

---

Subject: Looking for new names in new callbacks schema

Posted by [mirek](#) on Tue, 09 Feb 2016 08:05:14 GMT

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

---

I am now sort of refactoring Core to fully 'embrace' C++11. One of things that I decided needs update is Callbacks.

Current (U++2015) version supports lambdas fine, but as sort of 'external' addition. It is not quite effective, lambdas are stored through `std::function` indirectly, so invoking requires unnecessary pointer dereferencing.

While I plan to use a little bit more of `std::` things, I came to conclusion that we need our `std::function` equivalent because

- `std::function` is not Moveable (but should be)
- `std::function` throws exception if it is empty and function is invoked, while more reasonable behaviour is NOP
- our version provides "combination" operator<< that combines several callables into one

Right now, I call this `Upp::Function`, but I do not quite like the name. I am considering `Upp::Fn`, `Upp::Functor`, `Upp::Callable`, `Upp::Delegate` ...

Next thing... Even if we have `Upp::Function`, we still need derived `Callback` and `Gate`. The fundamental reason for that is overloading resolution, `std::function/Upp::function` need to have "catch all" constructor to convert lambdas (which are always of "unknown" type), so using `Function` directly make impossible to e.g. have `Ctrl::operator<=>`.

Meanwhile, with C++11 template parameter packs (template varargs), it is relatively possible to avoid most of current `callback/callback1/callback2...` mess and define single class instead of `Callback...Callback5`. Unfortunately, this class cannot be just "`Callback`", because in parameterless form, it still needs an empty list of parameters ("`Callback<>`") so it would be backward incompatible. (And yes, we still need to provide all those `Callback[N]` and `callback[n]` things for backward compatibility too).

So, so far I have named those new classes `CallbackN` and `GateN`, but I do not like these names either...

Any suggestions about how to name those new entities?

Mirek

---