Subject: Re: Interactive Objects, Was: Simple GUI question
Posted by Paul Van Delst[1] on Wed, 23 Apr 2003 14:45:30 GMT
View Forum Message <> Reply to Message

"Richard G. French" wrote:
>
> On 4/21/03 10:47 AM, in article MPG.190db260977ddbea989b5c@news.frii.com,
> "David Fanning" <david@dfanning.com> wrote:
>
>> I'm not exactly sure why people are afraid of objects.
>> Certainly the basics of object programming can be taught
>> in half a day. I make sure I include at least this much time
>> in any IDL programming class I teach.
>>
>
> I have used IDL for a long time and I have not yet taken the time to learn
> how to use objects in any language. I've gotten as far as using an
> occasional widget or two, and in some cases a rather complex set of compound
> widgets, but not objects. Nearly all of my programming involves either
> scientific image processing or computations, and the sequence of things is
> typically as follows:

[snip reasons he uses IDL...mostly same as mine btw]

> To follow David's analogy, I sometimes think that my widget programming
> style ends up being a design for an all-in-one blender, buzz saw, and CD-ROM
> burner, without doing any of them well. Object programming leaves me at a
> loss in terms of these fundamental design principles. The usual thing that I
> see is an example that says: "Start with an object that does one little
> thing, such as squaring a number or an array." But then I end up with about
> 100 lines of code which, in the end, just do what I can already do in one
> line of IDL - square a number or an array. What I need to see is an example
> of simple objects that can do something useful in IDL that I can't already
> do in a few lines of code. That might help me get over the hump so that I
> could join David's bandwagon.

My personal opinion is that it depends on what you're doing. You example above is a good one - using objects for every little thing leads to unimaginable code bloat (even for someone as programming-ly verbose as I am.) I've experimented with using objects in IDL and found them to be totally UNuseful for what I use IDL for -- mucking about with data, prototyping algorithms, and producing eye-candy for papers/proposals. I'm not knocking object oriented coding at all -- I try to use OO coding principles in my Fortran95 code as much as possible and it has helped enormously with code reuse and ease of use; but I also have 1000's more lines of code.

My wish for IDL is to produce more complete and easy-to-use display tools as part of the package. I was recently on a field trip where I had only matlab available to inspect the data and I was impressed with a simple little thing like how the equivalent of the plot command worked - you get a window where you can zoom in/out, annotate, etc -- all there right at your mouse/fingertips. I couldn't help but wonder why IDL doesn't have something like that (I won't mention Insight here....). I am aware there are plenty of user-produced things like that (I clobbered together my own a while back). Given what I use IDL for, and having seen what other data analysis packages (not restricted to matlab) can do, the price paid for an IDL license makes me hope that the introduction of objects leads to the production of some really powerful, fast, integrated, and easy to use tools in IDL in the future.

> I should be working on my NASA proposal....

I should be working on my NOAA report....

paulv

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC
Ph: (301)763-8000 x7748
Fax:(301)763-8545