
Subject: Re: Interesting....

Posted by [andrei_natanael](#) on Tue, 27 Oct 2009 09:59:53 GMT

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

tojocky wrote on Tue, 27 October 2009 10:44luzr wrote on Wed, 12 April 2006 14:20Ah,

well, it is a little bit hard to explain, but....

U++ layouts are designed in a way that leads to "dialogs as structs". Means each layout naturally generates some C++ struct that contains widgets as members. This is one of dominant reasons why coding in U++ is so fast.

Now with dialogs stored in XML, this feature would be lost, because you would have to dynamically create dialogs by parsing XML and best you could hope for is to access widgets using some sort of textual IDs. U++ productivity gone.

While it is quite possible to implement this in U++, the question is why? If some real world app needs this, it will have most likely other very specific requirement, satisfying them would be most likely harder than "XML dialog" skeleton code, which really is quite primitive.

Just a side note - I have couple of applications that store sort of dialog layouts in Oracle. Then others that store dialog layouts in proprietary text format. Still I do not see any advantage to provide some fixed facilities for dynamic dialogs.

Mirek

An interesting discussion i missed here.

About dynamic dialogs...

The main reason of dynamic dialogs is possibility to separate programming work by design work. With dynamic dialogs will be possibility to integrate styles too (visual styles)!

I think that is more important to make work easy even if is detrimental for running speed.

With this possibility will be possible to integrate forms in web browser! It is Not bad idea!

Ion Lupascu (Tojocky)

I don't see how you separate the code from the app design. You write handlers for widgets events. Even if you create a descriptive form of a window in XML file you still should parse that file, create widgets based on what you get from that file and your application should do something when user push a button or something. So the design of application imply also code.

How you will integrate styles? You cannot write CSS style attached to XML because U++ does know nothing about it.

I see the only benefit of a XML based design only when you write plug-ins which know nothing

about your GUI. You write a XML file with description of (allowed) widgets in it(containing widget name, position, etc) and then write event handlers(callbacks). Your app analyze the XML file, create a proper window based on info found in it and when an event occur from that window widgets it call your callbacks in plug-in, and your plug-in is doing his work. So, your plug-in may use the GUI without knowing nothing about it (you may use U++, Qt, GTK+, etc) and you don't have to change code in widget if for example you changed a widget name in your GUI library.
