

---

Subject: Re: Thoughts about alternative approach to multithreading

Posted by [zsolt](#) on Wed, 15 Oct 2008 19:49:33 GMT

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

---

In my current project we (the developers on the project) are using a a queuing framework.

Our experience is, that it is very easy to work with threads in an environment like this. It is very easy for beginners also.

We use UnitTest++ very intensively, and this aproach makes it very easy to create testable classes.

In our implementation, we don't use the old message id based design (used mostly in old embedded systems), as it needs a lot of administration. We use very hard template code. The logic of using the framework is very similar to the Callback system of U++.

We have special Callback-like template classes, but we use it with reversed logic (described a little bit later).

We have Tasks which are basically Thread classes with their own FIFOs. They are waiting for their FIFOs. The incoming items in the FIFO are CallbackXMethodAction like objects.

The logic is reversed if you compare it to U++'s Callbacks, as these Callback like objects are connected mostly to a method of the the same object they are in.

So this Callback like objects are callable from any threads. They are creating a CallbackXMethodAction like object on call and put it their Task's FIFO.

The Task reads this CallbackXMethodAction like object on its own thread and executes the method.

---