Subject: Re: How far is OO implemented in IDL?
Posted by mvukovic on Thu, 06 Sep 2001 18:16:03 GMT
View Forum Message <> Reply to Message

Paul van Delst <paul.vandelst@noaa.gov> wrote in message
news:<3B9690F0.94C40523@noaa.gov>...
> Mirko Vukovic wrote:
>>
>> David Fanning <david@dfanning.com> wrote in message
news:<MPG.15ffce623d91ea6989694@news.frii.com>...
>>> Olaf Stetzer (olaf.stetzer@imk.fzk.de) writes:
>>>
>>>> I know that Object Oriented Programming is supported
>>>> in IDL but I wonder how far this concept is supported?
>>>
>>> Almost never as far as you would have hoped if
>>> you know much about real object-oriented programming.
>>> Remember, objects were graphed onto a language
>>> that was nearly 20 years old at the time.
>>>
>> stuff deleted...
>>
>> With all due respect to David and RSI, I am wondering how valid this
>> argument in defense of RSI is. So, shooting off the hip:
>>
>> It seems that a 50 year old language is going object these days
>> (fortran).
>>
>> Yes, RSI will have a problem improving IDL if they keep to the old
>> core. Are we supposed to keep working with that old and tired
>> language? Will RSI get new users with such an outdated product?
>> Unless RSI works actively on rejuveniting IDL, they will loose out.
>
> It seems to me that you are assuming that any perceived failings of the implementation of
> object stuff in IDL (e.g. lack of an operator overloading capability) is due to the core of IDL
> being old and dusty. I don't know that's true. The implication of David's email is that it is
> but David does not represent RSI....anymore at least (although he'd know more than anyone
else
> I'm sure.)
>
> Since you brought up Fortran, I'll go with that. I've got keyboard/mouse elbow so here's a
> summary:
>
> - F77 is similar to procedural IDL, e.g. v3.6 (?)
> - F90/F95 has some OO components (modules, private/public atrributes allowing data
> encapsulation etc..)
> - IDL now has a lot of OO components
> - F2K is slated to have even more OO stuff (polymorphism etc.)

> - Future version of IDL will also have more OO stuff. Who's to say operator overloading won't
> be include in some future IDL release?
>
> So I really don't see what the issue is here. Everything seems to be progressing along quite
> smoothly. If it's a case of "I don't like how IDL does implements this or that", well - I can
> find about 10 people who think the exact same thing about 3-5 or so other languages just by
> walking down the hall and sticking my head in every other cubicle. Wot I would GIVE to have
the
> Fortran equivalent of the IDL WHERE function (yes, I know f90/95 has a WHERE construct but
it
> doesn't return the indices for using on other stuff)
>
>> As a side-note, IDL was written in fortran 20 years ago, and
>> re-written in C some 10 years ago.
>
> And now it's probably (I don't know) written in C++. So?
>
> paulv

I went back through my original post, and it does have a bit of a sharp edge to it. Blunting it some, the point I was trying to make was that IDL being an old language is not an insurmountable obstacle to modernizing it. David's numbers and Paul's comment make sense to me too.

Thus, a summary of the discussion might be that the resources to implement OO fully have not been comitted yet, for various reasons, which mostly boil down to cost vs. benefit (as expected).

Should RSI follow the GNU GCC model, and let the core language go to the community, and meanwhile develop and collect royalty on all the add-ons (all the procedures and functions, graphics, etc.)? I am not certain this would work too well. How many programmers out there would spend time upgrading the language?

Or, we can develop IDL++, a pre-processor like C++ was originally to C. Now there is a ``weekend'' project :-) (or maybe more realistically, a plant shut-down project)

Mirko