Deferred Operations (Deferreds)

It’s no secret and if fact by design, that you can’t execute methods that change a Subject’s structure when using the Solid semantics. That essentially means you can only change the state of the individual Traits themselves, but not add nor remove the Traits themselves. The main advantage of the Solid enchaining is that it provides for the concurrent Operating and it would certainly be great to have more flexibility while evaluating the multi-threaded processing.

So there come handy the deferred operations (or *Deferreds* for short). Like their naming implies, those are not executed immediately but are instead delayed for a later, more suitable occasion.

Say, we want to defer the Trait setting operation in a concurrent Operating. This would be implemented like so:

```
SolidChain->OperateConcurrently([Destination](FSolidSubjectHandle Unit,
FSelected Selected)
{
    Unit.SetTraitDeferred(FMoveTo{Destination});
});
```

From:
http://turbanov.ru/wiki/ - **Turbopedia**

Permanent link:
http://turbanov.ru/wiki/en/toolworks/docs/apparatus/deferred?rev=1653771574

Last update: **2022/05/28 20:59**