Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2023, gehouden op donderdag 30 maart 2023 jl. en georganiseerd door NHL Stenden Hogeschool). Bij elkaar bijna 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op donderdag 27 maart 2025 in Zwolle en wordt dan georganiseerd door Hogeschool Windesheim. Kijk op www.nioc2025.nl voor meer informatie.

Wil je op de hoogte blijven van de ontwikkeling rond Stichting NIOC en de NIOC kennisbank, schrijf je dan in op de nieuwsbrief via

www.nioc.nl/nioc-kennisbank/aanmelden nieuwsbrief

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.

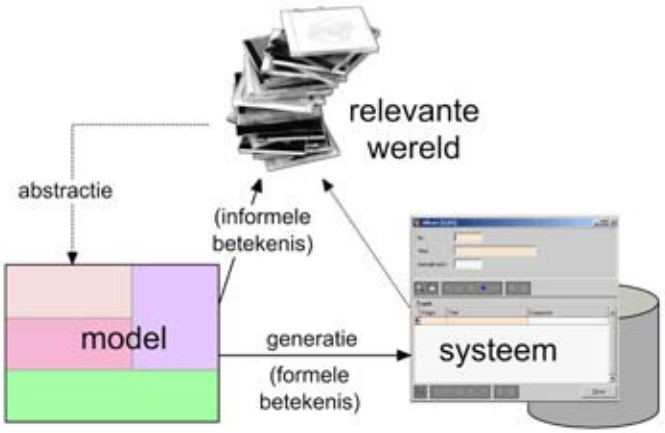
Ambachtelijk modelleren, met goed gereedschap

Leo Wiegerink,
Harold Pootjes,
Jeanot Bijpost en Marco de Groot
Bron: TINFON NIOC-special, 2007, nummer 1

Komende zomer brengt de Open Universiteit een nieuwe cursus uit over het ontwikkelen van informatiesystemen onder de titel Model-driven development. Dat is goed nieuws, want deze cursus geeft het ambacht modelleren weer de plek in het informaticaonderwijs die het verdient. Door de pragmatische aanpak en een prachtig stuk gereedschap gaan studenten modelleren zelfs weer leuk vinden.

Model-driven development

Model-driven development (MDD) is een ontwikkelaanpak waarbij de gemaakte modellen op twee manieren betekenis hebben. Allereerst is er een informele betekenis: het model als abstraherende beschrijving van de wereld, voor zover die relevant is voor het te ontwikkelen informatiesysteem. Maar er is ook een formele betekenis: het model als formele specificatie voor het te genereren systeem. Modelleren volgens MDD is dus evenzeer 'terugkijken' naar een bestaande wereld als 'vooruitkijken' naar de consequenties voor het systeem. Zie figuur 1.



Figuur 1:
Tweevoudige betekenis van MDD-model

Systeemgeneratie uit het model is de grondslag van een, waarbij de applicatie periodiek door belanghebbenden wordt beoordeeld op functionaliteit, usability, etc. Dit maakt modelleren tot een experimentele bezigheid.
De MDD-tool van de cursus, Cathedron geheten, ondersteunt dit proces optimaal.

Inhoud en didactische aanpak

We leren de studenten in deze cursus geen methoden die feilloos tot 'het juiste model' leiden, want 'het juiste model' bestaat niet. Wel leren ze, door kennis op te doen van een aantal steeds terugkerende basispatronen, in nieuwe problemen snel de overeenkomsten met oude problemen herkennen. Zo kunnen ze zich concentreren op wat echt moeilijk is. Ze ontwikkelen inzicht in de kracht en de beperkingen van een model door kritische reflectie, vergelijking met alternatieven en niet in het minst door de gegenereerde applicaties te onderzoeken.

Als student ben je in deze cursus (100 uur zelfstudie met vijf facultatieve begeleidingsbijeenkomsten) voortdurend getuige van experimenten én zelf aan het experimenteren. Je leert via voorbeelden, informatiepatronen en oefeningen dat modelleren betekent: steeds afwegingen maken. Bijvoorbeeld tussen een eenvoudige structuur met complexe constraints en een complexe structuur met eenvoudiger constraints. Of tussen harde constraints en voorschriften voor een workflow.

In de blokken 1 en 2 van de cursus ligt de nadruk op het informatiemodel. De student maakt kennis met platformonafhankelijke en platformafhankelijke (relationele) specificaties. Hij leert werken met het gereedschap (Cathedron) en leert dit in te zetten om modellen, via de default applicaties, op hun merites te beoordelen. Hij leert informatiepatronen herkennen en toepassen. Hij leert door een modulaire benadering complexe problemen op te delen in kleinere deelproblemen.
In blok 3 gaat het om het doorontwikkelen van een applicatie met non-defaults, op basis van een (min of meer) stabiel informatiemodel: de user interface wordt verder aangekleed en er wordt logica toegevoegd als implementatie van constraints en bedrijfsregels.
Blok 4 is gewijd aan meer gevorderde onderwerpen: het gebruik van subklassen met de vele afwegingen die daarbij horen, het streven naar flexibele, generieke oplossingen (waarbij subklassen veelal weer sneuvelen!) en een relativerende vergelijking van verschillende technieken voor informatiemodellering.

Inhoud cursus Model-driven development

Blok 1: Modelgestuurd ontwikkelen
- Inleiding
- Model-driven development met Cathedron

Blok 2: Structuur in informatie
- Informatie, objecten en feiten
- Analyseren en modelleren
- Informatiepatronen
- Methodisch modelleren

Blok 3: Interface en logica
- Gebruikersinterface (1)
- Gebruikersinterface (2)
- Logica (1)
- Logica (2)

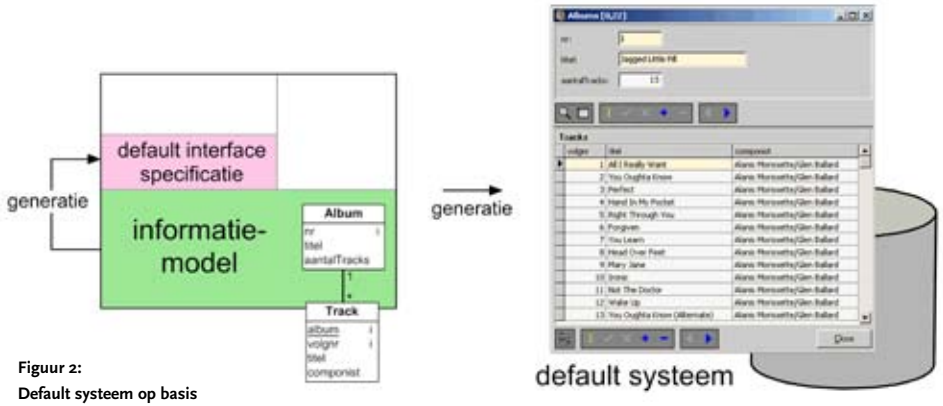
Blok 4: Voortgezet modelleren
- Generalisatie
- Generiek modelleren
- Casus: Vakantiepark

Informatiemodel, interfacespecificatie en logica

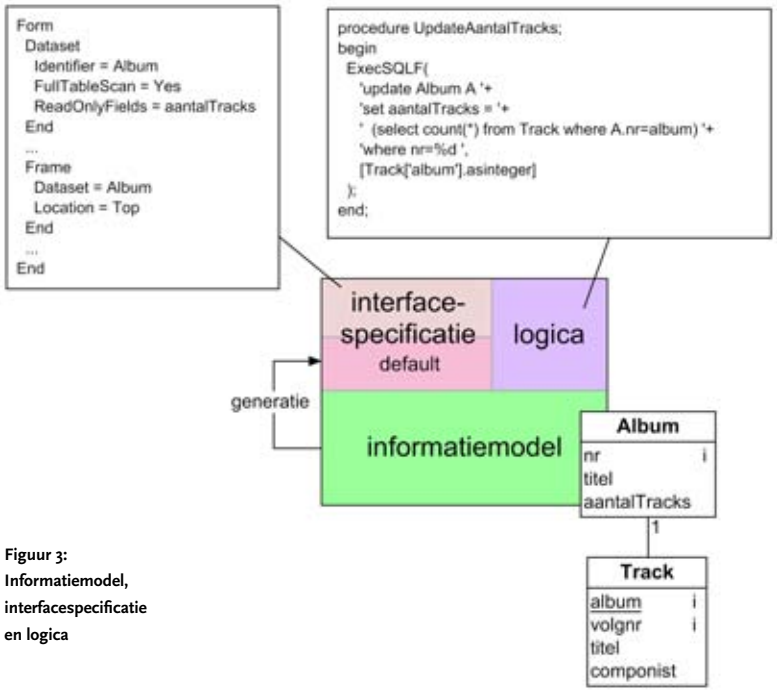
De modellen hebben drie componenten:
- informatiemodel
- interfacespecificatie
- logica.

De kern is het *informatiemodel,* waarvoor een UML-profile wordt gebruikt. De tool is in staat om op basis daarvan een zeer rijk default systeem te genereren. In veel gevallen is dit voldoende om als prototype aan gebruikers voor te leggen. In een iteratief proces kan het model zodoende gevalideerd worden, totdat het stabiel is. In de cursus wordt dit proces geoefend. Zie figuur 2, waarin dit wordt geïllustreerd voor het simpelste voorbeeld uit de cursus, de albumcollectie van een muziekliefhebber. Dit voorbeeld wordt in de loop van de cursus uitgebouwd tot onder meer een muziekwinkel, een muziekuitleen, een mediawinkel en een webmediawinkel. Doordat Cathedron snel genereert (er hoeft niet te worden gecompileerd), is het voor een student geen enkel probleem veelvuldig ver-

anderingen in het model aan te brengen en deze te testen. Bij aanpassing van het model blijft de oude structuur zoveel mogelijk intact. Cathedron zorgt dat zelfs bestaande populatie, voor zover mogelijk, behouden blijft. Het formulier van figuur 2 is dan ook niet voor niets van populatie voorzien.
De *interfacespecificatie* legt het uiterlijk en het gedrag van de interface vast: formulieren, knopjes, menu's, etc. Dit is inclusief de koppeling met event handlers. Voor het genereren van een default applicatie is het informatiemodel voldoende. De default applicatie wordt gegenereerd met behulp van een default interfacespecificatie, zoals getoond in figuur 2.
Figuur 3 laat een stukje interfacespecificatie zien, horend bij een master-detail-formulier voor muziekalbums. De gebruikte taal is een simpele hiërarchische taal, conform de hiërarchische opbouw van de interface zelf.
Een volgende uitbreiding is de *logica* om bedrijfsregels te implementeren. Zie figuur 3, met een stukje logica om het afleidbare veld aantal Tracks bij te houden. Hiervoor wordt een 3GL-taal gebruikt (Object Pascal).



Figuur 2:
Default systeem op basis van informatiemodel



Figuur 3:
Informatiemodel, interfacespecificatie en logica

Kracht en eenvoud

'Kracht en eenvoud' is het motto van de cursus en van het gebruikte gereedschap. We gebruiken een diagramtechniek die zo simpel is als maar kan, onder meer door de opmaakconventie een parent-klasse (master) hoger dan zijn child-klassen (details) te tekenen. Dit maakt het expliciet vermelden van standaardmultipliciteiten (0..1 en 0..n) overbodig, waardoor de diagrammen aan rust winnen en precies de essentie tonen. Geen spoorzoeken meer: in één oogopslag is de navigatierichting duidelijk van master naar detail ('van één naar veel'). Als vanzelf vallen de grotere diagrammen door deze conventie uiteen in modulaire onderdelen, zoals bijvoorbeeld een vraag-, aanbod- en transactiemodule.

Fundamenteel in het licht van kracht en eenvoud is dat 'primaire identiteit' een conceptuele hoofdrol heeft gekregen in de vorm van identiteitsregels (aangegeven met i). Dit in afwijking van UML. Die afwijkingen zijn er meer. Bijvoorbeeld in het vermelden van verwijzende attributen en van attribuuttypen. We zijn dus niet bang geweest UML aan te passen, daar waar het in onze visie tekortschoot.

De cursus maakt onderscheid tussen *model en diagram*. Een model is een specificatie en die moet volledig zijn voor succesvol genereren. De diagrammen dienen primair een communicatiedoel. Ze kunnen door de tool met meer of minder detail worden

afgebeeld. Aanvullende specificaties worden verstrekt via een edit-formulier, waarin ook platformonafhankelijke interfacespecificaties kunnen worden verstrekt. Er is een schetsmodus die alleen het hoognodige toont, voor beginnende studenten een erg prettige voorziening.

Model-driven architecture

De MDD-modellen van de cursus moeten gezien worden in het licht van de Model-driven architecture (MDA): het door de Object management group (OMG) op stapel gezette programma om modellen zoveel mogelijk platformonafhankelijk en volgens gestandaardiseerde techieken (waaronder UML-klassendiagrammen) te ontwikkelen.

In de MDA-visie worden informatiemodel, interface en logica volledig platformonafhankelijk gespecificeerd in een *platform independant model* (PIM). Voor een specifieke platformcombinatie (bijvoorbeeld Java met een relationele Oracle-database) wordt het PIM volautomatisch getransformeerd naar een *platform specific model* (PSM).

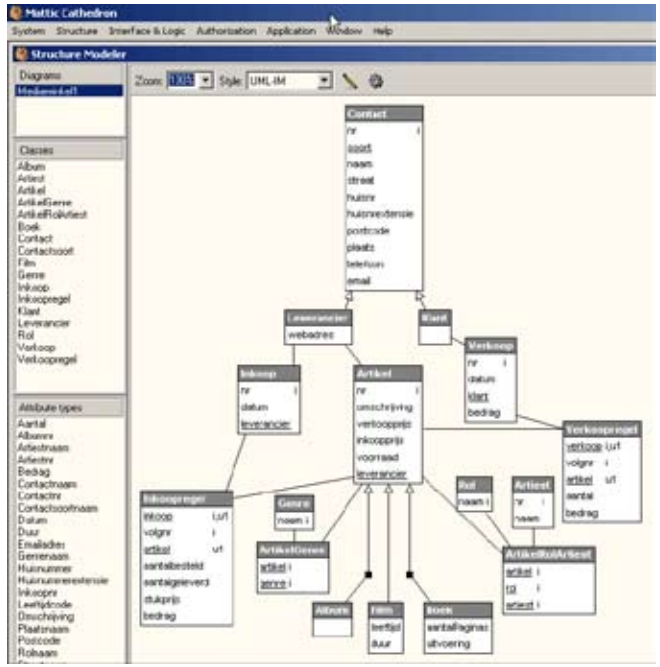
Compilatie leidt tot een werkende applicatie. Een alternatief is de modellen (op zo hoog mogelijk niveau, dat wil zeggen idealiter PIM) te laten uitlezen door een engine die in feite de applicatie is. Elke wijziging van het model (informatie, interface of logica) resulteert dan direct in een wijziging van de applicatie. De ontwikkelomgeving en de gebruikersomgeving zijn in dit geval twee views op hetzelfde systeem. Cathedron werkt op deze laatste manier: uiterst gemakkelijk en snel.

Cathedron

Een ambachtsman kan niet zonder goed gereedschap. Bij de cursus wordt de MDD-tool Cathedron geleverd. Twee jaar geleden werd toolbouwer Mattic met deze tool winnaar van de RAD-race (Rapid application development). Inmiddels is de tool in samenwerking met de Open Universiteit doorontwikkeld. Hij is nu voorzien van een grafische informatiemodelleringsinterface op PIM-niveau (zie figuur 4).
Uniek is dat met de tool niet alleen nieuwe systemen kunnen worden gegenereerd, maar dat er ook reverse engineering mee mogelijk is van bestaande databases, zelfs met behoud van populatie.

Cathedron is zeer geschikt om een iteratief ontwikkelproces te ondersteunen. In een handomdraai wordt de structuur van een model gewijzigd, wordt de interface aangepast of wordt een stukje logica toegevoegd. Natuurlijk kunnen we intrinsiek lastige dingen niet ineens eenvoudig maken. Een complexe bedrijfsregel zal dus een complex stukje programmeren vragen. Maar in een omgeving die zoveel rijkdom kado geeft, is dat voor studenten juist weer een aangename uitdaging.

Cathedron kan elk diagram (standaard in UML-IM, de keuze voor het cursusboek) met een druk op de knop aanpassen aan een andere techniek naar keuze, waaronder ERD, Bachman, ORM en puur UML (zonder identiteitsregels, verwijzende attributen en attribuuttypen). Didactisch is dit erg prettig: zo kunnen we studenten laten zien dat de tekentechniek er in wezen niet toe doet.



Figuur 4: Informatiemodelleringsinterface van Cathedron (schetsmodus en UML-IM)

Achter alle plaatjes zit dezelfde repository en dus dezelfde informatiestructuur.

We zouden Cathedron kunnen omschrijven als een MDD/MDA-tool, doordat het belangrijke stappen heeft gezet op de weg naar platformonafhankelijke specificatie van informatie, interface en (in mindere mate) logica.
Ter illustratie: door een vrijwel platformonafhankelijke interfacespecificatie kost het niet veel meer dan een druk op de knop om een web-client te genereren in plaats van een Windows-GUI.

Kenmerken van Cathedron op een rijtje

- Grafische informatiemodelleringsinterface, voor platformonafhankelijke specificatie van informatiestructuur
- Aanpassingen aan informatiemodel vanuit grafische interface, met behoud van gebruikersgegevens
- Krachtige default applicatie
- Ondersteuning meerdere diagramtechnieken, met meer of minder detail
- Reverse engineering, met behoud van gebruikersgegevens
- Genereren van Windows-applicatie of web-client
- Snelle installatie
- Hoge performance

Over de auteurs
Drs. L.J.G.M. Wiegerink is docent en cursusontwikkelaar aan de Open Universiteit. E-mail: leo.wiegerink@ou.nl. Drs. H. Pootjes is docent en cursusontwikkelaar aan de Open Universiteit. E-mail: harold.pootjes@ou.nl.

Ing. J.W. Bijpost is toolontwikkelaar bij Mattic Software. E-mail: jeanot@mattic.com.
Ing. M.H. de Groot is toolontwikkelaar bij Mattic Software. E-mail: marco@mattic.com.

Ontvangst met koffie in de Aula van de Universiteit van Amsterdam