

SEMINARIO 3: ONTOLOGÍAS

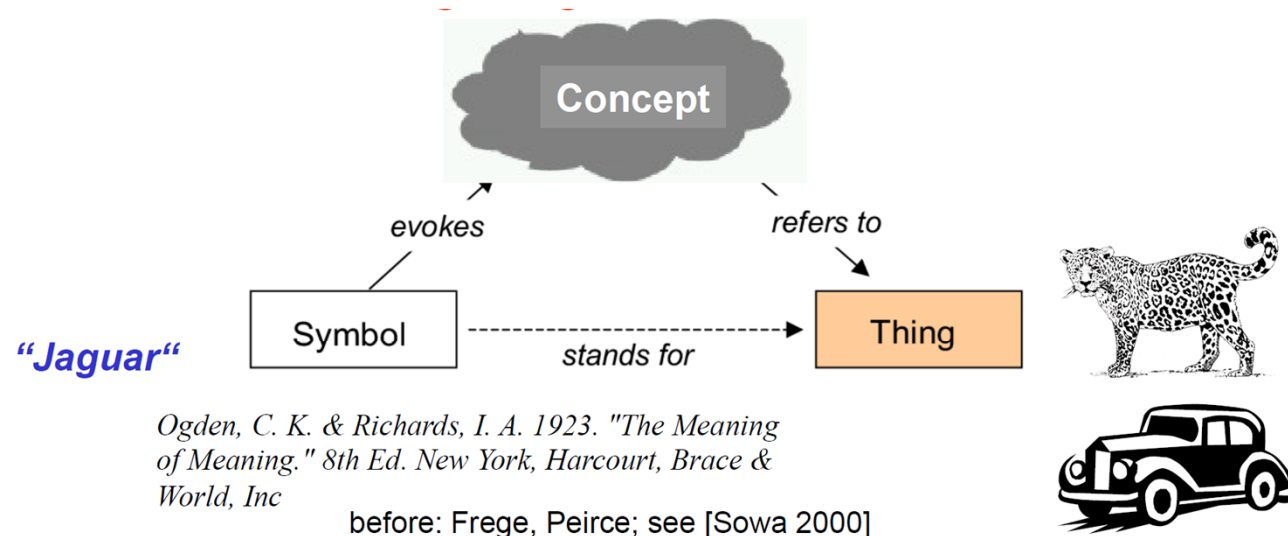


1. Introducción a las ontologías
2. RDF y RDFS
3. OWL
4. Protégé



Información vs. conocimiento

- en la web tenemos información: gran colección de hechos
- debemos traducir esos hechos a conceptos: verdades, creencias, perspectivas, juicios, metodologías, know-how...



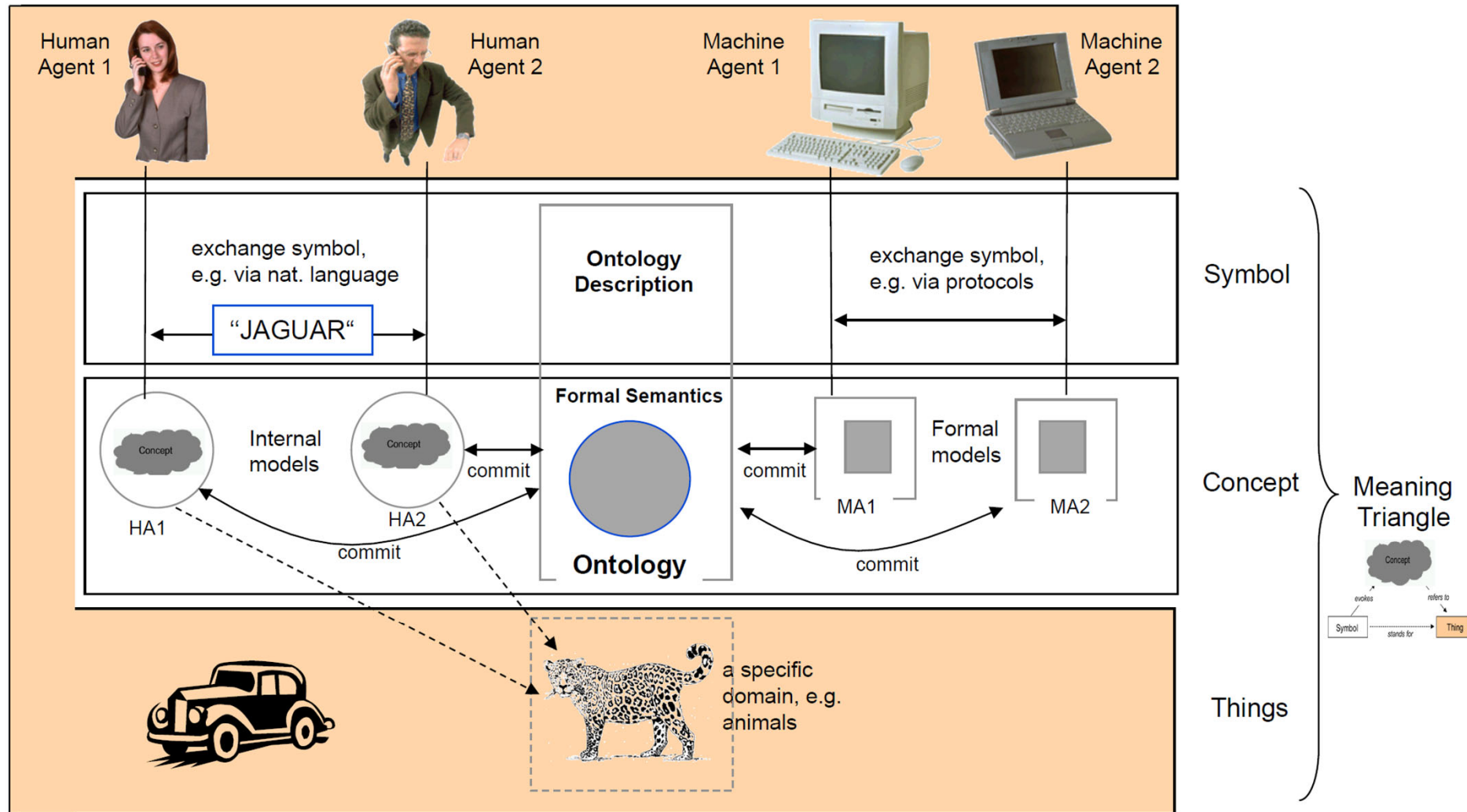
[Carole Goble, Nigel Shadbolt, Ontologies and the Grid Tutorial]



Problemas

1. ¿Cómo representamos el conocimiento para que lo pueda tratar una máquina?
2. ¿Cómo lo usamos en aplicaciones reales?





[Maedche et al., 2002]



Una ontología define los términos y conceptos comunes empleados para describir y representar un área de conocimiento.

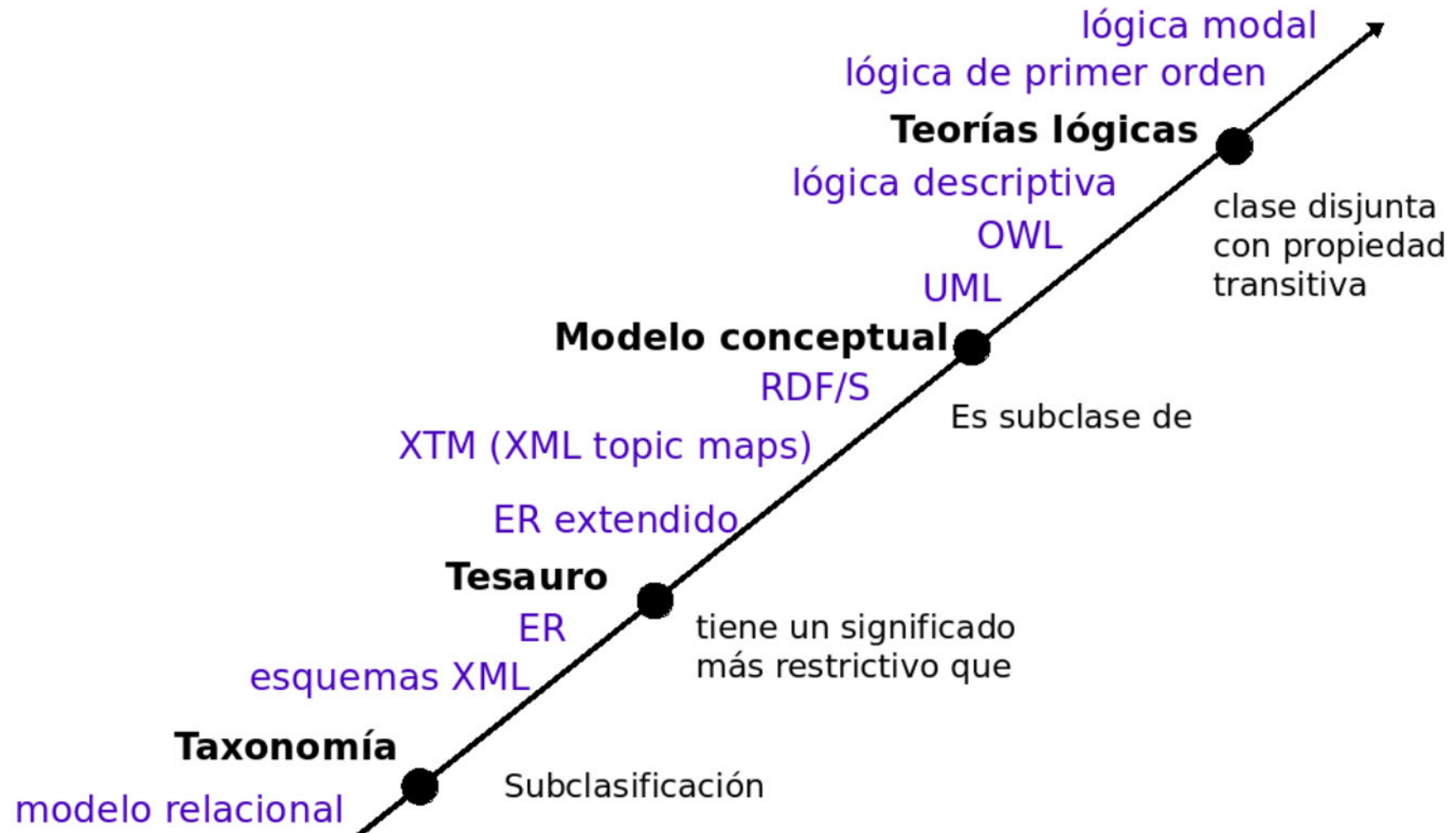
Descripción mediante

- clases
- instancias
- relaciones
- propiedades
- funciones/procesos
- restricciones

Representación

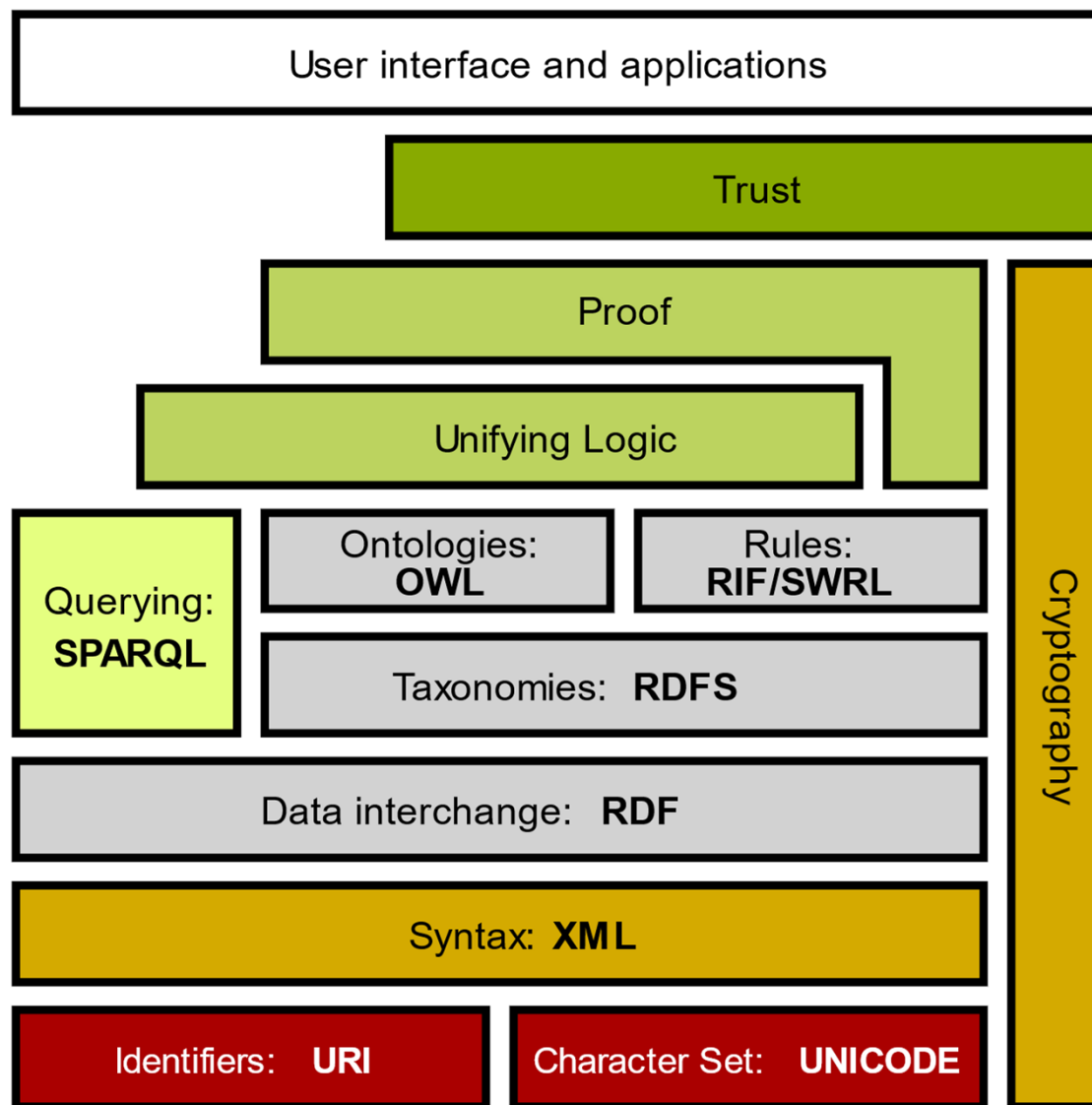
- frases que combinan la terminología para expresar relaciones entre los términos
- estas frases aportan **significado**.



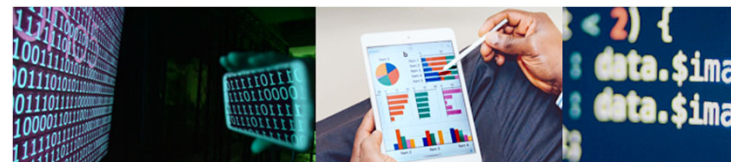


1. Determinar el alcance y el dominio de la ontología
 - Para qué se va a utilizar, qué tipo de preguntas debe contestar, ...
2. Considerar la reutilización de ontologías existentes
 - <https://lov4iot.appspot.com/>
3. Enumerar los términos importantes de la ontología
4. Definir las clases y la jerarquía de clases
5. Definir las propiedades de las clases - slots
 - Propiedades, relaciones
6. Definir las facetas de los slots
 - Tipo de valor (String, Number, Boolean, Enummerated, Instance)
 - Cardinalidad del valor
7. Definir las instancias





Fuente: Web for real people. Tim Berners-Lee



Resource Definition Framework (RDF), no es propiamente un lenguaje de ontologías pero está muy próximo

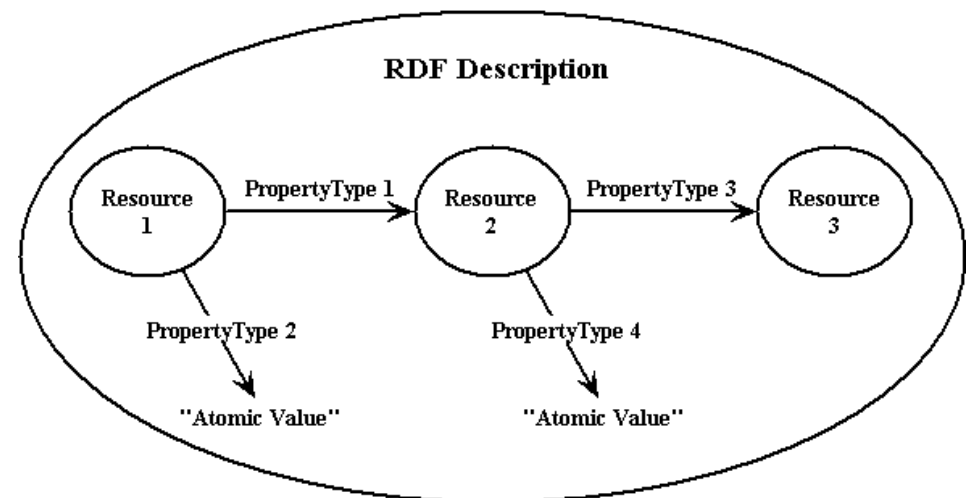
Objetivo: proporcionar un marco de representación del conocimiento estandarizado para la web

Sintaxis de RDF

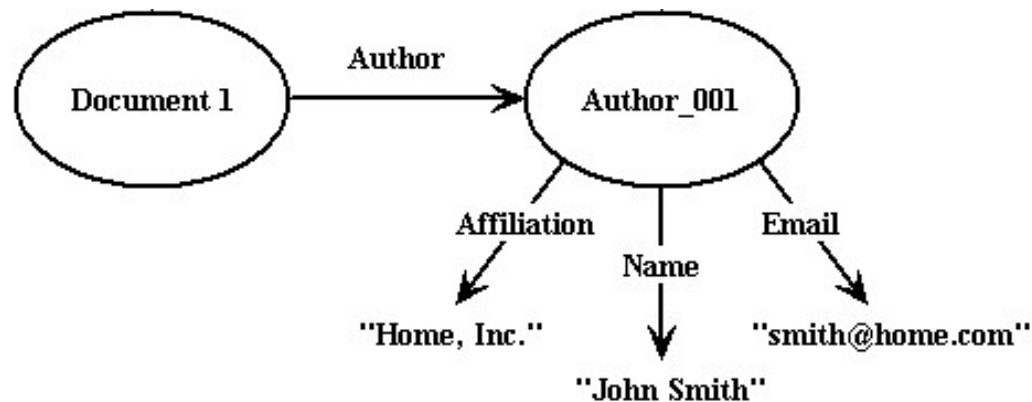
- Diferentes vocabularios de XML
- <http://www.w3.org/RDF/>

Modelo de datos de RDF:

- Tripletas (recurso, propiedad, valor)
 - Recurso (Sujeto)
 - Propiedad (Predicado)
 - Valor (Objeto)



- Recurso: Ente del cual se habla
- Propiedad: Define relaciones del ente con otros entes o valores, aspectos específicos, características, atributos o relaciones.
- Objeto : Es la entidad (persona, animal, cosas, ...) a la que se refiere el predicado.



RDF permite construir KB basadas en una ontología, pero ¿cómo se define la ontología?

RDF Schema (RDFS)

RDFS proporciona un lenguaje básico para definir nuestro vocabulario con

- jerarquía de clases e instancias
- restricciones (sobre las propiedades)
- jerarquía también de propiedades

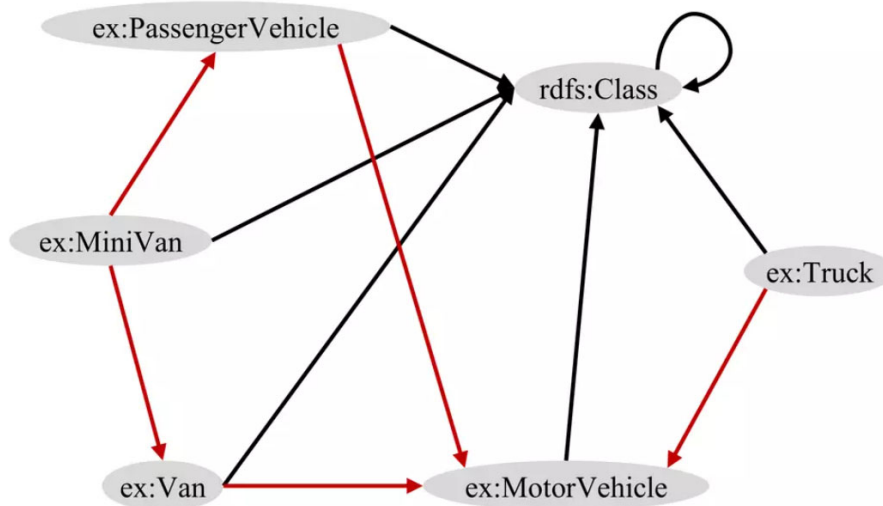
Se diferencia de los modelos de datos OO en que se debe seguir una **metodología bottom-up**: se definen las propiedades y luego se establece a qué clases caracterizan.

<http://www.w3.org/TR/rdf-schema/>



Vocabulario clases:

- rdfs:Resource: todos los elementos son recursos
- rdfs:Class: clase de recurso
- rdf:type: para indicar que un recurso es una instancia de una clase
- rdfs:subClassOf: para indicar que todas las instancias de una clase son instancias de otra



```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://example.org/schemas/vehicles">

  <rdfs:Class rdf:ID="MotorVehicle"/>

  <rdfs:Class rdf:ID="PassengerVehicle">
    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
  </rdfs:Class>

  <rdfs:Class rdf:ID="Truck">
    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
  </rdfs:Class>

  <rdfs:Class rdf:ID="Van">
    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
  </rdfs:Class>

  <rdfs:Class rdf:ID="MiniVan">
    <rdfs:subClassOf rdf:resource="#Van"/>
    <rdfs:subClassOf rdf:resource="#PassengerVehicle"/>
  </rdfs:Class>

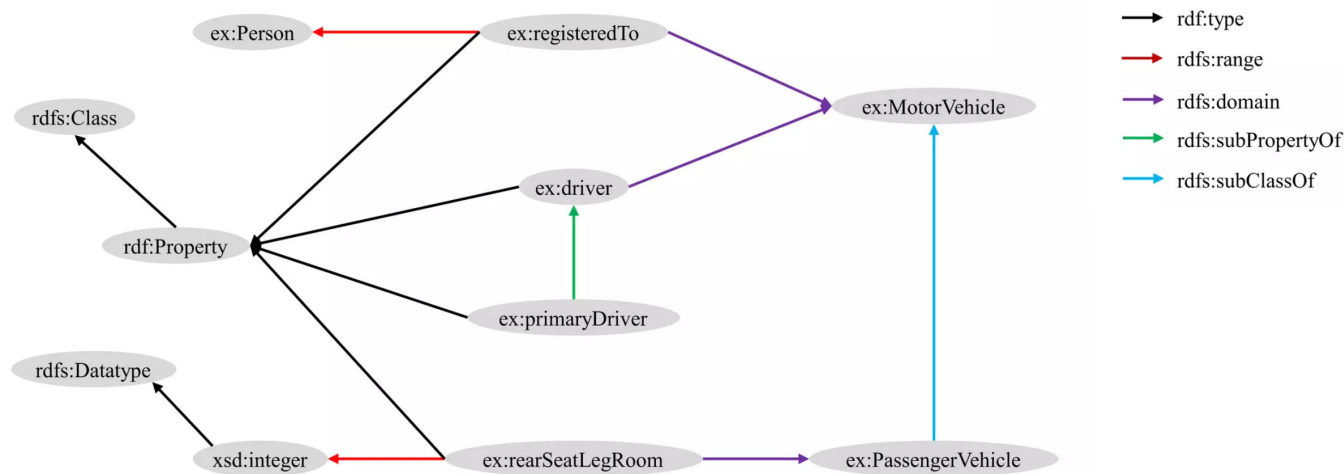
</rdf:RDF>
```



Vocabulario propiedades:

- `rdf:Property`: clase de las propiedades RDF
- `rdfs:subPropertyOf`: para indicar que todos los recursos relacionados con una propiedad también están relacionados con otra
- `rdfs:domain`: para indicar que un recurso que tiene una determinada propiedad es una instancia de una o más clases
- `rdfs:range`: para indicar que los valores de una propiedad son instancias de una o más clases





La propiedad “registeredTo” se aplica sobre instancias de la clase “MotorVehicle” (domain) y toma valores de instancias de la clase “Person” (range)

```
<rdf:RDF xml:lang="en"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <rdf:Description ID="registeredTo">
    <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="#MotorVehicle"/>
    <rdfs:range rdf:resource="#Person"/>
  </rdf:Description>

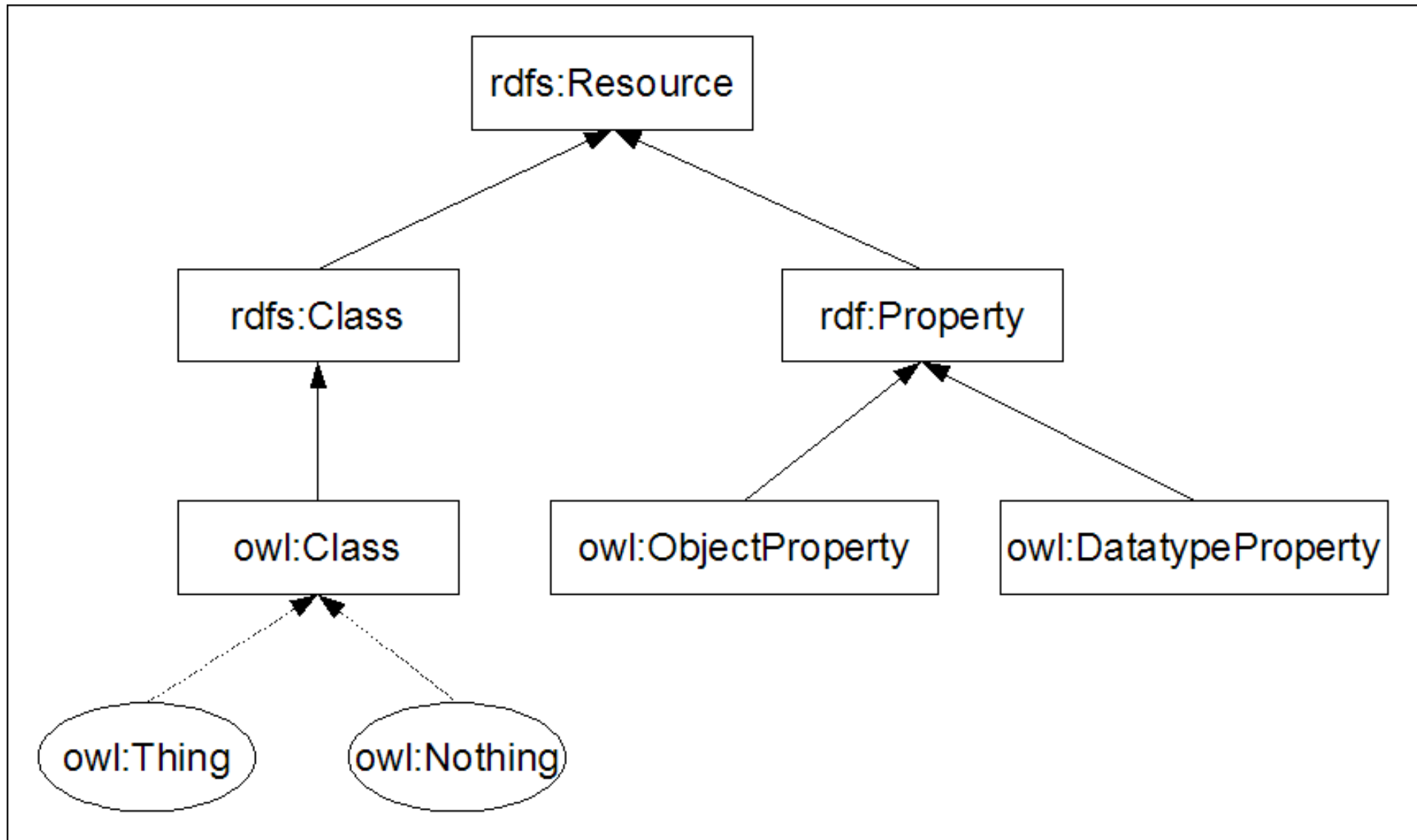
  <rdf:Description ID="rearSeatLegRoom">
    <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="#PassengerVehicle"/>
    <rdfs:domain rdf:resource="#Minivan"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/03/example/classes#Number"/>
  </rdf:Description>
</rdf:RDF>
```



OWL aporta mayor expresividad, pudiendo definir

1. clases como **combinaciones booleanas** de otras con union, intersection y complement.
2. clases como **disjoint**
3. dos clases como iguales **equivalentClass**
4. dos individuos (instancias) son o no el mismo **sameAs**, **differentFrom**
5. **cardinalidad** en las propiedades
6. propiedades **transitivas**





Classes and Resources		
Property	Used to say that...	Example
intersectionOf	...any instance of the first class is also an instances of all classes in the specified list	<code>:Mother owl:intersectionOf (:Woman :Parent)</code>
unionOf	...any instance of the first class is an instance of at least one of the classes in the specified list	<code>:Parent owl:unionOf (:Mother :Father)</code>
complementOf	...the first class is equivalent to everything not in the second class	<code>:Parent owl:complementOf :NonParent</code>
disjointWith	...the first class and second class have no members in common	<code>:Man owl:disjointWith :Woman</code>
equivalentClass	...the first class and the second class contain all the same members	<code>:AdultFemaleHuman owl:equivalentClass :Woman</code>
sameAs	...the first resource refers to the exact same thing as the second resource	<code>:JimFromWork owl:sameAs :MyNeighborJim</code>
differentFrom	...the first resource refers to something different from the second resource	<code>:BobFromWork owl:differentFrom :MyNeighborBob</code>

<https://cambridgesemantics.com/blog/semantic-university/learn-owl-rdfs/owl-references-humans/>



Basic Property Types			
Kind of Property	Used to say...	Example	Explanation
DatatypeProperty	...that this property links to simple data values	ex:hasBirthday	This property links to a date, which is a simple data value
ObjectProperty	...that this property links to another resource	ex:hasSpouse	This property links to a person, which is another resource
Logical Relationships			
Kind of Property	Used to say...	Example	Explanation
TransitiveProperty	...that if this property links A to B, and B to C, then it also links A to C.	ex:tallerThan	If Ann is taller than Bob, and Bob is taller than Chuck, then Ann is taller than Chuck
SymmetricProperty	...that if the property relates A to B, then it always relates B to A as well.	ex:hasSpouse	If Ann is Bob's spouse, then Bob is Ann's spouse too
AsymmetricProperty	...that if the property relates A to B, then it never relates B to A.	ex:tallerThan	If Ann is taller than Bob, then Bob can't be taller than Ann
ReflexiveProperty	...that this property always links something to itself.	ex:livesWith	Everybody lives with themselves
IrreflexiveProperty	...that this property never links something to itself.	ex:hasSpouse	Nobody is their own spouse
FunctionalProperty	...that this property only ever links to at most one thing.	ex:hasBirthday	You only have one birthday
InverseFunctionalProperty	...that the subject of this property is uniquely identified by the value of this property.	ex:hasDLNumber	I am the only person with my drivers license number
Properties Linking Properties			
Property	Used to say that...	Example	
inverseOf	...the two properties are the inverse of each other. For example, if Ann's child is Bob, then Bob's parent is Ann.	:hasChild owl:inverseOf :hasParent	
equivalentProperty	...two properties are exactly the same	:hasBirthPlace owl:equivalentProperty :hasBirthLocation	

Restrictions and Enumerations			
Parameter	Used to say...	Example	Explanation
cardinality min-cardinality max-cardinality	...that the property can have a certain number of values (objects).	<pre> :Automobile owl:equivalentclass [rdf:type owl:Restriction ; owl:cardinality "4"^^xsd:int ; owl:onProperty :hasWheel]. </pre>	All automobiles have 4 wheels (e.g., as opposed to a bicycle).
oneOf	...that all instances of a class come from the specified list	<pre> :BobsChildren owl:equivalentClass [rdf:type owl:Class ; owl:oneOf (:Bill :John :Mary)]. </pre>	The class 'BobsChildren' has the three items: Bill, John, and Mary



hasValue	...that all objects of that property have the specified value	<pre> :BobsChildren owl:equivalentClass [rdf:type owl:Restriction ; owl:onProperty :hasParent ; owl:hasValue :Bob]. </pre>	Each instance of BobsChildren has 'Bob' as the object of its :hasParent property.
someValuesFrom	...that at least one object of that property is a member of the specified class.	<pre> :Parent owl:equivalentClass [rdf:type owl:Restriction ; owl:onProperty :hasChild ; owl:someValuesFrom :Person]. </pre>	Any instance of the 'Parent' class has at least one child that is a Person



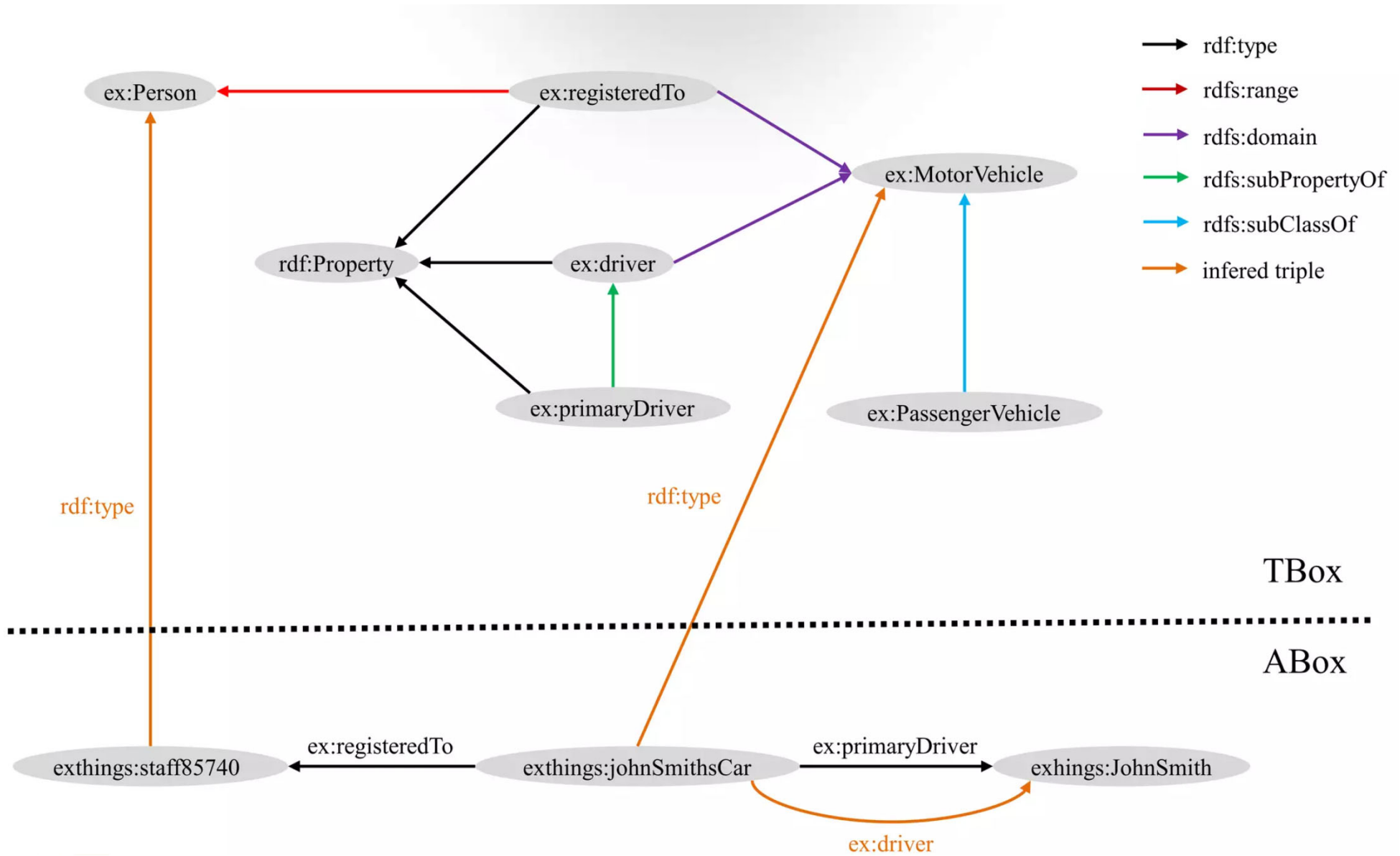
allValuesFrom	...that all objects of that property are members of the specified class	<pre> :Vegetarian owl:equivalentClass [rdf:type owl:Restriction ; owl:onProperty :eats ; owl:allValuesFrom :NonMeat]. </pre>	The class 'Vegetarian' is equivalent to the class of things that only eat non-meat.
---------------	---	--	---



Proceso de razonamiento sobre ontologías (lógica descriptiva):

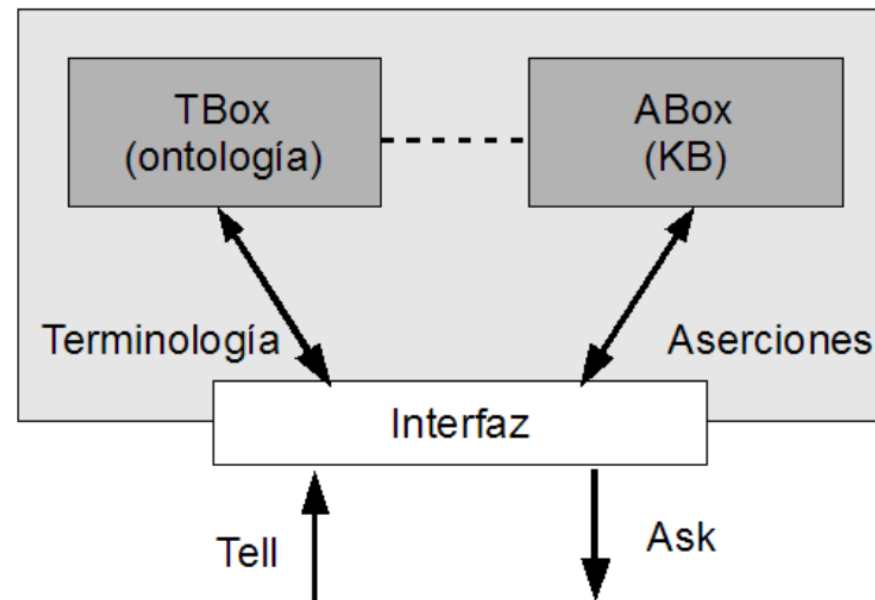
- ABox (Assertion Box): El ABox es la parte de una ontología que se encarga de representar hechos o instancias específicas del dominio que se está modelando. Contiene afirmaciones concretas sobre individuos, sus propiedades y las relaciones entre ellos.
- Por ejemplo, en una ontología médica, el ABox podría incluir afirmaciones sobre pacientes específicos, como "Paciente1 tieneSíntoma Fiebre" o "Paciente2 esTratadoCon MedicamentoX".
- TBox (Terminological Box): El Tbox se encarga de definir las clases, propiedades y restricciones conceptuales que estructuran el conocimiento en la ontología. Contiene información sobre las clases, las jerarquías de clases, las propiedades, las relaciones entre clases y las restricciones sobre las instancias.
- Por ejemplo, el TBox podría definir clases como "Enfermedad", "Síntoma", "Tratamiento" y establecer relaciones como "tieneSíntoma" y "esTratadoCon".
- La distinción entre ABox y TBox es fundamental en la representación del conocimiento, ya que permite separar los hechos concretos (ABox) de la estructura conceptual y las reglas (TBox). Esto facilita la representación y el razonamiento sobre el conocimiento de manera más organizada y eficiente. Los sistemas de razonamiento y consulta en ontologías utilizan esta distinción para realizar inferencias y responder preguntas sobre la base del conocimiento contenido en el ABox y el TBox.





Proceso de razonamiento sobre ontologías (lógica descriptiva):

Interfaz tell&ask



Añadir nuevo
conocimiento

Extraer información
de la ontología

