

Aplicación de lógicas descriptivas

Ontologías

Qué son las lógicas descriptivas?

- Las *lógicas descriptivas* (DL) son una familia de formalismos de representación del conocimiento.
- Definen los conceptos relevantes de un dominio y los relacionan para especificar propiedades.
- Se diseñaron como una extensión de marcos(frames) y redes semánticas, los cuales carecían de semántica basada en lógica formal.

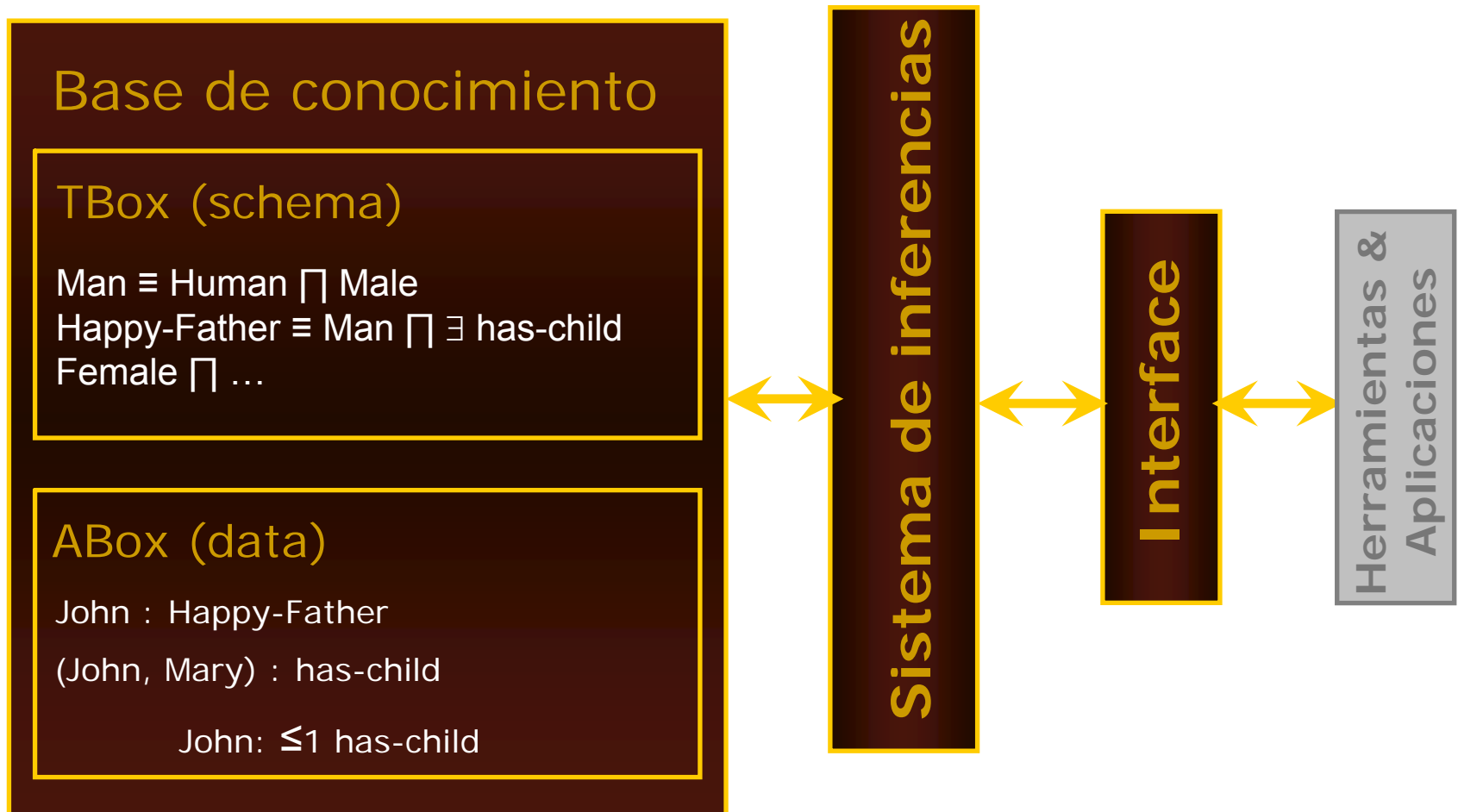
Conceptos básicos

- Los bloques de construcción sintácticos básicos son los conceptos atómicos (predicados unarios), los roles atómicos (predicados binarios) e individuos (constantes).
- El poder expresivo del lenguaje está restringido a un conjunto pequeño de constructores para construir conceptos complejos y roles.
- El conocimiento implícito sobre conceptos e individuos puede inferirse automáticamente con la ayuda de procedimientos de inferencia.

Base de conocimientos en LD

- Una base de conocimiento comprende :
- TBox : contiene sentencias describiendo conceptos jerárquicos (i.e. interrelaciones entre conceptos).
Ej: Cada empleado es una persona.
- ABox : contiene sentencias que indican a donde pertenecen los individuos en la jerarquía (i.e. relaciones entre individuos y conceptos).
Ej: Bob es un empleado.

Arquitectura de un sistema de LD



El lenguaje de descripción básico \mathcal{AL}

- Sintaxis:

$C, D \longrightarrow$	A		-concepto atómico
	\top		-concepto universal
	\perp		-concepto bottom
	$\neg A$		-negación atómica
	$C \sqcap D$		-intersección
	$\forall R.C$		-restricción de valor
	$\exists R.\top$		-cuantificación existencial limitada

- Ejemplos:

$\text{Person} \sqcap \neg \text{Female}$

$\text{Person} \sqcap \exists \text{hasChild}.\top$

$\text{Person} \sqcap \forall \text{hasChild}.\perp$

$\text{Person} \sqcap \forall \text{hasChild}.\text{Female}$

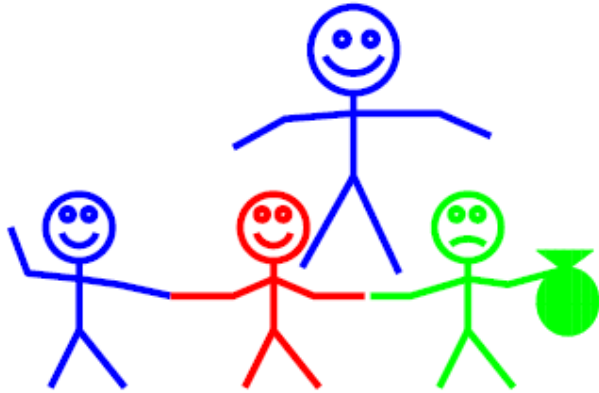
Personas no femeninas

Personas que tienen hijos

Personas que no tienen hijos

Personas que tienen solo hijas

Ejemplo: Padre Feliz



Man \sqcap (\exists has-child.Blue) \sqcap
(\exists has-child.Green) \sqcap
(\forall has-child.Happy \sqcup Rich)

Semántica formal de conceptos \mathcal{AL}

- La semántica es dada por medio de una interpretación \mathcal{I} que consiste de un conjunto no vacío $\Delta^{\mathcal{I}}$ (el dominio de interpretación) y una función de interpretación que le asigna a todo concepto atómico A un conjunto $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ y a cada rol atómico R una relación binaria $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.
- La función interpretación se puede extender a las descripciones de conceptos en forma inductiva.

Semántica formal de conceptos \mathcal{AL}

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$$

$$\perp^{\mathcal{I}} = \emptyset$$

$$(\neg A)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}$$

$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$$

$$(\forall R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \forall b. (a, b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$$

$$(\exists R.\top)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b. (a, b) \in R^{\mathcal{I}}\}.$$

Dos conceptos C y D son equivalentes ($C \equiv D$) si $C^{\mathcal{I}} = D^{\mathcal{I}}$.

Semántica formal (Otros constructores).

- La unión es interpretada como: $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$.
- La cuantificación existencial completa $\exists R.C$ es interpretada como:

$$(\exists R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b. (a, b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}.$$

- Y las restricciones numéricas:

$$(\geq n R)^{\mathcal{I}} = \left\{ a \in \Delta^{\mathcal{I}} \mid |\{b \mid (a, b) \in R^{\mathcal{I}}\}| \geq n \right\},$$

$$(\leq n R)^{\mathcal{I}} = \left\{ a \in \Delta^{\mathcal{I}} \mid |\{b \mid (a, b) \in R^{\mathcal{I}}\}| \leq n \right\},$$

Axiomas terminológicos

- Establecen como los conceptos o roles se relacionan unos con otros:

Sean C y D conceptos y R y S roles:

Axiomas de inclusión:

$$C \sqsubseteq D \quad (R \sqsubseteq S)$$

Axiomas de igualdades:

$$C \equiv D \quad (R \equiv S)$$

Una interpretación \mathcal{I} satisface una inclusión $C \sqsubseteq D$ si $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$

$$C \equiv D \quad C^{\mathcal{I}} = D^{\mathcal{I}}.$$

Semántica para ABox

- Si a, b y c son individuos, empleando conceptos como C y roles como R se pueden añadir a una ABox afirmaciones de dos tipos:
 - $C(a)$ (afirmaciones de conceptos)
 - $R(b, c)$ (afirmaciones de rol)
- La interpretación \mathcal{I} mapea cada nombre individual a un elemento $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.
- Asumimos que nombres de individuos distintos denotan objetos distintos.
- La interpretación \mathcal{I} satisface el concepto de afirmación $C(a)$ si $a^{\mathcal{I}} \in C^{\mathcal{I}}$ y satisface la afirmación de rol $R(b, c)$ si $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$.

Interpretaciones y modelos

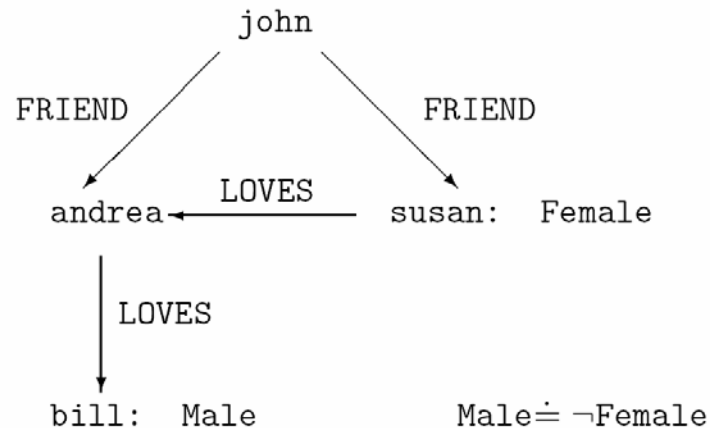
- Si \mathcal{T} es un conjunto de axiomas:
$$\mathcal{I} \text{ satisface } \mathcal{T} \Leftrightarrow \mathcal{I} \text{ satisface cada elemento de } \mathcal{T}.$$
- Si \mathcal{I} satisface un axioma decimos que es un modelo de ese axioma.
- Una interpretación satisface la ABox A si satisface cada afirmación en A . En este caso la interpretación se dice que es un modelo de la afirmación o de la ABox.
- Una ABox A es consistente con respecto a una TBox T , si hay una interpretación que es modelo de A y T

Diferencia entre ABox y BD

- Las semánticas de las ABoxes son caracterizadas como semánticas de “mundo abierto” mientras que las semánticas de BD están caracterizadas como semánticas de “mundo cerrado”.
- La ausencia de información en una instancia de BD es interpretada como información negativa mientras que la ausencia de información en una ABox solo indica carencia de conocimiento.
- Mientras una instancia de BD representa solo una interpretación, una ABox representa muchas diferentes interpretaciones (modelos).

Diferencia entre ABox y BD

- Toda evaluación en una BD no es un razonamiento lógico, sino un chequeo de modelo finito.
- Ejemplo:



$$(\exists \text{FRIEND. (Female} \sqcap (\exists \text{LOVES. Male)}))(\text{john})$$

- ¿Existe alguna amiga de John que ama a alguna persona que es hombre?
- BD: ???.
- ABox: ???

Respuestas al ejemplo

- BD: Existe un único modelo

$$\begin{aligned}\Delta^{\mathcal{I}} &= \{\text{john, susan, andrea, bill}\} \\ \text{Female}^{\mathcal{I}} &= \{\text{susan}\}\end{aligned}$$

- ABox: Existen 4 modelos posibles

$$\begin{array}{ll}\Delta^{\mathcal{I}_1} = \{\text{john, susan, andrea, bill}\} & \Delta^{\mathcal{I}_2} = \{\text{john, susan, andrea, bill}\} \\ \text{Female}^{\mathcal{I}_1} = \{\text{susan, andrea}\} & \text{Female}^{\mathcal{I}_2} = \{\text{susan}\} \\ \text{Male}^{\mathcal{I}_1} = \{\text{bill, john}\} & \text{Male}^{\mathcal{I}_2} = \{\text{bill, andrea, john}\}\end{array}$$

$$\begin{array}{ll}\Delta^{\mathcal{I}_1} = \{\text{john, susan, andrea, bill}\} & \Delta^{\mathcal{I}_2} = \{\text{john, susan, andrea, bill}\} \\ \text{Female}^{\mathcal{I}_1} = \{\text{susan, andrea, john}\} & \text{Female}^{\mathcal{I}_2} = \{\text{susan, john}\} \\ \text{Male}^{\mathcal{I}_1} = \{\text{bill}\} & \text{Male}^{\mathcal{I}_2} = \{\text{bill, andrea}\}\end{array}$$

Ontologías

- Las ontologías son teorías de contenidos sobre las clases de objetos, propiedades de objetos y relaciones entre objetos que son posibles en un dominio de conocimiento especificado, y como tales necesitan de un lenguaje de representación (ej:LD).
- En IA ontología significa 2 cosas relacionadas
 - Un vocabulario de representación especializado para algún dominio.
 - Un cuerpo de conocimiento que describe algún dominio, por lo general un dominio de conocimiento de sentido común.

Diferencia entre ontología y BC

- Una ontología provee la estructura básica o armadura alrededor de la cual una base de conocimiento puede construirse. Una ontología proporciona un conjunto de conceptos y términos para describir algún dominio, mientras que una BC utiliza estos términos para representar lo que es verdadero sobre algún mundo hipotético o real.
- Ejemplo: Una ontología médica podría contener definiciones de términos como leucemia o anemia, pero no debería contener afirmaciones como que cierto paciente particular tiene una enfermedad aunque una BC si podría contener esta afirmación.

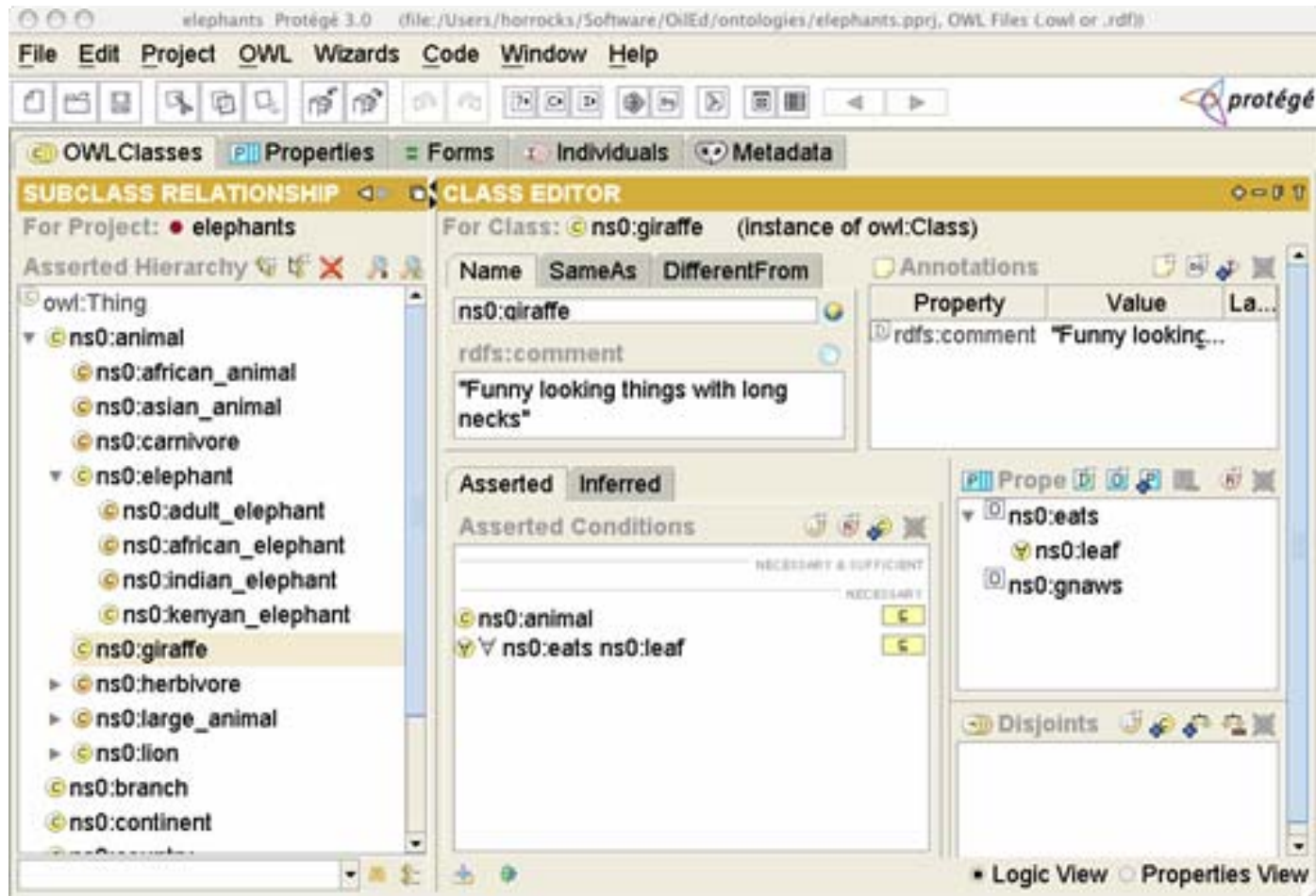
Especificación de ontologías

- La especificación de una ontología en los sistemas de conocimiento tiene 2 dimensiones:
 - **conocimiento factual de dominio**: proporciona conocimiento sobre las realidades objetivo en el dominio de interés (objetos, relaciones, eventos, estados, relaciones causales, etc).
 - **conocimiento para resolver problemas**: proporciona conocimiento sobre como alcanzar varios objetivos.

Una ontología puede representar creencias, intenciones, deseos, objetivos, hipótesis y predicciones sobre un dominio (además de hechos simples).

En general se necesita una ontología de propósito general y una ontología específica del dominio de discurso.

Ejemplo de ontología (Protégé)



Características comunes

- Hay objetos en el mundo
- Los objetos tienen propiedades o atributos que pueden tomar valores.
- Los objetos pueden existir en varias relaciones unos con otros.
- Las propiedades y relaciones pueden cambiar con el tiempo.
- Hay eventos que ocurren en diferentes instantes de tiempo.
- El mundo y sus objetos pueden estar en diferentes estados.
- Los eventos pueden causar otros eventos o estados como efectos.
- Los objetos pueden tener partes.

Criterios de diseño para ontologías

- **Claridad:** Las definiciones deben ser objetivas e independientes del contexto.
- **Coherencia:** Las inferencias deben ser consistentes con las definiciones.
- **Extendibilidad:** Debe anticipar el uso de vocabulario compartido. Se debe poder definir nuevos términos para usos especiales basados en el vocabulario existente, sin tener que revisar las definiciones existentes.
- **Sesgo de codificación:** Se debe evitar que las elecciones de representación sean hechas puramente por conveniencia de notación o implementación.
- **Compromiso ontológico mínimo:** Una ontología debe realizar tan pocas afirmaciones sobre el mundo modelado como sea posibles, permitiendo a las partes comprometidas con la ontología ser libres de especializar e instanciar la ontología cuando lo necesiten.

Interpretación de una ontología

- Una ontología especifica un conjunto de restricciones que declaran lo que debería cumplirse necesariamente en cualquier mundo posible.
- Una descripción de un mundo legal es un mundo posible que satisface las restricciones.
- La interpretación de una ontología se define como la colección de todas las descripciones de mundos legales.

Donde se utilizan las ontologías?

- **e-Science**, e.g., Bioinformáticas
 - Open Biomedical Ontologies Consortium (GO, MGED)
- **Medicina**
 - Construyendo/manteniendo terminologías como Snomed, NCI & Galen.
- **Organizando información compleja y semi-estructurada**
 - UN-FAO, NASA, General Motors, Lockheed Martin, ...
- **Gobierno/Militar**
 - DARPA, NIST, SAIC, Department of Homeland Security, ...
- **La Web Semántica**

Lenguajes de ontologías

- RDF, DAML, OIL, OWL
- RDF (Resource Description Framework) es un marco para metadatos en la WWW. Posee:
 - Clases y propiedades
 - Sub/super-clases (y propiedades)
 - Rango y dominio (de propiedades)
- Pero RDF es demasiado débil para describir los recursos con suficientes detalles, e.g:
 - No tiene restricciones de dominio y rango localizado.
 - No tiene restricciones de cardinalidad/existencia.
 - No tiene propiedades simétricas o inversas ni transitivas.
- Y RDF tiene semánticas no estándares
 - Que dificultan soportar el razonamiento

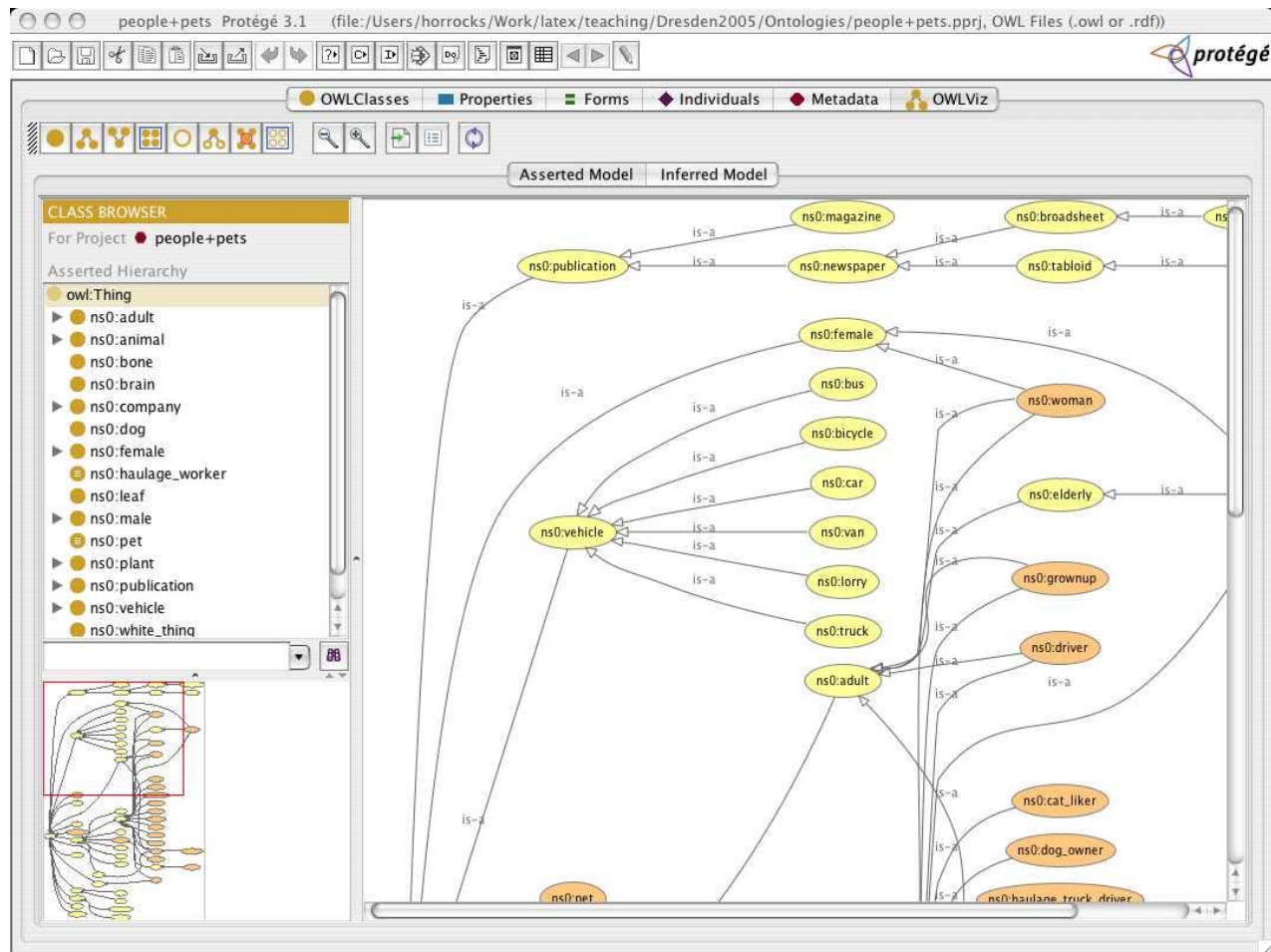
De RDF a OWL

- Dos lenguajes desarrollados para tratar las deficiencias de RDF
 - **OIL**: desarrollado por un grupo de investigadores (mayormente) europeos (varios del proyecto EU OntoKnowledge)
 - **DAML-ONT**: desarrollado por un grupo de investigadores (mayormente) de EU (en el programa DARPA **DAML**)
- Los esfuerzos se juntaron para producir **DAML+OIL**
 - El desarrollo fue realizado por “Joint EU/US Committee on Agent Markup Languages”
 - Extiende (“un subconjunto de LD”) RDF
- DAML+OIL subcribe a W3C como base para estandarización
 - Se formó el grupo de trabajo Web-Ontology (**WebOnt**)
 - El grupo WebOnt desarrollo el lenguaje **OWL** basado en DAML+OIL
 - El lenguaje OWL es ahora una Recomendación W3C (i.e., un estándar como HTML y XML)

Lenguaje OWL

- Existen 3 especies de OWL
 - **OWL full** es la union de la sintaxis OWL y RDF
 - La semántica RDF extendida con condiciones semánticas relevantes y tripletas axiomáticas.
 - **OWL DL** restringidos a fragmentos de DL/FOL ($\frac{1}{4}$ DAML+OIL)
 - Tiene un modelo de semántico estándar (primer orden)
 - Equivalente to **SHOIN**(D_n)
 - **OWL Lite** es “más fácil de implementar” subconjunto de OWL DL
 - Equivalente a **SHIF**(D_n)
- OWL DL/Lite el más usado
 - Amplio rango de herramientas/implementaciones disponibles.

Inferencias (Protegé)



Ontología (TBox/ABox)

OWL Syntax	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human \sqsubseteq Animal \sqcap Biped
equivalentClass	$C_1 \equiv C_2$	Man \equiv Human \sqcap Male
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter \sqsubseteq hasChild
equivalentProperty	$P_1 \equiv P_2$	cost \equiv price
transitiveProperty	$P^+ \sqsubseteq P$	ancestor ⁺ \sqsubseteq ancestor

OWL Syntax	DL Syntax	Example
type	$a : C$	John : Happy-Father
property	$\langle a, b \rangle : R$	$\langle \text{John}, \text{Mary} \rangle : \text{has-child}$

Aumentar el poder expresivo

- OWL no es lo suficientemente expresivo para algunas aplicaciones
 - Los constructores son principalmente para clases (predicados unarios)
 - No hay tipos de datos complejos o predicados integrados (e.g., aritméticos)
 - No hay variables
 - No hay predicados de mayor aridad
- Las extensiones (de OWL) que se están considerando incluyen:
 - Extensiones (decidibles) para la LD subyacente
 - Extensiones del lenguaje de “Reglas”
 - Lógica de primer orden (e.g., SWRL-FOL)
 - Extensiones (sintácticamente) de mayor orden (e.g., Lógica común)

Bibliografía

- D. Nardi, R. J. Brachman. An Introduction to Description Logics. In the Description Logic Handbook, edited by F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider, Cambridge University Press, 2002,
- F. Baader, W. Nutt. Basic Description Logics. In the Description Logic Handbook, edited by F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider, Cambridge University Press, 2002
- A. Borgida, R. J. Brachman. Conceptual Modelling with Description Logics. In the Description Logic Handbook, edited by F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider, Cambridge University Press, 2002
- B. Chandrasekaran, J. Josephson, V. Benjamins, What Are Ontologies, and Why Do We Need Them? IEEE Intelligent Systems, 1999.
- T. Gruber, Toward principles for the Design of Ontologies Used for Knowledge Sharing. In Formal Ontology in Conceptual Analysis and Knowledge Representation, 1993.
- H. Levesque, R. Brachman, Expressiveness and Tractability in knowledge representation and reasoning, 1987.
- Algunos links:
 - <http://www.w3.org/TR/owl-features/>
 - <http://www.ontopia.net/topicmaps/mat08:55> p.m. 14/10/2008erials/tmrdfoildaml.html