

Greg Murray houdt zich al tien jaar bezig met JavaScript. Hij zit namens Sun in de OpenAJAX Alliance en is oprichter en principal architect van Project jMaki. Murray heeft recent bijgedragen aan ontwerp en ontwikkeling van de Ajax-gebaseerde Java Pet Store 2.0 Demo. Java Magazine sprak met hem.

'JavaScript is onvermijdelijk'

Greg Murray over jMaki

Murray is naar eigen zeggen van huis uit een webapplicatie-ontwikkelaar. Hij bouwde elf jaar geleden zijn eerste website en begon toen al te experimenteren met animatie. Daarmee maakte hij het zichzelf niet gemakkelijk, want zelfs animated GIF's bestonden nog niet. Misschien komt het wel daardoor dat hij zich nu vooral bezig houdt met het vereenvoudigen van het schrijven van webapplicaties.

Murray: 'Er zijn heel veel verschillende toolkits voor de interactie met JavaScript, Dojo's, Yahoo UI widget, Spry van Adobe. Het stopt niet bij Ajax, we maken deel uit van de open Ajax organisatie. jMaki is onder meer een laag waar je al die toolkits samen kan gebruiken. Je gebruikt Java in de back-end, je schrijft een JSP- of een JSF-tag en aan het eind heb je al die JavaScripting op je pagina. Van het JavaScripting-standpunt hoef je de meeste complexiteit daarvan niet te zien, maar als je wilt, kun je erin duiken. Project jMaki is een manier om al die toolkits te gebruiken op een pagina. Je kunt er echter ook Java back-end of JSF back-end applicaties mee construeren die heel veel gebruik maken van JavaScript aan de front-end. Het idee is dat je nog steeds de voordelen hebt van je modeldata, je database, al je servlet api's, maar dat de meeste client interaction wordt gedaan in JavaScript. Ik wilde iets dat snel was, dynamisch, gemakkelijk, iets dat ik kan tunen terwijl het draait. Java-gebaseerde compo-

nenten die aan de serverkant draaien, bijvoorbeeld JSF-componenten met veel JavaScript erin, werden moeilijk te onderhouden. Als je JavaScript doet is dat erg iteratief, dat is trouwens de sleutel tot alle scripting development. Het platform was er eigenlijk niet op ontworpen om dat te doen. JSP laat je dat tot op zekere hoogte doen, maar als je begint scripts en componentmodellen begint te mixen wordt het moeilijk. Je hebt dan iets nodig dat alles samenvoegt en dat doet jMaki voor je. Het neemt bijvoorbeeld een lijst van dingen die je in een menu wilt stoppen, en brengt die op een zinvolle manier van een punt naar een ander punt. jMaki doet dat en ik heb daarbij geprobeerd zoveel mogelijk van het back-endmodel te abstraheren. Als je JDBC of EJB of servlets op de back-end wil gebruiken, of JSF managed beans, dan kan dat allemaal. Wanneer je slechts een kleine widget op je pagina wil hebben, alleen een combobox, gooi het op je pagina, ik schrijf geen compleet programmeermodel voor. Ik geloof echter wel in webstandaarden, CSS, JavaScript en HTML zouden niet door elkaar gehusseld moeten worden, en dat is precies wat in het verleden veel gebeurd is. Maar wanneer je in een wereld werkt met designers en anderen die webapps bouwen, die kunnen niet door je Javadoc of zelfs je JavaScript heen lopen, tenzij je het goed gescheiden houdt. Ik probeer dat model te promoten. We bieden daarom ook CSS en HTML *templates* aan.'

Geen Java meer nodig

De web 2.0 wereld is erg in beweging en Murray vindt dat fascinerend.

Murray: 'Er gebeuren heel veel interessante dingen. Als ik een maand op vakantie zou gaan dan zou ik in een andere wereld terugkomen. Dat komt door de iteratieve Web 2.0 stijl. Bij Google, Yahoo, en zelfs bij iTunes zie je dat hun services organisch, levend zijn, altijd veranderend maar altijd deployed. Applicaties blijven ook heel lang in bèta, als teken daarvan. Ik help ze dus met hun dingen in elkaar te zetten, maar ik werk ook veel samen met de Google services. Ik weet dat het in Nederland andere associaties oproept, maar er bestaat een foto sharing site die Flickr heet. Yahoo heeft de site gekocht. Je kunt met Flickr interacteren via RSS, je kunt dan content naar binnen halen en daar een soort *mash up* van maken, door items van Flickr, Yahoo en Google Maps naar gelieven samen te stellen. Veel mensen proberen te definiëren wat Web 2.0 is. Voor mij is het meer dynamisch gedrag, interacting, content van veel verschillende kanten inbrengen. Ik propageer jMaki als een goede oplossing daarvoor.'

Je zou je kunnen afvragen in hoeverre Java nog nodig is, nu JavaScript tot zoveel in staat is.

Murray: 'Veel mensen zeiden toen ze Ajax zagen: ik heb JavaScript op de frontend, ik heb geen Java meer nodig. Ik heb dat vaak gehoord. Maar we kunnen in feite de JavaScript omgeving versnellen of verbeteren, gebruik makende van JSP dat een hoop werk doet dat normaliter door JavaScript gedaan zou moeten worden. Ik kan Dojo – een toolkit in JavaScript - sneller laten lopen dan zonder Java omdat ik weet wat het doet en ik hints kan invoeren wanneer het laadt. Wij doen dat ook en maken deel uit van de Dojo foundation. We hebben technici die daaraan bijdragen, ik heb dat aanvankelijk zelf gedaan.'

De eeuwige bèta

Murray: 'In bèta zijn betekent in Web 2.0 ook iets heel anders, het zegt eigenlijk: we willen in staat zijn die te updaten. Dat is een hele nieuwe manier van denken. De Java development cyclus waarbij je een project na een half jaar oplevert kan niet meer, als je dat in de Web 2.0-wereld zou doen ben je verouderd. Ik was twee jaar geleden op JavaOne en toen ik JavaScript noemde werd de zaal meteen stil. De pers

'Als ik een maand op vakantie zou gaan, dan zou ik in een andere wereld terugkomen'

zegt dat Sun nu op de JavaScript-trein springt maar dat is niet waar, we doen dit al jaren. In 2003 begon Sun al te kijken naar het integreren van scripting talen in de JVM, ik denk JSR 223. Dat is voordat Ajax zelfs maar een gedachte was. Je ziet het ook aan JRuby, en ook aan het project Phobos. Dat maakt het mogelijk om op het JEE server platform te scripten op de backend. Wanneer ik terugdenk is het grappig want wat we nu doen is wat we zeven jaar geleden probeerden te doen. Het enige verschil is dat we nu aan de back-end een veel meer volwassen en geaccepteerd platform hebben. We hebben het ook gemakkelijker met Phobos door JPA. Dat is de volgende generatie van EJB waarmee je in wezen pojo's kunt gebruiken. JPA kun je nu direct vanuit JavaScript aanroepen. We vertellen je niet dat je alles opnieuw moet gaan bouwen, maar als je op een snelle manier applicaties wilt bouwen en scripten met JavaScript aan de front-end en het backend, en je wilt nog steeds gebruik maken van wat je hebt in Java, dan kan dat. Het is experimenteel en of het al dan niet een *full blown* product wordt is nog steeds nader te bepalen, maar het heeft de instelling van de applicatieserver-groep helemaal veranderd. Ik maak deel uit van die groep die J2EE maakt en wij hebben in feite gezegd dat we die nieuwe agile ontwikkeling waarin je dingen iteratief doet, beschikbaar moeten maken voor iedereen. We zijn zelfs opnieuw aan het nadenken over de opzet van onze servers. Dat betekent niet dat er geen J2EE meer zal zijn, maar dat het krachtiger zal zijn voor ontwikkelaars. Ik ben daar gelukkig mee, want het platform komt eindelijk op een punt waar ik als ontwikkelaar snel de dingen kan doen die ik wil. Toen ik begon met de Ajax-ontwikkelingen moest ik een applicatie deployen en werken op de deployde applicatie. Nu hoef ik dat niet meer te doen want we hebben tooling en dat soort zaken.'

Eén pagina

Murray: 'De volgende fase is dat we NetBeans plug-ins maken, om dezelfde

ontwikkelstijl mogelijk te maken. NetBeans had al de mogelijkheid om iteratief aan een webapplicatie te werken, maar nu hebben we Scripting eraan toegevoegd en het deployen van allerlei zaken in een webproject. Wanneer je nu uiteindelijk deployt krijg je een standaard warfile maar daar zit dan heel veel script in. En we hebben een manier om visueel drag and drop met jMaki widgets mogelijk te maken. We hebben dus ons eigen visual web pack, de oude Creator, dat laat je op een grafische manier webpagina's bouwen. Het is wel voor een andere doelgroep bestemd. Ik ben meer voor de codeview. Dat is nu ook al mogelijk, maar het is meer werk. Je moet aan de JSP-view werken en de meeste Creator-ontwikkelaars die op een grafische manier dingen willen bouwen zijn daar niet aan gewend.'

Kun je nu echt een goed werkende applicatie maken die geheel op JavaScript gebaseerd is?

Murray: 'Ja, en niet alleen dat, maar we kunnen het ook snel. Maar we hebben ook een nieuw programmeermodel nodig. Dat programmeermodel beginnen we nog maar pas te begrijpen, nu we Ajax sinds twee jaar gebruiken. Hoe bouw je een portal-achtige applicatie? Je zou portlets kunnen gebruiken of niet, jMaki gebruikt ze. Ik zal daar binnen Sun verder over praten, want ik ben meer voor standaard JSP-tags of voor JSF-componenten. Hoe verbind je alles met elkaar, hoe krijg je verschillende toolkits zover dat ze met elkaar interacteren, hoeveel modeldata houd je op de client versus de server, hoe page je door grote datasets op de client? Hier hebben we aan gewerkt, en ik ben daar aan begonnen met de Pet Store applicatie, van de eerste productie van Ajax. Dat begon als een één-pagina applicatie, je verliet nooit de beginpagina. Ik verving de verschillende regions terwijl ik verder door de cyclus kwam maar dat werd een beetje verwarrend. Dus maakten we meer pagina's; en sommige daarvan dynamischer, de cataloguspagina bij

voorbeeld laat het toe door de hele catalogus te scrollen zonder de pagina te verlaten.'

Murray is in het verleden wel eens fout geciteerd met betrekking tot webservices. Murray: 'Mensen zoals ik die bij het Web 2.0-kamp horen houden wel van webservices, maar van vrij rustige webservices, bij Web 2.0 gaat het om simpele dingen. WSDL is oké als je back-end applicaties doet, maar bij een front-end JavaScript heb je liever iets lichters. Het is dan meer: je geeft een URL en je krijgt iets terug. Een voorbeeld is Yahoo, waar je kunt zoeken via een webservice. Je kunt met Flickr praten, ze hebben een manier om naar foto's te zoeken via een webservice, maar ook door een RSS-feed, je geeft parameters en je krijgt een RSS-feed terug. Je kunt het uit JavaScript nemen en ermee werken. Dat is hoe ik in jMaki werk met beelden, ik kan ze keywords geven die tags zijn en een view genereren.'

Niet slecht

Murray: 'Webservices zijn niet slecht, ik ben op dat punt al eens verkeerd geciteerd, we houden er erg veel van (*lacht, red.*). Ik denk dat ze heel erg handig zijn waarvoor ze bedoeld zijn, solotype business-integratie, maar zo gauw je terugkomt en met een client wil communiceren wil je rustiger dingen. jMaki maakt deel uit van een proxy om webservices aan te roepen en die heeft de mogelijkheid om webservice data te converteren in iets wat

meer geschikt is voor JavaScript, dus we converteren het in Jason, JavaScript Object Notation, dat is native JavaScript, een alternatief voor XML. Het mist een paar voordelen van XML maar veel JavaScript-developers houden ervan omdat het in die omgeving gemakkelijker is om ermee te werken. Het voorkeursformaat is Jason maar we kunnen met XML omgaan als dat moet. Het ligt ook aan de toolkit, Dojo bijvoorbeeld gebruikt altijd Jason, dat is hun mentaliteit. Adobe heeft het Spry-framework en zij prefereren XML. Niets verkeerd mee, wij schrijven niets voor maar bevelen wel iets aan.'

Roept het zo krachtig maken van een scripttaal niet het gevaar op dat mensen die het niet zouden moeten doen te veel dingen gaan doen, ook met de verkeerde middelen?

Murray: 'Niet als je terug gaat naar de basisprincipes. Daarom preek ik steeds over de scheiding van content, style en behaviour. Over tien jaar, als er dan nog steeds een web is, zullen die nog steeds belangrijk zijn. Denk aan het design dat zijn tijd vooruit was, ga niet te ver, hussel de dingen niet door elkaar. Sommige mensen willen Web 2.0 of Ajax doen en gaan te ver, net zoals ik toen ik Pet Store als een één-pagina applicatie schreef. Maar dat is niet de mode die wij promoten nadat we ervan geleerd hebben. Er zijn plaatsen waar het wel heel prettig is, zoals bij GoogleMaps. Maar Google doet het niet overal, de Gmail-interface is anders bijvoorbeeld. Als je niet voorzichtig

bent krijg je ook accessibility issues en internationalisatie-issues. Internationalisatie is geen probleem, Dojo probeert accessibility in te bouwen al hun componenten, er nog beperkingen, ook aan de security-kant, waar gaten in je JavaScript ertoe zouden kunnen voeren dat mensen dingen doen met je services niet verwacht. Ik heb al een aantal van die dingen in mijn blog behandeld, maar ik denk niet dat het zo'n punt wordt.

Sommige mensen zeggen dat Web 2.0 een achilleshiel is. Dat geloof ik niet, maar er zijn wel *best practices* nodig. In feite is iedere website kwetsbaar, het maakt niet uit wat je op de front-end hebt. Je moet validatie aan de back-end kant doen, daar komt het op neer. Wat we nu ook zien is Java-ontwikkelaars, die dingen beginnen te schrijven in JavaScript. Maar scripten in JavaScript is een nieuw paradigma, het is geen Java. Ze zien er hetzelfde uit, ze delen vier letters, maar ze zijn heel anders. Maar het goede nieuws is: er is altijd al een JavaScript community geweest en die zijn al die problemen al tegengekomen de laatste vijf, zes jaar en die zijn in staat geweest ze bijna allemaal op te lossen. Ik hoop die informatie samen te brengen en op JavaOne tegen de developers te zeggen: dit zijn de dingen waar je op moet letten.'

Terugslag

Murray: 'We gaan ontwikkelaars niet dwingen, maar ik denk dat de tijden echt veranderd zijn. Als je dynamische webapplicaties wilt bouwen zul je hieraan moeten wennen. Er zal een terugslag komen, omdat mensen zich zullen realiseren dat ze een *toolkit* nodig hebben. De verschillen tussen de browsers maken dat gewoon nodig. Er zijn componentmodellen, JSF, GWT, er zijn manieren om veel JavaScript te verbeteren, maar uiteindelijk moet je JavaScript kennen als je die componenten schrijft. Ik denk dat het onvermijdelijk is voor een groot deel van de ontwikkelaarsgemeenschap, het is alleen een kwestie van acceptatie. We hebben alternatieven, zoals jMaki, maar het is ook niet zo'n grote verandering. Toen XML opkwam zeiden mensen niet: "Oh, ik wil geen XML leren want het is geen Java." Ze deden het omdat het een manier was om de ervaring een niveau verder te brengen. Zo moet je ertegenaan kijken. Je moet over JavaScript en Java denken dat ze samen werken om grotere problemen op te lossen.'



'We gaan ontwikkelaars niet dwingen, maar ik denk dat de tijden echt veranderd zijn.'