KRR: Knowledge Graphs

Artificial Intelligence Challenge / Introduction to Artificial Intelligence

ICM 2A + M1 CPS²

10th March 2023

antoine.zimmermann@emse.fr

Knowledge graphs in general

Definition, from Hogan et al. 2021:

"*a **graph of data** intended to accumulate and convey **knowledge of the real world**, whose **nodes** represent **entities of interest** and whose **edges** represent **relations between these entities***."

Knowledge graphs in general

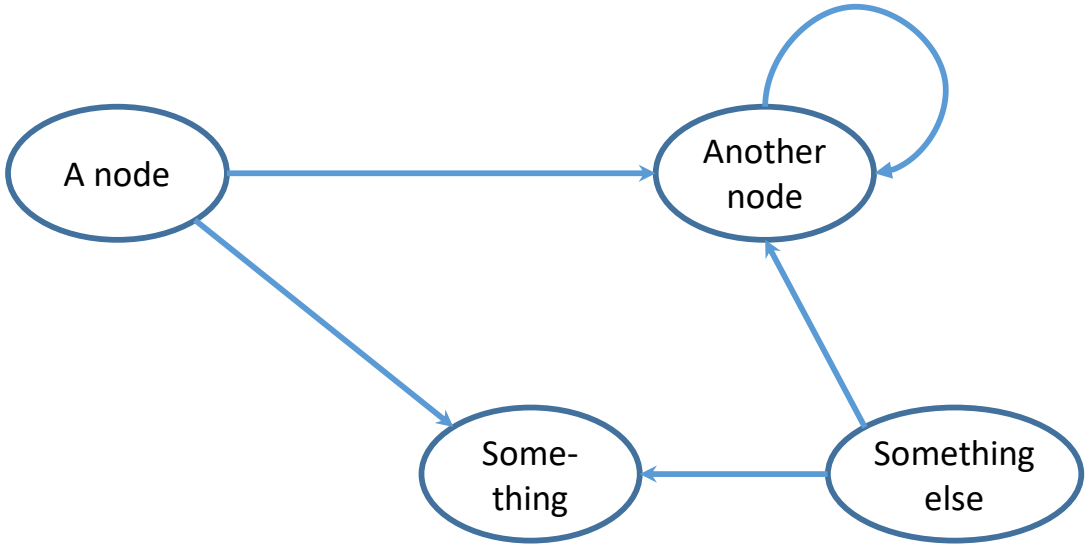
Definition, from Hogan et al. 2021:

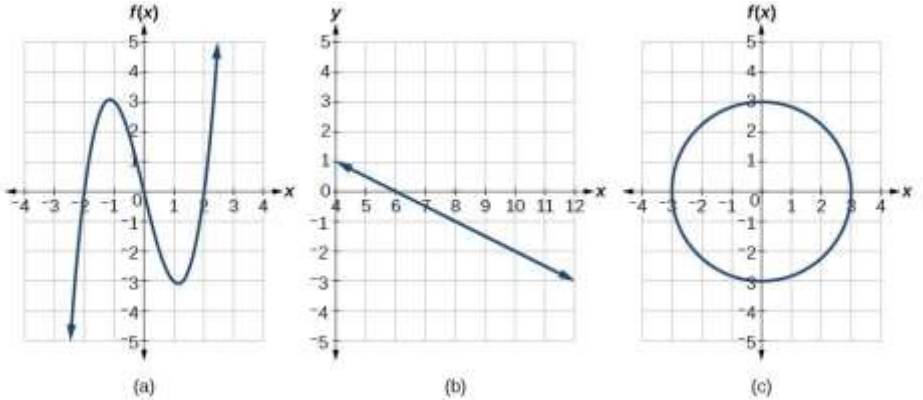
*"a **graph of data** intended to accumulate and convey **knowledge of the real world**, whose **nodes** represent **entities of interest** and whose **edges** represent **relations between these entities**."*

Syntax	Semantics
Graph of data	Knowledge of the real world
nodes	Entities of interest
edges	Relations between entities

A graphs of data

We are talking about graphs in the sense of *graph theory*, not the graphs of functions (a.k.a. plots)

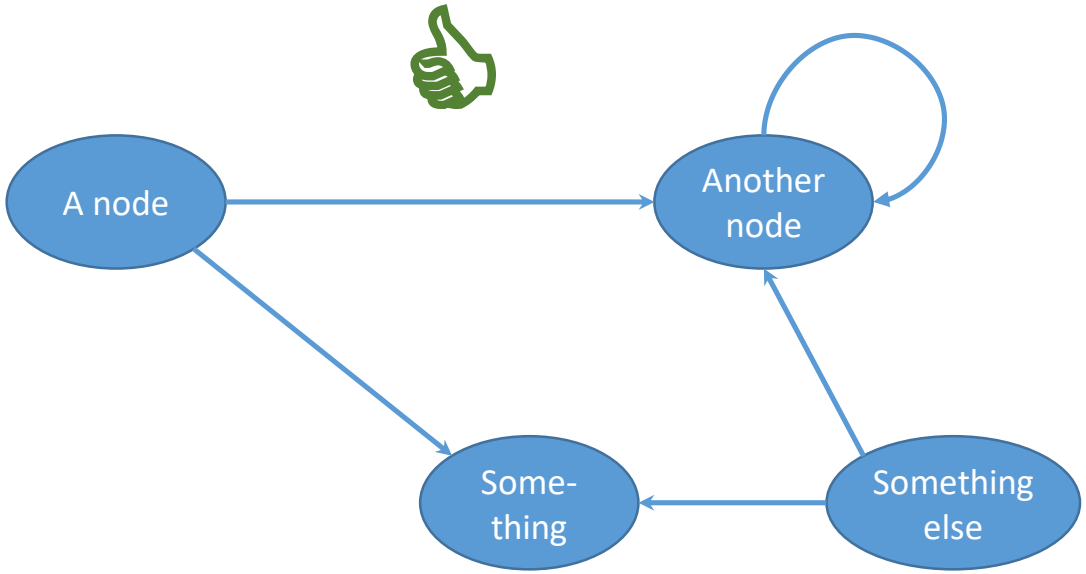


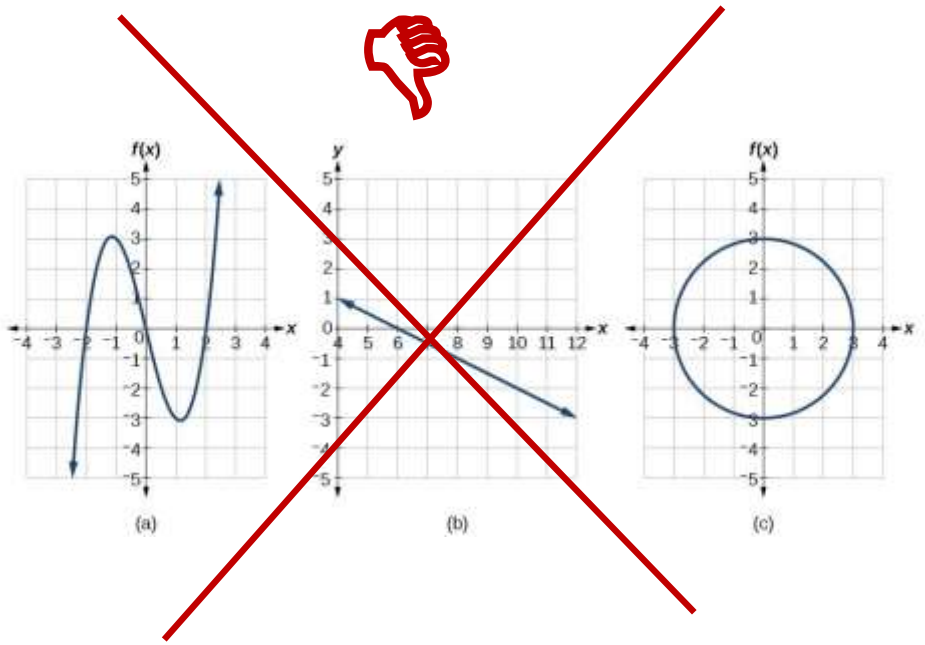


*Relevant section in the KG paper: Section 2.1

A graphs of data

We are talking about graphs in the sense of *graph theory*, not the graphs of functions (a.k.a. plots)





*Relevant section in the KG paper: Section 2.1

Graphs

In *graph theory*, a **graph** is:

- A set of **vertices** (or **nodes**)
- A set of **edges** (lines between nodes)

A **directed graph** has directed edges, or **arcs** (or arrows **from** one node **to** another)

A **multigraph** may have multiple edges between two nodes

A **directed multigraph** may have multiple arcs (in any direction) between nodes)

A **hypergraph** has **hyperarcs** (something that can connect more than 2 nodes)

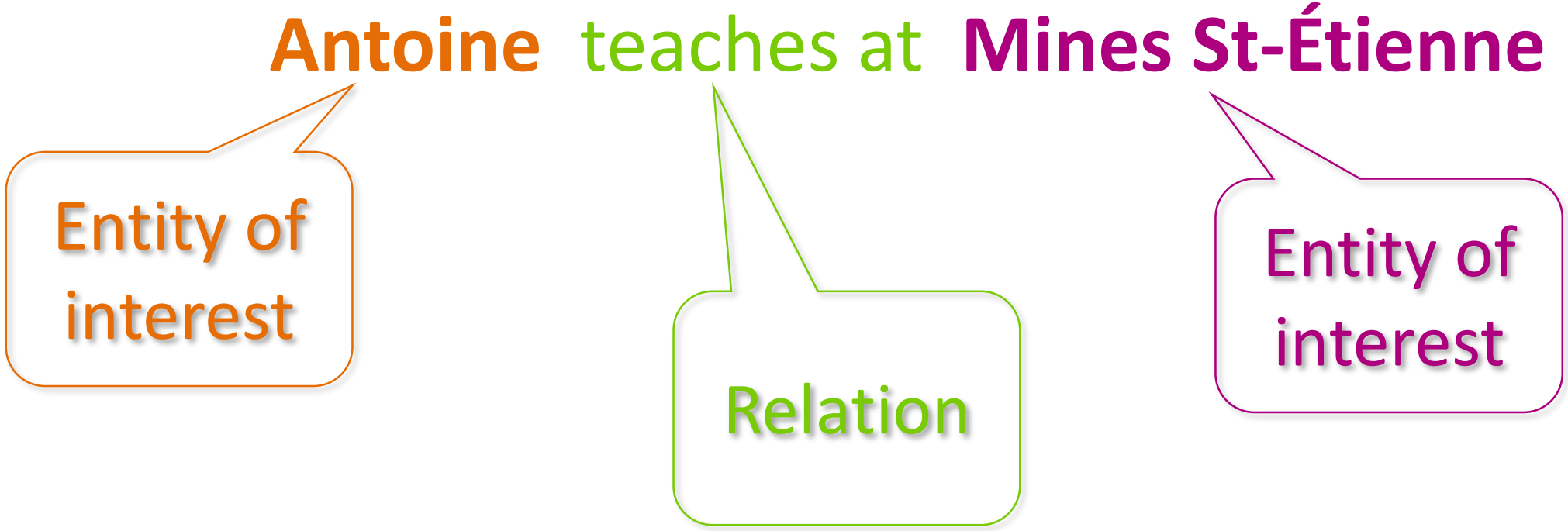
An **edge-labelled graph** assigns a label (usually some text, sometimes a data structure) to edges of an underlying graph

An **edge-labelled directed multigraph** is a directed multigraph in which a label is assigned to each arc

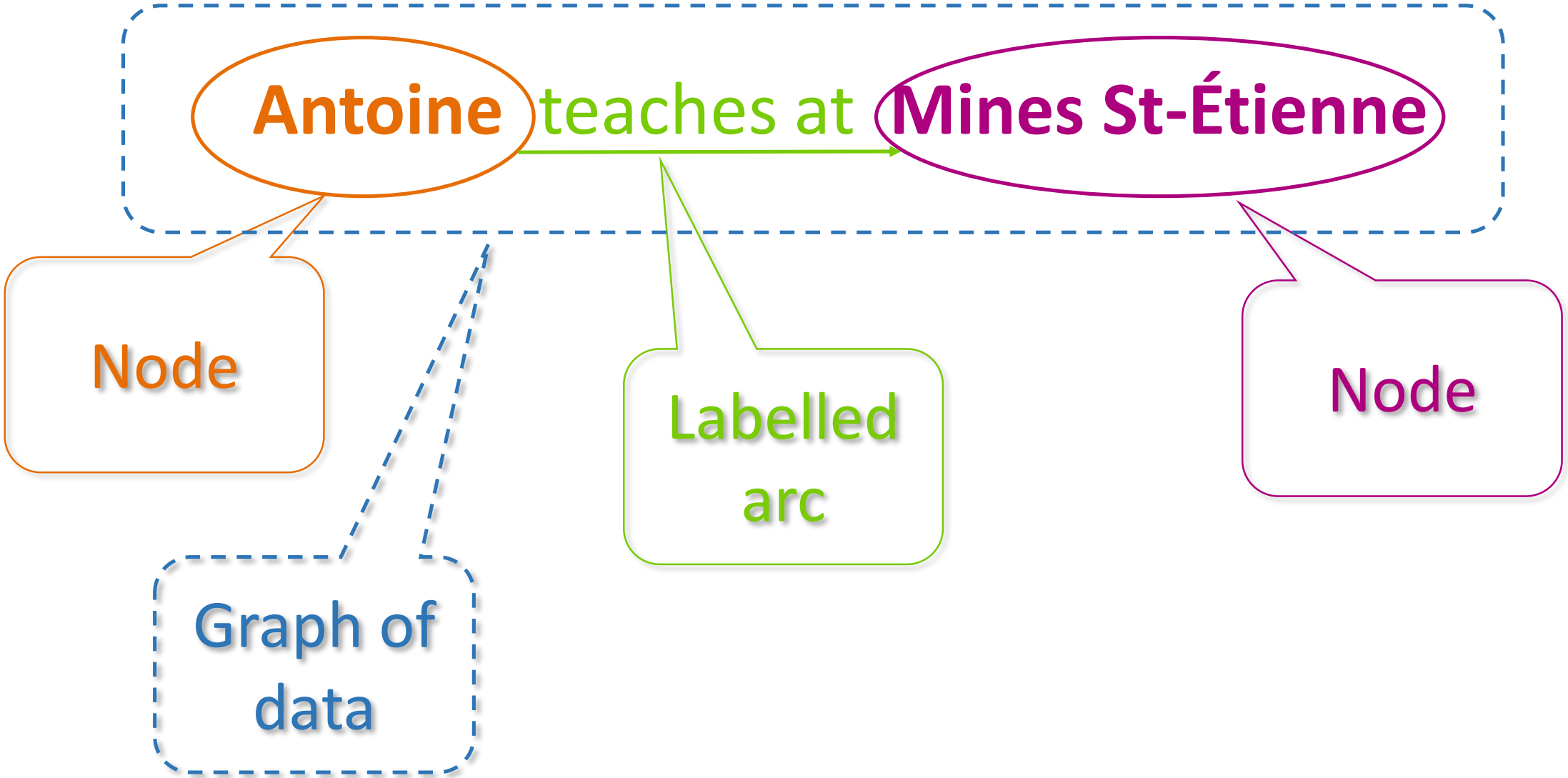
Labelled edges can represent simple facts

Antoine teaches at Mines St-Étienne

Labelled edges can represent simple facts



Labelled edges can represent simple facts



Labelled edges can be represented as **triples**

node	labelled-arc	**node**
source	arrow	**target**
subject	verb	**node**
subject	predicate	**object**

or simply: (s, p, o)

Reminder: Formalising all logics

A logic L is a 4-tuple $(\text{Sign}_L, \text{Form}_L, \text{Int}_L, \models_L)$ where:

- Sign_L is a set of *signatures* (the symbols of the language)
- Form_L is a set of *formulas* (that can be written from symbols)
- Int_L is a set of *interpretations* (that interpret the signatures)
- $\models_L$ is the *satisfaction relation*, such that $\models_L \subseteq \text{Int}_L \times \text{Form}_L$

Knowledge graphs as a **logic** (KG_{log})

- **Symbols** are Nodes, Arcs, and Labels
- **Formulas** are edge-labelled directed multigraphs (= a set of triples)
- **Interpretations** are composed of:
  - a **universe** of interpretation U
  - a function $\mathfrak{I}$ that **maps nodes to elements of the universe**
    (for all $n \in$ Nodes, $\mathfrak{I}(n) \in U$) and
    maps edge-labels to binary relations over the universe
    (for all $a \in$ Labels, $\mathfrak{I}(a) \subseteq U \times U$)
- An interpretation $\mathfrak{I}$ **satisfies** a graph G iff:
  for all $(s, p, o) \in G$, $(\mathfrak{I}(s), \mathfrak{I}(o)) \in \mathfrak{I}(p)$

Entailment

- In any logic *L* defined as before, we say that:

a set of formula *K* entails a formula *f* (in L)

if and only if

all interpretations $\mathfrak{I}$ in Int_L that satisfy all formulas in *K* also satisfy *f*

i.e.:

$$K \; L\text{-entails } f \text{ iff } \forall \mathfrak{I} \in \text{Int}_L, (\forall \alpha \in K, \mathfrak{I} \models_L \alpha) \Rightarrow \mathfrak{I} \models_L f$$

- In this case, we write $K \models_L f$

Entailment

- In KG_{log} defined as before, we say that:

a set of graphs *K* entails a graph *g* (in KG_{log})

if and only if

all interpretations $\mathfrak{I}$ that satisfy all graphs in *K* also satisfy *g*

i.e.:

K KG_{log}-entails *g* iff $\forall \mathfrak{I} \in Int_{KG^{log}}$, $(\forall \alpha \in K, \mathfrak{I} \models_{KG^{log}} \alpha) \Rightarrow \mathfrak{I} \models_{KG^{log}} g$

- In this case, we write $K \models_{KG^{log}} g$

Describe the following situation in KG_{log}

“Mines Saint-Étienne was founded in 1816. Its administrative address is 158 cours Fauriel, in Saint-Étienne, Loire, Auvergne-Rhône-Alpes, France, a country in the EU. The director of Mines Saint-Étienne is Pascal Ray. Antoine Zimmermann teaches the course *Knowledge Representation* in the AI Challenge of Mines Saint-Étienne. Students John Doe, Jane Doe are registered to the course *Knowledge Representation*.”

Methodology and challenges

- What are the entities of interest?
- How to identify them?
- Are there entities that not named explicitly but implicitly present?
- Identify binary relation
- Some relations are not binary, but directed graphs only have binary relations (arcs). How to represent them with a directed graph?
- Beware the homonyms!
- Choose good identifiers to avoid ambiguities and naming clashes
- Is your representation compatible with your neighbour’s?

Systems for designing and managing KGs

- Wikibase (Wikimedia foundation, open source)
- Amazon Neptune (Amazon Inc.)
- Neo4J (Neo tech, open source)
- GraphDB (Ontotext)
- Apache Tinkerpop (The Apache Foundation, open source)
- Grafo (data.world)
- Grakn (Grakn Labs, open source)
- Stardog
- Etc… (see https://www.g2.com/categories/graph-databases)

Formats for exporting knowledge graphs

- Many of these systems use the *Resource Description Framework* (RDF) to encode graphs, a Web standard from the World Wide Web Consortium (W3C), developed in the late 1990s and first standardised in 1999.
- In RDF, every entity of interest is identified using *Uniform Resource Identifiers* (URIs). Every URI identifies a unique entity world wide (as opposed to, e.g., primary keys, that identify a unique entity **in a table**; or serial numbers, that identify a unique item **within a series of products**).
- In RDF, edge-labels are URIs, so that they can be uniquely identified.

Identity: Identify entities of interest

- To describe an entity of interest, it must be identified

Does the string "**EMSE**" identifies "*école nationale supérieure des mines de Saint-Étienne*"? A river in Germany?

*Relevant section in the KG paper: Section 3.2

Identity: Identify entities of interest

- To describe an entity of interest, it must be identified

Does the string "**EMSE**" identifies "*école nationale supérieure des mines de Saint-Étienne*"? A river in Germany?

- Avoid ambiguities: use **persistent identifiers** (such as URIs)

http://www.emse.fr/

*Relevant section in the KG paper: Section 3.2

Identity: Identify entities of interest

- To describe an entity of interest, it must be identified

EMSE isLocatedIn **St-Étienne**

Does the string "**EMSE**" identifies "*école nationale supérieure des mines de Saint-Étienne*"? A river in Germany?

- Avoid ambiguities: use **persistent identifiers** (such as URIs)

http://www.emse.fr/

- If knowledge has to be shared or has to integrate other people's knowledge: reuse **external identity links**

http://www.wikidata.org/entity/Q3578252

*Relevant section in the KG paper: Section 3.2

Identity: Identify entities of interest

- To describe an entity of interest, it must be identified

Does the string "**EMSE**" identifies "*école nationale supérieure des mines de Saint-Étienne*"? A river in Germany?

- Avoid ambiguities: use **persistent identifiers** (such as URIs)

http://www.emse.fr/

- If knowledge has to be shared or has to integrate other people's knowledge: reuse **external identity links**

http://www.wikidata.org/entity/Q3578252

- Identity unknown: **existential nodes**

*Relevant section in the KG paper: Section 3.2

Identity: literal values and datatypes

- Some entities are values that can be completely encoded in a computer: integers, decimal numbers, character strings, dates and times
- These are special entities that we describe in a special way:

- To ensure we know what type of data this is describing, we put an explicit datatype; this datatype can itself be identified with a URI
- If we omit that datatype URI, we consider it is a character string (equivalent to **xsd:string**)

*Relevant section in the KG paper: Section 3.2.3

Represent context

- Pascal Ray is the director of Mines St-Étienne
- Philippe Jamet is the director of Mines St-Étienne
- Louis-Antoine Beaunier is the director of Mines St-Étienne
- Donald Trump is the elected president of the USA
- Audi TT is a sport car

Represent context

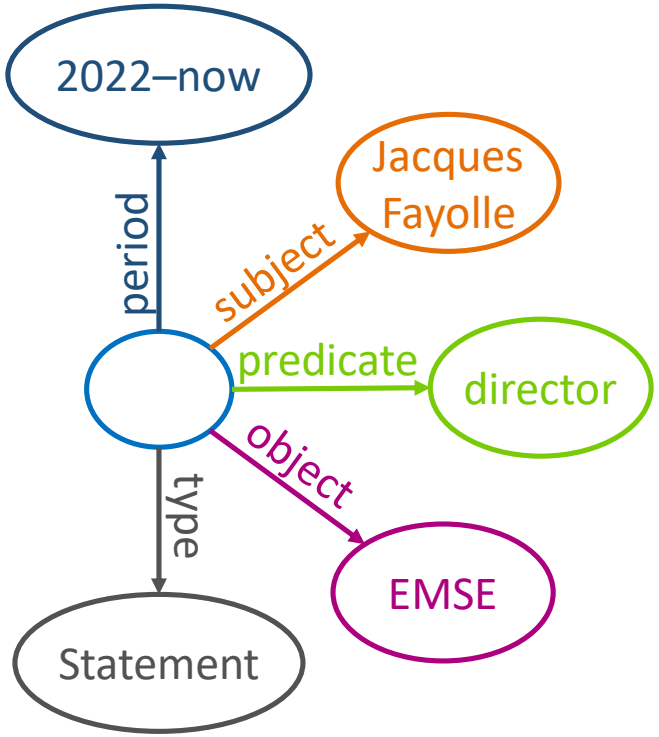
- Jacques Fayolle is the director of Mines St-Étienne (since 2022)
- Philippe Jamet is the director of Mines St-Étienne [2008–2014]
- Louis-Antoine Beaunier is the director of Mines St-Étienne [1816–1835]
- Donald Trump is the elected president of the USA [2017–2021]

  since 2021, according to Trump's supporters
- Audi TT is a sport car 80% true

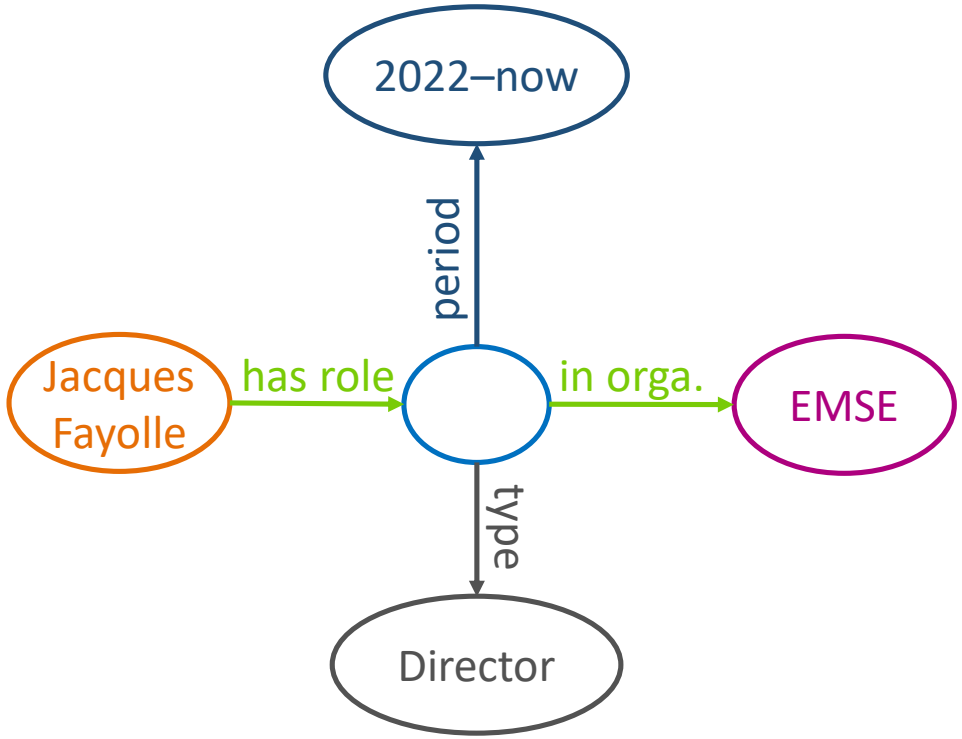
Graph models for context

Non-contextual statement

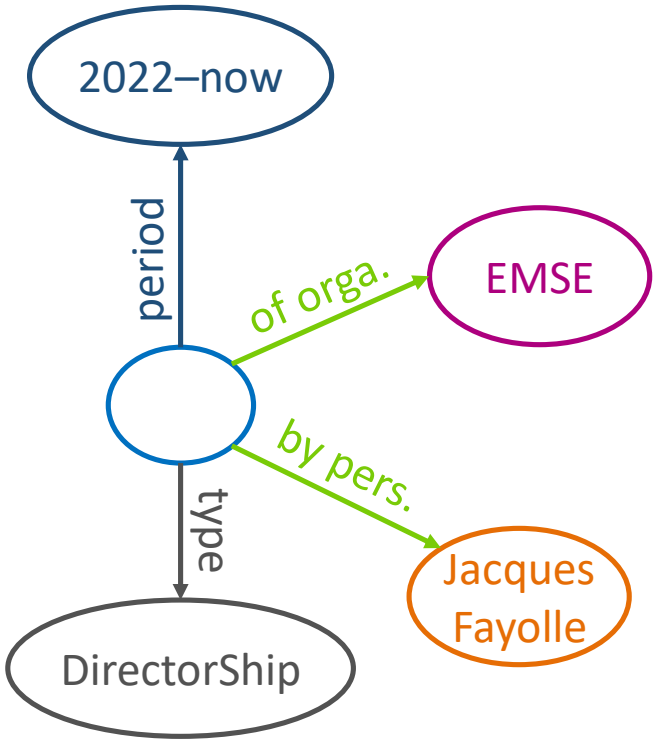
RDF Reification



N-ary rel. v1 (wikidata model)



N-ary rel. v2



[RDF, W3C Rec. 2004]
[RDF 1.1, W3C Rec. 2014]

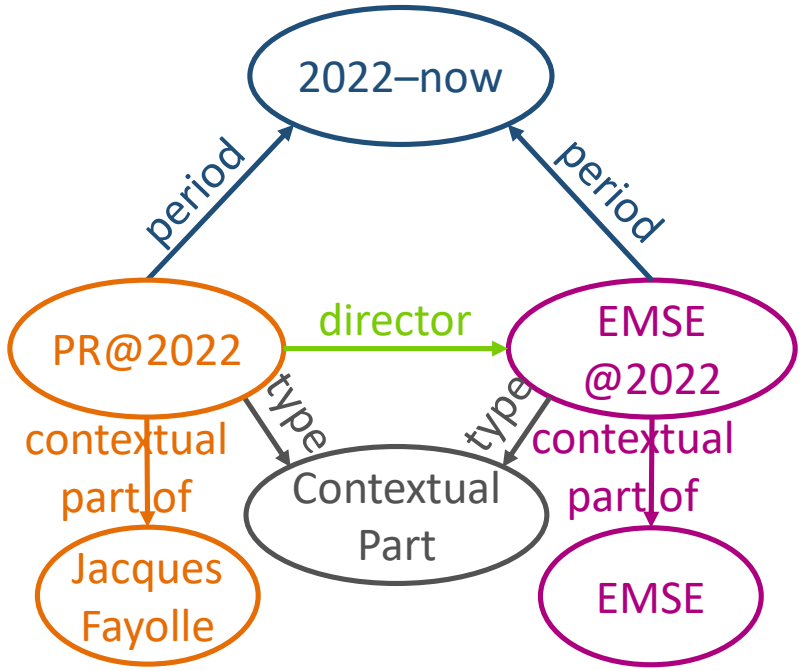
[Noy & Rector, W3C WG Note 2006]

*Relevant section in the KG paper: Section 3.3

Graph models for context

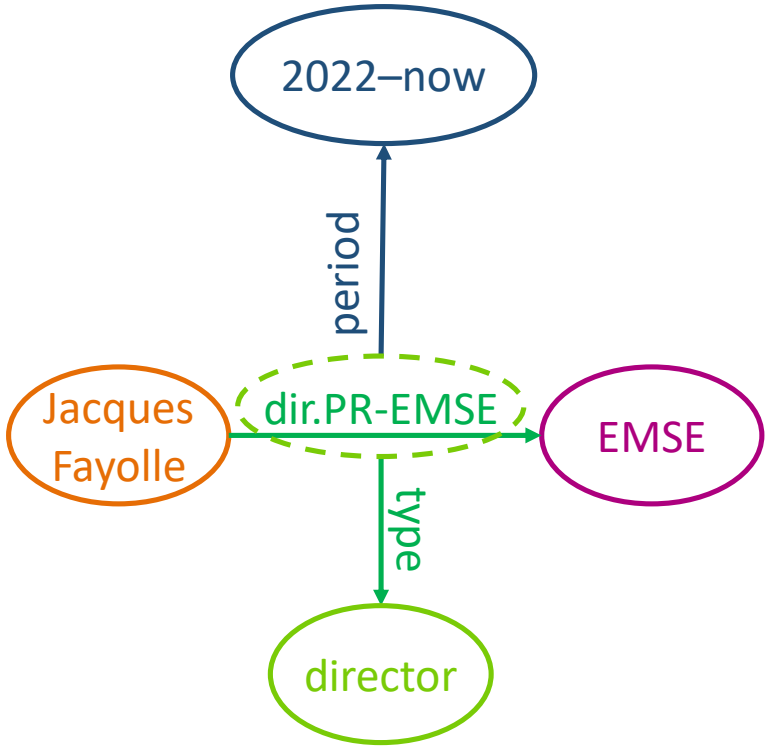
Non-contextual statement

4dFluents



[Giménez-García et al. 2017]

Singleton prop.



[Nguyen et al. 2014]

*Relevant section in the KG paper: Section 3.3