Subject: Re: A simple IDL canvas
Posted by Robert Barnett on Sun, 20 Mar 2005 22:39:58 GMT
View Forum Message <> Reply to Message

Hi Antonio,

I think that your application reflects how most of us feel about iTools. iTools is so complex and so feature driven, yet it doesn't seem to be extensible in the way that we would love it to be.

You are not reinventing the wheel, I haven't seen anything quite like your application yet. I'm hoping that people like you (and I) might be able to form OO libraries which embrace the concepts which *we* find important in the workplace.

I've worked on my own library which does some arranging and keeping a track of some kinds of graphics objects. However, my situation is very simple and doesn't require me to worry about event signalling. It's only really useful for certain medical applications.
http://www.zipworld.com.au/~retsil/idl/snaptools/doc/

There are several things that I would really like to see from a GUI based OO library.

* Something I can understand and comprehend in a reasonable timeframe
* The ability to only extend and use the functionality that I'm interested in, rather than having to worry about everything that the library does.
* Something that can be extended without ever having to modify the source code of the API.
* The ability to contribute and package additional features for the library in the same way that the library itself is implemented (a.k.a. the "Eat your own dogfood" concept)

I know that the ******** library addresses many of these issues, but I think that I am going to have difficulty costing, justifying and distributing that kind of solution.

I think I'll just have to put up with what RSI has provided and get over it :(

Antonio Santiago wrote:

> Hi,
>
> I'am working on an IDL canvas to draw lines, rectangles, ... (for the

> moment only this two types of item :) ). All this is object oriented.
> It is necessary for an application we are developing, I don't know if it
> was usefull from anybody more :(
> Well, if you know another IDL program (libraries) that does the same,
> please warn me to prevent me not reinvent the wheel :)
>
>
> For the moment you can create the canvas and create some items (lines or
> rectangles), also you can group items in "group items", that is, one
> group item is an "compound item" compossed by some other simple items.
>
> I attach an image example.
>
> The image shows the canvas with simple green line and a couple of simple
> rectangle items (blue line and grey filled rectangles).
>
> The items can receive, from now, button press/release and motion events
> and the user can specify which function, procedure or object method can
> handle the event (here I used the EventAction class I was created some
> time ago :) :
> http://asantiago.gentelibre.org/index.php/archives/2004/11/1
8/a-signal-mechanism-for-objects-in-idl).
>
>
> Based on the "group" class, I have created a more complex item called
> "product" (the orange rectangles). This class is a group item compossed
> by two rectangle items (one for black border and the other for the
> orange fill). Also I was configured the object to receive the events:
> button_press, button_release and motion on the "product" class methods
> and I can move thid elements arround the canvas.
>
> All this is because our main application allows create diferent product
> that can depend on other product. The result is a directed acyclic graph
> that represents products and dependencies among them.
> My idea is to create a more complex elements (nodes, edges and conector)
> based on simple rectangles, lines and groups items, that allows me to
> represent graphs in the canvas.
>
> I hope this will be usefull to other people :)
> Remember to warn me if exists this type of libraries for IDL and I am
> doing the monkey :)
>
> Bye,
> Antonio.
>
> ---------- ----------
>

--

nrb@
Robbie Barnett
imag
Research Assistant
wsahs
Nuclear Medicine & Ultrasound
nsw
Westmead Hospital
gov
Sydney Australia
au
+61 2 9845 7223