Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Semantic Web Technologies: Web Ontology Language

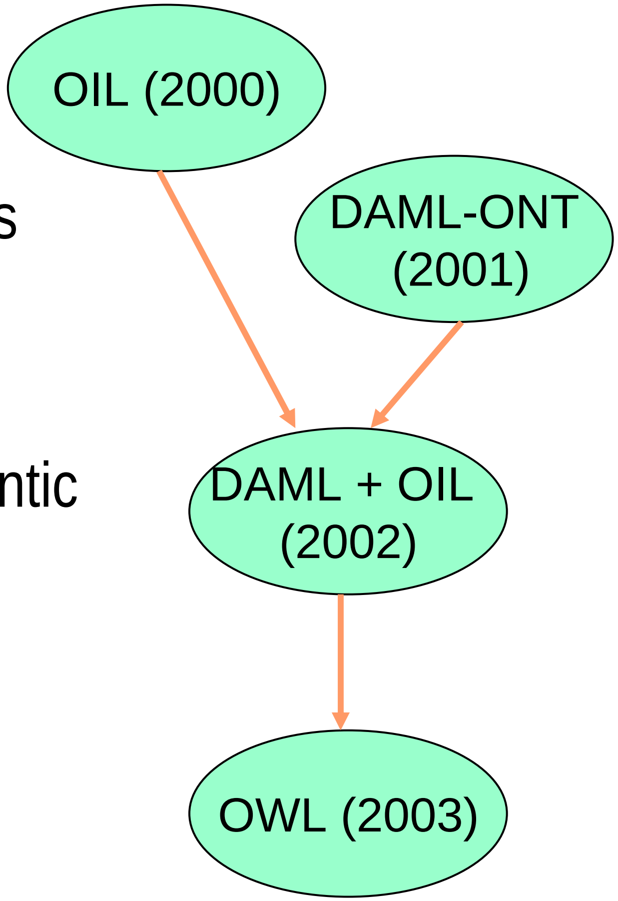
- Motivation
- OWL Formal Semantic
- OWL Synopsis
- OWL Programming

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Introduction

- XML / XML Schema provides a portable framework for defining a syntax
- RDF forms a datamodel + semantic to express relations between resources
- RDFS offers a vocabulary for describing RDF properties and classes combined with a semantic of hierarchies
- Missing: a vocabulary + formal semantic to describe more general logical relations

→ An Ontology language



Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

OWL Objectives

- Create / formulate a generally understandable structure of information, which allows for inference
- Enable reuse of present knowledge in different contexts and applications
- Provide a toolset to develop / adapt current knowledge according to changing conditions
- Find a technology to integrate existent information resources to form powerful knowledge-bases

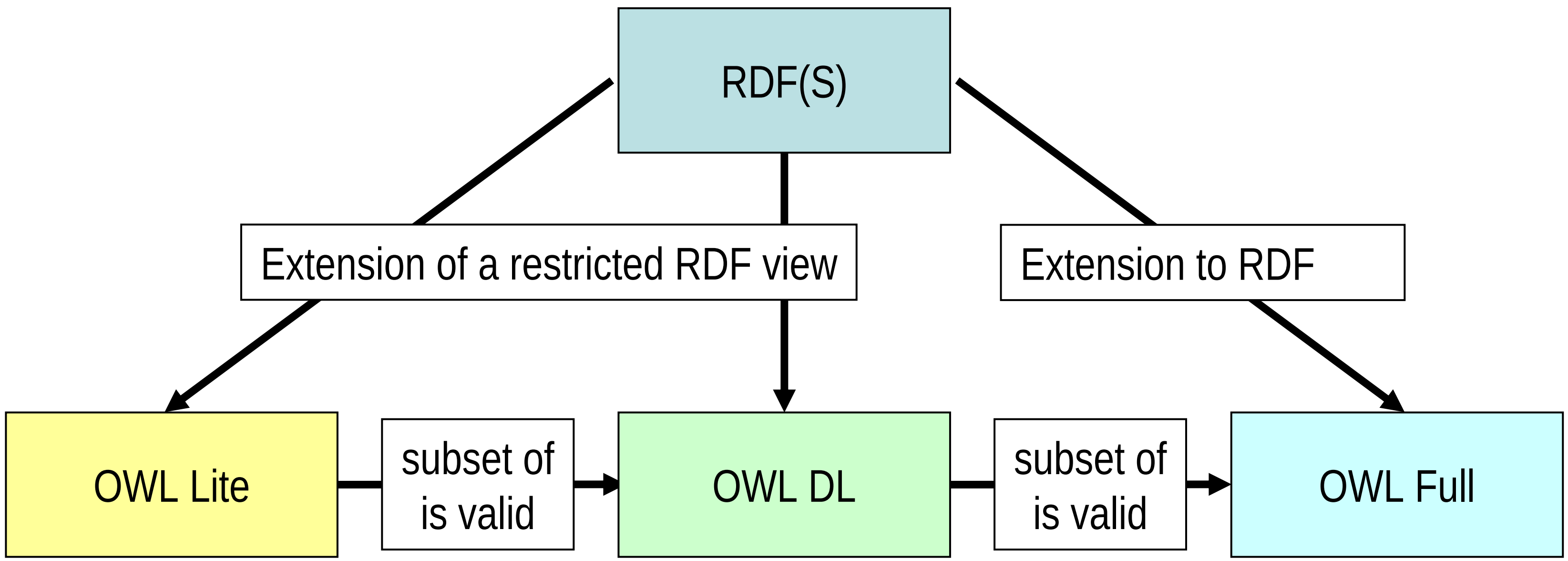
Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Six designated Use Cases

1. Web portals
   - Navigation and content retrieval
2. Multimedia collections
   - Media and content specific organisation and retrieval
3. Corporate Web site management
   - Adaptive access and presentation
4. Design documentation
   - Build and explore information model
5. Agents and services
   - Offer high-level tasks on integrated information
6. Ubiquitous computing
   - Provide interoperation in unchoreographed conditions

Classification of OWL in relation to RDF

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de



- Hierarchical classifications with simple constraints
- Cardinality of 0 and 1 only

- Maximum expressiveness while retaining computational completeness under certain constraints.
  (*eg. classes cannot be instances of another class*)

- Maximum expressiveness and the syntactic freedom of RDF with no computational guarantees

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Main Additions of OWL to RDFS

- New meanings of properties
  - local scope, type ($\forall$), value ($\exists$) + cardinalities
- Characteristics of properties
  - transitive, symmetric …
- Boolean expressions of classes
  - disjunction, conjunction, negation
- Defined classes
  - necessary and sufficient conditions

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

OWL Formal Semantic

- Comprises meaning beyond words:
- Inherent mapping to expressive description logic (DL):
  - eats value (meat or fish)

  $= \exists$ eats:meat $\cup \exists$ eats:fish
- Mapping used for reasoning support in DL reasoning systems

OWL Formal Reasoning

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Reason about class membership, equivalence and transitivity:

- herbivore ⇔ animal eats (plant or (part_of plant))
- tree ⇒ plant
- branch ⇒ part_of tree
- leaf ⇒ part_of branch
- giraffe ⇒ animal eats leaf
- part_of = transitive

Now we can derive:

- giraffe ⇒ herbivore

OWL Synopsis

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

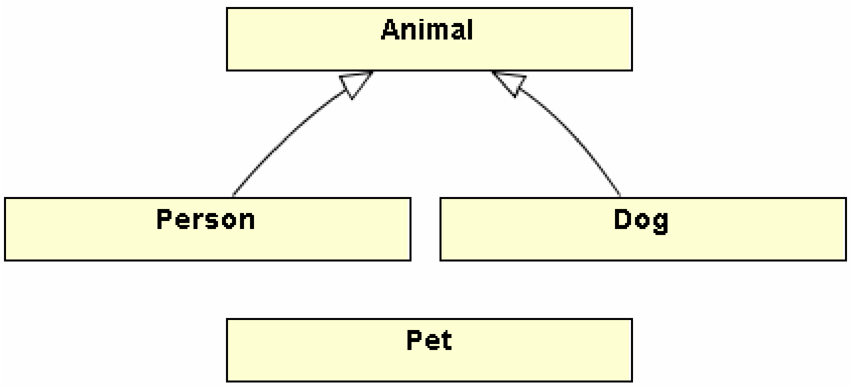
RDF Schema Features		
Term		**Description**
Class		A class defines a group of individuals that belong together because they share some properties
	Thing	Built-in class being the most general class and superclass of all OWL classes.
	Nothing	Built-in class being the most specific class and subclass of all OWL classes. This class cannot have any instances.
Individual		Individuals are instances of classes, and properties may be used to relate one individual to another.

- Includes the following features of RDF(s):
  `rdfs:subClassOf, rdf:Property, rdfs:subPropertyOf, rdfs:domain, rdfs:range`

OWL Classes & Instances

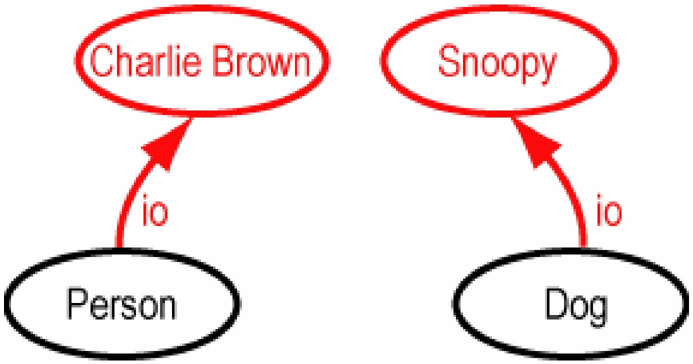
Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Defining Classes




```
<owl:Class rdf:ID="Animal" />
<owl:Class rdf:ID="Pet" />
<owl:Class rdf:ID="Person">
   <rdfs:subClassOf
   rdf:resource="#Animal" />
</owl:Class>
<owl:Class rdf:ID="Dog">
   <rdfs:subClassOf
   rdf:resource="#Animal" />
</owl:Class>
```

Declaring Instances




```
<Person rdf:about="CharlieBrown"/>
<Dog rdf:about="Snoopy" />
```

OWL Synopsis

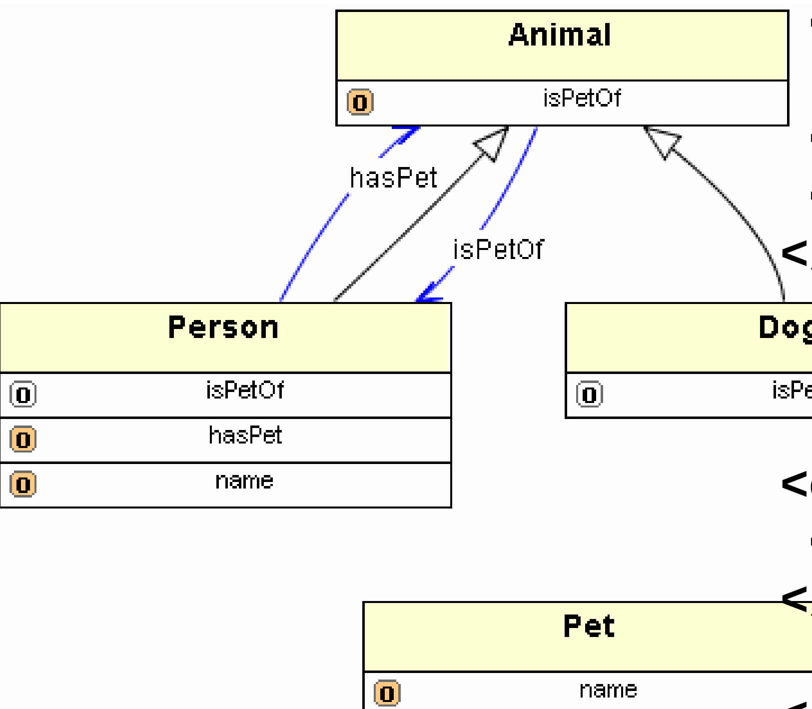
Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Property Characteristics

Term	Description
ObjectProperty	Relations between instances of two classes. (not intended to reflect a connection with **`rdf:object`**)
DatatypeProperty	Relations between instances of classes and RDF literals and XML Schema datatypes.
inverseOf	Property is stated to be the inverse of another property. *P1(x,y) iff P2(y,x)*
TransitiveProperty	Property is stated to be transitive. *P(x,y) and P(y,z) implies P(x,z)*
SymetricProperty	Property is stated to be symetric. *P(x,y) iff P(y,x)*
FunctionalProperty	Property is stated to be functional. *P(x,y) and P(x,z) implies y = z*
InverseFunctionalProperty	Property is stated to be inverse functional. *P(y,x) and P(z,x) implies y = z*

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

OWL Properties




```
<owl:ObjectProperty rdf:ID="likes"/>

<owl:ObjectProperty rdf:ID="hasPet"/>
 <rdfs:subPropertyOf
   rdf:resource=“#likes”/>
 <rdfs:domain rdf:resource="#Person"/>
 <rdfs:range rdf:resource="#Animal"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="isPetOf">
 <owl:inverseOf rdf:resource=“hasPet”>
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:ID="name">
 <rdfs:domain rdf:resource=“#Person” />
 <rdfs:range
   rdf:resource=“&xsd;string"/>
</owl:DatatypeProperty>
```

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

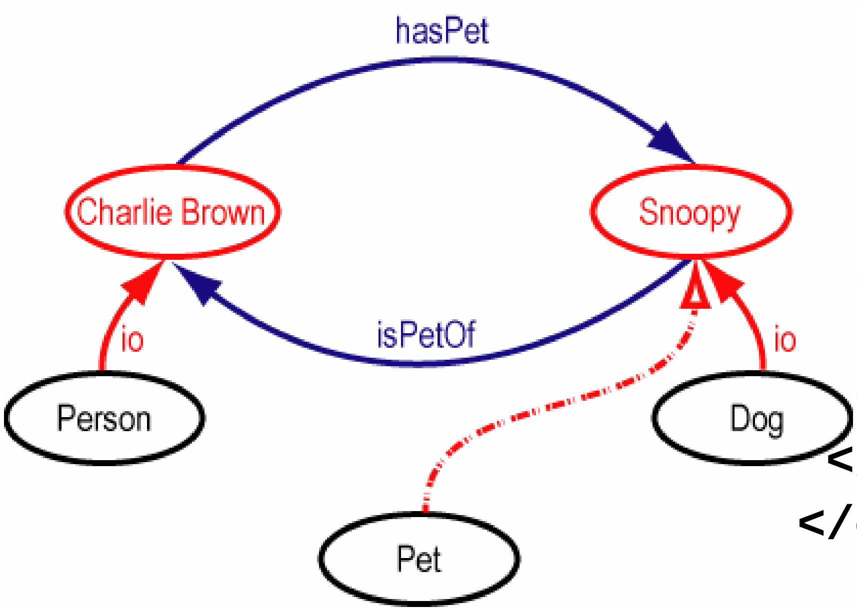
OWL Synopsis

(In) Equality

Term		Description
equivalentClass		Two classes may be stated to be equivalent. Equivalent classes have the same instances. Equality can be used to create synonymous classes.
equivalentProperty		Two properties may be stated to be equivalent. Equivalent properties relate one individual to the same set of other individuals. Equality may be used to create synonymous properties.
sameAs		Two individuals may be stated to be the same. These constructs may be used to create a number of different names that refer to the same individual.
differentFrom		An individual may be stated to be different from other individuals.
AllDifferent		A number of individuals may be stated to be mutually distinct in one AllDifferent statement.
	distinctMembers	States that all members of a list are distinct and pairwise disjoint.

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Declaring Equivalent Classes




```
<owl:Class rdf:ID="Pet">
 <owl:equivalentClass>
  <owl:Restriction>
   <owl:onProperty>
     <owl:ObjectProperty
        rdf:ID="isPetOf"/>
  </owl:onProperty>
  <owl:someValuesFrom
        rdf:resource="&owl;#Thing"/>
  </owl:Restriction>
 </owl:equivalentClass>
</owl:Class>
```

```
<Dog rdf:about="Snoopy">
   <isPetOf
   rdf:resource="#CharlieBrown" />
</Dog>
```

OWL Synopsis

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Property Restrictions

Term		Description
Restriction		Restrict value range of a property for a specific (sub) class.
	onProperty	Indicates the restricted property.
	allValuesFrom	The restriction allValuesFrom is stated on a property with respect to a class. It means that this property on this particular class has a local range restriction associated with it. Thus if an instance of the class is related by the property to a second individual, then the second individual can be inferred to be an instance of the local range restriction class.
	someValuesFrom	The restriction someValuesFrom is stated on a property with respect to a class. A particular class may have a restriction on a property that at least one value for that property is of a certain type.

OWL Synopsis

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Restricted Cardinality

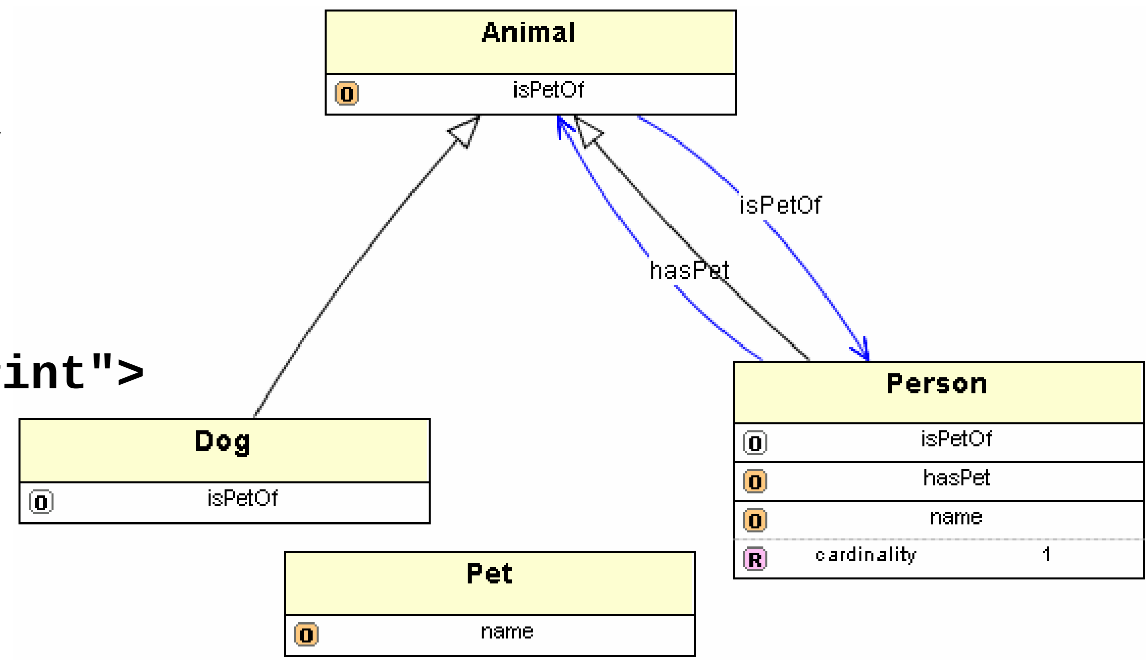
Term	Description
minCardinality	Cardinality is stated on a property with respect to a particular class. If a minCardinality of 1 is stated on a property with respect to a class, then any instance of that class will be related to at least one individual by that property. This restriction is another way of saying that the property is required to have a value for all instances of the class.
maxCardinality	Cardinality is stated on a property with respect to a particular class. If a maxCardinality of 1 is stated on a property with respect to a class, then any instance of that class will be related to at most one individual by that property.
cardinality	Cardinality is provided as a convenience when it is useful to state that a property on a class has both minCardinality 0 and maxCardinality 0 or both minCardinality 1 and maxCardinality 1.

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Restricting Values of a Class Property

```
<owl:Class rdf:ID="Person">
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty>
    <owl:DatatypeProperty
      rdf:ID="name"/>
   </owl:onProperty>
   <owl:cardinality
     rdf:datatype=“&xsd;#int">
    1
   </owl:cardinality>
   </owl:Restriction>
 </rdfs:subClassOf>

 <rdfs:subClassOf rdf:resource=“Animal”/>
</owl:Class>
```



Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

OWL Synopsis

Header Information

Term	Description
Ontology	Root tag of an ontology.
imports	Tag for including other ontology definitions.

Versioning

Term	Description
versionInfo	A standard tag intended to provide hooks for version control systems working with ontologies.
priorVersion	Reference to a prior version of this ontology.
backwardCompatibleWith	Reference to another compatible version of this ontology.
IncompatibleWith	Reference to another incompatible version of this ontology.
DeprecatedClass	Subclass of Class/Property. By deprecating a term, it means that the term should not be used in new documents that commit to the ontology.
DeprecatedProperty	

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

OWL Synopsis

Class intersection

Term	Description
intersectionOf*	This property links a class to a list of class descriptions. intersectionOf can be viewed as being analogous to logical conjunction.

Boolean Combinations of Class Expressions (OWL DL & FULL)

Term	Description
complementOf	This property links a class to a list of class descriptions.
unionOf	This property links to precisely one class description. unionOf is analogous to logical disjunction.

***Note:** OWL Lite restricts the usage of intersectionOf. The values of the intersectionOf list must be class identifiers and/or property restrictions. Thus, "complete class" axioms using enumeration, complement and union are not allowed in OWL Lite.

OWL Synopsis

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Annotation Properties

Term	Description
AnnotationProperty	Define a certain property being an annotation under the following conditions: • The sets of object properties, datatype properties, annotation properties and ontology properties must be mutually disjoint. • Annotation properties must have an explicit typing triple of the form: **`AnnotationPropertyID rdf:type owl:AnnotationProperty .`** • Annotation properties must not be used in property axioms. Thus, in OWL DL one cannot define subproperties or domain/range constraints for annotation properties. • The object of an annotation property must be either a data literal, a URI reference, or an individual.
OntologyProperty	Define a certain property being an annotation property in the ontology header.

- In addition OWL uses the following predefined properties from RDFS: **`rdfs:label, rdfs:comment, rdfs:seeAlso and rdfs:isDefinedBy`**

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

OWL Synopsis

Annotation Properties

Term	Description
oneOf*	Classes can be described by enumeration of the individuals that make up the class. The members of the class are exactly the set of enumerated individuals; no more, no less.
DataRange	An additional construct for defining a range of data values, namely an enumerated datatype.
disjointWith*	Classes may be stated to be disjoint from each other.

Annotation Properties

Term	Description
hasValue*	A property can be required to have a certain individual as a value (also sometimes referred to as property values).

***Note:** These properties are only available in OWL DL & OWL Full.

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Ontology Creation

```
<!DOCTYPE rdf:RDF [
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#”>
]>
<rdf:RDF xmlns:owl =“http://www.w3.org/2002/07/owl#”
         xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-
  ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
         xmlns:xsd =“&xsd;">

  <owl:Ontology rdf:about=“”>
        <rdfs:comment>An example OWL ontology</rdfs:comment>
        <rdfs:label>Simple University Ontology</rdfs:label>
  </owl:Ontology>

<!–- Define ontology here -->

</rdf:RDF>
```

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Another Example

```
<owl:Class rdf:ID="Person"/>

<owl:Class rdf:ID="Teacher">
  <rdfs:subClassOf rdf:resource="#Person" />
</owl:Class>

<owl:Class rdf:ID="Student">
  <rdfs:subClassOf rdf:resource="#Person" />
</owl:Class>

<owl:Class rdf:ID="UniversityStaff">
  <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:resource="#Teacher" />
      <owl:Class rdf:resource="#Student" />
  </owl:unionOf>
</owl:Class>
```

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

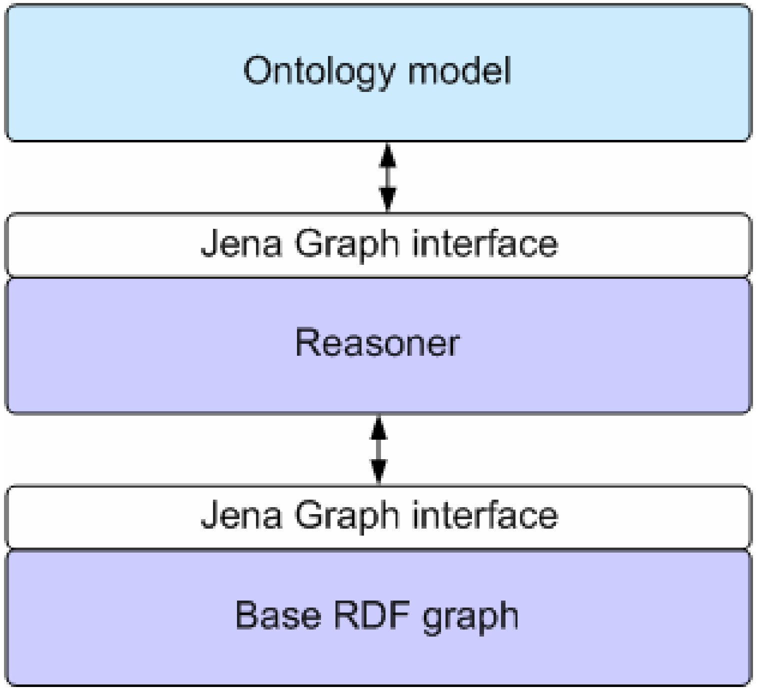
Another Example (2)

```
<owl:DatatypeProperty rdf:ID=“name”>
   <rdfs:domain rdf:resource=“#Person” />
   <rdfs:range rdf:resource=“&xsd;string” />
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID=“advises”>
   <rdfs:domain rdf:resource=“#Teacher” />
   <rdfs:range rdf:resource=“#Student” />
</owl:ObjectProperty>
```

Jena Ontology API Overview

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de



http://jena.sourceforge.net/ontology/index.html

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Programming OWL

- Model creation

```
OntModel om = ModelFactory.createOntologyModel(
ProfileRegistry.OWL_DL_LANG );
```

- Creating OWL classes:

```
OntClass person = om.createClass(NS+"Person");
OntClass teacher = om.createClass(NS+"Teacher");
OntClass student = om.createClass(NS+"Student");
```

- Inheritance

```
person.addSubClass(teacher);
student.addSuperClass(person);
```

- Features

```
RDFList list = om.createList(new RDFNode[] {student,teacher} );
OntClass universityStaff =
om.createUnionClass(NS+"UniversityStaff", list);
universityStaff.addSuperClass(person);
```

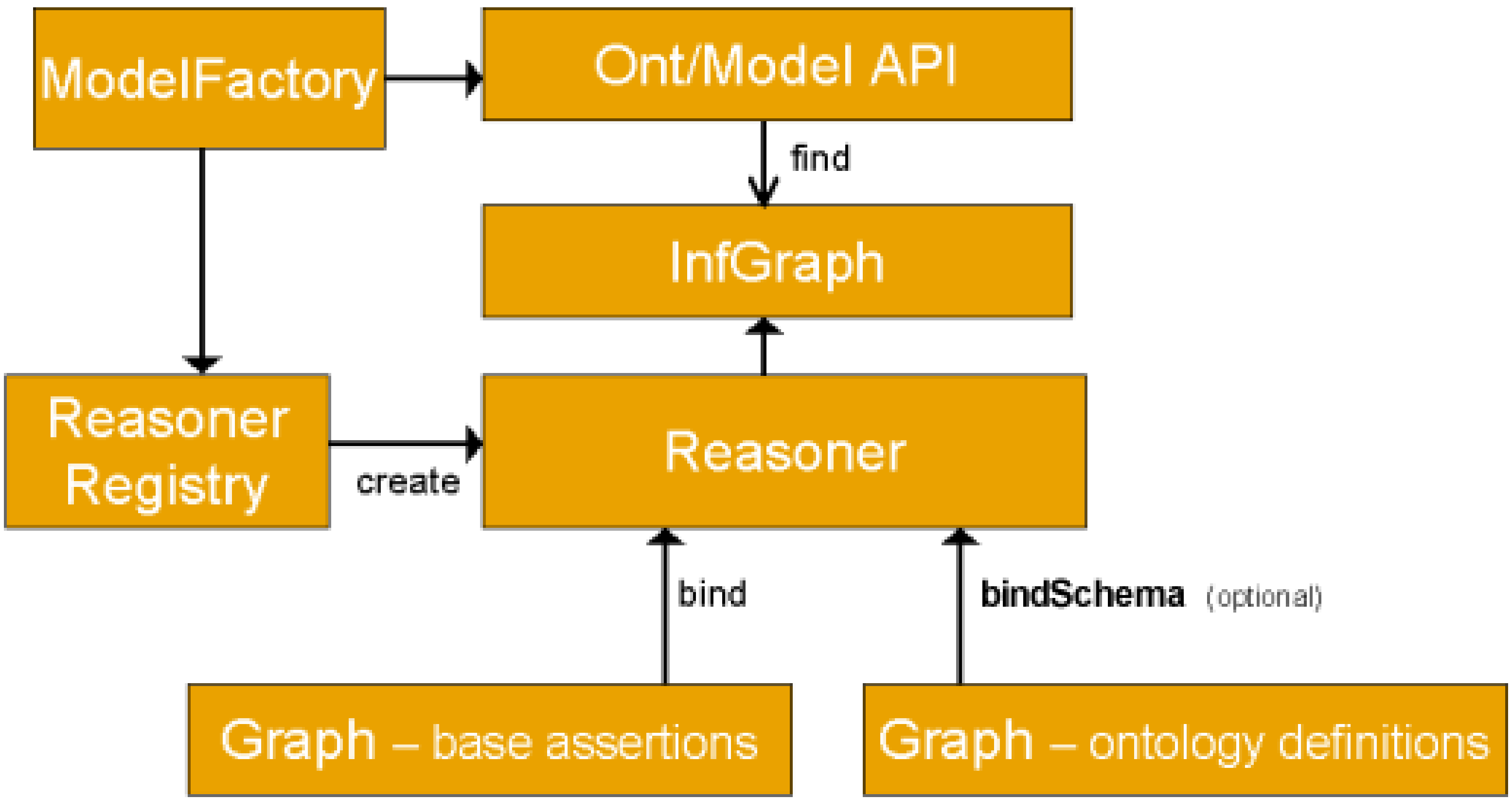
Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Creating Properties

- DatatypeProperties:
```
DatatypeProperty name =
om.createDatatypeProperty(NS+"name");
name.addDomain(person);
name.addRange(XSD.xstring);
```
- ObjectProperties:
```
ObjectProperty advises =
om.createObjectProperty(NS+"advises");
advises.addDomain(teacher);
advises.addRange(student);
```

OWL Inference with Jena

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de



http://jena.sourceforge.net/inference/index.html

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Obtaining a Reasoner

- A Reasoner can be obtained from the ReasonerRegistry:
  `Reasoner reasoner = ReasonerRegistry.getOWLReasoner();`
- Reasoners are configured by applying certain properties via setParameter(Property,Object) using vocabulary from ReasonerVocabulary

```
reasoner.setParameter(ReasonerVocabulary.PROPtraceOn,
                                new Boolean(true));
```

- A specific ontology (schema) can be bound by using bindSchema(Model) or bindSchema(Graph)

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

References

- Semantic Web @ W3C - http://www.w3.org/2001/sw/
- OWL Overview - http://www.w3.org/TR/owl-features/
- OWL Semantics & Abstract Syntax - http://www.w3.org/TR/owl-semantics/
- OWL Guide - http://www.w3.org/TR/owl-guide/
- Ubbo Visser et. al: Web Development, WWW Tutorial May 2004
- D. Fensel: Ontologies, 2nd Ed, Springer 2004.
- Daconta, Obrst, Smith: The Semantic Web, Wiley 2003.
- Jena Javadoc - http://jena.sourceforge.net/javadoc/
- S. Staab, R.Studer (Eds.): Handbook on Ontologies, Springer 2004.