Subject: A (maybe) dummy idea about widgets and control manager
Posted by mdelfede on Sat, 15 Sep 2007 07:36:31 GMT
View Forum Message <> Reply to Message

Lurking into IDE code, I've seen (if I'm right) that to draw controls on layout manager you use .usc files, that in practice duplicates the behaviour of drawing routines of controls just to show them at design time.
As I did write some delphi/bc++ controls for Borland tools in the past, I've noticed that they used a quite smart way to spare code.
Theyr controls all have (in base class) an 'in-designer' flag that changes the behaviour of control when it's in designing state.
This brings some advantages :

1- The most obvious : the control can paint by itself when put on the designer. No need to add external code to show it. And it's wysiwyg.

2- Mouse/keyboard events in design mode are not handled on the usual way, but redirected to the designer itself, so for example a button don't get clicked but can be moved/stretched on the designer

3- Having the true control on the designer (and not a graphical copy of it), with some RTTI you can set ALL the properties inside designer, with about no extra code.

Of course, borland put some language extensions in his products that allow things to be easy, in particular mode the __property extension. But it can be done also without extensions, it needs only RTTI and some conventions for property getters/setters
(for example, for a property called Color, should be getColor and setColor; borland simplifies it with a member property Color that calls automatically get/set when rode/written).
The base properties of a widget (position, size, anchors, parent widget) should be in the base class, so inherited by all widgets and handled by layout designer.

There is also a drawback in this approach : a buggy control can hang the ide completely, if no care is taken.

Mirek, if you think that it would be an interesting approach, I can try to make it working.... not an easy task anyways.

Ciao

Max

Subject: Re: A (maybe) dummy idea about widgets and control manager
Posted by unodgs on Sat, 15 Sep 2007 13:41:57 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Sat, 15 September 2007 03:36The most obvious : the control can paint by

itself when put on the designer. No need to add external code to show it. And it's wysiwyg.
Yes, this is a huge adventage. I vote for it;
Quote:
3- Having the true control on the designer (and not a graphical copy of it), with some RTTI you can set ALL the properties inside designer, with about no extra code.

It's better to avoid all that magic. Usc file should only contain code defining which properties are editable at design time and it should describe a way to build widget c++ code.
Quote:
There is also a drawback in this approach : a buggy control can hang the ide completely, if no care is taken.

Fortunately we are not Borland and usualy we fix bugs immediately (I used Builder C++ 5/6 for few years...)

Subject: Re: A (maybe) dummy idea about widgets and control manager
Posted by mirek on Sat, 15 Sep 2007 14:55:23 GMT

View Forum Message <> Reply to Message

mdelfede wrote on Sat, 15 September 2007 03:36
Theyr controls all have (in base class) an 'in-designer' flag that changes the behaviour of control when it's in designing state.
This brings some advantages :

1- The most obvious : the control can paint by itself when put on the designer. No need to add external code to show it. And it's wysiwyg.

We have followed this approach in the past.

The disadvantage is that it is not actually easy to get the widget to layout designer. You can easily add every CtrlLib widget, but for custom developed widgets things are more complicated, you would have to invent some .dll plugin system (not even sure how it should look like).

BTW, why do you think that usc script language tries to be similar to C++?

(The idea is to reuse as much of existing C++ code of Paint routine as possible straight to .usc).

Mirek

Subject: Re: A (maybe) dummy idea about widgets and control manager
Posted by mdelfede on Sat, 15 Sep 2007 18:39:18 GMT

luzr wrote on Sat, 15 September 2007 16:55
We have followed this approach in the past.

The disadvantage is that it is not actually easy to get the widget to layout designer. You can easily add every CtrlLib widget, but for custom developed widgets things are more complicated, you would have to invent some .dll plugin system (not even sure how it should look like).

Yes, of course. Borland used delphi packages for it, they're a sort of a dll with headers containing type info and other things.Their packages consist of 2 parts, design time part and run time part. The former has typeinfo and all needed to be dynamic linked inside IDE, the latter is a simple dll that gives component code.
I think that that system brought the most of Delphy success.
You compile the widget alone, import into the ide and than use it as a built in one. Quite easy job in windows, I dunno in Linux, I'm starting coding on Linux.

OTOH having Ide source, rebuilding it incorporating new controls should be even more simple. I'd prefere the plugin way, ever.

To make a component plugin I think you have to make heavy use of RTTI, or you have to clobber component definitions with functions used to emulate RTTI, which is not so clean code.

Quote:
BTW, why do you think that usc script language tries to be similar to C++?

(The idea is to reuse as much of existing C++ code of Paint routine as possible straight to .usc).

Yes, but you've got to mantain twice the same thing. So boring
job that most controls still don't have their .usc

Subject: Re: A (maybe) dummy idea about widgets and control manager
Posted by mdelfede on Sat, 15 Sep 2007 18:48:19 GMT

Quote:
It's better to avoid all that magic. Usc file should only contain code defining which properties are editable at design time and it should describe a way to build widget c++ code.

Can be done too, OTOH it requires a separate file with property definitions. It would be better to find a way to fit this code inside class definition. Borland did it with __published declaration section, that had the sole purpose of naming the exported properties.
Maybe something like that can be done with pure C++... I don't know yet.

Quote:

Fortunately we are not Borland and usualy we fix bugs immediately (I used Builder C++ 5/6 for few years...)

Me too, and I made quite many programs on it. I find it one of the best IDEs ever made. The only great mistake they did (not their fault, ever) was to make it not binary compatible with M$ compilers. The main fault is the lack of binary standardisation in c++, in particular regarding code mangling and templates.
I made some extensions of Autocad (which is done using M$ C) in the past, and dealing with binary incompatibility was a nightmare.
BC++ was not buggy, besides some small things; many components were buggy, and could crash the ide. They should have made a better error handling inside.

Ciao

Max

Subject: Re: A (maybe) dummy idea about widgets and control manager
Posted by mdelfede on Sat, 15 Sep 2007 20:59:11 GMT

View Forum Message <> Reply to Message

I started playing with the idea above; a little change in CtrlCore.h (ignoremouse = 1 if in designer), few code lines and widgets are created, moved and resized on my code main window.

So far, this works statically, so you must know in advance (ide compile time) the classes of widgets.

Next step would be creation by class name, and property set/get/list, which requires RTTI and a bit more coding but is manageable too. That would require some changes to existing controls, too, in order to 'export' the properties.
That would allow to instantiate controls by name (aka string), so they may be unknown at compile time.

A step further, controls linked dynamically; that's very platform / compiler dependent. On Linux, a good start is here :

http://www.linuxjournal.com/article/3687

On windoze, there are many more advanced DLL functions to deal with dynamic module loading/executing.

All that would allow to deploy controls (I'd call them components) as an include file + a precompiled shared library, that could be imported inside the ide with an 'import control' command. All without recompiling anything,

Again, if you think it's worth the effort, I can try to go further.

Ciao

Max

Subject: Re: A (maybe) dummy idea about widgets and control manager
Posted by mirek on Sat, 15 Sep 2007 21:25:41 GMT

View Forum Message <> Reply to Message

mdelfede wrote on Sat, 15 September 2007 16:59I started playing with the idea above; a little change in CtrlCore.h (ignoremouse = 1 if in designer), few code lines and widgets are created, moved and resized on my code main window.

So far, this works statically, so you must know in advance (ide compile time) the classes of widgets.

Next step would be creation by class name, and property set/get/list, which requires RTTI and a bit more coding but is manageable too. That would require some changes to existing controls, too, in order to 'export' the properties.
That would allow to instantiate controls by name (aka string), so they may be unknown at compile time.

A step further, controls linked dynamically; that's very platform / compiler dependent. On Linux, a good start is here :

http://www.linuxjournal.com/article/3687

On windoze, there are many more advanced DLL functions to deal with dynamic module loading/executing.

All that would allow to deploy controls (I'd call them components) as an include file + a precompiled shared library, that could be imported inside the ide with an 'import control' command. All without recompiling anything,

Again, if you think it's worth the effort, I can try to go further.

Well, I think that it is step back compared to current solution.

IMO, doing .usc part is really simple - almost as simple as implementing design mode for widget (that one does not come for free too - I know that because 4 years ago, we were using exactly this approach).

Maybe the reason why there are still missing .usc files is simply rather the fact that except for the most frequently used widgets, it is not really worth the effort for anything practical.

Just a note, U++ origins rather lie with disappointment with "Visual tools". It is really intended as non-visual library, only using basic visual design for the parts where it makes sense.

Well, over time, it evolved a bit... But I do not see any *practical* advantage for such solution, and believe me, I produce up to 20 dialogs / week for money...

The fact that sometimes I need to use "User class" does not really bother me a lot..

Mirek

Subject: Re: A (maybe) dummy idea about widgets and control manager
Posted by mdelfede on Sat, 15 Sep 2007 21:55:47 GMT
View Forum Message <> Reply to Message

luzr wrote on Sat, 15 September 2007 23:25
Well, I think that it is step back compared to current solution.

IMO, doing .usc part is really simple - almost as simple as implementing design mode for widget (that one does not come for free too - I know that because 4 years ago, we were using exactly this approach).

Of course, with no special compiler support is not so easy, but with a good ctrl class planning is almost for free. I made many
BC++ components that worked such a way and it's not difficult.

Quote:
Maybe the reason why there are still missing .usc files is simply rather the fact that except for the most frequently used widgets, it is not really worth the effort for anything practical.

That depends. I'm experimenting with TheIde (which I begin liking much, I must admit) and one of few things I miss is the complete visual layout editor. I miss it because it's so nice working with it (against, for example, wxwidget layout editors that with spacers/sizers/boxers are unusable for me).

I was designing a window with splitters, and I can't go visually, I must code all by hand. Ok, it's not difficult, but it would be nicer to drop a splitter on layout, drop inside 2-3 child widgets, a menu, some buttons and have the interface ready to use ! At the moment, you can't even with 'only the control frame' approach, as the editor can't deal with child widgets, and that's really a pity.

Quote:
Just a note, U++ origins rather lie with disappointment with "Visual tools". It is really intended as non-visual library, only using basic visual design for the parts where it makes sense.

That I did understand reading you about your fast arrays.
I also think an IDE must not mishave important things only to have a nice interface. But if you can have both....

Quote:
Well, over time, it evolved a bit... But I do not see any *practical* advantage for such solution, and believe me, I produce up to 20 dialogs / week for money...

let's say that the real *practical* advantage would be to have the contributed components "complete" by design
Hmmm.... there's another "advantage"... the ability to have closed source components imported inside the ide.

BTW, I think that the biggest improvement would be to allow child controls inserted correctly inside container widgets.
Doing my way or with .usc files is not quite important.

Just a question.... why did you change the layout designer approach 4 years ago ?

Ciao

Max

EDIT :

I added some RTTI to Ctrl classes, and it does work with very few overhead. I can now create controls by class name; adding properties should be quite straightforward.

The only really *BIG* caveat of this way of making widgets is the extensibility.
For controls built in TheIde, no problem, they work very well.
To add a control, I've 3 choices :

1- Add it to the UPP source and recompile the ide ( )
2- Use the same way as is it now, so 'user class' with only the empty rectangle on layout editor
3- Build a plugin system with shared libraries, so users can create controls and import inside the ide.

The point 3 is of course the cleanest but.... as controls use the upp library, this works ONLY building UPP as a shared library too. The system would be quite easy to code, but I think that the enforcement of having all built as shared lib would make it not so attractive for many people....
Borland uses indeed that way.
There are other ways to do it, but all involves the export of all UPP library by very complicated means.