

Tema 6: Ontologías

Anabel Gómez Ríos

2 de junio de 2016

1. Introducción

Motivación: ¿Cómo interpreto la frase *El agente cometió un error*? ¿A qué agente estoy haciendo referencia?

Hay varios problemas con el manejo de la información: inconsistencia, incompatibilidad, falta de completitud y falta de límites.

2. Web semántica vs web actual

La web actual representa la información utilizando documentos en lenguaje natural con poca estructura. Es de fácil comprensión por los humanos (html sólo define presentación) aunque es difícil soportar el procesamiento automático.

Algunas alternativas para facilitar el procesamiento de la información en la web son las siguientes:

1. Máquinas más inteligentes que comprendan el significado de la información que hay en la web. Procesamiento de lenguaje natural, reconocimiento de imágenes, etc.
2. Información más inteligente → representar la información de modo que sea sencilla de comprender a las máquinas: expresar contenidos en un formato procesable automáticamente y uso de metainformación.

¿Por qué ontologías? Hay un incremento de la necesidad de capturar el conocimiento, definir vocabulario común (consensuado), compartir entendimiento común (reutilizable) e interpretación y manipulación automática. Todo esto lleva al uso de ontologías.

3. Orígenes de las ontologías

En filosofía una ontología es una parte de la metafísica que trata del ser en general y de sus propiedades trascendentales. En ciencias de la computación algo existe si puede ser representado, descrito y definido (formalmente) para ser interpretado por una máquina.

Definiciones:

1. Una especificación de una conceptualización, una descripción de los conceptos y relaciones que pueden existir para un agente o una comunidad de agentes. T.R. Gruber
2. Una ontología es un catálogo de los tipos de cosas que se asume que existen en un dominio de interés D desde la perspectiva de una persona, la cual usa un lenguaje L para hablar sobre D. John F. Sowa
3. Una ontología trata sobre la exacta descripción de las cosas y sus relaciones. World Wide Web Consortium (W3C).

4. Componentes de una ontología

Está compuesta por conceptos o clases, instancias o individuos, propiedades o relaciones y axiomas. Los conceptos son colecciones de individuos (por ejemplo persona, árbol...). Las instancias son objetos del mundo (por ejemplo Joaquín) o valores (integer, string,...). Las propiedades describen las relaciones entre los conceptos. Hay dos tipos de propiedades, de objeto y de datos. Los axiomas son restricciones y meta-información sobre las relaciones. Definen el significado y permiten razonar con la ontología.

¿QUÉ APORTAN LAS ONTOLOGÍAS? **Clasificación y consultas automáticas:** localizar un concepto o un conjunto de conceptos basándose en la descripción y/o las relaciones e intercambio de vocabulario entre dominios. **Legible por humanos y computadoras.** **Chequeo de consistencia.** **Razonamiento automático:** Subsumpción: inferir que la clase A es más general que la clase B. Reconomiento: inferir que la instancia X debe ser un hijo de la clase B.

5. Clasificación de ontologías

1. **Ontologías genéricas.** Conceptos comunes de alto nivel, como por ejemplo *Individuo*, *Conjunto* o *Sustancia*. Son útiles para la reutilización e importantes cuando generamos o analizamos expresiones de LN.
2. **Ontologías de dominio.** Utilizan conocimiento específico de dominio y son generalizaciones del dominio.
3. **Ontologías orientadas a tareas.** Utilizan conocimiento específico de tareas y son generalización de tareas.
4. **Ontologías de aplicación.** Utilizan conceptos comunes de bajo nivel. Combinan, integran y extienden todas las sub-ontologías para una aplicación.

Las ontologías se usan en web semánticas, inteligencia artificial e IA distribuida, en sistemas expertos y KBS, especificación formal de requerimientos y estándares. Hay lenguajes para la definición de ontologías: RDF, RDF Schema y OWL.

6. Estándares básicos

Unicode es el estándar que proporciona el medio por el cual codificar un texto en cualquier forma e idioma. IRI (International Resource Identifier) es la cadena de caracteres que identifica inequívocamente un recurso (servicio, página, documento, et.) físico o abstracto. IRI identifica el recurso, pero no tiene por qué localizar su ubicación (URL). XML es un meta-lenguaje extensible de etiquetas usado para el intercambio de datos en la web. El uso de etiquetas con significado es intuitivo para humanos, pero no para las máquinas. XML estandariza formato, no significado, el nombre de las etiquetas XML no ofrece semántica por sí mismo.

RDF (Resource Definition Format) es un estándar W3C para describir recursos (cualquier concepto que tenga una URI) en la web. Utiliza un formato común para describir información que puede ser leído y entendido por una aplicación informática. Permite representar conceptos y relaciones mediante un conjunto de tripletas. Una triplete describe propiedades de un recurso identificado por una IRI: representa un documento o parte de él, o de una colección de documentos o un objeto. Una propiedad es siempre una IRI (predefinida y con significado preestablecido). Cada triplete combina un recurso (sujeto), una propiedad (predicado) y un valor para la propiedad (objeto).

RDF permite usar vocabularios semánticos definidos por expertos para describir recursos (dublin core para descripción de recursos digitales o FOAF, ontología para descripción de personas), es representable en forma de documentos XML (serialización RDF/XML) y da la posibilidad de usar lenguajes de consulta sobre tripletas RDF, como SPARQL (sintaxis tipo SQL sobre bases de datos de tripletas) o RDF.DBPEDIA (versión estructurada con tuplas RDF de la wikipedia).

RDF Schema es un lenguaje extensible que proporciona los elementos básicos para crear ontologías (vocabularios semánticos RDF). Permite definir clases, relaciones entre clases, restricciones sobre propiedades, etc.

1. `rdfs:Class` para declarar recursos como clases para otros recursos.
2. `rdfs:subClassOf` para definir jerarquías (relaciona clase con superclase).
3. `rdfs:property` para definir un subconjunto de recursos RDF que son propiedades.
4. `rdfs:subPropertyOf` definir jerarquías de propiedades.
5. `rdfs:domain` dominio de una propiedad (clase de recursos que aparecen como sujetos en las tripletas de ese predicado).
6. `rdfs:range` rango de una propiedad (clase de recursos que aparecen como objetos en las tripletas de ese predicado).

RDF Schema define el significado de los términos usados en las tripletas RDF.

OWL (Ontology Web Lenguaje) extiende a RDFS para permitir la expresión de relaciones complejas entre clases RDFS, y mayor precisión en las restricciones de clases y de propiedades. Deriva de la fusión de los lenguajes de ontologías DAML y OIL. Permite expresar relaciones entre clases, expresar y restringir clases (rango, dominio) y expresar y restringir

propiedades (cardinalidad).

Hay tres variantes/sublenguajes, de menor a mayor potencia expresiva:

1. OWL-lite: versión simplificada (representación de jerarquías simples)
2. OWL-DL: incluye constructores tomados de Description Logics (DL). Busca un compromiso entre máxima expresividad y eficiencia computacional (sólo constructores decidibles de DL).
3. OWL-full: soporte completo de constructores DL.

Tipos de propiedades:

1. Propiedades de tipo de dato: relaciones entre instancias de clases y literales RDF y XLM. Esquema de tipo de datos. Por ejemplo, la clase *Mascota* tiene la propiedad de datos *Nombre*.
2. Relaciones entre instancias de dos clases. Por ejemplo, la clase *DueñoDeMascota* tiene una propiedad objeto *esDueñoDe*.

Las propiedades ligam individuos de un dominio a individuos de un rango.

Constructores OWL: intersectionOf, unionOf, complementOf, oneOf, allValuesFrom, someValuesFrom, maxCardinality, minCardinality. AllValuesFrom es el cuantificador universal, para cada instancia de la clase que tiene instancias de una determinada propiedad, los valores de la propiedad son todos miembros de la clase indicada por la cláusula allValuesFrom. SomeValuesFrom es el cuantificador existencial, al menos uno de los valores de la propiedad debe ser miembro de la clase indicada por la cláusula someValuesFrom.

Axiomas OWL: subClassOf, equivalentClass, disjointWith, sameIndividualAs, differentFrom, subPropertyOf, equivalentProperty, inverseOf, transitiveProperty, functionalProperty, inverseFunctionalProperty.

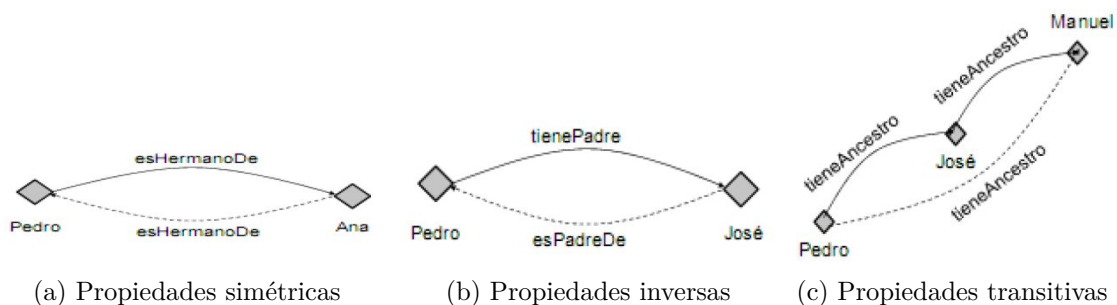


Figura 1: Propiedades

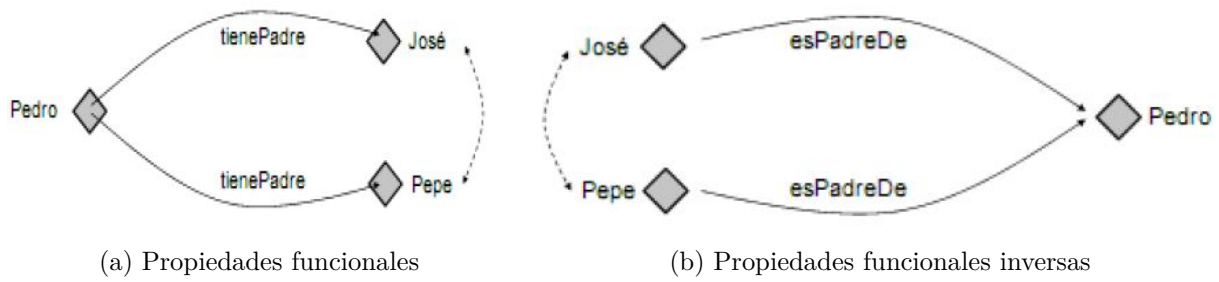


Figura 2: Propiedades

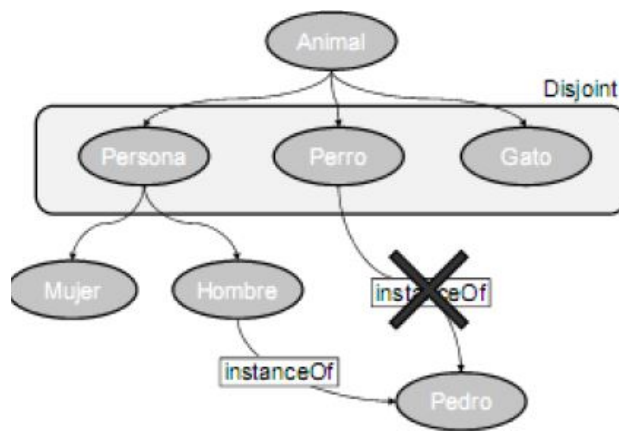


Figura 3: Disjunción

7. Razonamiento con ontologías

1. Clasificación automática

- $\text{DueñoDeMascota} \equiv \text{Persona} \sqcap \exists \text{esDueñoDe some Mascota}$
 $\sqcap \exists \text{quiereA some Mascota}$
 - $\text{Mascota} \equiv \text{Animal} \sqcap \exists \text{esQueridoPor some DueñoDeAnimal}$
 $\sqcap \exists \text{tieneDueño some DueñoDeAnimal}$
 - $\text{Gato} \equiv \text{Animal} \sqcap \forall \text{come only AlimentoCarnívoro}$
 - $\text{AnimalCarnívoro} \equiv \forall \text{come only AlimentoCarnívoro}$
- $\Rightarrow \text{DueñoDeMascota} \sqsubseteq \text{DueñoDeAnimal}$
 $\Rightarrow \text{Gato} \sqsubseteq \text{AnimalCarnívoro}$
- (a) (b)

Figura 4: Ejemplos de clasificación automática

2. Clasificación de instancias

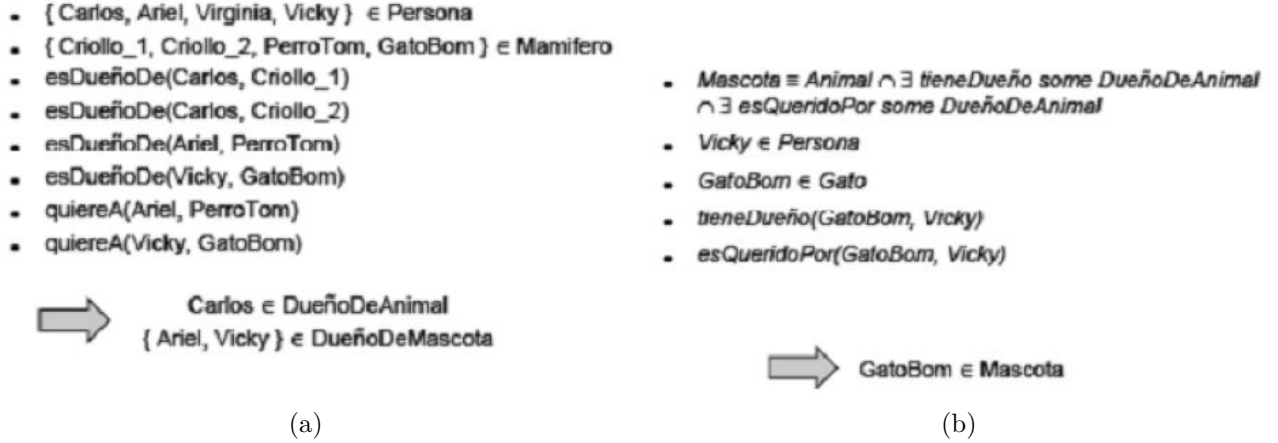


Figura 5: Ejemplo de clasificación de instancias

3. Detección de redundancia

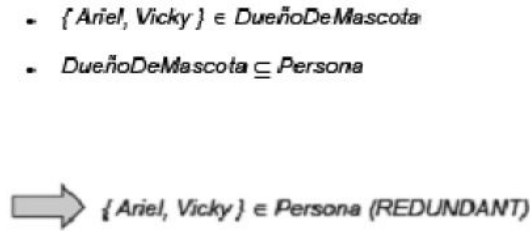


Figura 6: Ejemplo de detección de redundancia

4. Chequeo de consistencia (Disjoint y restricciones)

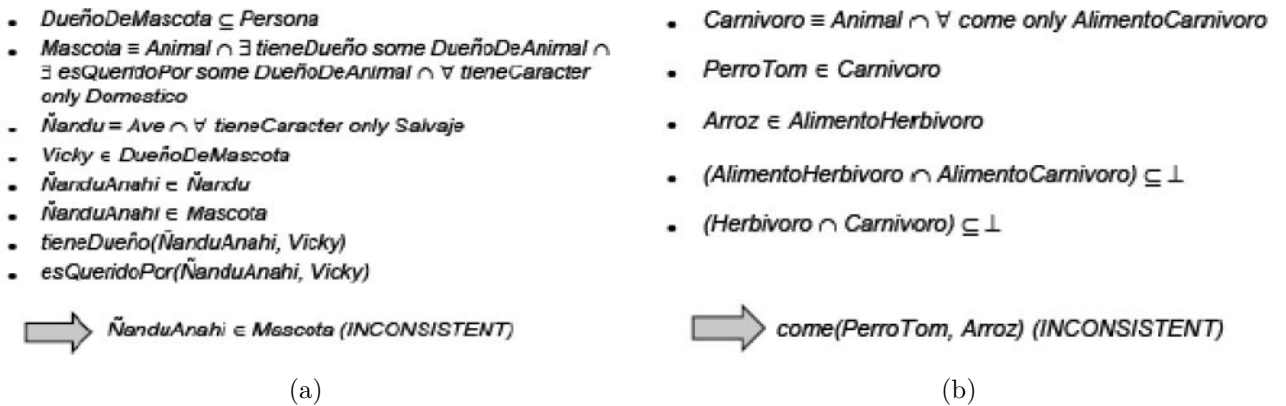


Figura 7: Ejemplos de chequeo de consistencia

8. Conclusiones

Las ontologías:

1. Definen vocabulario común.
2. Crean entendimiento compartido.
3. Proveen acceso común al conocimiento.
4. Permiten la extracción de nuevo conocimiento implícito a través de razonamiento automático.
5. Permiten compartir, integrar y re-utilizar conocimiento.
6. Proveen conocimiento entendible por humanos y computadoras.