Subject: Re: rewriting from IDL to PVwave
Posted by grunes on Wed, 11 Mar 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Ebeth Jones wrote:
> ick, this doesn't sound good. However, I'm not really worried about the
> GUI part, the code I have looks like it has just one program that takes
> care of all the interface stuff, and I think I can rewrite this because
> I've used Motif before. What I'm really worried about are all the
> number
> crunching routines that I have that are written as procedures - they do
> lots of spatial processing and such, and they look like they are just
> number crunchers without any specific graphics calls. I would prefer
> not to rewrite these if I can get away with it. Are there any things
> that I should be on the lookout for between IDL and PVwave in these that
> would make the thing hang up if I tried to just rewrite the GUI and
> stuff the *.pro number crunching code behind it?

I often try to write stuff to work in both. It is somewhat of a pain.
Number crunching isn't too bad--except for a few subtleties like the
data types returned by abs, and the shapes of some sub-arrays with
1 element in one of the dimensions, and the absence in PV-WAVE of double
precision complex (unless they have added it recently), they are quite
similar. I often use REFORM() to get over the shape problems.

There are also some interpolation and image reading functions that are
present
in one and not the other, some functions that have extra keywords in one
package,
and so forth.

Image display and widgets are the parts that are really dificult to make
work
in both.

What I suggest you do if some things won't compile or run because
something
is missing in one is to get the free trial version of IDL, start it up
and
type ? to bring up help, then search for the function or procedure in
question.
This will tell you what it was supposed to do. Anyway, for some
purposes
I find the IDL online help easier to read, especially if I don't know
what
some feature is called, or what sub-package it is a part of, though that
statement
is based on a very old version of PV-WAVE, and a recent version of IDL.

If you are lucky, and the software runs in less than the 10 minutes the free trial version allows (and you get rid of all file output), you can also cross check final and intermediate results between the two packages.

IDL has recently tried to incorporate "object oriented programming". If the code uses any of that, good luck.

Personally I've been seriously considering writing an IDL/WAVE-to-generic-Fortran/C converter for the number crunching component of IDL/WAVE, even though it would be quite hard to make perfect, and probably require some user inputs. Probably not enough interest out there to make it worth the considerable effort involved.

If the problem turns out to be dificult, maybe you should buy the other package too!
Lockheed-Martin isn't quite poor and starving.

As with most computer language translation, the best converter is an experienced programmer.