

Re: Writing portable applications

Source: <http://unix.derkeiler.com/Newsgroups/comp.unix.programmer/2005-08/0849.html>

From: Mike Meyer (mwm_at_mired.org)

Date: 08/30/05

Date: Mon, 29 Aug 2005 21:53:48 -0400

Ulrich Hobelmann <u.hobelmann@web.de> writes:

> *Mike Meyer wrote:*
>> *I'm still waiting for an answer to that one – where's the Java toolkit*
>> *that handles full-featured GUIs as well as character cell*
>> *interfaces. Without that, you aren't doing the job that the web*
>> *technologies do.*
> *Where is the text-mode browser that would even run part of the web*
> *apps I use, like home-banking, all web forums, server configuration*
> *interfaces, etc.? I think we should leave both these questions*
> *open. (In fact, as to UIs using Java, blech! Haven't seen a really*
> *good one...)*

If your web apps are well-written, any of them should work. As previously stated, Sturgeon's law applies to the web, so chances are good they aren't well-written.

>>>>> *I just wonder who wrote fully standards compliant web browsers for*
>>>>> *those 90 platforms.*
>>>> *Nobody. I doubt there's a fully standards compliant web browser*
>>> *Nobody, huh? Then how could you run just ANY web application on those*
>>> *platforms?*
>> *The same way you write POSIX applications in the face of buggy*
>> *implementations – by working around the bugs in the working part of*
>> *the implementation, and using conditional code where that makes a*
>> *serious difference.*
> *But as soon as some user of platform 54 tries your website, she'll*
> *encounter some weird behavior without even knowing why. And maybe so*
> *will you, especially if you don't have that platform there for*
> *testing. I don't understand how this web thing changes anything...*

The only difference is that the user of Platform 54 has a chance to use your app. Sure, it may not work because that platforms bugs are radically different from the bugs in the platforms you tested on. Raising the possibility of your app working from "no way in hell" to "maybe" is significant.

> *With POSIX at least you have a real bug-report for the guy responsible*
> *for it. If a platform keeps being buggy, with no fixes coming, screw*

comp.unix.programmer: Re: Writing portable applications

- > *them. Every user will see that sooner or later, and these platforms*
- > *die. Even Windows is quite stable/reliable after 10+ years NT!*

You do? All your porters reliably give you bug reports? Can I have some?

- >>>> *available for *any* platform, much less any non-trivial collection of*
- >>>> *them. You write portable web applications to the standards, and design*
- >>>> *them to degrade gracefully. Then you go back and work around any new*
- >>> *Oh right, they degrade gracefully. So without Javascript or cookies*
- >>> *(the former is often not implemented) you get a HTML page with an*
- >>> *error notice -- if you're lucky.*
- >> *You left off the important part of what I had to say -- that the*
- >> *application be written by a moderately competent web author.*
- > *But if you can cater for all kinds of sub-platforms, then why not just*
- > *provide a CLI as well as those GUI interfaces, when we're duplicating*
- > *work to begin with? ;)*

Because I'd rather not duplicate the work.

- >>> *A server AND client for a simple protocol designed for its task*
- >>> *(i.e. not FTP for instance) can be implemented in much less work than*
- >>> *even designing even part of a web application backend that does that*
- >>> *kind of stuff.*
- >> *Well, if it that easy (and web applications are dead simple), it*
- >> *should be done fairly frequently. Care to provide an example?*
- >
- > *We have all the web standards, with various extensions over the*
- > *years. Some FTP clients even don't crash if they see that some server*
- > *doesn't yet support the extension from RFC XY1234\$!@. Then there's*
- > *tons of inter-application traffic in XML already, growing fast. Then*
- > *there are s-expressions (Lisp XML if you want). Then probably*
- > *thousands of ad-hoc line-based text protocols, but I don't know how*
- > *well they can be extended. There's CORBA. Most web standards are*
- > *simple, at least if you would subtract the weird stuff (and IMHO there*
- > *should be new versions of everything with the crap removed). XML is*
- > *somewhat simple, just hook libxml.*

These don't answer the question. Maybe because I didn't explain it fully. Do you have an example of an application that implements a simple protocol along with a client and server where HTTP+HTML were considered as an alternative, and rejected as "more difficult" than the path actually chosen?

- > *There's NNTP. There's RSS. There's Atom. The latter two emerged*
- > *quite painlessly, even though you could maybe use some website for*
- > *what they provide. But this way you have lots of clients for lots of*
- > *platforms already.*

NNTP predates HTTP. Atom (and I assume RSS) uses HTTP as a transport, so there's no new protocol involved -- just a new file format.

>>>> *You think you're done. A lot of developers think you can stop with
>>>> the
>>>> first one or two. You're all right for some applications. For others,
>>>> you're not. Personally, I like applications that run on all the
>>>> platforms I use – and your set doesn't cover all three of those
>>>> systems.
>>> Ok, I'd be interested to hear what those are. VMS, RiscOS, Mac OS 9...?
>> FreeBSD, OS X and a Palm Vx.
> Didn't I say, a GUI for the Mac, for X11, and Windows? That only
> leaves out the Palm. I heard they aren't too hard to program for,
> either. But I haven't heard of a really decent browser for pre-OS5
> PalmOS (not sure about OS5).*

There are number of different browsers available for the pre OS5 Palms. They're more than adequate for the tools I want to use. In particular, Palm.net was built on HTML. On the palm side, you "compiled" a set of HTML files to create an application. Requests for remote objects went through a proxy that compressed them to save bandwidth. Very strange stuff.

>>>> *If a system's scheduler, or select implementation sucks, though, I'd
>>>> complain to the vendor or simply abandon the platform for
>>>> another. Competition is good :)
>> Complaining to the vendor doesn't always get the bug fixed. And
>> refusing to support a platform isn't always an option. Sometimes, you
>> have to bite the bullet and work around the bug on that platform.
> Sure, but you can tell your customers that unfortunately their system
> vendor refuses to fix a bug and ask THEM to ask that vendor. Boy,
> will they consider another platform in the future, where bugs do get
> fixed ;)*

Yup. You and that platform vendor are no win the set of vendors that don't fix bugs. Personally, I'd rather provide a workaround and keep the customer.

>>>> *same thing applies to threads, except such code typically includes a
>>>> third option of not using threads at all. And so on.
>>> Well, who doesn't do threads after several years of POSIX IMHO can't
>>> be taken seriously. Ok, the BSDs didn't until recently, but those are
>>> volunteer projects.
>> Not all platforms are POSIX. If you're ok limiting your application
>> to
>> a small subset of the total number of platforms available, then
>> there's no advantage to using web technologies. Some of us aren't
>> satisfied with that, though.
> Sure. You have to look where your users are. Chances are that with
> obscure systems they can't use most web-apps either.*

Right. Chances are they can only use well-written ones. If you write those, your stuff will stand out for them.

comp.unix.programmer: Re: Writing portable applications

>> *Well, marking up text is a pretty poor way to describe a UI – but
>> anything that is going to replace web technologies has to have a
>> media-independent way to describe the UI. One of the things that made
>> the web take off early was that anyone with a text editor could create
>> web pages. I think that's an important property to keep – you want the
>> tools that people use to create applications be as portable/flexible
>> as the applications. Since most GUI's are written in some programming
>> language or another, and most programming languages are still flat
>> text, a GUI description as flat text exists for most GUIs, so this
>> requirement isn't a handicap.
> That's true, though I think the future of development lies in
> overcoming that program-code-as-text thing (NOT visual programming,
> just tool-based, structured). Smalltalk did it decades ago.*

Last time I looked at smalltalk, it still presented program code as text. So I think you need to clarify what you mean.

>> *You don't have to guess – finding examples of XUL isn't hard at all. I
>> think XML gets used in a lot of places where it isn't appropriate. One
>> of the few places where it is appropriate is where you want a file
>> format that lots of independent implementations are going to be
>> reading. This could well be one of those times.
> Maybe, but for applications that aren't predominantly concerned about
> text, I'd really rather use a structured data type (like
> s-expressions), not text markup like XML. For hypertext, XHTML is
> fine, though, if a bit verbose.*

I'd rather use something which has a formal mechanism for defining what legal documents are. XML provides DTDs. If you really want S-expressions, we could use an SGML DTD that let you write them, and get the best of both worlds.

> *[follow-up set to comp.unix.programmer]*

comp.lang.python (where I've been reading this) added to followups.

> *(I just noticed I replaced my sig with something web-related
> yesterday. This is pure coincidence :D)
> --
> My ideal for the future is to develop a filesystem remote interface
> (a la Plan 9) and then have it implemented across the Internet as
> the standard rather than HTML. That would be ultimate cool.*

Yeah. I once worked on implementing the Rexx command interface as an internet protocol. It doesn't have the coolness factor of Plan 9's filesystem command interfaces, but it's not far from it.

<mike

--

Mike Meyer <mwm@mired.org> <http://www.mired.org/home/mwm/>

comp.unix.programmer: Re: Writing portable applications

Independent WWW/Perforce/FreeBSD/Unix consultant, email for more information.