Model View Controller Pattern – Behavioural

- Intent
  - » **Partition user-interactive applications into three parts**
    - > **Model**
    - > **View**
    - > **Controller**

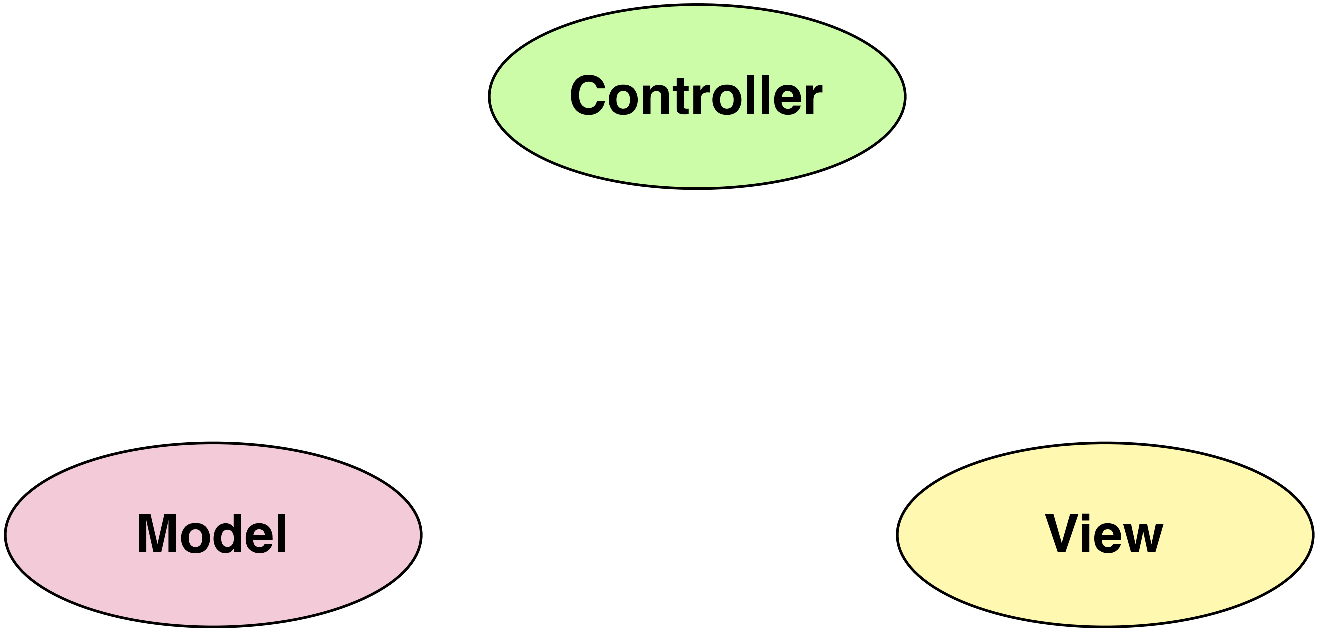
- Motivation
  - » **Use divide and conquer to simplify interactions among the parts of a program**

Participants – 1

- Model
  - » **The part that does the actual computation, data manipulation and processing**
- View
  - » **The purely graphical part of the application, taking care of presenting information visually and interacting with users. Notions such as buttons, other controls and events belong here**
- Controller
  - » **Connects the model with the view by specifying what operations from the model must be executed in response to what user events occur**

Participants – 2

Mediates between the model and the view



Controller

Model

View

What your application is

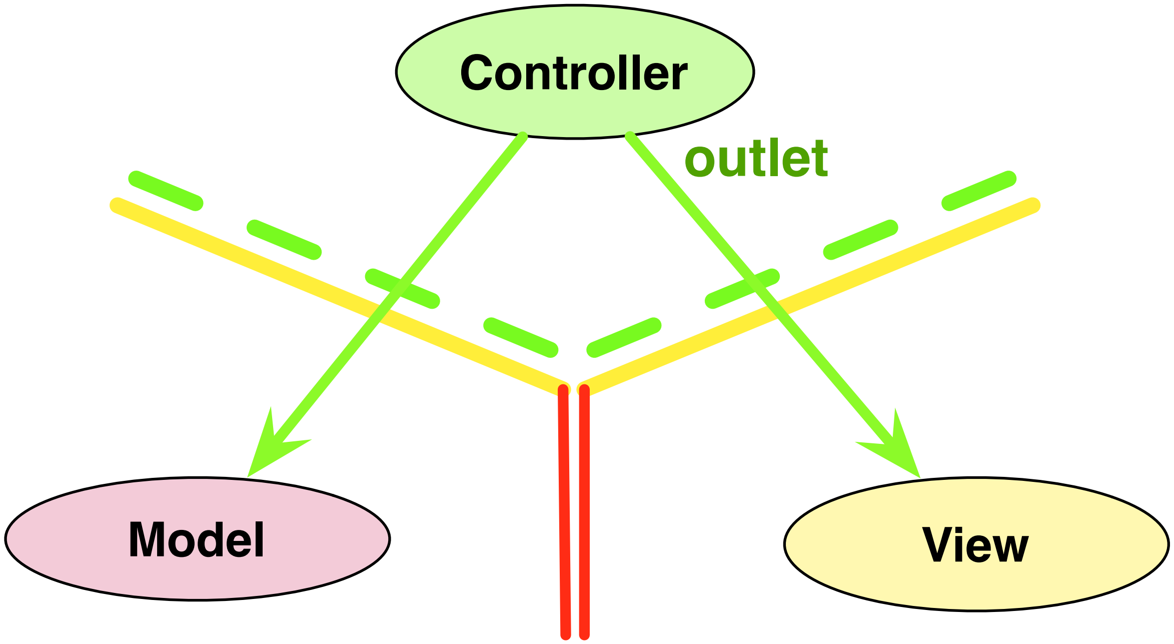
How a user sees and accesses the model

Diagrams based on Stanford CS193p Fall 2011 slides

Communications

It’s all about managing the communications

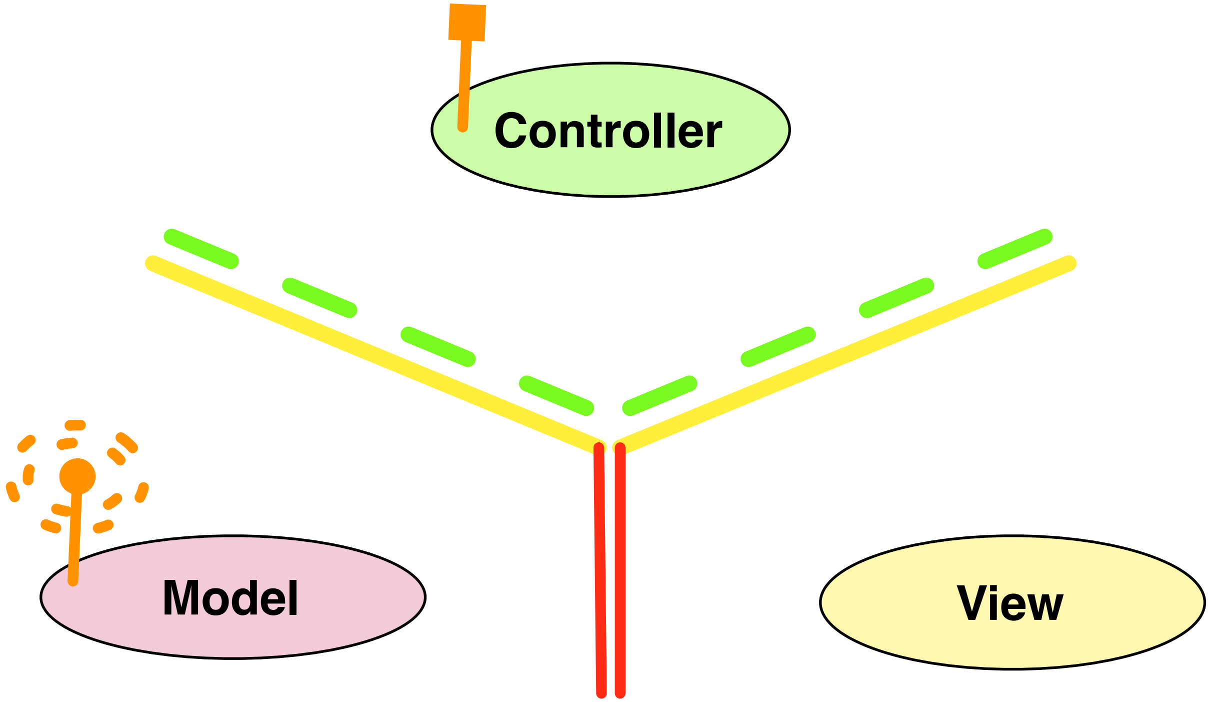
Controller can always talk to the model and view



Model and view never talk directly to each other

Communications – Model

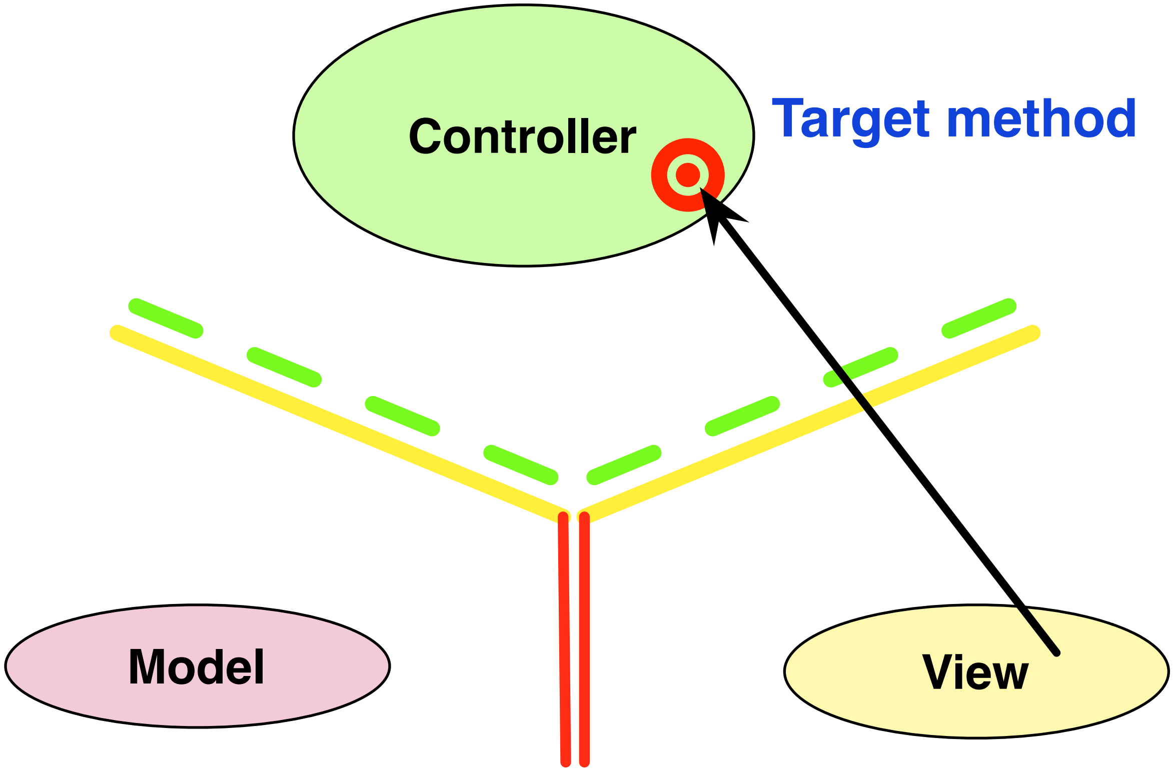
Controllers (and other models) tune into stations in which they are interested



Model has a radio station to broadcast change notifications

Models can only communicate directly with other models

Communications – View – 1 of 3

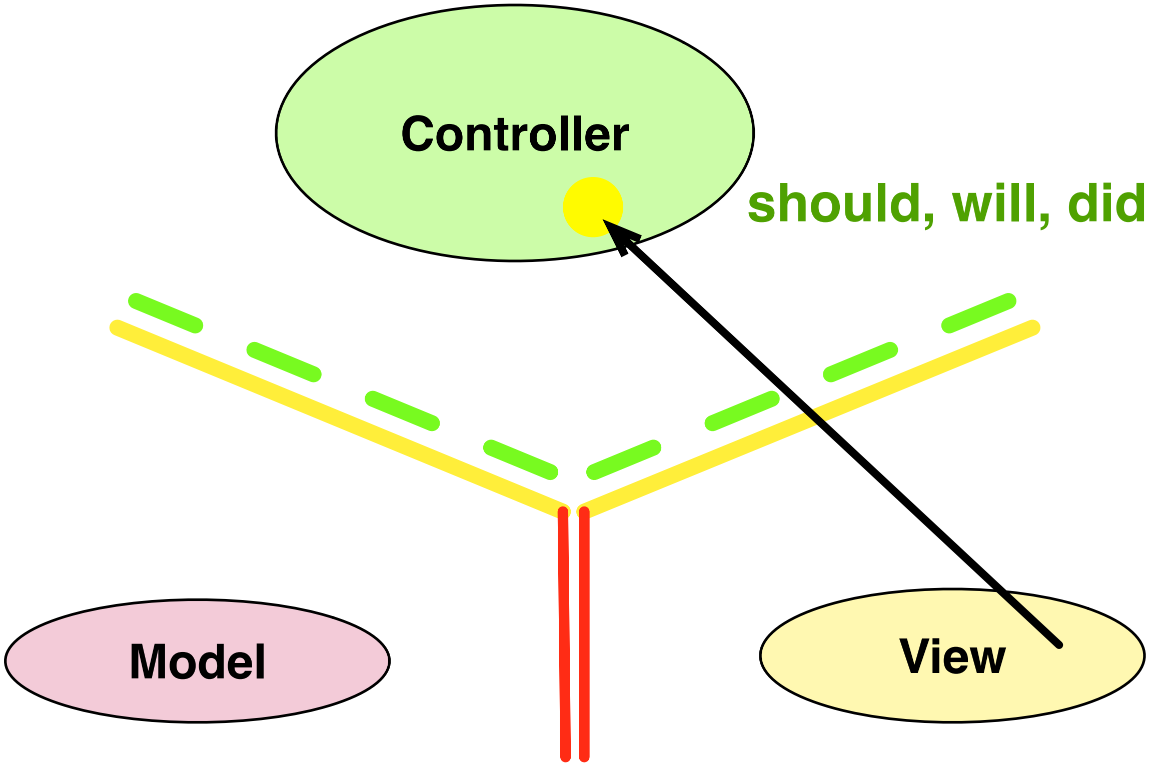


Target method

Action – button, gesture

Communications – View – 2 of 3

Controller sets itself as view's delegate



Controller

should, will, did

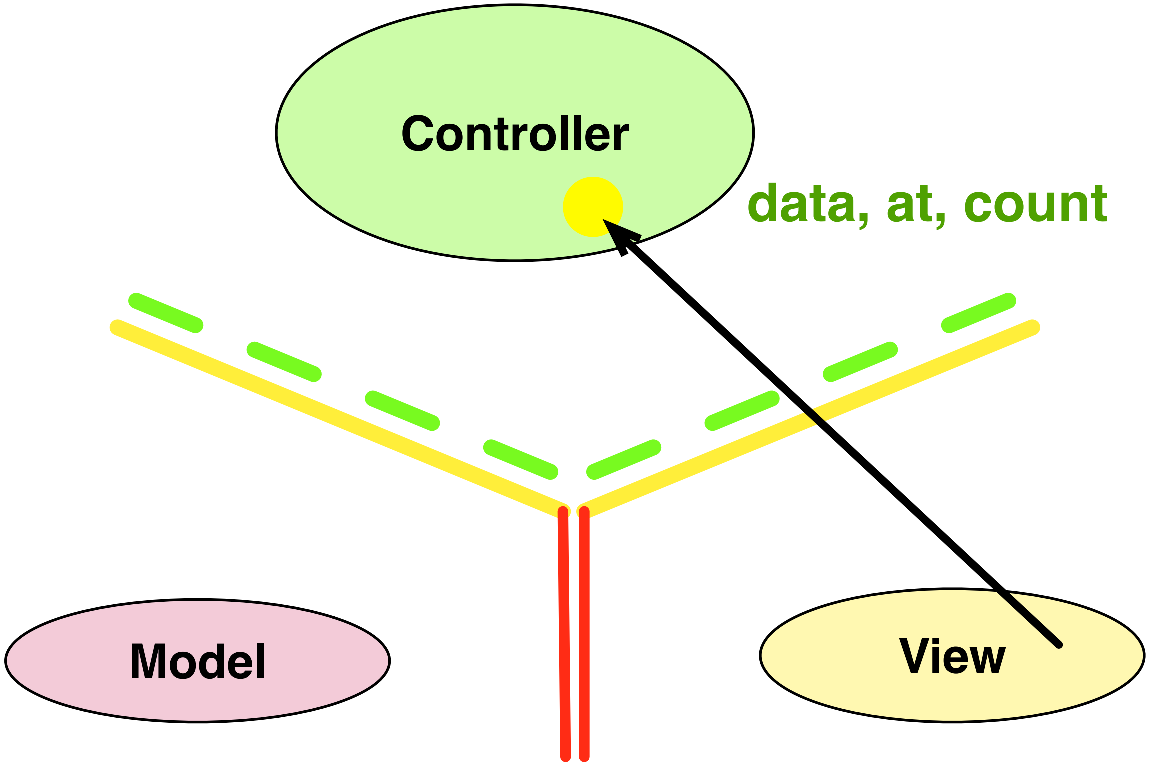
Model

View

View asks controller to do work, indirectly as its delegate, and notifies the controller when the view changes

Communications – View – 3 of 3

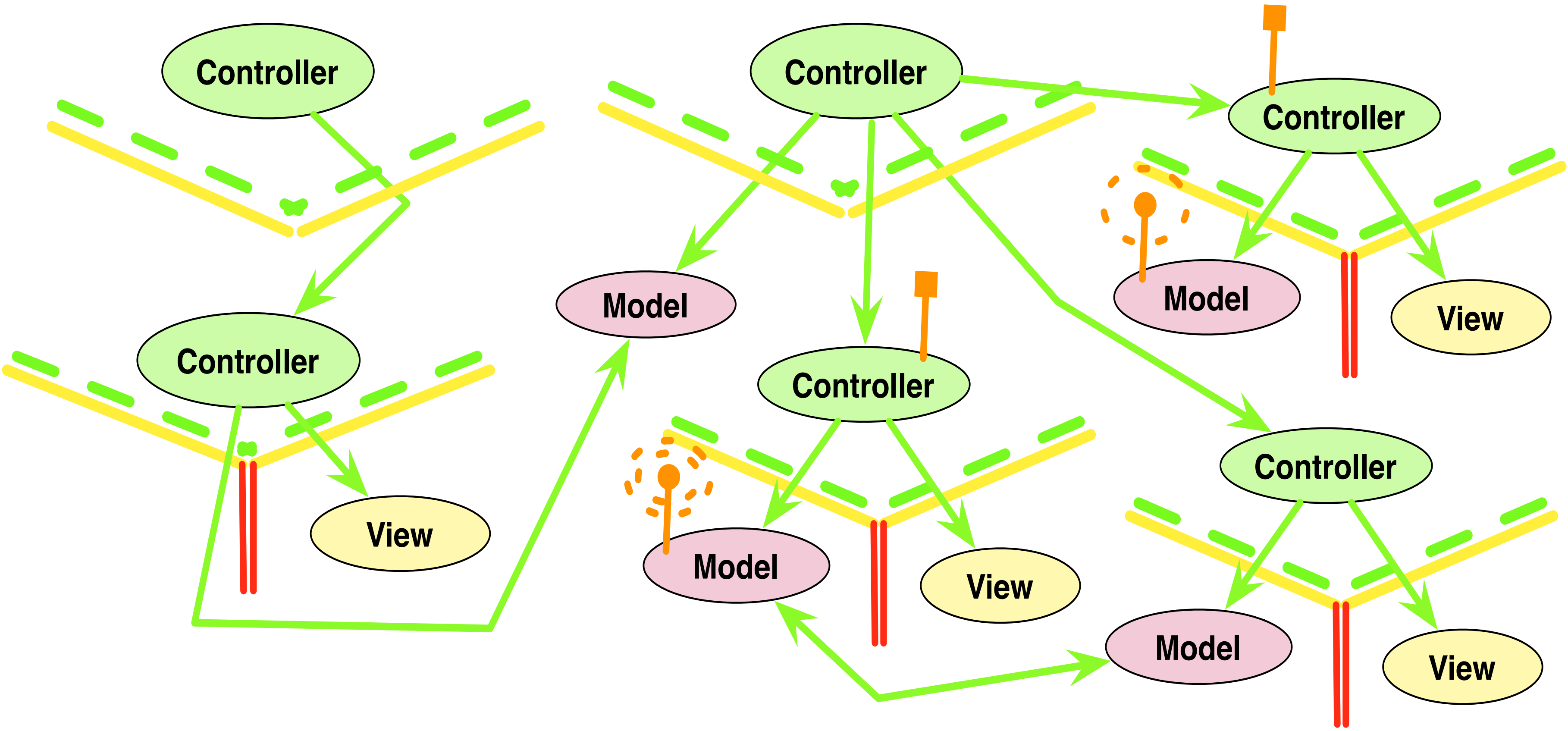
Controller sets itself as view’s data source



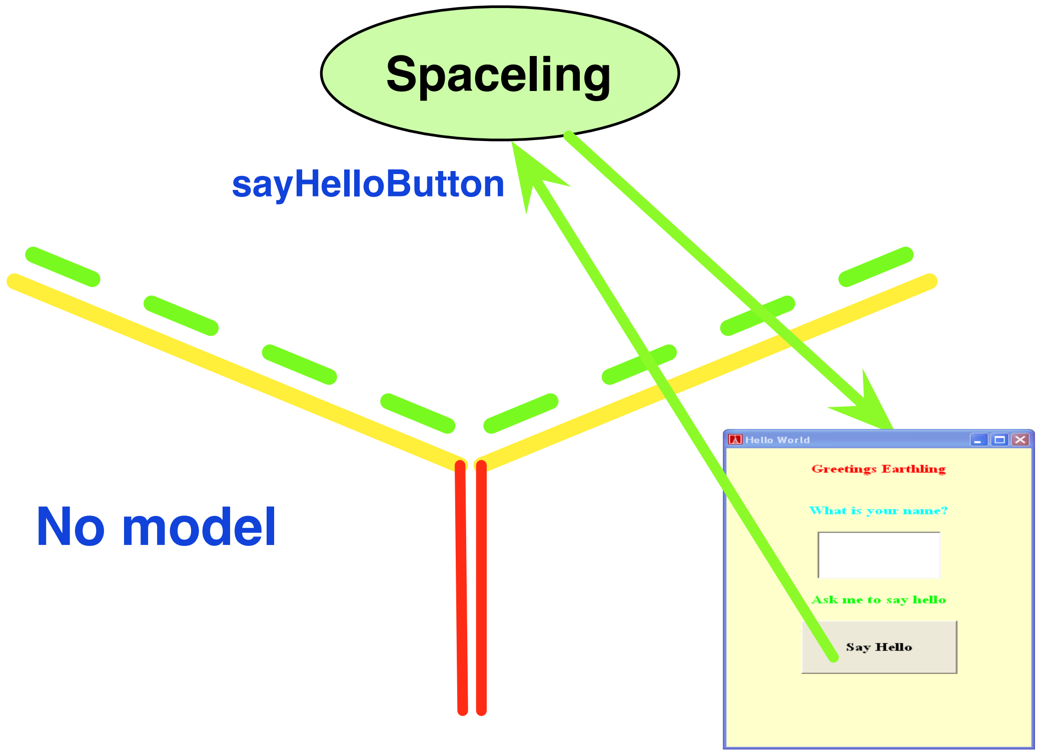
data, at, count

View asks controller for the data it needs

Multiple MVC patterns



Hello World Example

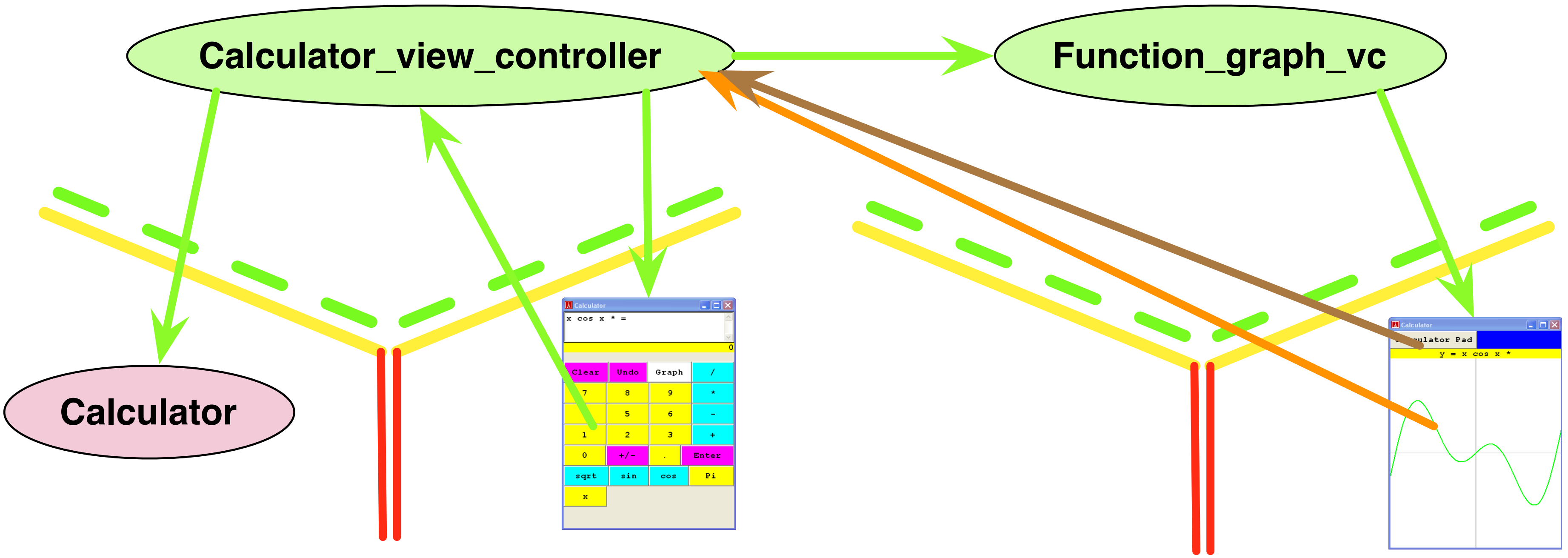


Spaceling

sayHelloButton

No model

Calculator Example



← delegate for “back” button

← data source

If the calculator has the data source protocol expected by the graphing function, then it could be the data source

Participants

- Model

  Has all the data and methods for maintaining the data

- View

  Displays relevant data, accepts user actions and notifies controller

- Controller

  » **Reacts to changes in the model to instruct how the view is to change**

  » **Reacts to view “actions” to instruct the view how to change and to instruct how the model is to change**

Applicability

- Use MVC in all interactive applications

Consequences

- The accepted way to deal with interactive applications

Related Patterns

- MVC is a special case of the Mediator pattern
  - » **The controller is the mediator between the model and the view**
- MVC has similarities to the Observer pattern.