Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Semantic Web Technologies: RDF + RDFS

The limits of my language are the limits of my world.
Ludwig Wittgenstein

- RDF Language
- RDF Schema
- RDF Expressiveness & Semantics
- RDF Programming

Introduction

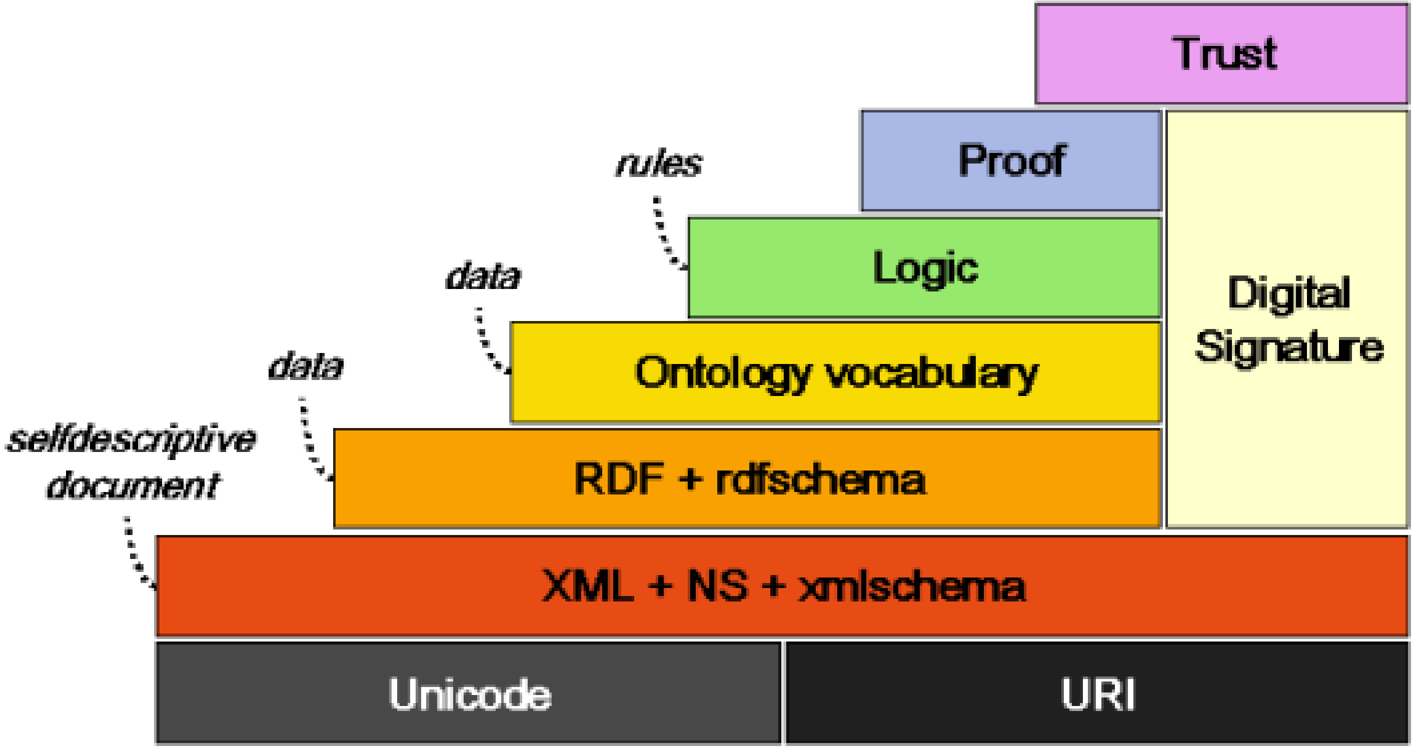
*“The **Semantic Web** provides a common framework that allows **data** to be shared and reused across application, enterprise, and community boundaries. […] It is based on the Resource Description Framework (RDF), which integrates a variety of applications using XML for syntax and URIs for naming.”*

http://www.w3.org/2001/sw/

- Is RDF just a complex way to write metadata that you can do with simple namespaces?
- Is RDF far beyond from any practical value and real world needs?

Semantic Web Layers

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de



Source: http://www.w3.org/2001/12/semweb-fin/w3csw

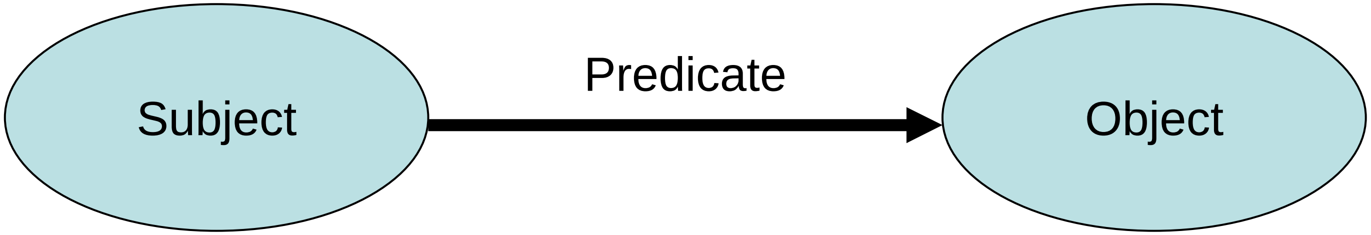
RDF - Resource Description Framework

- Model for the description of resources
- Common framework for representing Meta information
  - Applicable without assumptions on document structure/encoding
- Provides machine processable information
  - Provision of unambiguous syntax expressions
- Relies on the concept of URIs for identifying things

- RDF does not define a domain-specific vocabulary (but interoperates e.g. with XMLNS)
- RDF is not bound to a certain serialization syntax (but is commonly serialized with XML)

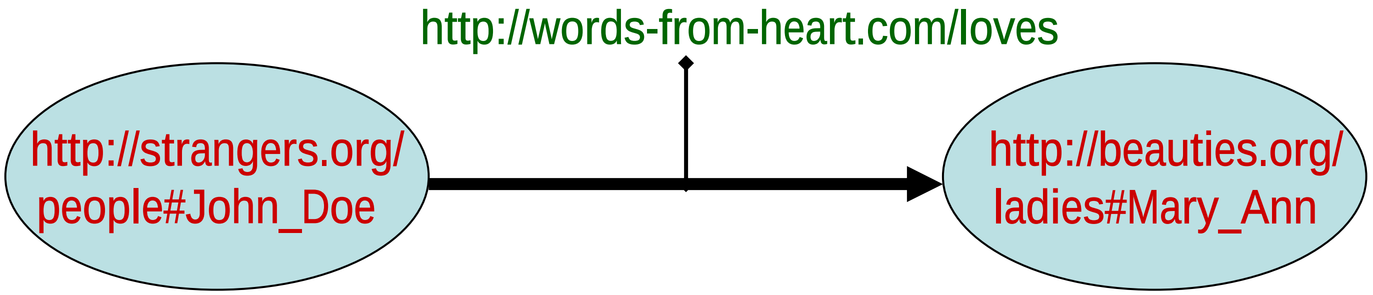
Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

RDF Model - Concepts



- RDF statement is represented by a triple of:
  - Subject → RDF URI reference or blank node
  - Predicate (property) → RDF URI reference
  - Object → RDF URI reference, literal or blank node
- Directed named graph of RDF triples:
  - Nodes (Subject / Object)
  - Arcs (Predicate)
- Meaning of a RDF graph is a conjunction (logical and)

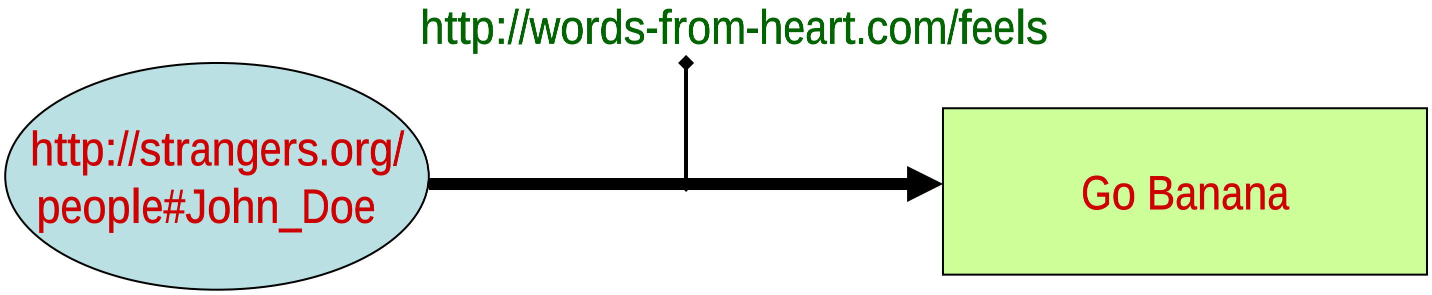
RDF Examples



http://words-from-heart.com/loves

http://strangers.org/
people#John_Doe

http://beauties.org/
ladies#Mary_Ann



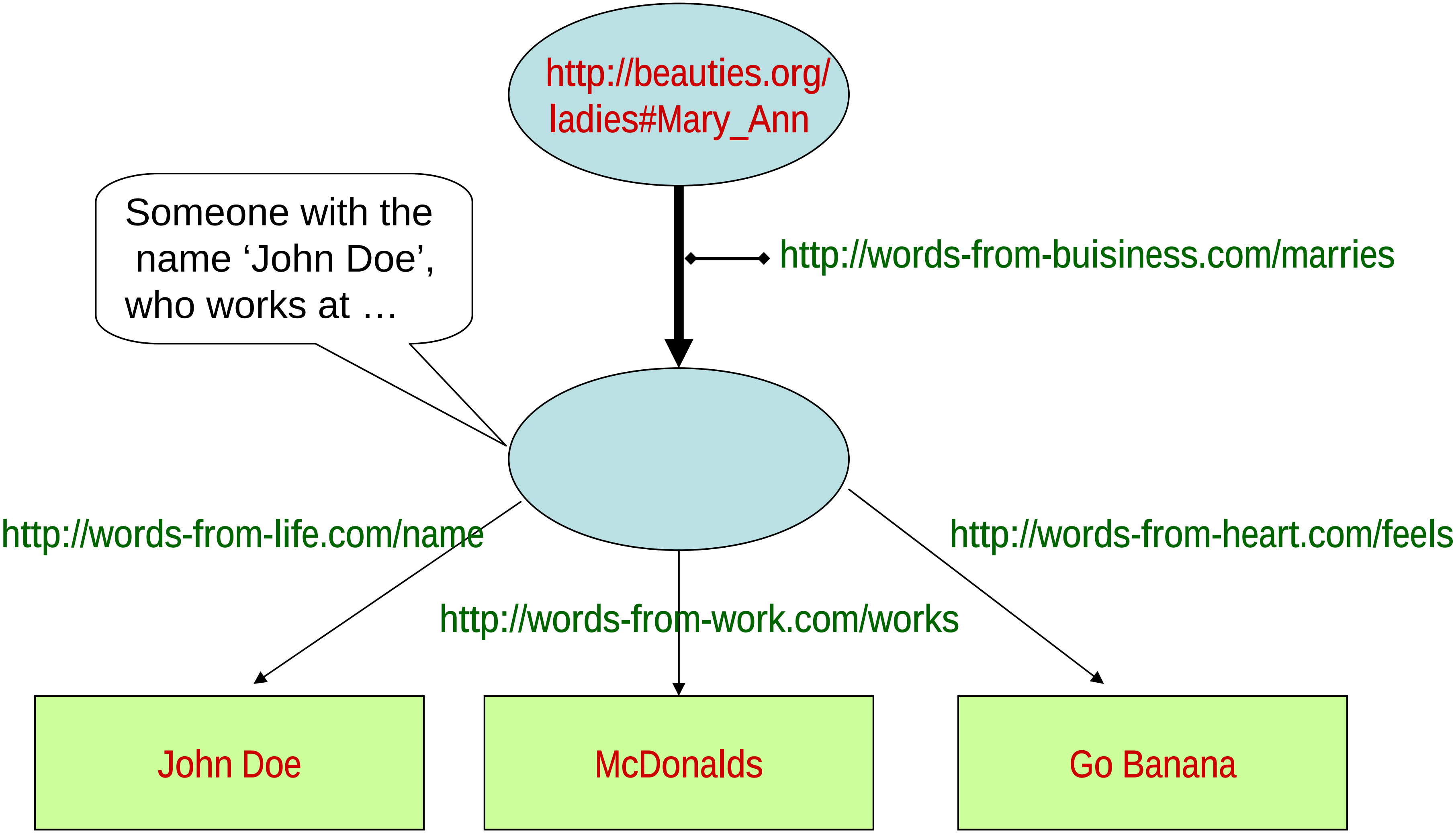
http://words-from-heart.com/feels

http://strangers.org/
people#John_Doe

Go Banana

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

RDF Examples: Blank Node

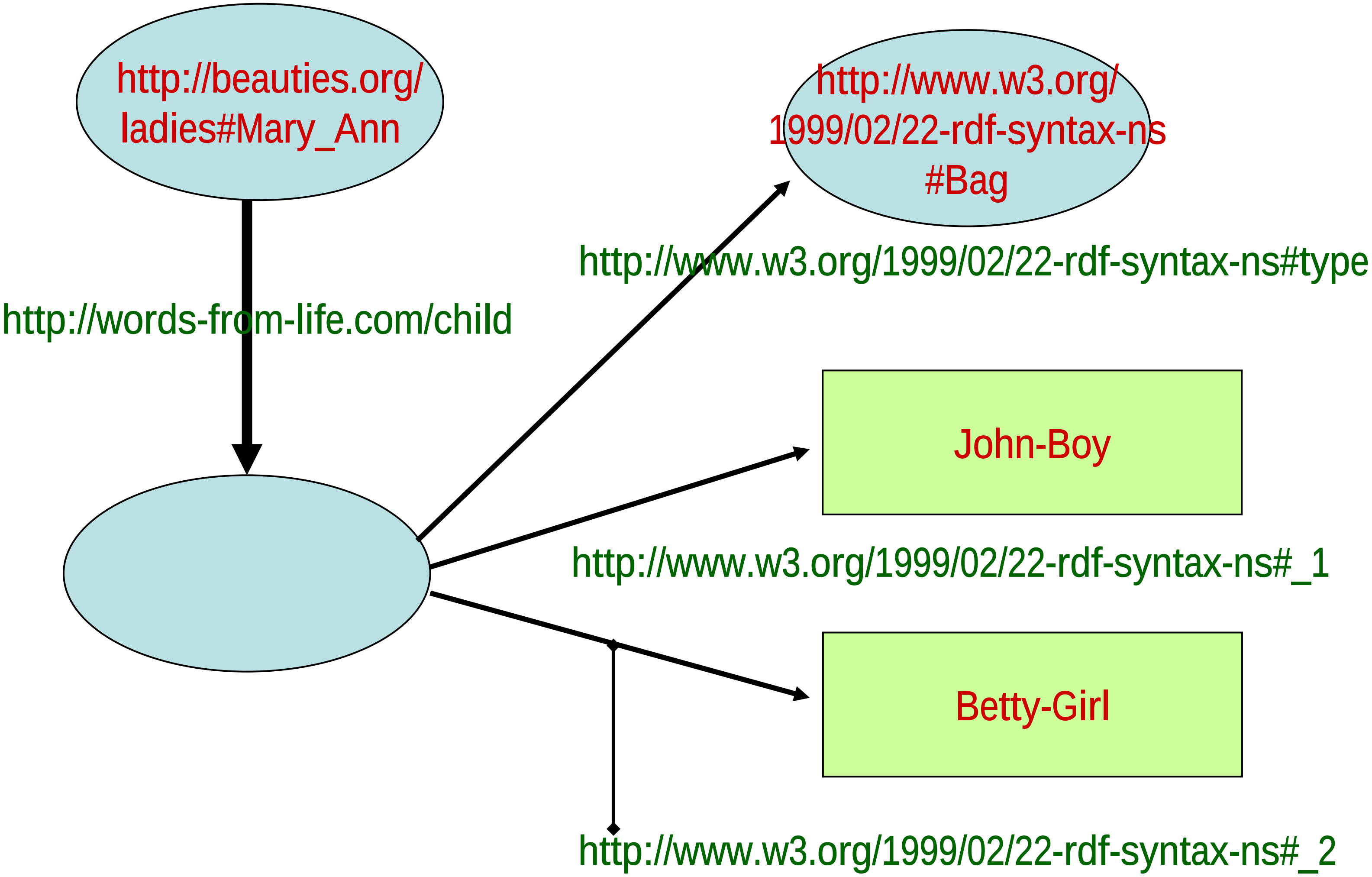

http://beauties.org/
ladies#Mary_Ann
Someone with the
name 'John Doe',
who works at …
http://words-from-buisiness.com/marries
http://words-from-life.com/name
http://words-from-heart.com/feels
http://words-from-work.com/works
John Doe
McDonalds
Go Banana

RDF Containers & Collections

- Containers:
  - rdf:Bag - group of unordered, possible duplicate resources or literals
  - rdf:Seq – group of ordered, possible duplicate resources or literals
  - rdf:Alt – group of literals or resources that are alternatives
- Collection:
  - rdf:parseType=“Collection” - Exhaustive enumeration of members terminated by rdf:nil
  - “linked list” by using rdf:first and rdf:rest

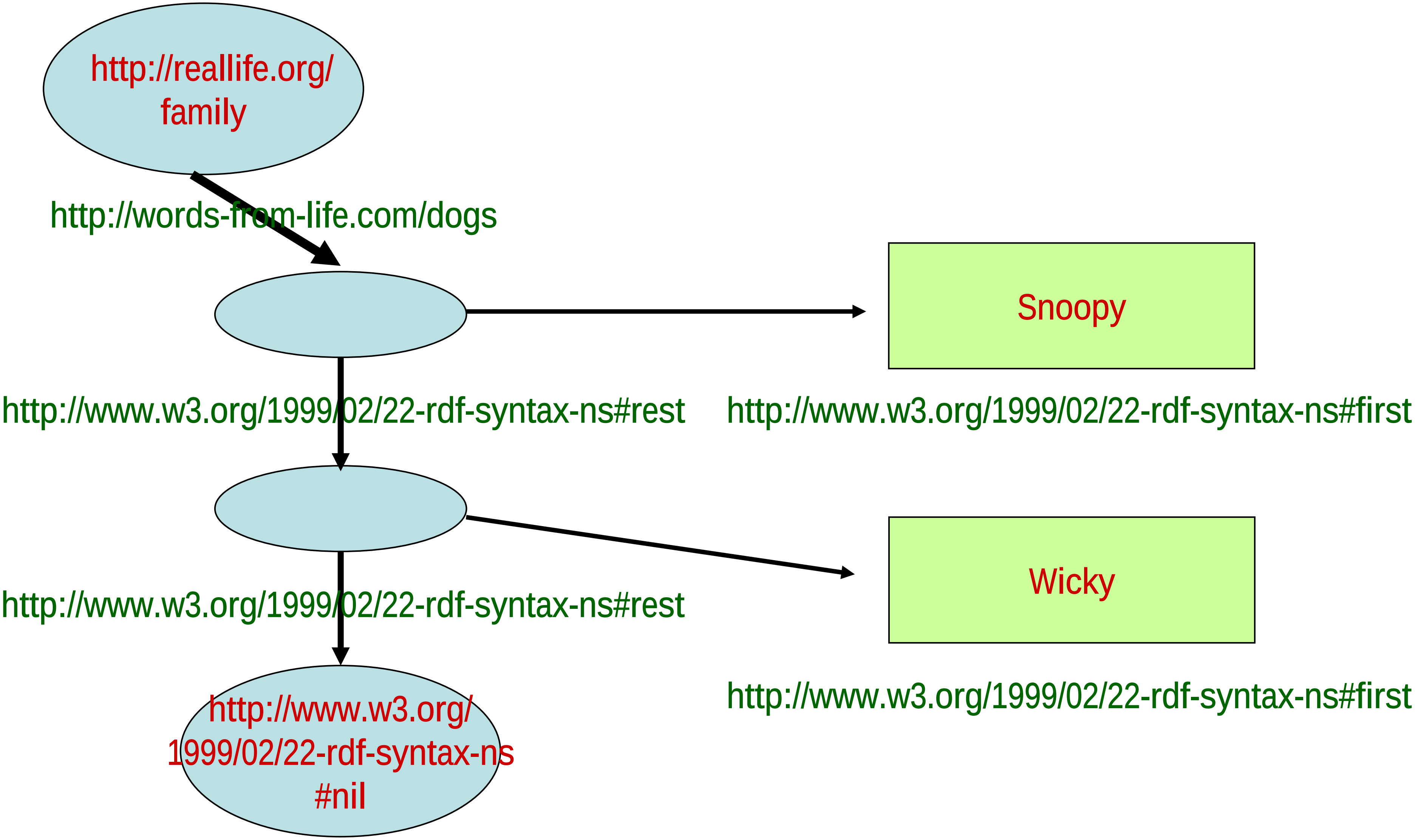
Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

RDF Examples: Containers


http://beauties.org/
ladies#Mary_Ann
http://words-from-life.com/child
http://www.w3.org/
1999/02/22-rdf-syntax-ns
#Bag
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
John-Boy
http://www.w3.org/1999/02/22-rdf-syntax-ns#_1
Betty-Girl
http://www.w3.org/1999/02/22-rdf-syntax-ns#_2

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

RDF Examples: Collections


http://reallife.org/ family
http://words-from-life.com/dogs
Snoopy
http://www.w3.org/1999/02/22-rdf-syntax-ns#rest
http://www.w3.org/1999/02/22-rdf-syntax-ns#first
Wicky
http://www.w3.org/1999/02/22-rdf-syntax-ns#rest
http://www.w3.org/1999/02/22-rdf-syntax-ns#first
http://www.w3.org/ 1999/02/22-rdf-syntax-ns #nil

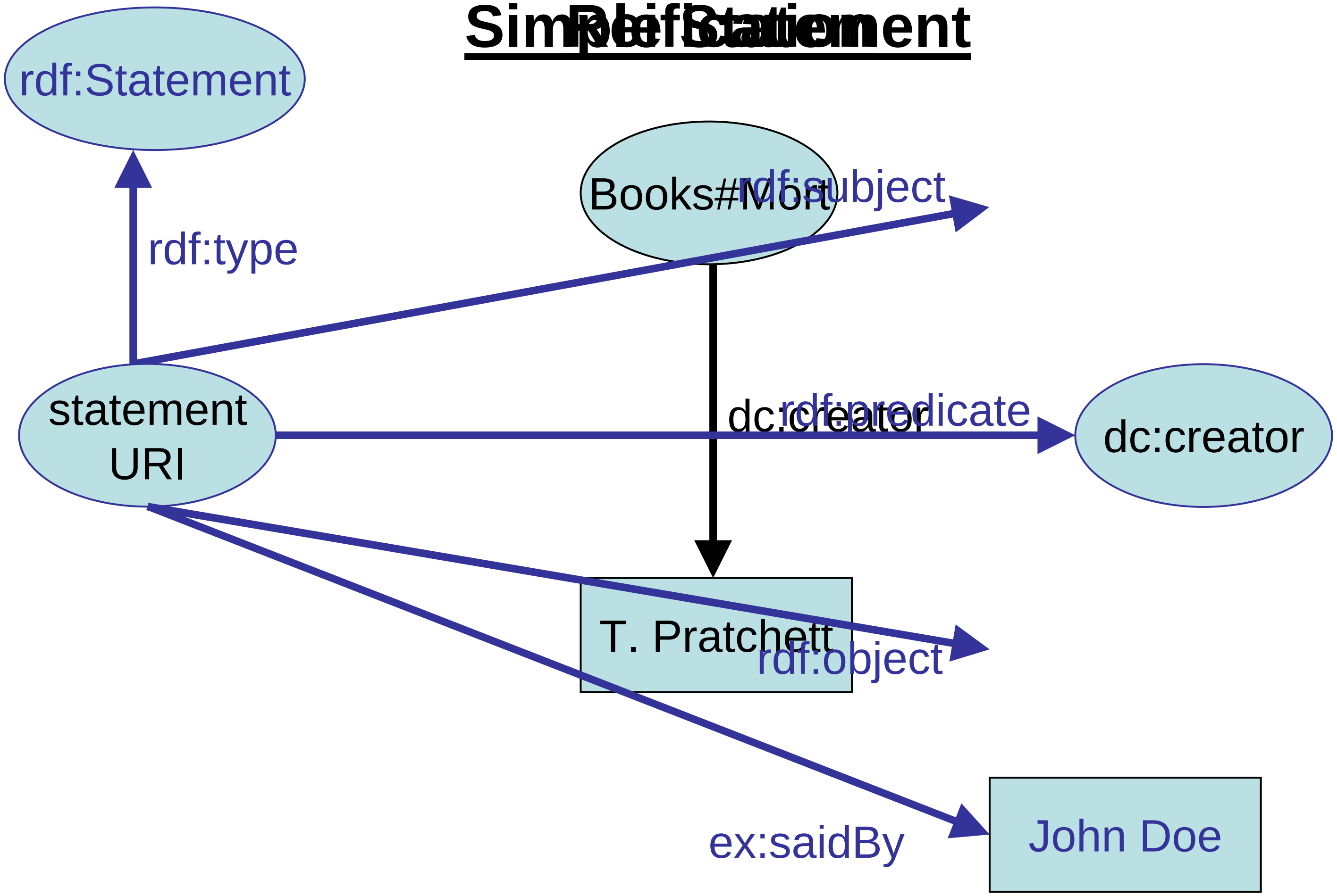
RDF – Reification

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

- Statements about statements

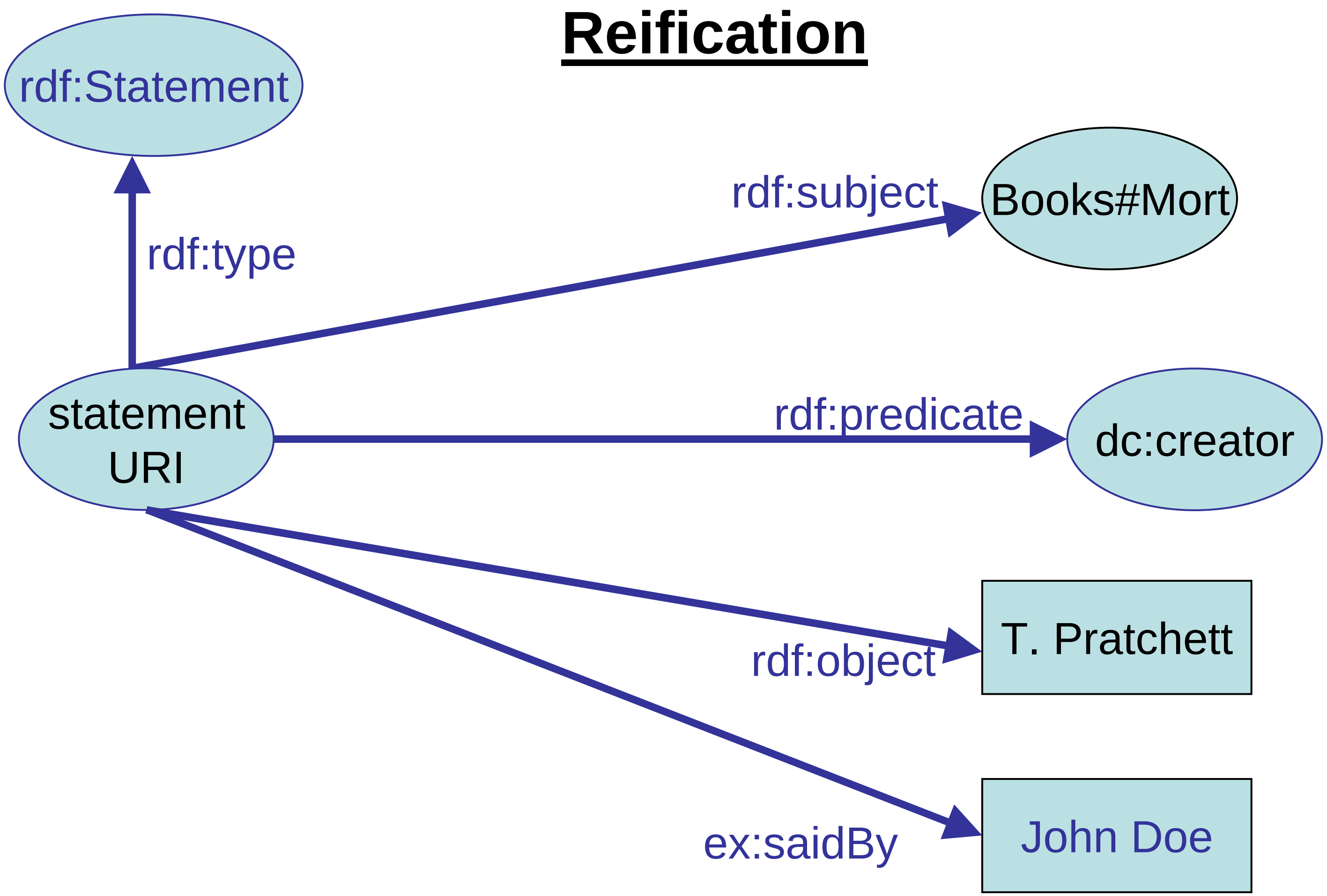
Simple Statement

Reification



RDF-Reification

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de



Reification Aspects

- Allows for expression of provenance of statements
- Describes the relation between a particular triple (S-P-O) and the resource, it refers to
“John said *this* ‘Book Mort has creator T. Pratchett’”
- RDF Reification cannot express
“’all statement instances of a given triple’ are said by John”
- There is no built-in meaning in RDF for the URI of a statement – left to applications …
- Introduces second order statements/logic: Problem for an inference service (but RDF has no 2nd order semantics…)

RDF Serialization: XML

- Namespace:
  - rdf: “http://www.w3.org/1999/02/22-rdf-syntax-ns#”
  - rdfs: “http://www.w3.org/2000/01/rdf-schema#”
  - xsd: “http://www.w3.org/2001/XMLSchema#”
  - Optional add namespaces from domain-specific vocabularies

```
<rdf:RDF xmlns:rdf=“..” xmlns:dc=“..”>

  <rdf:Description rdf:about=“http://../books#Mort”>
      <dc:creator>T. Pratchett</dc:creator>
      <dc:subject>Discworld novel</dc:subject>
  </rdf:Description>

</rdf:RDF>
```

RDF Serialization: Notation 3 (N3)

- Aims:
  - To optimize expression of data and logic in the same language
  - To allow for expression of RDF
  - To allow rules to be integrated smoothly with RDF
  - To allow quoting so that statements about statements can be made.

- The language achieves this by the following features:
  - URI abbreviation using prefixes which are bound to a namespace (using @prefix) a bit like in XML,
  - Enumeration of other objects related to the same subject and predicate by a comma ","
  - Enumeration of another predicate for the same subject using a semicolon ";"
  - Bnode with a certain properties just put the properties between [and]
  - Formulae allow N3 graphs to be quoted within N3 graphs using { and }
  - Variables and quantification allow rules, etc to be expressed
  - The grammar is simple and consistent.

http://www.w3.org/DesignIssues/Notation3.html

N3 - Examples

```
@prefix : <#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix wl: <http://words-from-life>

:Mort dc:creator “Terry Pratchett” ; dc:subject
  “Discworld novel”.

:Mary_Ann wl:child [wl:age 3],[wl:age 4].

:John_Doe :saydBy
      {:Mort  dc:creator “Terry Pratchett”}.
```

Programming RDF

- Jena Semantic Web Framework (JAVA)
  (http://jena.sourceforge.net)
  - Reading and writing RDF in RDF/XML, N3 and N-Triples
  - OWL API
  - In-Memory and persistent storage
  - RDQL query support
- Redland RDF Application Framework (C)
  - Language bindings to C#, Java, Obj-C, Perl, PHP, Python, Ruby and TCL
  - Reading and writing RDF in RDF/XML, N-Triples and Turtle Terse RDF Triple Language
  - In-Memory and persistent storage
  - RDQL and SPARQL support

Creating an RDF Model

```
//create default RDF model
Model model =  ModelFactory.createDefaultModel();

//create subject to make statement
Resource subject =
   model.createResource("http://publisher.com/Books#Mort");
//create predicate dc:creator
Property predicate =
   model.createProperty("http://purl.org/dc/elements/1.1/",
   "creator" );
//create object (value of predicate)
Literal object = model.createLiteral("Terry Pratchett");
//create statement (triple of subject, predicate and object)
Statement statement = model.createStatement(subject,
   predicate, object);
//append to model
model.add(statement);

// short hand
model.add(subject, DC.subject, "Discworld novel");
```

Serializing a RDF model

- RDF/XML:

```
model.write(System.out, "RDF/XML");
```

or

```
model.write(System.out, "RDF/XML-ABBREV");
```

- N-Triples:

```
model.write(System.out, "N-TRIPLE");
```

- N3:

```
model.write(System.out, "N3");
```

Parsing a RDF file

```
//get file
File f = new File(“example.rdf”);

//create new default model
Model model = ModelFactory.createDefaultModel();

//fill model
model.read( new FileInputStream(f), "", "RDF/XML");
```

Base to convert relative in absolute URIs

Language of source

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Programming Reification

```
Resource simpleSubject =
   model.createResource(“/Books#Mort");
Statement statement = model.createStatement(simpleSubject,
   DC.creator,"Terry Pratchett");

ReifiedStatement r_statement =
   model.createReifiedStatement(“/Reification", statement);

Property predicate = model.createProperty(“..", "saidBy");
Literal object = model.createLiteral("John Doo");

Statement st = model.createStatement(r_statement,
   predicate, object);

model.add(st);
```

RDF Schema

RDF (taken for pure) can only speak about instances …

- RDF vocabulary description language
- Semantic extension to RDF
- Domain independent
- Property-centric
- Defines Classes and Properties to describe other Classes, Properties and Resources
- Extended expressiveness: Define categories

RDFS-Classes

- Describe ‘kinds of things’.
- Classes are resources often identified by RDF URI references.
- Instantiation via **`rdf:type`** and its name
- Further specification through properties (e.g. “**`subClassOF`**”,…)
- Difference to OO classes: properties are defined independently of classes, possibly related via the **`rdf:domain`** or **`rdf:range`** properties (i.e. types are bound to properties)

Ex: The same ‘weight’ property can be applied to lemons and elephants …

Predefined Classes

Class name	Comment
rdfs:Resource	The class resource, base class.
rdfs:Literal	The class of literal values, e.g. textual strings and integers.
rdf:XMLLiteral	The class of XML literals values.
rdfs:Class	The class of classes.
rdf:Property	The class of RDF properties.
rdfs:Datatype	The class of RDF datatypes.
rdf:Statement	The class of RDF statements.
rdf:Bag	The class of unordered containers.
rdf:Seq	The class of ordered containers.
rdf:Alt	The class of containers of alternatives.
rdfs:Container	The class of RDF containers.
rdfs:ContainerMembershipProperty	The class of container membership properties, rdf:_1, rdf:_2, ..., all of which are sub-properties of 'member'.
rdf:List	The class of RDF Lists.

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Predefined Properties

Property Name	Comment	Domain	Range
rdf:type	The subject is an instance of a class.	rdfs:Resource	rdfs:Class
rdfs:subClassOf	The subject is a subclass of a class.	rdfs:Class	rdfs:Class
rdfs:subPropertyOf	The subject is a subproperty of a property.	rdf:Property	rdf:Property
rdfs:domain	A domain of the subject property.	rdf:Property	rdfs:Class
rdfs:range	A range of the subject property.	rdf:Property	rdfs:Class
rdfs:label	A human-readable name for the subject.	rdfs:Resource	rdfs:Literal
rdfs:comment	A description of the subject resource.	rdfs:Resource	rdfs:Literal

Predefined Properties

Collections

Property name	Comment	Domain	Range
rdf:first	The first item in the subject RDF list.	rdf:List	rdfs:Resource
rdf:rest	The rest of the subject RDF list after the first item.	rdf:List	rdf:List
rdf:nil	List terminator.	rdf:List	rdf:List

Reification

Property name	Comment	Domain	Range
rdf:subject	The subject of the subject RDF statement.	rdf:Statement	rdfs:Resource
rdf:predicate	The predicate of the subject RDF statement.	rdf:Statement	rdfs:Resource
rdf:object	The object of the subject RDF statement.	rdf:Statement	rdfs:Resource

Predefined Properties

Utility

Property name	Comment	Domain	Range
rdfs:seeAlso	Further information about the subject resource.	rdfs:Resource	rdfs:Resource
Rdfs:isDefinedBy	The definition of the subject resource.	rdfs:Resource	rdfs:Resource
rdf:value	Idiomatic property used for structured values.	rdfs:Resource	rdfs:Resource

Container and Classes

Property name	Comment	Domain	Range
rdfs:member	A member of the subject resource.	rdfs:Resource	rdfs:Resource

Definition of classes

- Class definition:

```
<rdf:Description rdf:ID=“Person">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Description>
```

Or short hand `<rdfs:Class rdf:ID=“Person” />`

- Instantitiation

```
  <ex:Person rdf:ID=“Donald E. Knuth” />
```

- Inheritance:

```
<rdfs:Class rdf:ID=“Teacher”>
    <rdfs:subClassOf rdf:resource=“#Person” />
    <rdfs:subClassOf rdf:resource=“#UniversityStaff” />
</rdfs:Class>
```

Describing Properties

```
<rdf:Property rdf:ID=“offers”>
    <rdfs:domain rdf:resource=“#Teacher” />
    <rdfs:range rdf:resouce=“#Course” />
</rdf:Property>

<rdf:Property rdf:ID=“age”>
    <rdfs:domain rdf:resource=“#Person” />
    <rdfs:range rdf:resource=“&xsd;integer” />
</rdf:Property>

<ex:Teacher rdf:ID=“Donald E. Knuth”>
    <ex:offers rdf:resource=“#AdvancedTeX” />
    <ex:age rdf:datatype=“&xsd;integer”>0..</ex:age>
</ex:Teacher>
```

Programming: Creating Classes

```
// setting default namespace
private static String NS = “urn:my:eg/”;


Resource cPerson = model.createResource(NS+"Person");
model.add( cPerson, RDF.type, RDFS.Class );

Resource cTeacher = model.createResource(NS+"Teacher");
model.add( cTeacher,  RDF.type, RDFS.Class );
model.add( cTeacher, RDFS.subClassOf, cPerson );

Resource cCourse = model.createResource(NS+"Course");
model.add( cCourse, RDF.type, RDFS.Class );
```

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Creating Properties

```
Resource pOffers = model.createResource(NS+"offers");
model.add( pOffers, RDF.type, RDF.Property );
model.add( pOffers, RDFS.domain, cPerson );
model.add( pOffers, RDFS.range, cCourse );

Resource pAdvises = model.createResource(NS+"advises");
model.add( pAdvises, RDF.type, RDF.Property );
model.add( pAdvises, RDFS.subPropertyOf, pOffers);
```

(Simple) Inference

```
@prefix :      <urn:fhtw:eg/> .
@prefix :rdfs <http://www.w3.org/2000/01/rdf-schema#> .
:Person a rdfs:Class .
:Course a rdfs:Class .
:Teacher a rdfs:Class ; rdfs:subClassOf :Person .
:#Michael Engelhardt a :Person .
:#DonaldKnuth a :Teacher .
```

Problem: How to get all persons from the model?

Solution: Resolving the class subClass relationship

Creating Inference Models

```
/* create inference model based on RDFS; schema holds RDFS
   statements; data contains RDF sample data statements */
InfModel infModel = ModelFactory.createRDFSModel(schema,
   data);
// get resource to query; :Person
Resource spec = infModel.getResource(NS+“Person");
/* obtain iterator to all statements having rdf:type of
   person */
ResIterator it =
   infModel.listSubjectsWithProperty(RDF.type, spec );
```

Returns:

```
:#MichaelEngelhardt
:#DonaldKnuth
```

The Limits of RDF/RDFS

RDFS follows a set oriented approach in expressing logic …

with the absence of following expressions:

- Cardinality constraints
- Transitivity
- Uniqueness
- Set operations: Unions, Intersections, Complements, …
- 'All' or empty set
- Quantifiers

References

- Semantic Web @ W3C - http://www.w3.org/2001/sw/
- RDF Primer - http://www.w3.org/TR/rdf-primer/
- RDF Model & Abstract Syntax - http://www.w3.org/TR/rdf-concepts/
- RDF Schema 1.0 - http://www.w3.org/TR/rdf-schema/
- RDF Semantics - http://www.w3.org/TR/rdf-mt/
- RDF Validator – http://www.w3.org/RDF/Validator/
- Shelley Powers: Practical RDF, O'Reilly, 2003.
- R. Eckstein, S. Eckstein: XML und Datenmodellierung, dpunkt 2004
- D. Fensel: Ontologies, 2nd Ed, Springer 2004.
- Daconta, Obrst, Smith: The Semantic Web, Wiley 2003.
- Jena Javadoc - http://jena.sourceforge.net/javadoc/