

Met de komst van .NET waren volgens velen de klassieke 4 GL's definitief ten dode opgeschreven. RADventure, distributeur van Clarion en producent van maatwerkoplossingen, kwam op het idee om de voordelen van de 4GL te vertalen naar het .NET platform. Binnen enkele maanden ontwikkelde een klein team een goed werkende bèta van Fenix, een ASP.NET generator. Software Release Magazine sprak met ing. Erik Pepping en Sebastian Talamoni, respectievelijk algemeen directeur en R&D-manager van RADventure.

Interview

Fenix: een uit de as herrezen 4GL

Hoe RADventure van Clarion een .NET-tool maakt

Een zeer belangrijke rol bij de ontwikkeling speelde een speciale taal om codegeneratoren mee te schrijven. Volgens RADventure was dat geen opzet, maar de naam Fenix suggereert uiteraard iets dat uit de as herrezen is. Clarion zelf is verre van dood, er komt binnenkort zelfs een nieuwe versie uit. Dat Clarion behoorlijk vitaal is, blijkt ook uit het feit dat het RADventure twee keer in successie erin slaagde de RAD Race te winnen. Toch zagen bij RADventure belangrijke redenen om Clarion opnieuw geboren te laten worden als Fenix.

met .NET te kunnen werken is enorm. Er zijn drie- of vierduizend classes, ieder met tien, twintig of dertig property's. Dat levert minstens veertigduizend property's op. Zelfs al hebben ze verschillende database providers, toch heeft iedere provider eigen settings. Een cursus VB .NET programmeren heeft nauwelijks zin. Ik kan je dat wel in een uurtje uitleggen, maar dan heb je nog zes maanden voor het framework nodig. Met Fenix kun je in tien minuten je eerste applicatie bouwen, maar voor je als kersverse .NET programmeur op de gewone manier zo'n applicatie van scratch hebt gebouwd, ben je vier maanden verder.'

'De kennis die je nodig hebt om met .NET te kunnen werken is enorm'

TIEN MINUTEN Erik Pepping: 'We zijn het project om twee redenen gestart. De eerste reden is, dat de kring van potentiële afnemers van Clarion niet echt groeit: iedereen richt zich op Java of .NET. We geloven nog steeds in Clarion, maar er is teveel marketinggeweld waar we niet tegenop kunnen. Dus hebben we besloten zelf wat met .NET te gaan doen. Dat had voor ons als producent van maatwerkoplossingen het voordeel dat we onze eigen applicaties eenvoudiger konden verkopen. Voor wat betreft het product, zien we productiviteit als belangrijkste toegevoegde waarde ten opzichte van Visual Studio .NET. De kennis die je nodig hebt om

Sebastian Talamoni: 'Daarbij speelt niet alleen de complexiteit van de code een rol. Er is ook geen hulp om het programmeurs gemakkelijker te maken, er is geen application framework point of view. Alles is *loosely coupled*. Vanuit het designstandpunt uit gezien is dat heel prettig, maar wanneer je echt een applicatie moet maken, wordt het heel moeilijk voor een programmeur. Tien tot twintig procent van de programmeurs zijn goeroes, maar die anderen zijn geen experts. Dus die tachtig procent hebben een manier nodig om op professionele wijze applicaties van hoge kwaliteit te maken.'

RIJKE RESOURCES

Hoe begin je nu als Clarion-ontwikkelaar aan een applicatie die toch maar heel weinig met Clarion en heel veel met .NET gemeen heeft?

Pepping: 'Met de Clarion modellen en met zes boeken. ASPX for beginners, ASPX for professionals (lacht).



FIGUUR 1. Formulier voor validatie

Een boek over .NET, eentje over het framework, eentje over VB .NET, over C#, over...'

Talamoni: 'En natuurlijk heel veel netresearch.'

Pepping: 'Microsoft heeft veel informatie en er zijn heel veel andere sites met .NET informatie.'

Talamoni: "Four guys from Loyola" was een geweldige bron voor de datagrid. Ze hebben twintig artikelen over hoe je bijvoorbeeld een datagrid kunt programmeren, een erg rijke resource.'

TEMPLATE-TAAL Door de hoeveelheid resources op Internet ging het schrijven van Fenix sneller dan aanvankelijk gedacht. Een groot voordeel was ook, dat de kennis over .NET alleen maar in de templates gebracht hoefde te worden, en niet zoals bij 'echte' .NET developers, steeds ter beschikking moest staan. De template taal van Clarion was - bleek dan ook - een belangrijke factor. Met behulp daarvan is het mogelijk templates aan te passen, maar ook om geheel nieuwe templates te genereren en in dit geval dus zelfs een compleet nieuwe tool. Technisch gezien (juridisch ligt dat heel anders) zijn zowel Fenix als Clarion eigenlijk open source. Zaken als de dictionary en de windows worden aangeduid met symbolen. Het is ook mogelijk zelf symbolen te maken.

Talamoni: 'Misschien is het de eerste keer dat je het gebruikt niet zo vriendelijk, maar het is heel erg krachtig en het concept erachter is geniaal.'

Pepping: 'Je kunt er iedere denkbare code mee genereren. Heel flexibel en dynamisch. We doen er alles mee. Als ik een drop table voor Oracle moet hebben, dan maak ik daar een template voor, dat zijn drie regels. Ik maak zelfs scripts om te controleren of de database nog integer is op die manier. Eigenlijk begin je heel grof, en vervolgens steeds fijner. Je maakt steeds meer stukjes code en templates. Maar wanneer je iets fout doet, dan

kun je dat gewoon topdown aanpassen. Alles wordt iedere keer opnieuw gegenereerd, dus je zit in zo'n iteratief verbeteringsproces.

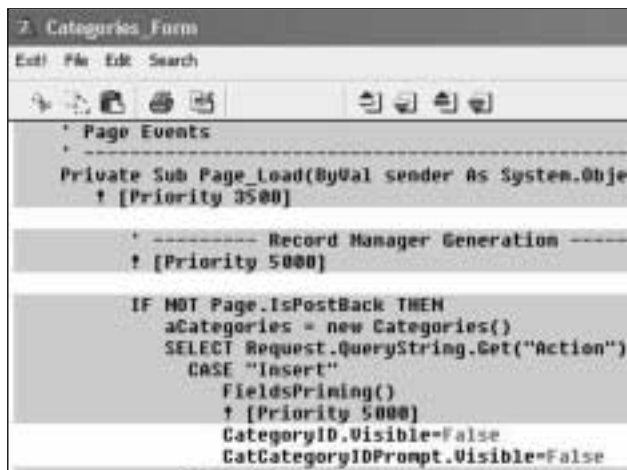
We kwamen er laatst achter dat de sizes van een bepaalde string iets te klein waren. Twee seconden later heb je dat aangepast, zet je de template ergens neer op de juiste plek, en het genereert weer de juiste code. Iteratief zit je in een verbeteringsslag, en tegelijk in een opbouwslag. We hebben bijvoorbeeld nog maar pas security toegevoegd aan Fenix. Hetzelfde geldt voor webservices: iedere keer wanneer een gebruiker er functionaliteit bij wil hebben, kunnen we dat heel snel toevoegen.'

Talamoni: 'De template-taal is de kracht van Clarion. Het is in wezen een verzameling blokken met hooks, referenties, die je als lego oneindig met elkaar kunt combineren. Met het zogenaamde hash embed statement kun je ook aangeven waar witte regels tussen de

'Iedere keer wanneer een gebruiker er functionaliteit bij wil hebben, kunnen we dat heel snel toevoegen'

gegenereerde code komt (witte regels zijn lege plaatsen waar je eigen code kunt toevoegen, red.). Zo kun je ook weer zelf templates schrijven, die bijvoorbeeld bij een bepaalde validatie iets extra's doen, en in alle windows terecht komen.

De template-taal is dan ook heel simpel, en heeft geen functies: alleen if then else, begin, case, tien woorden. Daarom maakt het voor ons ook geen verschil of we nu C# of VB .NET genereren. Overigens hebben we voorlopig voor VB .NET gekozen, maar dat is meer een marketing kwestie dan een technische.'



FIGUUR 2. De witte regels, waar plaats is om eigen code toe te voegen



FIGUUR 3. Global Embeds - File Manager's embed points

CODE DOM

Technisch gezien is zowel Fenix, Clarion als de template open source, in feite ASCII code. In principe is het daarom mogelijk alles aan te passen.

Pepping: 'Het is eigenlijk niet zo moeilijk. Moeilijk is

'Alles is loosely coupled; vanuit het design-standpunt gezien is dat heel prettig'

alleen het begrijpen van het concept. Het is in ieder geval abstracter dan code schrijven, want je genereert code. Je moet dus alles variabel maken, je moet altijd

denken: deze file, dit is er eentje uit de dictionary, dus het is een variabele. Dus wordt die specifieke file afhankelijk van de context ingevuld. Als ik in browse klanten zit, dan staat er customer in plaats van %file, dat is een slag die voor sommige mensen moeilijk kan zijn.'

Er is binnen .NET een technologie die code dom heet. Die geeft ook de mogelijkheid om code te genereren.

Talamoni: 'In het begin hebben we daar ook serieus naar gekeken, als alternatief voor onze template-taal. Wat ze in feite ermee proberen te doen is hele kleine structuren te genereren zoals: hoe kan ik een 'if' statement genereren, of hoe kan ik een case statement genereren. Het idee van de code dom is dat je kunt schrijven in een taal die onafhankelijk is van de echte .NET taal. Uiteindelijk is dat niet geschikt voor ons doel, wel biedt het handige dynamische code executie en compile en run functionaliteiten.'

CLARION TEMPLATES

De verzameling Clarion templates is enorm. Niet alleen is er een grote verzameling templates die standaard meegeleverd worden met Clarion, er is ook een grote hoeveelheid third party templates. Welk gedeelte daarvan is opgenomen in Fenix?

Pepping: 'Van de templates die kant-en-klaar meegeleverd worden ongeveer zestig procent, maar van de third party templates een stuk minder. Ook dat is een grote waardevolle verzameling. Maar er zijn natuurlijk een hoop templates in Clarion die we nu niet meer zullen gebruiken, bijvoorbeeld om Word aan te roepen. Er zullen ook nieuwe worden toegevoegd, die de functionaliteit van



Erik Pepping, algemeen directeur van RADventure



Sebastian Talamoni, R&D-manager bij RADventure



FIGUUR 4. Hotfields is één van de aardige advanced features van Fenix

het oude Clarion object framework te boven gaan, met name webservices.'

Vergeleken met Fenix moet je in Visual Studio .NET veel programmeren. Toch geeft Visual Studio .NET wel een productiviteitsverbetering, wanneer je het vergelijkt met VB 6.0.

Pepping: 'Ja, dan scheelt het wel. VB 6.0 programmeurs deden niets anders dan programmeren, afgezien van 'button click' werd er vrijwel niets voor ze gegeneerd, dus die gaan er wel op vooruit. Voor hen is het mooi. Als ze Fenix zien zeggen ze ook eerst: "maar dat kunnen wij nu ook." Als ik dan zeg: programmeer dan de functionaliteit maar die wij er nu in hebben zitten, dan komen ze erachter: "hé, ik moet dit zetten en dat zetten en dat object is wel een object, maar ik moet het helemaal aansturen". Het is allemaal vrij vervelend wanneer je zaken als sorting of het bewaren van sessions handmatig moet aansturen. Om maar te zwijgen van de meer advanced features, zoals hot fields in tabellen en record selectie. Eigenlijk stuur je de onderliggende component aan. Zo'n datagrid is wel een component, maar er zitten zoveel property's en methods in, dat je toch moet programmeren om het zo aan te sturen dat je zo'n hot field krijgt. Daarmee stuur je het gedrag. Het is een beetje zoals je kinderen opvoedt. Daar moet je ook steeds van alles tegen zeggen, en tegen de tijd dat ze twintig zijn heb je hele encyclopedieën tegen ze gesproken. Wij genereren dat allemaal.'

ALLE FRAMEWORK OBJECTS

Een belangrijke vraag bij code generatoren is ook, in hoeverre de functionaliteit beperkt is. Het feit dat op vrijwel iedere plaats in de gegenereerde applicatie handgeschreven code toegevoegd kan worden, heft al zeer veel beperkingen op.

Maar welk gedeelte van het .NET framework is standaard opgenomen in Fenix?

Pepping: 'Wij gebruiken zeer veel framework objects. Alles in het framework is een object, zelfs een kleur is een object. Browse, datagrid: bijna alle calls zijn framework calls. Je hebt het framework voor vrijwel alles nodig.'

Fenix biedt net als Clarion geen grote ondersteuning van modellering?

Pepping: 'Wij gaan niet van start met een conceptuele design-fase, bij ons heb je dat min of meer al gedaan. De generator is een abstractieniveau van de coding. De design is een ander abstractieniveau. Op databaseniveau is die koppeling er wel. Daar kun je een datamodel creëren of importeren.'

Talamoni: 'Je kunt wel een vreemde dll met classes importeren. Fenix kan zelfs C# met VB modellen mixen, wanneer je bijvoorbeeld een lege C# source module wilt maken met wat basiscommentaar. We ondersteunen dus het concept om vreemde code in een feitelijk lege klasse te importeren.'

PRAGMATISCHE AANPAK

Door het bouwen van een .NET generator krijg je natuurlijk een aardig beeld van het framework. Is dat beeld positief?

Pepping: 'Ja, de eerste keer dat ik het zag dacht ik: dit is een kopie van Java. Alles was zo identiek: C#, het con-

'Tien tot twintig procent van de programmeurs zijn goeroes, maar die anderen zijn geen experts'

cept van de infrastructuur, het idee van een heel kleine taal en een heel groot framework, zelfs de namen van de objecten en de klassen. Bijna alles komt uit de Java wereld, maar je herkent toch ook wel de pragmatische Microsoft-aanpak.'

De bètaversie van Fenix is nu al beschikbaar. De definitieve versie zal naar alle waarschijnlijkheid in juli verkrijgbaar zijn. RADventure zal dit jaar tijdens de RAD Race gebruik maken van Fenix.

Tekst en fotografie: Dré de Man