University of
Central Lancashire
UCLan

CO3409
Distributed Enterprise Systems

Summary: Semantic web
Summary: Semantic querying
Summary: Concurrent process models

Where opportunity creates success

University of Central Lancashire
UCLan

Summary: Semantic web

CO3409

17th March 2022

Semantic interoperability

What is semantic interoperability and how is it achieved?

- **Interoperability** vs. **compatibility**: Often distinguished by difference between a shared community standard (n → 1 → n conversion, requiring 2n conversion tools), and absence of a common standard (n → n conversion, requiring $n^2 - n$ conversion tools) or absence of conversion.

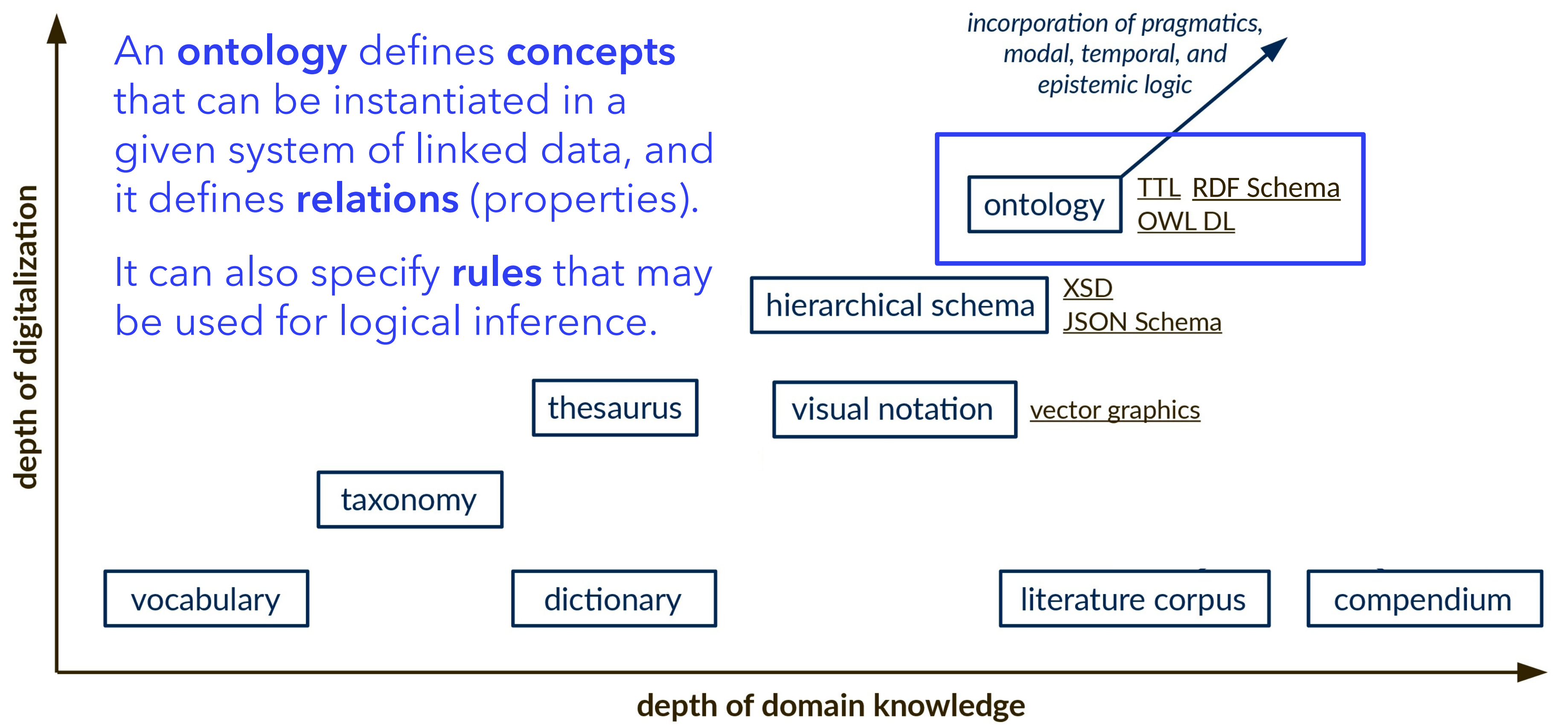
  However, some sources also use the two terms interchangeably.

- Interoperability requires working together in the realms of syntax, semantics, *and* pragmatics. **Semantic interoperability** permits communicating information with a **mutually agreed meaning**, based on a common framework of knowledge representation.

- Such frameworks are called **semantic artefacts** (or: *metadata standards*; or: *metadata schemas*), including ontologies following RDFS and OWL.

Semantic interoperability

Hierarchy of **semantic artefacts** (*i.e.*, metadata standards)



An **ontology** defines **concepts** that can be instantiated in a given system of linked data, and it defines **relations** (properties).

It can also specify **rules** that may be used for logical inference.

incorporation of pragmatics, modal, temporal, and epistemic logic

ontology
TTL RDF Schema
OWL DL

hierarchical schema
XSD
JSON Schema

thesaurus

visual notation
vector graphics

taxonomy

vocabulary

dictionary

literature corpus

compendium

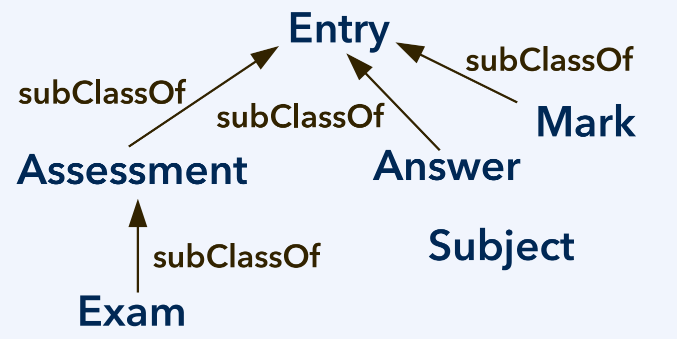
depth of digitalization

depth of domain knowledge

Knowledge bases: TBox and ABox

A **knowledge base** for linked data consists of two components:

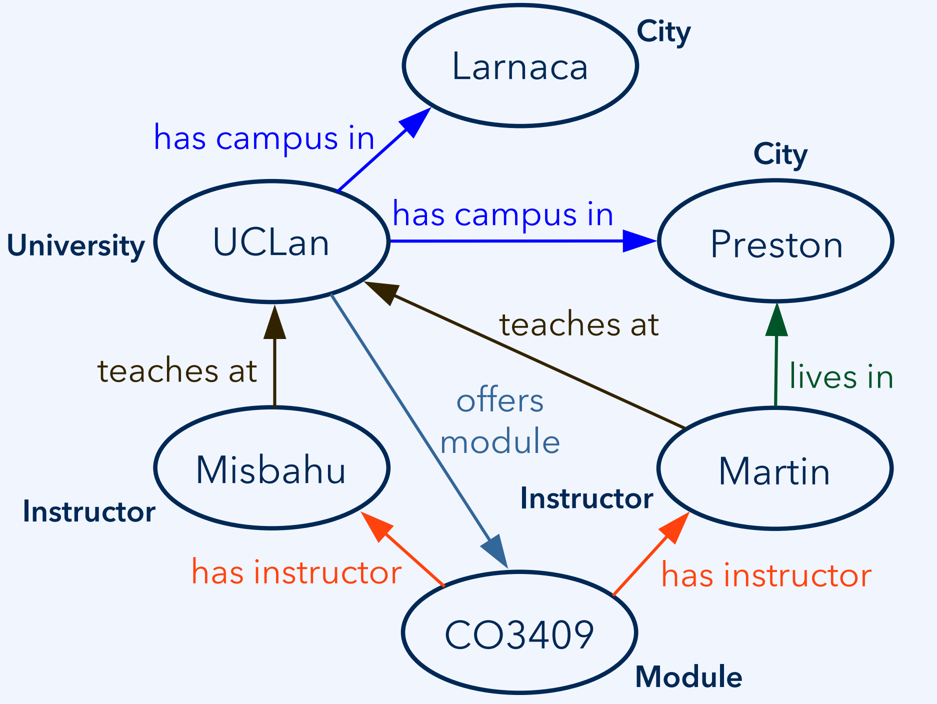
terminological box (TBox), **ontology**, or RDF schema



isMarkFor (Mark → Assessment)
isAssessmentAbout (Assessment → Subject)
isAttributedTo (Mark → Person)

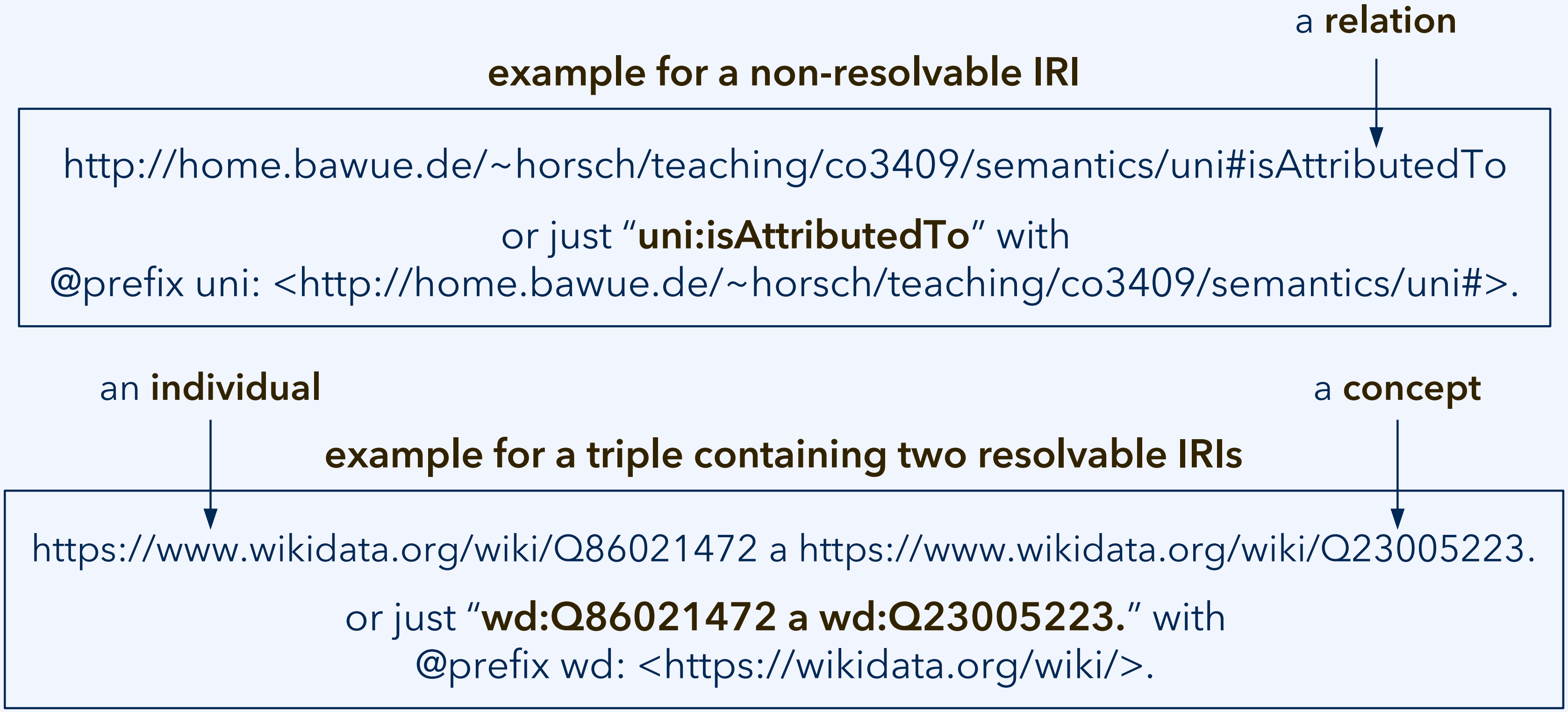
isNumericalMark (Mark → decimal)
hasMarkMaximum (Assessment → decimal)
hasSubmissionTimestamp (Entry → dateTime)

assertional box (ABox) or **knowledge graph**



Internationalized resource identifiers (IRIs)

In the Resource Description Framework (RDF), all **individuals** (objects), **relations** (properties), and **concepts** (classes) are “resources.” The *resource identifiers need not be resolvable*; if they are, they become *locators* (*e.g.*, URLs).



a **relation**

example for a non-resolvable IRI

http://home.bawue.de/~horsch/teaching/co3409/semantics/uni#isAttributedTo

or just “**uni:isAttributedTo**” with
@prefix uni: <http://home.bawue.de/~horsch/teaching/co3409/semantics/uni#>.

an **individual**

a **concept**

example for a triple containing two resolvable IRIs

https://www.wikidata.org/wiki/Q86021472 a https://www.wikidata.org/wiki/Q23005223.

or just “**wd:Q86021472 a wd:Q23005223.**” with
@prefix wd: <https://wikidata.org/wiki/>.

Key elements of RDF schema and OWL

animals:Fox a owl:Class.

concepts are of the type owl:Class

taxonomy specified using rdfs:subClassOf

animals:Fox rdfs:subClassOf animals:Canidae.
animals:Canidae rdfs:subClassOf animals:Mammalia.

animals:isNaturalEnemyOf a owl:ObjectProperty.

relations are of the type owl:ObjectProperty

animals:isNaturalEnemyOf rdfs:domain animals:Predator.

rdfs:domain and similarly **rdfs:range**

uni:hasSubmissionTimestamp a owl:DatatypeProperty.

uni:hasSubmissionTimestamp rdfs:range xs:dateTime;
rdfs:subPropertyOf uni:hasTimestamp.

elementary **datatype properties**

hierarchy for datatype properties and **relations** (object propertes)

FAIR principles[1]

Findability

F1. Globally unique **persistent identifiers (PID)**
F2. **Enriched with metadata**
F3. Data identifier included in metadata
F4. **Registered**/indexed in **searchable platform**

Accessibility

A1. **Retrievable from PID** via a standard protocol
A1.1. Open and freely implementable protocol
A1.2. … **authentication/authorization** if necessary
A2. Metadata remain accessible (beyond data)

Interoperability

I1. **Formal language** used for **knowledge representation**
I2. Metadata use **vocabularies** that are themselves FAIR
I3. Semantic web principles, **data can refer to other data**

Reusability

R1. Metadata include a plurality of accurate and relevant attributes
R1.1. Release data and metadata with an accessible **data usage licence**
R1.2. Data are annotated with a detailed **provenance description**
R1.3. Relevant **disciplinary and community standards** are fulfilled

[1] M. D. Wilkinson *et al.*, "The FAIR Guiding Principles …," doi:10.1038/sdata.2016.18, **2016**.

Persistent identifiers (PIDs)

FAIR principles directly related to persistent identifiers:

- F1. **Globally unique persistent** identifiers (PID)
- A1. Retrievable from PID via a **standard protocol**
- A1.1. Open and freely **implementable protocol**

Long-term: PID should remain constant even if organizations, hosting, etc., are subject to change.

Examples of PID systems:

- **Digital Object Identifier (DOI)**, *e.g.*, doi:10.1007/978-3-030-68597-3
- Uniform Resource Name[1] (URN), *e.g.*, urn:isbn:9783030685973
- Persistent URL (PURL), *e.g.*, http://purl.obolibrary.org/obo/BFO_0000001
- Open Researcher and Contributor ID (ORCID), *e.g.*, 0000-0002-9464-6739

To resolve a DOI, of the type doi:XXX, access https://dx.doi.org/XXX.

[1] https://www.w3.org/2001/tag/doc/URNsAndRegistries-50

Exam preparation advice

Summary of learning outcomes

- Principles of dealing with distributed data on the semantic web:
  - **Linked data:** Objects (individuals) connected by properties (relations)
  - Three kinds of resources: **Concepts** (classes), **individuals**, and **relations**
  - Three elements of an **RDF triple**: Subject, predicate, and object
  - **Knowledge base** and its two parts: TBox (schema, ontology) and ABox
  - **Knowledge graph** as representation of the content of the ABox
  - **Identifiers:** What is an IRI? What is a PID?
  - **Open world assumption** and non-unique name assumption
  - Be able to analyse and critically discuss compliance with **FAIR principles**
- Know, understand, and apply basic constructs from **RDFS/OWL**
- Denote TBox in **TTL format**, and ABox in **JSON-LD** or **TTL format**

University of Central Lancashire
UCLan

Summary: Semantic querying

CO3409

17th March 2022

Concepts

Linked data that are formalized as RDF triples can be stored:

- in **graph databases**, also called **triple stores** (*e.g.*, Apache Jena Fuseki)
- using JSON based **noSQL databases** such as MongoDB
- architectures containing such components need to facilitate **querying**

Linked data are queried using the SPARQL query language:

- in a “SELECT … WHERE …” query, a **graph pattern** is defined by **triples**
- the graph pattern contains free variables, some of which are selected
- the server needs to *identify all the occurrences* of the graph pattern
- the response **returns tabular data** with one column per selected variable

A **SPARQL end point** is a host that provides a SPARQL querying service, most frequently via a RESTful API and accessible through a web frontend.

SPARQL syntax

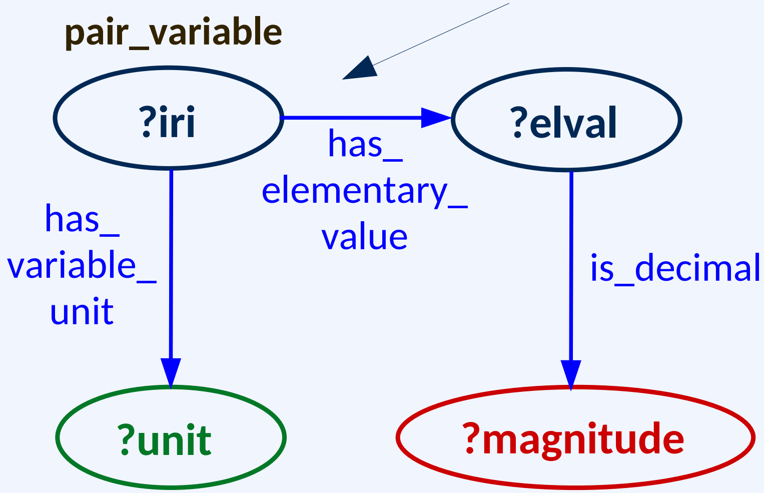
SELECT ?x ?y … WHERE {sequence of triples involving ?x, ?y, …}

selected free variables

there can be additional (non-selected) free variables

graph pattern

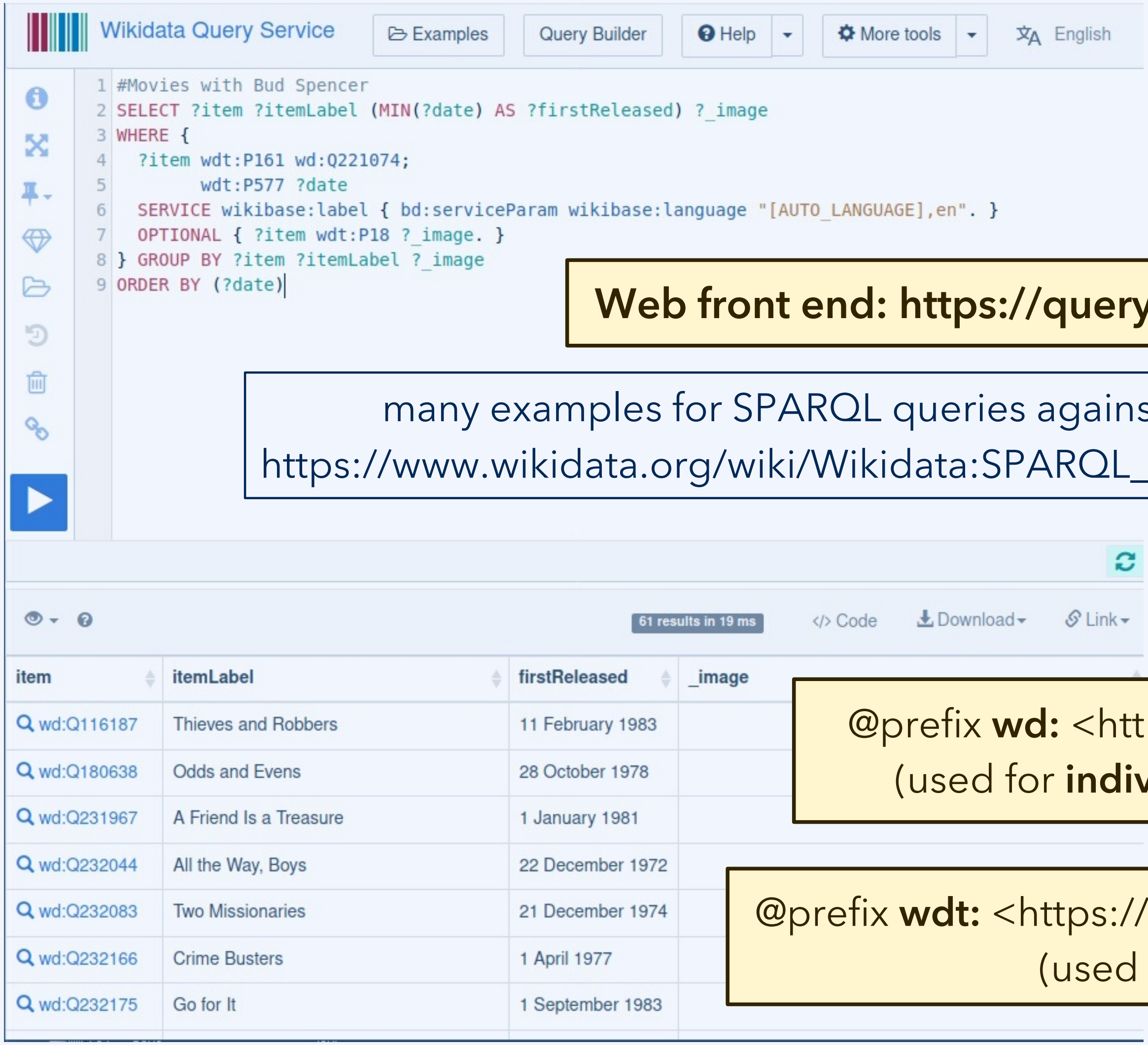
```
SELECT ?magnitude ?unit
WHERE {
   ?iri rdf:type :pair_variable.
   ?iri :has_elementary_value ?elval.
   ?elval :is_decimal ?magnitude.
   ?iri :has_variable_unit ?unit.
}
```



Observation: The WHERE clause consists of RDF triples with free variables.

[1] W3C recommendation, https://www.w3.org/TR/sparql11-query/, **2013**.

Wikidata SPARQL end point



```
#Movies with Bud Spencer
SELECT ?item ?itemLabel (MIN(?date) AS ?firstReleased) ?_image
WHERE {
  ?item wdt:P161 wd:Q221074;
        wdt:P577 ?date
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }
  OPTIONAL { ?item wdt:P18 ?_image. }
} GROUP BY ?item ?itemLabel ?_image
ORDER BY (?date)
```

item	itemLabel	firstReleased	_image
wd:Q116187	Thieves and Robbers	11 February 1983	
wd:Q180638	Odds and Evens	28 October 1978	
wd:Q231967	A Friend Is a Treasure	1 January 1981	
wd:Q232044	All the Way, Boys	22 December 1972	
wd:Q232083	Two Missionaries	21 December 1974	
wd:Q232166	Crime Busters	1 April 1977	
wd:Q232175	Go for It	1 September 1983	

Web front end: https://query.wikidata.org/

many examples for SPARQL queries against Wikidata are available at https://www.wikidata.org/wiki/Wikidata:SPARQL_query_service/queries/examples

@prefix **wd:** <https://wikidata.org/wiki/>
(used for **individuals** and **concepts**)

@prefix **wdt:** <https://wikidata.org/wiki/Property:>
(used for **relations**)

SPARQL and competency questions

Competency questions are queries that the system must or should/could better be able to **answer competently**. The employed **metadata standards** must be powerful enough to **express the competency questions in SPARQL**.

What are the numerical values of the marks?

```
SELECT ?mark ?value WHERE {
  ?mark uni:isNumericalMark ?value.
}
```

What exams are there marks for? What subjects are the exams on?

```
SELECT DISTINCT ?exam ?subject WHERE {
  ?mark uni:isMarkFor ?exam.
  ?exam rdf:type uni:Exam.
  ?exam uni:isAssessmentAbout ?subject.
}
```

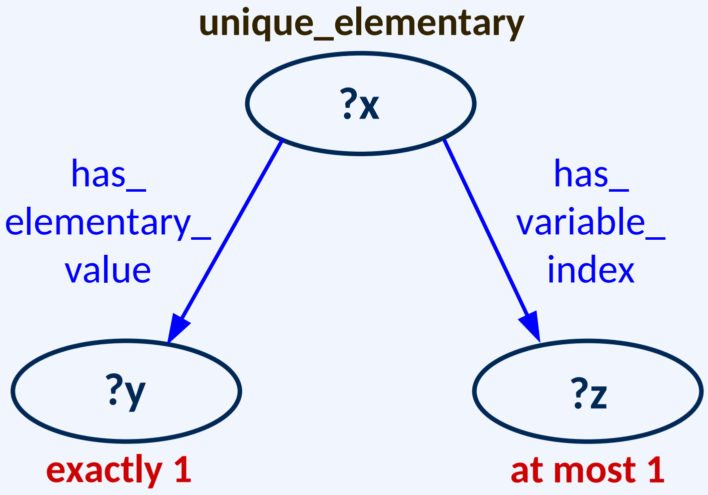
Competency questions can be acquired from users or, in general, from the project's or platform's stakeholders. It is a user-oriented technique that integrates well into an **agile requirements analysis** based on user stories.

SHACL constraints

RDF schema, combined with the open world assumption, is very liberal in what knowledge graphs it permits. However, an API will usually need to specify a concrete kind of information content to be exchanged for a particular action.

Shapes Constraint Language (SHACL) can be used for such specifications.[1]

```
:unique_elementary_shape a sh:Shape;
  sh:targetClass :unique_elementary;
  sh:property [
      sh:path :has_elementary_value;
      sh:minCount 1;
      sh:maxCount 1
    ], [
      sh:path :has_variable_index;
      sh:maxCount 1
    ].
```



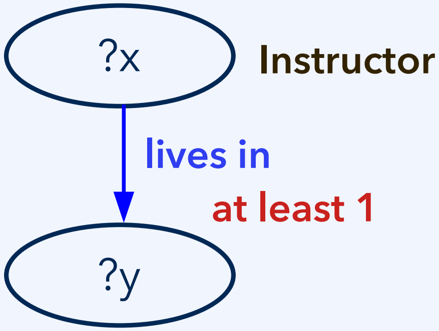
The open world assumption is **not** applied when evaluating SHACL constraints!

[1] W3C recommendation, https://www.w3.org/TR/shacl/, **2017**.

SHACL constraints vs. SPARQL queries

Shapes Constraint Language (SHACL) constraints share a major feature with SPARQL queries: Both are **conventions for specifying graph patterns**.

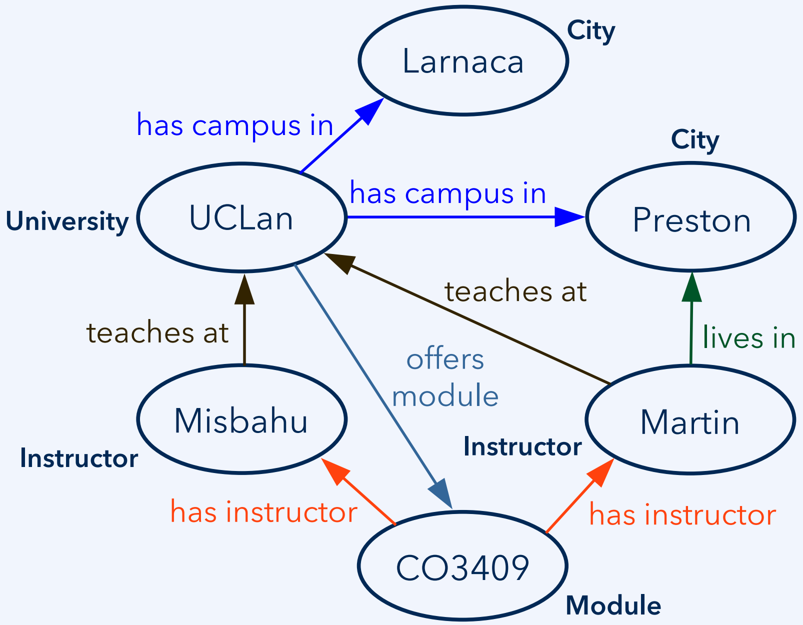
```
# SHACL constraint expressing:
# "Every instructor lives in at least one city."

@prefix uni: …
@prefix scenario-api: …

@prefix sh: <http://www.w3.org/ns/shacl#>.

scenario-api:instructor_city_constraint a sh:Shape;
  sh:targetClass uni:Instructor;
  sh:property [
      sh:path uni:livesIn;
      sh:minCount 1
    ].
```



```
SELECT DISTINCT ?x WHERE
{
  ?x rdf:type uni:Instructor.
  ?x uni:livesIn ?y.
}
```

SHACL constraints vs. SPARQL queries

No! Your knowledge graph does not conform with the constraint.



I found one occurrence of the pattern as a subgraph. The variable ?x has the value …:Martin.

SHACL

SPARQL

```
scenario-api:instructor_city_constraint a sh:Shape;
  sh:targetClass uni:Instructor;
  sh:property [
      sh:path uni:livesIn;
      sh:minCount 1
    ].
```

```
SELECT DISTINCT ?x WHERE
{
  ?x rdf:type uni:Instructor.
  ?x uni:livesIn ?y.
}
```

Exam preparation advice

Summary of learning outcomes

Semantic querying (only "SELECT … WHERE" queries using SPARQL):

- What is SPARQL used for? What is a SPARQL end point?
- Passive knowledge of SPARQL: **Read a query** and predict the response
- Active knowledge: **Formulate a query** that asks for needed information

Competency questions:

- What are competency questions? What are they used for, and why?
- **Formulate competency questions** (*e.g.*, as part of a requirements analysis)
- Why would we attempt to **express competency questions in SPARQL**?

Shape constraints using SHACL (*no need to learn SHACL syntax!*):

- How does the **use of graph patterns** in SHACL differ from that in SPARQL?
- Why is it important that SHACL disregards the **open world assumption**?

Summary: Concurrent process models

CO3409

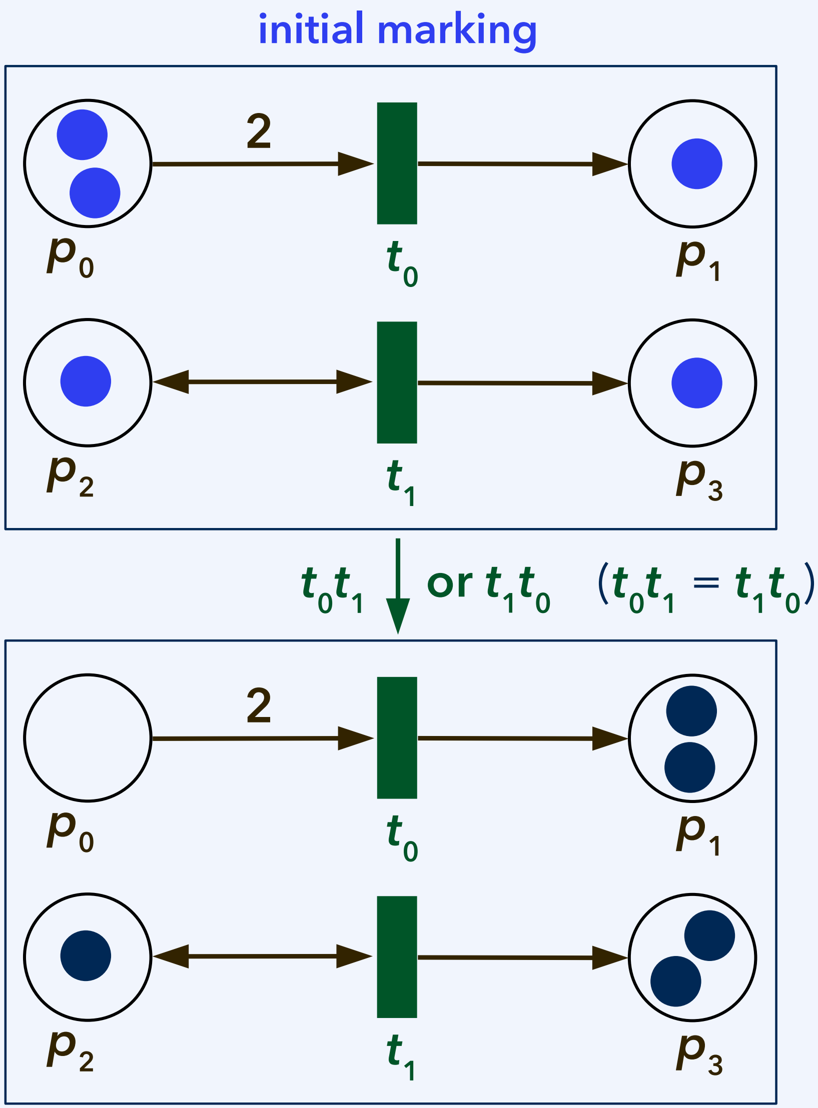
17th March 2022

Petri nets: Concepts

The **elements of a Petri net** are places, transitions, arcs, and tokens. Petri nets are also called place/transition nets (P/T nets). **Firing sequences** consist of the events, potentially concurrent, of firing transitions.

The state/configuration of a Petri net, given by the tokens present at each place, is also called a marking. The **state space of a Petri net** consists of the markings that can be reached from the **initial marking** via any of the possible firing sequences.

Common notations (see at the right): **Unlabelled arcs** and **bidirectional arcs**.



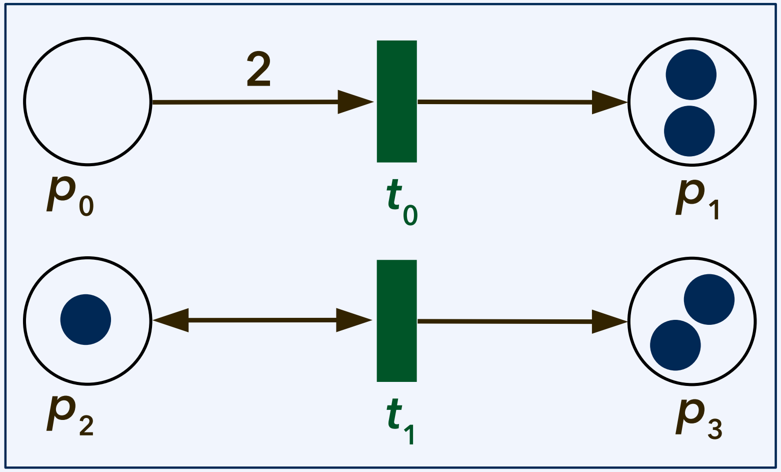
Petri nets: Concepts

A marking (state) of a Petri net is:

- … **reachable** if there is a firing sequence by which that marking is reached (generated) from the initial marking.
- … a **deadlock** if there are no active transitions.
  A transition is **active** in a given marking if it can be fired.
- … **denoted by** a tuple (or array, or vector) such as **(0, 2, 1, 2)** below.

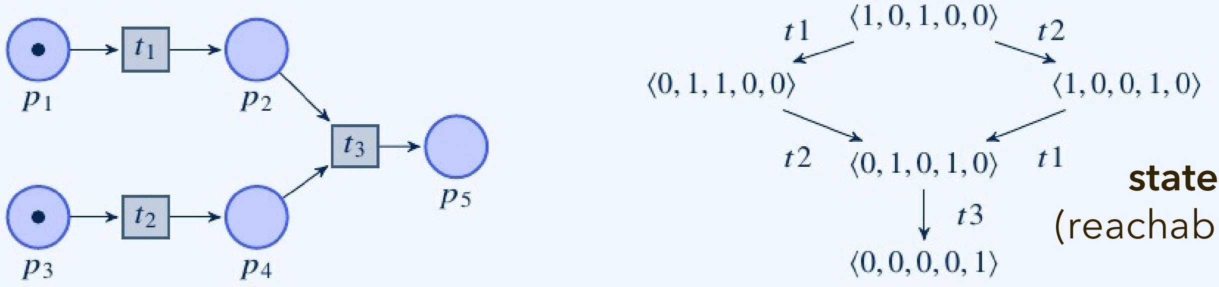
A Petri net is:

- **safe** if no reachable marking has a place with more than one token
- **bounded** if there is a finite upper bound for tokens in all places
- **deadlock-free** (also "weakly live") if there is no reachable deadlock



Petri nets, dependency (“linking”), and concurrency

From Buchs *et al.*, doi:10.1007/978-3-030-43946-0_7, **2020**.


p_1
t_1
p_2
t_3
p_5
p_3
t_2
p_4
$\langle 1,0,1,0,0\rangle$
$t1$
$t2$
$\langle 0,1,1,0,0\rangle$
$\langle 1,0,0,1,0\rangle$
$t2$
$t1$
$\langle 0,1,0,1,0\rangle$
$t3$
$\langle 0,0,0,0,1\rangle$

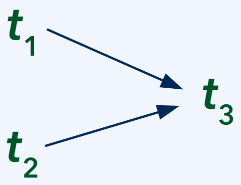
state space
(reachability graph)

Fig. 7.12: A simple Petri net and its reachability graph. The markings are encoded as $\langle p_1, p_2, \ldots, p_n\rangle$, where p_i are the number of tokens within the places, which the ordered by their index. $\langle 1,0,1,0,0\rangle$ encodes the initial marking on the left, stating that there is one token in the first and third place each (i.e. p_1 and p_3).

Two events are **directly or indirectly causally dependent** if one is specified to occur (conclude) before the other occurs (begins). Right: Firing events of $\boldsymbol{t_1}$ and $\boldsymbol{t_3}$.

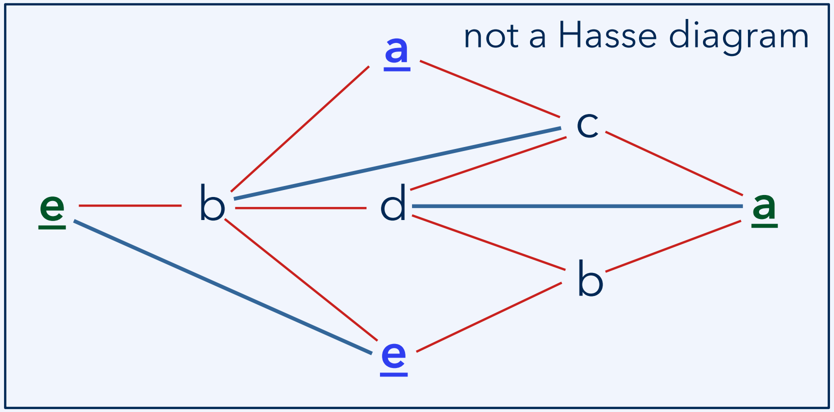
Events are **concurrent** if they are not directly or indirectly causally dependent – it does not matter which occurs first. Right: Firing events of $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$.

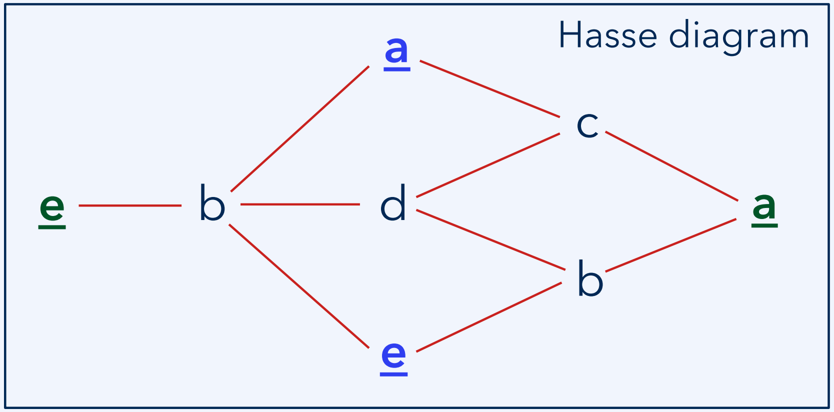
firing sequence
(Hasse diagram)



Petri nets, dependency (“linking”), and concurrency

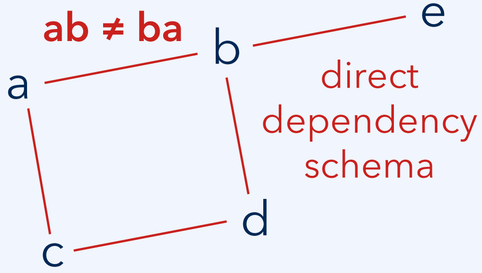
By convention, **Hasse diagrams** are often used to denote causal dependency of events. These diagrams remove *any **indirect** or **redundant** dependencies*:





Two events are **directly or indirectly causally dependent** if one is specified to occur (conclude) before the other occurs (begins). Above: e and a are indirectly dependent.

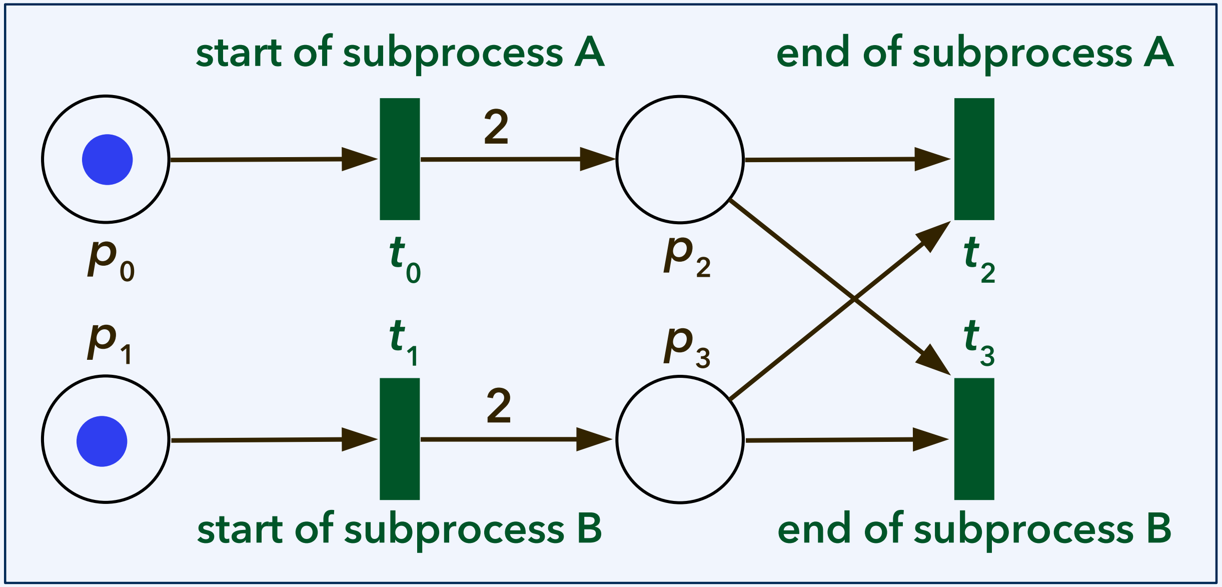
Events are **concurrent** if they are not directly or indirectly causally dependent – it does not matter which occurs first. Above: e and a are concurrent.

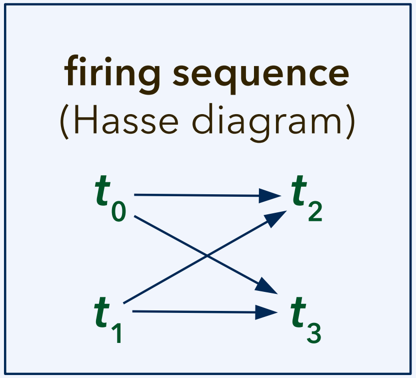


Petri nets and synchronicity (“coupling”)

Two subprocesses are synchronous (also, “coupled”) if it is specified that they must overlap temporally, *i.e.*, they must at least in part run at the same time.

Petri net representing two synchronous subprocesses A and B

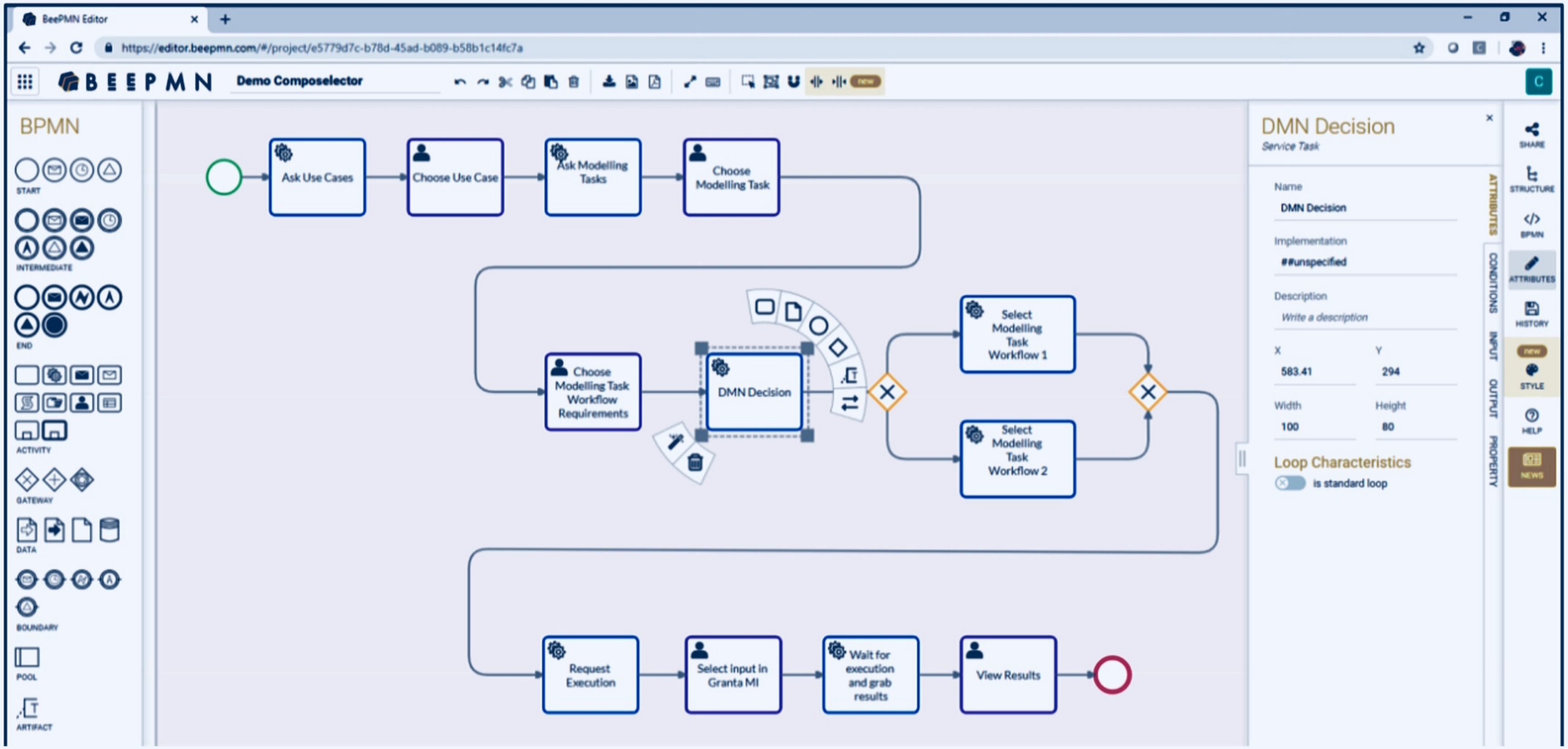




Note: **Synchronicity** (“**coupling**” – subprocesses must overlap) vs. **direct causal dependency** (“**linking**” – may not overlap) vs. **concurrency** (order unspecified).

Business process model and notation

BPMN is a powerful workflow notation, standardized[1] as ISO/IEC 19510:2013.



DMN Decision

Service Task

Name: DMN Decision

Implementation: ##unspecified

Description: *Write a description*

X: 583.41 Y: 294

Width: 100 Height: 80

Loop Characteristics

is standard loop

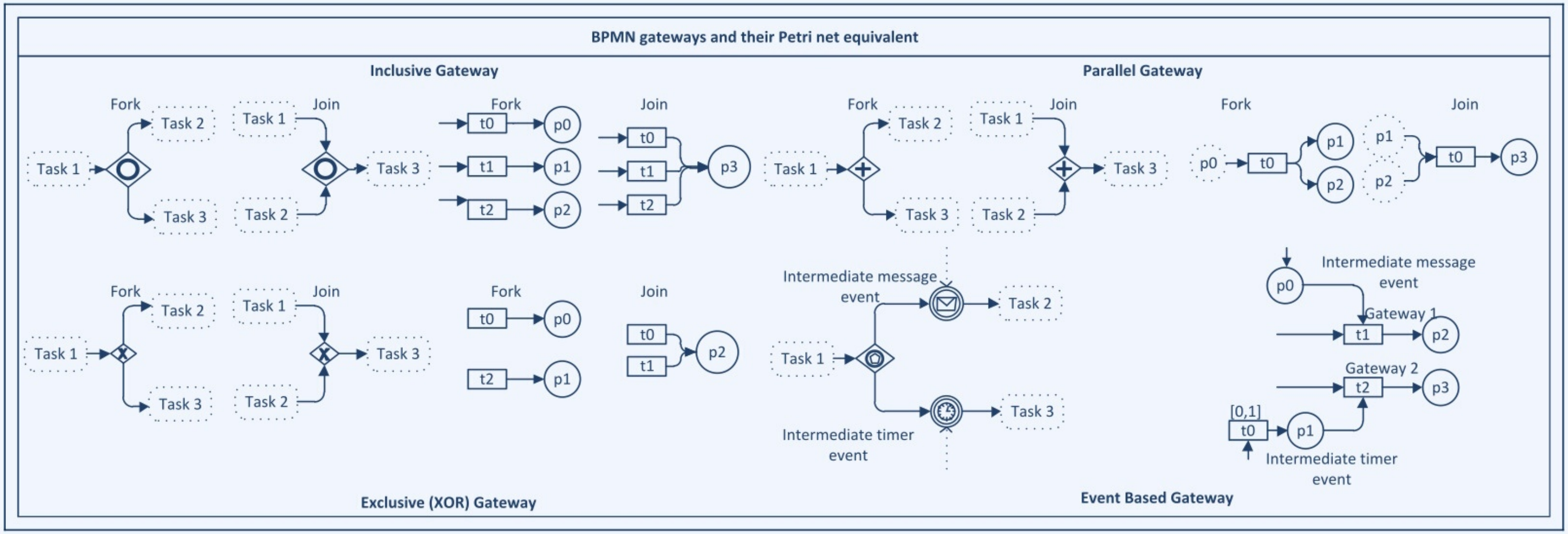
Example from A. Segatto, M. Milleri, C. Kavka, COMPOSELECTOR project deliverable 3.4, **2018**.

[1]See also the specification at https://www.omg.org/spec/BPMN/2.0.2/PDF.

Business process model and notation

BPMN workflows permit:

- XML input/output of workflows[1] based on an XML schema (XSD)
- Hierarchical inclusion of a subworkflow within an overarching workflow
- Orchestration via process automation systems[2] (*e.g.*, Camunda)
- … and there are algorithms that translate BPMN into Petri nets:[3]



[1] https://www.omg.org/spec/BPMN/2.0.2/PDF. [2] Ruecker, *Practical Process Automation*, O'Reilly, **2021**.
[3] U. Mutarraf *et al.*, *Adv. Mech. Eng.* 10(12), doi:10.1177/1687814018808170, **2018**.

Exam preparation advice

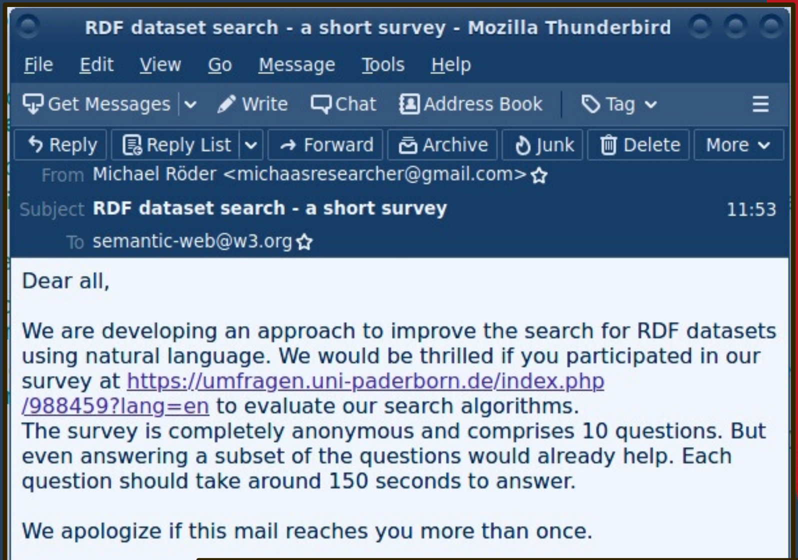
Summary of learning outcomes

Analysis of concurrent processes:

- What are the **events** that can occur in your system?
- What causal **dependency** (also, “linking”) is there between events?
- What **synchronicity** requirements (“coupling”) are there?
- Simple representation of concurrency: **Hasse diagram**

Analysis of Petri nets:

- What are the possible **firing sequences** in a Petri net?
- What transitions can be fired **concurrently**, and when is that the case?
- **Hasse diagram** representation of the firing sequence
- Is the Petri net **safe**? Is it **bounded**? Does it have a **deadlock**?
- How to **construct a Petri net** that represents a given concurrent process


RDF dataset search - a short survey - Mozilla Thunderbird
File Edit View Go Message Tools Help
Get Messages
Write
Chat
Address Book
Tag
Reply
Reply List
Forward
Archive
Junk
Delete
More

From Michael Röder <michaasresearcher@gmail.com>

Subject **RDF dataset search - a short survey** 11:53

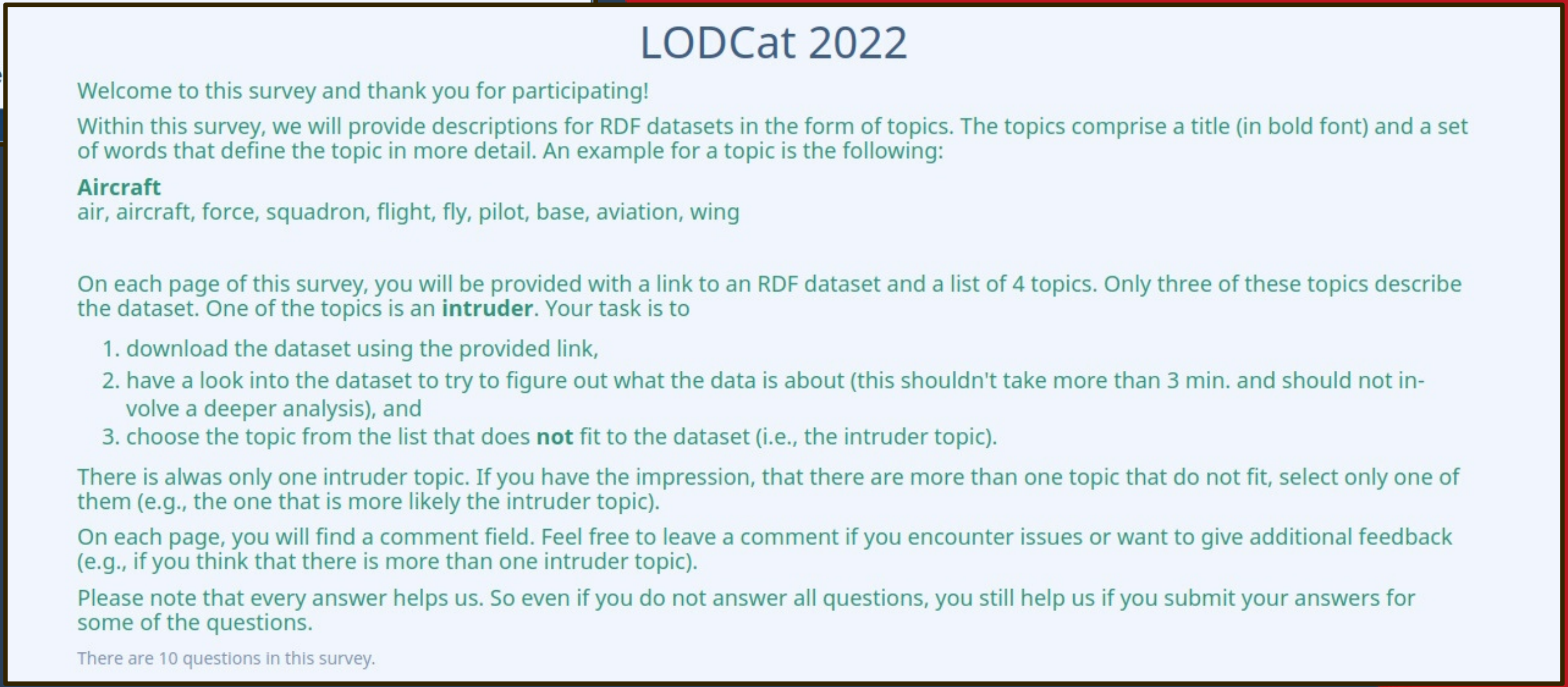
To semantic-web@w3.org

Dear all,

We are developing an approach to improve the search for RDF datasets using natural language. We would be thrilled if you participated in our survey at https://umfragen.uni-paderborn.de/index.php/988459?lang=en to evaluate our search algorithms.
The survey is completely anonymous and comprises 10 questions. But even answering a subset of the questions would already help. Each question should take around 150 seconds to answer.

We apologize if this mail reaches you more than once.

Yours sincerely,
Michael Röder
Paderborn Unive



LODCat 2022

Welcome to this survey and thank you for participating!

Within this survey, we will provide descriptions for RDF datasets in the form of topics. The topics comprise a title (in bold font) and a set of words that define the topic in more detail. An example for a topic is the following:

Aircraft
air, aircraft, force, squadron, flight, fly, pilot, base, aviation, wing

On each page of this survey, you will be provided with a link to an RDF dataset and a list of 4 topics. Only three of these topics describe the dataset. One of the topics is an **intruder**. Your task is to

1. download the dataset using the provided link,
2. have a look into the dataset to try to figure out what the data is about (this shouldn't take more than 3 min. and should not involve a deeper analysis), and
3. choose the topic from the list that does **not** fit to the dataset (i.e., the intruder topic).

There is alwas only one intruder topic. If you have the impression, that there are more than one topic that do not fit, select only one of them (e.g., the one that is more likely the intruder topic).

On each page, you will find a comment field. Feel free to leave a comment if you encounter issues or want to give additional feedback (e.g., if you think that there is more than one intruder topic).

Please note that every answer helps us. So even if you do not answer all questions, you still help us if you submit your answers for some of the questions.

There are 10 questions in this survey.

University of Central Lancashire
UCLan

CO3409
Distributed Enterprise Systems

Summary: Semantic web
Summary: Semantic querying
Summary: Concurrent process models

Where opportunity creates success