Subject: Re: IDL objects (not object graphics) tutorial?
Posted by b_gom on Thu, 24 Nov 2005 09:04:25 GMT
View Forum Message <> Reply to Message

Hi Richard,

Not knowing what your background is, I'll aim low..

Richard G. French wrote:

> I've scoured the web in vain
> looking for a simple tutorial on how and when to use objects in IDL. I've
> found a few generic tutorials praising the virtues of object-oriented
> programming, but almost none of the examples give me any sense of why one
> would go to the trouble.

'How' is covered reasonably well in the IDL manuals. 'When' is another question altogether.

> What I am looking for is something with a simple application or two in which
> it is both clear why using objects is superior AND which explains what is
> meant by self and methods and classes. Without some specific examples to
> look at, I am having a hard time making sense of the nomenclature or of the
> value of the approach.

Let me try to summarize in a simple-mined, 1:00 AM kind of way:

Think of what problem you want to solve. If it is something like 'calculate such-and-such' or 'perform operation X on an array', then you probably want a simple function of procedure. If you can think of your problem as if it were some sort of abstracted machine or 'object' (like a kitchen appliance or a tree or something), then you probably want to make an Object.

Most IDL widget applications fall into the latter category. Although they can almost always be written using non-OOP techniques, there are often times when the little extra effort to write them as objects pays off. It's hard to actually get a feel for this without trying both techniques first.

Here are two excercises that should illustrate some advantages of OOP:

Q) write a program that acts like a stack of 'things', where a thing is any IDL variable or type. The program should allow you to add things to the stack, pop them off the stack, tell you how many things are on the stack, etc. Also you need to be able to have multiple different stacks

of things all co-existing at the same time.

A) Look at the 'linkedlist' object on David Fannings webpage for an object-oriented solution. If you come up with a non-object-oriented solution, let me know.

Q) write a program that acts like a dynamic plot of some data. You need to be able to add lines to the plot, remove them, zoom, change colors, annotate, etc, and all interactively at run-time. Also, your program needs to be simple enough so that other people can call it from their program with a few simple lines of code, and have serveral plots open at one time.

A) Try the non-OOP approach for 15 minutes, then look at David's site. There's probably an object there that has most of what you need, and in 15 minutes you can probably add on a few custom methods to extend the functionality to do the rest. Chances are, after a few years, you'll still be adding features to your object and it will do just about everything you can think of. But, here's the key: after all these changes, the other programs you wrote that call the object years ago will still work, and you or others will still be able to extend the object further or adapt it for other uses.

> Someone must be out there just waiting to get rich writing a book on this
> topic.

Back in 1999, a good general introduction to OOP was being written up.
It's now available here:
http://mindview.net/Books/TICPP/ThinkingInCPP2e.html
This is for C++, but the general principles apply to IDL.

Brad
