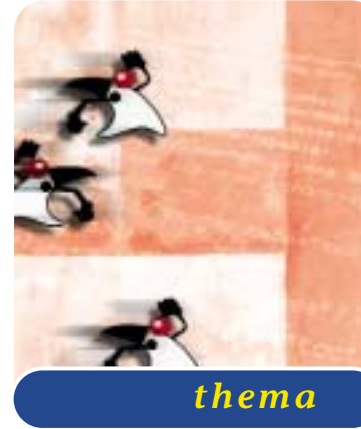


Java is voor veel mensen bijna synoniem met J2EE. Toch is Java ooit zonder begonnen, en sinds kort is er hernieuwde belangstelling voor Java op de desktop. Microweb is er midden in de dotcom-hype in geslaagd met een Java desktop applicatie succesvol te zijn, een applicatie die het makelaars mogelijk maakt hun huizen op Internet aan te bieden en foldermateriaal te produceren. Een gesprek met Klaasjan Tukker, manager software ontwikkeling Factotum Media B.V. Klaasjan was voorheen vice president software development bij Microweb.



# Java op de makelaars- desktop

## *Succes zonder J2EE*

De geschiedenis van Microweb Technology NV begon omstreeks 1994/95, toen nog als Microweb, en vrijwel gelijktijdig met Internet. Ook toen waren er al producten die poogden huizen, auto's en banen via Internet af te zetten. Voor die producten (die werkten met een Oracle Database en zelfs een Oracle application server) was echter nog nauwelijks vraag: Internet was nog te onbekend. Daarna is het bedrijf zich vooral gaan richten op *printing on demand* voor leermiddelen. Ter compensatie van het seizoenachtige karakter daarvan richtte men zich op een andere toepassing van printing on demand: het drukken van presentaties voor lokale reclamebureaus op 'tikbasis' (een gulden per print). In 1998 legde Microweb contact met de NVM met de vraag naar data om op basis daarvan webpresentaties van woningen te maken.

### **BROWSER WARS**

Tukker: 'Op dat moment was er een partij in Nederland, Factotum Media, die het alleenrecht op die data had. We konden die data wel krijgen, maar alleen onder strikte voorwaarden en we moesten per presentatie betalen. We besloten dat niet te doen, maar juist een eigen systeem te ontwikkelen waarbij de makelaar data invoert, centraal, op het Internet, maar niet - zoals in 1995 - geheel webgebaseerd. De applicatie draaide als Java-oplossing op de desktop, want we zaten midden in de tijd van de browser-wars waarin IE 4 iets anders deed dan Netscape 4, en je niet wist waarmee je klant werkte.'

*Daarmee was er wel een afhankelijkheid van de JVM.*

Tukker: 'Ja, dat wel, de makelaar moest wel de Java runtime geïnstalleerd hebben, die werd meestal door de accountmanager geïnstalleerd, en van daaruit werd gewoon opgestart. De eerste versies van Webhome hadden ook nog een eigen updatefaciliteit, waarmee de software vanaf Internet werd vervangen in latere versie hebben we dat vervangen door een Sun-standaard product, Java Webstart, dat die zaken voor jou doet. Door de installatie van de software op de desktop van de makelaar hebben wij onze handel in feite uitgebreid tot op zijn desktop. De makelaar voert rechtstreeks de data in en hij is daarvoor verantwoordelijk.'



De makelaar voert zelf de tekst voor advertenties en webpresentaties in

*Toen Klaasjan Tukker, samen met zijn collega Norbert Wassink in 1999 begon met de herbouw van WebHome, waren er een aantal argumenten die tegen het gebruik van een webapplicaties spraken.*

Tukker: 'Je weet niet welke browser je makelaar gebruikt, HTML is voor mij een presenteer-taal, geen userinterface-taal. Je kunt natuurlijk wel met Javascript en VB-script werken, maar op dat moment werkt het met de ene browser wel en met de andere niet. Het is nog steeds een probleem als iemand het uitschakelt, je gaat heel specifieke eisen stellen aan het gedrag van je browser.

Wij hebben gezegd: daarom bouwen wij een rich client waarmee je de volledige controle hebt over de user interface bij de klant, en waarmee je ook dingen kunt doen die je in HTML niet kunt doen zonder gebruik te maken van plug-ins. Wij lezen bijvoorbeeld afbeeldingen in. Die moeten in presentaties altijd in de



Tukker: 'HTML is voor mij een presenteer-taal, geen userinterface-taal'

3:2 verhouding zijn. Werk je met een digitale camera, staan die in een 4:3 verhouding. Door gebruik te maken van een Java-frontend kunnen wij automatisch de cut-out doen. Eigenlijk hebben wij in die jaren tegen de stroom ingezwommen. We werkten toen niet met J2EE, maar gebruikten servlets en http. Veel logica zit op de client en in feite is de server niets anders dan een doorgefluit of een barman die een bestelling aanneemt, doorgeeft, afmaakt en weer teruggeeft en kijkt of hij dat wel mag serveren aan die klant.'

## OO-CONCEPTEN

*De server doet in feite toch wel meer: data-integriteit en security worden bijvoorbeeld wel degelijk op de server gecontroleerd. Toch is dat ook iets wat pas later ingevoerd is.*

Tukker: 'De applicatie van 1995 werd in 1999 herschreven, in Java en volgens OO-concepten. Zo werd het mogelijk op basis van dezelfde software verschillende producten te schrijven, niet alleen voor de makerswereld producten beschikbaar, maar ook voor de autowereld, Webcar, en voor de catalogi-wereld als Webicat.'

*Anders dan 1995 was er nu wel vraag, maar soms was die zo groot dat die het softwareontwikkeling-proces in de weg zat.*

Tukker: 'Toen wij in 1999 begonnen had ik net mijn hogere informaticaopleiding afgerond, dus wij werden zo in de echte wereld gedropt waarbij je de noodzaak had om een echte oplossing te leveren. Het was tijdens de dotcom hype, maar wanneer je in een klein bedrijf werkt waar alles afhangt van de inkomsten die je op dat moment rechtstreeks genereert, dan kan een maand uitstel afstel betekenen.'

*Een deel van de applicatie was zelfs nog niet af, toen een ander deel al verkocht was.*

Tukker: 'We hebben eerst de client-applicatie geschreven, dus de klanten de mogelijkheid gegeven de data bij ons in te voeren, terwijl wij de hele achterkant nog niet operationeel hadden. Dus in de eerste weken met de eerste klanten hebben de grafisch designers gewoon op grafische werkstations met de hand de presentaties gemaakt.

Daardoor was de druk op de ontwikkeling heel hoog om ervoor te zorgen dat het geautomatiseerd werd. Zo hebben we in de loop der jaren ook de achterkant volledig in Java geschreven, waarmee volledig automatisch presentaties worden opgemaakt, aan de hand van standaard gedefinieerde templates voorbereid, en klaargezet voor digitale drukpersen.'

*Lees verder op pagina 49.*

Vervolg van pagina 8.

## DATASTRUCTUREN

*Tijd om software gestructureerd via korte iteraties met testcycli te ontwikkelen was er eigenlijk niet, volgens Tukker:*

Tukker: 'Er werd code geschreven, die code werd gerund, alleen om te kijken hij deed wat hij moest doen. Aan design patterns werd geen aandacht besteed, aan de structuur alleen globaal. Er was een globaal idee, er was een server, we moesten op dezelfde manier communiceren als browsers, ook mensen die achter firewalls of achter proxy's zitten, moeten ermee kunnen werken. We maakten gebruik van http communicatie. Er zat nog een andere gedachte achter, namelijk dat data eenvoudig uitbreidbaar moesten zijn zonder dat je nieuwe clients nodig had, oftewel er moest een soort modulair datamodel achter zitten. Er zaten dus wel een paar conceptuele ideeën achter, maar als je kijkt naar de manier waarop toentertijd de front-ends in elkaar zaten: de swing-based front-ends werden helemaal met de hand in elkaar geklopt, ontzettend veel werkelementen werden gewoon fixed gepositioneerd op een bepaalde locatie, en er zat een heel grote verwevenheid tussen de datastructuur en de front-end. In de begindagen was er één drang en dat was zorgen dat het product op de markt komt. Zo hebben we ruim drie jaar doorontwikkeld.'

*Wel werden veel van de zaken die achterwege gebleven waren, achteraf alsnog gedaan.*

Tukker: 'De Java documentatie werd aangepast, en er werden unit tests en UML-diagrammen van datastructuren geschreven. De belangrijkste stap was echter het aanpassen van de swingapplicaties. Die zien er niet altijd even mooi uit, daarom hebben we de user interface losgekoppeld. Thomas Zander, een van mijn medeontwikkelaars is zeer actief binnen KDE en was goed bekend met QT Designer van Trolltech, waarmee je user interfaces heel mooi platformafhankelijk kunt ontwikkelen. De user interfaces worden weggeschreven in een XML-bestand. De user interface compiler leest het XML-bestand in het geheugen in als document object model en vertaalt bepaalde widgets naar Javacode in Swing. Een QPushButton in QTDesigner wordt bijvoorbeeld vertaald naar een JButton in Swing-code.'

## OPEN SOURCE

*Bij het ontwikkelen van het user interface compiler project is er vrij intensief met Swing gewerkt. Swing bleek niet geheel bug-vrij.*

Tukker: 'Er zitten nog steeds problemen in de Swingomgeving. Op het moment dat je dat gaat abstraheren in de vorm van een XML-file die je gaat vertalen naar Java-code, kun je zorgen dat wanneer je met een



Tukker: 'Ontwikkelen doen we op het Linux-platform'

bepaalde button een probleem hebt, je al de workaround code meegenereert totdat Sun er een oplossing voor heeft. Voor een aantal veel voorkomende swingbugs waar we normaal meerdere instanties omheen moesten programmeren, wordt nu die code eenmalig gegenereerd. Dit was eigenlijk een project wat we hele-

**Dit project hebben we helemaal losgeweekt en als open source project gedefinieerd**

maal los hebben geweekt en als open source project hebben gedefinieerd, op verzoek van de ontwikkelaar in kwestie. Ik vond wel dat er een balans moest zijn tussen het aantal uren in werktijd en in privé-tijd die eraan besteed zouden worden, maar we hebben daarin vrij



gemakkelijk een evenwicht met elkaar kunnen vinden. De hele built-omgeving is in de loop der tijd ook aangepast. Toen ik begon te programmeren was dat in C++ en Java, dus dan gebruik je make met Makefiles. Niet altijd de meest plezierige omgeving en vooral wanneer je veel Java-files wilt compileren kost dat veel tijd. Langzamerhand ga je de overstap maken naar ANT, wat eigenlijk de standaard is binnen de Java-wereld. We hebben ook voor de UI-compiler een ANT-task geschreven die het compileren van de UI-files met Java-code gewoon als proces in zichzelf doet. Dus er is geen Java-code in onze CVS-Repository opgeslagen van user interfaces, alleen xml.'

*Behalve dat er binnen Microweb open source-projecten uitgevoerd worden, wordt er ook veel open source-software gebruikt binnen het bedrijf zelf.*

Tukker: 'Ontwikkelen doen we op het Linux-platform en de meest gebruikte editor in ons bedrijf is VI. Ik heb collega's die helemaal niets moeten weten van IDE's met code-generatie en dingen die je zelf niet in de hand hebt. De combinatie van VI met CVS en ctags waarmee je heel snel door code heen kunt browsen, wordt het meest gebruikt. Het meest arbeidsintensieve proces is echter het refactoren van code. Dus daar wordt tegenwoordig Eclipse voor gebruikt, wat heel sterk is in het refactoren van code en het groeperen en degroeperen, het renamen van methods. Eigenlijk ben ik de enige ontwikkelaar die nog op het Windows-platform draait - ik heb ook nog wat meer een consultant rol - waarop ik de Eclipse IDE volledig gebruik. Daar heeft het standaardiseren van je built omgeving naar ANT, het gebruik van CVS, het niet vastgepind zitten in een bepaalde user-interface design, het gebruik van XML-files die je in- en uit kunt checken met een automatische compile-processor erbij, voordeel.'



Ook webpresentaties kunnen op deze wijze automatisch gegenereerd worden.



Foto's kunnen door de applicatie op maat gemaakt worden

## PUBLICATIEPROCES

*Het bedrijf dat in het begin van de geschiedenis van Microweb indirect verantwoordelijk was voor het ontstaan van de huidige vorm van de applicatie, blijkt ook aan het eind van grote invloed op het bestaan van Microweb.*

**'Het meest arbeidsintensieve proces is het refactoren van de code'**

Tukker: 'Wij hebben Factotum enige tijd geleden benaderd met de vraag een OEM-versie van onze software te gaan gebruiken. Vervolgens kwam Factotum met het idee de twee bedrijven te combineren. Factotum was op dat moment goed voor ruim tweeduizend makelaars in de NVM-markt alleen al. Zij hadden een enorm klantenbestand en wij hadden de workflow en de oplossingen ervoor. Dat is met ingang van januari 2003 effectief geworden en sinds augustus is de gehele technologiepoot op verzoek van Factotum media daar ondergebracht en zijn zij de partij die het distribueert en in de onroerend goed-markt neerzet.'

Zo zijn het software pakket van Microweb en het marketingkennis- en marktaandeel van Factotum uiteindelijk gecombineerd. Microweb blijft echter voortbestaan en richt zich nu op het ontwikkelen van software om het publicatieproces van educatieve materialen te vereenvoudigen.

*Tekst en fotografie: Dré de Man*