Кратко обяснение:

Mediator Pattern е шаблон за поведение. Той решава проблема с комуникацията между класовете. Обикновено, когато имаме два класа и те искат да си комуникират, то единия знае за другия и те се навързват. Така coupling става доста голям и на помощ идва точно mediator pattern. Принципа за максимално разкачане тук е застъпен осезаемо. Медиаторът играе посредник между класовете. Всеки клас знае за медиатора, но не знае за останалите участници в системата. Така той играе роля на контролна точка между участниците и предава съобщения/заявки един към друг.

Приложимост:

* Когато набор от обекти общуват добре, но по-много сложен начин. Крайните взаимоотношения са неструктурирани и трудни за разбиране.
* Когато повторното използване на обект е трудно, тъй като се отнася до и комуникира с много други обекти.
* Когато поведение, което е разпределено между няколко класа трябва да бъде приспособено без много подкласове.

Структура:



Последствия:

* Mediator pattern ограничава създаването на подкласове. Той локализира поведение, което може да бъде разпределено между няколко обекта.
* Той разкача отделните класове и промотира loose coupling между определените обекти. Може да се промени или преизползва както медиатора, така и останалите обекти независимо.
* Медиатора заменя връзките много към много с едно към много между медиатора и другите обекти. Връзките едно към много е по-лесна за разбиране, поддържане и разширение.
* Абстрахира комуникацията между обектите. Направата на независим медиатор, в който се скрива комуникацията между останалите обекти, оставя да фокуса да падне върху това как обектите действат като част от тяхното индивидуално поведение.

Използван в:

Примери за използването на Composite може да се намерят в почти всички обектно-ориентирани системи.

* Et++
* THINK C class library
* Smalltalk/V
* Unidraw drawingframework

Свързани шаблони:

* Façade
* Observer