


MVVM

Prof. Dr.-Ing. Andreas Heil

 Licensed under a Creative Commons Attribution 4.0 International license. Icons by The Noun Project.

v1.0.0

Lernziele

Typische Web-Anwendungen

Die meisten Anwendungen sind nach einer Schichtenarchitektur aufgebaut:

- Die Benutzungsschnittstelle (View)
- Daten, die angezeigt und manipuliert werden (Data)
- Die Anwendungslogik, die das Verhalten der Anwendung ausmacht (Logic)

Entwurfsmuster versuchen

- die Schichten zu entkoppeln
- die Anwendung möglichst flexibel zu gestalten

Model-View-Controller (MVC)

- Entstand bereits in den 1980ern in den Anwendungen von Xerox (basierend auf Smalltalk)
- Ein sog. Controller verbindet ein View und das darunterliegende Model
- Der View nutzt das Model um die Ausgabe zu erzeugen
- Das Model enthält die Informationen
- Ein Model kann Ereignisse (Events) versenden, wenn sie Eigenschaften des Models ändern
- Die Events können sowohl vom Controller aber auch vom View genutzt werden

Model-View-Controller

- Unterschiedliche Aspekte der Anwendung werden getrennt (Separation of Concerns)
- Implementierungen weichen voneinander ab

Separation of Concerns - Vorteile

- Wird für Entkopplung benötigt
- Es kann mehrere Darstellungen geben (Desktop, Web, Mobil)
- Unterschiedliche Entwickler könnten sich um unterschiedliche Teile kümmern

Model

- Enthält die Daten
- Regelt den Zugriff und wann Änderungen stattfinden
- Basiert meist auf Objekten der realen Welt

View

- Darstellung des Inhaltes eines Models
- Zugriff auf die Daten (nur) durch das Model
- Darstellung des Models obliegt vollständig dem View

Controller

- Übersetzt Interaktionen mit dem View in entsprechende Aktionen
- Ausführung der Aktionen durch das Model
- Desktop-Anwendung: Maus-Klick; Web-Anwendung: HTTP-Request
- Aktionen können sowohl Geschäftsprozesse auslösen aber zu Statusänderungen im Model führen (vgl. HATEOS)
- Abhängig von Benutzerinteraktion und dem Ergebnis der Aktionen des Models stellt der Controller einen passenden View dar

Web Anwendungen

- View und Controller sind hier auf Client und Server verteilt
- Serverseitig wird ein sog. *Router* benötigt, um Anfragen auf den entsprechenden Controller weiterzuleiten
- Im Controller wird dann die entsprechende *Action* ausgeführt
- Abhängig vom Request Request wird dann das Model aktualisiert (vgl. HATEOS)
- Das Ergebnis wird in Form eines Views (hier HTML) angezeigt

s. Code

Referenzen

ToDElete

[https://patterns.florian-rappl.de/Category/Presentation patterns?full#slide-aa27a0a7-fb19-4ff0-850d-42fda0ca7ca9](https://patterns.florian-rappl.de/Category/Presentation%20patterns?full#slide-aa27a0a7-fb19-4ff0-850d-42fda0ca7ca9)