

Vorlesung Semantic Web



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Vorlesung im Wintersemester 2012/2013

Dr. Heiko Paulheim

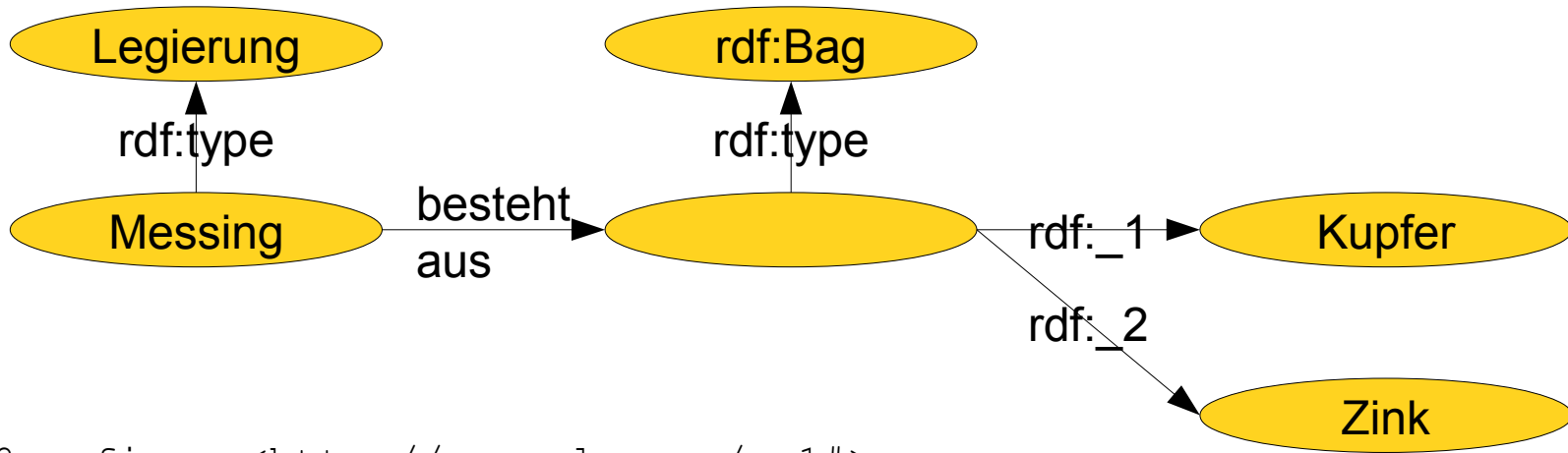
Fachgebiet Knowledge Engineering

Aufgabe 1

- Aussagen in RDF
 - Messing ist eine Legierung aus den Bestandteilen Kupfer und Zink.
 - Der SPIEGEL ist eine Zeitschrift, dessen Redaktion in Hamburg sitzt.
 - Die Ausarbeitung besteht aus den Teilen Einleitung, Hauptteil und Schluss.
 - Markus weiß, dass Jutta in Frankfurt wohnt.
 - Jutta sagt, dass Ihre Freundin in Darmstadt wohnt.
 - Stefan glaubt, dass Anna weiß, das er ihren Vater kennt.

Aufgabe 1

- Messing ist eine Legierung aus Kupfer und Zink



```
@prefix : <http://example.org/ex1#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
:Messing a :Legierung .  
:Messing :bestehtAus [  
  a rdf:Bag ;  
  rdf:_1 :Kupfer ;  
  rdf:_2 :Zink .  
].
```

Aufgabe 1

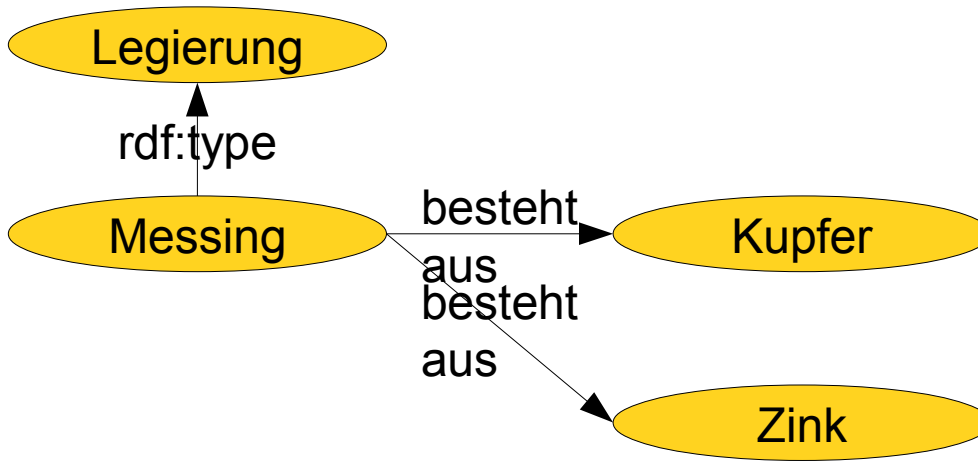
- **Messing ist eine Legierung aus Kupfer und Zink**

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://example.org/ex1#">
  <ex:Legierung rdf:about="http://example.org/ex1#Messing">
    <ex:bestehtAus>
      <rdf:Bag>
        <rdf:li rdf:resource="http://example.org/ex1#Kupfer" />
        <rdf:li rdf:resource="http://example.org/ex1#Zinn" />
      </rdf:Bag>
    </ex:bestehtAus>
  </ex:Legierung>
</rdf:RDF>
```

- **Testen:** <http://www.w3.org/RDF/Validator/>

Aufgabe 1

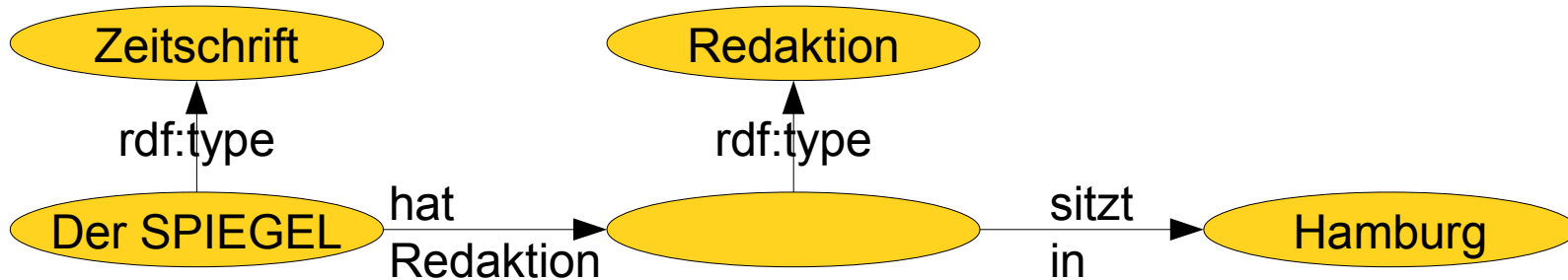
- Alternative: ohne Bag



```
@prefix : <http://example.org/ex1#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
:Messing a :Legierung .  
:Messing :bestehtAus :Kupfer, :Zink .
```

Aufgabe 1

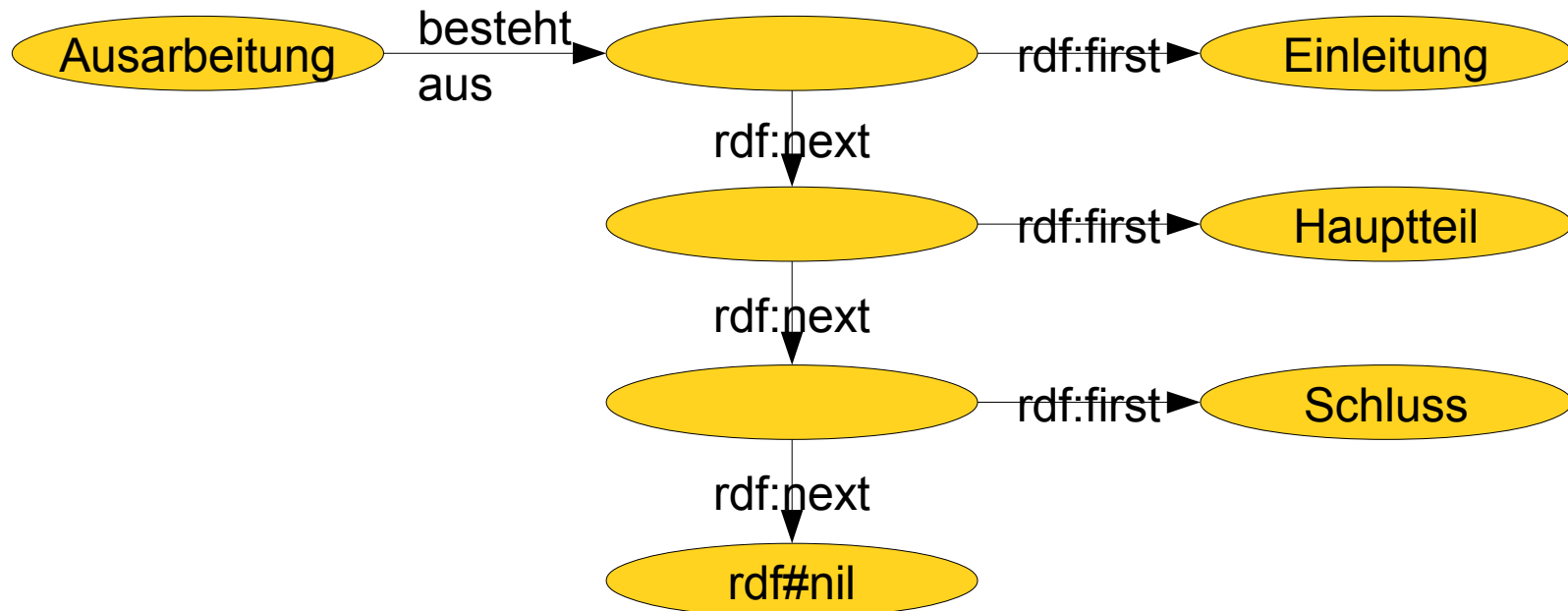
- Der SPIEGEL ist eine Zeitschrift, dessen Redaktion in Hamburg sitzt.



```
@prefix : <http://example.org/ex1#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
:DerSPIEGEL a :Zeitschrift ;
            :hatRedaktion [
                a :Redaktion ;
                :sitzt_in :Hamburg ] .
```

Aufgabe 1

- Die Ausarbeitung besteht aus den Teilen Einleitung, Hauptteil und Schluss.



```
@prefix : <http://example.org/ex1#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
:Ausarbeitung :bestehtAus (:Einleitung :Hauptteil :Schluss) .
```

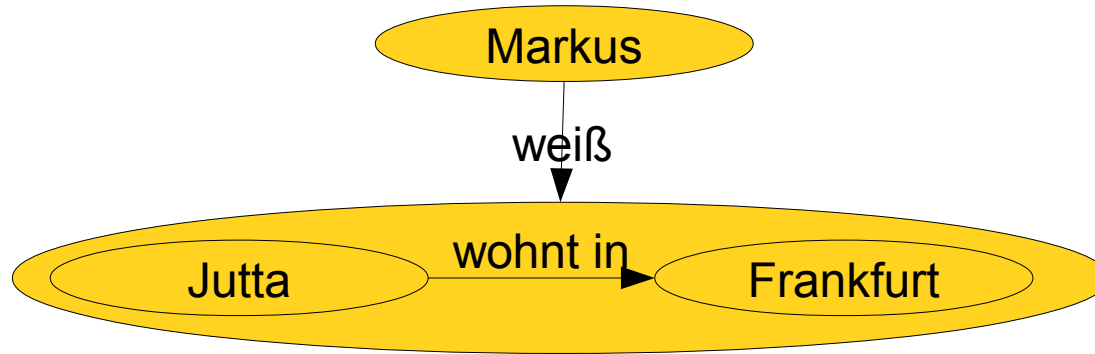
Aufgabe 1

- Markus weiß, dass Jutta in Frankfurt wohnt.
 - Schritt 1: Jutta wohnt in Frankfurt



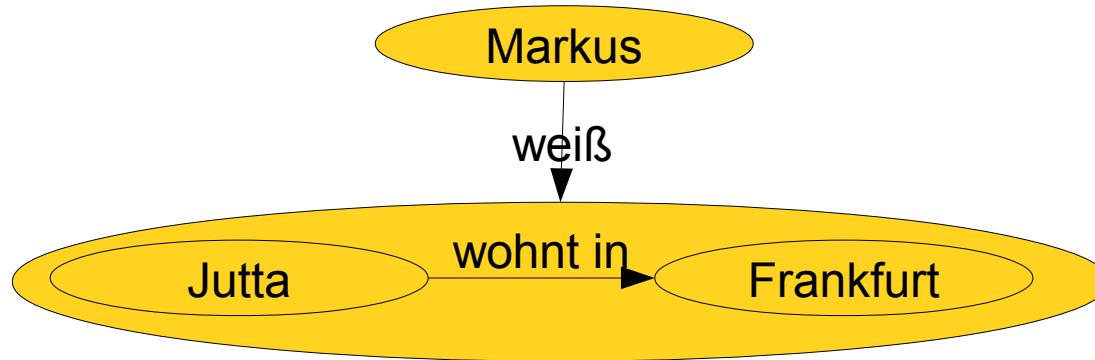
Aufgabe 1

- Markus weiß, dass Jutta in Frankfurt wohnt.
 - Schritt 2: ...und Markus weiß das.



Aufgabe 1

- Markus weiß, dass Jutta in Frankfurt wohnt.
 - Schritt 2: ...und Markus weiß das.



Aufgabe 1

▪ In N3:

```
@prefix : <http://example.org/ex1#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
:Markus :weiß [  
  a rdf:Statement;  
  rdf:subject :Jutta ;  
  rdf:predicate :wohnt_in ;  
  rdf:object :Frankfurt ] .
```

Aufgabe 1

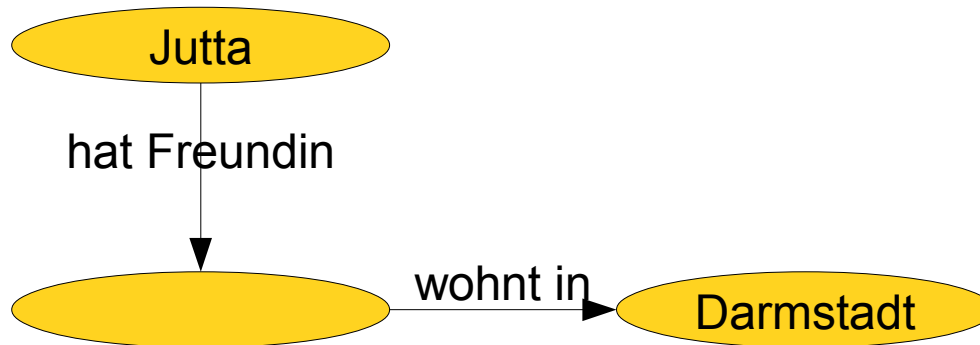


▪ In RDF/XML:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:ex="http://example.org/ex1#">
  <rdf:Description rdf:about="http://example.org/ex1#Markus">
    <ex:weiß>
      <rdf:Statement>
        <rdf:subject rdf:resource="http://example.org/ex1#Jutta" />
        <rdf:predicate rdf:resource="http://example.org/ex1#wohnt_in" />
        <rdf:object rdf:resource="http://example.org/ex1#Frankfurt" />
      </rdf:Statement>
    </ex:weiß>
  </rdf:Description>
</rdf:RDF>
```

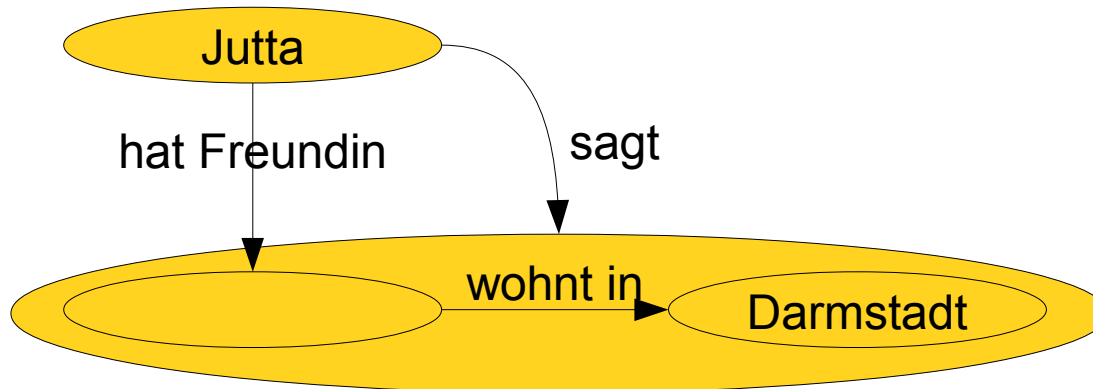
Aufgabe 1

- Jutta sagt, dass ihre Freundin in Darmstadt wohnt.
 - Schritt 1: Wir wissen nicht, wer diese Freundin ist
 - also modellieren wir sie als Blank Node



Aufgabe 1

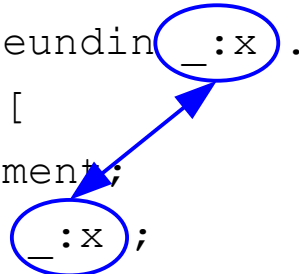
- Jutta sagt, dass ihre Freundin in Darmstadt wohnt.
 - Schritt 2: die Reifikation



Aufgabe 1

▪ In N3:

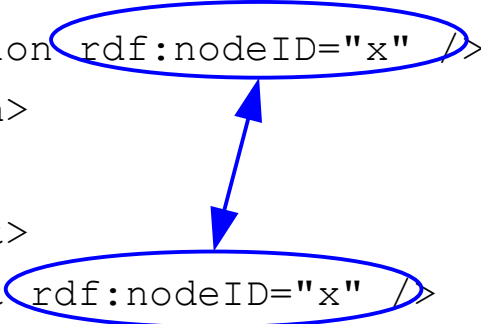
```
@prefix : <http://example.org/ex1#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
:Jutta :hatFreundin _:x .  
:Jutta :sagt [  
  a rdf:Statement;  
  rdf:subject _:x;  
  rdf:predicate :wohnt_in ;  
  rdf:object :Darmstadt ] .
```



Aufgabe 1

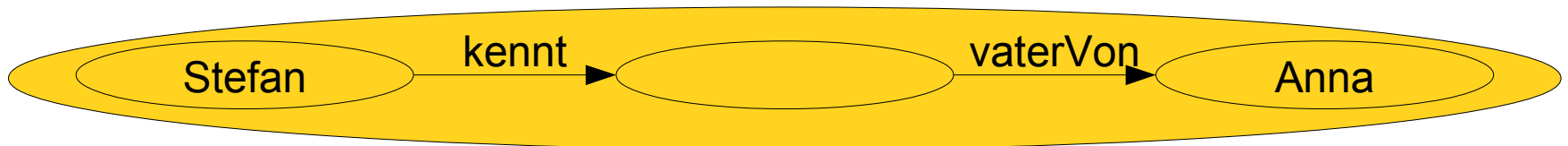
▪ In RDF/XML:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:ex="http://example.org/ex1#">
  <rdf:Description rdf:about="http://example.org/ex1#Jutta">
    <ex:hatFreundin>
      <rdf:Description rdf:nodeID="x" />
    </ex:hatFreundin>
    <ex:sagt>
      <rdf:Statement>
        <rdf:subject rdf:nodeID="x" />
        <rdf:predicate rdf:resource="http://example.org/ex1#wohnt_in" />
        <rdf:object rdf:resource="http://example.org/ex1#Darmstadt" />
      </rdf:Statement>
    </ex:sagt>
  </rdf:Description>
</rdf:RDF>
```



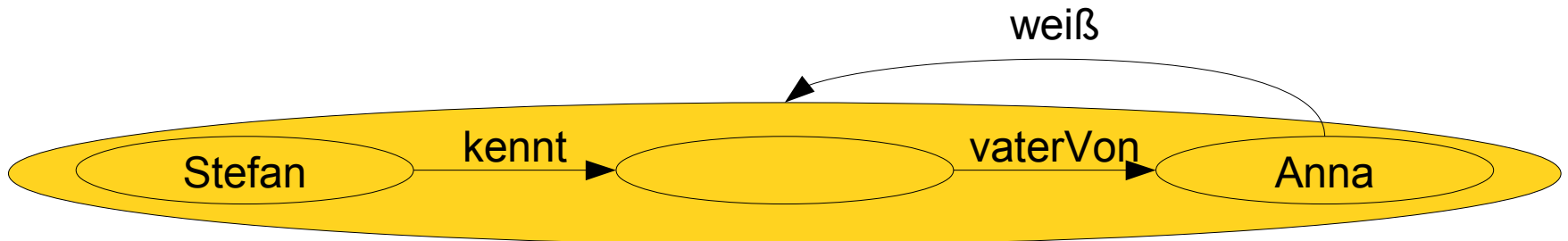
Aufgabe 1

- Stefan glaubt, dass Anna weiß, dass er ihren Vater kennt.
 - Schritt 1: Stefan kennt Annas Vater.



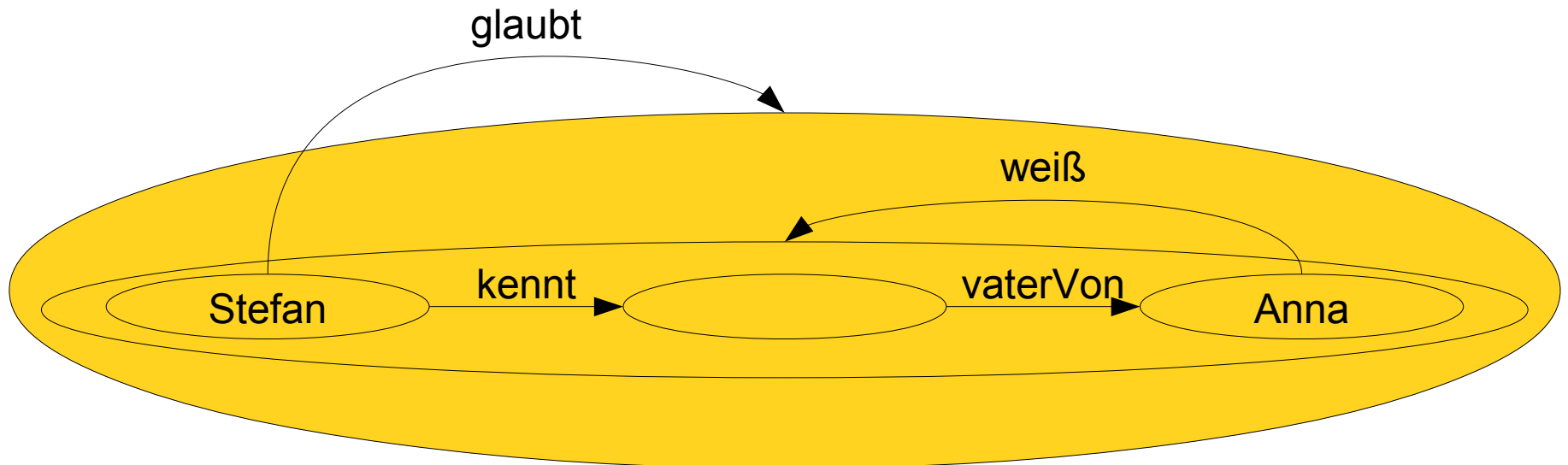
Aufgabe 1

- Stefan glaubt, dass Anna weiß, dass er ihren Vater kennt.
 - Schritt 2: ...und Anna weiß das.



Aufgabe 1

- Stefan glaubt, dass Anna weiß, dass er ihren Vater kennt.
 - Schritt 3: Stefan glaubt...



Aufgabe 1

▪ In N3:

```
@prefix : <http://example.org/ex1#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
:Stefan :glaubt [
  a rdf:Statement;
  rdf:subject :Anna ;
  rdf:predicate :weiß ;
  rdf:object [
    a rdf:Statement;
    rdf:subject :Stefan ;
    rdf:predicate :kennt ;
    rdf:object [
      :vaterVon :Anna
    ]
  ]
].
```

Aufgabe 1



▪ In RDF/XML:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://example.org/ex1#">
  <rdf:Statement rdf:nodeID="bnode1">
    <rdf:subject>
      <rdf:Description rdf:about="http://example.org/ex1#Stefan">
        <ex:glaubt>
          <rdf:Statement>
            <rdf:subject rdf:resource="http://example.org/ex1#Anna" />
            <rdf:predicate rdf:resource="http://example.org/ex1#weiß" />
            <rdf:object rdf:nodeID="bnode1" />
          </rdf:Statement>
        </ex:glaubt>
      </rdf:Description>
    </rdf:subject>
    <rdf:predicate rdf:resource="http://example.org/ex1#kennt" />
    <rdf:object>
      <rdf:Description>
        <ex:vaterVon rdf:resource="http://example.org/ex1#Anna" />
      </rdf:Description>
    </rdf:object>
  </rdf:Statement>
</rdf:RDF>
```

Aufgabe 2

- Gegeben ist folgende Menge von Tripeln:
 - :Peter :belegt :VL_Semantic_Web .
 - :Klaus :belegt :VL_Semantic_Web .
 - :Peter :belegt :VL_Verteilte_Systeme .
 - _:x :belegt :VL_Semantic_Web .
 - _:x :belegt :VL_Maschinelles_Lernen .
 - :Peter :belegt _:y .
 - :Jan :belegt _:y .
- Wie würden Sie diese Menge interpretieren? Stellen Sie sich dabei Fragen wie:
 - Wer belegt welche Veranstaltung?
 - Wie viele Studenten belegen eine bestimmte Veranstaltung?
 - Wie viele Veranstaltungen gibt es?

Aufgabe 2

- Beachten:
 - Non-unique name assumption
 - Open World Assumption

- Also:
 - Es gibt mindestens eine Veranstaltung und einen Student*
 - Peter, Klaus und Jan belegen jeweils mindestens eine Vorlesung
 - es kann sich dabei aber um 1-3 Personen handeln
 - Die Veranstaltungen Semantic Web, Verteilte Systeme und Maschinelles Lernen werden je von mindestens einem Studenten belegt
 - Es kann sich dabei aber um 1-3 Veranstaltungen handeln

* genauer: etwas belegtes und etwas belegendes

Aufgabe 2

- Beachten:
 - Non-unique name assumption
 - Open World Assumption

- Also:
 - Es gibt mindestens einen Studenten, der sowohl Semantic Web als auch Verteilte Systeme belegt (nämlich Peter)
 - es kann sich aber auch um eine Veranstaltung handeln
 - Es gibt mindestens einen Studenten, der sowohl Semantic Web als auch Maschinelles Lernen belegt
 - es kann sich aber auch um eine Veranstaltung handeln
 - Es gibt mindestens eine Veranstaltung, die sowohl Jan als auch Peter belegen
 - oder Jan und Peter sind dieselbe Person

Aufgabe 3

- Gegeben ist folgende XML-Darstellung eines RDF-Graphen:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:mhb="http://mhb.com/">
  <mhb:Lecture rdf:about="http://mhb.com/1001">
    <mhb:givenBy>
      <mhb:Lecturer>
        <mhb:name>Peter Müller</mhb:name>
      </mhb:Lecturer>
    </mhb:givenBy>
  </mhb:Lecture>
  <mhb:Lecture rdf:about="http://mhb.com/1002">
    <mhb:givenBy>
      <mhb:Lecturer>
        <mhb:name>Hans Meyer</mhb:name>
      </mhb:Lecturer>
    </mhb:givenBy>
  </mhb:Lecture>
  ...
</rdf:RDF>
```

Aufgabe 3

- Sie haben die Aufgabe, aus diesem RDF-Graphen eine Liste aller Dozenten zu extrahieren. In einem Forum schlägt der Nutzer XMLFan vor, folgendes XPath-Statement zu verwenden:
 - `//mhb:Lecture/mhb:givenBy/mhb:Lecturer/mhb:name`
- Was halten Sie von diesem Vorschlag?

Aufgabe 3



- RDF/XML erlaubt viele äquivalente Serialisierungen:

```
<mhb:Lecture rdf:about="http://mhb.com/1001">
  <mhb:givenBy>
    <mhb:Lecturer>
      <mhb:name>Peter Müller</mhb:name>
    </mhb:Lecturer>
  </mhb:givenBy>
</mhb:Lecture>
```

- ...ist äquivalent zu

```
<rdf:Description rdf:about="http://mhb.com/1001">
  <rdf:type rdf:resource="http://mhb.com/Lecture"/>
  <mhb:givenBy>
    <mhb:Lecturer>
      <mhb:name>Peter Müller</mhb:name>
    </mhb:Lecturer>
  </mhb:givenBy>
</rdf:Description>
```

- Was macht jetzt

```
//mhb:Lecture/mhb:givenBy/mhb:Lecturer/mhb:name ???
```

Aufgabe 3



- Merke: XPath und RDF/XML ist eine ganz schlechte Idee!
- Etwas besseres lernen wir in Kürze kennen...

Aufgabe 4

- Das Speicherschema von 3store sieht vor, zwischen Literalen und URIs im Objekt eines Tripels mit einem Flag zu unterscheiden. Warum gibt es ein solches Flag nicht für Subjekte und/oder Prädikate?

Subjekt (int64)	Prädikat (int64)	Objekt (int64)	ObjLiteral (bool)
0A7B22F834A043BC	423ABEF890432ABC	432FE8A8BE390A32	FALSE
432FE8A8BE390A32	B879EA890BA890FF	A3DE890234AB809E	FALSE
B879EA890BA890FF	B879EA890BA890FF	A3DE890234AB809E	FALSE
...	

- Weil Literale in RDF nur als Objekt erlaubt sind!

Aufgabe 5

- Erstellen Sie eine kleine HTML-Seite über ein Buch, einen Film o.ä. Ihrer Wahl und fügen Sie Informationen mit RDFa hinzu.

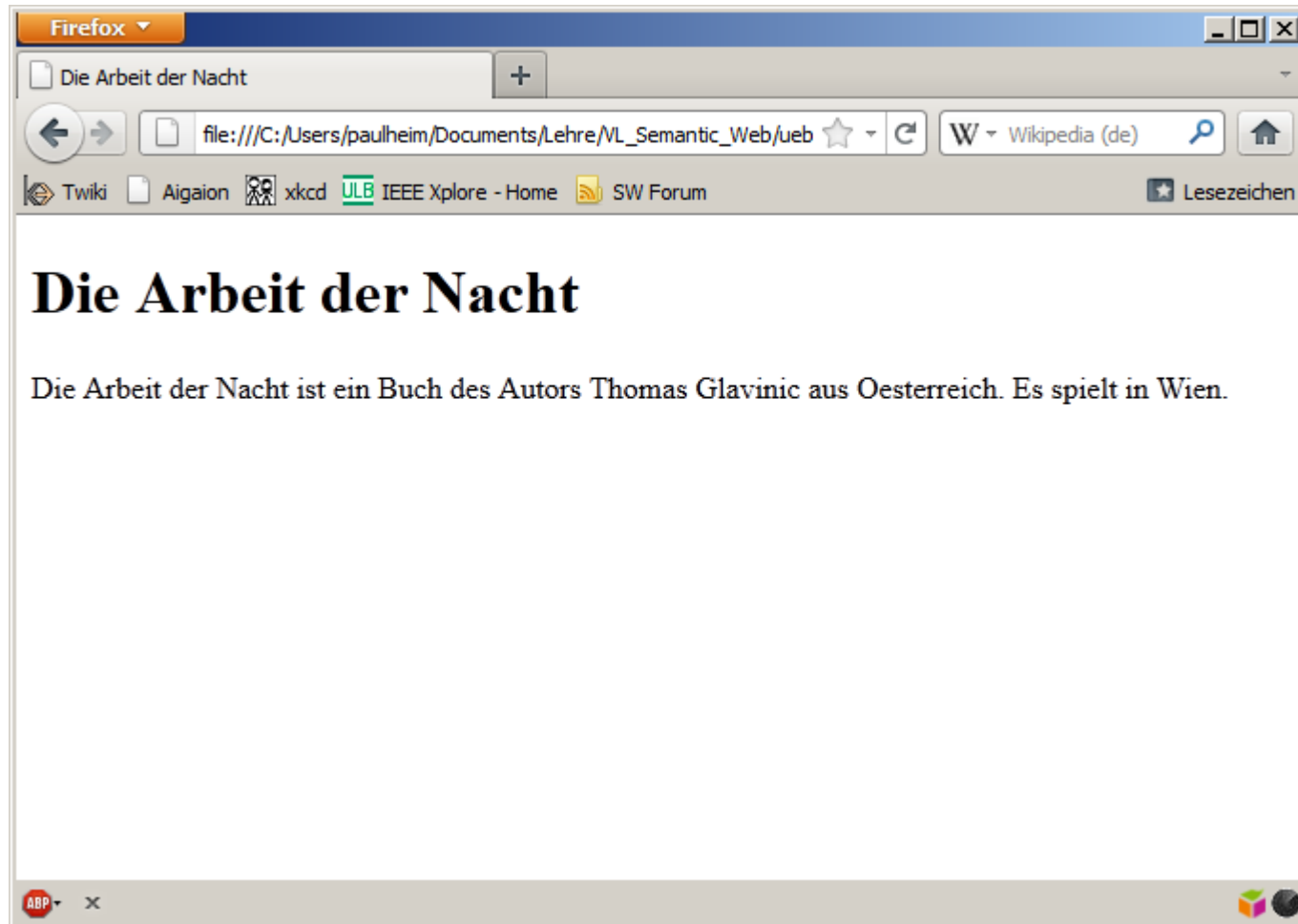
Aufgabe 5



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
  "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:ex="http://example.org/"
  version="XHTML+RDFa 1.0" xml:lang="de">
<head>
  <title>Die Arbeit der Nacht</title>
  <base href="http://example.org/#dadn" />
</head>
<body about="http://example.org/#dadn">
  <h1>Die Arbeit der Nacht</h1>
  <p><span property="ex:title">Die Arbeit der Nacht</span> ist ein Buch des
    Autors <span rel="ex:autor" href="ex:glavinic"><span property="ex:name">Thomas
    Glavinic</span> aus <span property="ex:kommtAus">Oesterreich</span></span>. Es
    spielt in <span property="ex:spieltIn">Wien</span>.</p>
  <span about="ex:glavinic" typeof="ex:Autor"/>
</body>
</html>
```



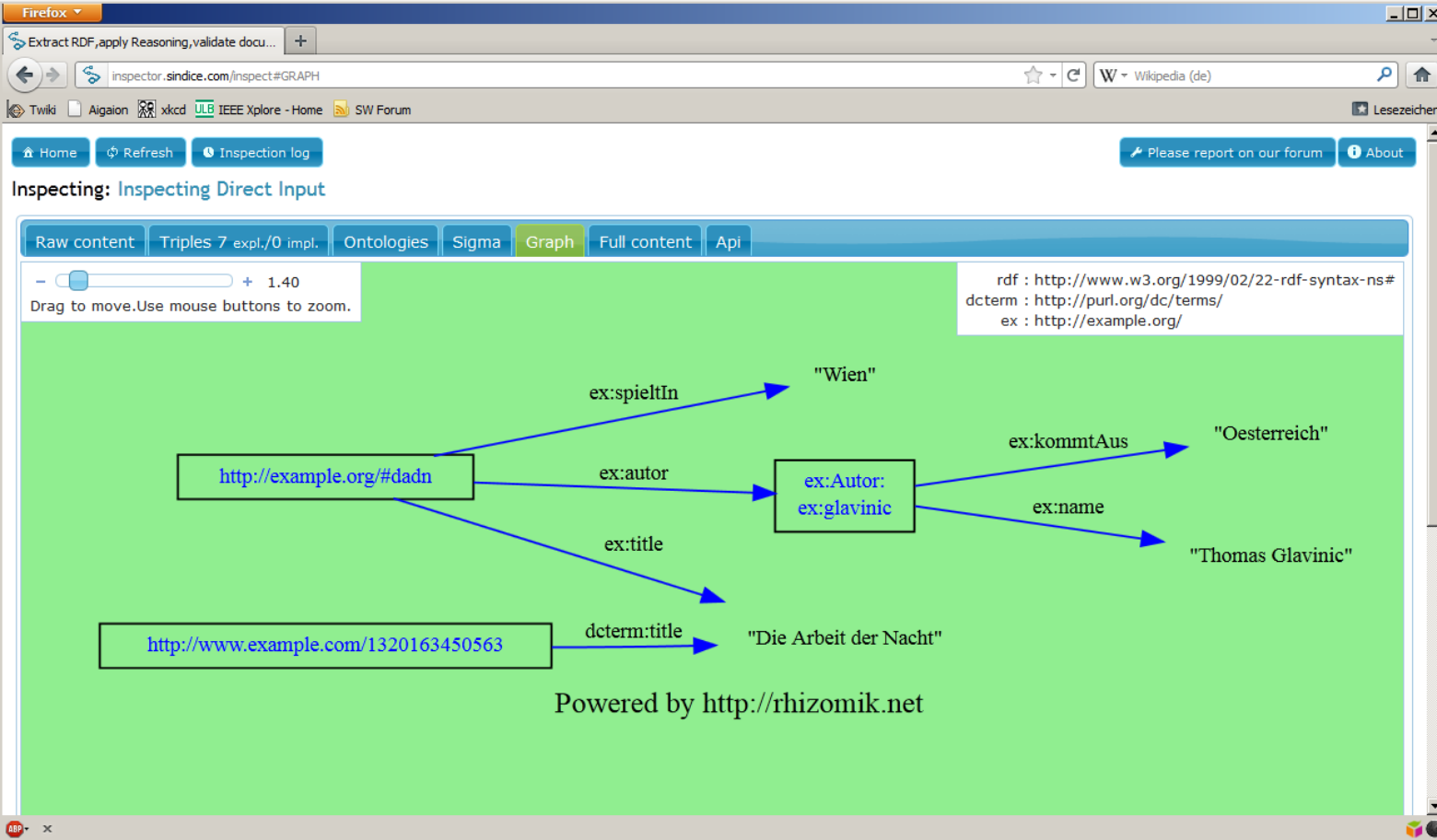
Aufgabe 5



Aufgabe 5

http://www.ke.tu-darmstadt.de/~paulheim/rdfa_example.xhtml

Aufgabe 5



Firefox

Extract RDF, apply Reasoning, validate docu... +

inspector.sindice.com/inspect#GRAPH

W - Wikipedia (de)

Home Refresh Inspection log

Please report on our forum About

Inspecting: Inspecting Direct Input

Raw content Triples 7 expl./0 impl. Ontologies Sigma Graph Full content Api

- 1.40
Drag to move. Use mouse buttons to zoom.

Legend:
rdf : <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
dcterm : <http://purl.org/dc/terms/>
ex : <http://example.org/>

Graph nodes and edges:
- Node: <http://example.org/#dadn>
- Node: <http://www.example.com/1320163450563>
- Node: ex:Autor: ex:glavinic
- Node: "Wien"
- Node: "Oesterreich"
- Node: "Thomas Glavinic"
- Node: "Die Arbeit der Nacht"

Edges:
- <http://example.org/#dadn> -- ex:spieltIn --> "Wien"
- <http://example.org/#dadn> -- ex:autor --> ex:Autor: ex:glavinic
- <http://example.org/#dadn> -- ex:title --> "Thomas Glavinic"
- <http://www.example.com/1320163450563> -- dcterm:title --> "Die Arbeit der Nacht"
- ex:Autor: ex:glavinic -- ex:kommtAus --> "Oesterreich"
- ex:Autor: ex:glavinic -- ex:name --> "Thomas Glavinic"

Powered by <http://rhizomik.net>

Vorlesung Semantic Web



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Vorlesung im Wintersemester 2012/2013

Dr. Heiko Paulheim

Fachgebiet Knowledge Engineering