

“Voor de toekomst van Linux is Mono de juiste weg om applicaties te ontwikkelen.”

EEN GESPREK MET MIGUEL DE ICAZA VAN PROJECT MONO

Miguel de Icaza is oprichter en president van de Gnome Foundation. Hij is ook medeoprichter en CTO van Ximian, dat onlangs door Novell werd overgenomen. In 1999 ontving Miguel de Icaza voor zijn werk de ‘Innovator of the Year Award’ van het befaamde Massachusetts Institute of Technology (MIT). Op dit moment werkt hij voornamelijk aan het Mono-project, een open source .NET-implementatie. Mono ontving dit jaar op de LinuxWorld conferentie de ‘Best Open Source Project Award’.

Mono is een .NET implementatie. Waarom hebt u niet gekozen voor een Java-implementatie of het ontwerpen van een eigen applicatieplatform?

Miguel de Icaza: “Er waren verschillende factoren die hierin een rol hebben gespeeld. De nadruk van Microsoft op cross-language interoperabiliteit als een mechanisme om het platform te laten groeien, terwijl we toch de vrije hand hadden om een taal te kiezen, was heel bekend voor ons. In het Gnome-project hebben we altijd geprobeerd onze API’s zo toegankelijk mogelijk te maken voor zo veel mogelijk programmeertalen. We zagen voor bepaalde taken de toegevoegde waarde van taalspecifieke voor- delen. Het ontwerp van .NET strookte met deze overtuiging. Vanuit ons perspectief had Microsoft bovendien een platform gecreëerd dat de beste features

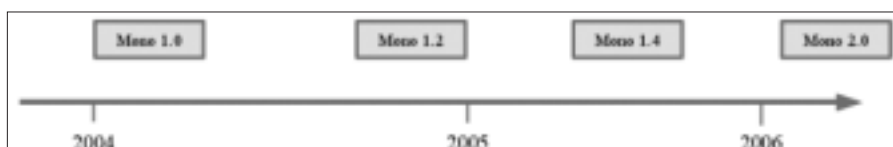
van Java had, en ook de door verschillende mensen voorgestelde verbeteringen van Java erin had betrokken. Mono is interessant om twee redenen: het is een hoognodige verbetering van het Unix ontwikkelplatform, en het maakt het ontwikkelaars die .NET onder Windows gebruiken mogelijk om hun software ook op Unix/Linux-systemen aan te bieden.”

Wat is de reikwijdte van het Mono-project? Wat zal worden geïmplementeerd en wat niet?

Miguel de Icaza: “Er is nu een roadmap voor Mono 1.0 en Mono 1.2. Op dit moment zijn de code-features die in Mono 1.0 zullen verschijnen, bijna compleet. We richten ons op het fixen van de bestaande bugs, het verbeteren van de performance en het tunen van Mono. De roadmap wordt onderhouden op: [\[mono.com/mono-roadmap.html\]\(http://mono.com/mono-roadmap.html\). In essentie zal Mono 1.0 de volgende zaken bevatten: een C#-compiler, een JIT execution system voor x86 and PPC, een ahead of time compilation engine om de opstart-snelheid te verbeteren, core class libraries \(mscorlib, System.XML\), ASP.NET, dat wil zeggen WebForms en webservices, en ADO.NET met uitgebreide ondersteuning voor third party libraries. Dit zijn de componenten die we denken te kunnen leveren in het 1.0 timeframe. Andere componenten zullen op dit punt alleen als bèta beschikbaar zijn. We zijn bijvoorbeeld bezig aan de nieuwe features van C#, en de benodigde features van de runtime om generics te ondersteunen. Verdere informatie is beschikbaar via onze roadmap.”](http://www.go-</p>
</div>
<div data-bbox=)

Mono 1.0 is nog niet klaar, terwijl Microsoft al werkt aan de 2.0-versie van .NET. Hoe blijven jullie bij?

Miguel de Icaza: “We blijven werken aan de ondersteuning van beschikbare features die in de nieuwere versies van .NET zullen verschijnen. We zullen echter nooit



in staat zijn de software op dezelfde tijd te shippen als Microsoft, en er zal waarschijnlijk altijd een achterstand in tijd blijven bestaan. Het goede nieuws is dat technologie niet hetzelfde is als bier; het wordt niet direct geassimileerd. Het kan maanden of jaren duren om de nieuwe features te deployen. Ontwikkelaars die alleen Unix gebruiken zullen op de features moeten wachten tot ze beschikbaar zijn. Ontwikkelaars die Windows gebruiken kunnen hun applicaties al bouwen en ze later op Unix/Linux-systemen deployen. Ik moedig ontwikkelaars zelfs aan Visual Studio .NET of SharpDevelop onder Windows te gebruiken om applicaties te ontwikkelen, omdat we nog geen eigen IDE hebben. We beperken ons echter niet tot de features van het .NET Framework. We hebben samen met andere teams libraries ontwikkeld die iedereen kan gebruiken, zowel Mono- als .NET-developers. Ook hebben we libraries ontwikkeld die profiteren van onze eigen platformspecifieke features (Unix-features en Gnome-features)."

Wat is de relatie tussen Ximian, het Gnome-project en Mono?

Miguel de Icaza: "Gnome is een project dat een desktop-omgeving produceert voor Unix-systemen. Een behoorlijk complete omgeving die toegankelijkheid voor slechtzienden biedt, en een heel sterke

focus op bruikbaarheid en stijl. Ximian, dat nu deel uitmaakt van Novell, is een bedrijf dat is gericht op productontwikkeling voor de Linux-desktop voor gebruik bij overheden en ondernemingen. Als onderdeel van deze inspanning hebben we grote bijdragen geleverd aan Gnome en OpenOffice, en bouwden we de eerste e-mail/PIM-client voor Linux: Evolution. Evolution is een behoorlijk uitgebreid stuk software, en was één van de redenen om het Mono-project te starten. Het was duidelijk dat we betere ontwikkeltools nodig hadden, en .NET had ze."

Naar .NET wordt wel eens verwezen als COM 2.0. Is Mono hetzelfde als Bonobo 2.0? Is er een relatie tussen beide producten?

Miguel De Icaza: "Ik denk dat het onjuist is om naar .NET te verwijzen als COM 2.0. Het is een volledig nieuwe codebase, maar kan gewoon met de oude en de nieuwe wereld communiceren. Hier is Microsoft altijd goed in geweest: binaire compatibiliteit en de integratie van oude technologieën met nieuwe. Het is duidelijk dat Microsoft wel degelijk de problemen van de ontwikkelaar begrijpt, en ze doen er een hoop aan om hun ontwikkelaars te ondersteunen. Dit was ook weer iets dat ons na aan het hart lag; we wilden niet al onze investeringen weggooiën en vanaf nul beginnen. We willen de

oude code-bases in C en C++ blijven behouden, maar de nieuwe code in C# schrijven. De verantwoordelijke personen voor de architectuur van het .NET Framework hebben dit begrepen, en dat is opnieuw goede reden om deze visie te volgen. Bonobo heeft verscheidene COM-concepten gekopieerd en verschillende interfaces om een componentmodel en een compound document system te implementeren. Het laatste is nooit volledig gerealiseerd, het werd niet echt gewaardeerd, en vandaag wordt het voornamelijk gebruikt om controls te 'embedden' en ingewikkelde RPC-calls in de Gnome-desktop te doen. We voorzien in bruggetjes naar de oude Gnome-technologieën als onderdeel van het Gtk#-project. Dit project is in wezen een 'binding' van onze oude technologieën zodat ontwikkelaars ze kunnen benaderen vanuit Mono."

De Open Source Community heeft wat bezwaren met betrekking tot Mono. Microsoft zou een niet-compatibele versie kunnen introduceren, patentzaken zouden een rol kunnen spelen, en de Open Source Community zou wel eens te afhankelijk van Microsoft kunnen worden.

Miguel de Icaza: "Daar kan ik duidelijk in zijn. Microsoft is erg goed in 'backward' compatibiliteit. Ze begrijpen dat als ze dit niet ondersteunen vele mensen ontsteld zullen zijn. Dus is het zaak de compatibiliteit met de bestaande code niet in de weg te zitten. Het potentiële gevaar met de applicatieportabiliteit zou zijn dat Microsoft het gebruik van niet algemeen gedocumenteerde 'private' API's zou aanmoedigen. Wanneer dergelijke API's worden gebruikt, zullen de migraties naar een ander systeem niet een rechttoe rechtaan drag-and-drop exercitie zijn, maar zouden ontwikkelaars hun code moeten herbouwen om gebruik te kunnen maken van een alternatieve API voor Unix. Het blijft mogelijk dat je in enkele gevallen toch een paar veranderingen moet aanbrengen aan jouw code om beter gebruik te kunnen maken van de features van een platform. Het probleem met patenten vindt niet plaats in de kern, maar zou zich



kunnen voordoen bij class-libraries van hoger niveau. De kern kan door iedereen worden gebruikt, zodat patenten alleen een effect hebben op hen die de niet-ECMA gestandaardiseerde API's gebruiken. Als er patenten zijn op de code, zullen we de open source versie daar niet mee belasten. We gebruiken het volgende standaardproces: Eerst proberen we een implementatie van onderdelen van het patent te vinden die ouder zijn dan het patent zelf, de zogenaamde 'prior art' (juridische term, red.). Is die er niet, implementeer dan dezelfde functionaliteit met een ander algoritme. Als dit niet mogelijk is, zullen we de code uit Mono moeten verwijderen. Als we gedwongen zijn de code uit Mono te verwijderen, kunnen mensen op plaatsen waar het patent niet van toepassing is nog steeds de code gebruiken, maar personen in landen waar het patent wel van toepassing is zullen een licentie van Microsoft moeten betrekken om die code te draaien. Wij zullen daarbij behulpzaam zijn."

Welke invloed heeft de recente acquisitie door Novell op de ontwikkeling van Mono?
Miguel de Icaza: "Die is goed, intern wordt het voor enkele projecten gebruikt, en het is ook onderdeel van onze Linux-strategie: we zijn erop uit om van Linux een levensvatbaar platform voor de toekomst te maken. Een levensvatbaar platform voor iedereen betekent niet alleen het verminderen van de hardwarekosten, maar het ontwikkelaars ook mogelijk maken complexere, krachtigere applicaties te bouwen; en Mono is een manier om dit te doen."

Zijn Ximian, Novell of een andere partij van plan om met Mono serieuze applicaties te bouwen? Zullen applicaties als Evolution uiteindelijk migreren?

Miguel de Icaza: "Absoluut, de komende versie van Evolution zal uitkomen met 'bindings' naar de CLI. Hierdoor kunnen Unix-ontwikkelaars extensies schrijven of data halen uit Evolution. Andere projecten zijn begonnen C# te gebruiken – zowel intern bij Novell/Ximian als in de community. Misschien heb je F-Spot gezien of Nat's Dashboard?"

Glossary

Gnome	Is een Open Source project met als doel de realisatie van een gebruiksvriendelijke desktopomgeving met bijbehorende applicaties. Gnome is de standaard grafische omgeving voor veel op Linux en op BSD gebaseerde distributies.
Ximian	Is een bedrijf dat voortkomt uit het Gnome-project. Het levert onder andere de Ximian Desktop, een kant en klare grafische werkomgeving, Evolution een Outlook-achtig programma, en Red Carpet, een product dat software-updates verzorgt.
Bonobo	Een door COM geïnspireerde technologie met als doel de Open Source wereld te voorzien van een gestandaardiseerde infrastructuur voor softwarecomponenten. Net als COM maakt Bonobo het mogelijk binaire componenten te hergebruiken op basis van interfaces.
Gtk(#)	De Gimp Toolkit biedt ontwikkelaars gereedschappen en een set widgets om GUI's mee te ontwikkelen. De toolkit is op vrijwel elk platform beschikbaar. Gtk# is de naam voor de bindings die het mogelijk maken om vanuit .NET Gtk te gebruiken.
ECMA	European Computer Manufacturers Association. Een organisatie die de '.NET-standaarden beheert'. De meeste van deze standaarden worden ook bewaakt door de ISO.
CLI	Common Language Infrastructure. De benaming van de runtime-omgeving in de ECMA-standaarden. In de .NET-wereld praten we over de Common Language Runtime: Microsoft's implementatie van een CLI. In de Mono-context praat men over de 'Mono runtime'.
Wine	is een Open Source implementatie van de Windows API.
Qt(#)	Qt is net als Gtk een multiplatform GUI-toolkit. Qt# zijn de bindings, zodat deze toolkit ook vanuit .NET kan worden gebruikt.
GPL - General Public License	Is een Open Source licentievorm. Al het werk dat op basis van GPL-code is gebouwd valt volgens de licentievoorzwaarden automatisch onder de GPL-licentie, en wordt daarmee Open Source. Een andere voorwaarde van de GPL is dat bij externe distributie van de software de broncode altijd beschikbaar moet worden gesteld aan het publiek.
X11	Is een Open Source Licentie die eigenlijk geen enkele voorwaarde aan de gebruiker of het gebruik van de betreffende source code stelt, anders dan dat het oorspronkelijke copyright wordt vermeld en de oorspronkelijke auteur wordt gevrijwaard van iedere aansprakelijkheid.

Het probleem voor crossplatform .NET applicaties zou het maken van GUI-applicaties kunnen zijn. Er zijn opties zoals het gebruik van GTK#, QT# of WineLib 'bindings'. Naar welke optie gaat uw voorkeur uit en waarom?

Miguel De Icaza: "Uiteindelijk zullen we Windows.Forms ondersteunen, maar het is een groot framework. Het zal dus wel enige tijd duren voordat we er zullen zijn. In de tussentijd is Gtk# snel vooruitgegaan. Mensen gebruiken het om crossplatform applicaties te ontwikkelen die zowel op Windows als Unix draaien; met native userinterfaces voor elk platform."

De ECMA-standaarden omvatten niet alle gebieden van het Microsoft .NET Framework zoals de System.Windows namespace. Hoe gaat u om met deze niet-gestandaardiseerde namespaces, en welke namespaces zou u graag gestandaardiseerd willen zien die dat nu nog niet zijn?

Miguel de Icaza: "De ECMA-specificatie is handig, omdat het de 'hard parts' van het .NET Framework bevat, zoals de C#-taal, de VM, de bestandsformaten, de low-level data lay-out. De zaken buiten de

ECMA-spec zijn relatief rechttoe rechtaan implementaties. Het zijn gewoon classes, en de publieke specificatie van Microsoft heeft alles wat nodig is om de functionaliteit te implementeren. Sommige zaken zijn complexer dan andere. De XmlSerializer bijvoorbeeld is een technologie die behoorlijk compleet is wat betreft features, maar het is niet gemakkelijk om alle hoeken en gaten vanuit de documentatie te behandelen. Je moet groei dus al in de architectuur incalculeren, en de code al doende verbeteren, refactoren, reviewen en vernieuwen."

Wat zijn de voornaamste verschillen tussen DotGNU en Mono? Werken jullie samen?

Miguel de Icaza: "Technisch zijn wij verder. We hebben een JIT-compiler, een pre-compiler, een heel volwassen C#-compiler, en uitgebreide regressie-testsuites voor elk component van het systeem. We willen Mono ook positioneren als een tool die iedere ontwikkelaar kan gebruiken: Open Source softwareontwikkelaars en commerciële ontwikkelaars, en we willen commerciële bedrijven ondersteunen in het gebruik.

(vervolg op pagina 89)