Subject: Re: VTK + IDL
Posted by Karl Schultz on Wed, 25 Aug 2004 14:33:38 GMT
View Forum Message <> Reply to Message

"Hee Chun" <chun.42@osu.edu> wrote in message
news:da6f35c7.0408241737.7a23a499@posting.google.com...
> Hello,
> I learned from the previous post(from Mike) that:
> 'OpenGL doesn't have the scientific analysis capabilities of IDL and
> IDL
> doesn't have the graphics constructs of OpenGL'.
> But I need both analysis capabilities and graphics display for the
> medical image data sets.

I'll echo the other replies and say that IDL should have most of what you would need. Particularly with IDL 6.1, *most* of the OpenGL features are now exposed via Object Graphics.

> Is there any way to interface between VTK and IDL using DLM? If there
> is, Is it very hard to implement it?

I've never done it or have heard of anyone who has. I think that it should be possible because VTK exposes an API that can be used from a C/C++ program. So, you should be able to code a DLM that implements some set of IDL function/procedures that turn around and call VTK functions.

I have not done an IDL-VTK graphics comparison lately, but I can't imagine that VTK graphics capabilites are that different from IDL's in terms of pure graphics features. There may be some differences in higher-level visualization techniques and features. If I found myself wanting to call VTK from IDL, I would think that it might be for using some non-graphical analysis function in VTK that does not exist in IDL.

> What kinds of problems do I need
> to consider?

If I remember right, VTK exposes a C++ API. So, your DLM would have to be coded in C++. IDL calls DLM's using a C interface. No problem here, just something to keep in mind.

> Where is the good starting point at least to open the 3d
> object rendered by VTK on the draw window of IDL?

You would need two things for this -

1) A way to get an OS-level window ID for the window that IDL created in a WIDGET_DRAW widget.
2) A way to pass this window ID to VTK and tell it to use that instead of

creating its own.

I am pretty sure that (1) does not exist, and the VTK docs would indicate if (2) is possible.

I once accomplished something like this, but in a slightly different manner:

If you can get an external library like VTK to make OpenGL calls without making its own GL context the current context, then you can call the library during the execution of the Window's Draw method. You would do this by implementing a subclass of one of the object graphics objects, say IDLgrPolygon. Override that object's Draw method with code that calls your DLM that in turn calls VTK to draw the VTK object, say an isosurface or somesuch.

This approach is really tricky and NOT supported at all by RSI. You can see an example of it in the RSI user contrib library on the RSI website. I put an example there that calls a Volume graphics library (VGL).

But I think that this will be hard to make work with VTK because I doubt that VTK exposes an entry point that will just make blind OpenGL calls without making its own GL context current or otherwise somehow relying on its ownership of the window. I could be wrong about that - you might find a low-level "draw myself" method on some VTK primitive you are interested in that would normally be called by a VTK scene graph traverser.

Karl
