



Deze zondag keerden we terug van vakantie. Van vier weken vakantie om precies te zijn. Heerlijk uitgerust. Het is goed toeven op de stranden van Bretagne. Niet te warm – net iets minder warm dan in uitdrogend Nederland en met een verkoelend zeebriesje. Precies goed om de jaarlijkse terugkerende leesachterstand weer eens helemaal goed te maken.

## Archetypisch

Dit jaar besloot ik een regelmatig terugkerende tip op te volgen en heb ik *Enterprise Patterns and MDA - Building Better Software with Archetype Patterns and UML* van Jim Arlow en Ila Neustadt in de koffer gepropt. Een hele mondvul. Dat belooft wat. De wedstrijd om de meeste buzzwords in één boektitel winnen de auteurs in elk geval glansrijk. Laat ik kort illustreren wat de auteurs beogen.

Stel dat een willekeurige bank of verzekeraar u vraagt voor een project waarin een applicatie wordt ontwikkeld voor het aanvragen van hypotheekoffertes. Grote kans dat in dit project een model wordt opgezet waarin alle kernbegrippen aan de orde komen, zoals hypotheek, offerte, tussenpersoon en onderpand.

Neemt u van mij aan dat er meer banken en verzekeraars zijn die zo'n applicatie ontwikkelen. Stel nu dat u – enkele jaren later – door een andere bank ook wordt gevraagd voor een project waarin een applicatie voor hypotheekoffertes wordt ontwikkeld. 'Dit model heb ik toch al eerder eens gezien?', verzucht u dan toch? Dit willekeurige voorbeeld heb ik meermaals meege maakt.

Precies daar ligt de kern van

*Enterprise Patterns and MDA*. Het boek bevat een ruime collectie archetype patterns. Zo'n pattern is een model van een veel voorkomend domein. De auteurs hebben hun uiterste best gedaan zo volledig mogelijk te zijn. Zo beschrijven ze in klassendiagrammen met begeleidende tekst nauwgezet patronen als Party, Product, Inventory, Order en Money.

Een nobel streven van de auteurs, zeker gezien het enorme detailniveau van de patronen. In dit detailniveau onderscheidt het boek zich van *Analysis Patterns* van Martin Fowler – iets dat de auteurs in de inleidende hoofdstukken graag benadrukken. Van mij mogen deze hoofdstukken in een volgende druk worden weggelaten. Ze lijken vooral bedoeld om de lezer duidelijk te maken wat een uniek idee het archetype pattern wel niet is. En om een aantal halfbakken ideeën te introduceren. Zo heet een model met begeleidende tekst plots 'literate modeling' en definiëren ze een eigen constraint language. Alsof OCL nog niet moeilijk genoeg is.

Wat overblijft zijn de patronen, die natuurlijk tot doel hebben het ontwerpen en eventueel genereren van applicaties over vergelijkbare domeinen te vergemakkelijken. Ik

vrees echter dat ze door de haast Duitse grondigheid van de auteurs vooral als naslagwerk gaan eindigen. De patronen zijn veelal te uitgebreid om te implementeren. Ze zijn het kleinste gemene veelvoud van wijdlopende gedachten. Zo omvat het patroon Money – toch niet het grootste – ruim twintig klassen, variërend van Money en Currency tot RoundingPolicy en ExchangeRateType. Bij het toepassen van de archetypen uit het boek, maar ook zelf ontwikkelde, zoals misschien een patroon Hypotheek, dat jammer genoeg in het boek ontbreekt, zal het altijd nodig zijn het patroon te strippen tot een vorm die geschikt is voor het onderhanden project. En dat is dan weer archetypisch ontwerpwerk.

Sander Hoogendoorn  
([www.sanderhoogendoorn.com](http://www.sanderhoogendoorn.com))

### Waardering

★★★★☆

Auteur: Jim Arlow & Ila Neustadt  
Titel: *Enterprise Patterns and MDA. Building Better Software with Archetype Patterns and UML*  
Uitg.: Addison Wesley

