Subject: Re: Object epiphany: A new way of building widget applications
Posted by John-David T. Smith on Thu, 05 Apr 2001 03:03:38 GMT
View Forum Message <> Reply to Message

Mark Hadfield wrote:
>
> "JD Smith" <jdsmith@astro.cornell.edu> wrote in message
> news:3ACBA2EF.493F496F@astro.cornell.edu...
>> Martin Schultz wrote:
>>>
>>> With almost a week delay, I finally get around to release the first
>>> version of a new class of IDL objects: the MGS_GUIObject hierarchy.
>>
>> I think it only fair to let people know that I tend to shy away from
>> distributed code with people's initials in the name. I know, it sounds
>> stupid, but I'm not sure I'm the only one. It seems to be a reasonably
>> common practice here (Craig, you listening?), but one which I think
>> might be best to avoid, for the following reasons:
>
> As one of the pioneers of this trend (he says modestly) may I present the
> opposing viewpoint:
>
> It's namespace management, pure and simple. It's desirable because IDL lacks
> built-in facilities.
>> And the way I think
>> about it, since IDL doesn't do any shadow checking (but cf. idlwave!),
>> the *best* routine with a given generic name will rise to the top.
>
> The one that rises to the top is somewhat unpredictable. (Well, strictly
> speaking it's predictable because yuu can control your PATH, though I have
> noticed recently that Windows 2000 expands path entries preceded by + in
> *reverse* alphabetical order, which caused me some grief.) The thing is, I
> don't remember exactly what is where on my PATH and I don't like relying on
> the search order. I have been bitten by duplicated routine names a number of
> times: CALDAT and CREATE_STRUCT are two I can remember.

I of course am very sensitive to this notion, which is why Carsten and I
developed an effective method for dealing with it in IDLWAVE. But in
any case, I was merely speaking metaphorically. If I write a routine
called "stack", and you write a routine called "stack", one or the other
will probably come into dominant usage. Is this ideal? No. Should we
attempt to relieve namespace collision by thinking ahead? Certainly.

>> 4. The author(s) can always be found in a proper documentation header.
>
> Sure, but it's not about claiming ownership, it's about namespace
> management.
>

> But hey, there's room for all points of view. If you don't prepend your
> initials and I do, then our routine names will never clash.
>
> Is there any other MGH out there?

If you need a prefix to differentiate your namespace, then by all means, chooose one. I was just arguing against that prefix being your initials. Here is an decent argument, simultaneously *for* namespace management, and *against* using your initials:

http://tiny-tools.sourceforge.net/emacs-code-body.html#about _lisp_symbol_naming

It's for lisp, but the same arguments apply. It's also pretty simplisitic, but the basic tone captures it I think. So, for your example, suppose you have a stack class which is fairly general. Why not super_stack, or fast_stack, or objStacker, etc? Yes, IDL started this whole ball rolling with their IDL_Blah series of classes, but I guess I just feel like a more open approach is available to us here.

If I were a company, JD, Inc., I would give my products a strong brand identity: JDI_Widget.pro. I'm not a company, and for this I'm glad. I don't make money from the things I contribute, nor do I guarantee their utility. If they solve your problem, great. If you rip them into pieces to something altogether different with them, great.

I'm not saying you *shouldn't* brand your contributions in the same way, but just pointing out a (perhaps not universal) connotation that branding engenders.

JD