Business Process Simulation

Lecture 2

Laura Genga

Department of Industrial Engineering and Innovation Sciences, Information Systems Group

Overview on lecture modules

a) Petri nets

b) UML class diagram

Recap Simulation Methodology

STEP 1: Project Definition

1.1 Decision frame

1.2 Research questions

1.3 Scope and level of detail

STEP 2: Design the study (black box & assumptions)

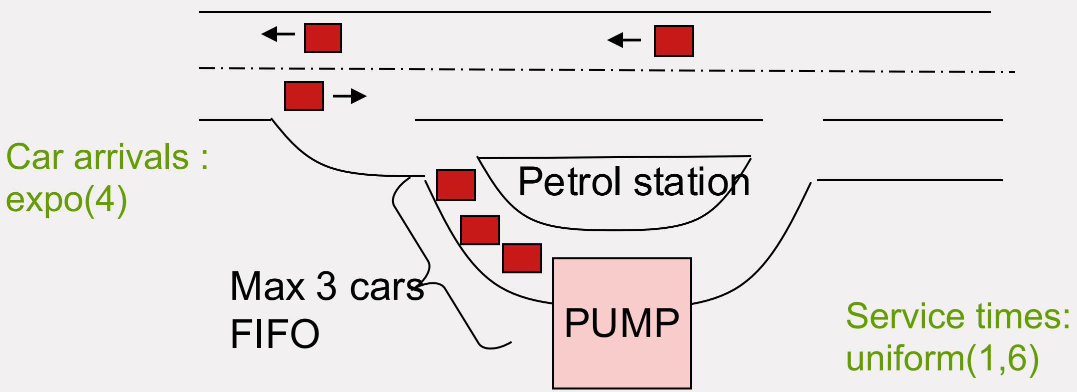
2.1 Black box representation

2.2 Assumptions and givens

2.3 Simulation suitable / needed?

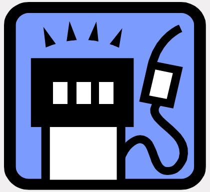
2.4 Number of models

EXAMPLE: The Petrol Station

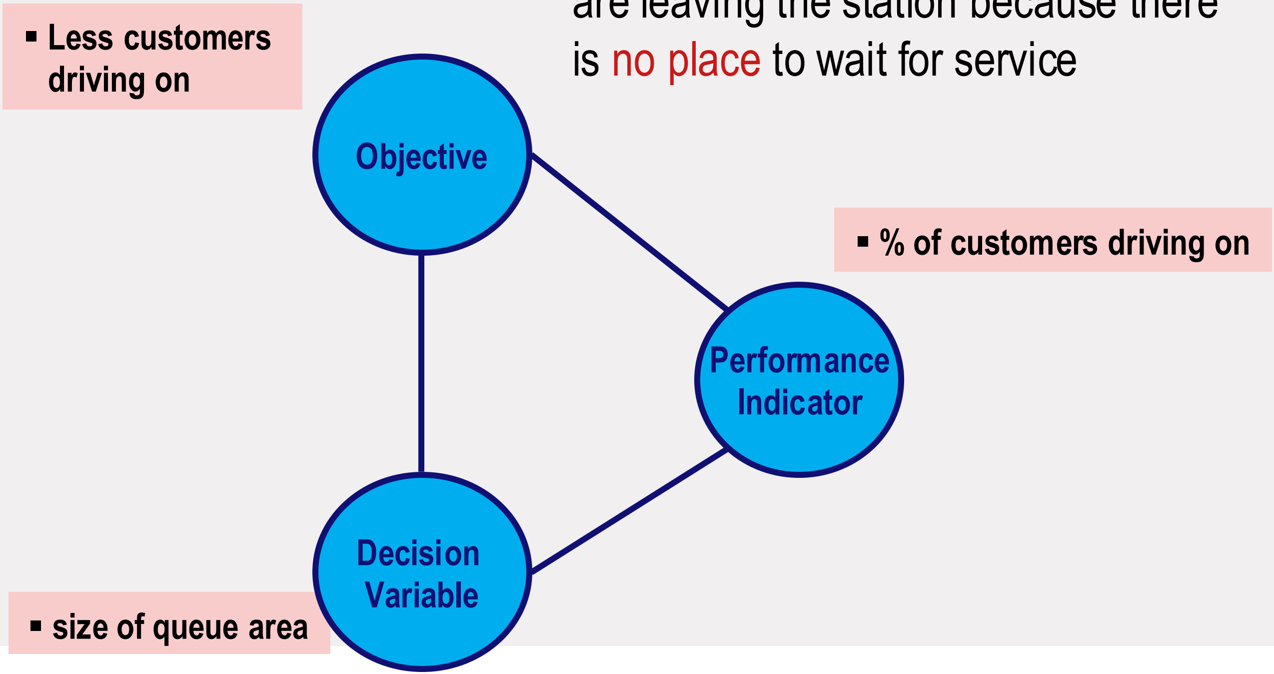


The owner of the petrol station has the feeling that some potential clients are leaving the station because there is no place to wait for service

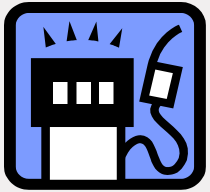
The Petrol Station – STEP 1 Decision Frame



…the feeling that some potential customers are leaving the station because there is no place to wait for service



The Petrol Station – STEP 2 Black box representation



Environmental Variables

- interarrival time of cars
- service time of cars

Decision variables → PETROL STATION → **Output variables**

- Size of the queue area

Output variables:

- percentage not served
- (mean waiting time)
- (number not served)

TU/e

Business Process Simulation

Lecture 2a – Petri nets

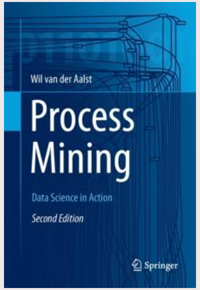
Laura Genga

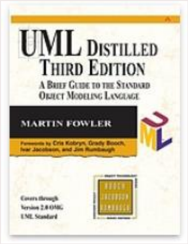
Department of Industrial Engineering and Innovation Sciences, Information Systems Group

@Thanks to Prof. Karolin Winter for the material

Literature for this Lecture

Chapter 3.2.2 and 3.2.3
W.M.P. van der Aalst. Process Mining Data Science in Action. Springer-Verlag 2016 Online ISBN 978-3-662-49851-4
https://link.springer.com/book/10.1007%2F978-3-662-49851-4

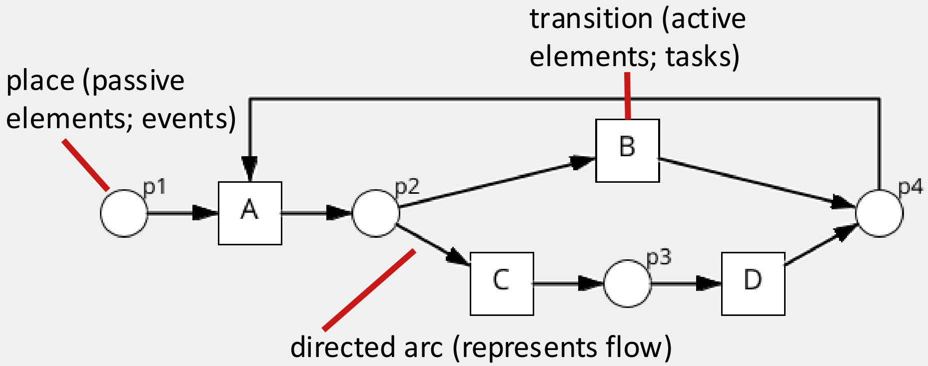




UML Distilled
https://martinfowler.com/books/uml.html

Petri Nets – Basic Modeling Elements

Petri net – graphical representation (example) ⟷ Petri net – formal representation



A *Petri net* is a triplet $N = (P, T, F)$ with

- P finite set of places
- T finite set of transitions such that $P \cap T = \emptyset$
- $F \subseteq (P \times T) \cup (T \times P)$ set of directed arcs (flow relations)

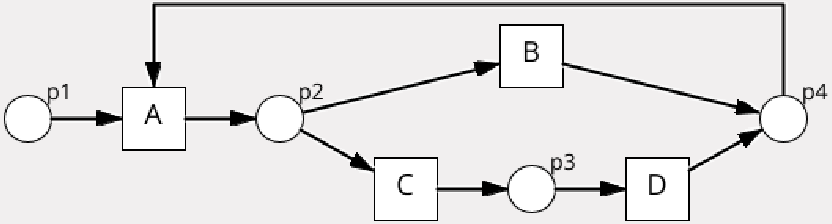
W. van der Aalst. Process Mining: Data Science in Action. Springer-Verlag 2016.
eBook ISBN 978-3-662-49851-4. https://doi.org/10.1007/978-3-662-49851-4

How does the formal representation of the example look like?

P = {p1,p2,p3,p4} T = {A,B,C,D} F = {(p1,A),(A,p2),(p2,B),(p2,C),(B,p4),(C,p3),(p3,D),(D,p4),(p4,A)}

www.menti.com
Code: *14219820*

Petri Nets – Input and Output Places



What are the sets of input and output nodes of transition A?

$\bullet$A={p1,p4}
A$\bullet$={p2}

A *Petri net* is a triplet $N = (P, T, F)$ with

- P finite set of places
- T finite set of transitions such that $P \cap T = \emptyset$
- $F \subseteq (P \times T) \cup (T \times P)$ set of directed arcs (flow relations)

W. van der Aalst. Process Mining: Data Science in Action. Springer-Verlag 2016.
eBook ISBN 978-3-662-49851-4. https://doi.org/10.1007/978-3-662-49851-4

For any node $x \in P \cup T$ the set of

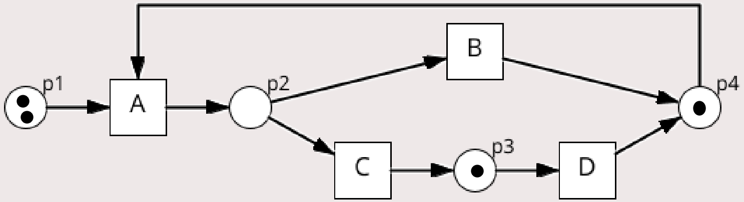
- *input nodes* is defined as $\bullet x = \{y | (y, x) \in F\}$
- *output nodes* is defined as $x \bullet = \{y | (x, y) \in F\}$

W. van der Aalst. Process Mining: Data Science in Action. Springer-Verlag 2016.
eBook ISBN 978-3-662-49851-4. https://doi.org/10.1007/978-3-662-49851-4

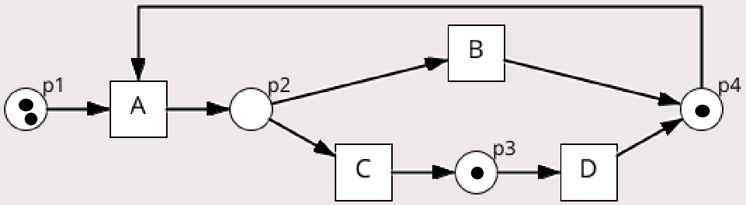
Petri Nets – Tokens

Tokens in a Petri net

- are represented by black dots
- reside in places
- are consumed and produced by transitions (active elements of the Petri net!); action is called *firing* (how is determined by firing rules)
- constitute the dynamic aspects and state of a Petri net
- represent currently running process instances (can usually not be distinguished for basic classes of Petri nets)



Petri Nets – Markings



The *marking* (or *state*) of a Petri net (P, T, F) is defined by a function $M : P \rightarrow \mathbb{N}$ mapping the set of places onto the natural numbers, where $\mathbb{N}$ is the set of natural numbers including 0.

Weske, M. (2019). Business Process Management: Concepts, Languages, Architectures. Springer-Verlag GmbH Germany, part of Springer Nature 2019. https://doi.org/10.1007/978-3-662-59432-2

What is the current marking of the Petri net?

0=M(p2)

1=M(p3)=M(p4)

2=M(p1)

Markings can also be expressed as

- array (places totally ordered by their identifier): M=[2,0,1,1]
- multi-set: M=[$p1^2$,p3,p4]

A pair (N,M) with N being a Petri net and M a marking for N is called a marked Petri net.

The set $\mathcal{M}_P$ denotes the set of all markings for a Petri net P.

What is the set $\mathcal{M}_P$ of the Petri net, i.e., what are all possible markings given this initial marking?

→ reachability analysis

Petri Nets – Different Classes

- Labeled Petri nets: A *labeled Petri net* $N = (P, T, F, A, l)$ is a Petri net together with a set of activity labels A and a labeling function $l \in T \to A$.

W. van der Aalst. Process Mining: Data Science in Action. Springer-Verlag 2016.
eBook ISBN 978-3-662-49851-4. https://doi.org/10.1007/978-3-662-49851-4

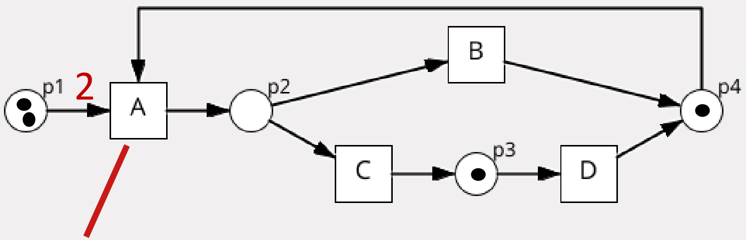
Note that sometimes it becomes necessary to introduce transitions which mainly serve to properly model a Petri net. We call those transitions silent transitions and label them as tau.[1]

- Place-Transition nets: transitions can consume and produce more than one token from one place
- Workflow nets: sub-class tailored towards business processes
- Colored Petri nets: can deal with data-related and time-related aspects
- ...

[1]More details on silent transitions see later in this lecture.

Place Transition Nets

We consider in the following *Place-Transition* nets which are Petri nets together with a weighting function that assigns a weight (natural number not including 0) to all arcs.
If no number is given for an arc, weight 1 is assigned to this arc per default.
This weighting function determines the number of tokens a transition consumes or produces.

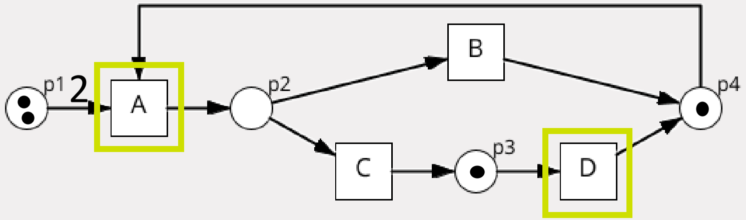


transition A consumes 2 tokens from place p1 and 1 token from place p4 and produces one token in p2

Petri Nets – Firing Rules

www.menti.com
Code: *14219820*

Which transition(s) in the given Place-Transition net are enabled ?



- A transition can fire if it is enabled.
- Firing of a transition is represented by a state change.
- Firing takes no time

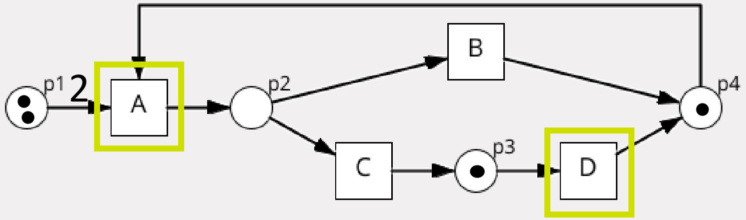
Firing rule for Place-Transition nets:
A transition *t* is enabled, if and only if each input place *p* of *t* contains at least the number of tokens defined as the weight of the connecting arc.

When firing, a transition

- *consumes* from each input place the number of tokens determined by the weighting function on the corresponding ingoing arc and
- *produces* the number of tokens determined by the weighting function on the corresponding outgoing arc in each output place.

Petri Nets – Reachability Analysis

The goal is to determine all markings that are reachable given an initial marking.



For each marking reachable from the initial marking:

1) Determine which transitions are enabled.
2) For each enabled transition determine the marking after firing.

Initial Marking: [2,0,1,1] with transitions A and D enabled.

Problem: Two transitions are enabled but which one fires first?

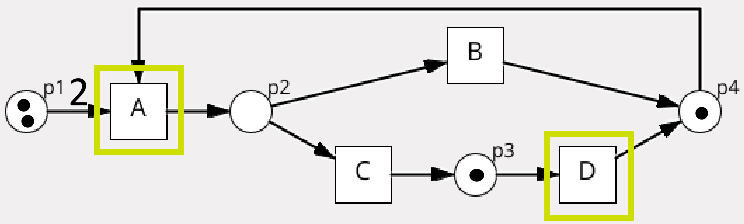
For reachability analysis this does not matter because we need to determine all markings of the Petri net!

In order to not miss out on markings, we should perform the reachability analysis in a systematic way.

Petri Nets – Reachability Analysis Table

We can use a table to perform the reachability analysis in a systematic way for the given Place-Transition net.[1]

Marking	Places				Activated Transitions			
	p_1	p_2	p_3	p_4	A	B	C	D
M_1	2	0	1	1	x			x → M_2
M_2								
M_3								
M_4								
M_5								
M_6								
M_7								



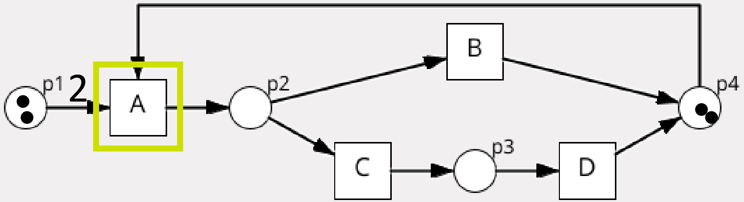
TU/e

[1]Idea based on lecture Einführung in die WI, Rinderle-Ma S., Winter K., TUM 2022/23

Petri Nets – Reachability Analysis Table

We can use a table to perform the reachability analysis in a systematic way for the given Place-Transition net.[1]

Marking	Places				Activated Transitions			
	p_1	p_2	p_3	p_4	A	B	C	D
M_1	2	0	1	1	x			x → M_2
M_2	2	0	0	2	x → M_3			
M_3								
M_4								
M_5								
M_6								
M_7								

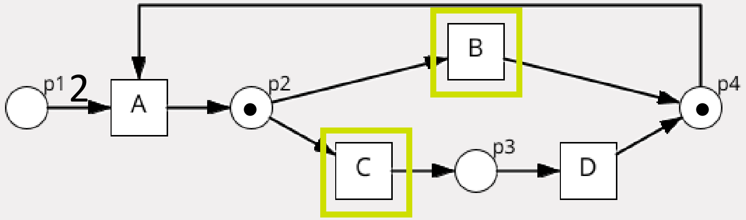


[1] Idea based on lecture Einführung in die WI, Rinderle-Ma S., Winter K., TUM 2022/23

Petri Nets – Reachability Analysis Table

We can use a table to perform the reachability analysis in a systematic way for the given Place-Transition net.[1]

Marking	Places				Activated Transitions			
	p_1	p_2	p_3	p_4	A	B	C	D
M_1	2	0	1	1	x			x → M_2
M_2	2	0	0	2	x → M_3			
M_3	0	1	0	1		x → M_4	x	
M_4								
M_5								
M_6								
M_7								

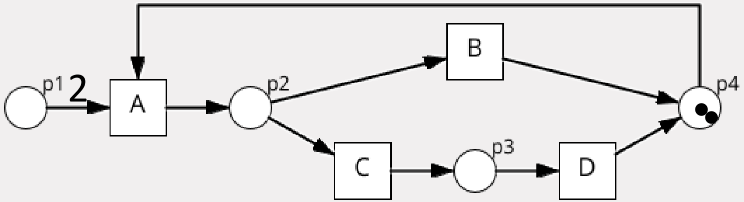


[1]Idea based on lecture Einführung in die WI, Rinderle-Ma S., Winter K., TUM 2022/23

Petri Nets – Reachability Analysis Table

We can use a table to perform the reachability analysis in a systematic way for the given Place-Transition net.[1]

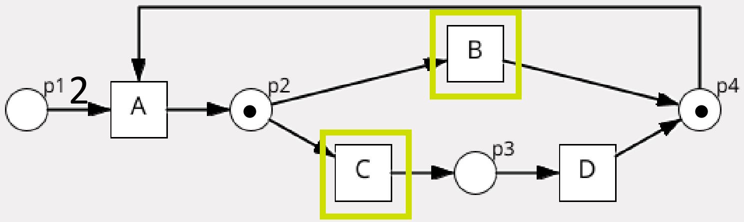
Marking	Places				Activated Transitions			
	p_1	p_2	p_3	p_4	A	B	C	D
M_1	2	0	1	1	x			x → M_2
M_2	2	0	0	2	x → M_3			
M_3	0	1	0	1		x → M_4	x	
M_4	0	0	0	2				
M_5								
M_6								
M_7								



[1] Idea based on lecture Einführung in die WI, Rinderle-Ma S., Winter K., TUM 2022/23

Petri Nets – Reachability Analysis Table

We can use a table to perform the reachability analysis in a systematic way for the given Place-Transition net.[1]



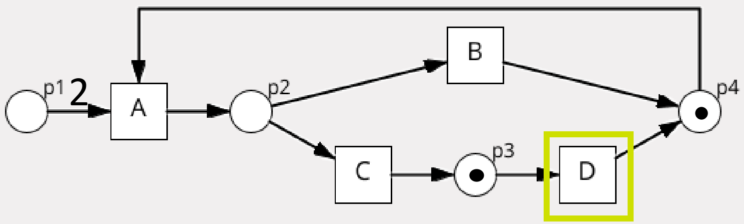
Marking	Places				Activated Transitions			
	p_1	p_2	p_3	p_4	A	B	C	D
M_1	2	0	1	1	x			x → M_2
M_2	2	0	0	2	x → M_3			
M_3	0	1	0	1		x → M_4	x → M_5	
M_4	0	0	0	2				
M_5								
M_6								
M_7								

[1]Idea based on lecture Einführung in die WI, Rinderle-Ma S., Winter K., TUM 2022/23

Petri Nets – Reachability Analysis Table

We can use a table to perform the reachability analysis in a systematic way for the given Place-Transition net.[1]

Marking	Places				Activated Transitions			
	p_1	p_2	p_3	p_4	A	B	C	D
M_1	2	0	1	1	x			x → M_2
M_2	2	0	0	2	x → M_3			
M_3	0	1	0	1		x → M_4	x → M_5	
M_4	0	0	0	2				
M_5	0	0	1	1				x → M_4
M_6								
M_7								

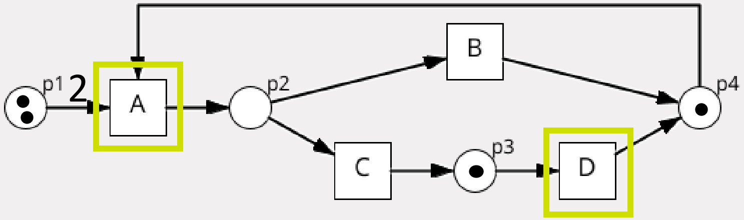


[1]Idea based on lecture Einführung in die WI, Rinderle-Ma S., Winter K., TUM 2022/23

Petri Nets – Reachability Analysis Table

We can use a table to perform the reachability analysis in a systematic way for the given Place-Transition net.[1]

Marking	Places				Activated Transitions			
	p_1	p_2	p_3	p_4	A	B	C	D
M_1	2	0	1	1	x → M_6			x → M_2
M_2	2	0	0	2	x → M_3			
M_3	0	1	0	1		x → M_4	x → M_5	
M_4	0	0	0	2				
M_5	0	0	1	1				x → M_4
M_6								
M_7								

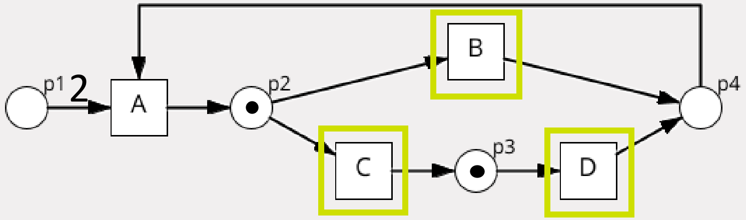


[1]Idea based on lecture Einführung in die WI, Rinderle-Ma S., Winter K., TUM 2022/23

Petri Nets – Reachability Analysis Table

We can use a table to perform the reachability analysis in a systematic way for the given Place-Transition net.[1]

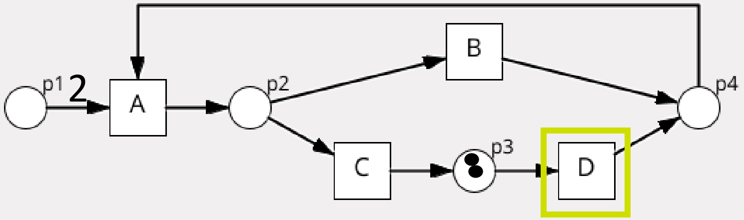
Marking	Places				Activated Transitions			
	p_1	p_2	p_3	p_4	A	B	C	D
M_1	2	0	1	1	x → M_6			x → M_2
M_2	2	0	0	2	x → M_3			
M_3	0	1	0	1		x → M_4	x → M_5	
M_4	0	0	0	2				
M_5	0	0	1	1				x → M_4
M_6	0	1	1	0		x → M_5	x → M_7	x → M_3
M_7								



[1]Idea based on lecture Einführung in die WI, Rinderle-Ma S., Winter K., TUM 2022/23

Petri Nets – Reachability Analysis Table

We can use a table to perform the reachability analysis in a systematic way for the given Place-Transition net.[1]

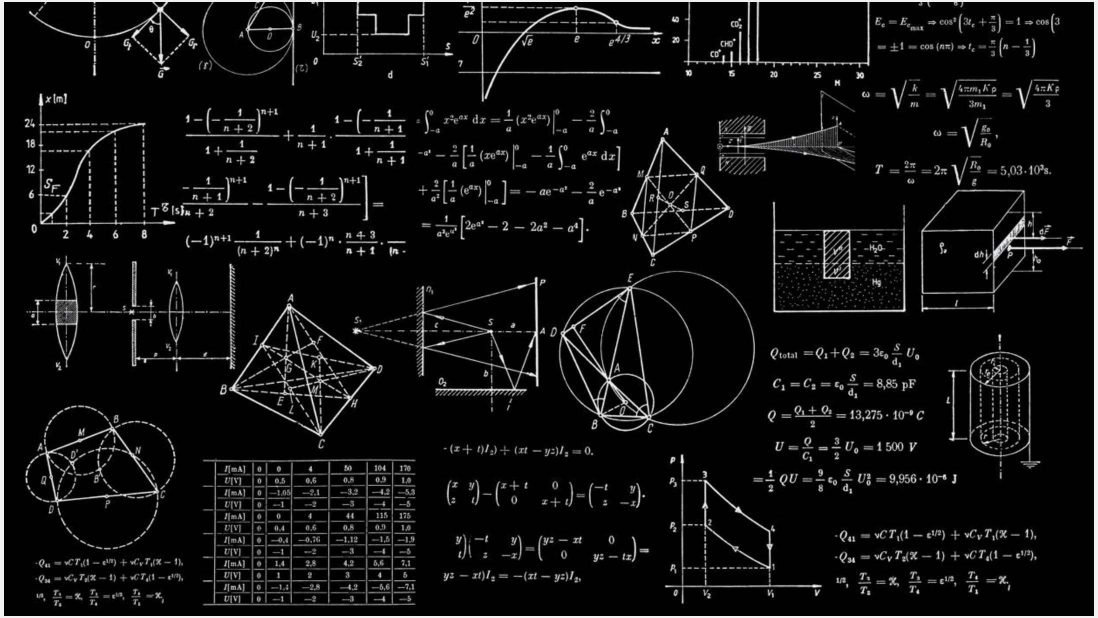


Marking	Places				Activated Transitions			
	p_1	p_2	p_3	p_4	A	B	C	D
M_1	2	0	1	1	x → M_6			x → M_2
M_2	2	0	0	2	x → M_3			
M_3	0	1	0	1		x → M_4	x → M_5	
M_4	0	0	0	2				
M_5	0	0	1	1				x → M_4
M_6	0	1	1	0		x → M_5	x → M_7	x → M_3
M_7	0	0	2	0				x → M_5

Note that depending on which transition fires first in your analysis, the rows can be in a different order and marking M_n could be labeled differently as M_m.
It is sufficient to write down each marking once.

[1]Idea based on lecture Einführung in die WI, Rinderle-Ma S., Winter K., TUM 2022/23

Take a Break from the Formal Stuff

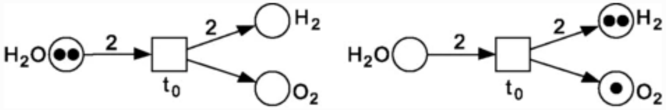


Picture by Dan Cristian Padure via Unsplash

Petri Nets – General Relevance

- Introduced by Carl Adam Petri
- Describe distributed systems
- Not only relevant for business process modeling but also in systems biology, chemistry, logistics, manufacturing, … .

Fig. 4.1



An example network modeling electrolysis of water, that is, $2\ H_2O \rightarrow 2\ H_2 + O_2$. The net shown on the *left* (on the *right*) illustrates a marking before (after) firing of transition t_0

Sackmann, A. (2011). Discrete Modeling. In: Koch, I., Reisig, W., Schreiber, F. (eds) Modeling in Systems Biology. Computational Biology, vol 16. Springer, London. https://doi.org/10.1007/978-1-84996-474-6_4

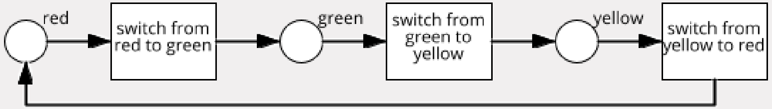
Conference Series on Petri nets: https://link.springer.com/conference/apn

Petri Nets – Relevance for Simulation

Petri nets have several advantages

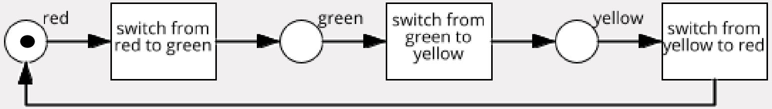
- only three/four modeling elements, i.e., easy to understand
- reachability analysis helps to identify (undesired) process behavior like
  - deadlocks, i.e., does the process end and if so, does it end the way I want it to end?
  - (not) reachable states, i.e., can I reach a certain point in the process?

Example traffic light:
What are the places/transitions? How should they be connected to guarantee the desired switching between the states?



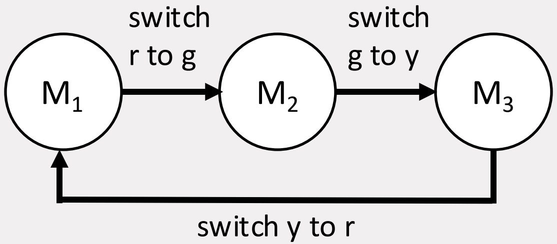
Picture by Pawel Czerwinski via Unsplash

Petri Nets – Reachability Graph



visual representation

reachability graph



reachability analysis

Marking	Places			Activated Transitions		
	red	green	yellow	switch r to g	switch g to y	switch y to r
M_1	1	0	0	x → M_2		
M_2	0	1	0		x → M_3	
M_3	0	0	1			x → M_1

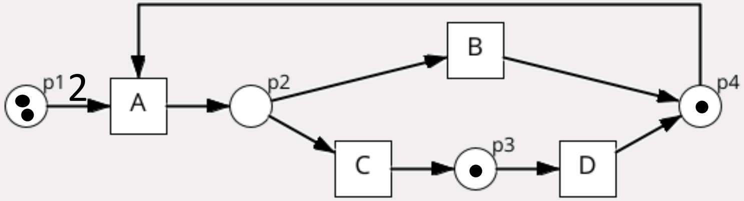
All states are reachable from all other states. The Petri net does not contain any deadlocks.

Petri Nets – Reachability Graph Exercise

<u>5-Minutes-Exercise:</u>

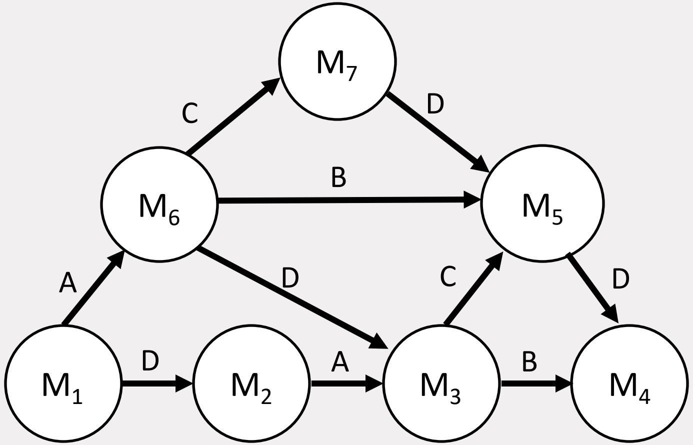
Draw the reachability graph for the given Place-Transition net using the previous reachability analysis results. Discuss the results with your neighbour.

Marking	Places				Activated Transitions			
	p_1	p_2	p_3	p_4	A	B	C	D
M_1	2	0	1	1	x → M_6			x → M_2
M_2	2	0	0	2	x → M_3			
M_3	0	1	0	1		x → M_4	x → M_5	
M_4	0	0	0	2				
M_5	0	0	1	1				x → M_4
M_6	0	1	1	0		x → M_5	x → M_7	x → M_3
M_7	0	0	2	0				x → M_5



Picture by Pexels via Pixabay

Petri Nets – Reachability Graph Exercise Solution



Marking	Places				Activated Transitions			
	p_1	p_2	p_3	p_4	A	B	C	D
M_1	2	0	1	1	$x \to M_6$			$x \to M_2$
M_2	2	0	0	2	$x \to M_3$			
M_3	0	1	0	1		$x \to M_4$	$x \to M_5$	
M_4	0	0	0	2				
M_5	0	0	1	1				$x \to M_4$
M_6	0	1	1	0		$x \to M_5$	$x \to M_7$	$x \to M_3$
M_7	0	0	2	0				$x \to M_5$

Petri Nets – Properties

k-bounded: A marked Petri net is k-bounded if no place ever holds more than k tokens. It is bounded if and only if there exists a $k \in \mathbb{N}$ such that it is k-bounded.

safe: A marked Petri net is safe if and only if it is 1-bounded.

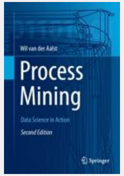
deadlock free: A marked Petri net is deadlock free if at every reachable marking at least one transition is enabled

liveness: A transition t in a marked Petri net is live if from every reachable marking it is possible to enable t. A marked Petri net is live if each of its transitions is live.

reversibility: For all reachable markings it holds that the initial marking can be reached again.

How can you recognize those properties based on the reachability table and/or reachability graph?

Further reading: Chapter 3.3 Model-Based Process Analysis

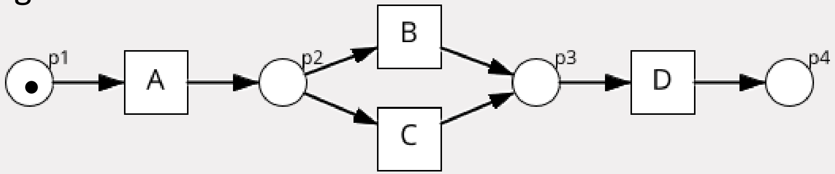


Petri Nets – Connection to Event Logs

www.menti.com
Code: *14219820*

Which transitions have fired in the Petri net is stored in event logs.

Event logs contain all observed traces, i.e., firing sequences.

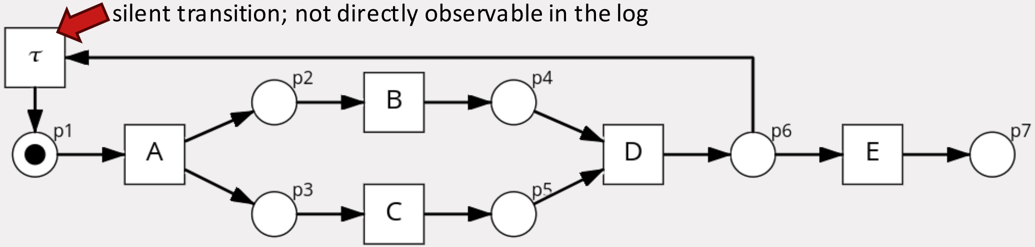


Example of a trace for the first Petri net: <A,B,D>

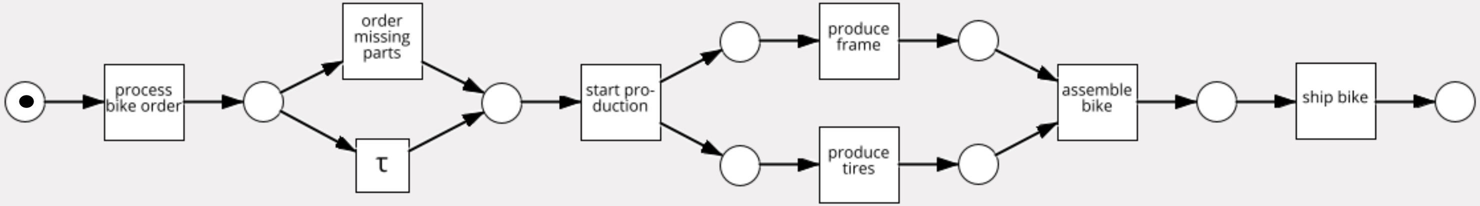
What would be another observable trace for the first Petri net?

Can you provide a log containing all observable traces for the second Petri net?

Note: Petri nets can be structurally different but trace equivalent.



Example Run of a Petri Net with Resulting Event Log



Trace 1:

<process bike order, order missing parts, start production, produce frame, produce tires, assemble bike, ship bike>

Trace 2:

<process bike order, start production, produce frame, produce tires, assemble bike, ship bike>

Trace 3:

<process bike order, start production, produce frame, assemble bike, produce tires, ship bike>

Event log consists of all the observed traces: L = [Trace1, Trace2, Trace3]

Workflow Nets

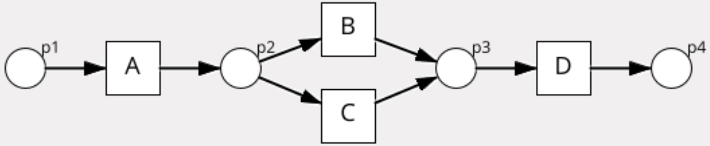
www.menti.com
Code: *14219820*

Workflow nets are (labeled) Petri nets with

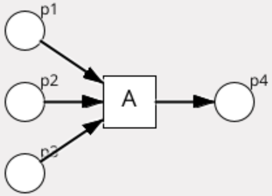
- a distinguished start place *s (source place)* that has no incoming arcs
- a distinguished end place *e (sink place)* that has no outgoing arcs
- every place and every transition being on a direct path between *s* and *e*

Which of the following Petri nets is/are a Workflow net(s) and why?

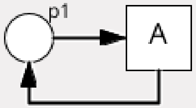
1)



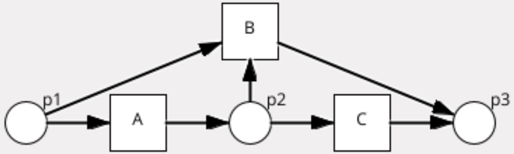
3)



2)



4)



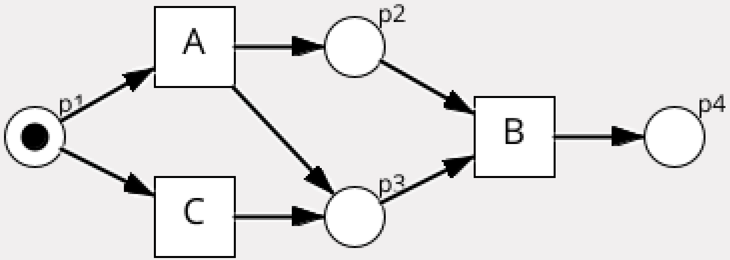
Workflow Nets – Soundness

www.menti.com
Code: *14219820*

A Workflow net is sound if and only if the following holds

- safeness: places cannot hold multiple tokens at the same time
- proper completion: if the sink place is marked, all other places are empty
- option to complete: it is always possible to reach the marking that marks just the sink place
- absence of dead parts: for any transition there is a firing sequence enabling it.

Is the given Petri net a sound Workflow net?



Business Process Simulation

Lecture 2b – UML class diagram

Laura Genga

Department of Industrial Engineering and Innovation Sciences, Information Systems Group

What is UML?

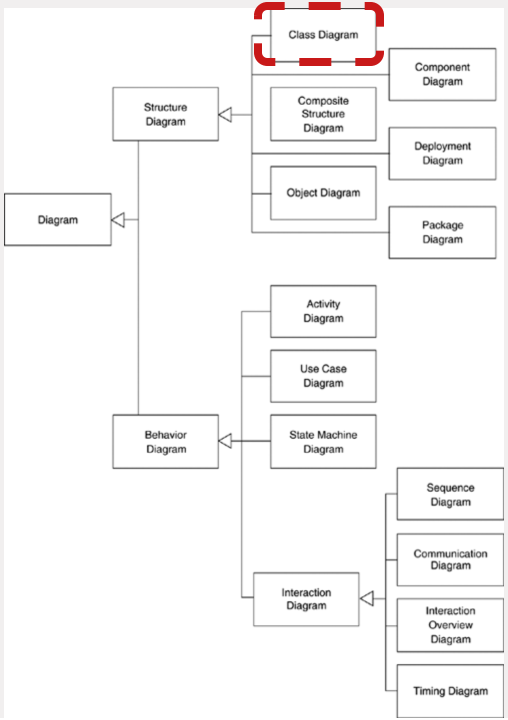
The Unified Modeling Language is a *family* of graphical notations to describe and design (Object-oriented) software systems

Released in 1977, managed by the Object Management GROUP

Main uses:

- A a **sketch** to describe (parts of) the system
- A **blueprint** to follow when implementing the system
- As a **programming language**

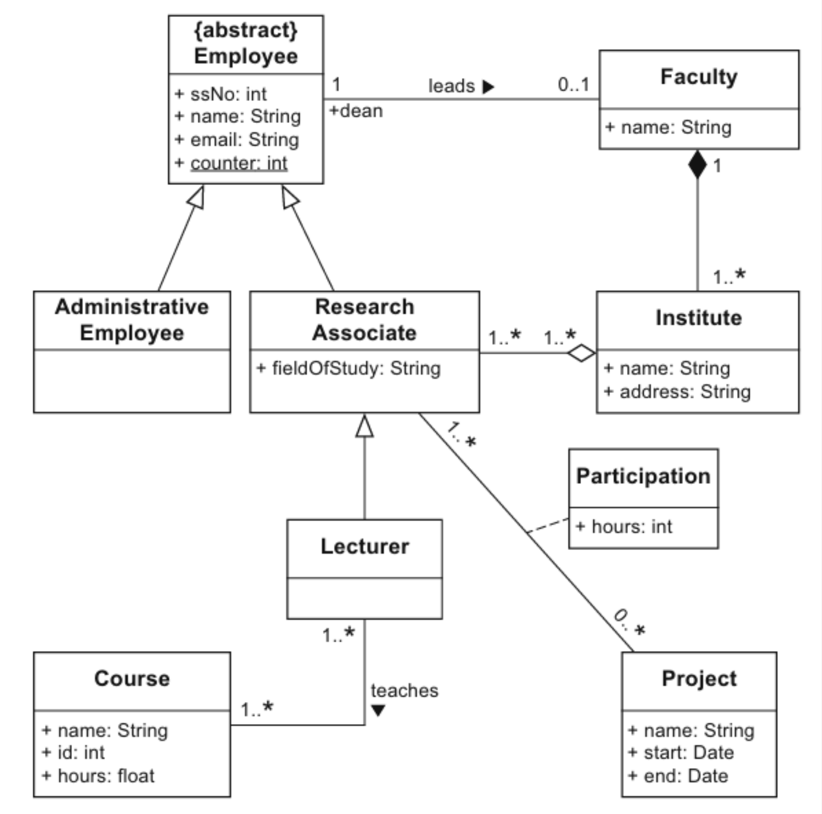
Different UML Diagrams



Class Diagram

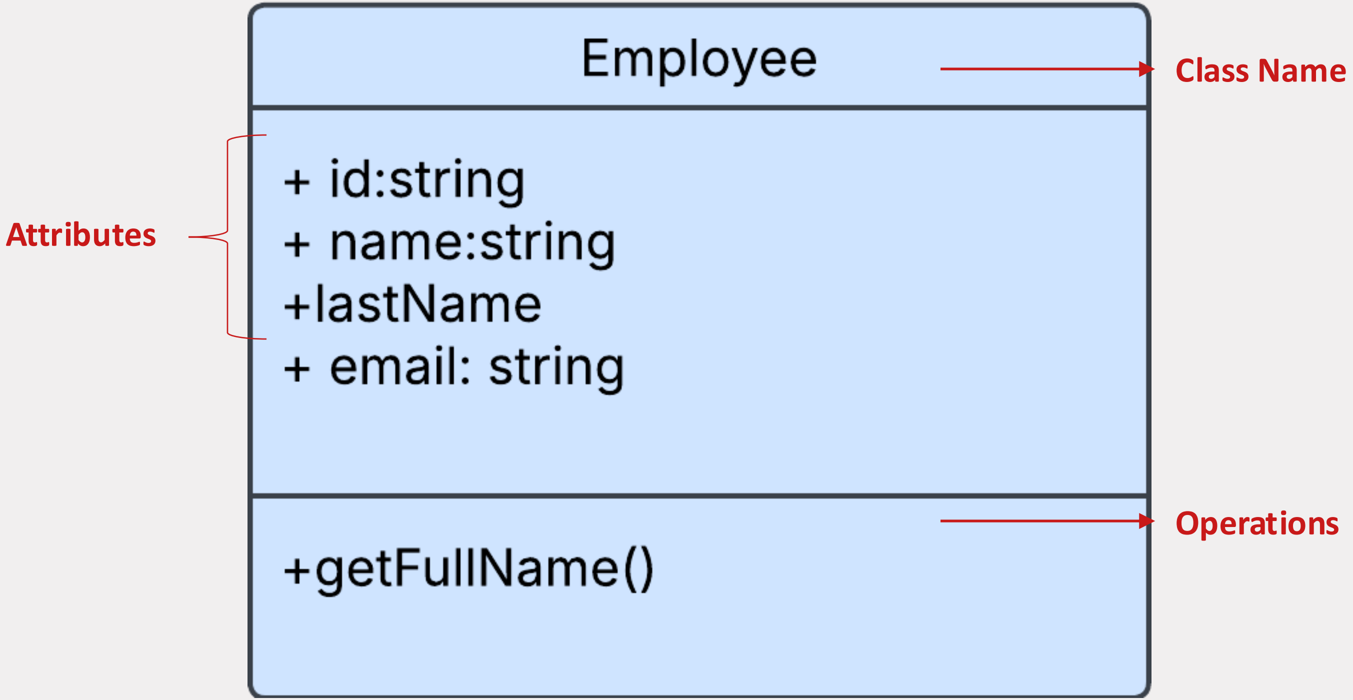
A class diagram describes:

- the **types of objects** in the system
- the kinds of static **relationships** that exist among them
- the **properties** and **operations** of a class
- the **constraints** that apply to the way objects are connected

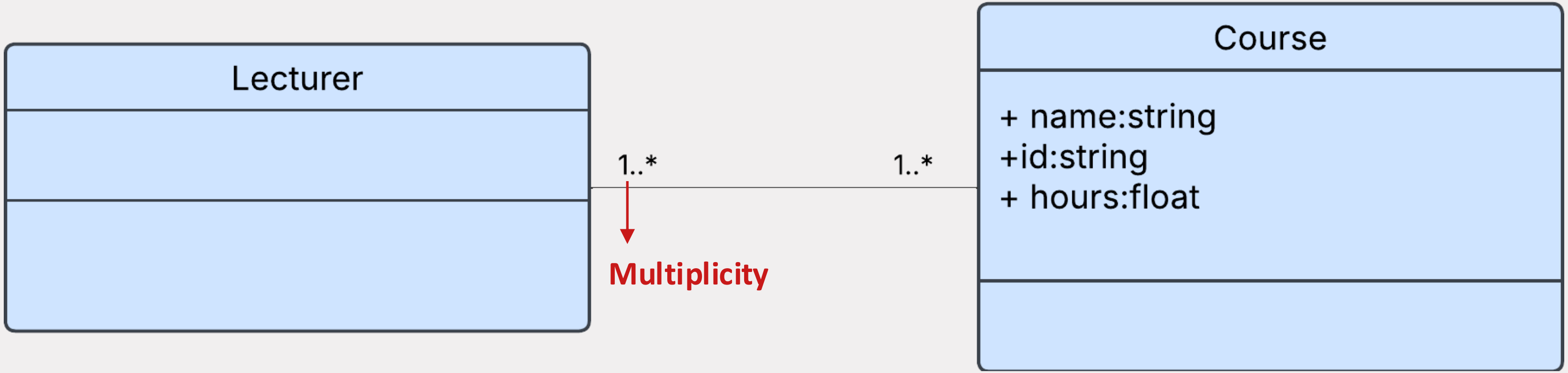


{abstract}
Employee
+ ssNo: int
+ name: String
+ email: String
+ counter: int
1
+dean
leads ▶
0..1
Faculty
+ name: String
1
1..*
Administrative Employee
Research Associate
+ fieldOfStudy: String
1..*
1..*
Institute
+ name: String
+ address: String
1..*
Participation
+ hours: int
Lecturer
0..*
1..*
teaches
Course
+ name: String
+ id: int
+ hours: float
1..*
Project
+ name: String
+ start: Date
+ end: Date

Class

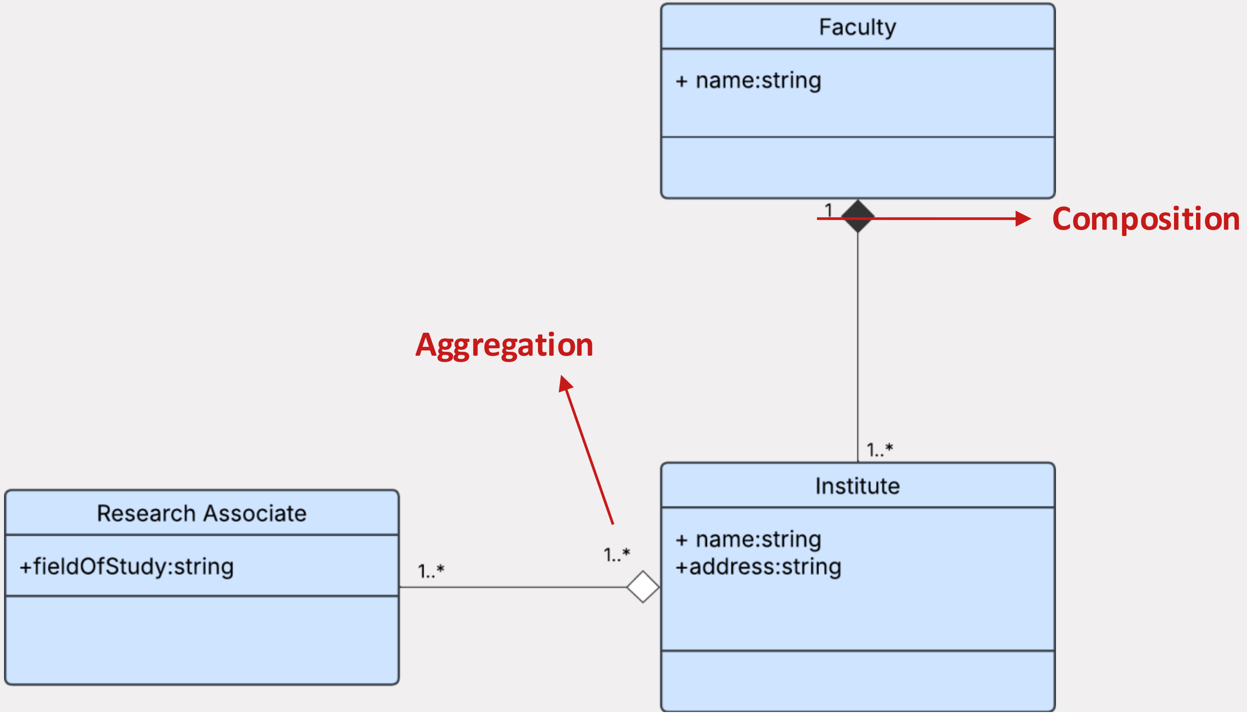

Employee
+ id:string
+ name:string
+lastName
+ email: string
+getFullName()
Class Name
Attributes
Operations

Associations

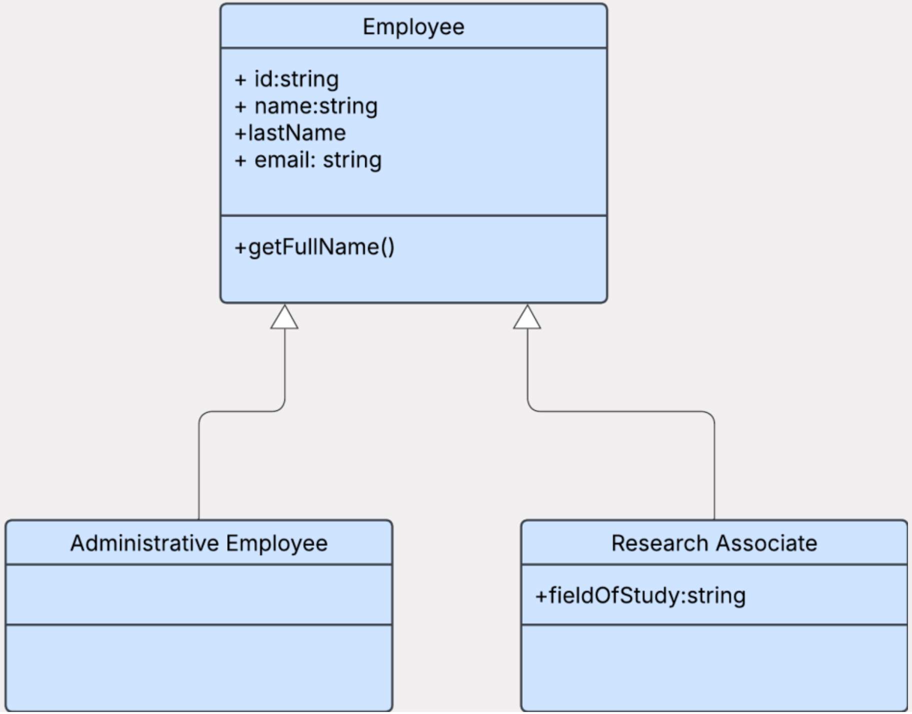


Lecturer.course, Course.Lecturer: Link-attributes

Association types



Generalization



Summary of today's lecture

Petri nets

- To represent process behaviors

UML class diagrams

- To represent entities and their relations