Het Model-View-Controller design pattern scheidt in de programmatuur de weergave, business logic , en gegevens van elkaar. Het pattern bestaat eigenlijk uit meerdere andere design patterns:

* Controller
  + Mapped gebruikeracties naar updates in de desbetreffnde Model(s).
  + Implementeert een manier van reageren op input.
  + De relatie tussen een controller en een view is een voorbeeld van het Strategy pattern.
* Model
  + Bevat gegevens die door de views weergegeven kunnen worden, en door de controller bewerkt kunnen worden.
  + Notifies Views bij statechanges door middel van bijvoorbeeld Observer Pattern.
  + Kan data opslagen in een database of xml.
* View
  + Presenteert de gegevens van een model.
  + Views kunnen genest zijn. Dat wil zeggen dat een view andere views kan bevatten. Dit is een voorbeeld van het Composite pattern.
  + Door middel van b.v. het Observer Pattern gekoppeld aan een Model. Een Model roept Update aan op de view als hij gewijzigde gegevens bevat.

Het MVC design pattern is toegepast bij ons design van WOZ: op de volgende klassen en onderdelen:

* Controllers:
  + Alles afgeleid van IMoveEngine, deze besturen beweegbare objecten(verschillende soorten moveengines, zoals keyboardcontrol en autowalkpath, kunnen aan alle iMovables gekoppeld worden);
* Models:
  + Alles afgeleid van IRoomObject, bevat status informatie van het object
  + Statefull event, bevat statusinformatie van een event
  + Room, bevat objecten die in de room aanwezig zijn
  + Game, bevat rooms, events, enz.
* Views:
  + ObjectDrawer -> Observer gekoppeld iMovable objecten
  + De views in de GUI zelf (Java AWT/SWT)

Het MVC pattern beschrijft eigenlijk 3 soorten klassen die met elkaar samen werken, controllers, models en views. In onze applicatie zit die samenwerking eigenlijk zo:

ObjectDrawer, tekent de context van een beweegbaar object. Een iMoveEngine bestuurd een object, en veranderd daarmee zijn positie. Het beweegbaar object is in feite het model. Maar de iMoveEngine zit niet gekoppeld aan de view, maar ook aan het model. Dus dit is een iets andere implementatie van het pattern.