MITK Clone Macro

Bugsquashing Seminar, 16.09.2015

Jan Hering

Clone Methods

Purpose: provide exact copy of a given object

We access our (data) classes via `mitk::BaseData::Pointer`

> **slicing** efects if using `virtual` copy constructors / assignment operators
>
> - standard copying functions `protected`
> - derived classes forced to override and implement clone method

Clone Methods

Avoid **slicing** efects by using `InternalClone`

```
class Base
{
public:
    Base * Clone() const { B *ptr = InternalClone();}
protected:
    Base( const Base &);

    virtual Base * InternalClone() const = 0;
};
```

```
class Derived : public Base
{
protected:
    Derived( const Derived &);

    virtual Base * InternalClone() const
    {
        return new Derived(*this);
    }
};
```

The ITK Methods

Clone macro provided by ITK:

```
#define itkCloneMacro(x)
  Pointer Clone() const
  {
    Pointer rval =
      dynamic_cast<x *>(this->InternalClone().GetPointer());
    return rval;
  }
```

is calling the protected method in `itk::LightObject`

```
class itk::LightObject
{
[...]
protected:
  /*
   * Actual implementation of the clone method. This method should be reimplemeted
   * in subclasses to clone the extra required parameters.
   */
  virtual LightObject::Pointer InternalClone() const;

[...]
};
```

MITK Clone Macro

- The clone macro defined in MITK provides an implementation of the **inherited** `itk::Object::InternalClone` method using the own copy-constructor

```
#define mitkCloneMacro(classname)
  virtual itk::LightObject::Pointer InternalClone() const override
{
  Pointer smartPtr = new classname(*this);
  smartPtr->UnRegister();
  return smartPtr.GetPointer();
}
```

and **must be placed** in the **`protected`** part of the class declaration.

for own types: create **custom** clone behavior by implementing the copy-constructor

```
MyClass::MyClass(const MyClass &) { [...] }
```

MITK Custom Class Declaration

```
class MyMitkClass : public BaseData
{
public:

    mitkClassMacro( MyMitkClass, BaseData )

    itkCloneMacro(Self)

/* [...] */

protected:
    mitkCloneMacro( Self )

    MyMitkClass();
    virtual ~MyMitkClass();

    MyMitkClass( const MyMitkClass & )
    {
        /* Copy all desired memeber variables here */
    }

};
```

Thank You!