

Vorlesung Semantic Web



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Vorlesung im Wintersemester 2012/2013

Dr. Heiko Paulheim

Fachgebiet Knowledge Engineering

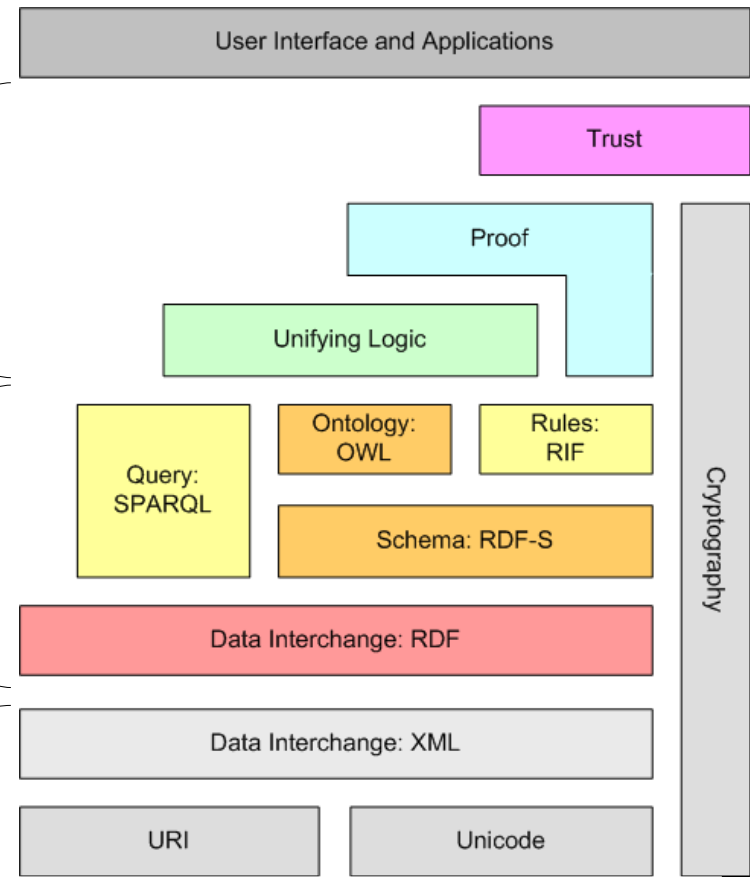
Semantic Web – Aufbau



here be dragons...

Semantic-Web-
Technologie
(Fokus der Vorlesung)

Technische
Grundlagen



Berners-Lee (2009): *Semantic Web and Linked Data*
<http://www.w3.org/2009/Talks/0120-campus-party-tbl/>

Was bisher geschah



- RDF: eine Sprache zur Beschreibung von Informationen
 - Graphenbasiert
 - Menge von Aussagen: Subjekt, Prädikat, Objekt
 - Notationen: RDF/XML, N3
- Besondere Sprachmittel
 - Leere Knoten
 - Mengen und Listen
 - Reifikation
- Semantik
 - Non-unique name assumption
 - Open world assumption

Was heute passiert

- Verarbeitung von RDF
 - Speicherung
 - Darstellung
- Kombination von RDF und HTML
- Einige Beispiele von RDF in der freien Wildbahn

RDF-Tools: Speicherung

- RDF ist eine Menge von Tripeln
- RDF-Speicher werden daher auch "Triple Store" genannt
- Unterschiedlichste Systeme
 - native Implementierung
 - Nutzung einer relationalen Datenbank

Naiver Triple Store

- naive Implementierung mit einer Tabelle
 - drei String-Spalten

Subjekt	Prädikat	Objekt
<http://foo.bar/Peter>	<http://foo.bar/vaterVon>	<http://foo.bar/Stefan>
<http://foo.bar/Peter>	<http://foo.bar/name>	"Peter Müller"
<http://foo.bar/Peter>	<rdf:type>	<http://foo.bar/Person>
<http://foo.bar/Stefan>	<rdf:type>	<http://foo.bar/Person>
<http://foo.bar/Peter>	<http://foo.bar/vaterVon>	<http://foo.bar/Julia>
<http://foo.bar/Peter>	<http://foo.bar/kennt>	_:genID01
_:genID01	<http://foo.bar/vaterVon>	<http://foo.bar/Markus>
...

Naiver Triple Store

- Abfragemechanismus
 - "Wer ist der Vater von Julia?"

```
SELECT subjekt FROM triples
  WHERE praedikat = '<http://foo.bar/vaterVon>'
  AND    objekt = '<http://foo.bar/Julia>'
```

- Wer kennt den Vater von Julia?

```
SELECT t1.subjekt FROM triples AS t1, triples AS t2
  WHERE t1.praedikat = '<http://foo.bar/kennt>'
  AND   t1.objekt = t2.subjekt
  AND   t2.praedikat = '<http://foo.bar/vaterVon>'
  AND   t2.objekt = '<http://foo.bar/Julia>'
```

Naiver Triple Store

- Jede komplexe Abfrage führt zu einem Self-Join

```
SELECT t1.subjekt FROM triples AS t1, triples AS t2
WHERE t1.praedikat = '<http://foo.bar/kennt>'
AND t1.objekt = t2.subjekt
AND t2.praedikat = '<http://foo.bar/vaterVon>'
AND t2.objekt = '<http://foo.bar/Julia>'
```

t1.Subjekt	t1.Prädikat	t1.Objekt	t2.Subjekt	t2.Prädikat	t2.Objekt
...
...
...

Naiver Triple Store

- Abfragemechanismus:
 - "Wer ist Vater, Mutter, Bruder, Schwester von Julia?"

```
SELECT praedikat,objekt FROM triples
WHERE (      praedikat = '<http://foo.bar/vaterVon>'
        OR praedikat = '<http://foo.bar/mutterVon>'
        OR praedikat = '<http://foo.bar/bruderVon>'
        OR praedikat = '<http://foo.bar/schwesterVon>')
AND      objekt = '<http://foo.bar/Julia>'
```

- Derartige Abfragen führen zu vielen gereihten OR-Statements

Naiver Triple Store

- Abfragemechanismus
 - Welche Informationen gibt es über Julia?

```
SELECT subjekt,praedikat,objekt FROM triples
WHERE subjekt='<http://foo.bar/Julia>'
SELECT subjekt,praedikat,objekt FROM triples
WHERE objekt='<http://foo.bar/Julia>'
```

Speicherung mit Property Table

- Property Table
 - Ein Eintrag pro Subjekt (statt pro Tripel)
 - eine Spalte pro Prädikat

Subjekt	rdf:type	foo:vaterVon	foo:kennt
foo:Peter	foo:Person	foo:Stefan	NULL
foo:Peter	NULL	foo:Julia	NULL
foo:Stefan	foo:Person	NULL	_:genID01
_:genID01	foo:Person	foo:Markus	NULL
...

Speicherung mit Property Table

- Abfragemechanismus:
 - "Wer ist der Vater von Julia?"

```
SELECT subjekt FROM properties WHERE  
    foo_vaterVon = "http://foo.bar/Julia"
```

- "Wer kennt den Vater von Julia?"

```
SELECT p1.subjekt FROM properties AS p1, properties AS p2  
WHERE  
    p1.foo_kennt p2.subjekt AND  
    p2.foo_vaterVon = "http://foo.bar/Julia"
```

Speicherung mit Property Table



- Abfragemechanismus:
 - "Welche Informationen gibt es über Julia?"
 - Mit Julia als Subjekt ist das trivial:

```
SELECT * FROM properties WHERE
    subjekt = "http://foo.bar/Julia"
```

- Mit Julia als Objekt nicht:

```
SELECT 'vaterVon',foo_vaterVon FROM properties WHERE
    objekt = "http://foo.bar/Julia"
```

```
SELECT 'kennt',foo_kennt FROM properties WHERE
    objekt = "http://foo.bar/Julia"
```

```
SELECT ...
```

```
...
```

Speicherung mit Property Table



- Geschwindigkeit
 - kürzere Tabelle → z.T. effizientere Abfragen
 - gut für Anfragen von ausgehenden Kanten
- Speicherbedarf
 - sehr kompakte Speicherung für ähnliche Ressourcen
 - schlecht bei unähnlichen Ressourcen
 - viele NULL-Werte
 - Probleme mit Relationen von hoher Kardinalität
 - z.B. "kennt"

Speicherung mit vertikaler Partitionierung

- Vertikale Partitionierung
 - Variante von Property-Table
 - eine Tabelle pro Property

Subjekt	rdf:type
foo:Peter	foo:Person
_:genID01	foo:Person
...	...

Subjekt	foo:vaterVon
foo:Peter	foo:Stefan
foo:Peter	foo:Julia
_:genID01	foo:Markus
...	...

Subjekt	foo:kennt
foo:Stefan	_:genID01
...	

Speicherung mit vertikaler Partitionierung

- Abfragemechanismus:
 - "Wer ist der Vater von Julia?"

```
SELECT subjekt FROM foo_vaterVon WHERE  
    objekt = '<http://foo.bar/Julia>'
```

- "Wer kennt den Vater von Julia?"

```
SELECT foo_kennt.subjekt FROM foo_kennt,foo_vaterVon WHERE  
    foo_kennt.objekt = foo_vaterVon.subjekt  
AND    foo_vaterVon.objekt = '<http://foo.bar/Julia>'
```


Speicherung mit vertikaler Partitionierung

- Abfragemechanismus:
 - "Welche Informationen gibt es über Julia?"

```
SELECT subjekt, 'foo:vaterVon', objekt FROM foo_vaterVon  
WHERE
```

```
    subjekt = '<http://foo.bar/Julia>'
```

```
SELECT subjekt, 'foo:vaterVon', objekt FROM foo_vaterVon  
WHERE
```

```
    objekt = '<http://foo.bar/Julia>'
```

```
SELECT subjekt, 'foo:kennt', objekt FROM foo_kennt  
WHERE
```

...

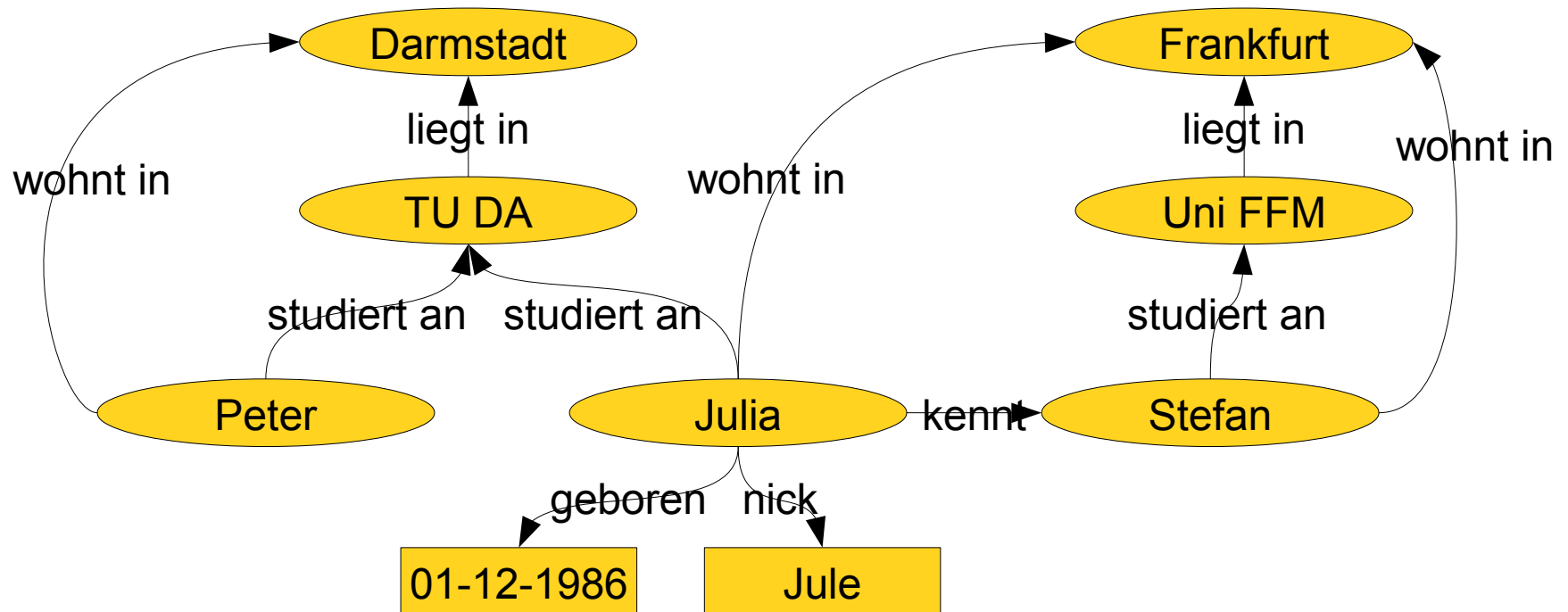
Speicherung mit vertikaler Partitionierung



- Geschwindigkeit
 - sehr kleine Tabellen → schnelle Joins
 - dafür oft viele einzelne Abfragen nötig
- Speicherbedarf
 - kompakte Speicherung bei sehr verschiedenen Ressourcen
 - gut bei Relationen hoher Kardinalität
 - etwas schlechter für ähnliche Ressourcen

Beispiel

- Ein sozialer Graph soll gespeichert werden



Beispiel



- Ein sozialer Graph soll gespeichert werden
 - 10000 Personen mit Wohnort, Uni, Nickname, Geburtsdatum
 - Jede Person kennt im Mittel 50 andere Personen
 - 100 Universitäten mit Ort
- Wie groß ist der Speicherbedarf?
- Wie aufwändig sind die Abfragen...
 - Finde alles über eine Person heraus
 - Erstelle eine Liste aller Nicknames und Geburtstage zu Freunden einer Person
 - Finde alle Personen, die in der Stadt wohnen, in der sie studieren

Beispiel: Naiver Triple Store



- Speicherbedarf
 - 10000 Personen mit Wohnort, Uni, Nickname, Geburtsdatum
 - $10000 * 4$ Tripel
 - Jede Person kennt im Mittel 50 andere Personen
 - $10000 * 50$ Tripel
 - 100 Universitäten mit Ort
 - 100 Tripel
- Gesamt: Tripel-Tabelle mit ~ 540.000 Einträgen
 - 1,6 Mio Strings
 - keine Null-Werte

Beispiel: Naiver Triple Store



- Finde alles über eine Person heraus

```
SELECT pred obj FROM triples WHERE subj="person"  
SELECT subj pred FROM triples WHERE obj="person"
```

- Erstelle eine Liste aller Nicknames und Geburtstage zu Freunden einer Person

```
SELECT tab2.obj AS nick, tab3.obj AS geburstag  
FROM triples AS tab1, triples AS tab2, triples AS tab3  
WHERE  
tab1.subj = "ich"; tab1.pred = "kennt";,  
tab1.obj = tab2.subj, tab2.pred = "nick",  
tab1.obj = tab3.subj, tab3.pred = "geburstag"
```

Beispiel: Naiver Triple Store



- Finde alle Personen, die in der Stadt wohnen, in der sie studieren

```
SELECT tab1.subj
FROM tripel AS tab1, tripel AS tab2, tripel AS tab3
WHERE
tab1.pred = "studiertAn",
tab1.obj = tab2.subj, tab2.pred = "liegt_in",
tab2.obj = tab3.obj, tab3.pred = "wohntIn"
tab1.subj = tab3.subj
```

Beispiel: Property Table

- Sechs verschiedene Prädikate
- 10000 Personen: je 50 Einträge
- 100 Universitäten: je 1 Eintrag

- Tabelle mit ~ 500000 Einträgen
 - $\sim 3,5$ Mio. Strings, davon viele Null-Werte

Beispiel: Property Table

- Finde alles über eine Person heraus
SELECT studiert_an, geboren, nick, kennt, wohnt_in FROM person
WHERE subj="person"
SELECT kennt FROM person WHERE obj="person"
- Erstelle eine Liste aller Nicknames und Geburtstage
zu Freunden einer Person
SELECT tab2.geburtstag, tab2.nickname FROM triples AS tab1, triples
AS tab2 WHERE
tab1.subj = "ich", tab1.kennt = tab2.subj

Beispiel: Property Table

- Finde alle Personen, die in der Stadt wohnen, in der sie studieren

```
SELECT tab1.subj FROM tripel AS tab1, tripel AS tab2, tripel AS tab3
WHERE
    tab1.studiertAn = tab2.subj
    tab2.liegtIn = tab1.wohntIn
```

Beispiel: vertikale Partitionierung



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Tabelle wohnt_in, studiert_in, geboren, nick: je 10000 Einträge
- Tabelle kennt: 500000 Einträge
- Tabelle liegt_in: 100 Einträge

- 6 Tabellen à 2 Spalten, ~540000 Einträge
- ~1,1 Mio Strings, keine Null-Werte

Beispiel: vertikale Partitionierung



- Finde alles über eine Person heraus

```
SELECT nick, FROM nick WHERE subj="person"
```

```
SELECT geboren FROM geboren WHERE subj="person"
```

```
SELECT studiertAn FROM studiertAn WHERE subj="person"
```

```
SELECT wohntIn FROM wohntIn WHERE subj="person"
```

```
SELECT kennt FROM kennt WHERE subj="person"
```

```
SELECT subj FROM kennt WHERE kennt="person"
```

- Erstelle eine Liste aller Nicknames und Geburtstage zu Freunden einer Person

- ```
SELECT nickname.object, geburtstag.object FROM kennt, nickname, geburtstag
```

```
WHERE
```

```
 kennt.subj = "ich", kennt.obj = nickname.subj,
```

```
 kennt.obj = geburtstag.subj
```

# Beispiel: vertikale Partitionierung



- Finde alle Personen, die in der Stadt wohnen, in der sie studieren

```
SELECT studiertAn.subj FROM studiertAn, liegtIn, wohntIn
WHERE
 studiertAn.obj = liegtIn.subj,
 liegtIn.subj = wohnortVon.subj,
 wohntInSubj = studiertAn.subj
```

# Vergleich der Lösungen

- Speicherbedarf:
  - Naiver Triple Store: 1,6 Mio Strings
  - Property Table: 3,5 Mio Strings
  - Vertikale Partitionierung: 1,1 Mio Strings
- Abfragen
  - Naiver Triple Store: große Joins, einfache „offene“ Abfrage
  - Property Table: wenige Joins
  - Vertikale Partitionierung: viele Joins, schwierige „offene“ Abfrage
- Fazit:
  - Trade-off zwischen Speicherplatz und Geschwindigkeit
  - Kein optimales Modell für *alle* Abfragen

# RDF-Tools: Speicherung

- Gemischte Lösungen
  - aus allen vorher kennen gelernten
  - z.B. Einzeltabellen für häufig verwendete Prädikate

| Subjekt                | Prädikat                  | Objekt                  |
|------------------------|---------------------------|-------------------------|
| <http://foo.bar/Peter> | <http://foo.bar/vaterVon> | <http://foo.bar/Stefan> |
| <http://foo.bar/Peter> | <http://foo.bar/vaterVon> | <http://foo.bar/Julia>  |
| <http://foo.bar/Peter> | <http://foo.bar/vaterVon> | <http://foo.bar/Markus> |
| _:genID01              |                           |                         |
| ...                    |                           |                         |

| Subjekt                 | Prädikat | Objekt                  |
|-------------------------|----------|-------------------------|
| <http://foo.bar/Peter>  | rdf:type | <http://foo.bar/Person> |
| <http://foo.bar/Stefan> |          | <http://foo.bar/Person> |
| <http://foo.bar/Julia>  |          | <http://foo.bar/Person> |
| ...                     |          | ...                     |

# RDF-Tools: Speicherung

- Performance-Verbesserung mit Indexierung
  - Problem: Vergleiche über URIs/Literale aufwändig
  - Lösung z.B.: Nutzen von Hashes
    - String-Vergleiche können durch int-Vergleiche ersetzt werden
  - Beispiel 3Store: URI-/Literal-Objekte werden mit Flag unterschieden

| Subjekt (int64)  | Prädikat (int64) | Objekt (int64)          | ObjLiteral (bool) |
|------------------|------------------|-------------------------|-------------------|
| 0A7B22F834A043BC | 423ABEF890432ABC | 432FE8A8BE390A32        | FALSE             |
| 432FE8A8BE390A32 | B879EA890BA00000 | 432FE8A8BE390A32        | FALSE             |
| B879EA890BA00000 | hash (int64)     | URI                     | SE                |
| ...              | 0A7B22F834A043BC | <http://foo.bar/Peter>  | SE                |
|                  | 423ABEF890432ABC | <rdf:type>              |                   |
|                  | 432FE8A8BE390A32 | <http://foo.bar/Person> |                   |
|                  | ...              | ...                     |                   |

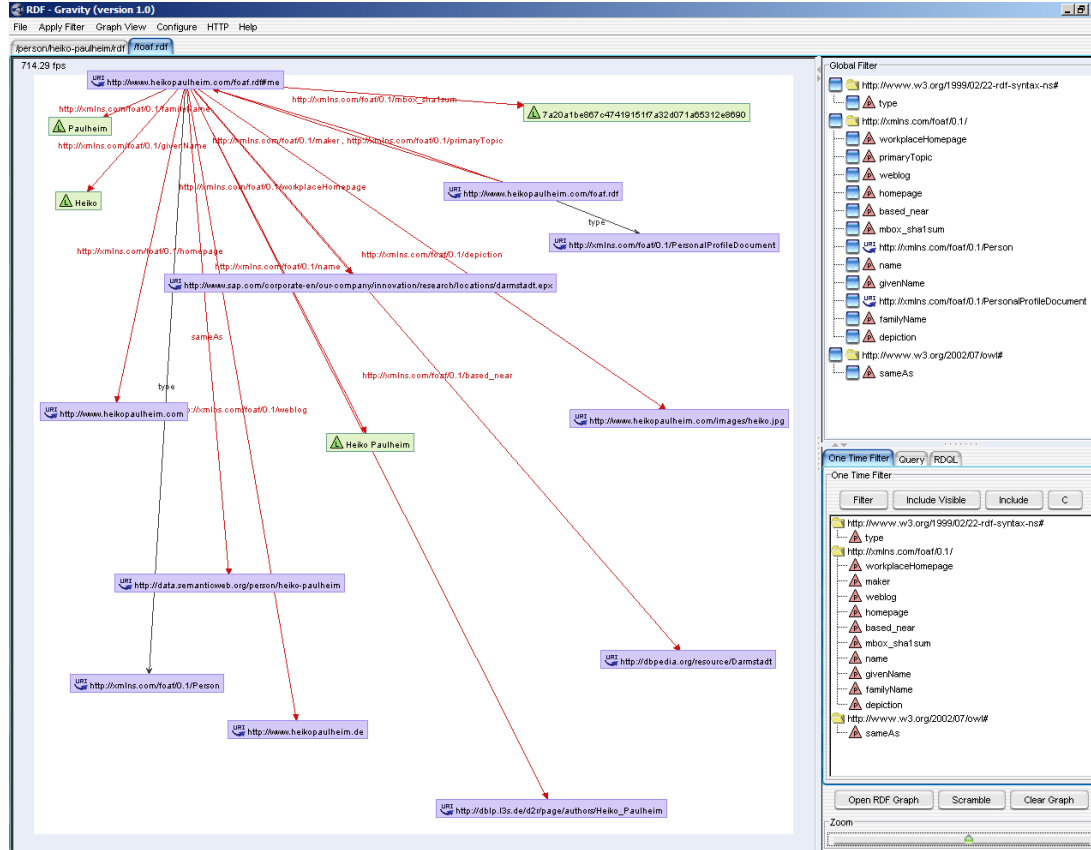


# RDF-Tools: Speicherung

- Hashing verhindert "teure" String-Vergleiche:
  - "...WHERE Subjekt = '<http://foo.bar/Peter>'" wird zu
  - "...WHERE Subjekt = hash('<http://foo.bar/Peter>')"
  - Hash-Funktion effizient zu berechnen

| Subjekt (int64)  | Prädikat (int64) | Objekt (int64)          | ObjLiteral (bool) |
|------------------|------------------|-------------------------|-------------------|
| 0A7B22F834A043BC | 423ABEF890432ABC | 432FE8A8BE390A32        | FALSE             |
| 432FE8A8BE390A32 | B879EA890BA...   | ...                     | FALSE             |
| B879EA890BA...   | hash (int64)     | URI                     | SE                |
| ...              | 0A7B22F834A043BC | <http://foo.bar/Peter>  | SE                |
| ...              | 423ABEF890432ABC | <rdf:type>              |                   |
| ...              | 432FE8A8BE390A32 | <http://foo.bar/Person> |                   |
| ...              | ...              | ...                     |                   |

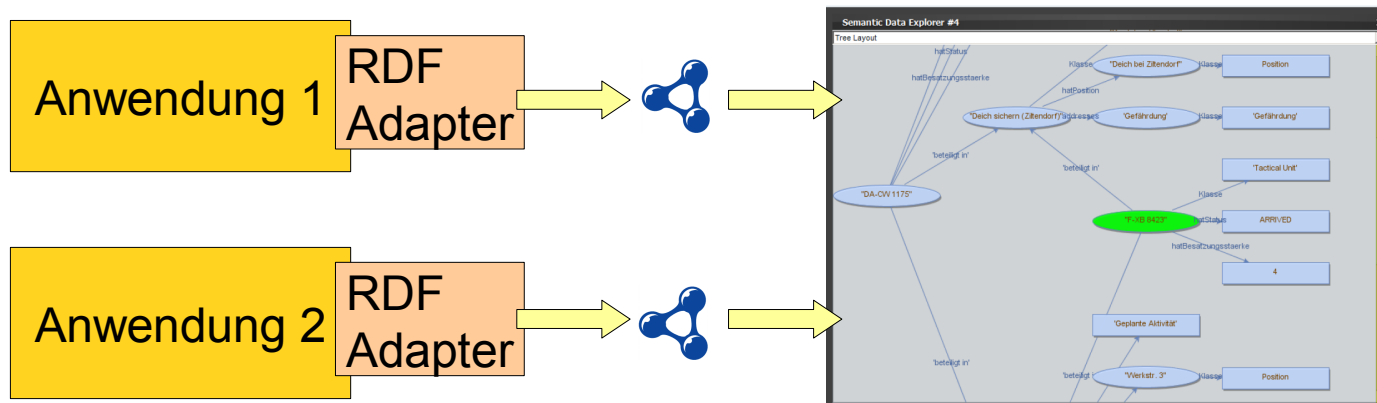
# RDF-Tools: Visualisierung



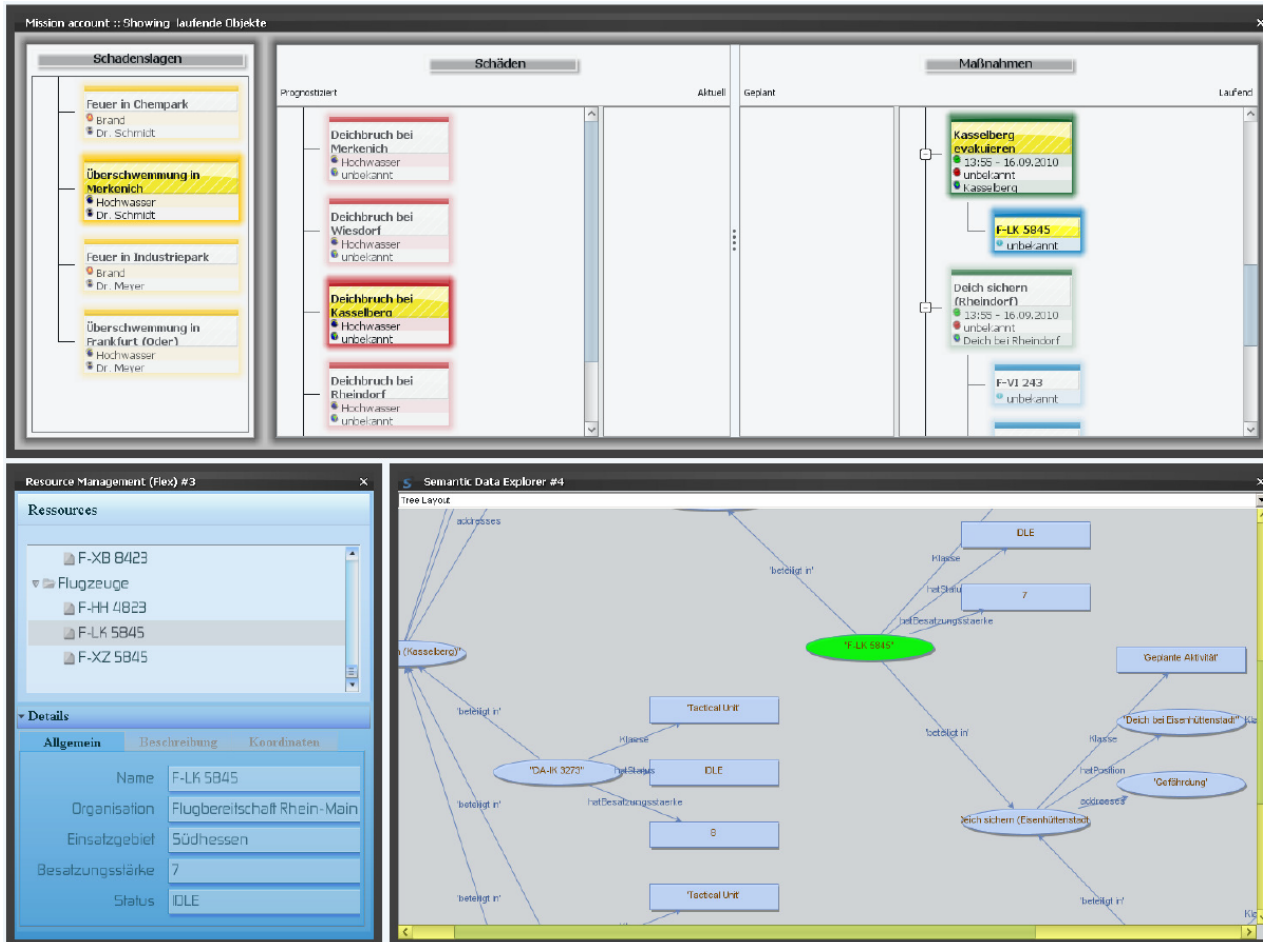
RDF Gravity  
<http://semweb.salzburgresearch.at/apps/rdf-gravity/>

# RDF-Tools: Visualisierung

- Experiment bei SAP Research 2010:
  - Daten aus Anwendungen als RDF zur Verfügung stellen
  - Den Graph direkt anzeigen
  - Verschränkte Interaktion:
    - Objekt im Graph wählen → wird in der Anwendung hervorgehoben
    - Objekt in der Anwendung wählen → wird im Graph hervorgehoben



# RDF-Tools: Visualisierung



The image displays two software interfaces used for mission management and data visualization.

**Mission account :: Showing laufende Objekte**

This interface is divided into three main sections:

- Schadenslagen (Damage States):** A list of damage events categorized by type and location, such as "Feuer in Chempark" (Fire in Chempark) and "Überschwemmung in Merkenich" (Flooding in Merkenich).
- Schäden (Damages):** A table showing the status of damages, categorized into "Prognostiziert" (Forecasted), "Aktuell" (Actual), and "Geplant" (Planned). Examples include "Deichbruch bei Merkenich" and "Deichbruch bei Kasselberg".
- Maßnahmen (Measures):** A table showing planned and ongoing measures, such as "Kasselberg evakuieren" (Evacuate Kasselberg) and "Deich sichern (Rheindorf)" (Secure dike (Rheindorf)).

**Resource Management (Flex) #3**

This interface shows a list of resources and their details. The resources listed are:

- F-XB 8423
- Flugzeuge (Aircraft)
- F-HH 4823
- F-LK 5845
- F-XZ 5845

The details for F-LK 5845 are shown below:

| Allgemein        | Beschreibung                | Koordinaten |
|------------------|-----------------------------|-------------|
| Name             | F-LK 5845                   |             |
| Organisation     | Flugbereitschaft Rhein-Main |             |
| Einsatzgebiet    | Südhessen                   |             |
| Besatzungsstärke | 7                           |             |
| Status           | IDLE                        |             |

**Semantic Data Explorer #4**

This interface shows a Semantic Data Explorer (RDF graph) with a tree layout. The graph illustrates relationships between various entities, including:

- Flugzeug (F-LK 5845):** The central entity, highlighted in green.
- Tactical Unit:** Multiple instances of tactical units are shown, with relationships like "hat Besatzungsstärke" (has crew strength) and "beteiligt in" (participates in).
- DLE (Darmstadt Luftwaffenstützpunkt):** A base or station entity.
- Geplante Aktivität (Planned Activity):** Activities like "Deich bei Eisenhüttenstadt" and "Gefährdung" (Hazard).
- Deich sichern (Eisenhüttenstadt):** A specific measure or activity.

# RDF-Tools: Visualisierung

- Experiment bei SAP Research 2010:
  - Nutzer müssen verschieden komplexe Aufgaben lösen
  - einmal mit, einmal ohne RDF-Visualisierung
- Ergebnis:
  - Für komplexe Tasks signifikant geringere Bearbeitungszeit
  - Signifikant höhere Nutzerzufriedenheit

# RDF-Tools: Validierung und Konvertierung

- W3C RDF Validator:
  - W3C RDF Validator: <http://www.w3.org/RDF/Validator/>
  - Gibt RDF/XML in Tripeln und als Graph aus
  
- RDF About Validator:
  - <http://www.rdfabout.com/demo/validator/>
  - kann auch N3

# RDF-Tools (weitere)

- Programmierframeworks
  - zum Entwickeln RDF-basierter Anwendungen
  - z.B. JENA, RDFReactor, ...
- Reasoner
  - können Schlussfolgerungen auf RDF-Graphen ziehen
  - können Anfragen beantworten
- Werden wir noch kennen lernen

- Das semantische Web verwendet also RDF
- Das "klassische" Web verwendet HTML
- Muss man jetzt jede Information doppelt codieren?

```
<html>
...
Dr. Mark Smith
<i>Physician</i>
Main St. 14
Smalltown
Mon-Fri 9-11 am
Wed 3-6 pm
...
</html>
```

```
:p a :Physician .
:p :hasDegree "Dr." .
:p :hasName "Mark Smith" .
:p :hasAddress :a .
:a :street "Main Street" .
:a :number "14"^^xsd:int .
:a :city "Smalltown" .
:p :hasOpeningHours [
 a rdf:Bag ;
 [:day :Monday;
 :from "9"^^xsd:int;
 :to "11"^^xsd:int;
]
...

```



# Gemeinsame Verwendung von HTML und RDF – Variante 1

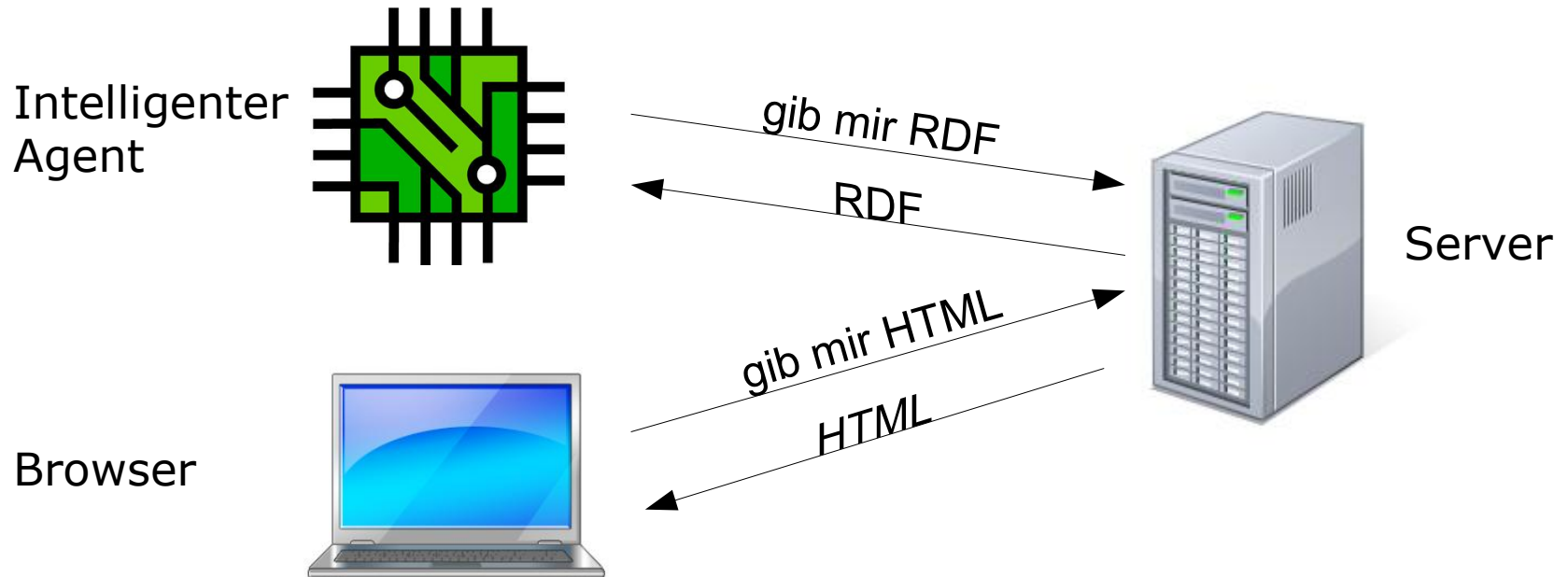
- Verweis auf RDF-Beschreibung – haben wir schon mal gesehen

```
<html>
 <head>
 <link rel="meta" type="application/rdf+xml" title="DC" href="dc.rdf" />
 </head>
 <body>
 ...
 </body>
</html>
```

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:dc="http://purl.org/dc/elements/1.1/">
 <rdf:Description rdf:about="http://www.ke.tu-darmstadt.de/lehre/semantic-web">
 <dc:publisher>TU Darmstadt</dc:publisher>
 <dc:subject>Semantic Web</dc:subject>
 <dc:creator>Heiko Paulheim</dc:creator>
 <dc:relation rdf:resource="http://www.w3.org/2001/sw/" />
 </rdf:Description>
</rdf:RDF>
```

# Parallele Verwendung von HTML und RDF – Variante 2

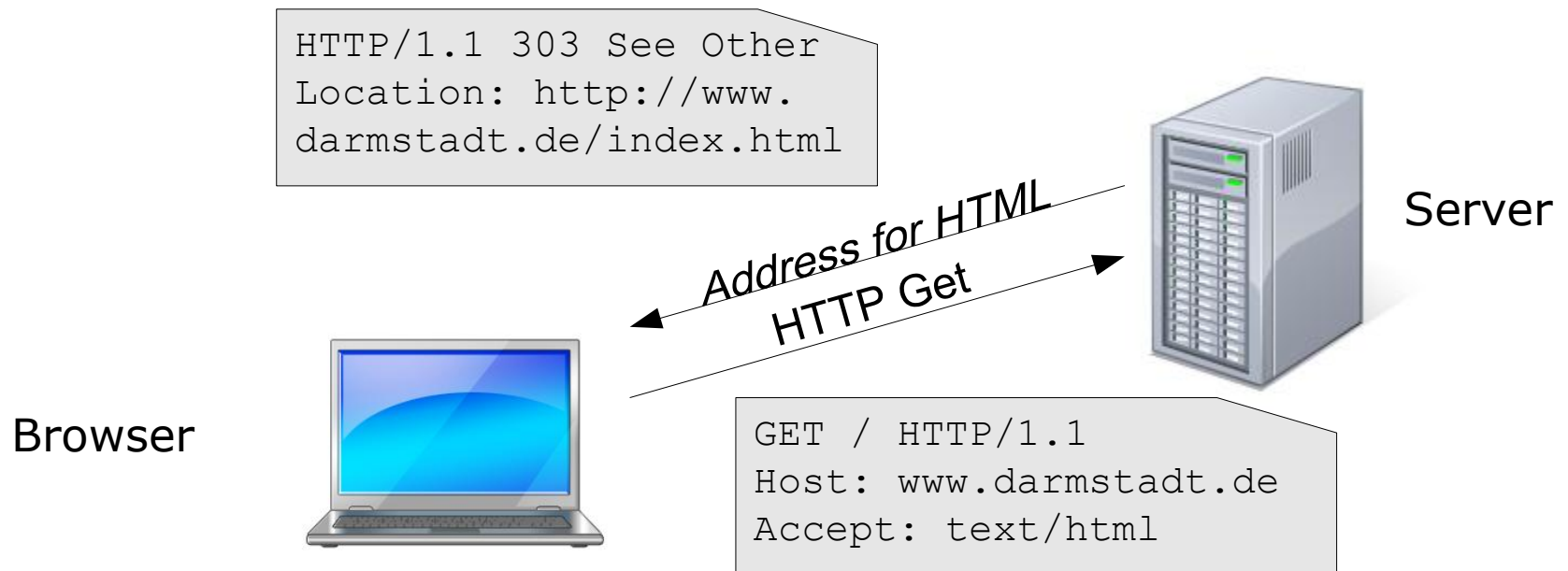
- Content Negotiation



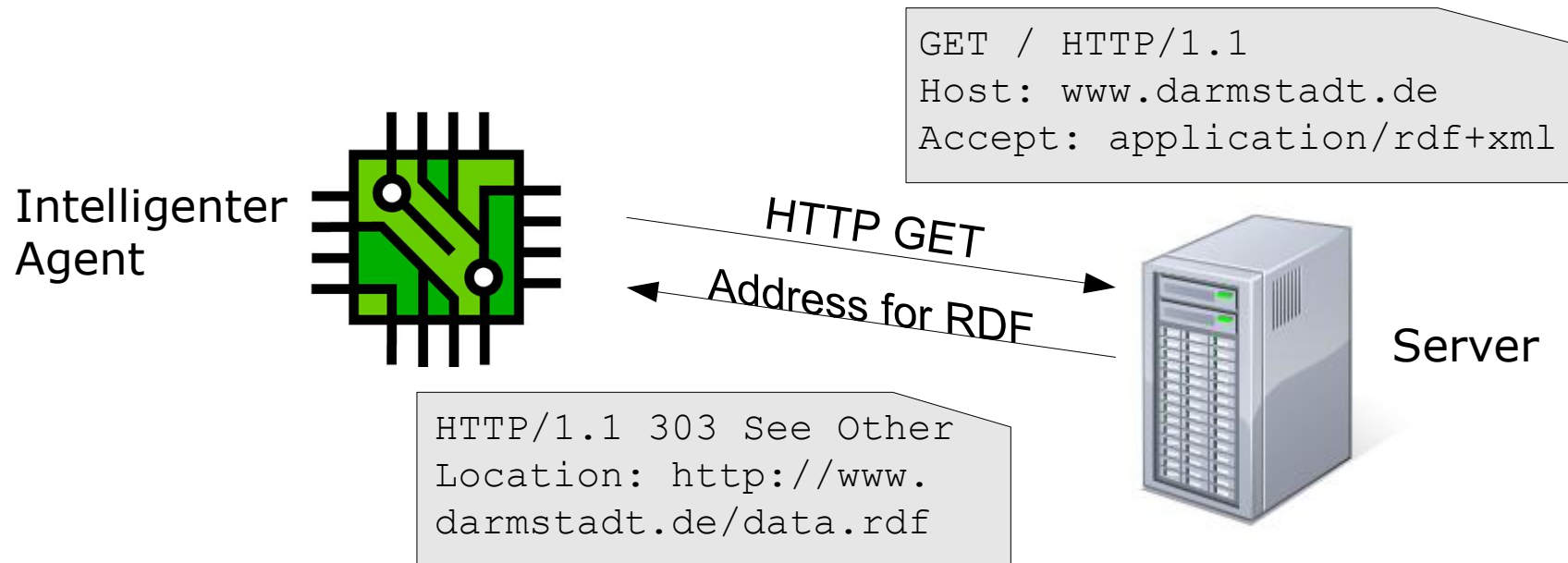
# Content Negotiation im Detail



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



# Content Negotiation im Detail



# Content Negotiation: MIME Types



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- MIME: Multipurpose Internet Mail Extensions
- Ursprünglich zur Kennzeichnung von Mail-Anhängen
  - Text, PDF, ..
- Erste Version: 1996
- Werden von der IANA verwaltet
  
- Für Semantic Web wichtige MIME Types:
  - application/rdf+xml
  - text/turtle
  - text/n3
  - application/sparql-query
  - application/sparql-results+xml



# Gemeinsame Verwendung von HTML und RDF

- Link auf RDF-Dokument
  - Umsetzung mit HTML-Bordmitteln möglich
- Content Negotiation
  - Aufwändigere Maßnahme
  - *Ein* URI für verschiedene Repräsentationen möglich
- In beiden Fällen
  - Zwei getrennte Repräsentationen nötig
  - "Doppelte Buchführung"
  - Potentielle Inkonsistenz

# RDF in Attributes (RDFa)

- Idee hinter RDF in Attributes (RDFa):
  - Warum nicht HTML und RDF in *einem* Dokument gemeinsam nutzen?
  - Information muss nur einmal codiert werden
- RDFa kombiniert XHTML mit RDF



```
<html>
...
<body about="http://www.marcsmith.com/MarcSmith">
 Dr. Mark Smith
 <i>Physician</i>

 Main Street
 14
 Smalltown

 ...
</body>
</html>
```

# Sprachmittel von RDFa



- `about="http://foo.bar/aSubject"`
  - Legt das Subjekt einer Seite oder eines Unterabschnittes fest
- `property = "http://foo.bar/aProperty"`
  - Definiert eine Datenrelation
  - Inhalte des Tags werden als Literal interpretiert
- `rel = "http://foo.bar/aRelation"`
  - Definiert eine Relation zu einer anderen Ressource
- `href = "http://foo.bar/aResource"`
  - Definiert das Objekt einer Relation
  - kann wieder Subjekt werden
- `typeof = "http://foo.bar/aType"`
  - legt den Typ (`rdf:type`) einer Ressource fest



# RDF in Attributes (RDFa)

```
<html>
...
<body about="http://www.marcsmith.com/MarcSmith">
 Dr. Mark Smith
 <i>Physician</i>

 Main Street
 14
 Smalltown

...
</body>
</html>
```

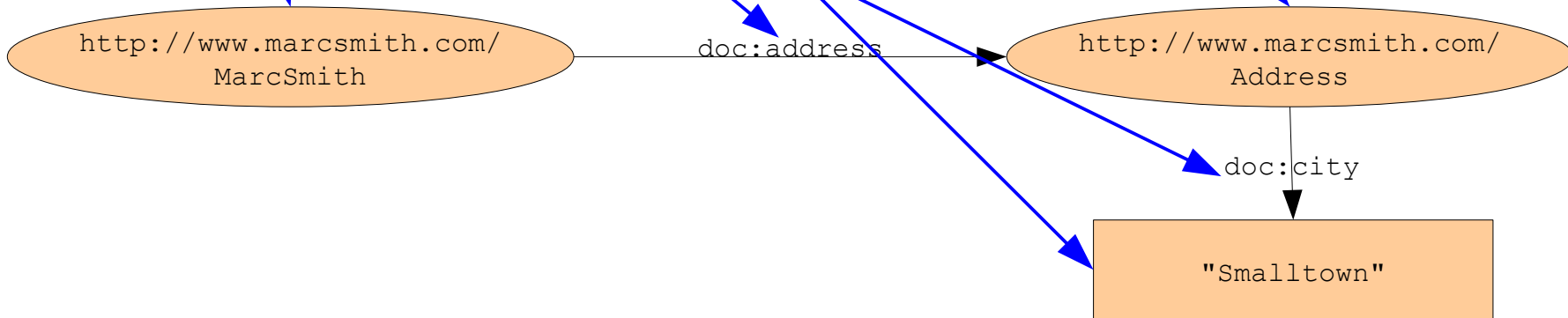


# RDF in Attributes (RDFa)

```
<html>
...
<body about="http://www.marcsmith.com/MarcSmith">
 Dr. Mark Smith
 <i>Physician</i>

 Main Street
 14
 Smalltown

...
</body>
</html>
```



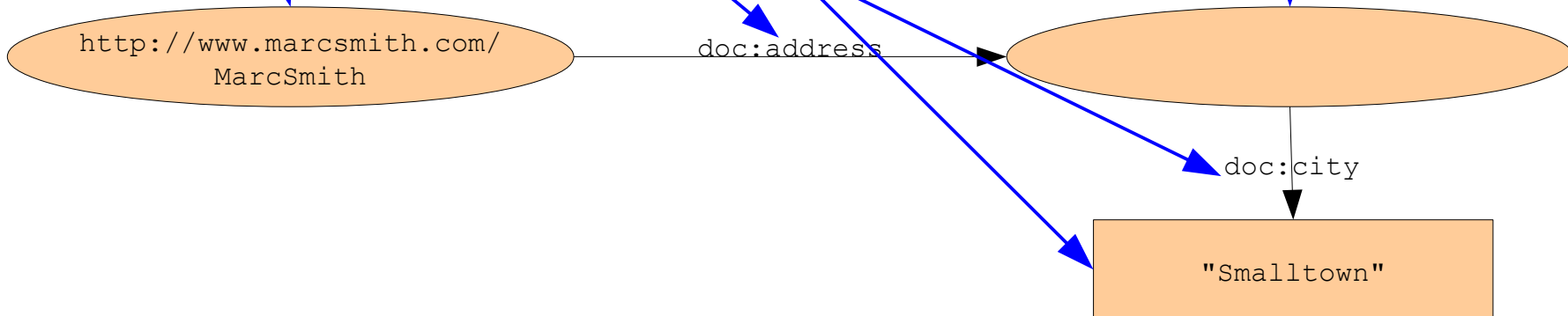
# RDF in Attributes (RDFa)

```
<html>
...
<body about="http://www.marcsmith.com/MarcSmith">
 Dr. Mark Smith
 <i>Physician</i>

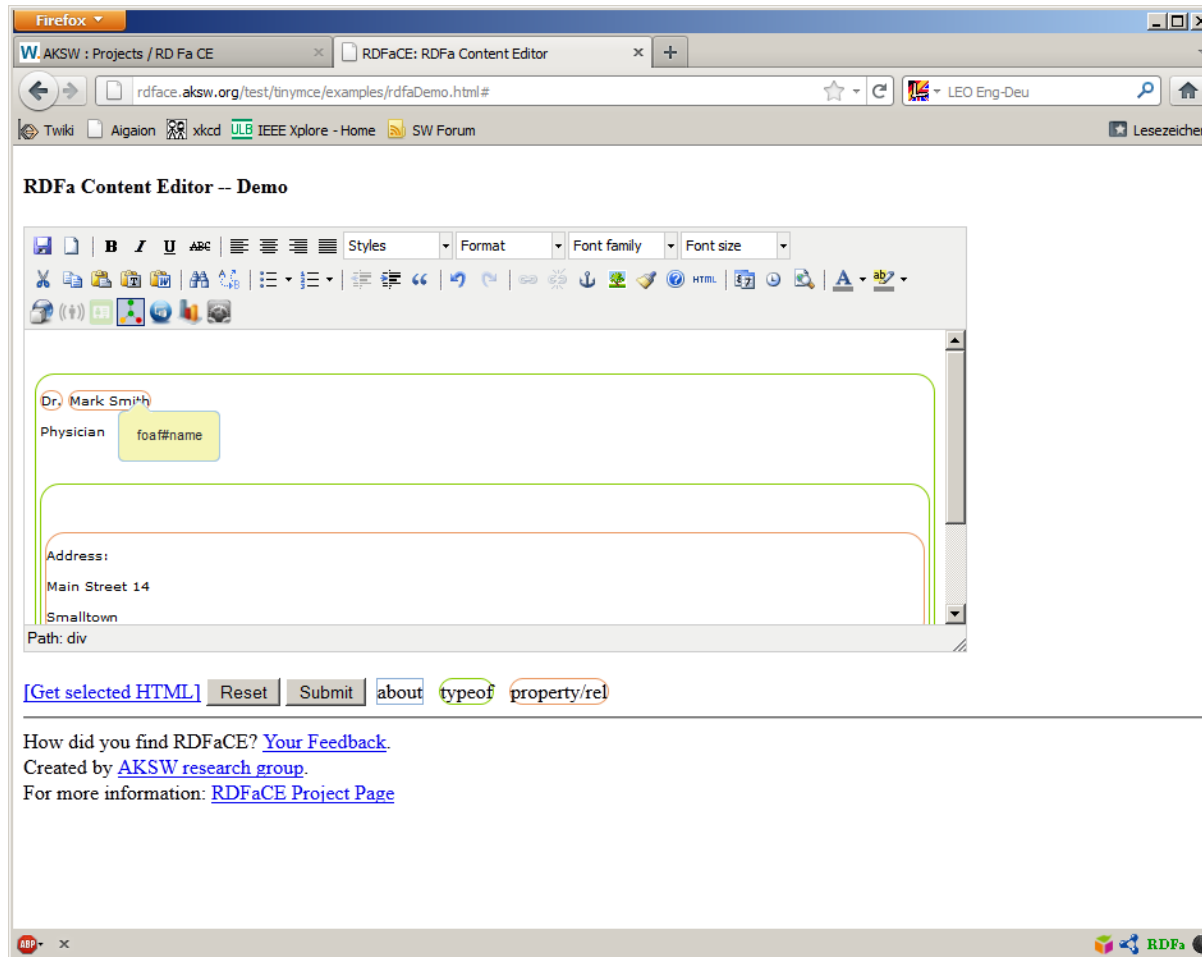
 Main Street
 14
 Smalltown

...
</body>
</html>
```

Relationen ohne  
"href" werden zu  
einem  
leeren  
Knoten!



# RDFaCE



The screenshot shows a Firefox browser window with the title "RDFaCE: RDFa Content Editor". The address bar shows the URL "rdface.aksw.org/test/tinymce/examples/rdfaDemo.html#". The page content includes a rich text editor with the following text: "Dr. Mark Smith", "Physician foaf#name", "Address:", "Main Street 14", and "Smalltown". Below the editor, there are buttons for "[Get selected HTML]", "Reset", "Submit", "about", "typeof", and "property/rel". At the bottom of the page, there is a footer with the text: "How did you find RDFaCE? [Your Feedback](#). Created by [AKSW research group](#). For more information: [RDFaCE Project Page](#)".

# RDF in der freien Wildbahn: FOAF-Profile



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

The screenshot shows a Firefox browser window displaying the homepage of Tom Heath. The browser's address bar shows the URL <http://people.kmi.open.ac.uk/tom/df>. The page content includes a navigation menu with links for Home, Blog, Research, Publications, PhD Thesis, Talks, Roles, Other Stuff, and Contact Details. The main content area is titled "Home - Tom Heath" and contains sections for "Research", "Recent Publications", and "From Elsewhere on the Web".

**Research**

I'm Lead Researcher at [Talis](#), working in the field of [Linked Data](#) and the [Semantic Web](#). Before joining [Talis](#) I studied for my PhD at the [Knowledge Media Institute of The Open University](#) in the UK.

My [PhD research](#) was focused on using the Semantic Web to support recommendation-seeking in social networks. Specifically the aim was to: understand how people choose word of mouth information sources from among members of their social networks, and the factors that affect these source-selection decisions; model these processes as computational algorithms so they can be replicated in an online environment; build Semantic Web-based systems that support people in information seeking through their trusted social networks. See [Research](#) and [Publications](#) for a few more details.

**Recent Publications** ([see all Publications](#))

- Christian Bizer, **Tom Heath**, Michael Hausenblas, Tim Berners-Lee (eds.) (2010) [Proceedings of the 3rd International Workshop on Linked Data on the Web \(LDOW2010\)](#) at WWW2010, Raleigh, North Carolina, USA, April 27, 2010.
- Abraham Bernstein, David R. Karger, **Tom Heath** Lee Feigenbaum, Diana Maynard, Enrico Motta, and Krishnaprasad Thirunarayan (eds.) (2009) [The Semantic Web - ISWC2009](#), Proceedings of the 8th International Semantic Web Conference (ISWC 2009), Chantilly, VA, USA, October 25 - 29, 2009. Lecture Notes in Computer Science, Vol. 5823, Springer.
- Christian Bizer, **Tom Heath** and Tim Berners-Lee (2009) [Linked Data - The Story So Far](#). International Journal on Semantic Web and Information Systems, Vol. 5(3), Pages 1-22. DOI: 10.4018/jswis.2009081901
- Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, **Tom Heath**, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, Elena Simperl, (eds.) (2009) [The Semantic Web: Research and Applications](#). Proceedings of the 6th European Semantic Web Conference (ESWC2009), Heraklion, Crete, Greece, May 31 - June 4, 2009. Lecture Notes in Computer Science, Vol. 5554, Springer.
- Christian Bizer, **Tom Heath**, Tim Berners-Lee (eds.) (2009) [Proceedings of the 2nd International Workshop on Linked Data on the Web \(LDOW2009\)](#) at WWW2009, Madrid, Spain, April 20, 2009.
- **Tom Heath** (2008) [Information-seeking on the Web with Trusted Social Networks - from Theory to Systems](#). Ph.D. Thesis, The Open University, Milton Keynes, UK.
- **Tom Heath** and Enrico Motta (2008) [Revyu: Linking reviews and ratings into the Web of Data](#). Journal of Web Semantics, Vol. 6 (4).
- **Tom Heath** (2008) [How Will We Interact with the Web of Data?](#) IEEE Internet Computing, Vol. 12(5), pp. 88-91.

**From Elsewhere on the Web**

**Feed Icon My Latest Semantic Web Bookmarks From Delicious**

- sgvizler - Javascript SPARQL result set visualizer - Google Project Hosting
- Indices of Deprivation explorer
- Things Near To
- swget
- schema.org - Home
- Linked Data Publishing Handbook - Data.gov.uk
- r3 » Blog Archive » SPARQL 1.1 Early Access Features
- Linked Data: Evolving the Web into a Global Data Space
- Big Lynx Productions Ltd
- Linked Data Book

**★ My Latest Reviews on Revyu.com**

- Review of No 1 Airport Lounge, Terminal 3, Heathrow Airport, London
- Review of Sophie the Giraffe Teething Toy
- Review of The Swan Inn, Swineford, nr Bristol
- Review of Extract (Film, 2009)
- Review of Riverford Farm Cook Book: Tales from the Fields, Recipes from the Kitchen
- Review of Airport Reviews - Budget Traveller Guide to Sleeping in Airports
- Review of Domainr
- Review of Tropic Thunder
- Review of The Guardian (Newspaper)
- Review of The Guardian (Film, 2006)

**Revyu.com - Review Anything**

REYU.COM  
REVIEW ANYTHING

**See Also**

- LDL META
- FOAF
- DELICIOUS
- WWW2009

**Recent Tweets**

- @amit\_p yeah, fine line between #outrageous and #justadamngoodidea :) #iswc2011
- RT @amit\_p: Probably more interesting that stuff so far tho not too #Outrageous: Augmenting Urban Environments <http://t.co/gteEogyY> #is ...
- @laroyo just to reiterate, i think it's great that you introduced this track at #iswc2011. gotta keep experimenting! #outrageous

**Visitors to this page come from:**

9,522 Visitors  
10 Jan 2010 - 14 Sep 2011

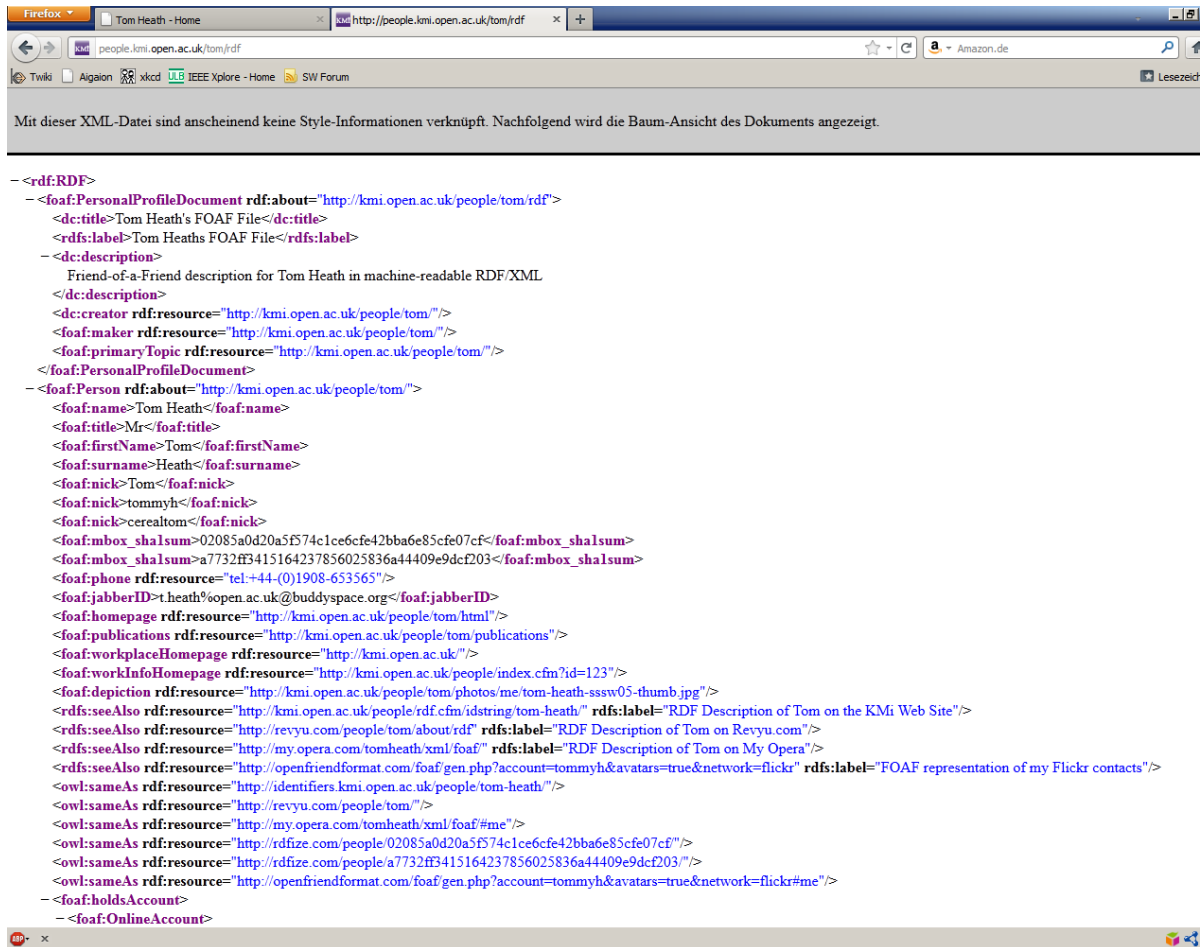
ClustrMaps

**Stuff I Like**

- Chipp (Online Shopping from Amazon)
- Dreamhost (Web Hosting)
- Dropbox (Online File)



# RDF in der freien Wildbahn: FOAF-Profile



```
- <rdf:RDF>
- <foaf:PersonalProfileDocument rdf:about="http://kmi.open.ac.uk/people/tom/rdf">
 <dc:title>Tom Heath's FOAF File</dc:title>
 <rdfs:label>Tom Heaths FOAF File</rdfs:label>
 <dc:description>
 Friend-of-a-Friend description for Tom Heath in machine-readable RDF/XML
 </dc:description>
 <dc:creator rdf:resource="http://kmi.open.ac.uk/people/tom/">
 <foaf:maker rdf:resource="http://kmi.open.ac.uk/people/tom/">
 <foaf:primaryTopic rdf:resource="http://kmi.open.ac.uk/people/tom/">
 <foaf:PersonalProfileDocument>
- <foaf:Person rdf:about="http://kmi.open.ac.uk/people/tom/">
 <foaf:name>Tom Heath</foaf:name>
 <foaf:title>Mr</foaf:title>
 <foaf:firstName>Tom</foaf:firstName>
 <foaf:surname>Heath</foaf:surname>
 <foaf:nick>Tom</foaf:nick>
 <foaf:nick>tommyh</foaf:nick>
 <foaf:nick>cerealtom</foaf:nick>
 <foaf:mbox_sha1sum>02085a0d20a5f574c1ce6cfe42bba6e85cfe07cf</foaf:mbox_sha1sum>
 <foaf:mbox_sha1sum>a7732ff3415164237856025836a44409e9dcf203</foaf:mbox_sha1sum>
 <foaf:phone rdf:resource="tel:+44-(0)1908-653565">
 <foaf:jabberID>t.heath%open.ac.uk@buddyspace.org</foaf:jabberID>
 <foaf:homepage rdf:resource="http://kmi.open.ac.uk/people/tom/html">
 <foaf:publications rdf:resource="http://kmi.open.ac.uk/people/tom/publications">
 <foaf:workplaceHomepage rdf:resource="http://kmi.open.ac.uk/">
 <foaf:workInfoHomepage rdf:resource="http://kmi.open.ac.uk/people/index.cfm?id=123">
 <foaf:depiction rdf:resource="http://kmi.open.ac.uk/people/tom/photos/me/tom-heath-sssw05-thumb.jpg">
 <rdfs:seeAlso rdf:resource="http://kmi.open.ac.uk/people/rdf.cfm/idstring/tom-heath" rdfs:label="RDF Description of Tom on the KMi Web Site">
 <rdfs:seeAlso rdf:resource="http://revyu.com/people/tom/about/rdf" rdfs:label="RDF Description of Tom on Revyu.com">
 <rdfs:seeAlso rdf:resource="http://my.opera.com/tomheath/xml/foaf/" rdfs:label="RDF Description of Tom on My Opera">
 <rdfs:seeAlso rdf:resource="http://openfriendformat.com/foaf/gen.php?account=tommyh&avatars=true&network=flickr" rdfs:label="FOAF representation of my Flickr contacts">
 <owl:sameAs rdf:resource="http://identifiers.kmi.open.ac.uk/people/tom-heath/">
 <owl:sameAs rdf:resource="http://revyu.com/people/tom/">
 <owl:sameAs rdf:resource="http://my.opera.com/tomheath/xml/foaf/#me">
 <owl:sameAs rdf:resource="http://rdfize.com/people/02085a0d20a5f574c1ce6cfe42bba6e85cfe07cf/">
 <owl:sameAs rdf:resource="http://rdfize.com/people/a7732ff3415164237856025836a44409e9dcf203/">
 <owl:sameAs rdf:resource="http://openfriendformat.com/foaf/gen.php?account=tommyh&avatars=true&network=flickr#me">
- <foaf:holdsAccount>
- <foaf:OnlineAccount>
```

# RDFa in der freien Wildbahn: Beispiel O'Reilly



Firefox

Examples-in-the-wild - RDFaWiki | Programming the Semantic Web - O'Reil... | Add-ons-Manager

http://oreilly.com/catalog/9780596153823/

Login - Mitarbeiter-Wiki | xkcd

Home Shop Answers Radar: News & Commentary Safari Books Online Conferences Training School of Technology

O'REILLY Books & Videos Learn something new today.

Buy 2, Get the 3<sup>rd</sup> FREE Use Discount Code opc10 All orders over \$29.<sup>95</sup> qualify for FREE SHIPPING within the US

Browse Contents

**Programming the Semantic Web**  
Build Flexible Applications with Graph Data  
By Toby Segaran, Colin Evans, Jamie Taylor  
Publisher: O'Reilly Media  
Released: July 2009  
Pages: 304  
Press Release

Buy 2 Get 3rd FREE | Free Shipping | 100% Guarantee

Buying Options

Save a Tree - Go Digital What is this?

Ebook \$31.99 Add to Cart  
Formats: DAISY, ePub, Mobi, PDF

Print+Ebook \$43.99 Add to Cart

Print \$39.99 Add to Cart

Safari Books Online Read Now >

Suchen: annotation

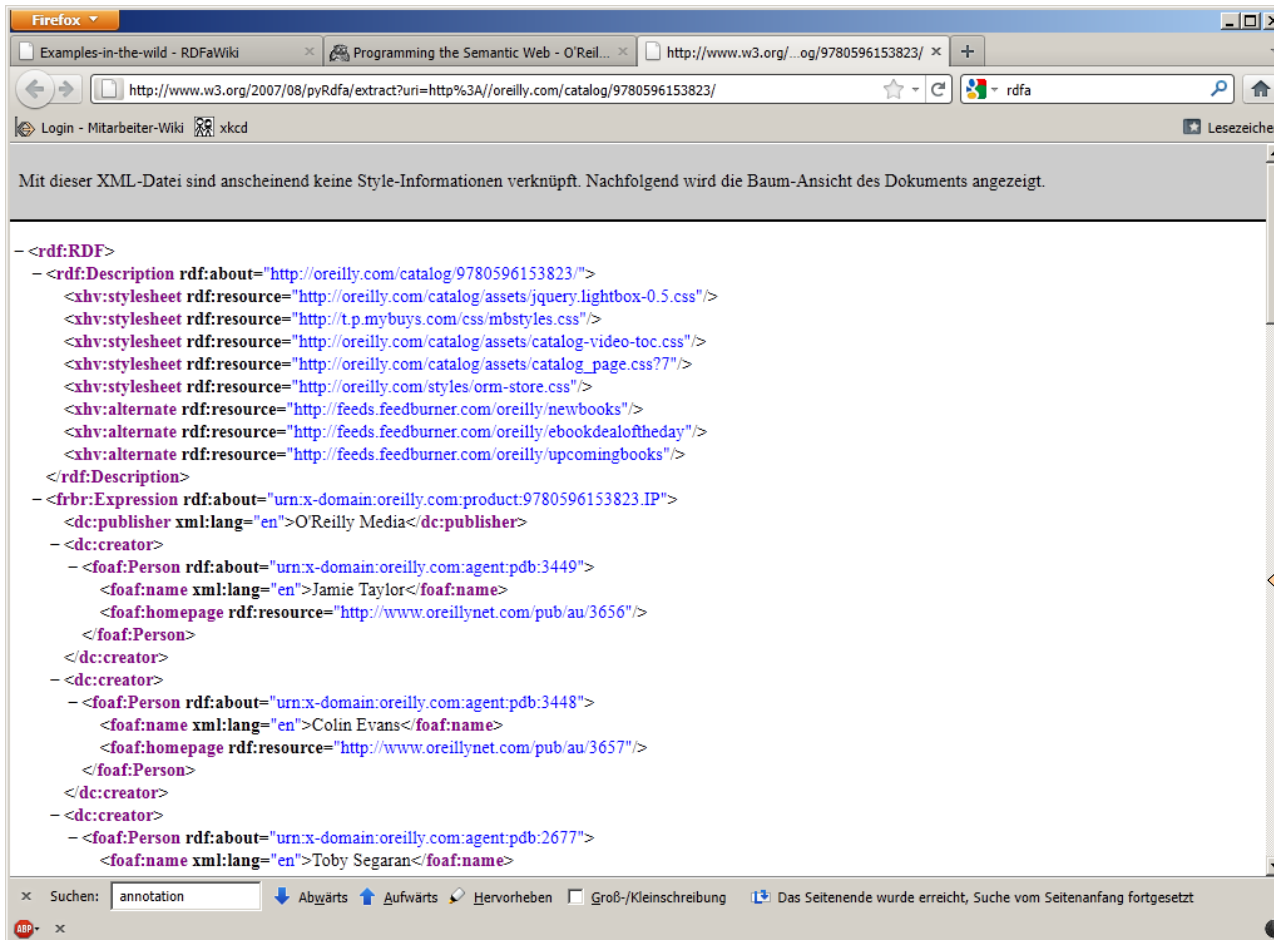
Abwärts Aufwärts Hervorheben Groß-/Kleinschreibung Das Seitenende wurde erreicht, Suche vom Seitenanfang fortgesetzt

RDFa

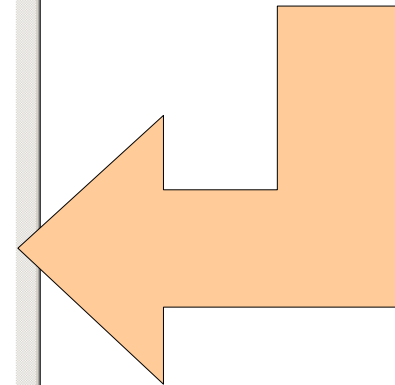
## Semantic Radar Plugin



# RDFa in der freien Wildbahn: Beispiel O'Reilly



```
- <rdf:RDF>
- <rdf:Description rdf:about="http://oreilly.com/catalog/9780596153823/">
 <xhv:stylesheet rdf:resource="http://oreilly.com/catalog/assets/jquery.lightbox-0.5.css"/>
 <xhv:stylesheet rdf:resource="http://t.p.mybuys.com/css/mbstyles.css"/>
 <xhv:stylesheet rdf:resource="http://oreilly.com/catalog/assets/catalog-video-toc.css"/>
 <xhv:stylesheet rdf:resource="http://oreilly.com/catalog/assets/catalog_page.css?7"/>
 <xhv:stylesheet rdf:resource="http://oreilly.com/styles/orm-store.css"/>
 <xhv:alternate rdf:resource="http://feeds.feedburner.com/oreilly/newbooks"/>
 <xhv:alternate rdf:resource="http://feeds.feedburner.com/oreilly/ebookdealoftheday"/>
 <xhv:alternate rdf:resource="http://feeds.feedburner.com/oreilly/upcomingbooks"/>
</rdf:Description>
- <frbr:Expression rdf:about="urn:x-domain:oreilly.com:product:9780596153823.IP">
 <dc:publisher xml:lang="en">O'Reilly Media</dc:publisher>
 <dc:creator>
 - <foaf:Person rdf:about="urn:x-domain:oreilly.com:agent:pdb:3449">
 <foaf:name xml:lang="en">Jamie Taylor</foaf:name>
 <foaf:homepage rdf:resource="http://www.oreillynet.com/pub/au/3656"/>
 </foaf:Person>
 </dc:creator>
 - <dc:creator>
 - <foaf:Person rdf:about="urn:x-domain:oreilly.com:agent:pdb:3448">
 <foaf:name xml:lang="en">Colin Evans</foaf:name>
 <foaf:homepage rdf:resource="http://www.oreillynet.com/pub/au/3657"/>
 </foaf:Person>
 </dc:creator>
 - <dc:creator>
 - <foaf:Person rdf:about="urn:x-domain:oreilly.com:agent:pdb:2677">
 <foaf:name xml:lang="en">Toby Segaran</foaf:name>
```





# RDFa in der freien Wildbahn: Beispiel O'Reilly

- Normale Suche ohne RDFa:

## [Programming the Semantic Web - O'Reilly Media](#)

[shop.oreilly.com/product/9780596153823.do](http://shop.oreilly.com/product/9780596153823.do) 

With this book, the promise of the **semantic web** -- in which machines can find, share, and combine data on the **Web** -- is not just a technical possibility, but a ...

- Verbessertes Ergebnis mit RDFa:



## [Programming the Semantic Web \[Book\]](#)

by Toby Segaran, Colin Evans, Jamie Taylor - O'Reilly (2009.07.01) - paperback - 280 pages

With this book, the promise of the **Semantic Web** -- in which machines can find, share, and combine data on the **Web** -- is not just a technical ...

★★★★☆ 8 reviews - [Add to Shopping List](#)

# RDFa in der freien Wildbahn: Beispiel BBC Music



The screenshot shows a Firefox browser window with the URL `http://www.bbc.co.uk/music/artists/b7ffd2af-418f-4be2-bdd1-22f8b48613da`. The page is the BBC Music artist profile for Nine Inch Nails. The browser's search bar contains the text "annotation".

**BBC Music** Sign in News Sport Weather iPlayer TV Radio More Search the BBC

MUSIC BETA GENRES REVIEWS NEWS BLOG NEW: SHOWCASE QUICK FIND Enter an artist name ...

BBC Music > Artists > Nine Inch Nails

## Nine Inch Nails

Formed 1988.

PLAYED MOST ON **BBC RADIO 6music**

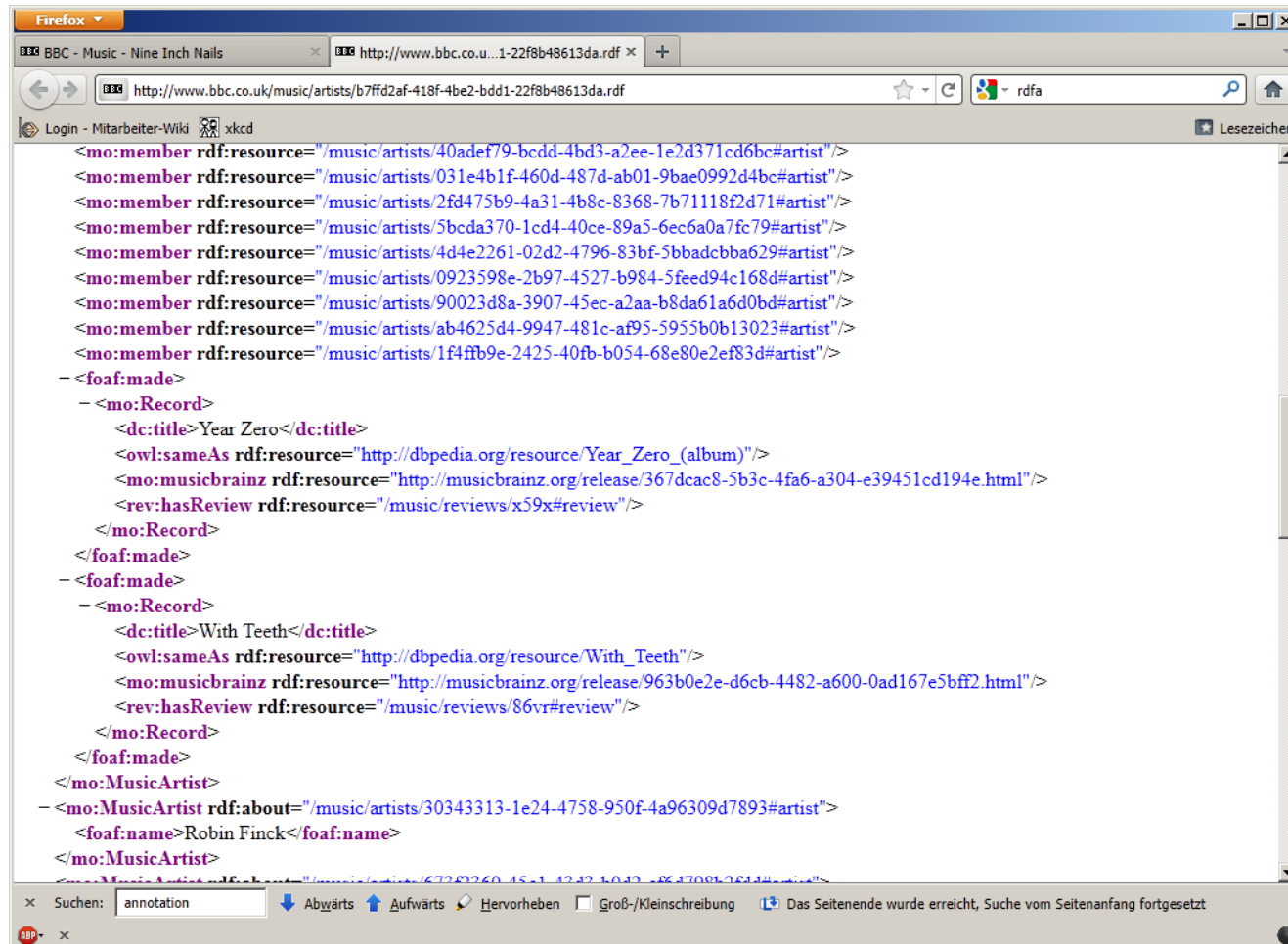
**BBC Music Showcase**  
Watch and listen to exclusive music clips

**Latest Tracks Played On The BBC**

- Head Like A Hole**  
BBC 6 Music | Late Night Jukebox 22/06/2011
- Right Where It Belongs**  
BBC 6 Music | Jarvis Cocker's Sunday Service Adam Curtis makes a welcome return
- The Hand That Feeds**  
BBC 6 Music | Shaun Keaveny Shrapnel

Suchen: annotation Abwärts Aufwärts Hervorheben Groß-/Kleinschreibung Das Seitenende wurde erreicht, Suche vom Seitenanfang fortgesetzt

# RDFa in der freien Wildbahn: Beispiel BBC Music



```
<mo:member rdf:resource="/music/artists/40adef79-bcdd-4bd3-a2ee-1e2d371cd6bc#artist"/>
<mo:member rdf:resource="/music/artists/031e4b1f-460d-487d-ab01-9bae0992d4bc#artist"/>
<mo:member rdf:resource="/music/artists/2fd475b9-4a31-4b8c-8368-7b71118f2d71#artist"/>
<mo:member rdf:resource="/music/artists/5bcda370-1cd4-40ce-89a5-6ec6a0a7fc79#artist"/>
<mo:member rdf:resource="/music/artists/4d4e2261-02d2-4796-83bf-5bbadcba629#artist"/>
<mo:member rdf:resource="/music/artists/0923598e-2b97-4527-b984-5feed94c168d#artist"/>
<mo:member rdf:resource="/music/artists/90023d8a-3907-45ec-a2aa-b8da61a6d0bd#artist"/>
<mo:member rdf:resource="/music/artists/ab4625d4-9947-481c-af95-5955b0b13023#artist"/>
<mo:member rdf:resource="/music/artists/1f4ffb9e-2425-40fb-b054-68e80e2ef83d#artist"/>
- <foaf:made>
- <mo:Record>
 <dc:title>Year Zero</dc:title>
 <owl:sameAs rdf:resource="http://dbpedia.org/resource/Year_Zero_(album)"/>
 <mo:musicbrainz rdf:resource="http://musicbrainz.org/release/367dcac8-5b3c-4fa6-a304-e39451cd194e.html"/>
 <rev:hasReview rdf:resource="/music/reviews/x59x#review"/>
 </mo:Record>
</foaf:made>
- <foaf:made>
- <mo:Record>
 <dc:title>With Teeth</dc:title>
 <owl:sameAs rdf:resource="http://dbpedia.org/resource/With_Teeth"/>
 <mo:musicbrainz rdf:resource="http://musicbrainz.org/release/963b0e2e-d6cb-4482-a600-0ad167e5bff2.html"/>
 <rev:hasReview rdf:resource="/music/reviews/86vr#review"/>
 </mo:Record>
</foaf:made>
</mo:MusicArtist>
- <mo:MusicArtist rdf:about="/music/artists/30343313-1e24-4758-950f-4a96309d7893#artist">
 <foaf:name>Robin Finck</foaf:name>
</mo:MusicArtist>
```

# Musikempfehlungen auf Basis von freilaufendem RDFa

The screenshot shows a Firefox browser window with the URL `data.seevl.net/entity/4kqQMGZV`. The page is titled "seevl" and displays information for "Nine Inch Nails".

**seevl** seevl.net / dev zone Semantic search

### Nine Inch Nails

Facebook Gefällt mir | Tweet 0 | Share

Nine Inch Nails is an American industrial rock project, founded in 1988 by Trent Reznor in Cleveland, Ohio. As its main producer, singer, songwriter, and instrumentalist, Reznor is the only official member ... [read more](#)

#### Videos

powered by YouTube

#### Fact sheet

Activity	1988-present 1989-present
Origin(s)	Cleveland Ohio United States
Genre(s)	Alternative metal Alternative rock Electronic music Industrial metal Industrial music Industrial rock
Label(s)	Interscope Records Island Records

### Related bands and artists

#### Tapeworm

Tapeworm was a defunct side project of Nine Inch Nails which existed in various forms from 1995 to roughly 2004. Tapeworm never released any recordings, but was frequently referenced in interviews. The band started as a side-project between Nine Inch Nails frontman Trent Reznor and live-band members Danny Lohner and ... [more info](#)

#### Marilyn Manson

Both have the same origin(s):  
• United States

Both are on the same label(s):  
• Interscope Records  
• Nothing Records

Both have the same member(s):  
• Chris Vrenna  
• Jeordie White

Marilyn Manson is an American rock band from Fort Lauderdale, Florida. Formed in 1989 by Brian Warner and Scott Putesky, the group was originally named Marilyn Manson & the Spooky Kids with their uniquely theatrical performances garnering a local cult following in the early '90s. This attention subsequently developed into a ... [more info](#)

#### Filter

Filter is a rock group formed in 1993 in Cleveland by Richard Patrick and guitarist/programmer Brian Liesegang. Filter has released five studio albums, the most recent being *The Trouble with Angels*, which was released

# RDFa in der freien Wildbahn: Beispiel Rotten Tomatoes



The screenshot shows the Rotten Tomatoes website interface for the movie "Requiem for a Dream (2000)". The page features a green header with the Rotten Tomatoes logo and navigation links. The main content area includes a movie poster, a Tomatometer score of 78% (Fresh), and an Audience score of 93% (Liked it). The page also displays a "MY RATING" section, a "MOVIE INFO" section with details about the film's runtime, genre, and release date, and a "MOVIES LIKE REQUIEM FOR A DREAM" section with recommendations like "Trainspotting" and "Fear and Loathing in Las Vegas".

**Requiem for a Dream (2000)**

**TOMATOMETER** 78% All Critics | Top Critics  
**AUDIENCE** 93% liked it

Average Rating: 7.3/10  
Reviews Counted: 131  
Fresh: 102 | Rotten: 29

Though the movie may be too intense for some to stomach, the wonderful performances and the bleak imagery are hard to forget.

Average Rating: 4.1/5  
User Ratings: 261,204

**MOVIE INFO**

Based on the novel by Hubert Selby Jr., this gritty drama concerns four people trapped by their addictions. Harry (Jared Leto), and his best friend Tyrone (Marlon Wayans) are impoverished heroin addicts living in Coney Island, NY, while Harry's girlfriend Marion (Jennifer Connelly) is a fellow addict trying to distance herself from her wealthy father. Harry dreams of scoring a pound of smack, from which he could make enough money to open a clothing boutique with Marion, but so far he and his ... [More](#)

R, 1 hr. 42 min. In Theaters: Oct 6, 2000 Wide  
Drama On DVD: May 22, 2001  
Directed By: Darren Aronofsky US Box Office: \$2.5M  
Written By: Darren Aronofsky, Hubert Selby, Jr. Artisan Entertainment

**Requiem for a Dream Trailer & Photos**

More Trailers and Videos (1)

**MOVIES LIKE REQUIEM FOR A DREAM**

- Trainspotting
- Fear and Loathing in Las Vegas
- The Basketball Diaries

**Latest News for Requiem for a Dream**

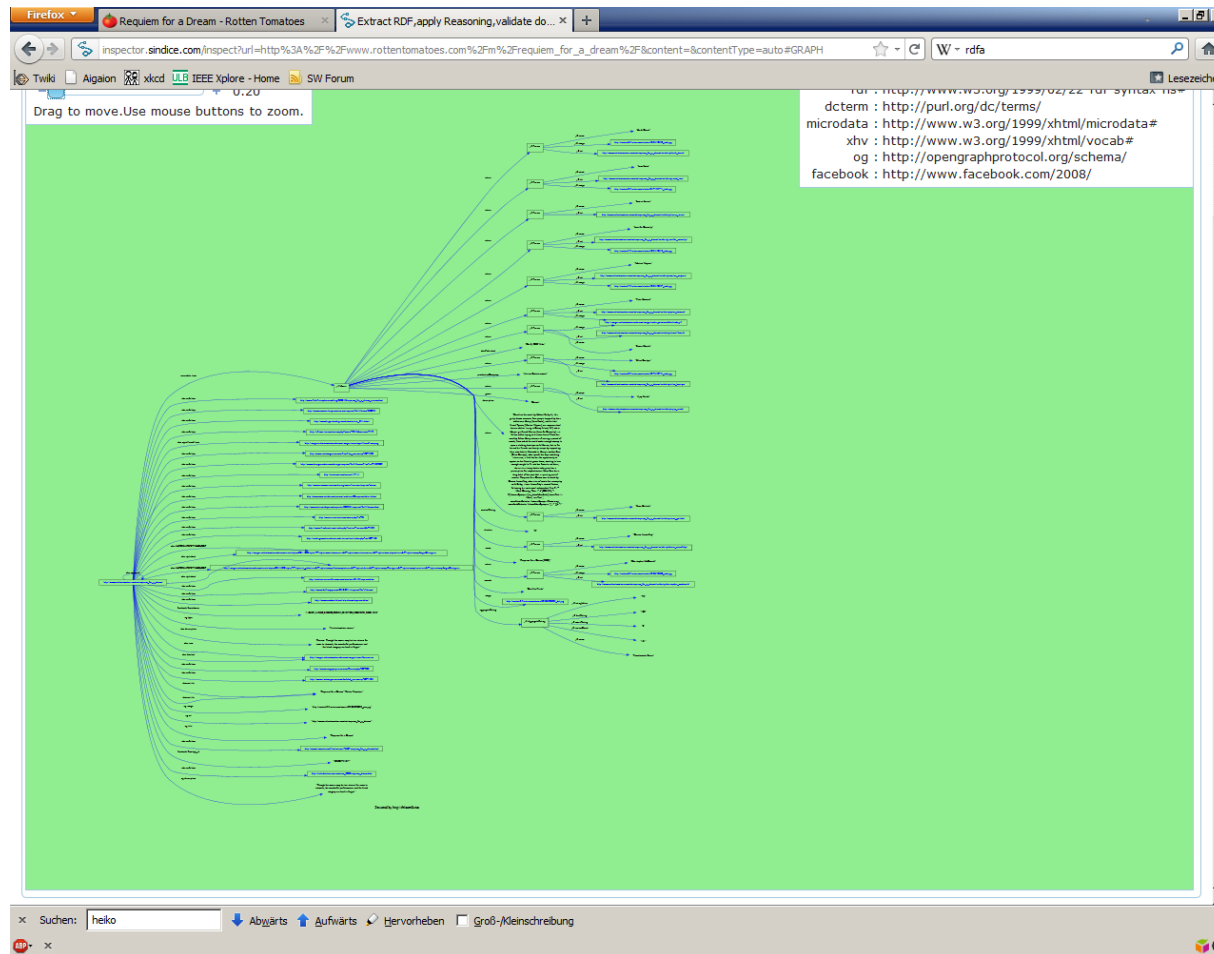
**August 25, 2009:**  
EW Lists 16 Classic "Dare You to Look" Scenes  
All the Nazi-scalpin' action in "Inglourious Basterds" has got the EW staff to thinking about some...

**January 9, 2008:**  
Lorenzo di Bonaventura Talks G.I. Joe, Transformers 2  
You think G.I. Joe and Transformers producer Lorenzo di Bonaventura doesn't hear your grouching, film...

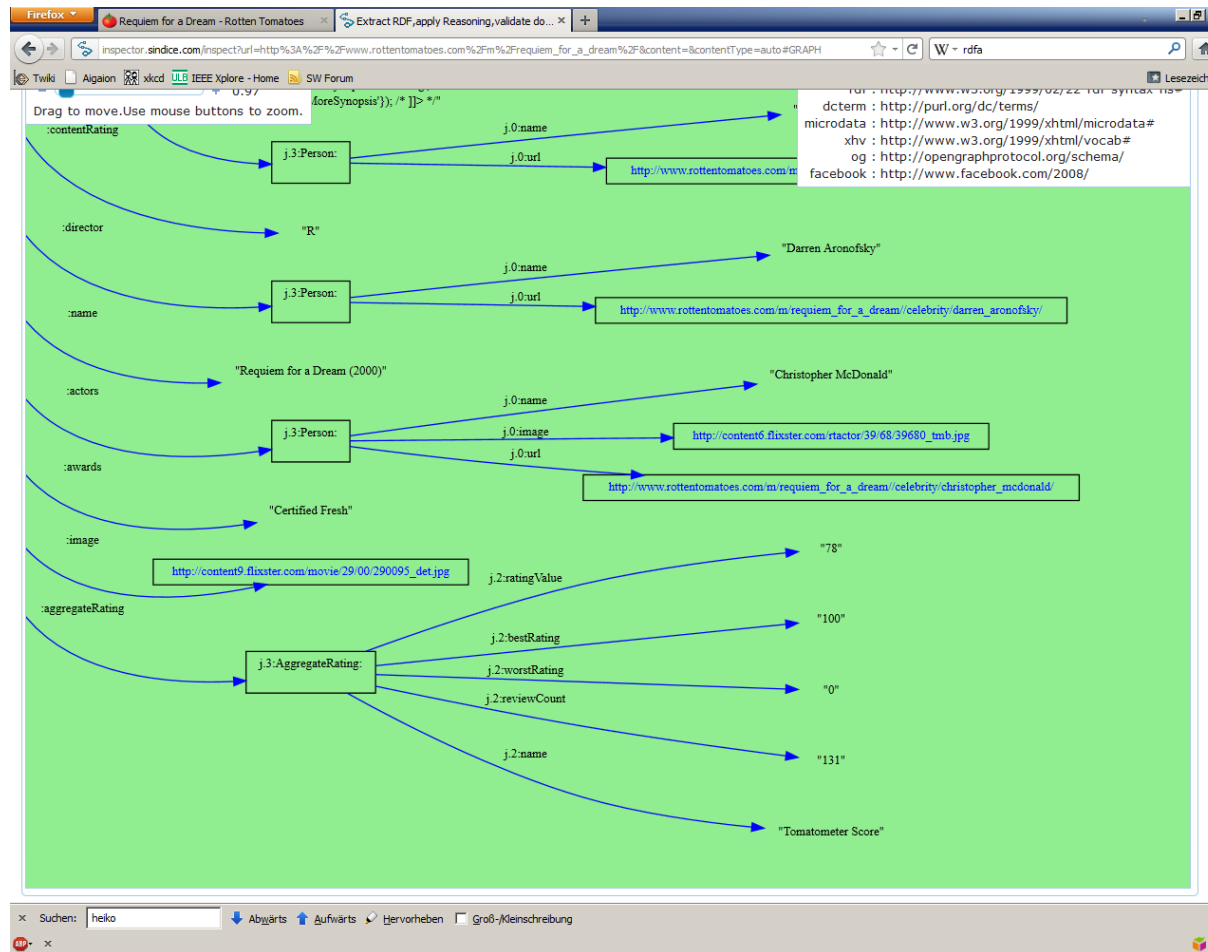
# RDFa in der freien Wildbahn: Beispiel Rotten Tomatoes



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



# RDFa in der freien Wildbahn: Beispiel Rotten Tomatoes



# Zusammenfassung RDF II

- Verschiedenste Tools für RDF existieren
  - Speicherung
  - Visualisierung
  - und vieles mehr
- RDFa ermöglicht die Verbindung von HTML und RDF
  - ohne Redundanz
- Einige Beispiele für RDF in der freien Wildbahn



# Zusammenfassung

- RDF ist ein Standard zur Beschreibung beliebiger Informationen
  - Graphenbasiert
  - Serialisierungen in Triples und XML
- Semantische Prinzipien
  - Non-unique Naming Assumption
  - Open World Assumption
- Tools zur Speicherung

# Ein kritischer Blick zurück

- Kann RDF mehr als XML?
- XML ist eine Auszeichnungssprache für Informationen
- In XML kann man beliebige Tags und Attribute definieren
- XML-Tagnamen haben für den Computer keine Bedeutung
- RDF ist eine Auszeichnungssprache für Informationen
- In RDF kann man beliebige Klassen / Relationen definieren
- RDF-Bezeichner haben für den Computer keine Bedeutung

# Ein kritischer Blick zurück

- Warum haben wir das dann eigentlich gelernt?



# Ein kritischer Blick zurück

HOW STANDARDS PROLIFERATE:  
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)



<http://xkcd.com/927/>

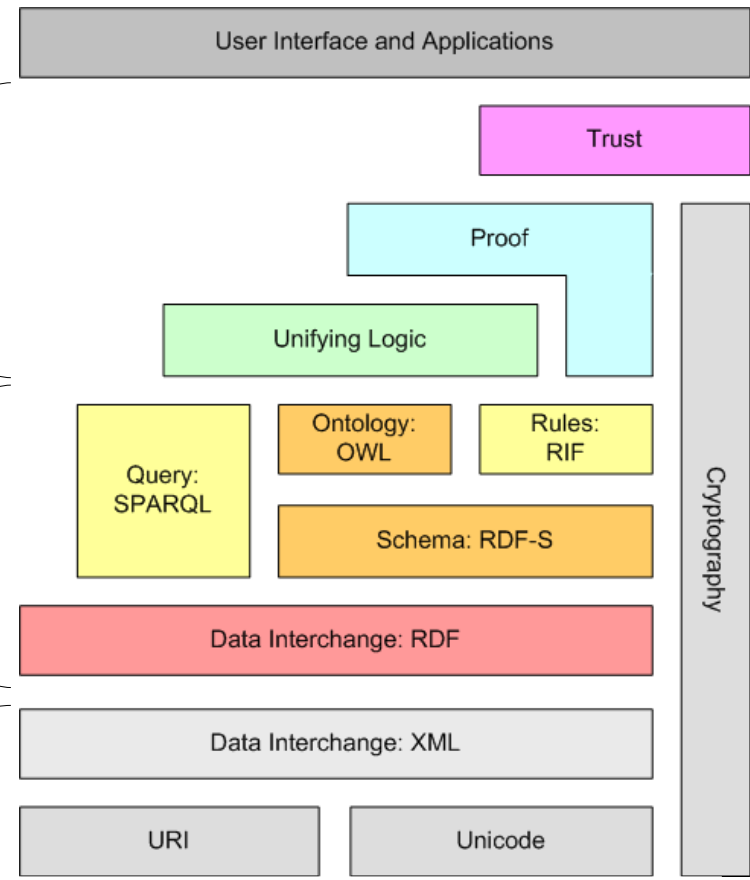
# Semantic Web – Aufbau



here be dragons...

Semantic-Web-  
Technologie  
(Fokus der Vorlesung)

Technische  
Grundlagen



Berners-Lee (2009): *Semantic Web and Linked Data*  
<http://www.w3.org/2009/Talks/0120-campus-party-tbl/>

# Vorlesung Semantic Web



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Vorlesung im Wintersemester 2012/2013

Dr. Heiko Paulheim

Fachgebiet Knowledge Engineering