

Modelos OWL

Boletín de prácticas

Práctica 4: Creación de un modelo OWL (Protégé) 1 sesión

Curso 2014-2015

Autores:

J. Javier Samper Zapater
José Fco. García Calderaro

Creación de un modelo OWL

1. Objetivos

Presentar al alumno algunas de las herramientas más utilizadas en el proceso de construcción de un modelo OWL. Uso de Protégé (editor de ontologías con numerosos plugins)

2. Introducción

Se desea crear un modelo semántico de información sobre la asignatura que está cursando el alumno, principalmente sobre los contenidos de la Guía Docente. Para ello se van a hacer uso de algunas de las herramientas que nos proporciona la Web de 3ª generación.

Los pasos a seguir son:

- Definición de un modelo semántico mediante el lenguaje OWL que permita especificar el dominio de la asignatura, recogiendo todos aquellos conceptos, propiedades y relaciones que se consideren necesarios, para cumplir los objetivos. Se tendrá en cuenta por ejemplo, los diferentes elementos que estructuran la guía de la asignatura.
- Instanciación de todos los temas (en protégé: creación de **individuales**), que permitan evaluar el modelo definido.
- Comprobación en cada paso de la consistencia del modelo mediante el razonador.

• 2.1 Datos a tener en cuenta

- Para que sea totalmente compatible con el software de la siguiente sesión, la ontología se guardará como fichero “.owl” **PERO utilizando la opción RDF/XML.**
- Conviene que la Ontología tenga **como raíz una única clase genérica** (subclase de THING).

3. Protégé

Se recomienda dar un vistazo a la información que se encuentra disponible en http://protegewiki.stanford.edu/wiki/Main_Page. En ella podemos encontrar desde las preguntas más frecuentes a como dar nuestros primeros pasos con Protégé.

OWL es una extensión de RDF y el enfoque del editor Protege-OWL está en la edición de ontologías OWL, aunque también se puede utilizar para editar modelos RDF y archivos RDF Schema o bases de datos. Nosotros vamos a desarrollar el modelo directamente con OWL.

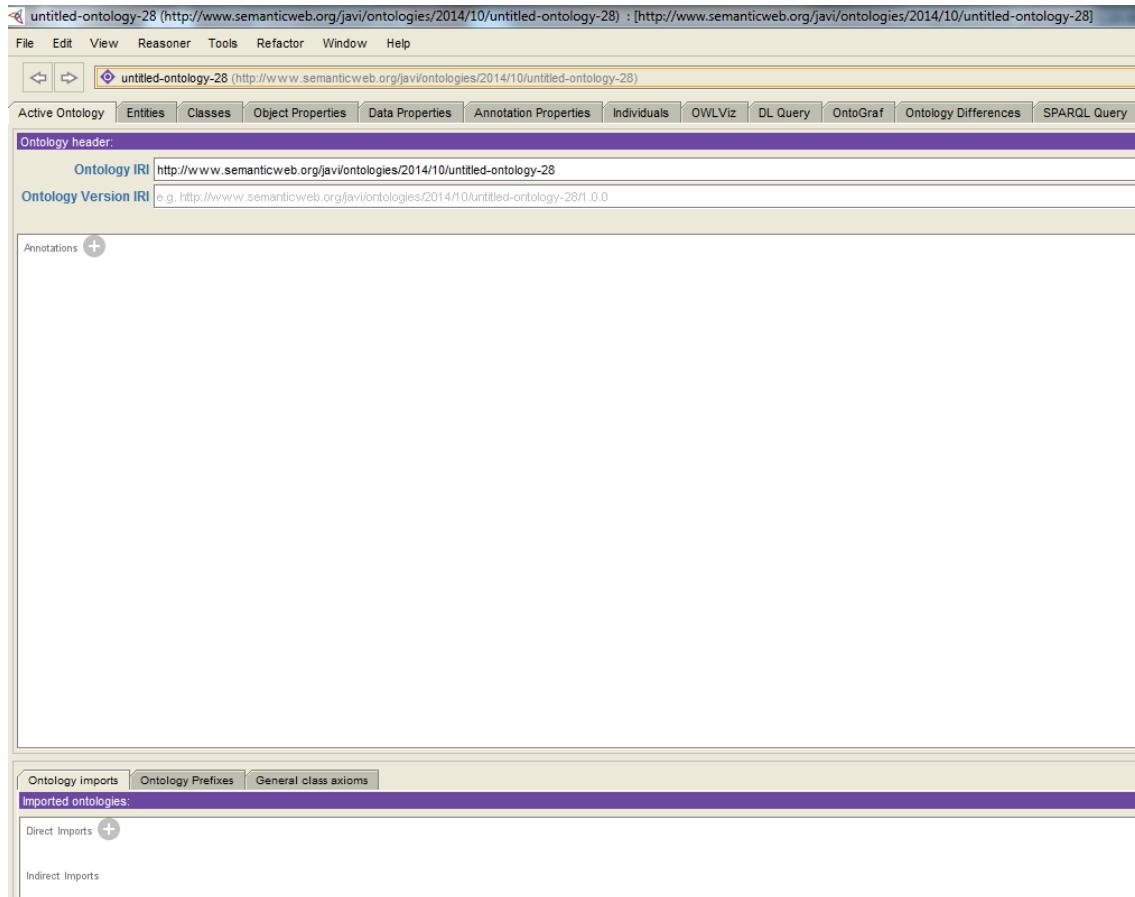
A fecha de hoy, la versión más reciente es la versión 5.0, pero está en fase beta. Por este motivo vamos a utilizar la versión anterior 4.3. Podemos descargar el archivo de instalación desde http://protege.stanford.edu/download/protege/4.3/installanywhere/Web_Installers/ .

Una vez instalada, instalaremos después los plugins que puedan ser necesarios.

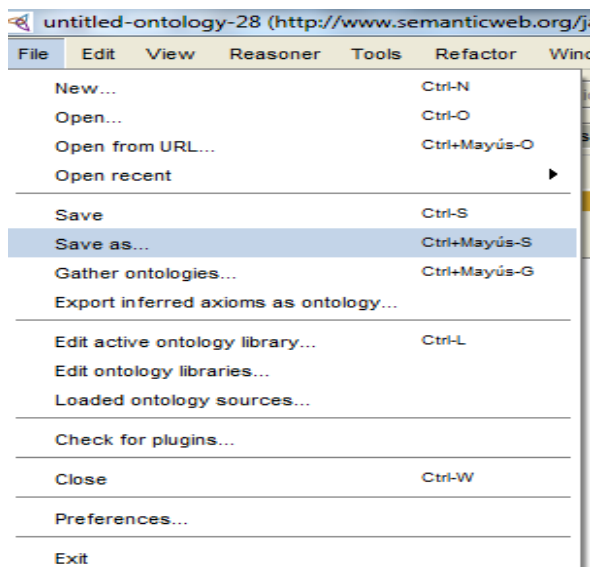
Información sobre como comenzar a trabajar con Protégé , la tenéis en:

- [Getting Started with Protégé-OWL](#) (4.x series).

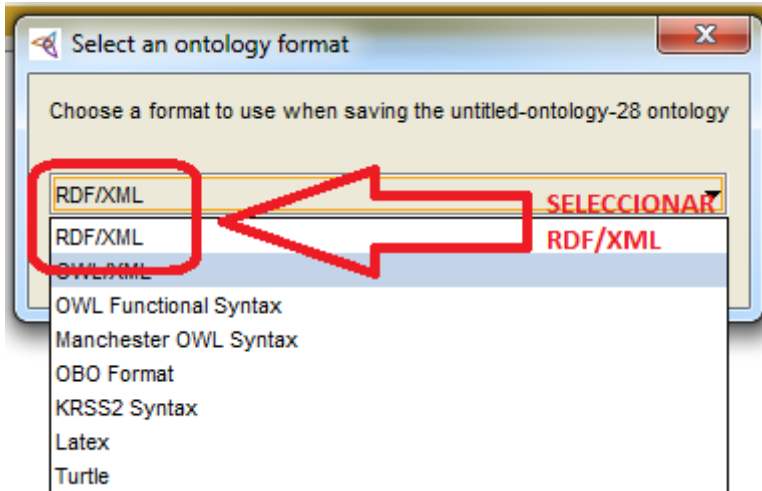
Básicamente, en nuestra primera vez, hay que indicar a Protégé que vamos a crear un nuevo proyecto y por defecto se abrirá una pantalla indicando la URI (nombre: “untitled-ontology-28”): <http://www.semanticweb.org/javi/ontologies/2014/10/untitled-ontology-28>



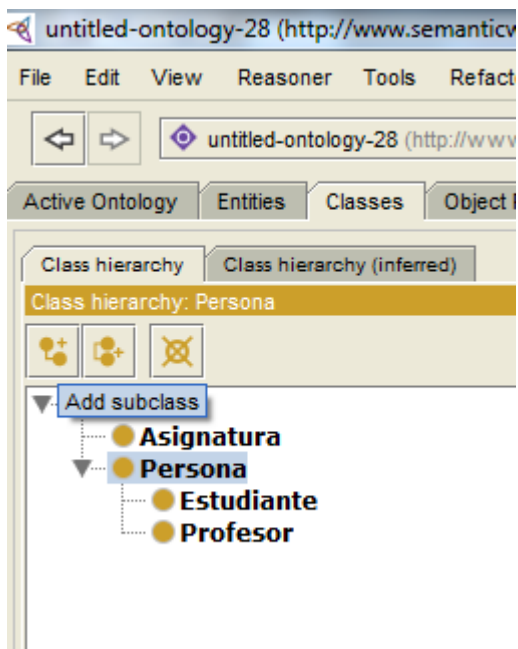
Es conveniente que antes de crear nada, guardéis con el nombre que queráis vuestro modelo.



Y después indiquemos el formato de nuestro modelo que deberá ser RDF/XML.



El uso de Protégé es muy intuitivo, por ejemplo, para poder crear una jerarquía de clases tendremos que seleccionar la pestaña de clases e ir añadiendo subclases según nos convenga:

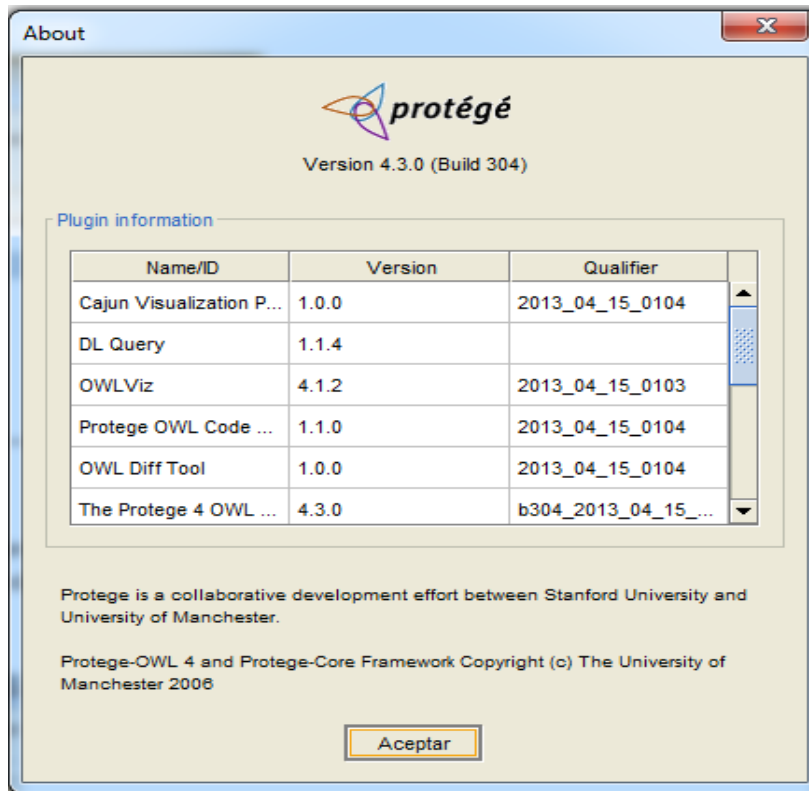


De la misma forma, también podremos ir añadiendo propiedades de tipo relación (Object Properties), propiedades de tipo de dato (Data properties) o anotaciones. Basta seleccionar las pestañas de al lado de “Classes”.

A su vez, es posible que contempléis el uso de ontologías o vocabularios que necesitareis importar.

3.1. Activación de Plugins en Protégé

En la opción de menú HELP podemos entrar en ABOUT , para ver el listado de plugins que están ya instalados.



Y de nuevo en HELP, también disponemos de un listado de plugins compatibles al menos con la versión 4.2.



[page](#) | [discussion](#) | [view form](#) | [view source](#) | [history](#)

Protege-OWL 4.2

Compatible Plugins

List of all plugins compatible with Protege-OWL 4.2.

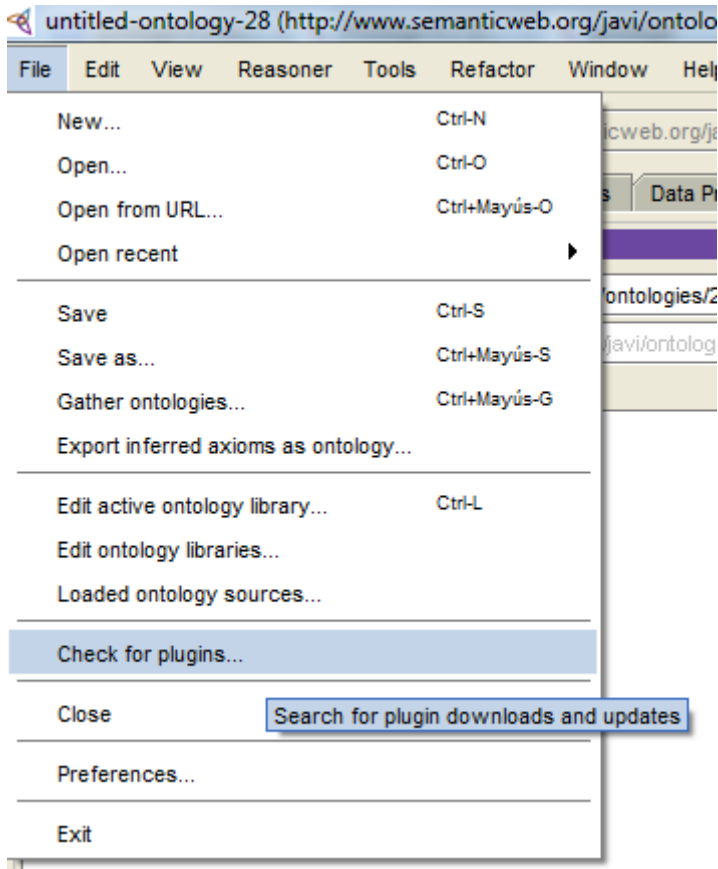
Plugin
Annotation Search View 1.1.1
Annotation Template View 1.1.1
Annotation Template View 2.1.0
BeanshellView 1.1.1
Bookmarks 1.2.2
Bookmarks 1.2.3
Cardinality View 1.1.2
Change View 1.1.1
Cloud Views 1.1.1
DL Query 1.1.0
DL-Learner 0.7 beta
ELK 0.3.0
ELK 0.4.0
HermiT 1.3.5

Navigation
 ■ [Main Page](#)
 ■ [All Categories](#)
 ■ [All Properties](#)
 ■ [All Templates](#)
 ■ [All Forms](#)
 ■ [Ontology Library](#)
 ■ [Plugin Library](#)
 ■ [Help](#)

Search
 Search

Toolbox
 ■ [What links here](#)
 ■ [Related changes](#)
 ■ [Special pages](#)
 ■ [Printable version](#)
 ■ [Permanent link](#)
 ■ [Browse properties](#)

Para instalar un plugin nuevo hay que ir al menú FILE y seleccionar la opción de “Check for Plugins” como vemos en la siguiente imagen:

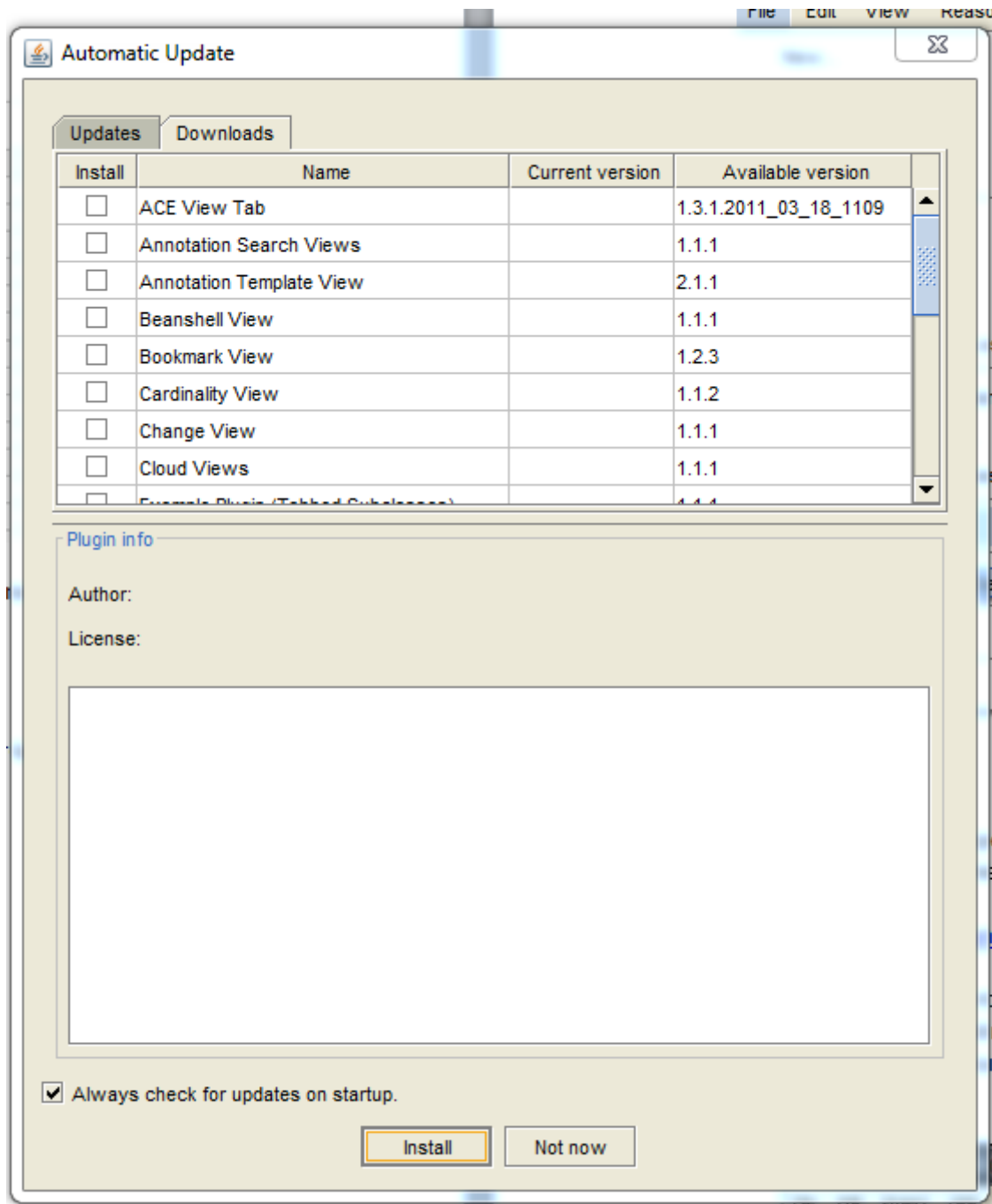


untitled-ontology-28 (<http://www.semanticweb.org/javi/ontology>)

File Edit View Reasoner Tools Refactor Window Help

- New... Ctrl-N
- Open... Ctrl-O
- Open from URL... Ctrl+Mayús-O
- Open recent
- Save Ctrl-S
- Save as... Ctrl+Mayús-S
- Gather ontologies... Ctrl+Mayús-G
- Export inferred axioms as ontology...
- Edit active ontology library... Ctrl-L
- Edit ontology libraries...
- Loaded ontology sources...
- Check for plugins...**
- Close Search for plugin downloads and updates
- Preferences...
- Exit

Al seleccionar “Downloads” nos aparecerá el listado de plugins disponibles para su posterior instalación:

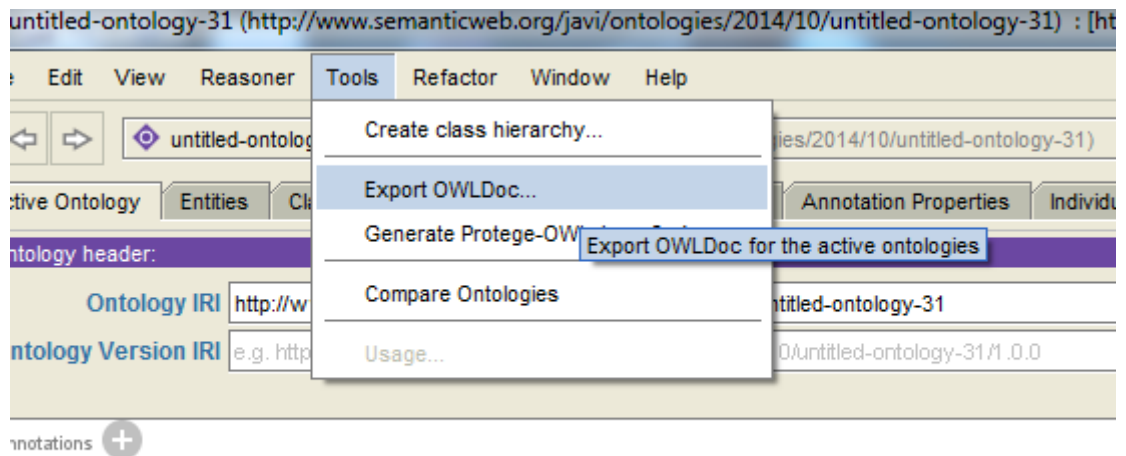


Lamentablemente, dependiendo de la versión descargada de Protégé algunos no estarán disponibles. Se recomienda visitar el apartado sobre plugins que aparece en la wiki de protégé:

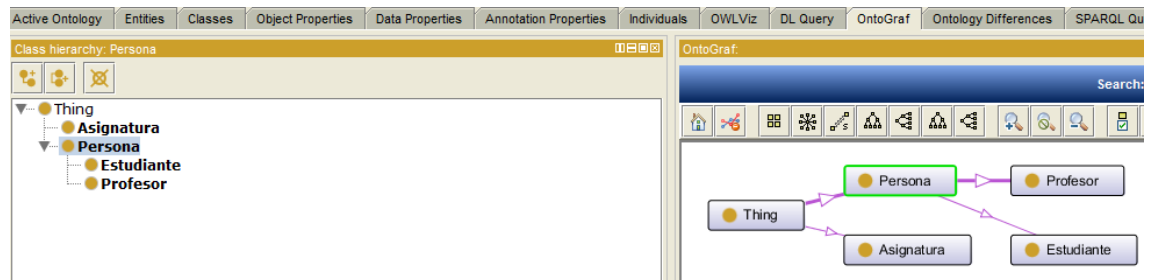
http://protegewiki.stanford.edu/wiki/Protege_Plugin_Library

A continuación se muestran una serie de plugins interesantes para la correcta realización de las prácticas.

- Plugin de documentación : "OWLDoc". Para generar el documento en HTML, basta con seleccionar la opción de "Export OWLDoc", en el menú de TOOLS. Se podría utilizar para generar la documentación del Modelo .



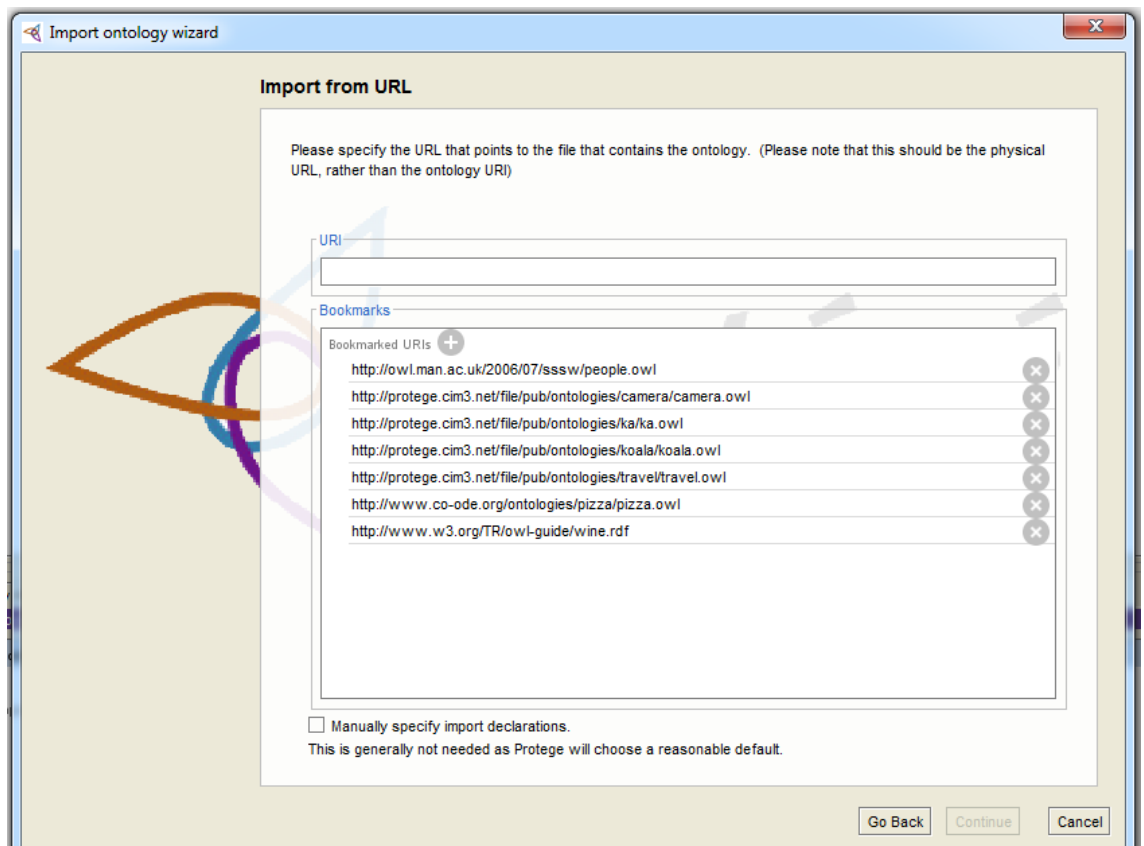
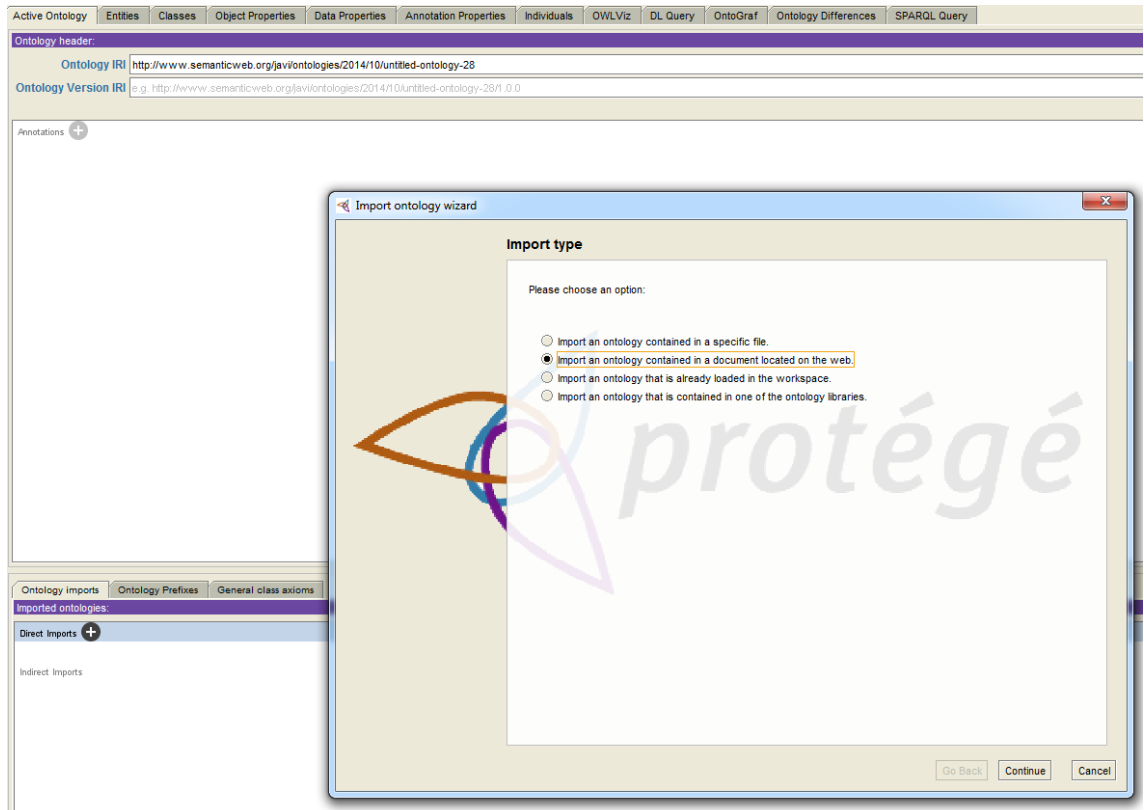
- Plugin gráficos para visualizar gráficamente el árbol taxonómico creado: : "OWLviz", OntoGraf , Jambalaya etc. Mediante los cuales se podrá complementar el documento anterior con capturas de pantalla.
 En nuestro caso haremos uso de OntoGraf el cual ya está instalado:



Si se probaran otros plugins, explicarlo en la memoria.

3.2. Importar otras ontologías y vocabularios

Por otra parte, para poder IMPORTAR otras ontologías y vocabularios , por ejemplo DC o FOAF o LOM etc. , basta con seleccionar en las carpetas desplegadas "ACTIVE ONTOLOGY" y en la parte de abajo clicar sobre el "+" de "DIRECT IMPORTS". Posteriormente seleccionar la opción de "...located on the web" e introducir la URL apropiada.



Algunas de las URLs más utilizadas son las siguientes:

Dublin Core

- <http://protege.stanford.edu/plugins/owl/dc/protege-dc.owl>
- <http://purl.org/dc/elements/1.1/>

FOAF

- <http://www.mindswap.org/2003/owl/foaf>
- <http://xmlns.com/foaf/spec/index.rdf>

Nota : esta última URL a veces ha dado problemas con Protégé, por lo que si eso ocurre usar la de Mindswap.

GEO

http://www.w3.org/2003/01/geo/wgs84_pos

OTROS.

Recursos para importar dependiendo de la versión de Protégé,:

[1] Protege 3:

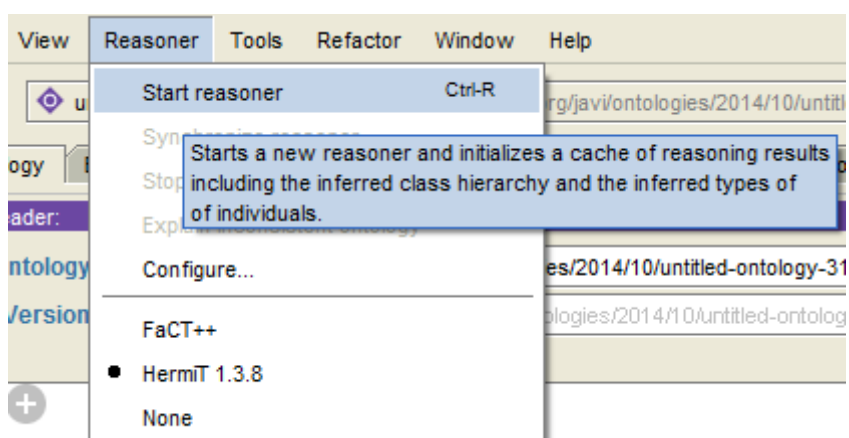
<http://protege.stanford.edu/doc/owl/owl-imports.html>

[2] Protege 4:

http://protegewiki.stanford.edu/wiki/Importing_Ontologies_in_P41

3.3. Chequear la Inconsistencia. Comprobación de modelo válido.

Usar el razonador por defecto o cualquier otro. Por defecto, tendremos 2 razonadores instalados, Fact y Hermit, y éste último estará activo. Podemos cambiarlo e incluso instalar otros como Pellet etc.



4. Implementación del modelo

Se desea desarrollar un modelo que recoja toda la información sobre la asignatura AWA, principalmente sobre los contenidos de la Guía Docente.

Información como:

- Nombre de la asignatura
- Carácter
- Profesores responsables
- Elementos del Contenido
- Contenidos de otra fuentes externas al sitio web de la asignatura
- ..

Para ello hay que completar los **cuatro pasos** siguientes:

1. Leer el siguiente documento sobre una metodología para crear un modelo ontológico u ontología: <https://sites.google.com/site/jojooa/inteligencia-artificial/definicion-de-ontology-101-guia-para-crear-tu-primera-ontologia-que-es-ontology-101>
2. Atendiendo a la información previa, diseñar en papel cómo va a ser el modelo que queréis definir. Qué clases tendrá, qué relaciones existirán entre estas clases, qué propiedades tendrá cada una y qué atributos almacenará cada clase. Se deberán crear también relaciones de tipo jerárquico (subclases y subpropiedades).
3. Crear el modelo anterior mediante Protégé y comprobar que **no existe ningún tipo de inconsistencia** mediante alguno de los razonadores integrados en Protégé: FaCT++, HermiT, etc.
4. Cuando el modelo sea consistente, hay que completarlo creando **instancias (individuales)**, relaciones y atributos de las clases que hayáis definido, de forma que tengáis varios ejemplos que aprovechen todas las capacidades de vuestro modelo. En este paso también es necesario comprobar que no existen inconsistencias.

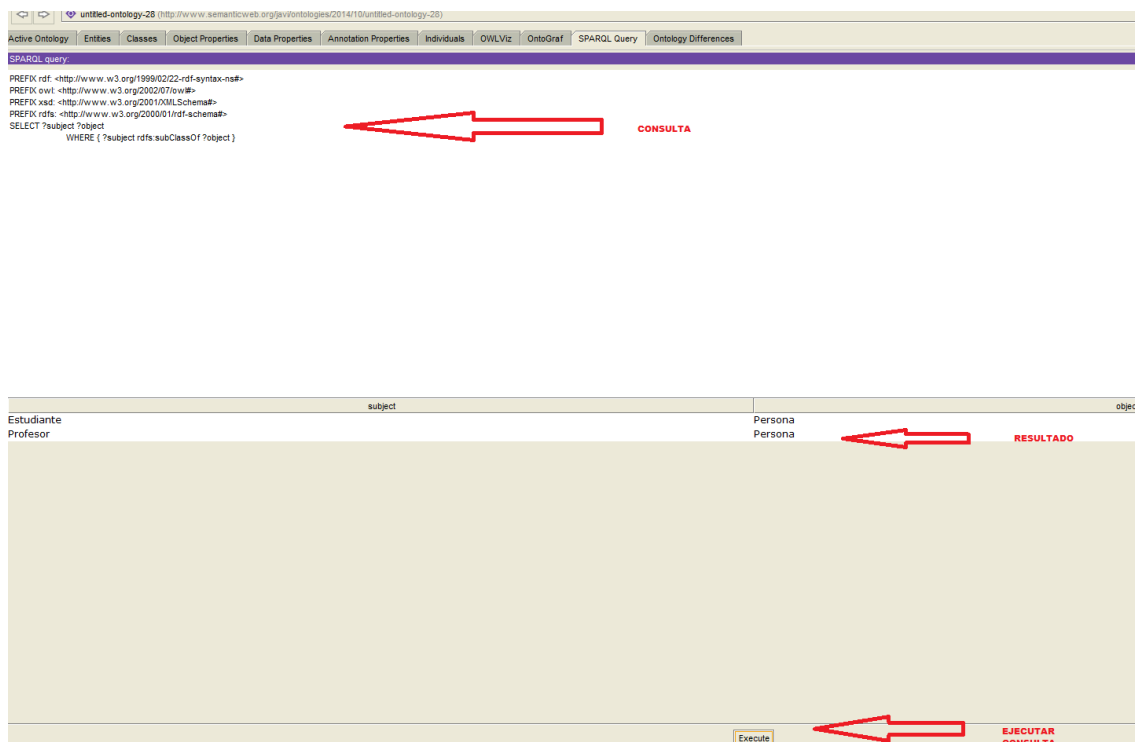
Tras finalizar los pasos anteriores, el resultado será un **fichero OWI (con extensión owl)** que almacenará toda la información de vuestro modelo. Para resolver cualquier duda sobre la creación de modelos podéis consultar la documentación de clase así como también los siguientes tutoriales que pueden ser bastante útiles:

- Guide to building a pizza ontology in 10 minutes:
. <http://protegewiki.stanford.edu/wiki/Protege4Pizzas10Minutes>
- Natalya F. Noy y Deborah L. McGuinness, “Desarrollo de Ontologías-101: Guía Para Crear Tu Primera Ontología”,
<http://ocw.uc3m.es/ingenieria-informatica/sistemas-avanzados-de-recuperacion-de-informacion/ejercicios/ontology101-es.pdf/view>

5. Consultas SPARQL

Con el modelo y sus correspondientes instancias bien definidas, podéis crear varias consultas SPARQL, como las que habéis visto en clase, que permitan obtener información sobre vuestra ontología.

El mismo protege posibilita realizar estas consultas, tal y cómo podéis ver en la siguiente imagen (pestaña SPARQL QUERY):



Si queréis probar una herramienta independiente, se recomienda utilizar la herramienta [Twinkle](http://www.idodds.com/projects/twinkle/). Una vez descargada sólo tenéis que extraerla y ejecutar el jar correspondiente. El aspecto del interfaz de Twinkle se muestra en la siguiente imagen.

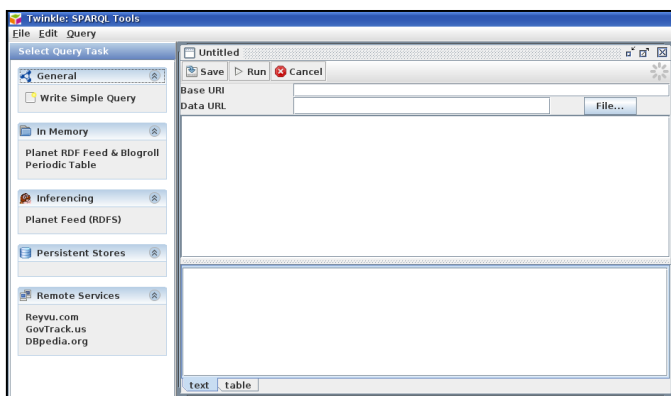


Imagen 1 – Interfaz de Twinkle 2.0. (<http://www.idodds.com/projects/twinkle/>)

Twinkle acepta como entrada ficheros RDF u OWL sobre los que se realizan las consultas SPARQL. Para ello ofrece dos opciones: seleccionar un fichero de forma local (Data URL), o bien indicar la URL en la que se encuentre almacenado este fichero (Base URI). Para este último caso, por ejemplo, podéis almacenar el fichero en vuestra cuenta web de la universidad: <http://mural.uv.es/usuario/fichero.owl>.

Tanto una opción como otra es válida y el funcionamiento de Twinkle es el mismo en los dos casos. Una vez le hayáis indicado a Twinkle cuál es el fichero RDF del modelo, es momento de utilizar el lenguaje SPARQL para realizar consultas contra ésta. Como ayuda para SPARQL podéis visitar <http://www.w3.org/2009/Talks/0615-qbe/> donde se ofrecen ejemplos que muestran el uso de distintas funciones de SPARQL. Se pueden ejecutar los ejemplos dentro de la propia web y comprobar los resultados que devuelve cada consulta.

Por otro lado, como documentación adicional en español del lenguaje SPARQL podéis consultar:

- SPARQL Lenguaje de consulta para RDF. Recomendación del W3C de 15 de enero de 2008: <http://skos.um.es/TR/rdf-sparql-query/>
- DBpedia español. Ejemplos de consultas SPARQL: <http://es.dbpedia.org/Wiki.jsp?page=Ejemplos%20de%20consultas%20SPARQL>
- Diapositivas de las clases de teoría.

5.1. Primera consulta SPARQL

Cada consulta que queráis crear **dependerá exclusivamente del diseño de vuestro modelo y de la información que queráis obtener de ella**. Por ejemplo, en caso que tengáis una ontología sobre la información de las asignaturas/profesores del Grado, podríais crear una consulta que permita obtener todas las asignaturas que imparte un profesor en concreto. Esta consulta podría realizarse tal como se muestra en el siguiente código. Aunque, como se ha comentado, ésta dependerá de las clases, relaciones y atributos de cada modelo en particular.

```
PREFIX p: <http://www.semanticweb.org/ontologies/2010/4/Master.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?asig
WHERE {
  ?prof p:nombre_profesor "Nombre Apellido1 Apellido2".
  ?prof p:imparte ?asig. }
```

En este caso, tenéis que sustituir Nombre Apellido1 Apellido2 por el nombre y apellidos del profesor definido en vuestro modelo del que queráis obtener la información. La ejecución de esta consulta sobre el modelo ejemplo da como resultado la tabla que se muestra en la imagen 2. Podéis comprobar que en los resultados obtenidos, el profesor 'Jose Javier Samper Zapater' imparte la asignatura SWWS . En este caso concreto no en un grado sino en un Máster cuyo modelo semántico de información estaba contenido en un fichero denominado "Master.owl".

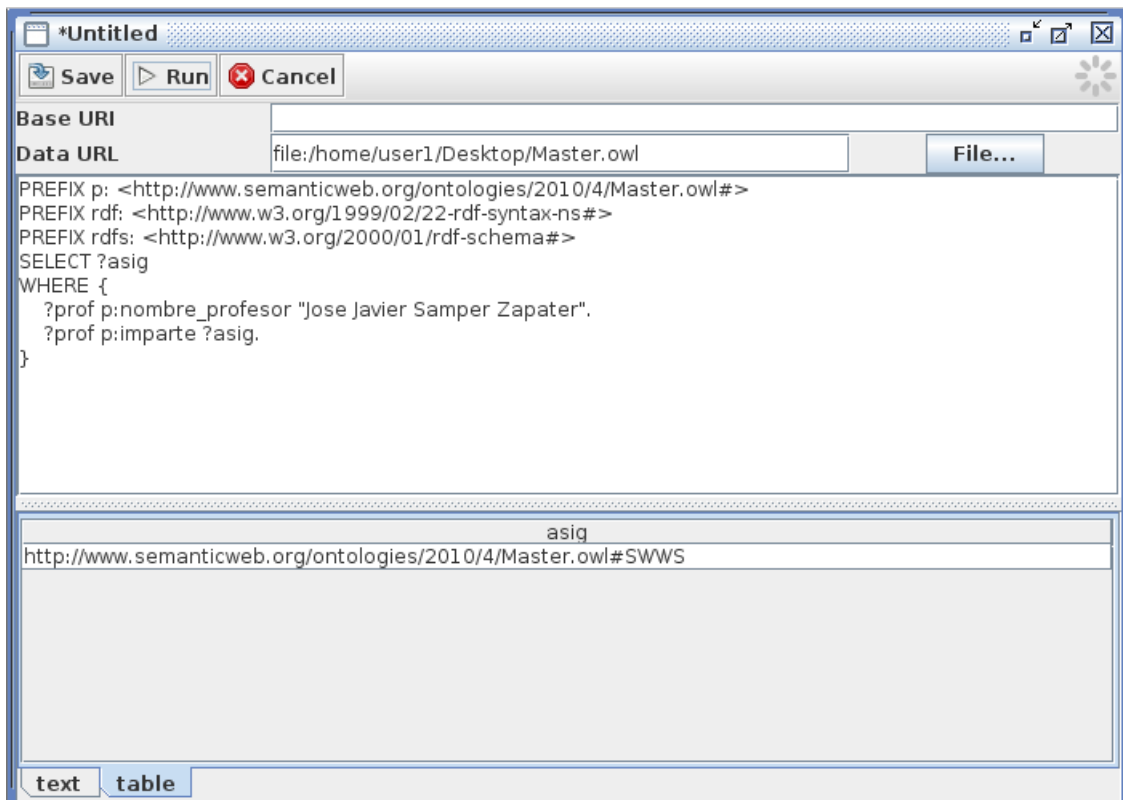


Imagen 2 - Primera consulta con SPARQL.

El proceso para realizar cualquier consulta es el mismo: **se escribe el código de la consulta en el recuadro superior de Twinkle, se ejecuta con 'Run' y el resultado se mostrará en el recuadro inferior.** Si existen errores sintácticos con la consulta, Twinkle ofrecerá un mensaje de error en el que se indicará el tipo de error y la línea en la que aparece.

6. [Sumario de la documentación a entregar](#)

Haciendo uso de los diferentes plugins que generan información del modelo, o sin ellos, se deberá entregar al profesor una memoria explicativa que contenga la explicación del modelo con capturas gráficas y detalles de cada uno de sus elementos **a través de la correspondiente actividad del aula virtual.**