
Subject: Object widget?

Posted by [Russell Ryan](#) on Fri, 22 Jun 2012 22:38:08 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi gang, I have a conceptual question. I've been developing a GUI to do some processing and modeling of astronomical images for a while now, which has gotten too "spaghettified" for my taste and so it's time for an overhaul. Since I'm about to rewrite ~10,000 lines of code to clean up all sorts of "features", I wanted to know what the virtues of writing this code as an object?

Thanks!

Russell

Subject: Re: Object widget?

Posted by [David Fanning](#) on Mon, 25 Jun 2012 17:58:41 GMT

[View Forum Message](#) <> [Reply to Message](#)

rryan@stsci.edu writes:

> I am very familiar with the Fanning's library and
> website. But I was hoping someone could describe the
> virtues of the OOP for widgets/GUIs. I'm rewriting the
> code anyway, and so thought to do it in OOP --- provided
> the reasons are better than simple intellectual curiosity.

In a word, objects are clever and widgets are dumb. Widgets have an extremely basic interface (SET_VALUE and GET_VALUE). Object widgets can be interacted with in any number of clever and creative ways. Plus, an added bonus, since they are so clever, they become idea-generating machines. The biggest problem you have with object-widgets is resisting the temptation to make them TOO clever! You can do too many things with them and you will find that you have to resist tinkering with them and just get the basic job done and move on. Otherwise, you will end up a geek instead of a scientist.

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Object widget?

Posted by [Michael Galloy](#) on Mon, 25 Jun 2012 18:42:55 GMT

[View Forum Message](#) <> [Reply to Message](#)

On 6/25/12 11:49 AM, rryan@stsci.edu wrote:

> I am very familiar with the Fanning's library and website. But I was hoping someone could describe the virtues of the OOP for widgets/GUIs. I'm rewriting the code anyway, and so thought to do it in OOP --- provided the reasons are better than simple intellectual curiosity.

Object-widgets have all the advantages of objects in general. In particular, encapsulation and a documented interface are important features. With a standard widget program, you basically have to give away your tlb widget identifier for another routine to interact with you. At which point, the interacting program can do anything they want to you. But an object-widget can publish an interface of actions it can take, properties that can be examined, etc. This makes working with object-widgets much easier and the code that interacts with them much easier to maintain.

Mike

--

Michael Galloy

www.michaelgalloy.com

Modern IDL: A Guide to IDL Programming (<http://modernidl.idldev.com>)

Research Mathematician

Tech-X Corporation

Subject: Re: Object widget?

Posted by [Russell Ryan](#) on Tue, 26 Jun 2012 00:29:13 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Monday, June 25, 2012 2:42:55 PM UTC-4, Mike Galloy wrote:

> On 6/25/12 11:49 AM, rryan@stsci.edu wrote:

>> I am very familiar with the Fanning's library and website. But I was hoping someone could describe the virtues of the OOP for widgets/GUIs. I'm rewriting the code anyway, and so thought to do it in OOP --- provided the reasons are better than simple intellectual curiosity.

>

> Object-widgets have all the advantages of objects in general. In
> particular, encapsulation and a documented interface are important
> features. With a standard widget program, you basically have to give
> away your tlb widget identifier for another routine to interact with
> you. At which point, the interacting program can do anything they want
> to you. But an object-widget can publish an interface of actions it can
> take, properties that can be examined, etc. This makes working with
> object-widgets much easier and the code that interacts with them much
> easier to maintain.

>

- > Mike
- > --
- > Michael Galloy
- > www.michaelgalloy.com
- > Modern IDL: A Guide to IDL Programming (<http://modernidl.idldev.com>)
- > Research Mathematician
- > Tech-X Corporation

Mike, David,

Thank you, that was what I was looking for. In fact, I think you two may have read my mind. The "main" GUI interacts with 4-6 "subGUIs" which do some other processing and manipulations. And, getting the data to pass back and forth between all the GUIs efficiently was a hassle. Finally, as the code progressed and new bits were added, tweaked, or removed, maintaining the connectivity was becoming a nightmare....

Sounds like the objects are the way to go, but I'm not too comfortable with developing my own. All the graphics handling is done with the object-graphics, so I'm familiar with the basics.

Thanks again fellas!

R
