

Re: Writing portable applications

Source: <http://unix.derkeiler.com/Newsgroups/comp.unix.programmer/2005-08/0867.html>

From: Ulrich Hobelmann (u.hobelmann_at_web.de)

Date: 08/30/05

Date: Tue, 30 Aug 2005 13:02:27 +0200

Mike Meyer wrote:

- > *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.*

:)

- >> *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.*

Ok, there's a chance...

- >> *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*
- >> *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?*

No, I mean, when you compile something on POSIX and it doesn't work, you have a bug to report. Of course many vendors might ignore them, and often don't send you any feedback at all (unfortunately), probably because contact between the company's programmer and you is considered bad for some reason (I'd consider it good support).

- > *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?*

No, because I don't know who considered HTTP+HTML before going the other way. HTTP certainly is used increasingly (for XML transfers, WebDAV, RSS, ...), because it's quite simple widely implemented. But as I said, there's RSS as a new standard, there's DAV and others. They don't use HTML, because custom protocols can be easy but have advantages. I guess it differs. Some applications make a lot of sense in a hypertext context; others don't.

>> *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.*

Yes. I consider HTML a kind of protocol in that sense, though. It's a format, and it needs interpretation, too.

>> *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.*

Short-term definitely. Long-term I'd try to migrate the customer (depending on how much influence I have in their IT context).

>>> *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.*

Very good point. You give me back my faith in web-apps ;)

>> *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 have class and method browsers, while most Java and C code is still edited file-wise (though with options to hide methods except for their declaration header). I think it makes sense to ignore the file thing and just use browsers like that (kind of like having every C function in one file, but with the headers and visibility of functions managed by your IDE/build system, not by you by hand.

>>> *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.*

Well, sometimes, sure. But quite often I think people don't bother with DTDs anyway (and of course you could design a DTD format for sexps and write a validator; I'd even guess there exists at least one somewhere out there). Lots of customers will probably ask for XML, so that mostly the way to go.

--

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.

Ken Thompson