Subject: Re: IDL routines dependencies map maker
Posted by Michael Galloy on Mon, 12 Mar 2012 17:53:22 GMT
View Forum Message <> Reply to Message

On 3/12/12 9:26 AM, DavidPS wrote:
> Hello IDL-gurus,
>
> I've been from time ago interested to find which routines are more critical than others in a library. In my case, I work with a huge library, and I'm certain that some routines are critical for the whole system. So, I've been thinking how to get that info, which procedures call what...
>
> ...after searching around I've found nothing (maybe because I'm looking for the wrong names). So I've decided to create my own mapper program for IDL routines.
>
> I've done it using bash scripts (lots of sed, grep and awk) and surprisingly, though far from efficient, it works. It creates a graphivz file (dot), and it also tries to generate a png from it (this step fails sometimes).
>
> So, if you want to test it, here it is:
>   https://github.com/dpshelio/IDL-mapper
>
> follow the instructions and visualise the png (if it works) or use one of the tools in here:
>   http://www.graphviz.org/Resources.php
>
> ZGRViewer is quite nice (http://zvtm.sourceforge.net/zgrviewer.html)
> but Gephi(not listed above) is amazing!: http://gephi.org/
>
> if you feel adventurous and want to improve it! please, do! My idea is to run it on that huge library (>3540 files), but I would be mad if I try this script on there...
>
> If you are interested, this is (in short) how it works:
>   1) Search for any pro, function definition in all the files
>   2) Search for any call of these saved pro/functions in all the files
>   3) Generates the dot file using pygraphivz
>
> Of course, the script is quite convoluted, and I'm sure there's better ways to do it, but my brain is a bit dry after all this weekend, so I would really appreciate some help, comments and suggestions.
>
> Also, it's possible that there's a way to do so within IDL.. but I did not find it.
>
> Cheers,
> David

Cool! I have the tool chain running and get a graph for the sample code:

http://michaelgalloy.com/wp-content/uploads/2012/03/map.png

Running on GPULib didn't take long (maybe a minute or so) and produced
useful results:

http://michaelgalloy.com/wp-content/uploads/2012/03/gpulib-d ependencies.png

I think this is a very useful project and have long thought about
including something like this in IDLdoc.

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL, A Guide to Learning IDL: http://modernidl.idldev.com
Research Mathematician
Tech-X Corporation

Subject: Re: IDL routines dependencies map maker
Posted by DavidPS on Mon, 12 Mar 2012 22:29:55 GMT

View Forum Message <> Reply to Message

> Running on GPULib didn't take long (maybe a minute or so) and produced
> useful results:
>
>
> http://michaelgalloy.com/wp-content/uploads/2012/03/gpulib-d ependencies.png

I'm actually impressed... definitely I don't have enough memory in my computer to run it properly.

> I think this is a very useful project and have long thought about
> including something like this in IDLdoc.

Cool to know that I was not completely mad at trying something like this. I'm happy other people find it useful. A step in the future would be to make it interactive, so we could click one of the dots and open the file (or see the documentation).

However, I'm still not completely sure that I'm having into account all the possible options... I still find some weird connections... I will try to improve it.
An open issue... and which I probably won't do, is handling objects. I have no clue of how they work, and I'm not sure if my approach (or something similar) would work.

Good Monday night to everyone,
David

Subject: Re: IDL routines dependencies map maker
Posted by Michael Galloy on Mon, 12 Mar 2012 23:05:22 GMT
View Forum Message <> Reply to Message

On 3/12/12 4:29 PM, DavidPS wrote:
>
>> Running on GPULib didn't take long (maybe a minute or so) and produced
>> useful results:
>>
>>
>> http://michaelgalloy.com/wp-content/uploads/2012/03/gpulib-d ependencies.png
>
> I'm actually impressed... definitely I don't have enough memory in my computer to run it properly.
>
>
>> I think this is a very useful project and have long thought about
>> including something like this in IDLdoc.
>
> Cool to know that I was not completely mad at trying something like this. I'm happy other people find it useful. A step in the future would be to make it interactive, so we could click one of the dots and open the file (or see the documentation).
>
> However, I'm still not completely sure that I'm having into account all the possible options... I still find some weird connections... I will try to improve it.
> An open issue... and which I probably won't do, is handling objects. I have no clue of how they work, and I'm not sure if my approach (or something similar) would work.

In general, 100% accuracy in static analysis like this is going to be impossible for a dynamic language like IDL. But, a lot of useful information can still be derived from the code.

It is impossible to handle the general case for objects. A routine can call a method on an object without knowing, until runtime, what class and method they are actually calling.

Also, CALL_PROCEDURE, CALL_FUNCTION, CALL_METHOD, and EXECUTE also make the general case impossible to deal with. I would just ignore these cases.

Another problem case is determining the difference between function calls and array indexing (IDL has a difficult time with this too, which is why the "compile_opt strictarr" option is so useful).

IDLdoc has a :Uses: field which could be used to pass hints to the dependency finding algorithm in the difficult cases, e.g., when using EXECUTE you could just specify what routine you are calling in the :Uses: field.

Mike

--
Michael Galloy
www.michaelgalloy.com
Modern IDL, A Guide to Learning IDL: http://modernidl.idldev.com
Research Mathematician
Tech-X Corporation

Subject: Re: IDL routines dependencies map maker
Posted by DavidPS on Mon, 12 Mar 2012 23:48:40 GMT

View Forum Message <> Reply to Message

> In general, 100% accuracy in static analysis like this is going to be
> impossible for a dynamic language like IDL. But, a lot of useful
> information can still be derived from the code.

Really useful if the map.dot file is opened with gephi. You can change the size of the circles, and get some statistics straight away.

> It is impossible to handle the general case for objects. A routine can
> call a method on an object without knowing, until runtime, what class
> and method they are actually calling.

Ok, so then I completely forget about objects ;-)

> Also, CALL_PROCEDURE, CALL_FUNCTION, CALL_METHOD, and EXECUTE also make
> the general case impossible to deal with. I would just ignore these cases.

I'm not sure in the CALL_ ones, but I think my scripts will find the function being called from EXECUTE. Not the procedures because I've set to look at the starting of the line.

> Another problem case is determining the difference between function
> calls and array indexing (IDL has a difficult time with this too, which
> is why the "compile_opt strictarr" option is so useful).

In the way it works, it find first the definition of the functions, and then search if they are called. So, yes, it could be a "false" edge if there's a variable called with the same name than the function. I was not aware of that "compile_opt"... I still don't know how I would use it.. but at least I've always used () just for functions.

> IDLdoc has a :Uses: field which could be used to pass hints to the
> dependency finding algorithm in the difficult cases, e.g., when using
> EXECUTE you could just specify what routine you are calling in the
> :Uses: field.

Then everything would be a lot easier!! ;-)

David