Subject: Re: Yet another object graphics question
Posted by Paul Van Delst[1] on Fri, 25 Feb 2005 22:16:21 GMT
View Forum Message <> Reply to Message

Michael Wallace wrote:

> Hey guys, thanks for all the responses. It really helps my thinking
> about things. I guess I'm now getting to the point in my IDL coding
> that the issues of more importance are not how something is done in IDL,
> but rather what are the best patterns and solutions to use when working
> with IDL, especially in the land of objects. I'm quite comfortable with
> objects in other languages (Java/C++), but I'm still trying to wrap my
> head around the IDL way. As such, some of the familiar ways of
> attacking problems in Java/C++ don't map too well to IDL and vice versa.
> One of these days I'll have to write my book on how to learn IDL if
> you've programmed in a real* programming language before. ;-)
>
> * FORTRAN doesn't count :-)

Wha...?!?!? The cheek! <insert arm flailing and much huffing and puffing>

FORTRAN (i.e. the 1977 and earlier standard) maybe. But Fortran95 can teach you a lot. In particular, that pointers are mostly redundant in a well designed language. :o) Linked lists, queues, trees etc notwithstanding of course (oh, and I guess they're good for array aliasing too. harumph.) And, with the approval of the Fortran2003 standard, the language has the usual complement of OOP stuff (for folks that like that sort of thing). Apart from the polymorphism stuff in f2003, I hang out for the PROTECTED attribute (as opposed to just the current PUBLIC/PRIVATE ones), allocatable components of derived types (aka structures. Now we have to use stoopid pointers in structures to mimic it), and, *FINALLY*, stream I/O. Soon, gone will be the days when we'll need to use the /F77_UNFORMATTED keyword in OPEN statements in IDL.

Woohoo!

I would suggest that Java/C++ are not good languages for learning how to design software. That should be language independent. (That's my dig. :o)

cheers,

paulv

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC