

---

*Einführung in die Informatik*

---

Modelle und Schichten

Meik Teßmer

---

# Inhalt der Veranstaltung

Lernziele:

- ▶ *Schichtung* als grundlegendes Organisationskonzept
- ▶ Erläuterung des Modellbegriffs
- ▶ Beispiele

# Motivation

*Wie könnte man eine Kommunikationsplattform wie Facebook/Twitter/Instagram aufbauen?*

Gibt es erste Ideen?

# Modell

Was ist ein Modell?

# Modell



Figure 1: Modell? (Quelle: Wikipedia)

# Modell

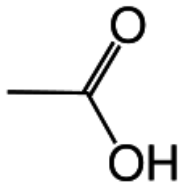


Figure 2: Auch Modell? (Quelle: Wikipedia)

# Modell

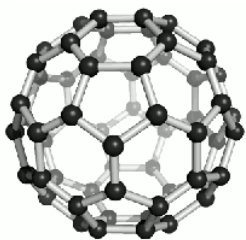


Figure 3: Und das? (Quelle: Wikipedia)

# Modell

Hauptgebäude

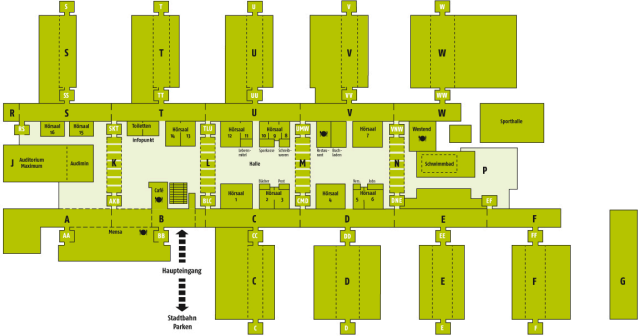


Figure 4: Auch das? (Quelle: Uni Bielefeld)



# Modell

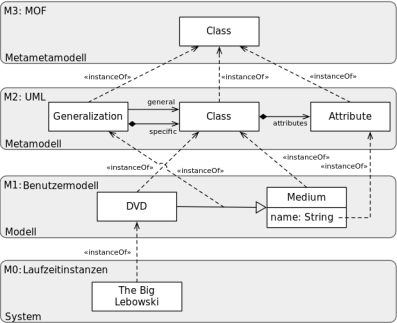


Figure 5: Und hier? (Quelle: Wikipedia)

# Modell

eine mögliche Definition:

*Ein Modell ist ein Abbild eines Ausschnitts der Realität.*

# Modelle - Sprachwelten

Beispiel: Mathematik

- ▶ Formeln:
  - ▶ eigene Zeichen
  - ▶ eigene Regeln
  - ▶ Fokus: Berechenbarkeit, Logik

Beispiel: Softwareentwicklung

- ▶ Unified Modeling Language (UML<sup>1</sup>)
  - ▶ verschiedene Diagrammtypen
  - ▶ eigene Symbole und Verknüpfungsregeln
  - ▶ Fokus: Erfassen von Anforderungen, Übersicht über die Systemstruktur usw.

---

<sup>1</sup>zu UML siehe auch Rupp, Queins, and SOPHISTen (2012)

# Konkretes Beispiel

unser erstes eigenes Modell: unser Computer



Figure 6: Computer

# Beispiel Variante 1

Ein Computer ist ein Tupel  $C$ :

$$C = (B, G, OS, A, D)$$

mit

- ▶  $B$ : Menge der Benutzer
- ▶  $G$ : Gerät
- ▶  $OS$ : Betriebssystem
- ▶  $A$ : Menge der Anwendungsprogramme
- ▶  $D$ : Menge von Daten

## Beispiel Variante 1

ein Laptop:

$C_1 = (\text{Peter Schmidt, Dell Latitude E6550, GNU/Linux, } \dots, \dots)$

## Beispiel Variante 2

„bessere“ Aufgabenstellung:

*Wie werden Probleme mit Hilfe eines Computers gelöst?<sup>2</sup>*

Im Detail:

- ▶ Welche Bestandteile unseres Rechners tragen zur Problemlösung bei?
- ▶ Wie hängen sie dabei voneinander ab?
- ▶ Darstellung: tabellarisch

---

<sup>2</sup>zu Problemlösen siehe auch Pólya (1957)

## Beispiel Variante 2

Schicht	Beispiel
Benutzer	wir
Anwendung	Firefox ...
...	...
Betriebssystem	GNU/Linux ...
...	...
Gerät	Dell ...



# Entwicklung eines Modells

bisher genutzte Definition:

*Ein Modell ist ein Abbild eines Ausschnitts der Realität.*

Wie kommt man dorthin?

# Entwicklung eines Modells

## Abgrenzung

- ▶ relevant: „mein“ Rechner
- ▶ nicht relevant: andere Rechner, Wandfarbe des Arbeitszimmers, Stuhlfabrikat, Geschwindigkeit der Internetanbindung

# Entwicklung eines Modells

## Reduktion

- ▶ relevant: Anwendungsprogramme, Betriebssystem ...
- ▶ nicht relevant: Farbe des Gehäuses, Geschwindigkeit der Festplatte, Anzahl USB-Anschlüsse

# Entwicklung eines Modells

## Dekomposition

- ▶ Reduktion der Komplexität
  - ▶ Prinzip Teile-und-Herrsche
  - ▶ kann hierarchisch strukturiert sein
  - ▶ vergleichbar einem top-down-Vorgehen
- wesentliche Schwierigkeit: geschickte Zerlegung finden

# Entwicklung eines Modells

## Aggregation

- ▶ detailreiche Teilsysteme zu größeren Teilsystemen kombinieren
- ▶ Gegenstück zur Dekomposition
- ▶ vergleichbar einem bottom-up-Vorgehen

# Entwicklung eines Modells

## Abstraktion

- ▶ Klassen-/Gruppenbildung: Einteilung von Objekten in homogene Mengen
- ▶ Eigenschaften: Objekte haben Eigenschaften
- ▶ Spezialisierung/Generalisierung

# Entwicklung eines Modells

## Zusammenfassung

- ▶ Abgrenzung: nicht relevante Objekte weglassen
- ▶ Reduktion: nicht relevante Details weglassen
- ▶ Dekomposition: Zerlegen in Teilsysteme
- ▶ Aggregation: Kombinieren von Teilsystemen
- ▶ Abstraktion: Klassen-/Gruppenbildung

# Gefahren

- ▶ Gleichsetzen von Modell und Realität

konkretes Beispiel: Wirtschaftskrisen der letzten Jahre,  
Pandemien

→ BWL-/VWL-Modelle funktionieren offensichtlich nicht  
zuverlässig

→ Ausbreitung von Krankheiten sind schwer zu modellieren

- ▶ Rückübertragung von Ergebnissen/Optimierungen

- ▶ schlechtes Modell



# Einstiegsfrage

*Wie könnte man eine Kommunikationsplattform wie Facebook/Twitter/Instagram aufbauen?*

Ideen?

# Strukturierung unserer Plattform

- ▶ Zuständigkeiten: Separation of Concerns
- ▶ Hauptbestandteile:
  - ▶ Schnittstelle(n) zum Anwender (Oberfläche)
  - ▶ Anwendungslogik
  - ▶ Datenhaltung
- ▶ Regeln:
  - ▶ Kommunikation nur von oben nach unten bzw. umgekehrt
  - ▶ keine Umgehung von Schichten

# Strukturierung unserer Plattform

unsere Plattform:

---

Schicht

---

Darstellung

Anwendungslogik

Datenhaltung

---

# Charakteristika einer Schicht

- ▶ aktiv in nur einer Richtung (nach unten)
- ▶ keine Brüche
- ▶ Unabhängigkeit, solange Schnittstellen gewahrt bleiben (Kapselung)
- ▶ haben jeweils eine konkrete Aufgabe

Auswirkungen:

- ▶ geringe Kopplung
- ▶ Wiederverwendbarkeit
- ▶ Wartbarkeit und Systemverstehen

# Probleme

- ▶ Performanceverlust (~10% pro Schicht)
- ▶ oftmals nur simples Durchreichen von Daten (Reports)

# Konkretes Beispiel

## Zwei-Schicht-Architektur

Table 4: Ein Office-Paket

Schicht	Ausprägung
Anwendungsschicht	Office
Datenschicht	Dateisystem

# Konkretes Beispiel

## Drei-Schicht-Architektur

Table 5: eKVV

Schicht	Ausprägung
Präsentationsschicht	Webbrowser
Logikschicht	Appl.-Server
Datenschicht	Datenbank

# Konkretes Beispiel: OSI

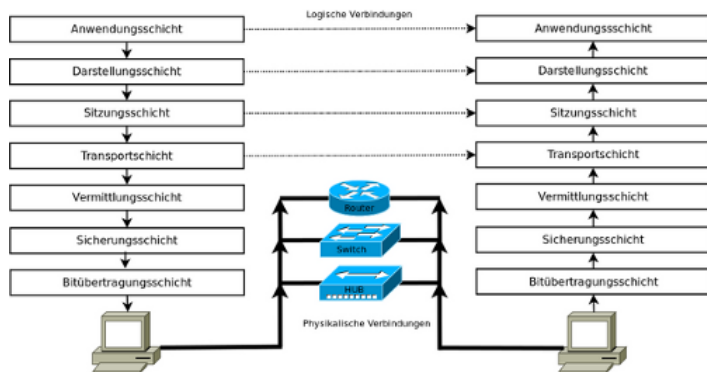


Figure 7: OSI-Referenzmodell



# Konkretes Beispiel: TCP/IP

OSI-Schicht	TCP/IP-Schicht	Beispiel
Anwendungen (7)	Anwendungen	HTTP, UDS, FTP, SMTP, POP, Telnet, OPC UA
Darstellung (6)		
Sitzung (5)		SOCKS
Transport (4)	Transport	TCP, UDP, SCTP
Vermittlung (3)	Internet	IP (IPv4, IPv6)
Sicherung (2)	Netzzugang	Ethernet, Token Bus, Token Ring, FDDI, IPoAC
Bitübertragung (1)		

Figure 8: TCP/IP-Modell

# Schichten in Python

- ▶ ganz unten: Anweisungen, Bedingungen, Schleifen
- ▶ darüber: Funktionen, die Anweisungen zusammenfassen
- ▶ darüber: Funktionen, die Funktionen aufrufen usw.

Code der Datenhaltungsschicht:

```
def lade_daten():  
    ...  
  
def speichere_daten(neue_daten):  
    ...
```

→ Schnittstelle: `lade_daten` und `speichere_daten`

# Schichten in Python

konkreter Code:

```
def lade_daten():
    handle = open("meine_daten.txt", "r")
    daten = []
    for zeile in handle:
        daten.append(zeile)
    handle.close()
    return daten

def speichere_daten(neue_daten):
    handle = open("meine_daten.txt", "w")
    for zeile in neue_daten:
        handle.write(zeile)
    handle.close()
```

# Ende

## Literatur

- Pólya, George. 1957. *How to Solve It: A New Aspect of Mathematical Method*. 2nd ed. Garden City, New York: Doubleday Anchor Books, Doubleday & Company, Inc.
- Rupp, Chris, Stefan Queins, and die SOPHISTen. 2012. *UML 2 Glaskar: Praxiswissen Für Die UML-Modellierung*. 4. ed. München Wien: Carl Hanser Verlag.