

# Materialización en Acceso a Datos Basado en Ontologías

**Sergio Alejandro Gómez (en colaboración con Pablo Rubén Fillotrani)**

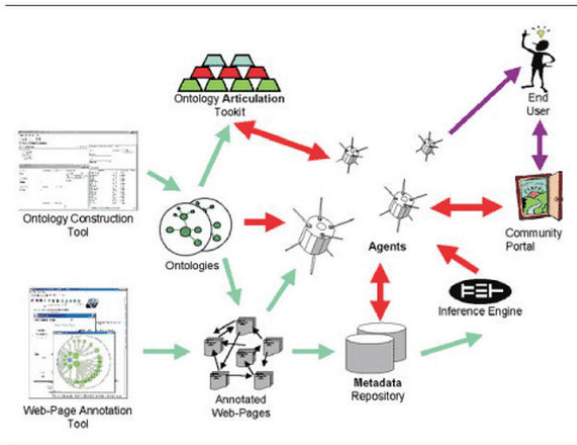
Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur



- Motivaciones
- Acceso a datos basado en ontologías
- Representación de ontologías: Description Logics
- Materialización de ontologías a partir de bases de datos relacionales
- Resultados obtenidos
- Sumario

# Motivaciones

## Web Semántica

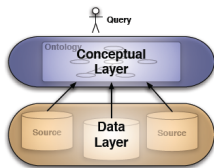


Berners-Lee, T., Hendler, J. & Lassila, O.  
The Semantic Web,  
*Scientific American* 284(5), 34–43, 2001.

# Motivaciones

## Acceso a datos basado en ontologías

Permite una visión de alto nivel de los datos, enriqueciendo el sistema con conocimiento de dominio y soportando consultas con múltiples fuentes de datos heterogéneas (dos enfoques: virtualización y materialización)



- **Capa conceptual:** El esquema de los datos se realiza en alto nivel con una ontología
- **Capa de datos:** Los datos están modelados con una instancia relacional (i.e. una BD o varias BD relacionales)
- **Mapeos:** Los mapeos indican cómo interpretar los datos de la BDR como aserciones en la ontología; e.g.

```
Man(DNI) ← select DNI from PERSONA where SEXO = 'Masculino'  
name(DNI, NOMBRE) ← select DNI, NOMBRE from PERSONA
```

Para modelar el esquema y la instancia de la base de datos se utiliza un formalismo de representación de conocimiento que permite razonar en forma algebraica sobre los diagramas relacionales.

La terminología representa el esquema con fórmulas:

- $Person \sqsubseteq \exists name$ : Las personas tienen un atributo nombre
- $\exists name^- \sqsubseteq String$ : El nombre es de tipo string
- $Person \sqsubseteq \exists drives$ : Las personas manejan
- $\exists drives^- \sqsubseteq Car$ : Las cosas manejadas son autos

La instancia se representa con aserciones:

- $Person(a)$ :  $a$  es una persona
- $name(a, 'Juan')$ : el nombre de  $a$  es Juan (se deduce que es un string)
- $drives(a, b)$ :  $a$  maneja a  $b$  (se deduce que  $b$  es un auto)

# Mapeo de una tabla con una clave primaria

## Formalización

- Sea  $T$  una tabla con esquema  $T(\underline{k}, a_1, \dots, a_n)$  e instancia  $\{(k^1, a_1^1, \dots, a_n^1), \dots, (k^m, a_1^m, \dots, a_n^m)\}$ .
- Para representar el esquema de la tabla, se debe crear un concepto  $T$  y para cada atributo  $a_i$  con dominio  $D_i$ , se deben crear dos axiomas:  $T \sqsubseteq \exists a_i$ , indicando que cada  $T$  tiene un atributo  $a_i$ , y  $\exists a_i^- \sqsubseteq D_i$ , diciendo que el tipo de  $a_i$  es  $D_i$ .
- La caja asercional para para representar la instancia de  $T$  contendrá:
  - Para cada individuo  $k_i$ , se usa

$$T(k^i)$$

para  $i = 1, \dots, n$   $T(k^1), \dots, T(k^m)$ .

- Para cada atributo  $a_i$  del esquema y valor de instancia  $a_i^j$ , se usa una aserción de rol

$$a_i(k^j, a_i^j)$$

para  $i = 1, \dots, n$  y  $j = 1, \dots, m$ .

# Mapeo de una tabla con una clave primaria

## Ejemplo

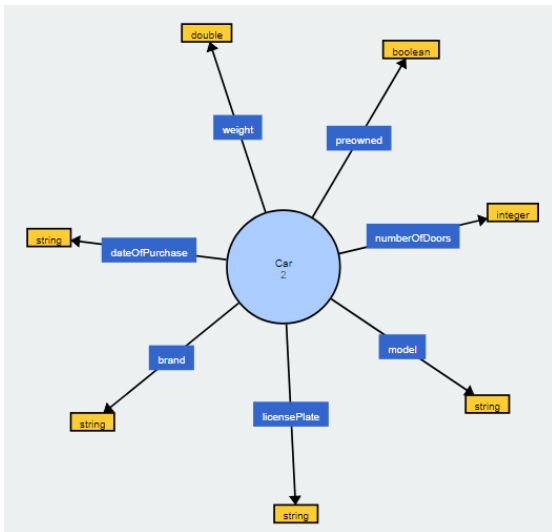
*Car*(*licensePlate*, *brand*, *model*, *weight*, *preowned*, *dateOfPurchase*, *numberOfDoors*)

<i>licensePlate</i>	<i>brand</i>	<i>model</i>	<i>weight</i>	<i>preowned</i>	<i>dateOfPurchase</i>	<i>numberOfDoors</i>
ABC123	Chevrolet	Corsa	1000.1	FALSE	2010-10-01	4
CDE456	VW	Suran	1000.4	TRUE	2013-10-01	5

- Filtro SQL:  
`select * from Car where dateOfPurchase <= 2011-12-31`  
(Sólo la primera fila pertenece al resultado de la consulta)
- Nombre de la clase: *Car*
- Los autos tienen una marca de tipo string:  $Car \sqsubseteq \exists brand$  y  $\exists brand^- \sqsubseteq String$ .
- ABC123 es un auto:  $Car(ABC123)$
- La marca de ABC123 es Chevrolet:  $brand(ABC123, Chevrolet)$
- Los autos tienen un número de puertas entero:  $Car \sqsubseteq \exists numberOfDoors$  y  $\exists numberOfDoors^- \sqsubseteq Integer$ .
- ABC123 tiene 4 puertas:  $numberOfDoors(ABC123, 4)$

# Mapeo de una tabla con una clave primaria

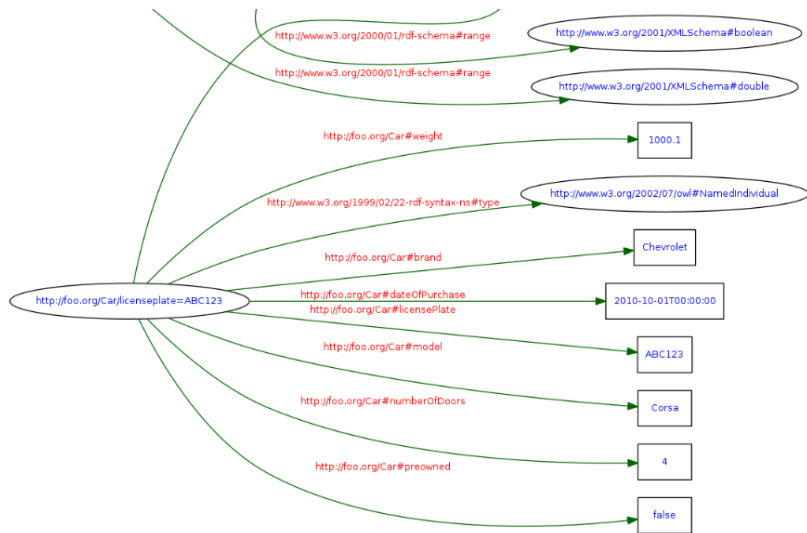
Visualización del nodo Car del grafo de la ontología





# Mapeo de una tabla con una clave primaria

Visualización del auto ABC123 del grafo de la caja asercional



# Mapeo de una tabla con una clave primaria

## Código OWL

```
<owl:Class rdf:about="http://foo.org#Car"/>

<owl:DatatypeProperty rdf:about="http://foo.org/Car#brand">
  <rdfs:domain rdf:resource="http://foo.org#Car"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="http://foo.org/Car#numberOfDoors">
  <rdfs:domain rdf:resource="http://foo.org#Car"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
</owl:DatatypeProperty>
...
<owl:NamedIndividual rdf:about="http://foo.org/Car/licenseplate=ABC123">
  <rdf:type rdf:resource="http://foo.org#Car"/>
  <Car:brand rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Chevrolet</Car:brand>
  ...
  <Car:numberOfDoors rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">4
</Car:numberOfDoors>
  ...
</owl:NamedIndividual>
```

En



Gómez, S.A., Fillottrani, P.R.

Towards a Framework for Ontology-Based Data Access: Materialization of OWL Ontologies From Relational Databases,

CACIC 2018, Tandil, Argentina, 2018.

se formalizan y ejemplifican cómo interpretar:

- Mapeo de tablas con una clave primaria
- Mapeo de relaciones uno a muchos
- Mapeo de relaciones muchos a muchos
- Mapeo de tablas sin claves primarias

En



Gómez, S.A., et al.

Desarrollo de Herramientas para Acceso a Bases de Datos Heterogéneas Basado en Ontologías en el Contexto de la Entrega de Servicios Públicos Digitales,

Primer Encuentro de Centros Propios y Asociados, Mar del Plata, Argentina, 2018.

se muestra cómo aplicar las tecnologías para exportar bases de datos de la Municipalidad de Bahía Blanca respecto de información de distribución de planes sociales.

- Estamos desarrollando una aplicación prototípica para la materialización de ontologías OWL a partir de bases de datos relacionales (<http://cs.uns.edu.ar/~sag/obda-test>).
- El prototipo ha sido codificado en Java y accede una base de datos H2 para exportar el esquema y la instancia de sus tablas en varios formatos OWL al usar la OWL API.
- Nuestra implementación sigue las directivas de la Direct Mapping Specification (DMS) para generar conocimiento asercional y las directivas de Kontchakov et al. para generar conocimiento terminológico.
- Este trabajo extiende las tecnologías del estado del arte al proveer definiciones generales de cómo las ontologías deben construirse ya que la DMS sólo provee ejemplos sin formalización.
- Esta investigación puede ser útil para desarrolladores de sistemas de información modernos donde fuentes de datos múltiples que necesitan accederse en una forma unificada con un lenguaje de consultas simple (como SPARQL) o usando un motor de inferencia Description Logics (como Hermit).

- Presentamos un marco de definiciones para permitir la interpretación de bases de datos en ontologías Description Logics
- Presentamos una aplicación que permite materializar una ontología OWL XML y OWL N3 a partir de una base de datos relacional en formato H2.
- La aplicación desarrollada programada en Java, utiliza JDBC como soporte de bases de datos y OWL API como soporte de OWL.
- El tema brinda posibilidades del desarrollos de trabajos de grado y posgrado en un área que involucra la intersección de lógica, tecnologías de bases de datos, ontologías, datos enlazados, web semántica, representación de conocimiento y razonamiento automatizado.

¡Gracias por escuchar! ¿Preguntas?