
Subject: Re: OOP Programming, Was: Something Else
Posted by [Reinhold Schaaf](#) on Fri, 06 Mar 1998 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

```
>> A third remark concerns event handling: It is not possible to define a
>> method of a class as the event handler of a widget.
>> As a consequence, one is forced to make the event handler a global
>> function. But global functions cannot access members of objects, so one
>> has to add methods to the class which would be unnecessary otherwise.
>
> I have grappled with this, too. Having all of a sudden become
> enamored with object programming, it is not much of a leap to
> believe that this is the way widget programs should work. Especially
> compound widgets. Like Reinhold, I've found imperfect, interim
> solutions.
>
>> Things could be ways simpler if global functions were allowed to be
>> event handling routines! I wonder whether this could be changed in
>> future versions of IDL.
>
> I haven't specifically discussed this with the folks at
> RSI (although I plan to), but I think OOP is probably in
> the same state that widgets were in when they were first
> introduced. That is to say, the first effort was pretty
> good, but nowhere near where they had to get to eventually.
```

And Mark Rivers wrote: I agree that this is somewhat inconvenient, but the workaround is really pretty easy. The trick I use is to stick the object itself in the uvalue of the top level base for this object. There is no need to keep the list of objects. Here is an example:

```
.*****
,
pro mca_display_event, event ; event processing for MCA application

    widget_control, event.top, get_uvalue = mca_display
    mca_display->event, event
end

.*****
,
pro mca_display::event, event ; event processing for MCA application

    case event.id of
```

.... Rest of event processing goes here.

```
.*****  
,
```

```
function mca_display::init, font_size=font_size, parent=parent
```

```
...
```

```
    base = self.widgets.base
```

```
    widget_control, base, set_uvalue=self
```

```
    xmanager, 'mca_display::init', base, event='mca_display_event', $
```

```
        cleanup = 'mca_display_cleanup', /no_block
```

```
...
```

Mark Rivers	(773) 702-2279 (office)
CARS	(773) 702-9951 (secretary)
Univ. of Chicago	(773) 702-5454 (FAX)
5640 S. Ellis Ave.	(708) 922-0499 (home)
Chicago, IL 60637	rivers@cars.uchicago.edu (e-mail)

or:

Argonne National Laboratory	(630) 252-0422 (office)
Building 434A	(630) 252-0405 (lab)
9700 South Cass Avenue	(630) 252-1713 (beamline)
Argonne, IL 60439	(630) 252-0443 (FAX)

=====

Mark,

I like your trick! Thanks.

I need to maintain a list of CFrame objects anyway, because this list enables me to render data automatically to all frames whenever the data is updated. I recently found out that my scheme of handling data and frames, in which the data is displayed, is a poor man's version of the document/view architecture of the Microsoft Foundation Class, which is part of Microsoft's Visual C++. My impression is that this architecture is a very sound foundation for building object oriented user interfaces (under any operating system, not only under Windows!), so it should be worth the effort to learn more about it.

Bye

Reinhold

--

***** *

Reinhold Schaaf
Ettighofferstr. 22
53123 Bonn

Tel.: 0228-625713

Email: schaaf@astro.uni-bonn.de

***** *