

# Tot ziens CLR, welkom DLR?



't Is opnieuw juni en de zomervakantie staat voor de deur. Ontwikkelaars op het Microsoft-platform hebben de afgelopen maanden met heel wat nieuwigheden kunnen kennismaken. Voor sommigen is de vakantieperiode de ideale tijd om de dagdagelijkse bezigheden even opzij te schuiven en tijd te maken voor wat nieuws. Zo verwacht ik dat leergierige ontwikkelaars de CLR even zullen laten voor wat hij is en een duik nemen in de DLR. De DLR, zegt U? Nu Google als merk meer waard is dan Microsoft, Yahoo niet staat te trappelen om te worden overgenomen, de populariteit van Second Life en andere virtuele werelden blijven toenemen, is het tijd voor Ray Ozzie om uit de schaduw van z'n voorganger Bill Gates te treden. En zo was Mix07 – Microsoft's webconferentie in Las Vegas – het ideale platform om Microsoft's visie voor het bouwen van rich internet applications wereldkundig te maken. Heel wat nieuwigheden zoals Microsoft Silverlight, Windows Live Developer Platform API's, Expression Studio, Silverlight Streaming en de CLR met volledige ondersteuning voor de DLR hebben dit voorjaar het licht gezien. En dan vergeet ik nog de datagerelateerde projecten Astoria en Jasper, Silverlight voor Mobile applicaties, PopFly, et cetera. Microsoft's visie van Software + Services (S+S) heeft na de BizTalk Services nu met Silverlight Streaming een tweede mooi voorbeeld klaar dat aantoont hoe services een softwareplatform een toegevoegde waarde kunnen geven. Maar ik ging het hebben over de DLR, nietwaar?

Silverlight 1.1 omvat een crossplatform versie van het .NET Framework en maakt het mogelijk om rijke .NET-applicaties voor de browser te bouwen. De in Silverlight ingebouwde versie van Microsoft's Common Language Runtime (CLR) is dezelfde virtuele machine waarin alle .NET-applicaties werken. Dit alles betekent dat men binnenkort toepassingen, die gebruik maken van het WPF UI programmeermodel, .NET datatypes en LINQ-bibliotheken zal draaien op Mac OS X. Voorts zijn de performantieverbeteringen voor deze CLR-browser-toepassingen indrukwekkend vergeleken met de snelheid van geïnterpreteerde JavaScript-code.

Bovenop de CLR biedt Silverlight ondersteuning voor zogenaamde dynamische talen via de Dynamic Language Runtime (DLR). Terwijl de CLR vandaag reeds dynamische talen zoals IronPython 1.0 ondersteunt, voegt de DLR een aantal services aan de CLR toe. Voorbeelden hiervan zijn het dynamic type system, het standard hosting model en de support voor dynamic codegeneratie. Met deze nieuwe features wordt het veel eenvoudiger om code te gaan delen tussen de enerzijds krachtige statische talen zoals VB.NET of C# en anderzijds de dynamische talen. Vandaag worden vier talen op de DLR gebouwd: Python, JavaScript (EcmaScript 3.0), Visual Basic en Ruby.

Terwijl Microsoft-ontwikkelaars in het verleden reeds in staat waren om applicaties in deze dynamische talen te coderen, openen deze features samen met de DLR een nieuwe wereld voor heel wat ontwikkelaars.

Wil dit zeggen dat ontwikkelaars afscheid nemen van de statische talen ten voordele van dynamische talen? Vaarwel CLR, welkom DLR? Zo'n vaart zal het waarschijnlijk niet lopen. De kracht

van de DLR op het .NET-platform ligt namelijk in de integratie van de CLR- en DLR-omgevingen. Beide runtimes worden met elkaar verweven en zullen leiden tot een nieuw programmeermodel. Vanaf heden kan ik kiezen in welke taal ik client-code wil schrijven.

Ik voorspel anderzijds wel een toenemende interesse voor dynamische programmeertalen. Het flexibele typingsysteem van deze talen geeft ontwikkelaars vandaag de mogelijkheid om zeer

snel prototypes te bouwen of zelfs volledige webapplicaties te ontwikkelen. Ruby en Python hebben hieraan een groot deel van hun succes te danken. Beide talen zijn eenvoudig te leren en uitermate geschikt voor scripting- en prototyping-doeleinden. Met de integratie van de CLR en de DLR geeft Microsoft ontwikkelaars nu de keuze om de geschiktste taal voor elke taak te kiezen, zonder de ondersteuning voor vertrouwde bibliotheken te verwaarlozen.

Alsof dit alles nog geen voldoende vernieuwing brengt, heeft Microsoft de volledige broncode van zowel IronPython als het volledige DLR-platform onder Microsoft Permissive License ter beschikking gesteld. Ondertussen hebben de open source-adepten een versie van IronPython werkend op Mono. Dit betekent dat de Silverlight plug-in nu ook IronPython ondersteunt onder Linux-omgevingen met Mono. Het lijkt dus enkel een kwestie van tijd om IronRuby, VBx en Smalltalk te laten draaien op Linux. Van mobiele telefoons tot Solaris: de DLR komt eraan! Ik kan enkel besluiten dat ontwikkelaars van Rich Internet Applicaties – dekt de vlag de lading? – een mooie tijd tegemoet gaan nu Microsoft uitpakt met ondersteuning van het desktop programmeermodel in de browser, de integratie van dynamische talen op het .NET-platform, een aantal nieuwe Windows Live API's die het mogelijk maken om echte S+S-toepassingen te bouwen. De ontwikkel- & design-omgevingen Visual Studio en Expression Studio vormen - met de uitwisseling van XAML - de kers op de taart tot een productieve samenwerking tussen designer en developer. Microsoft heeft, na de verovering van de desktop, een beweging ingezet om een belangrijke rol te gaan spelen in de wereld van de browsertoepassingen. Adobe maakt net de omgekeerde beweging. Laat u niet verblinden, de slag tussen beide softwaregiganten zal niet in de browser of op de desktop worden gewonnen. Het type toepassingen dat jullie ontwikkelaars zullen bouwen bovenop deze programmeermodellen (XAML) en het gebruik van de beste ontwikkelomgevingen, zullen het gevecht beslissen.

Kunnen jullie ontwikkelaars de innovaties nog baas? Laat ons hopen dat de zomer jullie aanzet tot de ontwikkeling van Silverlight-applicaties. Of misschien wil je wel een mashup bouwen met Popfly? Of een Silverlight DLR-applicatie integreren met SharePoint? ... Of verkiezen jullie toch om de CLR te laten varen en voortaan enkel DLR-toepassingen te bouwen? Alvast een prettige vakantie!

David Boschmans  
Development Platform Advisor  
Microsoft Belux