alhoewel het boek door de flegmatieke schrijfstijl van de auteur zeker goed leesbaar is en een prima opbouw kent, heeft het toch niet de diepgang die het onderwerp vereist. Het geeft daarom een goede aanzet tot modelleren in de praktijk, een definitieve gids voor systeemontwikkelprojecten is het niet. Ook 1.000 voet is wellicht te hoog.

*Sander Hoogendoorn, Partner Ordina groep
(e-mail: sander.hoogendoorn@ordina.nl).*

Titel: Use Case Driven Object Modeling with UML.
Auteur: Doug Rosenberg/ea
Uitgave: Addison Wesley