

Re: kilyx

Source:

<http://coding.derkeiler.com/Archive/Delphi/borland.public.delphi.non-technical/2007-02/msg01576.html>

- *From:* Brian Moelk <bmoelk@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Sat, 10 Feb 2007 11:10:40 -0500
-

Marco van de Voort wrote:

We already have the LCL, and a huge installed base in VCL (which LCL is somewhat compatible with, for good reason)

There is a large VCL install base and if LCL can leverage that *and* bring the third parties along with them, that would be great. However I have my doubts.

Many languages hook GUI layers on top that aren't built in their native tongue.

Usually the non critical scripting languages. How many C++ programs base themselves on Swing?

There's a difference between what the IDE is built in and what GUI framework the language supports.

If look back at CBuilderX, Borland built a C++ IDE around a Java core (Primetime). I was skeptical at first for the same reasons you are about hooking FPC into other IDE's (it's not a C++ IDE, etc.). But when I actually looked at it, I thought it showed promise. Eclipse and NetBeans both support C++ development.

If we look at GUI frameworks that are used from non-native languages, Mozilla is a good example. Here you have C++ doing all the "heavy lifting" with XPCOM, but they primarily use XUL/XBL and Javascript to build out the UI. There's no reason why this strategy won't work for Pascal.

The VCL wrapped Win32, written in C/C++, IMO there's no reason to believe that pascal wrappers could be built to get at XPCOM and integrate with UI's built with XUL/XBL and Javascript.

Re: kilyx

But has a procedural interface which matches Pascal as well as C/C++. (since even C/C++ need modification due to the requirement to support stdcall).

XPCOM has an IDL, any language can bind to it.

IOW, the win32 API was designed to be lowlevel accessible to a procedural languages. Swing/AWT is not. It is an OOP Java implementation, and to access it at all, you need to be Java, or compile to Java, or use JNI. (comparable to COM)

Swing/AWT/SWT are simply going to be used for the IDE services, not necessarily the deployment UI. Most of the IDE services are already built out in Eclipse and NetBeans. What's required is a sufficiently good grammar and some additional integration points customized for Pascal. Building an entire IDE is **not** required.

XUL is dog slow, overly complex etc. The requirements for XUL (having an infinitely configurable application GUI wise, read: Mozilla) are not universal. It is possible that making an application that makes good/safe use of XUL is more complex by a magnitude.

And yet we see really great, usable applications being built with XUL. They are cross platform and perform "good enough" for mass consumption.

I'm not interested in XUL, Swing, or other way too highlevel widget sets. I already have doubts about QT.

I'm not sure what I can say about that. The VCL doesn't fall in the same category?

Keep in mind that the focus of FPC is to write applications alike Delphi,

Understood, however I think even Delphi is showing its age. IOW, Delphi compatibility is very important, but FPC already does different things that Delphi does (cross platform, 64bit, generics). There's no reason to be unnecessarily confined by the Delphi-way.

while e.g. Python GUI development seems to revolve about making as short and simple possible controlapplet like programs for Linux distro's. (and they change language every 2 years. Perl->Python-> Ruby)

Re: kilyx

Re: kilyx

IMO, there has never been an emphasis on GUIs in that community, however it's interesting to look at a product like Komodo. It's written in Python and uses....Mozilla. :)

So the main focus for FPC would be to build a suitable IDE to write/debug code with.

We already have that.

These IDE frameworks already have tons of hooks for syntax highlighting, intellisense/code insight, debugging, etc.

We already have that too, and the latter part is false. They don't have debugging, but use GDB, just like we do.

I think you're missing my point. If you're happy with Lazarus, that's great. Keep using it, keep developing it. I'll keep an open mind and track its progress, but I have strategic concerns.

I disagree. There are those that are innovating on top of an IDE core and do have distinctive products.

Sure. I don't say that is not possible, but not for a company the size of Borland/CG.

I disagree. Because of CodeGear's size, I think it's an even better strategy. If they were a big enough company with a lot of money it would be more of a viable strategy to build an IDE core.

As it is now, they have to not only compete with MS and IBM and Sun on the higher level features, but they must also build out all the infrastructure while doing so.

The Java products favour a Java solution, and also IBM created a huge buzz around Eclipse in the Java scene (and NetBeans is trying to do the same for Sun). But that is not universal.

A lot of VS.NET is written in C++, but is it not capable of supporting ..NET development? Eclipse and NetBeans support C++ development and a lot of scripting languages that have nothing to do with Java.

Re: kilyx

Re: kilyx

If that's not universal, it's getting very close, very fast.

Lazarus/FPC already has a critical mass. Moreover, they are all volunteers, and you can't simply reassign them. They'll be as reluctant as I am. What you call "Not invented here", they'll call "throwing away the work of years based on a marketing fad".

I'm not trying to "simply reassign" anyone; clearly I can only assign myself. I'm simply expressing some thoughts. If my ideas resonate with some people, great. If they don't, that's fine too.

Actually George Birbilis has suggested stuff similar to you on the Lazarus maillist, resulting in huge discussions, but had the same problem, "no time" or "no skill" to backup the claims.

Again, I'm not trying to make any "claims" or back them up. But perhaps I should contact George Birbilis. :)

--

Brian Moelk
Brain Endeavor LLC
bmoelk@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

.