Subject: Re: How Object-oriented?
Posted by davidf on Thu, 22 May 1997 07:00:00 GMT
View Forum Message <> Reply to Message

David Ritscher <david.ritscher@zibmt.uni-ulm.de> writes:

> Is anyone ready yet to comment on how object- oriented IDL 5.0 is?

I am *not* an expert on OOP. My only qualification here is that reticence is not usually associated with my name. Please keep that in mind.

And my remarks are going to be colored by the fact that I was up until the wee hours of the morning trying to do the very simplest thing in IDL 5.0: display a surface with some axes on it. I have spent the better part of three days on this, so far without success, so I am feeling peevish about OOP just this moment. So please keep that in mind.

And finally, to give RSI every benefit of the doubt, I am still working with the beta version of the software, which everyone expects to be buggy. So *please* keep that in mind. :-)

> OO has become a big buzz word, so it has become important to at least
> give lip service to this concept. Can anyone comment on IDL 5.0?
> Is it more on the lip-service level, or are the changes significant
> enough that one can write real OO software?

I believe that RSI's commitment to OOP is more than lip service. The pointer implementation is just outstanding and I like very much the way they have implemented objects. I think the object structures are straightforward. I like the way they can be automatically initialized. I especially appreciate the almost complete flexibility you have in writing methods for your objects. And the elegance and simplicity of the Data Miner object implementation may be the best thing RSI has ever done with respect to building easy-to-use software.

I think, in short, that yes, the changes are significant enough that you will be able to begin to write real OO software. You only have to work with objects for a few minutes to start getting all kinds of ideas for powerful programs that could be written with them. I think, eventually, that objects will have as big an impact on IDL programs as widgets did when they were first introduced.

But I can remember when widgets were first introduced and how frustrated I got at times. The documentation was sketchy

at best, often wrong or misleading. The software was buggy. Things changed from one version of IDL to another, usually without warning, etc. (To give RSI credit, they usually changed for the better. I just sometimes didn't want to hear it after I had spent many days modifying my programs to work around some bug.)

I am in that frustrated mode right now after trying to write what I consider to be the very simplest graphics displays with the new object graphics routines. How in the world can something touted to be the very latest in programming innovation be so infernally hard to program!

I want to do to two simple things: (1) Display a surface with axes, and (2) Change the size of the axes labeling to something other than the grotesquely large default size. (I was actually embarrassed for RSI when I first saw the plots that came up in the Insight demo on my Windows NT machine.)

I expected to have to do more low-level programming with the object graphics system, especially with the first release of the software. But I did not expect to have to do so much low-level programming without adequate documentation. Last night I was reduced to the kind of programming I hate: making almost random changes in my programs in the futile hope that I might see a pattern in the results that gave me a clue as to what in the world was going on.

Just to give an example. I cannot figure out how to put axes on my surface so that it looks like a normal surface plot. (Yes, I know there are examples. I've run them. They work fine when the data is DIST(40), but they don't work at all if I try to display *real* data!) There is a LOCATION keyword in the Axis object that would seems to be what I need. Here is the explanation of that keyword:

> "Set this keyword to a two- or three-element vector of the form [x, y] or [x, y, z] to specify the coordinate through which the axis should pass. The default is [0, 0, 0]."

"The coordinate through which the axis should pass!?" What can this possibly mean? I can imagine an X axis having to pass through a point in the YZ plane, but beyond this I am stumped. And in any case, two hours worth of plugging all kinds of values into this keyword led to no insights. (A pun.)

And although I can now change the size of an axis title,

I have still discovered no way to change the size of the
axes annotation, beyond specifying each individual
annotation separately, which cannot possibly be what
RSI has in mind.

So, bottom line? I think RSI is moving in the right direction
with their OOP ideas. I think eventually it will revolutionize
the way IDL programs are written. But the current implementation
of object graphics (in my beta version, please remember) is
not ready for prime time.

Let's just say I'm glad I'm not an RSI technical support
engineer right now!

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
Customizable IDL Programming Courses
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com