Design Pattern

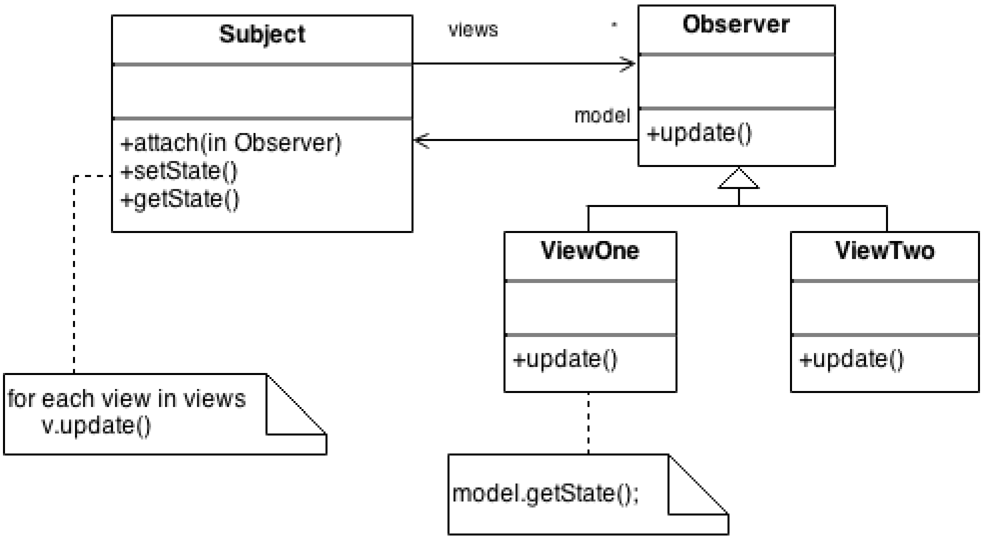
Summary

For this project we used two Design patterns.

- Observer (behavioural pattern)
- Factory (Creational pattern)
- Managed Extensibility Framework (MEF)

Observer

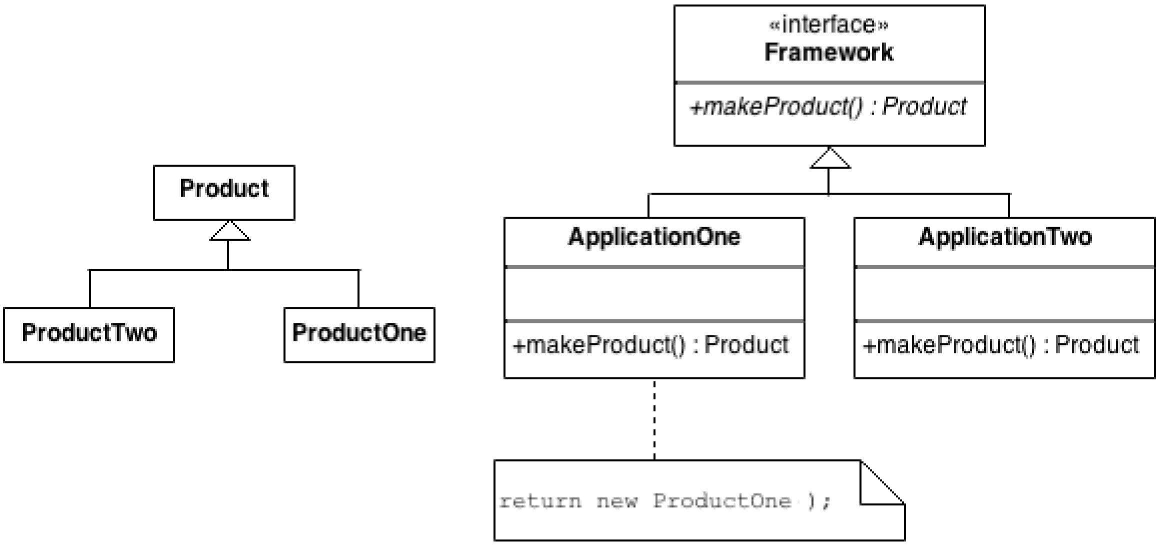
In the Hotel Simulation project the observer is used to notify if there is an event for the object. The object will react on that event based on the status it will get. Instead having one class that gets all the data we decided to only let the class observe on the data it should get. For example a guest should not get data to clean a room. And a maid should not get data to check in a hotel.



Factory

We also used a factory in the Hotel Simulation project but we decided to give it a small twist and combine it with MEF. A Factory method has 1 big disadvantage and that is for not being dynamic. For example if the product owner wanted a new facility in the hotel simulation. In order to implement the new facility we had to write a lot of code in order to work.

So we did some research and looked into MEF and decided to combine these two to make a dynamic factory. it implements the MEF framework to capture new types of areas and can directly work with them. If an area is added, it has to implement the IArea interface and by filling in the create area method. It is compatible with the program. this way it is possible to add a lot of new area's without having to deal with their implementation in the program


«interface»
Framework
+makeProduct() : Product
Product
ProductTwo
ProductOne
ApplicationOne
+makeProduct() : Product
ApplicationTwo
+makeProduct() : Product
return new ProductOne);