W3C Web Ontology Language: Introduction

KAIST, 2007년 4월 26일

장민수 <minsu@etri.re.kr>, ETRI 지식및추론연구팀

Contents

- What is OWL?
- OWL vs DL
- OWL Organization
- OWL Vocabulary
- OWL Inference
- Beyond OWL
- Usage & Tools

What is OWL? (1)

OWL = **W**eb **O**ntology **L**anguage

- A Data Representation Language for the Vision of the **Web of Data**, the Semantic Web.
- A **Language** Recommended in 2004 By **W3C** for Representing **Ontologies** in an **Interchangeable Format**
- A **Standard Language** for Specifying **Description Logic Knowledge Base**

"Knowledge Representation goes Global" by TBL

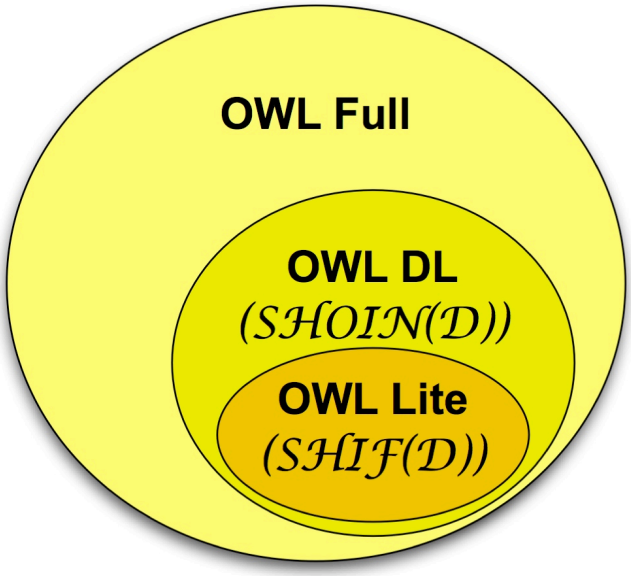
What is OWL? (2)

Ontology Definitions

- **Explicit** specification of a conceptualization according to (Gruber, 1993).
- **Formal** specification of a **shared** conceptualization according to (Borst, 1997)
- formal, explicit specification of a shared conceptualization of a **domain of interest** according to (Handschuh et al, 2001)
- ontologies enable to describe conceptual vocabulary **shared by a community** in an organization (Dieng et al, 2001)
- Ontologies aim to capture consensual knowledge in a generic way, and they may be **reused and shared** across applications and by groups of people (Gómez Pérez et al, 2004)

What is OWL? (3)

- OWL is composed of three sublanguages: OWL Lite, DL, and Full
- OWL Lite/DL corresponds to Description Logics - each of which similar to SHIF and SHOIN(D).
- Every ID is URI in OWL.

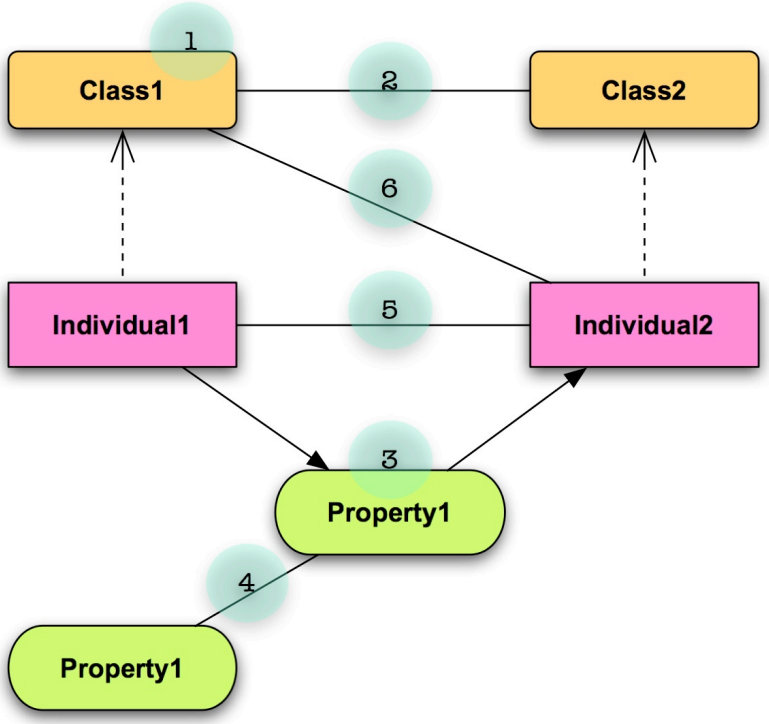


OWL vs DL (OWL DL is SHOIN(D)) (1)

Terms

DL	OWL
Concepts	Classes
Roles	Properties
Constants	Individuals

Knowledge Structure



OWL vs DL (OWL DL is SHOIN(D)) (2)

Class Descriptions (1,2,6)

Descriptions (C)		
A (URI reference)	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
`owl:Thing`	$\top$	$\texttt{owl:Thing}^{\mathcal{I}} = \Delta^{\mathcal{I}}$
`owl:Nothing`	$\bot$	$\texttt{owl:Nothing}^{\mathcal{I}} = \{\}$
`intersectionOf(`C_1 C_2 ...`)`	$C_1 \sqcap C_2$	$(C_1 \sqcap D_1)^{\mathcal{I}} = C_1^{\mathcal{I}} \cap D_2^{\mathcal{I}}$
`unionOf(`C_1 C_2 ...`)`	$C_1 \sqcup C_2$	$(C_1 \sqcup C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
`complementOf(`C`)`	$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
`oneOf(`o_1 ...`)`	$\{o_1, \ldots\}$	$\{o_1, \ldots\}^{\mathcal{I}} = \{o_1^{\mathcal{I}}, \ldots\}$
`restriction(`R `someValuesFrom(`C`))`	$\exists R.C$	$(\exists R.C)^{\mathcal{I}} = \{x \mid \exists y.\langle x,y\rangle \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}$
`restriction(`R `allValuesFrom(`C`))`	$\forall R.C$	$(\forall R.C)^{\mathcal{I}} = \{x \mid \forall y.\langle x,y\rangle \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
`restriction(`R `hasValue(`o`))`	$R : o$	$(\forall R.o)^{\mathcal{I}} = \{x \mid \langle x, o^{\mathcal{I}}\rangle \in R^{\mathcal{I}}\}$
`restriction(`R `minCardinality(`n`))`	$\geqslant n\, R$	$(\geqslant n\, R)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x,y\rangle \in R^{\mathcal{I}}\}) \geqslant n\}$
`restriction(`R `minCardinality(`n`))`	$\leqslant n\, R$	$(\geqslant n\, R)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x,y\rangle \in R^{\mathcal{I}}\}) \leqslant n\}$
`restriction(`U `someValuesFrom(`D`))`	$\exists U.D$	$(\exists U.D)^{\mathcal{I}} = \{x \mid \exists y.\langle x,y\rangle \in U^{\mathcal{I}} \text{ and } y \in D^{\mathbf{D}}\}$
`restriction(`U `allValuesFrom(`D`))`	$\forall U.D$	$(\forall U.D)^{\mathcal{I}} = \{x \mid \forall y.\langle x,y\rangle \in U^{\mathcal{I}} \rightarrow y \in D^{\mathbf{D}}\}$
`restriction(`U `hasValue(`v`))`	$U : v$	$(U : v)^{\mathcal{I}} = \{x \mid \langle x, v^{\mathcal{I}}\rangle \in U^{\mathcal{I}}\}$
`restriction(`U `minCardinality(`n`))`	$\geqslant n\, U$	$(\geqslant n\, U)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x,y\rangle \in U^{\mathcal{I}}\}) \geqslant n\}$
`restriction(`U `maxCardinality(`n`))`	$\leqslant n\, U$	$(\leqslant n\, U)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x,y\rangle \in U^{\mathcal{I}}\}) \leqslant n\}$

OWL vs DL (OWL DL is SHOIN(D)) (3)

Class Relations (2,6)

Class(A partial C_1 ...C_n)	$A \sqsubseteq C_1 \sqcap \ldots \sqcap C_n$	$A^{\mathcal{I}} \subseteq C_1^{\mathcal{I}} \cap \ldots \cap C_n^{\mathcal{I}}$
Class(A complete C_1 ...C_n)	$A = C_1 \sqcap \ldots \sqcap C_n$	$A^{\mathcal{I}} = C_1^{\mathcal{I}} \cap \ldots \cap C_n^{\mathcal{I}}$
EnumeratedClass(A o_1 ...o_n)	$A = \{o_1, \ldots, o_n\}$	$A^{\mathcal{I}} = \{o_1^{\mathcal{I}}, \ldots, o_n^{\mathcal{I}}\}$
SubClassOf(C_1 C_2)	$C_1 \sqsubseteq C_2$	$C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
EquivalentClasses(C_1 ...C_n)	$C_1 = \ldots = C_n$	$C_1^{\mathcal{I}} = \ldots = C_n^{\mathcal{I}}$
DisjointClasses(C_1 ...C_n)	$C_i \sqcap C_j = \bot, i \neq j$	$C_i^{\mathcal{I}} \cap C_j^{\mathcal{I}}\{\}, i \neq j$
Datatype(D)		$D^{\mathcal{I}} \subseteq \Delta_{\mathbf{D}}^{\mathcal{I}}$

OWL vs DL (OWL DL is SHOIN(D)) (4)

Property Descriptions & Relations (3,4)

DatatypeProperty(U super(U_1)...super(U_n)	$U \sqsubseteq U_i$	$U^{\mathcal{I}} \subseteq U_i^{\mathcal{I}}$
domain(C_1) ...domain(C_m)	$\geqslant 1\, U \sqsubseteq C_i$	$U^{\mathcal{I}} \subseteq C_i^{\mathcal{I}} \times \Delta_{\mathbf{D}}^{\mathcal{I}}$
range(D_1) ...range(D_l)	$\top \sqsubseteq \forall U.D_i$	$U^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times D_i^{\mathcal{I}}$
[Functional])	$\top \sqsubseteq \leqslant 1\, U$	$U^{\mathcal{I}}$ is functional
SubPropertyOf(U_1 U_2)	$U_1 \sqsubseteq U_2$	$U_1^{\mathcal{I}} \subseteq U_2^{\mathcal{I}}$
EquivalentProperties(U_1 ...U_n)	$U_1 = \ldots = U_n$	$U_1^{\mathcal{I}} = \ldots = U_n^{\mathcal{I}}$
ObjectProperty(R super(R_1)...super(R_n)	$R \sqsubseteq R_i$	$R^{\mathcal{I}} \subseteq R_i^{\mathcal{I}}$
domain(C_1) ...domain(C_m)	$\geqslant 1\, R \sqsubseteq C_i$	$R^{\mathcal{I}} \subseteq C_i^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
range(C_1) ...range(C_l)	$\top \sqsubseteq \forall R.C_i$	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times C_i^{\mathcal{I}}$
[inverseOf(R_0]	$R = ({}^{-}R_0)$	$R^{\mathcal{I}} = (R_0^{\mathcal{I}})^{-}$
[Symmetric]	$R = ({}^{-}R)$	$R^{\mathcal{I}} = (R^{\mathcal{I}})^{-}$
[Functional]	$\top \sqsubseteq \leqslant 1\, R$	$R^{\mathcal{I}}$ is functional
[InverseFunctional]	$\top \sqsubseteq \leqslant 1\, R^{-}$	$(R^{\mathcal{I}})^{-}$ is functional
[Transitive])	$Tr(R)$	$R^{\mathcal{I}} = (R^{\mathcal{I}})^{+}$
SubPropertyOf(R_1 R_2)	$R_1 \sqsubseteq R_2$	$R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$
EquivalentProperties(R_1 ...R_n)	$R_1 = \ldots = R_n$	$R_1^{\mathcal{I}} = \ldots = R_n^{\mathcal{I}}$
AnnotationProperty(S)		

OWL vs DL (OWL DL is SHOIN(D)) (5)

Individual Descriptions (5)

Individual(o type(C_1) ...type(C_n) value(R_1 o_1)...value(R_n o_n) value(U_1 v_1)...value(U_n v_n))	$o \in C_i$ $\langle o, o_i \rangle \in R_i$ $\langle o, v_i \rangle \in U_i$	$o^{\mathcal{I}} \in C_i^{\mathcal{I}}$ $\langle o^{\mathcal{I}}, o_i^{\mathcal{I}} \rangle \in R_i^{\mathcal{I}}$ $\langle o^{\mathcal{I}}, v_i^{\mathcal{I}} \rangle \in U_i^{\mathcal{I}}$
SameIndividual(o_1 ...o_n)	$o_1 = \ldots = o_n$	$o_i^{\mathcal{I}} = o_j^{\mathcal{I}}$
DifferentIndividuals(o_1 ...o_n)	$o_i \neq o_j, i \neq j$	$o_i^{\mathcal{I}} \neq o_j^{\mathcal{I}}, i \neq j$

OWL DL vs OWL Lite

Why OWL Lite?

- OWL DL is too complex for casual users.
- Inferencing with OWL DL is very hard.

OWL Lite = Restricted OWL DL

- Unions and Complements are prohibited.
- Individuals cannot be used inside class descriptions or axioms (unionOf, hasValue are prohibited).
- Cardinalities are limited to 0 or 1.
- Embedded class descriptions should always be named classes or restrictions
- etc

OWL Organization

Layering

Syntax

- Abstract Syntax
- Various Concrete Syntax: RDF/XML, N3, NTriple,...

Semantics

- Direct Model-Theoretic Semantics
- RDF-Compatible Model-Theoretic Semantics

OWL Syntax: Abstract Syntax

```
Class(pp:sheep partial
  pp:animal
  restriction(pp:eats allValuesFrom pp:grass))
Class(pp:giraffe partial
  pp:animal
  restriction(pp:eats allValuesFrom pp:leaf))
Class(pp:cow partial
  pp:vegetarian)
Class(pp:mad+cow complete
  intersectionOf(
    pp:cow
    restriction(
      pp:eats
      someValuesFrom(
        intersectionOf(
          pp:brain
          restriction(
            pp:part_of
            someValuesFrom pp:sheep)
        )
      )
    )
  )
)
```

OWL Syntax: N3

```
pp:sheep a owl:Class;
   rdfs:subClassOf pp:animal;
   rdfs:subClassOf
     [a owl:Restriction;
      owl:onProperty pp:eats;
      owl:allValuesFrom pp:grass].
pp:giraffe a owl:Class;
   rdfs:subClassOf pp:animal;
   rdfs:subClassOf
     [a owl:Restriction;
      owl:onProperty pp:eats;
      owl:allValuesFrom pp:leaf].
```

OWL Syntax: RDF/XML

```
<owl:Class rdf:about="&pp;sheep">
 <rdfs:subClassOf rdf:resource="&pp;animal"/>
 <rdfs:subClassOf>
   <owl:Restriction>
     <owl:onProperty rdf:resource="&pp;eats"/>
     <owl:allValuesFrom rdf:resource="&pp;grass"/>
   </owl:Restriction>
 </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="&pp;giraffe">
 <rdfs:subClassOf rdf:resource="&pp;animal"/>
 <rdfs:subClassOf>
   <owl:Restriction>
     <owl:onProperty rdf:resource="&pp;eats"/>
     <owl:allValuesFrom rdf:resource="&pp;leaf"/>
   </owl:Restriction>
 </rdfs:subClassOf>
</owl:Class>
```

OWL Vocabulary: Namespaces

- owl = http://www.w3.org/2002/07/owl#
- 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#

OWL Vocabulary: Simple Class Description

OWL Vocabulary	Abstract Syntax	Semantics
`owl:Class`	Class(c)	$EC(c) \subseteq \Delta^I$
`owl:Thing`	Thing	$EC(owl:Thing) = \Delta^I$
`owl:Nothing`	Nothing	$EC(owl:Nothing) = \varnothing$

Examples

```
<owl:Class rdf:ID="Animal"/>
<owl:Thing rdf:ID="Lash"/>
```

owl:Nothing cannot have an individual! If an OWL document with an individual declared as of type owl:Nothing, the document is inconsistent.

OWL Vocabulary: Class-Class Relationship

OWL Vocabulary	Abstract Syntax	Semantics
owl:equivalentClass	Class(c complete d), EquivalentClasses(c d)	EC(c) = EC(d)
rdfs:subClassOf	Class(c partial d), SubClassOf(c d)	EC(c) ⊆ EC(d)

Examples

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

OWL does not support UNA(Unique Name Assumption). Resources with different IDs may be equivalent.

OWL Vocabulary: Class-Class Relationship

OWL Vocabulary	Abstract Syntax	Semantics
owl:intersectionOf	intersectionOf(c d)	EC(c) ∩ EC(d)
owl:unionOf	unionOf(c d)	EC(c) ∪ EC(d)

Examples

```
<owl:Class rdf:ID="WhiteBurgundy">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Burgundy" />
    <owl:Class rdf:about="#WhiteWine" />
  </owl:intersectionOf>
</owl:Class>
```

- EC(WhiteBurgundy) = EC(Burgundy) ∩ EC(WhiteWine)

```
<owl:Class rdf:ID="Fruit">
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#SweetFruit" />
    <owl:Class rdf:about="#NonSweetFruit" />
  </owl:unionOf>
</owl:Class>
```

- EC(Fruit) = EC(SweetFruit) ∪ EC(NonSweetFruit)

OWL Vocabulary: Class-Class Relationship

OWL Vocabulary	Abstract Syntax	Semantics
owl:complementOf	complementOf(c)	Δ^I - EC(c)
owl:disjointWith	DisjointClasses(c d)	EC(c) ∩ EC(d) = ∅

Examples

```
<owl:Class rdf:ID="ConsumableThing" />
<owl:Class rdf:ID="NonConsumableThing">
   <owl:complementOf rdf:resource="#ConsumableThing" />
</owl:Class>
```

- EC(ConsumableThing) ∩ EC(NonConsumableThing) = ∅
- EC(owl:Thing) - EC(ConsumableThing) = EC(NonConsumableThing)

```
<owl:Class rdf:ID="Pasta">
   <rdfs:subClassOf rdf:resource="#EdibleThing"/>
   <owl:disjointWith rdf:resource="#Meat"/>
   <owl:disjointWith rdf:resource="#Fowl"/>
   <owl:disjointWith rdf:resource="#Seafood"/>
   <owl:disjointWith rdf:resource="#Dessert"/>
   <owl:disjointWith rdf:resource="#Fruit"/>
</owl:Class>
```

- EC(Pasta) ⊆ (EC(EdibleThing) - EC(Meat) - EC(Fowl) - EC(Seafood) - EC(Dessert) - EC(Fruit))

Without declared or inferred owl:disjointWith, any two classes can overlap.

OWL Vocabulary: Class-Individual Relationship

OWL Vocabulary	Abstract Syntax	Semantics
owl:oneOf	oneOf(o1 o2)	{o1 o2}

Examples

```
<owl:Class rdf:ID="WineColor">
  <rdfs:subClassOf rdf:resource="#WineDescriptor"/>
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#White"/>
    <owl:Thing rdf:about="#Rose"/>
    <owl:Thing rdf:about="#Red"/>
  </owl:oneOf>
</owl:Class>
```

- EC(WineColor) = {White, Rose, Red}
- White ∈ WineColor ∧ Rose ∈ WineColor ∧ Red ∈ WineColor

OWL Vocabulary: Property Description

OWL Vocabulary	Abstract Syntax	Semantics
owl:ObjectProperty	ObjectProperty(p)	ER(p) ⊆ Δ^I X Δ^I
owl:DatatypeProperty	DatatypeProperty(p)	ER(p) ⊆ Δ^I X $\Delta_D{}^I$
rdfs:domain	domain(c)	ER(p) ⊆ EC(c) X $\Delta_D{}^I$
rdfs:range	range(c)	ER(p) ⊆ Δ^I X EC(c)

Examples

```
<owl:ObjectProperty rdf:ID="hasChild">
   <rdfs:domain rdf:resource="#Human">
   <rdfs:range rdf:resource="#Human"/>
</owl:ObjectProperty>
```

- ER(hasChild) ⊆ EC(Human) X EC(Human)

```
<owl:ObjectProperty rdf:ID="hasAge">
   <rdfs:domain rdf:resource="#Human">
   <rdfs:range rdf:resource="&xsd;positiveInteger"/>
</owl:ObjectProperty>
```

- ER(hasAge) ⊆ EC(Human) X xsd:positiveInteger

OWL Vocabulary: Property-Property Relationship

OWL Vocabulary	Abstract Syntax	Semantics
owl:equivalentProperty	EquivalentProperties(p q)	ER(p) = ER(q)
rdf:subPropertyOf	SubPropertyOf(p q), ObjectProperty(p super(q)), DatatypeProperty(p super(q))	ER(p) ⊆ ER(q)
owl:inverseOf	ObjectProperty(p inverseOf(q))	(x,y) ∈ ER(p) ↔ (y,x) ∈ ER(q)

Examples

```
<owl:ObjectProperty rdf:ID="hasChild">
   <owl:equivalentProperty rdf:resource="#hasOffspring">
   <owl:inverseOf rdf:resource="#hasParent">
</owl:ObjectProperty>
<Human rdf:ID="human01">
   <hasChild rdf:resource="#human02">
</Human>
```

- (human01, human02) ∈ ER(hasChild)

OWL Vocabulary: Property Description

OWL Vocabulary	Abstract Syntax	Semantics
owl:TransitiveProperty	ObjectProperty(p Transitive)	(x,y) ∈ ER(p) ∧ (y,z) ∈ ER(p) → (x,z) ∈ ER(p)
owl:SymmetricProperty	ObjectProperty(p Symmetric)	(x,y) ∈ ER(p) → (y,x) ∈ ER(p)
owl:FunctionalProperty	ObjectProperty(p Functional)	(x,y1) ∈ ER(p) ∧ (x,y2) ∈ ER(p) → y1 = y2
owl:InverseFunctionalProperty	ObjectProperty(p InverseFunctional)	(x1,y) ∈ ER(p) ∧ (x2,y) ∈ ER(p) → x1 = x2

Examples

```
<owl:ObjectProperty rdf:ID="hasVintageYear">
  <rdf:type rdf:resource="&owl;FunctionalProperty" />
  <rdfs:domain rdf:resource="#Vintage" />
  <rdfs:range  rdf:resource="#VintageYear" />
</owl:ObjectProperty>
```

OWL Vocabulary: Individual Description

Examples

```
<Human rdf:ID="Jane">
  <hasChild rdf:resource="#John" />
  <hasAge rdf:datatype="&xsd;positiveInteger">35</>
</Human>
```

- Jane ∈ Human ∧ (Jane, John) ∈ ER(hasChild) ∧ (Jane, 35) ∈ ER(hasAge)

OWL Vocabulary: Individual-Individual Relationship

OWL Vocabulary	Abstract Syntax	Semantics
owl:sameAs	SameIndividuals(i1 i2)	i1 = i2
owl:differentFrom	DifferentIndividuals(i1 i2)	i1 ≠ i2
owl:AllDifferent, owl:distinctMembers	Syntactic Sugar...	

Examples

```
<WineSugar rdf:ID="Dry" />
<WineSugar rdf:ID="Sweet">
  <owl:differentFrom rdf:resource="#Dry"/>
</WineSugar>
<WineSugar rdf:ID="OffDry">
  <owl:differentFrom rdf:resource="#Dry"/>
  <owl:differentFrom rdf:resource="#Sweet"/>
</WineSugar>

<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <vin:WineColor rdf:about="#Red" />
    <vin:WineColor rdf:about="#White" />
    <vin:WineColor rdf:about="#Rose" />
  </owl:distinctMembers>
</owl:AllDifferent>
```

Without declared or inferred owl:differentFrom, any individuals can be the same one.

OWL Vocabulary: Class Description by Property Restrictions

OWL Vocabulary	Abstract Syntax	Semantics
owl:allValuesFrom	restriction(p allValuesFrom(c))	{x | ∀y[(x,y) ∈ ER(p) → y ∈ c]}

Examples

```
<owl:Class rdf:ID="Wine">
  <rdfs:subClassOf rdf:resource="&food;PortableLiquid" />
  ...
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasMaker" />
      <owl:allValuesFrom rdf:resource="#Winery" />
    </owl:Restriction>
  </rdfs:subClassOf>
  ...
</owl:Class>
```

*For all wines, **if they have makers**, all the makers are wineries.*

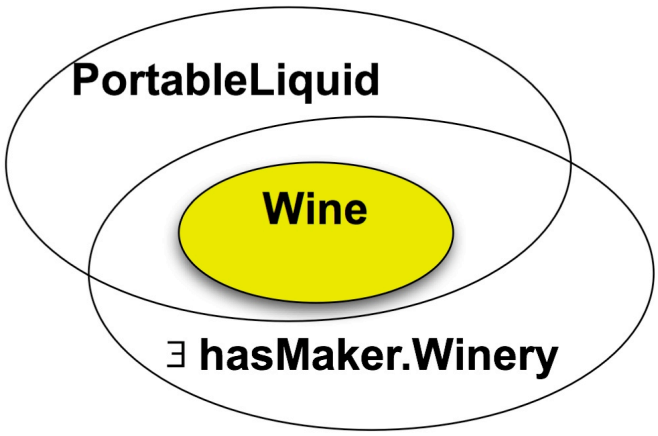
OWL Vocabulary: Class Description by Property Restrictions

OWL Vocabulary	Abstract Syntax	Semantics
owl:someValuesFrom	restriction(p someValuesFrom(c))	{x | ∃ y[(x,y) ∈ ER(p) ∧ y ∈ c]}

Examples

```
<owl:Class rdf:ID="Wine">
  <rdfs:subClassOf rdf:resource="&food;PortableLiquid" />
  ...
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasMaker" />
      <owl:someValuesFrom rdf:resource="#Winery" />
    </owl:Restriction>
  </rdfs:subClassOf>
  ...
</owl:Class>
```

For all wines, they have at least one maker that is a winery.



OWL Vocabulary: Class Description by Property Restrictions

OWL Vocabulary	Abstract Syntax	Semantics
owl:minCardinality	restriction(p minCardinality(n))	{x | #{y.(x,y) ∈ ER(p)} ≥ n}
owl:maxCardinality	restriction(p maxCardinality(n))	{x | #{y.(x,y) ∈ ER(p)} ≤ n}
owl:cardinality	restriction(p cardinality(n))	{x | #{y.(x,y) ∈ ER(p)} = n}

Examples

```
<owl:Class rdf:ID="Wine">
  <rdfs:subClassOf rdf:resource="&food;PotableLiquid"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#madeFromGrape"/>
      <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">
     1
      </owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  ...
</owl:Class>
```

Open-World Assumption: owl:minCardinality cannot contribute to consistency checking! owl:maxCardinality can!

```
Human a owl:Class;
      [a owl:Restriction;
       owl:onProperty hasFather;
       owl:maxCardinality "1"].
John a Human; hasFather Bob; hasFather Robert.
Bob owl:differentFrom Robert.
```

OWL Vocabulary: Class Description by Property Restrictions

OWL Vocabulary	Abstract Syntax	Semantics
owl:hasValue	restriction(p hasValue(v))	{x | (x,v) ∈ ER(p)}

Examples

```
<owl:Class rdf:ID="Burgundy">
  ...
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasSugar" />
      <owl:hasValue rdf:resource="#Dry" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

- Burgundy(x) → hasSugar(x,Dry)

OWL Vocabulary: Data Range

Examples

```
<owl:DatatypeProperty rdf:ID="tennisGameScore">
 <rdfs:range>
  <owl:DataRange>
   <owl:oneOf>
    <rdf:List>
     <rdf:first rdf:datatype="&xsd;integer">0</rdf:first>
     <rdf:rest>
      <rdf:List>
       <rdf:first rdf:datatype="&xsd;integer">15</rdf:first>
       <rdf:rest>
        <rdf:List>
         <rdf:first rdf:datatype="&xsd;integer">30</rdf:first>
         <rdf:rest>
          <rdf:List>
           <rdf:first rdf:datatype="&xsd;integer">40</rdf:first>
           <rdf:rest rdf:resource="&rdf;nil" />
          </rdf:List>
         </rdf:rest>
        </rdf:List>
       </rdf:rest>
      </rdf:List>
     </rdf:rest>
    </rdf:List>
   </owl:oneOf>
  </owl:DataRange>
 </rdfs:range>
</owl:DatatypeProperty>
```

- ER(tennisGameScore) = Δ^I X {0, 15, 30, 40}

OWL Inference: Common Tasks

- Consistency Check
- Subsumption & Equivalence
- Membership Determination

OWL Inference: Consistency Check (1)

Checking the consistency of an OWL ontology.

The following ontology is consistent.

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix owl:  <http://www.w3.org/2002/07/owl#>.
@prefix a:    <http://www.ex.org/2005/ex#>.

a:Vegetarian
       owl:intersectionOf
          (a:Animal
           [a owl:Restriction;
            owl:onProperty a:eats;
            owl:allValuesFrom a:Plant
           ]
          ).
a:Plant owl:complementOf a:Animal.
a:Sheep rdfs:subClassOf a:Vegetarian.
a:Cow rdfs:subClassOf a:Vegetarian.
a:MadCow
       owl:intersectionOf
          (a:Cow
           [a owl:Restriction;
            owl:onProperty a:eats;
            owl:someValuesFrom a:Sheep
           ]
          ).
```

MadCow ≡ Animal ∩ ∀eats.Plant ∩ ∃eats.Sheep

MadCow is an unsatisfiable concept, thus equivalent to owl:Nothing.

OWL Inference: Consistency Check (2)

Checking the consistency of an OWL ontology.

The following ontology is inconsistent.

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix owl:  <http://www.w3.org/2002/07/owl#>.
@prefix a:    <http://www.ex.org/2005/ex#>.

a:Vegetarian
       owl:intersectionOf
          (a:Animal
           [a owl:Restriction;
            owl:onProperty a:eats;
            owl:allValuesFrom a:Plant
           ]
          ).
a:Plant owl:complementOf a:Animal.
a:Sheep rdfs:subClassOf a:Vegetarian.
a:Cow rdfs:subClassOf a:Vegetarian.
a:MadCow
       owl:intersectionOf
          (a:Cow
           [a owl:Restriction;
            owl:onProperty a:eats;
            owl:someValuesFrom a:Sheep
           ]
          ).
a:Doll a a:MadCow.
```

Problem: *The unsatisfiable concept MadCow, owl:Nothing, has a member individual.*

OWL Inference: Subsumption (1)

Deriving hidden subsumption structure from an OWL ontology.

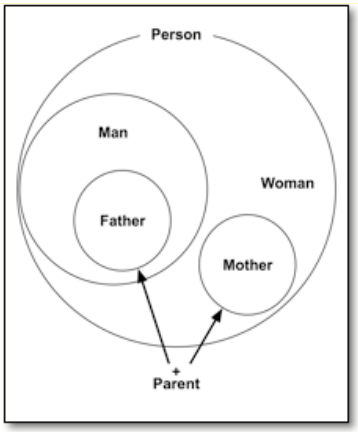
For example,

```
Class(Woman complete intersectionOf(Person,Female)))
Class(Man complete
           intersectionOf(Person, complementOf(Woman)))
Class(Mother complete
          intersectionOf(Woman,
           restriction(hasChild, someValuesFrom(Person))))
Class(Father complete
          intersectionOf(Man,
           restriction(hasChild, someValuesFrom(Person))))
Class(Parent complete unionOf(Mother, Father))
```

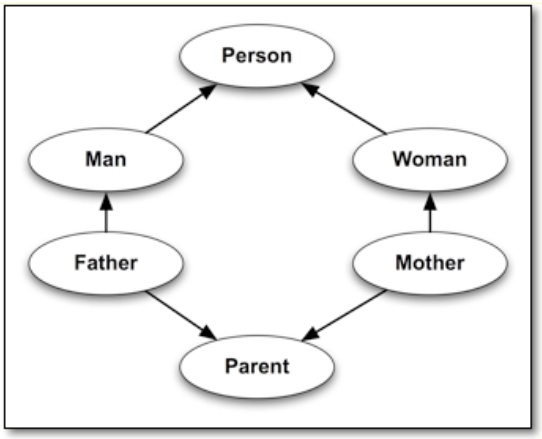
Woman ≡ Person ∩ Female.
Man ≡ Person ∩ ¬ Woman.
Mother ≡ Woman ∩ ∃ hasChild.Person.
Father ≡ Man ∩ ∃ hasChild.Person.
Parent ≡ Mother ∪ Father.

OWL Inference: Subsumption (2)

Ontology description represent...



Implied subsumption structue is...



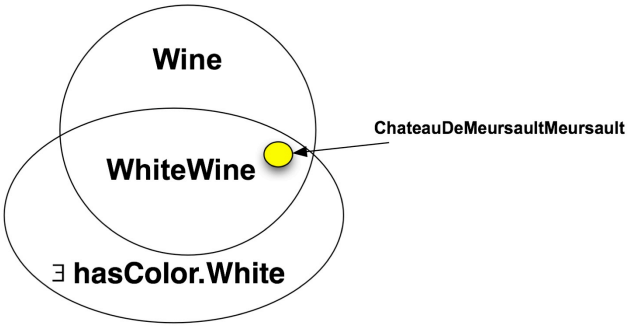
OWL Inference: Membership

Determine the types of individuals.

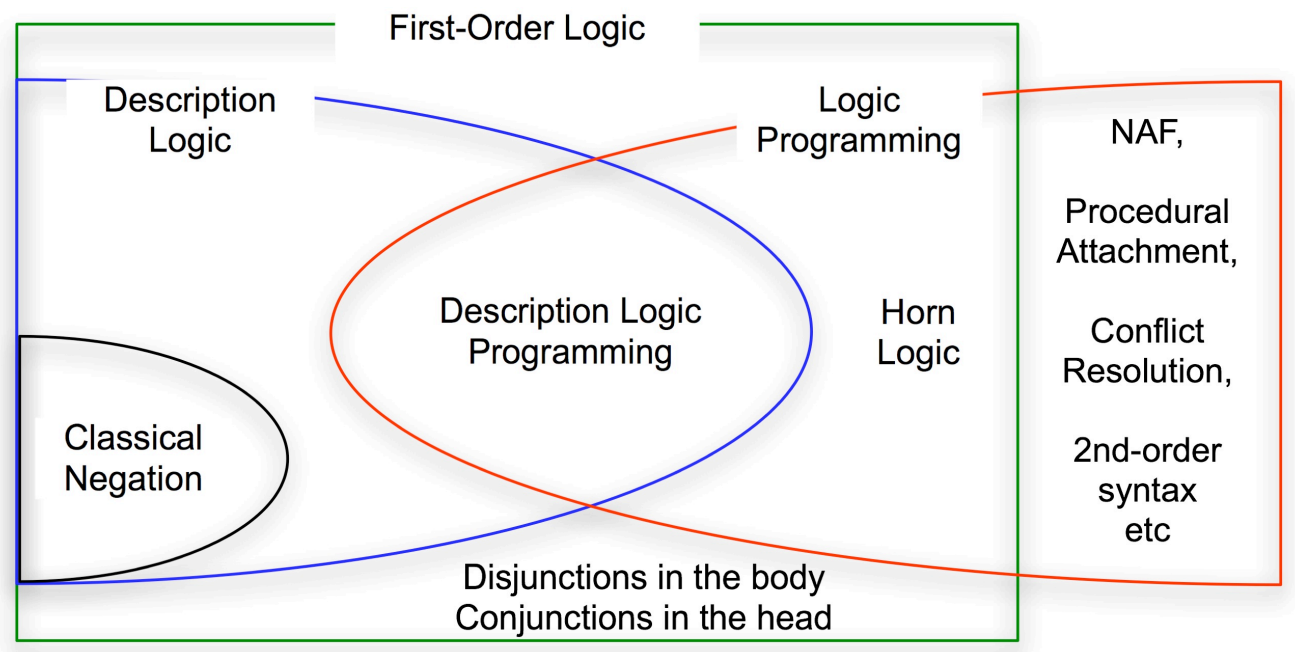
For example,

```
Class(WhiteWine complete
          intersectionOf
             (Wine,
               restriction(hasColor, hasValue(White)))
      )
Individual(ChateauDeMeursaultMeursault type(Wine) hasColor(White)
```

An implication: *ChateauDeMeursaultMeursault is a WhiteWine*.



Beyond OWL: Expressiveness



Beyond OWL: OWL's Limit

- Property Compositions
  - *The brother of somebody's father is that somebody's uncle.*
- Expressions & Functions
  - *ex1) Teenager is a class of people whose age is greater than 10 and less than 20.*
  - *ex2) MiddleClassCar is a class of cars with a displacement of larger than 2000cc.*
- Non-monotonic Reasoning (Negation-As-Failure)
  - *ex1) NiceGame is a class of games that have not received any bad reputation.*

Ontology Usage & Tools

Ontology Building

- Ontology Builders
- e.g. Protege, KAON, exOWL, etc

Ontology Sharing

- Ontology Alignment/Merging Tools
- e.g. Chimera, PROMPT, MoA, etc

Ontology Query/Reasoning

- OWL Ontology Reasoners
- e.g. Pellet, KAON2, Bossam, Saseme etc

References

- W3C Web Ontology Language Specifications @ W3C Site
- Tim Berners-Lee, "What the Semantic Web can Represent?", http://www.w3.org/DesignIssues/RDFnot.html
- Ian Horrocks et al., "From SHIQ and RDF to OWL"

Appendix 1: Description Logics Designators

Notation	Description
AL	Atomic Negation, Concept Intersection, Universal Restriction, Limited Existential Quantification
C	Complex Concept Negation
S	An abbreviation for ALC
H	Role Hierarchy
O	Nominals (owl:oneOf, owl:hasValue)
I	Inverse Properties
F	Functional Properties
N	Cardinality Restrictions
(D)	Datatype Properties, Data Values, Data Types