1.1 Multisync Thread Manager Pattern

1.1.1 Name and Source

Multisync Thread Manager
Page 122-125 in the book ".NET Patterns: Architecture, Design, and Process " [NET 03]

1.1.2 Also Known As

-

1.1.3 Type

1.1.4 Intent

To let the client choose whether it wants to be notified or to make regularly polling.

1.1.5 Problem

Essentially, the problem is similar to the “Event Monitor Pattern”, so you may want to refer to that pattern for a description of the problem.

How can a web service client invoke a long-running web service call without getting the client application blocked while waiting for the response ?

1.1.6 Forces

1.1.7 Solution

You can choose to combine the patterns “Notifying Thread Manager” and “Pollable Thread Manager” as an alternative to implement only one of them.

In other words, the class diagram would be the ‘union’ of the diagrams of these two patterns. You can create a class “MultisyncThreadManager” with the methods from the both classes “PollableThreadManager” and “NotifyingThreadManager” in those two patterns. The client class will have to implement the “CallbackInterface”.

The method “ThreadClass.DoExecution” will have to do two things when it has got the result from the “WebService.slowMethod()”. The first thing to do is to save the result in an attribute so that it can be returned later (this is the “Pollable Thread” part of the pattern). The other thing to do is to notify the Client by calling a method in the “CallbackInterface”, but that must be done within an if statement that checks if there is a reference to such a class, i.e. you must check if the client has used the “Notifying Thread” part of the pattern by invoking the method “MultisyncThreadManager.ExecuteAsync” instead of the method “MultisyncThreadManager.BeginExecution”.

1.1.8 Consequences

While the pattern provides flexibility by letting the client choose how to use the MultisyncThreadManager class, you do not achieve “High Cohesion” when it is used. (“High Cohesion” is a so-called GRASP, i.e. “General Responsibility Assignment Software Pattern” [Larman 01])

1.1.9 Related patterns

The pattern is a combination of the “Notifying Thread Manager” and “Pollable Thread Manager” patterns. The solution implements both these patterns, and the user can choose which of these patterns to use.