Subject: Re: true dynamic dispatching with Upp?
Posted by Didier on Thu, 18 Oct 2012 21:15:55 GMT
View Forum Message <> Reply to Message

Hi Kohait,

if you can't change you're elements, maybe you can encapsulate them in a helper class that would manage the editing part:

```
class ElementHelperBase {
  public:
  virtual Element* get() = 0;
  virtual void Edit() = 0;
}

template<class ElementType>
class ElementHelper : public ElementHelperBase
{
  private:
  ElementType& element;  // initialized by some constructor

  public:
  virtual Element* get() { return &element; }
  virtual void Edit() { EditElement(element); }
}

// using function overloading
// you can add an 'EditElement()' function for each type

void EditElement(ElementA& element)
{
  ElementAEditor editor;
  ... do you're stuff
}

void EditElement(ElementB& element)
{
  ElementBEditor editor;
  ... do you're stuff
}
```

and finally you can do:

```
void EditElement(ElementHelperBase& e)
```

```
{
  e.Edit();
}
```

No need for dynamic_cast<>  any more

This will work, but you need to create ElementHelper classes and instead of keeping track of 'Element's you need to keep track of 'ElementHelper's.

I used somthing close to this in ma GraphCtrl class to manage editing the axis properties depending on axis class type

Hope this idea help's you

---