
Subject: Re: widgets and objects

Posted by [Reinhold Schaaf](#) on Thu, 13 Aug 1998 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

<HTML>

<TT>mirko_vukovic@notes.mrc.sony.com wrote:</TT>

<BLOCKQUOTE TYPE=CITE><TT>Now, the combination of widgets and objects has been mentioned but not</TT>

<TT>described by several folks. Can one define a widget to be an object, and the</TT>

<TT>events to be methods? Then self is naturally defined, and at least I do not</TT>

<TT>have to worry about accessing it. The code does not get much cleaner, but at</TT>

<TT>least one level of pointer access is eliminated. However, it is not clear to</TT>

<TT>me that the event handlers can be methods. In that sense, one needs the</TT>

<TT>event handler to call the method, yet another layer of complexity.</TT><TT></TT>

<P><TT>Ideally, when defining the widget, a special type of variable would be defined</TT>

<TT>that exists in the widget space and in all the event handlers. It would be</TT>

<TT>like a hidden common block, but would automatically exist in all the event</TT>

<TT>handlers, and there would be a separate instance of these widget variables</TT>

<TT>for separately created base widget (the biggest problem with common blocks).</TT><TT></TT>

<P><TT>Thus, a couple of hints to RSI. Allow the use of methods to be event handlers</TT>

<TT>and/or allow the creation of these special widget common variables.</TT><TT></TT>

<P><TT>mirko</TT><TT></TT>

<P><TT>-----== Posted via Deja News, The Leader in Internet Discussion
==-----</TT>

<TT>http://www.dejanews.com/rg_mkgrp.xp p;

Create Your Own Free Member Forum</TT></BLOCKQUOTE>

<TT> </TT>

<TT>I worked quite a bit in this sort of programming and came up with the following scheme, which was initialized by a remark of Mark Rivers from the University of Chicago:</TT><TT></TT>

[illegible][illegible]

<P><TT>;some checks deleted</TT><TT></TT>

[illegible]


```
<P><TT>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~> Events from base widgets</TT>  
<BR><TT>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~> 'WIDGET_BASE': $</TT>  
<BR><TT>&nbsp;&nbsp;&~> BEGIN</TT>  
<BR><TT> ~> bEventHandled =  
oFrame->OnResize(sEvent.id,  
sEvent.x, sEvent.y)</TT>  
<BR><TT> ~> oFrame->OnUpdate</TT>  
<BR><TT>&nbsp;&~> END</TT>  
<BR><TT>&nbsp;&~> 'WIDGET_KBRD_FOCUS': $</TT>  
<BR><TT>&nbsp;&~> CASE sEvent.enter OF</TT>  
<BR><TT> ~> 0: bEventHandled =  
oFrame->OnLooseKbrdFocus(sEvent.id)</TT>  
<BR><TT> ~> 1: bEventHandled =  
oFrame->OnGainKbrdFocus(sEvent.id)</TT>  
<BR><TT> ~> ELSE:</TT>  
<BR><TT>&nbsp;&~> ENDCASE</TT>  
<BR><TT>&nbsp;&~> 'WIDGET_KILL_REQUEST': $</TT>  
<BR><TT>&nbsp;&~> bEventHandled =  
oFrame->OnKillRequest(sEvent.id)</TT><TT></TT>
```

[illegible]

<P><TT>etc etc</TT>
<TT></TT> <TT></TT>

The class CFrame, which is used as an abstract base class for other classes, which implement concrete frames, provides only non-functional event-handlers:

```
<BR><TT></TT>&nbsp;     
<UL><TT>FUNCTION CFrame::OnResize, wID, iX, iY</TT>  
<BR><TT>&nbsp;    RETURN, 0B</TT>  
<BR><TT>END</TT><TT></TT>
```

```
<P><TT>FUNCTION CFrame::OnButtonPress, wID</TT>  
<BR><TT>&nbsp; bEventHandled = 0B</TT>  
<BR><TT>END</TT><TT></TT>
```

<P><TT>etc.</TT>

server.</TT>

<TT></TT> <TT></TT>

<P><TT>Best regards</TT><TT></TT>

<P><TT>Reinhold</TT>

<TT></TT> <TT></TT>

<P><TT>--</TT>

<TT> ***** * </TT>

<TT> Reinhold Schaaf</TT>

<TT> Ettighofferstr. 22</TT>

<TT> 53123 Bonn</TT>

<TT> Germany</TT><TT></TT>

<P><TT> Tel.: (49)-228-625713</TT>

<TT> Email: schaaf@astro.uni-bonn.de</TT>

<TT> ***** * </TT>

<TT> </TT></HTML>
