

# OWL

DESCRIPCION DE LA ONTOLOGÍA



Universidad

Rey Juan Carlos

# INDICE

OWL

- Introducción
- Utilidades de las ontologías para la Web
- OWL conceptos
- 3 Sub-lenguajes
- Sinopsis Lite
- Sinopsis DL y Completo
- Descripción de la lengua (Lite)
- Descripción OWL DL y Completo

# INTRODUCCION <sup>OWL</sup>

## **ONTOLOGIA:**

En Informática para **definir vocabularios que las máquinas puedan entender** y que sean especificados con la suficiente precisión como para permitir diferenciar términos y referenciarlos de manera precisa.

Las ontologías incluyen definiciones de **conceptos básicos del dominio, y las relaciones entre ellos, que son útiles para los ordenadores** [...]. Codifican el conocimiento de un dominio y también el conocimiento que extiende los dominios. En este sentido, hacen el **conocimiento reutilizable**.

# INTRODUCCION OWL

**Un lenguaje de marcado** o lenguaje de marcas es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación. El más conocido es **HTML**, pero hay otras importantes variantes, como es el caso de **OWL**.

*Web Ontology Language ¿OWL en lugar de WOL?*

- Owl de *Winnie the Pooh*, quien escribe su nombre WOL en lugar de OWL.
- Fácilmnte pronunciabile en inglés.
- Relación con el prestigioso proyecto de representación del conocimiento de los años 70 de Bill Martin "One World Language".

# Utilidad de las Ontologías para la Web

- **Portales Web**
  - Reglas de categorización utilizadas para mejorar la búsqueda
- **Colecciones Multimedia**
  - Búsquedas basadas en contenido para medios no textuales
- **Administración de Sitios Web Corporativos**
  - Organización taxonómica automatizada de datos y documentos
  - Asignación entre Sectores Corporativos (¡fusiones!)

# Utilidad de las Ontologías para la Web

## • Documentación de Diseño

- Explicación de partes "derivadas" (el ala de un avión)
- Administración explícita de Restricciones

## • Agentes Inteligentes

- Expresión de las Preferencias y/o Intereses de los usuarios
- Mapeo de contenidos entre sitios Web

## • Servicios Web y Computación Ubicua

- Composición y Descubrimiento de Servicios Web
- Administración de Derechos y Control de Acceso

# OWL Conceptos OWL

- Ordenadores necesitan **acceso a colecciones estructuradas** de información.
- Para cubrir esta necesidad se usa “**Representación del conocimiento**”
- **OWL** es un **gran paso** en el desarrollo de ese potencial.
- Lenguajes **anteriores** han sido utilizados para desarrollar herramientas y ontologías destinadas a **comunidades específicas**.
- No fueron definidos para ser compatibles con la arquitectura de la World Wide Web en general, y la Web Semántica en particular.

# OWL Conceptos

OWL

- **¿Qué mejoras aporta OWL?**
  - Capacidad de ser distribuidas a través de varios sistemas
  - Escalable a las necesidades de la Web
  - Compatible con los estándares Web de accesibilidad e internacionalización
  - Abierto y extensible
- **Su actualidad:**
  - Inversiones de EEUU (a través de DARPA y la NSF) y la Unión Europea (mediante las infraestructuras de 5ª y 6ª generación del programa IST).
  - Hay un gran número de ontologías OWL disponibles en la Web.
  - Existe una biblioteca de ontologías en la biblioteca de ontologías DAML, la cual contiene unos 250 ejemplos hechos con OWL o DAML+OIL (hay un convertidor de DAML+OIL a OWL disponible en la Web).



# OWL Conceptos OWL

Extensión con respecto a RDF:

- Los recursos para limitar las propiedades de clases con respecto a número y tipo.
- Los recursos para inferir qué elementos que tienen varias propiedades son miembros de una clase en particular.
- Los recursos para determinar si todos los miembros de una clase tendrán una propiedad en particular, o si puede ser que sólo algunos la tengan.

# OWL Conceptos OWL

- Los recursos para distinguir entre relaciones uno a uno, varios-a-uno o uno-a-varios, permitiendo que las "claves externas" de las bases de datos puedan representarse en una ontología.
- Los recursos para expresar relaciones entre clases definidas en diferentes documentos en la Web.
- Los recursos para construir nuevas clases a partir de uniones, intersecciones y complementos de otras.
- Los recursos para restringir rangos y dominios para especificar combinaciones de clases y propiedades.

## 3 Sub-Lenguajes <sup>OWL</sup>

- **OWL Lite**

- Apoya a esos usuarios que necesitan sobre todo una jerarquía de la clasificación y apremios simples.
- Permite solamente valores de 0 o 1. proporciona una trayectoria rápida de la migración para otras taxonomías.
- OWL Lite tiene una complejidad formal más baja que OWL DL

# 3 Sub-Lenguajes <sup>OWL</sup>

- **OWL DL**

- Expresividad máxima
- El OWL DL incluye todas las construcciones de la lengua del OWL, pero pueden ser utilizadas solamente bajo ciertas restricciones

mientras que una clase puede ser una subclase de muchas clases, una clase no puede ser un caso de otra clase

- OWL DL es así que nombrada debido a su correspondencia con lógicas de la descripción

## 3 Sub-Lenguajes <sup>OWL</sup>

- **OWL (Completo)**

- Expresividad máxima y la libertad sintáctica de RDF (sin garantías de cómputo).

Una clase se puede tratar por completo simultáneamente como colección de individuos y como individuo por derecho propio.

- OWL permite que un ontología aumente el significado (RDF o OWL) del vocabulario predefinido

## 3 Sub-Lenguajes <sup>OWL</sup>

- Por supuesto:
  - Cada ontología legal de Lite OWL es una ontología legal del OWL DL.
  - Cada ontología legal OWL DL es una ontología del OWL legal.
  - Cada conclusión válida de Lite OWL es una conclusión válida del OWL DL.
  - Cada conclusión válida OWL DL es una conclusión completa del OWL válido.

# Sinopsis Lite OWL

## RDF Schema Features:

- Class (Thing, Nothing)
- rdfs:subClassOf
- rdf:Property
- rdfs:subPropertyOf
- rdfs:domain
- rdfs:range
- Individual

## (In)Equality:

- equivalentClass
- equivalentProperty
- sameAs
- differentFrom
- AllDifferent
- distinctMembers

## Property Characteristics:

- ObjectProperty
- DatatypeProperty
- inverseOf
- TransitiveProperty
- SymmetricProperty
- FunctionalProperty
- InverseFunctionalProperty

## Property Restrictions:

- Restriction
- onProperty
- allValuesFrom
- someValuesFrom

## Restricted Cardinality:

- minCardinality (only 0 or 1)
- maxCardinality (only 0 or 1)
- cardinality (only 0 or 1)

## Header Information:

- Ontology
- imports

## Class Intersection:

- intersectionOf

## Versioning:

- versionInfo
- priorVersion
- backwardCompatibleWith

## Annotation Properties:

- rdfs:label
- rdfs:comment
- rdfs:seeAlso

# Sinopsis DL y Completo

OWL

## Class Axioms:

- oneOf, dataRange
- disjointWith
- equivalentClass  
(applied to class expressions)
- rdfs:subClassOf  
(applied to class expressions)

## Arbitrary Cardinality:

- minCardinality
- maxCardinality
- cardinality

## Boolean Combinations of Class Expressions:

- unionOf
- complementOf
- intersectionOf

## Filler Information:

- hasValue



# Descripción Lengua (Lite)

**OWL Lite** solamente tiene **algunas de las características** de lengua del OWL y **tiene más limitaciones** en el uso de las características que el DL o el OWL (completo).

- Las clases se pueden definir **solamente en términos de superclases nombradas** (no expresiones arbitrarias)
- Solamente **ciertas clases de restricciones de la clase pueden ser utilizadas**
- **Restricciones** en el OWL Lite utilizan **solamente clases nombradas**
- Las únicas **cardinalidades** permitidas son explícitamente son **0 o 1**

# Descripción Lengua (Lite)

## Características del esquema

- **RDFS CLASE:**

Una clase define un grupo de individuos que comparten algunas características.

*Marta y Manolo: Clase persona.*

Las clases se pueden organizar en una jerarquía de la especialización usando el **subClassOf**.

- **RDFS SUBCLASSOF:**

Las jerarquías de las clases. Una clase es una subclase de otra clase.

Clase persona podría ser indicada para ser una subclase de la clase mamífero. De esto se puede deducir que si un individuo es una persona, entonces es también un mamífero.

# Descripción Lengua (Lite)

## Características del esquema

- **RDFS CARACTERISTICAS:**

Las características se pueden utilizar para indicar relaciones entre los individuos o de individuos a los valores de los datos.

- **RDFS SUB-CARACTERISTICAS:**

Las jerarquías de las características pueden ser creadas haciendo una o más declaraciones.

Si un individuo es relacionado con otro por una característica la cual es subproperty de otra, podremos decir que ambos individuos también están relacionados por esa característica.

# Descripción Lengua (Lite)

## Características del esquema

- **RDFS DOMINIO:**

Un dominio de una característica que limita los individuos a los cuales la característica puede ser aplicada.

*Ana es mamífero. Manolo es hijo de Ana, entonces Manolo debe ser un mamífero.*

- **RANGO:**

El rango de una característica limita a los individuos en el valor de sus características. Si una característica relaciona a individuo con otro individuo, y la característica tiene un rango, entonces el otro individuo debe pertenecer a la clase del rango.

- **INDIVIDUOS:**

Los individuos son casos de clases, y las características se pueden utilizar para relacionar a un individuo con otro.

*Marta (individuo) :Clase persona.*

*UniversidadReyJuanCarlos (individuo): Clase universidad*

*Tiene\_empleo (característica) que relaciona a marta con otro individuo de otra clase (UniversidadReyJuanCarlos)*

# Descripción Lengua (Lite)

## Propiedades de las características

- **InverseOf:**  
Una característica se puede indicar para ser lo contrario de otra característica. *“Tiene\_hijo” es lo contrario de “Tiene\_pariente” y Marta y Manolo son parientes, entonces se puede deducir que Manolo y Marta no son padre e hija.*
- **TransitiveProperty:** Las características se pueden indicar para ser transitivas.  
*(Sara, Marta) son un caso de la característica antepasado y Marta es un antepasado de Débora (es decir, (Marta, Débora) son un caso de la característica antepasado), entonces se puede deducir que Sara es un antepasado de Débora es decir, (Sara, Débora) son un caso de la característica antepasado.*
- **SymmetricProperty:** Las características se pueden indicar para ser simétricas.  
*Amigo se puede indicar para ser una característica simétrica. Entonces se puede deducir que si Manolo es una migo de Marta, Marta es amiga de Manolo.*

# Descripción Lengua (Lite)

## Propiedades de las características

- **FunctionalProperty:**

Las características se pueden indicar para tener un valor único.

La cardinalidad mínima de la característica es cero y la máxima es 1.

*“Primer\_empleo” se puede indicar para ser un FunctionalProperty. Esto implica que nadie trabajo varias veces por primera vez.*

- **InverseFunctionalProperty:**

Las características se pueden indicar para ser funcionales inverso. Si una característica es funcional inverso entonces lo contrario de la característica es funcional.

*“El valor de dicha característica será único en su clase”.*

# Descripción Lengua (Lite)

## Restricciones de las características

- **AllValuesFrom:**

Significa que esta característica en esta clase particular tiene una restricción local del rango asociado a ella. Por ejemplo, la clase persona puede tener una característica llamada “Tiene\_hija” restringido allValuesFrom para la clase mujer. Esto significa que si el “Tiene\_hija” de la característica relaciona a una persona individual Louise con Débora individual, después de esto se puede deducir que Débora es un caso de la clase mujer.

- **SomeValuesFrom:**

El “someValuesFrom” de la restricción se indica en una característica con respecto a una clase. Una clase particular puede tener una restricción en una característica que por lo menos un valor para esa característica de cierto tipo.

# Descripción Lengua (Lite)

## Restricciones de Cardinalidad en OWL Lite

- **Restricciones locales**, puesto que se indican en características con respecto a una clase particular
- Permiten solamente declaraciones referentes a cardinalidades del **valor 0 o 1**.
  - **Cardinalidad**: La Cardinalidad se proporciona como conveniencia cuando es útil indicar que una característica en una clase tiene el minCardinalidad 0 y maxCardinalidad 0 o el minCardinalidad 1 y el maxCardinalidad 1. Por ejemplo, la clase persona tiene exactamente un valor para el “Tiene\_madre\_nacimiento” de la característica. De esto se puede deducir que ninguno de los dos casos individuales distintos de la clase madre pueden ser valores para la característica del “Tiene\_madre\_nacimiento” de la misma persona.



# Descripción Lengua (Lite)

## Restricciones de Cardinalidad en OWL Lite

- **MaxCardinalidad:** Si un maxCardinalidad de 1 se indica en una característica con respecto a una clase, entonces cualquier caso de esa clase será relacionado con a lo más un individuo por esa característica. (Voto)

En cambio, un buen ejemplo de maxcardinalidad de 0, es “Tiene\_esposo” en la clase “Divorciado”, donde no aceptamos ningún valor.

- **MinCardinalidad:** Si un minCardinalidad de 1 se indica en una característica con respecto a una clase, entonces cualquier caso de esa clase será relacionado con por lo menos un individuo por esa característica. En caso de que dicha cardinalidad fuese 0, apenas tendríamos información para saber si dicha característica es opcional.

# Descripción OWL DL y Completo

- OWL Y OWL DL usan mismo vocabulario, aunque **DL mas restringido.**
- DL: Una clase no puede también ser un individuo o la característica, una característica no puede también ser un individuo o una clase. (OWL SI)
- DL: requiere que las características sean ObjectProperties o DatatypeProperties. (En OWL pueden ser incluso clases)
  - DatatypeProperties son relaciones entre los casos de clases y de literales de RDF y de datatypes del esquema de XML
  - ObjectProperties es relaciones entre los casos de dos clases

# Descripción OWL DL y Completo

- **OneOf:** (clases enumeradas): La clase de “dias\_semana” puede ser descrita simplemente enumerando los individuos lunes, martes, miércoles, jueves, viernes, sábado, domingo.
- **HasValue:** (valores de característica): Los casos de la clase de “Ciudadano\_Español” se pueden caracterizar como esa gente que tenga “España” como valor de su nacionalidad. (El valor de la nacionalidad, “España”, es un caso de la clase de nacionalidades).
- **DisjointWith:** Hombre y la Mujer pueden ser indicados para ser clases de desunión. Si A es un caso del hombre, después A no es un caso de la mujer.
- **MinCardinality, maxCardinality, cardinality (cardinalidad completa):** Mientras que en el OWL Lite, las cardinalidades se restringen a exactamente 1 o 0, OWL permite las declaraciones de las cardinalidades en números enteros no negativos arbitrarios.

# Descripción OWL DL y Completo

- **UnionOf, complementOf, intersectionOf (combinaciones booleanas):** Usando el unionOf, podemos indicar que una clase contiene los casos “ciudadano\_aleman” o “ciudadano\_Español”. Usando el complementOf, podríamos indicar que los niños no son “ciudadanos\_adultos”. (es decir los niños de la clase son una subclase del complemento de “ciudadanos\_adultos”). La ciudadanía de la unión europea se podía describir como la unión de la ciudadanía de todos los Estados miembro.
- **Clases complejas:** En muchas construcciones, el OWL Lite restringe la sintaxis para escoger nombres de la clase (en declaraciones del subClassOf o de los equivalentClass). El OWL amplía por completo esta restricción para permitir arbitrariamente descripciones complejas de la clase, clases enumeradas que consisten en, restricciones de la característica, y combinaciones booleanas. También, el OWL permite que las clases sean utilizadas como casos (OWL DL y OWL Lite no).