Unit 1 – a “Bird’s Eye” View on RDF(S), OWL & SPARQL

Axel Polleres

Siemens AG Österreich

VU 184.729 Semantic Web Technologies

Unit Outline

1. Motivation – Aggregating Web Data

2. How can I publish data? RDF

3. How can I query that data? SPARQL

4. What does that data mean? Ontologies described in RDFS + OWL

5. What's next?

Prerequisites

- Some basic knowledge about first-order logics.
- Some basic knowledge about databases (mainly: SQL).
- Some basic knowledge about HTML and HTTP.
- Some basic knowledge about XML would be nice.

Unit Outline

1. Motivation – Aggregating Web Data

2. How can I publish data? RDF

3. How can I query that data? SPARQL

4. What does that data mean? Ontologies described in RDFS + OWL

5. What's next?

Finding reviewers for a scientific Journal

Tim Berners-Lee, Dan Connolly, Lalana Kagal, Yosi Scharf, Jim Hendler: **N3Logic: A logical framework for the World Wide Web**. Theory and Practice of Logic Programming (TPLP), Volume 8, p249-269.

Assume you are the editor of a scientific journal:

- Who are the right reviewers?

Finding reviewers for a scientific Journal

Tim Berners-Lee, Dan Connolly, Lalana Kagal, Yosi Scharf, Jim Hendler: **N3Logic: A logical framework for the World Wide Web**. Theory and Practice of Logic Programming (TPLP), Volume 8, p249-269.

Assume you are the editor of a scientific journal:

- Who are the right reviewers?
- Which qualified people do I know?

Finding reviewers for a scientific Journal

Tim Berners-Lee, Dan Connolly, Lalana Kagal, Yosi Scharf, Jim Hendler: **N3Logic: A logical framework for the World Wide Web**. Theory and Practice of Logic Programming (TPLP), Volume 8, p249-269.

Assume you are the editor of a scientific journal:

- Who are the right reviewers?
- Which qualified people do I know?
- How can I assess their expertise?

Finding reviewers for a scientific Journal

Tim Berners-Lee, Dan Connolly, Lalana Kagal, Yosi Scharf, Jim Hendler: **N3Logic: A logical framework for the World Wide Web**. Theory and Practice of Logic Programming (TPLP), Volume 8, p249-269.

Assume you are the editor of a scientific journal:

- Who are the right reviewers?
- Which qualified people do I know?
- How can I assess their expertise?
- Which reviewers are in conflict?

Finding reviewers for a scientific Journal

Tim Berners-Lee, Dan Connolly, Lalana Kagal, Yosi Scharf, Jim Hendler: **N3Logic: A logical framework for the World Wide Web**. Theory and Practice of Logic Programming (TPLP), Volume 8, p249-269.

Assume you are the editor of a scientific journal:

- Who are the right reviewers?
- Which qualified people do I know?
- How can I assess their expertise?
- Which reviewers are in conflict?
- Observation: Much of the necessary data is available on the Web!

Finding reviewers for a scientific Journal

Tim Berners-Lee, Dan Connolly, Lalana Kagal, Yosi Scharf, Jim Hendler: **N3Logic: A logical framework for the World Wide Web**. Theory and Practice of Logic Programming (TPLP), Volume 8, p249-269.

Assume you are the editor of a scientific journal:

- Who are the right reviewers?
- Which qualified people do I know?
- How can I assess their expertise?
- Which reviewers are in conflict?
- Observation: Much of the necessary data is available on the Web!

Finding reviewers for a scientific Journal

Tim Berners-Lee, Dan Connolly, Lalana Kagal, Yosi Scharf, Jim Hendler: **N3Logic: A logical framework for the World Wide Web**. Theory and Practice of Logic Programming (TPLP), Volume 8, p249-269.

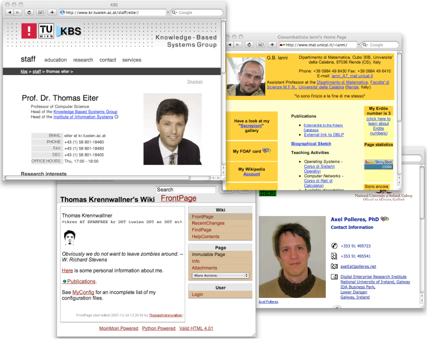
Assume you are the editor of a scientific journal:

- Who are the right reviewers?
- Which qualified people do I know?
- How can I assess their expertise?
- Which reviewers are in conflict?
- Observation: Much of the necessary data is available on the Web!

Questions:

- Where do I get the right data?
- What is the format & structure (schema) of this data?
- Which rules and query languages do I use to aggregate this data?
- Which systems are out there to support me?

Where is the data? 1/4



KBS

http://www.kr.tuwien.ac.at/staff/eiter/

TU WIEN KBS Knowledge-Based Systems Group

staff education research contact services

kbs > staff > thomas eiter >

Drucken

Prof. Dr. Thomas Eiter

Professor of Computer Science
Head of the Knowledge Based Systems Group
Head of the Institute of Information Systems

EMAIL:	eiter at kr.tuwien.ac.at
PHONE:	+43 (1) 58 801-18460
FAX:	+43 (1) 58 801-18493
SEC:	+43 (1) 58 801-18405
OFFICE HOURS:	Thu, 17:00 - 18:00

Research interests

Giovambattista Ianni's Home Page

http://www.mat.unical.it/~ianni/

G.B. Ianni

Dipartimento di Matematica, Cubo 30B, Universita' della Calabria, 87036 Rende (CS), Italy

Phone: +39 0984 49 6430 Fax: +39 0984 49 6410
E-mail: ianni_AT_mat.unical.it

Assistant Professor at the Dipartimento di Matematica, Facolta' di Scienze M.F.N., Universita' della Calabria (Rende, Italy).

"Io sono l'inizio e la fine di me stesso"

Have a look at my "Secrezioni" gallery

My FOAF card

My Wikipedia Account

Publications

- External link to the Polaris Database
- External link to DBLP

Biographical Sketch

Teaching Activities

- *Operating Systems* - Corso di Sistemi Operativi
- *Computer Networks* - Corso di Reti di Calcolatori

My Erdös number is 3 (click here to learn about Erdös numbers)

Page statistics

ShinyStat 23304

Sono ancora

Search

Thomas Krennwallner's Wiki FrontPage

Thomas Krennwallner
`<tkren AT SPAMFREE kr DOT tuwien DOT ac DOT at>`

Obviously we do not want to leave zombies around. -- W. Richard Stevens

Here is some personal information about me.

Publications.

See MyConfig for an incomplete list of my configuration files.

FrontPage (last edited 2007-12-24 13:20:54 by ThomasKrennwallner)

MoinMoin Powered Python Powered Valid HTML 4.01

DERI

National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh

Axel Polleres, PhD

Contact Information

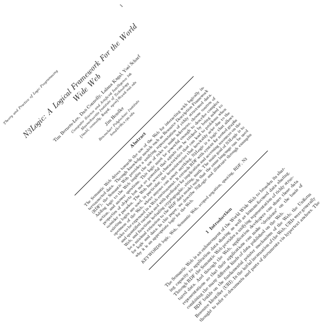
+353 91 495723

+353 91 495541

axel[at]polleres.net

Digital Enterprise Research Institute
National University of Ireland, Galway
IDA Business Park,
Lower Dangan
Galway, Ireland

Axel Polleres



Theory and Practice of Logic Programming

1

N3Logic: A Logical Framework For the World Wide Web

Tim Berners-Lee, Dan Connolly, Lalana Kagal, Yosi Scharf
Computer Science and Artificial Intelligence lab
Massachusetts Institute of Technology
{timbl, connolly, lkagal, syosi}@csail.mit.edu

Jim Hendler
Rensselaer Polytechnic Institute
hendler@cs.rpi.edu

Abstract

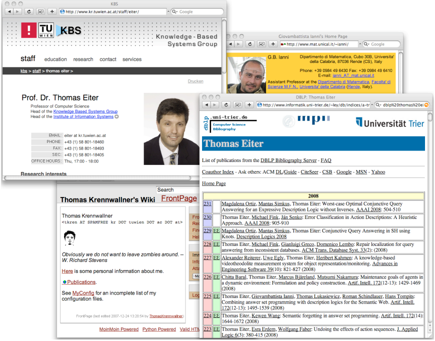
The Semantic Web drives towards the use of the Web for interacting with logically interconnected data. Through knowledge models such as Resource Description Framework (RDF), the Semantic Web provides a unifying representation of richly structured data. Adding logic to the Web implies the use of rules to make inferences, choose courses of action, and answer questions. This logic must be powerful enough to describe complex properties of objects but not so powerful that agents can be tricked by being asked to consider a paradox. The Web has several characteristics that can lead to problems when existing logics are used, in particular, the inconsistencies that inevitably arise due to the openness of the Web, where anyone can assert anything. N3Logic is a logic that allows rules to be expressed in a Web environment. It extends RDF with syntax for nested graphs and quantified variables and with predicates for implication and accessing resources on the Web, and functions including cryptographic, string, math. The main goal of N3Logic is to be a minimal extension to the RDF data model such that the same language can be used for logic and data. In this paper, we describe N3Logic and illustrate through examples why it is an appropriate logic for the Web.

KEYWORDS: logic, Web, Semantic Web, scoped negation, quoting, RDF, N3

1 Introduction

The Semantic Web is an enhancement of the World Wide Web to broaden its sharing capacity to application data sharing as well as human-focused data sharing. Through RDF the Semantic Web provides a unifying representation of richly structured data. And through the Web, application developers can share these data representations so that their applications can make "decisions" on the basis of combining the many different kinds of data published on the Web. RDF builds on the fundamental pointer mechanism of the Web; the Uniform Resource Identifier (URI). In the initial incarnation of the Web, URIs were generally thought to refer to documents and parts of documents via hypertext anchors. The

Where is the data? 1/4



KBS

http://www.kr.tuwien.ac.at/staff/eiter/

TU WIEN KBS Knowledge - Based Systems Group

staff education research contact services

kbs > staff > thomas eiter >

Drucken

Prof. Dr. Thomas Eiter

Professor of Computer Science
Head of the Knowledge Based Systems Group
Head of the Institute of Information Systems

EMAIL:	eiter at kr.tuwien.ac.at
PHONE:	+43 (1) 58 801-18460
FAX:	+43 (1) 58 801-18493
SEC:	+43 (1) 58 801-18405
OFFICE HOURS:	Thu, 17:00 - 18:00

Research interests

Giovambattista Ianni's Home Page

http://www.mat.unical.it/~ianni/

G.B. Ianni

Dipartimento di Matematica, Cubo 30B, Universita' della Calabria, 87036 Rende (CS), Italy

Phone: +39 0984 49 6430 Fax: +39 0984 49 6410
E-mail: ianni_AT_mat.unical.it

Assistant Professor at the Dipartimento di Matematica, Facolta' di Scienze M.F.N., Universita' della Calabria (Rende, Italy).

Search

Thomas Krennwallner's Wiki FrontPage

Thomas Krennwallner
<tkren AT SPAMFREE kr DOT tuwien DOT ac DOT at>

Obviously we do not want to leave zombies around. -- W. Richard Stevens

Here is some personal information about me.

Publications.

See MyConfig for an incomplete list of my configuration files.

FrontPage (last edited 2007-12-24 13:20:54 by ThomasKrennwallner)

MoinMoin Powered Python Powered Valid HTM

DBLP: Thomas Eiter

http://www.informatik.uni-trier.de/~ley/db/indices/a-tr

dblp.uni-trier.de Computer Science Bibliography

mpii Universität Trier

Thomas Eiter

List of publications from the DBLP Bibliography Server - FAQ

Coauthor Index - Ask others: ACM DL/Guide - CiteSeer - CSB - Google - MSN - Yahoo

Home Page

2008		
231		Magdalena Ortiz, Mantas Simkus, Thomas Eiter: Worst-case Optimal Conjunctive Query Answering for an Expressive Description Logic without Inverses. AAAI 2008: 504-510
230		Thomas Eiter, Michael Fink, Ján Senko: Error Classification in Action Descriptions: A Heuristic Approach. AAAI 2008: 905-910
229	EE	Magdalena Ortiz, Mantas Simkus, Thomas Eiter: Conjunctive Query Answering in SH using Knots. Description Logics 2008
228	EE	Thomas Eiter, Michael Fink, Gianluigi Greco, Domenico Lembo: Repair localization for query answering from inconsistent databases. ACM Trans. Database Syst. 33(2): (2008)
227	EE	Alexander Reiterer, Uwe Egly, Thomas Eiter, Heribert Kahmen: A knowledge-based videotheodolite measurement system for object representation/monitoring. Advances in Engineering Software 39(10): 821-827 (2008)
226	EE	Chitta Baral, Thomas Eiter, Marcus Bjäreland, Mutsumi Nakamura: Maintenance goals of agents in a dynamic environment: Formulation and policy construction. Artif. Intell. 172(12-13): 1429-1469 (2008)
225	EE	Thomas Eiter, Giovambattista Ianni, Thomas Lukasiewicz, Roman Schindlauer, Hans Tompits: Combining answer set programming with description logics for the Semantic Web. Artif. Intell. 172(12-13): 1495-1539 (2008)
224	EE	Thomas Eiter, Kewen Wang: Semantic forgetting in answer set programming. Artif. Intell. 172(14): 1644-1672 (2008)
223	EE	Thomas Eiter, Esra Erdem, Wolfgang Faber: Undoing the effects of action sequences. J. Applied Logic 6(3): 380-415 (2008)

1

Theory and Practice of Logic Programming

N3Logic: A Logical Framework For the World Wide Web

Tim Berners-Lee, Dan Connolly, Lalana Kagal, Yosi Scharf
Computer Science and Artificial Intelligence lab
Massachusetts Institute of Technology
{timbl, connolly, lkagal, syosi}@csail.mit.edu

Jim Hendler
Rensselaer Polytechnic Institute
hendler@cs.rpi.edu

Abstract

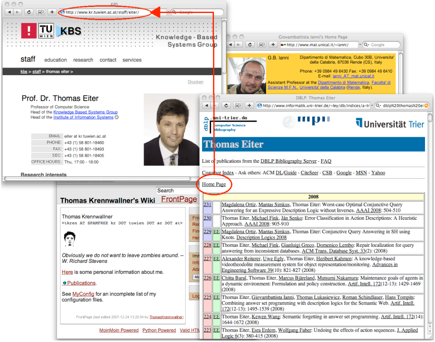
The Semantic Web drives towards the use of the Web for interacting with logically interconnected data. Through knowledge models such as Resource Description Framework (RDF), the Semantic Web provides a unifying representation of richly structured data. Adding logic to the Web implies the use of rules to make inferences, choose courses of action, and answer questions. This logic must be powerful enough to describe complex properties of objects but not so powerful that agents can be tricked by being asked to consider a paradox. The Web has several characteristics that can lead to problems when existing logics are used, in particular, the inconsistencies that inevitably arise due to the openness of the Web, where anyone can assert anything. N3Logic is a logic that allows rules to be expressed in a Web environment. It extends RDF with syntax for nested graphs and quantified variables and with predicates for implication and accessing resources on the Web, and functions including cryptographic, string, math. The main goal of N3Logic is to be a minimal extension to the RDF data model such that the same language can be used for logic and data. In this paper, we describe N3Logic and illustrate through examples why it is an appropriate logic for the Web.

KEYWORDS: logic, Web, Semantic Web, scoped negation, quoting, RDF, N3

1 Introduction

The Semantic Web is an enhancement of the World Wide Web to broaden its sharing capacity to application data sharing as well as human-focused data sharing. Through RDF the Semantic Web provides a unifying representation of richly structured data. And through the Web, application developers can share these data representations so that their applications can make “decisions” on the basis of combining the many different kinds of data published on the Web. RDF builds on the fundamental pointer mechanism of the Web; the Uniform Resource Identifier (URI). In the initial incarnation of the Web, URIs were generally thought to refer to documents and parts of documents via hypertext anchors. The

Where is the data? 1/4


KBS
http://www.kr.tuwien.ac.at/staff/eiter/
Google
TU WIEN
KBS
Knowledge - Based Systems Group
staff
education
research
contact
services
kbs > staff > thomas eiter >
Drucken

Prof. Dr. Thomas Eiter

Professor of Computer Science
Head of the Knowledge Based Systems Group
Head of the Institute of Information Systems

EMAIL:	eiter at kr.tuwien.ac.at
PHONE:	+43 (1) 58 801-18460
FAX:	+43 (1) 58 801-18493
SEC:	+43 (1) 58 801-18405
OFFICE HOURS:	Thu, 17:00 - 18:00

Research interests

Giovambattista Ianni's Home Page
http://www.mat.unical.it/~ianni/
Google

G.B. Ianni

Dipartimento di Matematica, Cubo 30B, Universita' della Calabria, 87036 Rende (CS), Italy

Phone: +39 0984 49 6430 Fax: +39 0984 49 6410
E-mail: ianni_AT_mat.unical.it

Assistant Professor at the Dipartimento di Matematica, Facolta' di Scienze M.F.N., Universita' della Calabria (Rende, Italy).

Thomas Krennwallner's Wiki FrontPage

Thomas Krennwallner
`<tkren AT SPAMFREE kr DOT tuwien DOT ac DOT at>`

Obviously we do not want to leave zombies around. -- W. Richard Stevens

Here is some personal information about me.

Publications.

See MyConfig for an incomplete list of my configuration files.

FrontPage (last edited 2007-12-24 13:20:54 by ThomasKrennwallner)

MoinMoin Powered Python Powered Valid HTM

DBLP: Thomas Eiter
http://www.informatik.uni-trier.de/~ley/db/indices/a-tr
dblp%20thomas%20ei
dblp.uni-trier.de
Computer Science Bibliography
mpii
Universität Trier

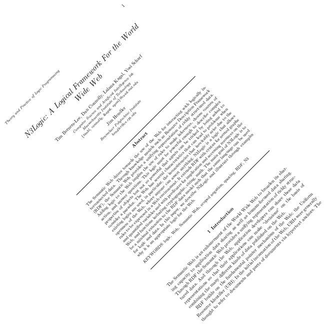
Thomas Eiter

List of publications from the DBLP Bibliography Server - FAQ

Coauthor Index - Ask others: ACM DL/Guide - CiteSeer - CSB - Google - MSN - Yahoo

Home Page

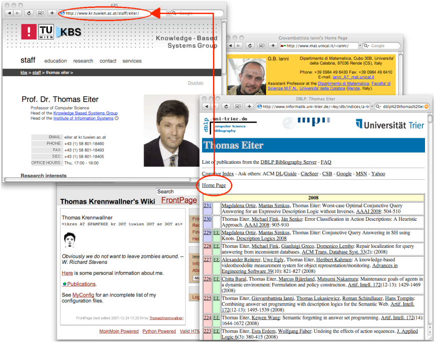
		2008
231		Magdalena Ortiz, Mantas Simkus, Thomas Eiter: Worst-case Optimal Conjunctive Query Answering for an Expressive Description Logic without Inverses. AAAI 2008: 504-510
230		Thomas Eiter, Michael Fink, Ján Senko: Error Classification in Action Descriptions: A Heuristic Approach. AAAI 2008: 905-910
229	EE	Magdalena Ortiz, Mantas Simkus, Thomas Eiter: Conjunctive Query Answering in SH using Knots. Description Logics 2008
228	EE	Thomas Eiter, Michael Fink, Gianluigi Greco, Domenico Lembo: Repair localization for query answering from inconsistent databases. ACM Trans. Database Syst. 33(2): (2008)
227	EE	Alexander Reiterer, Uwe Egly, Thomas Eiter, Heribert Kahmen: A knowledge-based videotheodolite measurement system for object representation/monitoring. Advances in Engineering Software 39(10): 821-827 (2008)
226	EE	Chitta Baral, Thomas Eiter, Marcus Bjäreland, Mutsumi Nakamura: Maintenance goals of agents in a dynamic environment: Formulation and policy construction. Artif. Intell. 172(12-13): 1429-1469 (2008)
225	EE	Thomas Eiter, Giovambattista Ianni, Thomas Lukasiewicz, Roman Schindlauer, Hans Tompits: Combining answer set programming with description logics for the Semantic Web. Artif. Intell. 172(12-13): 1495-1539 (2008)
224	EE	Thomas Eiter, Kewen Wang: Semantic forgetting in answer set programming. Artif. Intell. 172(14): 1644-1672 (2008)
223	EE	Thomas Eiter, Esra Erdem, Wolfgang Faber: Undoing the effects of action sequences. J. Applied Logic 6(3): 380-415 (2008)


N3Logic: A Logical Framework For the World Wide Web

Tim Berners-Lee, Dan Connolly, Lalana Kagal, Yosi Scharf, Jim Hendler

- A lot of Web data already available “out there”
- it’s linked

Where is the data? 1/4



KBS

http://www.kr.tuwien.ac.at/staff/eiter/

TU WIEN KBS Knowledge - Based Systems Group

staff education research contact services

kbs > staff > thomas eiter >

Drucken

Prof. Dr. Thomas Eiter

Professor of Computer Science
Head of the Knowledge Based Systems Group
Head of the Institute of Information Systems

EMAIL:	eiter at kr.tuwien.ac.at
PHONE:	+43 (1) 58 801-18460
FAX:	+43 (1) 58 801-18493
SEC:	+43 (1) 58 801-18405
OFFICE HOURS:	Thu, 17:00 - 18:00

Research interests

Giovambattista Ianni's Home Page

http://www.mat.unical.it/~ianni/

G.B. Ianni

Dipartimento di Matematica, Cubo 30B, Universita' della Calabria, 87036 Rende (CS), Italy

Phone: +39 0984 49 6430 Fax: +39 0984 49 6410
E-mail: ianni_AT_mat.unical.it

Assistant Professor at the Dipartimento di Matematica, Facolta' di Scienze M.F.N., Universita' della Calabria (Rende, Italy).

Search

Thomas Krennwallner's Wiki FrontPage

Thomas Krennwallner
<tkren AT SPAMFREE kr DOT tuwien DOT ac DOT at>

Obviously we do not want to leave zombies around. -- W. Richard Stevens

Here is some personal information about me.

Publications.

See MyConfig for an incomplete list of my configuration files.

FrontPage (last edited 2007-12-24 13:20:54 by ThomasKrennwallner)

MoinMoin Powered Python Powered Valid HTM

DBLP: Thomas Eiter

http://www.informatik.uni-trier.de/~ley/db/indices/a-tr

dblp.uni-trier.de Computer Science Bibliography

Universität Trier

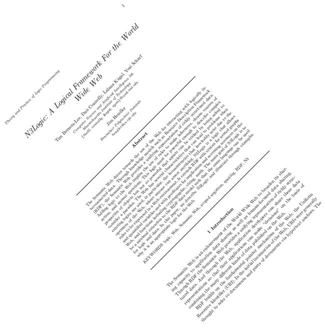
Thomas Eiter

List of publications from the DBLP Bibliography Server - FAQ

Coauthor Index - Ask others: ACM DL/Guide - CiteSeer - CSB - Google - MSN - Yahoo

Home Page

2008		
231		Magdalena Ortiz, Mantas Simkus, Thomas Eiter: Worst-case Optimal Conjunctive Query Answering for an Expressive Description Logic without Inverses. AAAI 2008: 504-510
230		Thomas Eiter, Michael Fink, Ján Senko: Error Classification in Action Descriptions: A Heuristic Approach. AAAI 2008: 905-910
229	EE	Magdalena Ortiz, Mantas Simkus, Thomas Eiter: Conjunctive Query Answering in SH using Knots. Description Logics 2008
228	EE	Thomas Eiter, Michael Fink, Gianluigi Greco, Domenico Lembo: Repair localization for query answering from inconsistent databases. ACM Trans. Database Syst. 33(2): (2008)
227	EE	Alexander Reiterer, Uwe Egly, Thomas Eiter, Heribert Kahmen: A knowledge-based videotheodolite measurement system for object representation/monitoring. Advances in Engineering Software 39(10): 821-827 (2008)
226	EE	Chitta Baral, Thomas Eiter, Marcus Bjäreland, Mutsumi Nakamura: Maintenance goals of agents in a dynamic environment: Formulation and policy construction. Artif. Intell. 172(12-13): 1429-1469 (2008)
225	EE	Thomas Eiter, Giovambattista Ianni, Thomas Lukasiewicz, Roman Schindlauer, Hans Tompits: Combining answer set programming with description logics for the Semantic Web. Artif. Intell. 172(12-13): 1495-1539 (2008)
224	EE	Thomas Eiter, Kewen Wang: Semantic forgetting in answer set programming. Artif. Intell. 172(14): 1644-1672 (2008)
223	EE	Thomas Eiter, Esra Erdem, Wolfgang Faber: Undoing the effects of action sequences. J. Applied Logic 6(3): 380-415 (2008)



1

Theory and Practice of Logic Programming

N3Logic: A Logical Framework For the World Wide Web

Tim Berners-Lee, Dan Connolly, Lalana Kagal, Yosi Scharf
Computer Science and Artificial Intelligence lab
Massachusetts Institute of Technology
{timbl, connolly, lkagal, syosi}@csail.mit.edu

Jim Hendler
Rensselaer Polytechnic Institute
hendler@cs.rpi.edu

Abstract

The Semantic Web drives towards the use of the Web for interacting with logically interconnected data. Through knowledge models such as Resource Description Framework (RDF), the Semantic Web provides a unifying representation of richly structured data. Adding logic to the Web implies the use of rules to make inferences, choose courses of action, and answer questions. This logic must be powerful enough to describe complex properties of objects but not so powerful that agents can be tricked by being asked to consider a paradox. The Web has several characteristics that can lead to problems when existing logics are used, in particular, the inconsistencies that inevitably arise due to the openness of the Web, where anyone can assert anything. N3Logic is a logic that allows rules to be expressed in a Web environment. It extends RDF with syntax for nested graphs and quantified variables and with predicates for implication and accessing resources on the Web, and functions including cryptographic, string, math. The main goal of N3Logic is to be a minimal extension to the RDF data model such that the same language can be used for logic and data. In this paper, we describe N3Logic and illustrate through examples why it is an appropriate logic for the Web.

KEYWORDS: logic, Web, Semantic Web, scoped negation, quoting, RDF, N3

1 Introduction

The Semantic Web is an enhancement of the World Wide Web to broaden its sharing capacity to application data sharing as well as human-focused data sharing. Through RDF the Semantic Web provides a unifying representation of richly structured data. And through the Web, application developers can share these data representations so that their applications can make "decisions" on the basis of combining the many different kinds of data published on the Web. RDF builds on the fundamental pointer mechanism of the Web; the Uniform Resource Identifier (URI). In the initial incarnation of the Web, URIs were generally thought to refer to documents and parts of documents via hypertext anchors. The

- A lot of Web data already available "out there"
- it's linked
- More and more of it available in in a machine-readable format (RDF) following the *Linked Data* principles (cf. introductory slides from last week)

Where is the data? 2/4

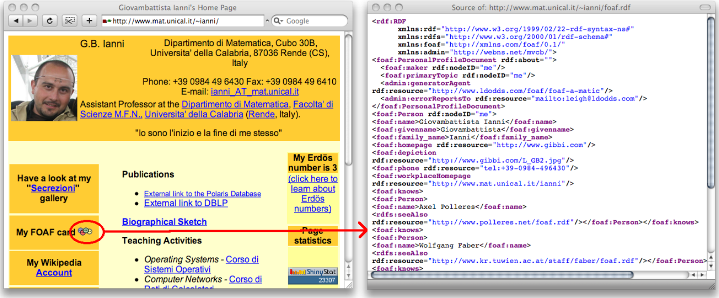
Obtaining Machine-Readable RDF data
(i) RDF directly by the publishers, (ii) as RDFa by content management systems, or (iii) by 3rd-party wrappers:

Where is the data? 2/4

Obtaining Machine-Readable RDF data

(i) RDF directly by the publishers, (ii) as RDFa by content management systems, or (iii) by 3rd-party wrappers:

"FOAF-files" in RDF linked from a home page: personal data (`foaf:name`, `foaf:phone`, etc.), relationships `foaf:knows`, `rdfs:seeAlso`)


Giovambattista Ianni's Home Page
http://www.mat.unical.it/~ianni/
Google
Source of: http://www.mat.unical.it/~ianni/foaf.rdf
ShinyStat
23307

G.B. Ianni

Dipartimento di Matematica, Cubo 30B,
Universita' della Calabria, 87036 Rende (CS),
Italy

Phone: +39 0984 49 6430 Fax: +39 0984 49 6410
E-mail: ianni_AT_mat.unical.it

Assistant Professor at the Dipartimento di Matematica, Facolta' di Scienze M.F.N., Universita' della Calabria (Rende, Italy).

"Io sono l'inizio e la fine di me stesso"

Have a look at my "Secrezioni" gallery

My FOAF card

My Wikipedia Account

Publications

- External link to the Polaris Database
- External link to DBLP

Biographical Sketch

Teaching Activities

- *Operating Systems* - Corso di Sistemi Operativi
- *Computer Networks* - Corso di Reti di Calcolatori

My Erdös number is 3 (click here to learn about Erdös numbers)

Page statistics

```
<rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:foaf="http://xmlns.com/foaf/0.1/"
      xmlns:admin="http://webns.net/mvcb/">
<foaf:PersonalProfileDocument rdf:about="">
  <foaf:maker rdf:nodeID="me"/>
  <foaf:primaryTopic rdf:nodeID="me"/>
  <admin:generatorAgent
rdf:resource="http://www.ldodds.com/foaf/foaf-a-matic"/>
  <admin:errorReportsTo rdf:resource="mailto:leigh@ldodds.com"/>
</foaf:PersonalProfileDocument>
<foaf:Person rdf:nodeID="me">
<foaf:name>Giovambattista Ianni</foaf:name>
<foaf:givenname>Giovambattista</foaf:givenname>
<foaf:family_name>Ianni</foaf:family_name>
<foaf:homepage rdf:resource="http://www.gibbi.com"/>
<foaf:depiction
rdf:resource="http://www.gibbi.com/L_GB2.jpg"/>
<foaf:phone rdf:resource="tel:+39-0984-496430"/>
<foaf:workplaceHomepage
rdf:resource="http://www.mat.unical.it/ianni"/>
<foaf:knows>
<foaf:Person>
<foaf:name>Axel Polleres</foaf:name>
<rdfs:seeAlso
rdf:resource="http://www.polleres.net/foaf.rdf"/></foaf:Person></foaf:knows>
<foaf:knows>
<foaf:Person>
<foaf:name>Wolfgang Faber</foaf:name>
<rdfs:seeAlso
rdf:resource="http://www.kr.tuwien.ac.at/staff/faber/foaf.rdf"/></foaf:Person>
<foaf:knows>
```

Different Options:

e.g. linking RDF/XML [Beckett and McBride (eds.), 2004] from (X)HTML,

Let's check, e.g.`http://www.w3.org/People/Berners-Lee/`, or

`http://www.cs.rpi.edu/~hendler/`

Where is the data? 3/4

Obtaining Machine-Readable RDF data
(i) directly by the publishers, (ii) as RDFa by content management systems, or (iii) by 3rd-party wrappers:

Some sites provide RDF in terms of microformats, or RDFa (=RDF embedded in HTML), e.g. on `http://bestbuy.com`, more and more using `http://schema.org` metadata

- ... try using W3C's RDFa Distiller: `http://www.w3.org/2007/08/pyRdfa/`
- This RDFa is often generated directly by the underlying CMS (e.g. Drupal provides modules for RDFa)

Where is the data? 4/4

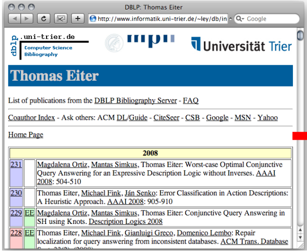
Obtaining Machine-Readable RDF data
(i) directly by the publishers, (ii) as RDFa by content management systems,, or (iii) by 3rd-party wrappers:

L3S' RDF export of the DBLP citation index, see http://dblp.l3s.de/d2r/

Where is the data? 4/4

Obtaining Machine-Readable RDF data
(i) directly by the publishers, (ii) as RDFa by content management systems,, or (iii) by 3rd-party wrappers:

L3S' RDF export of the DBLP citation index, see http://dblp.l3s.de/d2r/



DBLP: Thomas Eiter

http://www.informatik.uni-trier.de/~ley/db/in

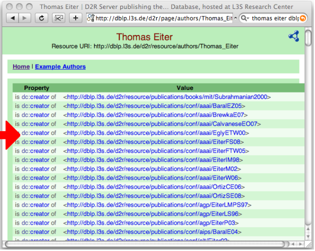
Thomas Eiter

List of publications from the DBLP Bibliography Server - FAQ

Coauthor Index - Ask others: ACM DL/Guide - CiteSeer - CSB - Google - MSN - Yahoo

Home Page

2008		
231		Magdalena Ortiz, Mantas Simkus, Thomas Eiter: Worst-case Optimal Conjunctive Query Answering for an Expressive Description Logic without Inverses. AAAI 2008: 504-510
230		Thomas Eiter, Michael Fink, Ján Senko: Error Classification in Action Descriptions: A Heuristic Approach. AAAI 2008: 905-910
229	EE	Magdalena Ortiz, Mantas Simkus, Thomas Eiter: Conjunctive Query Answering in SH using Knots. Description Logics 2008
228	EE	Thomas Eiter, Michael Fink, Gianluigi Greco, Domenico Lembo: Repair localization for query answering from inconsistent databases. ACM Trans. Database



Thomas Eiter | D2R Server publishing the... Database, hosted at L3S Research Center

http://dblp.l3s.de/d2r/page/authors/Thomas_Eit

Thomas Eiter
Resource URI: http://dblp.l3s.de/d2r/resource/authors/Thomas_Eiter

Home | Example Authors

Property	Value
is dc:creator of	<http://dblp.l3s.de/d2r/resource/publications/books/mit/Subrahmanian2000>
is dc:creator of	<http://dblp.l3s.de/d2r/resource/publications/conf/aaai/BaralEZ05>
is dc:creator of	<http://dblp.l3s.de/d2r/resource/publications/conf/aaai/BrewkaE07>
is dc:creator of	<http://dblp.l3s.de/d2r/resource/publications/conf/aaai/CalvaneseEO07>
is dc:creator of	<http://dblp.l3s.de/d2r/resource/publications/conf/aaai/EglyETW00>
is dc:creator of	<http://dblp.l3s.de/d2r/resource/publications/conf/aaai/EiterFS08>
is dc:creator of	<http://dblp.l3s.de/d2r/resource/publications/conf/aaai/EiterFTW05>
is dc:creator of	<http://dblp.l3s.de/d2r/resource/publications/conf/aaai/EiterIM98>
is dc:creator of	<http://dblp.l3s.de/d2r/resource/publications/conf/aaai/EiterM02>
is dc:creator of	<http://dblp.l3s.de/d2r/resource/publications/conf/aaai/EiterW06>
is dc:creator of	<http://dblp.l3s.de/d2r/resource/publications/conf/aaai/OrtizCE06>
is dc:creator of	<http://dblp.l3s.de/d2r/resource/publications/conf/aaai/OrtizSE08>
is dc:creator of	<http://dblp.l3s.de/d2r/resource/publications/conf/agp/EiterLMPS97>
is dc:creator of	<http://dblp.l3s.de/d2r/resource/publications/conf/agp/EiterLS96>
is dc:creator of	<http://dblp.l3s.de/d2r/resource/publications/conf/agp/EiterP03>
is dc:creator of	<http://dblp.l3s.de/d2r/resource/publications/conf/aips/BaralE04>

- Gives unique URIs to authors, documents, etc. on DBLP! E.g., http://dblp.l3s.de/d2r/resource/authors/Thomas_Eiter, http://dblp.l3s.de/d2r/resource/authors/Tim_Berners-Lee, http://dblp.l3s.de/d2r/resource/publications/journals/tplp/Berners-LeeCKSH08, etc.
- Provides RDF version of all DBLP data + query interface!
- Other nice example: RDF+query interface for large parts of wikipedia: http://dbpedia.org/

How can I query that data? SPARQL

SPARQL – W3C approved standardized query language for RDF:

- look-and-feel of "SQL for the Web"
- allows to ask queries like
- "*All documents created by Thomas Eiter*"
- "*Names of all persons who co-authored with authors of the present paper*"
- "*Names of persons who know Tim Berners-Lee or who are known by Tim Berners-Lee*"
- "*All people who have published in TPLP but have not co-authored with any of the authors of the present paper*"

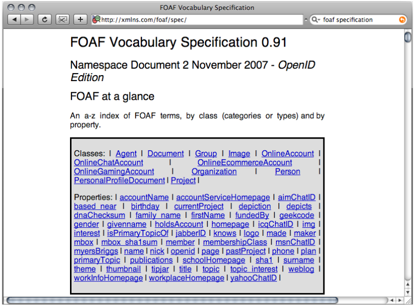
Example (**query1.sparql**):

```
SELECT ?D
FROM <http://dblp.l3s.de/d2r/data/authors/Thomas_Eiter>
WHERE {?D dc:creator <http://dblp.l3s.de/d2r/resource/authors/Thomas_Eiter>}
```

What does the data mean?

Data, i.e. the used *vocabulary* to write down RDF is described by *ontologies*,themselves published in RDF, e.g.:

- Friend-of-a-Friend (FOAF) [Brickley and Miller, 2007]
- Socially-Interlinked-Online-Communities (SIOC) [Bojārs *et al.*, 2007]
- Dublin Core [Nilsson *et al.*, 2008]



FOAF Vocabulary Specification 0.91

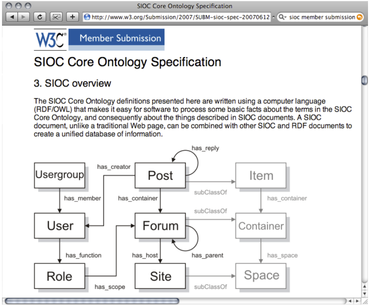
Namespace Document 2 November 2007 - *OpenID Edition*

FOAF at a glance

An a-z index of FOAF terms, by class (categories or types) and by property.

Classes: | Agent | Document | Group | Image | OnlineAccount | OnlineChatAccount | OnlineEcommerceAccount | OnlineGamingAccount | Organization | Person | PersonalProfileDocument | Project |

Properties: | accountName | accountServiceHomepage | aimChatID | based_near | birthday | currentProject | depiction | depicts | dnaChecksum | family_name | firstName | fundedBy | geekcode | gender | givenname | holdsAccount | homepage | icqChatID | img | interest | isPrimaryTopicOf | jabberID | knows | logo | made | maker | mbox | mbox_sha1sum | member | membershipClass | msnChatID | myersBriggs | name | nick | openid | page | pastProject | phone | plan | primaryTopic | publications | schoolHomepage | sha1 | surname | theme | thumbnail | tipjar | title | topic | topic_interest | weblog | workInfoHomepage | workplaceHomepage | yahooChatID |


SIOC Core Ontology Specification
http://www.w3.org/Submission/2007/SUBM-sioc-spec-20070612
sioc member submission
W3C
Member Submission

SIOC Core Ontology Specification

3. SIOC overview

The SIOC Core Ontology definitions presented here are written using a computer language (RDF/OWL) that makes it easy for software to process some basic facts about the terms in the SIOC Core Ontology, and consequently about the things described in SIOC documents. A SIOC document, unlike a traditional Web page, can be combined with other SIOC and RDF documents to create a unified database of information.

Unit Outline

1. Motivation – Aggregating Web Data

2. How can I publish data? RDF

3. How can I query that data? SPARQL

4. What does that data mean? Ontologies described in RDFS + OWL

5. What's next?

Semantic Web Data: The Resource Description Framework (RDF)

- RDF is describing *resources* per triples/statements
  Subject **P**redicate **O**bject.

Semantic Web Data: The Resource Description Framework (RDF)

- RDF is describing *resources* per triples/statements

  Subject Predicate Object.
- "simplest possible database schema", data just a set of triples:

Semantic Web Data: The Resource Description Framework (RDF)

- RDF is describing *resources* per triples/statements

  Subject **P**redicate **O**bject.

- "simplest possible database schema", data just a set of triples:

  axel isA Person .
  axel hasName "Axel Polleres".
  axel knows gb .
  axel knows thomas.
  thomas hasCreated an Article
  titled "Rules and Ontologies for the Semantic Web".

Semantic Web Data: The Resource Description Framework (RDF)

- RDF is describing *resources* per triples/statements

  Subject **P**redicate **O**bject.

- "simplest possible database schema", data just a set of triples:

  axel isA Person .
  axel hasName "Axel Polleres".
  axel knows gb .
  axel knows thomas.
  $\exists X$ *thomas hasCreated* X . X *isA Article .*
  X *hasTitle "Rules and Ontologies for the Semantic Web".*

Semantic Web Data: The Resource Description Framework (RDF)

- RDF is describing *resources* per triples/statements

  Subject **P**redicate **O**bject.

- "simplest possible database schema", data just a set of triples:

  axel isA Person .
  axel hasName "Axel Polleres".
  axel knows gb .
  axel knows thomas.
  $\exists X$ *thomas hasCreated* X . X *isA Article .*
  X *hasTitle "Rules and Ontologies for the Semantic Web".*

- abstracts away from tables (RDBMS) or tree-like (XML) schemas

Semantic Web Data: The Resource Description Framework (RDF)

- RDF is describing *resources* per triples/statements

  Subject **P**redicate **O**bject.

- "simplest possible database schema", data just a set of triples:

  axel isA Person .
  axel hasName "Axel Polleres".
  axel knows gb .
  axel knows thomas.
  $\exists X$ *thomas hasCreated* X *.* X *isA Article .*
  X *hasTitle "Rules and Ontologies for the Semantic Web".*

- abstracts away from tables (RDBMS) or tree-like (XML) schemas
- triples can be viewed as edges of a labeled,directed graph.

Semantic Web Data: The Resource Description Framework (RDF)

- RDF is describing *resources* per triples/statements

  Subject **P**redicate **O**bject.

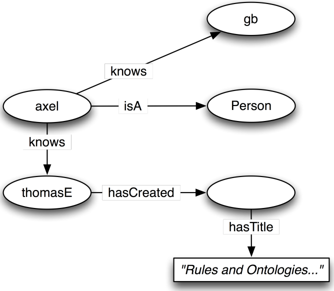
- "simplest possible database schema", data just a set of triples:

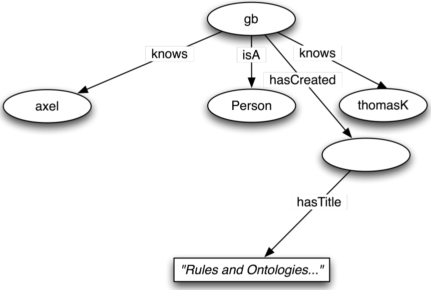
  axel isA Person .
  axel hasName "Axel Polleres".
  axel knows gb .
  axel knows thomas.
  $\exists X$ *thomas hasCreated* X *.* X *isA Article .*
  X *hasTitle "Rules and Ontologies for the Semantic Web".*

- abstracts away from tables (RDBMS) or tree-like (XML) schemas
- triples can be viewed as edges of a labeled,directed graph.
- main advantage: Graphs are easy to merge! (Trees,Tables aren't)

axel isA Person .
axel knows gb .
axel knows thomasE.
thomasE hasCreated X . X isA Article .
X hasTitle "Rules and Ontologies...".

gb isA Person .
gb knows axel .
gb knows thomasK.
gb hasCreated Y . Y isA Article .
Y hasTitle "Rules and Ontologies...".

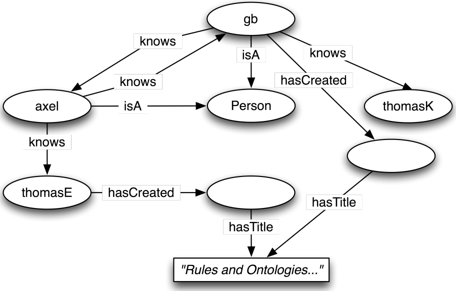




Observe: the "existential variables" became "blank" nodes in the Graph.

axel isA Person .
axel knows gb .
axel knows thomasE.
thomasE hasCreated X . X isA Article .
X hasTitle "Rules and Ontologies. . .".

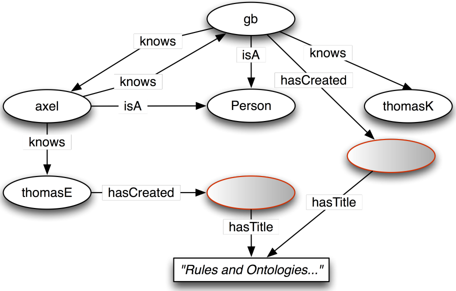
gb isA Person .
gb knows axel .
gb knows thomasK.
gb hasCreated Y . Y isA Article .
Y hasTitle "Rules and Ontologies. . .".



Observe: the "existential variables" became "blank" nodes in the Graph.

axel isA Person .
axel knows gb .
axel knows thomasE.
thomasE hasCreated X . X isA Article .
X hasTitle "Rules and Ontologies. . .".

gb isA Person .
gb knows axel .
gb knows thomasK.
gb hasCreated Y . Y isA Article .
Y hasTitle "Rules and Ontologies. . .".



Observe: the "existential variables" became "blank" nodes in the Graph. Note that we have no reason to assume that the two blank nodes are the same.

A Syntax for RDF: Turtle

There are different syntaxes for RDF

- RDF/XML [Beckett and McBride (eds.), 2004]
- Turtle [Beckett *et al.*, 2008], N3 [Berners-Lee and Connolly, 2008]
- RDFa [Adida *et al.*, 2008] (i.e., RDF "embedded" in (X)HTML)
- RDF in JSON

We'll use Turtle syntax in this lecture:

- it is a subset of Notation 3 [Berners-Lee and Connolly, 2008]
- sufficient to write all RDF
- almost human-readable
- also the basis for SPARQL
- upcoming W3C standard!
- tools and APIs to convert from one syntax into the other

Resources in RDF, Turtle Syntax

- Resources are identified by URIs (to encourage web-wide unique identifiers)
- There are special URIs, defined in vocabularies (FOAF, SIOC, RDF, etc.)
- Objects can be literals,

axel isA Person .
axel hasName "Axel Polleres".

Resources in RDF, Turtle Syntax

- Resources are identified by URIs (to encourage web-wide unique identifiers)
- There are special URIs, defined in vocabularies (FOAF, SIOC, RDF, etc.)
- Objects can be literals,

axel isA Person .
axel hasName “Axel Polleres”.

becomes:

```
<http://polleres.net/foaf.rdf#me> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
                    <http://xmlns.com/foaf/0.1/Person>.
<http://polleres.net/foaf.rdf#me> <http://xmlns.com/foaf/0.1/name>
                    ‘‘Axel Polleres’’.
```

Resources in RDF, Turtle Syntax

- Resources are identified by URIs (to encourage web-wide unique identifiers)
- There are special URIs, defined in vocabularies (FOAF, SIOC, RDF, etc.)
- Objects can be literals, occasionally with a datatype

axel isA Person .
axel hasName "Axel Polleres".

becomes:

```
<http://polleres.net/foaf.rdf#me> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
                    <http://xmlns.com/foaf/0.1/Person>.
<http://polleres.net/foaf.rdf#me> <http://xmlns.com/foaf/0.1/name>
                    ‘‘Axel Polleres’’^^<http://www.w3.org/2001/XMLSchema#string>.
```

Resources in RDF, Turtle Syntax

- Resources are identified by URIs (to encourage web-wide unique identifiers)
- There are special URIs, defined in vocabularies (FOAF, SIOC, RDF, etc.)
- Objects can be literals, occasionally with a datatype

axel isA Person .
axel hasName “Axel Polleres”.

becomes:

```
<http://polleres.net/foaf.rdf#me> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
                    <http://xmlns.com/foaf/0.1/Person>.
<http://polleres.net/foaf.rdf#me> <http://xmlns.com/foaf/0.1/name>
                    ‘‘Axel Polleres’’^^<http://www.w3.org/2001/XMLSchema#string>.
```

Ugly to read. . . more compact syntaxes like Turtle [Beckett *et al.*, 2008] allow prefix definitions á la XML:

```
@prefix :  <http://polleres.net/foaf.rdf#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
@prefix xs:  <http://www.w3.org/2001/XMLSchema#> .
:me rdf:type foaf:Person .
:me foaf:name ‘‘Axel Polleres’’^^xs:string.
```

More on RDF – Shortcuts in Turtle Syntax

```
@prefix :  <http://polleres.net/foaf.rdf#>
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
@prefix dc:  <http://purl.org/dc/elements/1.1/> .
:me rdf:type foaf:Person .
:me foaf:name ‘‘Axel Polleres’’ .
:me foaf:knows <http://dblp.l3s.de/d2r/data/authors/Giovambattista_Ianni> .
:me foaf:knows <http://dblp.l3s.de/d2r/page/authors/Thomas_Eiter> .
<http://dblp.l3s.de/d2r/page/authors/Thomas_Eiter> dc:creator X .
X rdf:type foaf:Document .
X dc:title ‘‘Rules and Ontologies for the Semantic Web’’.
```

More on RDF – Shortcuts in Turtle Syntax

```
@prefix :  <http://polleres.net/foaf.rdf#>
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
@prefix dc:  <http://purl.org/dc/elements/1.1/> .
:me rdf:type foaf:Person .
:me foaf:name ‘‘Axel Polleres’’ .
:me foaf:knows <http://dblp.l3s.de/d2r/data/authors/Giovambattista_Ianni> .
:me foaf:knows <http://dblp.l3s.de/d2r/page/authors/Thomas_Eiter> .
<http://dblp.l3s.de/d2r/page/authors/Thomas_Eiter> dc:creator _:X .
_:X rdf:type foaf:Document .
_:X dc:title ‘‘Rules and Ontologies for the Semantic Web’’.
```

- Blank nodes in Turtle are written as _:*Varname*

More on RDF – Shortcuts in Turtle Syntax

```
@prefix :  <http://polleres.net/foaf.rdf#>
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
@prefix dc:  <http://purl.org/dc/elements/1.1/> .
:me rdf:type foaf:Person ;
    foaf:name ‘‘Axel Polleres’’ ;
    foaf:knows <http://dblp.l3s.de/d2r/data/authors/Giovambattista_Ianni> ,
             <http://dblp.l3s.de/d2r/page/authors/Thomas_Eiter> .
<http://dblp.l3s.de/d2r/page/authors/Thomas_Eiter> dc:creator _:X .
_:X rdf:type foaf:Document ;
             dc:title ‘‘Rules and Ontologies for the Semantic Web’’ .
```

- Blank nodes in Turtle are written as _:*Varname*
- Turtle allows shortcuts:
  - Same subject triples can be grouped together with ';' , ','

More on RDF – Shortcuts in Turtle Syntax

```
@prefix :  <http://polleres.net/foaf.rdf#>
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
@prefix dc:  <http://purl.org/dc/elements/1.1/> .
:me rdf:type foaf:Person;
    foaf:name ‘‘Axel Polleres’’;
    foaf:knows <http://dblp.l3s.de/d2r/data/authors/Giovambattista_Ianni> ,
             <http://dblp.l3s.de/d2r/page/authors/Thomas_Eiter> .
<http://dblp.l3s.de/d2r/page/authors/Thomas_Eiter> dc:creator [
             rdf:type foaf:Document ;
             dc:title ‘‘Rules and Ontologies for the Semantic Web’’ ] .
```

- Blank nodes in Turtle are written as _:*Varname*
- Turtle allows shortcuts:
  - Same subject triples can be grouped together with ';' , ','
  - Blank nodes can be grouped/replaced using “bracket syntax” '[', ']'

More on RDF – Shortcuts in Turtle Syntax

```
@prefix :  <http://polleres.net/foaf.rdf#>
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
@prefix dc:  <http://purl.org/dc/elements/1.1/> .
:me a foaf:Person;
    foaf:name ‘‘Axel Polleres’’;
    foaf:knows <http://dblp.l3s.de/d2r/data/authors/Giovambattista_Ianni> ,
             <http://dblp.l3s.de/d2r/page/authors/Thomas_Eiter> .
<http://dblp.l3s.de/d2r/page/authors/Thomas_Eiter> dc:creator [
             a foaf:Document ;
             dc:title ‘‘Rules and Ontologies for the Semantic Web’’ ] .
```

- Blank nodes in Turtle are written as `_:`*Varname*
- Turtle allows shortcuts:
  - Same subject triples can be grouped together with ';' , ','
  - Blank nodes can be grouped/replaced using “bracket syntax” '[', ']'
  - `rdf:type` is often abbreviated with a.

More on RDF – Shortcuts in Turtle Syntax

```
@prefix :  <http://polleres.net/foaf.rdf#>
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
@prefix dc:  <http://purl.org/dc/elements/1.1/> .
:me  foaf:Person;
    foaf:name ‘‘Axel Polleres’’^^xs:string;
    foaf:knows <http://dblp.l3s.de/d2r/data/authors/Giovambattista_Ianni> ,
             <http://dblp.l3s.de/d2r/page/authors/Thomas_Eiter> .
<http://dblp.l3s.de/d2r/page/authors/Thomas_Eiter> dc:creator [
             a foaf:Document ;
             dc:title ‘‘Rules and Ontologies for the Semantic Web’’ ] .
```

- Blank nodes in Turtle are written as _:*Varname*
- Turtle allows shortcuts:
  - Same subject triples can be grouped together with ';' , ','
  - Blank nodes can be grouped/replaced using "bracket syntax" '[', ']'
  - `rdf:type` is often abbreviated with a.
  - typed literals l of type dt are written as l^^dt.

More on RDF – Shortcuts in Turtle Syntax

```
@prefix :  <http://polleres.net/foaf.rdf#>
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
@prefix dc:  <http://purl.org/dc/elements/1.1/> .
:me  foaf:Person;
    foaf:name ‘‘Axel Polleres’’^^xs:string;
    foaf:knows <http://dblp.l3s.de/d2r/data/authors/Giovambattista_Ianni> ,
             <http://dblp.l3s.de/d2r/page/authors/Thomas_Eiter> .
<http://dblp.l3s.de/d2r/page/authors/Thomas_Eiter> dc:creator [
             a foaf:Document ;
             dc:title ‘‘Rules and Ontologies for the Semantic Web’’@en ] .
```

- Blank nodes in Turtle are written as _:*Varname*
- Turtle allows shortcuts:
  - Same subject triples can be grouped together with ';' , ','
  - Blank nodes can be grouped/replaced using “bracket syntax” '[', ']'
  - `rdf:type` is often abbreviated with a.
  - typed literals l of type dt are written as l^^dt.
  - untyped literals can have a language tag [BCP-47, 2006].

More on RDF – Shortcuts in Turtle Syntax

```
@prefix :  <http://polleres.net/foaf.rdf#>
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
@prefix dc:  <http://purl.org/dc/elements/1.1/> .
:me  foaf:Person;
    foaf:name ‘‘Axel Polleres’’^^xs:string;
    foaf:knows <http://dblp.l3s.de/d2r/data/authors/Giovambattista_Ianni> ,
             <http://dblp.l3s.de/d2r/page/authors/Thomas_Eiter> .
<http://dblp.l3s.de/d2r/page/authors/Thomas_Eiter> dc:creator [
             a foaf:Document ;
             dc:title ‘‘Rules and Ontologies for the Semantic Web’’@en ] .
```

- Blank nodes in Turtle are written as _:*Varname*
- Turtle allows shortcuts:
  - Same subject triples can be grouped together with ';' , ','
  - Blank nodes can be grouped/replaced using “bracket syntax” '[', ']'
  - `rdf:type` is often abbreviated with a.
  - typed literals l of type dt are written as $l^{\wedge\wedge}dt$.
  - untyped literals can have a language tag [BCP-47, 2006].
  - (untyped literals with or without language tag are also called “plain” literals.)

Collecting RDF from the Web

- For us this is enough so far to “read” RDF on the Web.

[1] http://librdf.org/

[2] http://jena.sourceforge.net/

Collecting RDF from the Web

- For us this is enough so far to “read” RDF on the Web.
- For published RDF data there exists a machine-readable XML syntax. Lots of tools and APIs to read/process/convert this data (Redland (C++),[1] Jena (Java),[2] etc.)

```
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
@prefix :  <http://www.gibbi.com/foaf.rdf#> .

<http://www.gibbi.com/foaf.rdf> a foaf:PersonalProfileDocument.
<http://www.gibbi.com/foaf.rdf> foaf:maker :me .
<http://www.gibbi.com/foaf.rdf> foaf:primaryTopic :me .
:me a foaf:Person .
:me foaf:name "Giovambattista Ianni" .
:me foaf:homepage <http://www.gibbi.com> .
:me foaf:knows [ a foaf:Person ;
         foaf:name "Wolfgang Faber" ;
         rdfs:seeAlso <http://www.kr.tuwien.ac.at/staff/faber/foaf.rdf>].
:me foaf:knows [ a foaf:Person .
         foaf:name "Axel Polleres" ;
         rdfs:seeAlso <http://www.polleres.net/foaf.rdf> ].
:me foaf:knows [ a foaf:Person .
         foaf:name "Thomas Eiter" ] .
:me foaf:knows [ a foaf:Person .
         foaf:name "Alessandra Martello" ] .
```

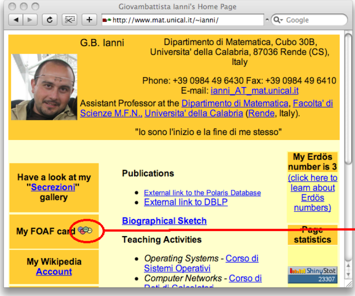
[1] http://librdf.org/

[2] http://jena.sourceforge.net/

Collecting RDF from the Web

- For us this is enough so far to "read" RDF on the Web.
- For published RDF data there exists a machine-readable XML syntax. Lots of tools and APIs to read/process/convert this data (Redland (C++),[1] Jena (Java),[2] etc.)

```
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
@prefix :  <http://www.gibbi.com/foaf.rdf#> .
```



Giovambattista Ianni's Home Page
http://www.mat.unical.it/~ianni/
Google

G.B. Ianni

Dipartimento di Matematica, Cubo 30B, Universita' della Calabria, 87036 Rende (CS), Italy

Phone: +39 0984 49 6430 Fax: +39 0984 49 6410
E-mail: ianni_AT_mat.unical.it

Assistant Professor at the Dipartimento di Matematica, Facolta' di Scienze M.F.N., Universita' della Calabria (Rende, Italy).

"Io sono l'inizio e la fine di me stesso"

Have a look at my "Secrezioni" gallery

My FOAF card

My Wikipedia Account

Publications

- External link to the Polaris Database
- External link to DBLP

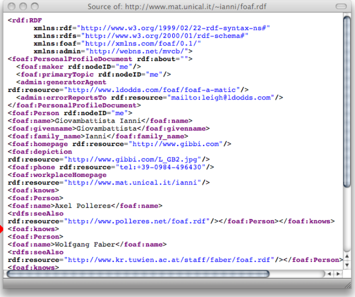
Biographical Sketch

Teaching Activities

- *Operating Systems* - Corso di Sistemi Operativi
- *Computer Networks* - Corso di Reti di Calcolatori

My Erdös number is 3 (click here to learn about Erdös numbers)

Page statistics



```
<rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:foaf="http://xmlns.com/foaf/0.1/"
      xmlns:admin="http://webns.net/mvcb/">
<foaf:PersonalProfileDocument rdf:about="">
  <foaf:maker rdf:nodeID="me"/>
  <foaf:primaryTopic rdf:nodeID="me"/>
  <admin:generatorAgent
rdf:resource="http://www.ldodds.com/foaf/foaf-a-matic"/>
  <admin:errorReportsTo rdf:resource="mailto:leigh@ldodds.com"/>
</foaf:PersonalProfileDocument>
<foaf:Person rdf:nodeID="me">
<foaf:name>Giovambattista Ianni</foaf:name>
<foaf:givenname>Giovambattista</foaf:givenname>
<foaf:family_name>Ianni</foaf:family_name>
<foaf:homepage rdf:resource="http://www.gibbi.com"/>
<foaf:depiction
rdf:resource="http://www.gibbi.com/L_GB2.jpg"/>
<foaf:phone rdf:resource="tel:+39-0984-496430"/>
<foaf:workplaceHomepage
rdf:resource="http://www.mat.unical.it/ianni"/>
<foaf:knows>
<foaf:Person>
<foaf:name>Axel Polleres</foaf:name>
<rdfs:seeAlso
rdf:resource="http://www.polleres.net/foaf.rdf"/></foaf:Person></foaf:knows>
<foaf:knows>
<foaf:Person>
<foaf:name>Wolfgang Faber</foaf:name>
<rdfs:seeAlso
rdf:resource="http://www.kr.tuwien.ac.at/staff/faber/foaf.rdf"/></foaf:Person>
<foaf:knows>
```

```
      foaf:name "Alessandra Martello" ] .
```

[1] http://librdf.org/

[2] http://jena.sourceforge.net/

Using *rapper*

Example tool for converting RDF between different syntaxesL: `rapper` (part of the Redland API, cf. `http://librdf.org/`), e.g.

```
rapper http://polleres.net/foaf.rdf -i rdfxml -o turtle
```

or

```
rapper http://polleres.net/teaching/SemWebTech_2012/testdata/foaf.ttl -i turtle -o rdfxml
```

cf. Part 1 of assignment 1:

- Create your own FOAF file. You can use a generator tool such as FOAF-a-Matic (returns RDF/XML) to generate a skeleton.
- Convert the FOAF file to Turtle syntax [...]

Unit Outline

1. Motivation – Aggregating Web Data
2. How can I publish data? RDF
3. How can I query that data? SPARQL
4. What does that data mean? Ontologies described in RDFS + OWL
5. What's next?

How can I query/aggregate RDF data? SPARQL

- First “ingredient”: a standardized query language – SPARQL [Prud’hommeaux and Seaborne, 2008;
  Harris and Seaborne, 2013] – based on graph pattern matching

Prologue:	**P**	PREFIX *prefix*: <*namespace-URI*>	
Head:	**C** or	CONSTRUCT { *template* }	. . . construct a new RDF graph
	S or	SELECT *variable list*	. . . select matching resources/literals in a graph
	A	ASK	. . . boolean query
Body:	**D**	FROM / FROM NAMED <*dataset-URI*>	
	W	WHERE { *pattern* }	
	M	ORDER BY *expression*	
		LIMIT *integer* > 0	
		OFFSET *integer* > 0	

How can I query/aggregate RDF data? SPARQL

- First "ingredient": a standardized query language – SPARQL [Prud'hommeaux and Seaborne, 2008;
  Harris and Seaborne, 2013] – based on graph pattern matching

Prologue:	**P**	`PREFIX` *prefix*: <*namespace-URI*>	
Head:	**C** or	`CONSTRUCT` { *template* }	. . . construct a new RDF graph
	S or	`SELECT` *variable list*	. . . select matching resources/literals in a graph
	A	`ASK`	. . . boolean query
Body:	**D**	`FROM` / `FROM NAMED` <*dataset-URI*>	
	W	`WHERE` { *pattern* }	
	M	`ORDER BY` *expression*	
		`LIMIT` *integer* > 0	
		`OFFSET` *integer* > 0	

- Let us start with `SELECT` queries and focus on the different patterns:
  - basic graph patterns (Conjunctive queries)
  - `FILTER`s
  - `UNION`s of patterns
  - `OPTIONAL` Patterns
  - `GRAPH` Patterns

Basic Graph Patterns (Conjunctive queries)

“select all names of persons known by G.B. from his FOAF file”

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
SELECT ?N
FROM <http://www.mat.unical.it/~ianni/foaf.rdf>
WHERE {
       <http://www.mat.unical.it/~ianni/foaf.rdf#me> foaf:knows ?X .
       ?X a foaf:Person .  ?X foaf:name ?N .
      }
```

- graph patterns (WHERE part) allow Turtle syntax

Basic Graph Patterns (Conjunctive queries)

"select all names of persons known by G.B. from his FOAF file"

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
SELECT ?N
FROM <http://www.mat.unical.it/~ianni/foaf.rdf>
WHERE {
       [ foaf:knows
        [ a foaf:Person; foaf:name ?N ]]
      }
```

- graph patterns (WHERE part) allow Turtle syntax
- all Turtle shortcuts allowed[3]

[3] We assume here that the only people declared "known" in G.B.'s FOAF file are those known by him.

Basic Graph Patterns (Conjunctive queries)

"select all names of persons known by G.B., Axel, and Thomas K. from their FOAF files"

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
SELECT ?N
FROM <http://www.mat.unical.it/~ianni/foaf.rdf>
FROM <http://www.polleres.net/foaf.rdf>
FROM <http://www.postsubmeta.net/foaf>
WHERE {
       [ foaf:knows
        [ a foaf:Person; foaf:name ?N ]]
      }
```

- graph patterns (WHERE part) allow Turtle syntax
- all Turtle shortcuts allowed[3]
- merge of several graphs can be queried at once

Basic Graph Patterns (Conjunctive queries)

"select all names of persons known by G.B., Axel, and Thomas K. from their FOAF files"

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
SELECT ?N
FROM <http://www.mat.unical.it/˜ianni/foaf.rdf>
FROM <http://www.polleres.net/foaf.rdf>
FROM <http://www.postsubmeta.net/foaf>
WHERE {
       [ foaf:knows
        [ a foaf:Person; foaf:name ?N ]]
      }
```

- graph patterns (`WHERE` part) allow Turtle syntax
- all Turtle shortcuts allowed[3]
- merge of several graphs can be queried at once
- Try it! E.g. using ARQ (`http://jena.sourceforge.net/ARQ/`)

  ```
  arq --query
  http://www.polleres.net/teaching/SemWebTech_2012/testdata/query2.sparql
  ```

FILTERs in Basic Graph Patterns

"select all names of persons known by GB, Thomas, and Axel from their FOAF files" (`query3.sparql`)

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
SELECT ?N
WHERE {
       [ foaf:knows
        [a foaf:Person ; foaf:name ?N] ]
      }
```

- graph patterns (`WHERE` part) allow Turtle syntax
- all Turtle shortcuts allowed
- Dataset can also be implicit, depending on the implementation

FILTERs in Basic Graph Patterns

"select all names of persons known by GB, Thomas, and Axel from their FOAF files" (query3.sparql)

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
SELECT ?N
WHERE {
       [ foaf:knows
        [a foaf:Person ; foaf:name ?N] ]
       FILTER ( ?N != "Giovambattista Ianni" &&
                ?N != "Thomas Krennwallner" && ?N != "Axel Polleres")
      }
```

- graph patterns (WHERE part) allow Turtle syntax
- all Turtle shortcuts allowed
- Dataset can also be implicit, depending on the implementation
- Now, let us try to filter out the authors' names from the result.

UNIONs (Disjunction)

"Names of persons who know Axel Polleres or *who are known by Axel Polleres"*

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
SELECT ?N
FROM ...
WHERE {
       { [ foaf:name "Axel Polleres" ] foaf:knows [foaf:name ?N ] }
       UNION
       { [ foaf:name ?N ] foaf:knows [foaf:name "Axel Polleres" ] }
      }
```

UNIONs (Disjunction)

"Names of persons who know Axel Polleres or who are known by Axel Polleres"

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
SELECT ?N
FROM ...
WHERE {
       { [ foaf:name "Axel Polleres" ] foaf:knows [foaf:name ?N ] }
       UNION
       { [ foaf:name ?N ] foaf:knows [foaf:name "Axel Polleres" ] }
      }
```

- UNION s allow alternative matching of several patterns, similar to UNIONs in SQL.

OPTIONALs 1/2 – Partial Matching

"Select all names of persons known by Axel from his FOAF file and – if available – their `rdfs:seeAlso` links" `query4.sparql`

```
PREFIX rdfs:  <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>

SELECT ?N ?L
FROM <http://www.polleres.net/foaf.rdf>
WHERE {<http://www.www.polleres.net/foaf.rdf#me> foaf:knows ?X .
       ?X foaf:name ?N . ?X rdfs:seeAlso ?L
      }
```

- "Normal" basic graph pattern doesn't work here, returns only those `?X` with **both** a name **and** a `rdfs:seeAlso` link.

?N	?L
"Dan Brickley"	<http://danbri.org/foaf.rdf>
"Ruben Lara Hernandez"	<http://nets.ii.uam.es/r̃lara/foaf.rdf>
...	

OPTIONALs 1/2 – Partial Matching

“Select all names of persons known by Axel from his FOAF file and – if available – their `rdfs:seeAlso` links” `query4.sparql`

```
PREFIX rdfs:  <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
SELECT ?N ?L
FROM <http://www.polleres.net/foaf.rdf>
WHERE {<http://www.www.polleres.net/foaf.rdf#me> foaf:knows ?X .
       ?X foaf:name ?N .  OPTIONAL  { ?X rdfs:seeAlso ?L }
      }
```

- “Normal” basic graph pattern doesn’t work here, returns only those `?X` with **both** a name **and** a `rdfs:seeAlso` link.
- `OPTIONAL` allows partial variable bindings in the solutions.

?N	?L
"Dan Brickley"	<http://danbri.org/foaf.rdf>
"Ruben Lara Hernandez"	<http://nets.ii.uam.es/r̃lara/foaf.rdf>
...	
"Thomas Eiter"	
...	

CONSTRUCT

CONSTRUCT queries in SPARQL allow to generate new RDF graphs from the results of a query, e.g.

> *"Create a graph which establishes 'foaf:knows relations for all persons who I have co-authored with according to DBLP."* (query7.sparql)

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
PREFIX dc:  <http://purl.org/dc/elements/1.1/>
PREFIX: <http://dblp.l3s.de/d2r/resource/authors/>
CONSTRUCT  { <http://polleres.net/foaf.rdf#me> foaf:knows ?Y }
WHERE { ?D dc:creator :Axel_Polleres;
        dc:creator ?Y . FILTER( ?Y != :Axel_Polleres )
      }
```

CONSTRUCT

CONSTRUCT queries in SPARQL allow to generate new RDF graphs from the results of a query, e.g.

> *"Create a graph which establishes 'foaf:knows relations for all persons who I have co-authored with according to DBLP."* (query7.sparql)

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
PREFIX dc:  <http://purl.org/dc/elements/1.1/>
PREFIX: <http://dblp.l3s.de/d2r/resource/authors/>
CONSTRUCT  { <http://polleres.net/foaf.rdf#me> foaf:knows ?Y }
WHERE { ?D dc:creator :Axel_Polleres;
        dc:creator ?Y . FILTER( ?Y != :Axel_Polleres )
      }
```

- "Output pattern" is a basic graph pattern
- similar to "views" in SQL
- May be viewed as a "rules language" itself.

ASK

ASK queries are "yes/no" queries without explicit output, e.g.

> *"Does Axel know one of the co-authors of <http://dblp.l3s.de/d2r/resource/publications/journals/tplp/Berners-LeeCKSH08>?"*

```
PREFIX rdfs:  <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
PREFIX dc:  <http://purl.org/dc/elements/1.1/>
ASK
FROM <http://polleres.net/foaf.rdf>
FROM <http://dblp.l3s.de/d2r/data/publications/journals/tplp/Berners-LeeCKSH08>
WHERE { <http://polleres.net/foaf.rdf#me> foaf:knows ?A .
        <http://dblp.l3s.de/d2r/resource/publications/journals/tplp/Berners-LeeCKSH08>
                                         dc:creator ?A }
```

ASK

ASK queries are "yes/no" queries without explicit output, e.g.

"Does Axel know one of the co-authors of <http://dblp.l3s.de/d2r/resource/publications/journals/tplp/Berners-LeeCKSH08>?"

```
PREFIX rdfs:  <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
PREFIX dc:  <http://purl.org/dc/elements/1.1/>
ASK
FROM <http://polleres.net/foaf.rdf>
FROM <http://dblp.l3s.de/d2r/data/publications/journals/tplp/Berners-LeeCKSH08>
WHERE { <http://polleres.net/foaf.rdf#me> foaf:knows ?A .
        <http://dblp.l3s.de/d2r/resource/publications/journals/tplp/Berners-LeeCKSH08>
                                           dc:creator ?A }
```

Interestingly, this query returns "no"... why? Because SPARQL doesn't know that

- <http://dblp.l3s.de/d2r/resource/authors/Jim_Hendler> = <http://www.cs.rpi.edu/ hendler/foaf.rdf#jhendler>

ASK

`ASK` queries are "yes/no" queries without explicit output, e.g.

> *"Does Axel know one of the co-authors of <http://dblp.l3s.de/d2r/resource/publications/journals/tplp/Berners-LeeCKSH08>?"*

```
PREFIX rdfs:  <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
PREFIX dc:  <http://purl.org/dc/elements/1.1/>
ASK
FROM <http://polleres.net/foaf.rdf>
FROM <http://dblp.l3s.de/d2r/data/publications/journals/tplp/Berners-LeeCKSH08>
WHERE { <http://polleres.net/foaf.rdf#me> foaf:knows ?A .
        <http://dblp.l3s.de/d2r/resource/publications/journals/tplp/Berners-LeeCKSH08>
                                                           dc:creator ?A }
```

Interestingly, this query returns "no"... why? Because SPARQL doesn't know that

- `<http://dblp.l3s.de/d2r/resource/authors/Jim_Hendler>` = `<http://www.cs.rpi.edu/ hendler/foaf.rdf#jhendler>`

although, in `http://polleres.net/foaf.rdf` there is a triple:

`<http://polleres.net/foaf.rdf#me> foaf:knows <http://www.cs.rpi.edu/∼hendler/foaf.rdf#jhendler>`

ASK

`ASK` queries are "yes/no" queries without explicit output, e.g.

> *"Does Axel know one of the co-authors of* `<http://dblp.l3s.de/d2r/resource/publications/journals/tplp/Berners-LeeCKSH08>`*?"*

```
PREFIX rdfs:  <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
PREFIX dc:  <http://purl.org/dc/elements/1.1/>
ASK
FROM <http://polleres.net/foaf.rdf>
FROM <http://dblp.l3s.de/d2r/data/publications/journals/tplp/Berners-LeeCKSH08>
WHERE { <http://polleres.net/foaf.rdf#me> foaf:knows ?A .
        <http://dblp.l3s.de/d2r/resource/publications/journals/tplp/Berners-LeeCKSH08>
                                                  dc:creator ?A }
```

Interestingly, this query returns "no"... why? Because SPARQL doesn't know that

- `<http://dblp.l3s.de/d2r/resource/authors/Jim_Hendler>` = `<http://www.cs.rpi.edu/ hendler/foaf.rdf#jhendler>`

although, in `http://polleres.net/foaf.rdf` there is a triple:

`<http://polleres.net/foaf.rdf#me> foaf:knows <http://www.cs.rpi.edu/∼hendler/foaf.rdf#jhendler>`

More on that later...

Exercise

Using the SPARQL interface to DBLP at `http://dblp.l3s.de/d2r/snorql/` write a query that outputs the following:

Task

Names of people who have published in TPLP or have co-authored with any of the authors of

`http://dblp.l3s.de/d2r/resource/publications/journals/tplp/Berners-LeeCKSH08`

- Can you do it in one query?
- Which of the constructs discussed do you need?

SPARQL summary

- We have only "scratched the surface" here
- Particularly, we didn't treat SPARQL1.1 ... more on that in later lectures:
- Extensions of SPARQL (updates (DELETE, INSERT, ...), aggregate functions (SUM, MAX, COUNT,...), etc.) currently being standardized
- Rigid investigation of SPARQL's semantics and complexity [Pérez *et al.*, 2006; Gutiérrez *et al.*, 2004]
- Peculiarities in SPARQL's semantics (multiset semantics, joins over unbound variables, etc. [Prud'hommeaux and Seaborne, 2008])
- SPARQL only does RDF graph pattern matching, what about ontologies?
  . . . Let's take a look at this next!

Unit Outline

1. Motivation – Aggregating Web Data

2. How can I publish data? RDF

3. How can I query that data? SPARQL

4. What does that data mean? Ontologies described in RDFS + OWL

5. What's next?

What does RDF data mean?

- *Ontologies* are formal descriptions of what the *vocabulary* used in an RDF document means.

What does RDF data mean?

- *Ontologies* are formal descriptions of what the *vocabulary* used in an RDF document means.
- By vocabulary, we mean here mostly:
  - *properties*,i.e., predicates
  - *classes*, i.e., objects of `rdf:type` triples
  - (*individuals*, i.e., concrete objects)

What does RDF data mean?

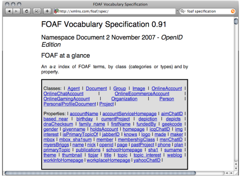
- *Ontologies* are formal descriptions of what the *vocabulary* used in an RDF document means.
- By vocabulary, we mean here mostly:
  - *properties*,i.e., predicates
  - *classes*, i.e., objects of `rdf:type` triples
  - (*individuals*, i.e., concrete objects)
- Ontologies describe ***relations*** among properties, classes and individuals (subclasses, subproperties, equivalence, domain, range, etc.)

What does RDF data mean?

- *Ontologies* are formal descriptions of what the *vocabulary* used in an RDF document means.
- By vocabulary, we mean here mostly:
  - *properties*,i.e., predicates
  - *classes*, i.e., objects of `rdf:type` triples
  - (*individuals*, i.e., concrete objects)
- Ontologies describe ***relations*** among properties, classes and individuals (subclasses, subproperties, equivalence, domain, range, etc.)
- The W3C has published two standards to describe ontologies, namely *RDF Schema (RDFS)* [Brickley and Guha (eds.), 2004] and the *Web Ontology language (OWL)* [W3C OWL 2 Working Group, 2012]
  - RDFS . . . simple schema language with minimal expressivity, mostly expressible in simple forward chaining inference rules (*Horn Rules*)
  - OWL . . . higher expressivity, foundations in *Description Logics*
  - both RDFS and OWL ontologies are RDF graphs themselves, i.e., OWL and RDFS provide "an RDF vocabulary to describe RDF vocabularies"

Example Vocabulary 1 – The FOAF ontology:

- **Properties:** `name`, `knows`, `homepage`, `primaryTopic` etc.
- **Classes:** `Person`, `Agent`, `Document`, `Organisation`, etc.
- **Relations:** e.g.
  - *Each* `Person` *is a* `Agent` (subclass)



FOAF Vocabulary Specification 0.91

Namespace Document 2 November 2007 - *OpenID Edition*

FOAF at a glance

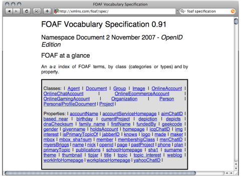
An a-z index of FOAF terms, by class (categories or types) and by property.

Classes: | Agent | Document | Group | Image | OnlineAccount | OnlineChatAccount | OnlineEcommerceAccount | OnlineGamingAccount | Organization | Person | PersonalProfileDocument | Project |

Properties: | accountName | accountServiceHomepage | aimChatID | based_near | birthday | currentProject | depiction | depicts | dnaChecksum | family_name | firstName | fundedBy | geekcode | gender | givenname | holdsAccount | homepage | icqChatID | img | interest | isPrimaryTopicOf | jabberID | knows | logo | made | maker | mbox | mbox_sha1sum | member | membershipClass | msnChatID | myersBriggs | name | nick | openid | page | pastProject | phone | plan | primaryTopic | publications | schoolHomepage | sha1 | surname | theme | thumbnail | tipjar | title | topic | topic_interest | weblog | workInfoHomepage | workplaceHomepage | yahooChatID |

Example Vocabulary 1 – The FOAF ontology:

- **Properties:** `name`, `knows`, `homepage`, `primaryTopic` etc.
- **Classes:** `Person`, `Agent`, `Document`, `Organisation`, etc.
- **Relations:** e.g.
  - *Each* `Person` *is a* `Agent` (subclass)
  - *The* `img` *property is more specific than* `depiction` (subproperty)



FOAF Vocabulary Specification 0.91

Namespace Document 2 November 2007 - *OpenID Edition*

FOAF at a glance

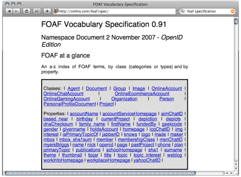
An a-z index of FOAF terms, by class (categories or types) and by property.

Classes: | Agent | Document | Group | Image | OnlineAccount | OnlineChatAccount | OnlineEcommerceAccount | OnlineGamingAccount | Organization | Person | PersonalProfileDocument | Project |

Properties: | accountName | accountServiceHomepage | aimChatID | based_near | birthday | currentProject | depiction | depicts | dnaChecksum | family_name | firstName | fundedBy | geekcode | gender | givenname | holdsAccount | homepage | icqChatID | img | interest | isPrimaryTopicOf | jabberID | knows | logo | made | maker | mbox | mbox_sha1sum | member | membershipClass | msnChatID | myersBriggs | name | nick | openid | page | pastProject | phone | plan | primaryTopic | publications | schoolHomepage | sha1 | surname | theme | thumbnail | tipjar | title | topic | topic_interest | weblog | workInfoHomepage | workplaceHomepage | yahooChatID |

Example Vocabulary 1 – The FOAF ontology:

- **Properties:** `name`, `knows`, `homepage`, `primaryTopic` etc.
- **Classes:** `Person`, `Agent`, `Document`, `Organisation`, etc.
- **Relations:** e.g.
    - *Each* `Person` *is a* `Agent` (subclass)
    - *The* `img` *property is more specific than* `depiction` (subproperty)
    - `img` *is a relation between* `Persons` *and* `Imgages` (domain/range)



FOAF Vocabulary Specification 0.91

Namespace Document 2 November 2007 - *OpenID Edition*

FOAF at a glance

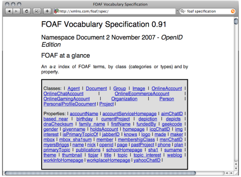
An a-z index of FOAF terms, by class (categories or types) and by property.

Classes: | Agent | Document | Group | Image | OnlineAccount | OnlineChatAccount | OnlineEcommerceAccount | OnlineGamingAccount | Organization | Person | PersonalProfileDocument | Project |

Properties: | accountName | accountServiceHomepage | aimChatID | based_near | birthday | currentProject | depiction | depicts | dnaChecksum | family_name | firstName | fundedBy | geekcode | gender | givenname | holdsAccount | homepage | icqChatID | img | interest | isPrimaryTopicOf | jabberID | knows | logo | made | maker | mbox | mbox_sha1sum | member | membershipClass | msnChatID | myersBriggs | name | nick | openid | page | pastProject | phone | plan | primaryTopic | publications | schoolHomepage | sha1 | surname | theme | thumbnail | tipjar | title | topic | topic_interest | weblog | workInfoHomepage | workplaceHomepage | yahooChatID |

Example Vocabulary 1 – The FOAF ontology:

- **Properties:** `name`, `knows`, `homepage`, `primaryTopic` etc.
- **Classes:** `Person`, `Agent`, `Document`, `Organisation`, etc.
- **Relations:** e.g.
  - *Each `Person` is a `Agent`* (subclass)
  - *The `img` property is more specific than `depiction`* (subproperty)
  - *`img` is a relation between `Persons` and `Imgages`* (domain/range)
  - *`knows` is a relation between two `Persons`* (domain/range)



FOAF Vocabulary Specification 0.91

Namespace Document 2 November 2007 - *OpenID Edition*

FOAF at a glance

An a-z index of FOAF terms, by class (categories or types) and by property.

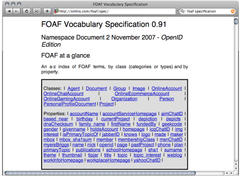
Classes: | Agent | Document | Group | Image | OnlineAccount | OnlineChatAccount | OnlineEcommerceAccount | OnlineGamingAccount | Organization | Person | PersonalProfileDocument | Project |

Properties: | accountName | accountServiceHomepage | aimChatID | based_near | birthday | currentProject | depiction | depicts | dnaChecksum | family_name | firstName | fundedBy | geekcode | gender | givenname | holdsAccount | homepage | icqChatID | img | interest | isPrimaryTopicOf | jabberID | knows | logo | made | maker | mbox | mbox_sha1sum | member | membershipClass | msnChatID | myersBriggs | name | nick | openid | page | pastProject | phone | plan | primaryTopic | publications | schoolHomepage | sha1 | surname | theme | thumbnail | tipjar | title | topic | topic_interest | weblog | workInfoHomepage | workplaceHomepage | yahooChatID |

Example Vocabulary 1 – The FOAF ontology:

- **Properties:** name, knows, homepage, primaryTopic etc.
- **Classes:** Person, Agent, Document, Organisation, etc.
- **Relations:** e.g.
  - *Each Person is a Agent* (subclass)
  - *The img property is more specific than depiction* (subproperty)
  - *img is a relation between Persons and Imgages* (domain/range)
  - *knows is a relation between two Persons* (domain/range)
  - *homepage denotes **unique** homepage of an Agent* (uniquely identifying property)

⋮



FOAF Vocabulary Specification 0.91

Namespace Document 2 November 2007 - *OpenID Edition*

FOAF at a glance

An a-z index of FOAF terms, by class (categories or types) and by property.

Classes: | Agent | Document | Group | Image | OnlineAccount | OnlineChatAccount | OnlineEcommerceAccount | OnlineGamingAccount | Organization | Person | PersonalProfileDocument | Project |

Properties: | accountName | accountServiceHomepage | aimChatID | based_near | birthday | currentProject | depiction | depicts | dnaChecksum | family_name | firstName | fundedBy | geekcode | gender | givenname | holdsAccount | homepage | icqChatID | img | interest | isPrimaryTopicOf | jabberID | knows | logo | made | maker | mbox | mbox_sha1sum | member | membershipClass | msnChatID | myersBriggs | name | nick | openid | page | pastProject | phone | plan | primaryTopic | publications | schoolHomepage | sha1 | surname | theme | thumbnail | tipjar | title | topic | topic_interest | weblog | workInfoHomepage | workplaceHomepage | yahooChatID |

Examples 2 – A simple ontology about reviewers:

- **Properties:** `title`, `isAuthorOf`, `publishedIn`, etc.
- **Classes:** `Senior`, `Paper`, `Publication`, etc.
- **Relations:**
  - *A* ***Publication*** *is a* ***Paper*** *which has been* ***published*** (subclass + existential condition on property)

[4] reuse of external ontologies!

Examples 2 – A simple ontology about reviewers:

- **Properties:** `title`, `isAuthorOf`, `publishedIn`, etc.
- **Classes:** `Senior`, `Paper`, `Publication`, etc.
- **Relations:**
  - *A `Publication` is a `Paper` which has been `published`* (subclass + existential condition on property)
  - *`isAuthorOf` is the opposite of Dublin Core's `dc:creator` Property*[4]

[4] reuse of external ontologies!

Examples 2 – A simple ontology about reviewers:

- **Properties:** `title`, `isAuthorOf`, `publishedIn`, etc.
- **Classes:** `Senior`, `Paper`, `Publication`, etc.
- **Relations:**
  - *A `Publication` is a `Paper` which has been `published`* (subclass + existential condition on property)
  - *`isAuthorOf` is the opposite of Dublin Core's `dc:creator` Property*[4]
  - *A `Senior` researcher is a `foaf:Person` who `isAuthorOf` 10+ `Publications`* (subclass + condition on cardinality)

[4] reuse of external ontologies!

Examples 2 – A simple ontology about reviewers:

- **Properties:** `title`, `isAuthorOf`, `publishedIn`, etc.
- **Classes:** `Senior`, `Paper`, `Publication`, etc.
- **Relations:**
  - *A `Publication` is a `Paper` which has been `published`* (subclass + existential condition on property)
  - *`isAuthorOf` is the opposite of Dublin Core's `dc:creator` Property*[4]
  - *A `Senior` researcher is a `foaf:Person` who `isAuthorOf` 10+ `Publications`* (subclass + condition on cardinality)
  - *Each item can be `publishedIn` at most one venue* (functional property)

⋮

[4] reuse of external ontologies!

RDF(S) vocabulary: RDF and RDFS themselves are vocabularies!

- **Properties:** `rdf:type`, `rdfs:domain`, `rdfs:range`, `rdfs:subClassOf`, `rdfs:subPropertyOf`, `rdf:first`, `rdf:rest` etc.
- **Classes:** `rdf:XMLLiteral`, `rdfs:Literal`, `rdfs:Resource`, `rdf:Property`, `rdfs:Class`, `rdf:List`, etc.
- **Relations:**

RDF(S) vocabulary: RDF and RDFS themselves are vocabularies!

- **Properties:** `rdf:type`, `rdfs:domain`, `rdfs:range`, `rdfs:subClassOf`, `rdfs:subPropertyOf`, `rdf:first`, `rdf:rest` etc.
- **Classes:** `rdf:XMLLiteral`, `rdfs:Literal`, `rdfs:Resource`, `rdf:Property`, `rdfs:Class`, `rdf:List`, etc.
- **Relations:**The semantics of the RDFs vocabulary is defined in [Hayes, 2004]; it is a "meta vocabulary" used to define the semantics of other vocabularies

The Semantics of RDF graphs:

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
<http://www.mat.unical.it/~ianni/foaf.rdf> a foaf:PersonalProfileDocument.
<http://www.mat.unical.it/~ianni/foaf.rdf> foaf:maker _:me .
<http://www.mat.unical.it/~ianni/foaf.rdf> foaf:primaryTopic _:me .
:me a foaf:Person .
:me foaf:name "Giovambattista Ianni" .
:me foaf:homepage <http://www.gibbi.com> .
:me foaf:phone <tel:+39-0984-496430> .
:me foaf:knows [ a foaf:Person ;
                 foaf:name "Wolfgang Faber" ;
                 rdfs:seeAlso <http://www.kr.tuwien.ac.at/staff/faber/foaf.rdf>].
:me foaf:knows [ a foaf:Person .
                 foaf:name "Axel Polleres" ;
                 rdfs:seeAlso <http://www.polleres.net/foaf.rdf>].
:me foaf:knows [ a foaf:Person .
                 foaf:name "Thomas Eiter" ] .
:me foaf:knows [ a foaf:Person .
                 foaf:name "Alessandra Martello" ] .
```

The Semantics of RDF graphs:

As we will see in the next Units, each RDF graph can “roughly” be viewed as a first-order formula:

$$
\begin{array}{l}
\exists \mathit{b1}, \mathit{b2}, \mathit{b3}, \mathit{b4}\\
(\mathit{triple}(\texttt{foaf.rdf}, \texttt{rdf:type}, \texttt{PersonalProfileDocument})\\
\ \wedge\ \mathit{triple}(\texttt{foaf.rdf}, \texttt{maker}, \mathit{me})\\
\ \wedge\ \mathit{triple}(\texttt{foaf.rdf}, \texttt{primaryTopic}, \mathit{me})\\
\ \wedge\ \mathit{triple}(\mathit{me}, \texttt{rdf:type}, \texttt{Person})\\
\ \wedge\ \mathit{triple}(\mathit{me}, \texttt{name}, \texttt{"Giovambattista Ianni"})\\
\ \wedge\ \mathit{triple}(\mathit{me}, \texttt{homepage}, \texttt{http://www.gibbi.com})\\
\ \wedge\ \mathit{triple}(\mathit{me}, \texttt{phone}, \texttt{tel:+39-0984-496430})\\
\ \wedge\ \mathit{triple}(\mathit{me}, \texttt{knows}, \mathit{b2}) \wedge\ \mathit{triple}(\mathit{b1}, \texttt{rdf:type}, \texttt{Person})\\
\ \wedge\ \mathit{triple}(\mathit{b1}, \texttt{name}, \texttt{"Wolfgang Faber"})\\
\ \wedge\ \mathit{triple}(\mathit{b1}, \texttt{rdfs:seeAlso}, \texttt{http://www.kr.tuwien...})\\
\ \wedge\ \mathit{triple}(\mathit{me}, \texttt{knows}, \mathit{b1}) \wedge\ \mathit{triple}(\mathit{b1}, \texttt{rdf:type}, \texttt{Person})\\
\ \wedge\ \mathit{triple}(\mathit{b2}, \texttt{name}, \texttt{"Axel Polleres"})\\
\ \wedge\ \mathit{triple}(\mathit{b2}, \texttt{rdfs:seeAlso}, \texttt{http://www.polleres...})\\
\ \wedge\ \mathit{triple}(\mathit{me}, \texttt{knows}, \mathit{b3}) \wedge\ \mathit{triple}(\mathit{b1}, \texttt{rdf:type}, \texttt{Person})\\
\ \wedge\ \mathit{triple}(\mathit{b3}, \texttt{name}, \texttt{"Thomas Eiter"})\\
\ \wedge\ \mathit{triple}(\mathit{me}, \texttt{knows}, \mathit{b4}) \wedge\ \mathit{triple}(\mathit{b1}, \texttt{rdf:type}, \texttt{Person})\\
\ \wedge\ \mathit{triple}(\mathit{b4}, \texttt{name}, \texttt{"Alessandra Martello"}))
\end{array}
$$

The Semantics of RDF graphs:

Alternatively, especially the OWL favors unary/binary predicate representation:

$$\begin{array}{l}\exists me, b1, b2, b3, b4\ (\texttt{PersonalProfileDocument}(\texttt{foaf.rdf}) \\ \ \wedge \texttt{maker}(\texttt{foaf.rdf}, me) \\ \ \wedge \texttt{primaryTopic}(\texttt{foaf.rdf}, me) \\ \ \wedge \texttt{Person}(me) \wedge \cdots)\end{array}$$

- unary predicates for `rdf:type` predicates
- binary predicates for all other predicates

The Semantics of the RDFS vocabulary:

The formal semantics of RDF(S) [Hayes, 2004] is accompanied by a set of (informative) entailment rules . . . can be written down roughly as the following first-order formulas:

$$\forall S, P, O\ (\mathit{triple}(S, P, O) \supset \mathit{triple}(S, \mathtt{rdf{:}type}, \mathtt{rdfs{:}Resource}))$$

$$\forall S, P, O\ (\mathit{triple}(S, P, O) \supset \mathit{triple}(P, \mathtt{rdf{:}type}, \mathtt{rdf{:}Property}))$$

$$\forall S, P, O\ (\mathit{triple}(S, P, O) \supset \mathit{triple}(O, \mathtt{rdf{:}type}, \mathtt{rdfs{:}Resource}))$$

$$\forall S, P, O\ (\mathit{triple}(S, P, O) \wedge \mathit{triple}(P, \mathtt{rdfs{:}domain}, C) \supset \mathit{triple}(S, \mathtt{rdf{:}type}, C))$$

$$\forall S, P, O, C\ (\mathit{triple}(S, P, O) \wedge \mathit{triple}(P, \mathtt{rdfs{:}range}, C) \supset \mathit{triple}(O, \mathtt{rdf{:}type}, C))$$

$$\forall C\ (\mathit{triple}(C, \mathtt{rdf{:}type}, \mathtt{rdfs{:}Class}) \supset \mathit{triple}(C, \mathtt{rdfs{:}subClassOf}, \mathtt{rdfs{:}Resource}))$$

$$\begin{aligned}\forall C_1, C_2, C_3\ (&\mathit{triple}(C_1, \mathtt{rdfs{:}subClassOf}, C_2) \wedge \\ &\mathit{triple}(C_2, \mathtt{rdfs{:}subClassOf}, C_3) \supset \mathit{triple}(C_1, \mathtt{rdfs{:}subClassOf}, C_3))\end{aligned}$$

$$\forall S, C_1, C_2\ (\mathit{triple}(S, \mathtt{rdf{:}type}, C_1)\ \wedge \mathit{triple}(C_1, \mathtt{rdfs{:}subClassOf}, C_2) \supset \mathit{triple}(S, \mathtt{rdf{:}type}, C_2))$$

$$\forall S, C\ (\mathit{triple}(S, \mathtt{rdf{:}type}, C) \supset \mathit{triple}(C, \mathtt{rdf{:}type}, \mathtt{rdfs{:}Class}))$$

$$\forall C\ (\mathit{triple}(C, \mathtt{rdf{:}type}, \mathtt{rdfs{:}Class}) \supset \mathit{triple}(C, \mathtt{rdfs{:}subClassOf}, C))$$

$$\begin{aligned}\forall P_1, P_2, P_3\ (&\mathit{triple}(P_1, \mathtt{rdfs{:}subPropertyOf}, P_2) \wedge \\ &\mathit{triple}(P_2, \mathtt{rdfs{:}subPropertyOf}, P_3) \supset \mathit{triple}(P_1, \mathtt{rdfs{:}subPropertyOf}, P_3))\end{aligned}$$

$$\forall S, P_1, P_2, O\ (\mathit{triple}(S, P_1, O) \wedge \mathit{triple}(P_1, \mathtt{rdfs{:}subPropertyOf}, P_2) \supset \mathit{triple}(S, P_2, O))$$

$$\forall P\ (\mathit{triple}(P, \mathtt{rdf{:}type}, \mathtt{rdf{:}Property}) \supset \mathit{triple}(P, \mathtt{rdfs{:}subPropertyOf}, P))$$

plus the axiomatic triples from [Hayes, 2004, Sections 3.1 and 4.1].

The Semantics of the RDFS vocabulary:

The formal semantics of RDF(S) [Hayes, 2004] is accompanied by a set of (informative) entailment rules . . . can be written down roughly as the following first-order formulas:

$$\forall S, P, O\ (\mathit{triple}(S, P, O) \supset \mathit{triple}(S, \texttt{rdf:type}, \texttt{rdfs:Resource}))$$

$$\forall S, P, O\ (\mathit{triple}(S, P, O) \supset \mathit{triple}(P, \texttt{rdf:type}, \texttt{rdf:Property}))$$

$$\forall S, P, O\ (\mathit{triple}(S, P, O) \supset \mathit{triple}(O, \texttt{rdf:type}, \texttt{rdfs:Resource}))$$

$$\forall S, P, O\ (\mathit{triple}(S, P, O) \wedge \mathit{triple}(P, \texttt{rdfs:domain}, C) \supset \mathit{triple}(S, \texttt{rdf:type}, C))$$

$$\forall S, P, O, C\ (\mathit{triple}(S, P, O) \wedge \mathit{triple}(P, \texttt{rdfs:range}, C) \supset \mathit{triple}(O, \texttt{rdf:type}, C))$$

$$\forall C\ (\mathit{triple}(C, \texttt{rdf:type}, \texttt{rdfs:Class}) \supset \mathit{triple}(C, \texttt{rdfs:subClassOf}, \texttt{rdfs:Resource}))$$

$$\begin{aligned}\forall C_1, C_2, C_3\ (&\mathit{triple}(C_1, \texttt{rdfs:subClassOf}, C_2) \wedge \\ &\quad \mathit{triple}(C_2, \texttt{rdfs:subClassOf}, C_3) \supset \mathit{triple}(C_1, \texttt{rdfs:subClassOf}, C_3))\end{aligned}$$

$$\forall S, C_1, C_2\ (\mathit{triple}(S, \texttt{rdf:type}, C_1) \wedge \mathit{triple}(C_1, \texttt{rdfs:subClassOf}, C_2) \supset \mathit{triple}(S, \texttt{rdf:type}, C_2))$$

$$\forall S, C\ (\mathit{triple}(S, \texttt{rdf:type}, C) \supset \mathit{triple}(C, \texttt{rdf:type}, \texttt{rdfs:Class}))$$

$$\forall C\ (\mathit{triple}(C, \texttt{rdf:type}, \texttt{rdfs:Class}) \supset \mathit{triple}(C, \texttt{rdfs:subClassOf}, C))$$

$$\begin{aligned}\forall P_1, P_2, P_3\ (&\mathit{triple}(P_1, \texttt{rdfs:subPropertyOf}, P_2) \wedge \\ &\quad \mathit{triple}(P_2, \texttt{rdfs:subPropertyOf}, P_3) \supset \mathit{triple}(P_1, \texttt{rdfs:subPropertyOf}, P_3))\end{aligned}$$

$$\forall S, P_1, P_2, O\ (\mathit{triple}(S, P_1, O) \wedge \mathit{triple}(P_1, \texttt{rdfs:subPropertyOf}, P_2) \supset \mathit{triple}(S, P_2, O))$$

$$\forall P\ (\mathit{triple}(P, \texttt{rdf:type}, \texttt{rdf:Property}) \supset \mathit{triple}(P, \texttt{rdfs:subPropertyOf}, P))$$

---

plus the axiomatic triples from [Hayes, 2004, Sections 3.1 and 4.1].

The Semantics of the RDFS vocabulary:

Note 1:
All those rules were Datalog expressible, i.e. Horn rules, no negation, no function symbols.

The Semantics of the RDFS vocabulary:

Note 1:
All those rules were Datalog expressible, i.e. Horn rules, no negation, no function symbols.

Note 2:
Writing entailment rules in unary/binary representation would yield second order, e.g.:

$$\forall S, C_1, C_2\ (\mathit{triple}(S, \texttt{rdf:type}, C_1)\ \wedge \mathit{triple}(C_1, \texttt{rdfs:subClassOf}, C_2) \supset \mathit{triple}(S, \texttt{rdf:type}, C_2))$$

The Semantics of the RDFS vocabulary:

Note 1:
All those rules were Datalog expressible, i.e. Horn rules, no negation, no function symbols.

Note 2:
Writing entailment rules in unary/binary representation would yield second order, e.g.:

$$\forall S, C_1, C_2\ (C_1(S)\ \wedge \texttt{rdfs:subClassOf}(C_1, C_2) \supset C_2(S))$$

RDFS Semantics Example: The FOAF ontology

FOAF Ontology:

- *Each Person is a Agent* (subclass)
- *The img property is more specific than depiction* (subproperty)
- *img is a relation between Persons and Imgages* (domain/range)
- *knows is a relation between two Persons* (domain/range)
- *homepage denotes* ***unique*** *homepage of an Agent* (uniquely identifying property)

⋮

RDFS: Semantics

⋮

$$\forall S, C_1, C_2\ (triple(S, \texttt{rdf:type}, C_1)\ \land triple(C_1, \texttt{rdfs:subClassOf}, C_2) \supset triple(S, \texttt{rdf:type}, C_2))$$

⋮

Data:

```
:me rdf:type foaf:Person .
```

RDFS Semantics Example: The FOAF ontology

FOAF Ontology in RDF:

- `foaf:Person rdfs:subClassOf foaf:Agent .`
- `foaf:img rdfs:subPropertyOf foaf:depiction .`
- `foaf:img rdfs:domain foaf:Person ; rdfs:range foaf:Image .`
- `foaf:knows rdfs:domain foaf:Person ; rdfs:range foaf:Person .`
- ???

$\vdots$

RDFS: Semantics

$\vdots$

$$\forall S, C_1, C_2\ (triple(S, \texttt{rdf:type}, C_1)\ \wedge triple(C_1, \texttt{rdfs:subClassOf}, C_2) \supset triple(S, \texttt{rdf:type}, C_2))$$

$\vdots$

Data:

```
:me rdf:type foaf:Person .
:me rdf:type foaf:Agent .
```

RDFS Semantics Example: The FOAF ontology

FOAF Ontology in RDF:

- `foaf:Person rdfs:subClassOf foaf:Agent .`
- `foaf:img rdfs:subPropertyOf foaf:depiction .`
- `foaf:img rdfs:domain foaf:Person ; rdfs:range foaf:Image .`
- `foaf:knows rdfs:domain foaf:Person ; rdfs:range foaf:Person .`
- ???

$\vdots$

RDFS: Semantics

$\vdots$

$$\forall S, C_1, C_2\ (triple(S, \texttt{rdf:type}, C_1)\ \wedge triple(C_1, \texttt{rdfs:subClassOf}, C_2) \supset triple(S, \texttt{rdf:type}, C_2))$$

$\vdots$

Data:

```
:me rdf:type foaf:Person .
:me rdf:type foaf:Agent .
```

The OWL vocabulary:

- *foaf:homepage denotes **unique** homepage of an Agent* (uniquely identifying property)

For expressing this, we need more than the RDFS vocabulary. OWL is again an RDF vocabulary, extending RDF(S), fixed semantics that adds more expressivity on top of RDFS:

- **Properties:** `owl:sameAs`, `owl:differentFrom`, `owl:inverseOf`, `owl:onProperty`, `owl:allValuesFrom`, `owl:someValuesFrom`, `owl:minCardinality`, `owl:maxCardinality` etc.
- **Classes:** `owl:Restriction`, `owl:DatatypeProperty`, `owl:ObjectProperty`, `owl:FunctionalProperty`, `owl:InverseFuncionalProperty`, `owl:SymmetricProperty` etc.
- **Relations:** The semantics of OWL is defined in
  - in terms of its RDF reading (RDF-based-semantics), and

[5] direct semantics puts some restrictions on the use of the OWL and RDF vocabulary, fragment sometimes called OWL DL

The OWL vocabulary:

- *foaf:homepage denotes **unique** homepage of an Agent* (uniquely identifying property)

For expressing this, we need more than the RDFS vocabulary. OWL is again an RDF vocabulary, extending RDF(S), fixed semantics that adds more expressivity on top of RDFS:

- **Properties:** `owl:sameAs`, `owl:differentFrom`, `owl:inverseOf`, `owl:onProperty`, `owl:allValuesFrom`, `owl:someValuesFrom`, `owl:minCardinality`, `owl:maxCardinality` etc.
- **Classes:** `owl:Restriction`, `owl:DatatypeProperty`, `owl:ObjectProperty`, `owl:FunctionalProperty`, `owl:InverseFuncionalProperty`, `owl:SymmetricProperty` etc.
- **Relations:** The semantics of OWL is defined in
  - in terms of its RDF reading (RDF-based-semantics), and
  - in terms of its Description Logics reading (direct semantics)[5]

[5] direct semantics puts some restrictions on the use of the OWL and RDF vocabulary, fragment sometimes called OWL DL

The Semantics of the OWL vocabulary (DL reading):

Description Logics:

- syntactic variant of first-order logic with equality
- especially tailored for talking about concepts (classes, sets) and roles (properties)
- dedicated symbols for class membership and subclass/subproperty relation:

```
foaf:Person rdfs:subClassOf foaf:Agent
:me rdf:type  foaf:Person
```

$Person \sqsubseteq Agent$

$me \in Person$

OWL DL in one slide

Expressing property characteristics:

OWL property axioms as RDF triples	DL syntax	FOL short representation
P rdfs:domain C .	$\top \sqsubseteq \forall P^{-}.C$	$\forall x, y.P(x, y) \supset C(x)$
P rdfs:range C .	$\top \sqsubseteq \forall P.C$	$\forall x, y.P(x, y) \supset C(y)$
P owl:inverseOf P_0 .	$P \equiv P_0^{-}$	$\forall x, y.P(x, y) \equiv P_0(y, x)$
P rdf:type owl:SymmetricProperty.	$P \equiv P^{-}$	$\forall x, y.P(x, y) \equiv P(y, x)$
P rdf:type owl:FunctionalProperty.	$\top \sqsubseteq \leqslant 1P$	$\forall x, y, z.P(x, y) \wedge P(x, z) \supset y = z$
P rdf:type owl:InverseFunctionalProperty.	$\top \sqsubseteq \leqslant 1P^{-}$	$\forall x, y, z.P(x, y) \wedge P(z, y) \supset x = z$
P rdf:type owl:TransitiveProperty.	$P^{+} \sqsubseteq P$	$\forall x, y, z.P(x, y) \wedge P(y, z) \supset P(x, z)$

OWL DL in one slide

Expressing property characteristics:

OWL property axioms as RDF triples	**DL syntax**	**FOL short representation**
P rdfs:domain C .	$\top \sqsubseteq \forall P^-.C$	$\forall x, y.P(x, y) \supset C(x)$
P rdfs:range C .	$\top \sqsubseteq \forall P.C$	$\forall x, y.P(x, y) \supset C(y)$
P owl:inverseOf P_0 .	$P \equiv P_0^-$	$\forall x, y.P(x, y) \equiv P_0(y, x)$
P rdf:type owl:SymmetricProperty.	$P \equiv P^-$	$\forall x, y.P(x, y) \equiv P(y, x)$
P rdf:type owl:FunctionalProperty.	$\top \sqsubseteq\ \leqslant 1P$	$\forall x, y, z.P(x, y) \wedge P(x, z) \supset y = z$
P rdf:type owl:InverseFunctionalProperty.	$\top \sqsubseteq\ \leqslant 1P^-$	$\forall x, y, z.P(x, y) \wedge P(z, y) \supset x = z$
P rdf:type owl:TransitiveProperty.	$P^+ \sqsubseteq P$	$\forall x, y, z.P(x, y) \wedge P(y, z) \supset P(x, z)$

Expressing complex class descriptions:

OWL complex class descriptions*	**DL syntax**	**FOL short representation**
owl:Thing	$\top$	$x = x$
owl:Nothing	$\bot$	$\neg x = x$
owl:intersectionOf $(C_1 \dots C_n)$	$C_1 \sqcap \dots \sqcap C_n$	$C_1(x) \wedge \dots \wedge C_n(x)$
owl:unionOf $(C_1 \dots C_n)$	$C_1 \sqcup \dots \sqcup C_n$	$C_1(x) \vee \dots \vee C_n(x)$
owl:complementOf (C)	$\neg C$	$\neg C(x)$
owl:oneOf $(o_1 \dots o_n)$	$\{o_1, \dots, o_n\}$	$x = o_1 \vee \dots \vee x = o_n$
owl:restriction $(P$ owl:someValuesFrom $(C))$	$\exists P.C$	$\exists y.P(x, y) \wedge C(y)$
owl:restriction $(P$ owl:allValuesFrom $(C))$	$\forall P.C$	$\forall y.P(x, y) \supset C(y)$
owl:restriction $(P$ owl:value $(o))$	$\exists P.\{o\}$	$P(x, o)$
owl:restriction $(P$ owl:minCardinality $(n))$	$\geqslant nP$	$\exists y_1 \dots y_n . \bigwedge_{k=1}^{n} P(x, y_k) \wedge \bigwedge_{i<j} y_j \neq y_j$
owl:restriction $(P$ owl:maxCardinality $(n))$	$\leqslant nP$	$\forall y_1 \dots y_{n+1} . \bigwedge_{k=1}^{n+1} P(x, y_k) \supset \bigvee_{i<j} y_i = y_j$

*For reasons of legibility, we use a variant of the OWL abstract syntax [Patel-Schneider *et al.*, 2004] in this table.

OWL DL in two slides: 2/2

Relating Class descriptions:

C_1 `rdfs:subClassOf` C_1	$C_1 \sqsubseteq C_2$
C_1 `owl:equivalentClass` C_2	$C_1 \equiv C_2$
C_1 `owl:disjointWith` C_2	$C_1 \sqcap C_2 \sqsubseteq \bot$

Relating individuals:

o_1 `owl:sameAs` o_1	$o_1 = o_2$
o_1 `owl:differentFrom` o_2	$o_1 \neq o_2$

OWL DL in two slides: 2/2

Relating Class descriptions:

C_1 `rdfs:subClassOf` C_1	$C_1 \sqsubseteq C_2$
C_1 `owl:equivalentClass` C_2	$C_1 \equiv C_2$
C_1 `owl:disjointWith` C_2	$C_1 \sqcap C_2 \sqsubseteq \bot$

Relating individuals:

o_1 `owl:sameAs` o_1	$o_1 = o_2$
o_1 `owl:differentFrom` o_2	$o_1 \neq o_2$

Examples:

```
<http://www.polleres.net/foaf.rdf#me> owl:sameAs
       <http://dblp.l3s.de/d2r/resource/authors/Axel_Polleres> .
<http://polleres.net/foaf.rdf#me> owl:differentFrom
       <http://www.mat.unical.it/~ianni/foaf.rdf#me> .
```

OWL DL in two slides: 2/2

Relating Class descriptions:

C_1 `rdfs:subClassOf` C_1	$C_1 \sqsubseteq C_2$
C_1 `owl:equivalentClass` C_2	$C_1 \equiv C_2$
C_1 `owl:disjointWith` C_2	$C_1 \sqcap C_2 \sqsubseteq \bot$

Relating individuals:

o_1 `owl:sameAs` o_1	$o_1 = o_2$
o_1 `owl:differentFrom` o_2	$o_1 \neq o_2$

Examples:

```
<http://www.polleres.net/foaf.rdf#me> owl:sameAs
       <http://dblp.l3s.de/d2r/resource/authors/Axel_Polleres> .
<http://polleres.net/foaf.rdf#me> owl:differentFrom
       <http://www.mat.unical.it/~ianni/foaf.rdf#me> .
```

Many more features in OWL2 (2009) . . . wouldn't have fit in 2 slides ;-)

OWL Example: The FOAF ontology

- *homepage denotes* ***unique*** *homepage of an Agent* (uniquely identifying property)

. . . in OWL/RDF syntax:

```
foaf:homepage rdf:type owl:InverseFunctionalProperty .
```

OWL Example: The FOAF ontology

- *homepage denotes* ***unique*** *homepage of an Agent* (uniquely identifying property)

. . . in OWL/RDF syntax:

```
foaf:homepage rdf:type owl:InverseFunctionalProperty .
```

. . . in DL syntax:

$$\top \sqsubseteq \leqslant 1 homepage^{-}$$

OWL Example: The FOAF ontology

- *`homepage` denotes* ***unique*** *homepage of an `Agent`* (uniquely identifying property)

. . . in OWL/RDF syntax:

```
foaf:homepage rdf:type owl:InverseFunctionalProperty .
```

. . . in DL syntax:

$$\top \sqsubseteq \leqslant 1 homepage^{-}$$

Example inference:

```
<http://www.polleres.net/foaf.rdf#me> foaf:homepage
        <http://www.polleres.net/> .
<http://dblp.l3s.de/d2r/resource/authors/Axel_Polleres> foaf:homepage
        <http://www.polleres.net/> .
```

OWL Example: The FOAF ontology

- *homepage denotes* ***unique*** *homepage of an Agent* (uniquely identifying property)

...in OWL/RDF syntax:

```
foaf:homepage rdf:type owl:InverseFunctionalProperty .
```

...in DL syntax:

$$\top \sqsubseteq \leqslant 1 homepage^{-}$$

Example inference:

```
<http://www.polleres.net/foaf.rdf#me> foaf:homepage
        <http://www.polleres.net/> .
<http://dblp.l3s.de/d2r/resource/authors/Axel_Polleres> foaf:homepage
        <http://www.polleres.net/> .
```

$$\models$$

```
<http://www.polleres.net/foaf.rdf#me> owl:sameAs
        <http://dblp.l3s.de/d2r/resource/authors/Axel_Polleres> .
```

Reasoning with Ontologies

Tools:

- Special purpose DL reasoners:
  Pellet [Sirin *et al.*, 2005], Racer [Haarslev and Möller, 2001], Fact++ [Tsarkov and Horrocks, 2006], Hermit [Motik *et al.*, 2007]
- General purpose FOL theorem provers:
  SNARK [Stickel *et al.*, online], SPASS [SPASS, online], Vampire [Riazanov and Voronkov, 2002]
- For special fragments of OWL [Motik *et al.*, 2012]:
  - **Rule/LP engines** (OWL RL)
  - **Relational databases** (OWL QL)

SPARQL & Ontologies

SPARQL on top of ontologies not trivial:

SPARQL & Ontologies

SPARQL on top of ontologies not trivial:

- Challenge 1: infinite RDFS/OWL inferences on a finite graph
- Challenge 2: blank nodes as existentials
- Challenge 3: SPARQL is SQL inspired (CWA), OWL/RDFS are (OWA):
  e.g., OPTIONAL patterns are non-monotonic, RDFS+OWL inference are both monotonic

SPARQL & Ontologies

SPARQL on top of ontologies not trivial:

- Challenge 1: infinite RDFS/OWL inferences on a finite graph
- Challenge 2: blank nodes as existentials
- Challenge 3: SPARQL is SQL inspired (CWA), OWL/RDFS are (OWA):
  e.g., OPTIONAL patterns are non-monotonic, RDFS+OWL inference are both monotonic

→ W3C's SPARQL1.1 WG has defined Entailment regimes for RDFS and OWL [Glimm *et al.*, 2013], stay tuned for later lectures.

Unit Outline

1. Motivation – Aggregating Web Data
2. How can I publish data? RDF
3. How can I query that data? SPARQL
4. What does that data mean? Ontologies described in RDFS + OWL
5. What's next?

Summary

- We should all have a rough idea about where to find RDF now.
- We should all have a rough idea about how to read RDF now.
- We should all have a rough idea about how to write RDF now
  → *Homework!*
- We should all have a rough idea of how to query RDF (SPARQL)
  → *Homework!*
- We should all have a rough idea of how the semantics of RDF vocabularies and data can be described (RDFS + OWL)

Details to come!

What's next?

- Details on the semantics of RDF+RDFS
- Details on the semantics of SPARQL+SPARQL1.1
- OWL2 and efficient reasoning for some fragments (particularly, OWL RL & OWL QL
- Linked Data
- Using SPARQL1.1 + RDFS + OWL on Linked Data
- Time allowed: applications.

- Ben Adida, Mark Birbeck, Shane McCarron, and Steven Pemberton (eds.).
  Rdfa in xhtml: Syntax and processing, October 2008.
  W3C Recommendation, available at http://www.w3.org/TR/rdfa-syntax/.

- Tags for identifying languages, September 2006.
  available at http://www.rfc-editor.org/rfc/bcp/bcp47.txt.

- Dave Beckett and Brian McBride (eds.).
  Rdf/xml syntax specification (revised), February 2004.
  W3C Recommendation, available at http://www.w3.org/TR/rdf-syntax-grammar/.

- Dave Beckett, Tim Berners-Lee, Eric Prud′hommeaux, and Gavin Carothers.
  Turtle - Terse RDF Triple Language, January 2008.
  W3CCandidarte Recommendation, http://www.w3.org/TR/turtle/.

- Tim Berners-Lee and Dan Connolly.
  Notation3 (N3): A readable RDF Syntax, January 2008.
  W3C Team Submission, http://www.w3.org/TeamSubmission/n3/.

- Uldis Bojārs, John G. Breslin, Diego Berrueta, Dan Brickley, Stefan Decker, Sergio Fernández, Christoph Görn, Andreas Harth, Tom Heath, Kingsley Idehen, Kjetil Kjernsmo, Alistair Miles, Alexandre Passant, Axel Polleres, Luis Polo, and Michael Sintek.
  SIOC Core Ontology Specification, June 2007.
  W3C member submission.

- Dan Brickley and R.V. Guha (eds.).
  RDF vocabulary description language 1.0: RDF Schema, February 2004.

W3C Recommendation, available at http://www.w3.org/TR/rdf-schema/.

Dan Brickley and Libby Miller.
FOAF Vocabulary Specification 0.91, November 2007.
http://xmlns.com/foaf/spec/.

Birte Glimm, Chimezie Ogbuji, Sandro Hawke, Ivan Herman, Bijan Parsia, Axel Polleres, and Andy Seaborne.
SPARQL 1.1 Entailment Regimes, March 2013.
W3C Recommendation, available at http://www.w3.org/TR/sparql11-entailment/.

Claudio Gutiérrez, Carlos A. Hurtado, and Alberto O. Mendelzon.
Foundations of semantic web databases.
In *PODS*, pages 95–106, 2004.

V. Haarslev and R. Möller.
RACER System Description.
In *Proceedings of the First International Joint Conference on Automated Reasoning* (*IJCAR 2001*), volume 2083 of *Lecture Notes in Computer Science* (*LNCS*), pages 701–705. Springer Verlag, 2001.

Steve Harris and Andy Seaborne.
SPARQL 1.1 Query Language, March 2013.
W3C Recommendation at http://www.w3.org/TR/sparql11-query/.

P. Hayes.
RDF semantics, 2004.
http://www.w3.org/TR/rdf-mt/.

Boris Motik, Rob Shearer, and Ian Horrocks.
Optimized Reasoning in Description Logics using Hypertableaux.
In Frank Pfenning, editor, *Proc. of the 21st Conference on Automated Deduction (CADE-21)*, volume 4603 of *LNAI*, pages 67–83, Bremen, Germany, July 17–20 2007. Springer.

Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten Lutz (eds.).
OWL 2 web ontology language profiles (second edition), December 2012.
W3C Recommendation, available at
`http://www.w3.org/TR/2012/REC-owl2-profiles-20121211/`.

Mikael Nilsson, Andy Powell, Pete Johnston, and Ambjörn Naeve.
Expressing dublin core metadata using the resource description framework (rdf), January 2008.
DCMI Recommendation.

Peter F. Patel-Schneider, Patrick Hayes, and Ian Horrocks.
OWL Web Ontology Language Semantics and Abstract Syntax, February 2004.
W3C Recommendation.

Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez.
Semantics and complexity of sparql.
In *International Semantic Web Conference (ISWC 2006)*, pages 30–43, 2006.

SPARQL Query Language for RDF, January 2008.
W3C Recommendation, available at
`http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/`.

Alexandre Riazanov and Andrei Voronkov.
The design and implementation of vampire.
AI Communications, 15(2-3):91–110, 2002.

Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz.
Pellet: A practical OWL-DL reasoner.
Technical Report 68, UMIACS, University of Maryland, 2005.

Spass: An automated theorem prover for first-order logic with equality, online.
Available at `http://www.spass-prover.org/`, last checked 04/2013.

Mark E. Stickel, Richard J. Waldinger, and Vinay K. Chaudhr.
A guide to snark, online.
Available at `http://www.ai.sri.com/snark/tutorial/tutorial.html`, last checked 04/2013.

Dmitry Tsarkov and Ian Horrocks.
Fact++ Description Logic Reasoner: System Description.
In *Proceedings of the Third International Joint Conference on Automated Reasoning (IJCAR 2006)*, 2006.

W3C OWL 2 Working Group.
OWL 2 Web Ontology Language Document Overview (Second Edition), December 2012.
W3C Recommendation, available at `http://www.w3.org/TR/owl2-overview/`.