Material and some slide content from:
- Krzysztof Czarnecki
- Ian Sommerville

MVC / MVP

Reid Holmes

Lecture 19 - Thursday November 18 2010.

Background

- MVC started w/ Smalltalk-80
- Java UI frameworks & EJBs have brought it back
- Also prevalent in GWT and .NET development.

MVC Motivation

- ?
- The same data is often displayed in different ways.
  - e.g., table view vs chart view.
  - The same business logic can drive both.
- ?
- ?
- Main Goal: Decouple models and views.
  - Increase maintainability/testability of system.

Model

- Contains application data.
  - This is often persisted to a backing store.
- ?
- Is domain independent.
- Are often Subjects in the Observer pattern.

View

- Presents the model to the user.
- ?
- Does not store data.
- Is configurable to display different data.

Controller

- Glues Model and View together.
- Updates the view when the Model changes.
- Updates the model when the user manipulates the view.
- Houses the application logic.
- Loose coupling between Model and others.
- ?

Abstract topology

Concrete topology

Interaction mechanism

- User interacts with the UI (View).
- UI (View) notifies controller of changes.
- ?
- Controller modifies the model as required.
- If the model changes, it fires modification events.
- ?

Benefits and tradeoffs

- Pro:
  - Decouple view from model
    - Support multiple views
    - ?
    - Split teams
  - Testability
- Con:
  - ?
  - Efficiency

MVP Motivation

- Take MVC a tiny bit further:
  - Enhance testability.
  - Further separate Designers from Developers.
- Leveraged by both GWT and .NET.

Model

- Contains application data.
  - This is often persisted to a backing store.
- Does not know how to present itself.
- Is domain independent.
- ?

View

- ?
- Does not store data.
- Can be interchanged easily.
- Does not ever see or manipulate Model objects.
- Only interacts with primitives.
  - e.g., (setUser(String) instead of setUser(User))

Controller

- Glues Model and View together.
- Updates the view when the Model changes.
- Updates the model when the user manipulates the view.
- Houses the application logic.

MVP Topology

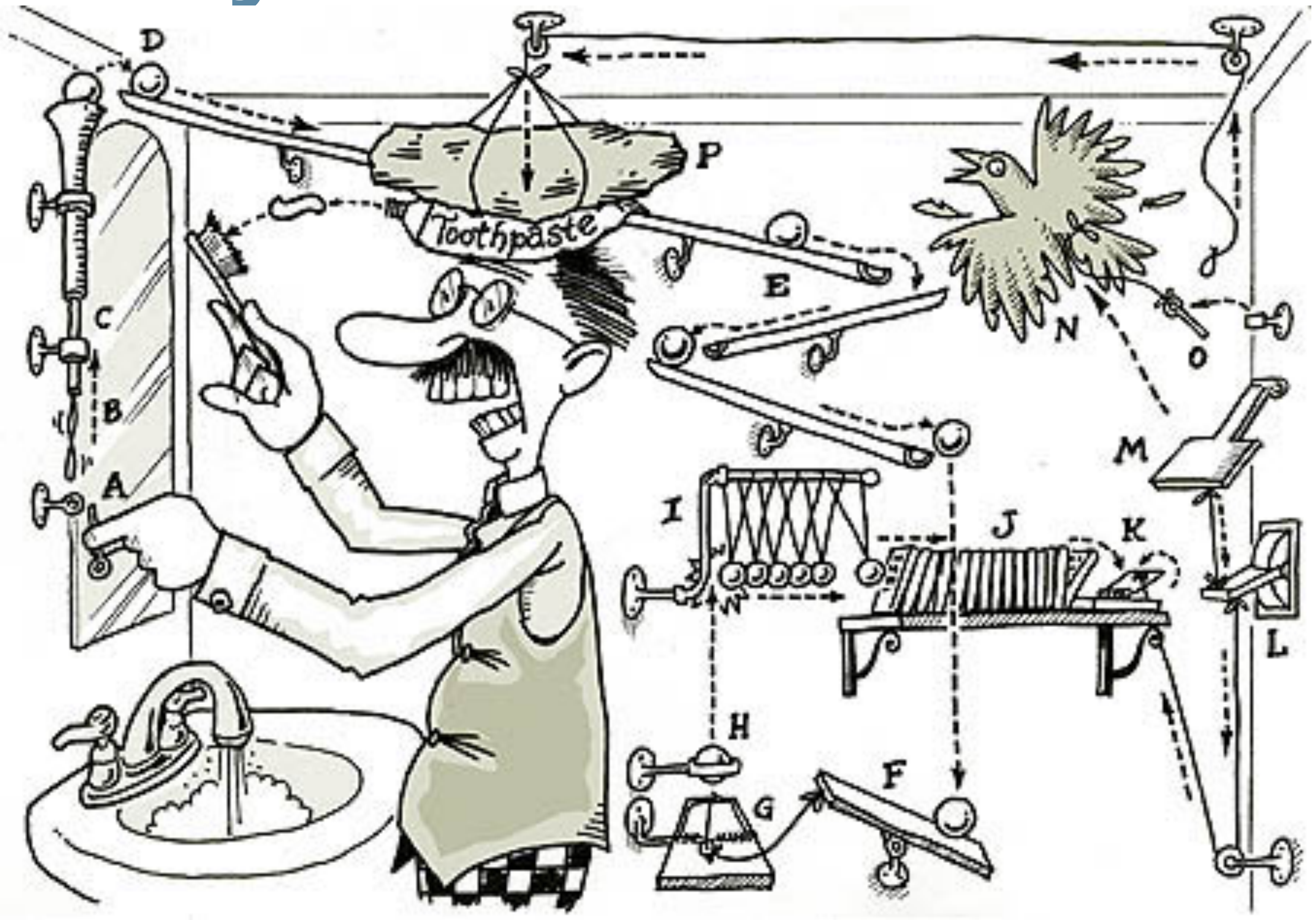
Concrete MVP Topology

Concrete Example

Benefits and tradeoffs

- Same as MVC with improved:
  - Decoupling of views from the model
    - Split teams
  - Testability
  - A little less complex than MVC

Activity



- Apply MV<C|P> to your system.
- Identify your events and interfaces.
- Groups will present to each other in 15 minutes.