HOCHSCHULE FÜR TECHNIK UND WIRTSCHAFT DRESDEN
UNIVERSITY OF APPLIED SCIENCES

ALADIN goes OPAL (OPALADIN)

T**a**sks and so**l**ution(-hint)s gener**a**tor in the fiel**d** of **I**nformatics and adjacent discipli**n**es goes OPAL

Outline

- 1. Motivation for the development of (OP)ALADIN
- 2. Objectives of (OP)ALADIN
- 3. Generation of semantically plausible exercises and exercise types in (OP)ALADIN
- 4. Learning management and didactics in (OP)ALADIN
- 5. Summary and outlook

Outline

- **1. Motivation for the development of (OP)ALADIN**
- 2. Objectives of (OP)ALADIN
- 3. Generation of semantically plausible exercises and exercise types in (OP)ALADIN
- 4. Learning management and didactics in (OP)ALADIN
- 5. Summary and outlook

1. Motivation for the development of (OP)ALADIN

- Low availability of…
  - …exercises and mock exams
  - …unknown exercises for independent practice
- No motivating impulses for individual learning processes
- Teaching process is tied to location and time slots
- No individualization of exercises in terms of complexity
- No self-organized and self-activated learning

- High expenditure regarding the…
  - …creation of new exercies
  - …creation of new exercise types
  - …correction of…
    - …tutorials
    - …laboratory practicals
    - …exams
  - …creation of solution hints
- Individual tutoring requires synchronous communication

Outline

1. Motivation for the development of (OP)ALADIN
2. **Objectives of (OP)ALADIN**
3. Generation of semantically plausible exercises and exercise types in (OP)ALADIN
4. Learning management and didactics in (OP)ALADIN
5. Summary and outlook

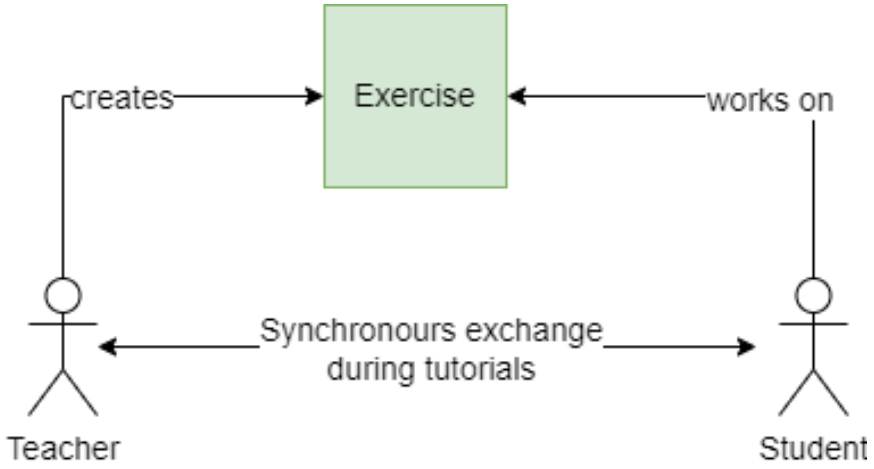
2. Objectives of (OP)ALADIN

- Adaption of exercise complexity to individual performance capability
- Learn in their own speed
- Promoting of high problem-solving competence
  ➔ higher success rates
- Generation of exercises is parametrizable
  ➔ Teaching content can be actively co-designed
- Promoting student networking
- Asynchronous feedback to/from teachers

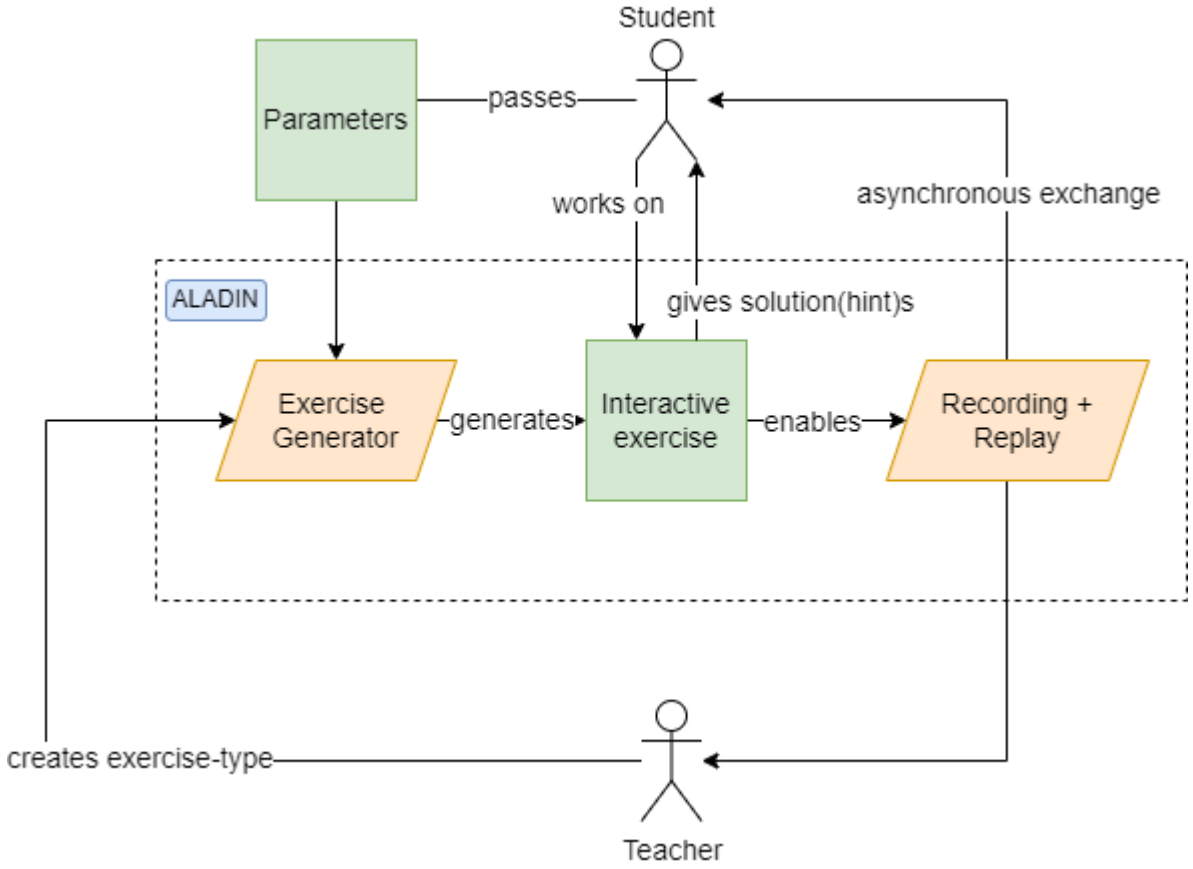
- Generation of competence-oriented exercises for heterogeneous target groups
- Generation of online self-tests and electronic mock exams
  ➔ Immediate automatic and performance-based feedback
- Reusability is unrestricted by time and subject matter
- Flexible applicability in terms of time, space and institution
- Extensible by new exercise types
- Reduction of effort with regard to generating exercises and solution aids and their correction

2.1 Learning and teaching with and without ALADIN

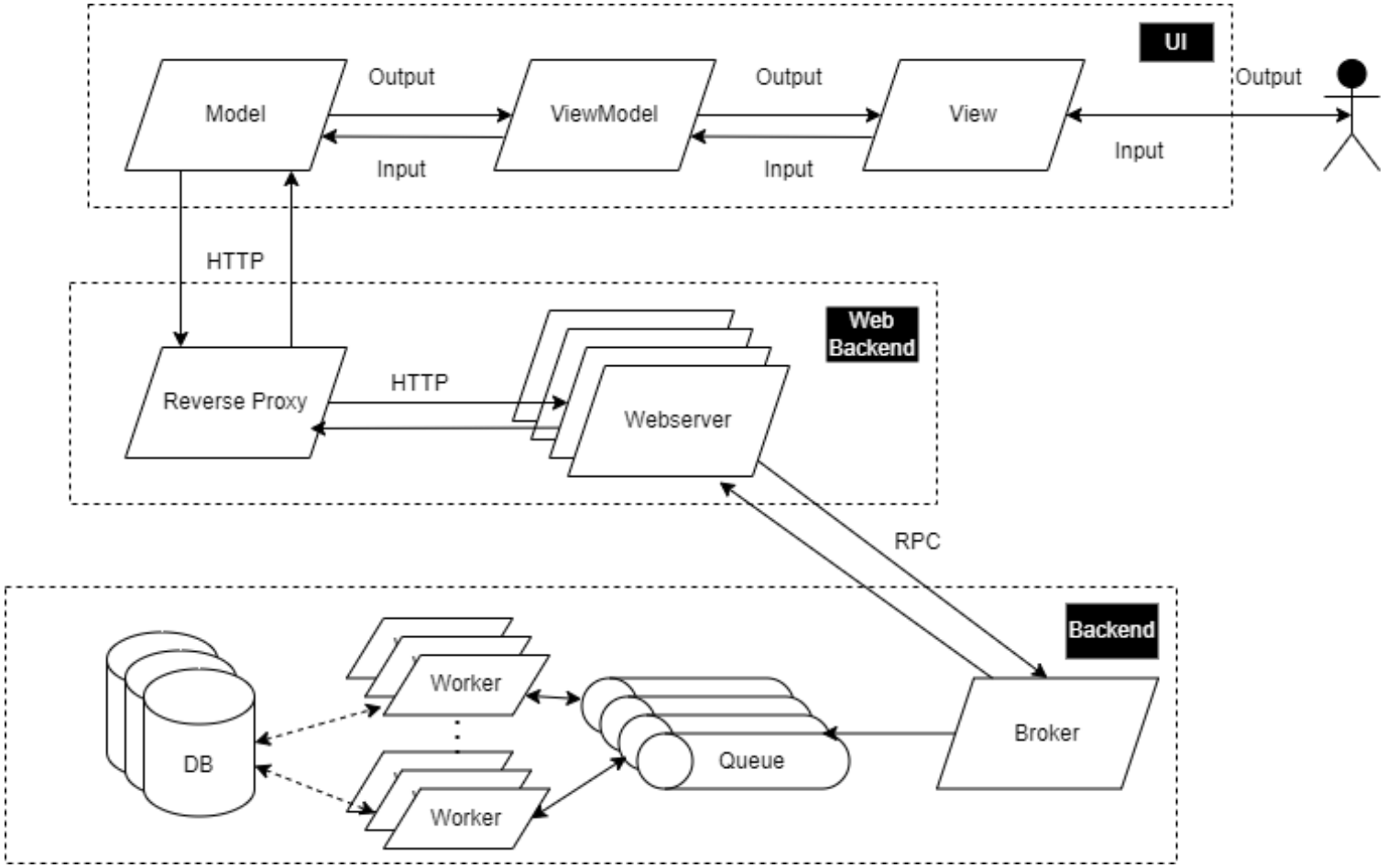
Teaching process without ALADIN



Teaching process with ALADIN



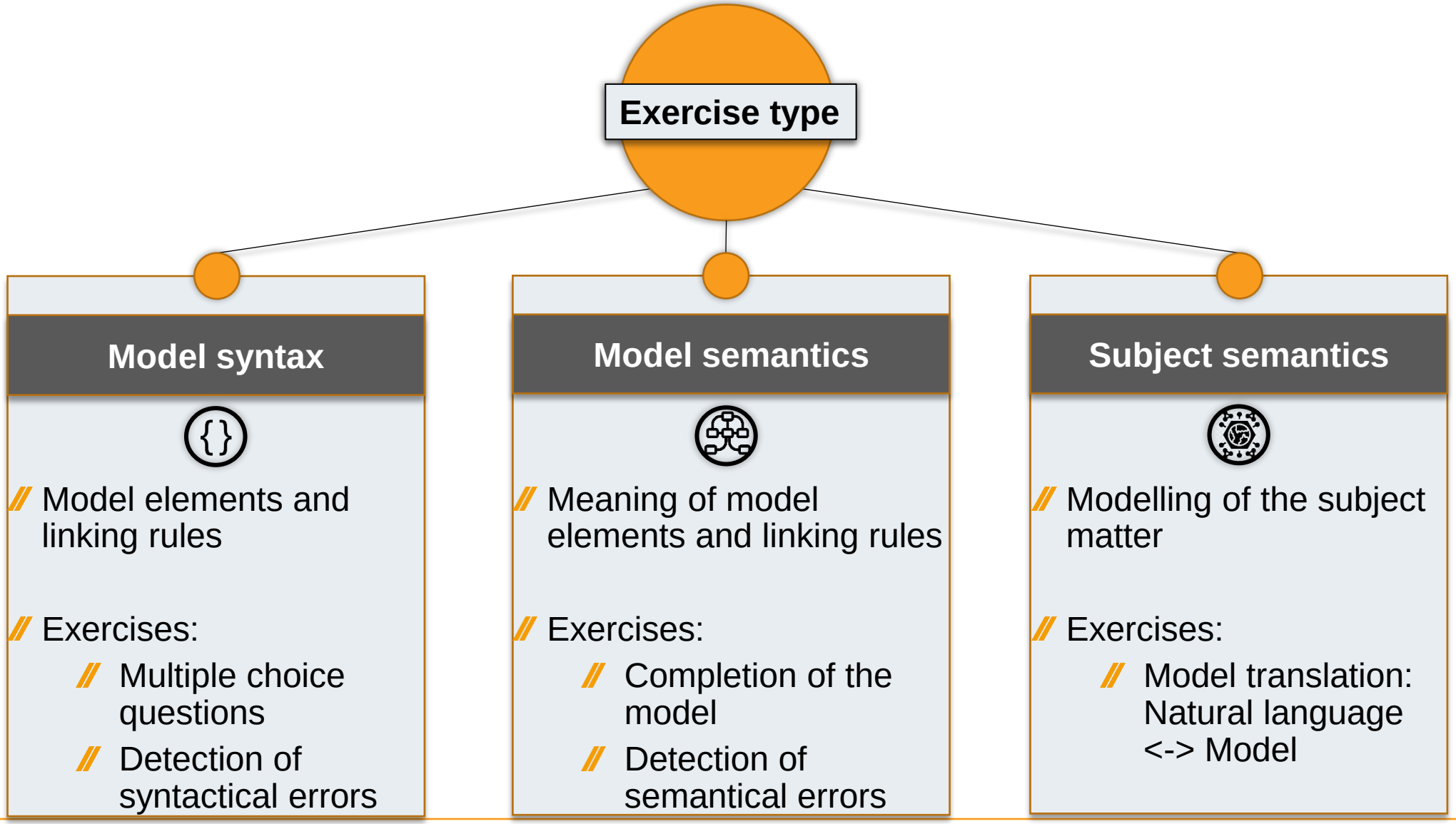
2.2 Software architecture of ALADIN



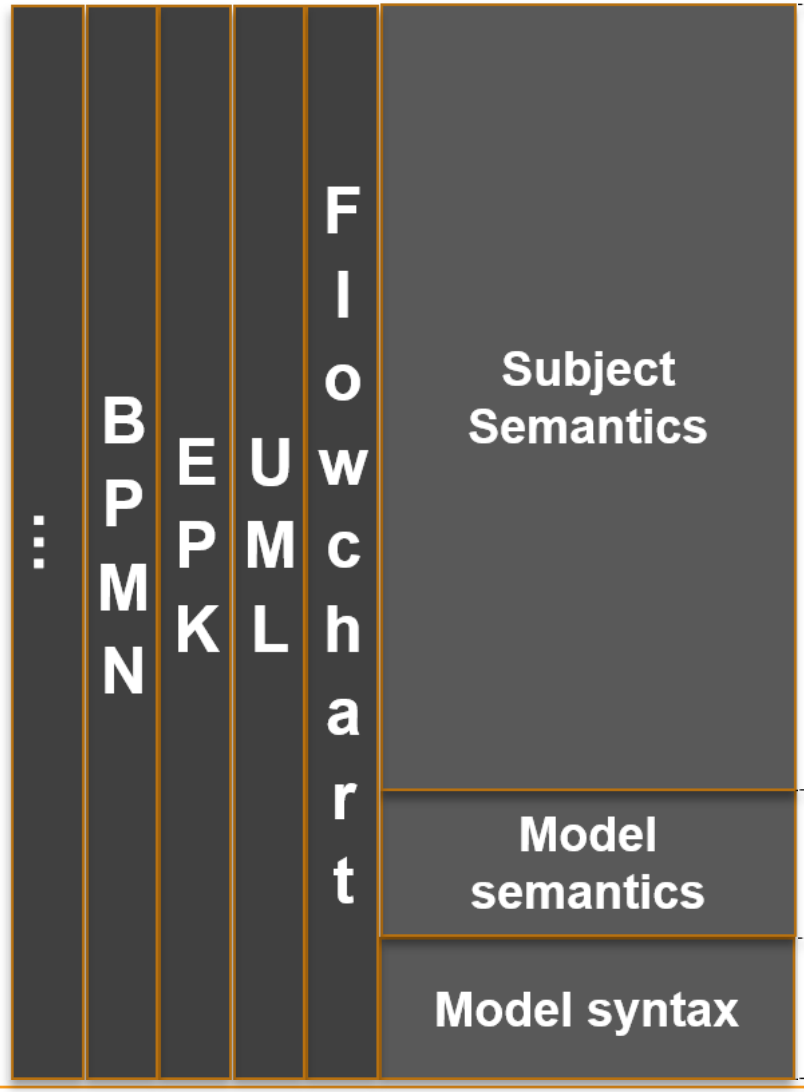
Outline

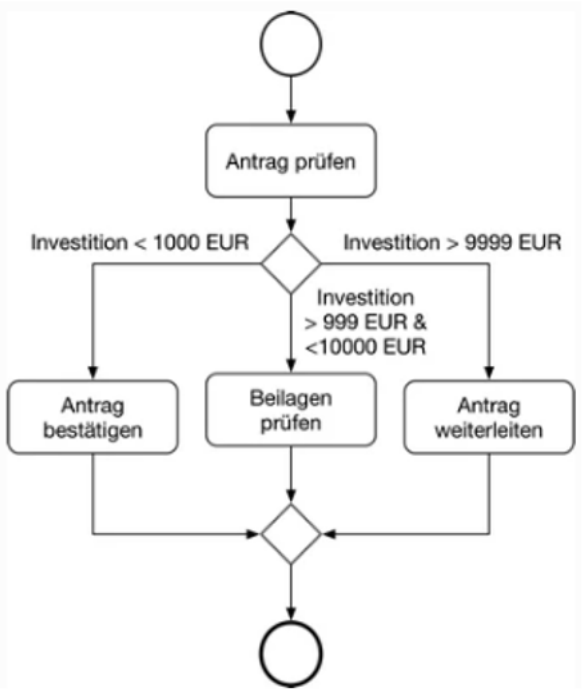
- 1. Motivation for the development of (OP)ALADIN
- 2. Objectives of (OP)ALADIN
- **3. Generation of semantically plausible exercises and exercise types in (OP)ALADIN**
- 4. Learning management and didactics in (OP)ALADIN
- 5. Summary and outlook

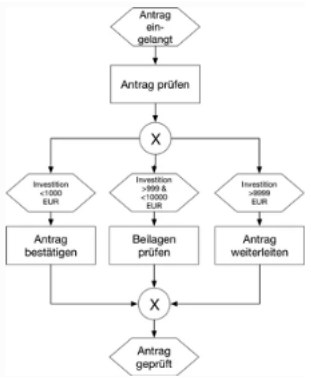
3.1 Exercise types for modeling exercises in (OP)ALADIN

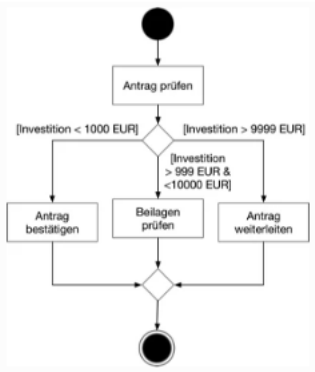


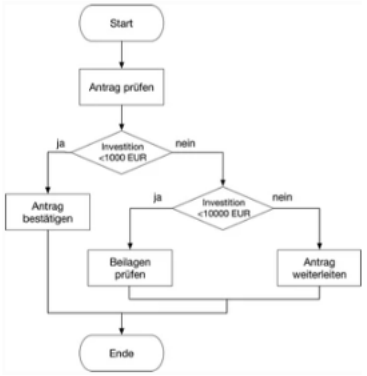
3.2 Generalisation of the exercise generation for behavioral diagrams

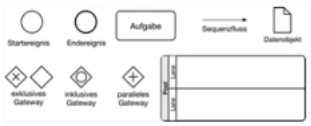


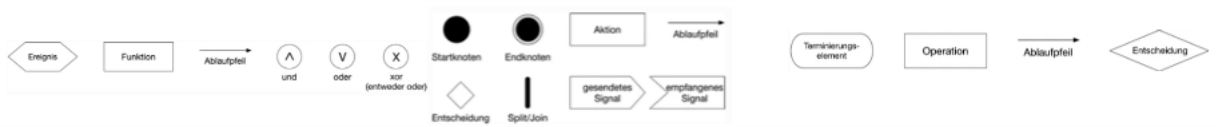


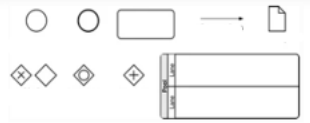




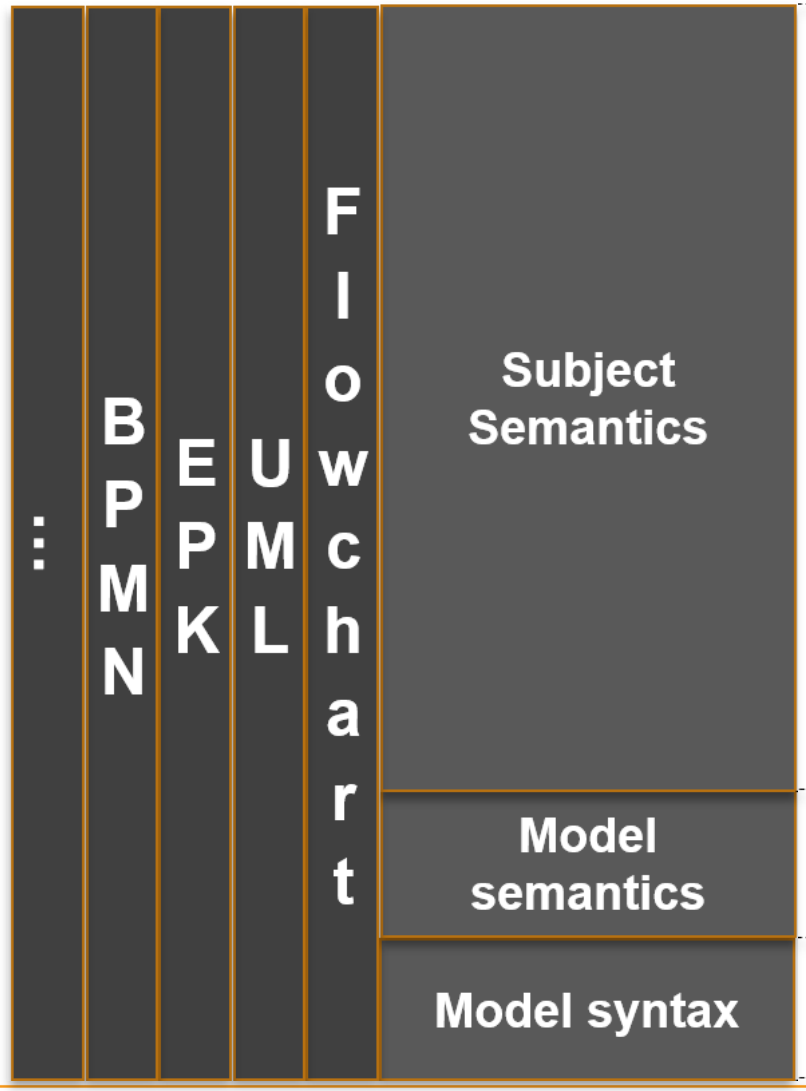






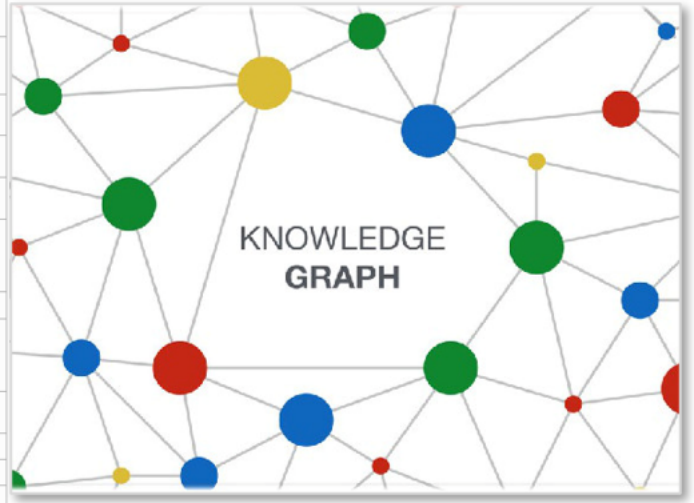


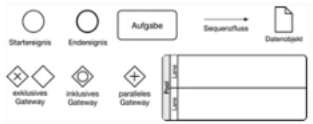
3.2 Generalisation of the exercise generation for behavioral diagrams

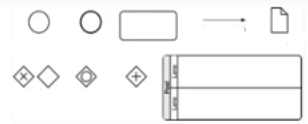


Definitionsteil	Konzept	Flowchart	eEPK	UML Aktivitäts-diagramm	BPMN	S-BPMN
1a	Anfang	Terminierungselement	Ereignis	Startknoten	Startereignis	Startzustand
	Input	–	Informationsobjekt	Datenobjekt/ empfangenes Signal	Datenobjekt, Nachricht	Geschäftsobjekt, Nachricht
1b	Ende	Terminierungselement	Ereignis	Endknoten	Endereignis	Endzustand
	Ergebnis	–	Informationsobjekt	Datenobjekt/_gesendetes Signal	Datenobjekt, Nachricht	Geschäftsobjekt, Nachricht
1c	Kundenbedürfnis	–	–	–	–	–
2a	Aktivitäten/Aufgaben	Operation				Funktionszustand
2b	Startereignis	Nur unspezifisch				Startzustand, meist via Empfang einer Nachricht
	HandelInder	–	Org. Einheiten	Partition	Pool, Lane	Subjekt
2c	Sachlogische/ zeitliche Reihenfolge	Ablaufpfeil & Entscheidung	Ablaufpfeil & Konnektoren für alternative und parallele Abläufe, Datenfluss	Ablaufpfeil, Entscheidung, Split/Join	Sequenzfluss, Gateways für alternative und parallele Abläufe, Nachrichtenfluss, Ausnahmebe-handlung, Transaktionen, Choreografie	Nachrichten zwischen Subjekten, Bedingungen für Zustandsübergänge in Subjektverhalten
2d	Geschäftsobjekt	–	Informationsobjekt	Datenobjekt	Datenobjekt, Nachricht	Geschäftsobjekt, Nachricht
3a	Mensch	–	Org. Einheiten	Partition	Pool, Lane, Benutzer-Task, Manueller Task	Subjekt
	Maschine	–	–	-	Service-Task	Subjekt
3b	Sachmittel	–	–	-	–	–
	Information	–	Informationsobjekt	Datenobjekt	Datenobjekt, Nachricht, Datenspeicher	Geschäftsobjekt, Nachricht
	Anwendungsprogramm	–	Anwendunssystem	–	Service-Task	Subjekt
	Hilfsmittel (allgemein)	–	–	–	–	–

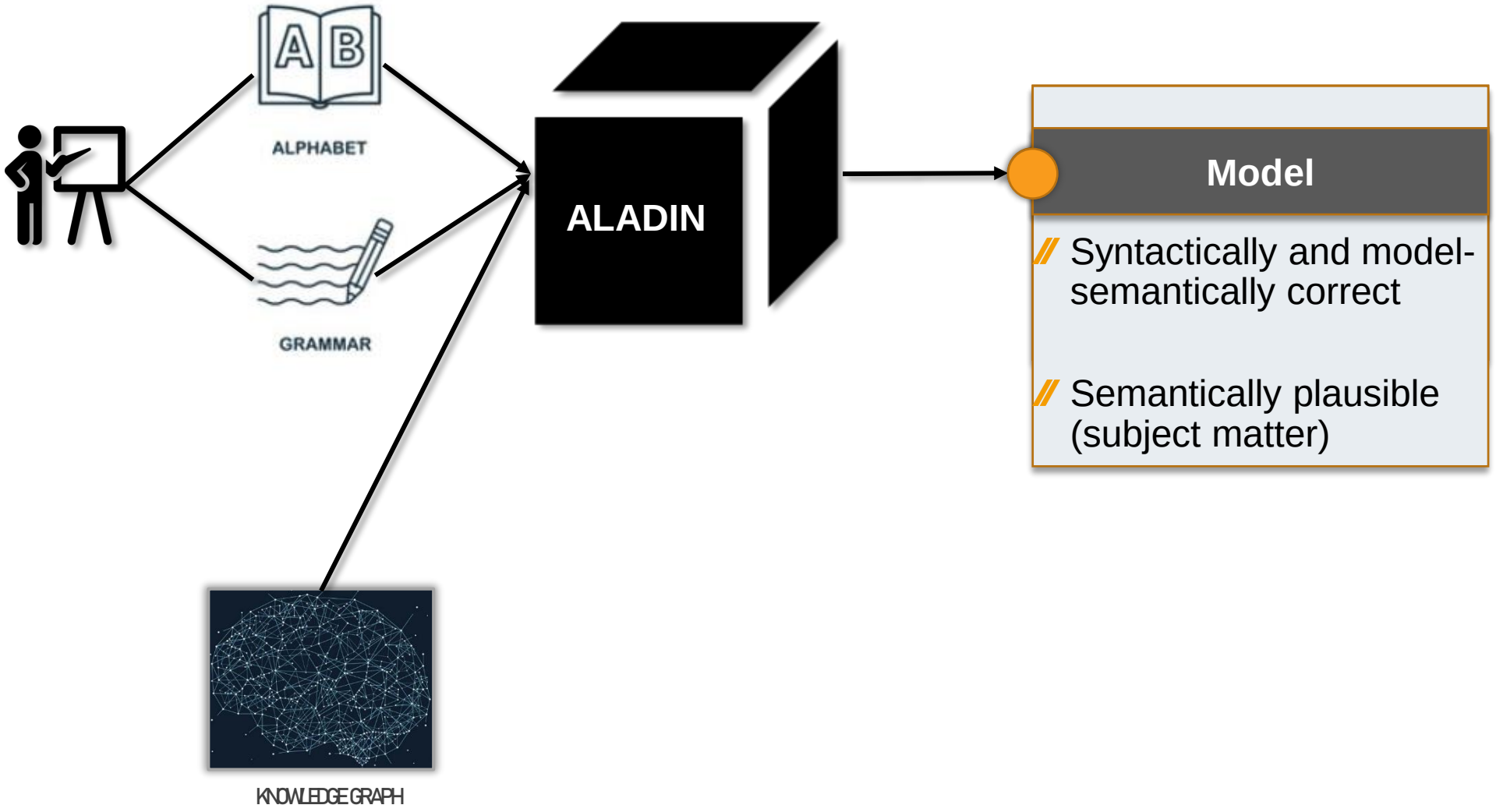
Meta-model







3.4 Implementation of modeling exercise generators in (OP)ALADIN

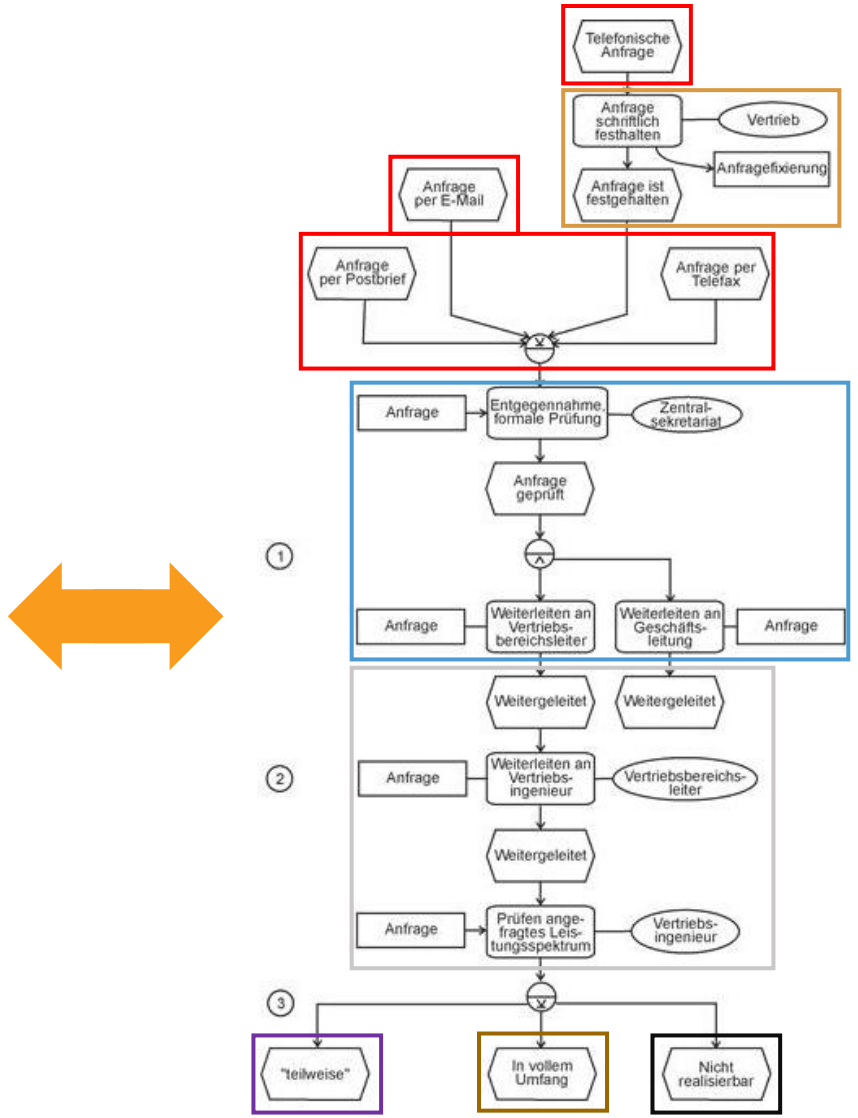


3.5 Requirements for a meaningful EPC modeling exercise

Exercise definition:
Model an event-driven process chain that represents the business process described in the text.

Natural language description:

The business process is started on the basis of a customer inquiry. This can be received by mail, fax, e-mail or telephone. If it is received by phone, the sales department prepares a written document that records the request (request fixation).
Regardless of the type of receipt, the central secretariat receives the request, checks it for formal correctness and then forwards it to the responsible sales area manager and the management.
The sales area manager passes the request on to one of his sales engineers, who checks the feasibility of the offer. This initially concerns the requested range of services. If this is not within the company's range of services, a rejection will be made. If only parts of the requested services can be provided, partner companies are asked for support and involved either as cooperation partners or subcontractors. If no suitable partners are found, the request is rejected.



3.6.1 Requirements for a meaningful SQL-query exercise

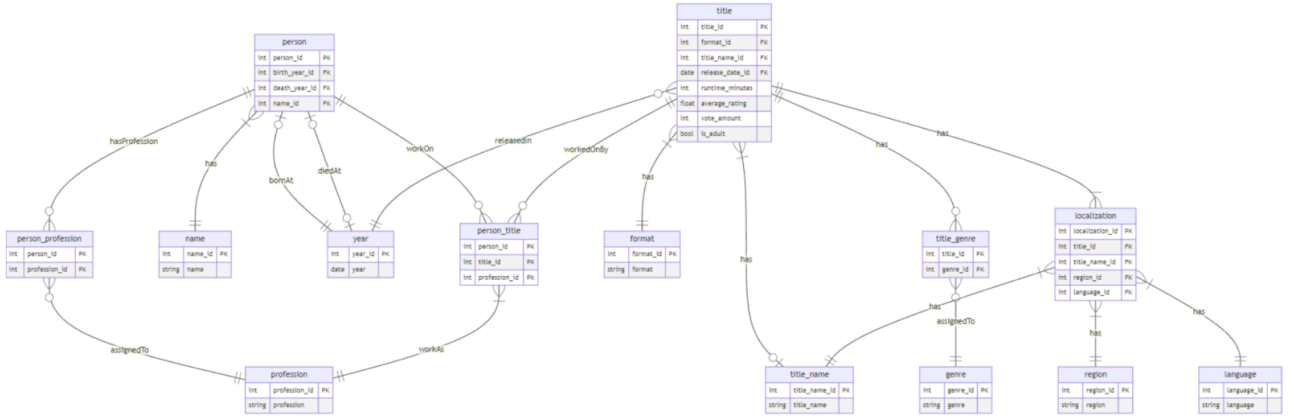
Exercise definition:

Write an SQL query that extracts the information described in the text from the database shown below.

Natural language description:

Find the count of title-ids, the runtime, the number of votes and the language-ids for all persons and the titles they were involved with, if the region contains ‘ge’ and the count of title_id is not '4883174'. Sort the result by the person-id in ascending order.

Database schema:




```
SELECT COUNT(t.title_id), t.runtime_minutes, t.vote_amount, lo.
    language_id
FROM imdb2.person as p
INNER JOIN imdb2.year as y
ON p.birthyear_id = y.year_id
INNER JOIN imdb2.title as t
ON y.year_id = t.release_date_id
INNER JOIN imdb2.localization as lo
ON t.title_id = lo.title_i
INNER JOIN imdb2.region as r
ON lo.region_id = r.region_id
WHERE r.region LIKE 'ge%'
GROUP BY t.runtime_minutes, t.vote_amount, lo.language_id
HAVING COUNT(t.title_id) <> '4883174'
ORDER BY p.person_id ASC;
```

Zeig mir die Lösung | Kopieren! | Ausführen!

```
1 SELECT t.region_id, t.territory_id
2 FROM northwind.employee_territories as et
3 CROSS JOIN northwind.territories as t
4 WHERE t.region_id = '1'
5 GROUP BY t.region_id, t.territory_id;
```

Digression - Limitations on semantic level 1.1

Patient

ID	Surname	Name	Date of birth	Gender
0	Mustermann	Max	01.01.2000	m
1	Decker	Dirk	31.12.1999	m
2	Räubertochter	Ronja	03.02.1952	f
3	Lustig	Lea	04.05.1965	f

PatientCondition

ID	PatientID	Status	Entry date
0	0	Recovered	14.04.2020
1	1	Vaccinated	01.06.2021
2	2	Vaccinated	21.08.2021
3	3	Infected	05.12.2020
4	1	Infected	01.01.2022

- Which patients were infected, despite being vaccinated?
- SQL-Abfrage:

```
SELECT p.Surname, p.Name FROM Patient AS p
JOIN PatientCondition AS pz ON p.ID = pz. PatientID
WHERE pz.Status = ‘Infected‘
AND pz.PatientID IN
        (SELECT PatientID FROM PatientenCondition
        WHERE Status = ‘Vaccinated‘ AND „Entry date“ < pz.„Entry date“);
```

Result

Surname	Name
Decker	Dirk

Digression - Limitations on semantic level 1.2

Create the intersection that contains the corresponding entries of the two tables "Patient" and "Patient condition". Output the columns "Name" and "First name". Only data for which "State" is "Infected" and the values of "ID" are in a subset, for which "State" is "Vaccinated" and "Entry date" is smaller than "Entry date" of the superset are to be output.

```
SELECT p.Name, p.Vorname FROM Patient AS p
JOIN Patientenzustand AS pz ON p.ID = pz.PatientenID
WHERE pz.Status = ‘Infiziert‘
AND pz.PatientenID IN
        (SELECT PatientenID FROM Patientenzustand
        WHERE Status = ‘Geimpft‘
        AND Erfassungsdatum < pz.Erfassungsdatum);
```

Find the surname and first name of the patients who have the status 'Vaccinated' and subsequently the status 'Infected'.

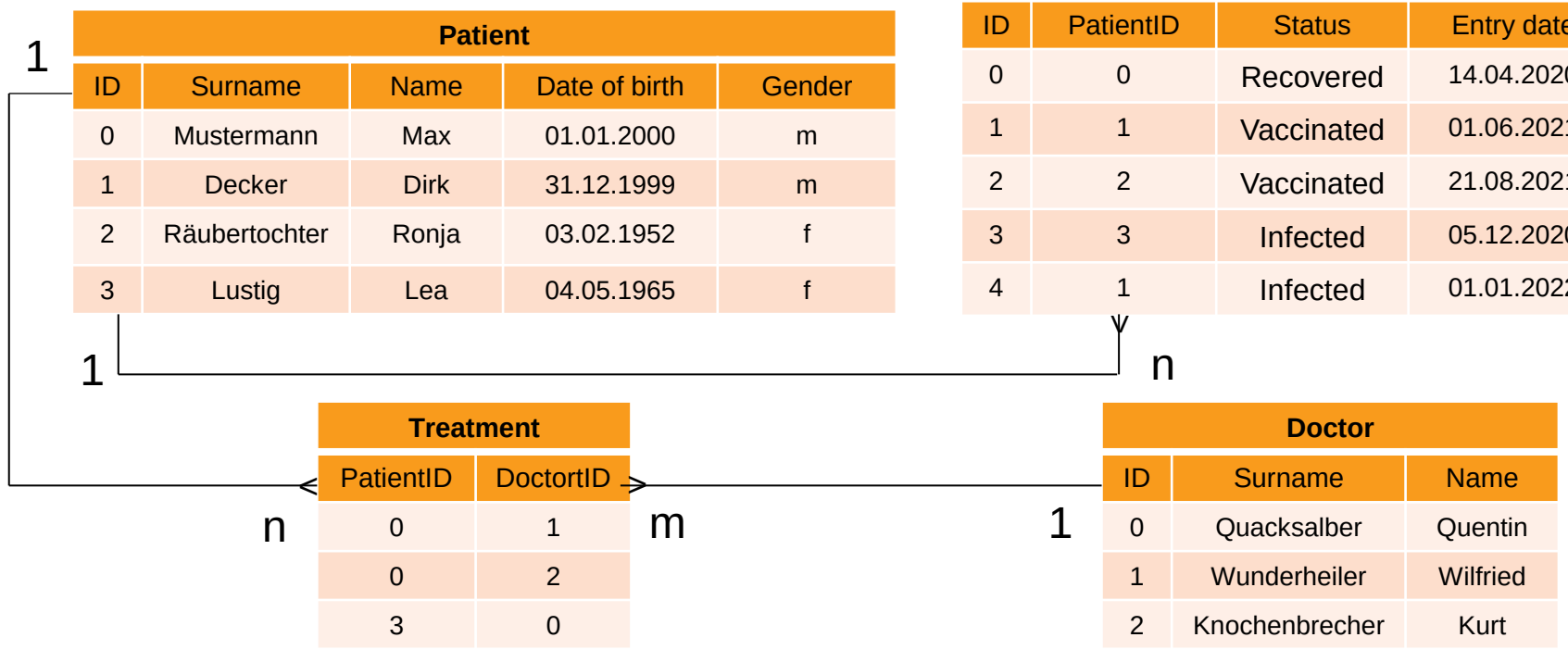
Which patients were infected despite vaccination?

- Not sufficiently inferable information
  - Cardinality
  - Semantic relation

Digression - Limitations on semantic level 1.3

Find the name and first name of the patients who have the status 'Vaccinated' and subsequently the status 'Infected' and are treated by 'Quentin Quacksalber'.

Patient	---Treatment ---	Doctor
Recipient	<-------Action--------	Acteur



Patient

ID	Surname	Name	Date of birth	Gender
0	Mustermann	Max	01.01.2000	m
1	Decker	Dirk	31.12.1999	m
2	Räubertochter	Ronja	03.02.1952	f
3	Lustig	Lea	04.05.1965	f

PatientCondition

ID	PatientID	Status	Entry date
0	0	Recovered	14.04.2020
1	1	Vaccinated	01.06.2021
2	2	Vaccinated	21.08.2021
3	3	Infected	05.12.2020
4	1	Infected	01.01.2022

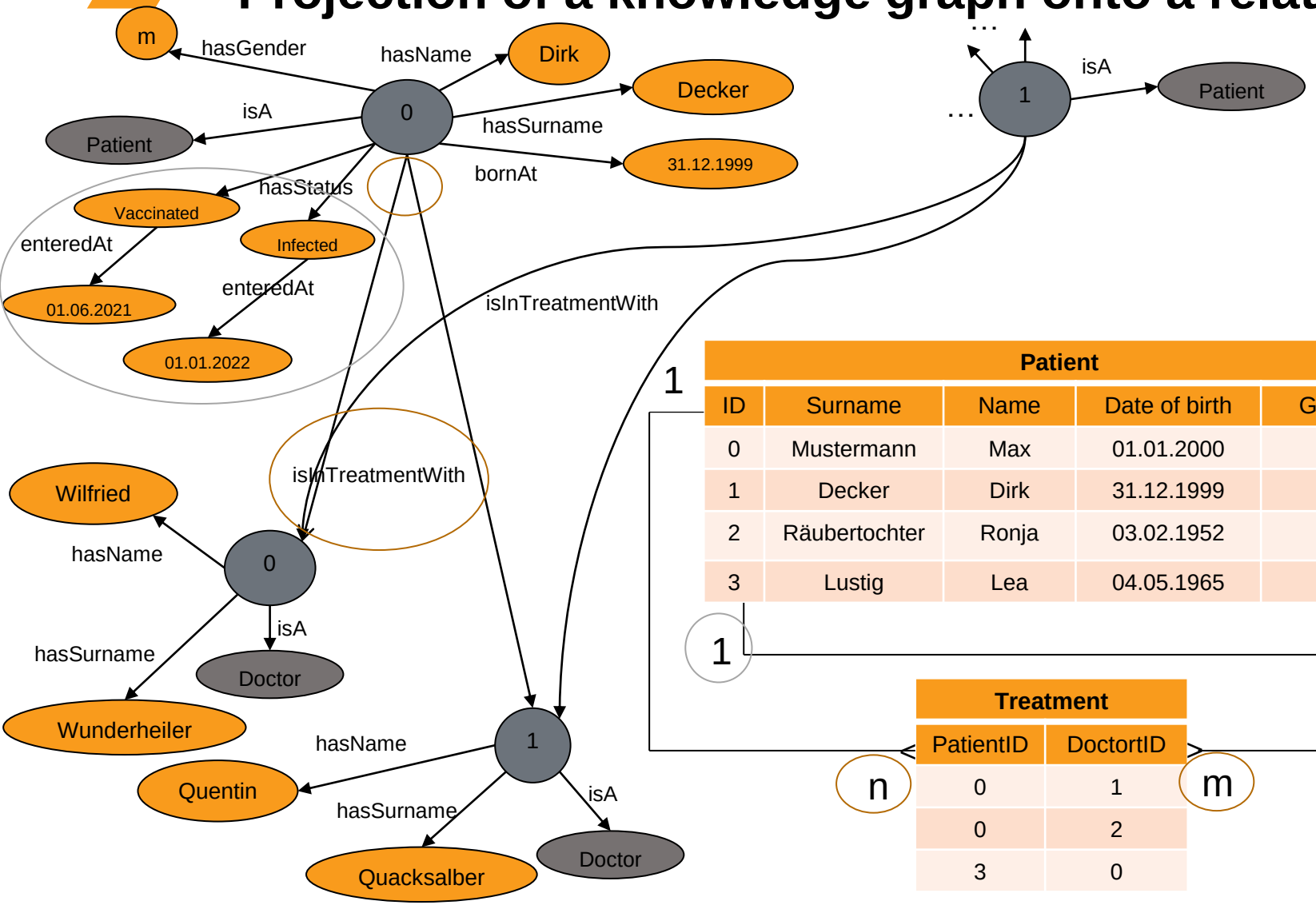
Treatment

PatientID	DoctortID
0	1
0	2
3	0

Doctor

ID	Surname	Name
0	Quacksalber	Quentin
1	Wunderheiler	Wilfried
2	Knochenbrecher	Kurt

Digression - Limitations on semantic level 1.4: Projection of a knowledge graph onto a relational database schema



Patient

ID	Surname	Name	Date of birth	Gender
0	Mustermann	Max	01.01.2000	m
1	Decker	Dirk	31.12.1999	m
2	Räubertochter	Ronja	03.02.1952	w
3	Lustig	Lea	04.05.1965	w

PatientCondition

ID	PatientID	Status	Entry date
0	0	Recovered	14.04.2020
1	1	Vaccinated	01.06.2021
2	2	Vaccinated	21.08.2021
3	3	Infected	05.12.2020
4	1	Infected	01.01.2022

1 — 1 — n

Treatment

PatientID	DoctortID
0	1
0	2
3	0

n — m

Doctor

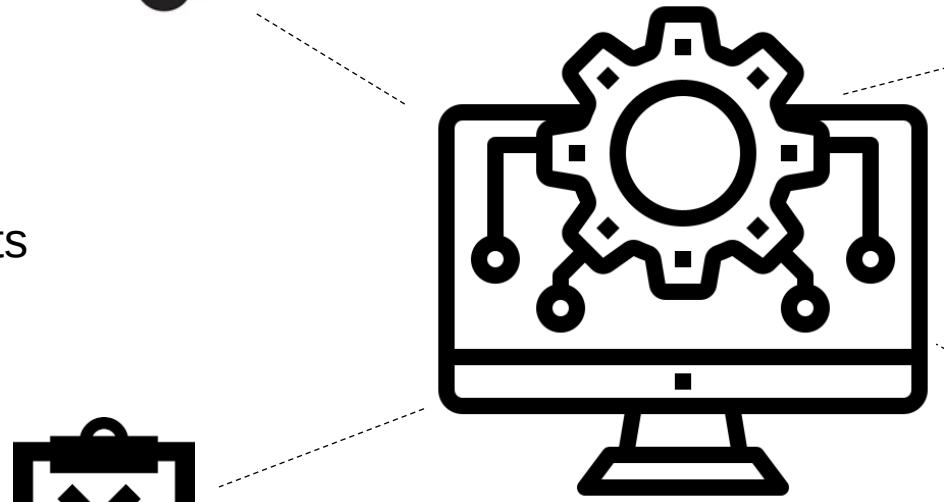
ID	Surname	Name
0	Quacksalber	Quentin
1	Wunderheiler	Wilfried
2	Knochenbrecher	Kurt

1

3.6.2 Requirements for a system for generating meaningful SQL query tasks

Parameterization of the SQL query

- SQL components (e.g. WHERE clause)
- SQL components (e.g. WHERE clause)
- Number of components (e.g. 3-5 constraints)



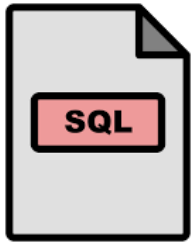
Evaluation of solution attempts

SQL query task generation system

Conclusive

Complete

Natural language description

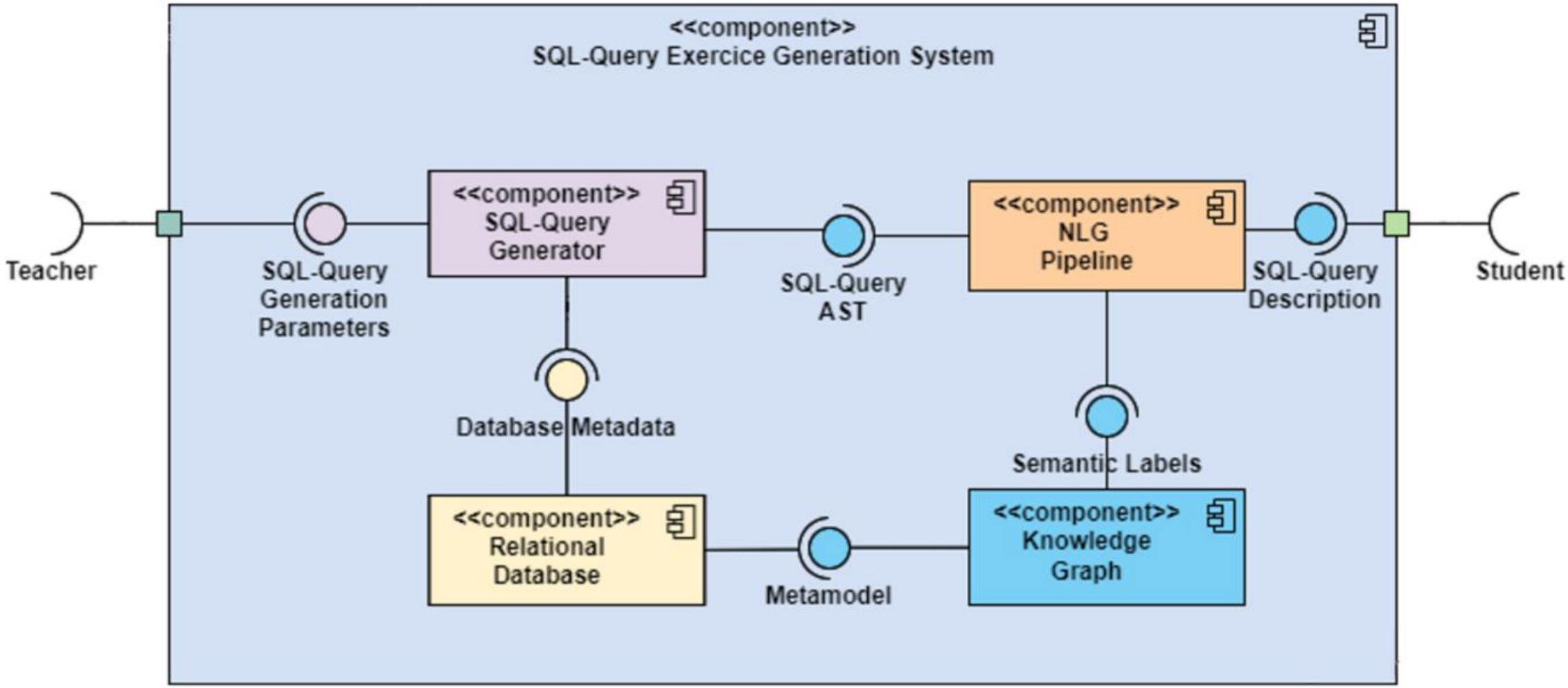


Syntactically correct

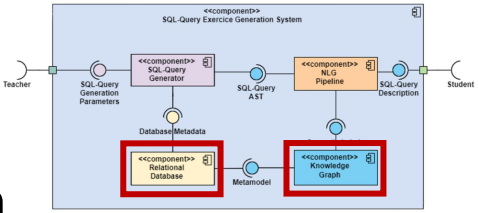
Semantically plausible

SQL-Query

3.6.3 Overview of a system for generating meaningful SQL query tasks



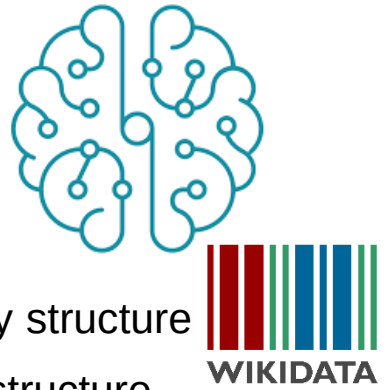
3.6.4 Creating metamodels and knowledge graphs for SQL query tasks

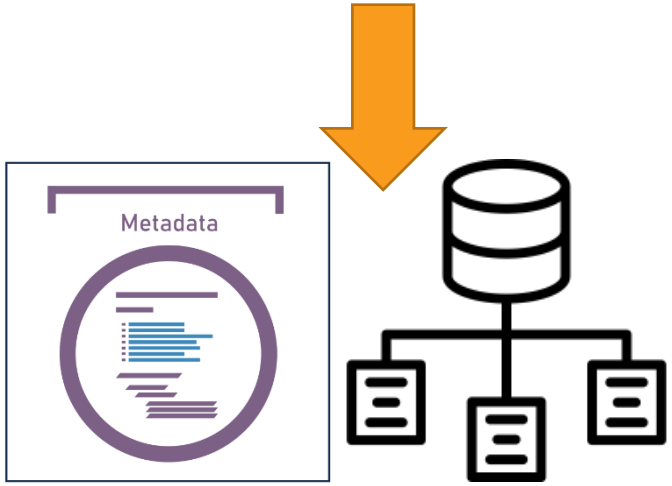


(Partially) Automated generation

Knowledge Graph:

- Entities
- Instances
- Hyperonymy structure
- Meronymy structure





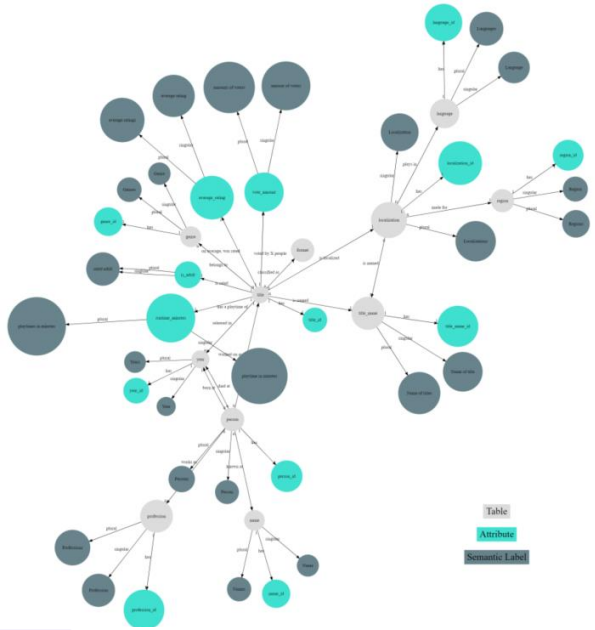
Limitations:

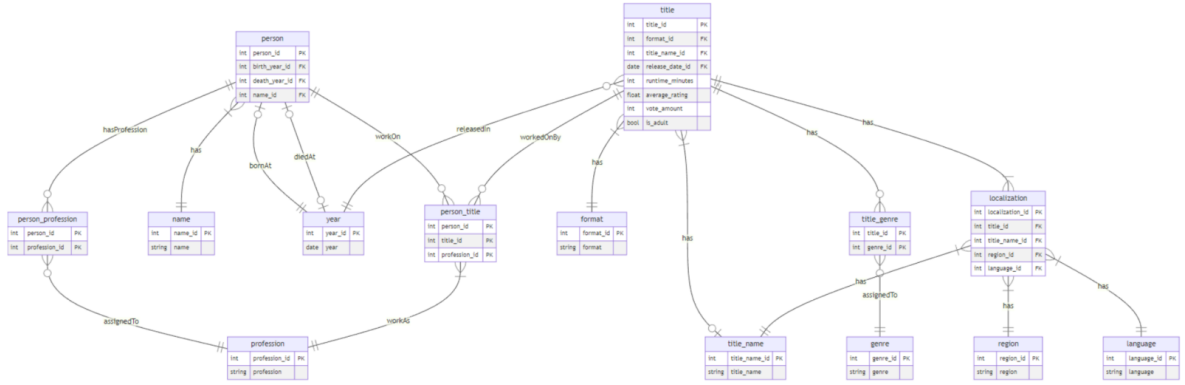
- Automatic derivation of table structures either too specific or too generic
- Attribute assignment unclear, because on instance level instead of class level
- When is entity a class/instance? Context is not modeled in Wikidata
- …

Manual generation

Film- and media database

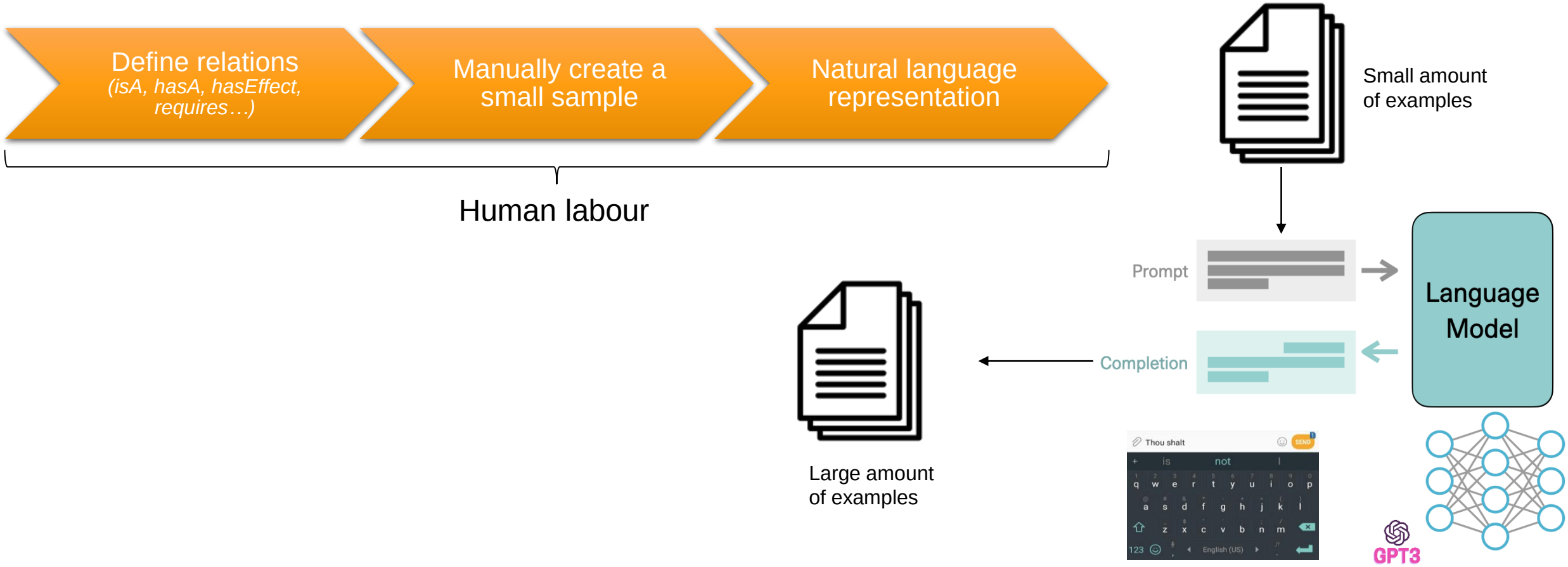
IMDb





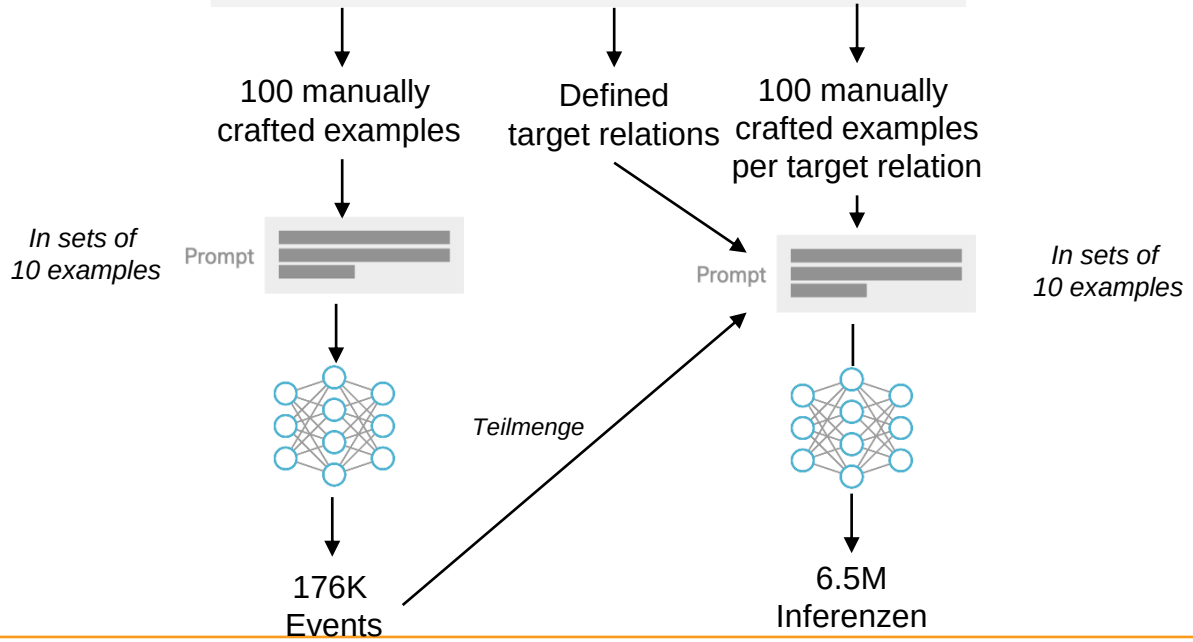
Digression - Dynamic generation of knowledge graphs 2.1

Generative creation of the knowledge database according to the requirements profile of the respective task type, with minimum effort



Digression - Dynamic generation of knowledge graphs 2.2: Symbolic Knowledge Distillation [1]

Event	Relation	Inference
X starts running	**xEffect** *so, X*	gets in shape
X and Y engage in an argument	**xWant** *so, X wants*	to avoid Y
X learns to type fast	**xNeed** *X needed*	to have taken typing lessons
X steals his grandfather's sword	**xEffect** *so, X*	is punished by his grandfather
X takes up new employment	**xIntent** *because X wants*	to be self sufficient



Relation	ATOMIC$^{20}_{20}$	ATOMIC^{10x}
HinderedBy	77,616	**1,028,092**
xNeed	100,995	**760,232**
xWant	109,098	**730,223**
xIntent	54,839	**965,921**
xReact	62,424	**1,033,123**
xAttr	113,096	**884,318**
xEffect	90,868	**1,054,391**
Total Count	608,936	**6,456,300**
Est Total Cost	~$40,000	**~$6,000**
Est Cost Per Triple	~$0.06	**~$0.001**

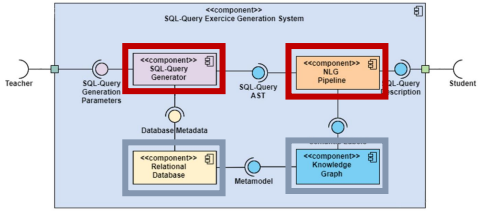
Corpus	Accept	Reject	N/A	Size	Size (div)
ATOMIC$^{20}_{20}$	86.8	11.3	1.9	0.6M	0.56
ATOMIC^{10x}	78.5	18.7	2.8	**6.5M**	**4.38**
	88.4	9.5	2.1	5.1M	3.68
	91.5	6.8	1.7	4.4M	3.25
	94.3	4.6	1.1	3.6M	2.74
	95.3	3.8	1.0	3.0M	2.33
	96.4	2.7	0.8	2.5M	2.00

Cutoffs of the „Critic“-model

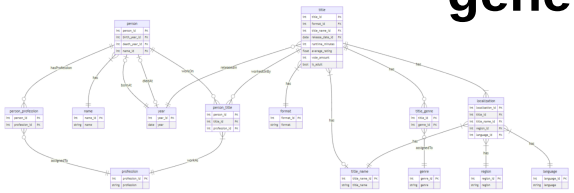
Introduction of the "Critic“-model
Classification model trained on 10K random tuples evaluated by crowdsourcing.

[1] Symbolic Knowledge Distillation: from General Language Models to Commonsense Models

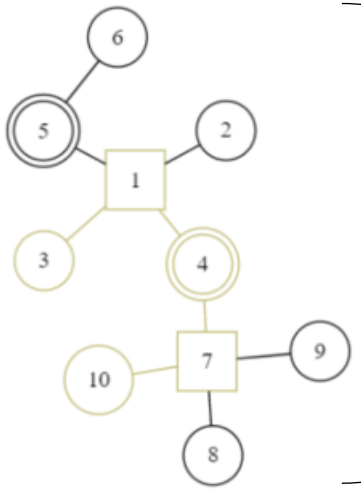
3.6.5 Generators for SQL queries and associated natural language descriptions



SQL-query generation

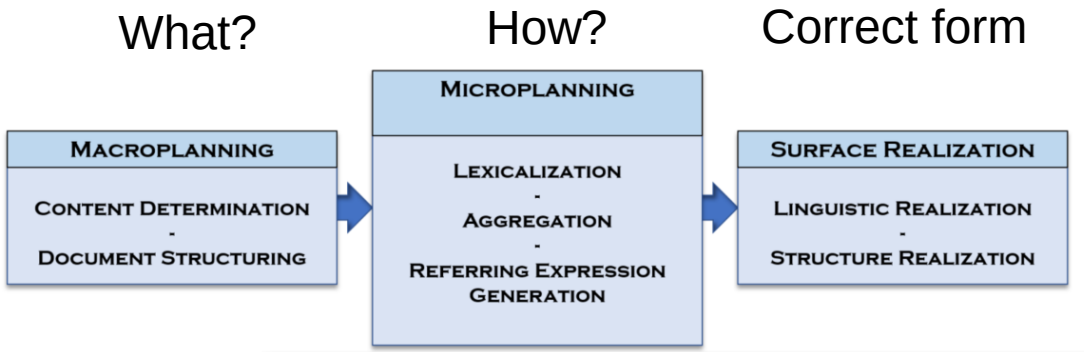


```
SELECT COUNT(t.title_id), t.runtime_minutes, t.vote_amount, lo.
    language_id
FROM imdb2.person as p
INNER JOIN imdb2.year as y
ON p.birthyear_id = y.year_id
INNER JOIN imdb2.title as t
ON y.year_id = t.release_date_id
INNER JOIN imdb2.localization as lo
ON t.title_id = lo.title_i
INNER JOIN imdb2.region as r
ON lo.region_id = r.region_id
WHERE r.region LIKE 'ge%'
GROUP BY t.runtime_minutes, t.vote_amount, lo.language_id
HAVING COUNT(t.title_id) <> '4883174'
ORDER BY p.person_id ASC;
```



1. **FROM CLAUSE**
2. [WHERE CONSTRAINTS]
3. SELECT COLUMN
4. [GROUP BY]
5. [HAVING CONSTRAINTS]
6. [ORDER BY]

Generation of a natural language description



Isomorphic Translation

Form the intersection that contains the corresponding entries of the tables person and year and the intersection that contains the corresponding entries of the tables year and title and the intersection that contains the corresponding entries of the tables title and localization and the intersection that contains the corresponding entries of the tables localization and region. Return the columns the amount of title_id, runtime_minutes, vote_amount and language_id. Only return the data for which region contains 'ge'. A further constraint is the amount of title_id doesn't equal '4883174' . Group the result by runtime_minutes, vote_amount, title_id, language_id and person_id. Sort the result ascending by person_id.

Homomorph Translation

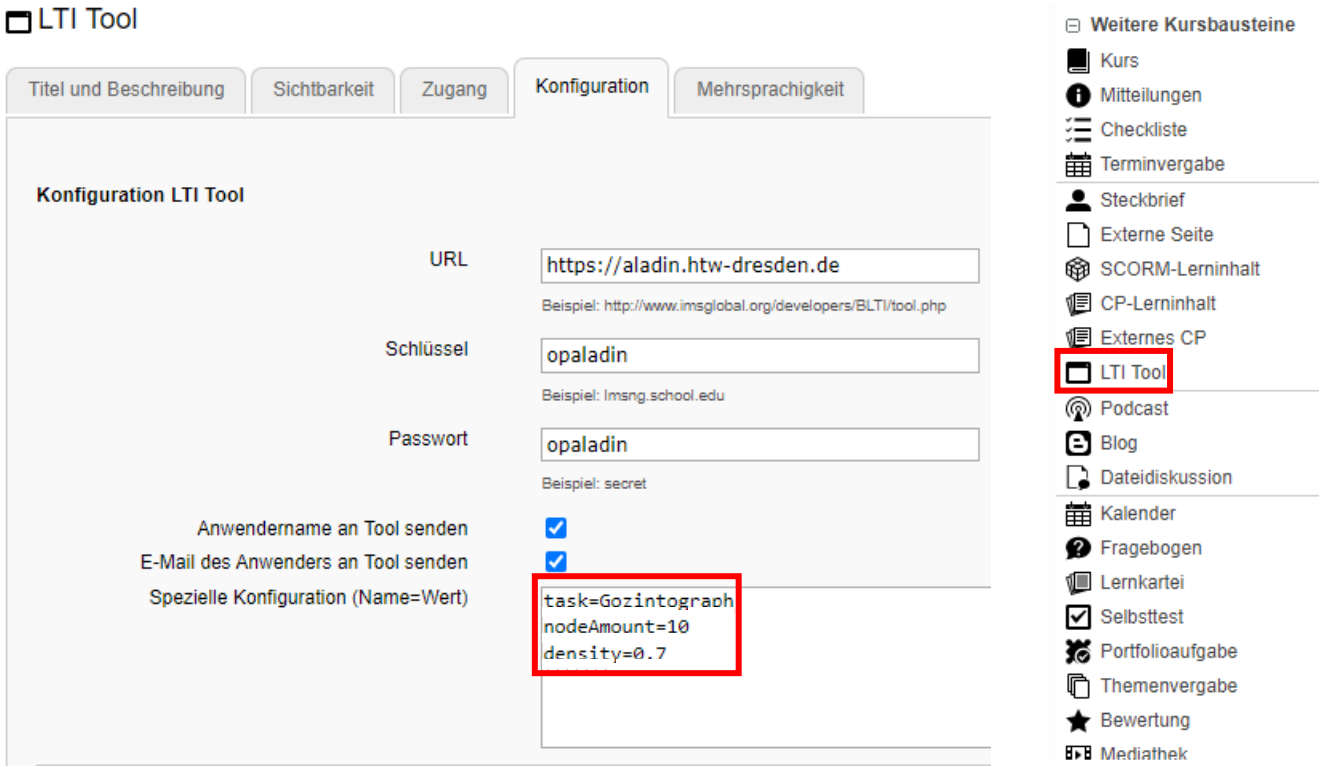
Find the count of title-ids, the runtime, the number of votes and the language-ids for all persons and the titles they were involved with, if the region contains 'ge' and the count of title_id is not '4883174'. Sort the result by the person-id in ascending order.

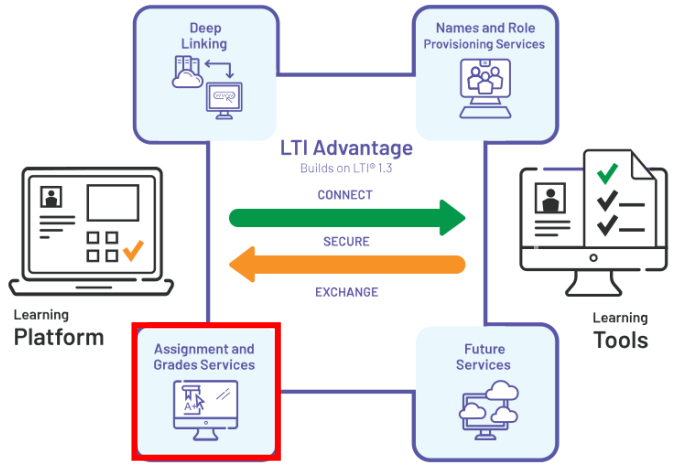
Outline

- 1. Motivation for the development of (OP)ALADIN
- 2. Objectives of (OP)ALADIN
- 3. Generation of semantically plausible exercises and exercise types in (OP)ALADIN
- **4. Learning management and didactics in (OP)ALADIN**
- 5. Summary and outlook

4.1 Integration into OPAL

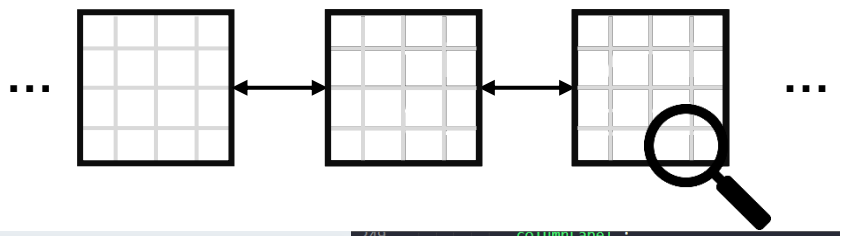
- Publicly accessible hosting of ALADIN
- Embedding into OPAL via the LTI-Tool-Course-Module
- Transmission of ALADIN configurations via the "*Special Configuration*", for the selection of the exercise type and the parameterization of the exercises to be generated
- Transmission of student specific grading information via the „*Assignment und Grades Service*“-extension of the LTI-standard

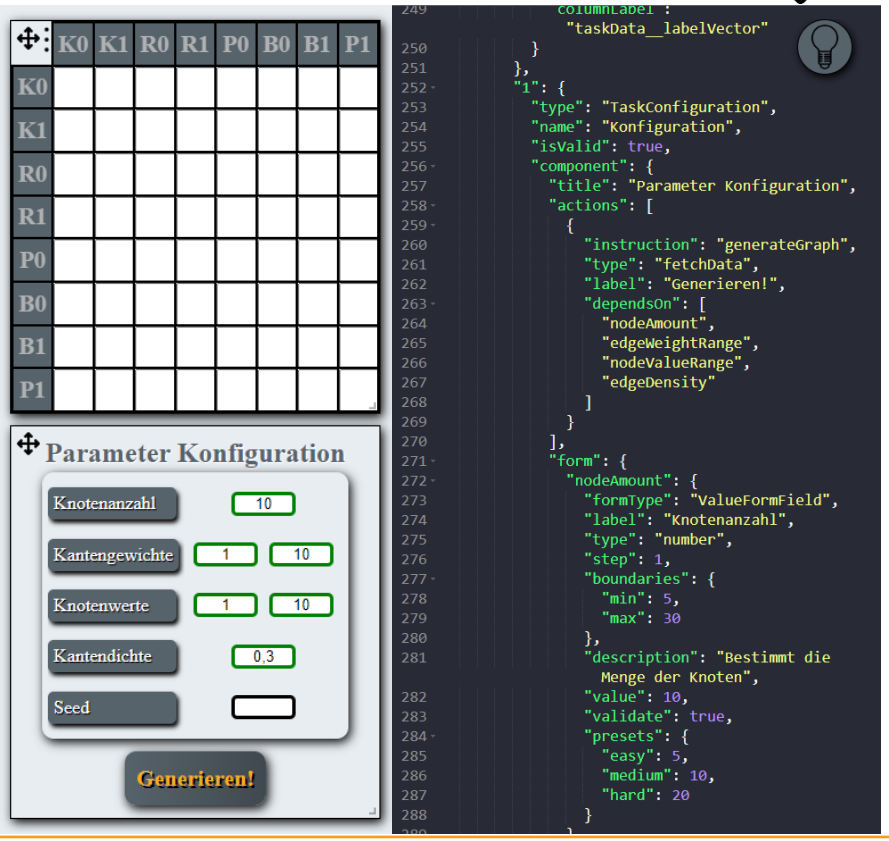




4.2 Declarative Exercise type authoring tool

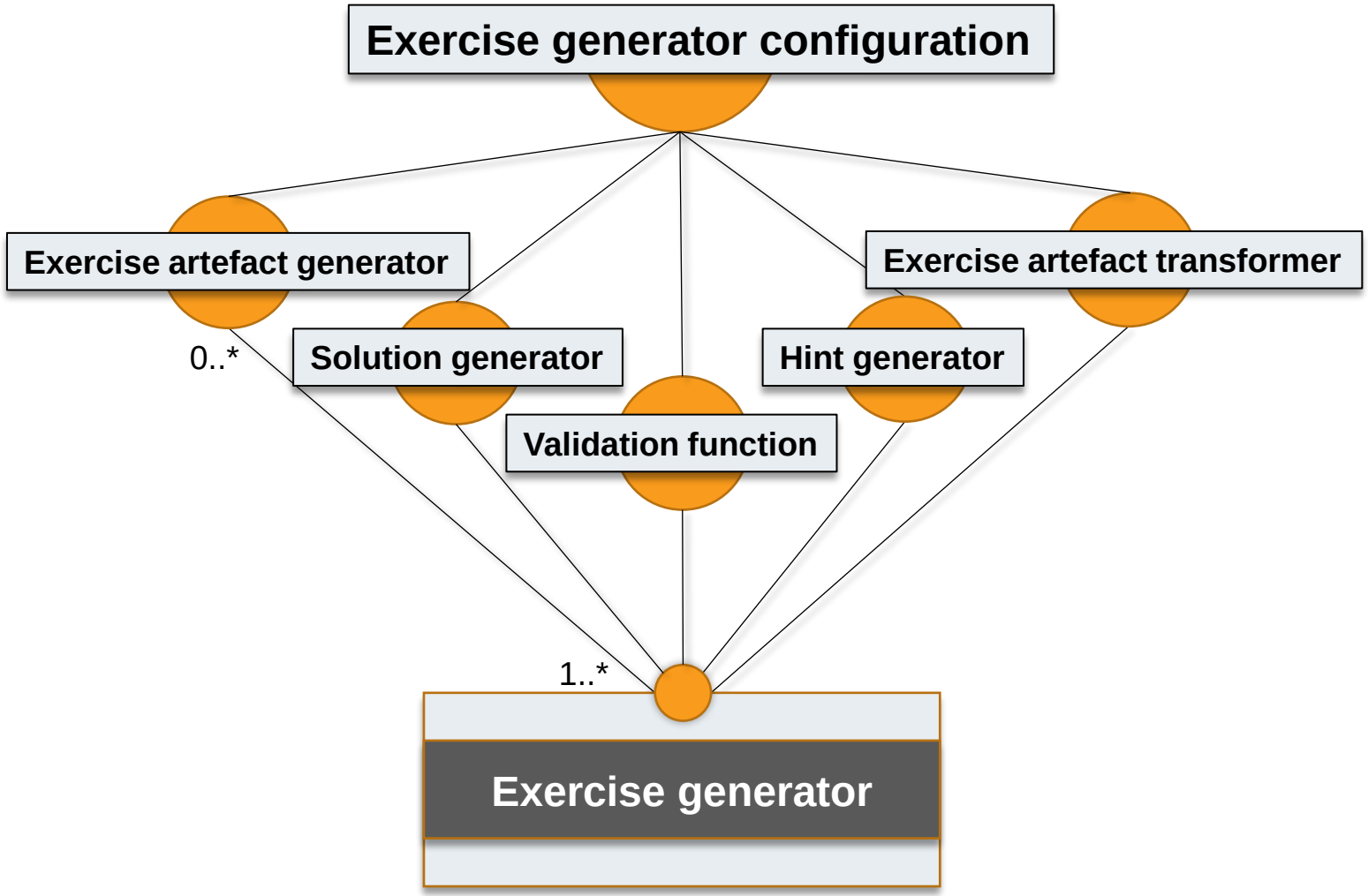
UI





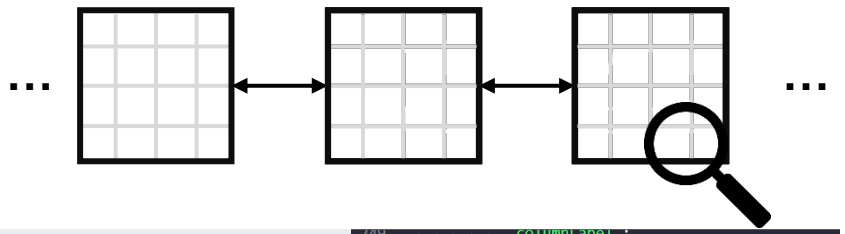

```
        "columnLabel":
          "taskData__labelVector"
      }
    },
    "1": {
      "type": "TaskConfiguration",
      "name": "Konfiguration",
      "isValid": true,
      "component": {
        "title": "Parameter Konfiguration",
        "actions": [
          {
            "instruction": "generateGraph",
            "type": "fetchData",
            "label": "Generieren!",
            "dependsOn": [
              "nodeAmount",
              "edgeWeightRange",
              "nodeValueRange",
              "edgeDensity"
            ]
          }
        ],
        "form": {
          "nodeAmount": {
            "formType": "ValueFormField",
            "label": "Knotenanzahl",
            "type": "number",
            "step": 1,
            "boundaries": {
              "min": 5,
              "max": 30
            },
            "description": "Bestimmt die
              Menge der Knoten",
            "value": 10,
            "validate": true,
            "presets": {
              "easy": 5,
              "medium": 10,
              "hard": 20
            }
```

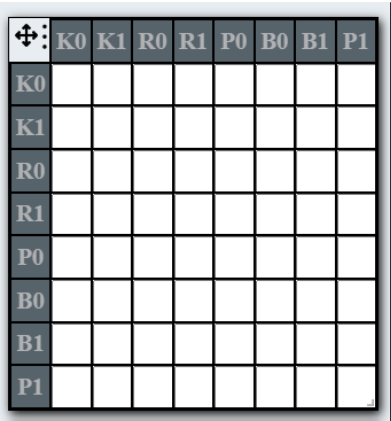
Exercise generator



4.2 Declarative Exercise type authoring tool

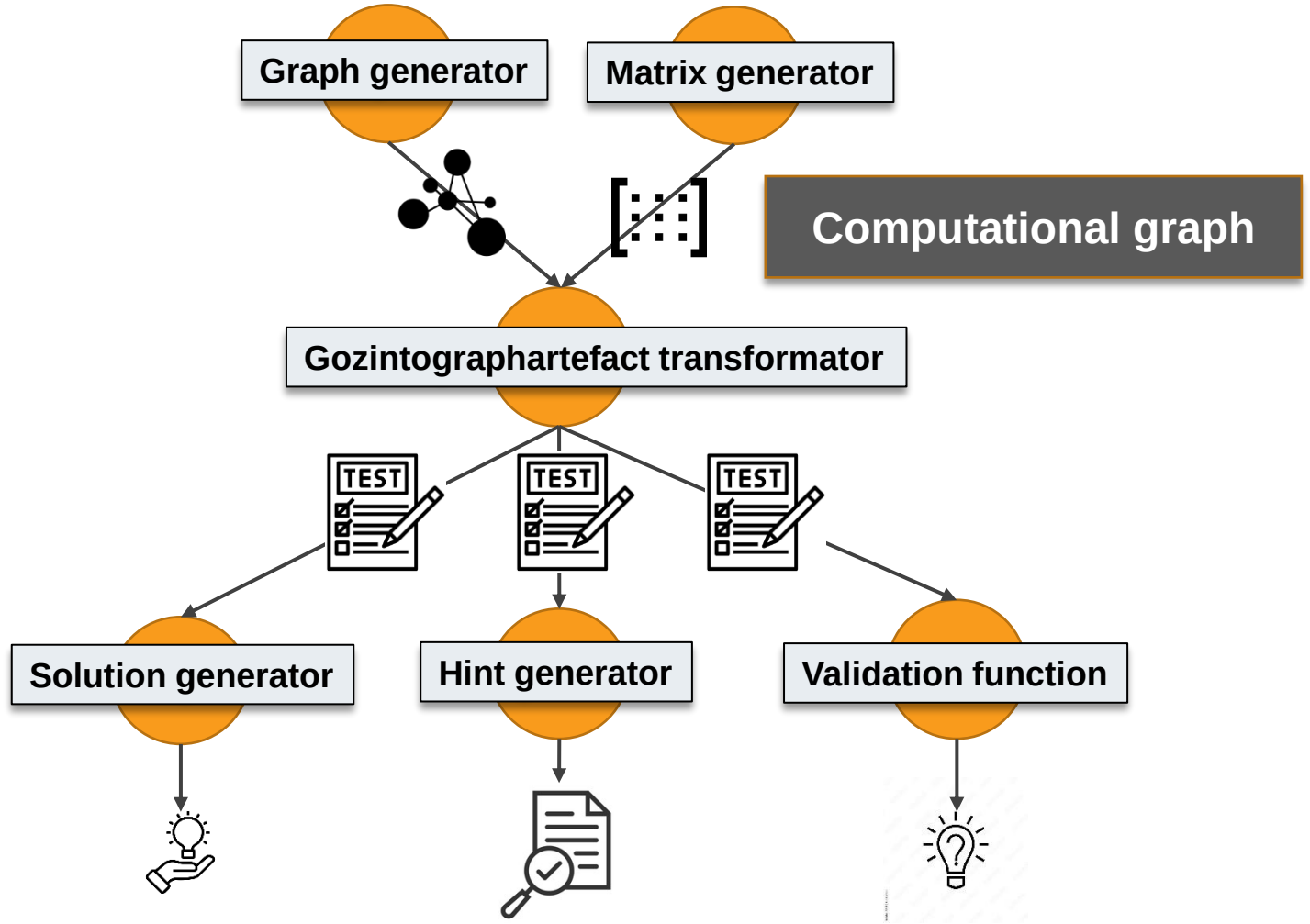
UI





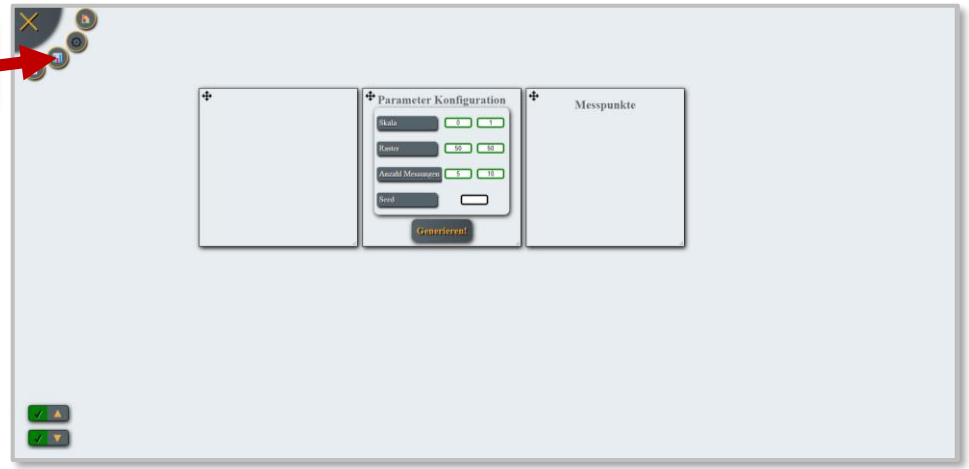
```
            "columnLabel":
              "taskData__labelVector"
          }
        },
        "1": {
          "type": "TaskConfiguration",
          "name": "Konfiguration",
          "isValid": true,
          "component": {
            "title": "Parameter Konfiguration",
            "actions": [
              {
                "instruction": "generateGraph",
                "type": "fetchData",
                "label": "Generieren!",
                "dependsOn": [
                  "nodeAmount",
                  "edgeWeightRange",
                  "nodeValueRange",
                  "edgeDensity"
                ]
              }
            ],
            "form": {
              "nodeAmount": {
                "formType": "ValueFormField",
                "label": "Knotenanzahl",
                "type": "number",
                "step": 1,
                "boundaries": {
                  "min": 5,
                  "max": 30
                },
                "description": "Bestimmt die
                  Menge der Knoten",
                "value": 10,
                "validate": true,
                "presets": {
                  "easy": 5,
                  "medium": 10,
                  "hard": 20
                }
```

Exercise generator

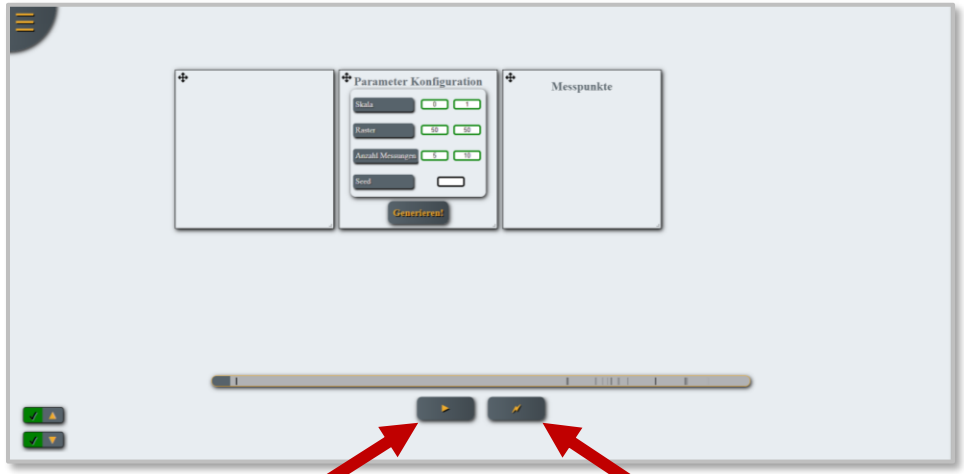


4.3 The 4R-principle (+A) in (OP)ALADIN

1. Record



2. Redirect



3. Replay

4. Resume

- **A**nnotation of the solution attempt…
  - …by the students
  - …by the teaching staff
- Permits…
  - …concrete exercise related questions
  - …specific individual feedback

Outline

- 1. Motivation for the development of (OP)ALADIN
- 2. Objectives of (OP)ALADIN
- 3. Generation of semantically plausible exercises and exercise types in (OP)ALADIN
- 4. Learning management and didactics in (OP)ALADIN
- **5. Summary and outlook**

5. Summary and Outlook

Summary

- Integration of graph replacement systems for declarative creation of exercise type generators
- 4R-principle for asynchronous exchange
- Generation of technically meaningful…
  - …SQL query exercises
  - …EPC modeling exercises
- Prototypical integration into OPAL using the LTI interface
- Modularization of the exercise generation functionalities analogous to the interface modularization

Outlook

- Exercises on nomenclature of molecules and chemical reaction equations
- Declarative task type authoring tool
- Spaced Repetition and Gamification
- Extensive experiments and tests on Spaced Repetition and Gamification (follow-up application ALADIN-X)
- Metamodel and knowledge graph generation (follow-up application METALADIN)