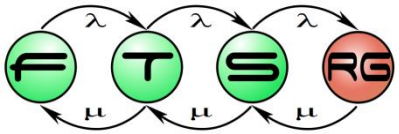
Structural Modelling

Budapesti Műszaki és Gazdaságtudományi Egyetem
Hibatűrő Rendszerek Kutatócsoport



Topics of the Lecture

- Goals and Applications of Structural Modelling
- Decomposition
- Description of the Model Elements by Graphs
- Property Modelling

STATIC MODELLING

How can the structure of complex systems be clearly modelled?

Examples:

- Architecture – Building
- Corporate network

Definition: Structural Model

Structural models are static. Their basis are (sub-)systems that are divided into their ingredients by the relation „**`Part of`**”.

Die **ingredients** can be the followings:

- further divided **subsystems** or
- further not divided (*elementary*) **components**.

The structural model represents the structure of the system according

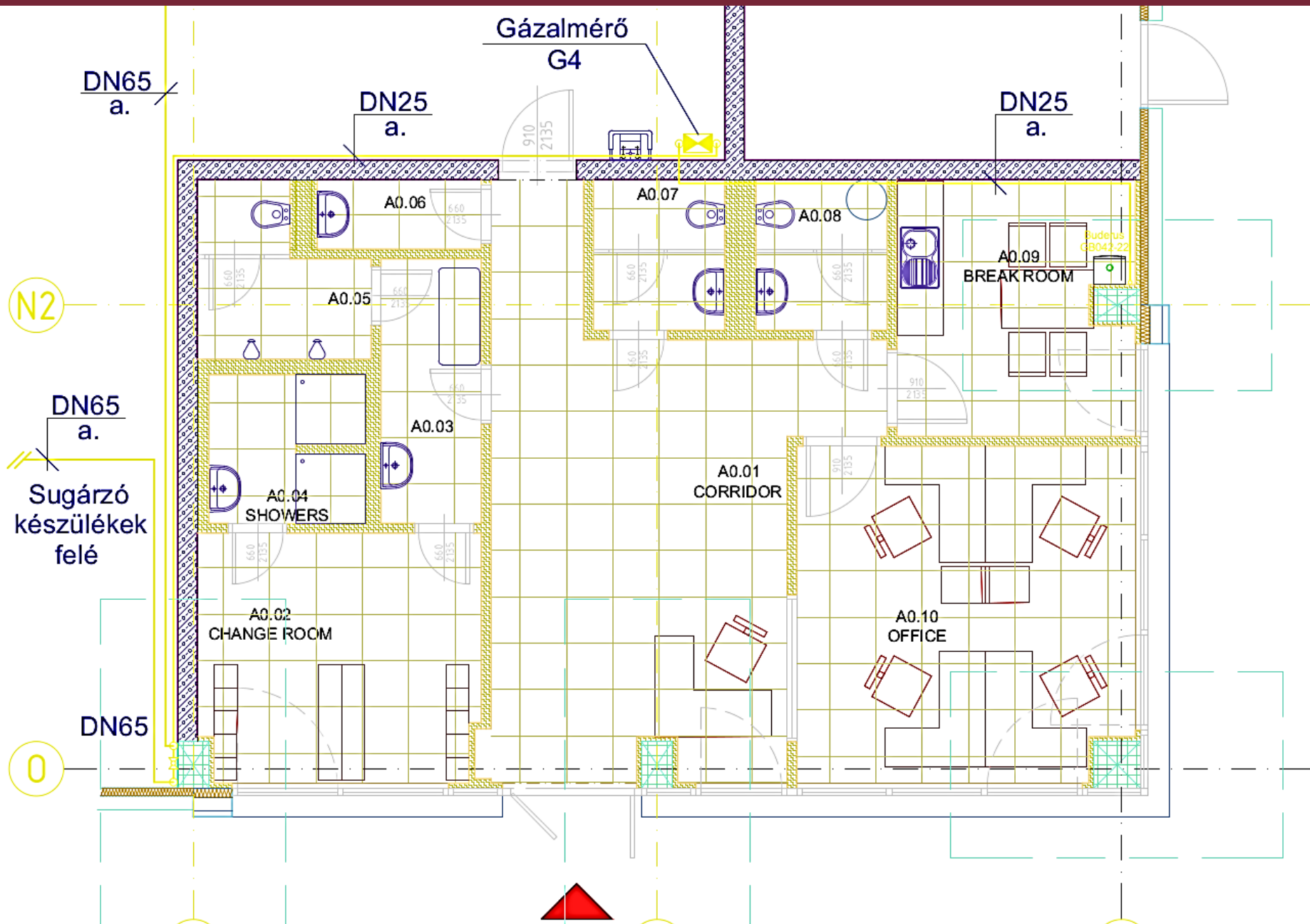
- its ingredients,
- the **properties** of the ingredients and
- their **relationships** among each other.

Example: Architecture Models

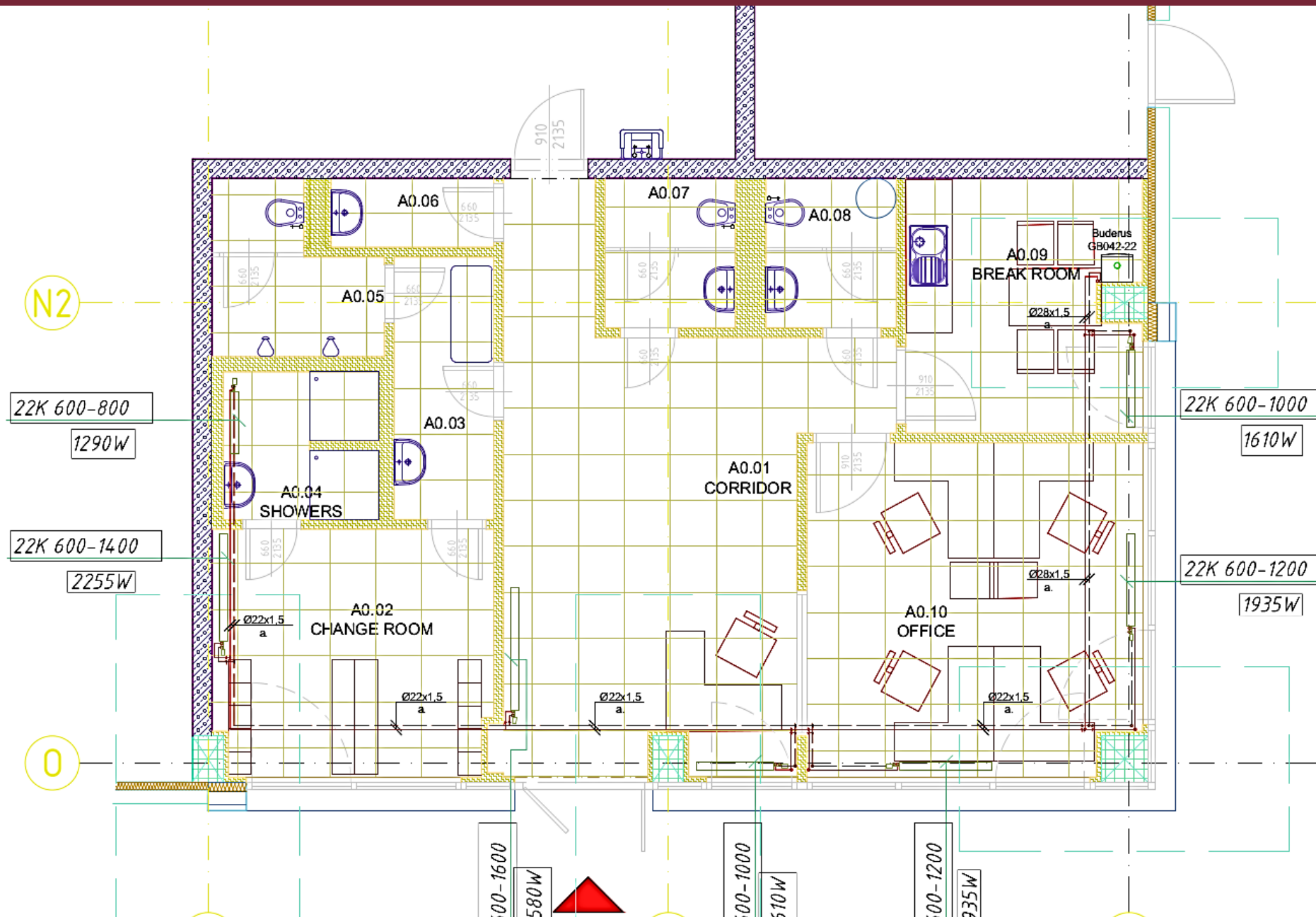
- BIM (Building Information Model)
- Modelling the whole building in a single model
- Views
  - Gas pipe system
  - Isolation
  - Water pipe and sewerage system
  - Electric installation plan
  - etc.

Office in a Production Hall

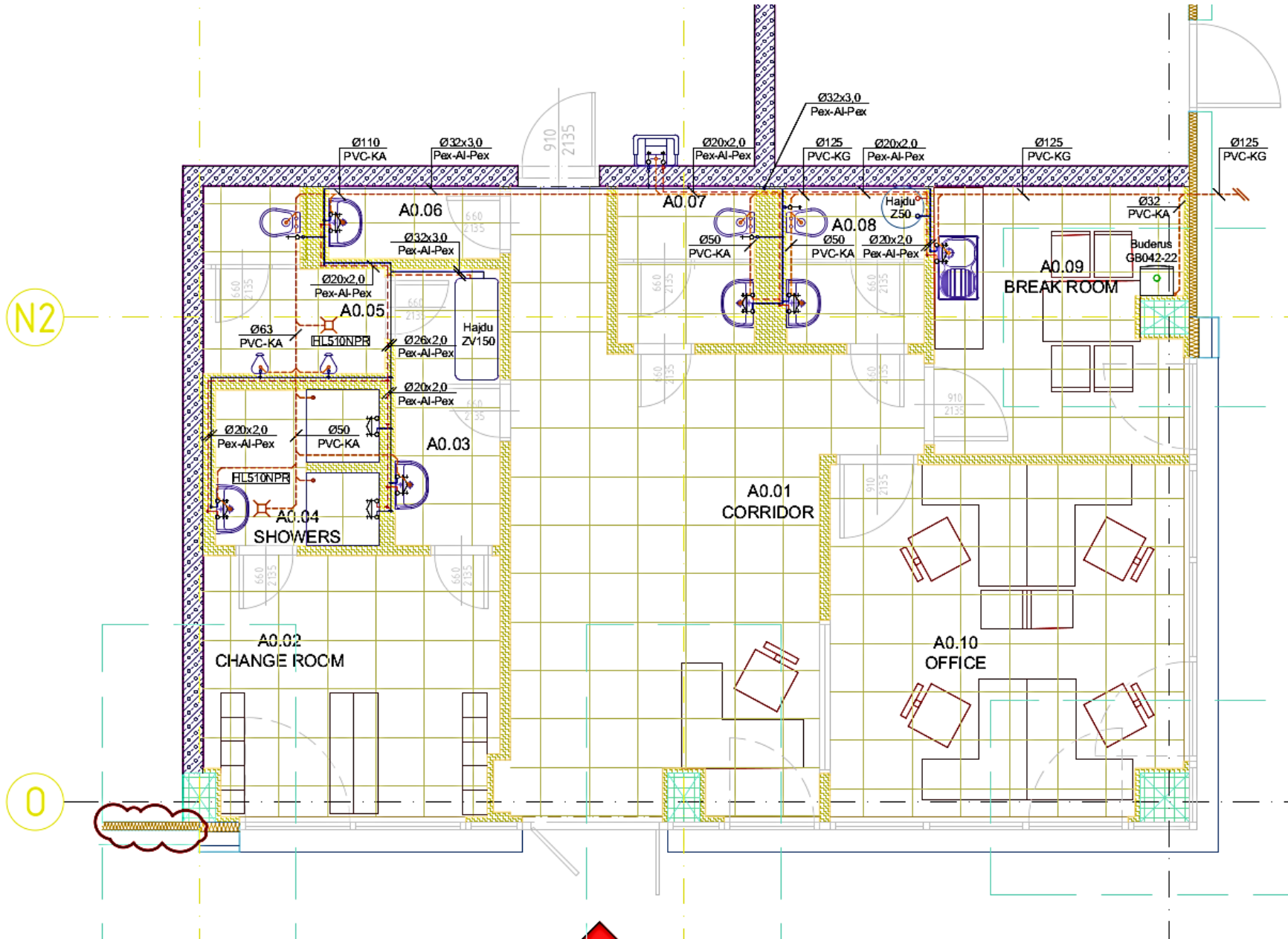
Gas Pipe System



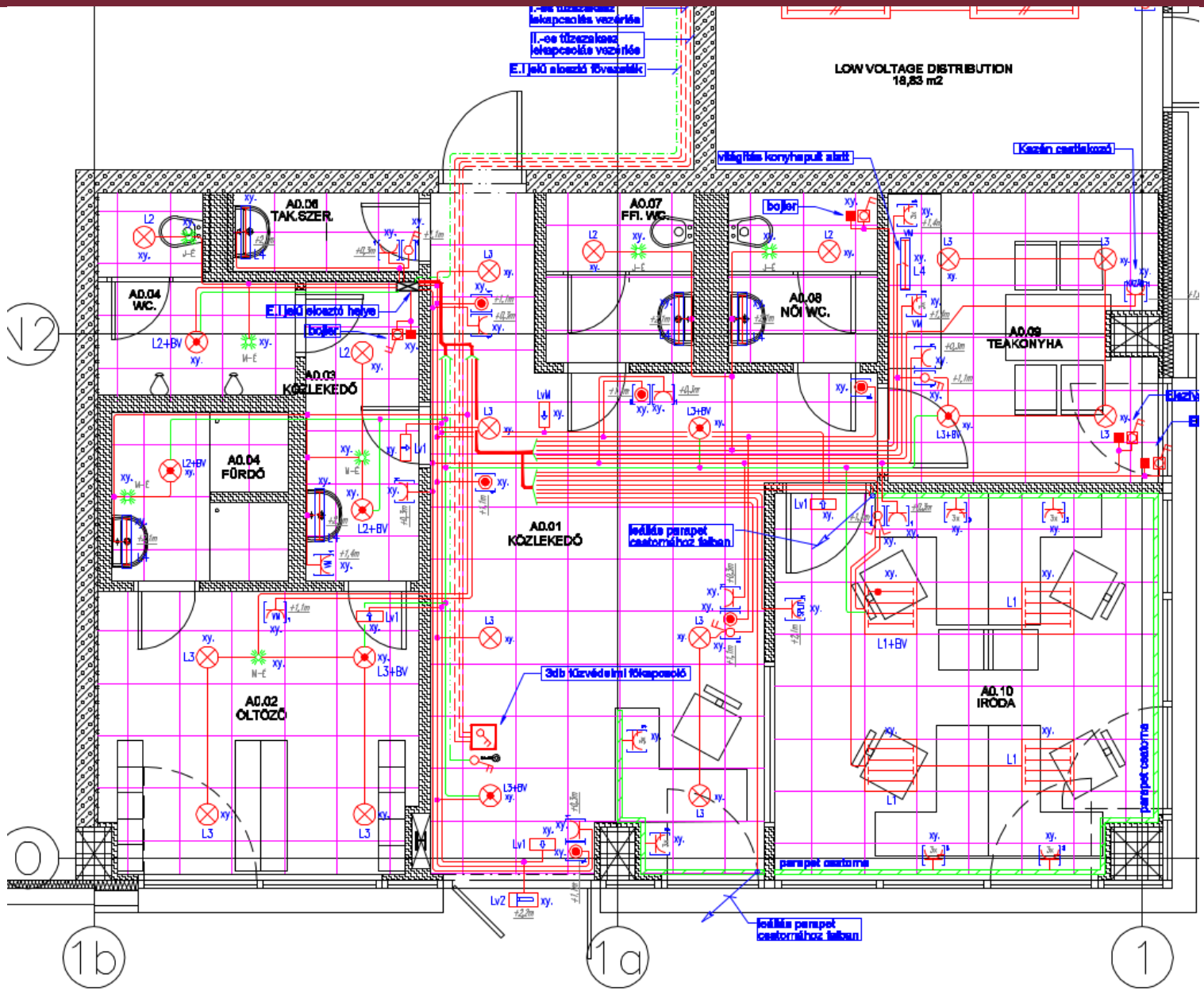
Isolation


N2
0
22K 600-800
1290W
22K 600-1400
2255W
22K 600-1000
1610W
22K 600-1200
1935W
A0.01
CORRIDOR
A0.02
CHANGE ROOM
A0.03
A0.04
SHOWERS
A0.05
A0.06
A0.07
A0.08
A0.09
BREAK ROOM
A0.10
OFFICE
Buderus
GB042-22
Ø28x1.5
a.
Ø22x1.5
a.

Water and Sewerage System



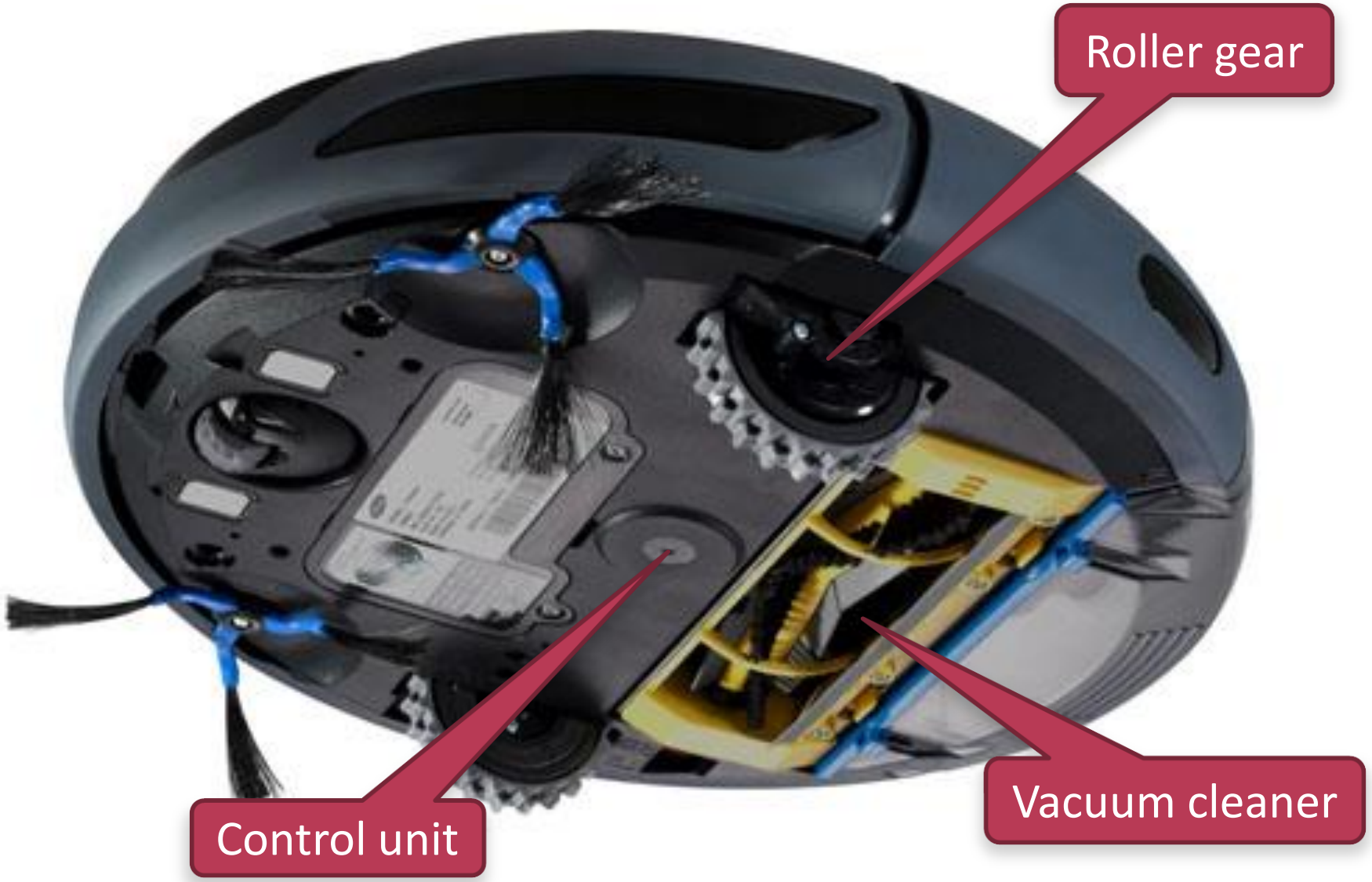
Electric Installation Plan



Goal of Structural Modelling

- Dividing a system
  - It is simpler to design smaller units
  - Usage of already existing parts
  - Usage of commercial components (COTS - Commercial off-the-shelf)
- Documentation of an existing system
  - „System overview”
- Defining data structures
  - What pieces of information should be there?
  - Relationships of the pieces

Example: Robot Vacuum Cleaner



Structural and Behavioural Modelling

- Structural
  - static
  - whole and part, components
  - connections
- Behavioural
  - dynamic
  - timeliness
  - states, processes
  - reactions to the environment (context)
- This categorization is neither full nor disjoint ...

The main components of the robot vacuum cleaner are the control unit, the roller gear and the vacuum cleaner.

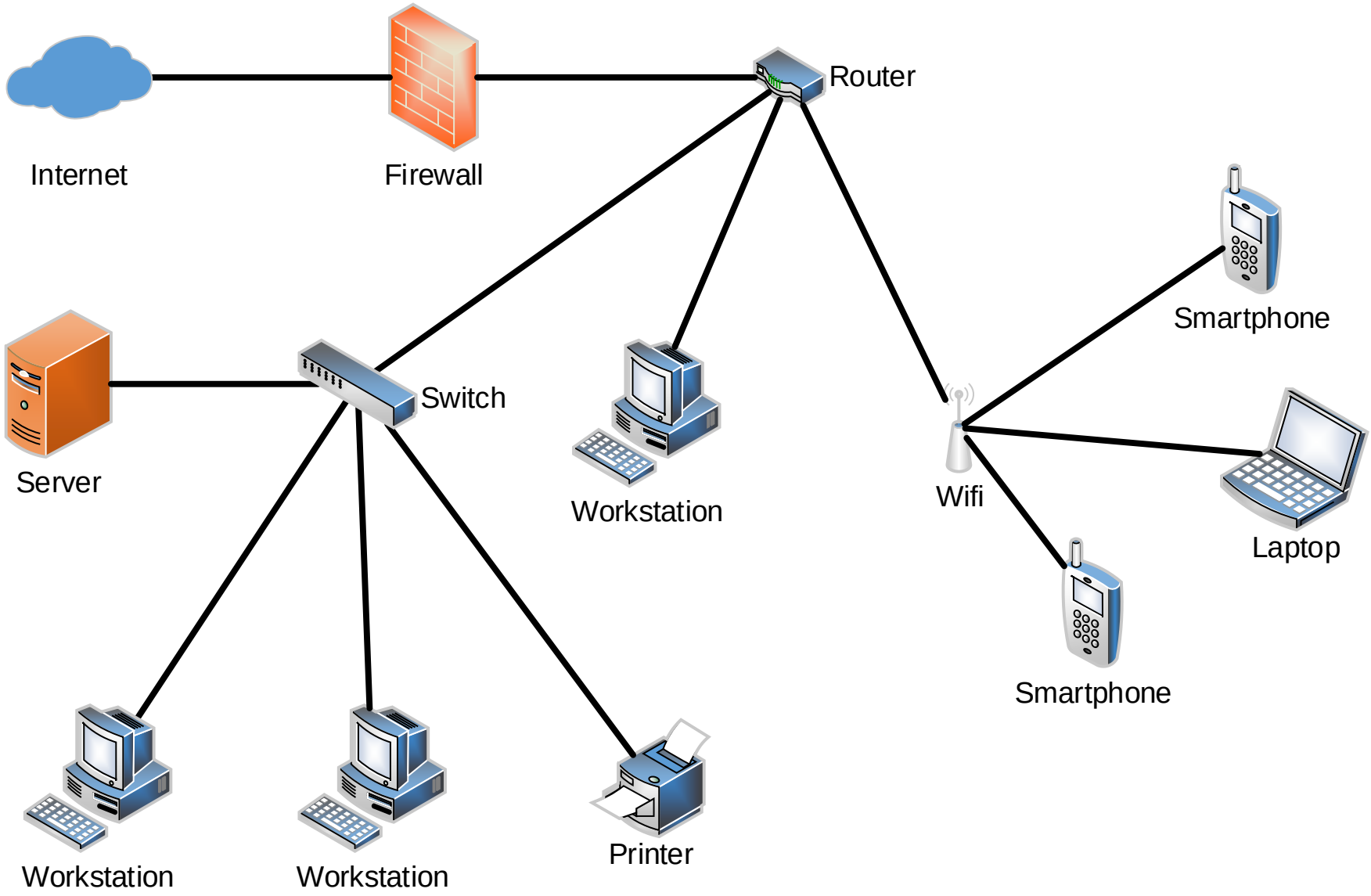
For the command „to right” changes the roller gear its operational mode to „turn”.

The roller gear of the robot intervenes when signals of the sensors are received. (When, how?)

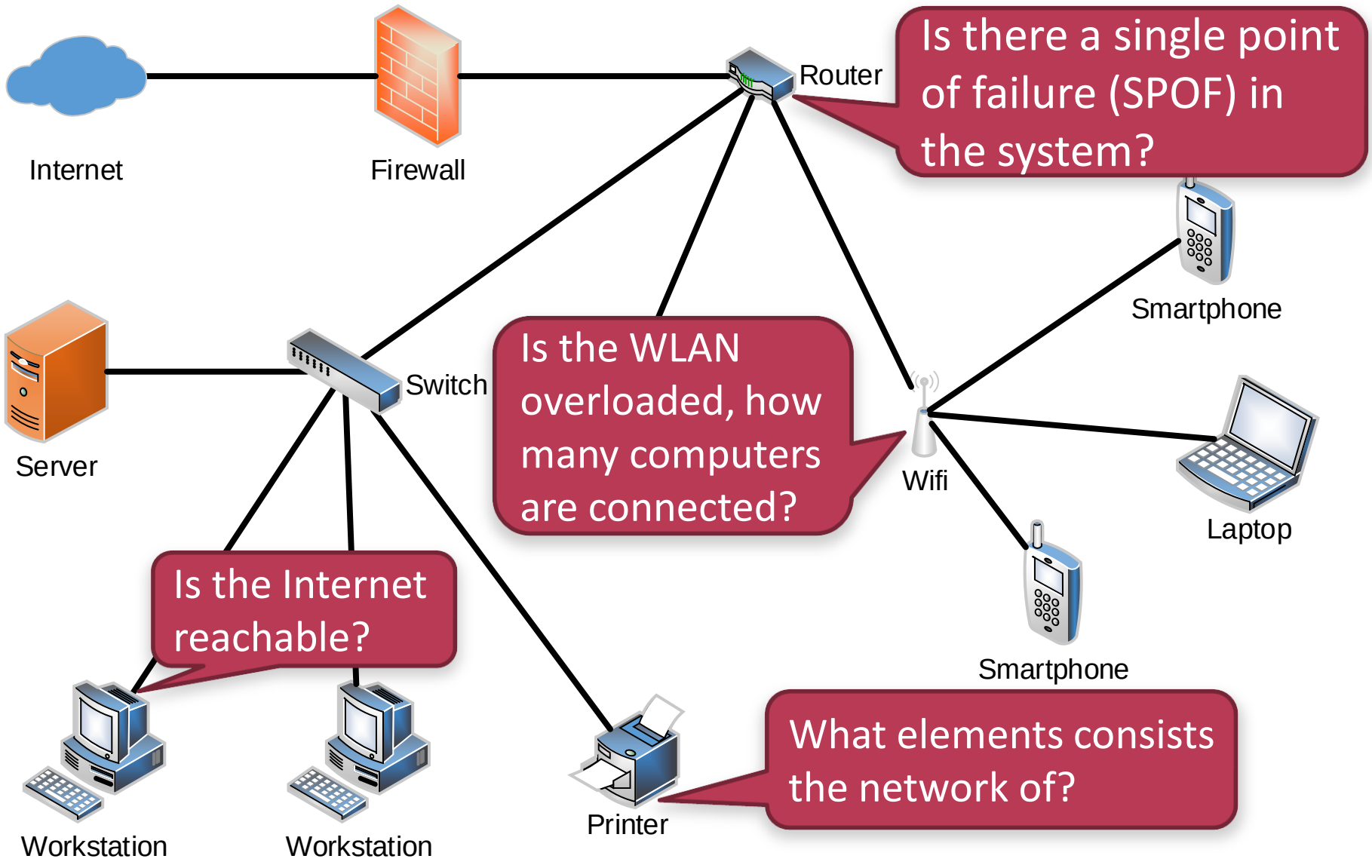
Structural Model

- The knowledge about the structure of the system
  - What parts consists the system of?
  - How are they connected?
  - What kind of properties do the elements have?

Example: Corporate Network



Example: Corporate Network



STRUCTURAL DECOMPOSITION

What containedness relationships are there among the subsystems?

Examples:

- Robot vacuum cleaner
- Registration office

Definition: Decomposition

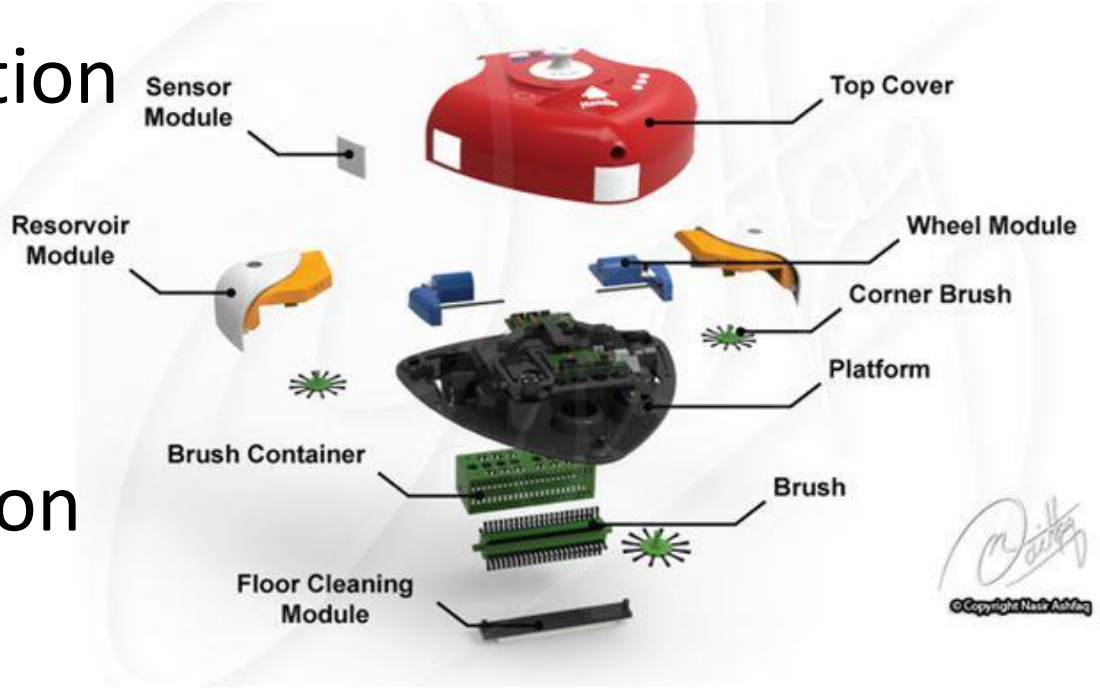
„**Structural models** are static. Their basis are (sub-)systems that are divided into their ingredients by the relation „**`Part of`**”.

Decomposition („faktoring”) is the division of a complex problem or system into smaller parts that are easier to conceive, understand, design, implement/program and maintain.

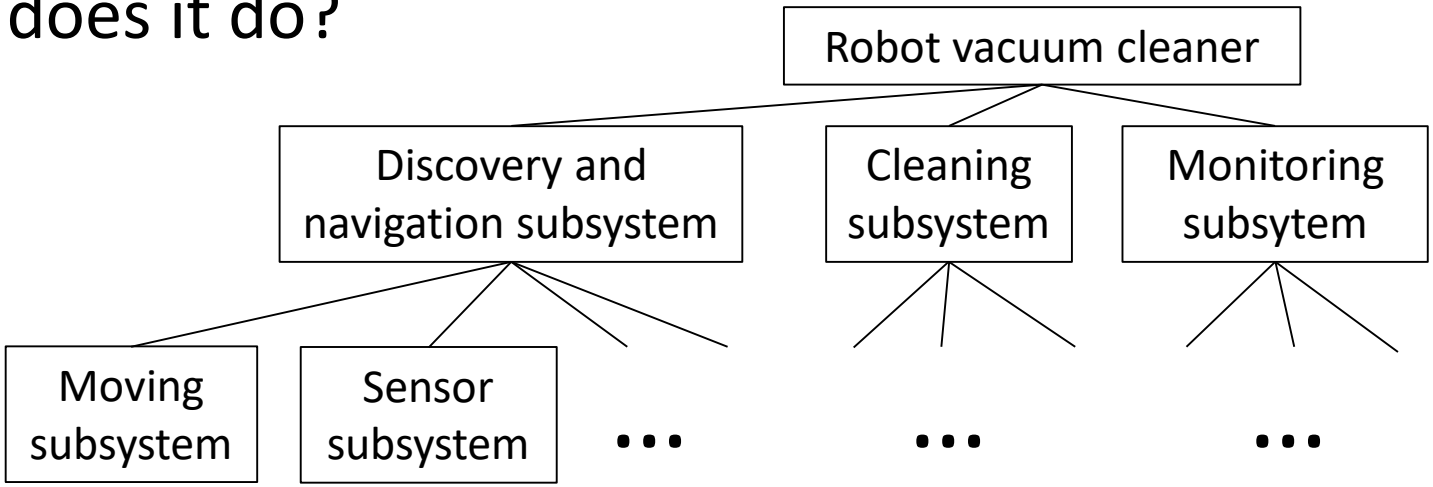
- Hierarchical Decomposition:
  - Multi-level whole-part relationship
- Meaning of the relation `Part of`:
  - Physical Decomposition: `Part of` = spatial division
  - Logical Decomposition: `Part of` = functional division

Types of Decomposition

- Physical decomposition
  - What pieces does it consists of?



- Logical decomposition by functions
  - What does it do?



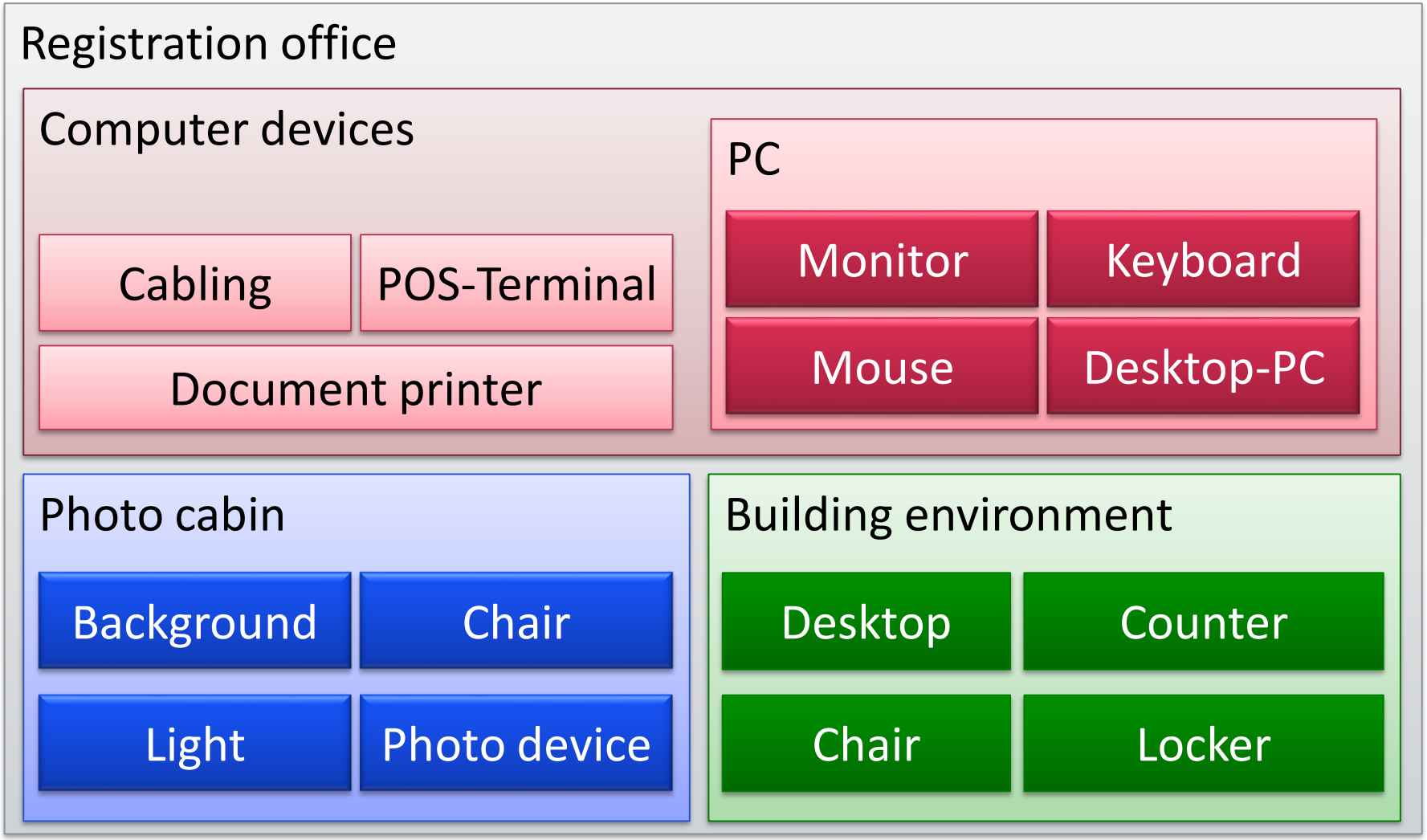
Definition: Correctness of Decomposition

A decomposition is **correct**, if

- each element of the system resulting from the decomposition corresponds to an element of the original system, and
- each element of the original system corresponds to some elements of the system resulting from the decomposition.

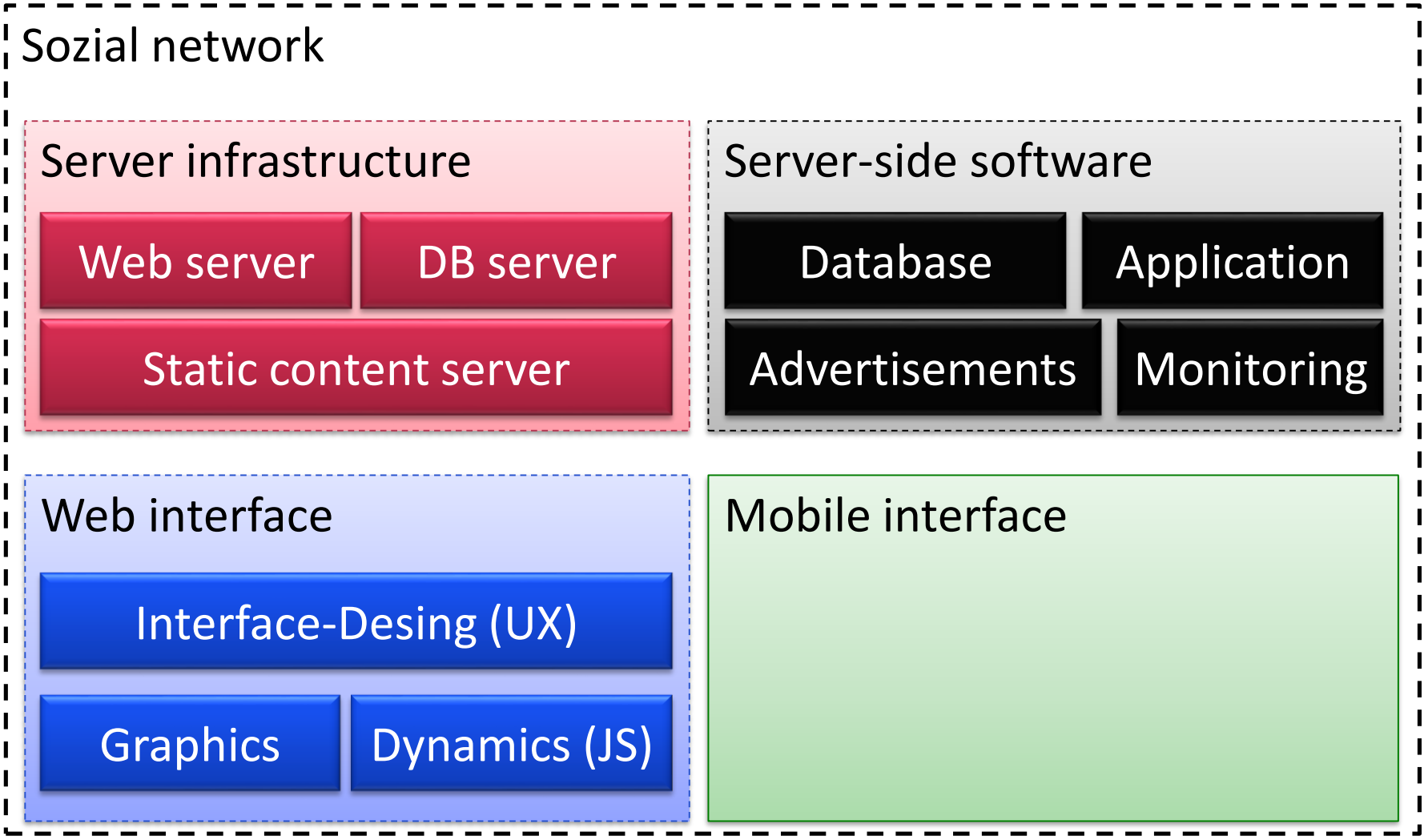
Top-down Development

- Elementary step: Decomposition



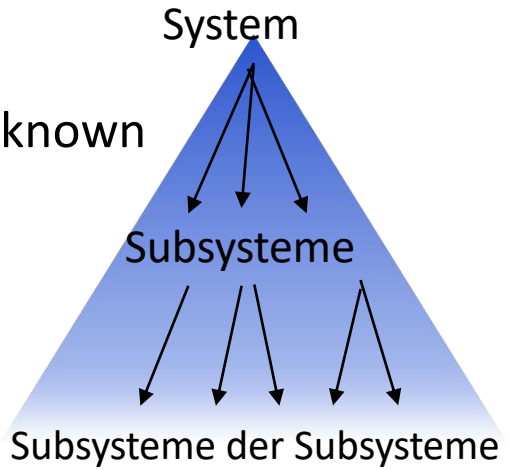
Bottom-up Development

- Elementary step: Composition

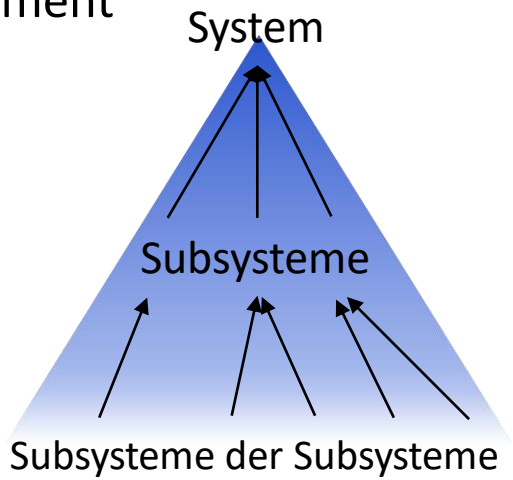


Top-down and Bottom-up

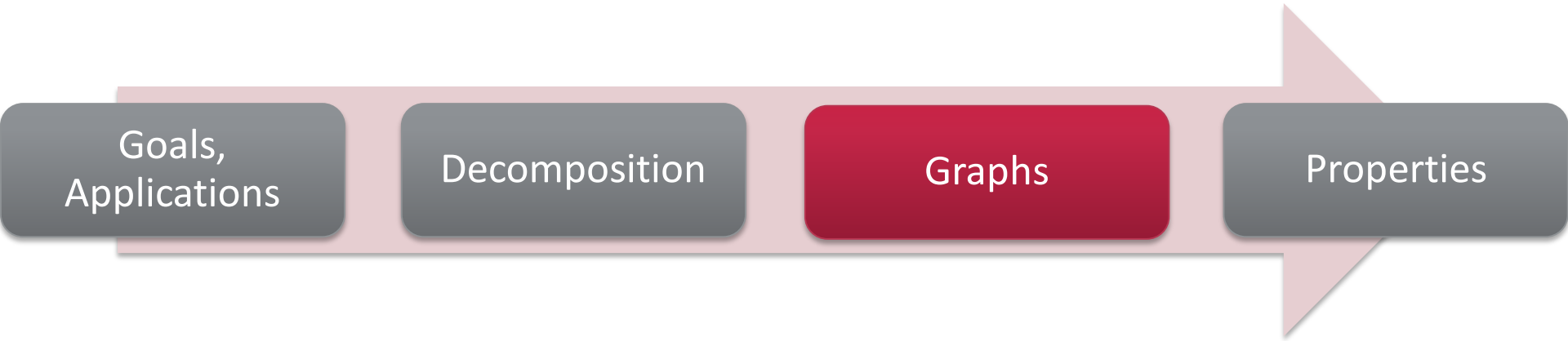
- Top-down
  - ☺ Roles of the subsystems are during development already known
  - ☹ There are no functioning parts during development
  - ☹ Problems/Reqs of the subsystems are revealed late



- Bottom-up
  - ☺ Subsystems can be tested step-by-step
  - ☺ There are always some functioning parts during development
  - ☹ Exact roles of the subsystems are revealed late



- Not only in structural modelling
- Mixed approaches, iterative Development



STRUCTURAL MODELS

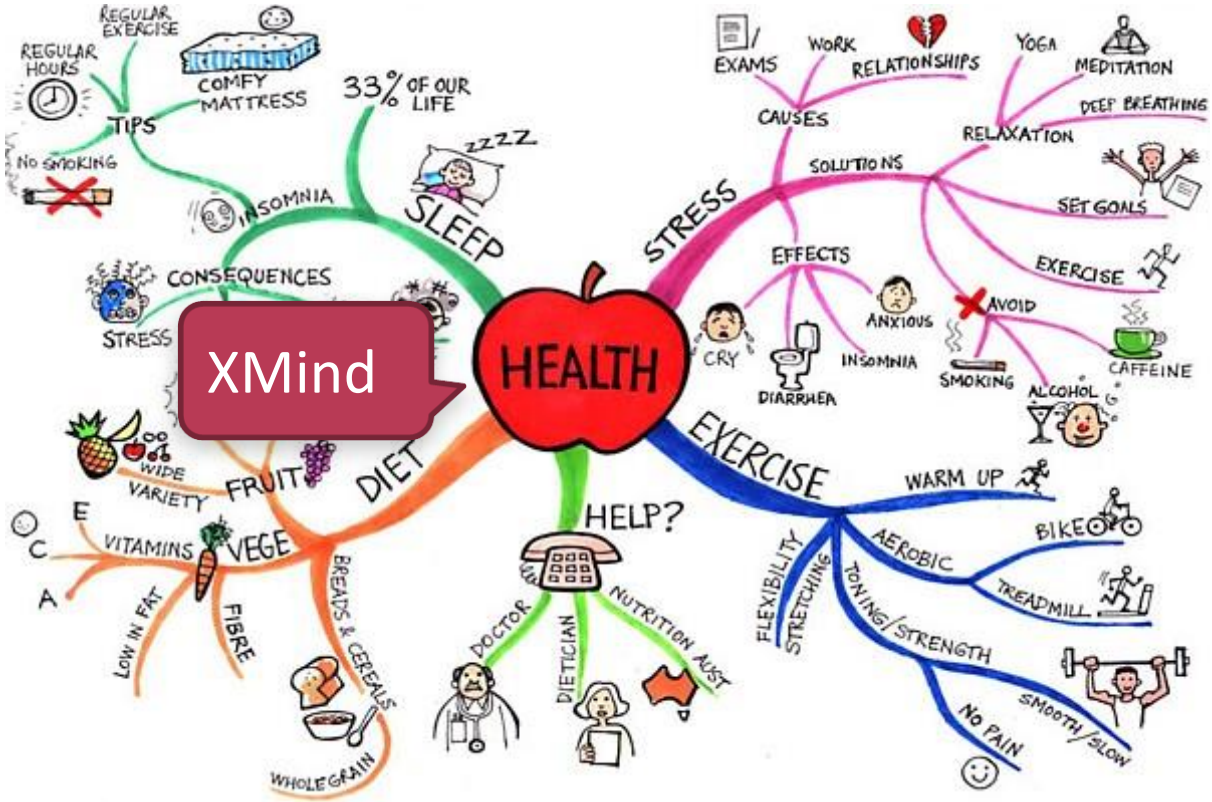
What kind of relationships are there among the subsystems?

Examples

- Transport network
- File system

Graph Representation

- Human way of thinking: ,,Connections of Things”
  - **Things:** persons, airports, areas
  - **Connections:** dependency, acquaintance, flights, containment
- Mathematical Formalism:
  Graph
  - Nodes
  - Edges
  - (Properties)



Definitions in Computer Engineering

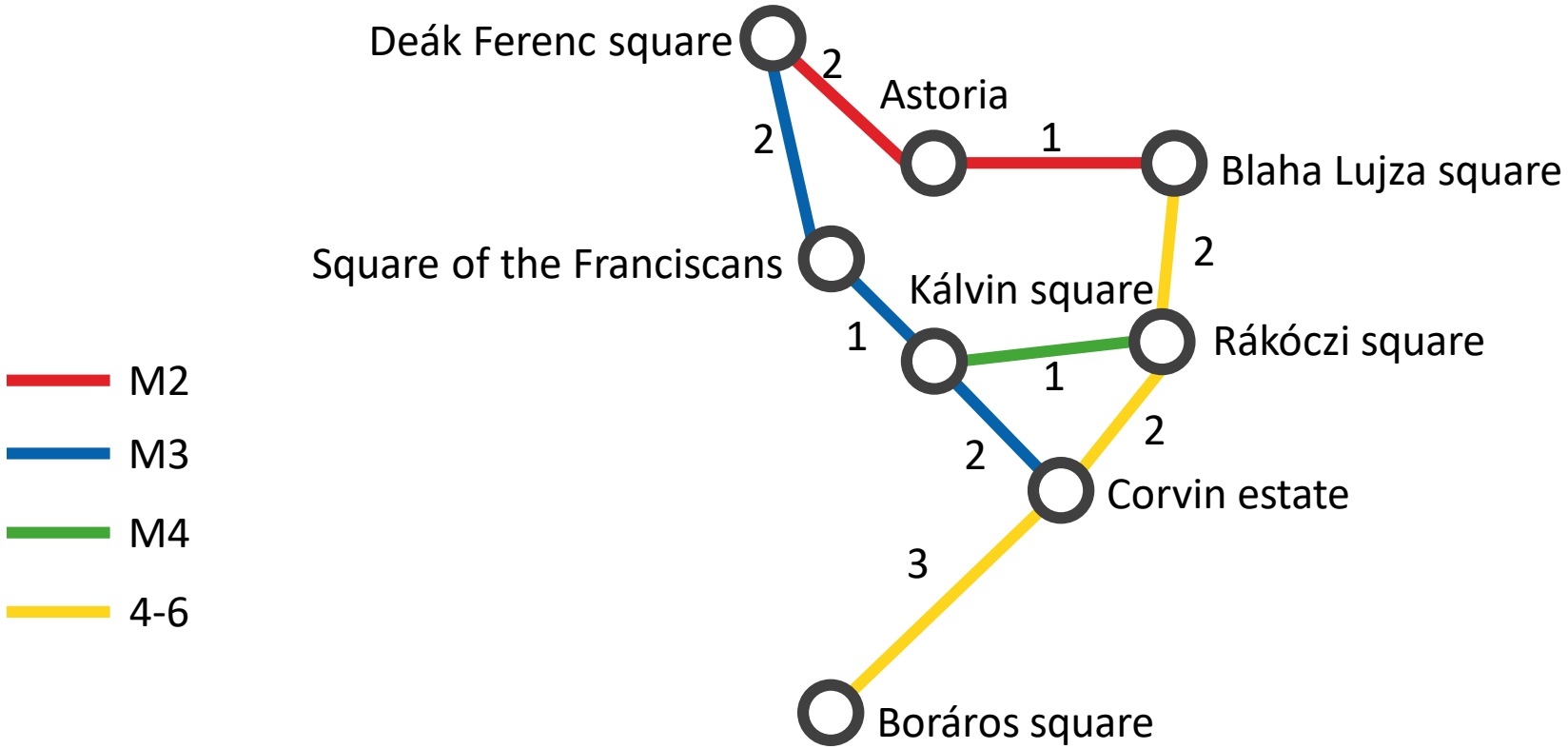
- In the computer engineering everything has several, often contradicting definitions
  - System = ?
  - Model = ?
- The same term is often named with different names
  - node, vertex, object, concept
  - edge, link, arc, connection, relationship

Always pick one!

Budapest – Rail-bound Transport Network

M2
M3
M4
4-6

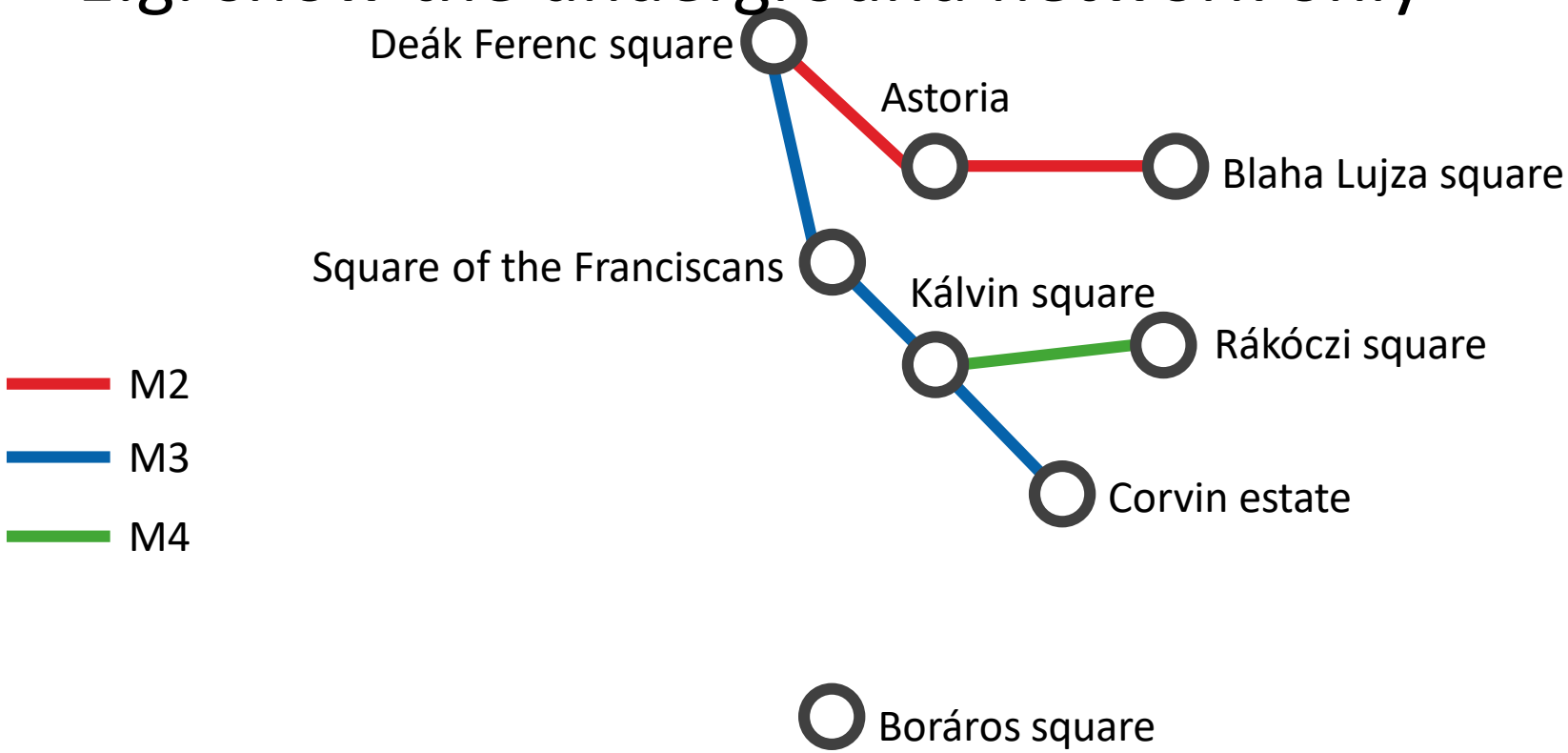
Budapest – Rail-bound Transport Network



Shortest path?

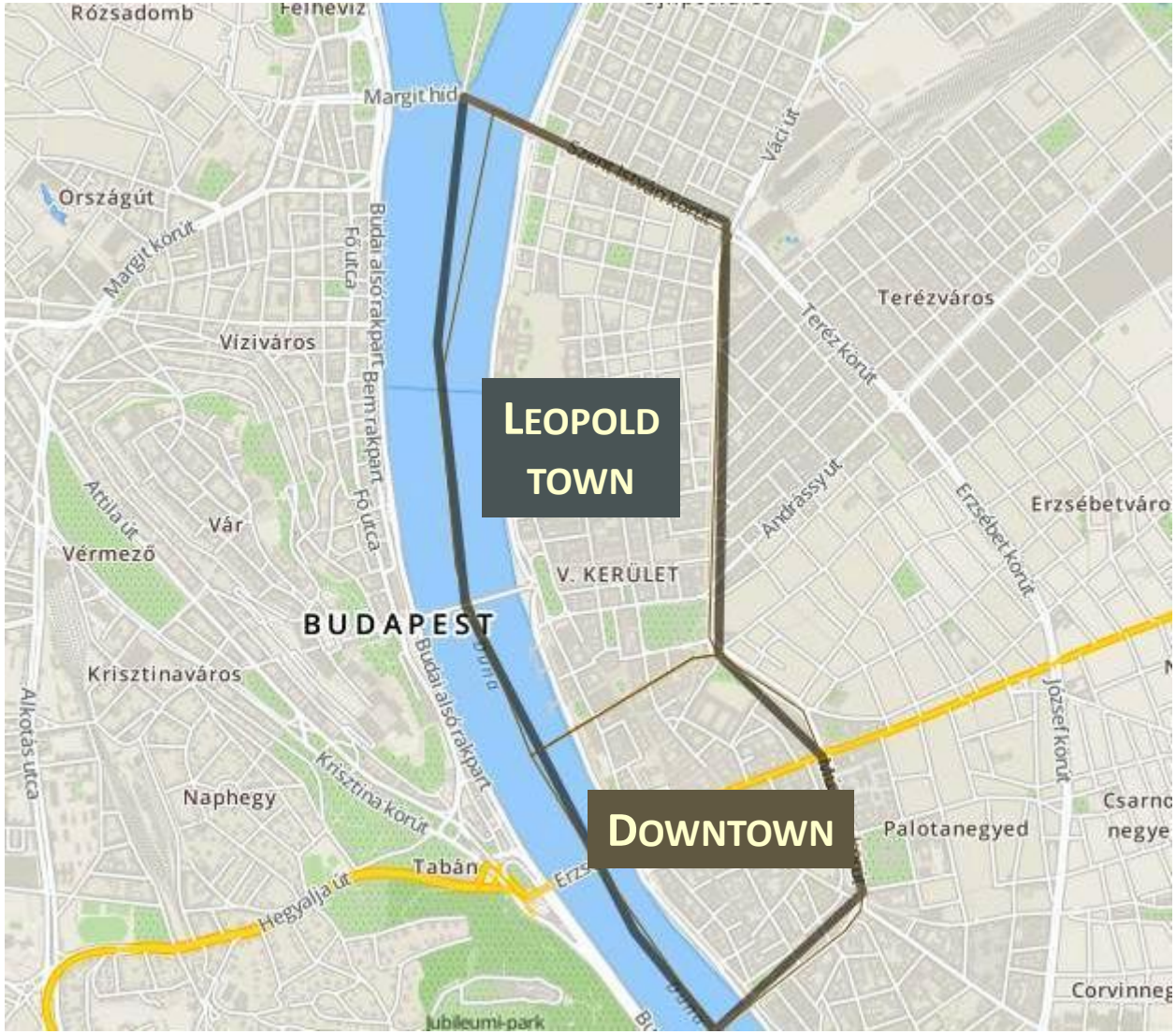
Filtering: by edge labels (Subgraphs)

- E.g. show the underground network only

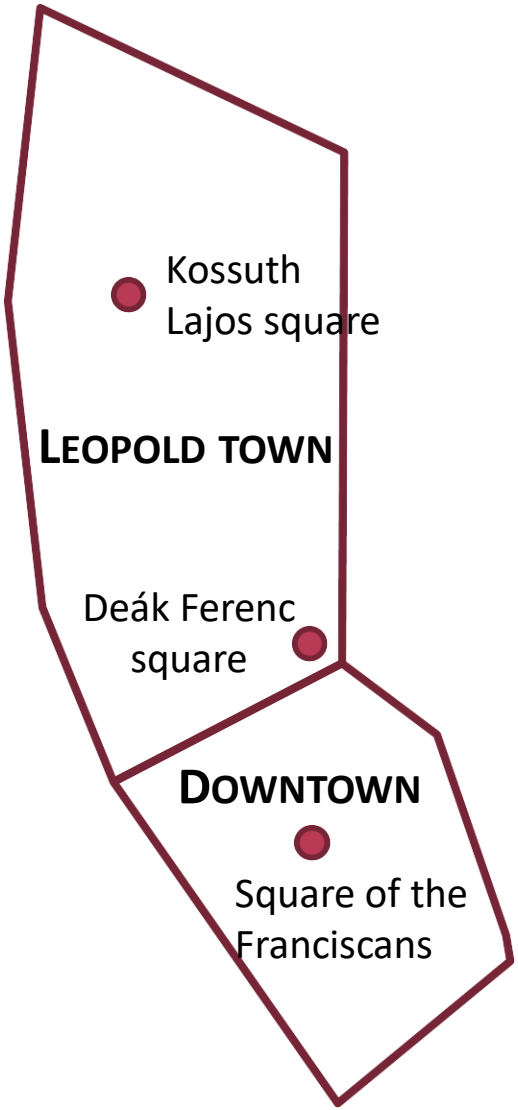


- Was is reachable by the underground only?

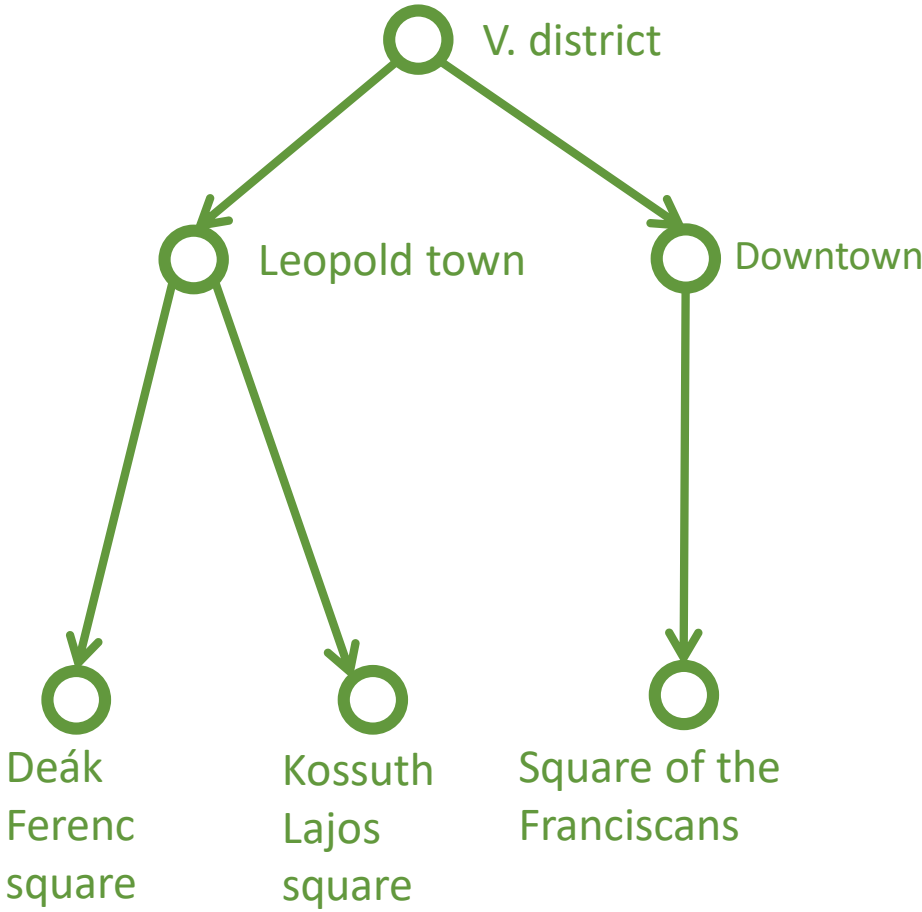
Budapest V. district



Budapest V. district – Hierarchical Model



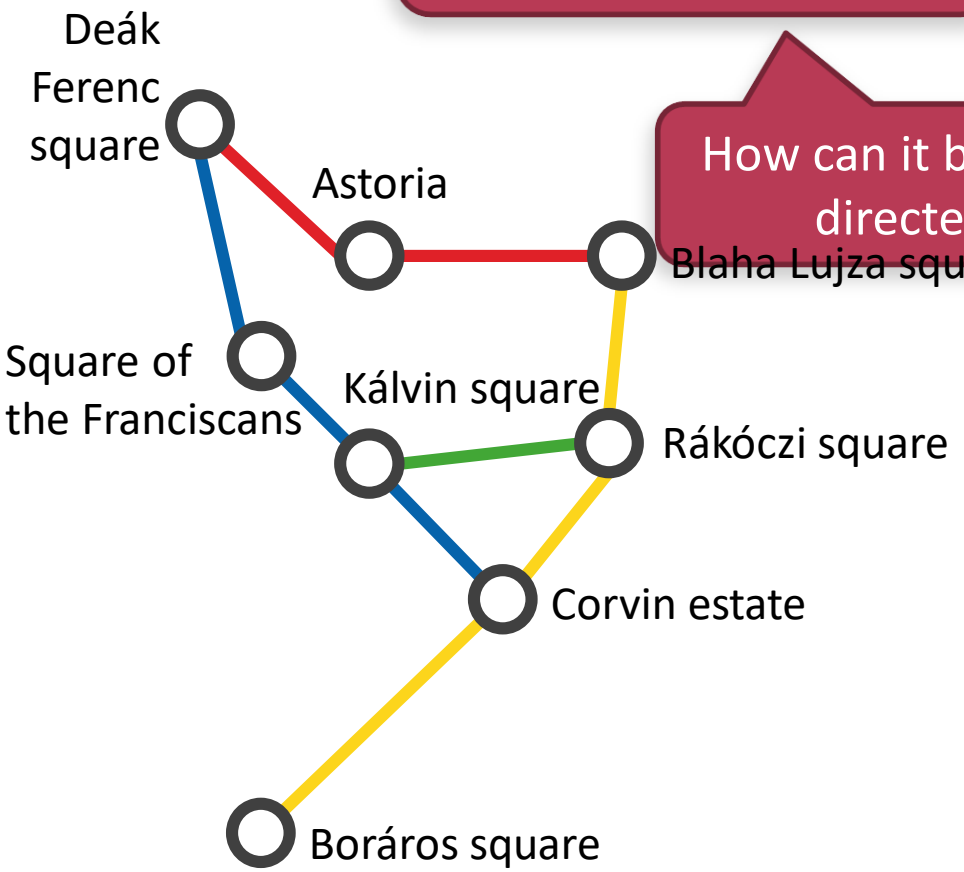
≡



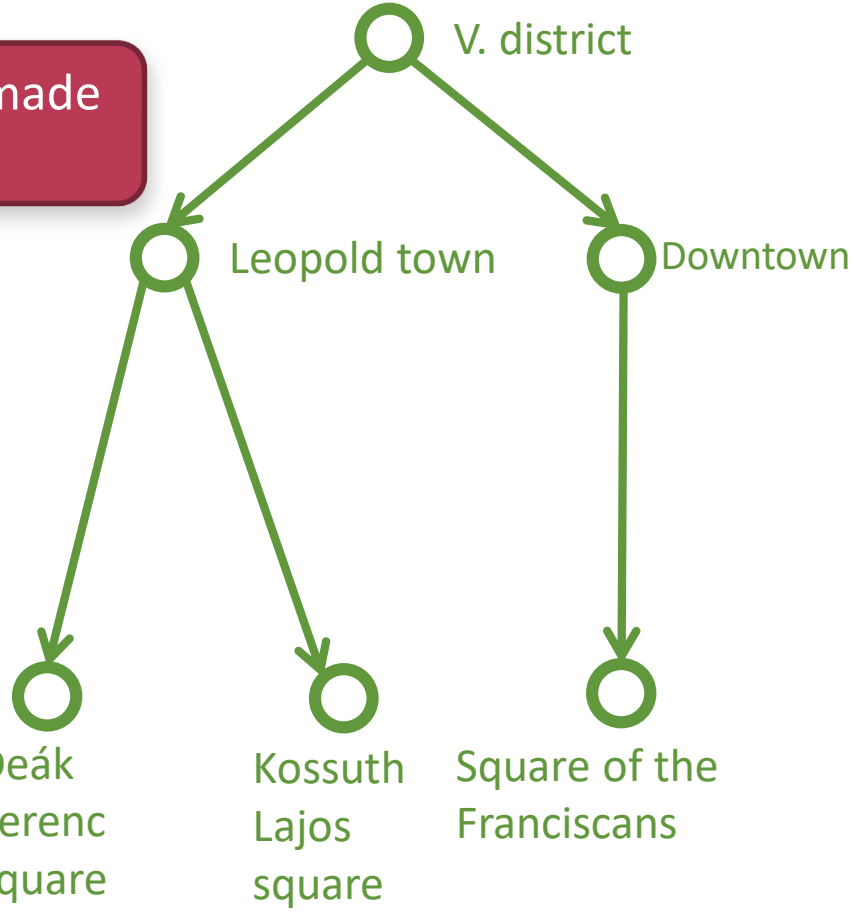
Simple Graphs

Transport network: undirected graph

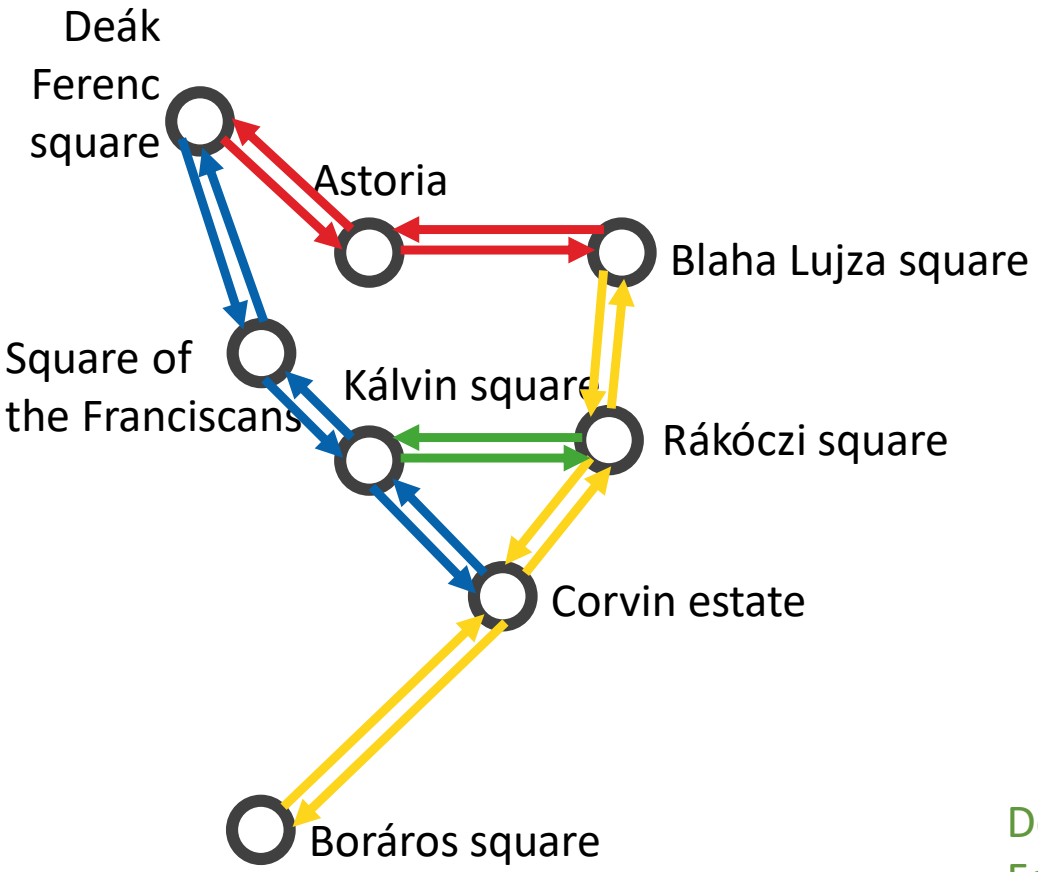
Areas: directed graph

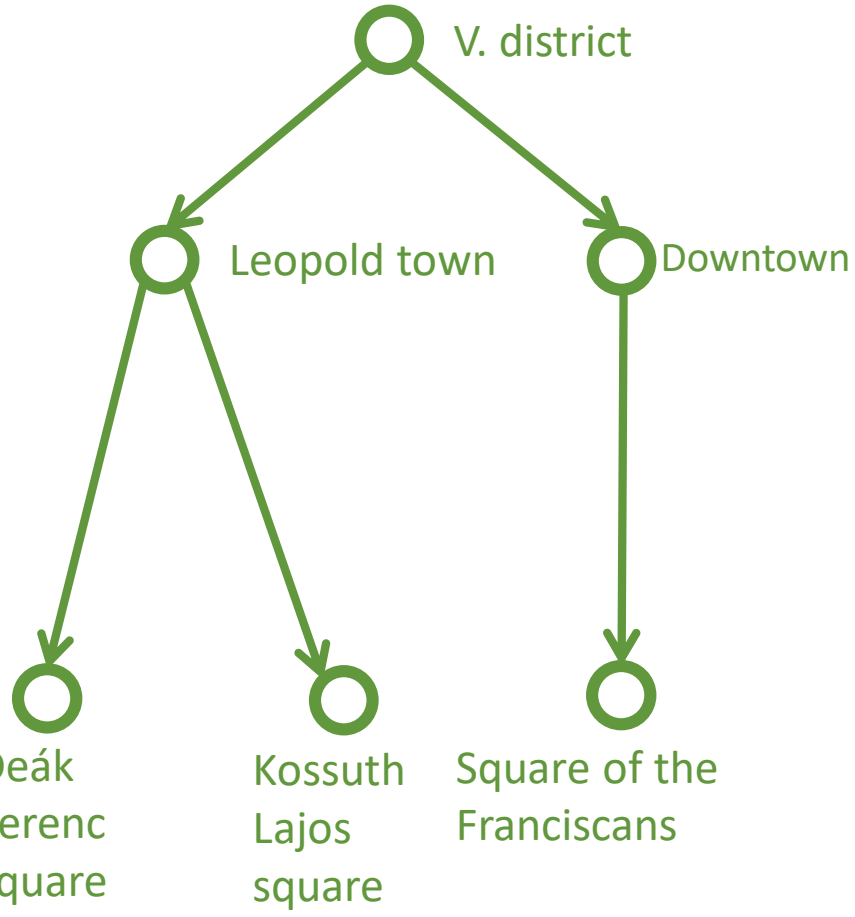


How can it be made directed?

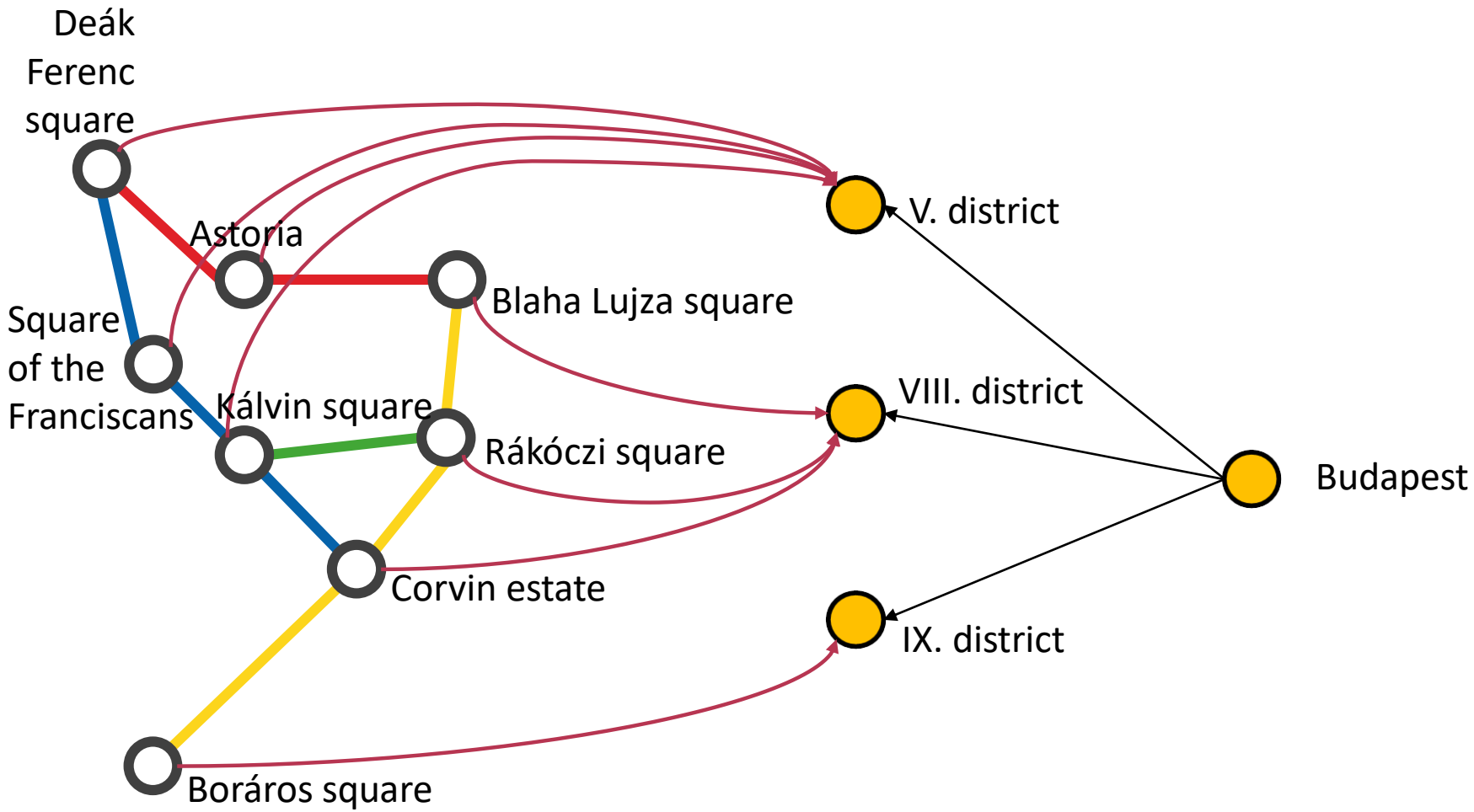


Simple Graphs

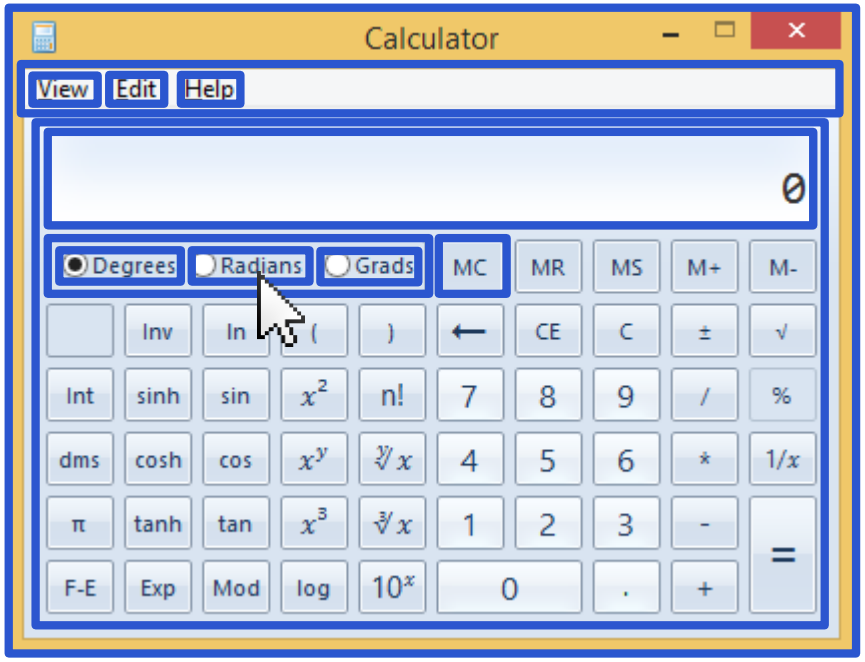




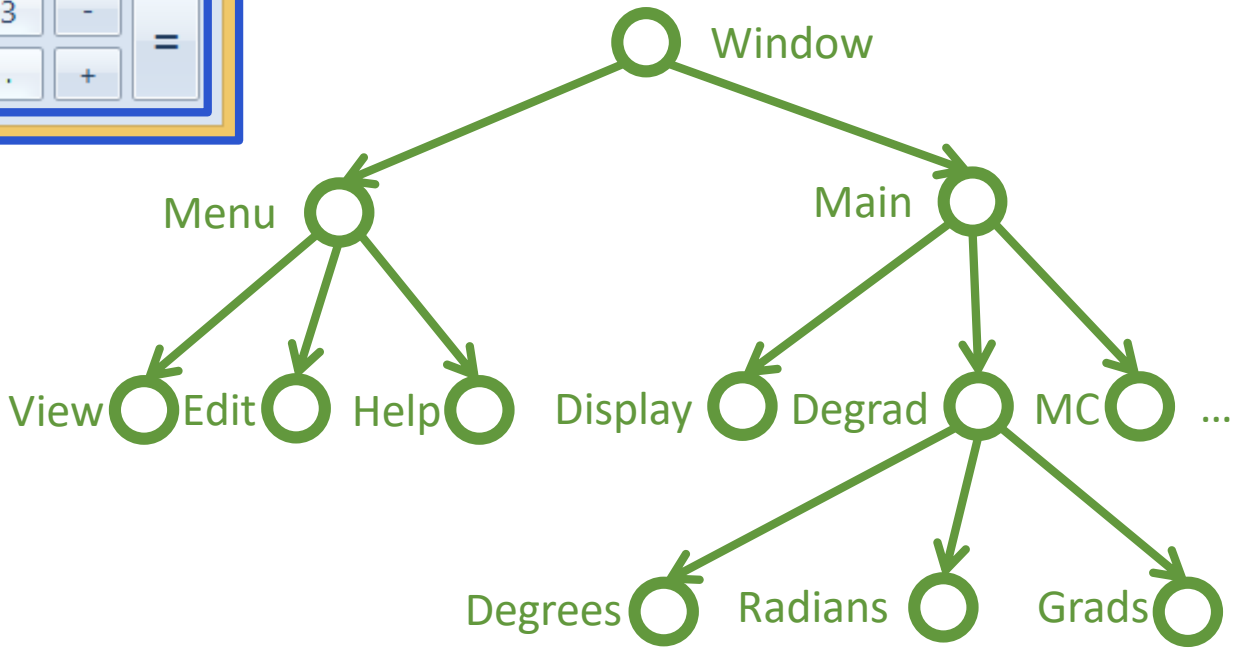
Typed Graph


Deák Ferenc square
Astoria
Blaha Lujza square
Square of the Franciscans
Kálvin square
Rákóczi square
Corvin estate
Boráros square
V. district
VIII. district
IX. district
Budapest

Graphical User Interface



- Representation of a tree structure
  - as graph
  - as containment relation



Representation of a Tree Structure

File system

C:

\Documents

\Pictures

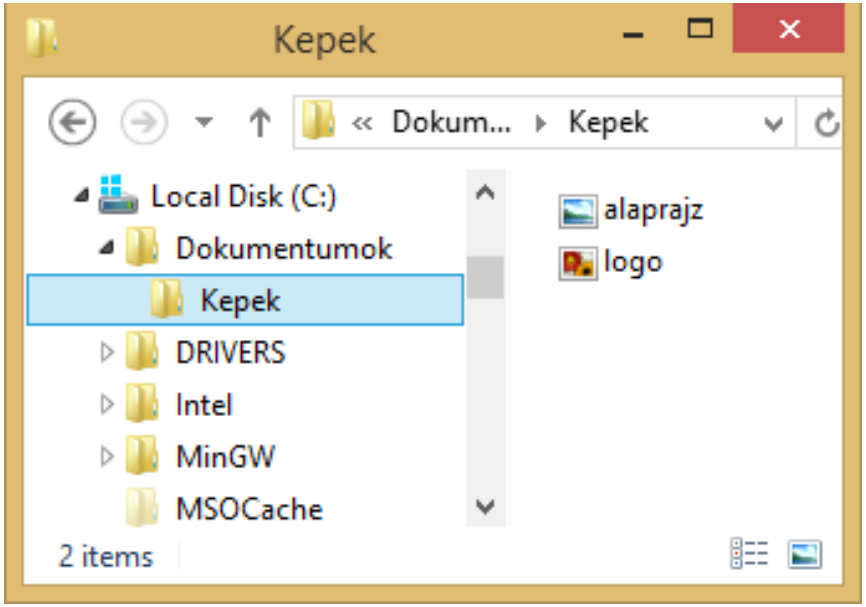
\logo.png

\groundplot.jpg

\contracts.pdf

\Drivers

…



How much storage capacity can be gained by deleting the directory „Documents”?

Characteristics of a File System

- How much storage capacity can be gained by deleting the directory „Documents”?
- Some further Parameter are required

Name	Type	Size (kB)	Last modified
Documents	directory		2016.02.02
Contracts.pdf	file	569	2015.11.09
Pictures	directory		2016.02.02
Logo.png	file	92	2015.03.06
Groundplot.jpg	file	1226	2016.02.02

- Knowing these pieces of information and the file hierarchy the question can be answered.

MODELLING THE PROPERTIES

What properties do have the individual components?

Examples

- Tables
- Databases

Definition: Property

- Properties of the elements of a model can also be described.

A **property** is a **function** that is
- defined **over the elements of a model** and
- **partial** (values may be undefined), and
- the **value set** of which is the set of the possible values of the given property.

Tabular Representation

- **Rows of the table** = Model elements
- **Columns of the table** = Properties

Name	Type	Size (kB)	Last modified
Documents	directory		2016.02.02
Contracts.pdf	file	569	2015.11.09
Pictures	directory		2016.02.02
Logo.png	file	92	2015.03.06
Groundplot.jpg	file	1226	2016.02.02

NULL / NA Attributes

Definition: Filtering

- During **filtering** a **filtering condition** is **evaluated** over the elements of the model, and
- the **submodel** is kept that consists exactly of the elements **satisfying the condition**.

- `Type = „file"`

Name	Type	Size (kB)	Last modified
Contracts.pdf	file	569	2015.11.09
Logo.png	file	92	2015.03.06
Groundplot.jpg	file	1226	2016.02.02

- `Size > 1000 kB`

Name	Type	Size (kB)	Last modified
Groundplot.jpg	file	1226	2016.02.02

Definition: Projection

During **projection** some **parameters** of the model are chosen, others are omitted.

Name	Type	Size (kB)	Last modified
Documents	directory		2016.02.02
Contracts.pdf	file	569	2015.11.09
Pictures	directory		2016.02.02
Logo.png	file	92	2015.03.06
Groundplot.jpg	file	1226	2016.02.02

Projection:

{Name, Size}

Name	Size (kB)
Documents	
Contracts.pdf	569
Pictures	
Logo.png	92
Groundplot.jpg	1226

SUMMARY

Definition: Structural Model

Structural models are static. Their basis are (sub-)systems that are divided into their ingredients by the relation „**`Part of`**”.

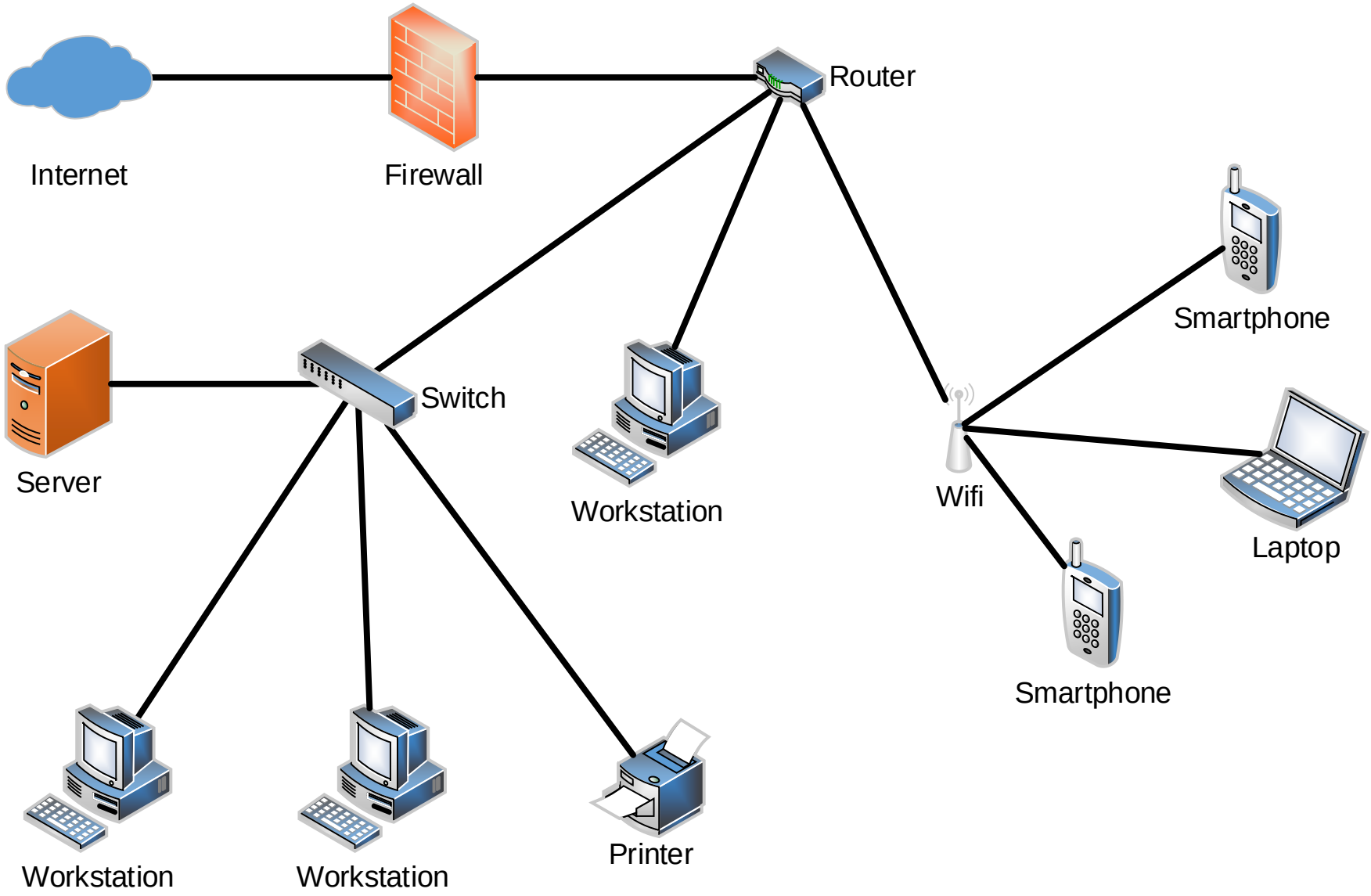
Die **ingredients** can be the followings:

- further divided **subsystems** or
- further not divided (*elementary*) **components**.

The structural model represents the structure of the system according

- its ingredients,
- the **properties** of the ingredients and
- their **relationships** among each other.

Example: Corporate Network



Definition: Decomposition

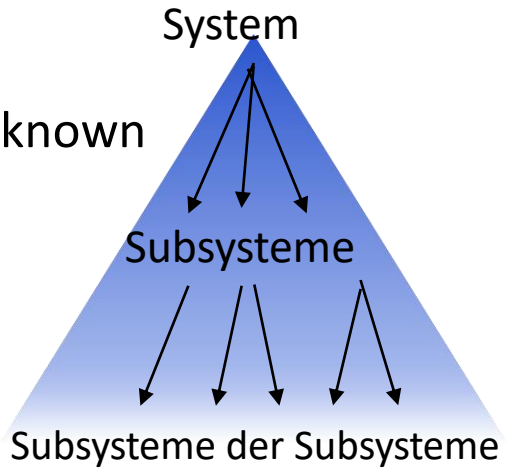
„**Structural models** are static. Their basis are (sub-)systems that are divided into their ingredients by the relation „**`Part of`**”.

Decomposition („faktoring”) is the division of a complex problem or system into smaller parts that are easier to conceive, understand, design, implement/program and maintain.

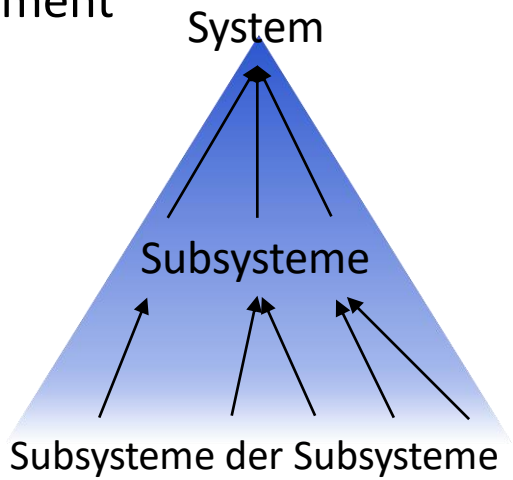
- Hierarchical Decomposition:
  - Multi-level whole-part relationship
- Meaning of the relation `Part of`:
  - Physical Decomposition: `Part of` = spatial division
  - Logical Decomposition: `Part of` = functional division

Top-down and Bottom-up

- Top-down
  - ☺ Roles of the subsystems are during development already known
  - ☹ There are no functioning parts during development
  - ☹ Problems/Reqs of the subsystems are revealed late

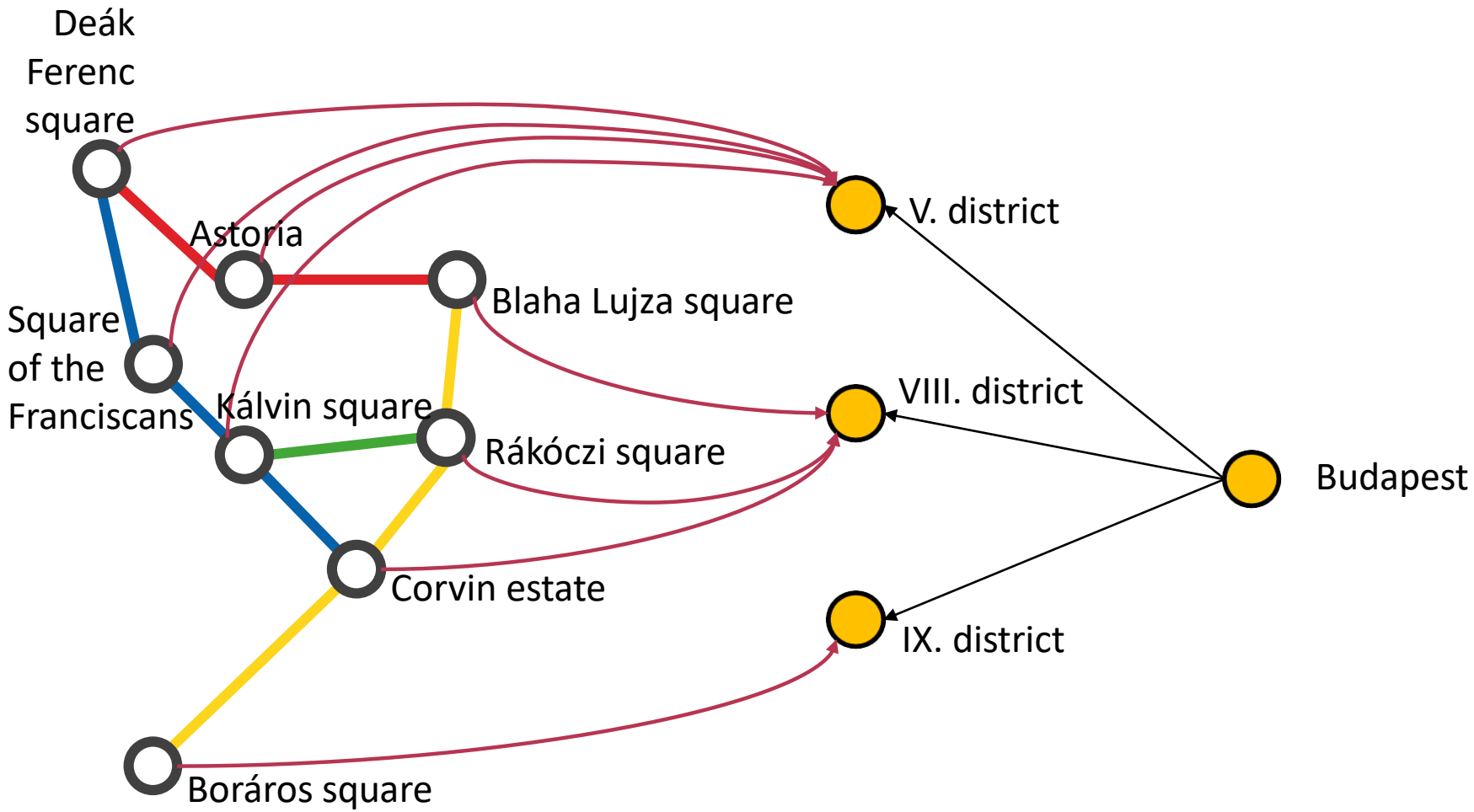


- Bottom-up
  - ☺ Subsystems can be tested step-by-step
  - ☺ There are always some functioning parts during development
  - ☹ Exact roles of the subsystems are revealed late



- Not only in structural modelling
- Mixed approaches, iterative Development

Typed Graph


Deák Ferenc square
Astoria
Blaha Lujza square
Square of the Franciscans
Kálvin square
Rákóczi square
Corvin estate
Boráros square
V. district
VIII. district
IX. district
Budapest

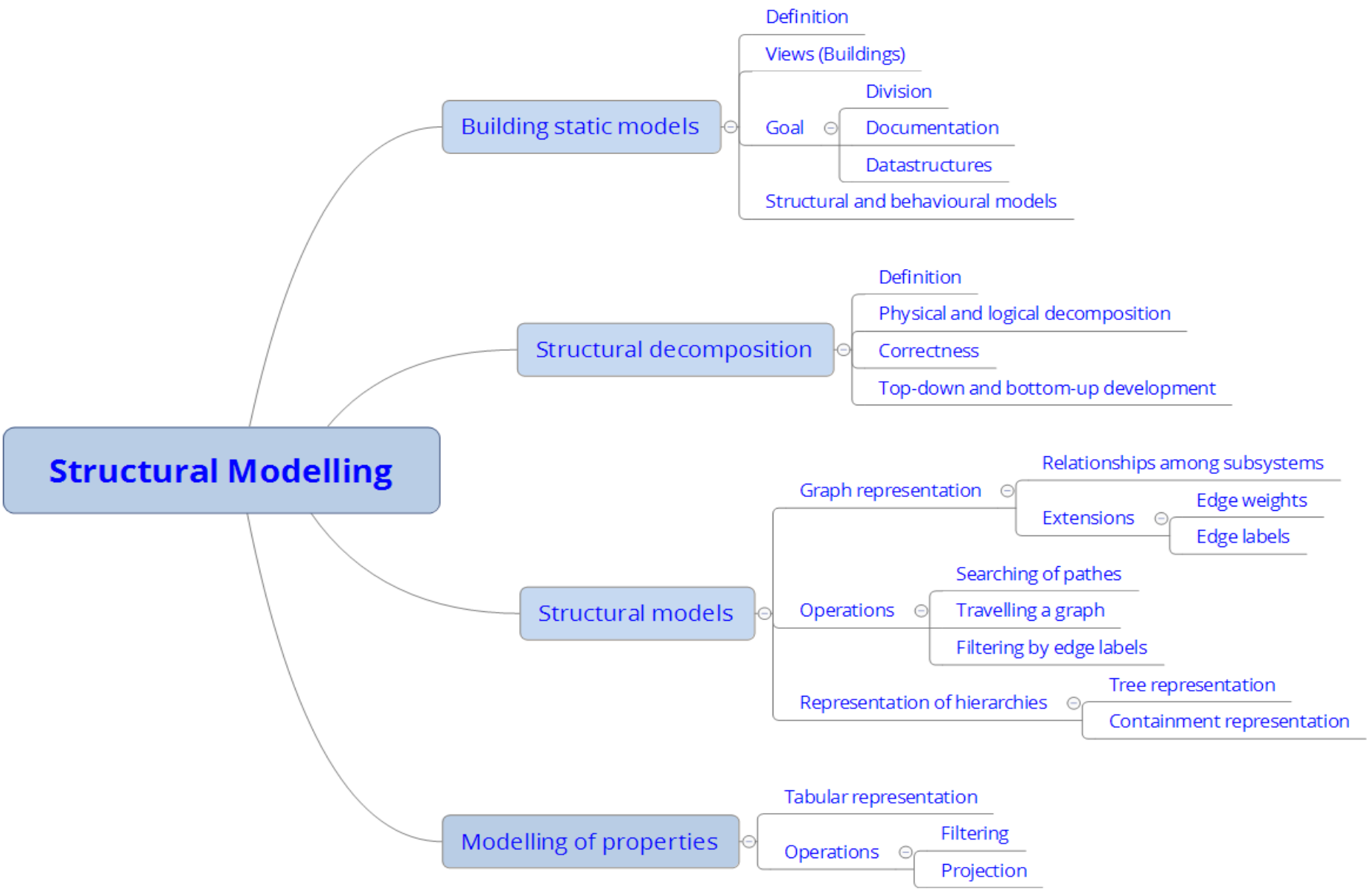
Tabular Representation

- **Rows of the table** = Model elements
- **Columns of the table** = Properties

Name	Type	Size (kB)	Last modified
Documents	directory		2016.02.02
Contracts.pdf	file	569	2015.11.09
Pictures	directory		2016.02.02
Logo.png	file	92	2015.03.06
Groundplot.jpg	file	1226	2016.02.02

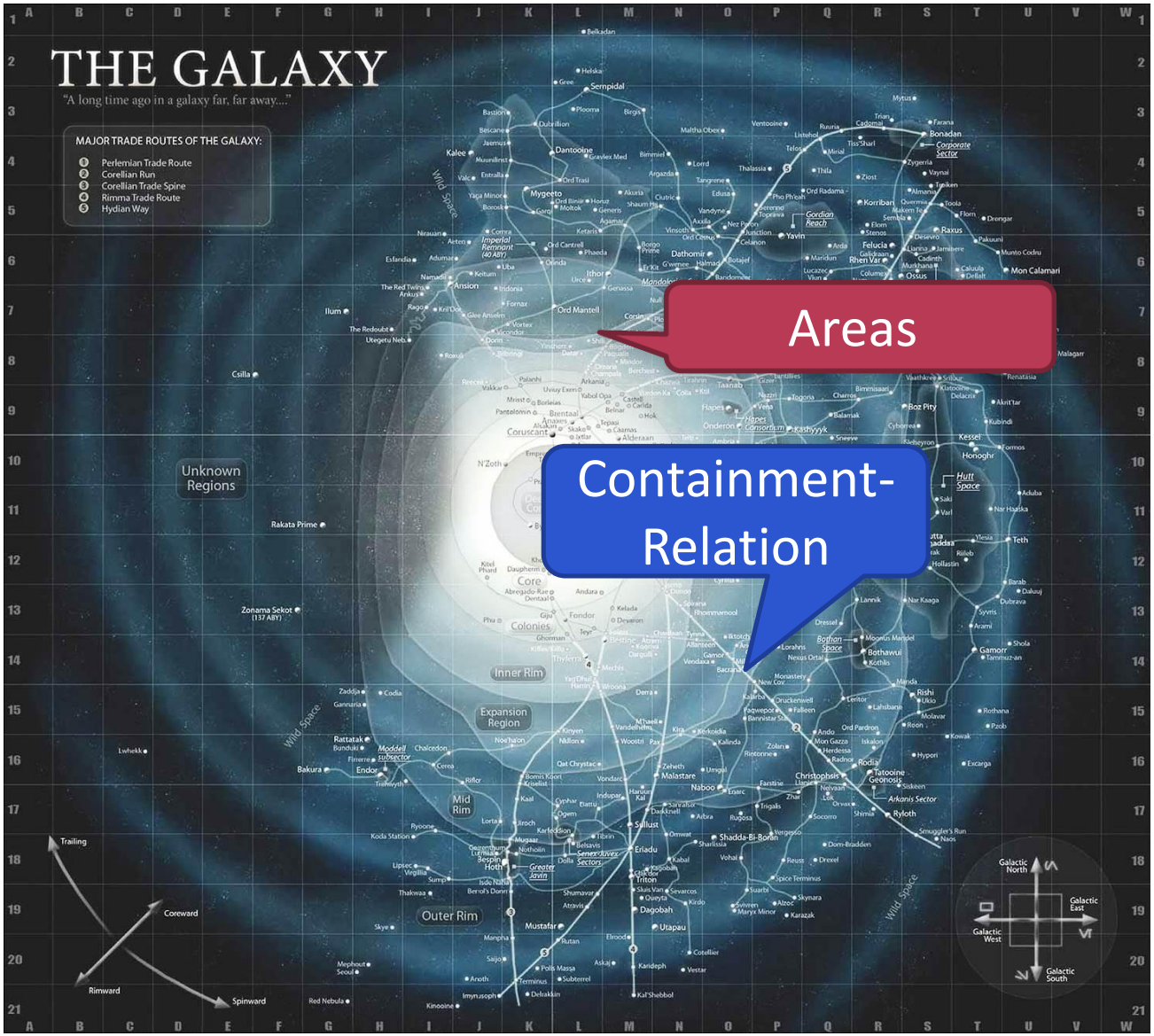
NULL / NA Attributes

Summary

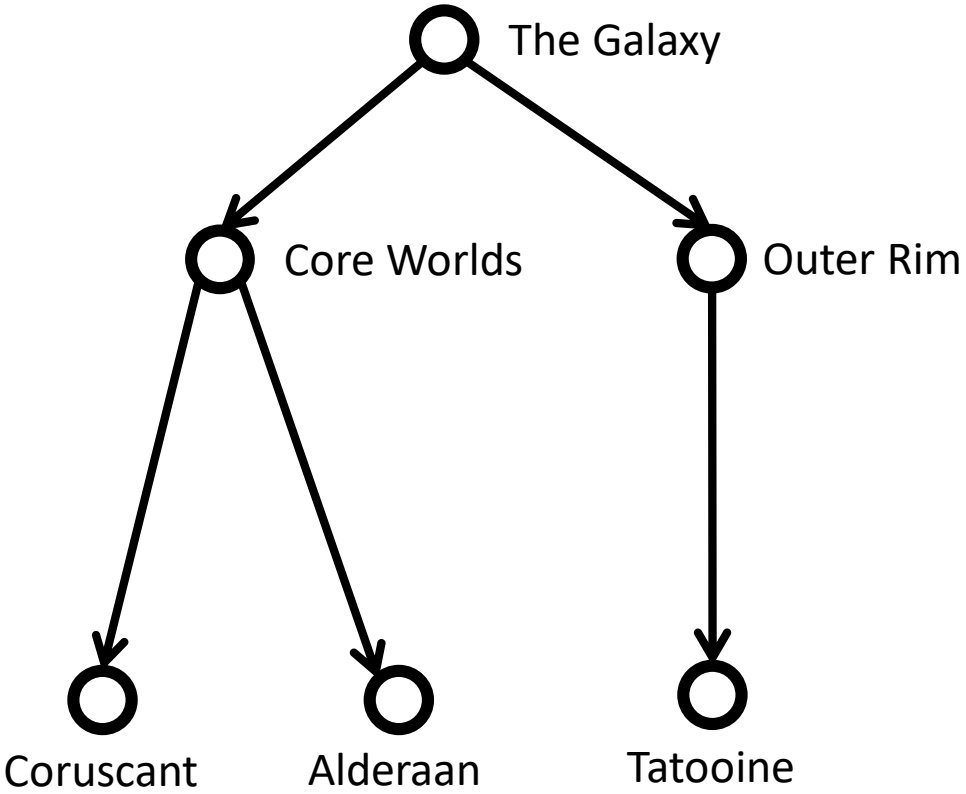

Structural Modelling
Building static models
Definition
Views (Buildings)
Goal
Division
Documentation
Datastructures
Structural and behavioural models
Structural decomposition
Definition
Physical and logical decomposition
Correctness
Top-down and bottom-up development
Structural models
Graph representation
Relationships among subsystems
Extensions
Edge weights
Edge labels
Operations
Searching of pathes
Travelling a graph
Filtering by edge labels
Representation of hierarchies
Tree representation
Containment representation
Modelling of properties
Tabular representation
Operations
Filtering
Projection

MODELLING TYPES

Example: The Galaxis



Example: The Galaxis



Categorisation of Model Elements in Types

- Example: Can the „Colour of Lightsaber” of Yoda change?
  - No
    Yoda – an entity with green lightsaber
    Important/relevant attribute?
  - Yes
    „Colour of Lightsaber” no characteristic attribute

- Type: Entirety of elements that are distinguished by similar attributes

Type and Attributes

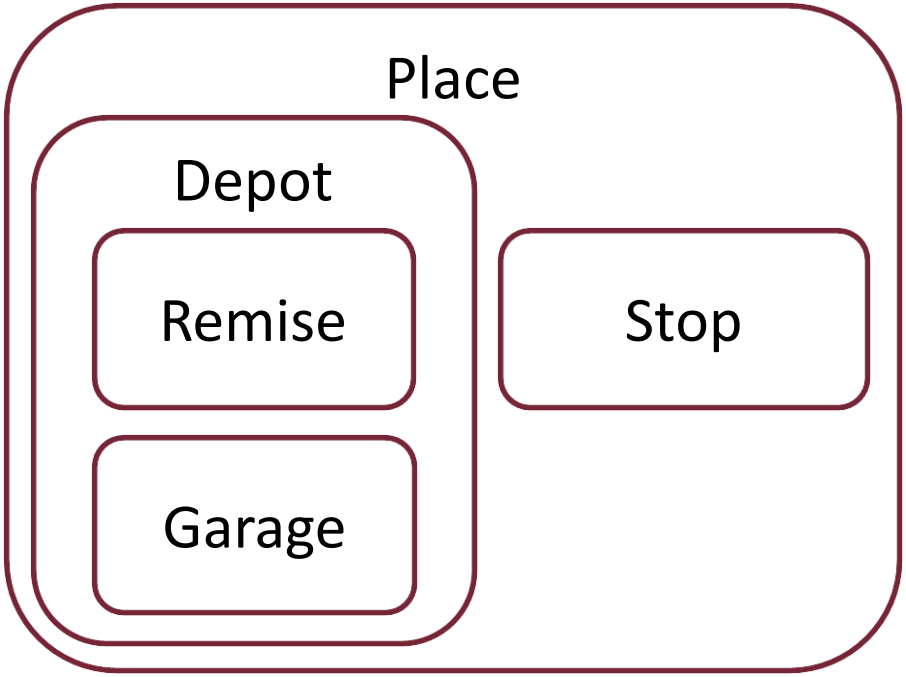
- Type is a part of the knowledge, implicitly expressed in the label
  - *Type* is a special property
  - Other properties: *attributes*

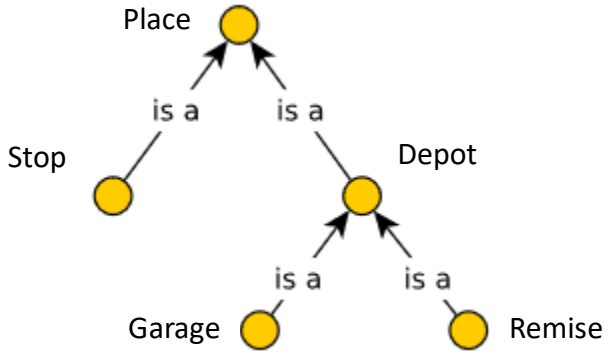
- Widely used Convention:
  - Attributes may sometimes change (but they do not have to)
  - Type is permanent (forever)

Type Graph

- A *type node* for each node type
- A *type edge* for each edge type
- …

Metamodell

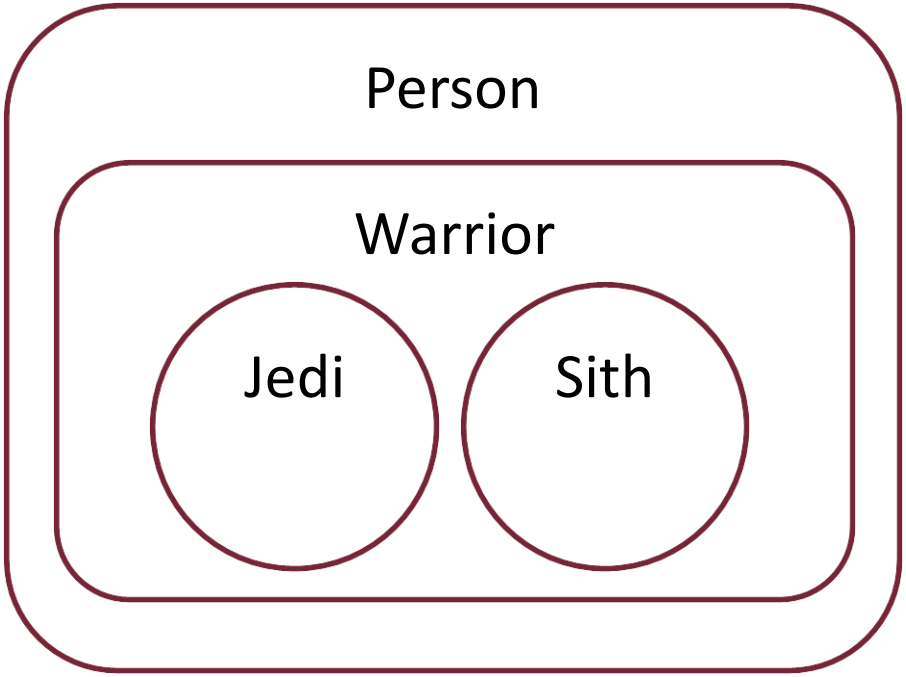


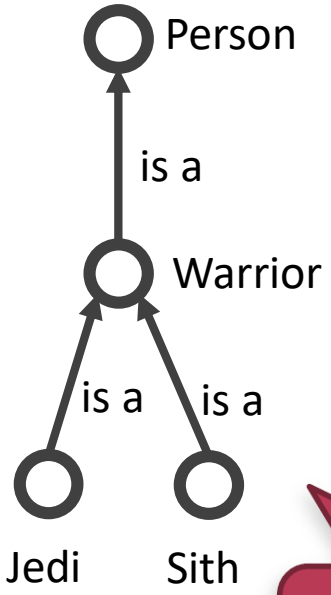


Type Graph

- A *type node* for each node type
- A *type edge* for each edge type
- …

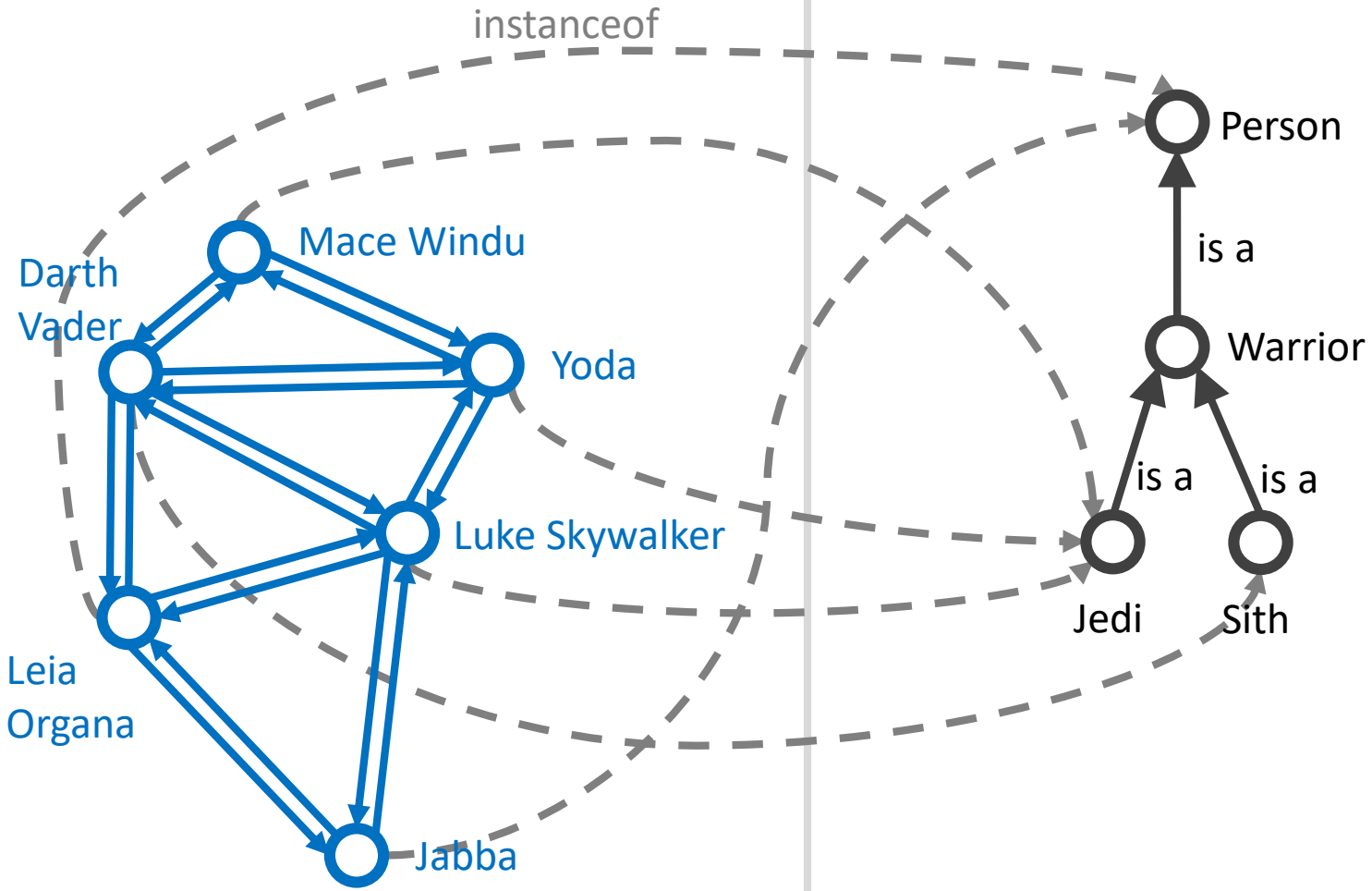
Metamodell



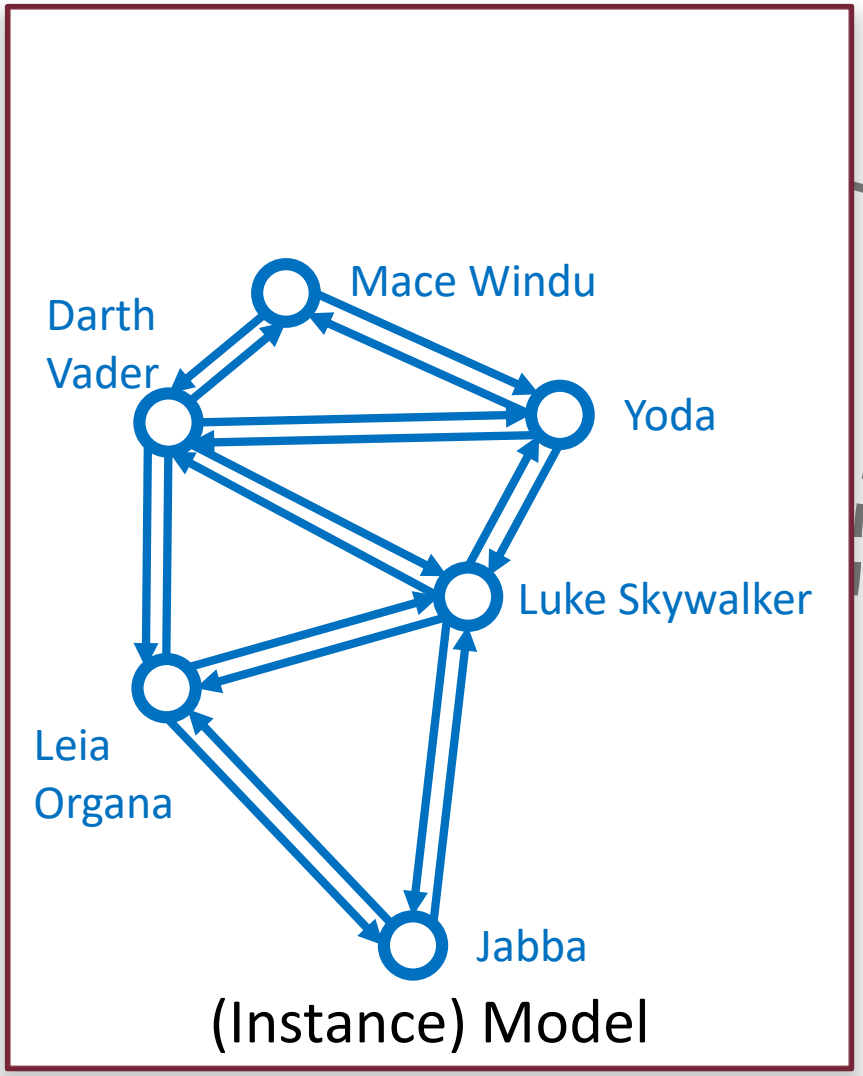


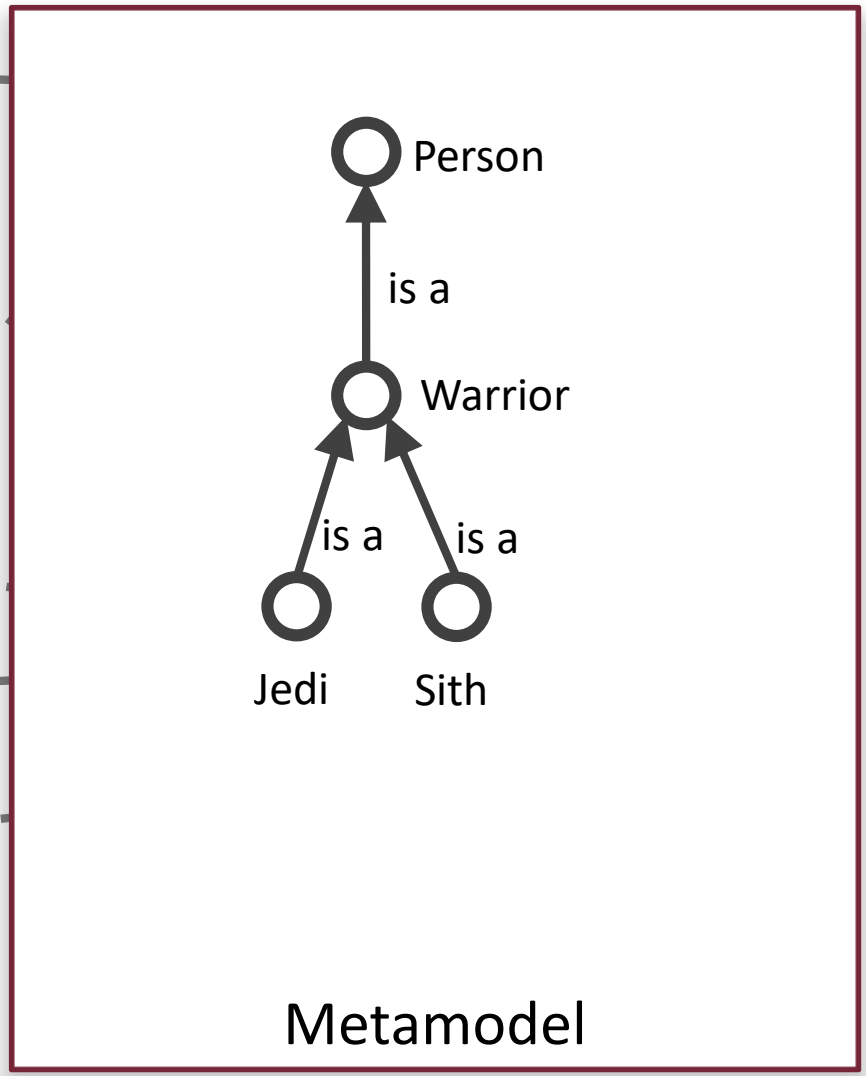
Hierarchic Decomposition

Representation of Type-Instance Relations

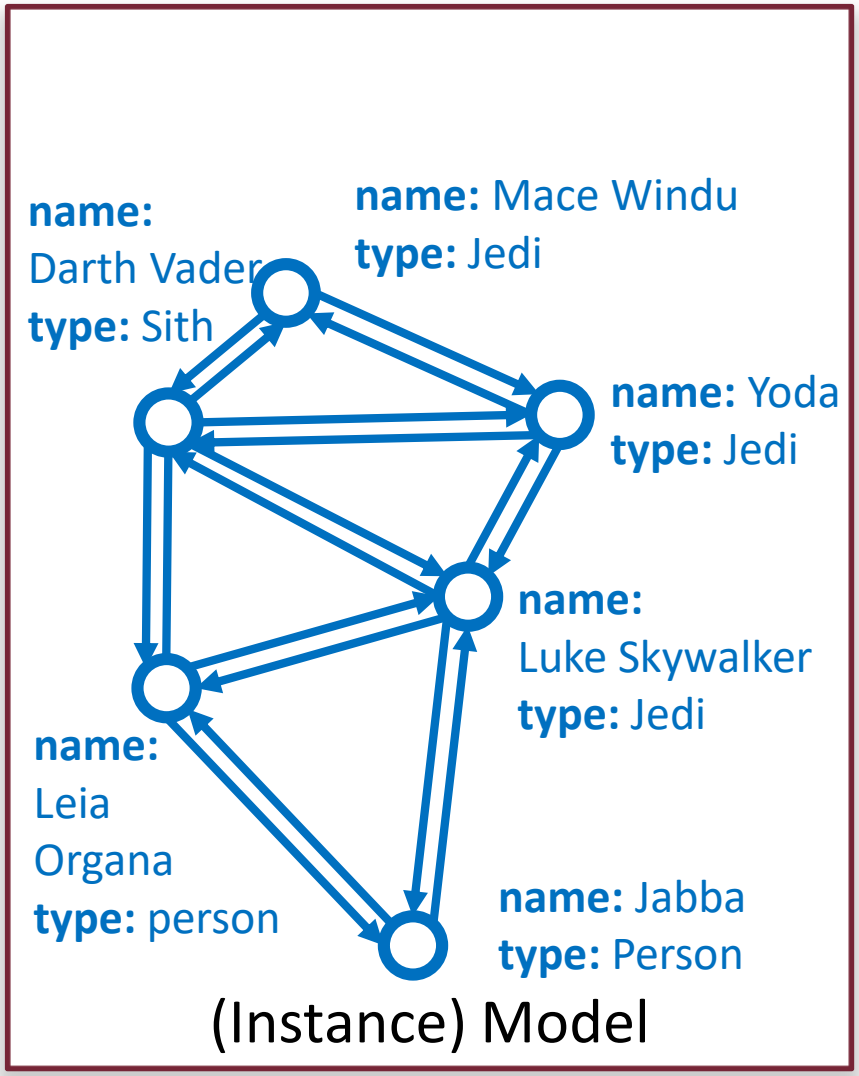


Representation of Type-Instance Relations



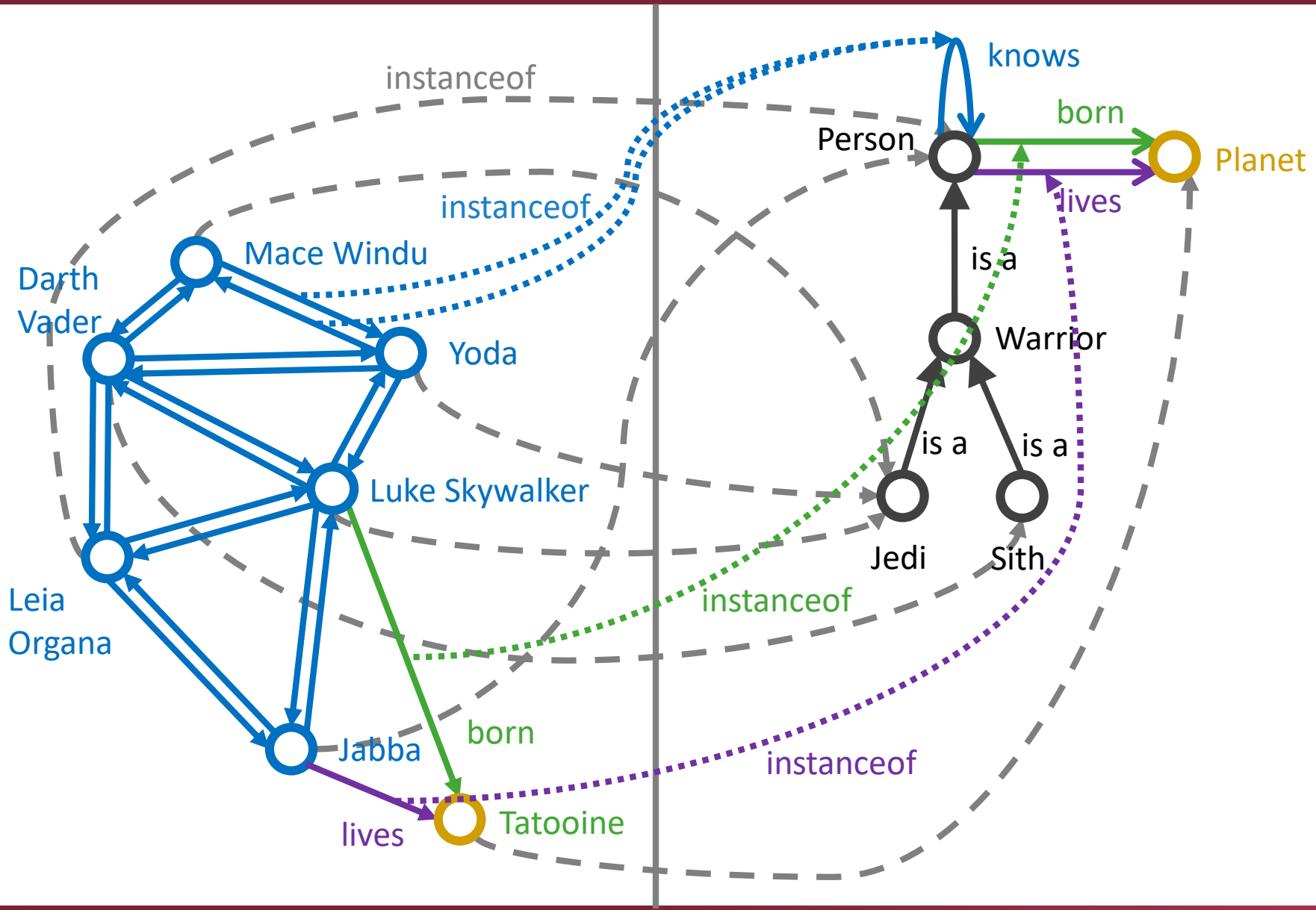


Representation of Types as Attributes



(Instance) Model

Edges with Different Edge Types



REPRESENTING STRUCTURAL MODELS IN PROGRAMS

Programming Paradigms

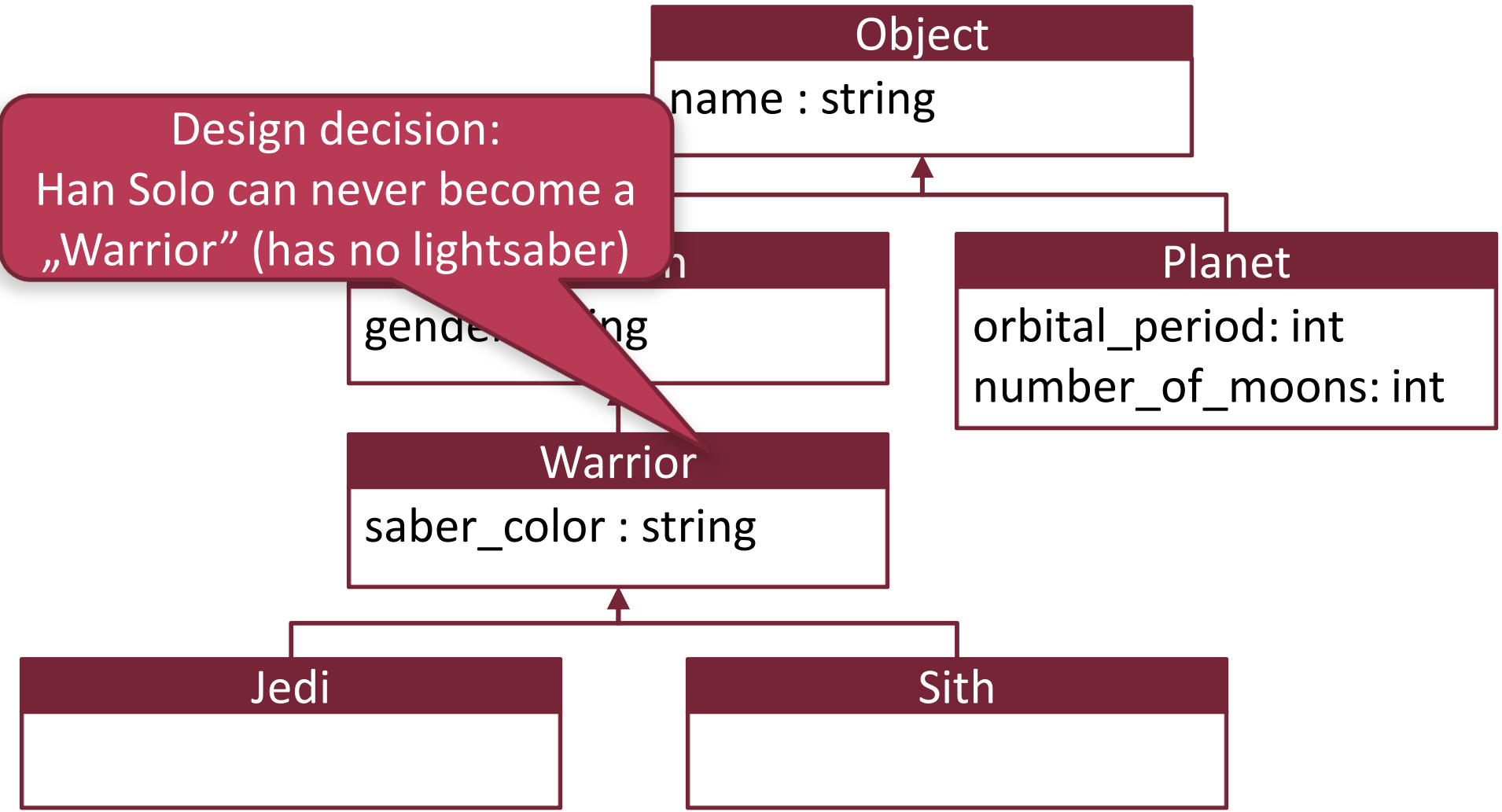
- **Programming paradigm:**
  theoretical model of a programming language

- **Structured programming**
  (C, Pascal, Modula)
  - Organising variables into structures: struct/record

Programming Paradigms

- **Objekt-oriented programming**, OOP (C++, Java)
  - type: *Class*
  - instance: *Object*
  - attribute: *Attribute field*
  - operation: *Methode*
  - Visibility of attributes/methodes, data encapsulation, etc.: see *Basics of Programming 2*
  - Pointer fields vs. nesting
    - Terms: reference, composition, aggregation

OOP: Inheritance



Representing Models

- Textual representation
  - XML, JSON, …
- Graphical representation
  - UML, AADL, SysML, EMF, …

Textual Representation

- XML (Extensible Markup Language)
  - standardised, universal technique for the definition of description languages
  - (human?) readable
- JSON (JavaScript Object Notation)
  - standardised notation for readable data exchange

XML-Example: Weather Web Service

```
<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-Example: 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"
}
```

Graphical Representation

- UML (Unified Modeling Language)
  - a universal modelling language
  - in SW development widely used
- AADL (Architecture Analysis & Design Language)
  - for describing of architectures
- SysML (Systems Modeling Language)
  - UML based general modellings language for system planing and modelling
- EMF (Eclipse Modeling Framework, Ecore)
  - for describing modelling languages

FURTHER ILLUSTRATIVE EXAMPLES

Illustration – Structural Models

Nested representation

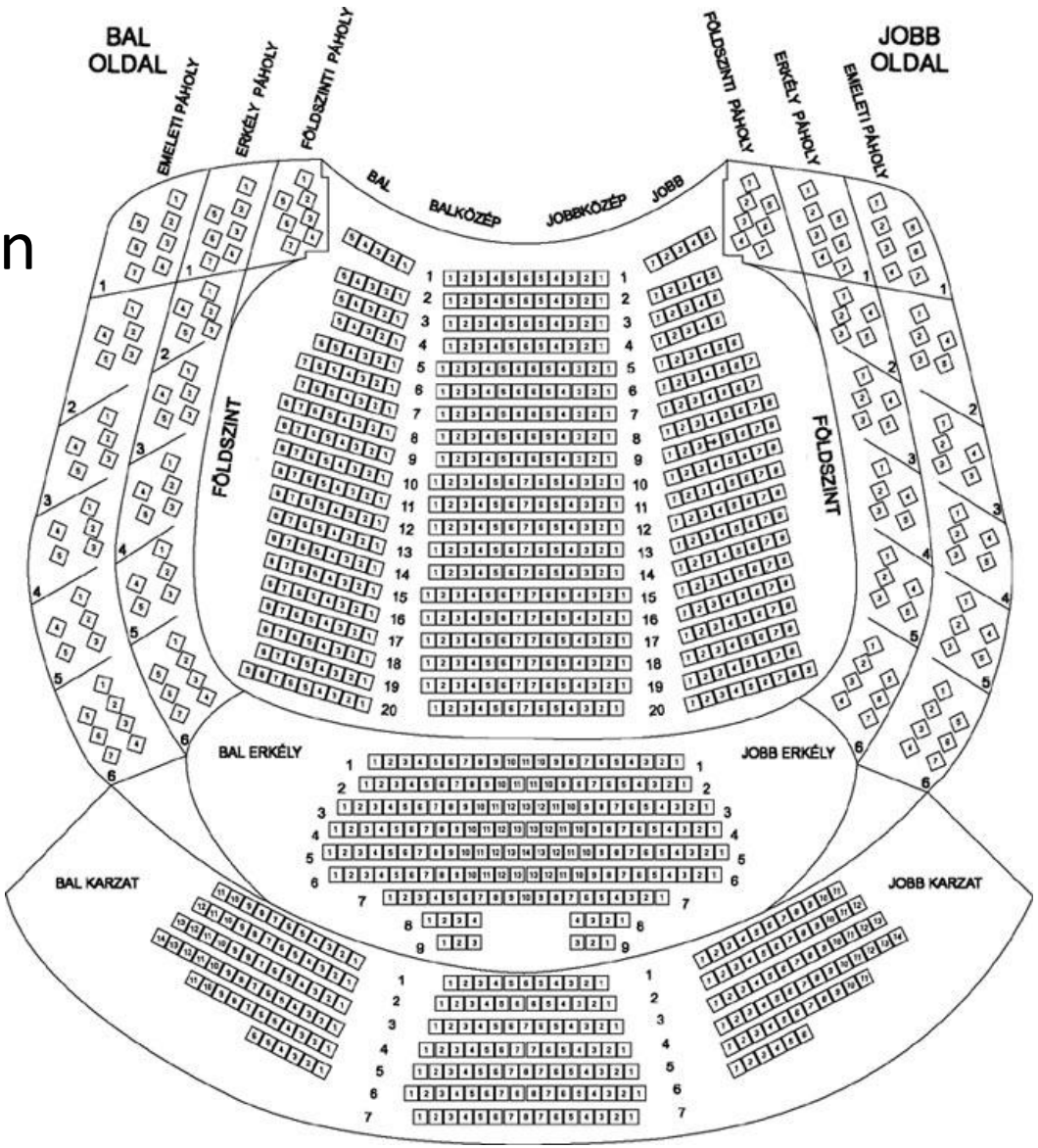


Illustration – Structural Models

Frame structure