

Ontologies

1. Motivation

1.1 Communicating Agents

Entendimiento mutuo:

- Traducción entre lenguajes de representación.
- Compartir el contenido semántico del lenguaje.

Tres componentes en la comunicación:

- Protocolo de interacción: ¿Cómo se estructuran las conversaciones/diálogos?
- Lenguaje de comunicación: ¿Qué significa cada mensaje?
- Protocolo de transporte: ¿Cómo envían y reciben los mensajes los agentes?

Los agentes pueden ser considerados como Bases de Conocimiento (virtuales). 3 niveles de representación:

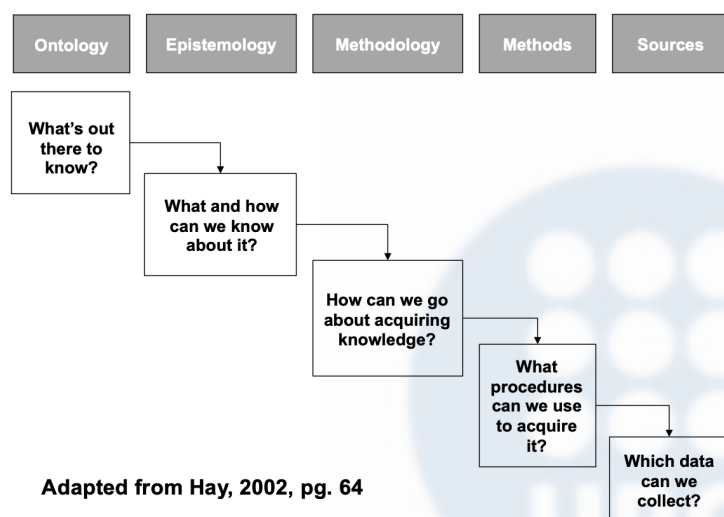
- Un lenguaje/formalismo para representar el conocimiento del dominio: **Ontología**.
- Un lenguaje para expresar proposiciones (para intercambiar conocimientos): **Lenguaje del contenido** (para mensajes).
- Un lenguaje para expresar actitudes para esas proposiciones: **Lenguaje de comunicación del agente** (para idiomas).

1.2 Ontology and Epistemology

Ontología: la cuestión de cómo es la realidad, los elementos básicos que contiene.

Epistemología: estudio de los criterios por los cuales podemos saber qué constituye y qué no constituye conocimiento garantizado o científico.

Metodología: teorías de recopilación de conocimiento, ¿cómo podemos saber lo que somos capaces de saber?



La forma en que pensamos que el mundo es (**ontología**) influye: lo que pensamos se puede saber sobre él (**epistemología**); cómo pensamos que se puede investigar (**metodología y técnicas de investigación**); los tipos de teorías que creemos que se pueden construir al respecto.

Si parece un pato, nada como un pato y grazna como un pato, entonces probablemente sea un pato.

1.3 Ontologies (origin)

El desarrollo de la ontología está directamente relacionado con la Filosofía.

Aristóteles acuñó el término **Categoría** como la palabra para describir las diferentes clases en las que se pueden dividir las cosas en el mundo.

El primer sistema de clasificación de Aristóteles es de seres (τὰ ὄντα). La división procede a través de dos conceptos:

- (1) **dicho de (said-of)**
- (2) **presente en (present-in)**

Los seres que se dicen de otros (said-of others) son **universales**, mientras que los que no se dicen de otros son **particulares**.

Los seres que están presentes-en (present-in) otros son **accidentales**, mientras que los que no están presentes-en otros son **no accidentales**.

Los seres no accidentales que son universales se describen más naturalmente como **esenciales**.

El término Ontología es bastante moderno (siglo XIX). Proviene del griego Ontos (de ser/ser) y Logos (palabra). Su uso empezó para diferenciar los estudios sobre las categorías del ser de aquellas categorías en biología. De hecho, la categorización es una tarea común en varias áreas de la ciencia (Filosofía, Biología, Medicina, Lingüística...)

1.4 Ontologies (definitions)

“Una ontología define los términos y relaciones básicos que comprenden el vocabulario de un área temática, así como las reglas para combinar términos y relaciones para definir extensiones del vocabulario”

“Una Ontología es una especificación explícita de una conceptualización”

[...] una ontología es una descripción (como una especificación formal de un programa) de los conceptos y relaciones que pueden existir para un agente o una comunidad de agentes.

La ontología tiene como objetivo estudiar las categorías que existen en un dominio dado.

El resultado de este estudio es una ontología: un catálogo de los diferentes tipos de objetos que asumimos como existentes en un dominio D dado, desde la perspectiva de alguien que usa un lenguaje L para hablar de D.

Los elementos en ontologías representan **predicados**, **constantes**, **conceptos** y **relaciones**. Una ontología puede verse como el vocabulario que los agentes necesitan usar para hablar sobre un dominio dado.

1.5 Ontologies (some assumptions)

- Es posible que nunca se encuentre la **verdad absoluta**.
- Los **datos**, la **evidencia** y las **consideraciones racionales** dan forma al conocimiento.
- La investigación es un proceso de hacer afirmaciones y luego probarlas, refinarlas o abandonar algunas de ellas por otras afirmaciones más justificadas.
- La investigación busca desarrollar afirmaciones verdaderas y relevantes que puedan servir para explicar la situación de interés o que describa la relación causal de interés.
- Los investigadores deben examinar sus métodos y conclusiones y controlar o limitar el sesgo (limit bias).

2. Several views of ontologies

2.1 Ontologies vs. data models

No hay una línea estricta en el medio, pero las ontologías son:

- más general.
- más reutilizables.
- destinado a múltiples propósitos, objetivos y usuarios.
- más fácil de compartir.
- tomar posición sobre la semántica de los conceptos (en oposición a la mera estructura e integridad).

En **Ciencias de la Información**: “Una ontología es una descripción (como una especificación formal de un programa) de los conceptos y relaciones que pueden existir para un agente o una comunidad de agentes”. (Tom Gruber)

En **Filosofía**: “La ontología es la ciencia de lo que es, de los tipos y estructuras de objetos, propiedades, eventos, procesos y relaciones en cada área de la realidad”. (Barry Smith)

2.2 Ontologies in Logics

La capacidad de producir deducciones a partir de la información representada en una ontología viene dada por una Lógica.

La lógica por sí misma tiende a ser neutral con respecto al significado:

si $P \rightarrow Q$ y P , entonces Q

El razonamiento anterior no se refiere a nada si no le damos significado a los átomos (P = "llueve", Q = "estoy mojado").

Cuando combinamos una lógica con una ontología, la ontología proporciona al formalismo lógico la capacidad de expresar significados con sus enunciados.

Esto es especialmente útil en lógicas de alto nivel.

Lógica de primer orden:

- $\text{ciudadano}(x) \rightarrow \text{humano}(x)$
- ¿Qué significan 'ciudadano' y 'humano'?

Lógica dinámica:

- $[\text{pedido}(x, \text{artículo}, t)] \text{ack-order-received}(x, \text{artículo})$
- ¿Cuál es el significado de la acción "order"?
- ¿Qué significa el predicado 'ack-order-received'?

Lógica deóntica:

- Si $\text{won}(x, \text{auct}, \text{item}, \text{pr}) \rightarrow O(\text{pay}(x, \text{item}, \text{pr}))$
- ¿Cuál es el significado del predicado 'won'?
- ¿Cuál es el significado de su segundo parámetro?
- ¿Qué significa 'pay'?
- ¿Hay más de una forma de pago (pay)?

Description Logic (Descripción Lógicas)

- superar las ambigüedades de las primeras redes y marcos semánticos.
- realizado por primera vez en el sistema KL-One [Brachman y Schmolze, 1985].
- Bien estudiado y decidible (la mayoría de los idiomas DL).
- Estrecho acoplamiento entre la teoría y la práctica.

TBox and ABox

TBox (terminología):

- El vocabulario de un dominio de aplicación:
 - Conceptos: conjuntos de individuos. (Persona, Hembra, Madre)
 - Roles: relaciones binarias entre individuos. (hasChild, lo que significa que una persona es hijo de otra)

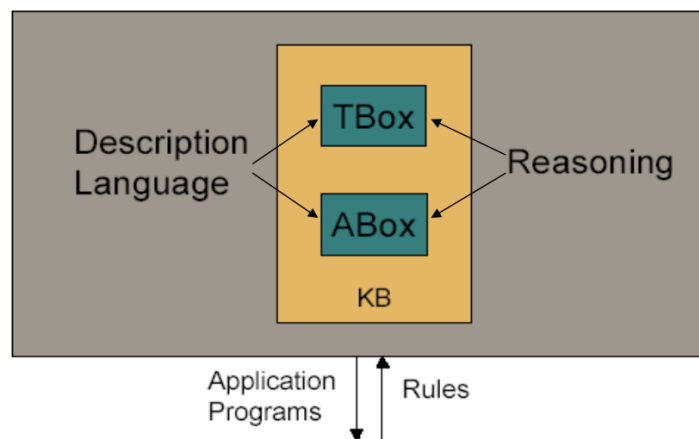
ABox (afirmaciones):

- Sobre individuos nombrados en términos de este vocabulario
- Ejemplo: Isabel y Carlos son Personas. Escribimos esto como $\text{Persona}(\text{Elizabeth})$ y $\text{Persona}(\text{Charles})$.

Los individuos, como "myCar", tienen atributos, como "color", y esos atributos tienen valores, como "red". Cuando esto sucede, decimos que el rojo es el atributo colorOf de myCar. Escribimos esto como $\text{colorOf}(\text{myCar}, \text{red})$.

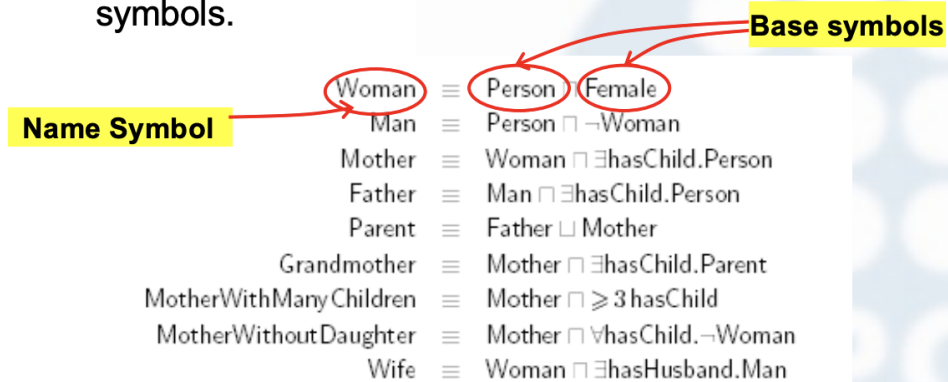
Base de conocimientos = TBox + ABox

Architecture of a DL System



Name Symbols vs Base Symbols

- atomic concepts occurring in a TBox T can be divided into two sets, **name symbols** N_T (or defined concepts) and **base symbols** B_T (or primitive concepts, occur only on the right-hand side)
- a **base interpretation** for T only interprets the base symbols.



2.3 Ontologies in Knowledge Engineering

- Permitir que los expertos humanos modelen y analicen su experiencia en resolución de problemas; este análisis de experiencia debe ser adecuado para alimentar el proceso de diseño del sistema de conocimiento conceptual.
- Permitir y apoyar la reutilización de bases de conocimiento construidas previamente. Reutilizar el significado de descompilación del conocimiento.
- Las ontologías dividen el conocimiento del dominio del conocimiento operativo: permite desarrollar de forma independiente las técnicas y algoritmos para resolver un problema a partir del conocimiento concreto sobre el problema.
- Permiten el análisis sobre el conocimiento del dominio: una vez que tenemos una especificación de conocimiento, se puede analizar mediante métodos formales (corrección, completitud...).

2.4 Ontologies in Distributed AI

Permitir compartir una interpretación de la estructura de la información entre personas/agentes:

- Al crear una ontología sobre un dominio, los agentes pueden entenderse entre sí (sin ambigüedades) y saber qué quiere decir el otro con cada mensaje.

Para permitir la reutilización del conocimiento:

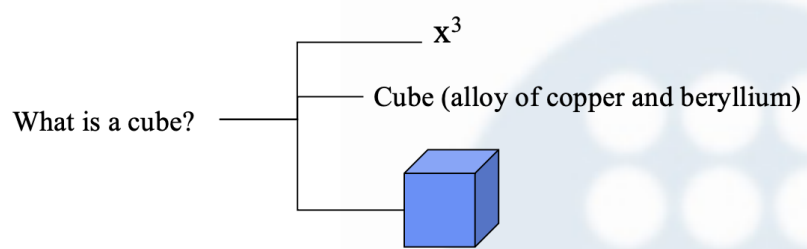
- Cree una descripción de dominio que puedan usar otras aplicaciones que deberían usar/compartir conocimientos sobre ese dominio.

Hacer explícitas las interpretaciones sobre el dominio:

- Se pueden comparar interpretaciones sobre conceptos, predicados... Si surgen conflictos, se puede acordar una interpretación común.

3. Ontological agreements

Acuerdos para utilizar el vocabulario de forma coherente y consistente.



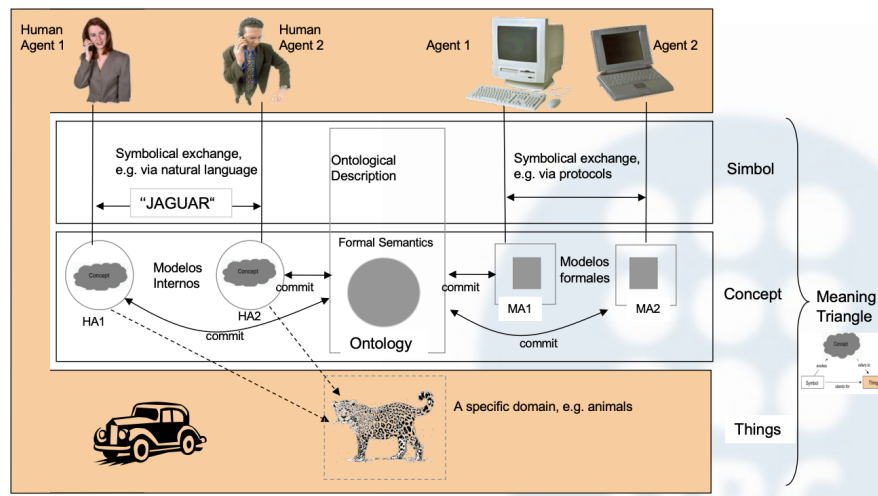
Why Ontologies?

- Para compartir un entendimiento común de la estructura de la información entre personas o agentes de software.
- Para permitir la reutilización del conocimiento del dominio.
- Para hacer explícitos los supuestos de dominio.
- Separar el conocimiento del dominio del conocimiento operativo.
- Para analizar el conocimiento del dominio.

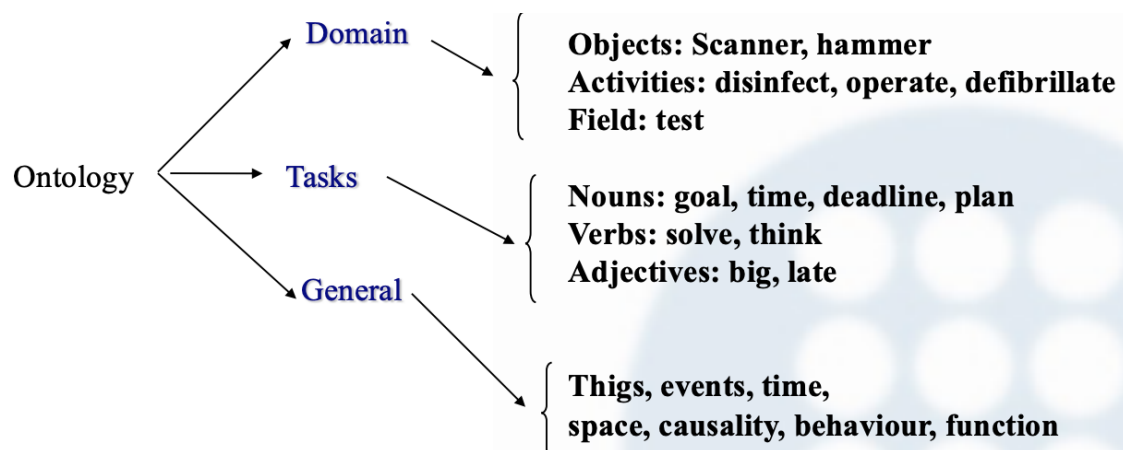
Ontological agreements

Man-machine communication

[Maedche et al., 2002]

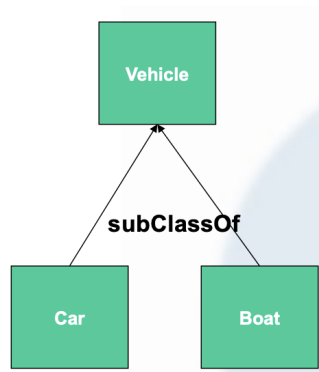


4. Ontology types & examples

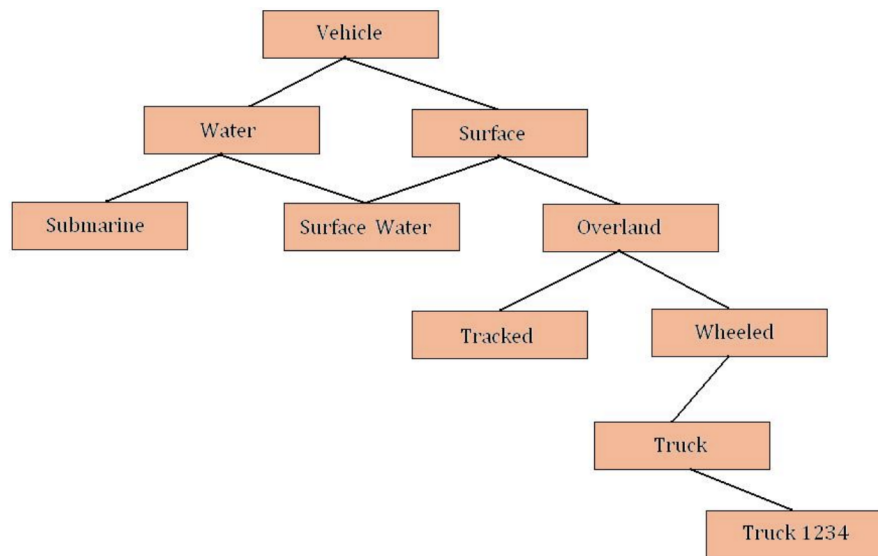


Ask & Tell ontologies : limited subset of what can be queried and answered
Meta-Ontologies

A simple example ontology:



Hierarchy in ontology:



4.1 Fundamental Design Questions

Sintagmático vs Paradigmático:

- Sintagmático –cuando las palabras aparecen juntas en una unidad.
- Paradigmático –si las palabras están enlazadas en un recurso léxico

Cuando escuchamos una palabra, muchas palabras vienen a nuestra mente por asociación.

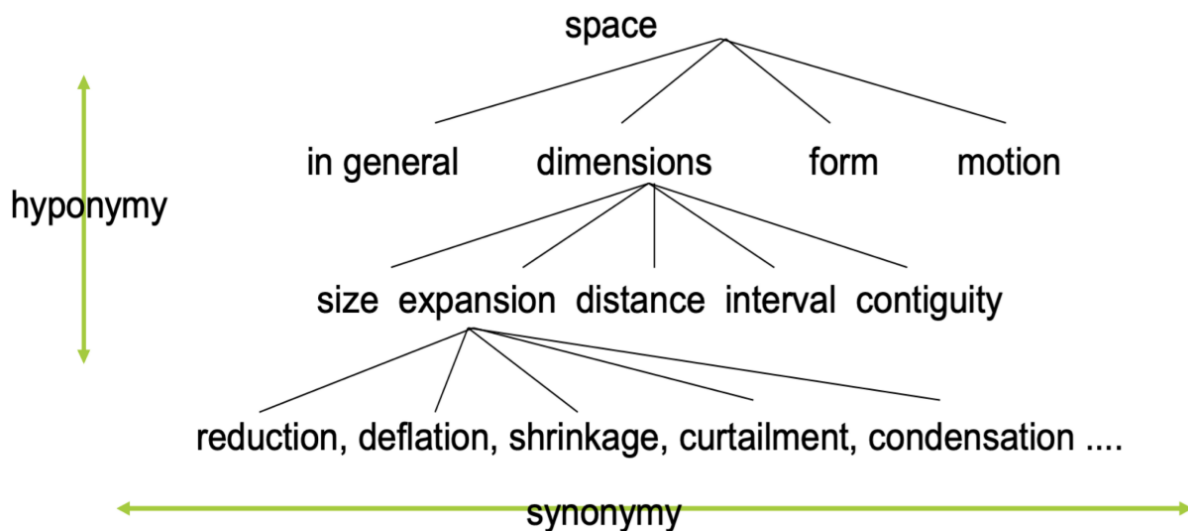
Por ejemplo, para gato:

- animal, mamífero –Paradigmático.
- maullido, ronroneo, peludo –Sintagmático.

4.2 Major Lexical Relations

Synonymy, Polysemy, Metonymy, Hyponymy/Hypernymy, Meronymy/Holonymy, Antonymy...

Hyponymy vs Synonymy



Hyponymy (more detailed), synonymy (same meaning)

Hyponymy (ISA relation):

- Relacionado con el nivel Superordinado y Subordinado
- Categorías:
 - Hipónimo (petirrojo, pájaro).
 - Hipónimo (emú, pájaro).
 - Hipónimo (pájaro, animal).
 - Hiperónimo (animal, pájaro).

A es un **hiperónimo** de B si B es un tipo de A

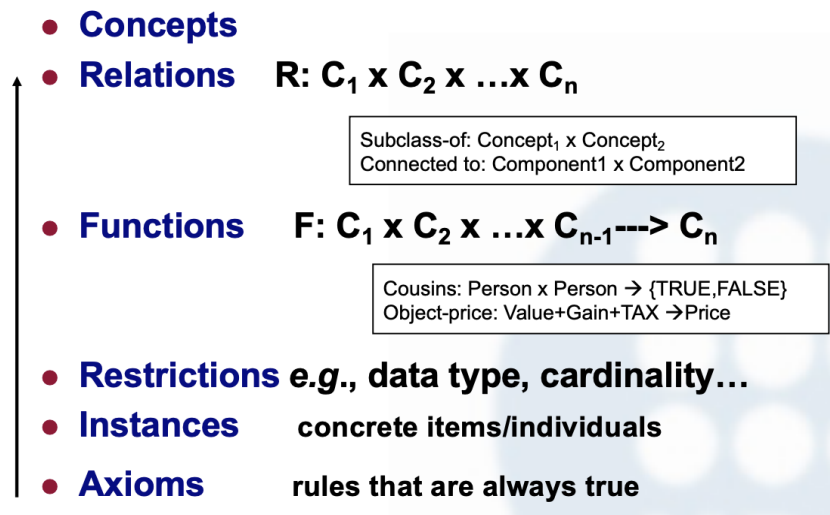
A es un **hipónimo** de B si A es un tipo de B

5. Ontology development

5.1 Elements

- **Conceptos:** sirven para modelar varios elementos incluyendo tareas, funciones, acciones, estrategias, planes, etc. Algunos lenguajes de ontología se refieren a ellos como clases compuestas por propiedades.
- **Relaciones:** modelan un tipo de interacción entre conceptos de dominio.
- **Funciones:** un tipo especial de relación.

- **Restricciones:** reglas que pueden restringir el dominio de valores para algunos conceptos y relaciones.
- **Instancias:** constituyen los elementos/individuos concretos representados por la ontología.
- **Axiomas.**



5.2 Design and development

Crear una ontología requiere:

- Definir las clases en el dominio.
- Organizar las clases en una jerarquía taxonómica.
- Definir las propiedades de cada clase e incluir cualquier restricción en sus valores.
- Asignar valores a cada propiedad para crear instancias.

Es importante tener en cuenta que:

- No existe una metodología estándar única para desarrollar ontologías.
- No existe un único método correcto para modelar un dominio. La mejor solución depende de la aplicación/dominio dado.
- El proceso de desarrollo tiende a ser iterativo.
- Los objetos en la ontología deben estar cerca de los objetos y relaciones usados para describir el dominio. Suelen corresponder a sustantivos y verbos que aparecen en oraciones que describen el dominio.

Fases en el desarrollo de ontologías (en la mayoría de las metodologías están presentes las siguientes 7 fases):

- **Fase 1:** Determinar el dominio y la cobertura de la ontología:
 - ¿Cuál es el dominio a cubrir por la ontología?
 - ¿Para qué usaremos la ontología?
 - ¿Qué tipo de preguntas debe responder la ontología?
 - ¿Quién usará y mantendrá la ontología?
- **Fase 2:** Considere reutilizar ontologías existentes:
 - Las ontologías se construyen para comunicar el conocimiento en un dominio dado. Por lo tanto, están contruidos para ser compartidos.
 - No es necesario rehacer un trabajo ya hecho por otra persona: si ya existe una (buena) ontología para nuestro dominio, podemos usarla.
- **Fase 3:** enumerar los términos importantes en la ontología:
 - Escribir una lista de los términos útiles para referirse a nuestro dominio, creando oraciones que podríamos usar para preguntar cosas sobre él o para explicar/describir el dominio a otros.
 - ¿Qué propiedades tienen estas cosas/términos?
 - ¿Qué nos gustaría decir sobre ellos?
- **Fase 4:** Definir las clases/conceptos y su jerarquía:

Hay varios enfoques:

 - **Top-down:** definimos los conceptos más generales y luego los refinamos mediante un proceso de especialización.
 - **Bottom-up:** definimos los conceptos más específicos y luego los agrupamos según propiedades comunes, en un proceso de generalización.
 - **Bidireccional:** definimos los conceptos más relevantes y luego generalizamos/especializamos para completar la ontología.

Ninguno de estos métodos es el mejor, depende del dominio y la granularidad objetivo de nuestro modelo.
- **Fase 5:** Definir los atributos y relaciones para cada clase:
 - Debe describir la estructura interna de nuestras clases.
 - Definir una lista de características y a qué clase corresponden.
 - Podemos tener diferentes tipos de propiedad:
 - Propiedades descriptivas, p. calidad.
 - Identificar propiedades, p. nombres.
 - Partes.
 - Relaciones con otras instancias de clase.

- Las propiedades deben asignarse a la clase más general, las subclases heredarán estas propiedades.
- **Fase 6:** Definir las características de los atributos/relaciones:
 - Para atributos:
 - Cardinalidad (número de valores permitidos).
 - Dominios de tipo y valor.
 - Valores predeterminados.
 - Obligatorio/opcional.
 - Para las relaciones:
 - Cardinalidad.
 - Rango.
- **Fase 7:** Crear instancias:
 - Si es necesario, crear instancias que formarán parte de la ontología en todo momento.

5.3 Ontology development principles

Claridad y objetividad:

- Una ontología debe proporcionar al usuario el significado del término definido de forma objetiva y en una representación cercana al lenguaje natural.

Integridad:

- Las definiciones deben expresarse en términos de lo necesario y suficiente.

Coherencia:

- Asegurar (permitir) que las inferencias derivadas de la ontología sean consistentes con las definiciones.

Máxima extensibilidad monótona:

- Se pueden incluir nuevos términos generales o especializados en la ontología de tal manera que no requiera la revisión de las definiciones existentes.

Principio de distinción ontológica:

- Las clases en una ontología deben ser disjuntas.

Diversificación:

- Diversificación de las jerarquías incluidas para aumentar el poder de los mecanismos de herencia múltiple.

Modularidad:

- Reducir el acoplamiento entre módulos.

Estandarización de nombres:

- Estandarización de nombres donde sea posible.

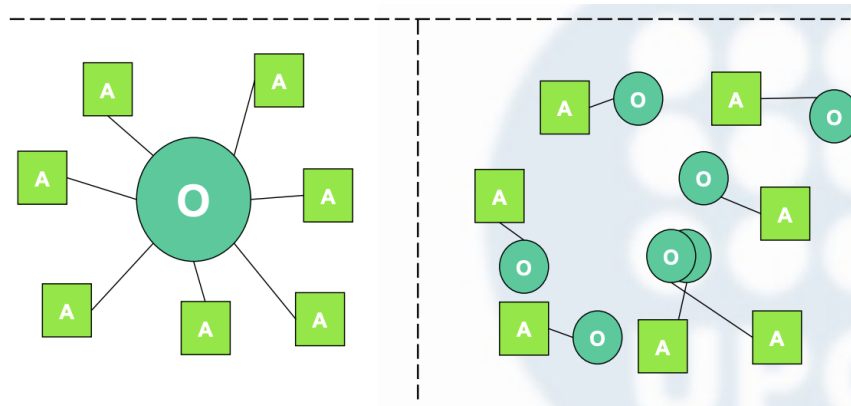
Minimizar la distancia semántica:

- Minimizar la distancia semántica entre conceptos estrechamente relacionados. Los conceptos similares se agruparán y representarán utilizando las mismas primitivas.

Size and scope of an ontology:

Dos extremos (la realidad algo intermedio):

- Una gran ontología que lo captura todo.
- Una ontología (pequeña) para cada aplicación específica.



Una gran ontología que lo captura todo:

Beneficios:

- pocas o ninguna inconsistencia interna.
- para un desarrollador de aplicaciones más fácil de encontrar.
- documentación uniforme.
- ¡Menos trabajo superpuesto!

Inconvenientes:

- ¿Quién lo mantiene?
- ¿Quién es responsable?
- Pesado y lento de usar (tanto para el usuario humano como para la aplicación).
- Difícil tener en cuenta las opiniones y los deseos de todos en el momento del diseño y la actualización.
- Ejemplo: Ciclo

Una ontología (pequeña) para cada aplicación específica:

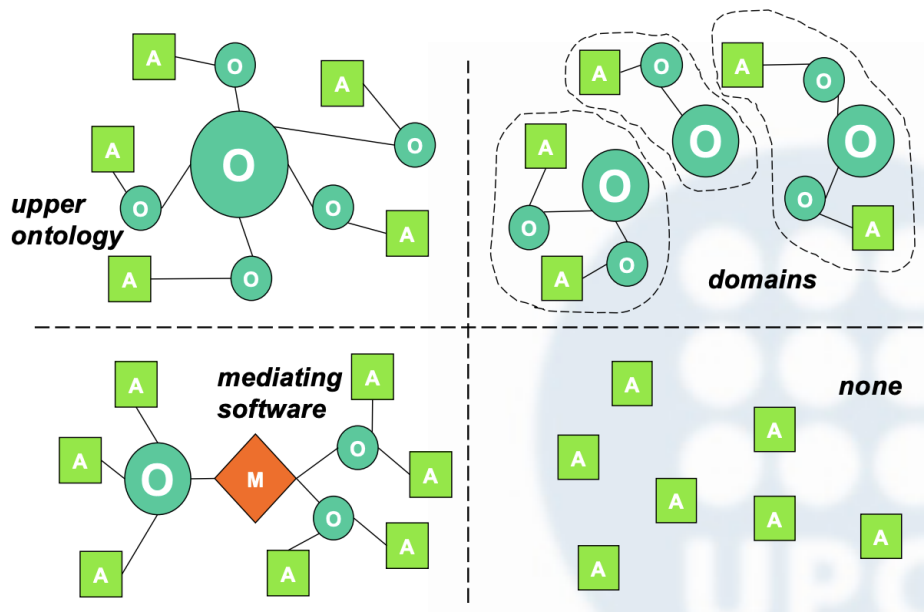
Beneficios:

- Las ontologías se adaptan bien a las demandas de la aplicación.
- Más rápido de usar.
- Más fácil de formar una imagen completa de una ontología (menos conceptos e interrelaciones).

Inconvenientes:

- Diferentes ontologías no encajan sin ninguna de las dos.
 - Organismo central de coordinación o software de alineación de ontologías.
 - Coincidencia.
- Trabajo superpuesto: los mismos conceptos definidos en múltiples ontologías, ya sea de la misma manera o (¡peor aún!) de manera diferente.

Algunas mixtas:



5.4 Some advices on ontology development

- No incluir versiones singulares y plurales de una palabra (la mejor política es usar solo nombres en singular o plural).
- Los nombres no son clases:
 - Debemos distinguir el nombre de la clase que le damos, podemos tener sinónimos, pero todos representan la misma clase.
- Asegurarse de que la jerarquía se haya creado correctamente.
- Ver relaciones de transitividad y ver si son correctas.
- Evitar ciclos en la jerarquía.
- Todas las subclases de una clase deben tener el mismo nivel de generalidad.
- No existe un criterio con respecto al número de clases, la experiencia es que un número entre dos y doce, pero usted sugeriría que necesitamos estructurar las clases agregando más niveles. (¿Cuándo introducir nuevas clases?)
- Muchas veces es incómodo navegar por jerarquías demasiado planas o muy profundas, se debe elegir un punto medio, algunas indicaciones serían.

- Las nuevas clases tienen propiedades adicionales que no tienen la superclase.
- Tienen diferentes restricciones. Participar en diferentes relaciones.
- Decidir si tenemos que usar una propiedad o crear una clase. A veces un atributo es lo suficientemente importante como para considerar que sus diferentes valores corresponden a diferentes objetos.
- Decidir dónde está el nivel para las instancias.
- Pensar qué nivel mínimo de granularidad necesita.
- Limitar el alcance de la ontología.
 - No es necesario que la ontología incluya todas las clases de dominio posibles, solo las necesarias para la aplicación que se está desarrollando.
 - Tampoco necesitamos incluir todos los atributos/restricciones/posibles relaciones.

6. Formal foundations of Ontologies

La **mereología** es una colección de teorías axiomáticas de primer orden que se ocupan de las partes y sus respectivos todos.

Su noción central es **meronomic**, lo que significa que se basa en relaciones parte-todo.

Ha sido la base formal de las ontologías desde Platón hasta principios del siglo XX. Y el creador de la mereología ha sido: el lógico y matemático polaco Stanisław Lesniewski.

Mereología fundamental: la partidar es un orden parcial reflexivo:

- x is_a part_of x (**irreflexivity - nothing is a part of itself**).
- If x is_a part_of y and y is_a part_of x then $x=y$.
- If x is_a part_of y then y is_not_a part_of x (**asymmetry**).
- If x is_a part_of y and y is_a part_of z then x is_a part_of z (**transitivity**).

Hay dos definiciones básicas: **suma mereológica** y **fusión mereológica**. Supongamos que tenemos un conjunto Z de objetos:

- un objeto x es una **suma mereológica** de todos los elementos de Z si y sólo si cada elemento de Z es un ingrediente de x y cada ingrediente de x se superpone a algún elemento z de Z ,
- un objeto x es una **fusión** de todos los elementos de Z si y sólo si para cada objeto y , y se superpone a x si y sólo si y se superpone a algún elemento z de Z .

- x is proper part of y : x is part of y and y is not part of x
- x overlaps y : there is a part of x that is also a part of y
- x and y are disjoint: x and y do not overlap
- *Binary Product*: $x \cdot y$: individual that is part of both x and y and any common part of x and y is a part of the product.
- *Binary Sum*: $x + y$: individual that overlaps something iff it overlaps at least x or y .
- *Difference*: $x - y$: largest individual contained in x that has no part in common with y .

member-of ~~≠~~ instance-of

In other words, **set** ~~≠~~ **class**

Un conjunto que no tiene su descripción intencional no puede ser una clase.

- {car, 3, set, sushi} is not a concept, and hence not a class.
- {human-1, human-2, ... (all humans)} can have two interpretations:
 - a class Human when human- i is interpreted as an instance of Human.
 - an instance of a class "human-set" whose members are human- i .
- {tree-1, tree-2, ..., tree- n } -> a forest (member-of).
- {tree-1, tree-2, ..., tree- i , ... (all trees)} -> a class Tree (instance-of).

La definición ontológica de una clase y una instancia:

- Una cosa que es una conceptualización de un conjunto X puede ser una clase si y sólo si cada elemento x de X pertenece a la clase X si y sólo si la "propiedad intrínseca" de x satisface la condición intencional de X . Y, entonces y solo entonces, $\langle x \text{ instancia-de } X \rangle$ se mantiene.

6.1 Is-a vs part-of

La principal diferencia entre is-a y part-of: al crear una instancia de una clase:

- is-a caso: Solo se genera una instancia de una de las subclases.
- part-of caso: Se generan instancias de cada una de sus subclases.

gato parte de mamífero? perro parte de mamífero?

Si los interpretamos como especies, es correcto.

6.2 Formal foundations of Ontologies

- Necesidad de encontrar una representación computable.
- La lógica de primer orden (FOL) es demasiado expresiva y computacionalmente costosa.
- Descripción Lógica es una opción válida:
 - Permite razonar sobre clases, subclases, instancias y definiciones.
 - ¿Es una restricción de FOL basada en la teoría de conjuntos?
 - También es la base de la semántica formal orientada a objetos.
 - Existen algoritmos eficientes basados en cuadros analíticos.
- FOL + new operators and symbols
 - $\dot{=}$ (if and only if), \sqsubseteq (if)
 - \sqcup union, \sqcap intersection
 - \top (universal set, theorem), \perp (empty set, contradiction)
- Distinction between two kinds of predicates
 - Concepts (C)
 - Relations (R)
- Quantified formulae are rewritten:

$$\begin{aligned}\exists R.C &\equiv \exists y(R(x, y) \wedge C(y)) \\ \forall R.C &\equiv \forall y(R(x, y) \rightarrow C(y))\end{aligned}$$

7. Ontology Languages

Necesidad de expresar ontologías en un lenguaje computable máquina (usable por entidades computacionales en sus mensajes y en su razonamiento):

- Un lenguaje lo suficientemente simple para facilitar el desarrollo de ontologías.
- Un lenguaje con semántica formal: se necesita semántica formal para obtener deducciones de la información en la ontología.
- Un lenguaje que permite a los agentes y otros sistemas inteligentes razonar con él.
- El costo computacional debe ser razonable.

7.1 Advantages of Ontologies

Vista formal de la comunidad:

- Permite formalizar un punto de vista compartido sobre un cierto universo de discurso, por ejemplo, acuerdo sobre cómo modelar el tiempo.

Interoperabilidad:

- Puede apoyar la comunicación y la cooperación entre sistemas desarrollados en diferentes sitios.
- Los compromisos ontológicos realizados por un sistema se hacen explícitos.
- Por ejemplo, los sistemas médicos de diagnóstico y control de la terapia pueden compartir la misma ontología médica genérica subyacente.

ej., noción de estado patológico, procedimiento terapéutico

Adquisición de conocimiento basada en modelos:

- Utilizar la ontología de las directrices médicas para adquirir conocimientos sobre determinadas directrices médicas de forma estructurada.

Validación y verificación del nivel de conocimiento, use la ontología de la guía médica para verificar los documentos de la guía.

7.2 KIF

Idea: tener un formato de intercambio entre aplicaciones, independiente de sus representaciones internas.

Basado en lógica de primer orden (FOL): Notación de prefijos + definiciones.

Semántica: Lógica de descripción (Definiciones + condiciones necesarias)

Operators: Boolean values (True, False), Connectives (and, or, not, if, only if...), Quantifiers (forall, exists...), Vars(?x).

- **Functions can be defined**

```
(deffunction abs (?x) := ((if (>= ?x 0) ?x (- ?x))))
```

- **Relations can also be defined**

```
(defrelation number (?x) :=  
  (or (integer ?x) (real ?x) (complex ?x)))  
≡  
(<=> (number ?x)  
  (or (integer ?x) (real ?x) (complex ?x)))
```

- **Metaknowledge expressions**

```
(believes john '(exists (?x) (> ?x 3)))
```

- **Class person**

```
(defrelation name (?x) := (string ?x))  
  
(defrelation age (?x) := (integer ?x))  
  
(defrelation person (?x ?y) :=  
  (listof (name ?x) (age ?y)))  
  
(defobject juan:= (person "Juan" 25))  
  
(defrelation adult (?x) :=  
  (and (= ?x (person ?x ?y))  
    (> ?y 18)))  
  
(person ((name.X) (age.Y)))
```

7.3 XML

Idea de una Web Semántica: Información anotada semánticamente en un lenguaje analizable por máquina.

HTML no es suficiente (Lenguaje orientado a la presentación).

Por lo que surgió la idea de usar XML (derivado de SGML)

Ventajas:

- Permite describir atributos en la información.
- Ya utilizado por iniciativas industriales.
- Permite la integración desde diferentes fuentes de datos (mediante reglas de traducción XSLT).
- Lenguaje no propietario.

Un documento XML puede contener definiciones de tipos de datos en su interior o puede hacer referencia a un archivo DTD.

Uno puede crear repositorios de definiciones reutilizables (namespaces).

De XML a DAML+OIL, problemas:

- XML es demasiado rígido (estructuras en forma de árbol).
- Difícil incluir relaciones con las estructuras definidas.
- Difícil de afirmar predicados.

Extensión: RDF + RDFS

- RDF permite afirmar sentencias.
- RDFS declara clases, atributos y relaciones.
- Las definiciones de RDFS se pueden instanciar.

Extensión aún más potente: DAML+OIL

- Lenguaje de marcado del agente DARPA.
- Capa de inferencia de ontología.

7.4 OWL

Ampliación adicional: OWL (Ontology Web Language):

- Fusión de DAML+OIL, estándar de W3C.
- También se ha convertido en estándar de facto para la mayoría de las aplicaciones de la Web Semántica y los Servicios de la Web Semántica.
- 3 niveles:
 - OWL lite: define taxonomías y restricciones simples.
 - OWL DL: proporciona expresividad como descripción lógica.
 - OWL full: máxima expresividad (pero no razonadores disponibles).

Hay otra extensión, llamada OWL-S, que se utiliza para los servicios web y la Web semántica.

Características:

Integra todos los elementos en DAML+OIL, creando un nuevo namespace: owl:

- Todas las propiedades que incluye DAML+OIL se han integrado en el owl namespace.
- Sintax se amplía para especificar instancias y se añade una nueva etiqueta, owl:Thing. (todas las instancias pertenecen a owl:Thing)
- Permite la definición de todos los tipos primitivos incluidos en los XML Schema Datatypes utilizando el namespace xsd.

OWL agrega nuevas restricciones a clases, propiedades e individuos:

- owl:allValuesFrom: los valores de propiedad deben pertenecer a cierta clase.
- owl:equivalentClass, owl:equivalentProperty: Equivalencia entre clases y propiedades.
- owl:sameAs, owl:differentFrom: individuos iguales/diferentes.
- owl:SymmetricProperty: una propiedad simétrica a otra.

8. Ontologies and Knowledge Representation

En la IA distribuida, las ontologías son la base del intercambio de conocimientos. Pero, ¿son las ontologías una representación del conocimiento? La representación del conocimiento es una materia multidisciplinar que aplica teorías y técnicas de otros tres campos:

1. La **lógica** proporciona la estructura formal y las reglas de inferencia.
2. La **ontología** define los tipos de cosas que existen en el dominio de la aplicación.
3. La **computación** apoya las aplicaciones que distinguen la representación del conocimiento de la filosofía pura...

El papel de las ontologías en la comunicación entre agentes:

- Utilizado por los agentes cuando intentan darse sentido unos a otros.
- Caso de FIPA: la ontología se indica mediante un parámetro en un mensaje de ACL (junto con otros parámetros: remitente, destinatario, contenido, respuesta con, respuesta por, en respuesta, sobre, idioma, protocolo e ID de conversación).

Se hace referencia a la ontología desde dentro de un mensaje de ACL y determina la semántica de los conceptos utilizados en el lenguaje del contenido.

- Sin embargo, las propias ontologías pueden referirse a cualquier tema.
- Un caso interesante sería ontologizar la comunicación entre agentes.
- Por ejemplo, los protocolos de interacción (capa de conversación) pueden describirse en una ontología externa a los agentes.

Los agentes podrían descargar las descripciones del protocolo de interacción y modificar su comportamiento en consecuencia:

- Ontología de acciones o tareas vs. ontología de hechos.
- Know-how vs know-that

“Sin **lógica**, una representación del conocimiento es vaga, sin criterios para determinar si las declaraciones son redundantes o contradictorias. Sin **ontología**, los términos y símbolos están mal definidos, confusos y confusos. Y sin **modelos computables**, la lógica y la ontología no pueden implementarse en programas de computadora. **La representación del conocimiento es la aplicación de la lógica y la ontología a la tarea de construir modelos computables para algún dominio.**”

Some ontology tools: Protégé, Jena, Pellet, RDF Validator, Ontolingua...