

Programmeertalen zijn weinig anders dan een verzameling jargonuitdrukkingen in de Engelse taal. WOCAS4all vormt met de bijbehorende scripttaal Pronto één van de uiterst zeldzame uitzonderingen op deze regel. De uit 1996 stammende en voor die tijd zeer moderne omgeving wordt vrijwel uitsluitend gebruikt om applicaties voor woningcorporaties te bouwen en te onderhouden. Centric – sinds jaren eigenaar - werkt er bovendien hard aan om ervoor te zorgen dat de Nederlandse enclave binnen de computertalen klaargestoomd wordt voor de toekomst.

Modeldriven wordt modelgedreven

‘Het stimuleert om in het Nederlands over je omgeving na te denken’

De wereld van de woningcorporaties kent zo zijn eigen problemen. Ieder jaar stelt de Tweede Kamer vast met hoeveel de huren verhoogd mogen worden. Regelmatig veranderen ook de regels met betrekking tot onder meer toetsing van huren. Bezwaarschriften en uitspraken van huurcommissies zorgen voor nog meer eisen aan applicaties voor woningcorporaties. Deze vormden twaalf jaar geleden de aanleiding om een applicatie te ontwikkelen die uiterst eenvoudig aan te passen moest zijn. In 1995 bracht een groep woningcorporaties daarom een ontwikkelkrediet bij elkaar om een zogenaamde ‘applicatieshell’ te laten ontwikkelen. Het oorspronkelijke idee ging nog uit van C++ code, maar in het eerste jaar van de ontwikkeling besloot men tot een MDA-achtige aanpak. (Model Driven Architecture, red)

Huisman: “Het werd geen MDA in de klassieke zin van een PIM (Platform Independant Model, red)) met een platformtransformatie, maar een zogenaamde direct actieve omgeving. Je hebt een model, een modellaag in de database en een interpreter die het model rechtstreeks

leest. In die beginjaren waren er wel wat strubbelingen: gaan we er wel mee door of niet, willen we het tool wel hebben?



Leen Blom is verantwoordelijk voor research en development voor de twee softwaredivisies van Centric: IT Solutions en Software Engineering. MDE was en is een van researchspeerpunten bij Centric.



Edwig Huisman, werkt bij business unit woningcorporaties die onderdeel uitmaakt van de Centric-divisie IT Solutions. Hij is verantwoordelijk voor het technische portfolio van alle producten van woningcorporaties, in het verleden verantwoordelijk voor het MDA-tool waarmee WOCAS gebouwd is. Nu is hij verantwoordelijk voor de techniek in de brede zin.



Dré de Man
Tekst en foto's

‘Er bestaat geen **automatiseringsterm** waar wij geen Nederlands equivalent voor bedacht hebben.’

Maar al spoedig bleek dat je met zo'n omgeving een productiviteitswinst kunt halen die je met traditionele 3GL-tools niet kunt krijgen.”

De ontwikkelomgeving werkt met twee lagen in de database, waarbij één laag de modellaag is en de andere de data laag. Dezelfde applicatieshell (Pronto) zorgt ervoor dat er een ontwikkelomgeving kan draaien op basis van een metamodel, en dat de applicatie zelf op basis van het model kan draaien. Er is een BRE, een Business Rule Editor en een WOCAS, een WOningCorporatie Administratie-Systeem.

Huisman: “De grap is dat je op één desktop een ontwikkelomgeving kunt zetten,

een eindgebruikersomgeving, via de ontwikkelomgeving het model kunt aanpassen en dus rechtstreeks kunt testen wat je gebouwd hebt. Er zitten geen compilatieslagen tussen. Zo'n tachtig procent van wat de applicatie doet, wordt in het model ingevuld door middel van modellen, schermen, rapporten, brieven, batches, velden en knoppen. De rest wordt in een scripttaal ingevuld. De scripttaal is eigenlijk ontstaan omdat in de oorspronkelijke opdracht stond dat alles aanpasbaar moest zijn in het Nederlands. Dat namen we letterlijk. We hadden ook een vertaalster die alles naar het Nederlands omzette, zodat we geen *thin clients* maar *dunne klanten* hadden, en dat evolueerde. Programmeurs waren aanvankelijk

wat recalcitrant, iets wat sommige uitdrukkingen ook weerspiegelen, bijvoorbeeld: aantal gelezen regels in objectlijst. Er is een formele definitie opgesteld, en er is een parser geschreven. De standaardcommando's, de standaardstructuren, zijn vrij simpel en alle uitbreidingen zijn ook in het Nederlands geschreven. Het is een objectgeoriënteerde omgeving dus ook een objectgeoriënteerde taal. We hebben een aantal keer een presentatie gegeven voor Hogeschool Zuid-Holland en de TU Delft en daar was men heel verrast. Het is helemaal uitgebouwd, compleet met documentatie. De Nederlandse scripttaal vormt wel één van de bijzondere aspecten van deze omgeving. Uiteindelijk is het echter ook één van de aspecten die een beetje tegen je kunnen werken, omdat maar heel weinig mensen deze taal beheersen. Een van de ideeën is dan ook om er een C#-parser voor te zetten. In principe verander je dan niets aan de functionaliteit.”

Hoe kijken de mensen die er nu mee werken tegen aan?

Huisman: “Ja, die weten niet beter, die praten ook helemaal in die Nederlandse termen, ook daar waar vele onbekend zijn.



Edwig Huisman (links) en Leen Blom: “De mensen hier praten ook helemaal in Nederlandse termen”

Een tabel is een klasse en een view is een presentatieklasse, en zo hebben we overal Nederlandse termen voor. Soms zijn die duidelijk, soms moet je er even over nadenken en soms heb je er ook uitleg bij nodig. Een SQL-statement is een rekenregel. Tja, hoe kom je daarop? Maar je kunt zo wel mensen leren SQL te schrijven zonder dat ze ooit SQL gezien hebben. We kennen rekenregel invoeren, rekenregel uitvoeren, rekenregel triggers, rekenregel associaties, de connecties tussen de from-clausule en de where exists-clausule is dan een alias. Pronto verandert het dan in SQL. Je blijft op die manier ook modelmatig werken. Overal in het model zit ook referentiële integriteit. Zou iemand een kenmerk uit een klasse weggooien, een kolom uit een tabel met andere woorden, dan zou de omgeving daarvoor waken. Als je omgeving goed het type onderhoudswerkzaamheden ondersteunt, dus analyse, verbouwing, en controle, dan breng je de onderhoudskosten omlaag. Ook die controles hebben Nederlandse namen gekregen, het stimuleert om in het Nederlands over je omgeving na te denken. Ik zal nooit het woord deployment gebruiken, dat is Engels, bij ons heet dat uitrol. Er bestaat geen automatiseringsterm waar wij geen Nederlands equivalent voor bedacht hebben.”

Pronto werkt volgens het principe dat bij

Ruby Convention over Configuration heet: Huisman: “In het model staan vensters en velden en knoppen. Die kunnen ingevuld worden en hebben attributen in het model: wijzig, opslaan en sluiten. Als er geen scripttaal wordt ingevuld, dan is er sprake van standaardfunctionaliteit, dat geldt voor alles, die zit er al in. Als ik niets doe, dan wil ik bij een mutatievenster blijkbaar gegevens invoeren en opslaan. Op een venster met het attribuut queryvenster wil ik blijkbaar zoeken naar gegevens. Ik wil hier een venster hebben, ik wil een deel-scherm hebben en ik wil daar velden en knoppen hebben. Als ik geen afwijkende functionaliteit wil hebben, dan kan ik daar gewoon gegevens mee opslaan en dus binnen een uur voor een applicatie zoekschermen en mutatieschermen bouwen. Er ontstaat een nieuwe nomenclatuur in het

Nederlands met het uitgangspunt dat je het programmeren zo lang mogelijk uitstelt. Alles wat generieke functionaliteit is, zit in die applicatieshell.”

De logica zit dus voor een deel achter de schermen en voor een deel in de database?

Huisman: “De logica wordt in de database gegenereerd. We genereren uit het model stored procedures, ook vanwege de performance. Op de client zitten veldcontroles en logica van veldvolgordes. Maar grote stukken verwerking weer niet, want anders zou je grote hoeveelheden via de ODBC-connectie moeten ophalen en weer terugschrijven. Maandelijkse huurprolongatie, huurincasso, dat soort dingen gebeuren allemaal op de database. Uiteindelijk zit er altijd te veel logica in de vensters. Wel heb je daarop een soort bewaking. Omdat het hele model in de database zit, kun je met patroonherkenning dit soort dingen controleren en het onderhoud vergemakkelijken. In het geval van een bug kun je ook heel snel zien hoe vaak hij

‘De IDE waar de programmeur mee werkt hebben we zelf gemaakt met een meta-metamodel en daarboven bevindt zich een meta-meta-metamodel dat in zijn eigen staart bijt.’

voorkomt. In 1995 was dit revolutie. Toen was er zelfs nog wel twijfel of we dit wel aankonden? We maken financiële software en die moet betrouwbaar en saai zijn. We hebben daarvoor ook echt andere ontwikkelaars moeten inhuren. Afwijkende functionaliteit zit in de Nederlandse scripttaal, Pronto geheten. Iedere keer is dus de afweging: is het generiek of specifiek, want in dat laatste geval moeten we het met taal inrichten. Dat willen we zo weinig mogelijk. Desondanks is die taal behoorlijk uitgebreid om allerlei dingen te kunnen die je in de praktijk tegenkomt, want de praktijk is weerbarstig. De hele applicatie draait op basis van ODBC en er zit een database-onafhankelijke laag in.”

De applicatie is geschreven in C++, maar wat produceert hij uiteindelijk?

Huisman: “Een team van tien man heeft het in C++ geschreven en hij wordt nog steeds in C++ onderhouden. Wat eruit komt? Deels wordt er nog steeds model gelezen en van de scripttaal Pronto wordt een tussenvariant gecompileerd. Dat is nog net geen bytecode, maar een lijstachtige taal met heel korte commando's van twee of drie tekens lang die in een apart veld in de database opgeslagen wordt. Die kun je runnen zonder opnieuw dat script te interpreteren. Overigens hebben we voor het model ook een soort caching gemaakt uit de database in een binair bestand. Daardoor start de applicatie veel sneller op en het uitrollen wordt ook eenvoudiger.

De IDE waar de programmeur mee werkt, hebben we zelf gemaakt met een meta-metamodel en daarboven bevindt zich een meta-meta-metamodel dat in zijn eigen staart bijt. We hebben een programma waarmee je die modellen naar boven kunt kopiëren. Het ontwikkelteam maakt dus niet alleen die C++-bouwstenen, maar er zit ook nog een stukje ontwikkel-

werk voor de ontwikkelomgeving van de programmeurs die eigenlijk in zichzelf gedefinieerd is. Wensen of ideeën om het programmeren sneller te maken worden intern door de ploeg verwerkt, bijvoorbeeld voor extra versiecontrole op vroegere expressies. Er

zijn allerlei ontwikkelingen geweest om de productiviteit te verhogen. Omdat je dat zelf onder controle hebt, ben je ook niet afhankelijk van derden, zoals Microsoft of Oracle.”

Blom: “Het is mooi om te zien dat al in de periode 1995-2000 een aantal aspecten zitten die nu bij modeldriven engineering heel gangbaar worden. Domain Specific Languages onder andere: dat meta-meta-metamodel aan de bovenkant. Daaruit wordt een ontwikkelomgeving gemaakt waarmee de ontwikkelaars alleen dat kunnen maken wat je in die hogere taal hebt gespecificeerd. Het andere is dat je gebruikmaakt van frameworks. Dat is iets wat je tegenwoordig standaard ergens anders vandaan haalt. Met betrekking tot die wisselwerking van modellen en frame-

WOCAS4all: informatiesysteem en ontwikkelomgeving ineen

WOCAS4all biedt een complete set van basisfunctionaliteit, gecombineerd met een eigen, Nederlandse ontwikkelomgeving. De basisfunctionaliteit is gebouwd rondom het dienstenmodel, met woningverhuur als primaire dienst. In de geïntegreerde ontwikkelomgeving is zoveel mogelijk generieke functionaliteit ondergebracht in bouwstenen. De variabele functionaliteit wordt in de database vastgelegd door middel van business rules. Deze zijn een vertaling van het bedrijfsmodel van de corporatie. Ook is voorzien in een speciale Business Rules Editor (BRE4all). De totaaloplossing WOCAS4all is geïntegreerd met een document informatiesysteem (DIS), een geografisch informatiesysteem (GIS) en een technisch informatiesysteem (TIS).

works, eigen code en andermans code, begint een soort gemeenschappelijke aanpak te ontstaan in softwareland.”

Huisman: “Je ziet heel veel parallellen. Bijvoorbeeld doordat wij het mogelijk gemaakt hebben om .Net of ActiveX-componenten te integreren, zodat je kunt overstappen op andermans werk. Je ziet ook dat daar waar je dingen generiek wilt veranderen je dat in een klap in de shell kunt doen. Denk bijvoorbeeld aan vormgevingsaspecten. Geen enkele ontwikkelaar hoeft zich daarover zorgen te maken. Bij de volgende versie is het er gewoon. Je hoeft niet al je programma's langs, of alles om te zetten of te porten. Daar waar klanten heel specifieke generieke eisen hadden, bijvoorbeeld bepaalde financiële bedrijven zodanig splitsen in de boekhouding zodat ik kan bewijzen aan de accountant dat deze journaalposten nooit naar het andere bedrijf kunnen gaan, daar hebben wij zulke voorzieningen voor getroffen. Daardoor controleert de omgeving de programmeur zodanig dat hij niet van de rechte weg af kan wijken.”

Zijn woningcorporaties typisch Nederlands? Met andere woorden bestaan er ook buitenlandse producten?

Huisman: “Onze grootste concurrent, NCCW, heeft een Oracle-product uit Engeland gehaald en aangepast aan de Nederlandse markt. Wij hebben een nieuw product gebouwd op basis van onze ken-

nis van de Nederlandse volkshuisvesting. Dat is heel bijzonder, omdat de huurprijzenwet een grote rol speelt. Die maakt het bijvoorbeeld zelfs mogelijk om in afgesloten boekjaren nog volledige prijswijzigingen door te voeren. Klachten bij de huurcommissie kunnen er namelijk toe leiden dat de huurprijs met terugwerkende kracht, tot bijvoorbeeld twee jaar geleden, verlaagd moet worden. In Engeland heb je dergelijke uitzonderingen niet en in Zweden evenmin. Wat onze applicatie bijzonder maakt, is dat we het mogelijk gemaakt hebben om dat soort historische wijzigingen in te voeren en zowel qua geldigheid als qua transactietijd terug te halen. Alle dataklassen hebben twee timestamps en die kunnen op terugwerkende kracht-mutatie gezet worden. Dat betekent dat ik in het verleden kan muteren en dat er tussentijdse corrigerende journaalposten gemaakt worden en dat ik beeldschermrowsers terug kan kijken naar de situatie zoals die toen was, of zoals het toen had moeten zijn. Een soort data-warehousing in een notendop die ook nog eens aan alle eisen met betrekking tot controleerbaarheid en herleidbaarheid voldoet. Het heeft uiteraard ook een grote inspanning gevegd om alles in de applicatie als het ware historisch transparant te laten werken.”

Wie kan er nu business rules formuleren? Moet een programmeur dat doen of kunnen anderen dat ook?

Huisman: “Er zijn klanten die mensen in dienst hebben die dat zelf kunnen en in mindere mate kunnen klanten zelf regels aanpassen. Daar zijn verschillende niveaus in. De kleinere corporaties, die wat meer de focus hebben op hun basisproces, kunnen middels een paar simpele beheerschermen grenswaardes aanpassen, bijvoorbeeld de hoogte van bepaalde puntprijzen of andere parameters. Grotere corporaties die ontwikkelaars in dienst hebben, kunnen uiteraard veel meer aanpassen. De configuratiebouwsteen is in staat om modellen samen te voegen. Die modellen zijn allemaal SQL, dus je beschikt over de superset van de twee formuleren. Je kunt daar verschillend over denken, maar het was oorspronkelijk begonnen om de flexibiliteit. Het uitgangspunt was: de corporatie moet simpel en in het Nederlands alles kunnen aanpassen. Waar het om de eis met betrekking tot het Nederlands gaat is dat gelukt,

Centric en de markt

Centric heeft circa veertig procent van de markt voor woningcorporatiesoftware in handen. Grootste concurrent is NCCW, daarnaast duiken hier en daar ook SAP-gebaseerde oplossingen op.

maar het blijft toch programmatuur, functionaliteit. De winst zit echter in het feit dat wij heel snel kunnen ontwikkelen.”

WOCAS heeft een heel lange historie. Er zit heel veel kennis in. Maar hoe zou Centric het doen als het opnieuw gebouwd moest worden?

Blom: “In deze omgeving zit in feite het idee van DSL's, maar wel heel proprietary. We bestuderen nu DSL's. Aan de andere kant zie je frameworks. Klanten kiezen daar soms voor en dan moet je software daaraan voldoen. Zover is het nog niet, maar die tendens is er wel. Dus daar gaan we wat mee doen. De ontwikkelomgevingen worden steeds vaker gemodelleerd naar de DSL's. Dus als we opnieuw moesten beginnen, dan zouden we werken op basis van de standaarden die nu ontstaan in de wereld.”

Huisman: “Toen Pronto al bestond, kwam SOA op. De vraag was: gaan we dat zelf bouwen of gaan we standaardtechnologie gebruiken. We hebben voor dat laatste gekozen. We hebben besloten aan te sluiten bij de technologie van Microsoft en werken met Visual Studio 2008. We gebruiken DSL's en bouwen webservices ermee, in plaats van opnieuw het wiel proberen uit te vinden.”

Blom: “Grofweg zal het zo zijn dat onze applicaties integreerbaarder worden, door middel van standaardtechnieken zoals webservices, en als dat gebeurt zullen ze ook opgeknipt worden in functionele modules, en die kun je dan gaandeweg vervangen door nieuwere technologie, mocht dat nodig zijn. Dat is de weg die we opgegaan zijn.” «