Subject: Re: How far is OO implemented in IDL?
Posted by David Fanning on Wed, 05 Sep 2001 12:54:11 GMT
View Forum Message <> Reply to Message

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.

> I am thinking of operator overloading in special. My
> Idea is the following:
>
> There exists a struct called something like SQL-Timestamp.
> This struct holds int's/longint's for year, month, day,
> hour, minute, second, fraction, to hold a comlete date/time.
>
> If I make an object out of this struct and define the
> operators + and - it should be possible to add or substract
> seconds or days to/from a date stored in this struct. The
> functions are then hidden inside the object, I simply use
> the operators + and - for the operations. In C++ this would
> be possible, even for different kinds of variables but I don't
> know if this is possible in IDL too!

It is a good idea, but operator overloading in IDL
is not going to work like operator overloading in
C, that's for sure. You could add ADD and SUBTRACT
methods to your IDL object to do what you like, but
you will have to leave the actual meaning of operators
in IDL alone.
>
> My second thought would be the following: If I access a database
> with dataminer, fields of type DATETIME or TIMESTAMP are returned
> as the mentioned struct SQL-Timestamp which is defined by the system.
> Would it be possible to replace this struct by the aforementioned
> object?

I doubt it, but I don't know much about DataMiner. It would
be easy enough to write a method for the object that would
copy the structure into it, but whether it is worth it to you
depends on what plans you have for the object.

> So, this would be the most elegant way to provide more funcionality

> to date/time-handling for database access. In practise however I
> think that using the existing functions for date/time-conversion
> (I mean the functions in the JHU/APL/S1R IDL libraries)
> added by some functions for converting sqltimestamp to/from Julian
> Seconds (which I am currently writing) is the easier and faster
> approach...
>
> What do you think?

I think the latter method will work admirably for you. :-)

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: How far is OO implemented in IDL?
Posted by mvukovic on Wed, 05 Sep 2001 15:47:24 GMT
View Forum Message <> Reply to Message

Olaf Stetzer <olaf.stetzer@imk.fzk.de> wrote in message
news:<3B95D216.8F7BD6C1@imk.fzk.de>...
> Hello,
>
> I know that Object Oriented Programming is supported
> in IDL but I wonder how far this concept is supported?
>
> I am thinking of operator overloading in special. My
> Idea is the following:
>
stuff deleted

Operator overloading is not supported. Neither are friends, or
abstract objects.

As recently pointed out, you can cludge abstract objects by having the
INIT function return 0 and calling it from a derived object.

Mirko

Subject: Re: How far is OO implemented in IDL?
Posted by mvukovic on Wed, 05 Sep 2001 19:47:49 GMT
View Forum Message <> Reply to Message

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.

As a side-note, IDL was written in fortran 20 years ago, and re-written in C some 10 years ago.

feeling much better :-), and expecting corrections to the arguments above,

Mirko

Subject: Re: How far is OO implemented in IDL?
Posted by David Fanning on Wed, 05 Sep 2001 20:23:44 GMT
View Forum Message <> Reply to Message

Mirko Vukovic (mvukovic@taz.telusa.com) writes:

> David Fanning <david@dfanning.com> 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.
>
> As a side-note, IDL was written in fortran 20 years ago, and
> re-written in C some 10 years ago.
>
> feeling much better :-), and expecting corrections to the arguments
> above,

Before the spelling police get all over me again,
I just want you to know that I know the difference
between "graphed" and "grafted". But I was thinking
about the guys at RSI working out the details on object
implementation on graph paper and ... oh, never mind.
It was 5:30 AM and I hadn't had a cup of coffee yet. :-(

But in response to Mirko (and, heaven help me, I really
feel uncomfortable being in a position that appears to
be defending RSI) here are some numbers:

  FORTRAN Users: 1,345,493,398
  IDL Users: 45,384

If you had to pick a language to completely re-write
every 10 years, which would you choose? :-)

Cheers,

David

P.S. Let's just say I didn't spend too much time
researching the numbers above, but the IDL numbers

would have to be normalized downward when you
consider the number who would actually *pay* to
have someone work on the language re-write.

--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: How far is OO implemented in IDL?
Posted by Paul van Delst on Wed, 05 Sep 2001 20:54:08 GMT
View Forum Message <> Reply to Message

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

--
Paul van Delst Religious and cultural
CIMSS @ NOAA/NCEP purity is a fundamentalist
Ph: (301)763-8000 x7274 fantasy
Fax:(301)763-8545 V.S.Naipaul

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

Subject: Re: How far is OO implemented in IDL?
Posted by Randall Skelton on Fri, 07 Sep 2001 11:08:07 GMT
View Forum Message <> Reply to Message

On 6 Sep 2001, Mirko Vukovic wrote:

[big snip]
> 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?

Well... I for one think it would be great to have the ability to extend the IDL language. Slow progress is being made on IBM's data explorer (now OpenDX) which was released by IBM to the community. I will simply be happy when the interface for pointer/heap variable is released so I can properly build IDL objects in C and fully encapsulate the internal structure of my data!

On the subject of software models, lets all hope we don't see IDL `XP` and idl.net anytime soon ;)

Cheers,
Randall
