

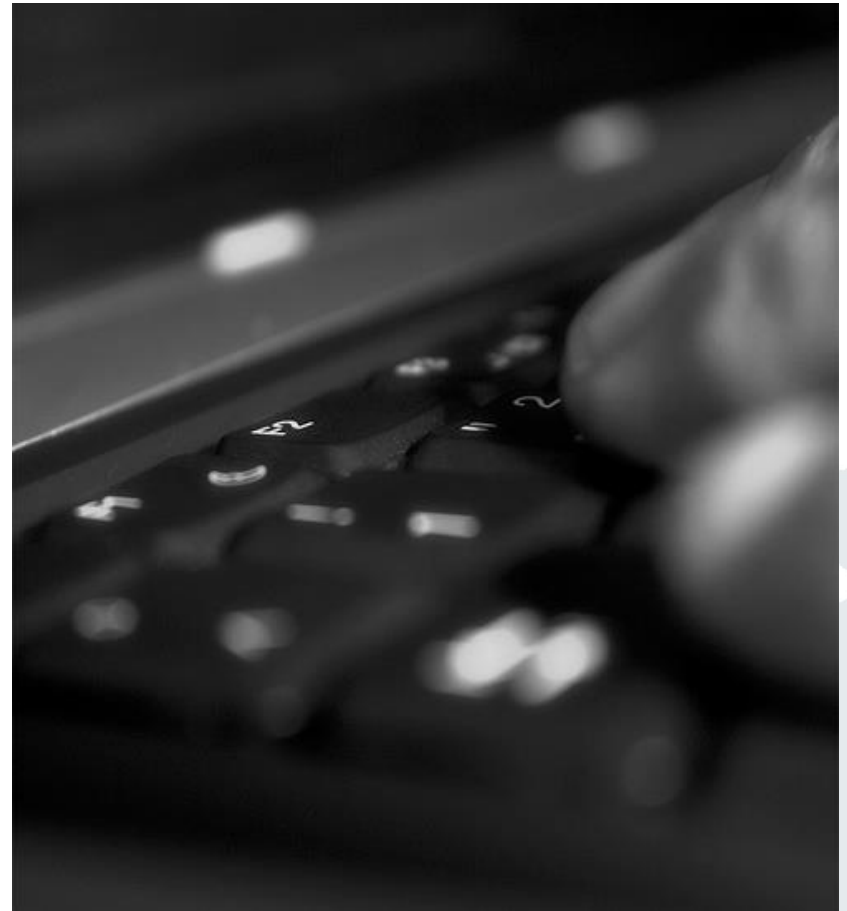
## Lecture 14

### Business Informatics 2 (PWIN)

FAQ

SS 2015

Stephan Heim, Christopher Schmitz  
[www.m-chair.de](http://www.m-chair.de)



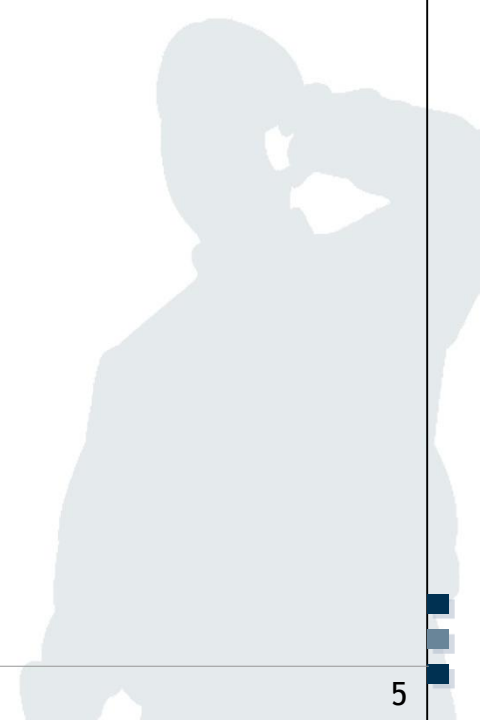
Jenser (Flickr.com)

Ich würde gerne wissen, was sich denn geändert hat auf den mit "updated" gekennzeichneten Folien.

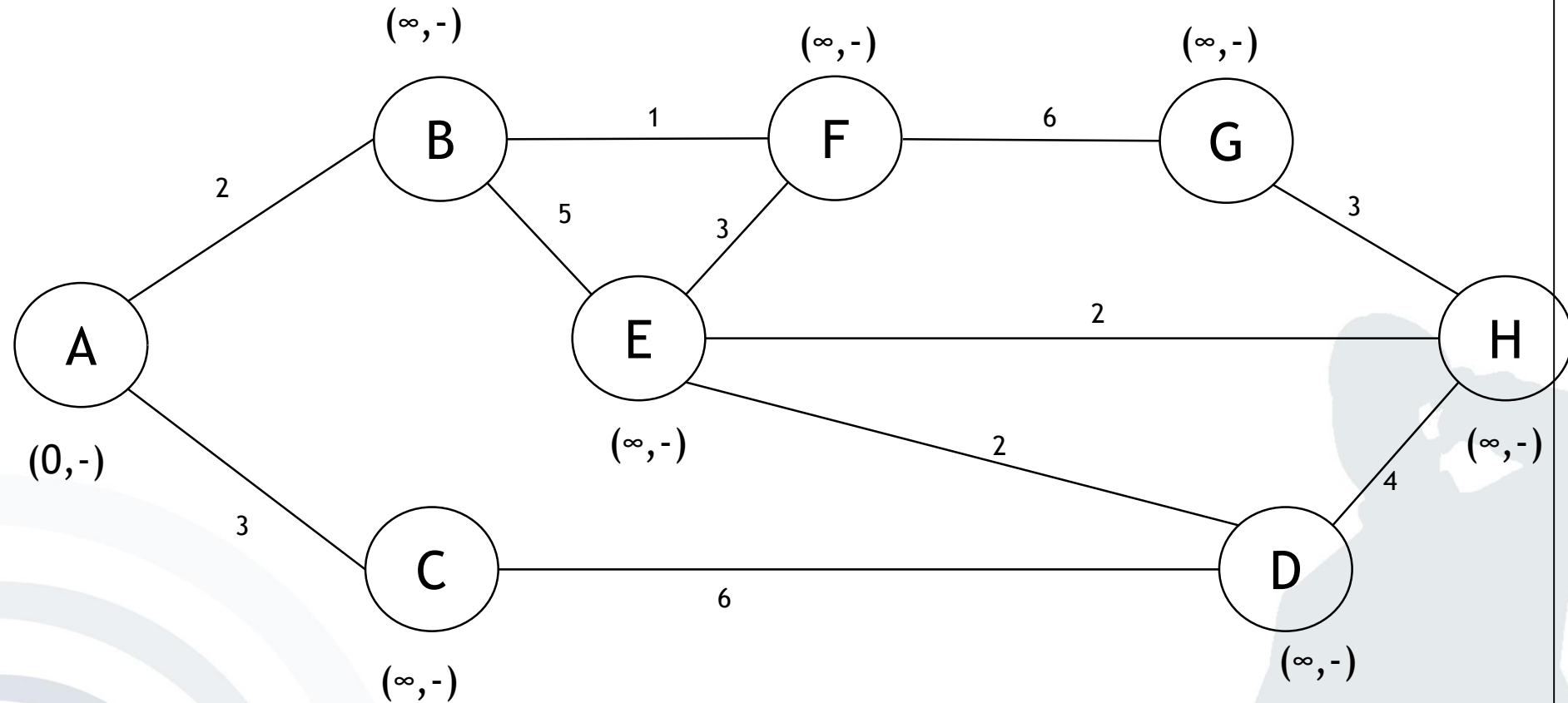
- **Slide Set 7:**
  - Minor changes
- **Slide Set 8:**
  - Minor changes
- **Slide Set 9:**
  - p.27: Arrows for <<include>> and <<extend>> relationships corrected
  - Names of abstract classes formatted to italic.
- **Slide Set 10:**
  - Minor changes
- **Slide Set 11**
  - p.37: Primary keys corrected
- **Slide Set 12**
  - p.6: Description of data types changed
  - p.27,28: Tables corrected
- **Exercise6**
  - 2a): ER model “Employee - Company“ changed

Wird die Klausur auf Deutsch oder Englisch gestellt?  
Und in welcher Sprache darf ich antworten?

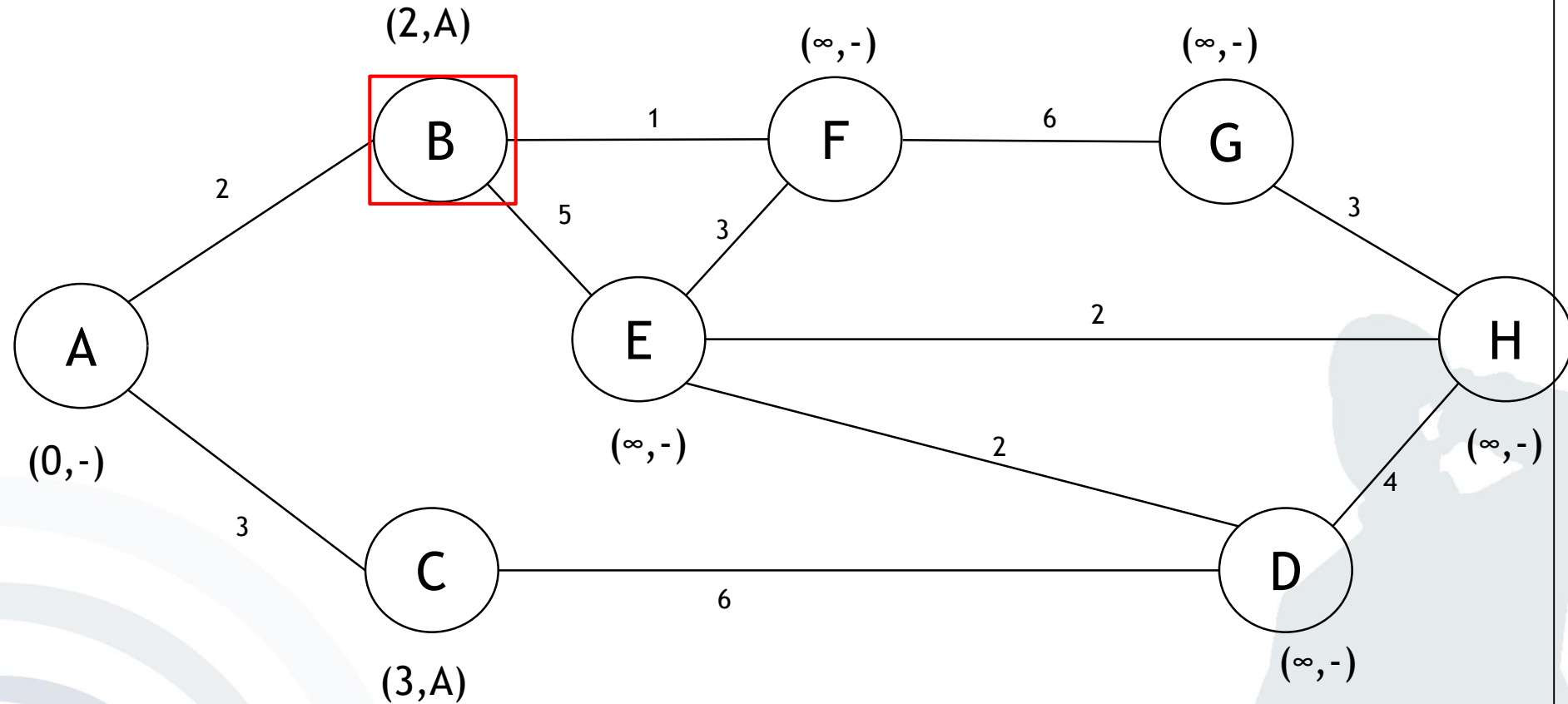
- Die Aufgaben werden auf Deutsch und Englisch gestellt.
- Sie dürfen auf Deutsch und Englisch antworten.



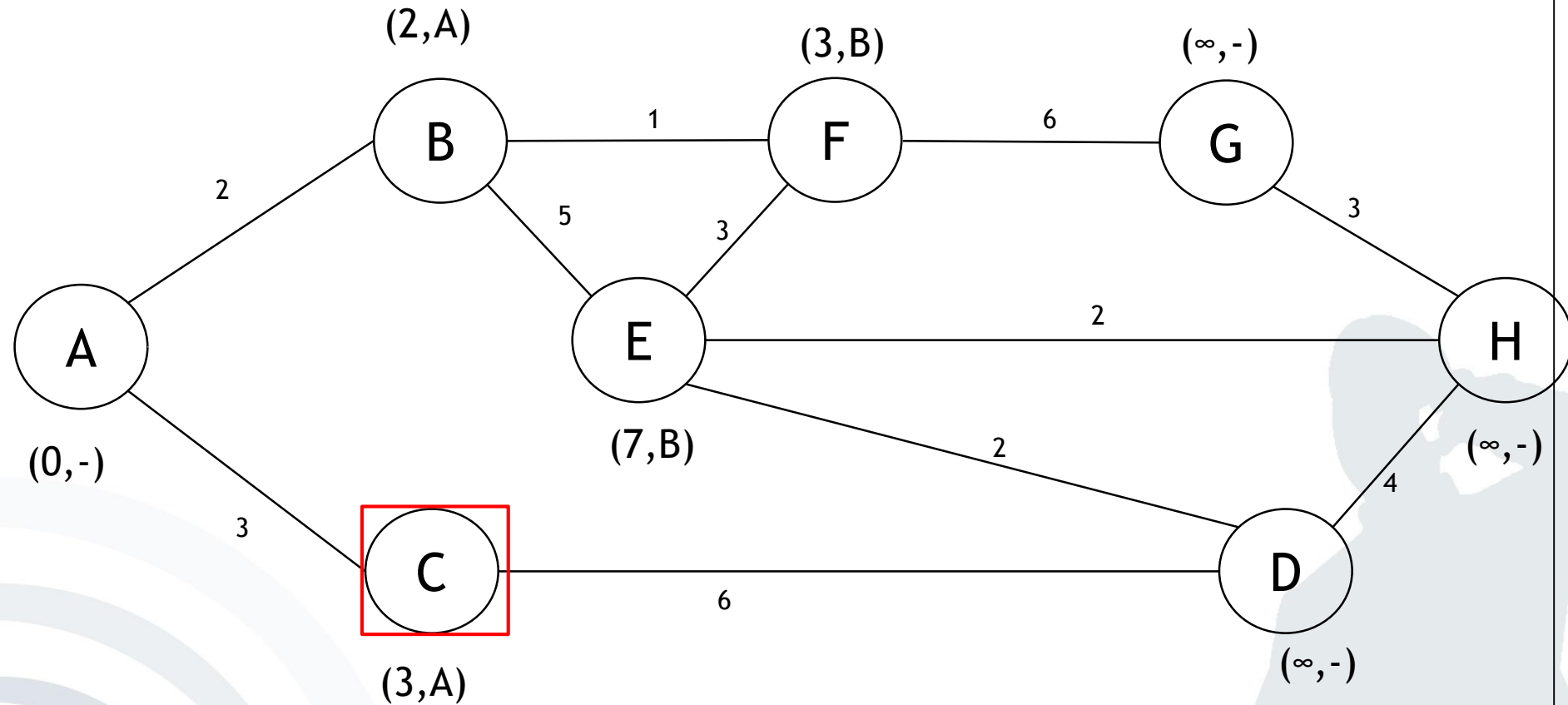
# Dijkstra Algorithm (1)



# Dijkstra Algorithm (1)

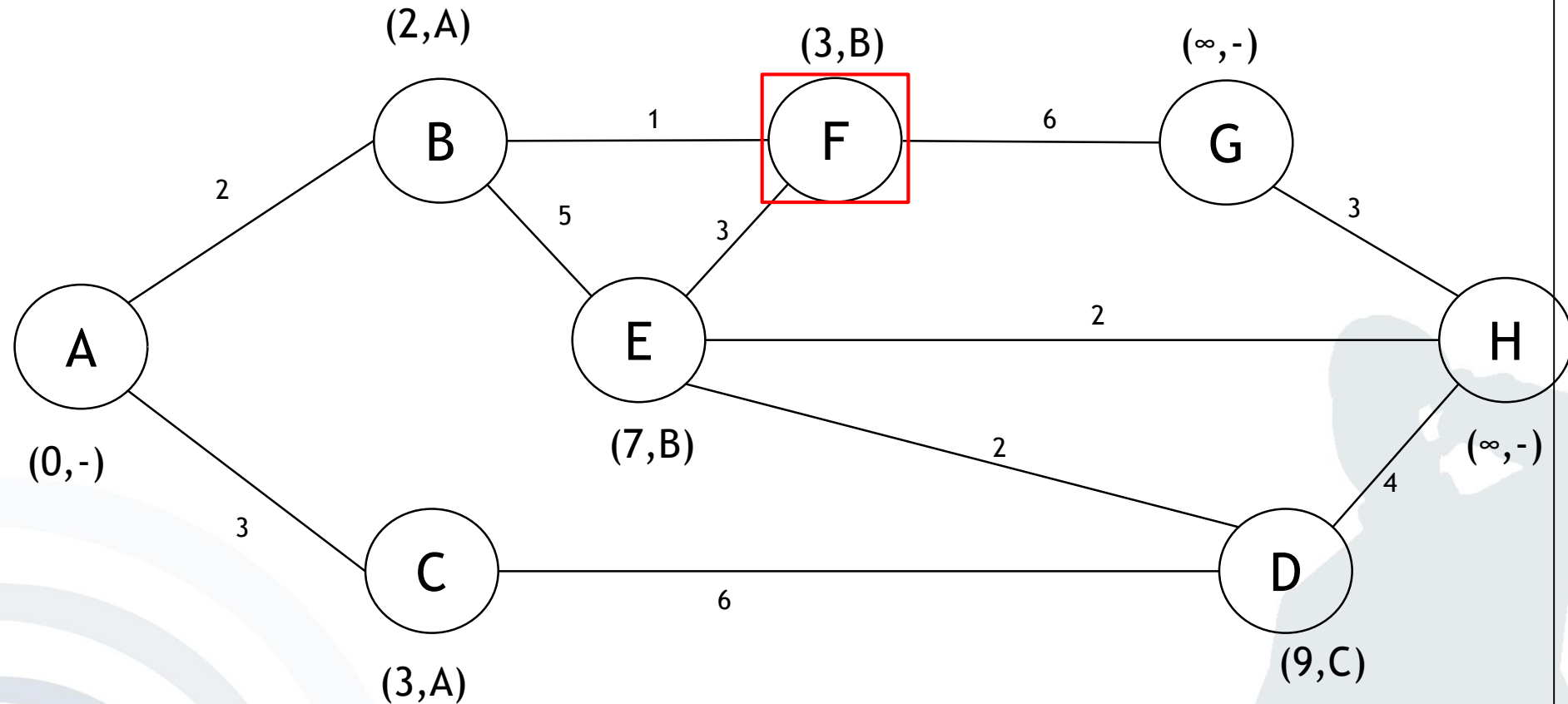


# Dijkstra Algorithm (1)

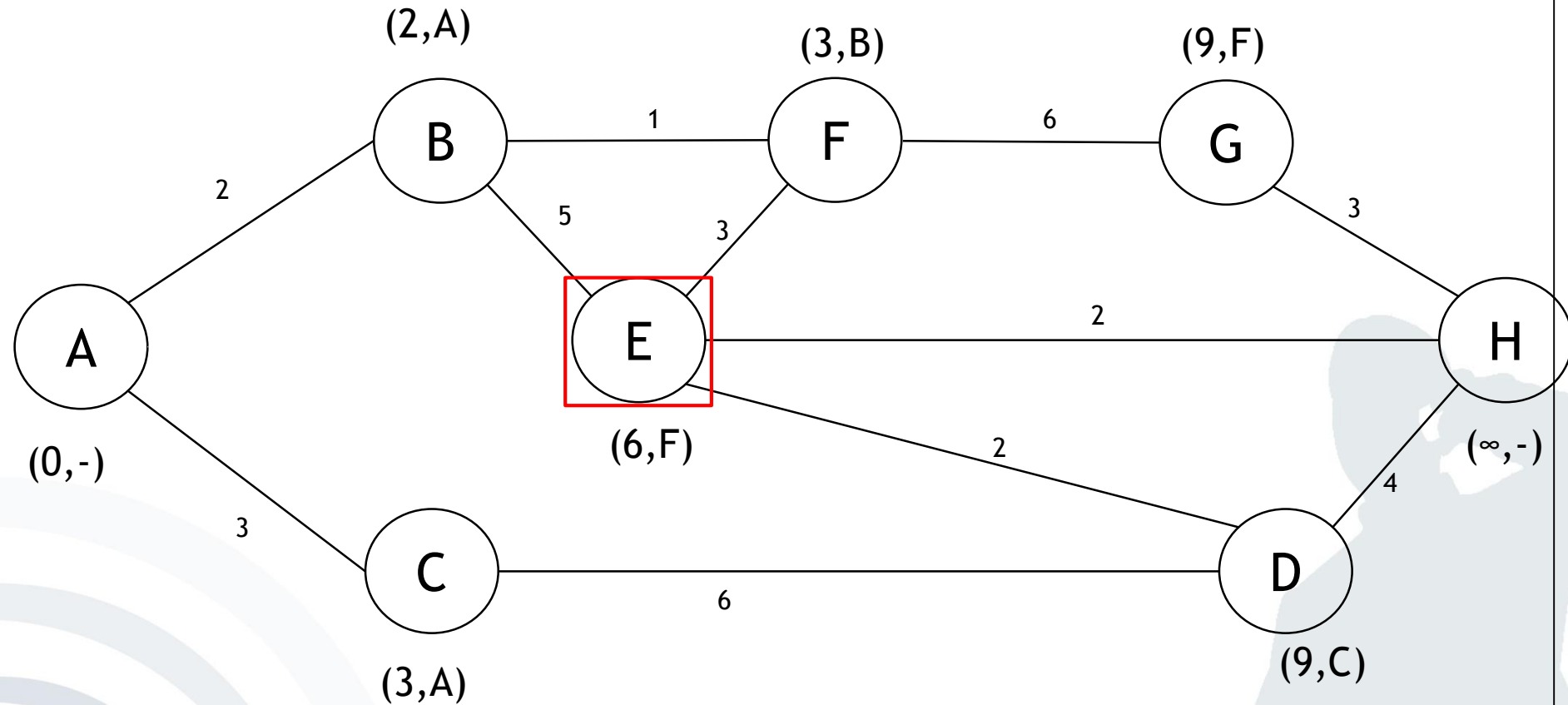




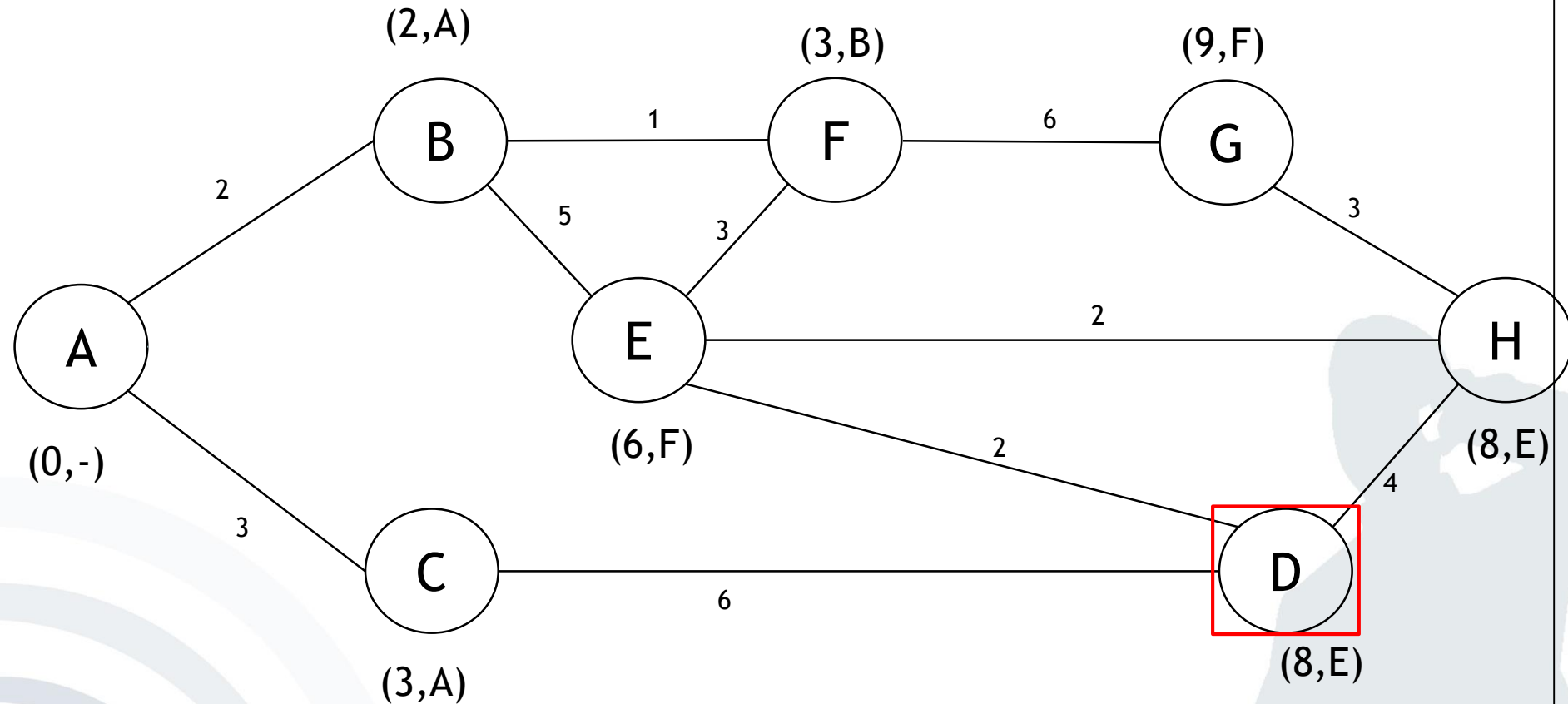
# Dijkstra Algorithm (1)



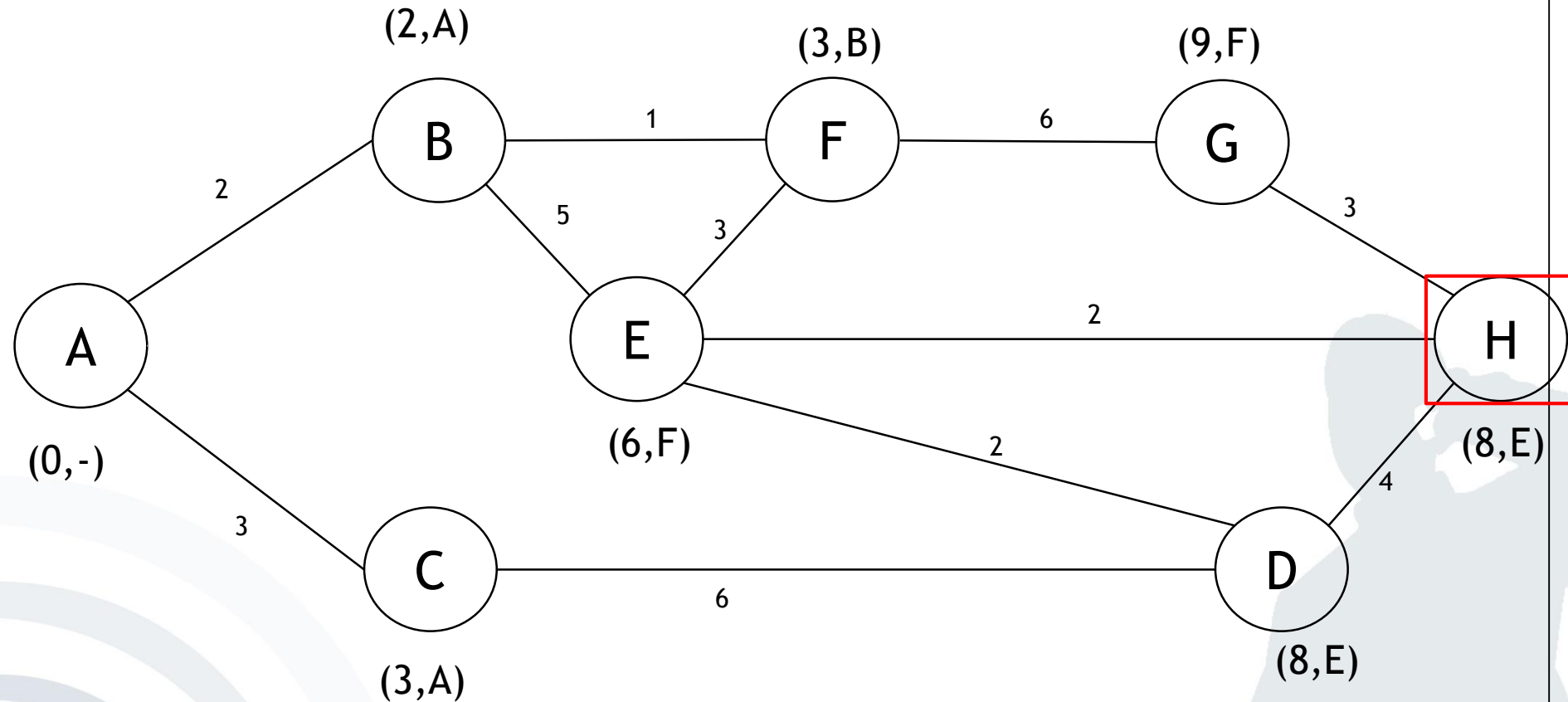
# Dijkstra Algorithm (1)



# Dijkstra Algorithm (1)



# Dijkstra Algorithm (1)



Wie erklärt sich die Musterlösung zu Aufgabe 5 b) der Klausur SoSe 2004?

Schreibe Sie eine XML-DTD zu Ihrem Dokument aus Teil b), so dass in  
b) ein "gültiges" Adress-XML-Dokument entsteht

```
<?xml version="1.0"?>
<Adresse>
  <Firma>...</Firma>
  <Name>...</Name>
  <Strasse>...</Strasse>
  <PLZ>...</PLZ>
  <Stadt>...</Stadt>
</Adresse>
```

Schreibe Sie eine XML-DTD zu Ihrem Dokument aus Teil b), so dass in  
b) ein "gültiges" Adress-XML-Dokument entsteht

```
<?xml version="1.0"?>
<Adresse>
  <Firma>...</Firma>
  <Name>...</Name>
  <Strasse>...</Strasse>
  <PLZ>...</PLZ>
  <Stadt>...</Stadt>
</Adresse>
```

```
<!DOCTYPE Adresse [
]
]>
```

Schreibe Sie eine XML-DTD zu Ihrem Dokument aus Teil b), so dass in b) ein "gültiges" Adress-XML-Dokument entsteht

```
<?xml version="1.0"?>
<Adresse>
  <Firma>...</Firma>
  <Name>...</Name>
  <Strasse>...</Strasse>
  <PLZ>...</PLZ>
  <Stadt>...</Stadt>
</Adresse>
```

```
<!DOCTYPE Adresse [
  <!ELEMENT Adresse(Firma, Name,
                    Strasse, PLZ, Stadt)>
]>
```



Schreibe Sie eine XML-DTD zu Ihrem Dokument aus Teil b), so dass in  
b) ein "gültiges" Adress-XML-Dokument entsteht

```
<?xml version="1.0"?>
```

```
<Adresse>
```

```
  <Firma>...</Firma>
```

```
  <Name>...</Name>
```

```
  <Strasse>...</Strasse>
```

```
  <PLZ>...</PLZ>
```

```
  <Stadt>...</Stadt>
```

```
</Adresse>
```

```
<!DOCTYPE Adresse [
```

```
  <!ELEMENT Adresse(Firma, Name,  
                    Strasse, PLZ, Stadt)>
```

```
  <!ELEMENT Firma    (#PCDATA)>
```

```
  <!ELEMENT Name     (#PCDATA)>
```

```
  <!ELEMENT Strasse  (#PCDATA)>
```

```
  <!ELEMENT PLZ      (#PCDATA)>
```

```
  <!ELEMENT Stadt    (#PCDATA)>
```

```
]>
```

Ich würde mich freuen, könnten wir während der Wiederholung nochmals auf die UML Diagramme eingehen und dabei die Bedeutung der verschiedenen Pfeilarten, etc. erklärt bekommen. Vielen Dank.

# UML Diagrams Overview

- Use case diagram

Use case diagram

- Class diagram
- Object diagram

Structural diagrams

- Activity diagram
- Sequence diagram
- Collaboration diagram
- State diagram

Behavioural diagrams

- Component diagram
- Deployment diagram

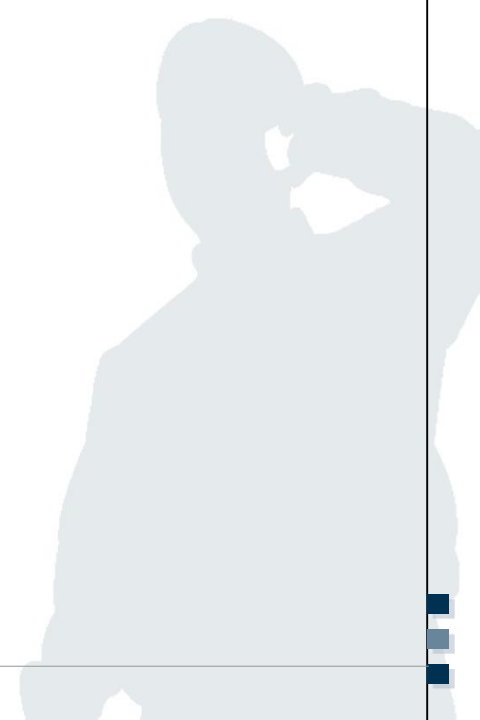
Architectural diagrams

Static elements

Dynamic elements

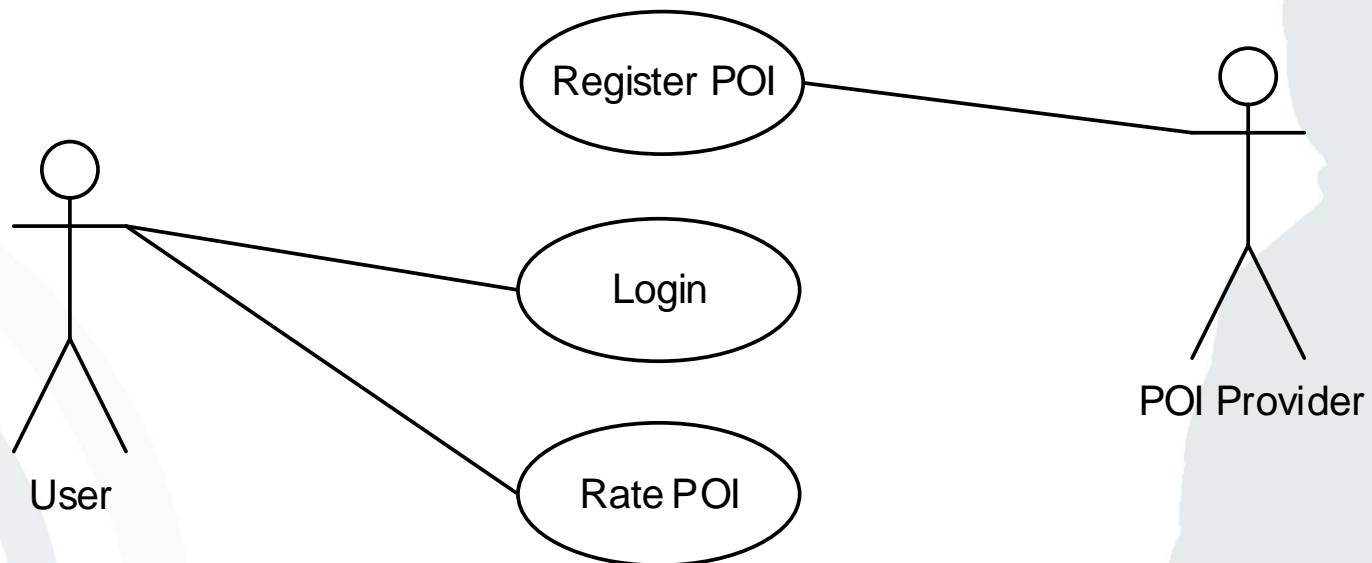
Architectural elements

# UML - Use Case Diagram



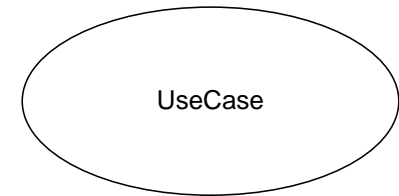
## Use case diagrams

- Use cases describe the functionality, which a system has to provide
- The sum of all “use cases” comprises the technical requirements of a system.
- Use cases define the interfaces between a user and the system
- Specification is developed together with the client/customer



- **Use Case**

- Representation of a sequence of actions that provides value to an actor.

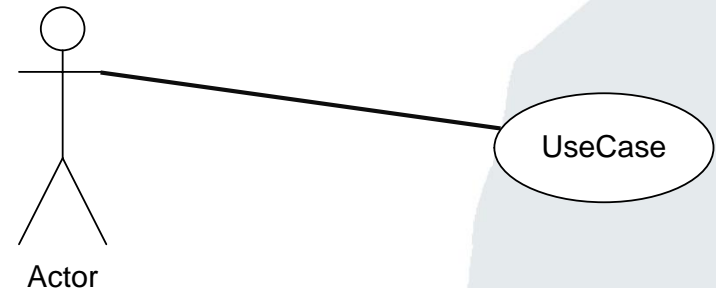


- **User of the system**



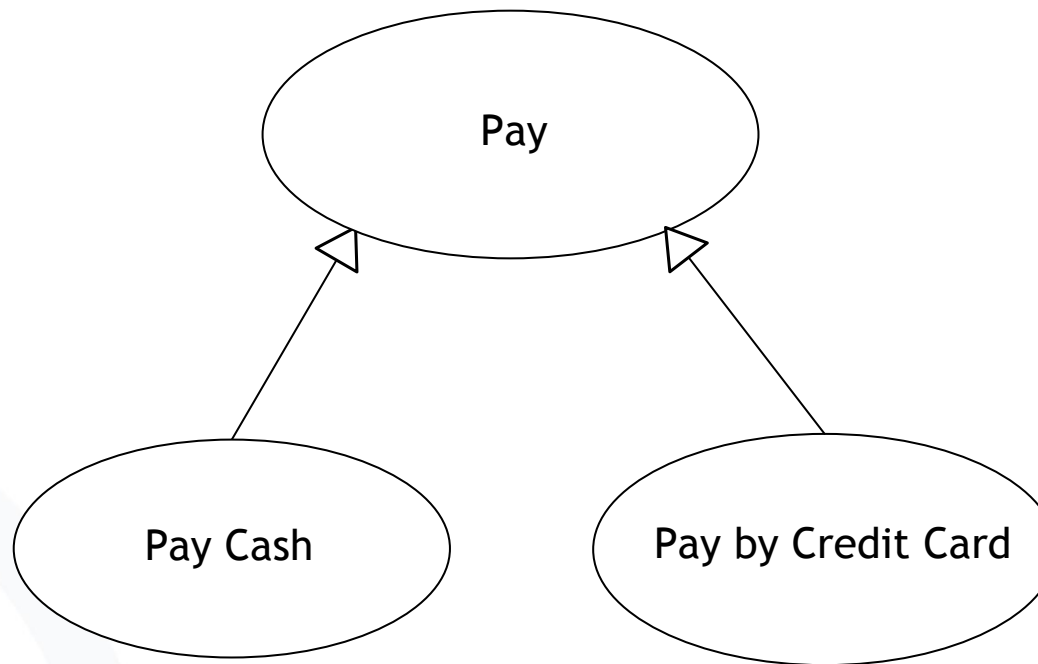
- **Association**

- Interaction of an actor with a use case



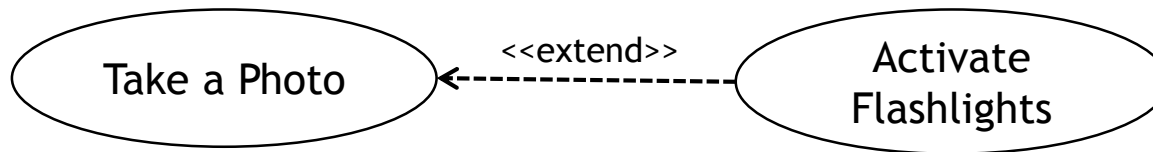
- **Generalisation**

- Generalisation of Use Cases
- UseCase2 generalises the behaviour of UseCase1



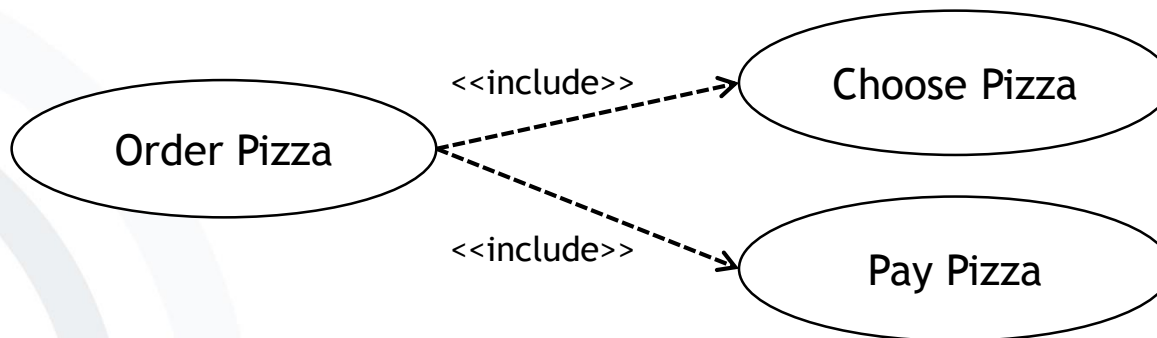
- **<<extend>>**

- Extends an Use Case
- UseCase2 extends UseCase1



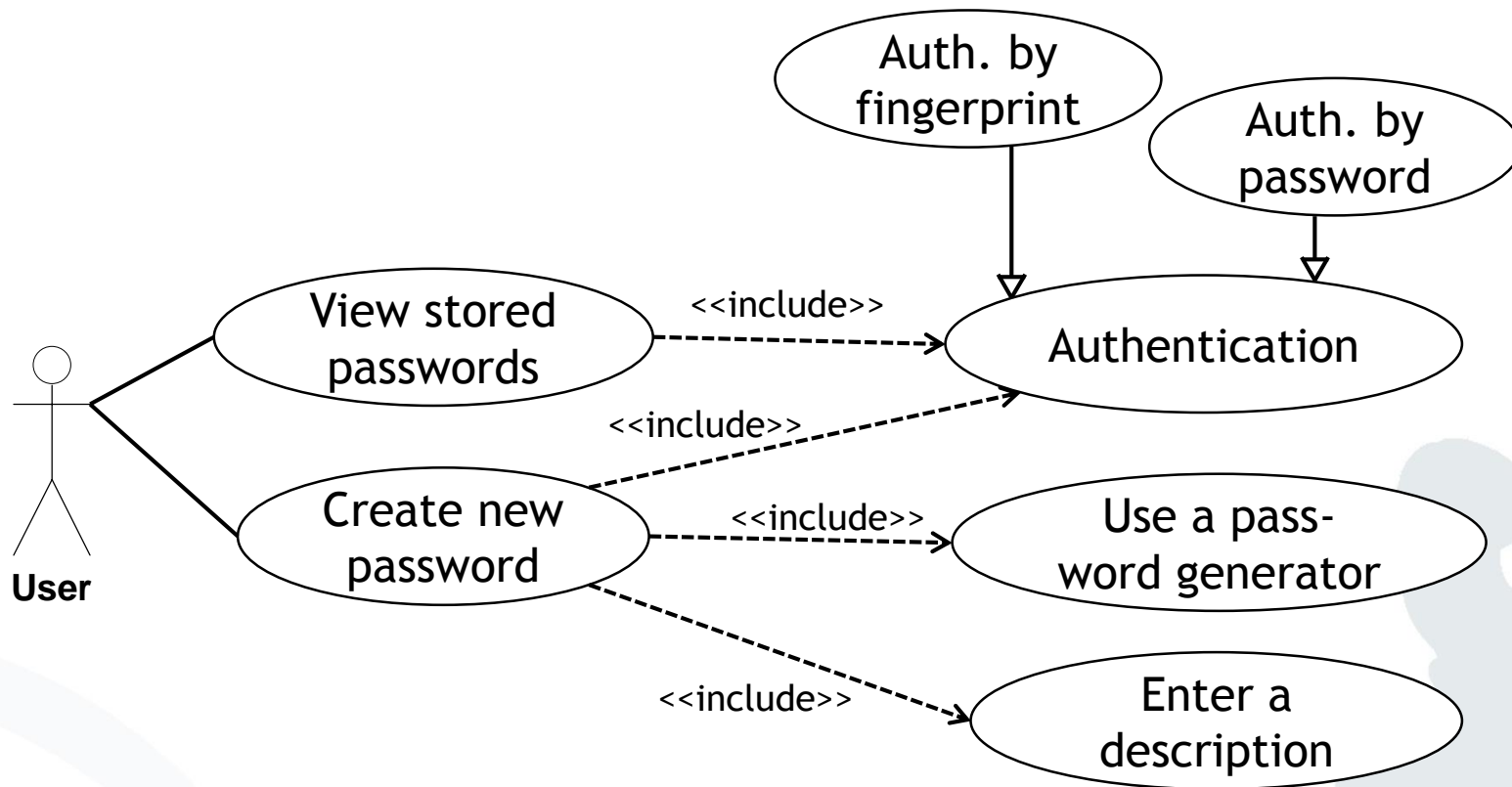
- **<<Include>>**

- Inclusion of an Use Case
- UseCase1 includes the behaviour of UseCase2

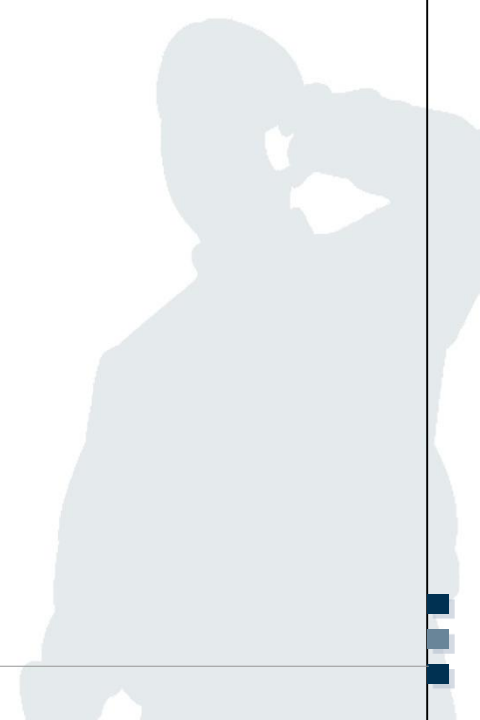




# Use Case Diagram

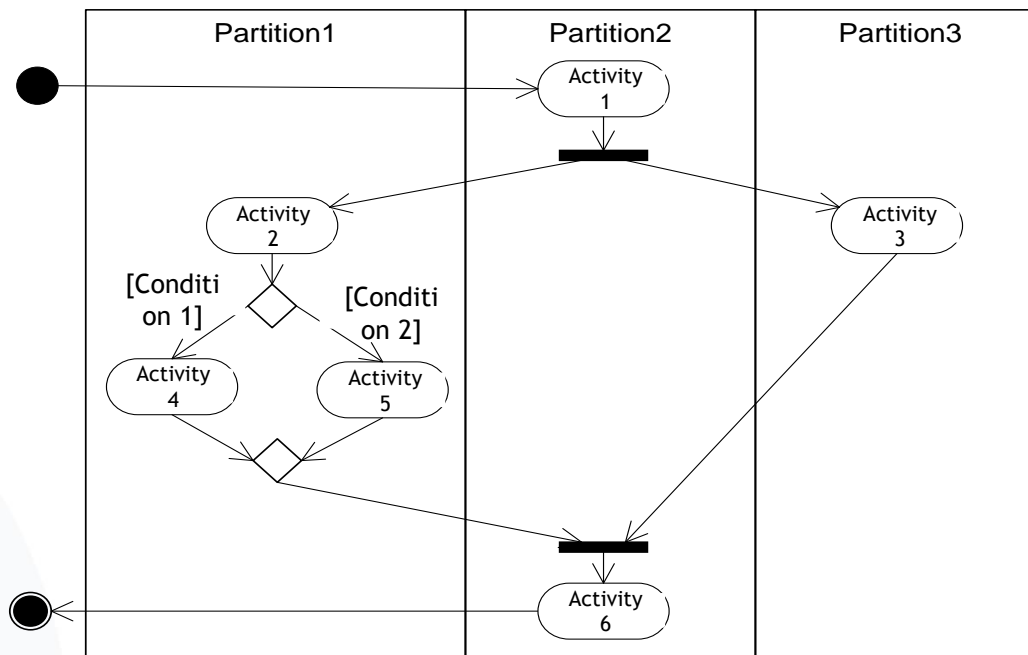


# UML - Activity Diagram



## Activity diagrams

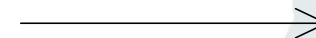
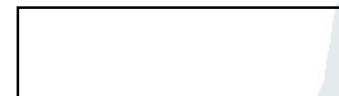
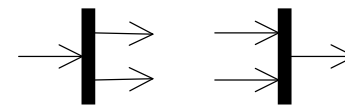
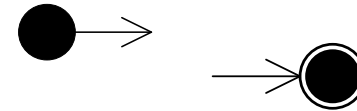
- Activity diagrams are used to model workflows in a system.
- Central element: Activity  
An activity is an “action” within a process.
- Activities are structured by responsibilities.

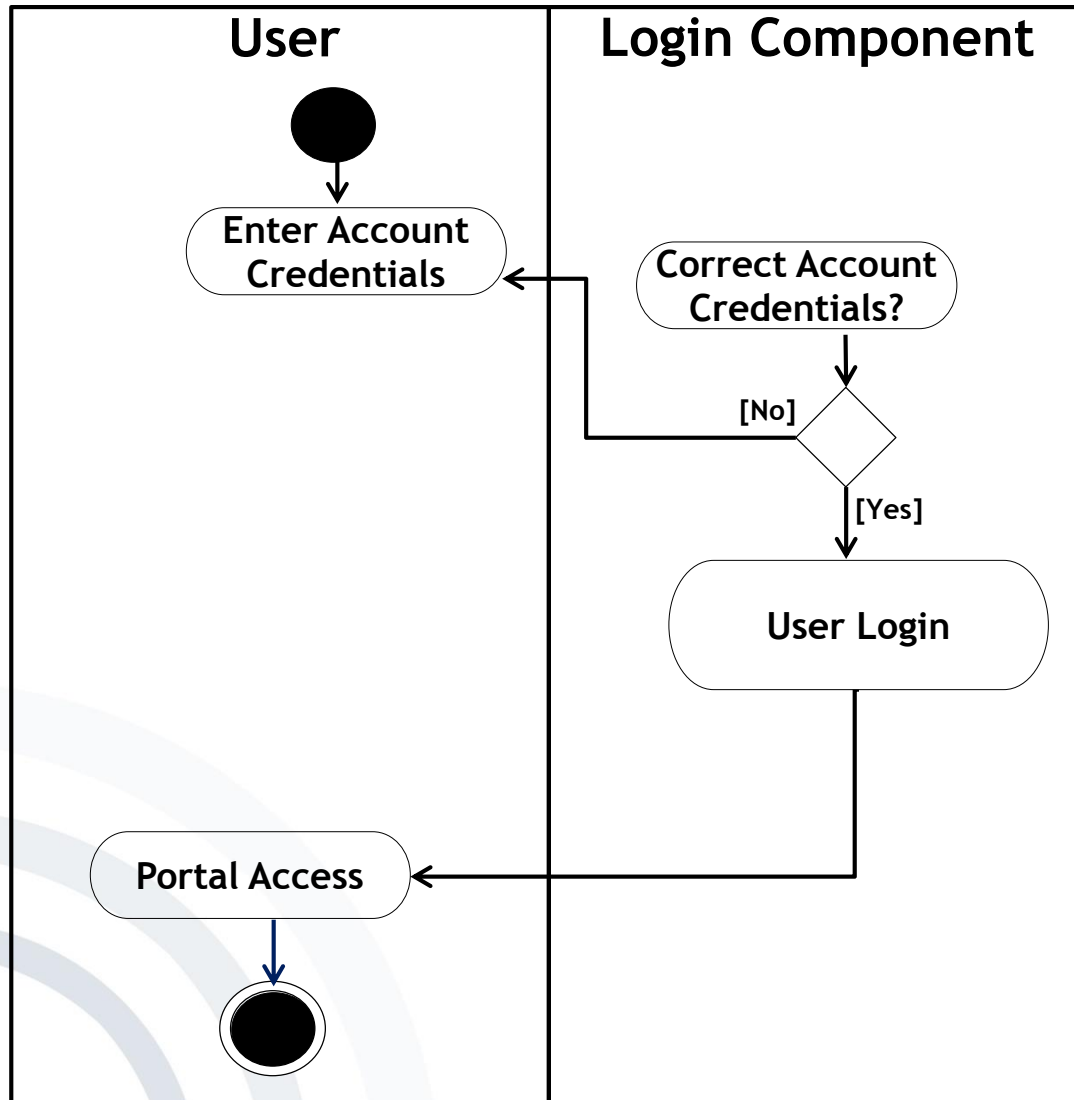


## Notation elements

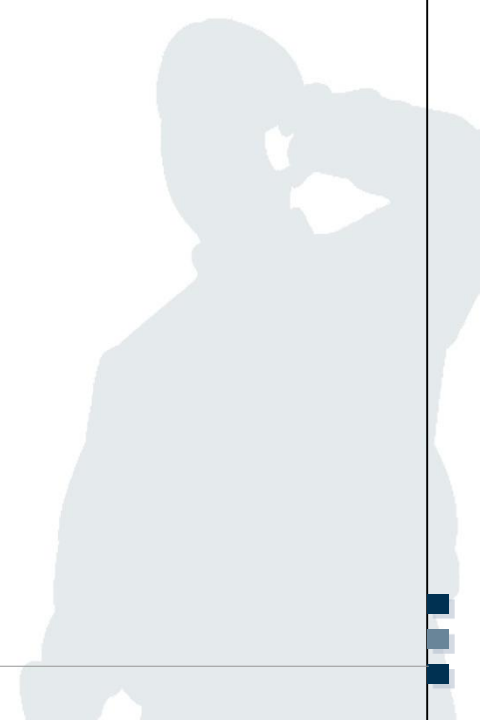
- Initial state/final state
- Activity
- Decision
- Split/join
- Responsibility
- Activity flow

# Activity Diagram



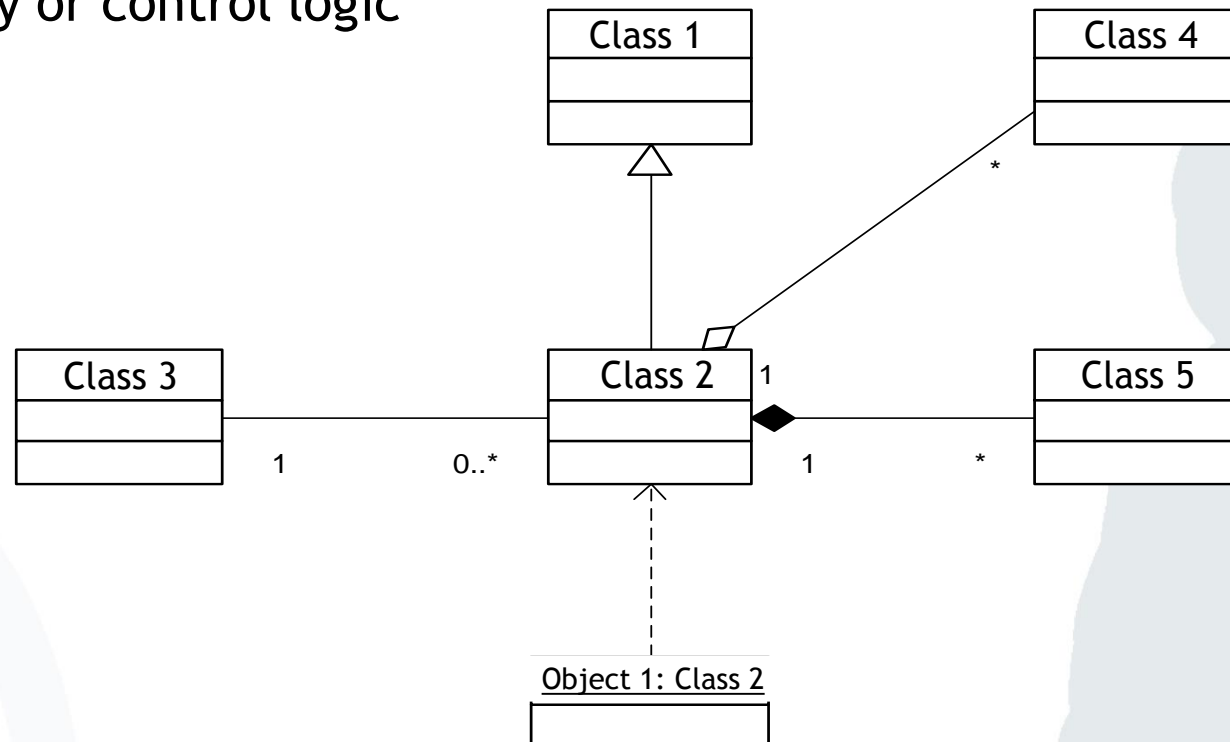


# UML - Class Diagram

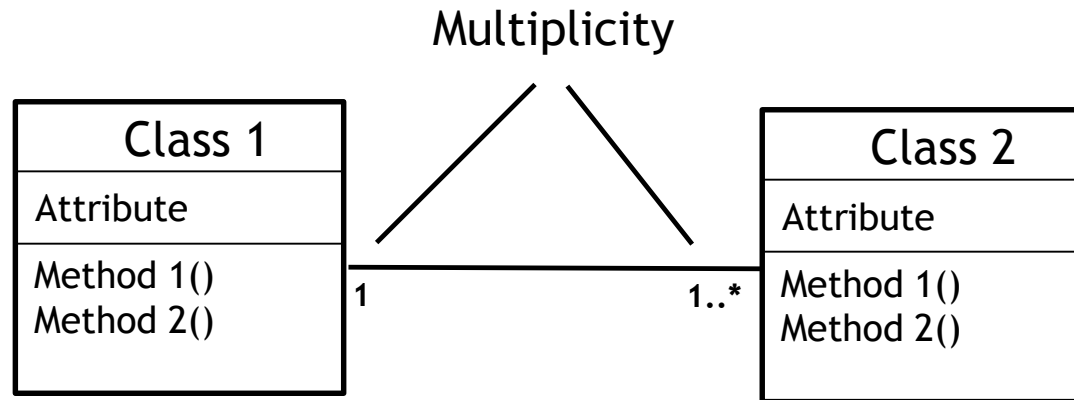


## Class diagrams

- Representation of the static structure of a software system
- Description of logical relations between structural elements
- No activity or control logic



# Class Diagram - Associations

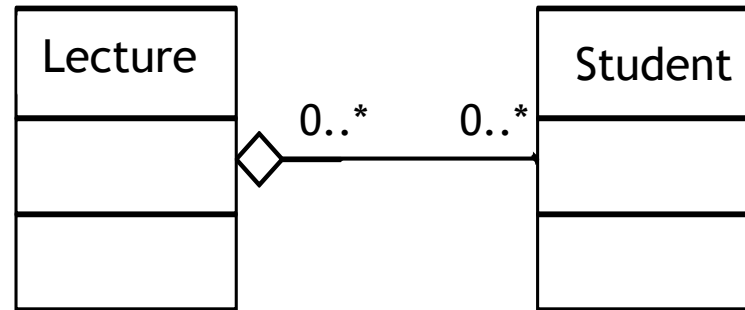


- Describes the relationship between two classes
- It is represented by a line connecting the two classes.
- The multiplicity min..max attached to the association defines the minimal or maximal number of associations between the objects of the two classes.

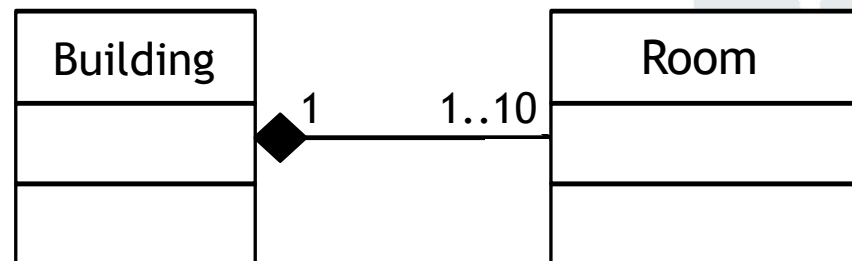
(\*) denotes any number of objects.



- Aggregation
  - Denotes a „has a“ relationship

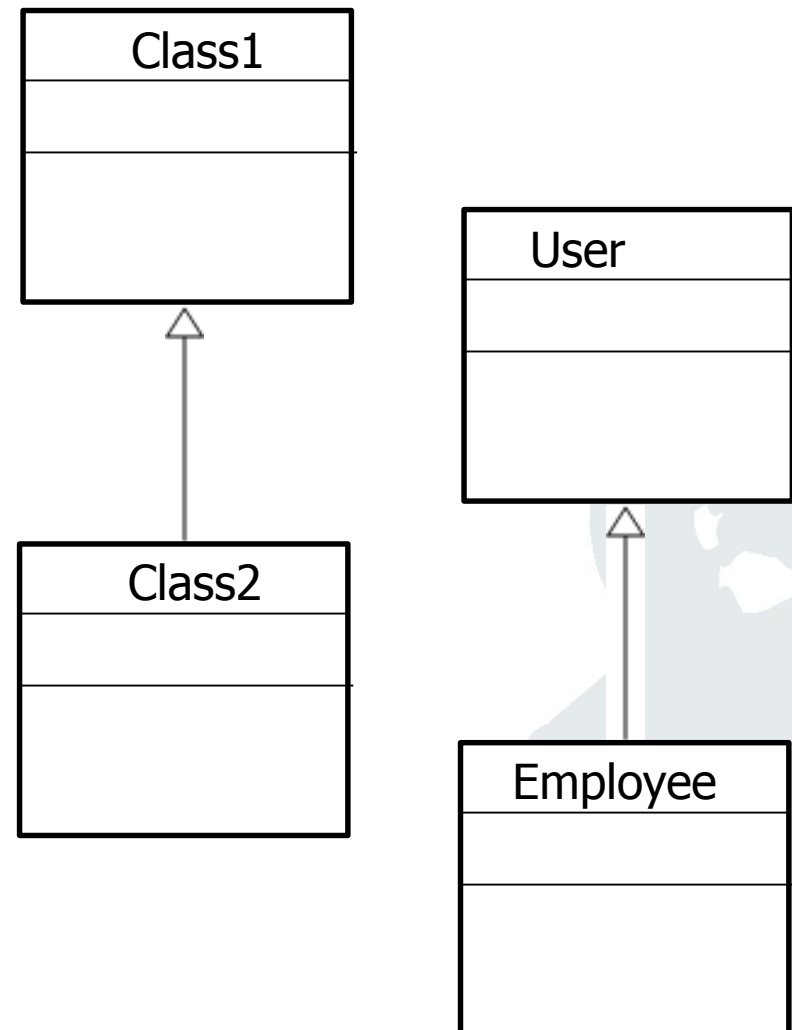


- Composition
  - Composition is a stronger variant of the aggregation
  - Denotes an „owns a“ relationship



# Class Diagram - Inheritance

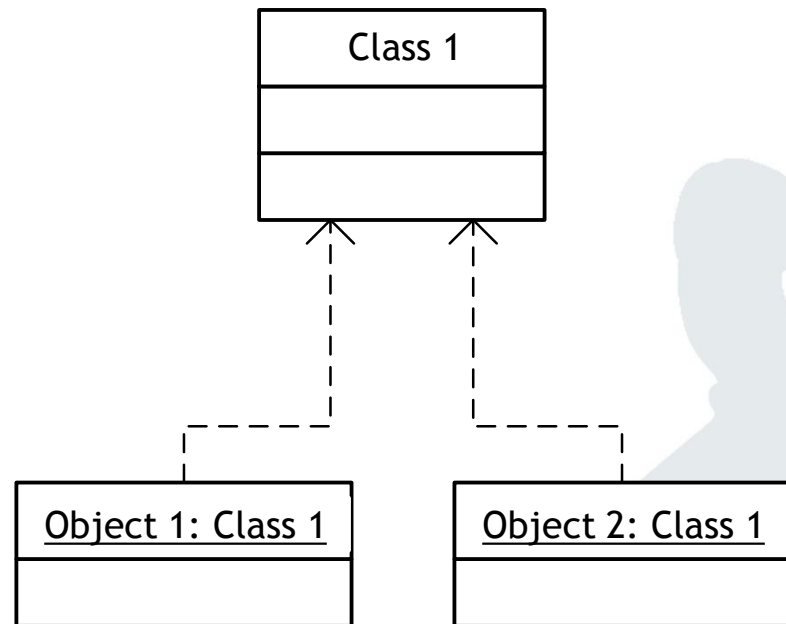
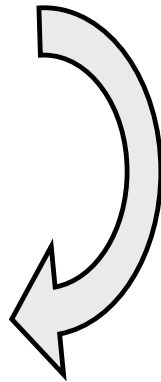
- Denotes an relation between parent class and sub class
- Is represented by a line with an empty arrow at the end, pointing towards the parent class
- Class2 inherits from Class1.
- Purpose:
  - Reuse code, by objects which can be based on previously created objects



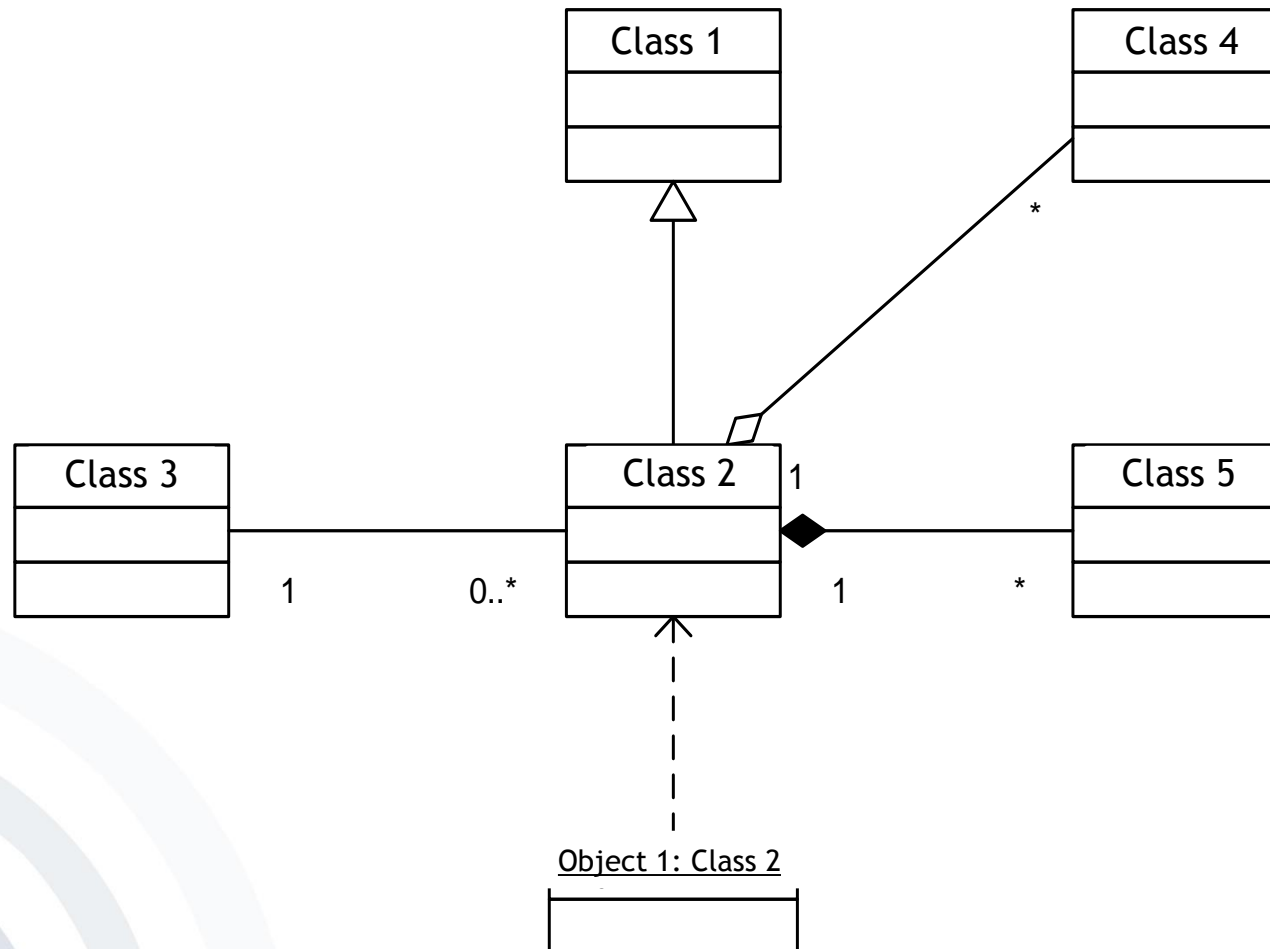
# Class Diagram - Instantiation

- Representation of the relation “class-object”
- An object is an instance of a class.

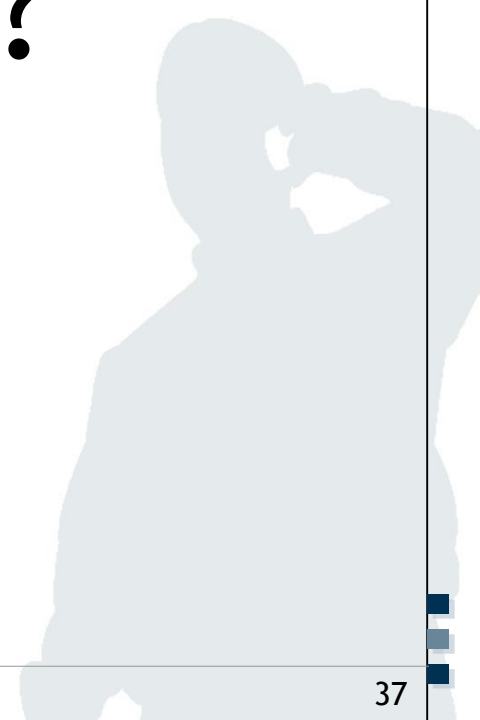
- Class
  - Attributes
  - Methods
- Object
  - Attribute values
  - Messages



# Class Diagram Example



# Open Questions?



## Lecture 14

### Business Informatics 2 (PWIN)

FAQ

WS 13/14

Prof. Dr. Kai-Inn Voigt  
[www.m-chair.net](http://www.m-chair.net)

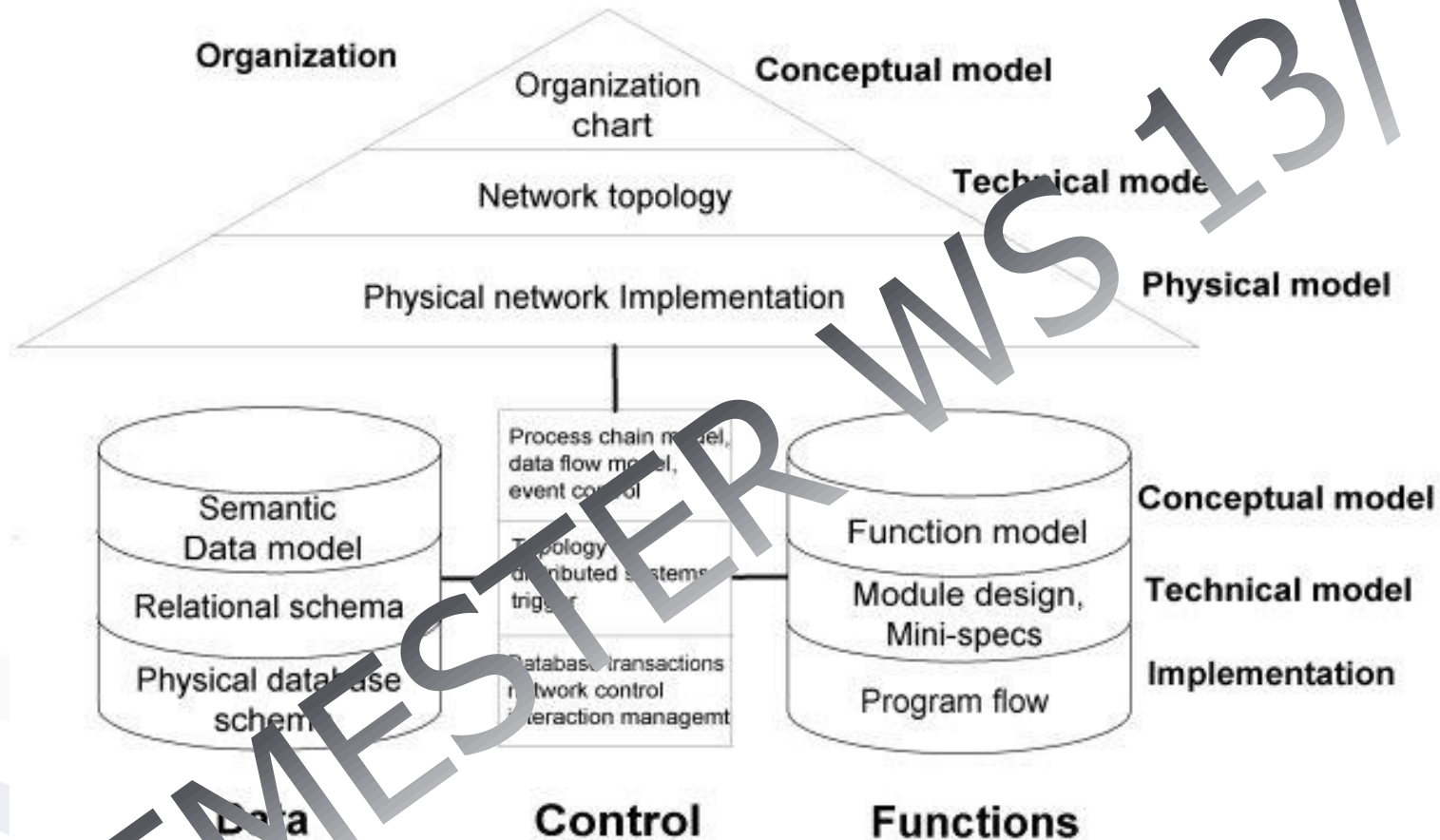


Jenser (Flickr.com)

Student:

- Detaillierte Untergliederung von ARIS definieren/anwenden können?
- Organisation (Conceptual, technical, physical model) noch einmal genauer erklären
- Genaue Differenzierung zw. Ressource und Organisational view

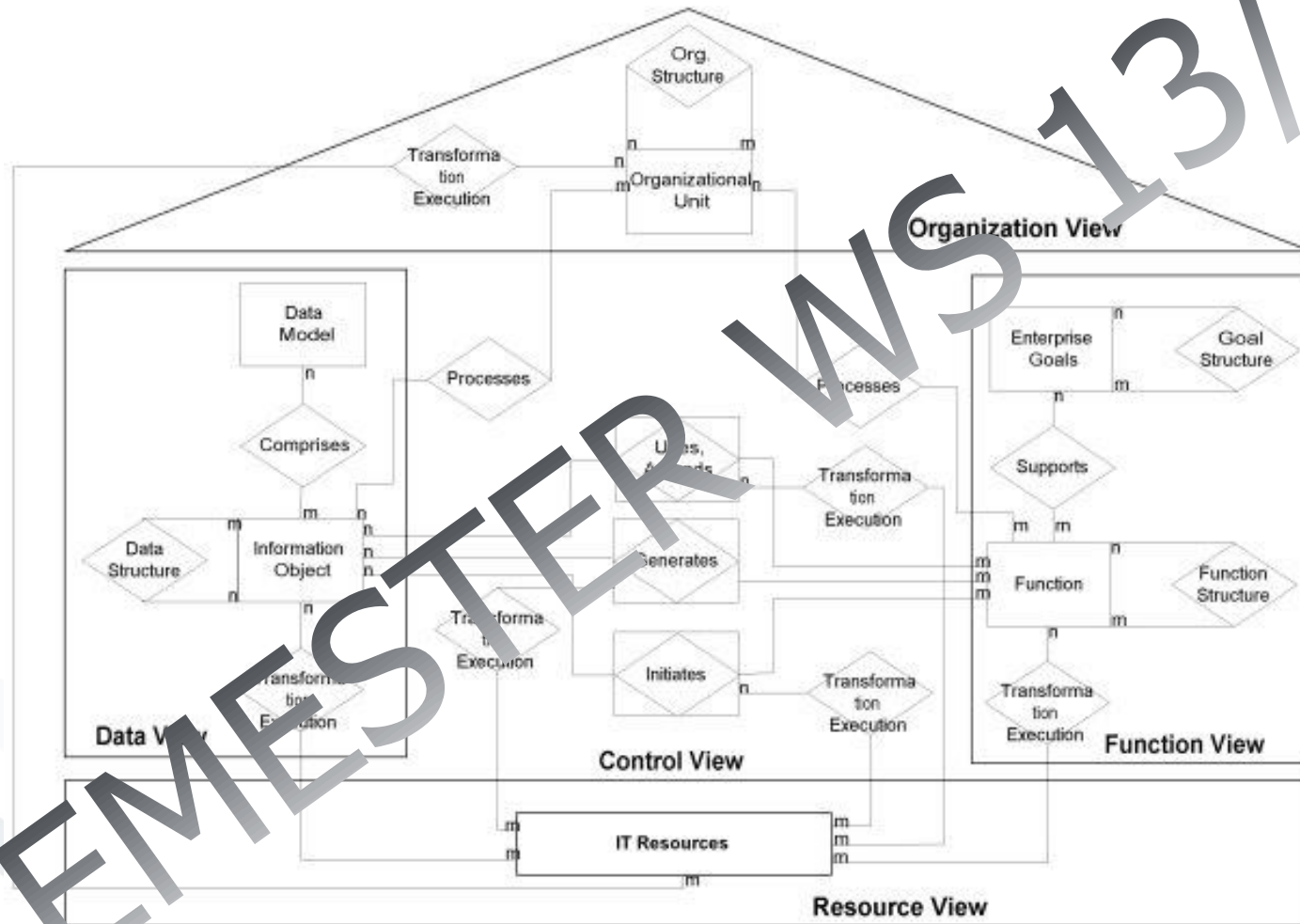
# ARIS - Description Layers



## ARIS Architecture

Source: pera.net, 2011





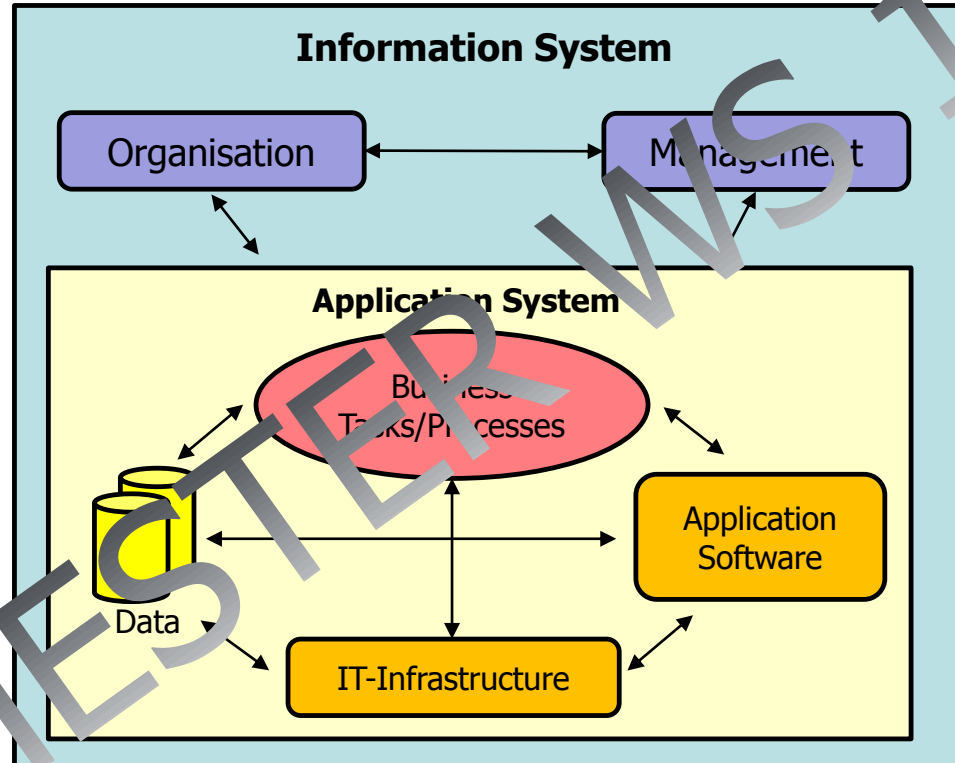
Information model of the ARIS architecture

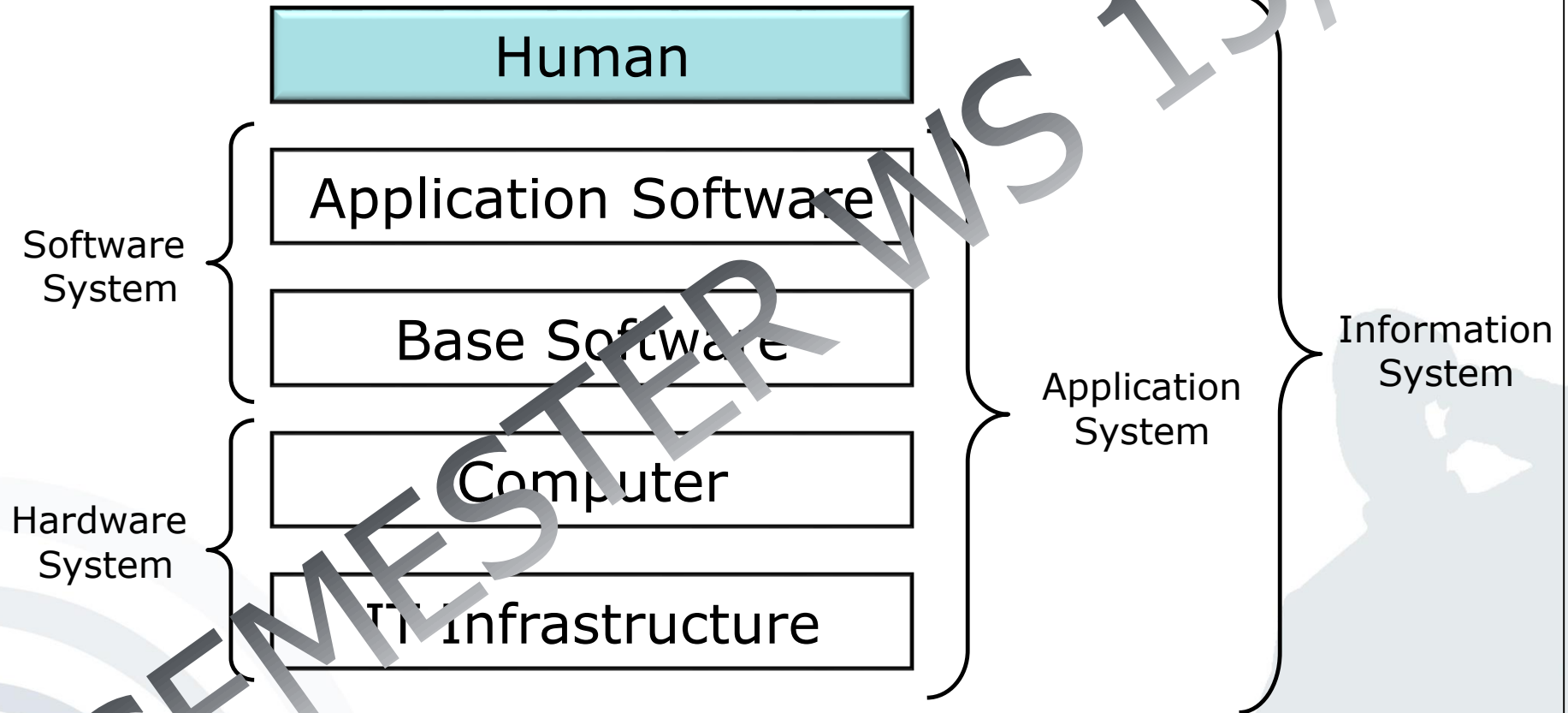
Source: pera.net, 2011

Mentorium 1, Folie 17:

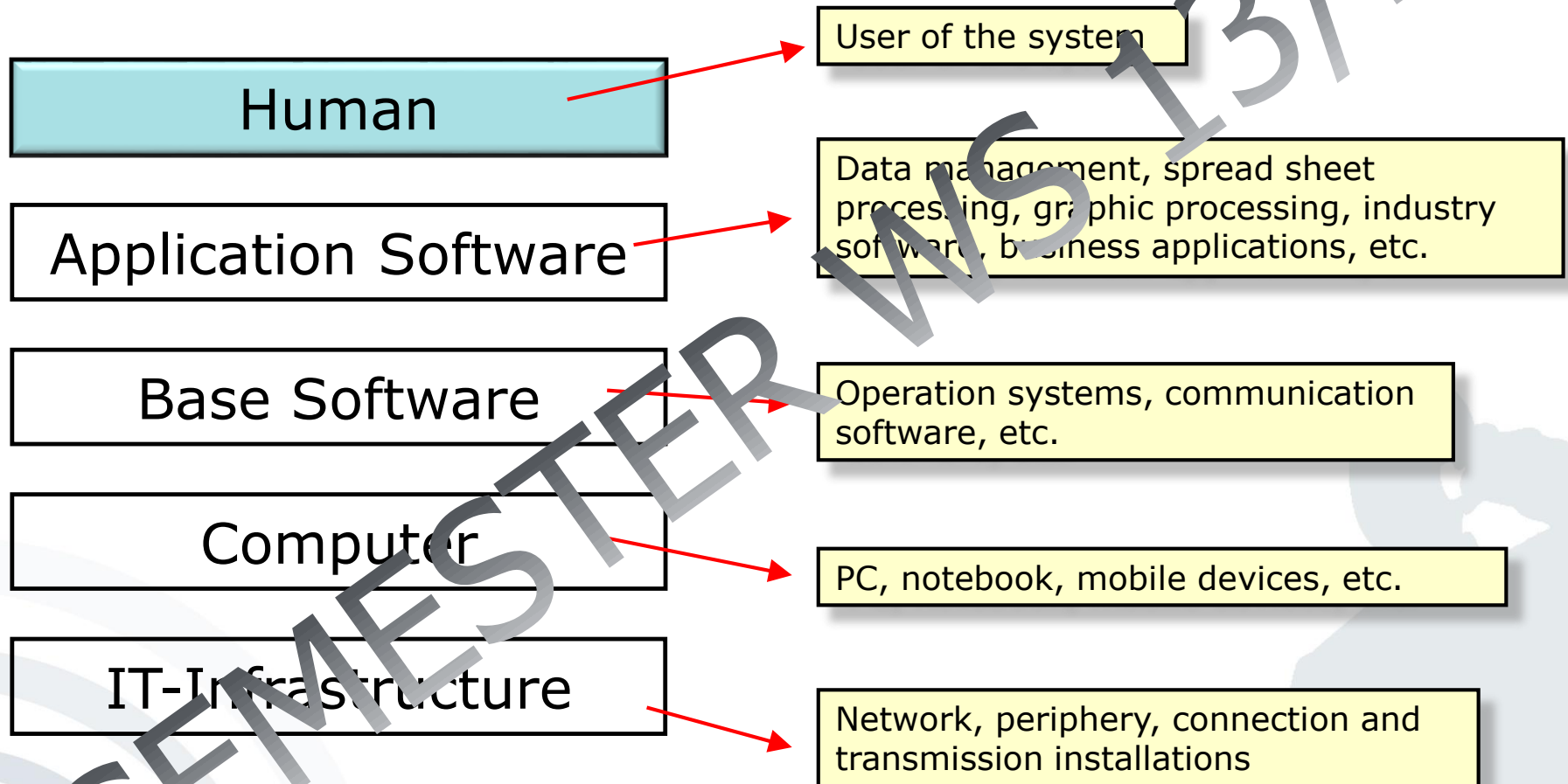
- Woher weiß man, welche Sicht man nimmt (Unternehmenssicht/Individuum)?
- Was ist mit „Beziehung zueinander“ gemeint?

- Describe the difference between **Information Systems** and **Application Systems**.
- Referring to the myPlace Service, give an example for an Information System as well as an Application System and describe their relation to each other.





Source: Teubner (1999)



Source: Teubner (1999)

Unterschied zw. Enterprise model/modelling noch  
einmal erläutern

- Enterprise Modelling

- Enterprise modelling is the abstract representation, description and definition of the structure, processes, information and resources of an identifiable business, government body, or other large organization.

(Source: Leondes and Frymuth Jackson 1992.)

- Enterprise Model

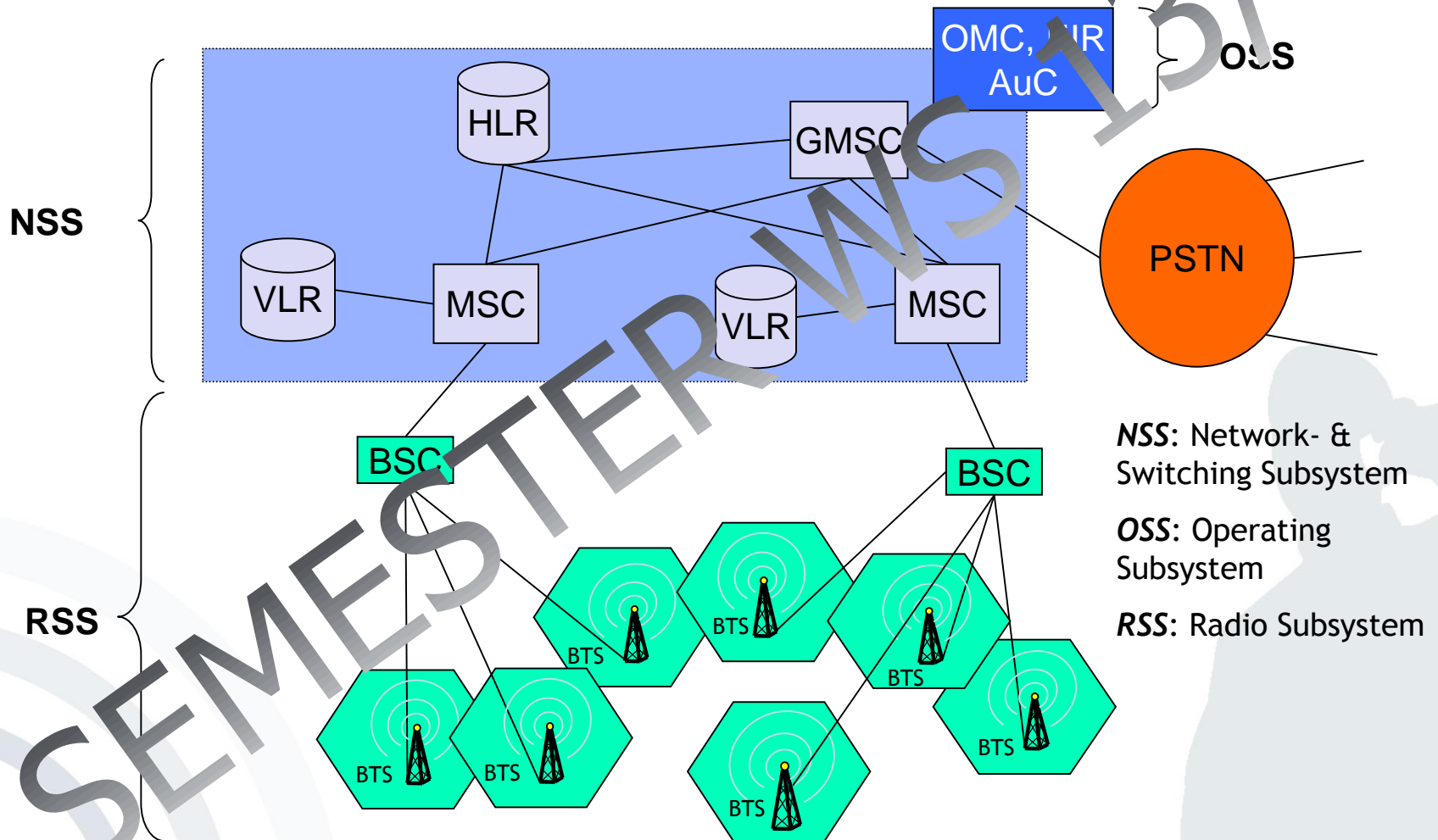
- An enterprise model is a representation of the structure, activities, processes, information, resources, people, behaviour, goals, and constraints of a business, government, or other enterprises.

(Source: F.B. Vernadat 1997)



Wie tief müssen wir denn bei CSM einsteigen? Es gibt  
wahnsinnig viele Abkürzungen innerhalb der  
Architektur  
und Verknüpfungen (Chapter 6)..

# GSM (2G) System Architecture



Based on [Schiller2003]

Können Sie noch einmal den genauen Unterschied zwischen Schicht 3 und 4 im OSI-Modell erklären?

- The Network Layer takes care of the transmission of packets **from the source to the destination**. This may also include *passing through* network segments on the way between two routers.
  - The main task of this layer is **routing**.
  - Internet Protocol (IP) is the protocol used in the Network Layer.

- Routing is the process of selecting paths and forwarding packets while making sure the best possible path to the destination network is used.
- Best possible means:
  - Low transport costs
  - Fastest possible transport
  - Bug-proof transmission
  - Tamper resistant transmission
  - Optimized network load
  - Weighted according to a combination of the criteria above
  - etc

- Various routing algorithms exist:
  - Bellman equation (“Principle of Optimality”)
  - **Dijkstra Algorithm**
  - Ford Algorithm
  - Distance Vector Routing
  - Link State Routing
  - Hierarchical Routing
  - Broadcast Routing
  - Multicast Routing
  - ..

- Transport Layer (Layer 4) constitutes an “interface” between the transport-oriented layers below and the application-oriented layers above.
- It takes care of the data transport from the source to the destination (logical end-to-end connection), independent from physical networks.
- The following services are provided to higher layers by the Transport Layer:
  - **Setup, coordination and termination of connections (3-way-handshake)**
  - Flow control and Buffering
  - Multiplexing
- Important protocols:
  - Transmission Control Protocol (TCP)
  - User Data Protocol (UDP)

- Example from everyday life - making an appointment via correspondence

Prof. Rannenberg wants to make an appointment with Prof. König via correspondence.

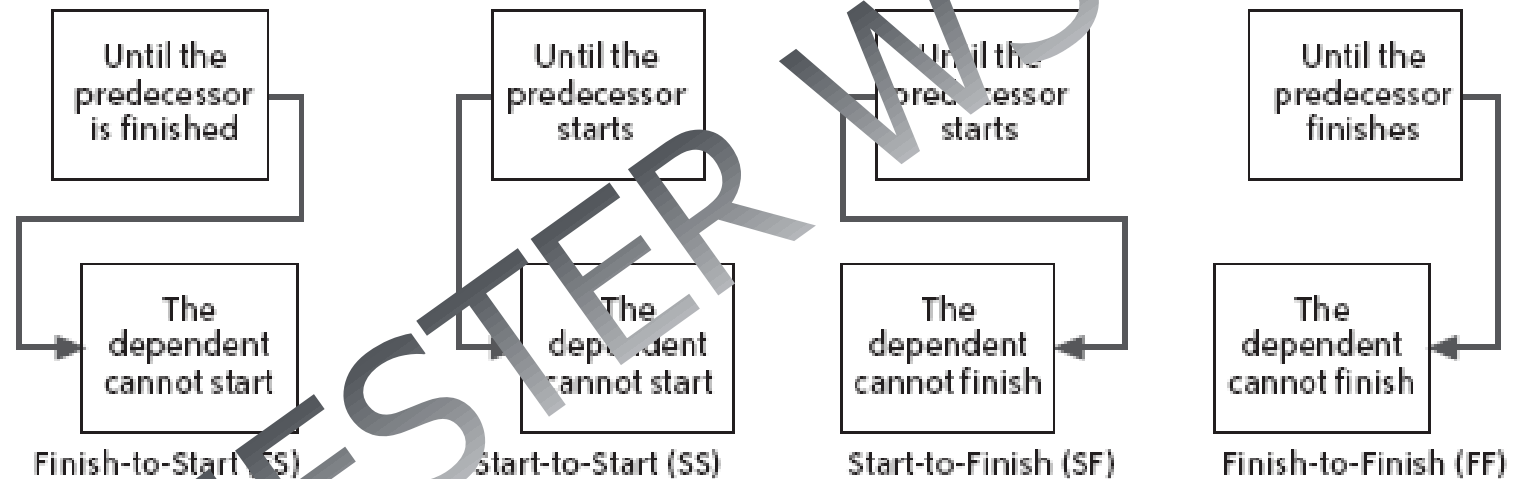
1. Prof. Rannenberg sends a message to Prof. König to suggest an appointment date.
2. Prof. König confirms the appointment date by sending a message back to Prof. Rannenberg.
3. Prof. Rannenberg sends a message to Prof. König to let him know that he received the confirmation message.

Step 3 is necessary in order for Prof. König to know that Prof. Rannenberg has received the confirmation. Message No. 2 could have gotten lost and then Prof. König would show up alone for the meeting.

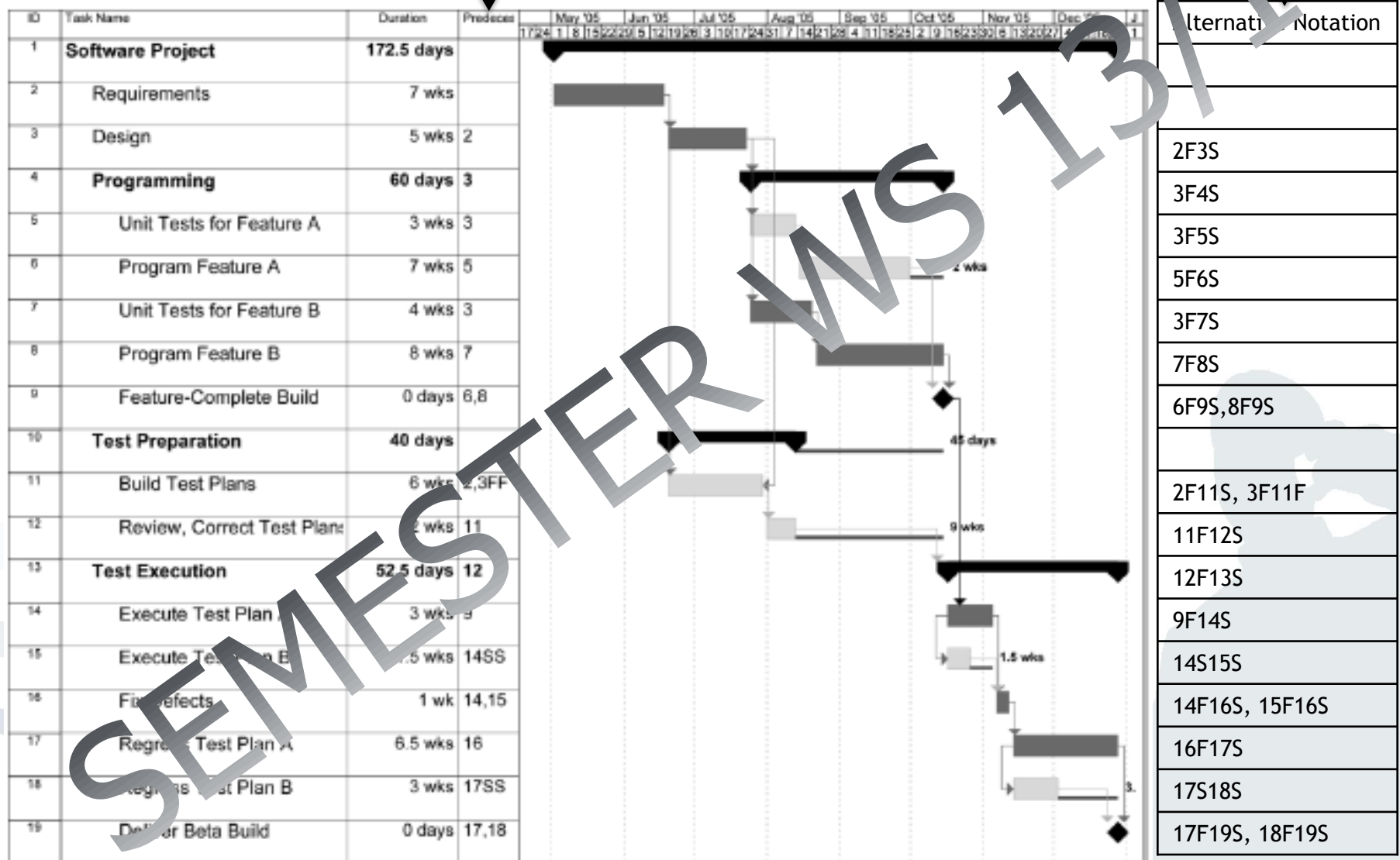


Create a Schedule

- Possible dependencies between tasks:



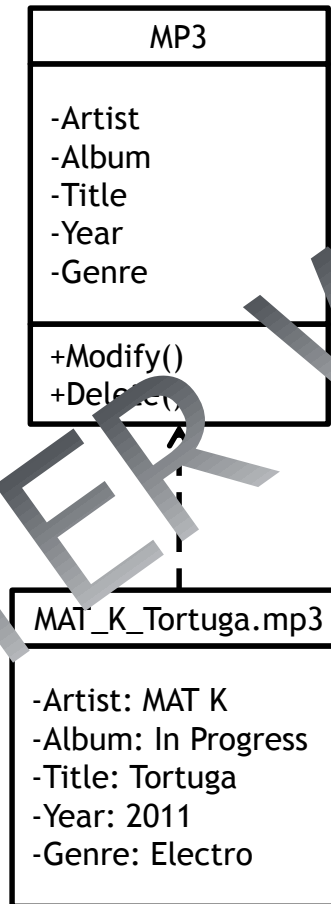
Source: Stellmann, Greene (2006)



Create a Class Diagram

- Create a class diagram including the class “MP3” and the associated instance (object) „MAT\_K\_Tortuga.mp3“. MP3 files have ID-Tags, which store information about the artist, album, title, year, genre and track number. This information can be modified or deleted. The file with the name „MAT\_K\_Tortuga.mp3“ was composed by „MAT K“ and has the title „Tortuga“, which is released in the album „In Progress“. This electro title was recorded in 2011.

- Create a class diagram including ...
- ... the class “MP3” and ...
- ... the associated instance (object) „MAT\_K\_Tortuga.mp3“.
- MP3 files have information about the artist, ...
- ... album, ...
- ... title, ...
- ... year, and
- ... genre.
- This information can be modified or ...
- ... deleted.
- The file with the name „MAT\_K\_Tortuga.mp3“ ...
- ... was composed by „MAT K“ and ...
- ... has the title „Tortuga“, which is released ...
- ... in the album „In Progress“.
- This electro title ...
- ... was recorded in 2011.



Create an Activity Diagram



- *Create an activity diagram based on the following description:*
  - The activity starts when Joe uploads a file into the portal which transfers the file to the server which then receives the file and checks it for errors.
  - In case of an error the server sends a message to the portal which notifies Joe of an error who then uploads his file again.
  - In case of success the server notifies the portal which then notifies Joe.
  - If the file is okay, the server waits for Sarah to start a second process in order to start the processing of the file after which the whole activity ends.

- (1) The activity starts when Joe uploads a file into the portal ...
- (2) ... that transfers the file to the server ...
- (3) ... which then receives the file ...
- (4) ... and checks it for errors.
- (5) In case of an error the server sends a message to the portal ...
- (6) ... which notifies Joe of an error
- (7) ... who then uploads his file again.
- (8) In case of success ...
- (9) ... the server notifies the portal ...
- (10) ... which then notifies Joe.
- (11) If the file is okay, ...
- (12) ... the server waits for Sarah ...
- (13) ... to start a second process ...
- (14) ... in order to start the processing of the file ...
- (15) ... after which the whole activity ends.

