



Knowledge Graphs

Lecture 4 – Ontologies as Key to Knowledge Representation

4.5 Unlocking the Potential of OWL

Prof. Dr. Harald Sack

FIZ Karlsruhe – Leibniz Institute for Information Infrastructure

AIFB – Karlsruhe Institute of Technology

Autumn 2023



FIZ Karlsruhe

Leibniz-Institut für Informationsinfrastruktur

Knowledge Graphs

Lecture 4: Ontologies as Key to Knowledge Representation

4.1 From Aristotle to AI: Exploring Ontologies in Computer Science

4.2 The Crucial Role of Mathematical Logic

Excursion 5: Essential Logics in a Nutshell

Excursion 6: Description Logics

4.3 The Web Ontology Language OWL

4.4 From simple to complex: Scaling up with OWL

4.5 Unlocking the Potential of OWL

The Semantic Web Technology Stack (not a piece of cake...)

Most apps use only a subset of the stack

Querying allows fine-grained data access

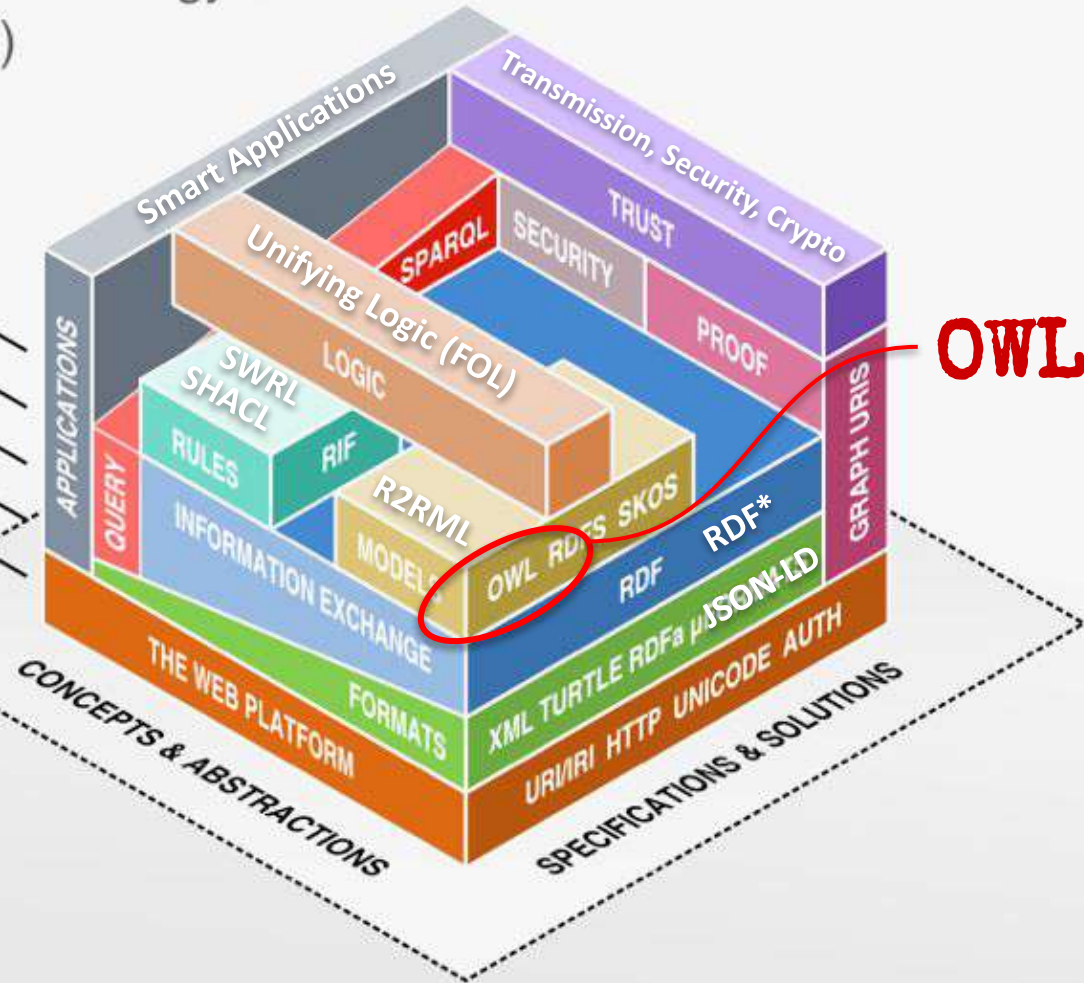
Standardized information exchange is key

Formats are necessary, but not too important

The Semantic Web is based on the Web

Linked Data uses a small
selection of technologies

LINKED DATA



OWL2 Property Relationships

- **Property hierarchies** can be created via specializations: `rdfs:subPropertyOf`
- **Inverse properties** are defined via `owl:inverseOf`
- **Identical properties** are defined via `owl:equivalentProperty`

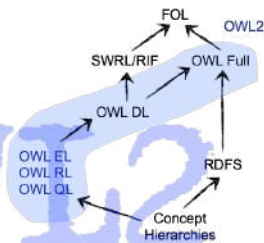
```

:madeOf a owl:ObjectProperty ;
    rdfs:subPropertyOf :consistsOf .

:reads a owl:ObjectProperty ;
    owl:inverseOf :isReadBy .

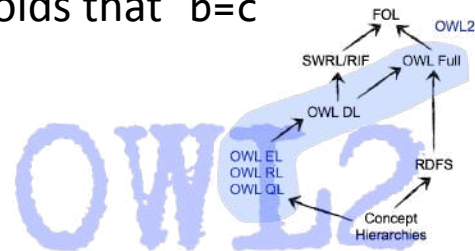
:composedOf a owl:ObjectProperty ;
    owl:equivalentProperty :consistsOf .
  
```

`isMadeOf` \sqsubseteq `consistsOf`
`reads-` \equiv `isReadBy`
`composedOf` \equiv `consistsOf`



OWL2 Property Relationships

- `owl:TransitiveProperty`
e.g.: if `partOf(a,b)` and `partOf(b,c)` then it holds that `partOf(a,c)`
- `owl:SymmetricProperty`
e.g.: if `neighborOf(a,b)` then it holds that `neighborOf(b,a)`
- `owl:FunctionalProperty`
e.g.: if `hasMother(a,b)` and `hasMother(a,c)` then it holds that `b=c`
- `owl:InverseFunctionalProperty`
e.g.: if `isMotherOf(b,a)` and `isMotherOf(c,a)` then it holds that `b=c`

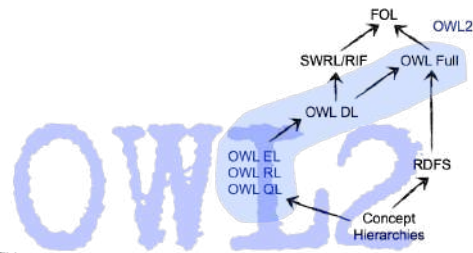


OWL2 Transitive Properties

```
:isPublishedBefore a owl:ObjectProperty ;  
                    a owl:TransitiveProperty ;  
                    rdfs:domain owl:Book ;  
                    rdfs:range owl:Book .  
  
:AnimalFarm a owl:Book ;  
             :isPublishedBefore :NineteenEightyfour .  
  
:BraveNewWorld a owl:Book ;  
               :isPublishedBefore :AnimalFarm .
```

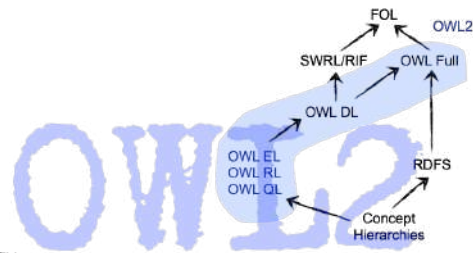
Via inference it can be entailed that:

```
:BraveNewWorld :isPublishedBefore :NineteenEightyfour .
```



More Property Relationships

- **Asymmetric properties** via `owl:AsymmetricProperty`
e.g.: if it holds that `isLeftOf(a,b)`
then it is not possible that `isLeftOf(b,a)`
- **Reflexive properties** via `owl:ReflexiveProperty`
e.g.: `isRelatedTo(x,x)`
- **Irreflexive properties** via `owl:IrreflexiveProperty`
e.g.: if `isParentOf(x,y)` then $x \neq y$



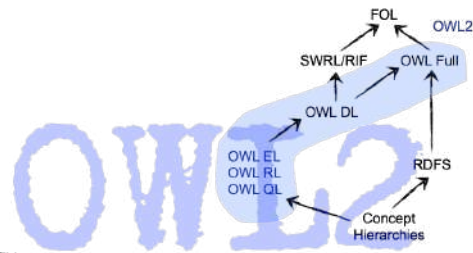
Disjunctive Properties

- Two properties R and S are **disjunctive**,
if two individuals x,y are never related via both properties

```
:hasParent a owl:ObjectProperty ;  
          owl:propertyDisjointWith :hasChild .
```

- Shortcut** for several **disjunctive properties**

```
[] rdf:type owl:AllDisjointProperties ;  
   owl:members  
     ( :hasParent  
       :hasChild  
       :hasGrandchild ) .
```

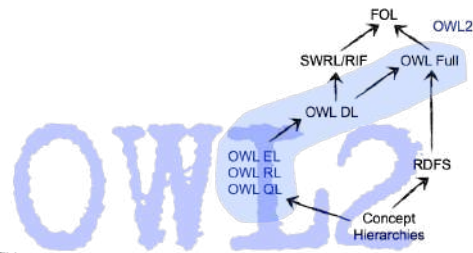


OWL2 Negation

Two individuals can explicitly be defined to be **not related with each other** via a given property:

`-isBrother(GeorgeOrwell,IsaacAsimov)`

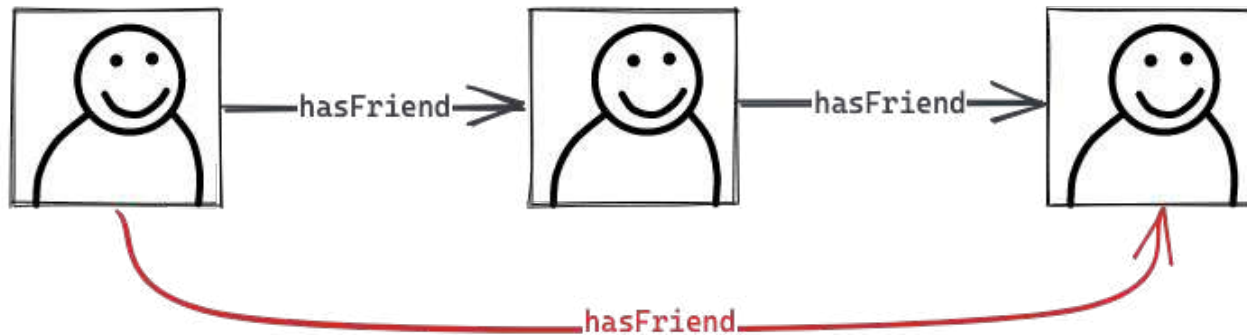
```
[] rdf:type owl:NegativePropertyAssertion ;  
  owl:sourceIndividual :GeorgeOrwell ;  
  owl:assertionProperty :isBrother ;  
  owl:targetIndividual :IsaacAsimov .
```



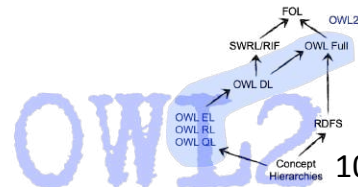
OWL2 Property Chaining

Complex Roles (properties) can be constructed from simple roles (**RBox**):

- "The friends of my friends are also my friends."



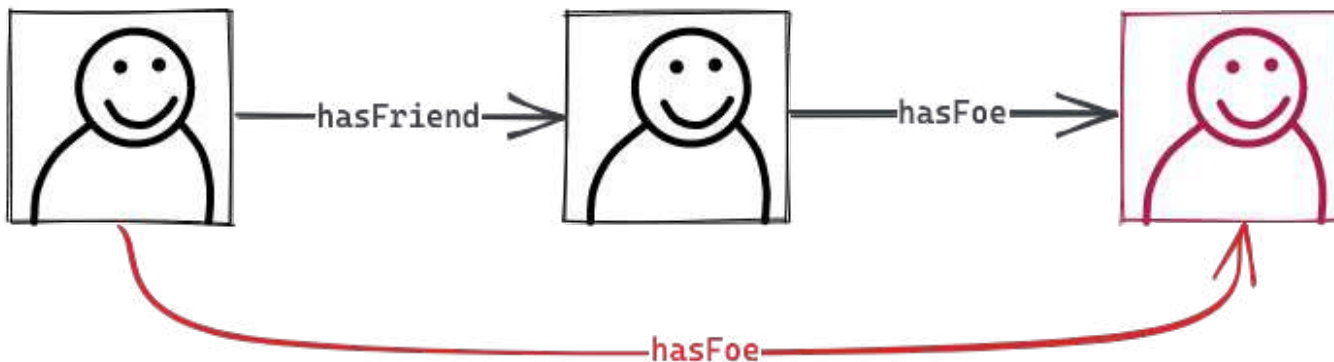
- `hasFriend a owl:TransitiveProperty .`



OWL2 Property Chaining

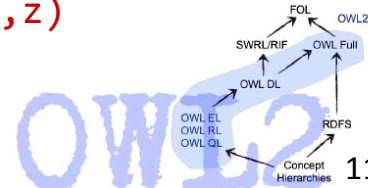
- But what about: *"The foes of my friends are also my foes."*

cannot be expressed via Transitivity



- In FOL it can be expressed as a rule (axiom):

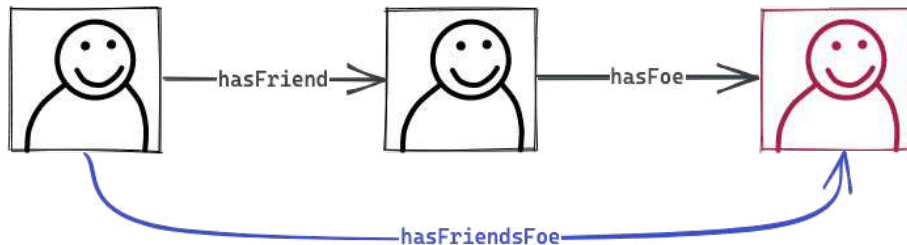
$$\forall x,y,z: \text{hasFriend}(x,y) \wedge \text{hasFoe}(y,z) \rightarrow \text{hasFriendsFoe}(x,z)$$



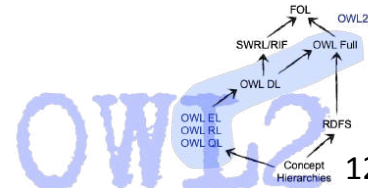
OWL2 Property Chaining

- General Role Inclusion (property chaining)

```
:hasFriendsFoe a owl:ObjectProperty ;  
  owl:propertyChainAxiom ( :hasFriend :hasFoe ) .
```



- Not allowed for datatype properties



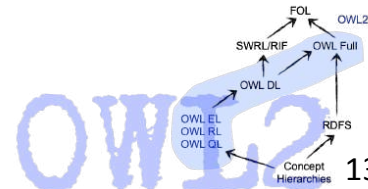
OWL2 Qualified Number Restriction

- Class constructors with **number restrictions on properties connected with a range constraint**:

SuccessfulAuthor $\sqsubseteq \geq 1$ notableWork.Bestseller

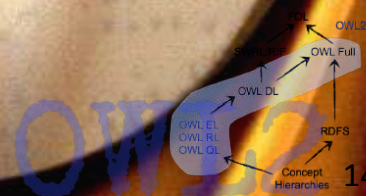
```
:SuccessfulAuthor a owl:Class;  
  rdfs:subClassOf [  
    a owl:Restriction ;  
    owl:onProperty :notableWork ;  
    owl:minQualifiedCardinality "1"^^xsd:nonNegativeInteger ;  
    owl:onClass :Bestseller ] .
```

- `owl:maxQualifiedCardinality`, `owl:minQualifiedCardinality`,
`owl:qualifiedCardinality`



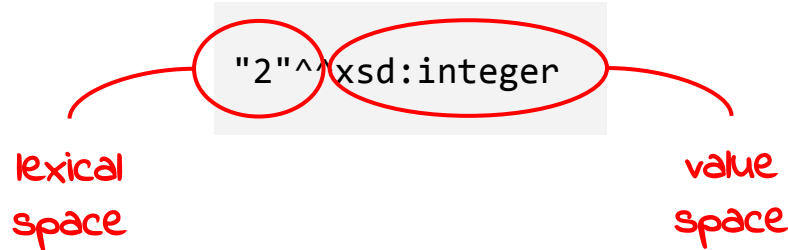

```
:Philosopher a owl:Class ;
  rdfs:subClassOf [
    a owl:Restriction ;
    owl:onProperty :knows ;
    owl:hasSelf "true"^^xsd:boolean ] .
```

[3]

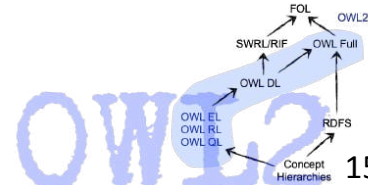


OWL2 Datatypes

- OWL also supports datatypes through the definition of a datatype map.
- As in RDF(S), a datatype in OWL is identified by an **IRI** and is defined in terms of a **mapping from a lexical space** (e.g., the string “2”) **to a value space** (e.g., the number two).



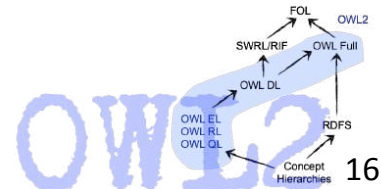
- Not all XML Schema datatypes are supported, e.g. `xsd:date`, `xsd:time`, `xsd:duration`, the Gregorian datatypes like `xsd:gYear`, and syntactic XML datatypes like `xsd:QName` have been omitted.



OWL2 Datatype Facet Restriction

- New OWL2 datatypes: `owl:real` and `owl:rational`
- Unlike RDF(S), each datatype in OWL is additionally associated with a **facet space**, which allows for deriving new datatypes by **restricting facets of old datatypes**.

```
:MinorAge a rdfs:Datatype ;  
  owl:equivalentClass  
    [ a rdfs:Datatype ;  
      owl:onDatatype xsd:integer ;  
      owl:withRestrictions (  
        [ xsd:minInclusive 0 ]  
        [ xsd:maxExclusive 18 ] ) ] .
```



Next Lecture...

An owl wearing a white spacesuit with an American flag patch is floating in space. The owl has large yellow eyes and its wings are spread. In the background, the reddish-orange surface of Mars is visible, along with the blue and white Earth. A small satellite is also seen in the distance.

Ontology Engineering for Smarter Knowledge Graphs

Bibliographic References:

- Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, Sebastian Rudolph (eds., 2012), [OWL 2 Web Ontology Language Primer \(Second Edition\)](#), W3C Recommendation 11 December 2012
- Aidan Hogan (2020), [The Web of Data](#), Springer.
 - Chap. 5.4.2 Properties, pp. 200–215.
 - Chap. 5.4.4 Negation, 243–244.
 - Chap. 5.4.5 Datatypes, 244–250.

Picture References:

- [1] “A large owl watching over a crowded street in a Bladerunner like dystopian city environment.”, created via ArtBot, Deliberate, 2023, [CC-BY-4.0], <https://tinybots.net/artbot>
- [2] Benjamin Nowack, *The Semantic Web - Not a Piece of cake ...*, at bnode.org, 2009-07-08 , [CC BY 3.0], <https://web.archive.org/web/20220628120341/http://bnode.org/blog/2009/07/08/the-semantic-web-not-a-piece-of-cake>
- [3] “The Greek philosopher Heraclitus”, created via ArtBot, Ceipher Female Model, 2023, [CC-BY-4.0], <https://tinybots.net/artbot>
- [4] “A large owl in a space suit floating in deep space next to its spaceship over the surface of Mars.”, created via ArtBot, Deliberate, 2023, [CC-BY-4.0], <https://tinybots.net/artbot>