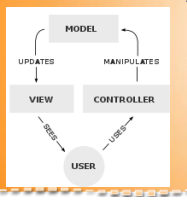
User Interfaces Software Architecture



Paulo Dias, Beatriz Sousa Santos

deti departamento de electrónica telecomunicações e informática
universidade de aveiro

1

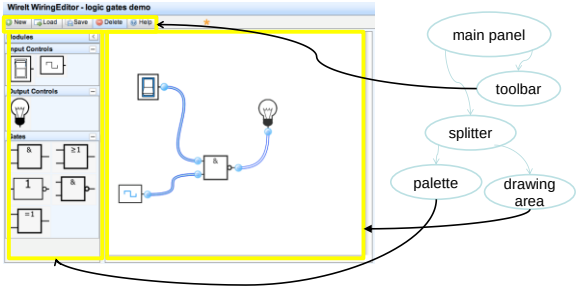
Outline

- Design patterns for GUIs
  - View tree
  - Listener
  - Widget
  - Model-view-controller

2

View Tree

- A GUI is structured as a tree of views
  - A view is an object that displays itself on a region of the screen



3

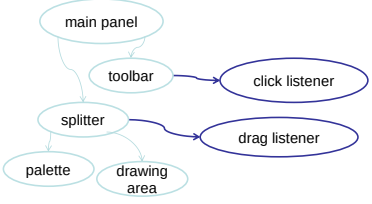
Main elements of the View Tree

- Output
  - GUIs change their output by **mutating** the view tree
  - A redraw algorithm automatically redraws the affected views
- Input
  - GUIs receive keyboard and mouse input by attaching listeners to views
- Layout
  - Automatic layout algorithm traverses the tree to calculate positions and sizes of views

4

Input Handling

- Input handlers are associated with views
  - Also called **listeners**, event handlers, subscribers, observers



5

Listener Pattern

- GUI input handling is an example of the Listener pattern
  - aka Publish-Subscribe, Event, Observer
- An event source generates a stream of discrete events
  - e.g., mouse events
- Listeners register interest in events from the source
  - Can often register only for specific events – e.g., only want mouse events occurring inside a view's bounds
  - Listeners can unsubscribe when they no longer want events
- When an event occurs, the event source distributes it to all interested listeners

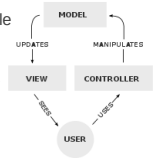
6

Separating Frontend from Backend

- Input and output in GUIs are separated
  - Output is represented by the view tree
  - Input is handled by listeners attached to views
- Missing piece is the backend of the system
  - Backend (aka **model**) represents the actual data that the user interface is showing and editing
  - Why do we want to separate this from the user interface?

Model–view–controller (MVC)

- MVC is an architectural pattern commonly used for developing user interfaces
- It expresses the "core of the solution" to a problem while allowing it to be adapted for each system
- It divides an application into three interconnected parts
  - Model
  - View
  - Controller



- Separating internal representations of information from the ways it is presented to and accepted from the user
- It decouples these major components allowing for efficient code reuse and parallel development.

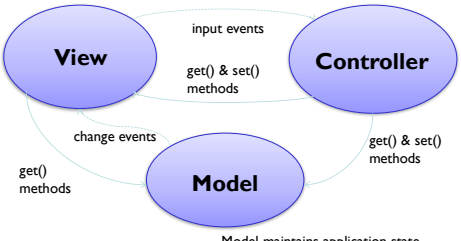
MVC Components

- The ***model*** is the central component of the pattern
  - expresses the application's behavior in terms of the problem domain, independent of the UI. It directly manages the data, logic and rules of the application
- A ***view*** can be any output representation of information
  - Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants
- The ***controller*** accepts input and converts it to commands for the model or view

Model-View-Controller Pattern

View handles output
• gets data from the model to display it
• listens for model changes and updates display

Controller handles input
• listens for input events on the view tree
• calls mutators on model or view

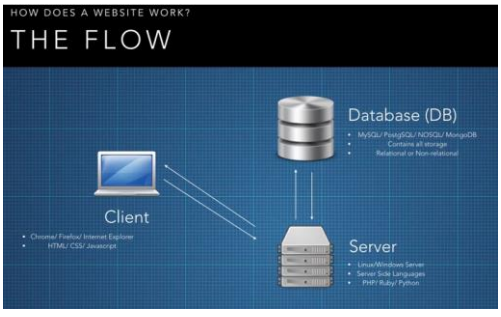


Model maintains application state
• implements state-changing behavior
• sends change events to views

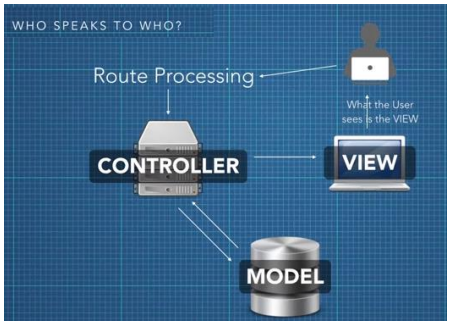
Model–view–controller (MVC)

- Advantages
  - Separation of responsibilities
    - Model: data
    - View: output
    - Controller: input
  - Decoupling
    - Views and models can be changed independently, reused or share models
    - Multiple views for a shared model
    - Simultaneous development
  - Ease of modification
- Disadvantages
  - Code navigability
  - Multi-artifact consistency
  - Pronounced learning curve

Model–view–controller (MVC)



Model–view–controller (MVC)



https://www.youtube.com/watch?v=1IsL6g2ixak

15

Widget: Tightly Coupled View & Controller

- The MVC idea has largely been superseded by a MV (Model-View) idea
- A widget is a reusable view object that manages both its output and its input
  - Widgets are sometimes called components (Java) or controls (Windows)
- Examples: scrollbar, button, menubar

16