

Agentes Inteligentes Para la Web

Trabajo Práctico Obligatorio N°1 - Tecnologías Semánticas

Manuel Latorre FAI-1931
manuel.latorre@est.fi.uncoma.edu.ar

Primer cuatrimestre 2023



Índice

1. Ejercicio 1	1
1.1. Validación con Protégé	4
1.2. Definición de relación en OWL y RDF	4
2. Ejercicio 2	6
3. Ejercicio 3	9
3.1. Web 2.0 vs Web Semantica	9

Índice de figuras

1. Modelo entidad relación del dominio planteado junto con la inconsistencia	1
2. Detección de inconsistencia en Bípedo	4
3. Explicación del razonador HermiT a la inconsistencia encontrada en Bípedo	5
4. Grafo de “Awards” de Borges obtenido con la ultima consulta SPARQL	8

1. Ejercicio 1

El dominio consistirá en la definición de diferentes tipos de animales junto con sus alimentos y habitats. De los animales se conocerá su nombre, tamaño promedio y peso promedio. De los alimentos se conoce su nombre y descripción. Del habitat en donde viven los animales se sabe el nombre, ubicación y clima. Los animales siempre serán o terrestres o acuáticos, sin poder ser ambos a la vez. Además de los animales terrestres se pueden clasificar como cuadrúpedos o bípedos, mientras que los acuáticos se clasifican en cetáceos y peces. Finalmente de los animales acuáticos se conoce el tipo de agua en la que habitan, es decir dulce o salada.

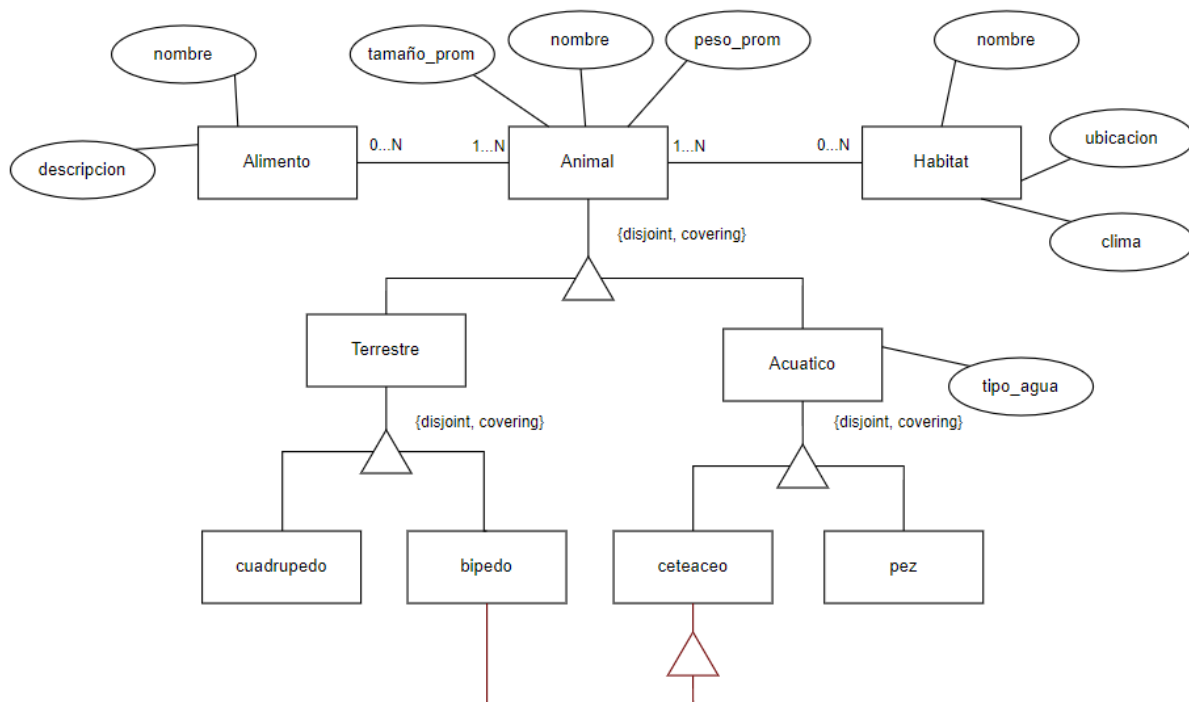


Figura 1: Modelo entidad relación del dominio planteado junto con la inconsistencia

En el MER planteado se puede observar en rojo una generalización especialización entre “Cetaceo” y “Bipedo”, esta constituye una inconsistencia en el modelo debido a que las generalizaciones/especializaciones son totales y disjuntas por lo que si un “Animal” es “Terrestre” nunca podrá ser “Acuático” y al no poder ser “Acuático” entonces un “Bipedo” nunca debería poder ser un “Cetaceo”. Esta relación se representa a través de la lógica de la siguiente manera:

Sea x un animal:

$$\begin{aligned}Cuadrupedo(x) \vee Bipedo(x) &\longrightarrow Terrestre(x) \\Terrestre(x) &\longrightarrow \neg Acuatico(x) \\Cuadrupedo(x) &\longrightarrow \neg Bipedo(x) \\Bipedo(x) &\longrightarrow \neg Cuadrupedo(x) \\Cetaceo(x) \vee Pez(x) &\longrightarrow Acuatico(x) \\Acuatico(x) &\longrightarrow \neg Terrestre(x) \\Cetaceo(x) &\longrightarrow \neg Pez(x) \\Pez(x) &\longrightarrow \neg Cetaceo(x)\end{aligned}$$

Como se tiene que los “Bipedos” son animales “Terrestres” entonces se tiene que estos no pueden ser animales “Acuaticos” y por extension no pueden ser un “cetaceos”, ya que si lo fueran también serian “Acuaticos”. Por lo tanto queda claro que la especializacion/generalizacion entre “Cetaceo” y “Bipedo” no es posible dada la disyunción existente entre “Terrestres” y “Acuaticos”. Por lo tanto, lo representado en rojo de la figura 1 es efectivamente una inconsistencia en el modelo

Definicion con Lógicas Descriptivas**Habitat:** $\exists clima \sqsubseteq Habitat$ $\exists nombre \sqsubseteq Habitat$ $\exists ubicacion \sqsubseteq Habitat$ **Alimento:** $\exists nombre \sqsubseteq Alimento$ $\exists descripcion \sqsubseteq Alimento$ **Animal:** $\exists nombre \sqsubseteq Animal$ $\exists peso_prom \sqsubseteq Animal$ $\exists tamano_prom \sqsubseteq Animal$ **Terrestre:** $Terrestre \sqsubseteq Animal \sqcap \neg Acuatico$ **Acuatico:** $Acuatico \sqsubseteq Animal \sqcap \neg Terrestre$ $\exists tipo_agua \sqsubseteq Acuatico$ **Cuadrupedo:** $Cuadrupedo \sqsubseteq Terrestre \sqcap \neg Bipedo$ **Bipedo:** $Bipedo \sqsubseteq Terrestre \sqcap \neg Cuadrupedo$ **Cetaceo:** $Cetaceo \sqsubseteq Acuatico \sqcap \neg Pez$ **Pez:** $Pez \sqsubseteq Acuatico \sqcap \neg Cetaceo$ **Relacion Animal \leftrightarrow Habitat:** $Animal \sqsubseteq \geq 0 habita.Habitat$ $Habitat \sqsubseteq \exists esHabitado.Animal$ **Relacion Animal \leftrightarrow Alimento:** $Animal \sqsubseteq \geq 0 come.Alimento$ $Alimento \sqsubseteq \exists esComido.Animal$

1.1. Validación con Protégé

Al modelar la ontología en Protégé y validarla con el razonador HermiT, se puede detectar la presencia de inconsistencias. En la Figura 1, se muestra un ejemplo en el cual el razonador HermiT identifica una inconsistencia y proporciona una explicación para la misma.

La explicación del razonador indica que la clase “Bipedeo” es una subclase de “Cetaceo”, y a su vez, “Cetaceo” es una subclase de “Acuatico”. Sin embargo, también se establece que “Bipedeo” es una subclase de “Terrestre” y que “Acuatico” y “Terrestre” son clases disjuntas, lo que implica que no pueden tener instancias en común.

Esta explicación muestra que la existencia de “Bipedeo” como subclase tanto de “Cetaceo” (que está relacionado con “Acuatico”) como de “Terrestre” es contradictoria debido a la disyunción planteada entre “Acuatico” y “Terrestre”. Por lo tanto, se demuestra la inconsistencia en el modelo ontológico y coincide con lo planteado a través de la lógica previamente.

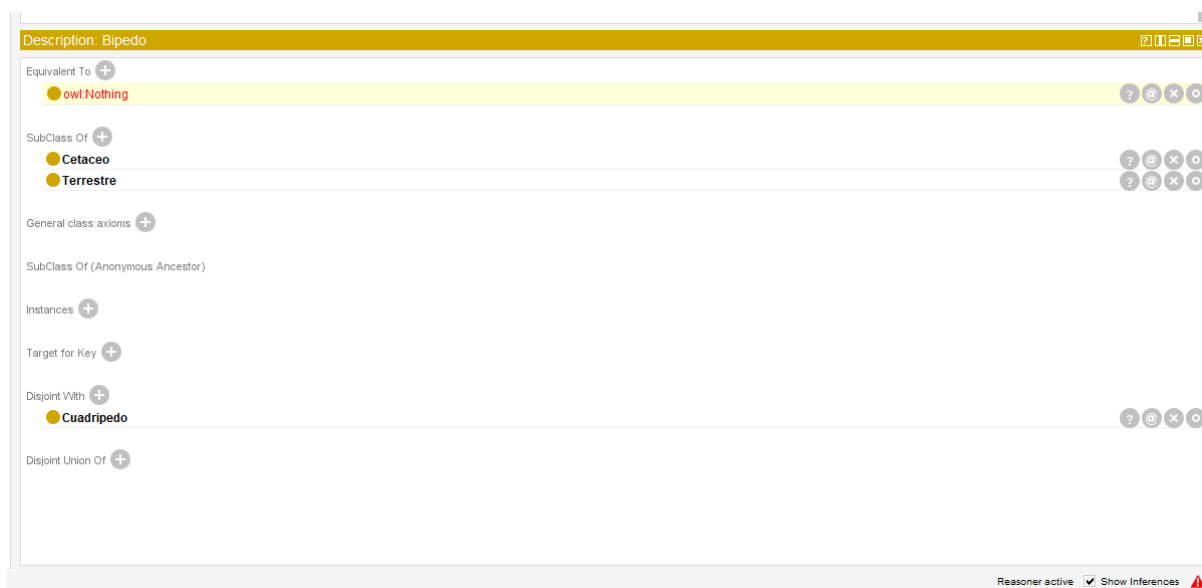


Figura 2: Detección de inconsistencia en Bípedeo

1.2. Definición de relación en OWL y RDF

Además de las diferencias sintácticas que se pueden apreciar entre las definiciones de Owl y RDF de los cuadros de mas abajo, aunque en el ejemplo planteado no se puede observar claramente, se debe destacar que OWL se basa en lógica descriptiva y permite expresar relaciones complejas y razonar sobre ellas. Además proporciona constructores específicos para definir jerarquías de clases, propiedades y restricciones que facilitan el razonamiento lógico.

En cambio RDF es un modelo de datos básico que se utiliza para la representación de la información en la web semántica y no proporciona la capacidad de definir restricciones complejas o realizar razonamiento lógico directamente. RDF posee una expresividad mas limitada que OWL y se centra principalmente en la representación de tripletas sujetos-predicados-objetos sin proporcionar construcciones específicas para definir jerarquías o restricciones complejas

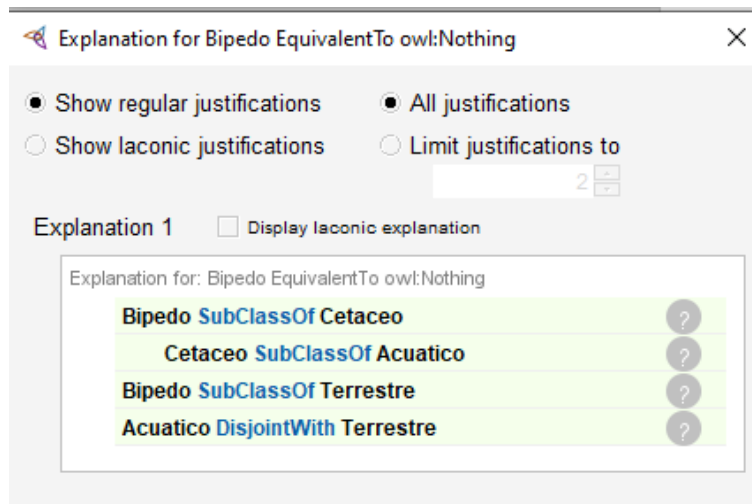


Figura 3: Explicación del razonador HermiT a la inconsistencia encontrada en Bipedo

Definición en RDF de “Acuatico” subclase de “Animal” y disjunto con “Terrestre”

```
<owl:Class rdf:about="http://tpo1#Animal">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf
    -schema#Class" />
</owl:Class>

<owl:Class rdf:about="http://tpo1#Acuatico">
  <rdfs:subClassOf rdf:resource="http://tpo1#Animal" />
  <owl:disjointWith rdf:resource="http://tpo1#Terrestre" />
</owl:Class>

<owl:Class rdf:about="http://tpo1#Terrestre">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf
    -schema#Class" />
</owl:Class>
```

Definición en RDF de “Acuatico” subclase de “Animal” y disjunto con “Terrestre”

```
<rdf:Description rdf:about="http://tpo1#Animal">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
    Class"/>
</rdf:Description>

<rdf:Description rdf:about="http://tpo1#Acuatico">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
    Class"/>
  <rdfs:subClassOf rdf:resource="http://tpo1#Animal"/>
  <rdfs:disjointWith rdf:resource="http://tpo1#Terrestre"/>
</rdf:Description>

<rdf:Description rdf:about="http://tpo1#Terrestre">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#
    Class"/>
</rdf:Description>
```

2. Ejercicio 2

Listar solo 10 propiedades del recurso Jorge Luis Borges <http://www.wikidata.org/entity/Q909>

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX wd: <http://www.wikidata.org/entity/>

SELECT ?property ?value
WHERE {
  wd:Q909 ?property ?value.
  ?property rdf:type owl:ObjectProperty.
}
LIMIT 10
```


Listar lugar de nacimiento y muerte. La consulta debe retornar también los rdfs:label filtrados por lenguaje inglés (usar FILTER)

```
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX wd: <http://www.wikidata.org/entity/>

SELECT *
WHERE {
    wd:Q909 wdt:P19 ?bornPlace.  # Obtiene la propiedad "place
    of birth" de Q909
    wd:Q909 wdt:P20 ?deathPlace.  # Obtiene la propiedad "place
    of death" de Q909
    ?bornPlace rdfs:label ?bornPlaceLabel.  # Obtiene los labels
    de los valores
    ?deathPlace rdfs:label ?deathPlaceLabel.  # Obtiene el label
    del lugar de muerte
    FILTER(LANG(?bornPlaceLabel) = "en").
    FILTER(LANG(?deathPlaceLabel) = "en").
}
```

Usando CONSTRUCT crear un nuevo grafo con todos los premios recibidos por Borges (award received). Recuperando también los rdfs:label de esos premios y filtrarlos por idioma inglés (en).

```

PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX wd: <http://www.wikidata.org/entity/>

CONSTRUCT {
  wd:Q909 wdt:P166 ?award.
  ?award rdfs:label ?awardLabel.
}
WHERE {
  wd:Q909 wdt:P166 ?award. # Obtiene awards de q909
  ?award rdfs:label ?awardLabel. # Obtiene los labels de
    los awards
  FILTER(LANG(?awardLabel) = "en").
}

```

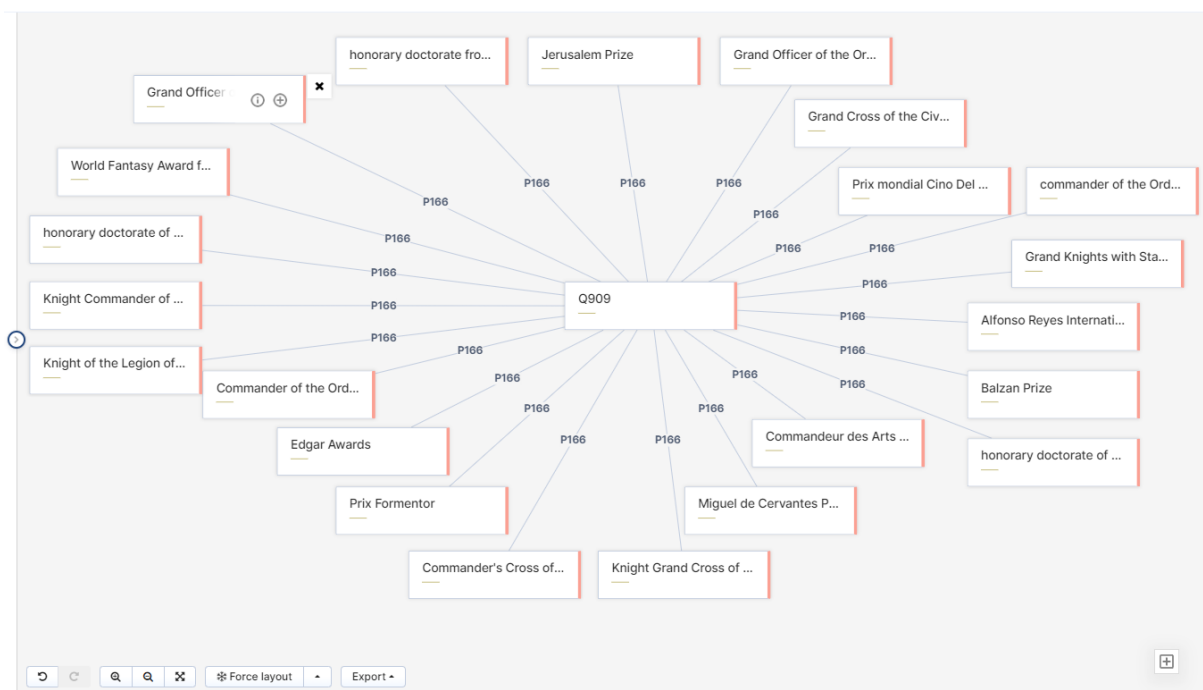


Figura 4: Grafo de “Awards” de Borges obtenido con la ultima consulta SPARQL

3. Ejercicio 3

Un agente debería tener las siguientes cualidades para navegar la Web Semántica

- El agente debe ser capaz de comprender cada declaración que recopila. Una forma de lograrlo es entendiendo los términos comunes y las relaciones que se utilizan para crear estas declaraciones, lo cual implica comprender los datos en formato RDF ya que proporcionan un modelo estandar para representar información semántica en la web, utilizando tripletas sujetos-predicados-objetos para expresar relaciones entre entidades. El agente debe ser capaz de interpretar dichas tripletas y completar la estructura y el significado de los datos RDF
- El agente debe ser capaz de realizar razonamientos basados en su comprensión de los términos y las relaciones comunes. Por ejemplo, al saber que los recursos A y B tienen la misma dirección de correo electrónico y considerar el conocimiento expresado por los términos y las relaciones comunes, debería ser capaz de concluir que A y B son en realidad el mismo recurso. Para esto se utilizan lógicas descriptivas ya que le dan la capacidad al agente de inferir nuevas conclusiones a partir de información disponible en las ontologías y los datos RDF. Estas lógicas proporcionan un marco formal para realizar inferencias y deducciones, lo que permite al agente obtener un mayor nivel de comprensión y conocimiento de los datos semánticos
- El agente debe ser capaz de procesar algunas consultas comunes que se realicen sobre las declaraciones que ha recopilado. Ya que, si no se proporciona una interfaz de consulta, las declaraciones recopiladas no serán de utilidad alguna. Aquí entra en juego SPARQL permitiéndole al agente ejecutar consultas sobre los datos RDF para obtener información específica, brindándole la capacidad de buscar y filtrar datos según ciertos criterios.
- El agente debería ser capaz de utilizar ontologías para comprender la semántica de los datos y realizar razonamiento lógico sobre ellos. Las ontologías son modelos conceptuales que describen las relaciones y propiedades entre los diferentes elementos de un dominio de conocimiento, proporcionando una estructura formal para representar dicho conocimiento y permitiendo al agente inferir nuevas relaciones y conclusiones a partir de la información disponible.

3.1. Web 2.0 vs Web Semantica

Comparado con un agente para la Web 2.0, un agente para la Web Semántica tiene capacidades más avanzadas en términos de interpretación y comprensión del contenido. Mientras que un agente para la Web 2.0 generalmente se centra en la extracción de información basada en palabras clave y la búsqueda de contenido relevante, un agente para la Web Semántica puede entender la semántica subyacente de los datos y realizar inferencias lógicas sobre ellos. Esto permite al agente obtener un mayor nivel de conocimiento y comprensión, y proporciona la capacidad de realizar consultas más sofisticadas y contextualmente más relevantes. Además, un agente para la Web Semántica puede aprovechar la estructura y el significado de las ontologías para navegar y descubrir información de manera más precisa y eficiente, mientras que un agente para la Web 2.0 se basa principalmente en algoritmos de búsqueda de texto plano.

Un ejemplo claro donde se pueden ver las ventajas de la Web semántica sobre la Web 2.0 es en las implementaciones de “mashups”. Un “mashup” es una aplicación web que recopila datos estructurados producidos por terceros a través de APIs ofrecidas por dichos terceros, y procesa los datos de alguna manera para luego representarlos a los usuarios de una forma que difiere de su apariencia original. Normalmente, una aplicación de mashup mejora la presentación visual de los datos o ofrece un valor añadido a sus usuarios al combinar datos de diferentes fuentes, o ambas cosas. Este concepto está más relacionado con la Web 2.0, donde cada vez más sitios web esconden sus datos a través de sus APIs.

Un “mashup” implementado en Web 2.0 presenta las siguientes desventajas y complicaciones:

- Limitada escalabilidad: La construcción de un “mashup” basado en APIs requiere aprender cada conjunto de APIs de diferentes proveedores, lo que implica un proceso constante de aprendizaje. Además, cada vez que haya un nuevo proveedor, será necesario aprender un nuevo conjunto de APIs. Por lo tanto, la construcción y mantenimiento de este tipo de “mashup” no es escalable y puede resultar costoso.
- Cobertura limitada: Un “shopbot” de una tienda digital, por ejemplo, solo puede comprender las APIs que se han programado para que entienda. No puede explorar por sí mismo, lo que limita la cobertura de datos. Cualquier decisión basada en este “shopbot” probablemente no sea óptima debido a la limitada cobertura de datos.
- Desconexión con los proveedores de datos: Una vez que el “shopbot” recupera y consume los datos, se pierde el enlace entre este y el proveedor de datos original. Los usuarios no pueden regresar al sitio original que proporcionó los datos. Incluso si se incluyen enlaces que direccionan de vuelta a los proveedores de datos, estos enlaces suelen ser superficiales y no pueden llevar a ubicaciones precisas de los componentes de datos específicos. Esto limita la capacidad del usuario para aprovechar ofertas especiales u otras características ofrecidas en el sitio original.

En cambio con Web semántica a través del uso de datos enlazados (*Linked Data*) se obtendrán los siguientes resultados:

- Buena escalabilidad del método en sí mismo: Bajo la Web de Datos Enlazados, los datos estructurados se expresan utilizando grafos y estándares RDF, que son un conjunto de estándares comunes a todos los sitios. Esto hace que la construcción y el mantenimiento de un mashup sean escalables, ya que no es necesario aprender constantemente nuevas APIs.
- Cobertura ilimitada de conjuntos de datos: A diferencia de los mashups de Web 2.0, las aplicaciones de Linked Data (Web semántica) operan sobre un espacio de datos global y sin límites. Esto les permite proporcionar respuestas más completas a medida que aparecen nuevas fuentes de datos en la Web.
- Enlaces cruciales para volver a los proveedores de datos: En los mashups de Linked Data, todos los elementos (recursos) se identifican mediante URIs controladas por el proveedor de datos. Si un usuario busca una de estas URIs, puede ser direccionado de vuelta al proveedor de datos original, quien puede proporcionar contenido adicional para dirigir el tráfico entrante. Esta capacidad de enlace es una diferencia clave entre los mashups de Web 2.0 y los mashups de Linked Data, y es donde reside el potencial valor comercial.

- Los mashups de Linked Data también ofrecen a los usuarios la posibilidad de enlazar casi ilimitadamente con otros recursos. Cada elemento en el mashup se identifica mediante una URI, que puede estar vinculada a otros recursos en otros conjuntos de datos. Estos enlaces también están tipificados, lo que permite al usuario seguirlos y visitar recursos específicos en otros conjuntos de datos. Esta capacidad de enlace ilimitado ofrece un valor significativo.