

**Sun, Microsoft en Adobe zijn op dit moment in een hevig gevecht gewikkeld wie het grootste stuk van de RIA-markt gaat veroveren. Het programma van de JFall maakte dit ook duidelijk: veel aandacht voor zowel de RIA-oplossingen van Adobe als voor die van Sun. Adobe heeft in ieder geval een voorsprong in de tijd op haar twee concurrenten. James Ward, evangelist bij Adobe, is van mening dat die voorsprong zich nog verder uitstrekt.**

# De moeder van alle RIA's

## Interview met James Ward, Flex-Evangelist bij Adobe

*Macromedia heeft het concept van RIA's al in 2002, geïntroduceerd. Het heeft echter nog vijf jaar geduurd voordat het echt populair werd. Al vóór 2002 gaf Flash gestalte aan die visie, in 2004 werd Flex geïntroduceerd. Zou je Flash nu echt een voorloper van Flex kunnen noemen, de moeder van alle RIA's?*

Dura: "Jazeker, mensen gebruikten Flash om er RIA's mee te bouwen. Maar het was moeilijk om applicaties te bouwen met het timeline paradigma. Voor designers is een timeline geweldig, maar niet voor ontwikkelaars die code willen schrijven en OO-concepten willen hebben. Flex wilde diezelfde ervaringen brengen maar met een programmeertaal in plaats van een timeline. Flex is er nu meer dan drie jaar en er zijn tienduizenden zo niet honderduizenden apps gebouwd met Flex nu. Het is geweldig dat de industrie heeft herkend dat dat de weg is die we ingeslagen zijn en dat Microsoft en Sun daarin volgen en met hun eigen platformen komen maar ze lopen een beetje achter ze hebben niet de penetratie van de runtime die wij hebben.

*De distributie van de runtime lijkt me voor Sun en Microsoft niet zo'n probleem. Maar om het geadopteerd te krijgen is een ander verhaal dan het te bouwen Het is*

een kip- en ei-probleem. Je krijgt mensen niet zo gek om applicaties te bouwen voor een platform dat niemand heeft en niemand gaat een platform installeren als er geen applicaties voor zijn.

*Maar ze kunnen het allebei met hun updates verspreiden.*

Ja, maar Microsoft heeft ervoor gekozen dat niet te doen en bij Sun valt dat ook nog af te wachten. We prijzen ons gelukkig met de Flash player die een zeer grote verspreiding kent. Dat is onze VM voor Flex-applicaties. Op 93% van de met internet verbonden pc's is Flashplayer 9.0 geïnstalleerd, de laatste versie dus. We bereiken meestal 80% adoptie met een nieuwe versie binnen de eerste acht maanden, en 90% in de eerste twaalf maanden. Sun geeft geen officiële cijfers, maar ik heb begrepen dat 65% een versie van de Sun JVM heeft. Het is moeilijk om een hoge adoptiegraad te bereiken, vooral wanneer de download vrij groot is. Wij hebben een downloadgrootte van 1.2 MB voor de Flashplayer en de virtual machine, dus zelfs al heb je een inbelverbinding, dan is het nog eenvoudig. Er zijn ook veel mensen die Flex omarmen als UI-technologie op basis van Java backend. Het wordt heel goed opgepakt binnen de Java-wereld.

*Hoe zit het met de schaalbaarheid en de performance? Ik heb hier en daar geluiden opgevangen dat dat kan tegenvallen.*

Er zijn zeker scenario's waarvoor de VM die we in de Flash player hebben niet de juiste oplossing is. Sun's JVM is er nu een jaar of zeven, dat is een zeer rijpe VM. Onze VM is pas twee jaar oud, en is heel erg klein, 200 kb voor de basis VM. De performance is om eerlijk te zijn niet zo goed als die van de JVM maar het is ook afhankelijk van wat je doet. Als je zeer grote datasets sorteert is Java duidelijk sneller, maar als rendering je bottleneck vormt dan zijn er veel gevallen waarin Flash een heel stuk sneller is dan de Java. Er is daarvoor een aantal redenen. Eén daarvan is dat we nu in Flash multi threaded rendering doen. Als je een multi core machine hebt, dan kun je rendering via multiple threads laten verlopen en dat kan het flink versnellen. Het ligt er dus aan wat je bouwt; er zijn ook beperkingen. Er zijn ook dingen binnen Java zelf die langzamer zijn, bijvoorbeeld wanneer je met devices praat hebben we daar nog geen aparte libraries voor. Het is allemaal in ontwikkeling. Je kunt nu nog niet echt met apparaten praten, dat kan alleen via een socket die op zijn beurt met een apparaat praat.

*Terug naar de VM: Sun en Microsoft zijn allebei hard bezig om hun VM's kleiner te maken, Microsoft zit nu bij 2 MB voor Silverlight en Sun is ook bezig de VM klei-*

## Er zijn zeker scenario's waarvoor de VM die we in de **Flash player** hebben niet de juiste oplossing is

*ner te maken. Jullie brengen echter nieuwe functionaliteit en nieuwe libraries binnen de VM, dus mag ik aannemen dat die groottes elkaar gaan benaderen?*

Wij zijn heel voorzichtig met het groter maken van onze VM. Het passeren van de 1MB-grens was een heel grote stap voor ons. We hebben ook tests gedaan waaruit we geconcludeerd hebben dat dat wel een psychologische grens is, het downloadpercentage neemt daarboven echt af. We zijn dus heel voorzichtig. We hebben wel onlangs nieuwe edities uitgebracht, eentje met de h264-decoder voor video, waardoor HD-content binnen Flash afgespeeld kan worden, dat heeft de grootte wel iets doen toenemen.

*Een beetje verwarrend lijkt me het feit dat voor de Adobe VM geprogrammeerd moet worden in alweer een nieuwe taal.*

Het goede is dat we binnen Flex gekozen

hebben voor technologieën die bekend zijn. De twee programmeertalen binnen Flex zijn XML en een procedurele taal die in essentie Javascript is.

*Maar niet echt.*

Het is een implementatie van de nieuwere versie van Javascript, eentje die nog niet in de browsers zit, maar nu al wel in de specificatiecommissie. Het is ECMAScript 284 rev 4, de specificatie voor Javascript 2.0. Toen we Flash 9.0 bouwden hebben we op die taal gestandaardiseerd en we hebben ook de eerste versie van Tamarin ontworpen als een implementatie van een VM voor die taal. Daarna zeiden we: waarom zouden we iedereen ter wereld dwingen om zijn eigen VM te bouwen voor Javascript 2.0? We hebben dus besloten die VM te open sourcen met Mozilla zodat Mozilla het in de browser zou kunnen stoppen. Zo kan iedereen die VM gebruiken om Javascript 2 code uit te voeren. Tamarin en de programmeermodule zijn belangrijk om de ontwikkelaars in staat te stellen de technologie te leren. Javascript 2 lijkt nu heel erg op Java. Het is niet meer zo vreemd als Javascript was, we hebben een echt class base systeem nu, met overerving en dat soort dingen. De taal is nu heel gemakkelijk te leren voor Java-ontwikkelaars, en met XML en HTML is het heel erg bekend. Eén ding wat ik wil benadrukken is dat de vraag naar Flex-ontwikkelaars ver boven het aanbod ligt. We zijn nu op het punt in de adoptiecurve waar Java in de jaren 90 was. Wanneer mensen echter vragen waar ze Flex-ontwikkelaars kunnen vinden, dan vertel ik ze: neem een goede Java-programmeur, train hem twee weken op Flex en het zal een expert op het gebied van Flex zijn.

*Hoe zit het met de tools?*

Ja, ik vergat dat te vertellen, Flex Builder is er nu als een Eclipse plug in, wat voor



*'Als rendering de bottleneck vormt dan zijn er veel gevallen waarin Flash een heel stuk sneller is dan Java'*

veel Java ontwikkelaars heel vertrouwd overkomt. Je kunt ook de Flex SDK downloaden. En de command line compiler of ANT gebruiken om het te compilen. Veel verschillende opties, maar allemaal bekend voor Java ontwikkelaars. Flex is zelfs geschreven in Java.

*Hoe zit het met de testmogelijkheden?*

Er bestaan verschillende unit test frameworks. Flex unit is er een, er is nog een andere en wij werken aan eentje dat deel uit gaat maken van de open source Flex SDK. Er zijn ook geautomatiseerde tets frameworks zoals van Mercury, diverse zelfs.

*Als je tegen een Java-ontwikkelaar zou moeten zeggen waarom hij Flex zou moeten gebruiken, wat zou je dan zeggen?*

Ik zou hem aanmoedigen Flex Builder te

downloaden, er mee spelen, er zijn heel veel tutorials op Adobe.com.

*Is er ook open source materiaal voor Flex?*

Ja, er zijn heel veel open source projecten voor Flex, verschillende open source libraries. Daarnaast is er ook een open source collaboration server.


*Wat zijn nu goede voorbeelden van applicaties die geschikt zijn voor het bouwen met Flex?*

De meeste business apps die met data entry en data visualisatie van doen hebben, zijn perfecte kandidaten voor Flex-applicaties. Je creëert een betere gebruikerservaring die hetzelfde werkt binnen verschillende browsers en besturingssystemen. Meer consumentgeoriënteerde apps als Picknick.exe en Yahoo maps, Yahoo Web Messenger, zijn met Flex gebouwd,

de nieuwe on line tekstverwerker Buzzword (opgekocht door Adobe, DdM) is ook met Flex gebouwd. Overal waar je je wil onderscheiden door een gebruikerservaring is Flex op zijn plaats. Multimedia, recording al dat soort dingen. Data-visualisatie op BI-gebied is ook een belangrijke toepassing, vrijwel iedere BI-vendor die ik ken gebruikt Flex.

*Maar hoe zit het dan met de eerder genoemde performance-problemen?*


Er zijn verschillende aspecten. Eentje is dat je een binary serialisatie protocol tussen client en server kunt gebruiken om je data uit te wisselen waarbij je extreem snel data kunt parsen. Een ander aspect is Tamarin. Het heeft een JIT enabled VM, waardoor applicaties heel snel zijn aan de client kant, met native vector drawing.



**Compu'Train verzorgt ruim 500 ICT-cursussen op 9 locaties in Nederland. Jaarlijks begroeten wij ruim 20.000 cursisten.**

**Cursussen**

- JavaScript Basis
- JavaScript: Browser-servercommunicatie met AJAX
- Java SE Fundamentals
- Java SE Object Programming
- Java SE Workshop



**Compu'Train**

[www.computrain.nl](http://www.computrain.nl) • 0800-2667887

Utrecht • Amsterdam • Arnhem • Den Haag • Drachten • Eindhoven • Groningen • Sittard • Venlo