University of Central Lancashire
UCLan

Logic-based ontological reasoning for NP-hard problems

Martin Thomas Horsch

Where opportunity creates success

UCLan University of Central Lancashire

What do you see?

Use only simple sentences consisting of:

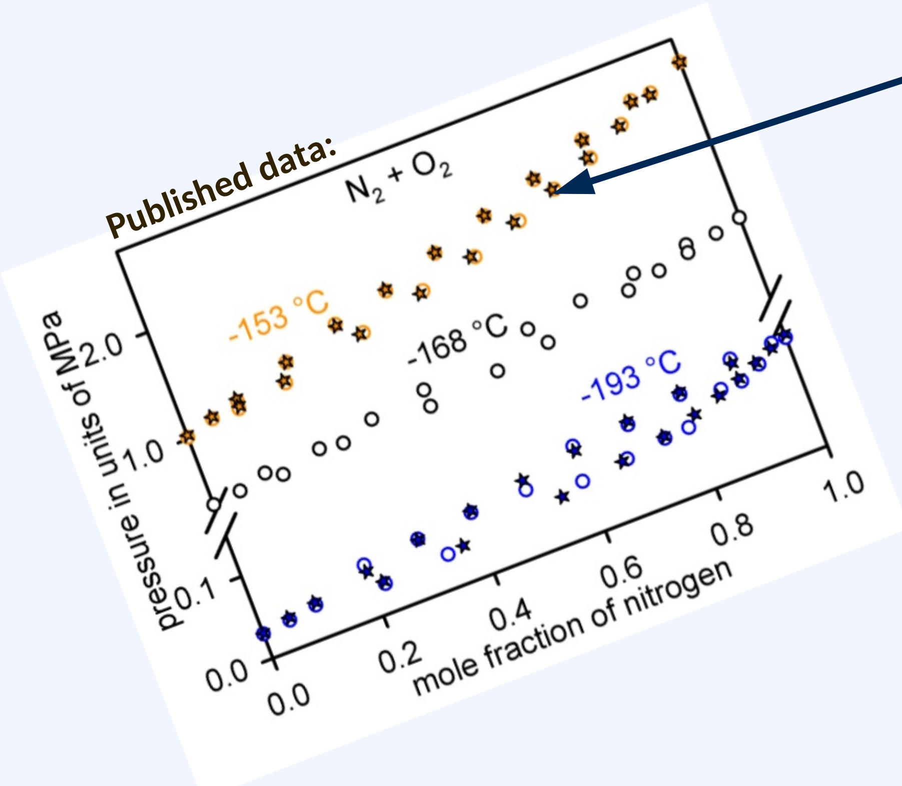
- A **subject**
- A **predicate**
- An **object**

Such as:

"**The-elephant** **is-dancing-in** **the-room**."

"**The-wheel** **is-part-of** **the-car**."

FAIR research data



What values did *x* and *p* have?

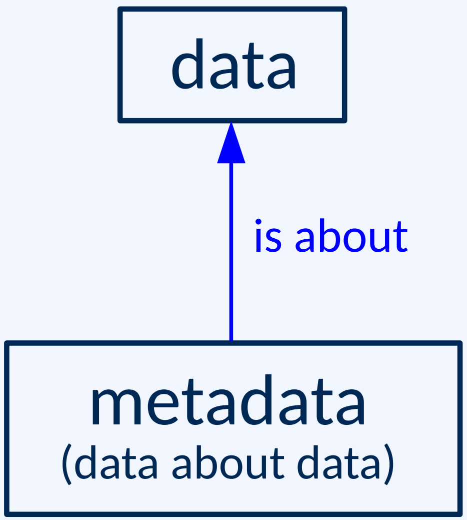
How was the data point obtained?

What is the margin of error, how was the error defined, and what software (or experimental setup) was used?

Good practice in managing research data:

Make all data findable, accessible, interoperable, and reusable (FAIR).

FAIR research data and metadata



What values did *x* and *p* have?

How was the data point obtained?

What is the margin of error, how was the error defined, and what software (or experimental setup) was used?

Competency questions:

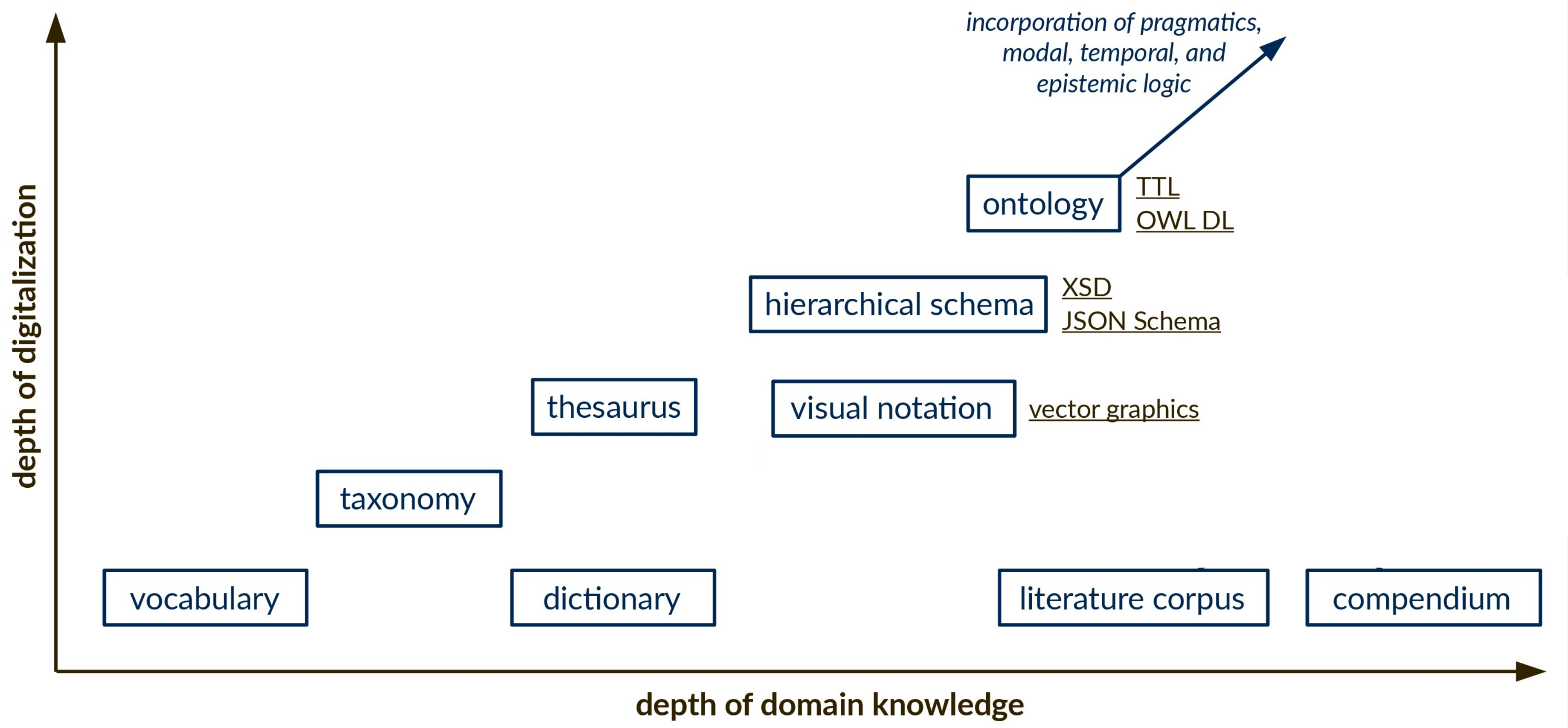
Representative queries (requests) for metadata, to be competently answered by a data infrastructure.

Good practice in managing research data:

Make all data findable, accessible, interoperable, and reusable (FAIR).

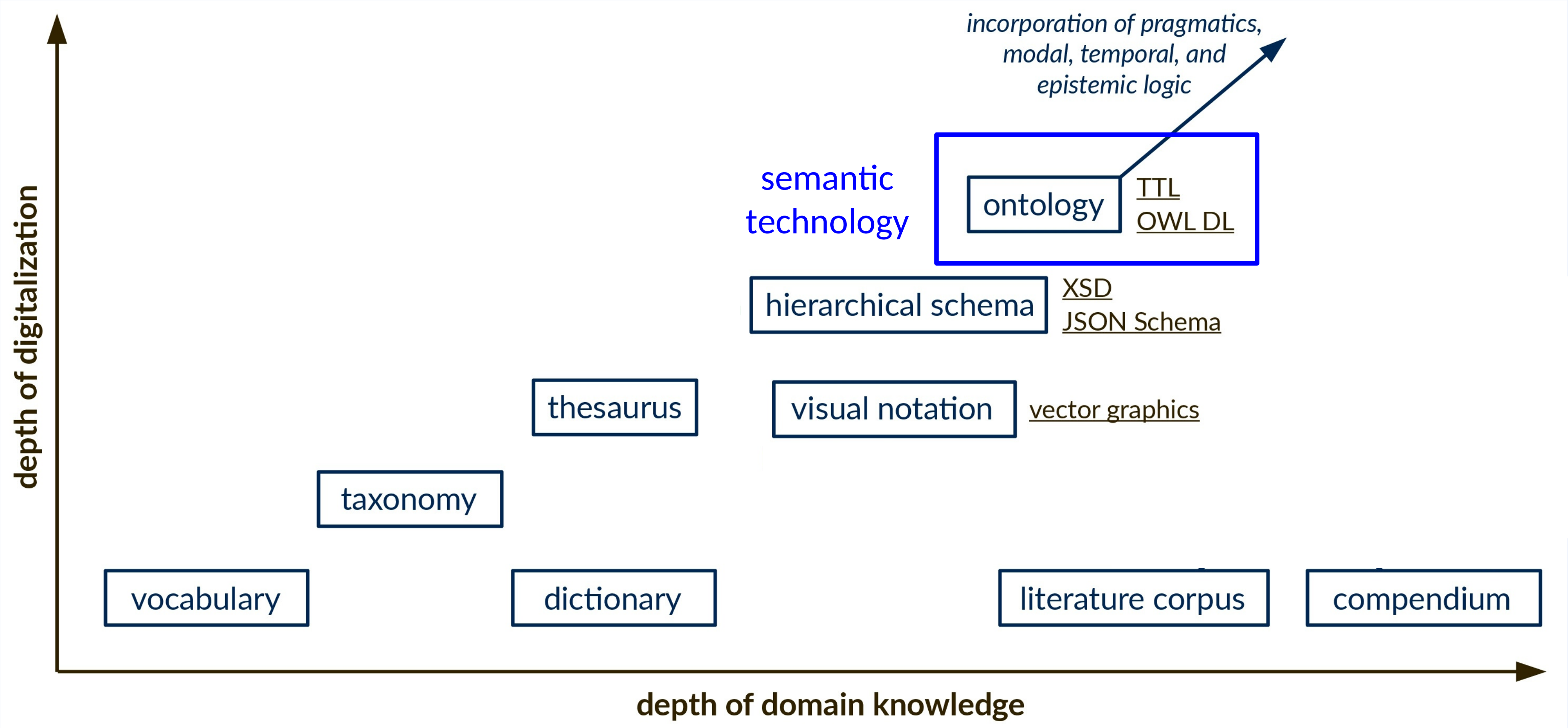
Metadata standardization

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



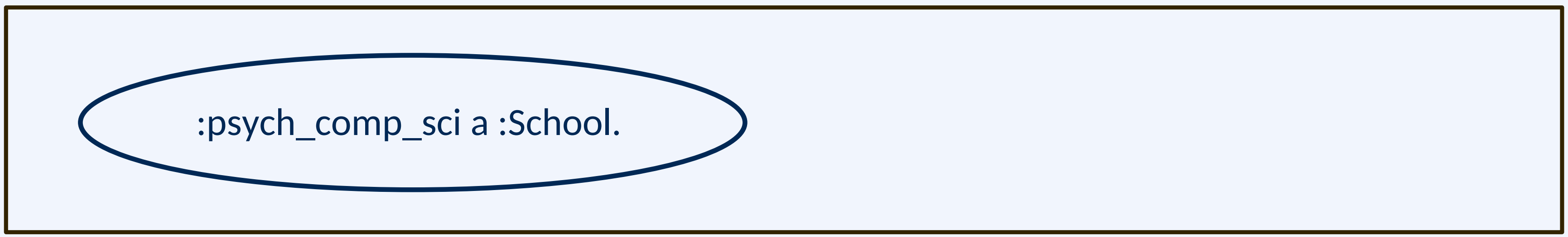
Metadata standardization

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



Knowledge graphs

Modern knowledge bases represent knowledge about the state of affairs as **knowledge graphs**, relying on semantic technology including RDF and OWL.



Knowledge graph; also: Scenario or assertional box (ABox)

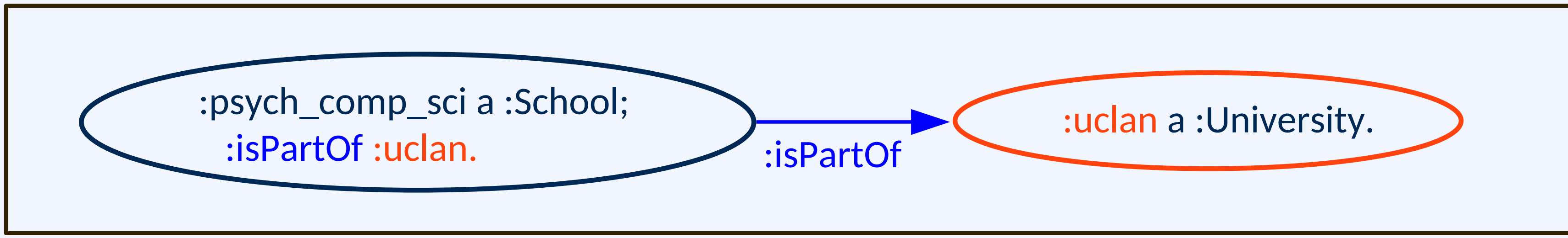
Semantics (*i.e.*, meaning) of the graph above: “The School of Psychology and Computer Science is a school.”

Interactions with the knowledge base take two forms:

- **Data ingest** (“tell”) to extend or update the information about the world.
- **Data retrieval** based on **querying** (“ask”).

Knowledge graphs

Modern knowledge bases represent knowledge about the state of affairs as **knowledge graphs**, relying on semantic technology including RDF and OWL.



Knowledge graph; also: Scenario or assertional box (ABox)

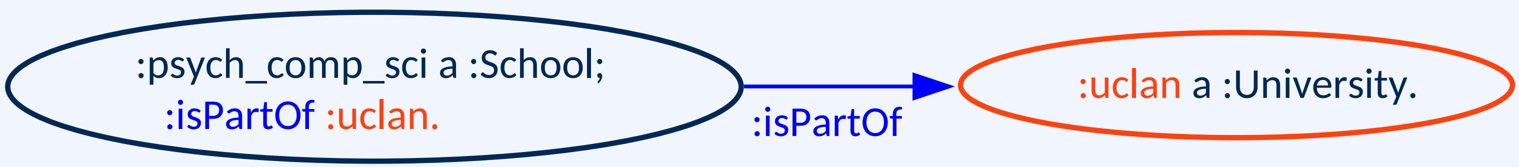
Semantics (*i.e.*, meaning) of the graph above: “The School of Psychology and Computer Science is a school. It is part of UCLan which is a university.”

Interactions with the knowledge base take two forms:

- **Data ingest** (“tell”) to extend or update the information about the world.
- **Data retrieval** based on **querying** (“ask”).

Knowledge graphs

Modern knowledge bases represent knowledge about the state of affairs as **knowledge graphs**, which can be decomposed into simple statements: **Triples**.



Knowledge graph; also: Scenario or assertional box (ABox)

:psych_comp_sci :isPartOf :uclan.

RDF triple, consisting of subject, predicate, and object

Interactions with the knowledge base take two forms:

- **Data ingest** ("tell") to extend or update the information about the world.
- **Data retrieval** based on **querying** ("ask").

Terse triple language (TTL)

Terse triple language, also known as **turtle format**, is a compact notation for triples that is easy to write in a text editor.

RDF: Resource Description Framework

TTL format

RDF triples

subject	a	class_of_subject;
	has_property	first_object, second_object;
	other_property	another_object.

Terse triple language (TTL)

Terse triple language, also known as **turtle format**, is a compact notation for triples that is easy to write in a text editor.

RDF: Resource Description Framework

TTL format

RDF triples

```
:CERTIFICATE a vivo:certificate;
  vivo:states_assessment :ASSESSMENT;
  vivo:has_certifier :CERTIFIER.
```

":CERTIFICATE is a certificate.
It states an assessment, namely, :ASSESSMENT.
It has a certifier, namely, :CERTIFIER."

```
subject   a                class_of_subject;
          has_property     first_object, second_object;
          other_property   another_object.
```

Internationalized resource identifiers (IRIs)

In the Resource Description Framework (RDF), all **individuals** (objects), **relations** (properties), and **concepts** (classes) are regarded as resources.

IRIs as resource identifiers

prefix:suffix

The prefix acts like a namespace. In TTL format, it may be empty, as in ":CERTIFICATE".

RDF triples in TTL format

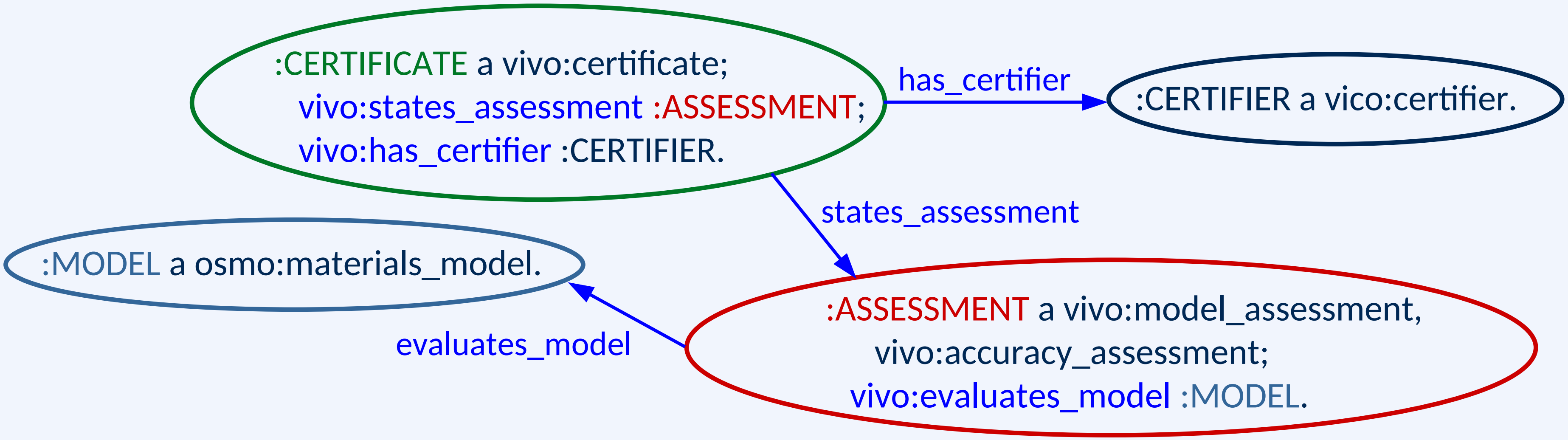
```
:CERTIFICATE a vivo:certificate;
  vivo:states_assessment :ASSESSMENT;
  vivo:has_certifier :CERTIFIER.
```

":CERTIFICATE is a certificate.
It states an assessment, namely, :ASSESSMENT.
It has a certifier, namely, :CERTIFIER."

These short prefixes act as abbreviations for the full first part of the IRI:

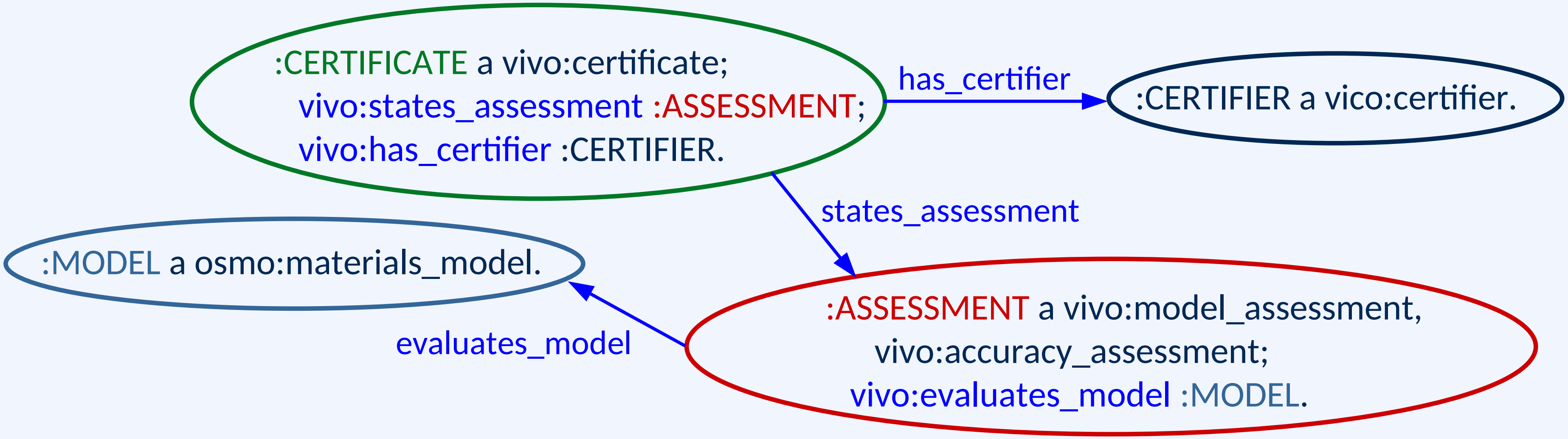
```
@prefix vivo: <https://purl.vimmp.eu/semantics/vivo/vivo.ttl#>.
```

Knowledge graphs



The certifier :CERTIFIER has issued a certificate (the IRI of which is :CERTIFICATE) stating a model accuracy assessment (the IRI of which is :ASSESSMENT) that evaluates the materials model :MODEL.

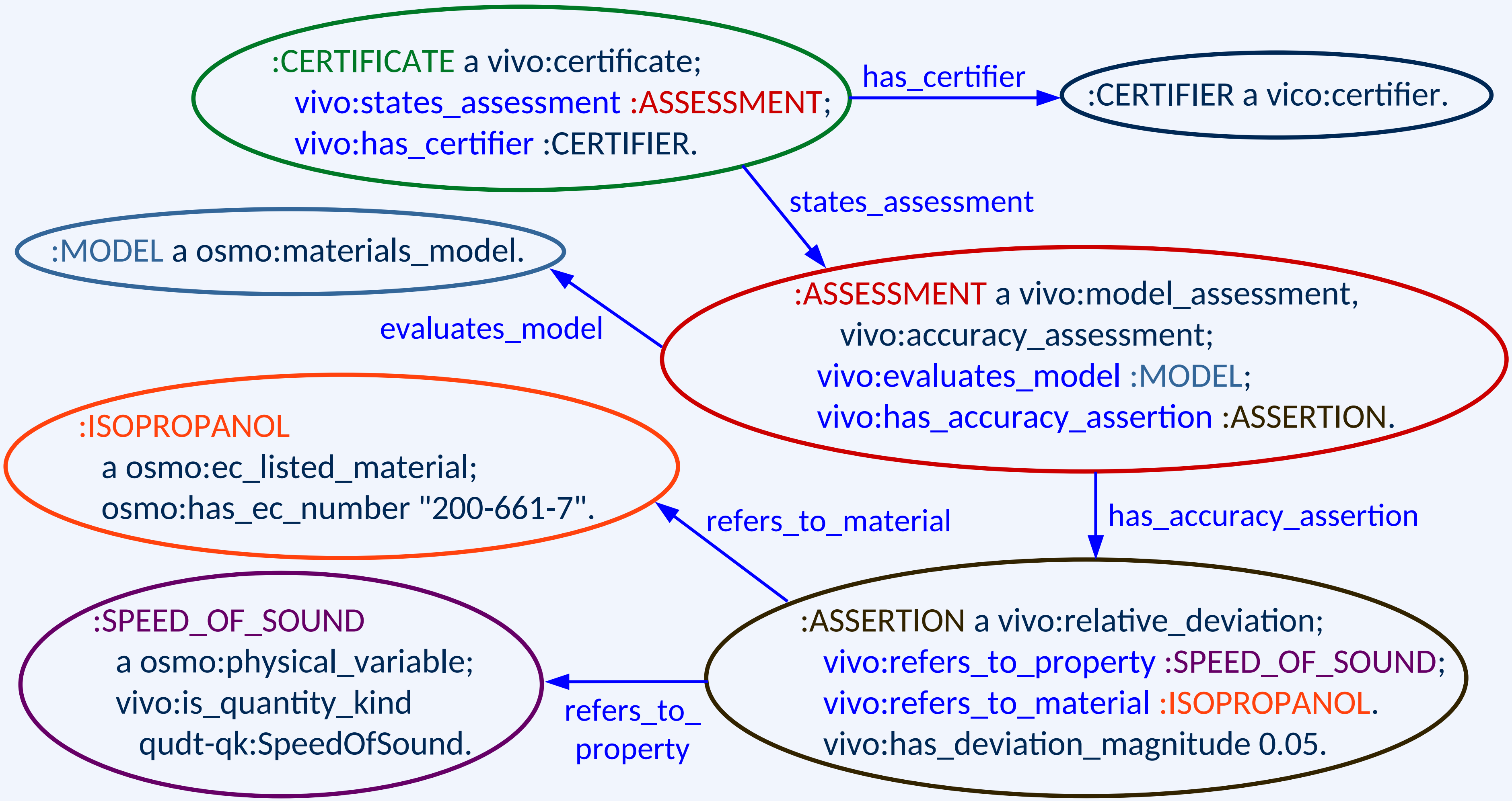
Knowledge graphs



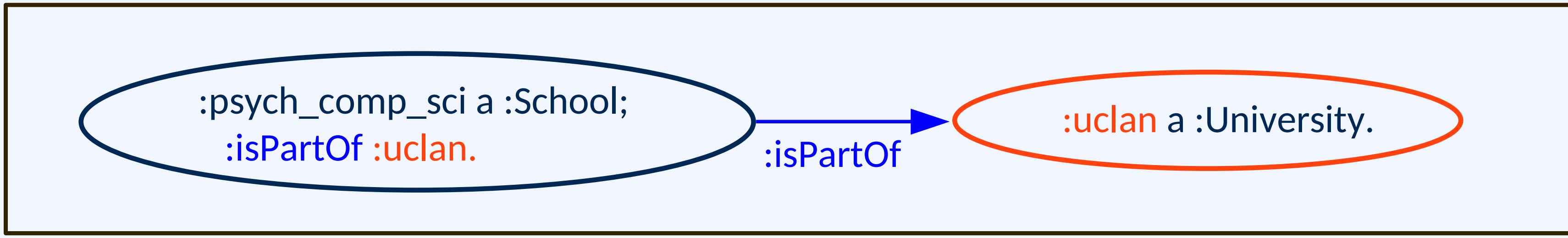
The certifier :CERTIFIER has issued a certificate (the IRI of which is :CERTIFICATE) stating a model accuracy assessment (the IRI of which is :ASSESSMENT) that evaluates the materials model :MODEL.

In the model accuracy assessment :ASSESSMENT, it is asserted that the materials model :MODEL has a relative error of 5% for the speed of sound of isopropanol.

Knowledge graphs



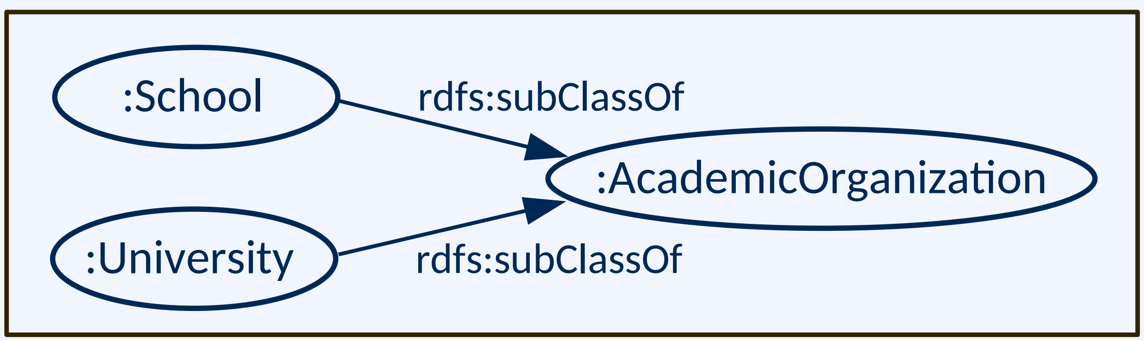
Parts of a knowledge base: ABox and TBox



Knowledge graph; also: Scenario or assertional box (ABox)

:psych_comp_sci :isPartOf :uclan.

RDF triple, consisting of subject, predicate, and object



Taxonomy, part of the ontology

also: TBox (terminological box)

The semantic web

Semantic technology can facilitate the integration of data and software into a coherent framework, permitting multiple components to become interoperable. **On the semantic web, data and metadata are provided in the form of triples:**

Triples: Individual Relation Individual. (Subject Predicate Object.)

Example: theFox eats theChicken.

The semantic web

Semantic technology can facilitate the integration of data and software into a coherent framework, permitting multiple components to become interoperable. **Ontologies characterize individuals (*i.e.*, objects), the concepts (*i.e.*, classes) to which they belong, the possible relations between them, and applicable restrictions (rules).**

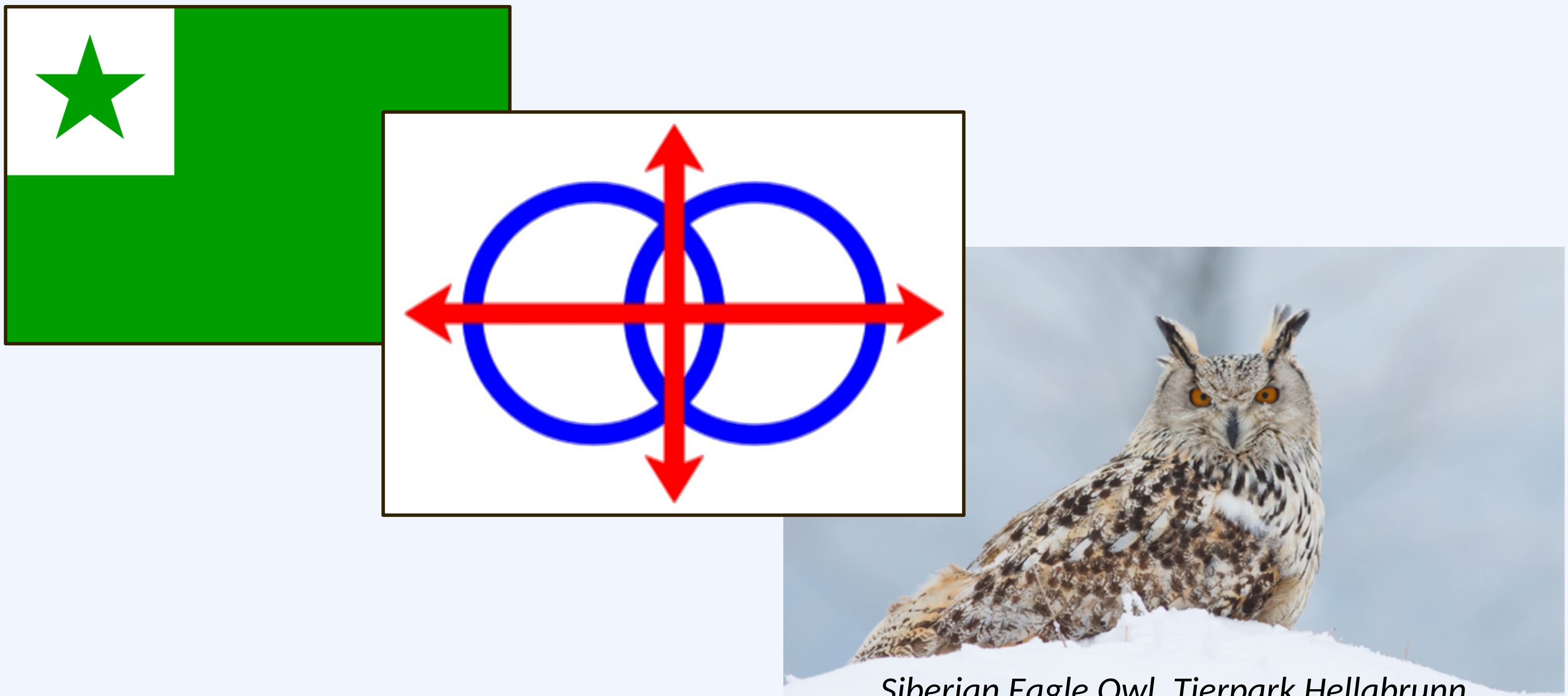
Triples: Individual Relation Individual. (Subject Predicate Object.)

Example: theFox eats theChicken.

- Resource description framework (RDF): Formalism for specifying triples.
- Web ontology language (OWL): Formalism for specifying ontologies, including rules that can be processed by automated reasoning.
- Terse triple language (TTL): Common syntax for denoting triples from RDF and OWL.

Web Ontology Language (OWL)

„One World Language“



Siberian Eagle Owl, Tierpark Hellabrunn

Web Ontology Language (OWL)

Ontology and reasoning are always **logic-based**. Hence, the title of this talk, "logic-based ontological reasoning for NP-hard problems," is redundant.

Web Ontology Language (OWL) is a logical framework for specifying ontologies. It is based on a **description logic (DL)** known as OWL DL.

animals:Fox a owl:Class.

taxonomy specified using rdfs:subClassOf

concepts are of the type owl:Class

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

Web Ontology Language (OWL)

University of Central Lancashire UCLan

Ontology and reasoning are always **logic-based**. Hence, the title of this talk, "logic-based ontological reasoning for NP-hard problems," is redundant.

Web Ontology Language (OWL) is a logical framework for specifying ontologies. It is based on a **description logic (DL)** known as OWL DL.

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.
animals:isNaturalEnemyOf rdfs:domain animals:Predator.

relations are of the type owl:ObjectProperty

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

Web Ontology Language (OWL)

Ontology and reasoning are always **logic-based**. Hence, the title of this talk, “logic-based ontological reasoning for NP-hard problems,” is redundant.

Web Ontology Language (OWL) is a logical framework for specifying ontologies. It is based on a **description logic (DL)** known as OWL DL.

What did you see?

One approach to designing ontologies consists in **describing example scenarios**.

Usually, different people describe the same scenario in different terms, causing **semantic heterogeneity**.

Reasoning over triples

Triples: Individual Relation Individual. (Subject Predicate Object.)

(1) Frank is_father_of Robert.

Q: "Is Daniel the father of Robert?"

Human is a concept.
Robert is a Human.

Cardinality restriction:
Every Human has exactly one father.

Reasoning over triples

Triples: Individual Relation Individual. (Subject Predicate Object.)

(1) Frank is_father_of Robert.

Human is a concept.
Robert is a Human.

Q: "Is Daniel the father of Robert?"
A: "We don't know."

Cardinality restriction:
Every Human has exactly one father.

Principle: Non-unique name assumption

Unless stated otherwise, it is possible that multiple names refer to the same thing.

This is useful for data integration from different sources, using multiple namespaces:

first-knowledge-base:Methane is_the_same_as second-knowledge-base:CH4.

Reasoning over triples

Triples: Individual Relation Individual. (Subject Predicate Object.)

(1) Frank is_father_of Robert.

(2) Frank is_different_from Daniel.

Q: "Is Daniel the father of Robert?"
A: "No, he is not."

Human is a concept.
Robert is a Human.

Cardinality restriction:
Every Human has exactly one father.

Principle: Non-unique name assumption

Unless stated otherwise, it is possible that multiple names refer to the same thing.

This is useful for data integration from different sources, using multiple namespaces.

Reasoning over triples

Triples: Individual Relation Individual. (Subject Predicate Object.)

(1) Frank is_father_of Robert.

(2) Frank is_different_from Daniel.

Q: "Is Daniel the father of Robert?"
A: "No, he is not."

Human is a concept.
Robert is a Human.

Cardinality restriction:
Every Human has exactly one father.

(3) Frank is_father_of Anna.

Q: "How many children does Frank have?"
A: "At least one."

"How many different entities X are there such that Frank is_father_of X?"

Principle: Open world assumption

Since relevant information may distributed over the entire semantic web, rather than the presently considered source only, the **available knowledge is assumed to be incomplete**. (Contrast with a closed, monolithic database architechture.)

Reasoning over triples

Triples: Individual Relation Individual. (Subject Predicate Object.)

(1) Frank is_father_of Robert.

(2) Frank is_different_from Daniel.

Q: "Is Daniel the father of Robert?"
A: "No, he is not."

(3) Frank is_father_of Anna.

Q: "How many children does Frank have?"
A: "At least two."

Human is a concept.
Robert is a Human.

Cardinality restriction:
Every Human has exactly one father.

Anna is_different_from Robert.

"How many different entities X are there such that Frank is_father_of X?"

Principle: Open world assumption

Since relevant information may distributed over the entire semantic web, rather than the presently considered source only, the **available knowledge is assumed to be incomplete**. (Contrast with a closed, monolithic database architechture.)

Reasoning example: Anselm's argument

Anselm of Canterbury's ontological "proof of God"

A: "What is your opinion about God?"

B: "God does not exist."

A: "But then what does 'God' mean?"

B: "The greatest being that I can *think* of."

Reasoning example: Anselm's argument

Anselm of Canterbury's ontological "proof of God"

A: "What is your opinion about God?"

B: "God does not exist."

A: "But then what does 'God' mean?"

B: "The greatest being that I can *think* of."

B: "Let us call it G. I think G does not exist."

A: "Can you try to think of a G', just like G, but such that you additionally think that G' exists?"

B: "Yes, but then I would be wrong."

Reasoning example: Anselm’s argument

Anselm of Canterbury‘s ontological “proof of God”

A: “What is your opinion about God?”

B: “God does not exist.”

A: “But then what does ‘God’ mean?”

B: “The greatest being that I can *think* of.”

B: “Let us call it G. I think G does not exist.”

A: “Can you try to think of a G‘, just like G, but such that you additionally think that G‘ exists?”

B: “Yes, but then I would be wrong.”

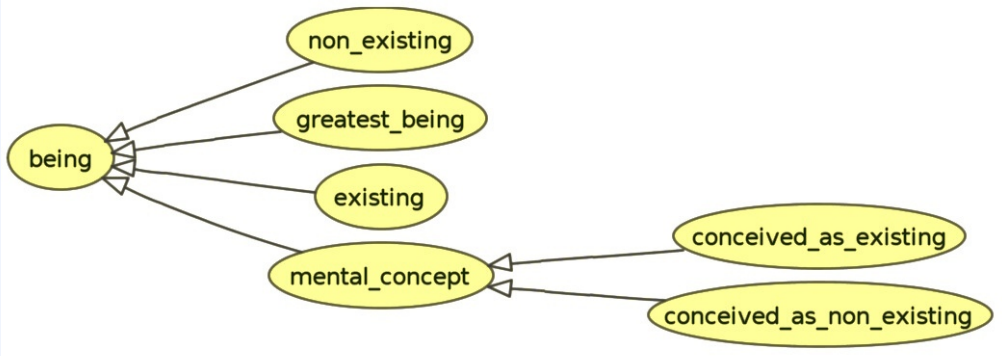
A: “Just do it anyway. Now which is greater?”

B: “Then G‘ is greater because it has existence.”

A: “Then G is not the greatest you can think of.”

Reasoning example: Anselm's argument

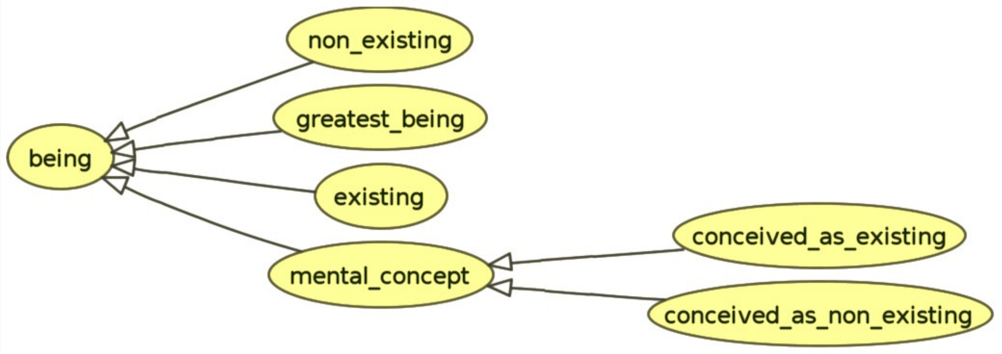
What is "ontological" about this argument?



discusses "what there is" in terms of the **concepts** that individuals (*i.e.*, objects) can instantiate

Reasoning example: Anselm's argument

What is "ontological" about this argument?



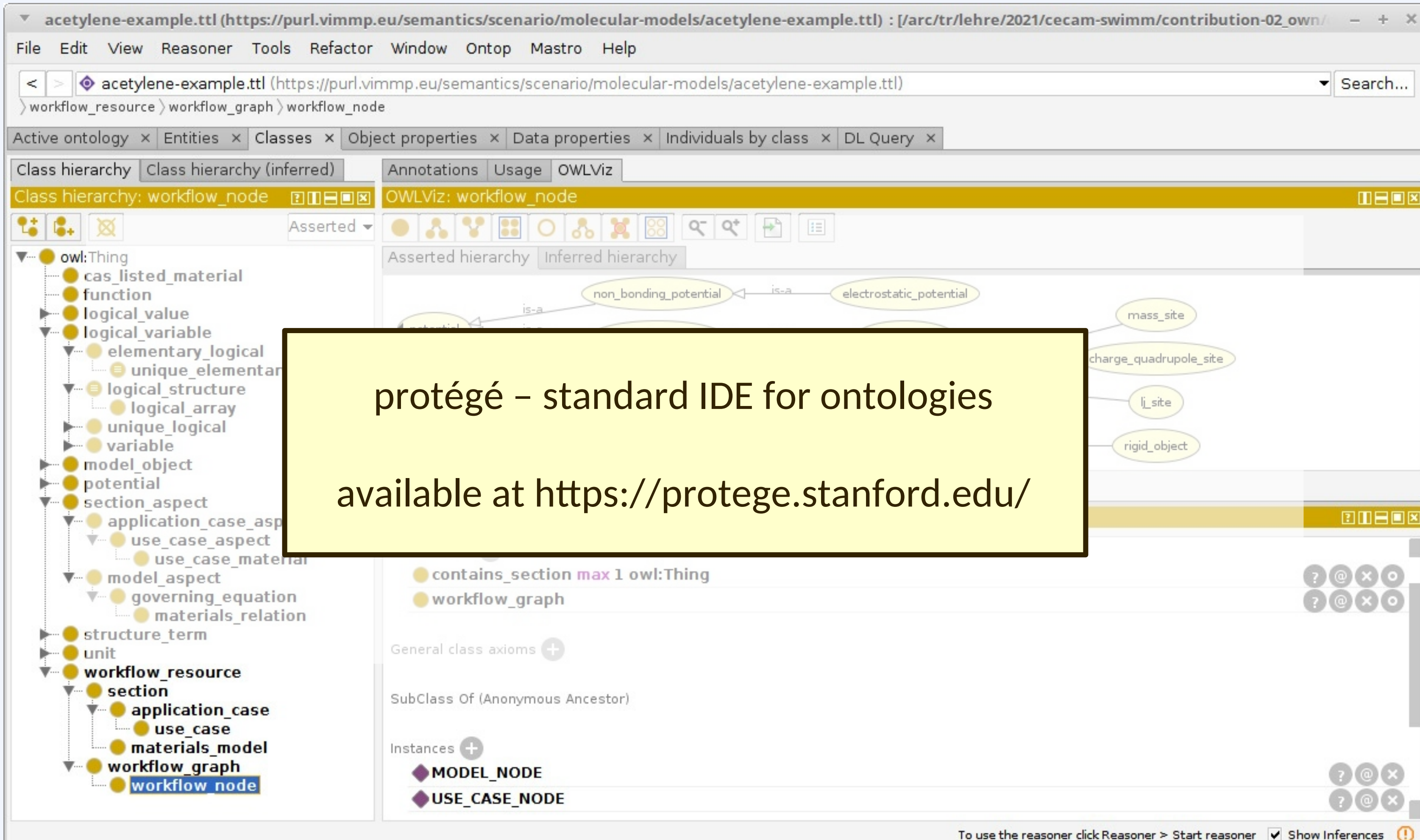
defines **relations** that can hold between pairs of individuals (objects)

cup of coffee thought of as existing **>** *cup of coffee thought of as non-existing*

If I can conceive of something as non-existing, then I can also conceive of something that is similar in all respects except that it also exists.

specifies **rules**

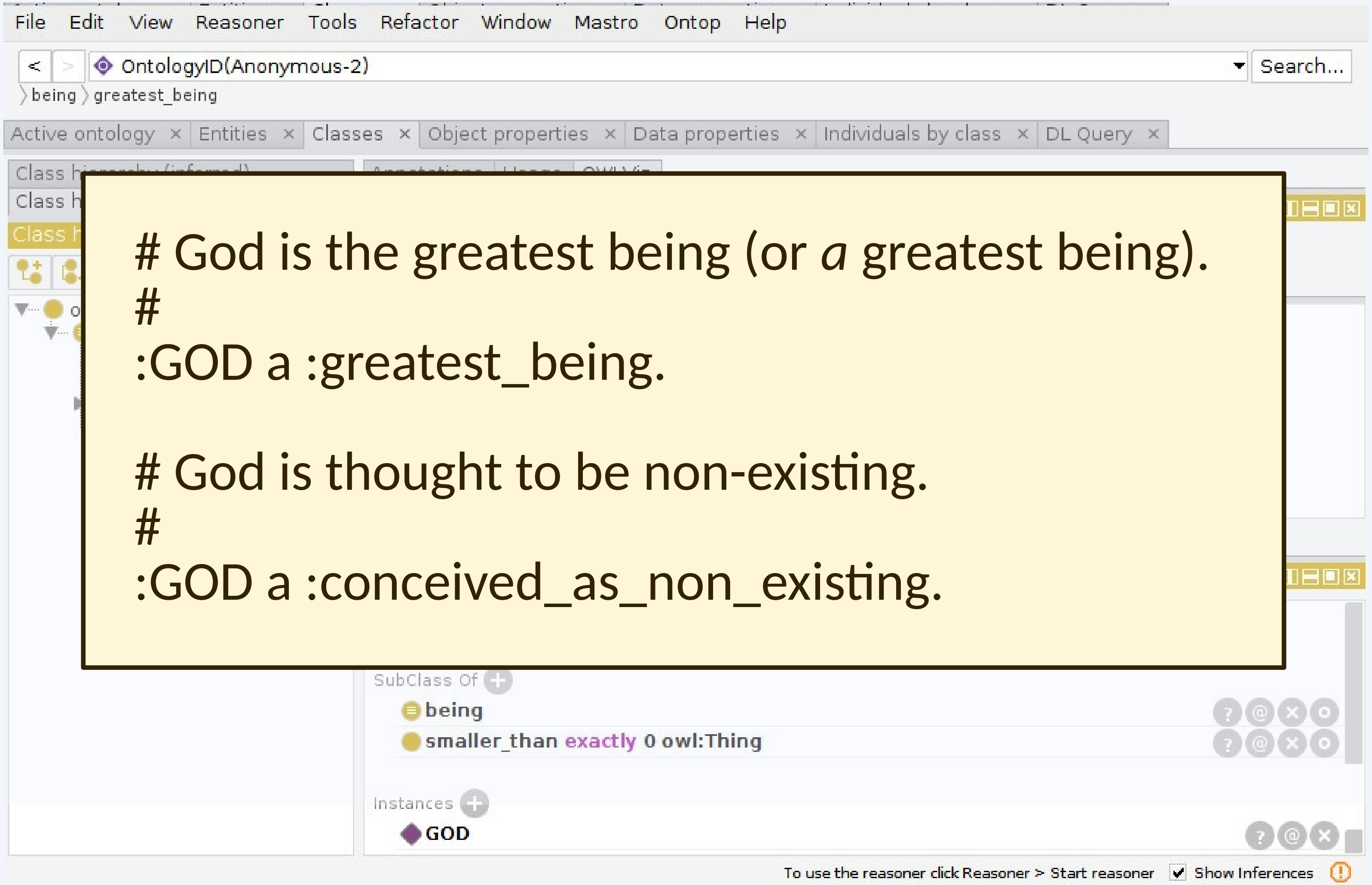
The protégé tool for working with ontologies



protégé – standard IDE for ontologies

available at https://protege.stanford.edu/

The protégé tool for working with ontologies



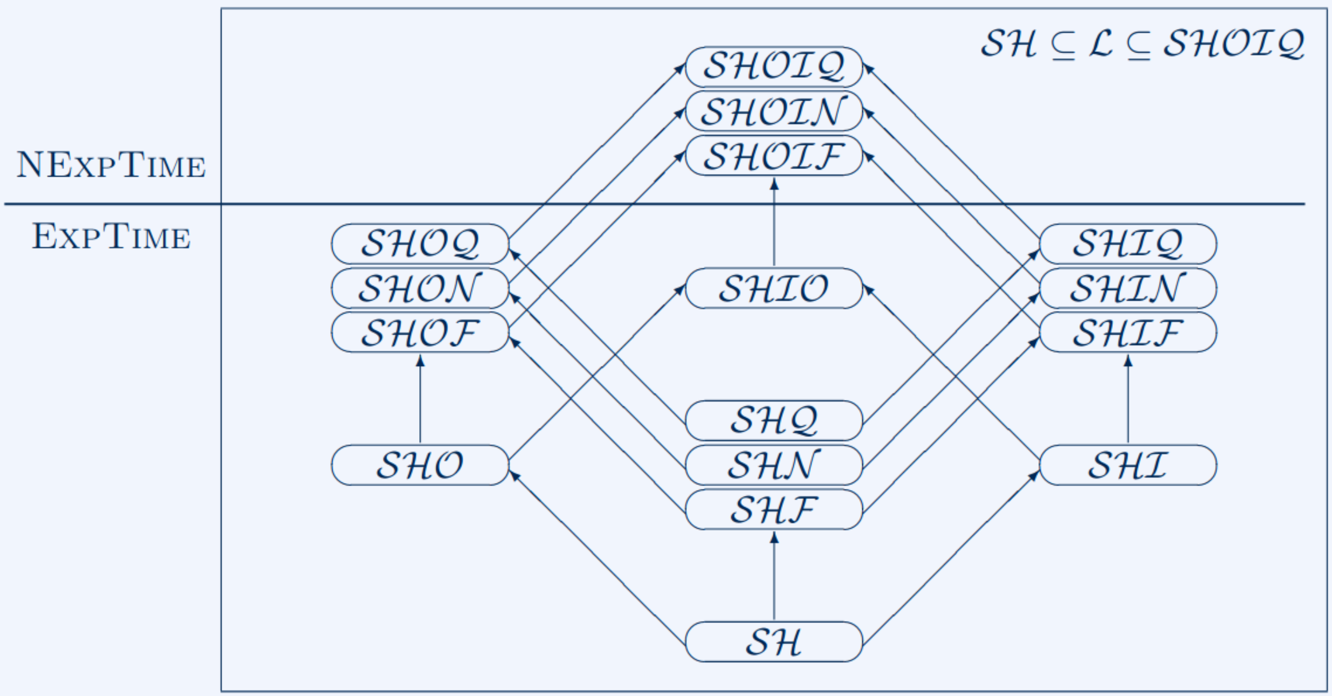
God is the greatest being (or *a* greatest being).
#
:GOD a :greatest_being.

God is thought to be non-existing.
#
:GOD a :conceived_as_non_existing.

Ontologies, reasoning, and NP-hard problems

Lecture title specified as:

"Logic-based ontology reasoning for NP-hard problems."



Complexity of ABox **consistency checking** for various fragments of description logic[1]

[1] E. Zolin, Complexity of reasoning in description logics, http://www.cs.man.ac.uk/~ezolin/dl/.

Ontologies, reasoning, and NP-hard problems

Lecture title specified as:

"Logic-based ontology reasoning for NP-hard problems."

Possible interpretations:

1) There are problems that are NP-hard for other logics, where for ontologies (and OWL description logic), these problems are tractable.
    - Special fragments of OWL are designed for improved computational tractability. However, for the popular fragments, common reasoning problems are still NP-hard.

Ontologies, reasoning, and NP-hard problems

Lecture title specified as:

"Logic-based ontology reasoning for NP-hard problems."

Possible interpretations:

1) There are problems that are NP-hard for other logics, where for ontologies (and OWL description logic), these problems are tractable.

   - Special fragments of OWL are designed for improved computational tractability. However, for the popular fragments, common reasoning problems are still NP-hard.

   - If expressivity is reduced very far (to RDF schema), most reasoning problems are tractable. However, we are not doing anything "for" the problem; we replace it by a simpler one.

Ontologies, reasoning, and NP-hard problems

Lecture title specified as:

"Logic-based ontology reasoning for NP-hard problems."

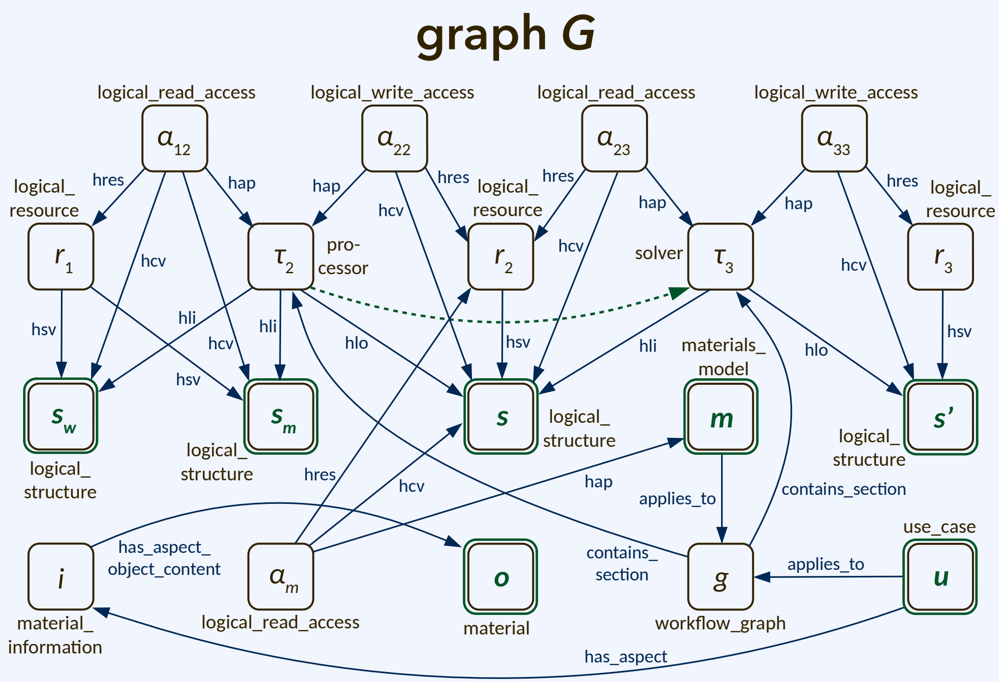
Possible interpretations:

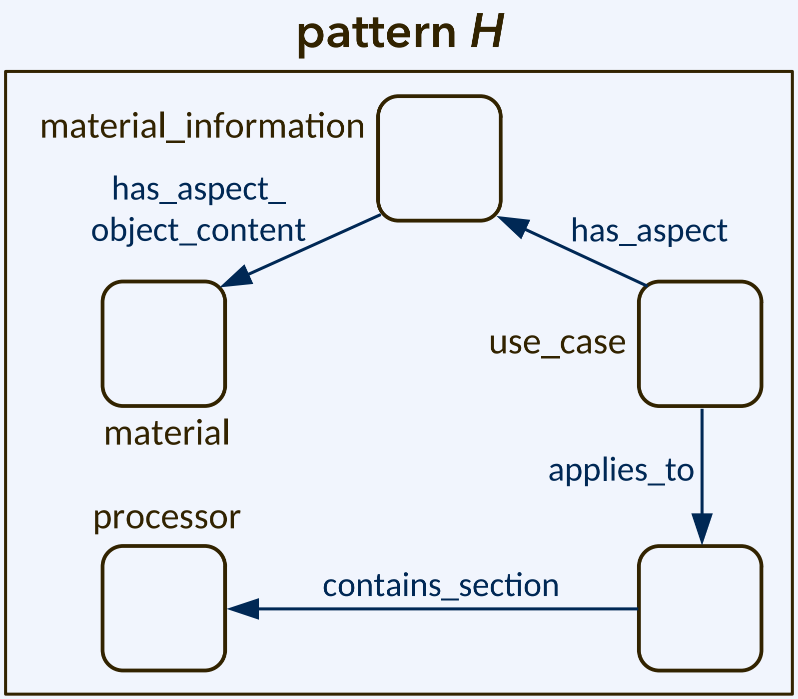
1) There are problems that are NP-hard for other logics, where for ontologies (and OWL description logic), these problems are tractable.
2) There are NP-hard problems with a use case outside ontology, and a technique of solving them is reducing them to a reasoning problem.
   - One such problem is **subgraph matching**. This has use cases in a variety of domains, *e.g.*, where graph transformation is applied.
   - Subgraph matching corresponds to **querying a knowledge base**.

Ontologies, reasoning, and NP-hard problems

Subgraph matching problem (NP-complete):

Given a graph *G* and a pattern *H*, does G contain a subgraph isomorphic to *H*?





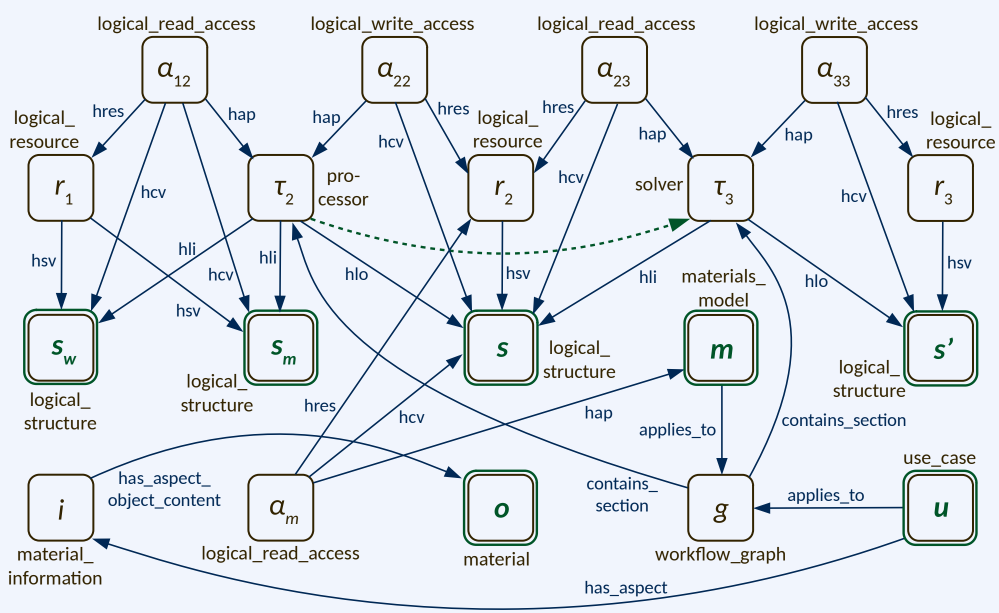
(example from Klein, Preisig, Horsch, Konchakova, in Proceedings of JOWO 2021)

Ontologies, reasoning, and NP-hard problems

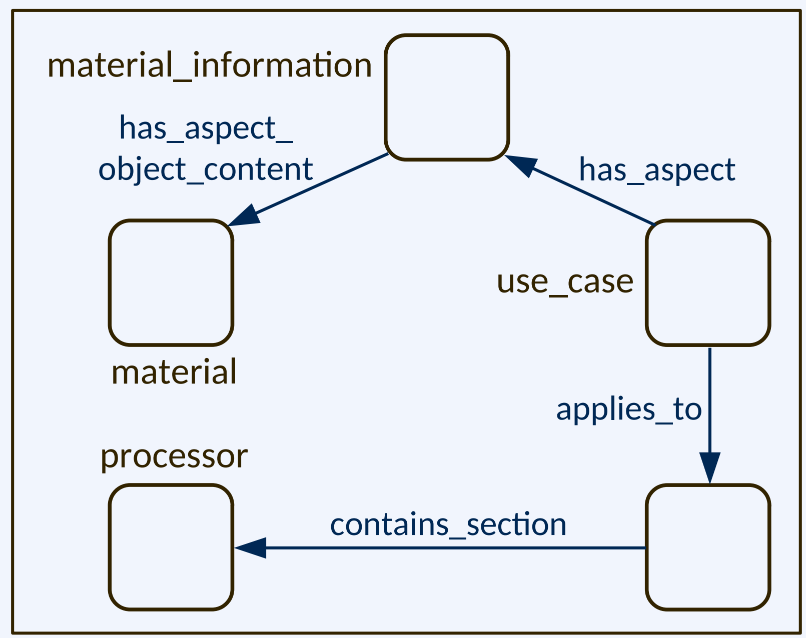
– This problem is closely related to **querying a knowledge base**: What are all the matches to a certain pattern within the knowledge graph?

– It is also related to the *Hamilton cycle* and *travelling salesman* problems.

graph *G*



pattern *H*



(example from Klein, Preisig, Horsch, Konchakova, in Proceedings of JOWO 2021)

Querying a knowledge base using SPARQL

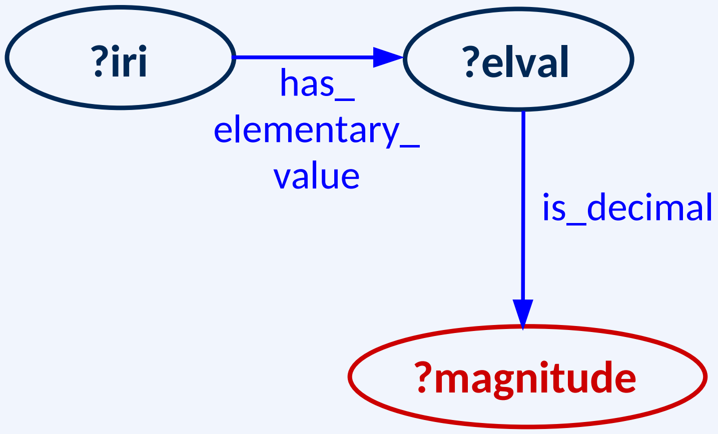
SPARQL („SPARQL Protocol and RDF Query Language“) for the semantic web

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

SELECT **?magnitude**
WHERE {

?iri osmo:has_elementary_value **?elval**.
?elval osmo:is_decimal **?magnitude**.

}

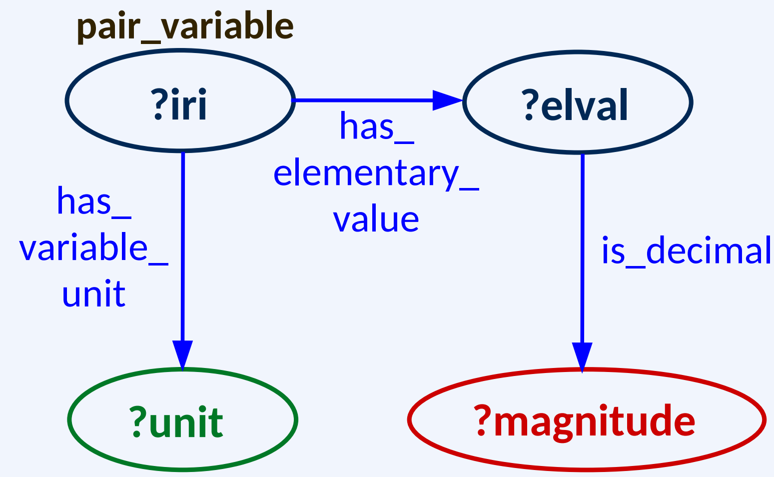


Querying a knowledge base using SPARQL

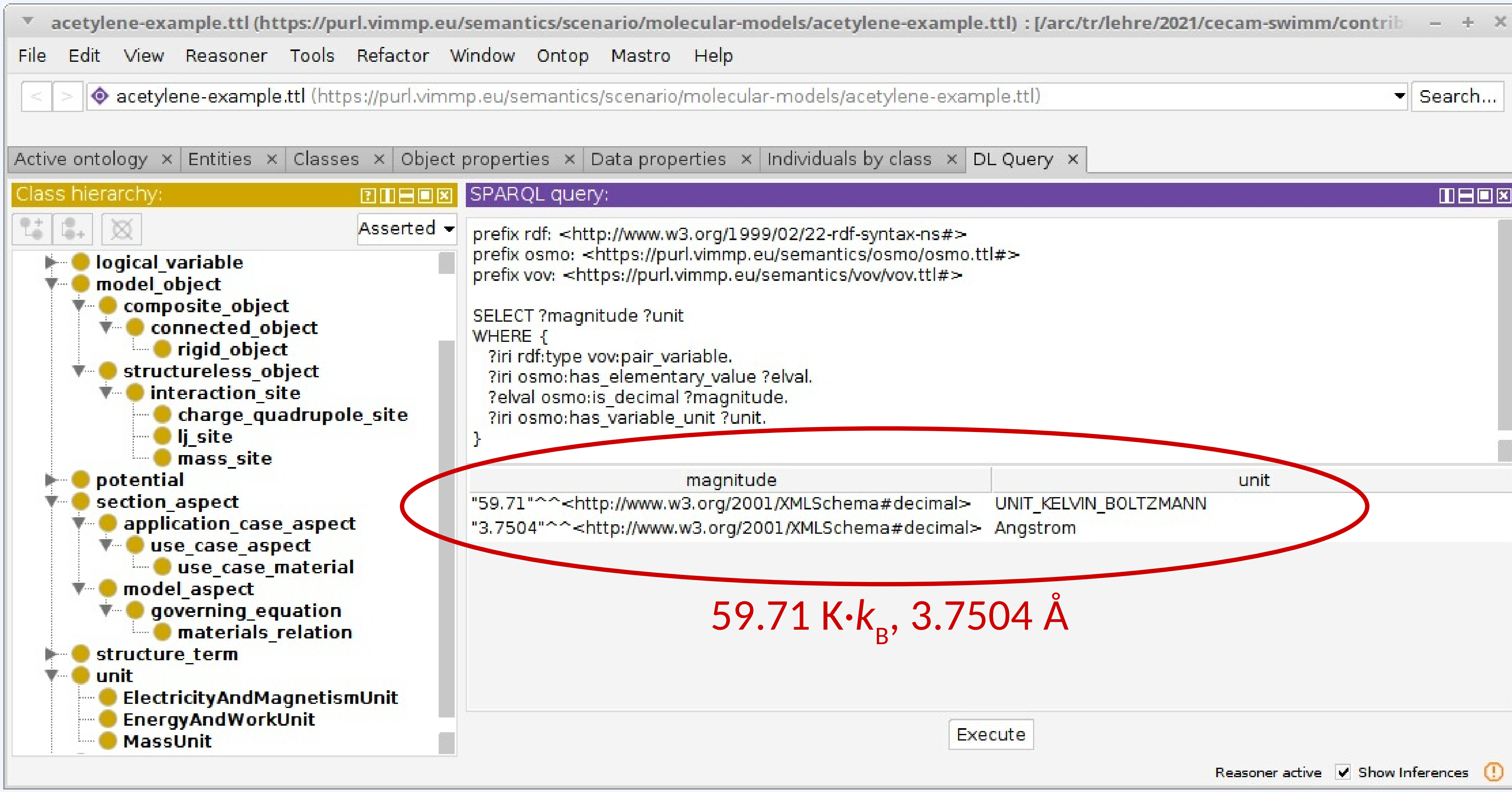
SPARQL („SPARQL Protocol and RDF Query Language“) for the semantic web

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

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



Querying a knowledge base using SPARQL



University of Central Lancashire
UCLan

Logic-based ontological reasoning for NP-hard problems

Martin Thomas Horsch

Where opportunity creates success