# Semantic Web Technologies

## Web Ontology Language OWL

Jos de Bruijn
jos.debruijn@deri.org

Digital Enterprise Research Institute (DERI)
University of Innsbruck, Austria

May 10, 2006





## Outline







## Limitation of RDFS

- No semantics for:
  - Containers
  - Collections
  - Reification
- Domain and range of property **infer** information rather than **check** data
  - Conjunctive interpretation of multiple restrictions
- Use of properties as objects
- RDF/XML syntax very verbose





## RDFS as an Ontology Language

- Classes
- Properties
- Class hierarchies
- Property hierarchies
- Domain and range restrictions





## Expressive limitations of RDF(S)

- Only binary relations
- Characteristics of Properties (e.g. inverse, transitive, symmetric)
- Local range restrictions (e.g. for Class Person, the property hasName has range xsd:string)
- Complex concept descriptions (e.g. Person is defined by Man and Woman)
- Cardinality restrictions (e.g. a Person may have at most 1 name)
- Disjointness axioms (e.g. nobody can be both a Man and a Woman)

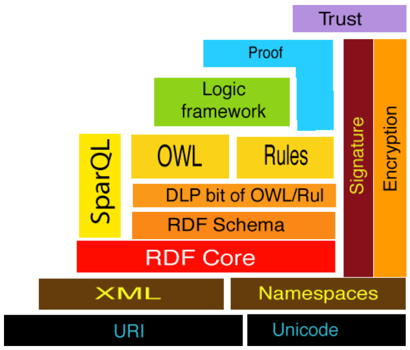




## Layering issues

- Syntax
  - Only binary relations in RDF
  - Verbose Syntax
  - No limitations on graph in RDF
    - Every graph is valid
- Semantics
  - Mayformed graphs
  - Use of vocabulary in language
    - e.g. ⟨rdfs:Class,rdfs:subClassOf,ex:a⟩
  - Meta-classes
    - e.g. ⟨ex:a,rdf:type,ex:a⟩

## Where were we?




[following slides in part due to Frank van Harmelen

http://www.cs.vu.nl/~frankh/spool/SemWebSlides/OWL.ppt]





## Stack of Languages

- XML
  - Surface syntax, no semantics
- XML Schema
  - Describes structure of XML documents
- RDF
  - Datamodel for "relations" between "things"
- RDF Schema
  - RDF Vocabulary Definition Language
- OWL
  - A more expressive Vocabulary Definition Language





## RDF Schema recap

- RDFS provides
  - Classes
  - Class hierarchies
  - Properties
  - Property hierarchies
  - Domain and range restrictions
- RDFS does not provide
  - Property characteristics (inverse, transitive, ...)
  - Local range restrictions
  - Complex concept definitions
  - Cardinality restrictions
  - Disjointness axioms





## Extending RDF Schema

- OWL extends RDF Schema to a full-fledged knowledge representation language for the Web
  - Logical expressions (and, or, not)
  - (in)equality
  - local properties
  - required/optional properties
  - required values
  - enumerated classes
  - symmetry, inverse





## Design Goals for OWL

- **Shareable**
- **Changing** over time
- **Interoperability**
- **Inconsistency** detection
- Balancing **expressivity and complexity**
- **Ease of use**
- Compatible with **existing standards**
- **Internationalisation**





## Requirements for OWL

- Ontologies are **object on the Web**
- with **their own meta-data**, versioning, etc...
- Ontologies are **extendable**
- They contain **classes, properties, data-types, range/domain, individuals**
- **Equality** (for classes, for individuals)
- **Classes as instances**
- **Cardinality** constraints
- **XML** syntax

## Objectives for OWL

Objectives:

- layered language
- complex datatypes
- digital signatures
- decidability (in part)
- local unique names (in part)

Disregarded:

- default values
- closed world option
- property chaining
- arithmetic
- string operations
- partial imports
- view definitions
- procedural attachments





## Language layers of OWL

- OWL Lite
  - Classification hierarchy
  - Simple constraints
- OWL DL
  - Maximal expressiveness
  - While maintaining tractability
  - Standard formalisation in a DL
- OWL Full
  - Very high expressiveness
  - Losing tractability
  - All syntactive freedom of RDF (self-modifying)





## Features of OWL language layers

- OWL Lite
  - (sub)classes, individuals
  - (sub)properties, domain, range
  - conjunction
  - (in)equality
  - cardinality 0/1
  - datatypes
  - inverse, transitive, symmetric properties
  - someValuesFrom
  - allValuesFrom
- OWL DL
  - Negation
  - Disjunction
  - Full cardinality
  - Enumerated types
  - hasValue
- OWL Full
  - Meta-classes
  - Modify language





## OWL Full

- **No restriction on use of vocabulary** (as long as legal RDF)
  - Classes as instances (and much more)
- **RDF style model theory**
  - Reasoning using FOL engine
  - Semantics should correspond to OWL DL for restricted KBs





## OWL DL

- Use of vocabulary restricted
  - Can't be used to do "nasty things" (e.g. modify OWL)
  - No classes as instances
  - Defined by abstract syntax
- Standard DL-based model theory
  - Direct correspondence with a DL
  - Reasoning via DL engines





## OWL Lite

- No explicit negation or union
- Restricted cardinality (0/1)
- No nominals (oneOf)
- DL-based semantics
  - Reasoning via DL engines (+datatypes)
- Semantically, only small restriction on OWL DL
  - No nominals
  - No arbitrary cardinality

## OWL and Description Logics

- OWL Lite corresponds to the DL $\mathcal{SHIN}(\mathbf{D})$
  - Named classes ($A$)
  - Named properties ($P$)
  - Individuals ($C(o)$)
  - Property values ($P(o, a)$)
  - Intersection ($C \sqcap D$)
  - Union(!) ($C \sqcup D$)
  - Negation(!) ($\neg C$)
  - Existential value restrictions ($\exists P.C$)
  - Universal value restrictions ($\forall P.C$)
  - Unqualified number restrictions ($\geq nP$, $\leq nP$, $= nP$)
- OWL DL corresponds to the DL $\mathcal{SHOIN}(\mathbf{D})$
  - Property value ($\exists P.\{o\}$)
  - Enumeration ($\{o_1, ..., o_n\}$)





## OWL constructs

| OWL Construct | DL | Example |
|---|---|---|
| intersectionOf | $C_1 \sqcap ... \sqcap C_n$ | $Human \sqcap Male$ |
| unionOf | $C_1 \sqcup ... \sqcup C_n$ | $Doctor \sqcup Lawyer$ |
| complementOf | $\neg C$ | $\neg Male$ |
| oneOf | $\{o_1, ..., o_n\}$ | $\{john, mary\}$ |
| allValuesFrom | $\forall P.C$ | $\forall hasChild.Doctor$ |
| someValuesFrom | $\exists P.C$ | $\forall hasChild.Lawyer$ |
| value | $\exists P.\{o\}$ | $\exists citizenOf.USA$ |
| minCardinality | $\geq nP.C$ | $\geq 2 hasChild.Lawyer$ |
| maxCardinality | $\leq nP.C$ | $\leq 1 hasChild.Male$ |
| cardinality | $= nP.C$ | $= 1 hasParent.Female$ |

+ XML Schema datatypes: int, string, real, etc...





## OWL axioms

| OWL Axiom | DL | Example |
|---|---|---|
| SubClassOf | $C_1 \sqsubseteq C_2$ | $Human \sqsubseteq Animal \sqcap Biped$ |
| EquivalentClasses | $C_1 \equiv ... \equiv C_n$ | $Man \equiv Human \sqcap Male$ |
| SubPropertyOf | $P_1 \sqsubseteq P_2$ | $hasDaughter \sqsubseteq hasChild$ |
| EquivalentProperties | $P_1 \equiv ... \equiv P_n$ | $cost \equiv price$ |
| SameIndividual | $o_1 = ... = o_n$ | $President\_Bush = G\_W\_Bush$ |
| DisjointClasses | $C_i \sqsubseteq \neg C_j$ | $Male \sqsubseteq \neg Female$ |
| DifferentIndividuals | $o_i \neq o_j$ | $john \neq peter$ |
| inverseOf | $P_1 \equiv P_2^-$ | $hasChild \equiv hasParent^-$ |
| Transitive | $P^+ \sqsubseteq P$ | $ancestor^+ \sqsubseteq ancestor$ |
| Symmetric | $P \equiv P^-$ | $connectedTo \equiv connectedTo^-$ |





## More on OWL species

- OWL Full is **not** a Description Logic
- OWL Lite has strong syntactic restrictions, but only limites semantics restrictions, compared with OWL DL
  - Negation can be encoded using disjointness
  - With negation an conjunction, you can encode disjunction

Class(C complete unionOf(B C))

is equivalent to:

DisjointClasses(notB B)
DisjointClasses(notC C)
Class(notBandnotC complete notB notC)
DisjointClasses(notBandnotC BorC)
Class(C complete notBandnotC)





## More on layering

- For an OWL DL-restricted KB, OWL Full semantics is **not** equivalent to OWL DL semantics

John friend Susan .

OWL Full entails:

John rdf:type owl:Thing . Susan rdf:type owl:Thing . friend rdf:type owl:ObjectProperty .

John rdf:type _:x . _:x owl:onProperty friend . _:x owl: minCardinality "1"^^xsd:nonNegativeInteger .





## Syntaxes of OWL

- RDF
  - Official exchange syntax
  - Hard for humans
  - RDF parsers are hard to write!
- UML
  - Large user base
- XML
  - Not the RDF syntax
  - Better for humans
  - More XML than RDF tools available
- Abstract syntax
  - Not defined for OWL Full
  - Human readable

# OWL in RDF/XML

Example from [OwlGuide]:

```
<!ENTITY vin
"http://www.w3.org/TR/2004/REC−owl−guide−20040210/wine#" >
<!ENTITY food
"http://www.w3.org/TR/2004/REC−owl−guide−20040210/food#" > ...
<rdf:RDF
xmlns:vin="http://www.w3.org/TR/2004/REC−owl−guide−20040210/wine#"
xmlns:food="http://www.w3.org/TR/2004/REC−owl−guide−20040210/food#"
 ... >

<owl:Class rdf:ID="Wine"> <rdfs:subClassOf
rdf:resource="&food;PotableLiquid"/> <rdfs:label
xml:lang="en">wine</rdfs:label> <rdfs:label
xml:lang="fr">vin</rdfs:label> ...   </owl:Class>

<owl:Class rdf:ID="Pasta"> <rdfs:subClassOf
rdf:resource="#EdibleThing" /> ... </owl:Class> </rdf:RDF>
```





# OWL Abstract syntax

Class( professor partial ) Class( associateProfessor partial academicStaffMember)

DisjointClasses ( associateProfessor assistantProfessor )
DisjointClasses ( professor associateProfessor )

Class( faculty complete academicStaffMember)





# OWL Abstract syntax

In DL syntax:

$\text{associateProfessor} \sqsubseteq \text{academicStaffMember}$
$\text{associateProfessor} \sqsubseteq \neg\, \text{assistantProfessor}$
$\text{professor} \sqsubseteq \neg\, \text{associateProfessor}$
$\text{faculty} \equiv \text{academicStaffMember}$





# More examples

DatatypeProperty(age range(xsd:nonNegativeInteger))
ObjectProperty( lecturesIn )

ObjectProperty(isTaughtBy domain(course) range(academicStaffMember))
SubPropertyOf(isTaughtBy involves)

ObjectProperty(teaches inverseOf(isTaughtBy) domain(academicStaffMember) range(course))

EquivalentProperties ( lecturesIn teaches)

ObjectProperty(hasSameGradeAs Transitive Symmetric domain(student) range(student))





# More examples

In DL syntax:

$\top \sqsubseteq \forall age.xsd : nonNegativeInteger$
$\top \sqsubseteq \forall isTaughtBy^{-}.course$
$\top \sqsubseteq \forall isTaughtBy.academicStaffMember$
$isTaughtBy \sqsubseteq involves$
$teaches \equiv isTaughtBy^{-}$
$\top \sqsubseteq \forall teaches^{-}.academicStaffMember$
$\top \sqsubseteq \forall teaches.course$
$lecturesIn \equiv teaches$
$hasSameGradeAs^{+} \sqsubseteq hasSameGradeAs$
$hasSameGradeAs \equiv hasSameGradeAs^{-}$
$\top \sqsubseteq \forall hasSameGradeAs^{-}.student$
$\top \sqsubseteq \forall hasSameGradeAs.student$





# More examples

Individual (949318 type( lecturer ))

Individual (949352 type(academicStaffMember) value(age "39"^^&xsd;integer))

ObjectProperty(isTaughtBy Functional)

Individual (CIT1111 type(course) value(isTaughtBy 949352) value(isTaughtBy 949318))

DifferentIndividuals (949318 949352) DifferentIndividuals (949352 949111 949318)

## More examples

In DL syntax:

$949318 : \mathit{lecturer}$
$949352 : \mathit{academicStaffMember}$
$\langle 949352, "39"\hat{}\hat{}\&xsd;\mathit{integer}\rangle : \mathit{age}$
$\top \sqsubseteq \leq 1 \mathit{isTaughtBy}$
$\mathit{CIT}1111 : \mathit{course}$
$\langle \mathit{CIT}1111, 949352\rangle : \mathit{isTaughtBy}$
$\langle \mathit{CIT}1111, 949318\rangle : \mathit{isTaughtBy}$
$949318 \neq 949352$
$949352 \neq 949111$
$949111 \neq 949318$
$949352 \neq 949318$





## More examples

Class( firstYearCourse partial restriction (isTaughtBy allValuesFrom ( Professor )))

Class(mathCourse partial restriction (isTaughtBy hasValue (949352)))

Class(academicStaffMember partial restriction (teaches someValuesFrom (undergraduateCourse)))

Class(course partial restriction (isTaughtBy minCardinality (1)))

Class(department partial restriction (hasMember minCardinality(10)) restriction (hasMember maxCardinality(30)))





## More examples

In DL syntax:

$\mathit{firstYearCourse} \sqsubseteq \forall \mathit{isTaughtBy}.\mathit{Professor}$
$\mathit{mathCourse} \sqsubseteq \exists \mathit{isTaughtBy}.\{949352\}$
$\mathit{academicStaffMember} \sqsubseteq \exists \mathit{teaches}.\mathit{undergraduateCourse}$
$\mathit{course} \sqsubseteq \geq 1 \mathit{isTaughtBy}$
$\mathit{department} \sqsubseteq \geq 10 \mathit{hasMember} \sqcap \leq 30 \mathit{hasMember}$





## More examples

Class(course partial complementOf(staffMember))

Class(peopleAtUni complete unionOf(staffMember student))

Class(facultyInCS complete intersectionOf( faculty restriction (belongsTo hasValue (CSDepartment))))

Class(adminStaff complete intersectionOf( staffMember complementOf(unionOf(faculty techSupportStaff))))





## More examples

In DL syntax:

$\mathit{course} \sqsubseteq \neg \mathit{staffMember}$
$\mathit{peopleAtUni} \equiv \mathit{staffMember} \sqcup \mathit{student}$
$\mathit{facultyInCS} \equiv \mathit{faculty} \sqcap \exists \mathit{belongsTo}.\{\mathit{CSDepartment}\}$
$\mathit{adminStaff} \equiv \mathit{staffMember} \sqcap \neg(\mathit{faculty} \sqcup \mathit{techSupportStaff})$





## Summary

Limitations of RDFS
- Expressive limitations
- Problems with layering

Web Ontology Language OWL
- Design of OWL
- OWL Layering
- OWL and Description Logics
- OWL Syntaxes

## Required reading


- OWL Guide: `http://www.w3.org/TR/owl-guide/`
- Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. **Journal of Web Semantics**, 1(1):7, 2003.


## Further reading


- Semantic Web Primer, Chapter 4
- I. Horrocks and P. F. Patel-Schneider. Reducing OWL entailment to description logic satisfiability. In **Proc. of the ISWC2003**.
- Jos de Bruijn: **Using Ontologies. Enabling Knowledge Sharing and Reuse on the Semantic Web.** DERI Technical Report DERI-2003-10-29, 2003. `http://www.deri.org/publications/techpapers/documents/DERI-TR-2003-10-29.pdf`
- OWL Reference: `http://www.w3.org/TR/owl-ref/`