Mono and GNOME

Miguel de Icaza
miguel@ximian.com

Feb 08, 2002

It is sentimental for me after reading all the nice e-mail on the GNOME lists. By contrast, to write software alone is not that interesting, the most interesting part is to interact with other developers, and watch how community projects grow.

I am not sure what people told Richard Stallman about my plans. Given the confusion surrounding .NET, it is very possible that people were asking 'Miguel wants to depend on Passport' or something just as bad as that.

My only intention is to write applications using the CLI as a development platform, which is really not very exciting for a news paper to report: "Programmer to use new compiler, new garbage collector, news at 11".

Really, programmer's lives are boring, I wish my life would be as exciting as other people's life appear to be.

Before starting though, I would like to ask my readers to forget everything they have heard about .NET, because it is a marketing term used to describe many different Microsoft projects, and there is a lot of information both correct and incorrect about it floating around.

My goals with Mono are very specific, and I will address those shortly, but for the sake of getting things done, please forget everything you have heard about .NET.

1 First, the Facts

GNOME is not adopting Mono or .NET as an implementation technology. The headline from the Register is misleading, for a number of reasons:

- The headline does not reflect any statements I made on the interview (if you read the interview you will notice this).
- The only future plans that have been approved by the GNOME team (which has 11 voting members on its board) are found here:

  http://developer.gnome.org/dotplan/
- I am not the GNOME foundation or control GNOME like Linus controls his kernel, I am just its founder and a contributor.
- GNOME is not built by an individual, its built by a team of roughly 500 contributors in many areas.
- Decisions in the GNOME world are done by active contributors and module maintainers. I have given my maintainership status on every module I maintained to other members of the GNOME team as I got more involved with Ximian and later on with Mono.

So effectively I have no "maintainer" control.

At this point on time, the GNOME team is working on shipping version 2.0 of the desktop and the development platform, a major upgrade to the desktop offering, and everyone is quite excited with this.

2 What is Mono?

Mono is an implementation of three pieces of technology:

- A compiler for a new programming language, similar to Java, called C#.
- A virtual machine for the Common Intermediate Language (CIL) byte codes.
- A set of libraries that encapsulate useful routines and classes: from hash tables, to XML manipulation, to database management, to GUI applications, to web construction tools.

These are usually referred in the Microsoft world as the '.NET Framework' as opposed to .NET. When I say '.NET Framework' here, I am talking about these technologies.

Seasoned industry programmers will notice that the above is very much like Java and the Java VM. They are right, the above is just like Java.

The CIL has one feature not found in Java though: it is byte code representation that is powerful enough to be used as a target for many languages: from C++, C, Fortran and Eiffel to Lisp and Haskell including things like Java, C#, JavaScript and Visual Basic in the mix.

I wish I had the time to go in more detail, but for the sake of this argument, the above will suffice.

Although Ximian can only finance the work of a C# compiler (that is all the resource I have at my disposal), I want to encourage other people to work on free implementations of other compilers.

I want to encourage other developers to look at targeting existing compilers and interpreters to the CLI: JavaScript, Basic, Perl, Python, C++, and maybe even get gcc core to generate CIL bytecodes.

2.1 The CIL and the promise of language independence:

Bertrand Meyer (the father of Eiffel) wrote an interesting article that encapsulates my excitement about the possibilities of the CIL:

http://eiffel.com/doc/manuals/technology/bmarticles/sd/dotnet.html

This technology allows programming languages to be considered on the basis of how they will perform for a given task, and not based on the runtime libraries that you will depend. Any software engineer should read this article:

http://www.fawcette.com/dotnetmag/2001_12/online/online_eprods/bme

So no longer should a software engineer pick Fortran, because that is the only language where his math libraries are available: he can now pick the right language for the problem at hand.

3 Mono and GNOME

GNOME had always tried to have a good support for multiple programming languages, because we realize that no matter how much we loved C as a programming language, there was a large crowd of people out there that would like to use the GNOME libraries from their favorite programming language, which might not necessarily be C.

This strategy has paid off very well. There are healthy and striving Python, Perl, Guile and Ada communities out there that use the Gtk+ and Gnome bindings to build applications. From rapid prototyping to robust applications: we wanted to empower developers.

Keeping language bindings up to date and shipping them on time has always been a consuming process, because no matter how automated this process has turned out to be, there is still a considerable amount of manual work that needs to be done.

I do go into more details about this at the following places:

http://www.go-mono.com/rationale.html

http://scriptingnews.userland.com/stories/storyReader$1275

4 An upgrade to the development platform: Part I.

Microsoft has terrible APIs to code against. Anyone who has used Win32 and any combination of the various layered cakes that have been built on top of it has stuck to that platform only because of the size of the market, but it is one of the most horrible APIs ever built.

To make things worse, an evolution of APIs, components, memory management contracts and patched up versions of COM have made the platform horrible.

Microsoft has injected fresh air into their platform by building and designing a new programming platform that addresses all these pains. They have incorporated many ideas from Java, and they have extended it to address new needs that developers had. They took where Java left off.

Now, the Unix platform, GNOME included has some of these problems: our APIs have been evolving. Libraries have been built by disconnected groups (PNG, JPEG, Gtk+, Xml, Bonobo, CORBA spec apis, etc) and the end result is that a developer eventually has to learn more than he wanted to in the course of developing a large application.

Ximian funded for a long time the work on the Perl bindings, and we had a lot of work going into Bonobo (more than we do today) because we believed that this would help us achieve language independence and empower scripting language developers (that is why we were so psyched about CORBA/Bonobo support all this time).

When C#, the CLR and the class libraries were launched, we looked at that, and we saw how they were solving the problem in a very nice way. At least it appealed to me and others from a purely technological standpoint. This new platform showed a lot of promise.

After much researching and debating, we decided that a couple of developers at Ximian will join me in working on a free implementation of these specifications. These people came precisely from the cross-language interoperability area: Dick Porter had been working before on ORBit and our SOAP implementation; Dietmar Maurer came from the Bonobo development world and Paolo Molaro was working on Gtk+/Gnome/Bonobo bindings for Perl. This is the original Mono developer lineup.

5 Evolution, Gnumeric and GNOME

I have written and maintained many lines of code as part of my GNOME work. Ximian has developed Evolution which consists of roughly 750,000 lines of code.

Large software projects expose a set of problems that can be ignored for smaller projects. Programs that have long life times have different dynamics when it comes to memory management than smaller programs.

There is a point in your life when you realize that you have written enough destructors, and have spent enough time tracking down a memory leak, and you have spend enough time tracking down memory corruption, and you have spent enough time using low-level insecure functions, and you have implemented way too many linked lists [1]

The .NET Framework is really about productivity: even if Microsoft pushes these technologies for creating Web Services, the major benefit of these is increased programmer productivity.

Evolution took us two years to develop and at its peak had 17 engineers working on the project. I want to be able to deliver four times as many free software applications with the same resources, and I believe that this is achievable with these new technologies.

My experience so far has been positive, and I have first hands experience on the productivity benefits that these technologies bring to the table. For instance, our C# compiler is written in C#. A beautiful piece of code.

It can be argued that I could be wrong, and that these technologies are too new. But my personal experience and the

[1] indeed, GNOME uses Glib which is a massive step up from the Unixy libc APIs.

experience of some of my friends with this platform has been amazing. I want to share with others this simplicity. And I want to empower developers: I want to enable a whole class of developers to create great desktop applications that integrate with GNOME.

6 Why is Mono related to GNOME?

It is no secret that I have been working on Mono as a new platform for software development, and it is also not a secret that I want to help the GNOME project with Mono. This has been the plan since the project was announced in July.

Mono will use Gtk+, Gnome-Db, Libart, Gnome-Print and other GNOME technologies as part of its implementation of its class libraries, because that is what my team and I are familiarized with.

So when you copy your binary from Windows that was compiled with the Visual Studio.NET and run it on your Unix platform, it will just integrate nicely with your GNOME desktop.

We are also exploring a port to MacOS X, and for that particular case, we will integrate with Aqua, not with Gtk+, but you get the idea.

7 GNU was based on a proprietary technology.

GNU is a free re-implementations of Unix. Linux is a re-implementation of the Unix kernel. Before the advent of Linux and the Berkeley Unix, Unix was a proprietary technology, built by ATT (which back in the day, was a monopoly).

Still, developers took what was good from Unix, and reimplemented a free version of it. Down to the Unix programming language: C (which was also invented at ATT). Even C++ was invented at ATT.

Think of Mono as following the same process: we are bringing the best technology out there to our beloved free software platform. And at the same time it serves to be a magnificent upgrade on the development platform.

8 I can not force anyone

Whether people in GNOME or elsewhere will use Mono is independent of my opinion. Mono will have to stand on its own feet, and will have to convince developers on its own merits before it succeeds.

When I made my comments to the Register reporter, I was envisioning that in a couple of years Mono would be a really solid technology: a good JIT engine, good class libraries and would be a useful platform for innovation: it would allow people to focus more on the problems at hand and worry less about the low-level details of the platform.

9 Rewriting GNOME

Havoc brought up an important point recently, an article from Joel Spolsky:

http://www.joelonsoftware.com/articles/fog0000000348.html

The short story is: rewriting code does not pay off, and I agree with the thesis of the article. Rewriting GNOME in C# with the CLR would be a very bad idea, if not the worst possible idea ever.

But what makes the .NET Framework technologies interesting is that they are evolutionary technologies:

9.1 The runtime can be linked into an application.

Example:

```
bash$ cat hello.c
```

```
#include <mono.h>
main (int argc, char *argv [])
{
      mono_init (argc, argv);
      mono_assembly_load ("classes.dll");
      mono_ves_execute ("Class.Main");
}
```

So existing applications can be “extended” with Mono, take a piece of code like Gnumeric, and write a new chunk of it using Mono for example.

9.2 There is no language switch required.

You can keep using your fav language, and gradually start writing new pieces of code in another language that runs with all the benefits of “managed” execution.

I go into some more detail here:

http://mail.gnome.org/archives/gnome-devel-list/2002-February/msg00021.html

10 GNOME 4

As you might realize by now, GNOME 4 is not planned, it is not possible to know what is in there. So my comments on GNOME 4 only reflect the fact that I personally believe that people will see that Mono is an interesting platform to write new applications.

So in the future the applications that will be shipped, very likely might contain Mono technologies. Whether this is limited to new applications only, or this is something used in more fundamental pieces of the system is an entirely different matter.

But for now, GNOME 4 is non-existant project.

11 Fighting the System

The .NET Framework will exist in the Windows world, and because of this they will be widely deployed. It is a pointless battle to pretend that boycotting the use of those technologies will have any kind of effect on their reach.

The .NET Framework stands on its own feet, and developers in the Windows world love it. Even if this was not the case, Microsoft is using these technologies and distributing to as many people as possible. We are witnessing the creation and deployment of a new standard. Sure, it has a lot of corporate support, but it will become a widely deployed technology.

12 Other uses of Mono

Despite my love for Mono as a tool for writing GNOME applications and giving developers new tools to write code in less time, there is an extra advantage in having a free implementation of the .NET Framework for Unix:

- Windows developers know how to write code for it.
- Lets make it easy to bring developers from the Windows world into our platform.
- Training materials, tutorials, documentation, tips and tricks are already available in large quantities, lets leverage this.

13 Mono Financing

Right now Mono is financed by Ximian because we believe that this will reduce our cost of development for future applications. And thats why we are really focused on Mono for the desktop (amusingly the ASP.NET support in Mono has evolved more rapidly, because Gaurav and Leen have been very excited about this, and just have been producing code like crazy).

So even in the Mono world, I do not get to make all the decisions: people work on what they are interested in developing.

The Mono community is great! Lots of passionate programmers work with us, and I feel very happy that I have had a chance to work with all of them.

At this point in time Ximian has only a small team of full time developers working on Mono (five) and a lot of the work is being done by contributors on their spare time, or hackers that want to see the .NET Framework run in other platforms, or people who share our enthusiasm for the platform, or people who just like to hack on a particular area and just love to code.

But I would like to hire more full time developers: the open source development model is great for getting the fun/short things done, but it is terrible to get the long-haul, boring, repetitive or dull things done.

I want to be able to bring more people to work full time on Mono. I would like to offer the services of Ximian as a project manager to keep driving this project forward, and get cash infusions to hire developers to work on this project.

The only restriction is that all of our work has to be free software. But other than that, I am ready to take money from anyone or listen to any kind of proposals for making this happen.

Some people wonder if we have got a Microsoft investment or contract (because I like this Microsoft technology). The answer is no. But I would take one if they wanted to fund my free software project ;-) Man, I wonder what that would be like!

Implementing the .NET Framework is a massive effort, and I want to enroll as many contributors as possible.

14 API compatibility

I believe that the 'Embrace and Extend' philosophy is bad for users and developers. Whether its a large corporation doing it, or ourselves. I want to be as compatible as possible with the APIs that were published by Microsoft.

This achieves various things:

- Allows developers to move back and forth.
- Reduces training.
- Helps us leverage existing knowledge.

Of course, this should not stop anyone from implementing new APIs. And I even encourage people to write new classes, APIs and components that will be reusable both on Unix and on Windows.

15 What if we never can keep up?

There is the issue that we might not be able to keep up (right now, we dont, as .NET Framework 1.0 is already out there, and we are, well still underway). Also, theoretically there is the risk of a given API being unimplementable on Unix.

Even if that is the case, we still win, because we would get this nice programming environment, that althought might not end up being 100improvement and would still help us move forward. So we can reuse all the research and development done by Microsoft on these ideas, and use as much as we can.

So far all it seems like everything in .NET can be emulated in our environment.

16 Further Debate

I have just scratched the surface in this email, I do like a lot the technology behind the .NET Framework as you might have noticed from the interviews, no secret there. I can go on for hours, but I have to set a limit to this email.

I hope this explanation will get us through, feel free to e-mail me if you believe I have missed something or if you are interested in contributing to make this vision happen.

I would like to thank all the people I have worked over the years: every GNOME developer past and present, every Mono developer past and present and all my friends at Ximian who have created a great place to work.

This community is great, and I have loved working with an increasing number of people as free software becomes more popular. I know sometimes I have been unreasonable, but I am trying to learn from my mistakes. Am just too good at being mistaken.