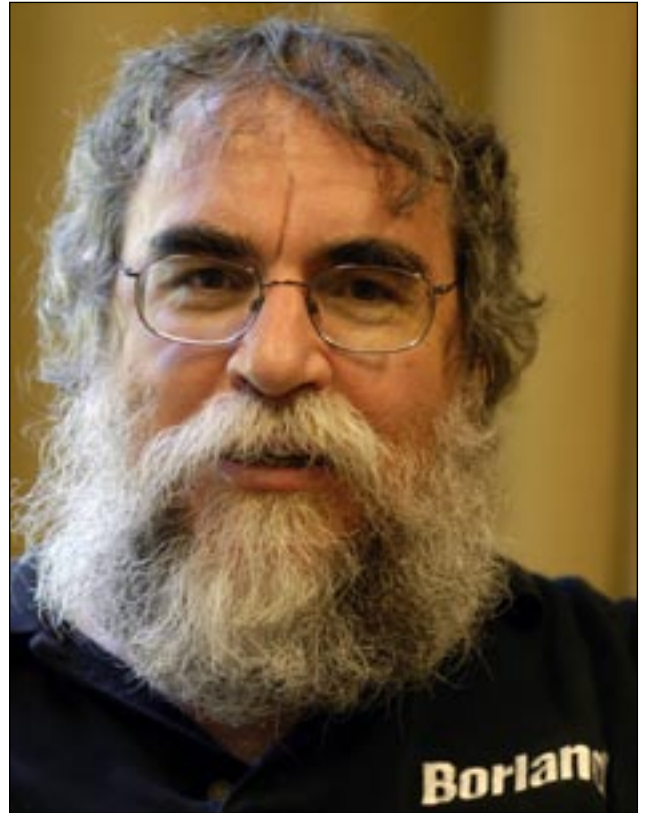


aandacht dus op de dingen die J2EE-programmeren en Java-programmeren weer simpel maken. Wij ondersteunen zeven applicatieservers. Zo kun je onafhankelijk van wat de back-end is een set van EJB's schrijven en zeggen: 'oké, deze keer wil ik het deployen naar BEA Weblogic, of naar Oracle, we zorgen daarvoor. We hebben de tooling om de juiste deployment descriptors onder water te gebruiken om de objecten te packagen enzovoorts'. Dus ja, er zullen wat andere concurrenten zijn maar we hebben al bewezen hoewel tooling voor Java-development over het hele development-proces we hebben en op dat gebied zullen we nog meer gaan doen.'

Waarom heeft Borland dit niet eerder gedaan?

Davidl: 'Eclipse was aan het begin nogal instabiel. JBuilder had een vliegende start, we hadden al refactoring en gedistribueerde development in versie 8 en Eclipse was in het begin niets bijzonders, pas met versie 3 met een aangepaste plug-in architectuur konden we de overstap naar Eclipse maken. We zijn grote voorstanders van het Java platform. Executive leden van de JCP werken aan een aantal JSR's, maar voor de tooling hebben we besloten om voor Eclipse te kiezen, in plaats van voor NetBeans, of ons eigen framework. De samenwerking met Sun gaat gewoon door, net zoals die met Oracle en anderen overigens. Onze klanten zijn hier echter heel blij mee en we gaan door met het geven van informatie en zullen ze testversies toespelen van Peloton - de codenaam van de nieuwe versie - en van de migratietools zijn en beginnen zich erop voor te bereiden. Sommige klanten zullen verder gaan met hun JBuilder-gebaseerde projecten en anderen zullen nieuwe Eclipse-gebaseerde projecten beginnen. De meeste ontwikke-



laars zullen het echter niet merken, want we zullen dezelfde rijke Java-omgeving bieden die we al jaren bieden. De roadmap laat het heel duidelijk zien: hier gaan we naartoe en dit is de weg. JBuilder 2006 dat dit jaar nog uitkomt is de laatste versie van JBuilder gebaseerd op het oude platform, begin 2006 komt dan JBuilder Peloton, de eerste Eclipse-gebaseerde versie en de basis voor alle volgende versies van JBuilder.'

OptimalJ naar Eclipse

Procesmodeluitbreiding en Pattern-ruilbeurs

OptimalJ 4.0 wordt uitgebreid met een procesmodel gebaseerd op het UML 2.0 Activity Diagram. Bovendien gaat Compuware over van NetBeans naar Eclipse. Edwin Schumacher Director Product Management voor Compuware OptimalJ licht een en ander toe.

Geheel in lijn met Borland en met de snel groeiende populariteit van het open source platform Eclipse, is de nieuwe Developer Edition van OptimalJ 4.0 nu 'built on Eclipse'. De andere edities zullen nog even moeten wachten. Alle toekomstige OptimalJ-edities zullen ook worden gebouwd op het Eclipse-platform. De plug-in voor IntelliJ blijft echter ook bestaan. Met de toevoeging van het pro-

cesmodel gebaseerd op het UML 2.0 Activity Diagram is het MDA model van OptimalJ redelijk compleet. Verder heeft Compuware het nieuwe online platform Pattern Plaza opgezet. Het platform voorziet ontwikkelaars van relevante en accurate informatie en fungeert als forum ter stimulatie van samenwerking en kennisdeling. Architecten kunnen zo OptimalJ-transformatiepatronen met elkaar delen en samenwerken aan implementatieverbeteringen.

Heeft de markt Compuware gedwongen voor het Eclipse-platform te kiezen?

Schumacher: 'Eclipse is populair, daar moeten we naar toe. Toen wij met OptimalJ begonnen bestond Eclipse nog niet, dus we zijn begonnen met NetBeans. Eclipse is

ook een tijd volledig gedomineerd geweest door IBM, om die reden was het toen ook niet handig om daarin te participeren, want we zijn concurrenten op een aantal gebieden. Maar nu Eclipse enerzijds onafhankelijk is en anderzijds zeg maar massaal geadopteerd wordt door developers, gaan we gewoon Eclipse doen. Dat is voor ons geen probleem want we deden NetBeans en onze hele architectuur was modulair opgebouwd. Voor NetBeans moeten we modules doen en voor Eclipse plugins, maar het concept daarachter is ongeveer hetzelfde.'

Zit er nog wel ruimte in die markt?

Schumacher: 'Wij zien de laatste tijd juist enorm veel interesse van system integrators. Want met name de Westerse system integrators concurreren keihard met India en China. Ze zitten met de vraag hoe ze hier even productief kunnen zijn als daar, én hoe ze de kwaliteit van die applicaties kunnen verhogen, en die consistentie te verhogen. Wij hebben discussie met Atos Origin, met Logica CMG, Capgemini en Ordina. Al die system integrators zijn *MDA centers of excellence* aan het opzetten. Daar bekijken ze hoe ze met MDA hun competitief voordeel kunnen vergroten. Aan de ene kant kunnen ze natuurlijk sneller en hogere kwaliteit software bouwen, maar ze kunnen met behulp van die pattern editing functionaliteit ook hun eigen patterns bouwen en dus hun eigen applicaties genereren. Dat is dus een stuk intellectueel property die ze in kunnen bouwen. Het andere probleem waar OptimalJ een oplossing voor biedt is, dat developers soms heel goed, en soms nogal junior zijn. Als je met OptimalJ werkt komt er altijd dezelfde gestructureerde applicatie uit. Daardoor kunnen ze gemakkelijker developers in een project inpassen dan wanneer ze alles handmatig moeten doen. Dan moet altijd het A-team sturen als iets complex is en het B-team als het misschien minder complex is.'

Kun je er uiteindelijk een ontwikkelstraat van maken?

Schumacher: 'Ja, dat is een van de dingen die we in OptimalJ nastreven. OptimalJ focust op modelleren en coderen, maar we hebben bijvoorbeeld ook de connectie gemaakt met requirements management en we gaan straks ook de connectie maken met functioneel testen. Compuware heeft ook het tool Vantage Analyser voor J2EE, en daarmee kun je al applicaties monitoren en we hebben DevPartner for Java, ook een profiling tool. We zijn nu DevPartner for Java en Vantage Analyser aan elkaar aan het knopen, zodat de feedback loop naar de developer heel snel wordt. Dan hebben we de kring weer gesloten en kunnen we die Application Life Cycle versnellen en dat is in feite een ontwikkelstraat.'

Worden frameworks als Hibernate en Spring door Compuware zelf geïmplementeerd of gaat dat gebeuren via Pattern Plaza?

Schumacher: 'Er zijn al implementaties van Hibernate en die komen op Pattern Plaza te staan. Wat we zien, is dat als je naar Hibernate kijkt, kun je binnen OptimalJ ongeveer vier verschillende architecturen onderscheiden om dat pattern te bouwen en dan is het maar de vraag welke je wilt. We maken een design waarin die vier verschillende architecturen te zien zijn, en dan kunnen developers kiezen en komt dat pattern beschikbaar. Een van die system integrators wilde graag client/server-applicaties bouwen en die heeft een Swing-pattern gemaakt. Prima, dat is wat we willen. Het zou ook kunnen dat er meerdere implementaties van hetzelfde pattern op komen te staan maar dat is niet erg. Laat de mensen maar zelf uitzoeken wat ze het beste vinden.'

Betekent de overgang naar Eclipse – open source software met het eraan gekoppelde idee dat het allemaal gratis moet zijn – niet ook dat er iets aan de prijzen gedaan moet worden?

Schumacher: 'Ik zie dat niet, want Eclipse is een IDE. De toegevoegde waarde van een IDE is er niet meer, iedereen heeft dat. Wat je ziet op open source, dat vooral commodity's open source worden. Je zou hetzelfde kunnen zeggen van IBM en BEA als je kijkt naar hun applicatieservers en JBoss. JBoss maakt dat open source, maar op het moment dat je applicatie wil schalen naar honderdduizenden users, dan lukt dat niet meer met JBoss. Voor kleine projecten, personal development, Eclipse en JBoss – fantastisch. Maar als je teamdevelopment moet doen met Eclipse en het project wordt groter en het wordt complexer, dan heb je met Eclipse bijvoorbeeld het geld niet in je tools zitten, maar moet je wel heel dure mensen inhuren. Je geeft dat geld toch uit. En dan zou het ook nog kunnen zijn dat je heel erg afhankelijk wordt van de skills van die specifieke mensen, dat zou nog een risico kunnen zijn ook. In OptimalJ zit al die kennis in het tool; die is van jezelf en die kun je zelfs uitbreiden. Wat we merken is dat de markt daar nog aan moet wennen, het is een soort van industriële revolutie voor softwareontwikkeling en die gewinning kost tijd. '

In welke zin, de system integrators hebben kennelijk al wel belangstelling.

Schumacher: 'Absoluut. Maar developers zeggen vaak: dat tool neemt mijn werk over. En in feite is dat niet zo, het tool maakt het werk alleen een stuk eenvoudiger. Het vervelende werk wordt afgenomen, maar de creativiteit die een developer ten toon moet spreiden om dingen op te lossen die niet te automatiseren zijn, blijft nodig. Dat zijn nou net de leuke dingen. In de developer-community zie je ook dat ze er steeds meer open voor gaan staan, omdat een heleboel bedrijven gaan outsourcen.'