Subject: Object-oriented IDL?
Posted by Ray Osborn on Mon, 26 Feb 1996 08:00:00 GMT
View Forum Message <> Reply to Message

I have only a limited acquaintance with IDL and am not a regular follower of this group, so forgive me if this has been discussed before. I've checked that it's not discussed in the FAQ. While I'm very impressed with many of the features of IDL (and presumably PV-Wave), and shall probably be using it a good deal in the near future, I have been put off by what I believe is a fundamental limitation in its abilities to handle composite data structures. I would be interested to hear from more experienced IDL users, and perhaps the company itself, if they think that this is a genuine limitation and whether it's something that can/will be addressed in the future.

IDL is an array-based language, but when I perform data analyses, I don't deal with arrays but rather with spectra. These spectra comprise multi-dimensional arrays grouped with other arrays defining dimension scales, data errors, spectrum titles, axis labels, sample temperature etc. I gather that in IDL, I can group all these arrays and other variables into a record structure, but if I want to add two such structures, I would have to do all the book-keeping by hand i.e. check that the x-arrays are compatible, add the y-arrays, propagate the errors in quadrature and copy all the labels and other variables over. Doing this once or twice would be no problem but for more complex analyses, I am liable not to bother, and drop, for example, error-propagation to keep it manageable.

If IDL were an object-based language, rather than an array-based one, I could overload the "+" operator to perform all these operations in a consistent fashion. As I understand it (and I'm not an OOP expert), operator overloading just involves defining a standard set of procedures that are invoked when performing mathematical operations on these spectrum "objects". Once written, I can perform these operations much more transparently with far fewer lines of IDL code knowing that, for example, the errors are properly handled with each operation. Incidentally, data plotting could also be simplified since the independent axes and their labels could be carried around with the spectrum, and would not have to be specified in the plot command.

I am not suggesting that IDL should have the complete flexibility of C++ or other object-oriented languages, but just defining a limited set of classes could, I think, enormously enhance the power of IDL, at least for the sorts of data analysis that I perform. I would be interested if other IDL users agree with this, or whether they know of other ways of getting what I want within the current version. If not, would it be possible for IDL to be developed along these lines without being rebuilt from scratch?

--
R. Osborn Tel: +1 (708) 252-9011
Materials Science Division Fax: +1 (708) 252-7777
Argonne National Laboratory E-mail: ROsborn@anl.gov
Argonne, IL 60439-4845