

De Eclipse Summit Europe – die vorige maand werd gehouden – is de belangrijkste Eclipse-conferentie aan deze kant van de Atlantische Oceaan. Voor Jos Warmer was deze bijzonder interessant, omdat zijn werkgever Ordina een aantal open source plugins voor Eclipse aan het ontwikkelen is. Dat was voor hem de reden om af te reizen en een kijkje te nemen bij deze conferentie. Hij doet verslag van zijn indrukken en bevindingen.

Eclipse Summit 2008

Platform wordt steeds breder

De Eclipse Summit werd gehouden in Ludwigsburg in Duitsland, dus besloot ik met de auto te gaan. De avond voor vertrek las ik nog net op tijd dat je in Ludwigsburg alleen mag komen als je een zogenaamde Umwelt-sticker op je auto hebt. De route maar aangepast, zodat ik onderweg langs een TÜV keuringsstation kwam waar ik zo'n sticker kon ophalen. Toen ik om een Umwelt-sticker vroeg, werd mij fijntjes toegevoegd dat dit een Feinstaubplakette heet. De man leek bijna beledigd door mijn woordgebruik. Het duurde gelukkig maar vijf minuten en nu ben ik een mooie groene sticker op mijn auto rijker.

Modeling Symposium

De Eclipse Summit begon met een dag van symposia, waarbij verschillende onderwerpen aan de orde kwamen. Zelf ben ik natuurlijk naar het Modeling Symposium gegaan. De ochtend werd gevuld met korte presentaties van vijftien minuten. Opvallend was het feit dat de presentaties gingen over geavanceerde onderwerpen, niet zozeer over 'kijk eens hoeveel code ik genereer'. Het lijkt erop dat de modeling community snel volwassen begint te worden. Met een aantal deelnemers dat drie keer zo groot was als vorig jaar, en dat in deze moeilijke tijden, is de populariteit van modeling duidelijk groeiende. Zelf mocht ik de laatste presentatie voor mijn rekening nemen om te vertellen over Mod4j (Modeling voor Java), het open source modeling project dat we bij Ordina aan het ontwikkelen zijn.

De middag werd gevuld met discussiegroepen over onderwerpen als 'modelleren versus programmeren', 'modularisatie van Domain Specific Languages (DSL's)', 'Evolutie van DSL's'. Op het symposium zijn ook concrete acties afgesproken.

Zo gaan Sven Efftinge en ondergetekende een voorstel schrijven voor een Eclipse Modeling project. Ook is afgesproken om gezamenlijk een online boek over best practices in EMF (het Eclipse Modeling Framework) te schrijven. Het idee komt van Ken Hussey, maar hij heeft geen tijd om het helemaal zelf te doen.

Interessante presentaties

De tweede dag begon met een keynote van Dave Thomas. Dave is één van de mensen die aan de wieg van Eclipse hebben gestaan. De keynote ging in op embedded development. De spreker wist op overtuigende wijze het publiek te vertellen dat de enorme onnodige complexiteit die we binnen de Java wereld gecreëerd hebben nergens goed voor is. De aanwezige Java ontwikkelaars moesten zich maar voorbereiden op vragen van hun (klein) kinderen als *'opa, heb jij echt Java gedaan?'*, *'En waarom was dat allemaal zo ontzetten ingewikkeld? Kijk maar hoe eenvoudig ik het kan!'*. Java als taal heeft volgens Dave geheel gefaald. Het was eigenlijk bedoeld als taal voor set top boxes, later voor mobile applicaties en heeft het in beide gevallen niet gehaald. Gelukkig heeft de enterprise wereld Java gered. Dit klinkt wel erg overdreven, maar raakte bij mij een gevoelige snaar. Ik vraag me ook wel eens af waarom we het zo verschrikkelijk complex maken. De toekomst van embedded en mobile devices ligt volgens Dave bij C++ in combinatie met JavaScript. Dave eindigde met de relativerende opmerking dat hij maar een professor was en dat die natuurlijk niets van de echte wereld begrijpen.

Enkele interessante, meer business gerichte, onderdelen van Eclipse zijn BIRT en Visual Rules. Beide onderdelen zijn niet alleen ontwikkeltools,

Jos Warmer

is Partner bij Ordina
(jos.warmer@ordina.nl)

maar een combinatie van een ontwerptool in Eclipse en een runtime welke in een gewone Java applicatie werkt.

Visual Rules is een tool waarmee met behulp van visuele modellen business rules gemodelleerd kunnen worden. Een interessante feature is onder meer de mogelijkheid tot het definiëren van unit tests voor business rules. Hiermee kun je op veilige wijze regels wijzigen en de gevolgen direct zien. Een ander aspect dat opviel was de mogelijkheid tot het tonen van statistieken van toepassing van regels in het visuele business rules model zelf. Hierdoor zie je direct welke regels vaak of juist heel weinig gebruikt worden. Toepassing van deze visualisatie op één case geeft een helder beeld van het gevolgde pad welke gebruikt kan worden om uit te leggen waarom een bepaald antwoord gegeven wordt. De modelleeromgeving is geïntegreerd met Eclipse, de runtime is gewoon Java. O ja, Visual Rules is helaas geen open source.

Voor het genereren van documentatie maakt Visual Rules gebruik van BIRT. BIRT is een reporting tool dat wel open source is, hoewel er ook een commerciële versie bestaat. BIRT kan wat faciliteiten betreft concurreren met commerciële tools en is binnen de Eclipse community steeds populairder aan het worden. De BIRT designer maakt het mogelijk om snel en zonder programmeren rapporten te genereren. Als output is een groot scala aan formaten beschikbaar zoals onder meer HTML, PowerPoint, Word en PDF, etc. Na een demo van vijftig minuten is duidelijk dat het aantal mogelijkheden overweldigend is. In de demo werden in een razend tempo alleen de *nieuwe* features van de laatste versie getoond. Naast de traditionele rapport generatie wordt BIRT binnen een aantal andere Eclipse projecten gebruikt voor het genereren van documenten. Als je op zoek bent naar een open source rapport generator is BIRT zeker de moeite waard om eens te bekijken.

Het nieuwe Rational Team Concert product van IBM Rational gebruikt ook BIRT voor het maken van rapporten op het visuele dashboard. Rational Team Concert, ook wel Jazz genoemd, is een omgeving voor het ondersteunen van grote gedistribueerde teams, Het Jazz team zelf is zo een team en het proces dat Jazz standaard ondersteund is dan ook in grote mate gebaseerd op de eigen ervaringen in het Jazz project. Aan de ene kant goed om je te baseren op ervaringen, aan de andere kant is dit wel een erg beperkte blik. En natuurlijk van belang, Rational Team Concert is geen open source.

In *“The Unbearable Stupidity of Modeling”* geeft Ed Merks een uitstekend overzicht van veel misvattingen rond modelleren. Ed is “Mister EMF”, de geestelijk vader en hoeder van het Eclipse Modeling Framework. In die hoedanigheid is hij in de loop der jaren alle mogelijke domme vragen over modelleren wel tegengekomen. Een van de mooiste was de misvatting “modeling is complex”. Het antwoord daarop is simpelweg “If modeling makes thing more complex, you are doing something wrong.” Een waarheid als een Hollandse koe! De keynote spreker Dave Thomas merkte op dat modeleren niets is maar domain specific languages een uitstekende oplossing zijn. Weer zo een misvatting die Ed ontkrachte, domain specific languages zijn immers ook modelleertalen dus is werken met DSL's ook modelleren. Een vermakelijke, maar ook leerzame sessie. Misschien leuk voor ons Nederlanders, Ed is zelf Nederlander, hoewel hij al vele jaren in Canada woont.

User Interfaces

Op het gebied van user interfaces zijn er meerdere parallele ontwikkelingen gaande binnen de Eclipse wereld. Het Eclipse Rich Client Platform (RCP) wordt steeds vaker gebruikt voor het ontwikkelen van desktop applicaties. Ruim de helft van de aanwezigen was met een of meer projecten op dit gebied bezig. Zou Java met RCP toch weer een serieuze kandidaat worden voor de desktop?

Dave Thomas, keynote spreker tijdens Eclipse Summit 2008



In het verlengde van RCP is een interessant project gaande RAP geheten. RAP staat voor Eclipse Rich Ajax Platform. RAP is dus een user interface framework voor gebruik in web applicaties. RAP maakt gebruik van hetzelfde programmeermodel als RCP, jawel het Rich Client Platform van Eclipse voor de desktop. Dat betekent dat RAP alle rijkdom van desktop applicaties heeft, je kunt hier dus een uitstekende rich web client mee bouwen. Omdat het programmeermodel dat van een desktop applicatie volgt is het in vergelijking met sommige Web frameworks relatief eenvoudig in gebruik. RAP is bovendien nu al bijna honderd procent source code compatible met RCP, waardoor je met één source zowel een desktop als een Web applicatie kunt ontwikkelen. In het geval van RAP blijft de client wel een thin client, dus alle processing vindt gewoon op de server plaats. RAP heeft dit voor elkaar gekregen door een speciale implementatie van de onderliggende SWT toolkit van Eclipse te maken.

Een andere ontwikkeling is de wijze waarop de user interface van de Eclipse workbench zelf ontwikkeld gaat worden. In de incubatie fase voor Eclipse 4.0 worden hier veel heel nieuwe technieken gebruikt die veel meer flexibiliteit mogelijk moeten maken.

Echt eenduidig is de richting van user interfaces nog niet. Vandaar dat er binnenkort ook een voorstel gedaan wordt voor een nieuwe Eclipse project, het Presentation Modeling Framework, kortweg PMF genoemd. Dit moet een framework worden waarmee op hoog niveau user interfaces gemodelleerd kunnen worden, die vervolgens naar verschillende technologische user interface frameworks code kan genereren. Een van de indieners is weer een Nederlander, Jim van Dam.

De toekomst van Eclipse

In een aantal sessies werd ingegaan op de vraag: 'Waar gaat Eclipse naartoe?' Voor de komende jaren is in juni 2009 versie 3.5 (Galileo) en juni 2010 versie 3.6 gepland. Deze versies zullen voortborduren op de huidige Eclipse (3.4) en vooral de focus hebben op betere bruikbaarheid, performance, gebruik van modernere technologie, portabiliteit, etc. Geen schokkende zaken, maar een gestage verbetering. Vanaf juni 2010 wordt het echt interessant. Op die datum wordt naast versie 3.6 ook versie 4.0 verwacht. De 3.* versies zullen blijven bestaan en verder ontwikkelen zolang er gebruikers zijn, terwijl daarnaast een geheel vernieuwde Eclipse 4.* serie ontstaat. De reden hiervoor is dat er dusdanig grote hertstructurering plaats dient te vinden dat backwards compatibiliteit niet gewaarborgd kan worden. In versie 4.0 zal Eclipse ook op het web gebruikt

kunnen worden, waarbij er één source de basis is voor de gehele user interface. Eclipse wordt in deze versie ook verder ontwikkeld als runtime platform voor desktop applicaties. Hiervoor wordt skinning, layout, styling etc. sterk uitgebreid zodat eenieder zijn eigen stijl applicatie kan bouwen. De datum voor Eclipse 4.0 wordt nog wel 'tentative' genoemd.

De workbench voor Eclipse 4.0 wordt momenteel geheel van scratch opnieuw ontworpen. Als basis wordt de workbench in het nieuwe ontwerp geheel met EMF gemodelleerd. Hiermee wordt het workbench model eenvoudiger en flexibeler. Bovendien kan de hele workbench met behulp van scripts aangepast worden aan de wensen van de gebruikers. Bovendien wordt de workbench op het scherm getoond door middel van rendering, wat betekent dat de look and feel drastisch gewijzigd kan worden door alleen de renderer te vervangen. Technisch mooi en het kan leiden tot prachtige visuele hoogstandjes.

Naast desktop applicaties wordt de Eclipse runtime ook aan de server kant steeds meer gebruikt. Hierbij ligt de focus op de OSGi componenten model. Eclipse zelf gebruikt het OSGi model het Eclipse plugin mechanisme te ondersteunen. Met OSGi is het mogelijk om runtime allerlei componenten te starten, stoppen of toe te voegen. OSGi wordt langzamerhand ook steeds meer gebruikt als componentenmodel op de server voor Java applicaties in het algemeen. Het Eclipse project waarin de OSGi implementatie ontwikkeld is heet Equinox.

Conclusie

Zoals duidelijk zal zijn is Eclipse een platform dat steeds breder wordt, niet alleen als platform voor ontwikkeltools, maar ook als runtime platform voor applicaties. Mocht je geïnteresseerd zijn in meer details van de genoemde onderwerpen of willen zien wat er allemaal nog meer besproken is, dan kan je op <http://www.eclipsecon.org/summiteurope2008/> het programma en de meeste presentaties bekijken. Het is natuurlijk niet hetzelfde als erbij zijn, maar zo kan je toch op de hoogte blijven. «

Rectificatie

In Java Magazine #3 (september) is bij het artikel over Groovy abusievelijk een verkeerde auteursnaam vermeld. "Even wennen, maar dan wordt het: 'Groovy, baby!'" is geschreven door Jan Ypma, Java Architect & MDA trekker bij Capgemini. Wij bieden hem hierbij onze verontschuldiging aan.