

EDCubed kwam als overwinnaar uit een erg spannende RAD Race editie

2003. Een reden genoeg voor Software Release Magazine om deze tool grondig aan de tand te voelen. EDCubed kwam aan de start met The EDCubed Technology, kortweg "TET". Ook in de editie 2002 was al opgevallen dat deze tool, of beter, deze omgeving duidelijk mikte op het zwaardere werk en heel wat in zijn mars had. En niet in het minst viel op, dat het hier een code generator betrof.

Bespreking

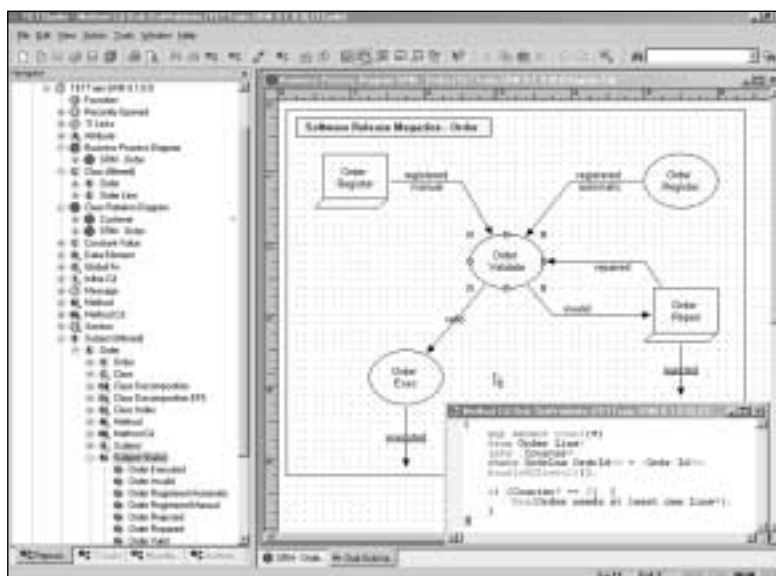
# EDCubed Technology: kracht en diepgang

## Code genereren op modelbasis

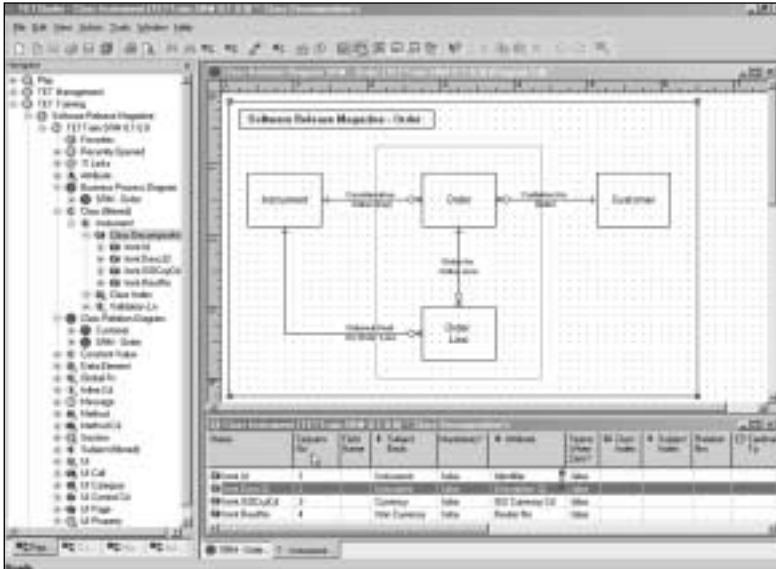
Nu hebben code-generatoren altijd een intrinsieke voorsprong op hun meer klassieke collega's. Gesteld dat de code correct wordt gegenereerd, dan kan er moeilijk iets mis gaan. De kneep zit hem natuurlijk in het goed en gedetailleerd aansturen van de generator zodat er precies de gewenste functionaliteit uitrolt. En daar wringt meestal de schoen: ofwel wordt het gedrag van de applicatie te generisch ofwel moet er oeverloos gespecificeerd worden om elk aanstuurbaar element vast te leggen. Dan nog kan alleen zeer ver maar nooit 100% codevrij gewerkt worden en is er altijd wel ergens een stuk handwerk nodig. Het moeilijkste is dan om dat op de goede plaats in het grotere geheel van de gegenereerde code te doen terechtkomen. Of hoe complexiteit nooit verdwijnt maar alleen verplaatst wordt. De vraag is altijd hoeveel code hiervoor nodig is en van welk type en complexiteit die is. Betreft het SQL statements, dan kan menig programmeur daar perfect mee weg. Maar wat als het meer wordt? Benieuwd hoe dit in TET zit. Software komt nooit vanzelf, maar dat wist u al.

**COMPONENTEN** De TET omgeving valt uiteen in drie grote componenten. De meest zichtbare daarvan is de TETStudio, waarin de programmeur werkt met de diverse editors die hun informatie opslaan in de TETRepository. Die wordt vervolgens uitgelezen door de TETFactory om de applicatiecode te fabriceren. Daarvoor worden eerst alle aangemaakte modellen gecheckt op hun volledigheid en integriteit; als alles in orde is wordt er een proprietary platformafhankelijke geparseerde tussencode aangemaakt. Het werk van de Expander is daarmee afgelopen, waarna de TETRenderer

instaat voor het genereren van de platformafhankelijke sourcecode voor het gewenste target platform. TET mikt dus duidelijk op het zwaardere werk en in het achterhoofd komen termen als "uit de kluiten gewassen"; "upper en lower CASE" spontaan opborrelen. Dat is blijkbaar ook in lijn met wat de TET ontwikkelaars voor ogen hadden: elk principe of theoretisch concept meenemen (event driven, data driven, OO maar evenzeer het afdekken van de life cycle van een object en het herbruiken ervan) zolang het maar bijdraagt tot het doel van de (hogere) productiviteit. Voorwaar een nobel streven, waaraan de eis van een formele én complete repre-



FIGUUR 1. TETstudio met daarin de BPR editor geopend en een stuk Tcode



FIGUUR 2. Class Relation Diagram editor van de TET Studio

sentatie van de applicatie werd toegevoegd. En vanaf daar moet het én volledig én automatisch gaan om zowel de applicatie - als de user interface logica van de mainstream platformen te realiseren. In de praktijk zit dat erg goed: het obligate Windows (32 bit only) maar evengoed Unix (IBM AIX 4.3 en Linux kernel 2.4, getest met Red Hat Linux 8.0) worden afgedekt maar ook het OS390 op de IBM mainframe. En dat is een categorie waar menig andere toolleverancier van droomt maar meestal voor past. TET bevindt zich daarmee in goed gezelschap van bijvoorbeeld het vroegere Texas Instruments Information Engineering Facility (IEF) dat

die pas hanteerbaar worden na een hoog aantal overervingen - maar zinvol inzetbare elementen. De Global repository bevat ook de autorisatie informatie over wie wat mag doen in de repository. Elke (zelfgemaakte) repository gaat hier automatisch van overerven. Als de TET Studio versie wijzigt, dan geldt dat ook voor de set van design patterns maar die zijn wel downward compatibel opgezet zodat de opgeslagen modellen niet hoeven te veranderen. Alvast een prima aanpak. De Play repository is aanvankelijk leeg en dient exact daarvoor: om rond te spelen en op te bouwen vooraleer over te gaan naar de productieve versie. EDCubed gaat ook hier verder dan heel wat tools en heeft overal een eigen versioning systeem ingebouwd dat tot op het laagste niveau en zeer granulaair is uitgewerkt. Het volstaat om een element een eerste keer te editen en de versie wordt aangemaakt. Dit is duidelijk in de design fase meegenomen en niet zomaar achteraf toegevoegd om het hokje over "versioning" te kunnen afvinken bij het vergelijken van features.

**EVENT DRIVEN** Het echte werk begint bij het vastleggen van het Business Process Diagram in de bijhorende editor. Het komt er op neer dat de Business en de ondersteunende objecten gemodelleerd worden in Subjects. Daarbij wordt de MDA filosofie van de OMG groep gevolgd. Een Subject krijgt een naam, een top level Class en eigen methoden, die alle verplichte en optionele faciliteiten van de life cycle van het subject moeten ondersteunen. In het fysiek systeem dat later uitgerold wordt, resulteren de Methoden in processen.

Processen communiceren op intrasysteem niveau, directe calls of door middel van database triggers of stored procedures. Elke intra process communicatie is binnen TET opgezet door het lezen en schrijven in queues die omwille van performantie in core memory staan. Elk process heeft één ingaande queue en mogelijk meerdere output queues; maar het houdt ook een aparte queue bij voor error logging van logische en fysieke fouten. Elk met TET gebouwd systeem is *de facto event driven*. De status van het subject wordt bijgewerkt en weggeschreven in de onderliggende relationele database. Mocht er iets misgaan, dan wordt een process hervat vanaf de meest recente teruggevonden status van een Business object. Het programma dat door de TET Expander en Renderer wordt gegenereerd, spuwt dan code uit die de IBM MQSeries (vanaf versie 1.3, alleen het messaging gedeelte van dit steeds groter wordende product is echt nodig), J2EE; WebSphere of WebLogic (J2EE compliant) uitrolt zonder enige extra inspanning.

De queueing, messaging en load-balancing middleware wordt op haar beurt gecontroleerd vanuit informatie die wordt betrokken vanuit een 7 x 24 uur opera-

## De kneep zit hem natuurlijk in het goed aansturen van de generator zodat de gewenste functionaliteit er uitrolt

steunde op de concepten van James Martin en na overnames nu onder dak zit bij Computer Associates en onder de naam Advantage- (voorheen Cool) Gen door het leven gaat. Het vermoeden is dat beide tools nogal wat vergelijkbare eigenschappen kunnen hebben.

**COCKPITGEVOEL** Dat krijg je wanneer TETStudio wordt opgestart dat de toegang tot alle onderliggende elementen bevat. Uiterst links worden de beschikbare repository's opgelijst. TET Global is de fundamentele repository die standaard wordt meegeleverd en onder andere de "TETCodeTables" bevat waarin elementaire, veelgebruikte (subject) definities voor het grijpen liggen. Hier - in tegenstelling tot nogal wat OO tools - geen primitieven op een extreem hoog abstractieniveau





FIGUUR 4. De UI painter met de Tcode

wordt de datastructuur afgelijnd door vastleggen van attributen en elementen als Primary Key, Candidate Key, (non) Clustered Index et cetera. Dit alles gebeurt op een logisch niveau in de tool dat niets van doen heeft met de onderliggende deployment database. De cardinaliteit van alle relaties wordt verplicht meegenomen in de (logische) modellering zodat de codege-

## Variabelen en dergelijke die van een hoger of ander niveau worden overgeërfd krijgen een speciale aanduiding mee

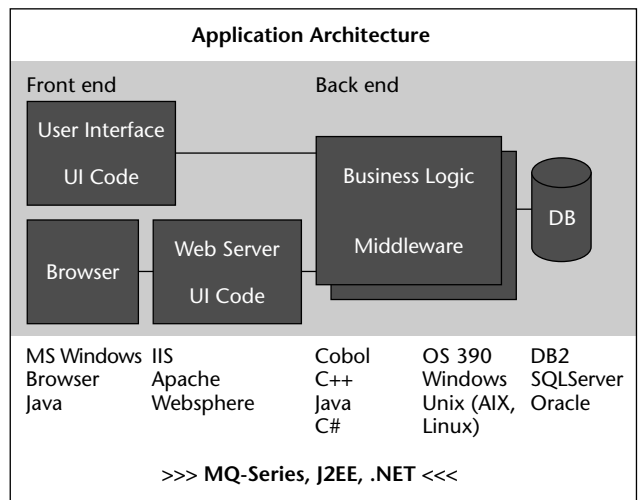
nerator daar later de DDL voor een optimale fysieke database mee gaat uitrollen. TET biedt vanzelf de voor de hand liggende attribuuttypes aan, ook weer los van enige specifieke database implementatie. Overigens levert TET tegen elke (ondersteunde) database SQL af zonder verplichte cursorgeneratie en dat scheelt op de performantie.

Als de modellen de interne consistentiecheck passeren, kan de generatiefase opgestart worden door middel van een Output ReQuest (ORQ of "ork") aan de hand van een bijbehorende template (voor de ondersteunde elementen: zie het kader aan het eind van dit artikel). Er kan gekozen worden tussen debug en release mode code die ook netjes in een ZIP file wordt neergezet. Vervolgens wordt de betreffende compiler (Visual Studio .NET in ons geval) opgestart en uiteindelijk wordt een "System Part" opgeleverd met daarin al het nodige om de business logica asynchroon op te roepen vanuit een queue, een andere methode of de User inter-

face. Daarnaast worden de UI's gebundeld in een "UI Part"; elk Part kan als applicatie of als componentenset opgeleverd worden. Voor de duidelijkheid: nu wordt alleen de .NET compiler gebruikt.

**TOETERS EN BELLEN** U heeft gelijk: het lijstje is obligaat en hier gaan we dan. UML? Ja en nee: EDCubed kiest voor een aanpak die het beste van dit verhaal afroemt maar gaat hierin niet blindelings meelopen want "...zelfs UML 2.0 capteert volgens ons niet alles wat nodig is om tot een applicatie te komen die volledig uit een model te genereren valt..." Anderzijds is men ook niet vies van een experimentje. Zo werd recent een business case vanuit Rational Rose door middel van een XML interface geïmporteerd in TET (UML 1.4), overgezet in een subject model en van daaruit met succes generender naar onder andere COBOL. In de volgende versie van TET (a rato van twee major releases per jaar) zal deze UML import officieel gemaakt worden. Halverwege 2004 moet ook de "echte" .NET ondersteuning er zijn. Daarbij zal zo veel mogelijk gebruikt worden wat beschikbaar is met zo weinig mogelijk "TET eigen" middleware. In eerste instantie komt er een C# renderer die managed code gaat opleveren maar het zou even goed een VB renderer kunnen gaan worden gezien de taalonafhankelijkheid van .NET. Pragmatisme alom dus. Dat geldt ook voor het omgaan met het OO paradigma: meenemen wat snelle en robuuste systemen oplevert en als zekere OO elementen dat niet doen, dan worden die niet gebruikt. Het doel voorop, de middelen zijn ondergeschikt.

**SPIERBALLEN** TET moet duidelijk in de categorie van "tools met spierballen" ingedeeld worden. Code genereren op modelbasis is op zich al minder evident, maar dat ook nog doen tegen de koning van alle platformen en binnen de krijtlijnen van wat men daar



FIGUUR 5. The EDCubed Technology - TET architectuur

gewoon is, doet niet iedereen na. En evengoed het hippe Java verhaal volgen. Natuurlijk zal de leercurve niet voor de hand liggen, maar dat zal ook niet tientallen malen erger zijn dan meer courante tools. Het feit dat EDCubed met een in Nederland ontwikkelde tool nu al doorgedrongen is in de markt van de bancaire applicaties én op het mainframe, boezemt vertrouwen in want dat soort klanten gaat niet over één nacht ijs. Goed leesbare COBOL blijft immers belangrijk en zeker als dat kan op basis van een UML model. De firma heeft ook volledige modellen beschikbaar en verkocht voor bijvoorbeeld "clearing and settlement" en ook dat weten banken en zware COBOL klanten te waarderen. In dit marktsegment is er ook bijzonder weinig keuze en EDCubed verdient het zeker om op de shortlist naast een tool als CA AdvantageGen te staan als je met dit soort van features van doen hebt. Het wekt dan ook geen verwondering dat grote (Nederlandse) systeemhuizen steeds nadrukkelijker het potentieel van deze tool bestuderen als alternatief voor hun courante COBOL ontwikkelstraten.

*Ivan Verborgh is onafhankelijk IT deskundige en docent aan de Provinciale Hogeschool Limburg (België).*

## TET Studio

Gebruikt zo veel mogelijk algemene standaarden zoals ANSI-SQL, J2EE en HTML. Specifieke, leverancier afhankelijk functionaliteit wordt zo veel mogelijk genegeerd. Bij de generatie kan gekozen worden uit :

### OS

Microsoft Windows (geadviseerd: NT sp4 en hoger, W2000, XP, W2003)  
IBM AIX 4.3  
Linux kernel 2.4 (getest met RedHat 8)  
IBM OS390 (Cobol 85 in combinatie met DB2)

### Browsers

HTML 3.2 compliant browsers (getest met MS IE versie 5 en hoger)

### Middleware

MQSeries vanaf versie 1.3  
J2EE  
WebSphere (J2EE compliant)  
WebLogic (J2EE compliant)  
Apache vanaf versie 1.3  
IIS vanaf versie 4

### Databases

DB2 vanaf versie 5  
Oracle vanaf versie 8  
SQL Server vanaf versie 6.5  
MS Access vanaf MSOffice 97

### Programmeertaal

Cobol 85  
C++ (.NET 2000 Studio, .NET 2003 Studio, GNU 3.0.2 (LINUX), VisualAge C++ Professional / C for AIX Compiler (Version 6))  
Java (1.4)  
HTML + Java script (3.2)

**PATCHES**

*Patches*

**PATCHES**

*Patches*

**PATCHES**

*Patches*

**PATCHES**

### webMethods introduceert op webservices gebaseerde Portal-oplossing

webMethods maakte onlangs de beschikbaarheid bekend van webMethods Portal. webMethods Portal is een volledig op webservicestandaarden gebaseerde Portal-oplossing, die alle informatie, toepassingen en processen binnen een onderneming toegankelijk maakt voor medewerkers. webMethods Portal is gebaseerd op webservicestandaarden, waardoor geprofiteerd kan worden van reeds gedane investeringen in applicatie-integratie. Alle uitgevoerde acties (zoals het publiceren, bekijken, bijwerken en verwijderen van informatie) worden als een webservice via de WSDL-standaard gepresenteerd. Hierdoor kunnen bijvoor-

beeld webservices van derden binnen de eigen portal veilig worden aangeroepen. Ontwikkelaars kunnen met webMethods Portal snel en eenvoudig onderdelen aan een eigen Portal toevoegen of koppelen aan bestaande bedrijfsonderdelen. Het toevoegen van nieuwe Portal-onderdelen gebeurt via zogenaamde portlet-wizards, die via een drag-and-drop interface de ontwikkeling vereenvoudigen en versnellen. En via de uitgebreide lijst van voorgedefinieerde portlets, kunnen eenvoudig en snel koppelingen worden gelegd naar bedrijfskritieke applicaties zoals SAP, Siebel, Microsoft Exchange of Lotus Notes. webMethods Portal kan zowel stand-alone als in combinatie met het webMethods Integratieplatform

worden gebruikt en kan zowel met .NET als J2EE overweg. webMethods Portal ondersteunt tevens nieuwe Portal-standaarden als WSRP en JSR 168.

### Gupta's Chuck Stevenson in Top 25 CTO 2004

Chuck Stevenson CTO van Gupta Technologies is door InfoWorld gekozen in de Top 25 Chief Technology Officers 2004. Infoworld: 'Waarschijnlijk werd geen CTO in 2003 met afschrikwekkender financiële beperkingen geconfronteerd dan Charles W. Stevenson. Hij bracht de ontwikkelkosten terug met twintig procent en de kosten voor support met veertig procent. Ondertussen heeft het twintig miljoen grote bedrijf tien kwartalen achter elkaar zwarte cijfers geschreven. Gupta werd popu-

lair door een van de eerste 4 GL's, SQLWindows, en haar 'onderhoudsvrije' database SQLBase. Nadat tijdens de Internet hype het toenmalige management de focus verlegde naar mobiele toepassingen, werd het onderhoud van de klassieke Gupta producten verwaarloosd, waardoor het bedrijf bijna ten onder ging (zie [www.release.nl](http://www.release.nl) > online archief: De reïncarnatie van Gupta, 3/2002). Stevenson – oorspronkelijk database ontwikkelaar - zorgde er met hulp van Platinum Equity voor dat het verlies van bestaande klanten een halt toegeroepen werd en dat een deel van de oude klanten terugwonnen werd. Zowel SQLBase als Team Developer werden op belangrijke punten verbeterd.