Subject: Re: Catalyst Library
Posted by David Fanning on Wed, 19 Nov 2008 22:45:02 GMT
View Forum Message <> Reply to Message

Giorgio writes:

> I am trying to implement a 1D plotting object to the catalyst
> library. The idea is to have a main object that sets up the
> coordinates for the plotting (x and y-ranges, fonts, background color
> and axis colors) and then different objects with the data that will do
> the plotting it self. Each one can have its own definition of a
> symbol, color, line, symbol size, line thickness and if the symbol is
> full or not. I know that it is ambitious and we'll see up to where I
> get since I am not doing this in a regular basis.

I have had a student working on this several hours a
week for the past couple of weeks. She just sent me
some code, but I haven't had time to review it yet.
But perhaps we can compare notes soon.

> Up to now, I've just
> defined the main class based on the catdataatom class. In doing this,
> I was surprised by the following: the new classes can only be
> inherited from classes with catatom as a superclass. I wonder myself
> why is so and which was the spirit behind this?

One of the problems with object programming is
cleaning up your objects properly. It is easy (too easy!)
to leak memory. Because we were using the notion of
a Catalyst program as an object container we thought to
take advantage of this for proper object cleanup. But,
it was not this simple. When you destroy a container, all
the objects in that container are destroyed. But sometimes
data objects were in *several* containers. If we destroy
a graphics window object that contains an image object,
we don't necessarily want that image object to be destroyed;
it might still be in use by some other part of the program.

So we decided we would reference count. When you add a data
object to a container, the container is added as a "parent"
the the data object's parent list. When you take an object
out of the container (or you destroy the container), we remove
that parent from the list. It is not until ALL the parents
are removed that the image object is actually destroyed.

This reference counting occurs at the CatAtom level.
Internal documentation occurs at the CatAtom level.
Message passing occurs at the CatAtom level. As well

as several other things that I am probably forgetting.

But, the bottom line is that every object has to inherit
the CatAtom object because much of what it means to be
a "catalyst" object is built into that single object.

On occasion, even I have thought maybe the restriction
is too strict. But I've always found (so far) a way to
accommodate it.

Cheers,

David

P.S. Thanks for the question. I have in mind to have
a separate section, eventually, on my web page with answers
to these kinds of questions. A kind of documentation that writes
itself as the need arises. :-)

--
David Fanning, Ph.D.
Coyote's Guide to IDL Programming (www.dfanning.com)
Sepore ma de ni thui. ("Perhaps thou speakest truth.")
