Linking Data with RDF

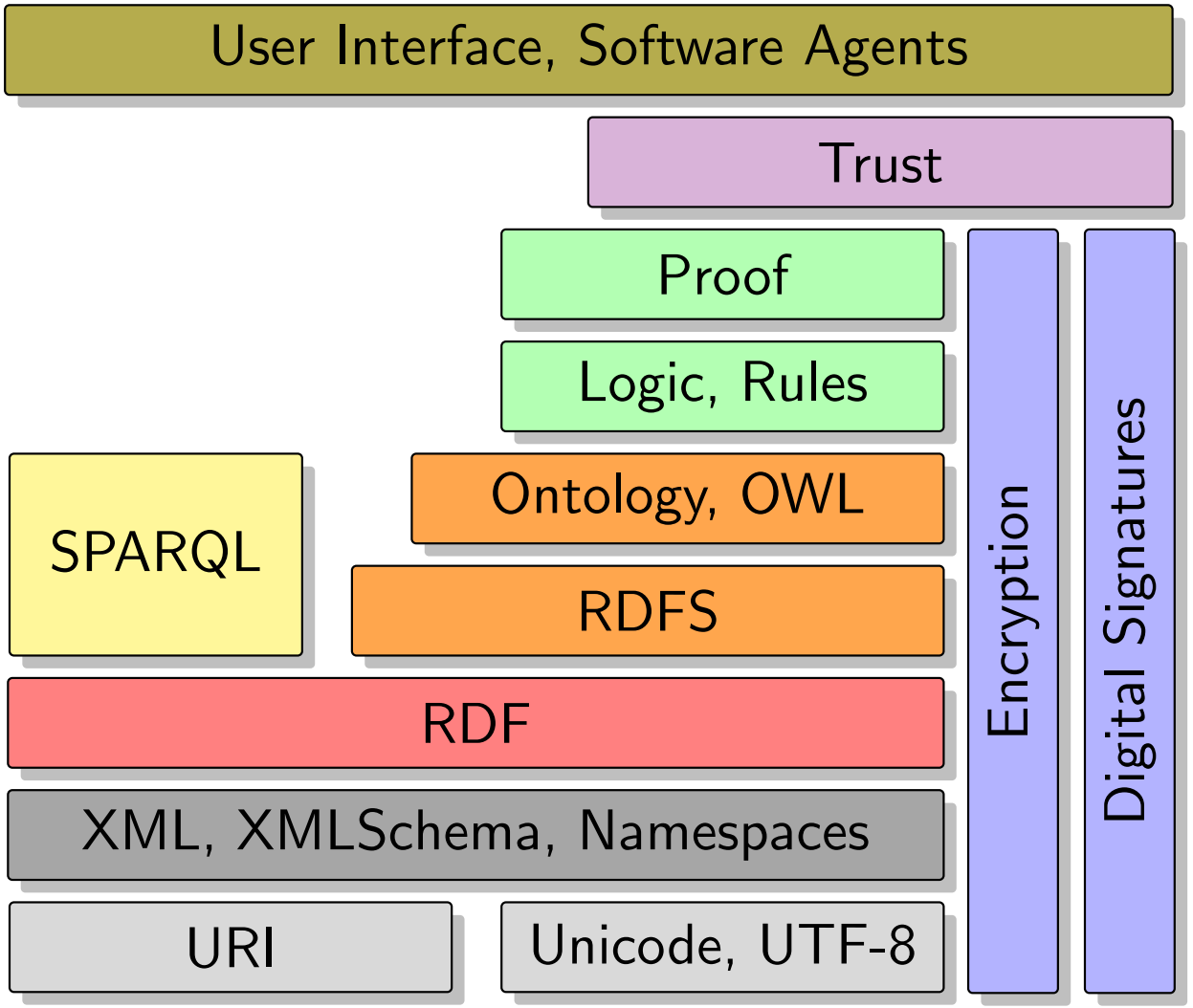
Camilo Thorne

Room 00.012
Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
+49 (0) 711 685-81369
camilo.thorne@ims.uni-stuttgart.de

Semantic Web, SS 2017
(based on slides by W. Kessler)

Institut für Maschinelle Sprachverarbeitung

The Semantic Web Stack [W3C, Tim Berners-Lee]



Institut für Maschinelle Sprachverarbeitung

Outline

Institut für Maschinelle Sprachverarbeitung

Exercise: Application and Data

> William Shakespeare was an English poet and playwright. Shakespeare was born and brought up in Stratford-upon-Avon. At the age of 18, he married Anne Hathaway, with whom he had three children: Susanna, and twins Hamnet and Judith. ... [Wikipedia]

- List all terms that you would like to make statements about (without worrying about overlaps or their connections)
- Draw model/ontology
- Discuss your solution with your neighbour

Institut für Maschinelle Sprachverarbeitung

Outline

Institut für Maschinelle Sprachverarbeitung

Why do we need more than XML and XML Schema?

```
<pet>
   <name>Theo</name>
   <petType>Cat</petType>
   <dateOfBirth>2008-07-21</dateOfBirth>
   <owner name="John Smith" city="Exampleville" />
</pet>
```

```
<宠物>
   <名字>西奥</名字>
   <动物种类>猫</动物种类>
   <出生日期>2008年07月21日</出生日期>
   <饲养人 名字="约翰·史密斯" 城市="例威乐" />
</宠物>
```

- XML Schema specifies only **syntactic** constraints on XML.
- The semantics of XML documents is expressed by the names of tags and thus is not accessible to machines, only to people.
- XML cannot capture relations between different XML elements (e.g., one pet and another pet).
- The nesting of tags does not have a standard meaning:

```
<pet>
   <owner>John Smith</owner>
   <petType>...</petType>
</pet>
```

vs.

```
<owner>
   <name>John Smith</name>
   <pet>...</pet>
</owner>
```

Institut für Maschinelle Sprachverarbeitung

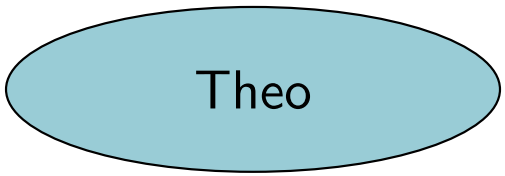
Resource Description Framework (RDF)

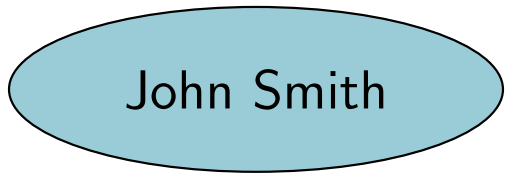
- W3C Recommendation since 2004 (first draft 1999)
- RDF is a **data model**
- With RDF we can describe **resources** and their relations
- The basic building blocks are subject-property-object **triples**, also called **statements** which form a (directed) graph
- RDF namespace (usually abbreviated as `rdf`): `http://www.w3.org/1999/02/22-rdf-syntax-ns#`
- RDF documents can import external namespaces, they are expected to be RDF documents defining resources, this allows the reuse of resources

Institut für Maschinelle Sprachverarbeitung

RDF Basic Items – Resouces

- In RDF **resources** are the things we talk about
- Resources are usually specific objects or individuals:
  - the cat Theo
  - the person John Smith
  - the book “Max und Moritz”
  - the room V5.02
  - ...





Institut für Maschinelle Sprachverarbeitung

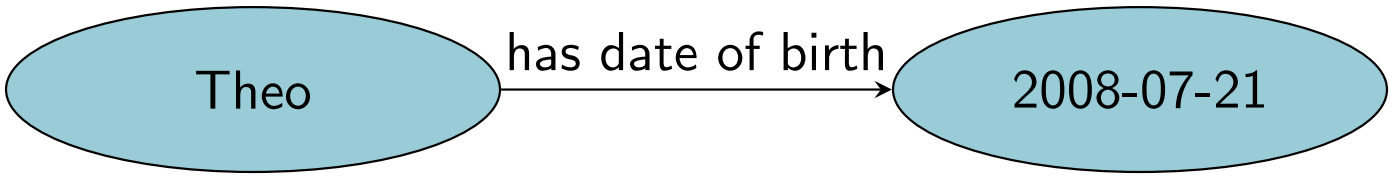
RDF Basic Items – Properties

- A **property** is a special kind of resource that describes a relation between two resources
- Properties are connections:
  - "is pet of" describes the relation of a pet with its owner
  - "has age" states the age of a person, animal or thing
  - "has author" specifies who wrote a book
  - ...

Institut für Maschinelle Sprachverarbeitung

RDF Basic Items – Literals

- A **literal** is an **atomic data value** that further describes a resource
- The birth date of Theo doesn't exist independently from him, but is used to further describe him (attribute)



Institut für Maschinelle Sprachverarbeitung

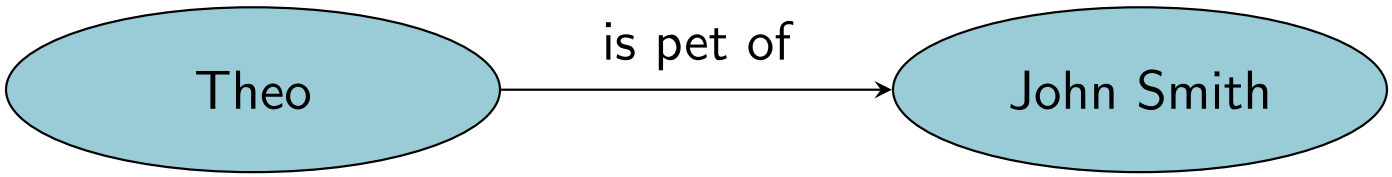
RDF Basic Items – Triples (a.k.a. Statements)

- A **triple (S, P, O)** asserts a property of a resource with

  subject S : a resource,
  predicate P : a property, and
  object O : a resource or a **literal**[1]

- Triples can be written as short sentences:

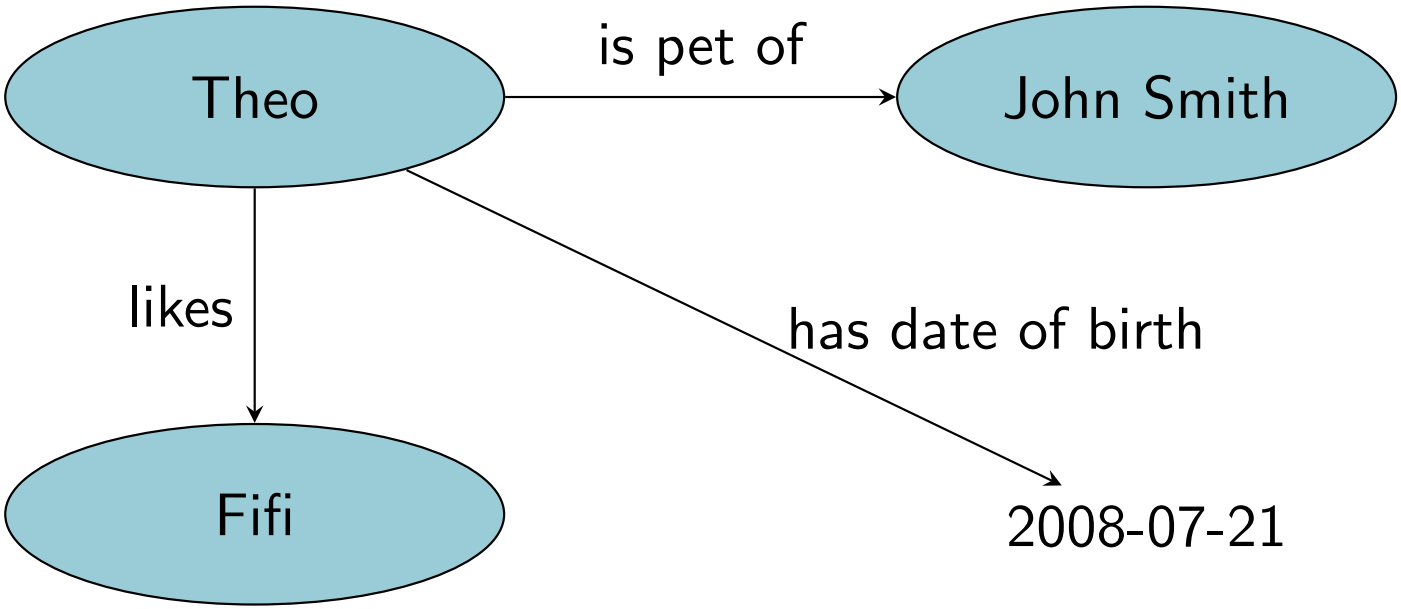
"Theo" "is pet of" "John Smith" .



[1] Literals are atomic values (of a certain data type).

RDF Triples form a Graph

"Theo" "is pet of" "John Smith" .
"Theo" "likes" "Fifi" .
"Theo" "has date of birth" "2008-07-21" .



Directed graph: "Theo" "is pet of" "John Smith" . $\neq$ "John Smith" "is pet" "of Theo" .

Institut für Maschinelle Sprachverarbeitung

RDF uses URIs

- **Resources** and **properties** are **referenced** by their URIs
- **Literals** can have data types identified by URIs (default: string)
- Use the types provided by XML Schema

Institut für Maschinelle Sprachverarbeitung

RDF uses URIs

- **Resources** and **properties** are **referenced** by their URIs
- **Literals** can have data types identified by URIs (default: string)
- Use the types provided by XML Schema

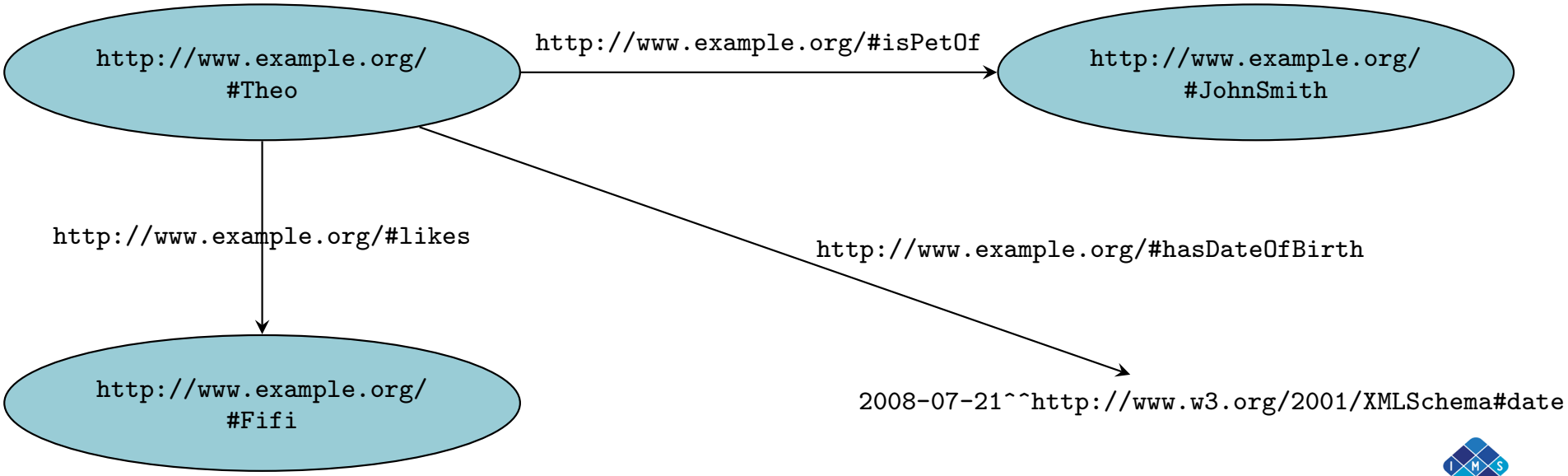
```
http://www.example.org/#Theo http://www.example.org/#isPetOf http://www.example.org/#JohnSmith .
http://www.example.org/#Theo http://www.example.org/#likes http://www.example.org/#Fifi .
http://www.example.org/#Theo http://www.example.org/#hasDateOfBirth
                                        2008-07-21^^http://www.w3.org/2001/XMLSchema#date .
```

Institut für Maschinelle Sprachverarbeitung

RDF uses URIs

- **Resources** and **properties** are **referenced** by their URIs
- **Literals** can have data types identified by URIs (default: string)
- Use the types provided by XML Schema

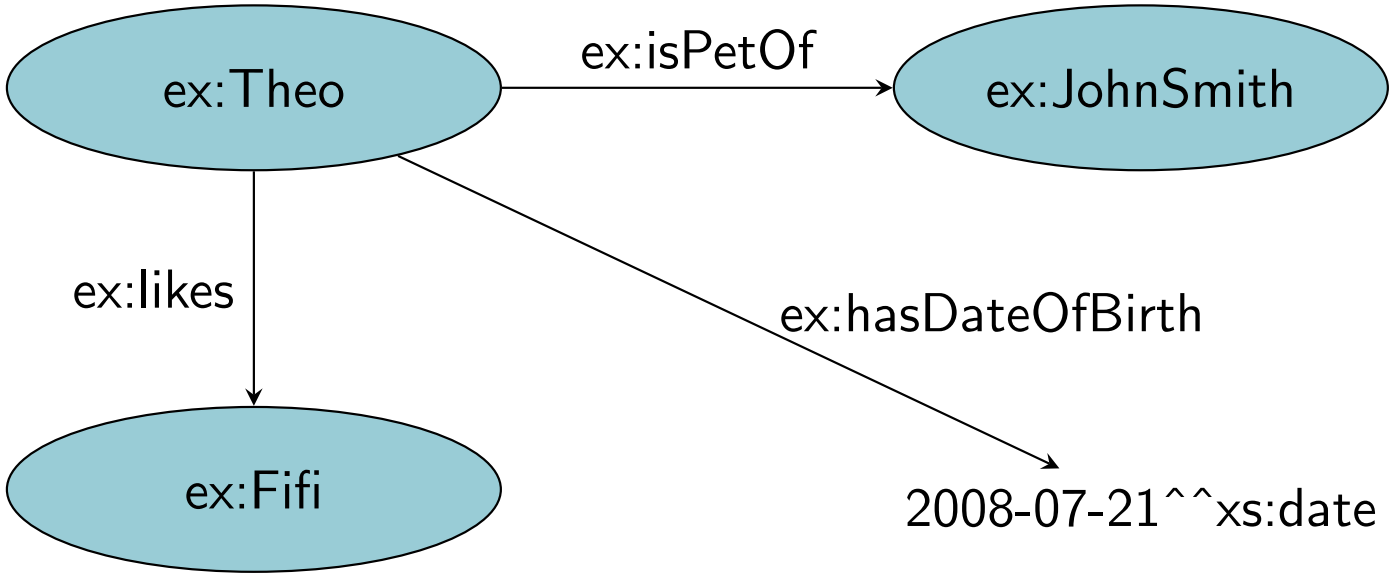
```
http://www.example.org/#Theo http://www.example.org/#isPetOf http://www.example.org/#JohnSmith .
http://www.example.org/#Theo http://www.example.org/#likes http://www.example.org/#Fifi .
http://www.example.org/#Theo http://www.example.org/#hasDateOfBirth
                                        2008-07-21^^http://www.w3.org/2001/XMLSchema#date .
```



Institut für Maschinelle Sprachverarbeitung

RDF uses URIs – with Namespaces

```
PREFIX ex: <http://www.example.org/#>
PREFIX xs: <http://www.w3.org/2001/XMLSchema#>
```



Institut für Maschinelle Sprachverarbeitung

Quiz: RDF data model

Which of these statements are true?

A) RDF statements form a directed graph.

B) A literal can be subject of a statement.

C) RDF defines relations between resources.

D) RDF is just a fancy way of writing XML.

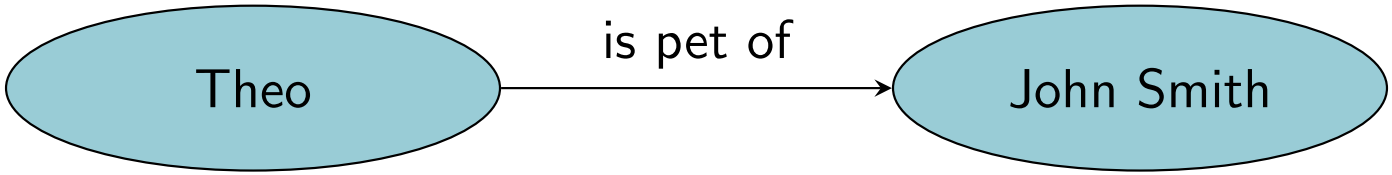
E) RDF graphs are always acyclic.

Institut für Maschinelle Sprachverarbeitung

Outline

Institut für Maschinelle Sprachverarbeitung

Writing RDF in Turtle



Turtle Notation

```
<URI of subject> <URI of predicate> <URI of object> .
```

- Example with long URIs:

```
http://www.example.org/#Theo  http://www.example.org/#isPetOf  http://www.example.org/#JohnSmith .
```

- Example with abbreviated URIs:

```
ex:Theo    ex:isPetOf    ex:JohnSmith .
```

Institut für Maschinelle Sprachverarbeitung

Example as Turtle Triples

- Define namespaces:

```
PREFIX ex: <http://www.example.org/#>
PREFIX xs: <http://www.w3.org/2001/XMLSchema#>
```

- “Theo” “is pet of” “John Smith” .

```
ex:Theo ex:isPetOf ex:JohnSmith .
```

- “Theo” “likes” “Fifi” .

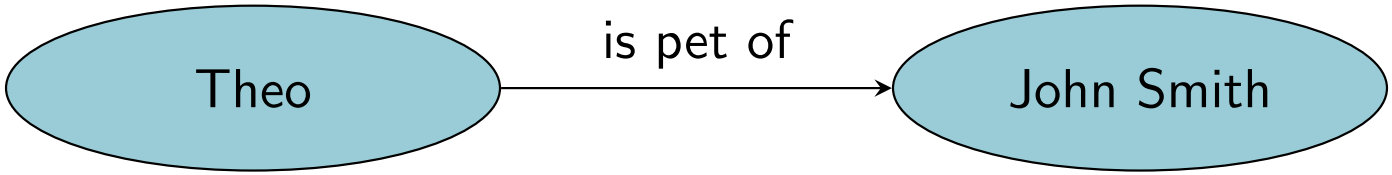
```
ex:Theo ex:likes ex:Fifi .
```

- “Theo” “has date of birth” “2008-07-21” .

```
ex:Theo ex:hasDateOfBirth 2008-07-21^^xs:date .
```

Institut für Maschinelle Sprachverarbeitung

Writing RDF in XML



- Many people write RDF using XML.
- The root element is `rdf:RDF` to specify that RDF is used.
- Resources are `rdf:Description` elements.
- The predicate of a statement is an element nested inside the subject's element, the object is the element content.

```
<rdf:Description rdf:about="http://www.example.org/#Theo">
   <ex:isPetOf>
      <rdf:Description rdf:about="http://www.example.org/#JohnSmith">
      </rdf:Description>
   </ex:isPetOf>
</rdf:Description>
```

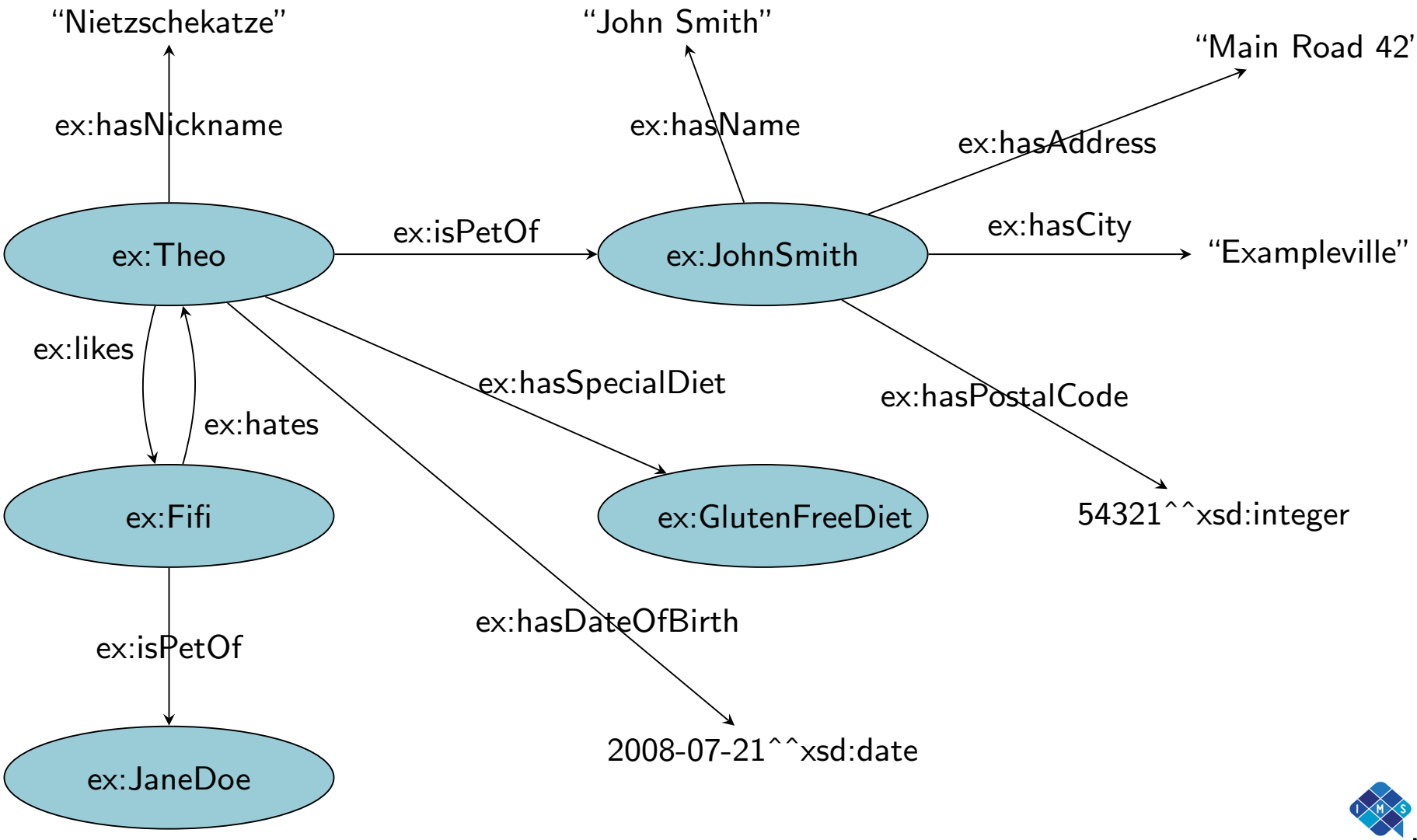
Institut für Maschinelle Sprachverarbeitung

Example as XML

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
      xml:base="http://www.example.org/"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema#date"
      xmlns:ex="http://www.example.org/#">
   <rdf:Description rdf:about="ex:Theo">
      <ex:hasDateOfBirth rdf:datatype="xsd:date">
         2008-07-21
      </ex:hasDateOfBirth>
   </rdf:Description>
   <rdf:Description rdf:about="ex:Theo">
      <ex:isPetOf>
         <rdf:Description rdf:about="ex:JohnSmith">
         </rdf:Description>
      </ex:isPetOf>
   </rdf:Description>
</rdf:RDF>
```

Institut für Maschinelle Sprachverarbeitung

Bigger Example



Institut für Maschinelle Sprachverarbeitung

Bigger Example as Turtle Triples

- Define namespaces:

```
PREFIX ex: <http://www.example.org/#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

- Information about Theo:

```
ex:Theo ex:isPetOf ex:JohnSmith .
ex:Theo ex:hasDateOfBirth 2008-07-21^^xsd:date .
ex:Theo ex:likes ex:Fifi .
ex:Theo ex:hasNickname "Nietzschekatze" .
ex:Theo ex:hasSpecialDiet ex:GlutenFreeDiet .
```

- More about owner John Smith and friend Fifi:

```
ex:JohnSmith ex:hasName "John Smith" .
ex:JohnSmith ex:hasAddress "Main Road 42" .
ex:JohnSmith ex:hasPostalCode 54321^^xsd:integer .
ex:JohnSmith ex:hasCity "Exampleville" .
ex:Fifi ex:isPetOf ex:JaneDoe .
ex:Fifi ex:hates ex:Theo .
```

Institut für Maschinelle Sprachverarbeitung

Exercise: RDF Modeling in Turtle

- Consider the "Shakespeare" model from the first slides
- Write it in RDF, in Turtle format
- Work with your neighbour to ensure that what you wrote has the intended meaning

Institut für Maschinelle Sprachverarbeitung

Outline

Institut für Maschinelle Sprachverarbeitung

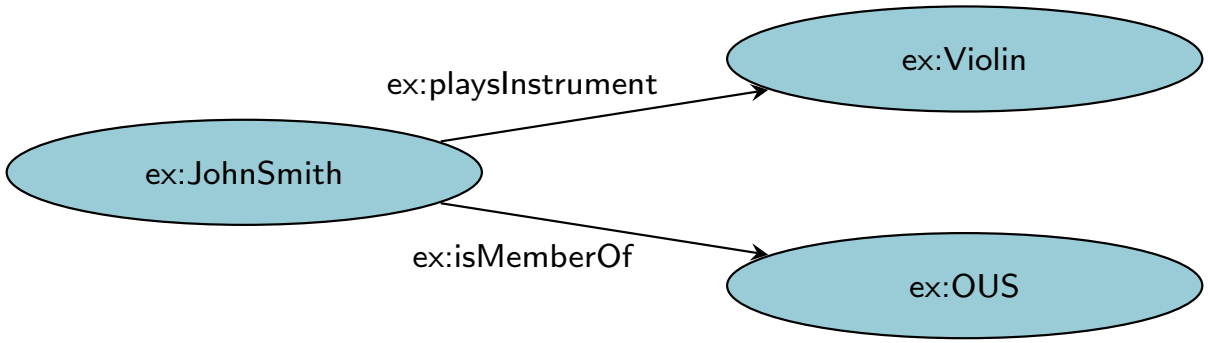
More than Binary

- Until now all the examples were about binary relations.
- But what about relations with more than two arguments?
- Example: John Smith plays the violin in the orchestra of the University of Stuttgart (OUS).
- Writing that John Smith plays the violin is easy:
  `ex:JohnSmith ex:playsInstrument ex:Violin .`
- Writing that John Smith is a member of the OUS is also easy:
  `ex:JohnSmith ex:isMemberOf ex:OUS .`
- But how do we combine the two?

Institut für Maschinelle Sprachverarbeitung

First Try

```
ex:JohnSmith ex:playsInstrument ex:Violin .
ex:JohnSmith ex:isMemberOf ex:OUS .
```

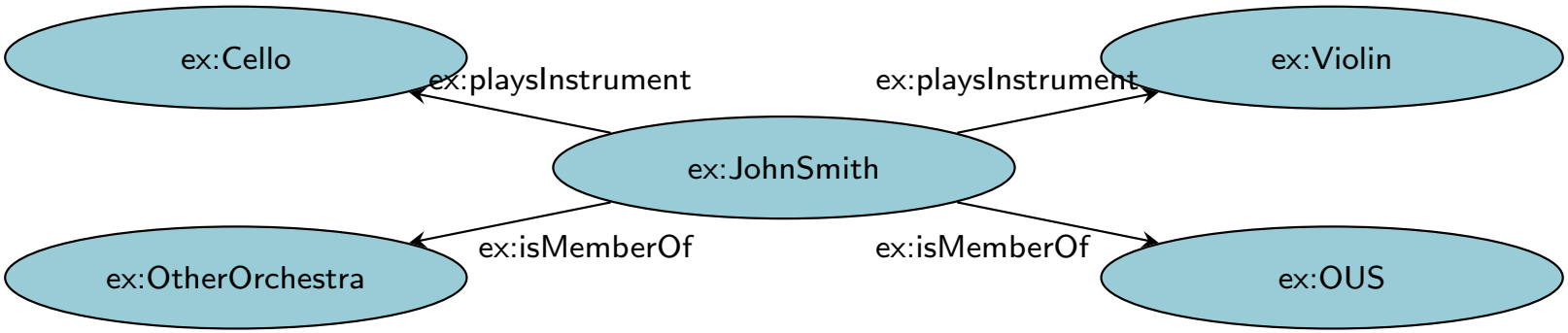


Institut für Maschinelle Sprachverarbeitung

A Cello Messes Up the First Try

```
ex:JohnSmith ex:playsInstrument ex:Violin .
ex:JohnSmith ex:isMemberOf ex:OUS .

ex:JohnSmith ex:playsInstrument ex:Cello .
ex:JohnSmith ex:isMemberOf ex:OtherOrchestra .
```

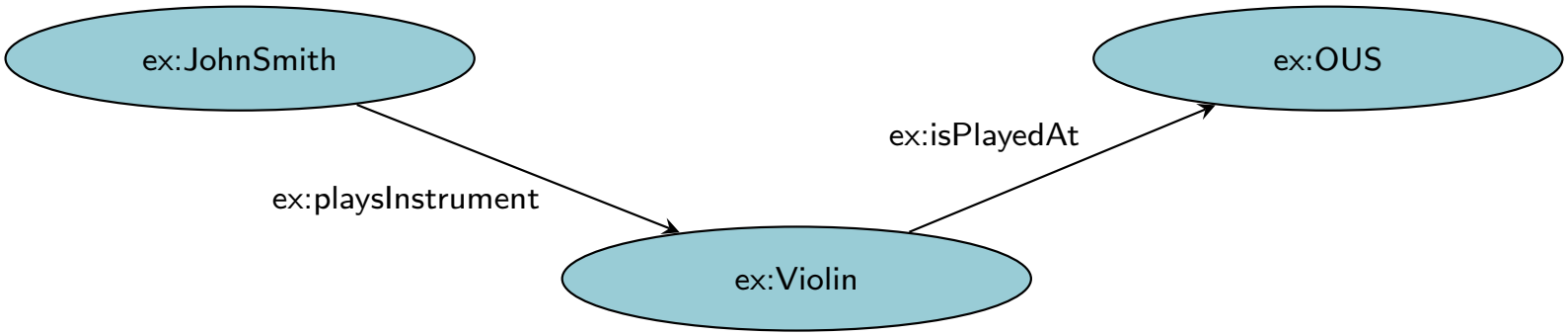


We cannot distinguish what is played where (it's a graph)!

Institut für Maschinelle Sprachverarbeitung

Second Try

```
ex:JohnSmith ex:playsInstrument ex:Violin .
ex:Violin ex:isPlayedAt ex:OUS .
```

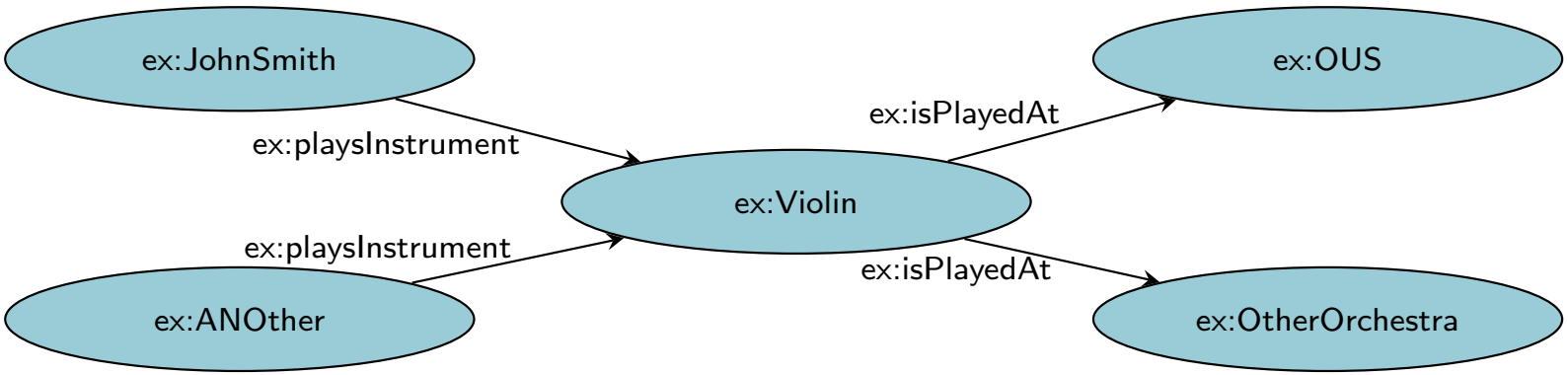


Institut für Maschinelle Sprachverarbeitung

A.N. Other Messes Up the Second Try

```
ex:JohnSmith ex:playsInstrument ex:Violin .
ex:Violin ex:isPlayedAt ex:OUS .
```

```
ex:ANOther ex:playsInstrument ex:Violin .
ex:Violin ex:isPlayedAt ex:OtherOrchestra .
```

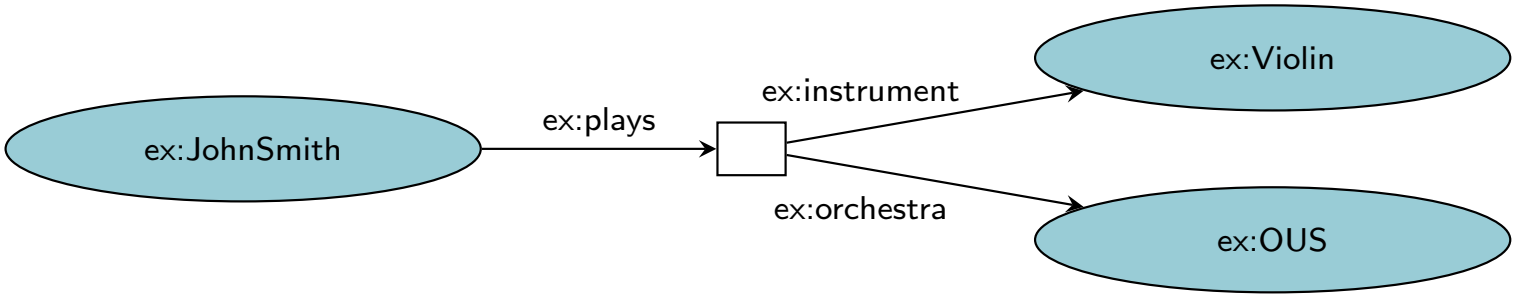


We cannot distinguish who is plays where (it's a graph)!

Institut für Maschinelle Sprachverarbeitung

Blank Nodes

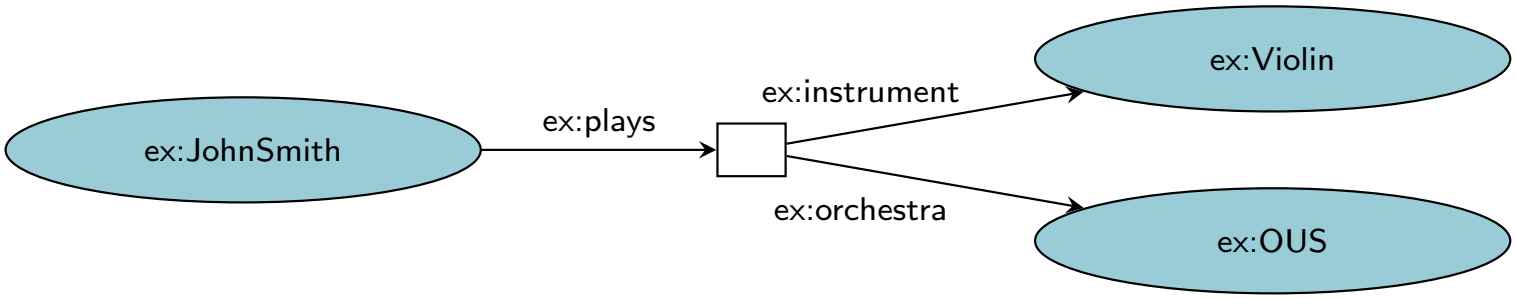
- N-ary relations cannot be expressed directly in RDF
- A **blank node** has to be introduced
- Blanks nodes are **pointers**



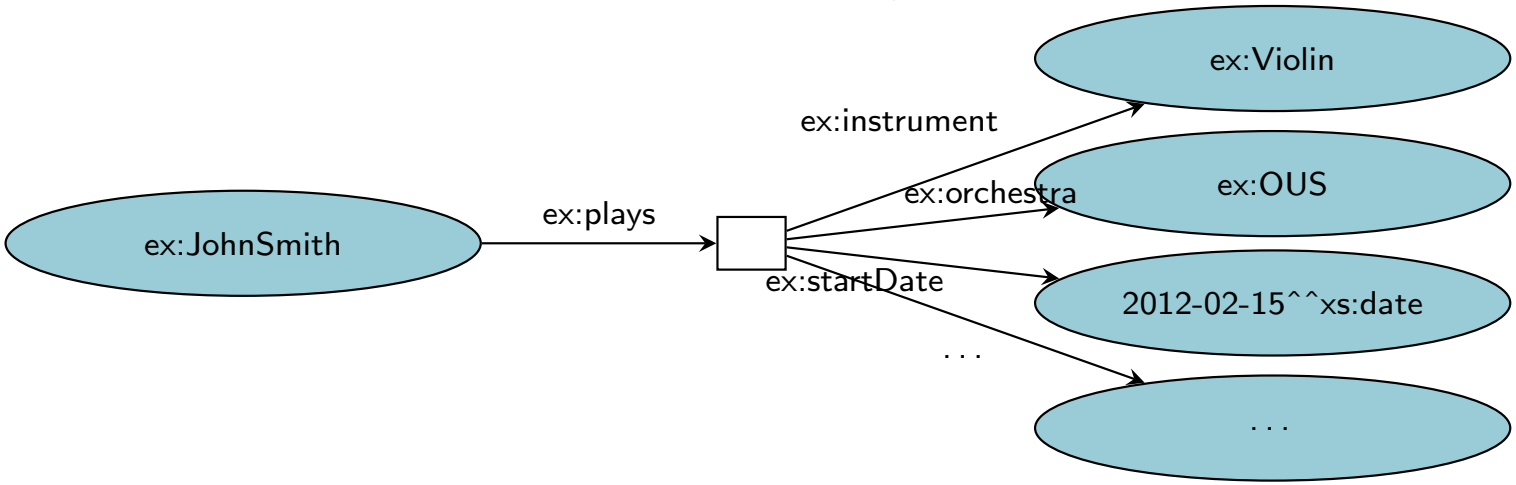
Institut für Maschinelle Sprachverarbeitung

Blank Nodes

- N-ary relations cannot be expressed directly in RDF
- A **blank node** has to be introduced
- Blanks nodes are **pointers**



- We can now add arbitrary many arguments:



Institut für Maschinelle Sprachverarbeitung

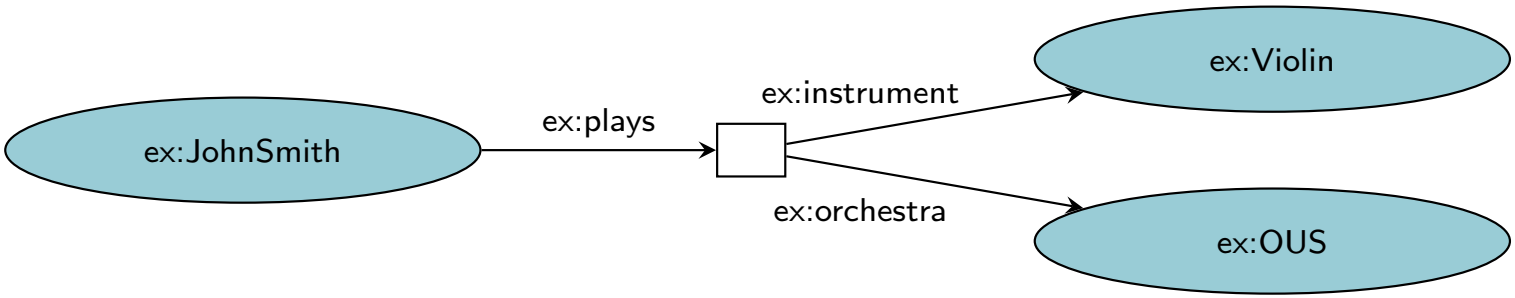
Blank Nodes in Turtle

- Blank node can be stated **explicitly**:

```
ex:JohnSmith ex:plays _:node1 .
_:node1 ex:instrument ex:Violin .
_:node1 ex:orchestra ex:OUS .
```

- Blank nodes can be stated **implicitly**:

```
ex:JohnSmith ex:plays [
        ex:instrument ex:Violin ;
        ex:orchestra ex:OUS
] .
```



Institut für Maschinelle Sprachverarbeitung

Exercise: RDF Modeling with N-Ary Relations

- For your application think about non-binary relations
- Model two of them using empty nodes
- Discuss your solution with your neighbor

▷ Example: "Heal the world" is track num. 7 on "Dangerous":

```
ex:HealTheWorld ex:isTrackInAlbum _:node1 .
_:node1 ex:hasAlbum ex:Dangerous .
_:node1 ex:hasTrackNumber 7^^xsd:int .
```

Institut für Maschinelle Sprachverarbeitung

More . . .

- This has been but a small part of what there is to say about RDF!

Institut für Maschinelle Sprachverarbeitung

More . . .

- This has been but a small part of what there is to say about RDF!
- ▷ Examples:
  - many more syntax elements exist
  - blank nodes can be used as containers
    (to group together resources or attributes)
  - reification can be used to alternatively express N-ary relations
  - RDF triples can be expressed/written as logical formulas
  - etc., etc., . . .

Institut für Maschinelle Sprachverarbeitung

Assignment – Guidelines

- Later today I'll post an exercise sheet on RDF on ILIAS
- It covers today's and next week's topics, you have 20 days to solve it
- You should submit it via email (to me)
- You can solve it in groups, but should be submitted individually

- Format:
  1. (at least) 3 files: .pdf file (for written answers)
     .xml and .ttl (for RDF models)
  2. please bundle them in a .zip file, whose name should contain:
     - full name
     - student ID number
     - exercise number

Institut für Maschinelle Sprachverarbeitung

Assignment – Guidelines

- Later today I'll post an exercise sheet on RDF on ILIAS
- It covers today's and next week's topics, you have 20 days to solve it
- You should submit it via email (to me)
- You can solve it in groups, but should be submitted individually

- Format:
  1. (at least) 3 files: .pdf file (for written answers)
     .xml and .ttl (for RDF models)
  2. please bundle them in a .zip file, whose name should contain:
     - full name
     - student ID number
     - exercise number

If possible, typeset in LaTeX

Institut für Maschinelle Sprachverarbeitung

Outline

Institut für Maschinelle Sprachverarbeitung

Summary: Linking Data with RDF

- RDF is a **data model** to describe relations between resources
- **Resources** are the things we talk about (referenced by URIs)
- **Properties** describe relations between resources
- RDF **statements** (triples) consist of

  subject S : a resource
  predicate P : a property
  object O : a resource or a literal
- RDF can be written using **Turtle** or **XML**
- All relations in RDF are **binary**, to express n-ary relations, **blank nodes** have to be used

Institut für Maschinelle Sprachverarbeitung

Outline

Institut für Maschinelle Sprachverarbeitung

Suggested Reading


Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph and York Sure. Semantic Web. Grundlagen. Springer textbook, 2008. (Chapter 3)


Pascal Hitzler, Markus Krötzsch and Sebastian Rudolph. Foundations of Semantic Web Technologies. Chapman & Hall/CRC, 2009. (Chapter 2)

Institut für Maschinelle Sprachverarbeitung