

---

Subject: Re: MVC example

Posted by [cbpporter](#) on Wed, 19 Mar 2008 06:58:02 GMT

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

---

If you want pure MVC, IMO U++ is not best suited for you. For one, a lot of GUI components which represent collections own their own data, so basically they are both views and models in the same time. You may think of this as unnatural from a MVC perspective, but in practice it makes a lot of sense for the control both to own and manage it's data. Also, MCV imposes some tight coupling between fairly redundant classes. While that coupling is tight logically, it is pretty loose from structural point of view (which means you have to use glue code to tie components together), which does not fit that well with U++ approach, where everything belongs somewhere, rule imposed also by code layout.

But if U++ was not built for MVC, I could accomplish it. It has something called a Display, which is a view used to alter the visual representation of the data from a control. Look over the [reference/Display](#) example to see a very simple example of it. Here you have a DropList which owns a list of fonts, providing default view for it's data, and that view is replaced with a custom one, which this time is a pure view. Such uses of explicit views is pretty much all you'll ever need in practice. The list box could have contained any data, which means you don't have to embed GUI items in your model.

Other than this, I can't give you any more samples of MVC pattern from U++ because I'm not aware of any. But, I'm also not aware of any success stories of applications written from this perspective. There is a huuuuge commercial MFC application which uses an imposed MVC model (not a pure one), and I can't stop cursing it because of it's unnecessary casts and other awkwardnesses that it imposes.

---