Subject: Object Programming in IDL
Posted by graham_wilson_1234 on Tue, 21 May 2002 12:26:38 GMT
View Forum Message <> Reply to Message

Just to appease Craig, I have started a new thread so I can avoid putting my comments after David's 'gosh golly' post ;) I am interested in hearing others comments...

The first point that we should all be very clear on is that IDL is _NOT_ a particularly good example of an object oriented language. You can certainly emulate OOP concepts using IDL's objects and a select few functions/proceedures but if often defeats the purpose of the OOP style. When someone mentions IDL objects, it is universally assumed that they really mean 'object graphics' which leads directly to point number 2; Object oriented programming != object graphics. Unfortunately, it is very difficult to dispel this myth using IDL because of point number 1.

With regard to writing object oriented code in IDL we are all rather stuck until RSI implements a more complete feature set. I generally define polymorphism it as the ability to process objects differently depending on their data type or class. In this respect, the lack of operator overloading is an example where IDL fails to offer the full OOP tool set. Yes, you can overload methods, but operators should be no different. To compensate for this missing functionality one can write functions and/or procedures but this better described as an overlay and you must rely on a naming convention or a path precidence to avoid conflicts. Personally, I'd like to see true polymorphism (with overloading) and public/private methods sooner rather than later (is anyone at RSI listening?).

A good technical book describing the merits of using objects in data analysis is "Programming with Data: A Guide to the S Language" (ISBN: 0-387-98503-4). The concepts described are specific to S-Plus but can be adapted to any OOP language. While they may seem abstract at first, they are very powerful way of manipulating and modelling data. A free alternative to S-Plus is R (www.r-project.com).

For what it is worth, Matlab has a slightly more complete implementation of OOP. The one glaring (and annoying) feature missing from Matlab, however, is the absence of pointers and therefore dynamic structures/sizing. This, of course, is a grip for a different newsgroup...

I lurk therefore I am.
Graham