

---

Subject: Re: NEW: Dispatcher (templateable dispatcher helper for MVC pattern and more)

Posted by [kohait00](#) on Mon, 12 Jul 2010 11:43:38 GMT

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

---

i dont know exactly how it could help here, the dispatcher is just a means to register multiple recipients for a typesafe messages.

what it could help is, ofcourse, to keep the data structure of ypur application clean. you can focus on data channels (Dispatchable<T>) and group things together, that belong together (recipients for the same data type)

i will try to provide some more examples on how to use the Dispatcher.

actually, the idea comes from one of my applications, where devices send data packets, that i want to be able to monitor in different ways, i.e. as a static text field, as an editfield somewhereelse, as a meter slider in some other place. so all the different controls simply listen for the data on their channels and display them, the dispatcher provides them with the data, and i only need to forward my stuff to the dispatcher, without need to take care everytime, that all controls get their info. thus the dispatcher was born. the version in my app is not that simple, the stuff i have learned there i tried to encorporate in this simple version.

basicly, the idea is not so new. dont know if you are familiar with Microsoft's DSS / CCR framework. they have the data flow control pattern (dont know if that oone exists, i am no prof). but the idea is that one bids up the applications as a reactional framework. reacting on income of messages, an application can have multiple data slots, waiting for different types of messages, having handlers that take care of each data element coming in. the total of that is, that you can create totally new applications by organizing and putting together other applications with their data channels linked together (even across networks, using serialisation of class instances). thats basicly the way a human beeing things of, when modeling a realworld problem. we tend to thing reactional not procedural.

anyway, to make a long story short: the dispatcher was kind of a step in this direction. as of now, ultimate has all the means to realize simiral behaviour. maybe we should go this direction as well to make Upp more attractive.

sorry for long post, sometimes its flowing

---