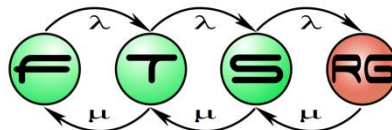


# Strukturelle Modellierung

**Budapesti Műszaki és Gazdaságtudományi Egyetem**  
**Hibatűrő Rendszerek Kutatócsoport**



# Praktische Informationen

- Bunte Stifte für die Übung!
- Skriptum : <http://docs.inf.mit.bme.hu/remo-jegyzet/>
  - Definitionen, Beispiele
  - Auch elektronisch lesbar (auf dem Tablet, Handy, ...)

## RENDSZERMODELLEZÉS

## Struktúra alapú modellezés

### 3.1. A struktúramodellezés alkalmazásai

Mind a természetben, mind a mesterséges rendszerekben fellelhetők bizonyos szabályszerűségek. Egyes szabályszerűségek a rendszer elemei közötti kapcsolatokat, míg mások magukat az elemeket jellemzik.

#### 3.1.1. Hálózatok

Egy rendszert gyakran úgy jellemezhetünk a legjobban, ha bizonyos elemeit megkülönböztetjük és leírjuk az ezek közötti *kapcsolatot*.

**Példa.** Egy nagyváros közlekedési hálózata szövevényes rendszere az út- és sínhálózatnak, a több százvezernyi járműnek és az ezeken utazó embereknek. A közlekedők mozgása mellett az infrastruktúra is folyamatosan változik a különböző fejlesztések, átalakítások és karbantartások miatt.

# Themen dieser Vorlesung

- Ziel und Anwendungen der strukturellen Modellierung
- Dekomposition
- Beschreibung der Modellelemente mit Graphen
- Eigenschaftsmodellierung

Ziele,  
Anwendungen

Dekomposition

Graphen

Eigenschaften

# STATISCHE MODELLIERUNG

*Wie kann die Aufbau eines komplexen Systems  
übersichtlich modelliert werden?*

Beispiele:

- Architektur– Gebäude
- Firmennetzwerk

# Definition: Strukturelles Modell

Das **strukturelle Modell** ist statisch. Sein Basis ist ein (Teil-) System, das durch die Relation „**Teil von**“ auf seine Bestandteile aufgeteilt wird.

Die **Bestandteile** können die folgenden sein:

- weiter aufgeteilte **Teilsysteme** oder
- weiter nicht aufgeteilte (*elementare*) **Komponenten**.

Das strukturelle Modell repräsentiert die Aufbau (Struktur) des Systems entsprechend

- dessen Bestandteilen,
- den **Eigenschaften** der Bestandteile und
- ihren **Verhältnissen** untereinander.

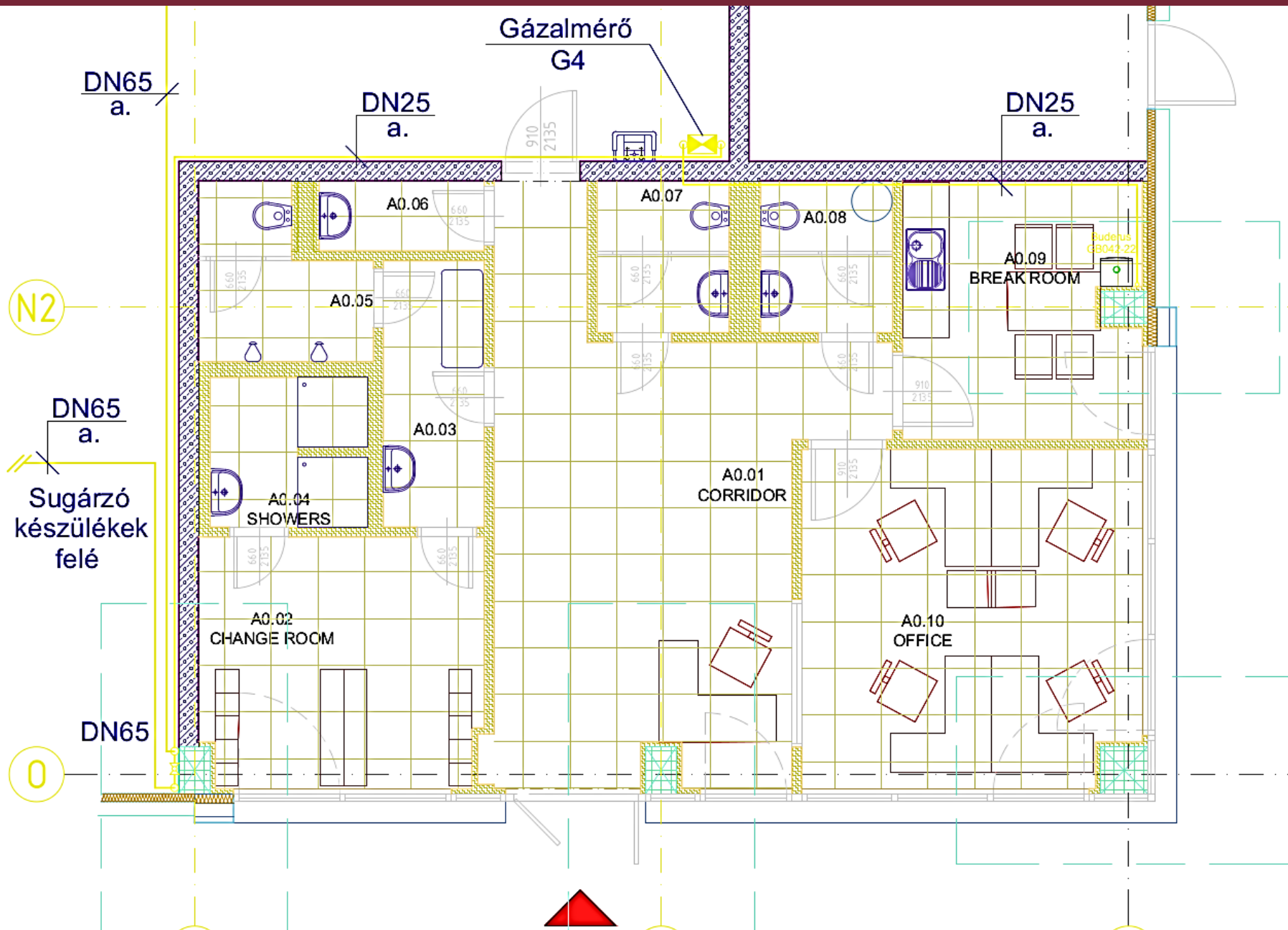
# Beispiel: Architektonische Modelle

- BIM (Building Information Model)
- Modellierung von Gebäuden in einem gemeinsamen Modell
- Ansichten
  - Gasleitungen
  - Isolierung
  - Wasserleitungen und Kanäle
  - Elektroinstallationsplan
  - usw.

# Büro einer Halle

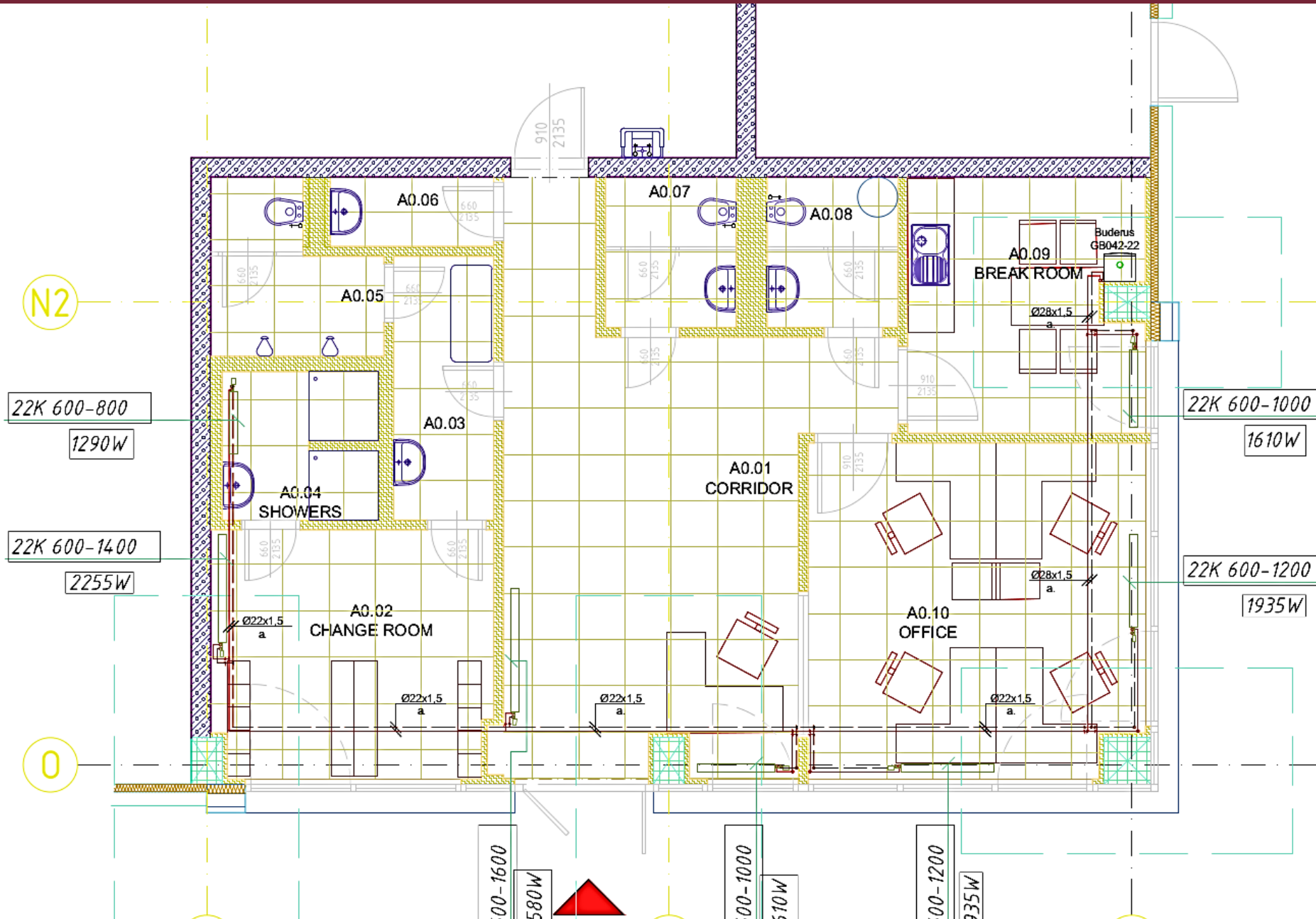


# Gasleitungen

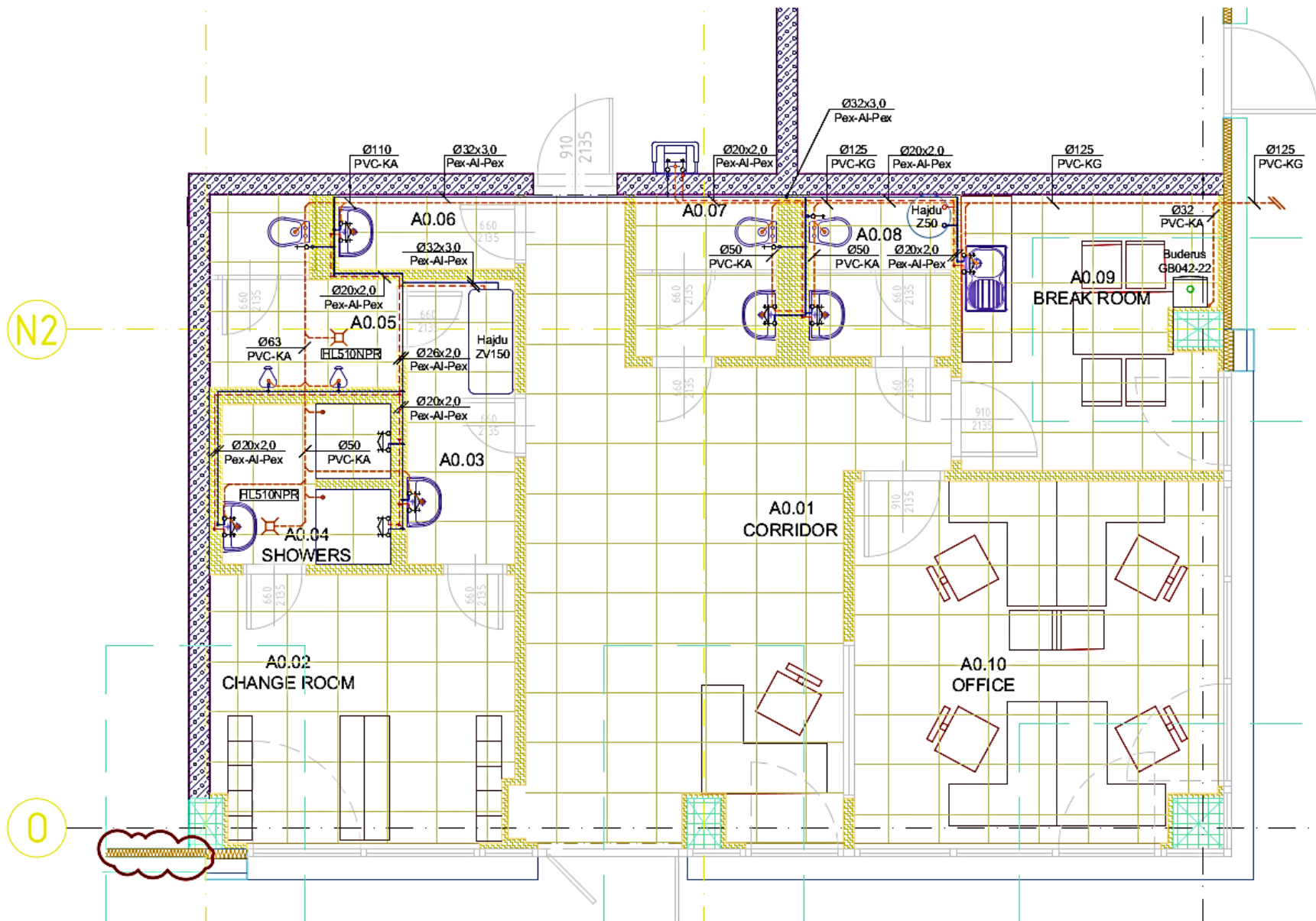




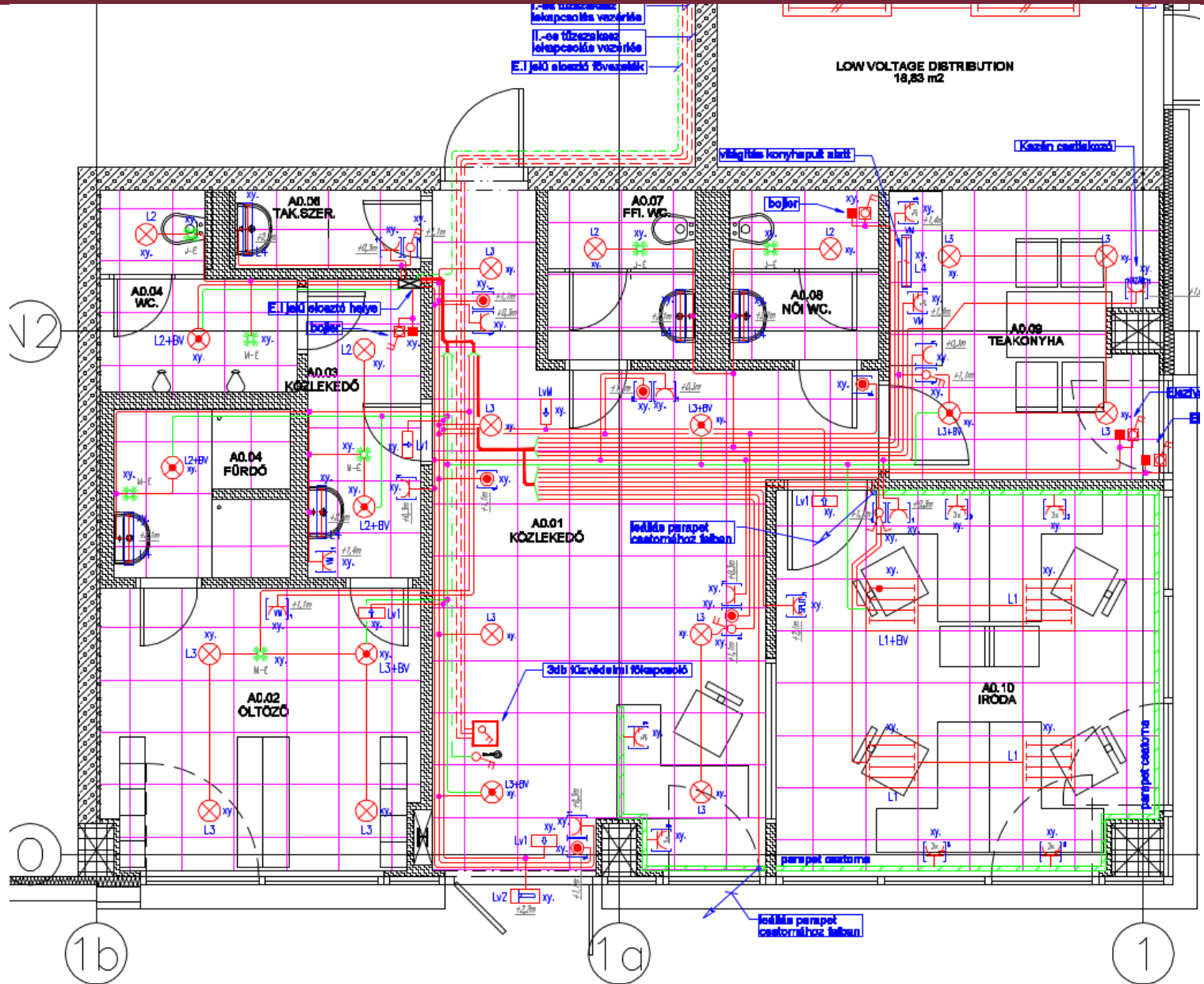
# Isolierung



# Wasserleitungen und Kanäle



# Elektroinstallationsplan

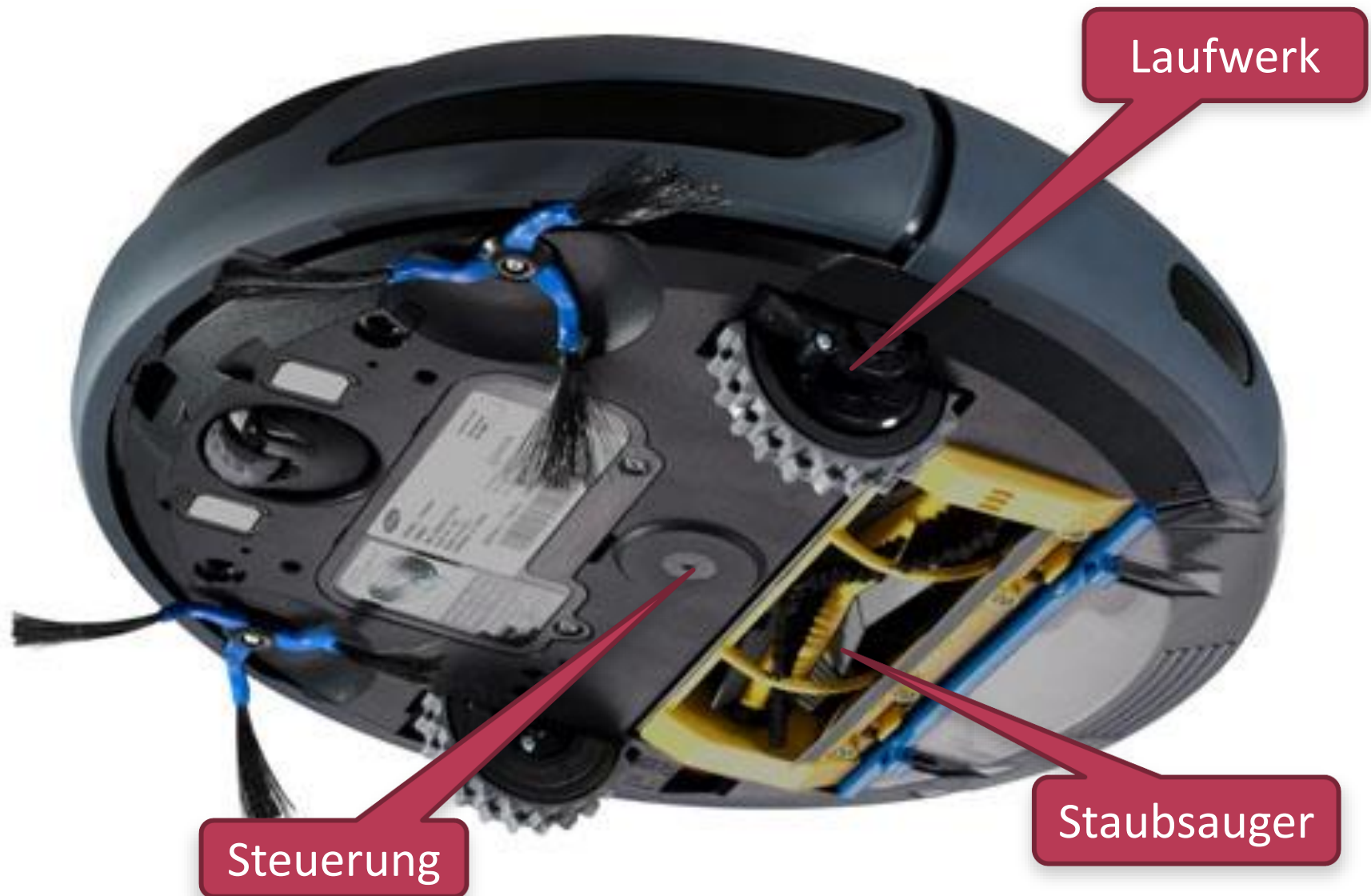


# Ziel der strukturellen Modellierung

- Aufteilung des Systems
  - Es ist einfacher, kleinere Einheiten zu entwerfen
  - Gebrauch bestehender Teileinheiten
  - Gebrauch allgemeiner Komponenten (COTS - Commercial off-the-shelf)
- Dokumentation existierender Systeme
  - „Systemübersicht“
- Erstellung der Datenstruktur
  - Welche Informationen werden behandelt?



# Beispiel: Roboterstaubsauger



# Strukturelle und Verhaltensmodellierung

## ■ Strukturelle (*structural*)

- statisch
- Teil und Ganzheit, Bestandteile
- Verhältnisse, Verbindungen

Die größere Bestandteile des Roboter-Staubsaugers sind das Steuerwerk, das Laufwerk und der Staubsauger.

## ■ Verhalten (*behavioral*)

- dynamisch
- zeitlicher Verlauf
- Zustände, Prozesse
- Reaktionen auf die Außenwelt

Auf dem Befehl „rechts“ wechselt das Laufwerk seine Betriebsart auf „Abbiegen“.

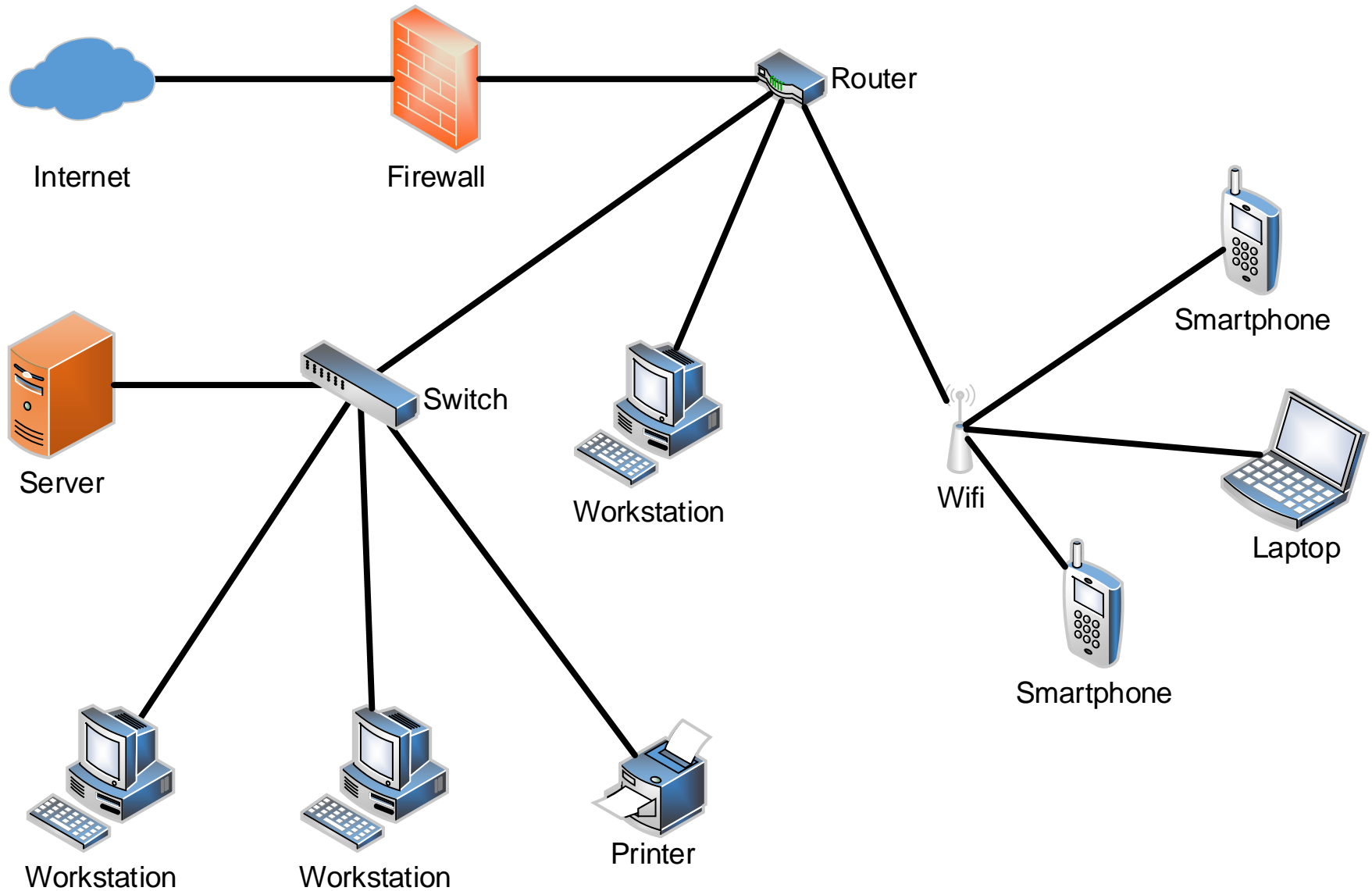
Das Laufwerk des Roboterstaubsauger greift ein, wenn Signale von den Sensoren empfangen werden.  
(Wann, wie?)

## ■ Deckt nicht alles ab, nicht immer klar trennbar ...

# Strukturelles Modell

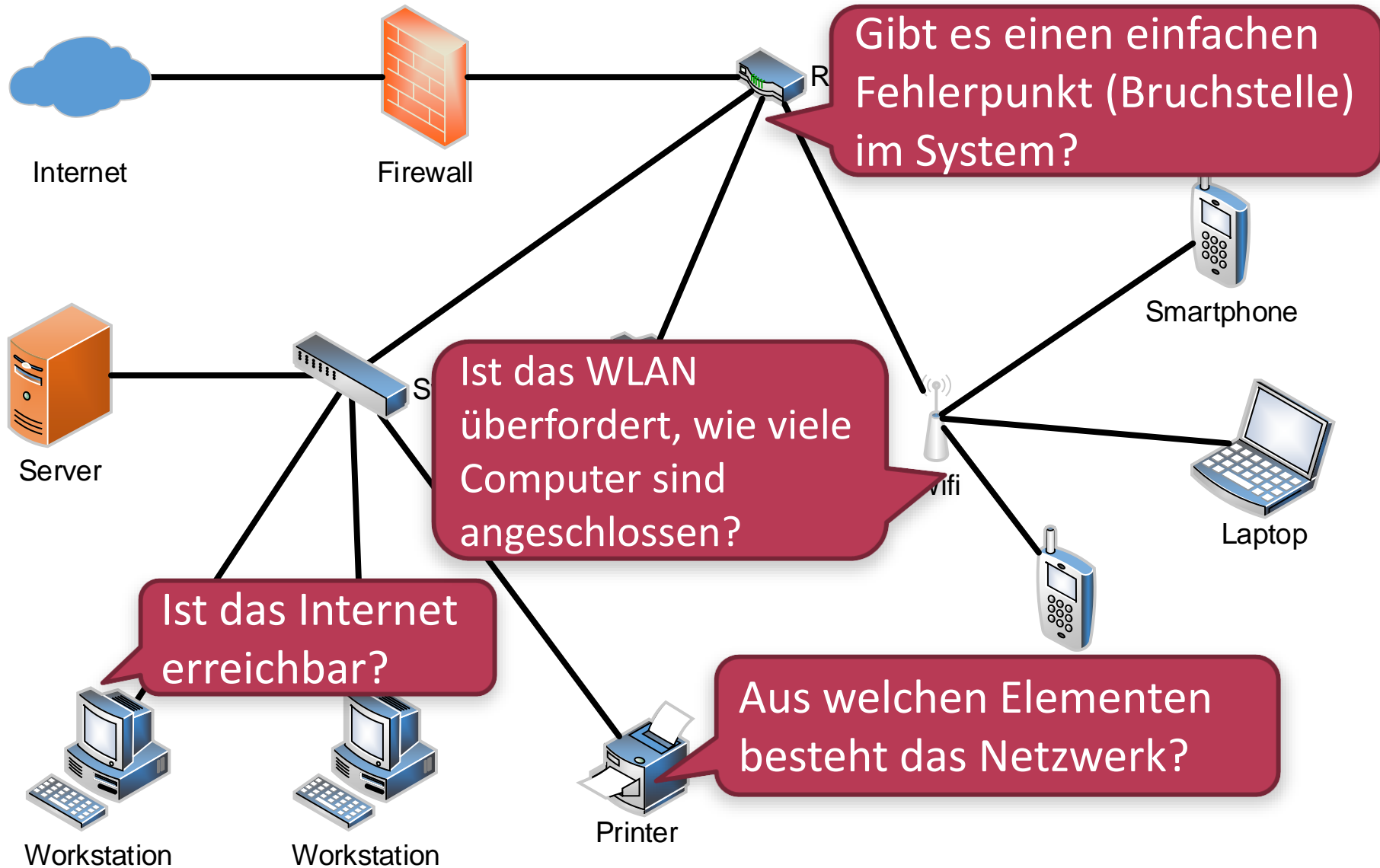
- Das Wissen bezüglich der Aufbau des Systems
  - Aus welchen Elementen besteht das System?
  - Wie sind die Elemente untereinander verknüpft?
  - Was für Eigenschaften haben die Elemente?

# Beispiel: Ein Firmennetzwerk





# Beispiel: Ein Firmennetzwerk



Ziele,  
Anwendungen

Dekomposition

Graphen

Eigenschaften

# STRUKTURELLE DEKOMPOSITION

*Was für Enthaltensein-Beziehungen sind zwischen den Teilsystemen?*

Beispiele:

- Roboterstaubsauger
- Bürgerbüro

# Definition: Dekomposition

- „Das **strukturelle Modell** ist statisch. Sein Basis ist ein (Teil-)System, das durch die Relation „**Teil von**“ auf seine Bestandteile aufgeteilt wird.“

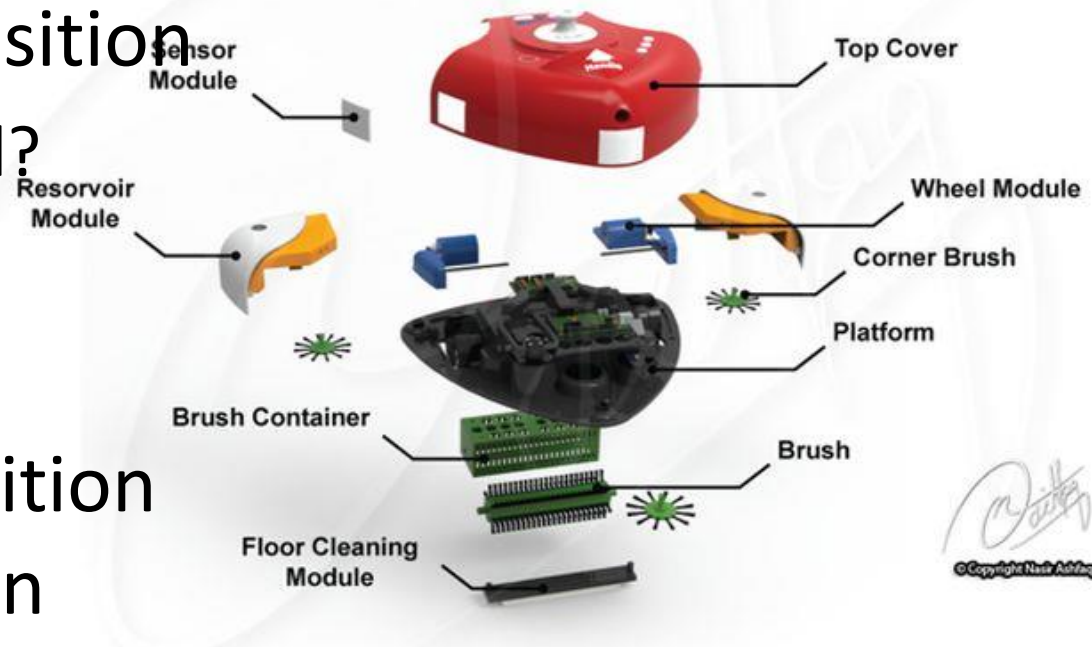
Die **Dekomposition** („faktoring“) ist die Aufteilung eines komplexen Problems oder Systems in kleinere Teile, die einfacher zu verstehen, zu entwickeln und zu warten sind.

- Hierarchische Dekomposition:
  - Mehrstufige Teil-Ganzheit-Verhältnis
- Bedeutung der Relation **Teil von**:
  - Physische Dekomposition: **Teil von** = räumliche Aufteilung
  - Logische Dekomposition: **Teil von** = funktionelle Aufteilung

# Typen der Dekomposition

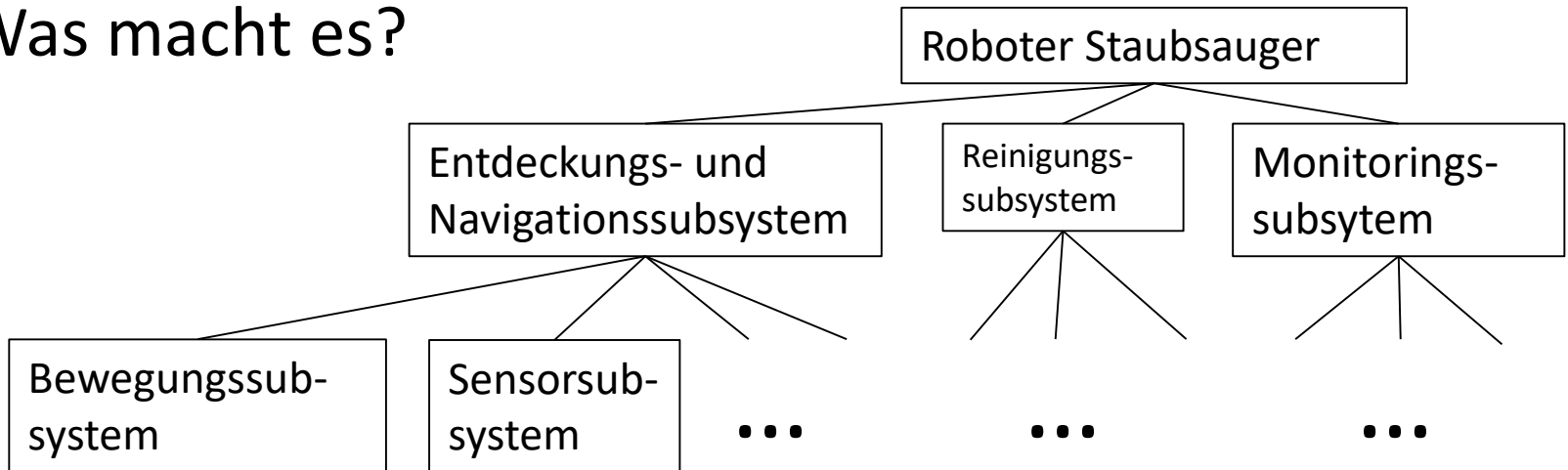
- **Physische Dekomposition**

- Welcher Bestandteil?



- **Logische Dekomposition nach den Funktionen**

- Was macht es?



# Definition: Richtigkeit der Dekomposition

Eine Dekomposition ist richtig, falls

- jedem Element des mit der Dekomposition gewonnenen Systems irgendein Element des originellen Systems entspricht, und
- jedem Element des originellen Systems ein oder mehrere Elemente des mit der Dekomposition gewonnenen Systems entspricht.

# Top-down-Entwicklung

## ■ Grundschrift: Dekomposition

Bürgerbüro

Rechentechnische Geräte

Verkabelung

POS-Terminal

Dokumentdrucker

PC

Monitor

Tastatur

Maus

Desktop-PC

Fotoabteil

Paravent

Stuhl

Beleuchtung

Fotoapparat

Bauliches Umfeld

Schreibtisch

Kundenschalter

Stuhl

Schliessfach

# Bottom-up-Entwicklung

## ■ Grundschrift: Komposition

Soziales Netzwerk

Server-Infrastruktur

Webserver

DB-Server

Statische-Content-Server

Server-seitige Software

Datenbank

Anwendung

Werbungen

Überwachung

Web-Schnittstelle

Interface-Desing (UX)

Graphik

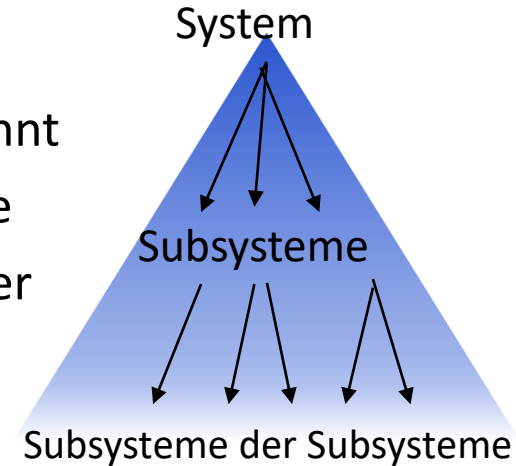
Dynamik (JS)

Handy-Schnittstelle

# Top-down und bottom-up

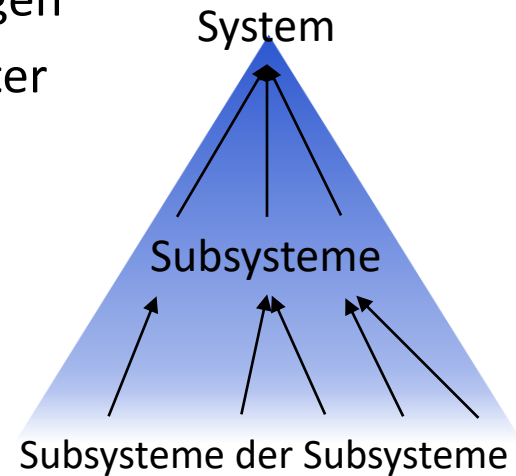
## ■ Top-down

- ☺ Rolle des Teilsystems ist bei der Entwicklung schon bekannt
- ☹ In der “Halbzeit” gibt es noch keine funktionierende Teile
- ☹ Probleme/Ansprüche der Teilsysteme kommen erst später ins Tageslicht



## ■ Bottom-up

- ☺ Teilsysteme sind selbstständig auszuprobieren / zu testen
- ☺ Während Entwicklung ist schon etwas zum Lauf zu bringen
- ☹ Die genaue Rolle der einzelnen Teile sieht man erst später



- Nicht nur in der strukturellen Modellierung
- Gemischte Vorgehensweise, iterative Entwicklung



Ziele,  
Anwendungen

Dekomposition

Graphen

Eigenschaften

# STRUKTURELLE MODELLE

*Wie sind die Verhältnisse unter den Subsystemen?*

Beispiele

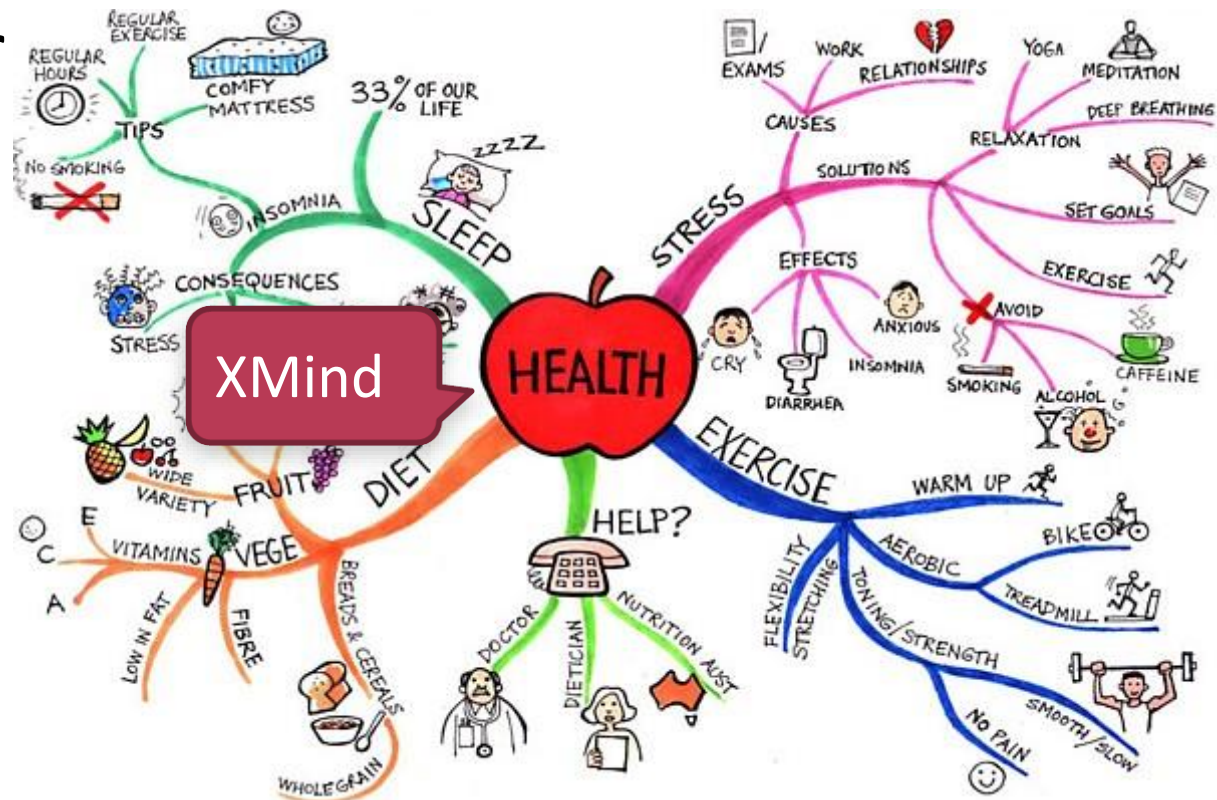
- Streckennetz
- Dateisystem

# Graphrepräsentation

- Menschliche Denkweise: „Verbindung der Dinge“
  - **Dinge:** Personen, Flughäfen, Gebiete
  - **Verbindung:** Dependenz, Bekanntschaft, Flug, Teil

- Mathematischer Formalismus:  
**Graph**

- Knotenpunkte
- Kanten
- (Eigenschaften)



# Definitionen in der Informatik

- In der Informatik hat jedes Ding mehrere, oft einander widersprechende Definitionen
  - System = ?
  - Modell = ?
- Oft wird derselbe Begriff mit verschiedenen Namen genannt
  - Knotenpunkt, Knoten, Punkt, Objekt  
(*node, vertex, object, concept*)
  - Kante, Verbindung, Anschluss  
(*edge, link, arc, connection, relationship*)

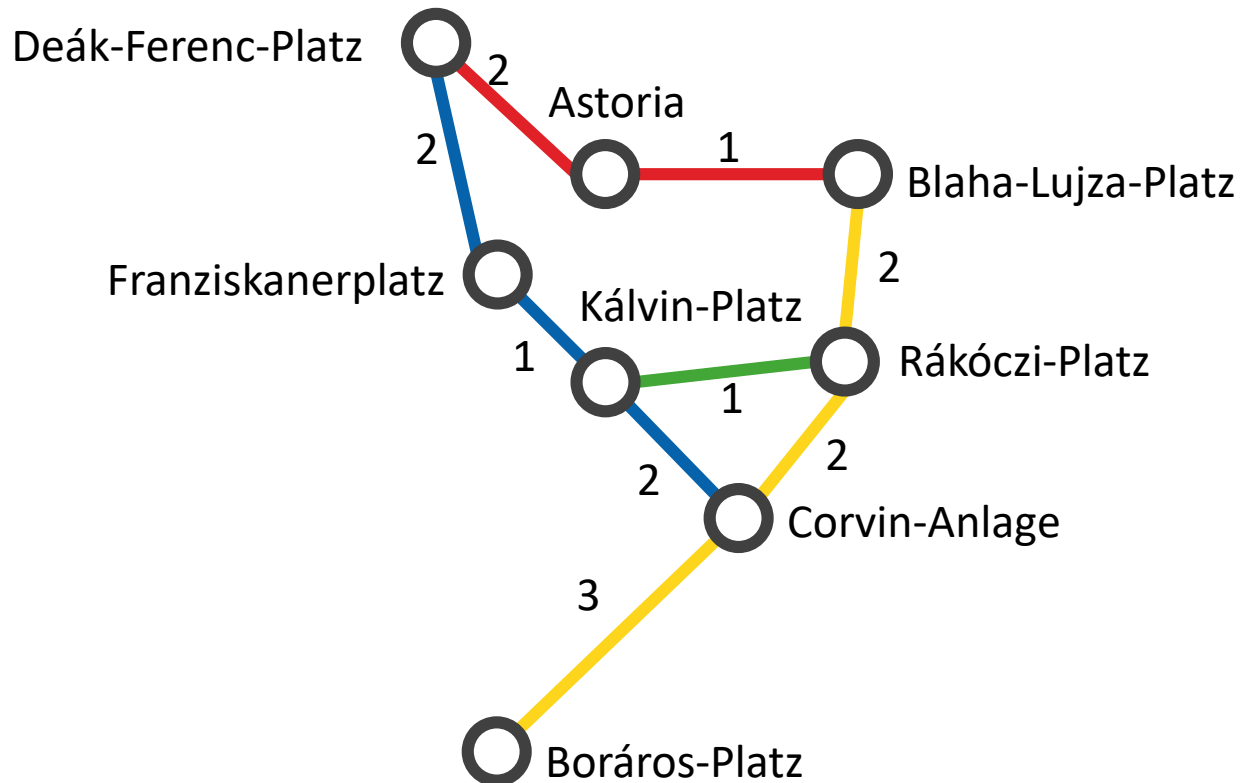
Eins auswählen!

# Budapest – Schienenverkehrsplan

- M2
- M3
- M4
- 4-6



# Budapest – Schienenverkehrsplan

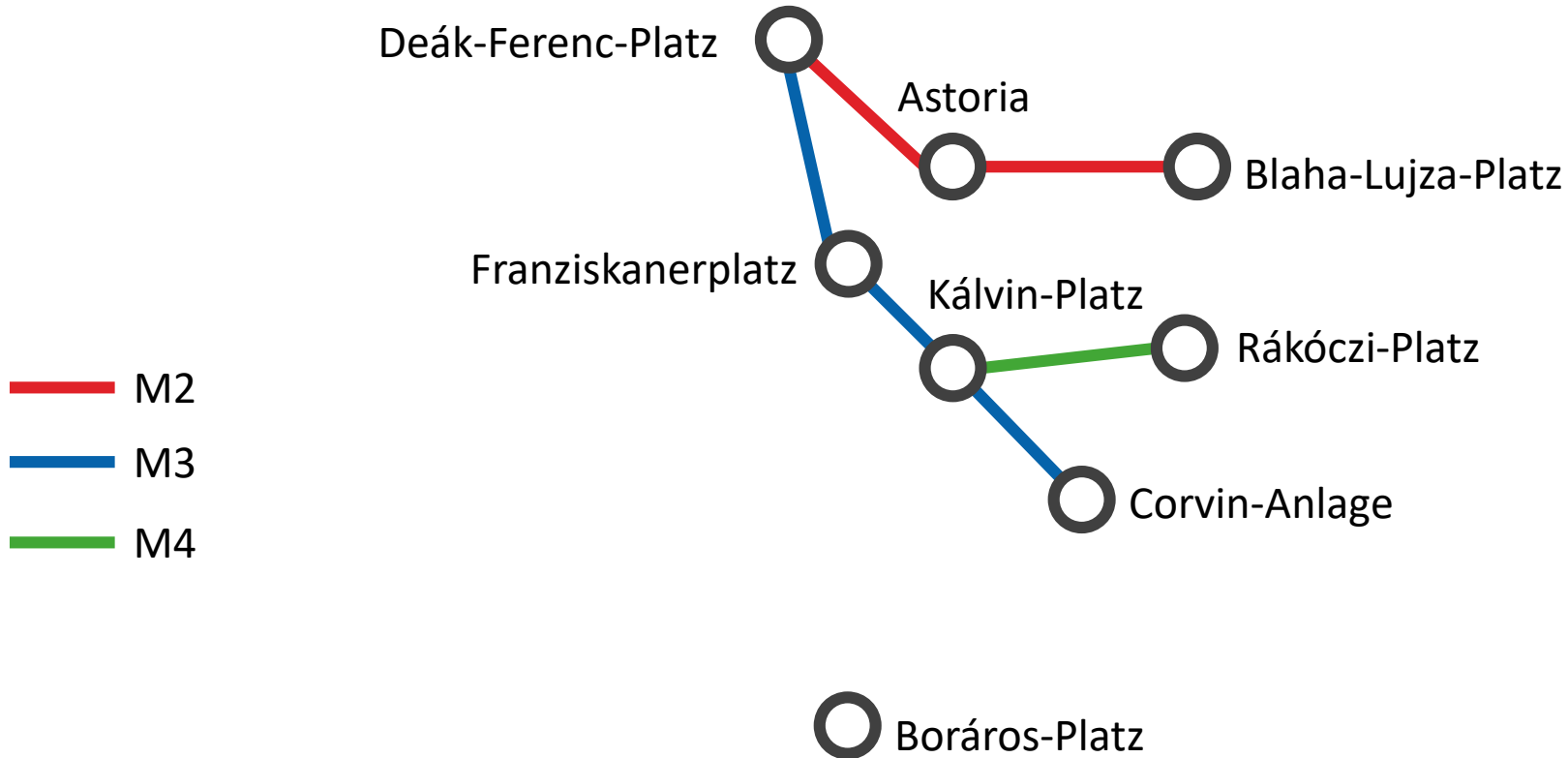


- M2
- M3
- M4
- 4-6

kürzeste Weg?

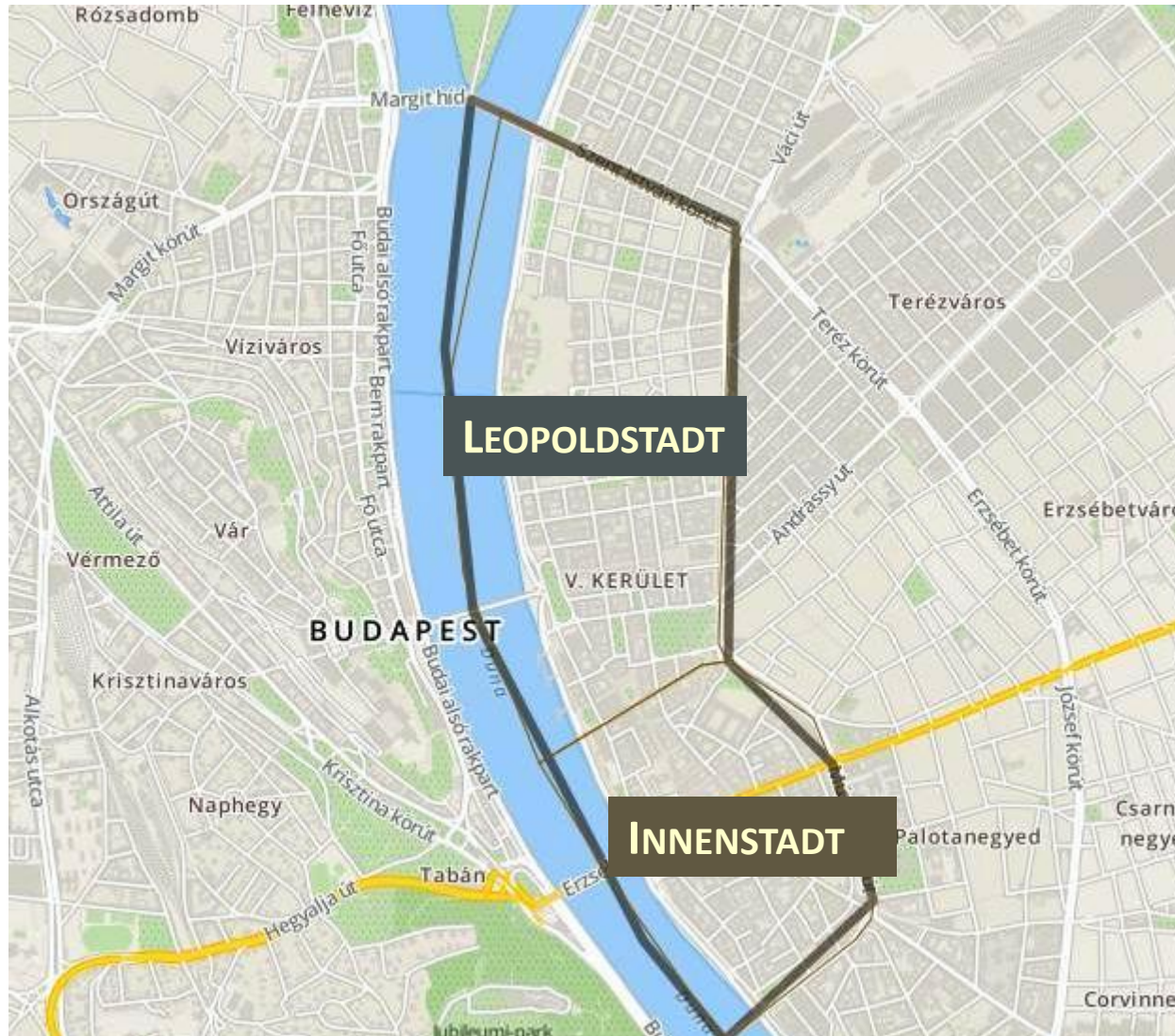
# Filterung: nach Kantenaufschrift (Teilgraph)

- z.B. man möchte nur die U-Bahn-Netzwerk sehen

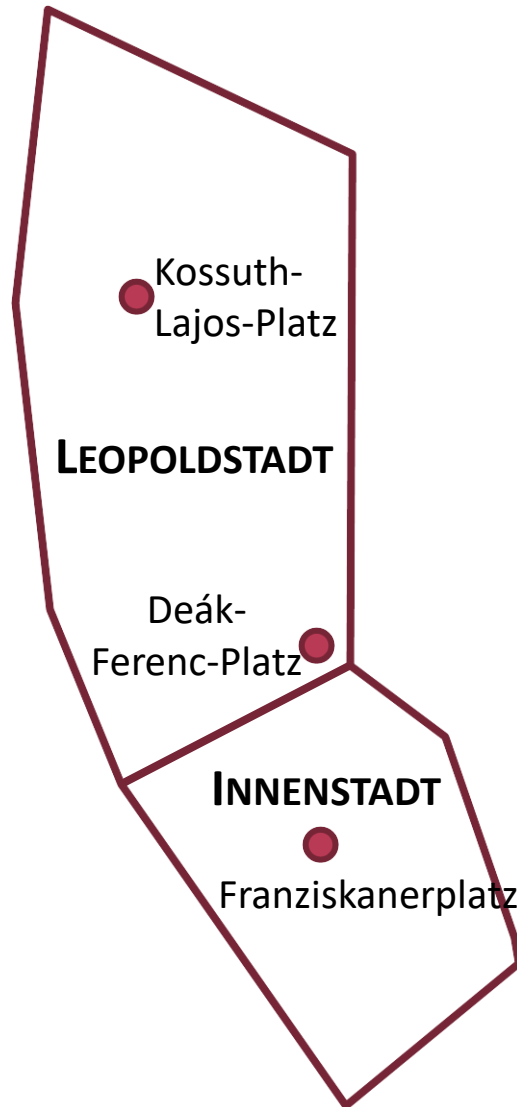


- Was ist mit U-Bahn erreichbar?

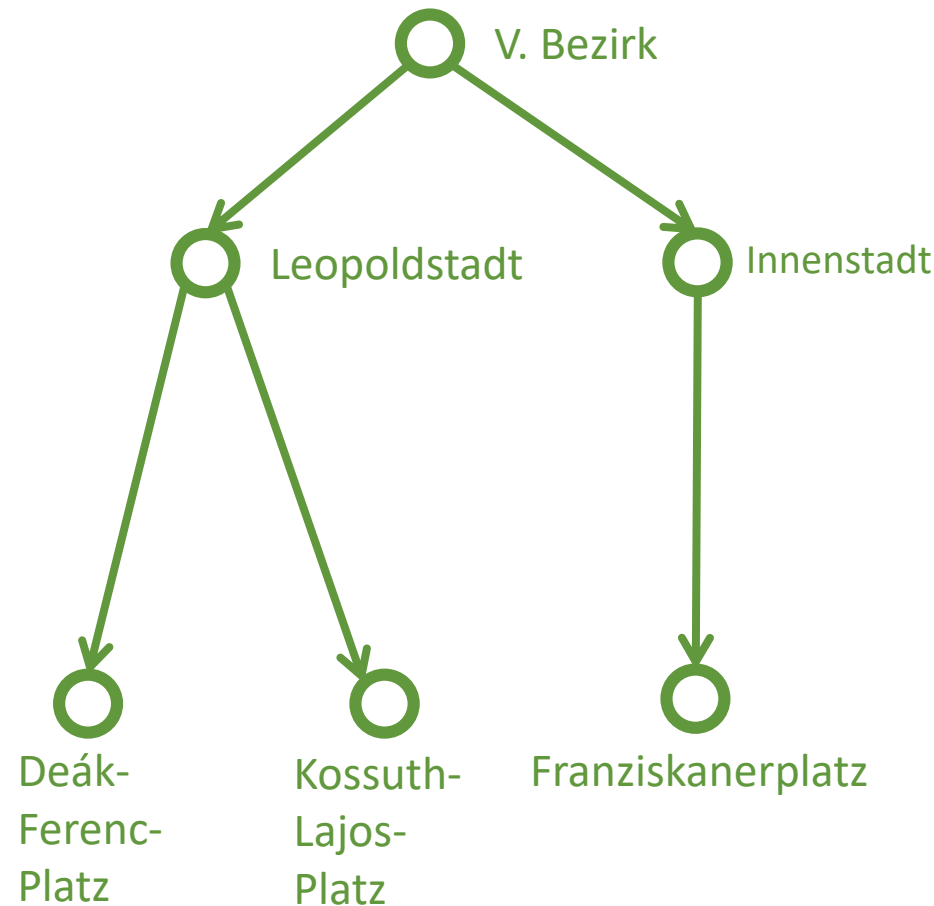
# Budapest V. Bezirk



# Budapest V. Bezirk – hierarchisches Modell

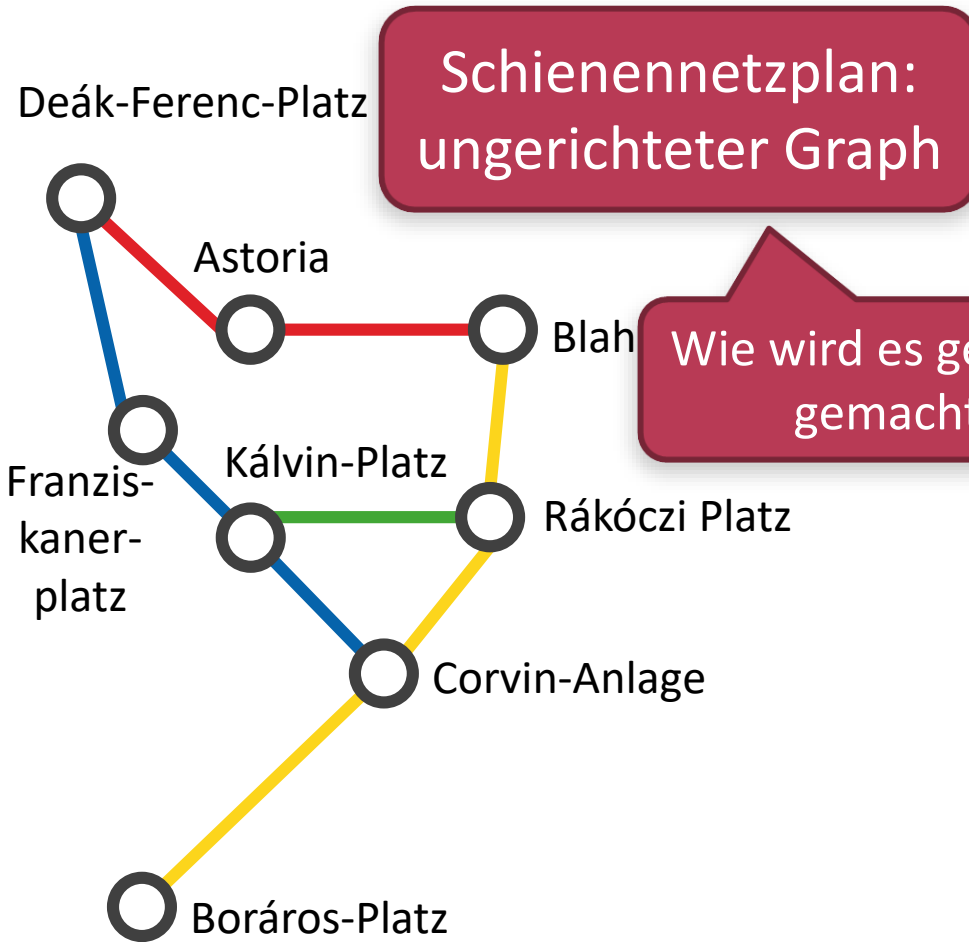


≡



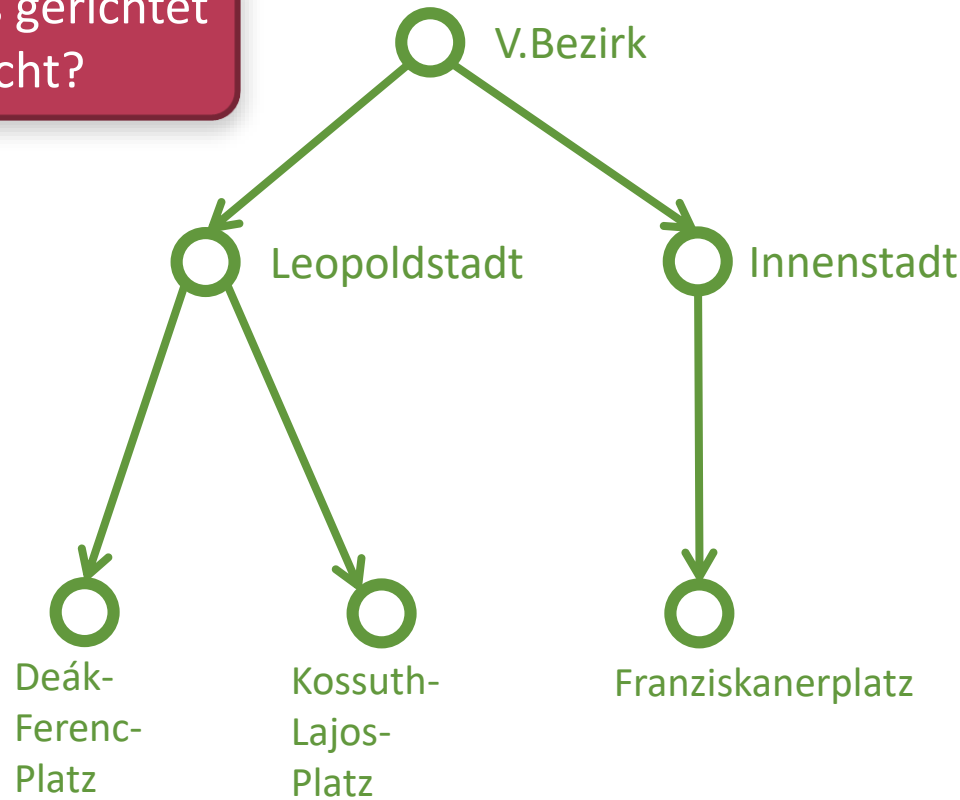


# Einfache Graphen



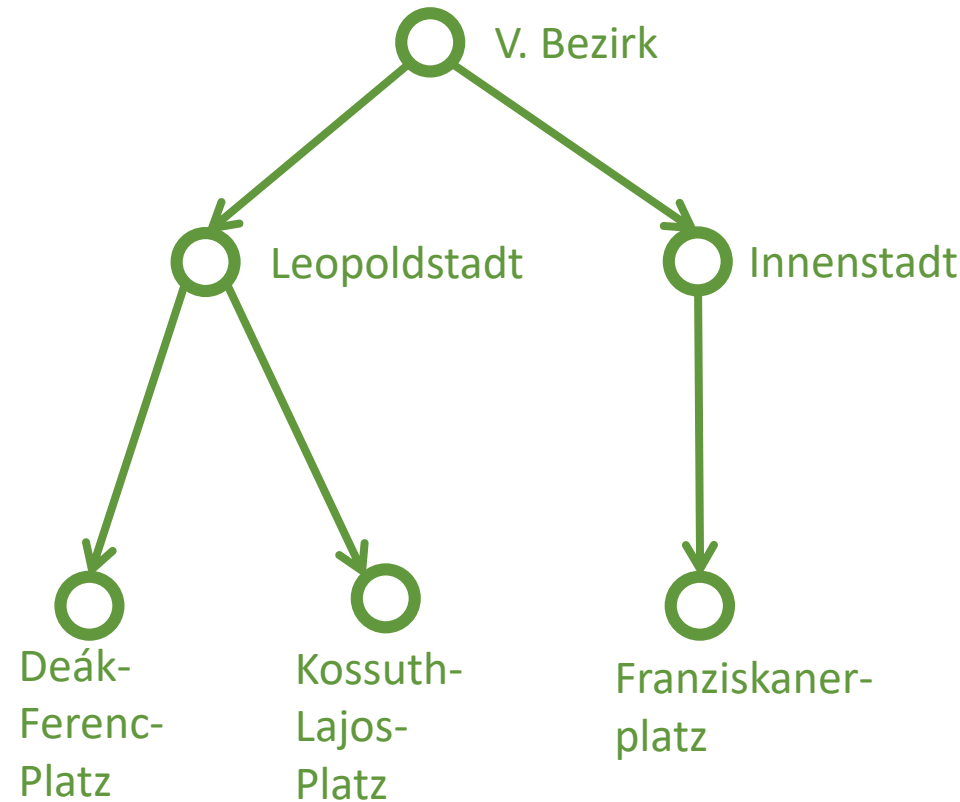
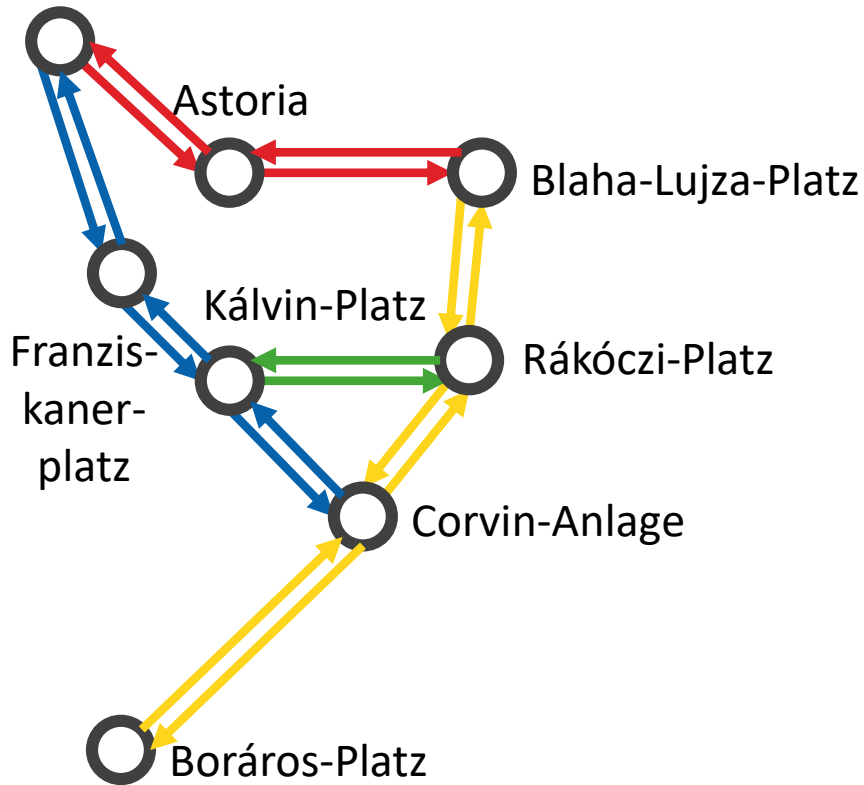
Wie wird es gerichtet gemacht?

Bereiche:  
gerichteter Graph

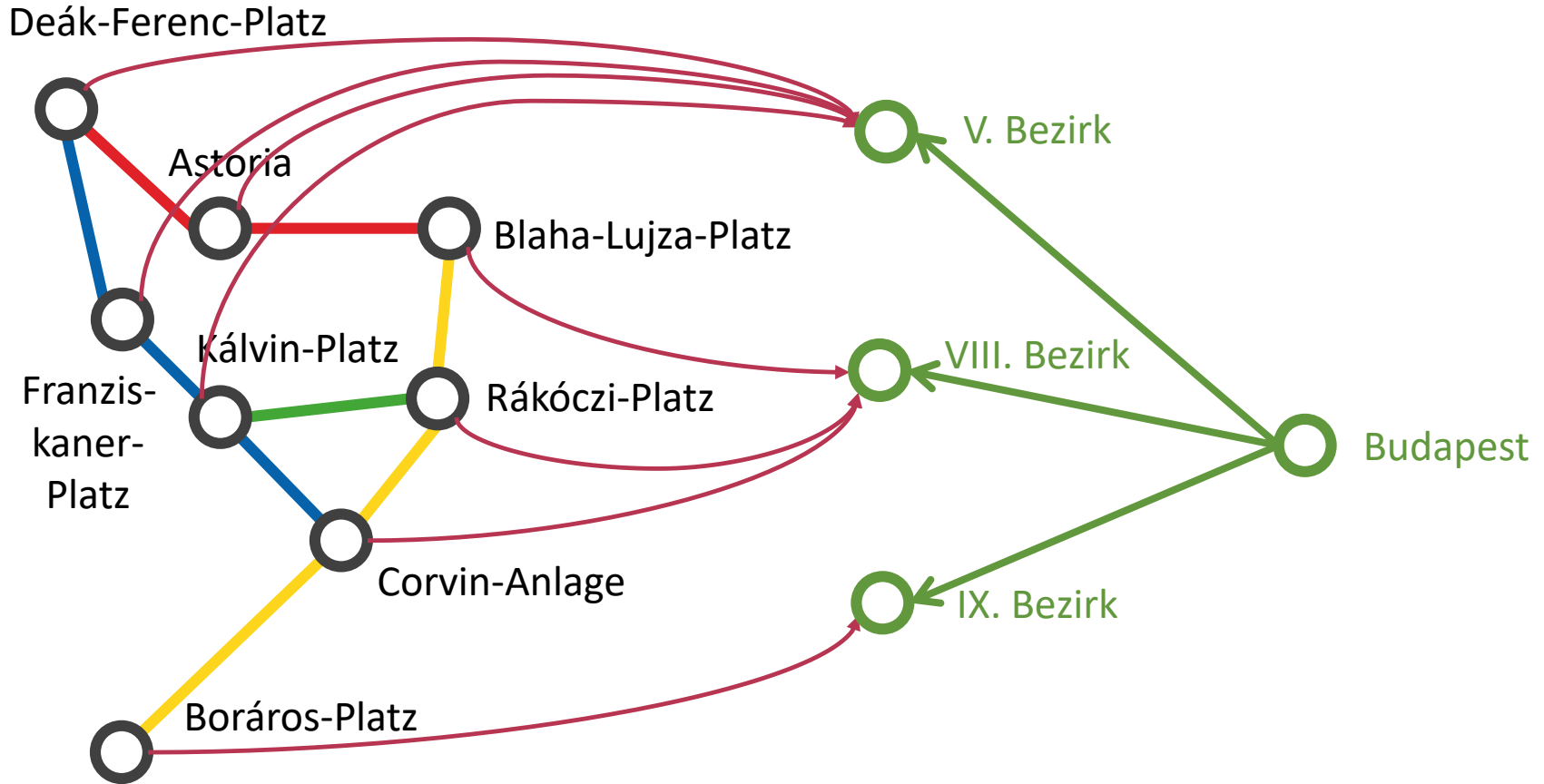


# Einfache Graphen

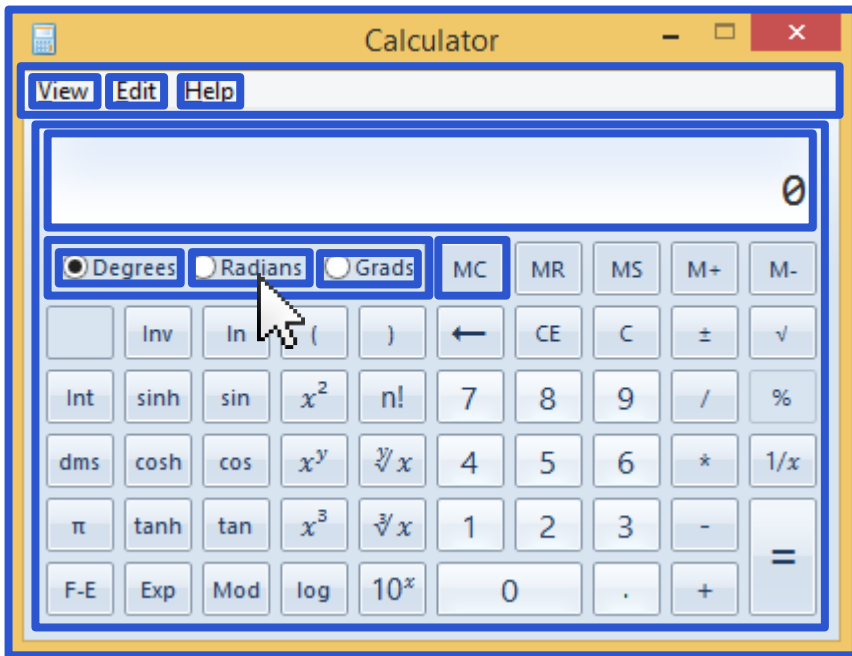
Deák-Ferenc-Platz



# Graph mit Typen

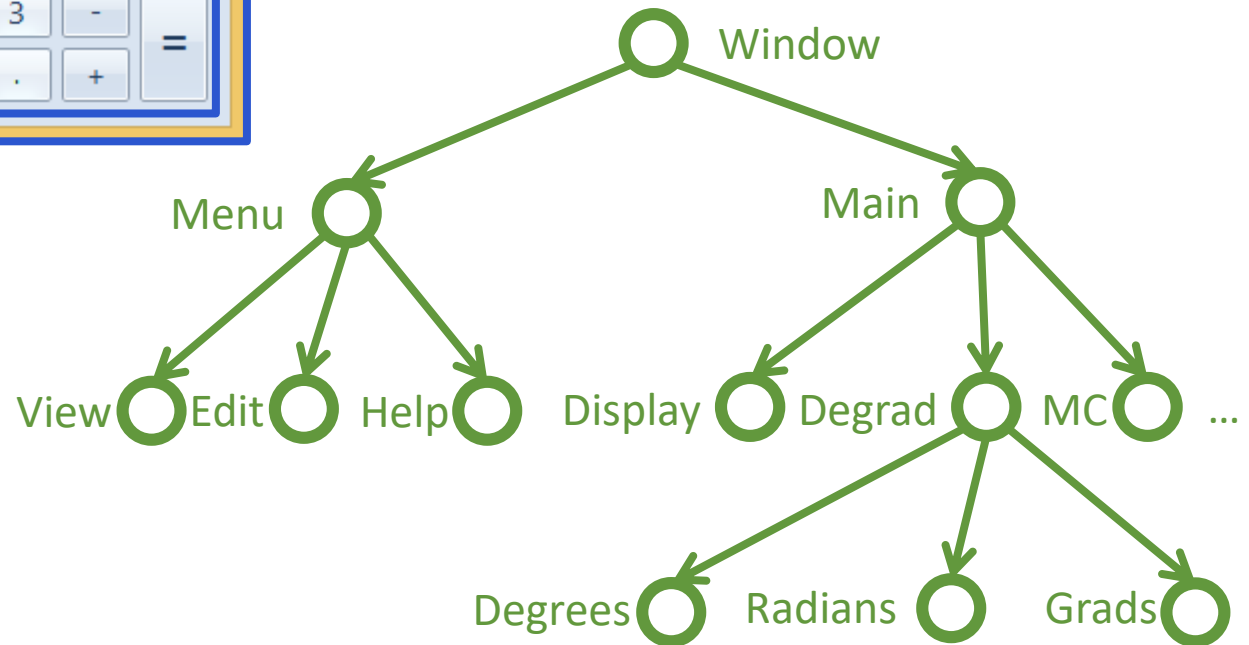


# Graphische Benutzerschnittstelle



## ■ Darstellung einer Baumstruktur

- als Graph
- integrative Darstellung



# Darstellung einer Baumstruktur

## Dateisystem

C:

\Dokumente

\Bilder

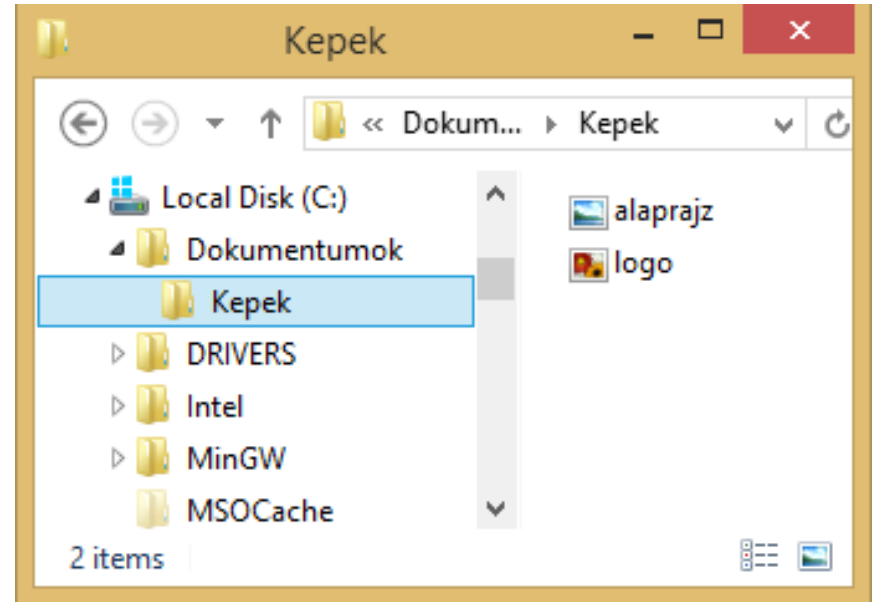
\Logo.png

\Grundriss.jpg

\Verträge.pdf

\Drivers

...



Wieviel Speicherplatz kann durch das Löschen des Verzeichnisses „Dokumente“ gewonnen werden?

# Charakteristiken eines Dateisystems

- Wieviel Speicher kann durch das Löschen des Verzeichnisses „Dokumente“ gewonnen werden?
- Dazu sind einige Parameter der Dateien nötig

Name ▾	Typ ▾	Grösse (kB) ▾	Letze Änderung ▾
Dokumente	Verzeichnis		2016.02.02
Verträge.pdf	Datei	569	2015.11.09
Bilder	Verzeichnis		2016.02.02
Logo.png	Datei	92	2015.03.06
Grundriss.jpg	Datei	1226	2016.02.02

- Mit Kenntnis dieser Informationen und der Dateihierarchie ist die Frage zu beantworten.

Ziele,  
Anwendungen

Dekomposition

Graphen

Eigenschaften

# MODELLIERUNG DER EIGENSCHAFTEN

*Welche Eigenschaften haben die einzelne Komponenten?*

Beispiele

- Tabelle
- Datenbanken

# Definitionen: Eigenschaft

- Eigenschaften der Elemente eines Modells können auch beschrieben werden

**Eigenschaft** ist eine **Funktion**, die

- auf den **Modellelementen** definiert und
- **partial** (Funktionswerte dürfen fehlen) ist, und
- dessen **Wertebereich** die Menge der Werte der Eigenschaften ist.



# Tabellarische Darstellung

- **Zeilen der Tabelle** = Modellelemente
- **Spalten der Tabelle** = Eigenschaften

Name ▾	Typ ▾	Grösse (kB) ▾	Letze Änderung ▾
Dokumente	Verzeichnis		2016.02.02
Verträge.pdf	Datei	569	2015.11.09
Bilder	Verzeichnis		2016.02.02
Logo.png	Datei	92	2015.03.06
Grundriss.jpg	Datei	1226	2016.02.02

NULL / NA  
Attributen

# Definition: Filterung

- Während der **Filterung** wird eine **Filterungsbedingung** auf den Elementen des Modells ausgewertet, und
- das **Teilmodell** wird behalten, welches den Bedingungen entsprechende Elemente hat.

○ Typ = "Datei"

Name ▾	Typ ▾	Grösse (kB) ▾	Letze Änderung ▾
Verträge.pdf	Datei	569	2015.11.09
Logo.png	Datei	92	2015.03.06
Grundriss.jpg	Datei	1226	2016.02.02

○ Grösse > 1000 kB

Name ▾	Typ ▾	Grösse (kB) ▾	Letze Änderung ▾
Grundriss.jpg	Datei	1226	2016.02.02

# Definition: Projektion

Während der **Projektion** werden einige **Parameter** des Modells ausgewählt, andere werden weggelassen.

Name ▼	Typ ▼	Grösse (kB) ▼	Letze Änderung ▼
Dokumente	Verzeichnis		2016.02.02
Verträge.pdf	Datei	569	2015.11.09
Bilder	Verzeichnis		2016.02.02
Logo.png	Datei	92	2015.03.06
Grundriss.jpg	Datei	1226	2016.02.02

Projektion:  
{Name, Grösse}

Name ▼	Grösse (kB) ▼
Dokumente	
Verträge.pdf	569
Bilder	
Logo.png	92
Grundriss.jpg	1226

# ZUSAMMENFASSUNG

# Definition: Strukturelles Modell

Das **strukturelle Modell** ist statisch. Sein Basis ist ein (Teil-) System, das durch die Relation „**Teil von**“ auf seine Bestandteile aufgeteilt wird.

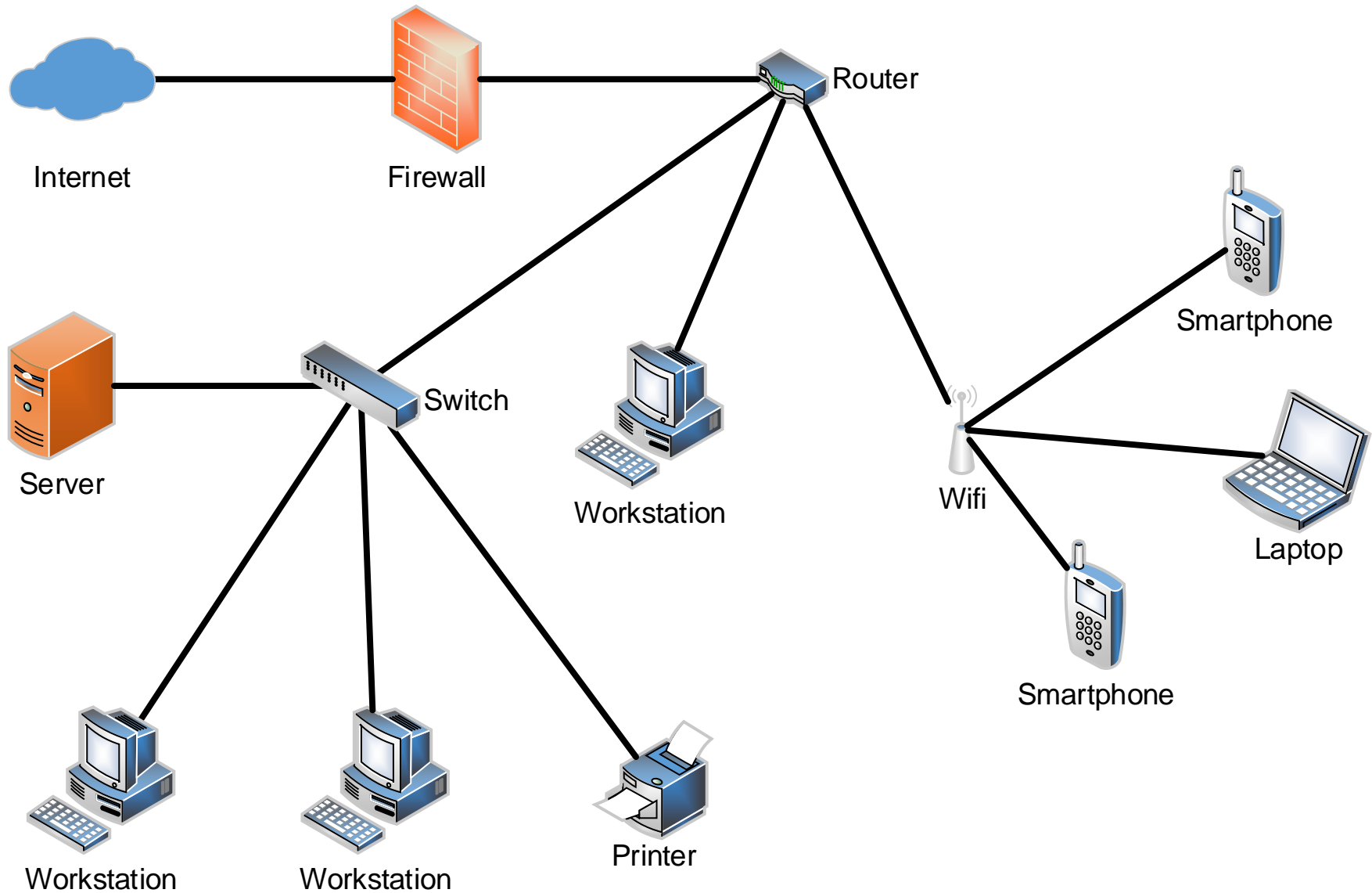
Die **Bestandteile** können die folgenden sein:

- weiter aufgeteilte **Teilsysteme** oder
- weiter nicht aufgeteilte (*elementare*) **Komponenten**.

Das strukturelle Modell repräsentiert die Aufbau (Struktur) des Systems entsprechend

- dessen Bestandteilen,
- den **Eigenschaften** der Bestandteile und
- ihren **Verhältnissen** untereinander.

# Beispiel: Ein Firmennetzwerk



# Definition: Dekomposition

- „Das **strukturelle Modell** ist statisch. Sein Basis ist ein (Teil-)System, das durch die Relation „**Teil von**“ auf seine Bestandteile aufgeteilt wird.“

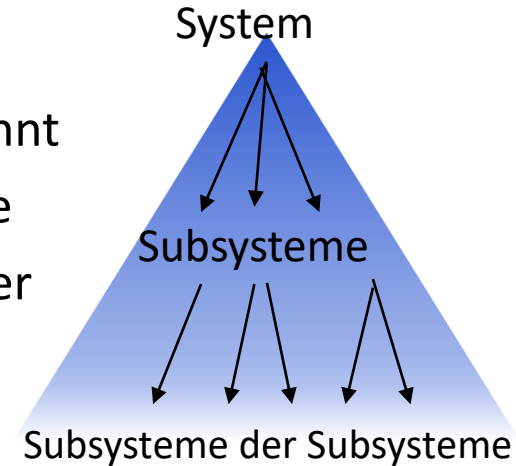
Die **Dekomposition** („faktoring“) ist die Aufteilung eines komplexen Problems oder Systems in kleinere Teile, die einfacher zu verstehen, zu entwickeln und zu warten sind.

- Hierarchische Dekomposition:
  - Mehrstufige Teil-Ganzheit-Verhältnis
- Bedeutung der Relation Teil von:
  - Physische Dekomposition: Teil von = räumliche Aufteilung
  - Logische Dekomposition: Teil von = funktionelle Aufteilung

# Top-down und bottom-up

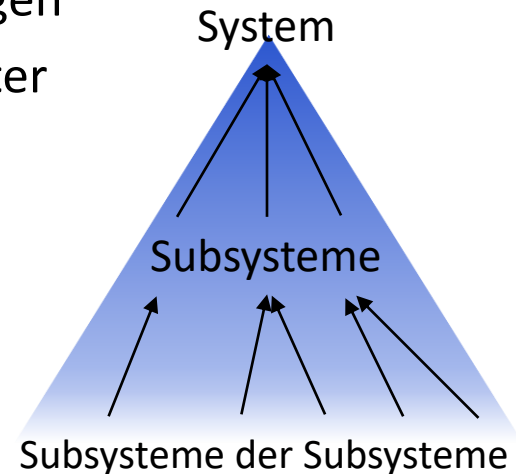
## ■ Top-down

- ☺ Rolle des Teilsystems ist bei der Entwicklung schon bekannt
- ☹ In der “Halbzeit” gibt es noch keine funktionierende Teile
- ☹ Probleme/Ansprüche der Teilsysteme kommen erst später ins Tageslicht



## ■ Bottom-up

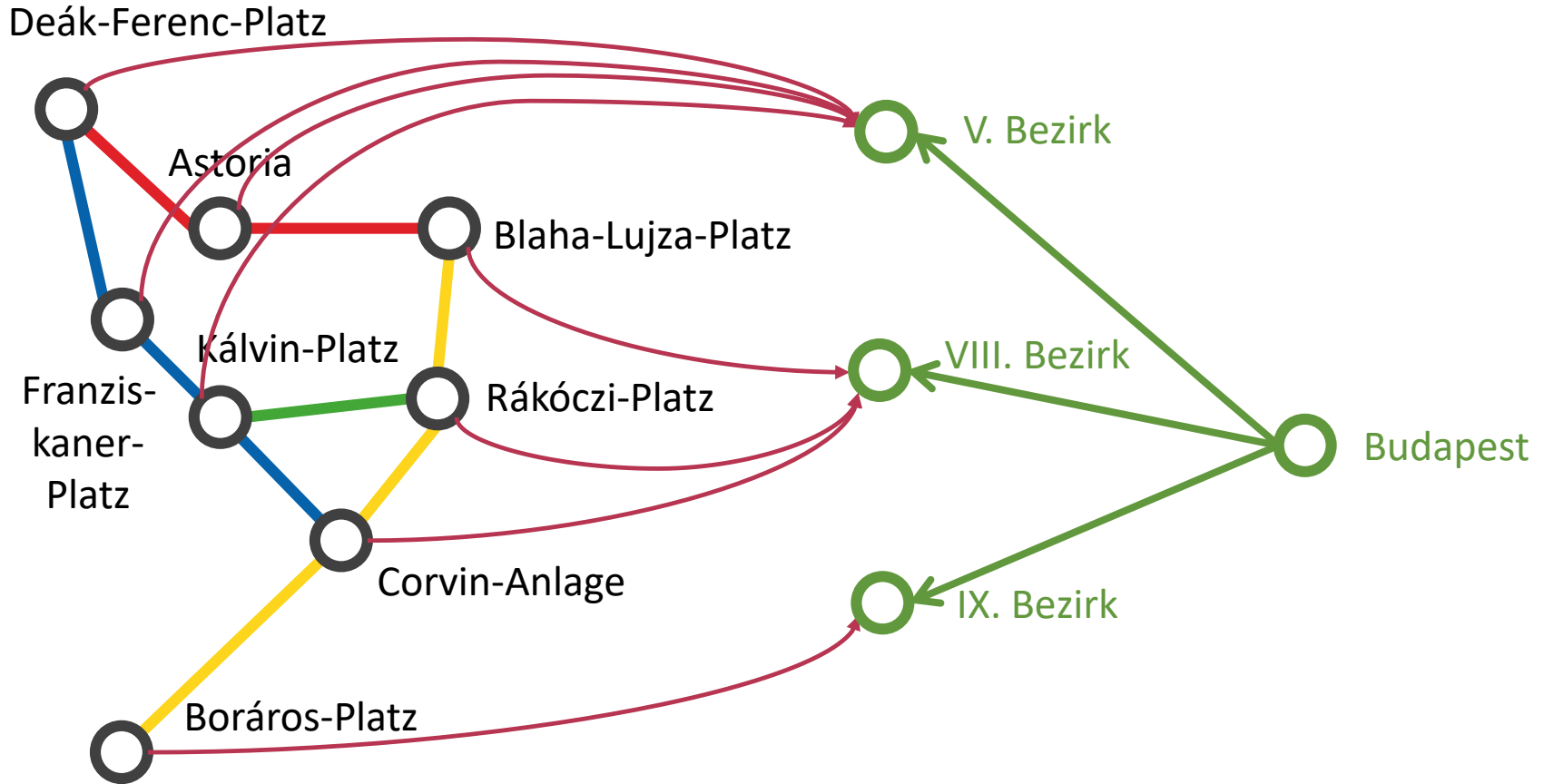
- ☺ Teilsysteme sind selbstständig auszuprobieren / zu testen
- ☺ Während Entwicklung ist schon etwas zum Lauf zu bringen
- ☹ Die genaue Rolle der einzelnen Teile sieht man erst später



- Nicht nur in der strukturellen Modellierung
- Gemischte Vorgehensweise, iterative Entwicklung



# Graph mit Typen



# Tabellarische Darstellung

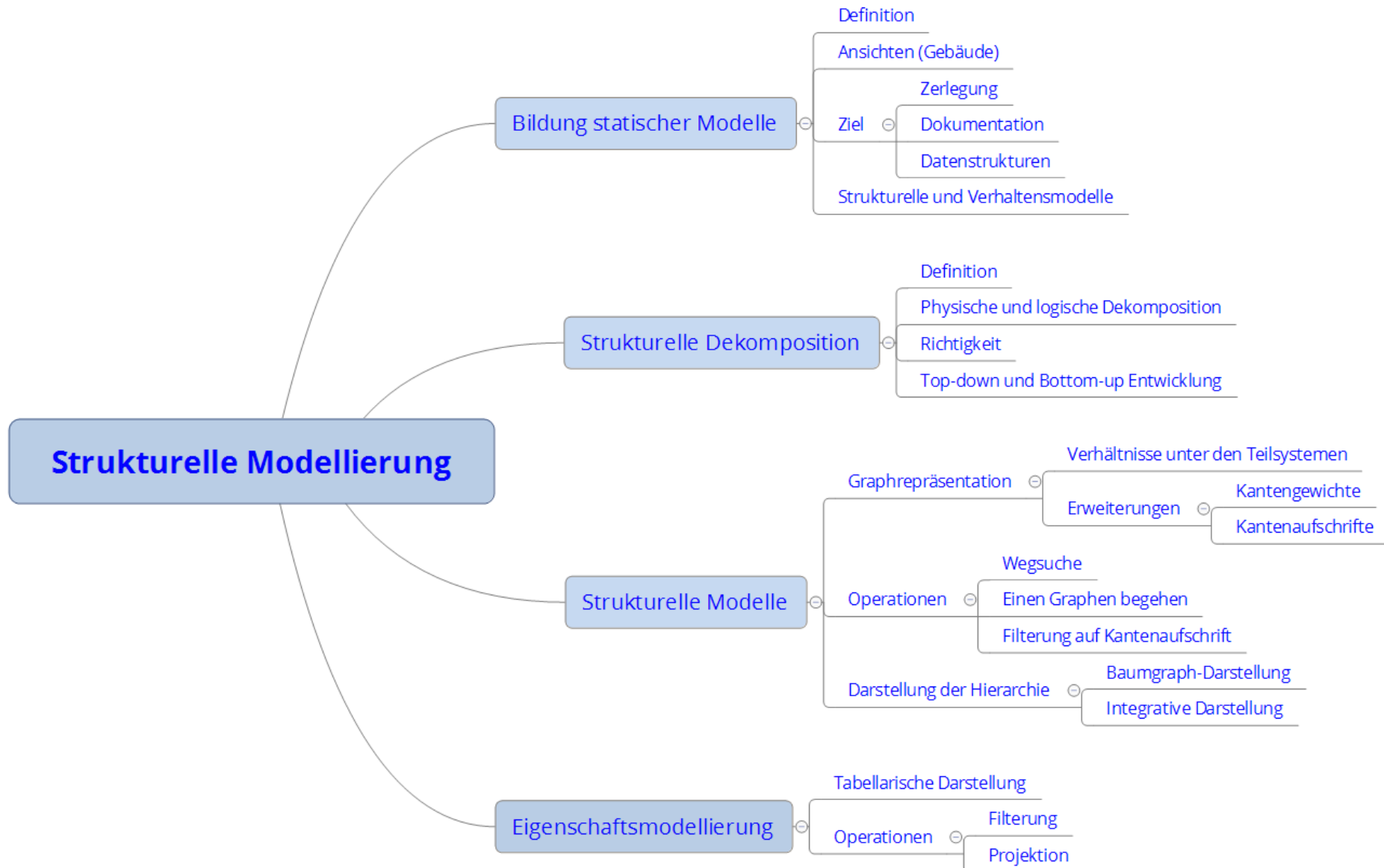
- **Zeilen der Tabelle** = Modellelemente
- **Spalten der Tabelle** = Eigenschaften

Name ▾	Typ ▾	Grösse (kB) ▾	Letze Änderung ▾
Dokumente	Verzeichnis		2016.02.02
Verträge.pdf	Datei	569	2015.11.09
Bilder	Verzeichnis		2016.02.02
Logo.png	Datei	92	2015.03.06
Grundriss.jpg	Datei	1226	2016.02.02

NULL / NA  
Attributen

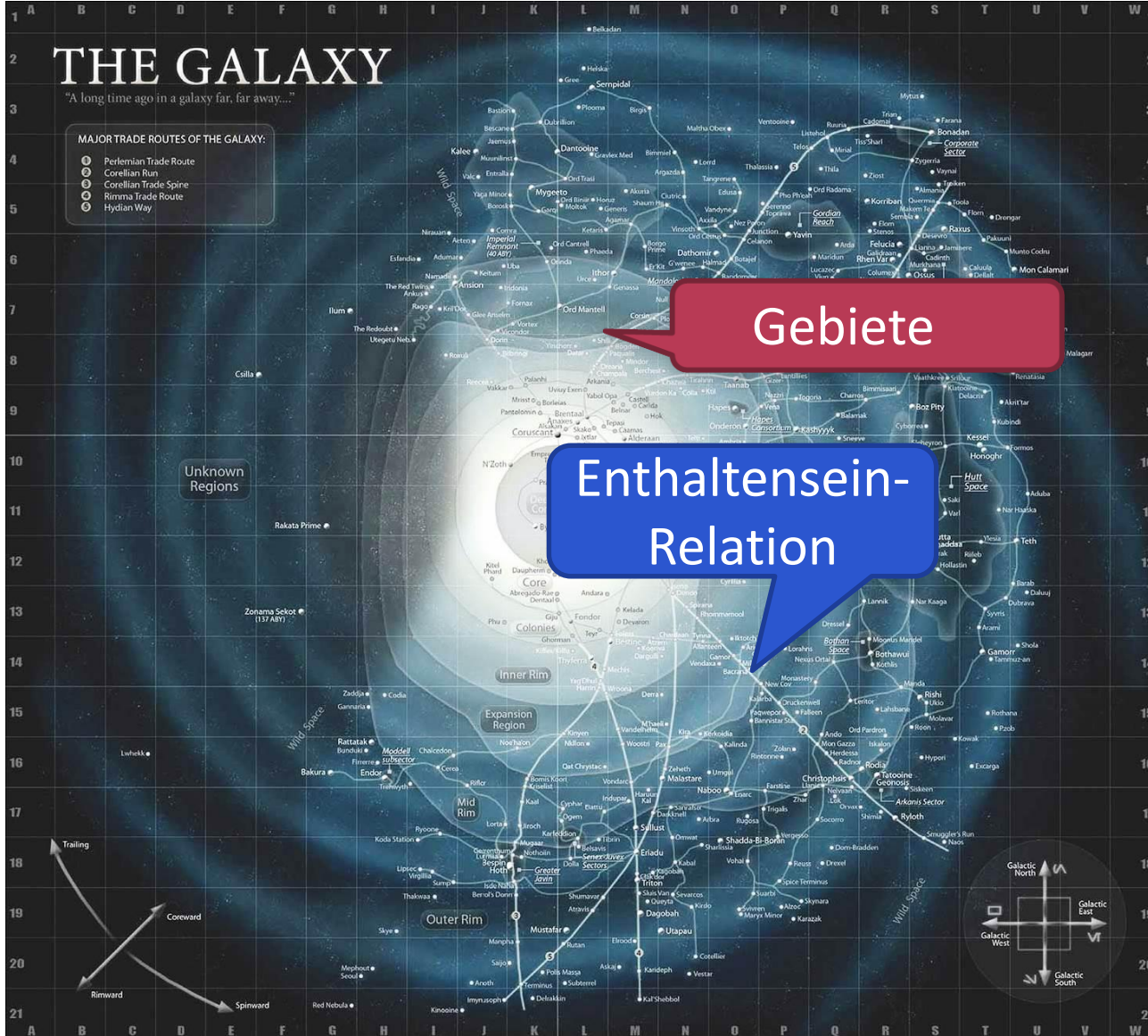
- Filtern
- Projizieren

# Zusammenfassung

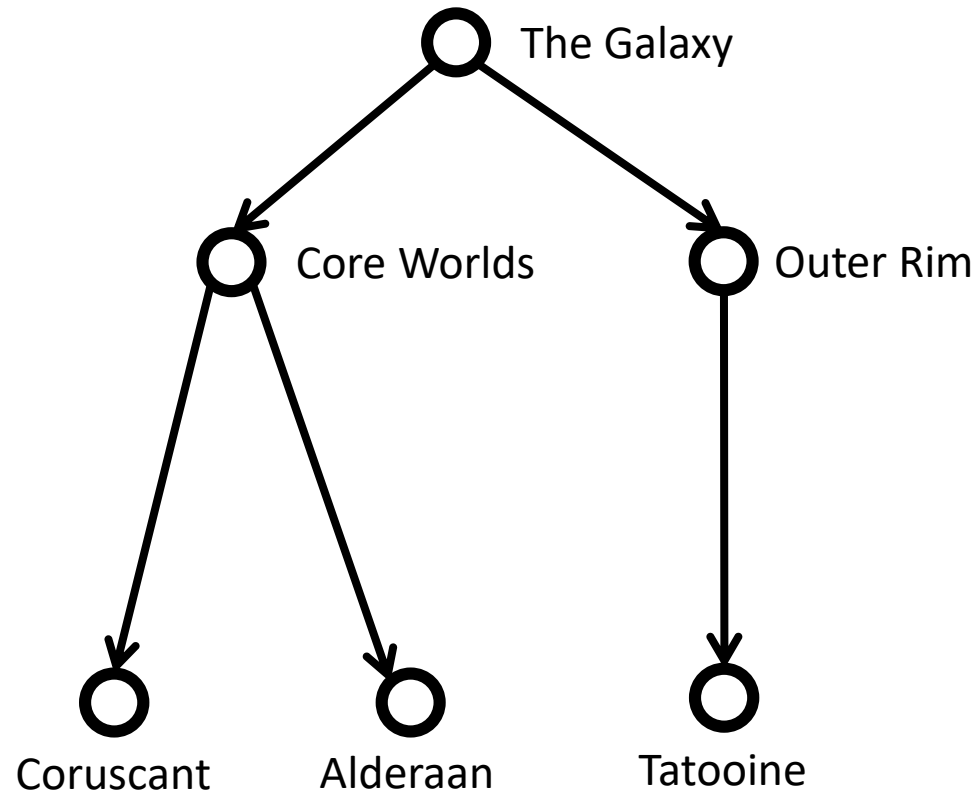


# TYPEN

# Beispiel: Die Galaxis



# Beispiel: Die Galaxis



# Einstufung von Modellelementen in Typen

- Beispiel: Kann die „Farbe des Lichtsäbels“ von Yoda sich ändern?
  - Nein  
Yoda – eine Entität mit grünem Lichtsäbel  
Wichtiges/relevantes Merkmal?
  - Ja  
Lichtsäbelfarbe kein charakteristisches Merkmal
- Typ: Gesamtheit der durch ähnlichen Attribute gekennzeichneten(!) Elemente

# Typen und Eigenschaften

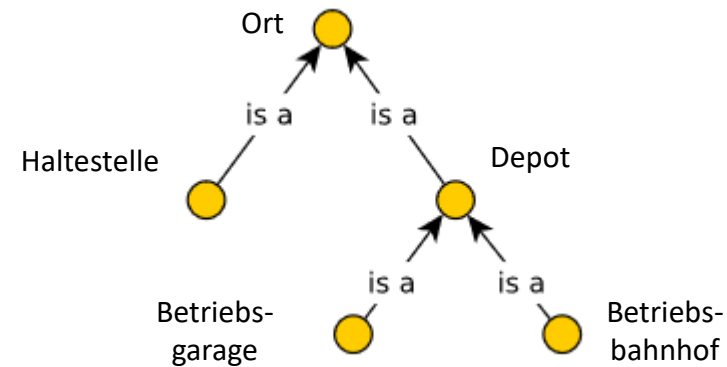
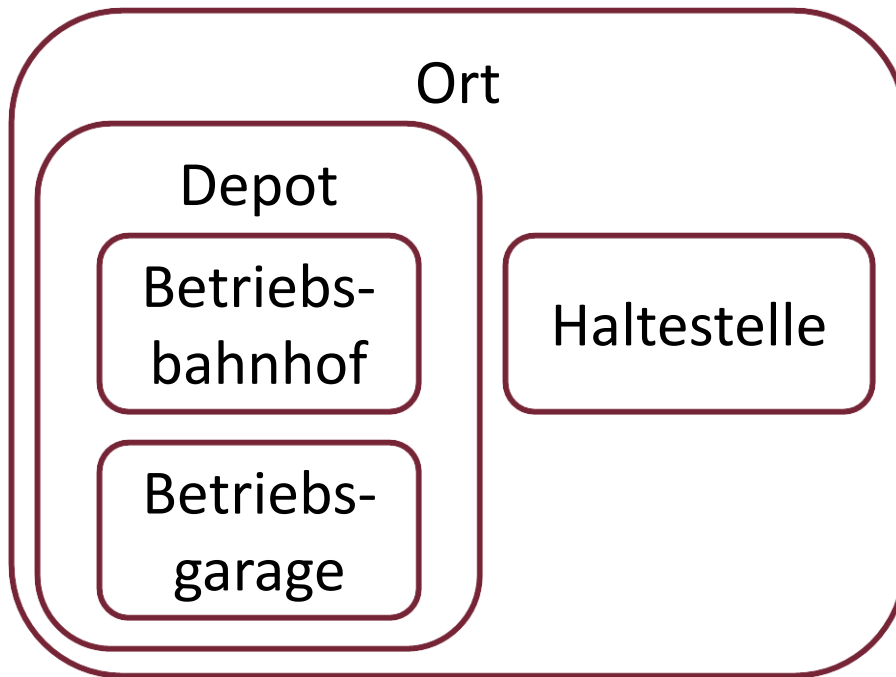
- Der Typ ist ein Teil des Wissens, impliziter Weise repräsentiert in der Markierung
  - der *Typ* ist also ein ausgezeichnetes Attribut
  - die andere Merkmale: die *Eigenschaften*
  
- Verbreitete Konvention:
  - die Eigenschaften dürfen sich verändern (müssen aber nicht)
  - der Typ eines Elementes ist fix (für IMMER)



# Typengraph

- Für jeden Knotentyp ein Typenknoten
- Für jeden Kantentyp eine Typenkante
- ...

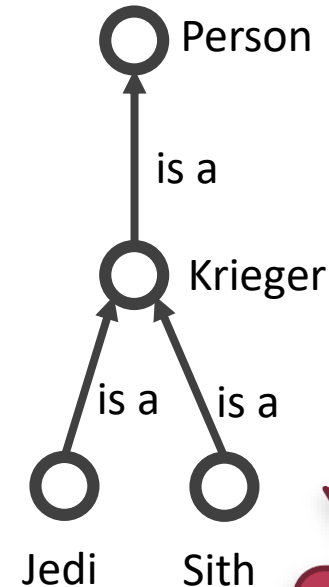
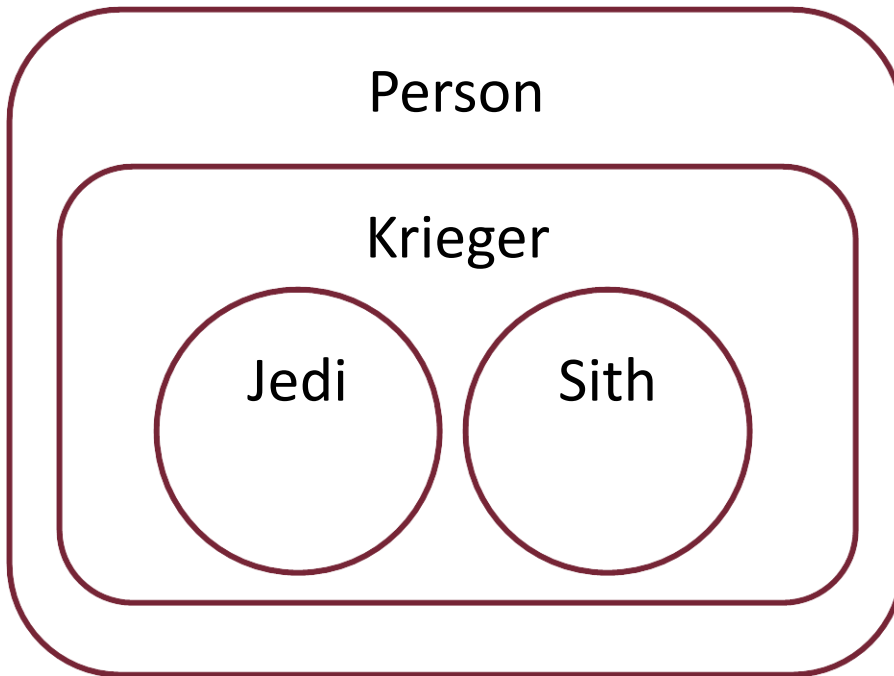
Metamodell



# Typengraph

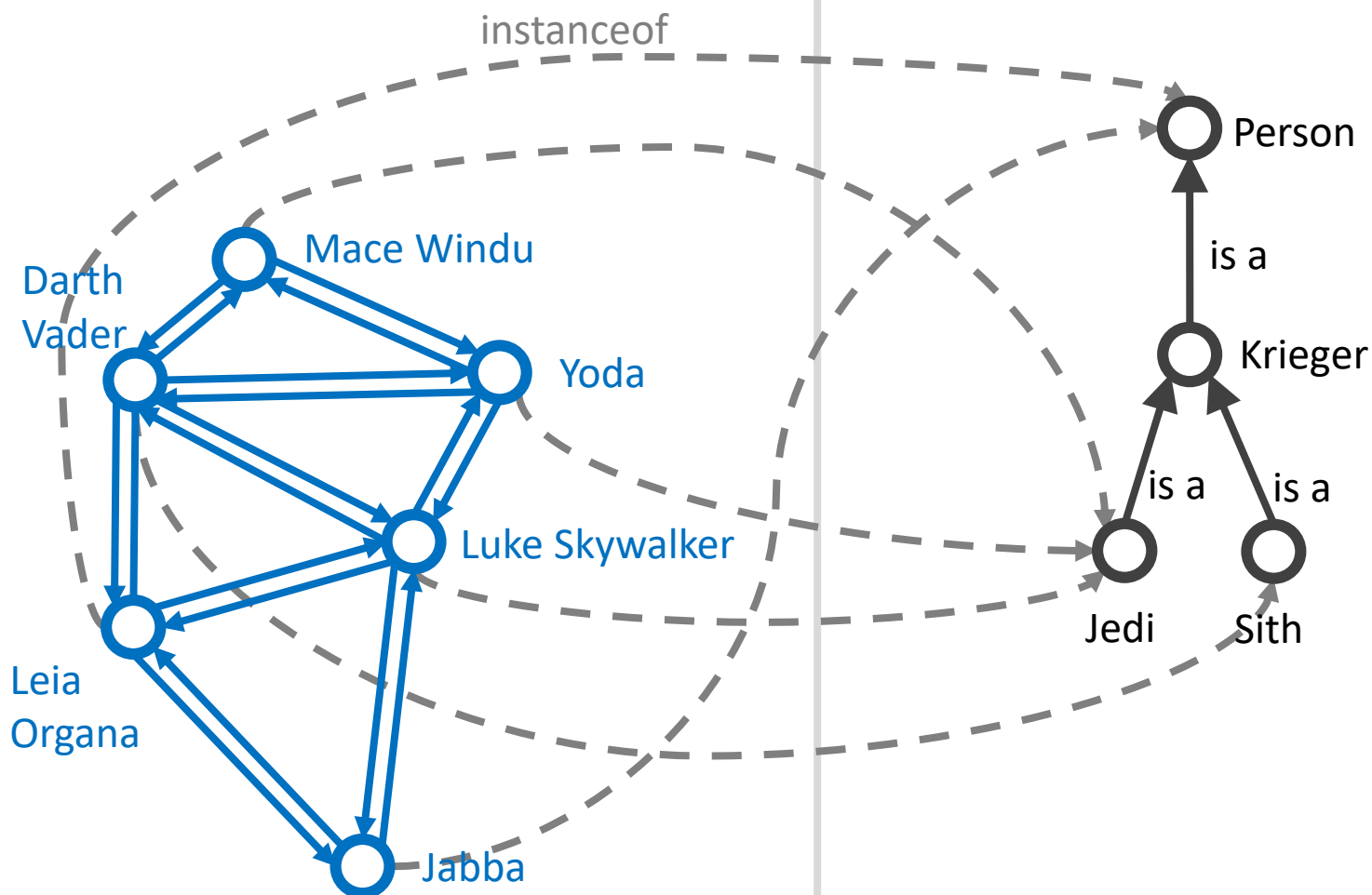
- Für jeden Knotentyp ein Typenknoten
- Für jeden Kantentyp eine Typenkante
- ...

Metamodell

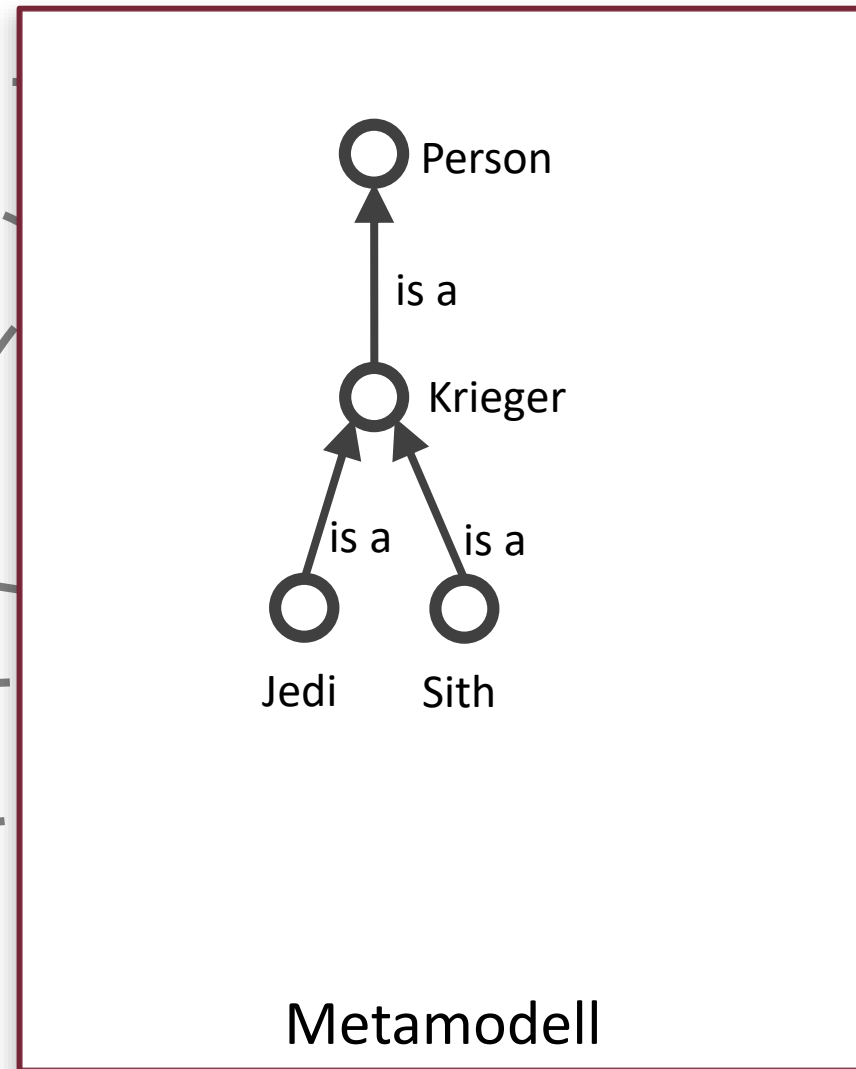
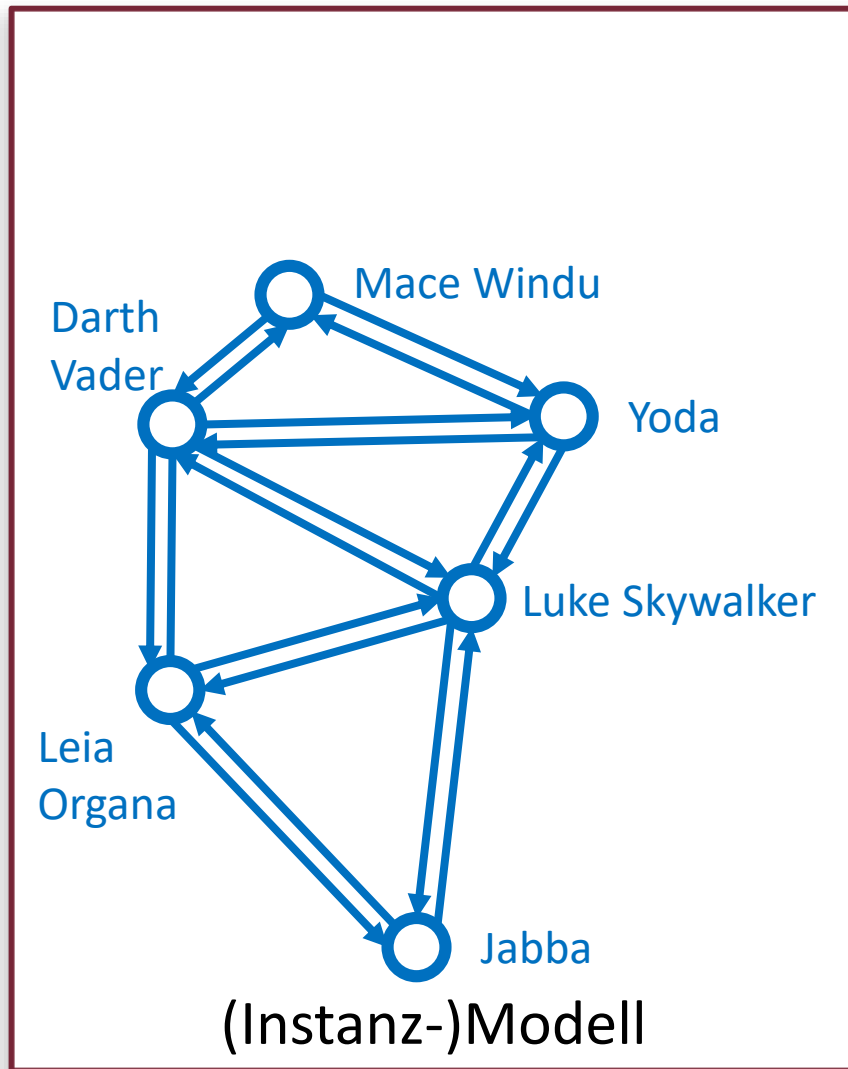


Hierarchische Dekomposition

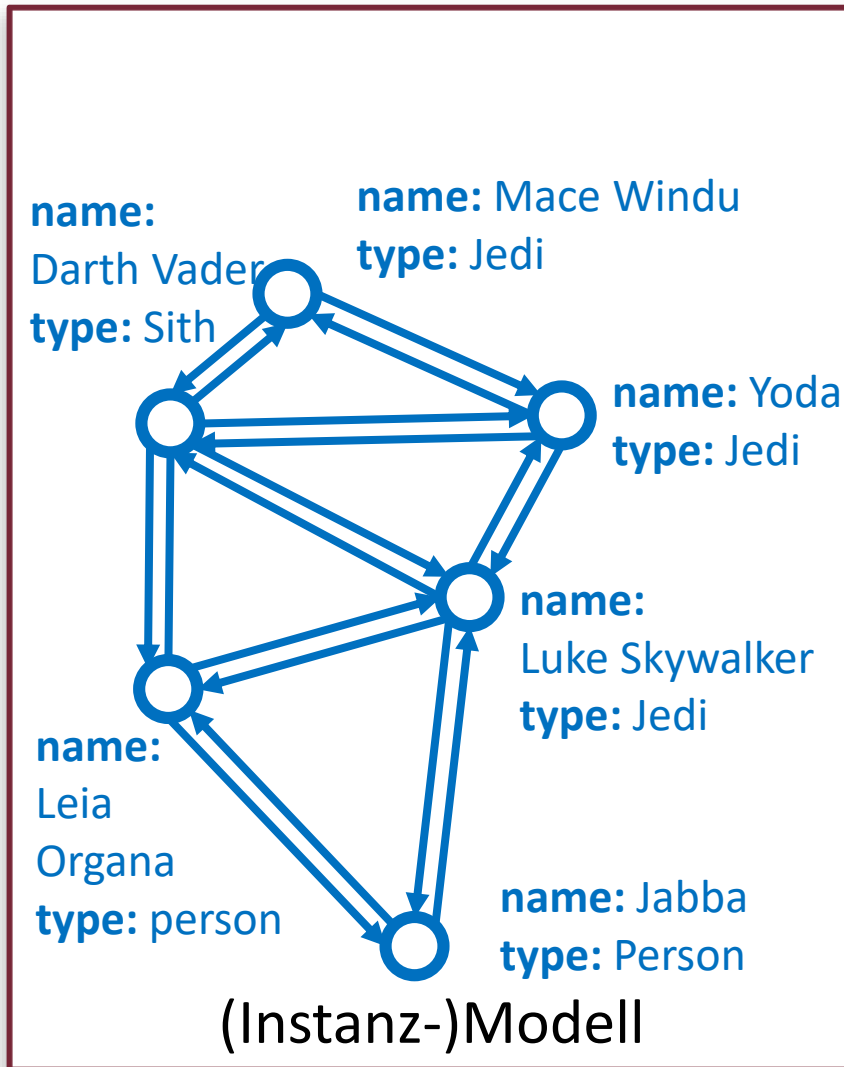
# Darstellung von Typ-Instanz-Verhältnissen



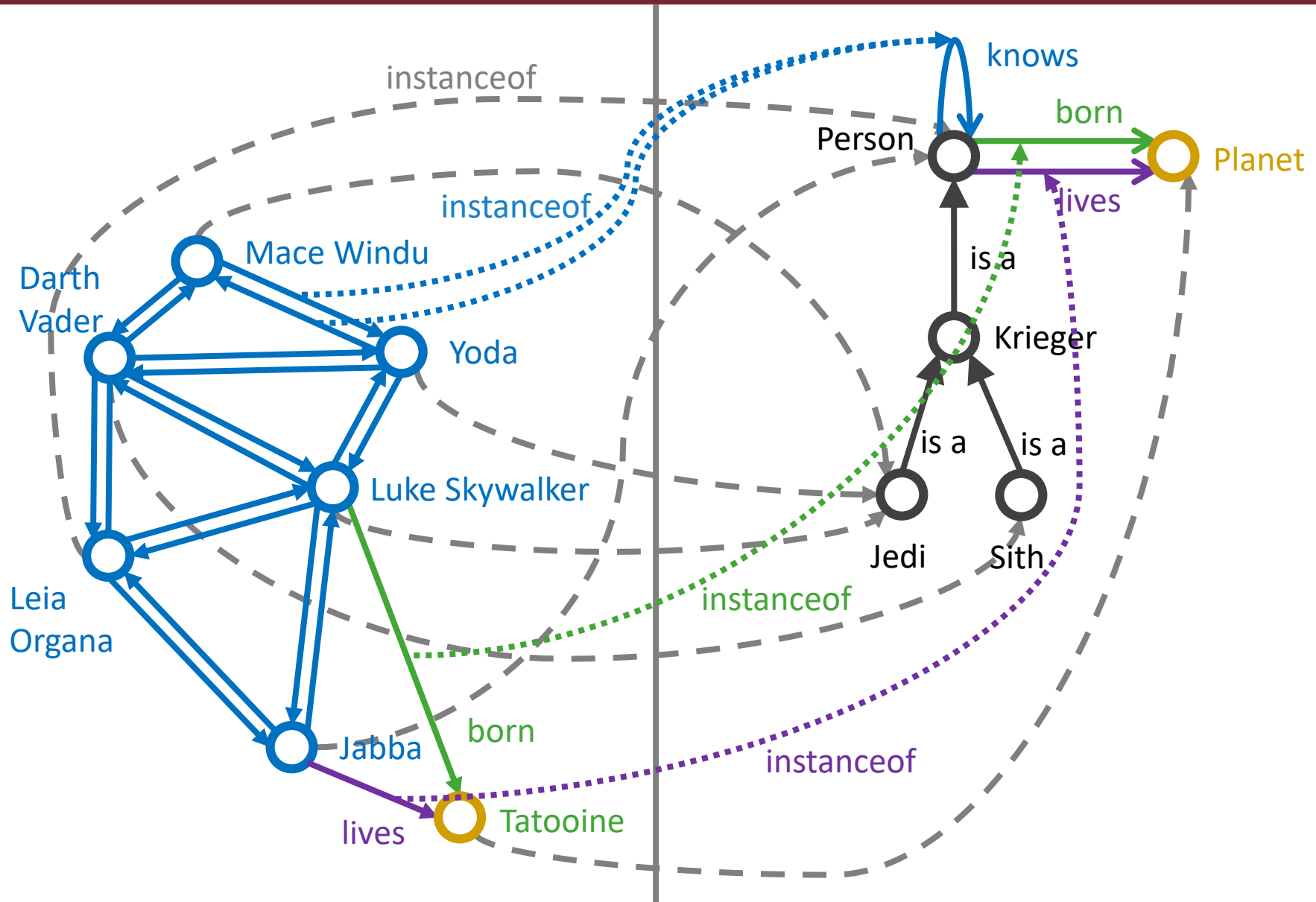
# Darstellung von Typ-Instanz-Verhältnissen



# Darstellung der Typen als Parameter



# Kanten mit verschiedenen Kantentypen



# REPRÄSENTATION DER STRUKTURELLEN MODELLE IN PROGRAMMEN

# Programmierungsparadigmen

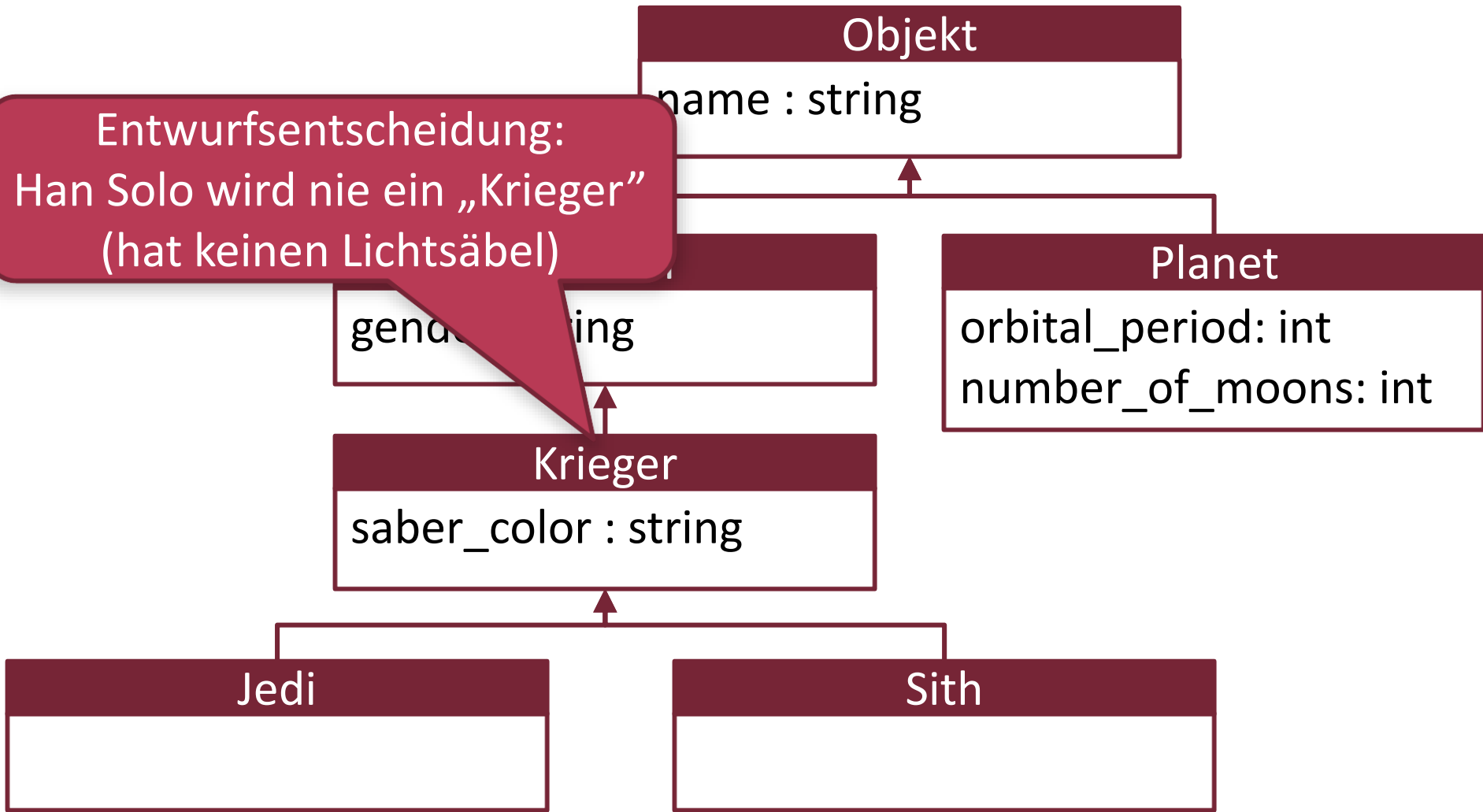
- **Programmierungsparadigma:**  
theoretisches Modell einer  
Programmierungssprache
- **Strukturierte Programmierung**  
(C, Pascal, Modula)
  - in Strukturen zusammengefassten Variablen: struct



# Programmierungsparadigmen

- **Objektorientierte Programmierung, OOP**  
(C++, Java)
  - Typ: *Klasse*
  - Instanz: *Objekt*
  - Merkmal: *Attribut*
  - Operation: *Methode*
  - Sichtbarkeit von Attributen/Methoden, Datenkapselung, usw.:  
siehe *Grundlagen der Programmierung 2*
  - Zeigerfelder vs. Enthalterelation
    - Begriffe: reference, composition, aggregation

# OOP: Vererbung



# Repräsentation der Modelle

- Textuelle Repräsentationen
  - XML, JSON, ...
- Graphische Repräsentationen
  - UML, AADL, SysML, EMF, ...

# Textuelle Repräsentation

- XML (Extensible Markup Language)
  - standardisierte, universelle Sprache für die Definition von Beschreibungssprachen
  - lesbar (auch für Menschen?)
- JSON (JavaScript Object Notation)
  - standardisierte Notation für lesbaren Datenaustausch

# XML-Beispiel: Wetter-Webdienstleistung

```
▼<current>
  ▼<city id="3054643" name="Budapest">
    <coord lon="19.04" lat="47.5"/>
    <country>HU</country>
    <sun rise="2015-02-17T05:45:24" set="2015-02-17T16:10:12"/>
  </city>
  <temperature value="268.061" min="268.061" max="268.061" unit="kelvin"/>
  <humidity value="83" unit="%"/>
  <pressure value="1034.42" unit="hPa"/>
  ▼<wind>
    <speed value="2.12" name="Light breeze"/>
    <direction value="52.0001" code="NE" name="NorthEast"/>
  </wind>
  <clouds value="0" name="clear sky"/>
  <visibility/>
  <precipitation mode="no"/>
  <weather number="800" value="Sky is Clear" icon="01n"/>
  <lastupdate value="2015-02-17T20:11:20"/>
</current>
```

# JSON-Beispiel: Google Maps API

```
{
  "results": [
    {
      "address_components": [
      ],
      "formatted_address": "1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA",
      "geometry": {
        "location": {
          "lat": 37.42291810,
          "lng": -122.08542120
        },
        "location_type": "ROOFTOP",
        "viewport": [
        ]
      },
      "types": [
      ]
    }
  ],
  "status": "OK"
}
```

# Graphische Modellierungssprachen

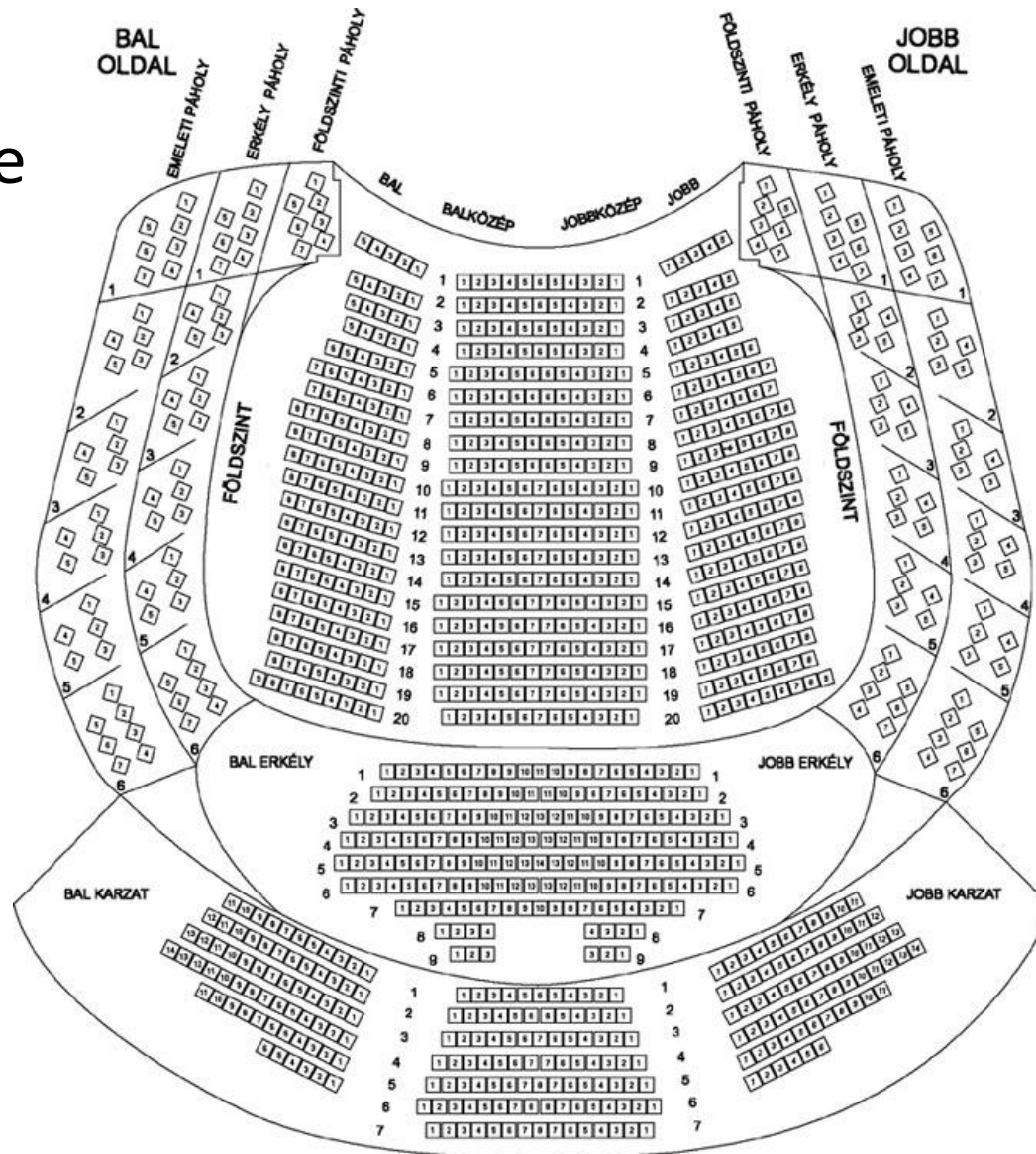
- UML (Unified Modeling Language)
  - eine universelle Modellierungssprache
  - in SW-Entwicklung sehr beliebt
- AADL (Architecture Analysis & Design Language)
  - für Beschreibung von Architekturen
- SysML (Systems Modeling Language)
  - UML-basierte allgemeine Modellierungssprache für Systemplanung, -Modellierung
- EMF (Eclipse Modeling Framework, Ecore)
  - für Beschreibung von Modellierungssprachen

# WEITERE ILLUSTRATIVE BEISPIELE



# Illustration – Strukturelle Modelle

## Geschachtelte Darstellung



# Illustration – Aufbauamodelle

## Rahmenstruktur

