LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
FAKULTÄT FÜR MATHEMATIK, INFORMATIK UND STATISTIK
INSTITUT FÜR INFORMATIK
LEHRSTUHL FÜR DATENBANKSYSTEME UND DATA MINING

MVVM (Model View ViewModel) in JavaFX

SEP Sommersemester 2019

Nicolas Brauner

30.04.2019

DBS

Wissenschaftlicher Betreuer:
Maximilian Hünemörder, Ludwig Zellner
Verantwortlicher Professor:
Prof. Dr. Peer Kröger

Übersicht

1. Motivation: Design Patterns
2. MVC (Model View Controller)
3. MVVM (Model View ViewModel)
    1. Definition
    2. MVVM in JavaFX

Motivation: Design Patterns

Was sind Design Patterns?

Entwurfsmuster (englisch design patterns) sind bewährte Lösungsschablonen für wiederkehrende Entwurfsprobleme [...].
Sie stellen damit eine wiederverwendbare Vorlage zur Problemlösung dar, die in einem bestimmten Zusammenhang einsetzbar ist.

(Quelle: Wikipedia, Entwurfsmuster)

Motivation: Design Patterns

Entstehung:

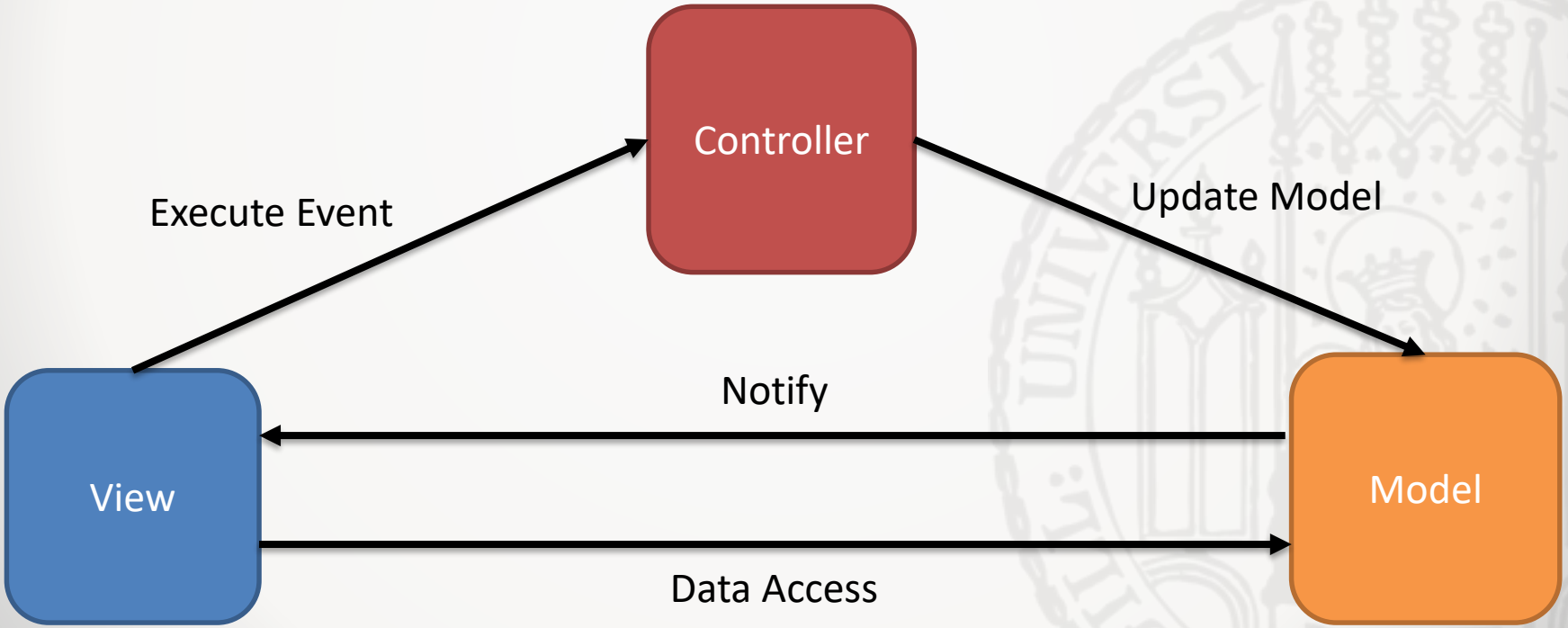
- Ursprünglich aus der Architektur
- Später: Für Erstellung grafischer Benutzeroberflächen mit Smalltalk
- Verbreitung durch Promotion Erich Gammas
- Publikation des Buches *Design Patterns – Elements of Reusable Object-Oriented Software* zusammen mit Richard Helm, Ralph Johnson, John Vlissides (Gang of Four)

Motivation: Design Patterns

- Vier Elemente eines Design Patterns:
  - 1. Pattern Name
  - 2. Problem
  - 3. Solution
  - 4. Consequences
- Drei Arten von Design Patterns:
  - Creational Patterns (Erzeugungsmuster)
  - Structural Patterns (Strukturmuster)
  - Behavioral Patterns (Verhaltensmuster)

MVC (Model View Controller)

- Trennung zwischen Datenmodell, Präsentation und Programmsteuerung
- Entwurfs- und/oder Architekturmuster



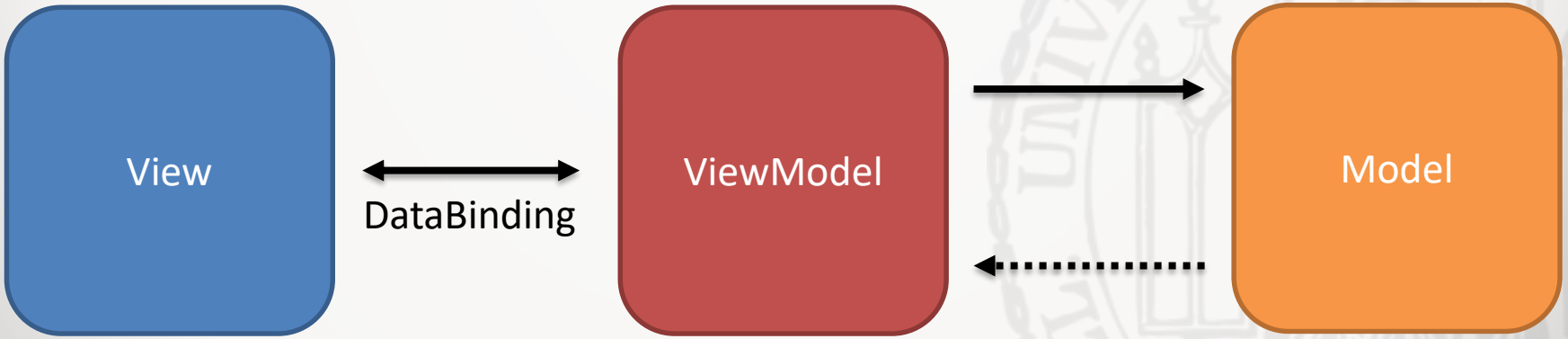
Probleme mit MVC

- Heutige Systeme sind viel komplexer als damalige -> MVC Character lässt sich auf verschiedenen Ebenen erkennen, aber unklare Beschreibung.
- Nicht immer klar trennbar: Model kann Geschäftslogik halten, Controller kann mehrere Modelle halten, View kann Felder validieren etc.

➔ Ausarbeitung spezifischerer Designpatterns wie **Model View ViewModel (MVVM)**

MVVM (Model View ViewModel)

- Trennung zwischen Ansicht, Daten und deren Verarbeitung wie bei MVC
- JavaFX arbeitet speziell mit MVVM (aufgrund des Datenbindungsmechanismus)



MVVM

- Im Gegensatz zum Controller in MVC, steuert ViewModel nicht die View, sondern stellt lediglich die Daten dafür bereit (Databinding)
- MVVM ist technologispezifisch aufgrund des *Binders*
- Auch GUI Unit Tests sind mit MVVM möglich
- Die View benötigt keinen hinterlegten Code

MVVM in JavaFX

```
StringProperty a = new
SimpleStringProperty();
StringProperty b = new
SimpleStringProperty();

a.bindBidirectional(b);

a.set("Hallo");

System.out.println(b.get());
// "Hallo"

b.set("Welt");

System.out.println(a.get());
// "Welt"
```

```
public class MyViewModel {
    private StringProperty labelText =
            new SimpleStringProperty("default");
    private StringProperty inputText =
            new SimpleStringProperty();
    private BooleanProperty buttonDisabled =
            new SimpleBooleanProperty();

    public MyViewModel() {
        buttonDisabled.bind(inputText.isEmpty());
    }

    public void changeLabel() {
        labelText.set(inputText.get());
        inputText.set("");
    }

    // getter, setter, property-accessors
}
```

MVVM in JavaFX

```
public class MyView {
    @FXML
    Label label;
    @FXML
    TextField input;
    @FXML
    Button button;

    MyViewModel viewModel = new MyViewModel();

    public void initialize() {
         label.textProperty().bind(viewModel.labelTextProperty());
        input.textProperty()
            .bindBidirectional(viewModel.inputProperty());

        button.disableProperty()
            .bind(viewModel.buttonDisabledProperty());
    }

    // will be called when button is pressed
    public void onAction() {
        viewModel.changeLabel();
    }
}
```

MVVM in JavaFX

- Views lassen sich als FXMLs erstellen und mittels *Property Binding* mit der Geschäftslogik (also ViewModel und Model) verbinden (uni- oder bi-direktional)
- Einbinden eines Controllers/ViewModels in fxml File

MVVM in JavaFX

```
@Override
public void start(Stage primaryStage) throws Exception{
        //create a root and load fxml code
        Parent root = FXMLLoader.load(getClass().getResource("views/HelloView.fxml"));
        //add root to the window and show it
        primaryStage.setTitle("Hello World");
        primaryStage.setScene(new Scene(root));
        primaryStage.show();
 }
```

MVVM in JavaFX

```
<GridPane alignment="center … fx:controller="sample.viewmodels.HelloViewModel">
        <columnConstraints>
                <ColumnConstraints hgrow="ALWAYS" … />
        </columnConstraints>
        <rowConstraints>
                <RowConstraints percentHeight="25.0" vgrow="ALWAYS" />
                …
        </rowConstraints>
        <children>
                <Label fx:id="messageLabel" …>
                        <font> <Font size="25.0" /> </font>
                </Label>
                <Label fx:id="messageLabelTwo“ …>
                        <font> <Font size="25.0" /> </font>
                </Label>
                </Button>
        </children>
</GridPane>
```

Vielen Dank für eure Aufmerksamkeit!