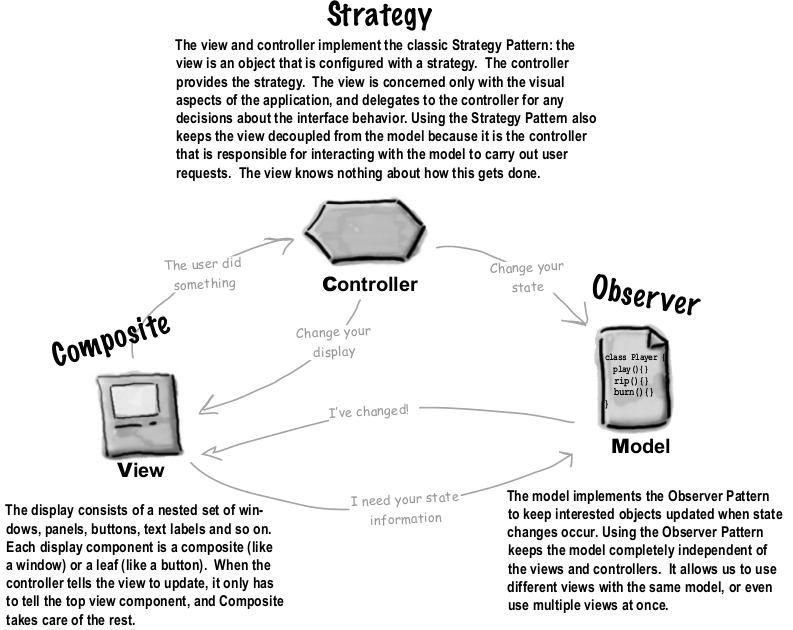
Short definition:

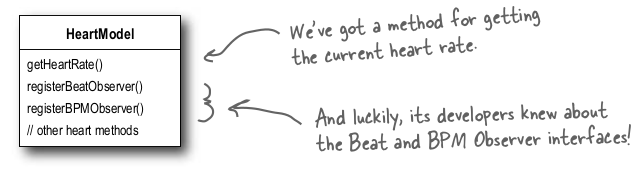
The view only worries about presentation, the controller worries about translating user input to actions on the model. Model is responsible for maintaining all the data, state and any application logic. The model doesn’t know anything about the views or controllers. In other words it is totally decoupled. All it knows is that it may have observers it needs to notify. That’s the beauty of the Observer Pattern. It also provides an interface the views and controllers can use to get and set its state.

Class diagram



Eg :

* (HFDP P546) It’s your time to be the DJ. When you’re a DJ it’s all about the beat. You might start your mix with a slowed, downtempo groove at 95 beats per minute (BPM) and then bring the crowd up to a frenzied 140 BPM of trance techno. You’ll finish off your set with a mellow 80 BPM ambient mix. How are you going to do that? You have to control the beat and you’re going to build the tool to get you there.
  + Design your views and menuitems (you may want to saperate the view that shows model’s state and other which manipulates it)
  + Define actions that can happen via your views and menuitems
  + Define role of the controller
  + Define your model and its functions
  + Dive deep and show how the flow of interactions will happen for each user use case b/w view-controller-model.
  + Define what kind & how many types of observers you need.
  + Define your ModelInterface with methods for init, controller, get/set of state, observers registration/de-registration.
  + Implement your model class now
  + Implement your view classes now
  + Define your controller interface
  + Implement your controller class.
  + Finally, write the client class.
* (HFDP P557) How about a heartbeat display? How can you reuse above view to use the below model?



* (HFDP P541) Imagine you’re using your favorite MP3 player, like iTunes. You can use its interface to add new songs, manage playlists and rename tracks. The player takes care of maintaining a little database of all your songs along with their associated names and data. It also takes care of playing the songs and, as it does, the user interface is constantly updated with the current song title, the running time, and so on.

(Solution : Not given in the book)