Subject: Re: IDL alternatives?
Posted by cavanaug on Wed, 31 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

In responding to an earlier post of mine, Tim Patterson wrote :

> I wonder if he was referring not to the actual language itself, but to the
> lack of development tools for IDL? There is no debugger for example. No
> integrated environemnt. No way to compile stand alone code. Perhaps
> this is what he meant by the programmer interface? (It's how I
> interpreted it anyway).
>
> As somebody who has often had to adapt and update other people's IDL/PV-WAVE
> code, I know how difficult it is to write easily maintainable code in these
> languages. And there are no tools for this purpose. Even the emacs IDL
> mode is not officially supported. This can be a real probelm when
> trying to develop and debug large systems written in IDL/PV-WAVE.

I will grant you that lack of an executable creator tool is a major deficit, but it is my opinion that of all the other deficits you mentioned, none are problems for competent, thorough programmers and engineers. Maintaining poorly written code is a problem in every language, and nothing (again in my opinion) specific to IDL makes this problem worse. And as far as a lack of an integrated environment, this is almost a given in the UNIX world. Also, have you ever tried debugging Fortran 90 code? I know of no robust UNIX Fortran 90 debugger. My co-workers and I have no way of looking inside our structures, following our pointers, etc., yet we still have developed tons of good scientific research programs. And what about debugging LISP code? Granted, I have not programmed in LISP for about 3 years, but back then we did not have a LISP debugger available to us. We learned that a good design is the best debugger. And not knowing PV-WAVE (though I wish I could try it out for a while), there is no better way (again in my opinion) than using IDL to create event-driven, point-and-click interfaces to our large data pools. IDL is also the best tool I've ever used for fast data-visualization prototyping. I have used Fortran and NCAR Graphics, but what took me a day with those languages takes me only a few hours with IDL.

One last point : as far as all the "How come IDL wont work with X machine, running Y system, when I try to do Z" questions (which I do also post), the volume of these posts is in my opinion comparable with the other comp.lang groups that I read. And until these language and compiler companies stop with the develop-and-screw mentality, these sorts of posts will continue, ad nauseum. (I would not include GNU, NCSA or most other freeware providers in the above group)

Sorry about my long diatribe, but I'm still not convinced that IDL is the black sheep of the computer language family.

Charles
--

Charles Cavanaugh | "Words are very unnecessary, they can only do harm"
cavanaug@ncar.ucar.edu | - Depeche Mode
NCAR Boulder, CO, USA | "Facts all come with points of view"
My opinions | - Talking Heads