Semantic Web Languages: RDF, RDF Schema, OWL

Frank van Harmelen
Vrije Universiteit Amsterdam

Shared content-vocabularies: Ontologies

Formal, ← machine processable

explicit specification ← concepts, properties, relations, functions

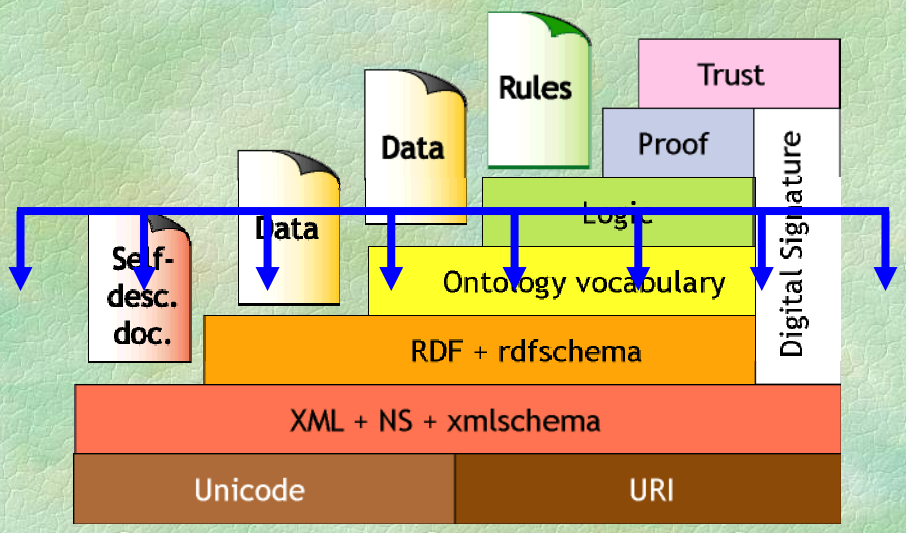
of a shared ← Consensual knowledge

conceptualisation ← Abstract model of some domain

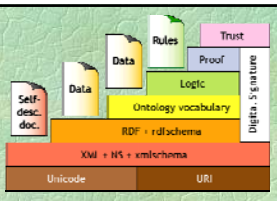
What's inside an ontology?

- Classes + class-hierarchy
- instances
- slots/values
- inheritance (multiple? defaults?)
- restrictions on slots (type, cardinality)
- properties of slots (symm., trans., ...)
- relations between classes (disjoint, covers)
- reasoning tasks: classification, subsumption

Stack of languages



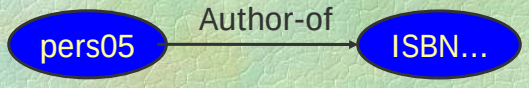
Stack of languages



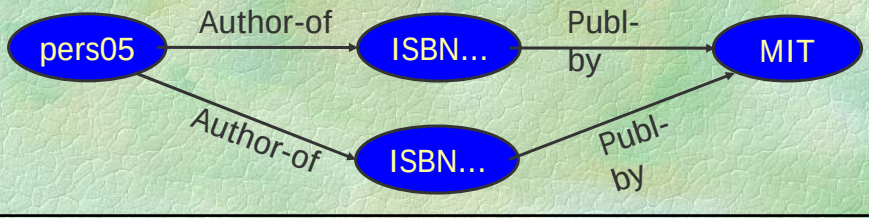
- **XML**:
  - Surface syntax, no semantics
- **XML Schema**:
  - Describes structure of XML documents
- **RDF**:
  - Datamodel for “relations” between “things”
- **RDF Schema**:
  - RDF Vocabulary Definition Language
- **OWL**:
  - A more expressive Vocabulary Definition Language

Bluffer’s guide to RDF (1)

- **Object ->Attribute-> Value** triples



- objects are **web-resources**
- Value is again an Object:
  - triples can be **linked**
  - data-model = graph



Bluffer's guide to RDF (2)

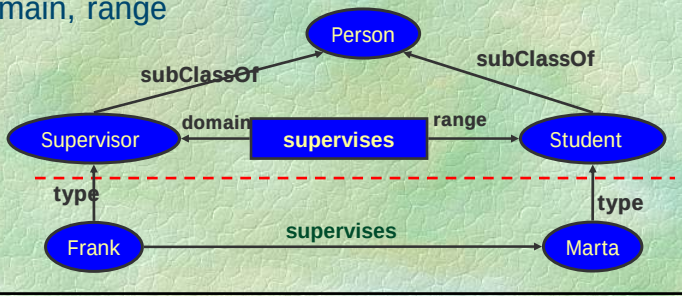
- Every identifier is a URL
  = world-wide unique naming!
- Has XML syntax

```
<rdf:Description rdf:about="#pers05">
  <authorOf>ISBN...</authorOf>
</rdf:Description>
```

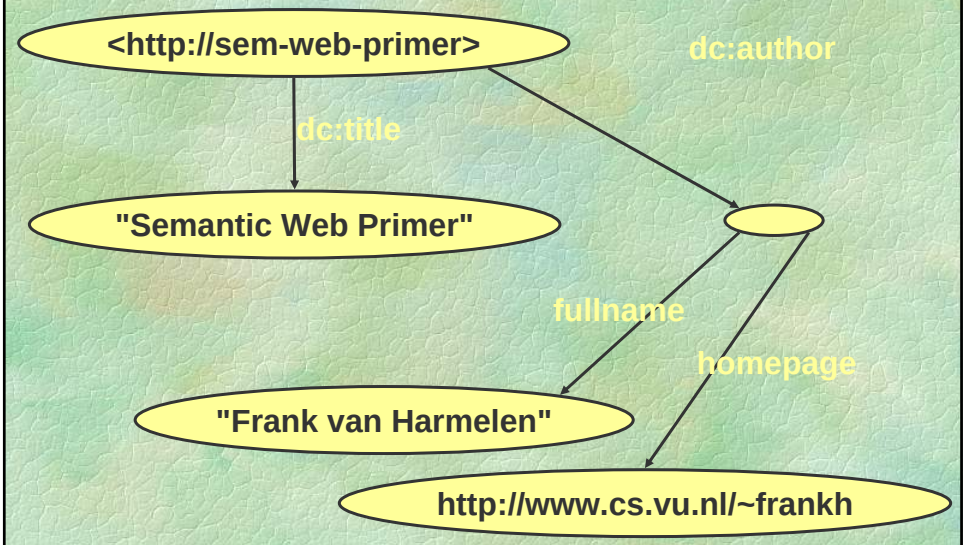
- Any statement can be an object
  - graphs can be **nested**

What does RDF Schema add?

- Defines **vocabulary** for RDF
- Organizes this vocabulary in a **typed hierarchy**
  - Class, subClassOf, type
  - Property, subPropertyOf
  - domain, range



RDF(S) syntax: graphics



RDF(S) syntax: XML

```
<rdf:RDF>
  <rdf:Description rdf:about="http://sem-web-primer"
   dc:title="Semantic Web Primer">
    <dc:author>
      <rdf:Description fullname="Frank van Harmelen">
        <homePage rdf:resource="http://www.cs.vu.nl/~frankh"/>
      </rdf:Description>
    </dc:author>
  </rdf:Description>
</rdf:RDF>
```

RDF(S) syntax: Turtle

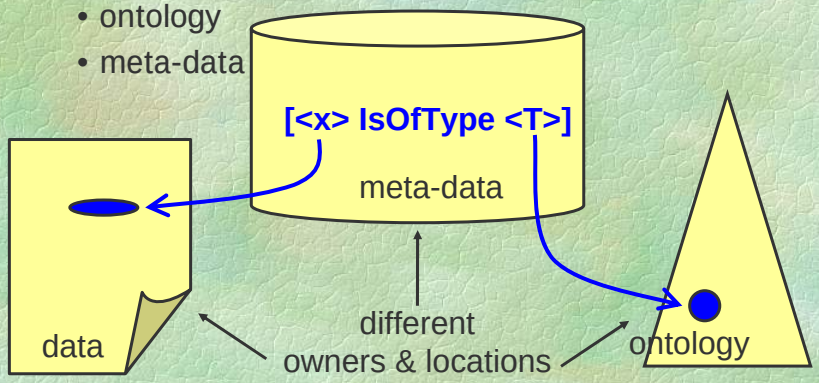
```
<http://sem-web-primer>
  dc:title "Semantic Web Primer" ;
  dc:author [
    fullname "Frank van Harmelen";
    homePage <http://www.cs.vu.nl/~frankh>
  ] .
```

RDF(S) / XML relationship

- XML is a just a syntax for RDF(S)
  - (one of many)
- RDF(S) assigns meaning to some terms
  - (XML doesn't)
- This allows greater interoperability:
  - tools/tools
  - thesaurus/thesaurus
  - tools/thesaurus

Crucial RDF(S) property:

- All identifiers are URL's
  - Allows total decoupling of
    - data
    - ontology
    - meta-data



RDF(S) is a logic

- only ground binary predicates
- only conjunction
- limited form of ∃ (anonymous individuals)
- implication between ∀ predicates (subsumption)

RDF(S) is a logic

- (Non-standard) model-theory (Pat Hayes)
  - E.g: classes members of themselves
- Simple model-theoretic properties:
  - Entailment,
  - skolemisation,
  - (strong) Herbrand property,
  - interpolation theorem
- Axiomatisations (Stanford, Essen, Lyon)
  - Some of these machine verified

RDF(S) have a (very small) formal semantics

- Defines what other statements are **implied** by a given set of RDF(S) statements
- Ensures mutual **agreement on minimal content** between parties without further contact
- Sound & complete "entailment rules"
- Very **simple to compute** (and not explosive in practice, compute full closure of > 10^9 statements)

Things RDF(S) can't do

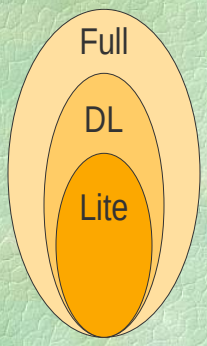
- (in)equality
- enumeration
- boolean algebra
  - Union, complement
- number restrictions
  - Single-valued/multi-valued
  - Optional/required values
- inverse, symmetric, transitive
- ...

Beyond RDF: OWL

- **OWL** extends RDF Schema to a full-fledged knowledge representation language.
  - logical expressions
  - data-typing
  - cardinality
  - quantifiers
- **OWL** is simply a Description Logic SHOIN(D) with an RDF/XML syntax

Layered language

- **OWL Lite:**
  - Classification hierarchy
  - Simple constraints
- **OWL DL:**
  - Maximal expressiveness
  - While maintaining tractability
  - Standard formalisation
- **OWL Full:**
  - Very high expressiveness
  - Loosing tractability
  - Non-standard formalisation
  - All syntactic freedom of RDF (self-modifying)

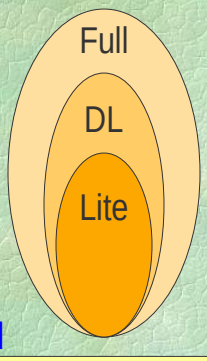


Syntactic layering
Semantic layering

Language Layers

- **OWL Light**
  - (sub)classes, individuals
  - (sub)properties, domain, range
  - conjunction

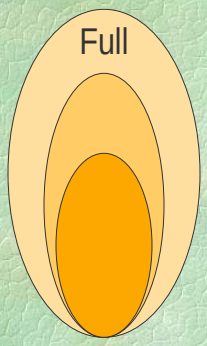
  RDF Schema
  - (in)equality
  - cardinality 0/1
  - datatypes
  - inverse, transitive, symmetric
  - hasValue
  - someValuesFrom
  - allValuesFrom



- **OWL DL**
  - Negation
  - Disjunction
  - Full Cardinality
  - Enumerated types
- **OWL Full**
  - Allow meta-classes etc

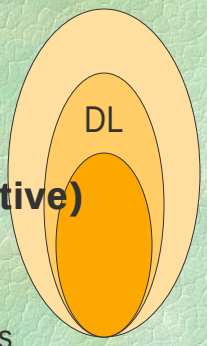
Language Layers: Full

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



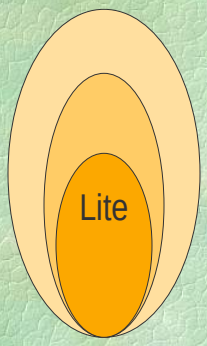
Language Layers: DL

- **Use of OWL vocabulary restricted**
  - Can't be used to do "nasty things" (I.e., modify OWL)
  - No classes as instances
  - Defined by abstract syntax
- **Standard FOL model theory (definitive)**
  - Direct correspondence with FOL
  - Reasoning via DL engines
  - Reasoning for full language via FOL engines
    - No need for axiomatisation (unlike full)
    - Would need built in datatypes for performance



Language Layers: Lite

- No explicit negation or union
- Restricted cardinality (0/1)
- No nominals (oneOf)
- Semantics as in DL
  - Reasoning via DL engines (+datatypes)
    - E.g., FaCT, RACER, Cerebra

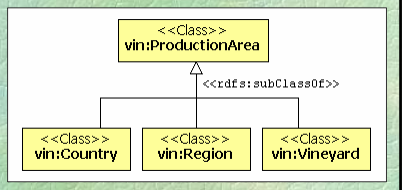


OWL also has a formal semantics

- Defines what other statements are **implied** by a given set of statements
- Ensures **mutual agreement** on content (both **minimal and maximal**) between parties without further contact
- Can be used for integrity/**consistency checking**
- Hard to compute (and *rarely/sometime/always explosive in practice*)

Different syntaxes for OWL

- **RDF**
  - Official exchange syntax
  - Hard for humans
- **UML**
  - Large user base
  - Masahiro Hori, IBM Japan
- **XML**
  - Not the RDF syntax
  - Better for humans
  - Masahiro Hori, IBM Japan
- **"Abstract" syntax**
  - Human readable/writeable




```
Class          Wine
  subClassOf                PotableLiquid

   Restriction
     onProperty             #madeFromGrape
     minCardinality  1

   Restriction
     onProperty             hasMaker
     allValuesFrom            #Winery

  equivalentClass           Vin
```

Things OWL doesn't do

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

Illustrating OWL in its abstract syntax

```
Class(professor partial)
Class(associateProfessor partial academicStaffMember)

DisjointClasses(associateProfessor assistantProfessor)
DisjointClasses(professor associateProfessor)

Class(faculty complete academicStaffMember)
```

```
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))
```

```
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)
```

```
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)))
```

```
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))))
```

```
EnumeratedClass(weekdays
                  Monday
                  Tuesday
                  Wednesday
                  Thursday
                  Friday
                  Saturday
                  Sunday)
```