



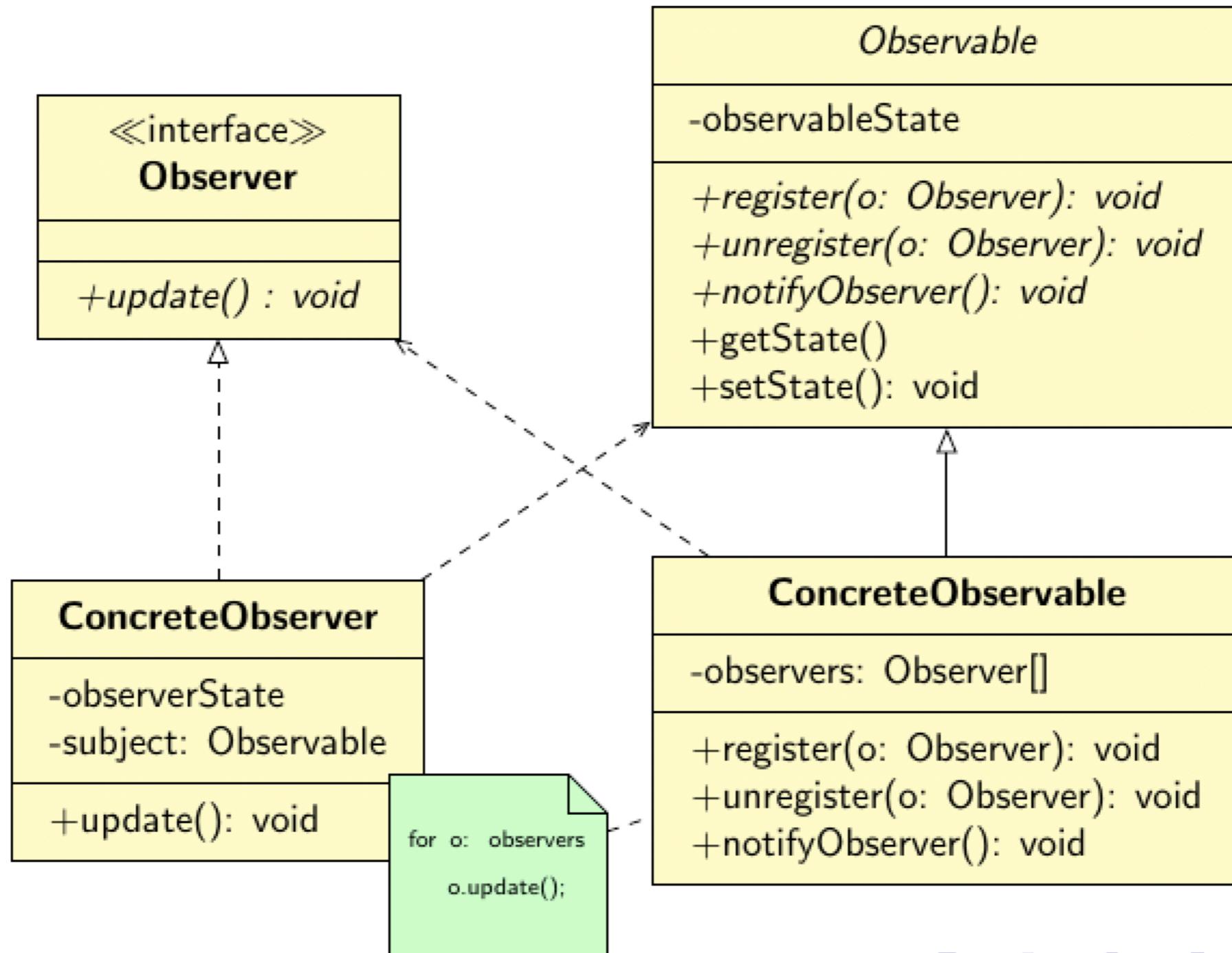
UNIVERSITY OF
TORONTO
MISSISSAUGA

CSC207

Week 8: MVC

Design Patterns: Observer Example

Observer Pattern UML

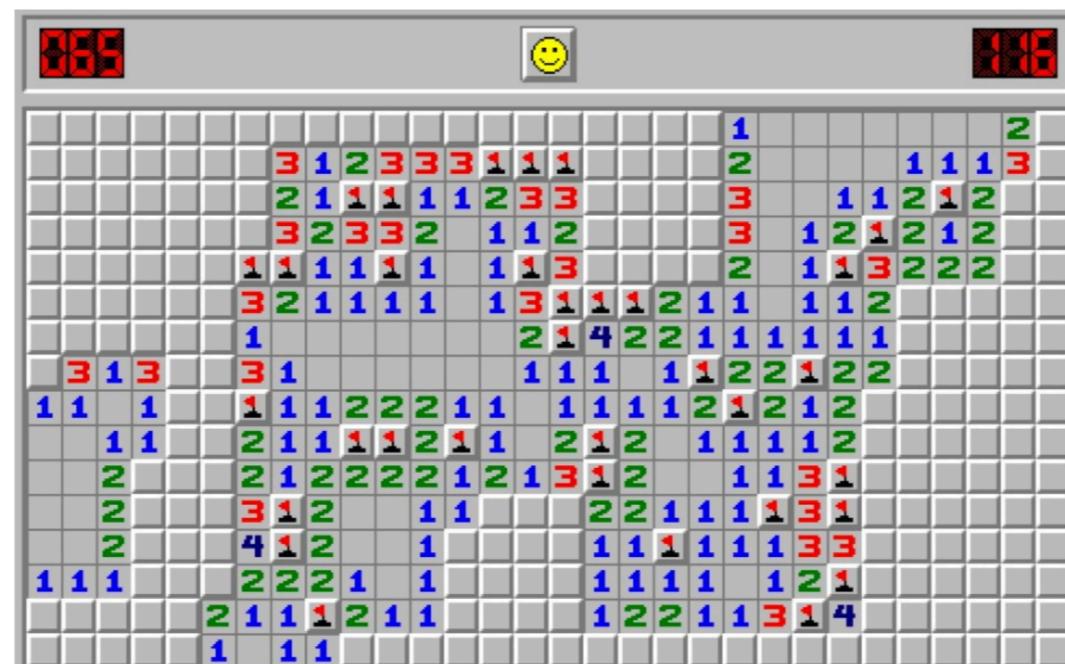


The Observer pattern in GUI Applications

A **GUI** is a **Graphical User Interface**

The Observer Pattern frequently plays an important role in GUI design

Observer patterns may be embedded within architectures that separate **models** of system state from corresponding **views**

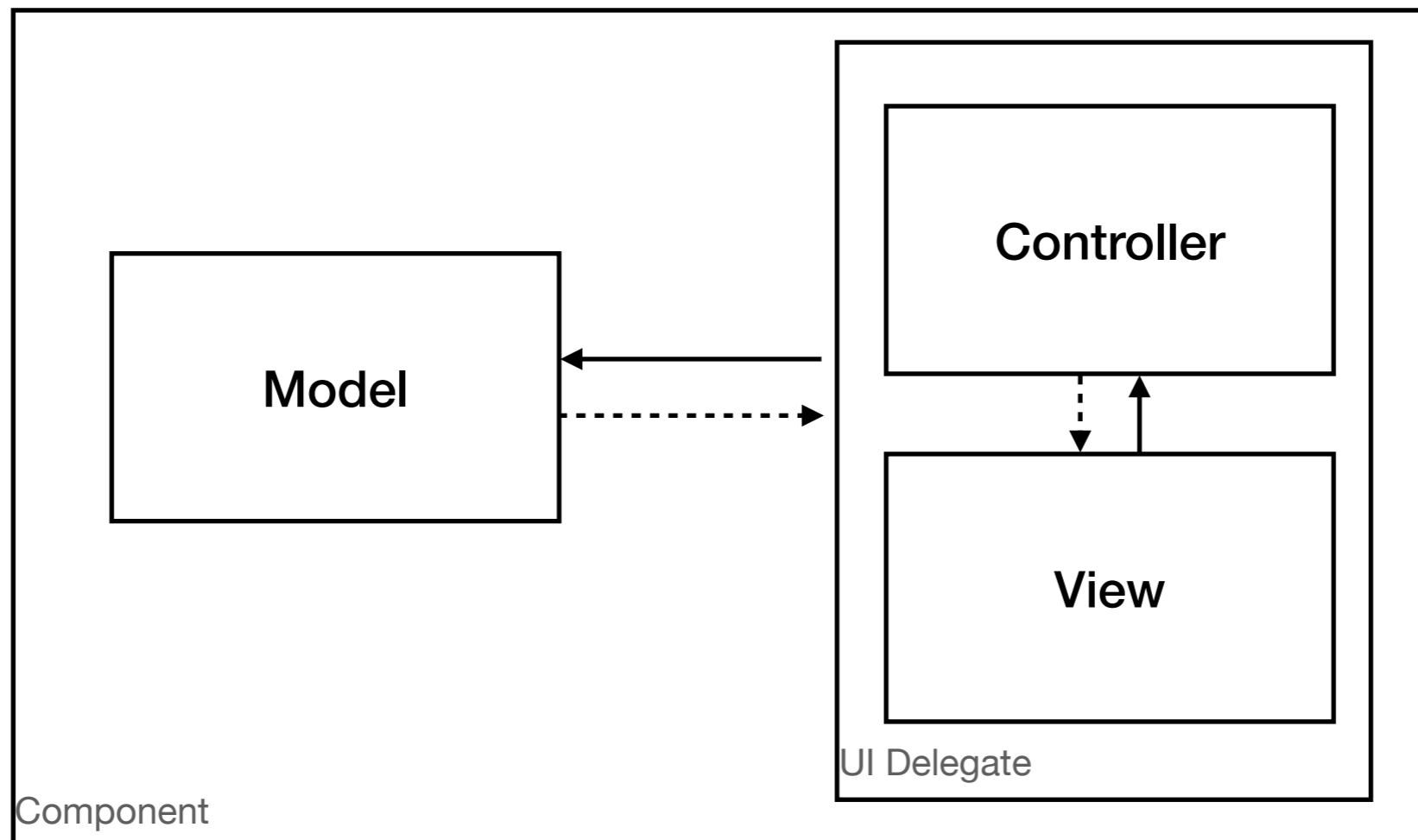


Board =
[[0,0,0,0],
[0,1,0,0],
[0,0,1,0]]



UNIVERSITY OF
TORONTO
MISSISSAUGA

Model-View-Controller Design

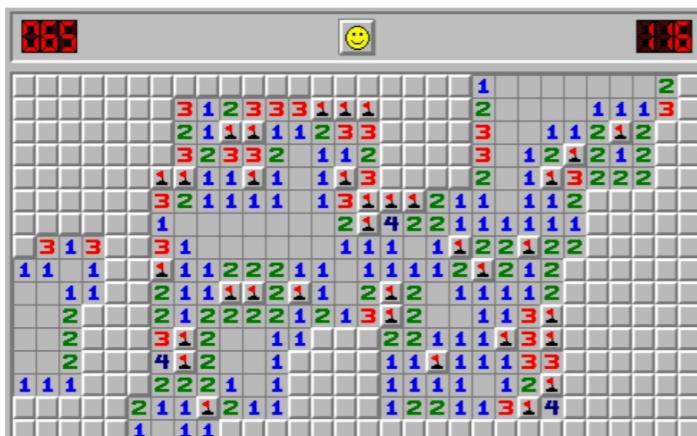


Originally introduced in 1979 at Xerox Palo Alto by Trygve Reenskaug

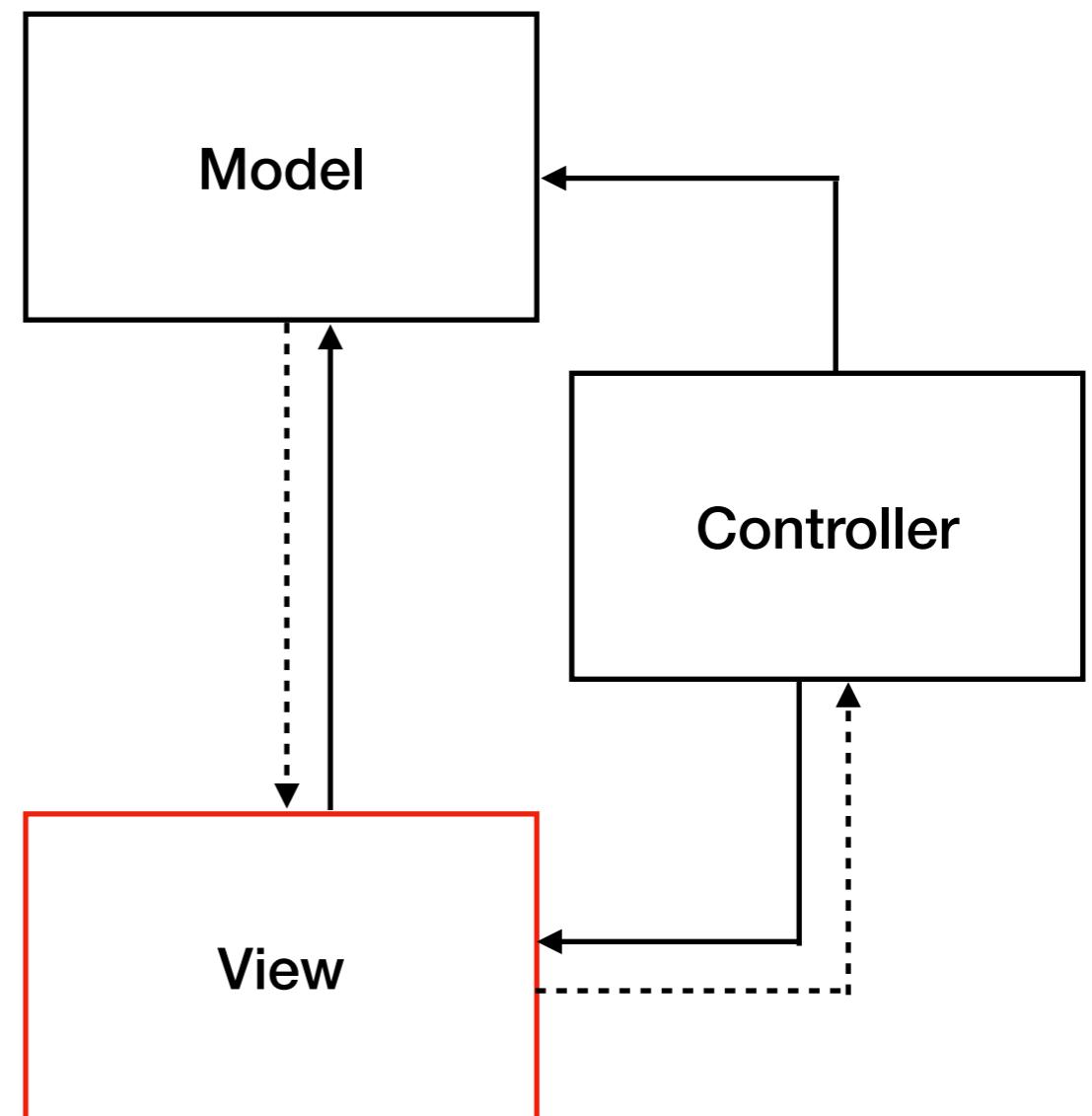
Model-View-Controller Design

View:

- Renders the models
- Listens for model updates
- Sends user input to controller



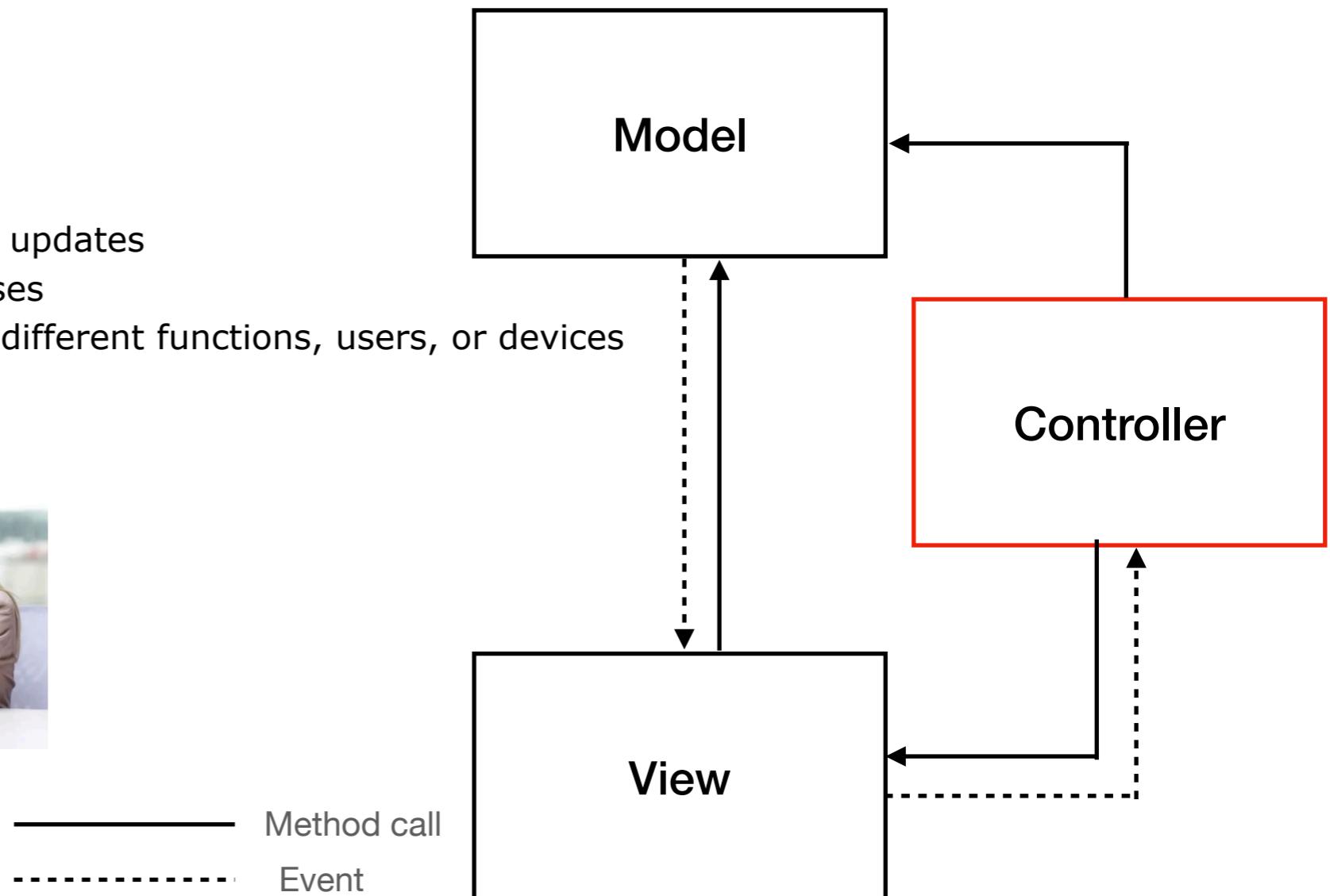
— Method call
- - - - Event



Model-View-Controller Design

Controller:

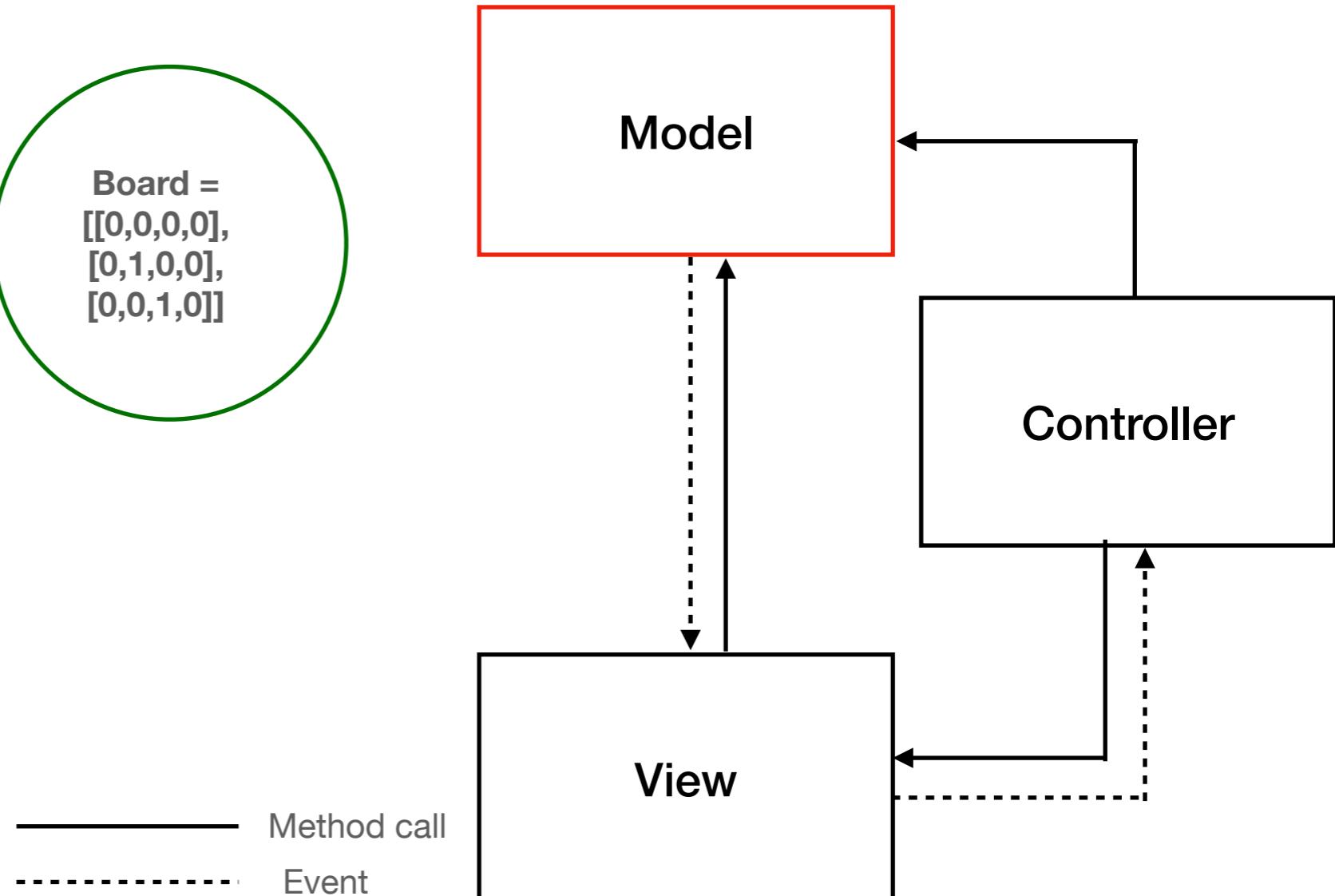
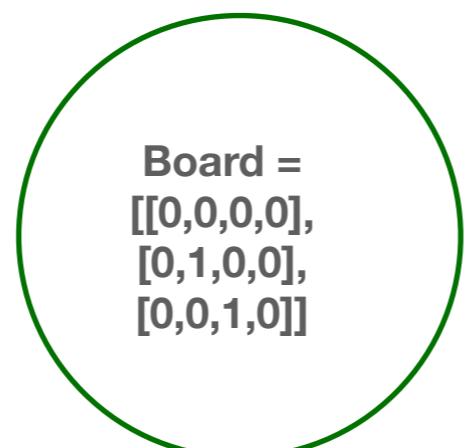
- Maps user actions onto model updates
- Selects views for user responses
- May select different views for different functions, users, or devices
- Contains application logic



Model-View-Controller Design

Model:

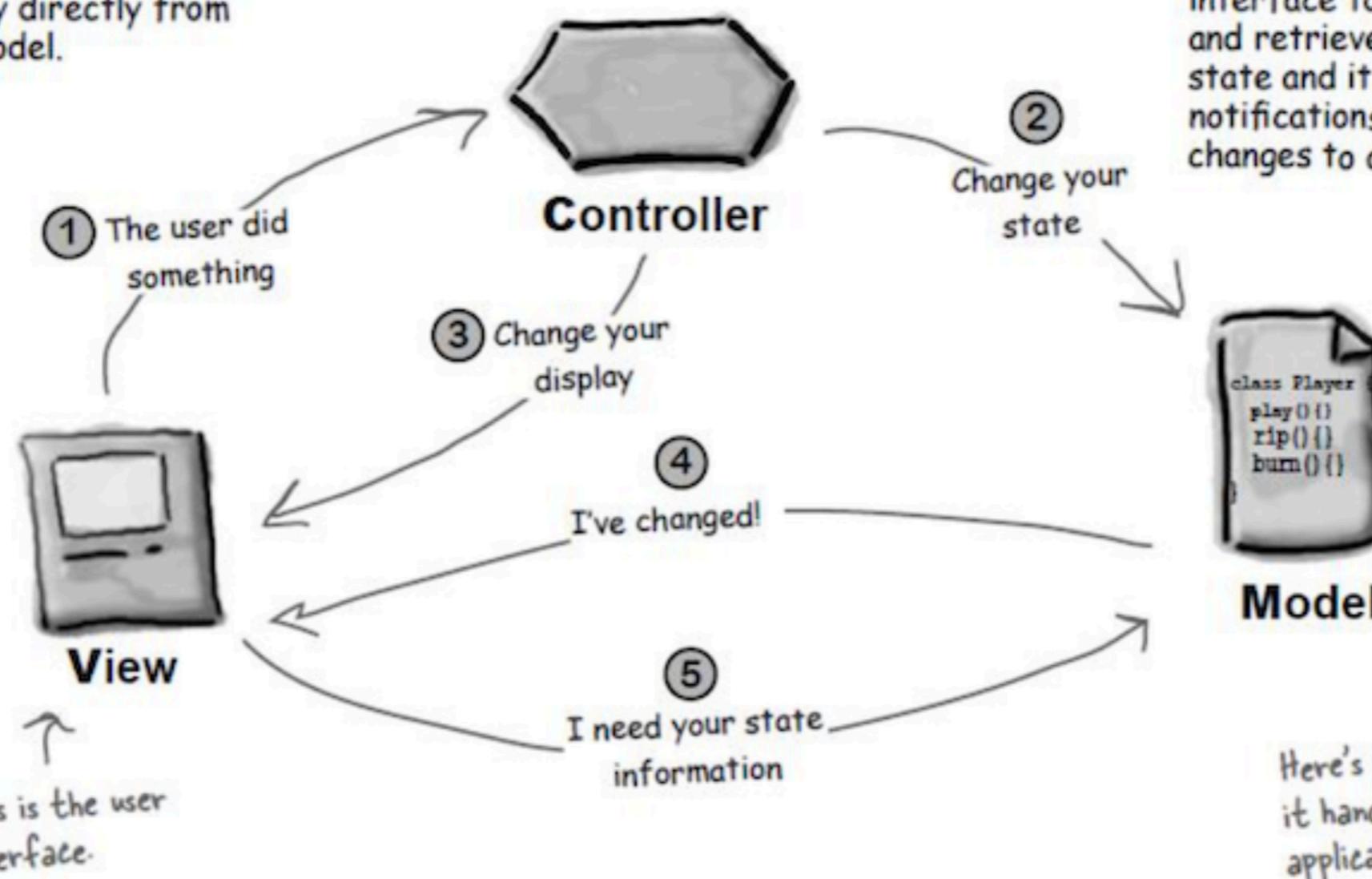
- Reflects application state
- Responds to state queries
- Notifies views of changes



Model-View-Controller Design

VIEW

Gives you a presentation of the model. The view usually gets the state and data it needs to display directly from the model.



CONTROLLER

Takes user input and figures out what it means to the model.

MODEL

The model holds all the data, state and application logic. The model is oblivious to the view and controller, although it provides an interface to manipulate and retrieve its state and it can send notifications of state changes to observers.

Model-View-Controller Design

MVC **separates concerns** among its components, which facilitates design, communication and testing.

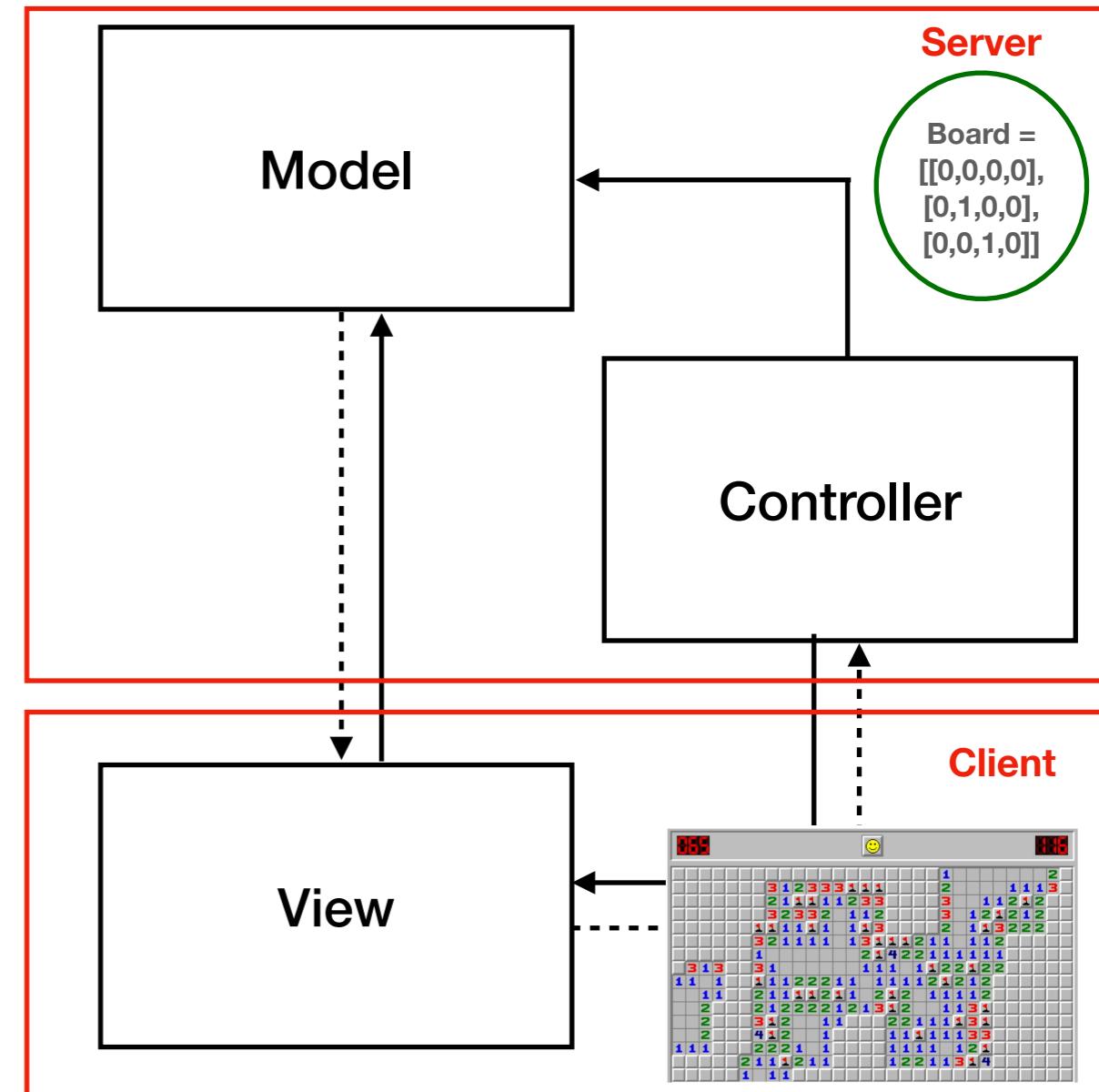
Separation of concerns in **software** is often related to separation of concerns in system **architecture**, too.

It maps naturally onto **interactive (GUI) systems**, including **web applications**.

Web programming platforms that now conform to MVC include:

*.NET (ASP.NET MVC)
JAVA (Spring Web MVC)
PHP (Joomla)
Python (Django)
Ruby on Rails
... and many others*

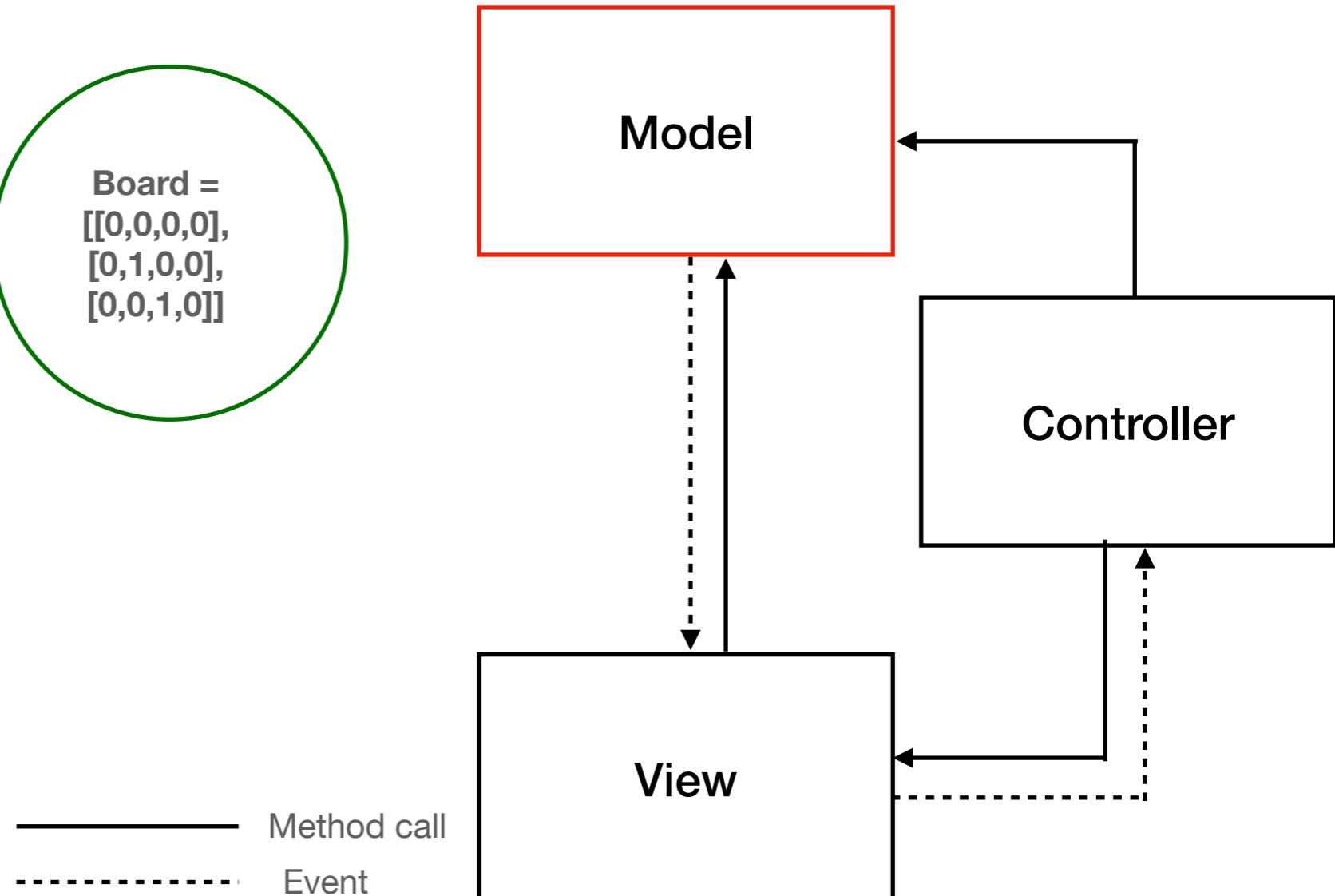
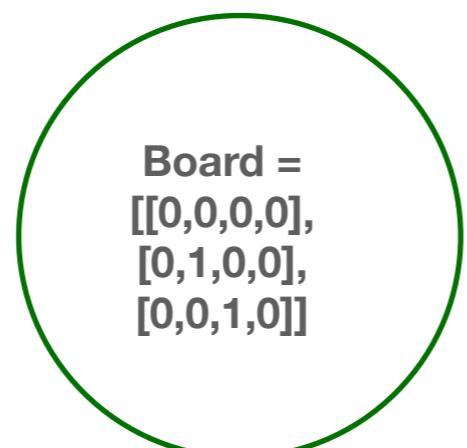
— Method call
- - - - Event



Model-View-Controller Design

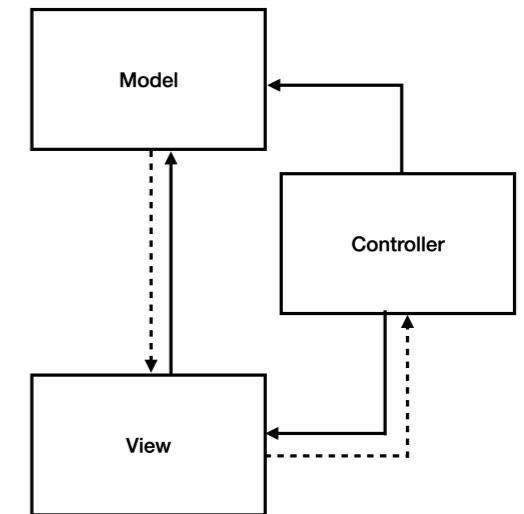
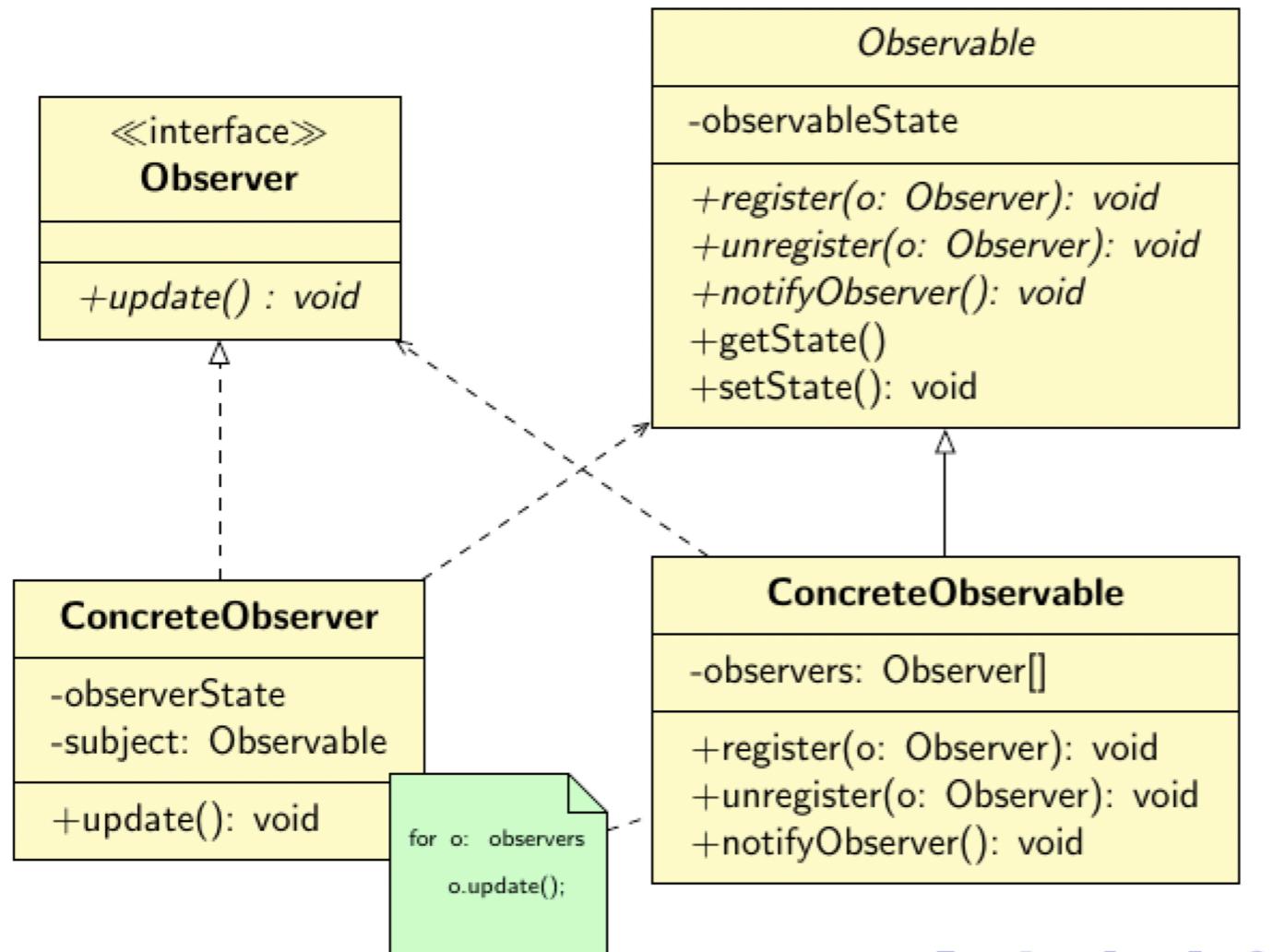
Model:

- Reflects application state
- Responds to state queries
- Notifies views of changes



MVC and the Observer Pattern

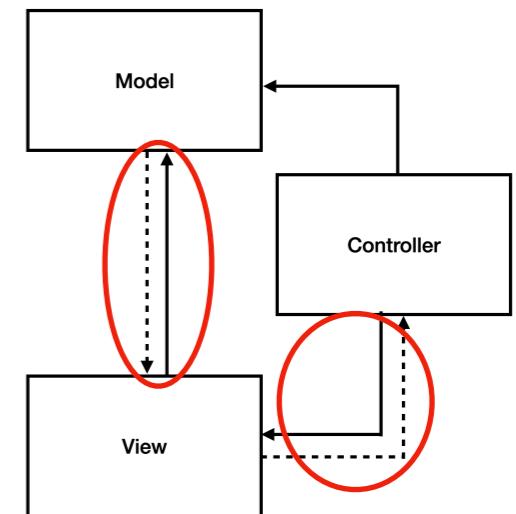
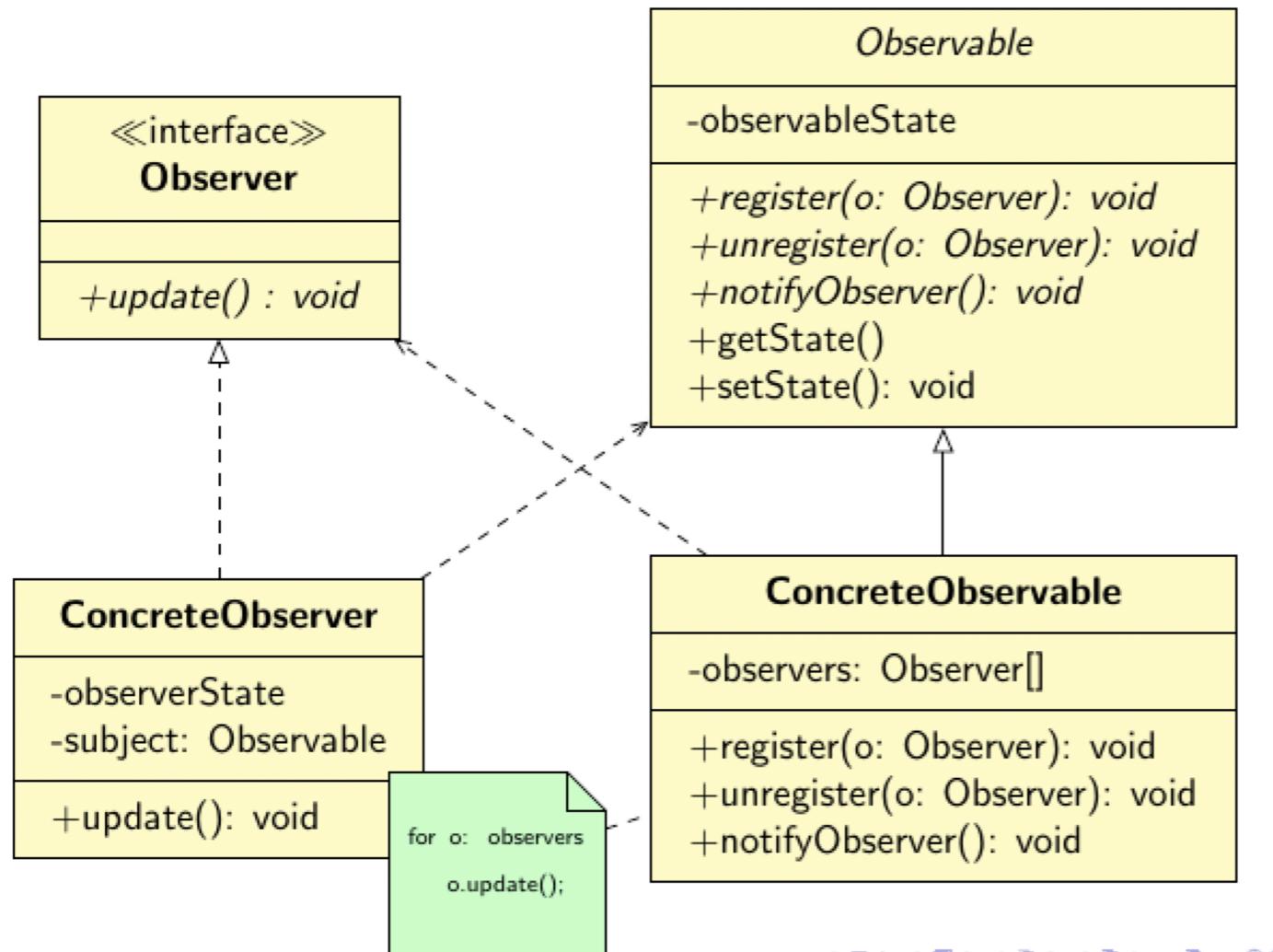
Observer Pattern UML



Where might this pattern fit within MVC?

MVC and the Observer Pattern

Observer Pattern UML



- A couple of places! Observers can allow the controller to **listen** for GUI events.
- Observers can allow the GUI to **listen** for changes in model state
- Observers can be used to facilitate **communication between modules**, while maintaining **separation of concerns**

Observer in GUI Applications

In Java, interactions with GUI elements (like buttons) are typically called **Events**.

Event Listeners act like **Observers**; work is delegated to **Event Handlers**.

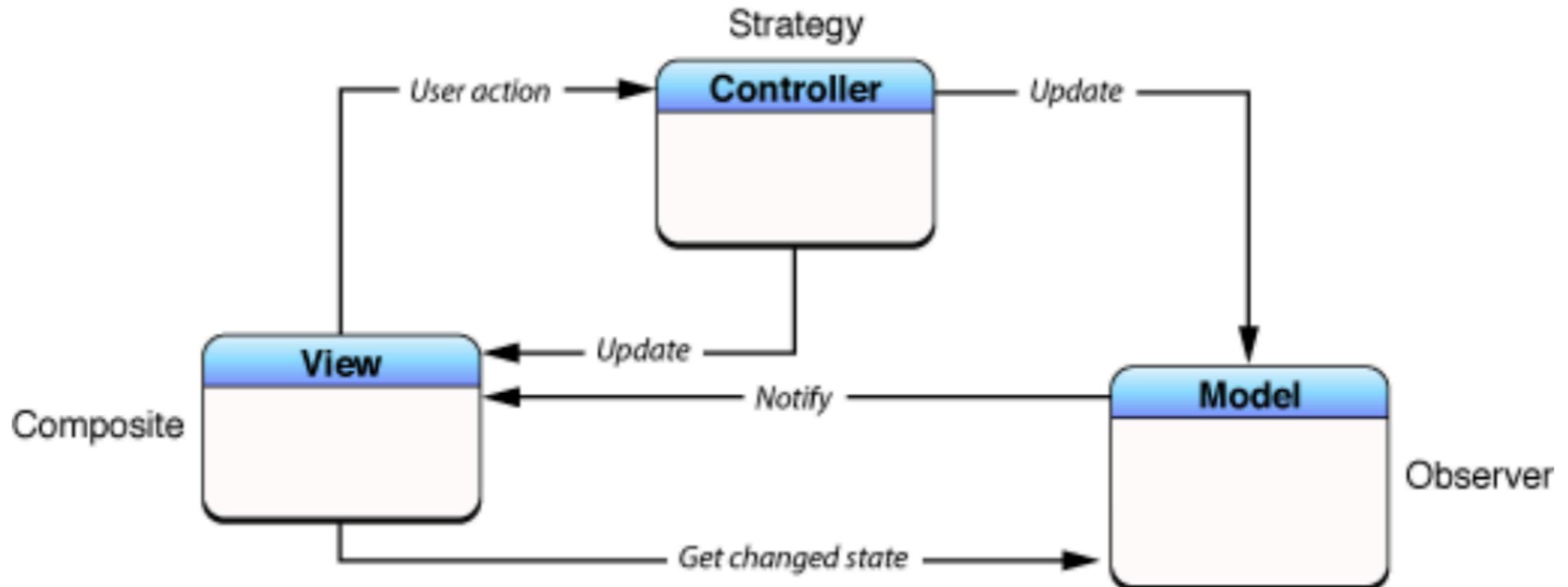
GUI Event Handling has been built into the **Java Delegation Event Model (DEM)**.

This makes it very easy to build observer relationships between **View** components and underlying **Models** of state.

Observer in GUI Applications

```
public class HiBye extends Application {  
  
    @Override  
    public void start(Stage stage) throws Exception {  
  
        Button bHi, bBye;  
        bHi = new Button("hi");  
        bBye = new Button("bye");  
        TextField tf = new TextField();  
        HiByeEventHandler hbh1 = new HiByeEventHandler(tf);  
        HiByeEventHandler hbh2 = new HiByeEventHandler(tf);  
        bHi.setOnAction(ActionEvent.ACTION , hbh1);  
        bBye.setOnAction(ActionEvent.ACTION , hbh2);  
  
        // more code  
    }  
}
```

Observer in GUI Applications



Other patterns (e.g. the **Composite** Pattern, or the **Strategy** Pattern) can be related to MVC components as well. **We will talk more about these in the coming weeks, but you are welcome to start reading ahead ... !**