

4156

date for Oct 29th
midterm ask Jason

page 1

10/8/15

MVC & other patterns

someone somewhere has already solved your problem or a very similar problem

- Sometimes this results in a library or framework where you can directly reuse open source code
- design & architecture patterns are instead a way of organizing code or code interactions, reusing experience
- patterns are "discovered", not "invented", find solutions that many developers have already used in many different projects

(anti-patterns - many developers already found this was a bad idea)

many books & websites will list design patterns, often w/ code samples in various languages (catalogs)

Original Gang of Four Design Patterns book (1995) listed 23 patterns - MVC was not only of them

- GoF patterns "Smaller"
- MVC made up of several GoF patterns

MVC is an architecture

- other architectures include client/server, peer-to-peer, N-tier
- MVC at center of most desktop, web, & mobile apps - which all have a GUI

what is MVC?

Model
View
Controller

model - data, classes & methods that manipulate data or states, business logic

view - renders / displays data for user to see
updated when model changes

controller - translates user actions to operations on model

4156

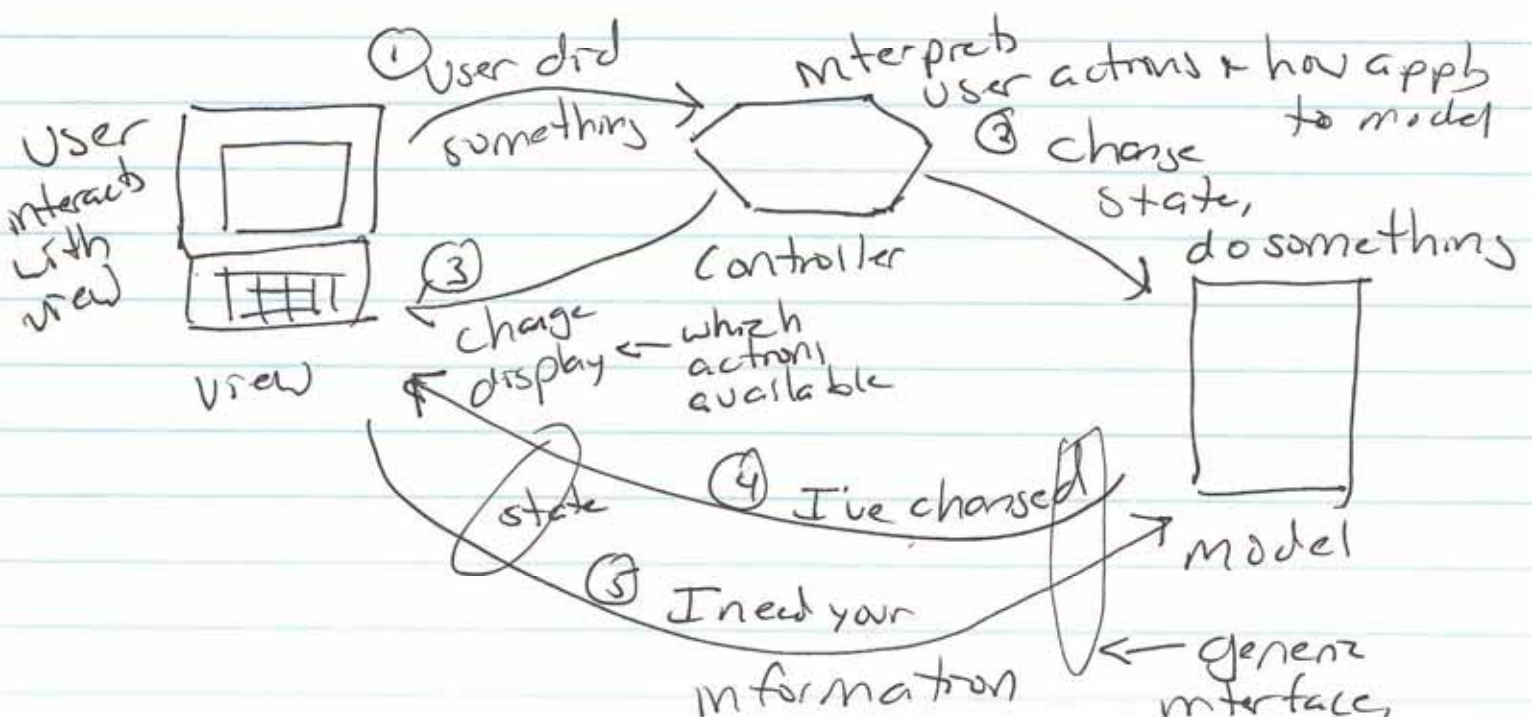
10/8/15

example - itunes or some other music player

view - shows play list, current song, etc.

controller - ~~except~~ accepts UI actions from user
tells model what to do

model - all state, data, application logic to play songs, rip, burn, etc.
tells view when state has changed



loosely coupled,
view & model independent
controller knows about both

general interface,
model knows
nothing about
view

4156

10/8/15

MVC, if implemented properly, reduces coupling & increases cohesion

- 2nd paper to read discusses good & bad ways to implement

Coupling = interdependencies among program units, "need to know"

cohesion = degree to which a program unit exhibits a single purpose

need to balance, since of course program units need to work together to implement application

MVC separates the UI from the computational model - developing a nice UI requires different skill set than developing good data structures & algorithms, & is likely to be changed more often

there can be multiple views &/or controllers, but only one model

→ show AP example

4156

10/8/15

ad lib : there are disagreements among experts as to where the "business logic" belongs in the MVC architecture, as well as whether or not the model & view ever interact directly n.t. always through the controller as mediator

the presentation was based on the AP Computer Science variant. Note the MVC Song takes the other approach.