Subject: IDL & iTools used as post-processor for other commercial software
Posted by mvukovic on Thu, 28 Sep 2006 19:12:41 GMT
View Forum Message <> Reply to Message

Yesterday I saw a presentation about Chemkin software (by Reaction Design). This software was developed at Sandia, and then commercialized. And something looked familiar:

iTools were used to show the simulation results.

Some of the other widgets used in pre-processing also had an IDL feel, but I am not 100% certain.

So, there you have it. That's why IDL is pushing object graphics :-)

Mirko

Subject: Re: IDL & iTools used as post-processor for other commercial software
Posted by Benjamin Hornberger on Wed, 04 Oct 2006 16:26:28 GMT
View Forum Message <> Reply to Message

Ed Hyer wrote:
> Comparing Mirko and Bob's viewpoints does show a remarkable degree of
> convergence in what is possible with the two approaches. The idea that
> you might be able to meet the requirements of a peer-reviewed
> profession with an interactive tool is pretty impressive, honestly. The
> cornerstone of these requirements is that any graphic produced can be
> re-generated on-demand starting from the data in a relatively raw form,
> or using different data. Bob's scripts contain the entire process,
> which I think is probably pretty typical for scientific users. My own
> scripts generally start with loading data (often several hundred MB).
>
> Does iTools have a mechanism (or perhaps, could one be built in) to
> keep the code used to distill from the "original" input datasets down
> to the numbers for the plot? With this feature, you truly could go from
> having a script associated with every plot to having an iTool. This
> might actually be an improvement over a script-based process, although
> I don't know if it would get me to switch. Since everything I do
> graphically is 96% recycled code, the cost-benefit calculation is
> pretty tight.

What I usually do interactively in iTools is not manipulating the data but the appearance of the plot: Adjusting data range, line style and thickness, colors, logscale, add a legend and move it around (much easier than in direct graphics, unless you have it automated, which is only worth doing if you generate the same type of plot over and over), adding an annotation, etc. The latter I guess some people do in

Illustrator, in which case it's not exactly reproducible either.

So I can say that I can exactly reproduce the *data* if necessary --
which is important -- but maybe not the exact appearance of the graphic.
In fact, a nice thing about iTools is that the data is stored in the
tool (and can be exported from it). If you use the NAME keyword, the
name is associated with a given visualization in the tool, so I think
it's very well reproducible where the data comes from.

Plus, graphics for publications and presentations which have to be
reproducible exactly are only a fraction of what I need IDL graphics
for. Often I just need to, e.g., visualize a function, which for me is
faster in iTools than direct graphics in particular if I want to print
it out. If I don't print and don't expect to change any plot parameters,
I also use direct graphics since they are faster in that case.

I also agree that if one wants to create the same visualization from the
same kind of data over and over, a script using direct graphics is
superior, and I also do that. However, that's not interactive and we
mostly emphasized the usefullness of iTools for *interactive*, ad hoc,
use. iTools might even be able to replace direct graphics scripts for
this kind of application if people understand how to manipulate iTools
from the command line. Not so easy ...

Benjamin
