

Projekty

W bieżącym wydaniu „Projektów pod lupą” przyjrzymy się otwartym odpowiednikom oferty Microsoftu, kryjącym się pod nazwą .NET: wydanej niedawno pierwszej oficjalnej wersji Novell Mono oraz mniej znanemu, choć równie ciekawemu, projektowi dotGNU.

ARTUR SKURA



Ronald Raefle, visipix.com

Mono

Fascynacja Miguela de Icazy małpami jest powszechnie znana w środowisku Wolnego Oprogramowania. Miguel, główny architekt GNOME i lider Mono, od wczesnej młodości interesował się wszelkiego typu czelokoształnymi stworzeniami. Przykładowo, koncepcję komponentów w GNOME ubrał w nazwę „bonobo”, niezwykle aktywnej seksualnie małpy, często angażującej się w związki z różnymi partnerami. Nie powinno więc nikogo dziwić, że „mono” po hiszpańsku znaczy „małpa”.

Kiedy Miguel oświadczył, że wraz z kolegami z Ximiana zamierza stworzyć otwartą implementację .NET dla Linuksa, nikt nie był zdziwiony: młody programista od dawna z powodzeniem rozwijał projekty stanowiące bezpośrednią konkurencję dla sztandarowych aplikacji Microsoftu, takie jak np. Gnumeric, będący niemal wierną kopią Excela („jeśli jakaś funkcjonalność istnieje w Excelu, a brak jej w Gnumericu, uważamy to za błąd i pracujemy nad jej dodaniem w naszej aplikacji”) czy Evolution, odpowiednik Outlooka. Icaza znany był również z sympatii do koncepcyjnych rozwiązań Microsoftu, co nie przysporzyło mu wielkiej popularności w uniksowo-linuksowym środowisku, gdzie Microsoft postrzegany

jest głównie przez pryzmat nie zawsze najwyższej jakości produktów, takich jak Windows 9x czy też kompromisów związanych z bezpieczeństwem, które doprowadziły do rozmnożenia się wszelkiego rodzaju wirusów i robaków. Miguel potrafił jednak dostrzec wartość, która tkwi w .NET, i przekonać do niej sporą część środowiska.

Czym jest .NET? Jeszcze kilka lat temu trudno było udzielić jednoznacznej odpowiedzi, zaś sami przedstawiciele Microsoftu gubili się w wyjaśnieniach. Na przykład według jednej z broszur marketingowych wydanych w 2000 roku .NET miało być platformą rodziny produktów Windows 2000, wiele osób uznało więc .NET za klasyczny przykład tzw. *rebrandingu*, czyli sprzedaży starych produktów pod nową nazwą. Dziś już znacznie łatwiej jest odróżnić marketingowe hasła od faktycznej oferty związanej z .NET i jej poszczególnych składników. Mamy zresztą do czynienia z dość dynamicznym procesem, w którym nie brak burzliwych zwrotów: przykładowo, jednym z fundamentów .NET według Microsoftu miały być usługi uwierzytelniające, znane jako Microsoft Passport, które wywołały burzliwą krytykę i odpowiedź w postaci Liberty Alliance Sun-a. Dziś Microsoft już nie uważa Passport za fundament .NET, zaś Liberty Alliance praktycznie przestało ist-

nieć. W istocie głównym zadaniem .NET jest utrzymanie dominacji Microsoftu na komputerach użytkowników w świecie, w którym coraz większą rolę odgrywa nie pojedyncze stanowisko, lecz komunikacja pomiędzy różnymi urządzeniami. Jest to również mocna odpowiedź na Javę Suna – zarówno jeśli chodzi o maszynę wirtualną, jak i język (C#).

W praktyce Microsoft udostępnia infrastrukturę .NET jako tzw. „.NET framework”, który zawiera dwa główne elementy: maszynę wirtualną dla języków zgodnych z .NET (Common Language Runtime) oraz biblioteki klas, zawierające trzy podstawowe zestawy komponentów: ASP.NET (dla aplikacji sieciowych), ADO.NET (komunikacja z bazami danych) oraz Windows.Forms (odpowiedzialne za graficzny interfejs użytkownika), zawierające funkcjonalność niezbędną do tworzenia nowoczesnych aplikacji. Mono ma ambicję stać się pełnowartościową implementacją infrastruktury .NET dla Linuksa (a przy okazji również innych systemów uniksowych, takich jak FreeBSD czy OS X, a nawet... dla samego Windows). Zawiera bowiem maszynę wirtualną, kompilator C# oraz część bibliotek (głównie ASP.NET i ADO.NET).

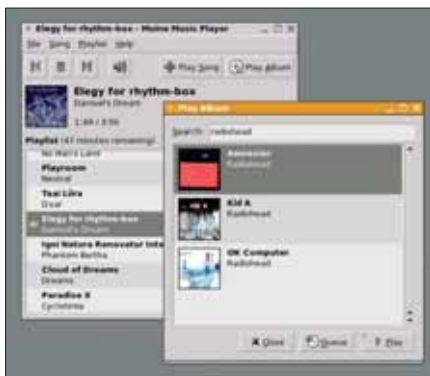
Jakie są szanse na sukces Mono? Odpowiedź na to pytanie nie jest łatwa, zależy

Linux Magazine Newsletter

Bądź zawsze na bieżąco!



<http://www.linux-magazine.pl/Newsletter>



Odtwarzacz Muine, jedna z najbardziej znanych aplikacji działających na platformie Mono.

bowiem w dużej mierze od postrzegania projektu przez całe środowisko, w tym potencjalnych programistów. W Microsoft nad .NET pracuje bezpośrednio kilkuset programistów, czyli znacznie mniej niż kompetentnych specjalistów tworzących Wolne Oprogramowanie. Z drugiej strony w Novellu nad Mono pracuje ich zaledwie kilkunastu. I choć potencjał nie zawsze znajduje odbicie w liczbach, zaś dokonania Mono (jak i konkurencyjnego dotGNU) już teraz zasługują na uznanie, można się spodziewać, że jedną ze strategii Microsoftu w walce z Mono będzie dodawanie kolejnych funkcji, których implementacja będzie wymagała czasu i wiecznego „ścigania” oryginału. Novell ma jednak asa w rękawie w postaci... całego systemu GNU/Linux, dla którego przecież powstaje (z założenia tylko neutralne co do platformy) Mono.

Tak więc Miguel liczy, że przyciągnie liderów nowych projektów dowiązaniem do bibliotek GTK+ i GNOME oraz innymi rozszerzeniami, np. większą ilością obsługiwanych silników baz danych w ADO.NET (np. Sybase i SQLite). Nie da się jednak ukryć, że z punktu widzenia zwykłych użytkowników niezwykle ważna jest dobra, tj. stabilna, szybka i bezbłędnie działająca implementacja Windows.Forms, bo to z nią będą mieli bezpośrednią styczność. Java nie zrobiła oszałamiającej kariery na biurkach, głównie z powodu zastrzeżeń co do wymagań i szybkości działania interfejsu użytkownika. Trudno oczekiwać, by chropowato działające Windows.Forms wzbudziły zaufanie do Mono jako stabilnej platformy, zarówno dla użytkowników jak i programistów – a dopracowanie ich wymaga jednak czasu i odpowiednich zasobów. Silną stroną projektu może być jednak ASP.NET i ADO.NET, zważywszy na fakt, że Linux już w tej chwili ma

bardzo silną pozycję na rynku serwerów WWW: możliwość wykorzystania mocno promowanych przez Microsoft rozwiązań na tej platformie może tylko wpłynąć pozytywnie na jej popularność.

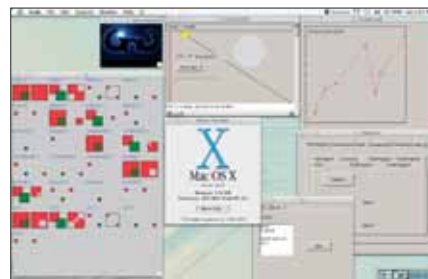
Po wydaniu wersji 1.0 Novel nie zasypia gruszek w popiele. Wersja 1.2, która ma się pojawić pod koniec roku, będzie zawierała wiele nowości i może stanowić faktyczny przełom w rozwoju projektu. Mono 1.2 ma zawierać m.in. kompilator C# 2.1 i obsługę Windows.Forms. Równocześnie część programistów intensywnie pracuje nad implementacją infrastruktury .NET 2.0, której rozmiary niemal dwukrotnie przekraczają ilość kodu zawartego w wersji 1.1. Jednak wersja końcowa Mono 2.0 ukaże się nie wcześniej niż w 2005 roku.

dotGNU

DotGNU to projekt mający na celu stworzenie alternatywy dla całej strategii .NET Microsoftu, nie tylko infrastruktury, której odpowiednik w dotGNU nazywa się Portable.NET. Nie da się jednak ukryć, że jest to najważniejszy element projektu, bowiem pozostałe – phpGroupWare, którego autorzy zdecydowali się włączyć do rozwoju dotGNU, oraz serwer aplikacji DGEE, bardzo związane z usługami WWW, nie budzą tak wielkiego zainteresowania jak infrastruktura .NET.

Od strony ideologicznej silną motywacją leżącą u podstaw projektu jest chęć powstrzymania Microsoftu przed zdobyciem kolejnej twierdzy, której opanowanie mogłoby przynieść ogromną szkodę nie tylko środowisku Wolnego Oprogramowania, ale wszystkim użytkownikom komputerów. Stworzenie z.NET najpopularniejszej platformy komunikacyjnej między różnego typu komputerami i urządzeniami mogłoby doprowadzić do sytuacji analogicznej jak w przypadku systemów operacyjnych na biurkach, gdzie użytkowanie komputera niemal zawsze wiąże się z wniesieniem opłaty Microsoftowi, choćby po to, by móc skorzystać z ulubionej aplikacji czy nowego sprzętu.

DotGNU jest więc w pewnym sensie ideologicznym rozwinięciem koncepcji leżących u podstaw GNU na platformę .NET. Widać więc wyraźnie różnicę między dotGNU a Mono: Novell rozwija Mono, szukając odpowiednich zastosowań biznesowych, które pozwoliłyby skapitalizować ogromny potencjał tkwiący w projekcie. Obecnie Mono jest wykorzystywane do rozwoju wewnętrznych projektów w Novellu,



Demonstracja możliwości Windows.Forms zaimplementowanych w Portable.NET na Mac OS X.

ale prędzej czy później można się spodziewać stworzenia kompletnej oferty związanej z Mono. Z kolei dla twórców dotGNU sukces biznesowy nie jest oznaczony wysokim priorytetem, stawiają oni bowiem na inne aspekty projektu, takie jak – zawarta już w nazwie – przenośność.

Od strony technicznej Portable.NET charakteryzuje się jedną ważną cechą, odróżniającą go od Mono: zawiera implementację Windows.Forms nie korzystającą z istniejących bibliotek, lecz wykonującą wszystkie operacje rysowania samodzielnie, przy czym wszystkie kontrolki zostały zaimplementowane w C#. Dzięki temu programy korzystające z GUI, uruchomione na Portable.NET, mają szansę znacznie bardziej przypominać aplikacje windowsowe niż uruchomione na Mono. W praktyce jednak obu implementacjom jeszcze dużo brakuje.

Osoby zainteresowane przyszłością rozwoju oprogramowania powinny zwrócić uwagę zarówno na Mono, jak i dotGNU. Trudno na obecnym etapie mówić jeszcze o sukcesie, trzeba jednak zdać sobie sprawę, że istnienie otwartych implementacji .NET jest niezbędne – w równym stopniu, co w przypadku systemów operacyjnych, pakietów biurowych czy kompilatorów. Bez nich bowiem środowisko użytkowników Wolnego Oprogramowania może znaleźć się w sytuacji, w której było jeszcze kilka lat temu, zaś zwykli użytkownicy komputerów mogą być zmuszeni do stworzenia kolejnej trudnej do zerwania więzi z dostawcą platformy do wykonywania aplikacji i komunikacji sieciowej. ■

INFO

Projekt Mono:
<http://www.mono-project.com>

Projekt dotGNU:
<http://www.dotgnu.org>