

Licenciatura en Ingeniería en Computación

Proyecto Tecnológico

*Extracción y Representación de servicios web escritos
en SAWSDL mediante una ontología*

Daniel Hernández Rubio

209200879

Dr. Maricela Claudia Bravo Contreras

Trimestre 2014 Invierno

10 de abril de 2014

Declaratorias

Yo, Dr. Maricela Claudia Bravo Contreras, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Firma

Yo, Daniel Hernández Rubio, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Firma

Resumen

Actualmente el uso de los servicios web, en el ámbito tanto educativo como laboral, es muy común y de mucha utilidad. Los desarrolladores publican dichos servicios con la finalidad de que otro grupo de desarrolladores pueda hacer uso de ellos. Conforme esta tecnología ha ido avanzando y desarrollándose cada vez más, se ha tenido que ir avanzando de igual manera en la estructura que compone el servicio web, añadiéndole componentes adicionales como son las anotaciones semánticas.

Los servicios web con anotaciones semánticas nos ayudan a definir de mejor manera los componentes del servicio web, tales como las estructuras de entrada y salida de mensajes, interfaces y operaciones. Las anotaciones proporcionan mecanismos por los cuales los conceptos de los modelos semánticos que se definen, ya sea dentro o fuera del documento del servicio web, pueda hacer referencia desde dentro de los componentes como anotaciones.

La relevancia de este proyecto es que colabora en la solución de invocar a los servicios web de forma automatizada ya que con ayuda de una herramienta la cual nos permita obtener las anotaciones semánticas de un servicio web, dichas anotaciones se modelarán en una ontología. Al usar un modelo ontológico se logra documentar de una mejor manera la descripción de los servicios ya que se asocian conceptos y relaciones del modelo ontológico con parámetros y operaciones de un servicio web.

Tabla de contenido

Introducción	6
Justificación	7
Antecedentes	7
Objetivo general.....	8
Objetivos específicos.....	8
Marco Teórico	9
Servicios Web	9
Servicio Web semánticos	9
La Web semántica	10
Ontologías	10
OWL.....	11
SAWSDL.....	12
Desarrollo del Proyecto.....	13
Arquitectura general	13
Extracción y representación de WSDL 1.1	14
Extracción y representación de WSDL 2.0	16
Implementación	17
Herramientas utilizadas	17
APIs utilizadas.....	18
Análisis y Discusión de Resultados	20
Resultados con archivos del tipo WSDL 1.1	20
Resultados con archivos del tipo WSDL 2.0	24
Conclusiones	25
Referencias.....	26
Manual de Usuario	28
1.- Instalación.....	28
2.- Interfaz de inicio.....	28
3.- Uso de la herramienta.....	28
4.- Uso de Protege.....	32

Tabla de Figuras

Figura 1. Arquitectura general del sistema.....	13
Figura 2. Estructura Archivos WSDL 1.1	14
Figura 3. Diseño ontológico para archivos WSDL 1.x	15
Figura 4. Estructura Archivos WSDL 2.0	16
Figura 5. Diseño ontológico para Archivos WSDL 2.0	17
Figura 6. Pantalla inicial de la pestaña Sistema	20
Figura 7. Despliegue de los diferentes lenguajes de descripción de servicios	20
Figura 8. Despliegado de la herramienta para subir los archivos WSDL 1.1.....	21
Figura 9. Pantalla de selección de archivos a subir.....	22
Figura 10. Pantalla con los archivos subidos al servidor	22
Figura 11. Pantalla inicial de Protégé.....	23
Figura 12. Estructura de la ontología creada.....	24

Introducción

El lenguaje de descripción de servicios web, WSDL¹ [1] por sus siglas en inglés, especifica una manera de describir las funcionalidades abstractas de un servicio y concretamente cómo y dónde invocarlo.

Las anotaciones semánticas para WSDL, SAWSDL¹ [2] por sus siglas en inglés, definen cómo añadir anotaciones semánticas para varias partes de un documento WSDL, en sus dos versiones existentes WSDL 1.1 [3] y WSDL 2.0 [4], tales como las estructuras de entrada y salida de mensajes, interfaces y operaciones. SAWSDL no especifica un lenguaje para la representación de los modelos semánticos, por ejemplo, ontologías. En su lugar, proporciona mecanismos por los cuales los conceptos de los modelos semánticos que se definen, ya sea dentro o fuera del documento WSDL, pueden hacer referencia desde dentro de los componentes de WSDL como anotaciones. Esta semántica, cuando se expresa en lenguajes formales, ayuda a eliminar la ambigüedad de la descripción de servicios web durante el descubrimiento automático y la composición de los servicios web.

La recomendación del W3C² para WSDL no incluye la semántica en la descripción de servicios web. Por lo tanto, la descripción del servicio web puede tener dos servicios similares, mientras que las descripciones significan cosas totalmente diferentes o pueden tener descripciones muy diferentes pero significado similar. La resolución de tales ambigüedades en las descripciones de servicios web es un paso importante hacia la automatización del descubrimiento y composición de servicios web.

Se necesita implementar una aplicación, que dados los servicios descritos en lenguaje SAWSDL, los transforme a su correspondiente representación en OWL³[5] para permitir registrarlos, modificarlos o eliminarlos dentro de la ontología[6] que se diseñará e implementará como parte del desarrollo del proyecto.

¹ SAWSDL Semantic Annotation for WSDL

² W3C World Wide Web Consortium

³ OWL Ontology Web Language

Justificación

Los desarrolladores de software a menudo buscan en los repositorios de servicios web públicos aquellos servicios web que cumplan con ciertos requerimientos para integrarlos a sus aplicaciones. Sin embargo, la implementación de las interconexiones de los servicios web debe realizarse de forma manual o semiautomática, esto se refiere a que el desarrollador no cuenta con una API⁴ que le facilite la invocación del o de los servicios, teniendo el que desarrollar el modulo de conectividad. Además la falta de una buena documentación de los servicios, hace más complicado su invocación. Derivado de estos problemas, para mejorar la documentación, se han propuestos diferentes soluciones, una de ellas es el uso de ontologías. La relevancia de este proyecto es que colabora en la solución de invocar a los servicios web de forma automatizada. Al finalizar el proyecto se obtendrá una aplicación que permita, a partir de varios archivos de descripción escritos en distintos lenguajes de descripción de servicios (SDL⁵), encontrar los elementos más relevantes y llevarlos a un modelo ontológico general, a partir del cual se podrá generar el objeto cliente para invocar al servicio de forma remota. Además podemos considerar el hecho de que las herramientas que describen los servicios web en SAWSDL son muy escasas. Al usar un modelo ontológico se logra documentar de una mejor manera la descripción de los servicios ya que se asocian conceptos y relaciones del modelo ontológico con parámetros y operaciones de un servicio web.

Antecedentes

Antecedentes internos:

Se encontraron los siguientes proyectos terminales los cuales trabajan con servicios web, ambos proyectos se encuentran concluidos.

Proyecto Terminal 1. Plataforma de acceso remoto mediante servicios web [7]

Similitudes: La propuesta tiene como base el trabajo con servicios web

Diferencias: No es una aplicación que pretenda resolver los problemas relacionados con la invocación de servicios web, y mucho menos hace mención sobre trabajar con SAWSDL

Proyecto Terminal 2. Extracción automatizada y representación de servicios web mediante ontologías [8].

⁴ API Application Programming Interface

⁵ SDL Service Description Language

Similitudes: La propuesta tiene como base el trabajo con servicios web, maneja modelos ontológicos para la representación de servicios web, propone un modelo de conectividad.

Diferencias: No trabaja con servicios descritos en SAWSDL, ninguno de los módulos propone un analizador sintáctico de SAWSDL.

Antecedentes externos:

Artículo de revista: Artículo de revista: *On the functional quality of service (FQoS) to discover and compose interoperable Web services*[9]

Similitudes: Estudio de similitudes y comparación de SDLs Planteamiento de un modelo general

Diferencias: Diseño e implementación del modelo para SAWSDL, implementación en java de la aplicación que permite representar SAWSDL en base al modelo diseñado.

Artículo de Investigación: *Semantic Web Service Selection with SAWSDL-MX*[10]

Similitudes: Manejo de servicios del tipo SAWSDL, implementación similar en el registro de servicios, realiza operaciones de inferencia sobre las ontologías.

Diferencias: No utiliza un analizador sintáctico de SAWSDL, el artículo no especifica si su módulo hará la modificación o eliminación de los servicios registrados. Por último el uso de APIs de desarrollo son diferentes a las propuestas en esta propuesta.

Objetivo general

Diseñar e implementar un sistema que lea descripciones de servicios web en SAWSDL para extraer los elementos representativos de éstos y almacenarlos en una ontología.

Objetivos específicos

- Diseñar e implementar un analizador sintáctico de SAWSDL.
- Diseñar e implementar una ontología para la representación de servicios web descritos en lenguaje SAWSDL.
- Diseñar e implementar un módulo que permita realizar las tareas de conectividad entre el sistema y la ontología, para el registro, modificación y eliminación de individuos y relaciones entre éstos.

Marco Teórico

Servicios Web

Un servicio Web es un componente de software identificado por una URI⁶, el cual puede ser accedido mediante protocolos y estándares basados en Internet utilizando la descripción de su interfaz de comunicación descrita en un formato procesable por una máquina, por ejemplo el lenguaje WSDL. El Servicio Web está diseñado para soportar interoperabilidad entre sistemas sobre una red. Otros sistemas interactúan con los Servicios Web de la manera descrita en su descripción usando mensajes SOAP⁷, típicamente enviados usando HTTP⁸ con una serialización XML⁹. Una característica de las aplicaciones compuestas por Servicios Web, es que éstas pueden estar dispersas por la red, escritos en diferentes lenguajes de programación y en ambientes de ejecución diferentes, sin que esto impida que puedan interactuar entre sí. De esta manera, la interoperabilidad entre los diferentes componentes de un sistema, e incluso entre aplicaciones, se favorece.

Servicio Web semánticos

Los servicios web semánticos surgen de la necesidad de realizar las operaciones de descubrimiento, selección, composición, negociación, invocación, monitorización y recuperación semiautomática de los servicios web tradicionales. Un servicio web semántico está formado por un servicio web y una anotación semántica sobre dicho servicio. La anotación semántica consiste en asociar conceptos y relaciones de una ontología con parámetros y operaciones de un servicio web. Los Servicio Web Semánticos son servicios que se describen a sí mismos y son susceptibles al descubrimiento, composición e invocación automática. Dicho de otra manera, los servicios Web semánticos están habilitados con una descripción de sí mismos más formal, lo cual permite que sean interpretados por otros programas. La descripción semántica de los Servicios Web va a permitir que las tareas de composición e invocación de servicios, entre diferentes usuarios y dominios sean automáticas. Existen tecnología para Servicios Web que solo proveen una descripción a nivel sintáctico, lo que provoca que tanto consumidores como proveedores

⁶ URI Uniform Resource Identifier

⁷ SOAP Simple Object Access Protocol

⁸ HTTP Hyper Text Transfer Protocol

⁹ XML eXtensible Markup Language

tengan dificultad para representar el verdadero significado de los parámetros de entrada y salida de los servicios. Estas limitaciones pueden ser superadas mediante la representación semántica de los servicios mediante la utilización de una ontología de servicios, habilitando a las máquinas interpretar sus capacidades y hacer la integración con el respectivo dominio de conocimiento al que pertenece cada servicios.

La Web semántica

La Web semántica es una visión del futuro de la Web donde la información está dando un significado explícito, permitiendo que las máquinas puedan procesar automáticamente e integrar la información disponible en la Web. La Web semántica se basará en la capacidad de XML para definir esquemas de etiquetas a medida y en la aproximación flexible de RDF¹⁰ para representar datos. El primer nivel requerido por encima de RDF para la Web semántica es un lenguaje de ontologías que pueda describir formalmente el significado de la terminología usada en los documentos Web. Si se espera que las máquinas hagan tareas útiles de razonamiento sobre estos documentos, el lenguaje debe ir más allá de las semánticas básicas del Esquema RDF.

La web semántica es una extensión de la web tradicional que permite a las máquinas interpretar el significado de los datos, en base a anotaciones semánticas basadas en descripciones ontológicas. Esta anotación semántica permite dotar a la gran cantidad de información de la web de una estructura lógica, basada generalmente en soluciones de gestión del conocimiento, que permite a los usuarios y las máquinas buscar y relacionar información de una forma más automática y directa.

Para anotar estos recursos de la web se emplean ontologías. Una ontología es una descripción de una pequeña parte del mundo real, de los tipos de elementos que aparecen en ese mundo, de las relaciones entre ellos, de los elementos existentes y de sus restricciones.

Ontologías

Una ontología se define como una especificación formal de una conceptualización compartida. En otras palabras, define los términos y relaciones básicas que comprenden el vocabulario de un tópico de área, así como las reglas para combinar los términos y

¹⁰ RDF Resource Description Framework

relaciones para definir extensiones al vocabulario. Además, puede verse a las ontologías como un sistema para la representación de una base de conocimiento.

Las ontologías se han desarrollado dentro de la comunidad de investigación de Modelado del Conocimiento, con el fin de facilitar el intercambio y reutilización del conocimiento. Estas proveen una gran capacidad de expresividad al momento de hacer el modelado del dominio de conocimiento, lo que permite que éste pueda ser comunicado entre personas y sistemas de aplicación heterogéneos y distribuidos.

Al igual que con la arquitectura de servicios Web, la Web Semántica establece estándares los cuales tiene su base en URIs y el esquema XML. Los actuales componentes del marco de trabajo de la Web Semántica son: RDF, Esquema-RDF, y el lenguaje OWL.

OWL

Una ontología en OWL se puede definir básicamente mediante tres elementos principales:

- Las clases representan entidades del entorno que se esté describiendo
- Las relaciones representan vínculos entre las entidades
- Las instancias son afirmaciones concretas sobre el mundo real.

OWL define además un conjunto de propiedades que pueden ser aplicadas a los elementos anteriores, como pueden ser: restricciones de cardinalidad para las relaciones, propiedades de herencia para las clases, de igualdad o desigualdad, etc. OWL ha sido diseñado para poder construir la Web semántica ya que se necesita poder representar el conocimiento de forma que sea legible por los ordenadores, esté consensuado, y sea reutilizable. OWL forma parte de un conjunto creciente de recomendaciones del W3C relacionadas con la Web semántica. XML proporciona una sintaxis superficial para documentos estructurados, pero no impone restricciones semánticas en el significado de estos documentos. XML Schema es un lenguaje que se utiliza para restringir la estructura de los documentos XML, además de para ampliar XML con tipos de datos. RDF es un modelo de datos para objetos (recursos") y relaciones entre ellos, proporcionando una semántica simple para éste. Este tipo de modelo de datos puede ser representado en una sintaxis XML. RDF Schema es un vocabulario utilizado para describir propiedades y clases de recursos RDF, con una semántica para la generalización y jerarquización tanto de propiedades como de clases.

OWL añade más vocabulario para describir propiedades y clases: entre otros, relaciones entre clases, cardinalidad, igualdad, más tipos de propiedades, características de propiedades, y clases enumeradas.

SAWSDL

Semantic Annotations for WSDL and XML Schema es una recomendación del W3C y gracias a él, se pueden añadir anotaciones semánticas con términos de una ontología a los componentes del Lenguaje de Descripción de Servicios Web para permitir la clasificación, descubrimiento, concordancia, composición e invocación de servicios Web.

SAWSDL proporciona mecanismos por los cuales los conceptos de los modelos semánticos definidos tanto dentro como fuera del documento WSDL se puede hacer referencia dentro de los componentes de WSDL como anotaciones. Esta semántica cuando se expresa en lenguajes formales puede ayudar a eliminar la ambigüedad de la descripción de los servicios de Internet durante el descubrimiento automático y la composición de los servicios Web. Los principios de diseño clave para SAWSDL son:

- La especificación permite anotaciones semánticas para los servicios Web utilizando y aprovechando el marco de extensibilidad de WSDL existentes.
- Es indiferente a lenguajes de representación semántica.
- Permite anotaciones semánticas para los servicios Web, no sólo por el descubrimiento de servicios Web, sino también para invocarlos.

Con base en estos principios de diseño, SAWSDL introduce tres nuevos atributos para extender WSDL y así permitir las anotaciones semánticas:

- *modelReference* para especificar la asociación entre un componente WSDL y un concepto en algún modelo semántico.
- *liftingSchemaMapping* y *loweringSchemaMapping* que se agregan a las declaraciones de elementos de esquema XML y las definiciones de tipo para especificar las asignaciones entre los datos semánticos y XML.

Desarrollo del Proyecto

Arquitectura general

A continuación se muestra el diseño del sistema, así como el de las ontologías que se utilizaron en el proyecto.

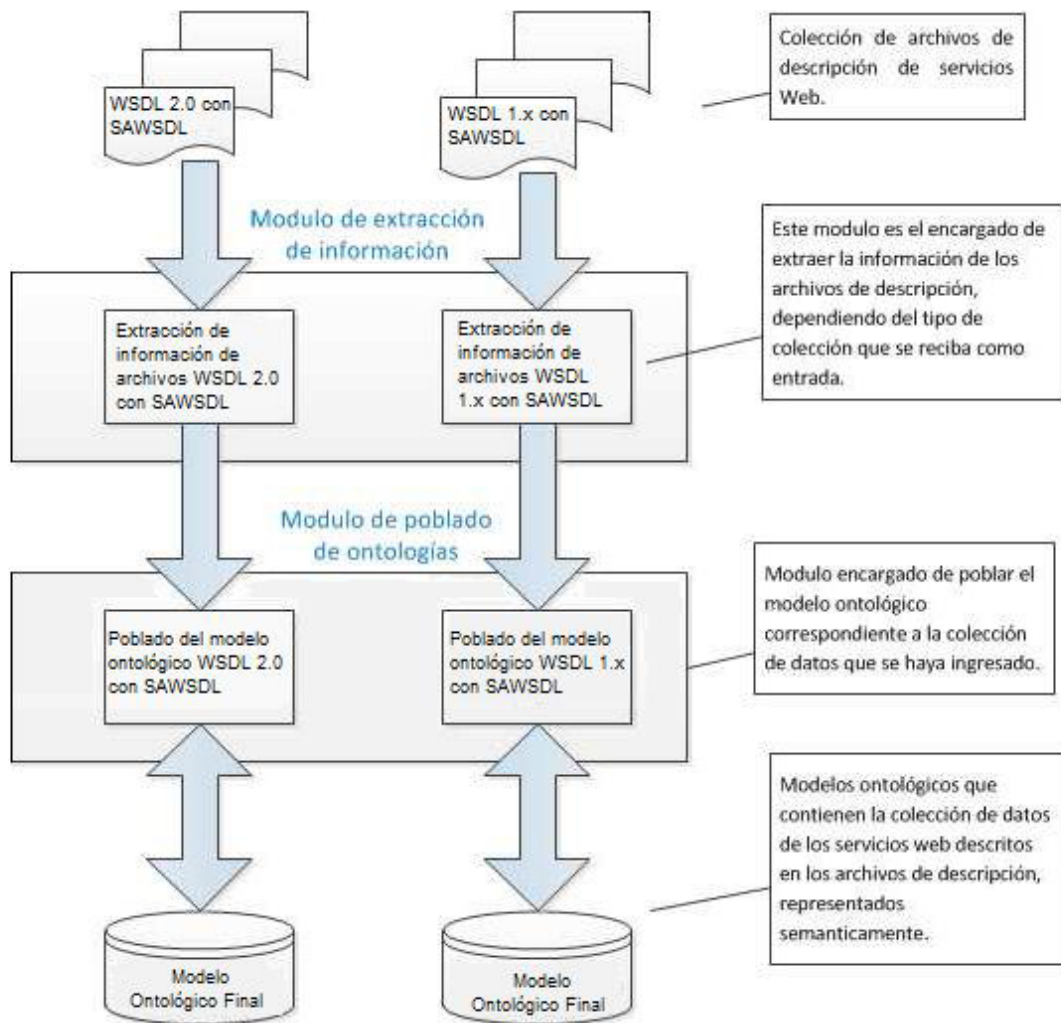


Figura 1. Arquitectura general del sistema

La Figura 1 muestra la arquitectura general del sistema, la aplicación recibe una colección de archivos de descripción de servicios Web escritos en alguno de los siguientes lenguajes de descripción:

1. WSDL 1.X con anotaciones SAWSDL
2. WSDL 2.0 con anotaciones SAWSDL

Cuando el archivo es recibido se proceso con un *parser* implementado que extrae los elementos del archivo WSDL así como sus anotaciones semánticas. El *parser* trabaja dependiendo del tipo de archivo WSDL que recibe:

Extracción y representación de WSDL 1.1

La Figura 2 muestra la estructura general de los archivos WSDL 1.1, a partir del ellos se obtendrá la información más importante contenida en cada uno de las secciones del archivo. Cada sección del WSDL tiene su correspondencia con una clase en el modelo ontológico. La sección *portType* del WSDL se corresponde con la clase *Interface* en el modelo, esto es con el fin de homologar el nombre con WSDL 2.0. De igual manera el *port* del *wsdl* 1.1 se corresponde con un *endpoint* en *wsdl* 2.0 y con la clase *Endpoint* en la ontología.

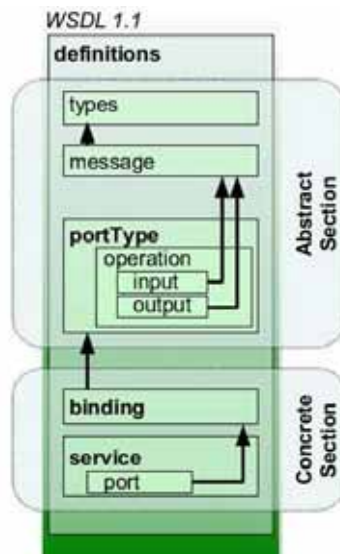


Figura 2. Estructura Archivos WSDL 1.1

Partiendo de la estructura general de los archivos de descripción escritos en WSDL 1.1 se creó el modelo ontológico para representar a los servicios descritos en este lenguaje de descripción.

La Figura 3 muestra el diseño del modelo ontológico para WSDL 1.1, todas las clases derivan de una clase principal denominada *owl:Thing*, una clase es una descripción formal

de una entidad del dominio que se quiere representar. Una clase puede tener subclases que representan conceptos que son más específicos que dicha clase. Por ejemplo *Parameter* representa a los parámetros que pueden existir en las operaciones, esto parámetros pueden ser de entrada o salida, para hacer esta distinción se crearon las clases *ParameterInput* y *ParameterOutput* ambas derivan de la clase *Parameter*, de manera similar ocurre con la clase *DataType*. Tanto *Parameter* como *DataType* son clases abstractas para evitar que existan instancias de ellas, solo se utilizan para agrupar e introducir un orden dentro de la jerarquía de clases

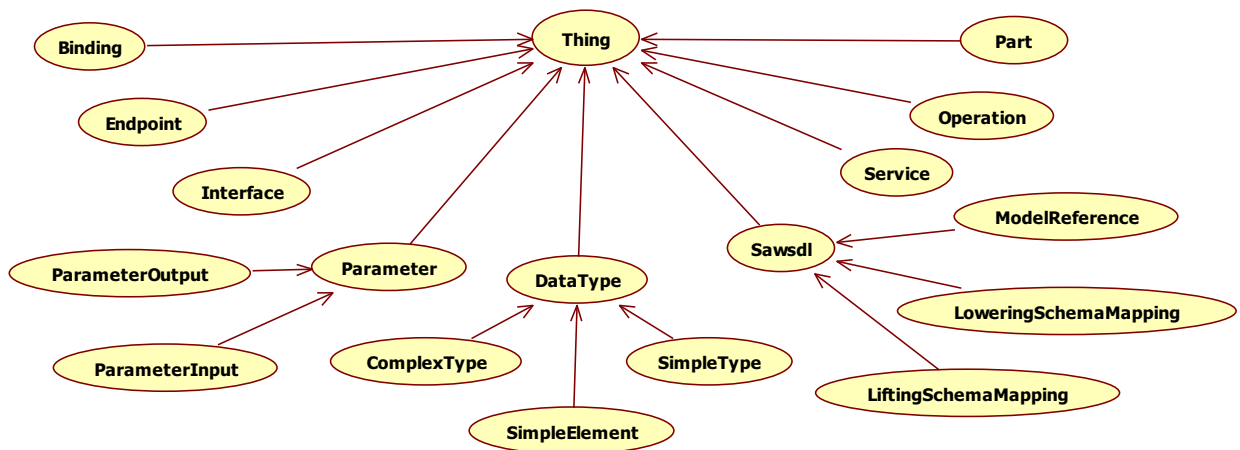


Figura 3. Diseño ontológico para archivos WSDL 1.x

Para representar las relaciones entre las clases, OWL permite establecer estas relaciones utilizando propiedades de objeto (*ObjectProperties*) y propiedades de datos (*DataProperties*).

Las propiedades de objetos permiten hacer relaciones entre las clases del modelo, lo cual permite afirmar hechos generales sobre los miembros de las clases.

Las propiedades de tipos de datos relacionan individuos con literales RDF o tipos simples de un esquema XML. Tanto las propiedades de objetos y propiedades de datos deben tener un dominio y un rango en el cual se utilicen, en ambas un dominio de propiedad reduce los individuos a los que se puede aplicar dicha propiedad. Mientras que el rango de un *ObjectProperties* reduce los individuos que una propiedad puede tener como su valor, el rango de un *DataProperties* reduce los tipos de datos que una propiedad puede tener como

su valor. Una vez terminado el proceso de estructuración de la ontología se llevo a cabo la identificación de las tres anotaciones características del lenguaje SAWSDL: *modelReference* , *liftingSchemaMapping* y *loweringSchemaMapping*, esto con la finalidad de añadirlas también al modelo ontológico como se muestra en la Figura 3.

Extracción y representación de WSDL 2.0

En la Figura 4 se tiene la estructura general de un archivo WSDL 2.0 a partir de la cual se diseño el modelo ontológico para representar este tipo de archivos. De igual forma se modelaron las relaciones que existen entre cada sección del WSDL con los datos que contienen.

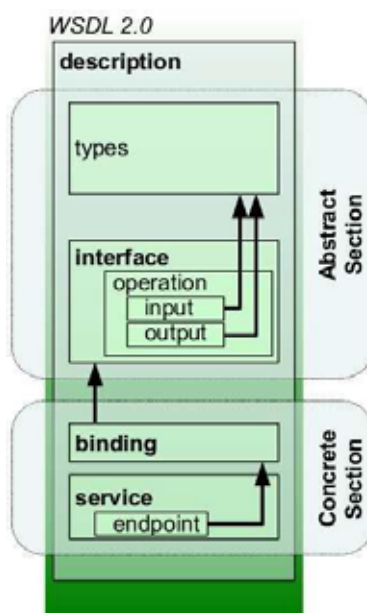


Figura 4. Estructura Archivos WSDL 2.0

La Figura 5 muestra el diseño del modelo ontológico para representar los servicios web descritos mediante la especificación del WSDL 2.0.

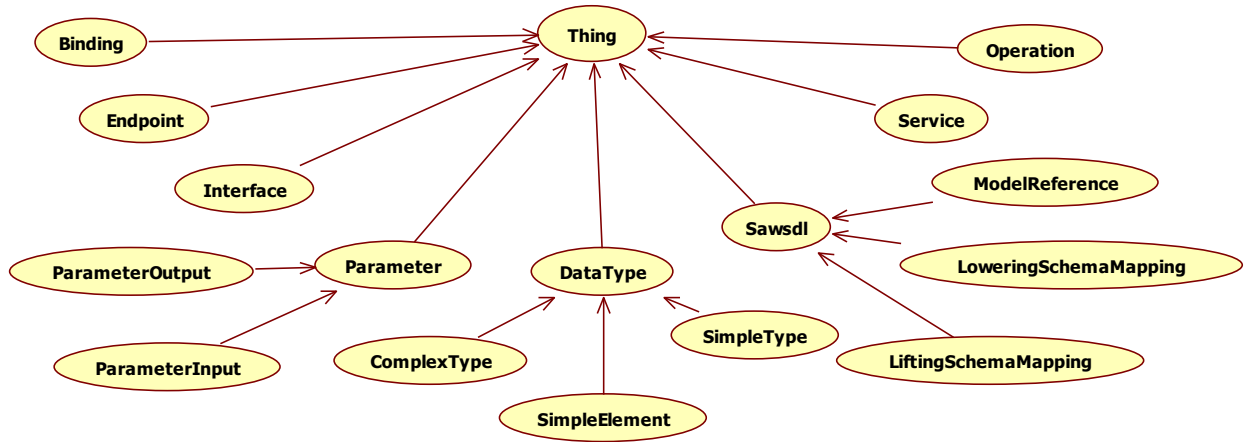


Figura 5. Diseño ontológico para Archivos WSDL 2.0

Ya con los elementos bien definidos por el *parser* se lleva a cabo la implementación de la ontología esto mediante el uso de las API's descritas más adelante. Cuando la ontología es creado se procede a poblar la ontología con los elementos parseados. La herramienta genera un archivo owl que se puede tratar con la aplicación Protégé.

Implementación

El proyecto se desarrollo en una PC con las siguientes características:

- Computadora con procesador Intel Core i3, 8 GB de memoria RAM, disco duro de 1 TB.
- Sistemas operativos Windows 7 Professional, Linux Debian 6.0
- Servidor de aplicaciones Apache-Tomcat-7.1.0

Herramientas utilizadas

El proyecto se implemento utilizando java como lenguaje de programación, mediante el uso de NetBeans [8] como entorno de desarrollo integrado (IDE). El uso de este IDE nos ayuda a programar en un desarrollo más ágil y una mejor estructuración del proyecto. Esta herramienta permitió una mejor documentación del código fuente de la aplicación así como la realización de las pruebas unitarias.

Mediante el uso de la opción *debug* se pudieron encontrar errores sobre el funcionamiento del sistema, ya que esta opción te permite marcar los puntos de interrupción en donde se tuviera algún funcionamiento inesperado del sistema. Debido a que este tipo de errores es complicado encontrar donde existe un error, con el *debug* se pudo seguir el cambio en las

variables y ver donde se tenía un valor que no era el esperado y corregir la lógica de programación.

Se utilizó Protégé 4.3 [9] como editor de ontologías, esta herramienta facilita la creación de los modelos ontológicos, el entorno gráfico del editor es intuitivo y tiene varias pestañas para trabajar con alguna sección específica que conforma la ontología, como puede ser las clases que conforman el modelo, las propiedades de las clases y las propiedades de tipos de datos, entre otros.

Permite agregar o eliminar clases del modelo de una forma intuitiva, también permite la creación de las propiedades de objetos y propiedades de datos, establecer las relaciones que existen entre las clases, agregar que propiedades tendrá cada objeto y definir el rango y el dominio que existe en cada relación que se defina entre las clases.

El uso de Protégé fue una parte fundamental en el desarrollo del proyecto, permitió el diseño de los modelos ontológicos, el desarrollo de los modelos fue de manera iterativa por lo que conforme se avanzaba en el proyecto se hicieron cambios a los modelos de tal manera que el diseño fuera mejorando paulatinamente conforme se encontraban mas requerimientos que agregar a los modelos.

APIs utilizadas

JDOM 2.0.2.- Es un API implementado en java para trabajar con archivos XML permite crear, leer y manipular este tipo de documentos. Proporciona una capa de abstracción entre un *parser* de XML y nuestra aplicación, los *parsers* que se pueden utilizar con JDOM [10] son SAX y DOM. JDOM está enfocado para trabajar en java, por lo que su uso se facilita si se tiene conocimiento de este lenguaje de programación, como los archivos WSDL y OWLS están basados en XML la manipulación de estos se facilita al utilizar esta API, al leer algún documento JDOM nos proporciona clases y métodos para manipularlo, a partir de la estructura del archivo se crea un árbol basado en nodos que nos permite recorrer los nodos y recuperar la información que se necesite. JDOM ayudo en la recuperación de la información de los archivos de descripción permitiendo una navegación a través del los nodos de una forma más fácil que si se hubiera utilizado solamente un parser XML como DOM o SAX. El uso de esta API fue de vital importancia en la implementación del proyecto, debido a que una parte fundamental del proyecto es la recuperación de la información.

OWL API.- Esta API proporciona clases y métodos para el trabajo con ontologías, permitiendo la creación, manipulación y lectura. Esta API nos permite realizar el poblado de la ontología, después de haber parseado el archivo de entrada, cargado una plantilla del modelo ontológico y de haber obtenido la información de los archivos de descripción de servicios.

Esta API esta implementada en java lo que permitió una fácil integración con el proyecto y con JDOM, de tal manera que la integración de ambas resulto sencilla y no se produjo ningún conflicto al trabajar con ella. OWL API [11] cumplió con el objetivo de crear una colección de servicios web con los datos obtenidos de los archivos de descripción, permitiendo establecer las relaciones definidas en el modelo ontológico.

Protege-OWL *Code Generator*.- Esta API permite la creación de clases java que representan a una ontología desarrollada en Protégé, una vez que se tiene elaborado el modelo ontológico y habiendo definido las clases, los *dataProperties*, los *objectProperties*, etc. esta API nos permite generar las clases necesarias para posteriormente trabajar con la ontología.

Cada clase definida en el modelo se mapea a una clase concreta en java con los métodos necesarios para poder manipular los *objectProperties* y *dataProperties* específicos a esa clase. Para la instanciación de las clases que representan a la ontología se implementa el patrón *Factory* en la creación de los objetos, de tal manera que cuando se necesite una instancia de un objeto determinado la clase *Factory* correspondiente es la encargada de retornar el objeto apropiado que se le solicite, es decir, se delega la responsabilidad de creación de los objetos al *Factory* específico al modelo ontológico correspondiente.

PrimeFaces 3.4.- Este framework fue utilizado para realizar la parte visual de la herramienta, nos facilita la creación de elementos web del tipo JSF.

Análisis y Discusión de Resultados

Resultados con archivos del tipo WSDL 1.1

1. Permitir subir al servidor únicamente archivos con extensión WSDL
2. Extracción de la información y poblado de la ontología.
3. Permitir la descarga del archivo generado, es decir, la ontología poblada.
4. Eliminación de los archivos subidos al servidor.
5. Lectura del archivo generado usando Protégé 4.3
6. Mostrar en Protégé 4.3 los datos creados a partir de la recuperación de la información de los archivos de descripción WSDL.

En las siguientes imágenes podremos ver los resultados obtenidos con las primeras pruebas del sistema:

En la figura 6 se muestra la pantalla correspondiente a la pestaña de sistema:



Figura 6. Pantalla inicial de la pestaña Sistema

En la figura 7 se observan los distintos lenguajes de descripción de servicios con los que puede trabajar el sistema. Cualquiera de las dos opciones es funcional pero para fines de esta demostración se ocupará la definición WSDL 1.x



Figura 7. Despliegue de los diferentes lenguajes de descripción de servicios

Una vez que se selecciona el lenguaje WSDL 1.x, se da clic en el botón iniciar con el cual nos desplegara la herramienta para poder seleccionar y subir los archivos WSDL para su análisis, como lo muestra la Figura 8.



Figura 8. Despliegado de la herramienta para subir los archivos WSDL 1.1

El sistema nos muestra una advertencia por si se subirán más de 200 archivos al servidor, para que carga de los archivos se realice en bloques de no más de 200, esto con el fin de evitar la carga del explorador que se esté usando. Actualmente es posible la modificación del tamaño de los archivos subidos al servidor, pero por fines prácticos este se dejo en bloques de 200 archivos.

Para buscar los archivos que se subirán al servidor y se van a utilizar presionamos el botón buscar, esta acción nos abrirá el explorador de archivos de nuestro sistema operativo, de tal forma que podamos navegar hasta la ubicación donde se encuentre la colección de WSDL, enseguida seleccionamos el o los archivos que se subirán al sistema y damos clic en el botón abrir, así como lo muestra la Figura 9.

Una vez que presionamos el botón abrir se nos mostraran en el componente los archivos seleccionados. Y procedemos a subir los archivos al servidor, esto se realiza al presionar el botón subir.

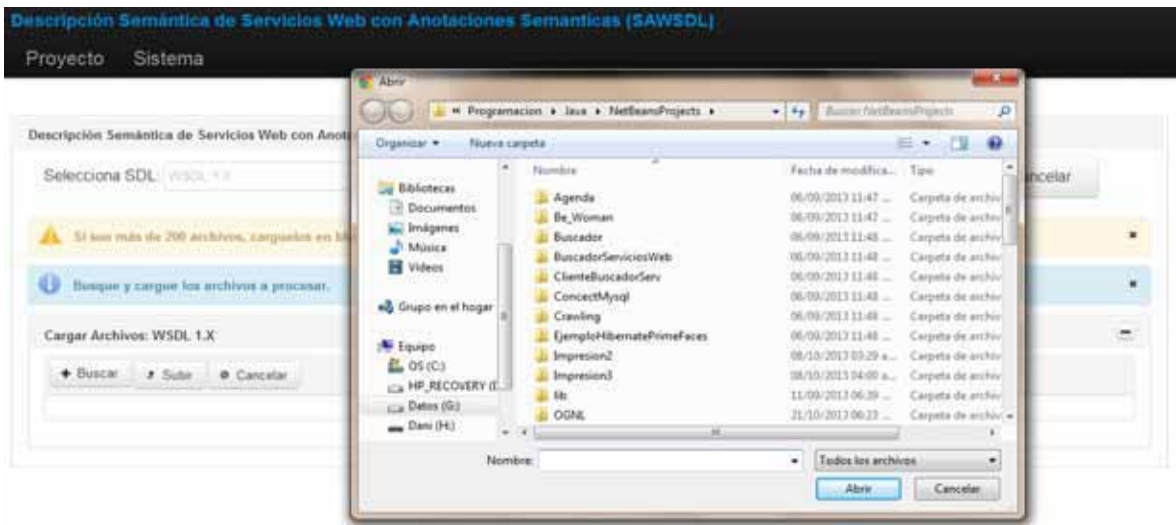


Figura 9. Pantalla de selección de archivos a subir

La Figura 10 nos muestra una lista con los nombres de los archivos que se subieron al servidor y el total de ellos. Si se requiere subir más archivos volvemos a realizar los pasos anteriores, es decir, buscar, seleccionar y subir los archivos.

Una vez realizado la carga de archivos procedemos a poblar la ontología, para empezar a realizar este proceso damos clic en el botón poblar. El proceso de poblado implica la extracción de la información de cada uno de los archivos, la inserción de esta información en el modelo ontológico y guardar los cambios que se realizan al modelo.

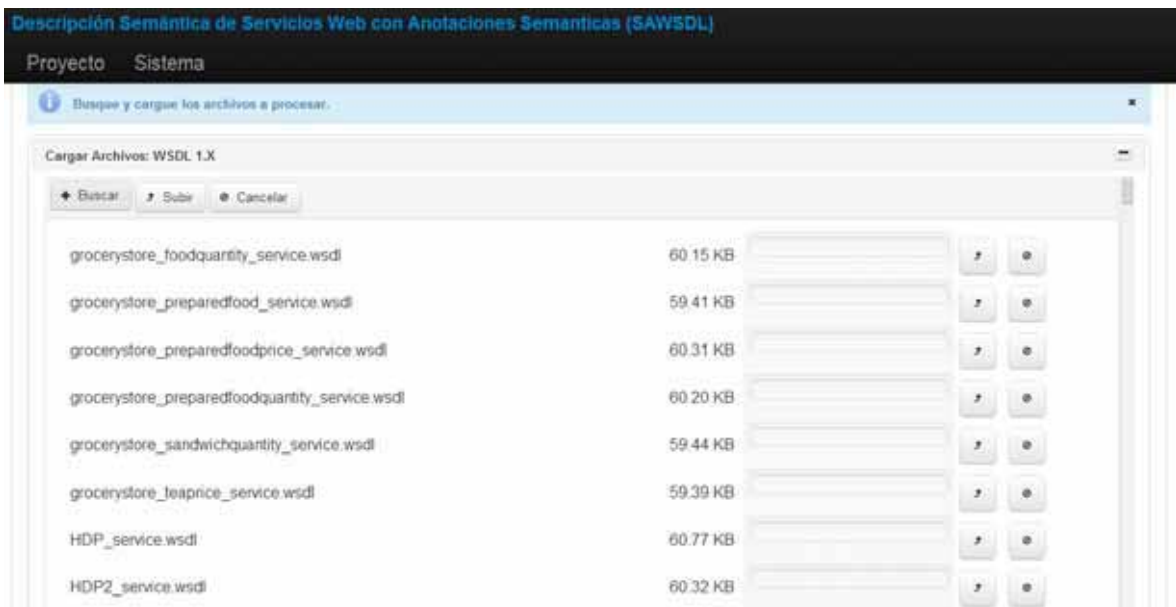


Figura 10. Pantalla con los archivos subidos al servidor

Una vez poblado el modelo se procese a descargar la ontología creada. Para ello se da clic en el botón de descargar. Después de haber realizado la descarga de la ontología procedemos a abrir el archivo, Protégé 4.3 nos permite trabajar con ontologías, por lo que utilizamos este programa para abrir el archivo OWL.

La figura 11 muestra la pantalla inicial de Protégé 4.3, para poder utilizar la ontología descargada damos clic en *File, Open* y seleccionamos el archivo descargado.

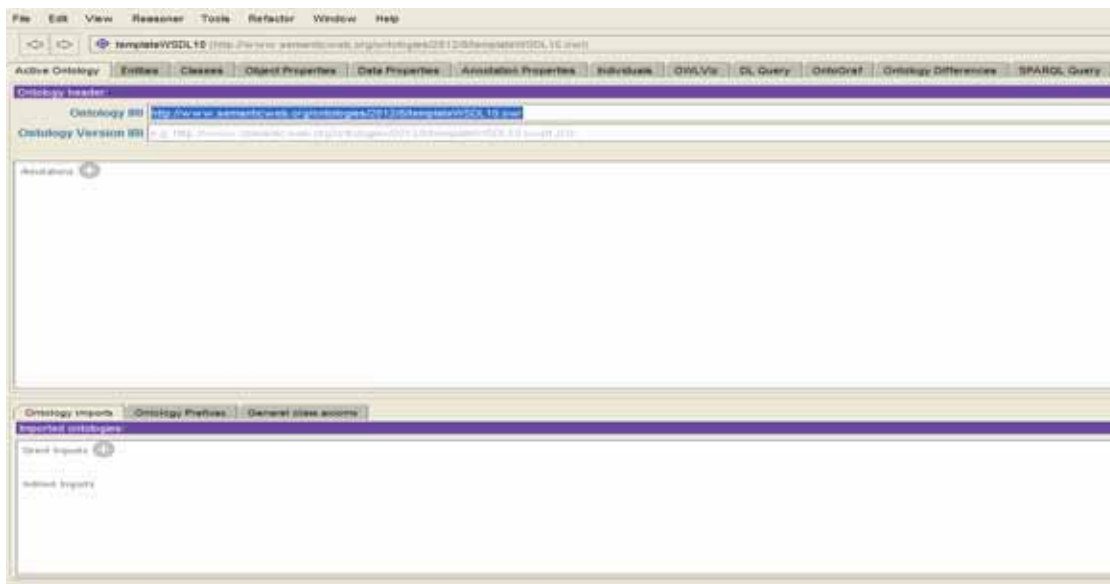


Figura 11. Pantalla inicial de Protégé

Una vez seleccionado el archivo nos dirigimos a la pestaña de *Entities* donde podremos ver la estructura de nuestra ontología así como sus individuos, como se muestra en la Figura 12.

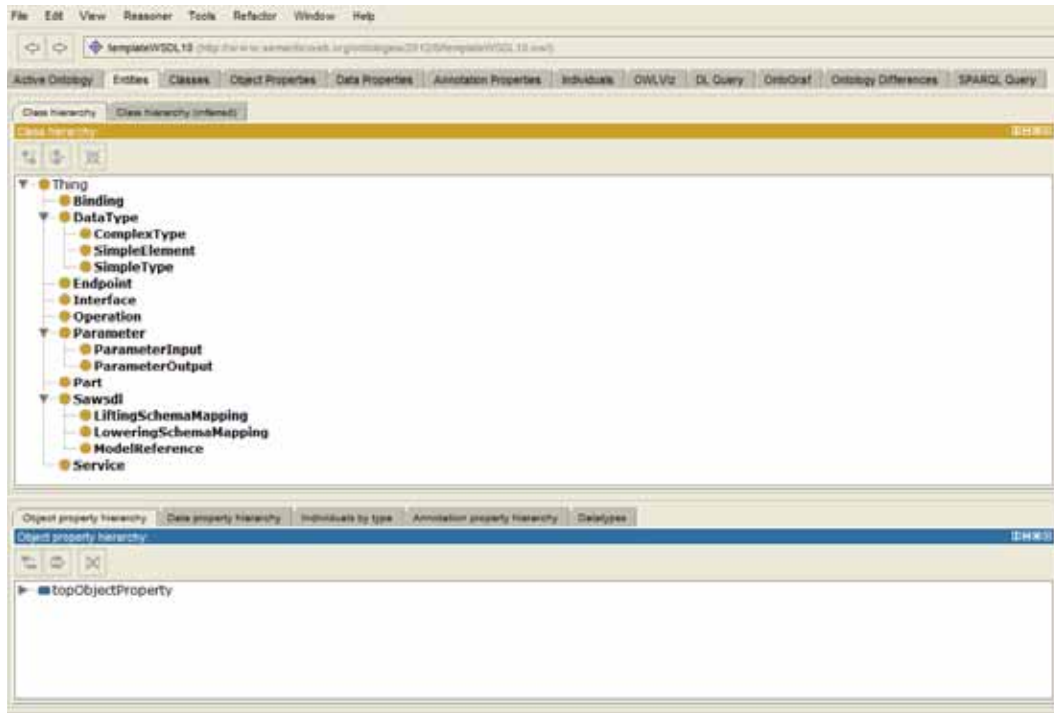


Figura 12. Estructura de la ontología creada

Seleccionando alguno de los elementos del árbol que se despliega podremos obtener mas información acerca de los individuos que forman la ontología. Cabe mencionar que los individuos que nos interesan se encuentran en el nodo de SAWSDL, que describen las propiedades de las anotaciones semánticas que arrojo el análisis de los archivos cargados al principio.

Resultados con archivos del tipo WSDL 2.0

1. Permitir subir al servidor únicamente archivos con extensión WSDL
2. Extracción de la información y poblado de la ontología.
3. Permitir la descarga del archivo generado, es decir, la ontología poblada.
4. Eliminación de los archivos subidos al servidor.
5. Lectura del archivo generado usando Protégé 4.3
6. Mostrar en Protégé 4.3 los datos creados a partir de la recuperación de la información de los archivos de descripción WSDL.

Los resultados obtenidos por la herramienta son similares al usar archivos del tipo WSDL 2.0, los pasos a seguir para la selección de archivos, la población de la ontología y la descarga de la ontología son similares. También se utiliza Protégé para poder trabajar con la ontología descargada siguiendo los mismos pasos.

Conclusiones

Actualmente la creación de un servicio web así como su publicación y la generación de un cliente para el mismo son tareas muy comunes, y con la ayuda de *Frameworks*, fáciles. Pero la labor de dotarlas de características que ayuden al desarrollador entender el para qué sirve cada elemento así como darles un sentido semántico para que funcionen en conjunto con otros servicios web es una tarea q hasta la fecha está en pleno desarrollo. Este proyecto me ayudo a entender la estructura de los archivos WSDL, así como sus anotaciones semánticas que lo llevan a ser un SAWSDL. Otra cosa importante fue el uso de ontologías, su diseño debe tener bien delimitado el área de conocimientos, porque de no ser así, el diseño puede estar mal y puede abarcar muchas cosas. Mas sin embargo una vez limitado el campo de estudios, una ontología es una herramienta muy poderosa y útil.

Una de las partes más laboriosas del proyecto fue el parseo y descomposición del archivo WSDL, aunque su estructura está definida ya por estándares, las anotaciones al no estar estandarizadas, le añaden un extra al parseo y a pesar de que actualmente hay muchas API's para el manejo de archivos en WSDL, ninguna sirve para archivos con anotaciones semánticas, es por ello que la elaboración del parser con estas características fue unas de las partes fundamentales del proyecto.

También se tuvo la dificultad de poblar la ontología, debido a que la utilización de la API era una tarea completamente nueva y hay poca documentación de la misma, es decir, una vez diseñado el modelo ontológico y creado el parser para recuperar los datos de los archivos de descripción con anotaciones semánticas, como hacer posible que los datos extraídos se guardaran en el modelo ontológico.

Esto se resolvió utilizando Protégé OWL API. Esta API nos permite cargar el archivo que representa el modelo ontológico y trabajar con él, nos permite editarlo, modificarlo, etc. Como cada modelo es diferente se implemento una clase por cada modelo para realizar el poblado de la ontología correspondiente.

Al final obtuvimos una herramienta que ayudara a crear repositorios de servicios web descritos semánticamente debido a las anotaciones, a partir de archivos WSDL. Además se podrán realizar búsquedas mediante reglas de inferencia sobre la o las ontologías creadas. Con esto podremos ayudar a los desarrolladores que buscan un servicio web específico, arregarles todos los servicios web que cumplan con sus características requeridas y que además cuenten con anotaciones que le ayuden a comprender mejor la utilización del mismo.

Referencias

- [1] J. Farrell (2013, Marzo 5). Semantic Annotations for WSDL and XML Schema. [En línea]. <http://www.w3.org/TR/sawSDL/>
- [2] E. Christensen et al. (2013, Febrero 25). *web Service Description Language (WSDL)*. [En línea]. <http://www.w3.org/TR/wsdl>
- [3] E. Christensen et al. (2013, Enero 25). *web Service Description Language*. [En línea]. <http://xml.coverpages.org/wsdl20000929.html>
- [4] D. Booth. (2013, Febrero 15). *web Service Description Language (WSDL) Version 2.0* [En línea]. <http://www.w3.org/TR/wsdl20primer/>
- [5] OWL Working Group (2013, Febrero 20) OWL 2 Web Ontology Language Document Overview (Second Edition <http://www.w3.org/TR/2012/REC-owl2-overview-20121211/>)
- [6] J. Heflin. (2013, Marzo 1). *OWL web Ontology Language Use Cases and Requirements*. [En línea]. Disponible: <http://www.w3.org/TR/2004/RECwebontreq20040210/>
- [7] M.M Ramírez, “Plataforma de archivos de acceso remoto mediante servicios web”, Propuesta de proyecto terminal, Universidad Autónoma Metropolitana Azcapotzalco, D.F., México, 2008.
- [8] J. P Martínez “Extracción automatizada y representación de servicios web mediante ontologías”, Propuesta de proyecto terminal, Universidad Autónoma Metropolitana Azcapotzalco, D.F., 2011.
- [9] B. Jeong, H Cho, C. Lee, “*On the functional quality of service (FQoS) to discover and compose interoperable Web services*”, International Journal of Expert Systems with Applications, vol. 33, no 3, pp 54115418, Abril 2009.
- [10] M. Klusch, P. Kapahnke “*Semantic Web Service Selection with SAWSDL^{MX}*”, German Research Center for Artificial Intelligence Stuhlsatzenhausweg 3, Saarbrücken, Germany. <http://ftp.informatik.rwth-aachen.de/Publications/CEURWS/Vol416/paper1.pdf>

[11] OWL api. Owl api[En linea]. "Website", Noviembre 2012.
<http://owlapi.sourceforge.net/>.

[12] Primefaces.org. Primefaces[En linea]. "Website", Noviembre 2012.
<http://primefaces.org/>.

Manual de Usuario

1.- Instalación

La aplicación se despliega mediante un archivo .war que se encuentra en /PTSAWSDL/dist/PTSAWSDL.war.

Este archivo debe ser pegado en la carpeta *webapps* de nuestro servidor Apache Tomcat.

Una vez pegado el archivo solo hay que reiniciar nuestro servidor.

2.- Interfaz de inicio

Para acceder a nuestra aplicación debemos acceder a la siguiente url:

<http://localhost:8080/PTSAWSDL/>

Y nos desplegará la siguiente ventana:



Para acceder a la herramienta damos clic en la pestaña de Sistema.

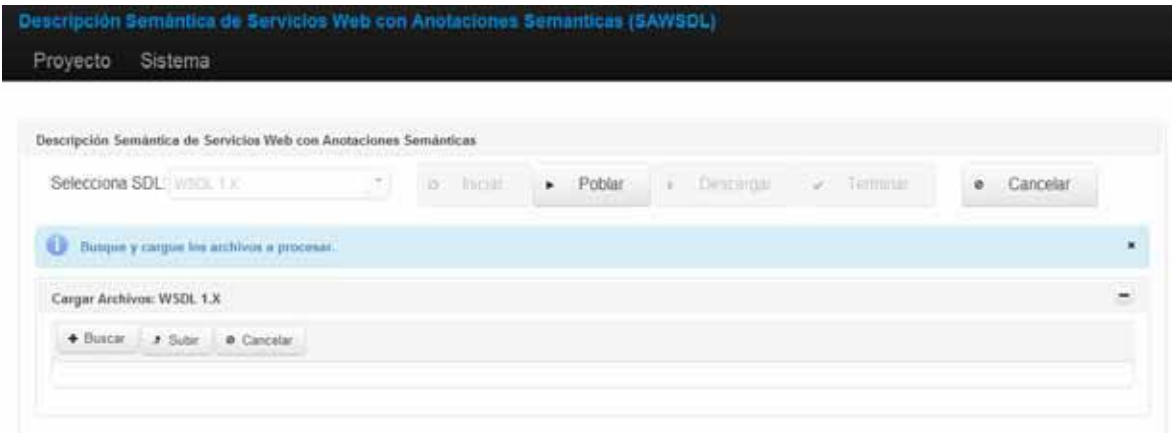
3.- Uso de la herramienta

La pantalla del sistema es la siguiente:

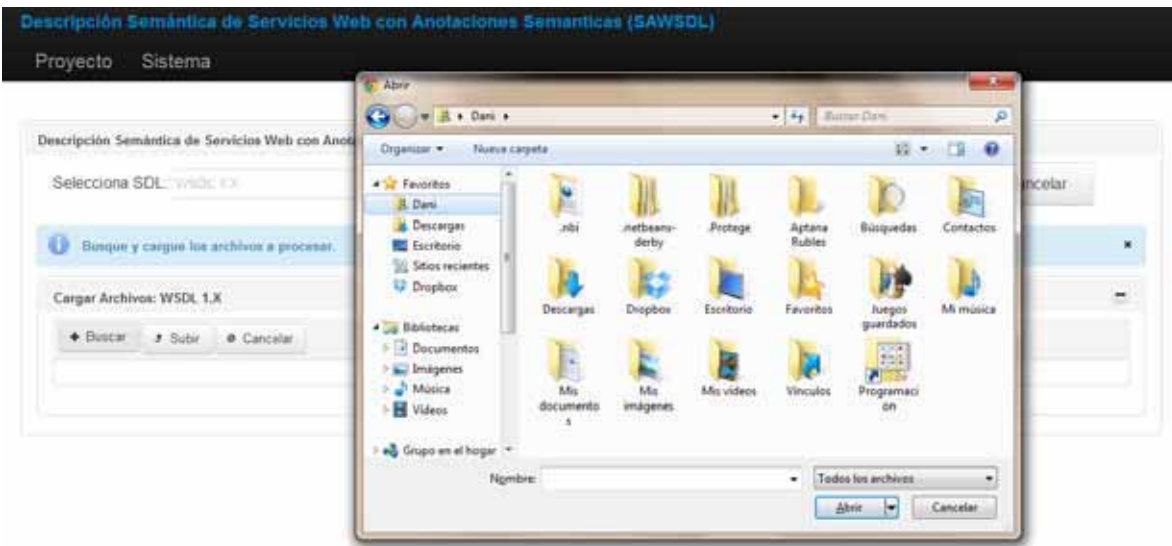


Para seleccionar el tipo de archivo que se subirá, ya sea WSDL 1.x o WSDL 2.0, le damos clic en el selector de SDL.

Para fines demostrativos de este manual, trabajaremos con la opción de WSDL 1.X. Damos clic en el botón Iniciar y se desplegará la siguiente pantalla:



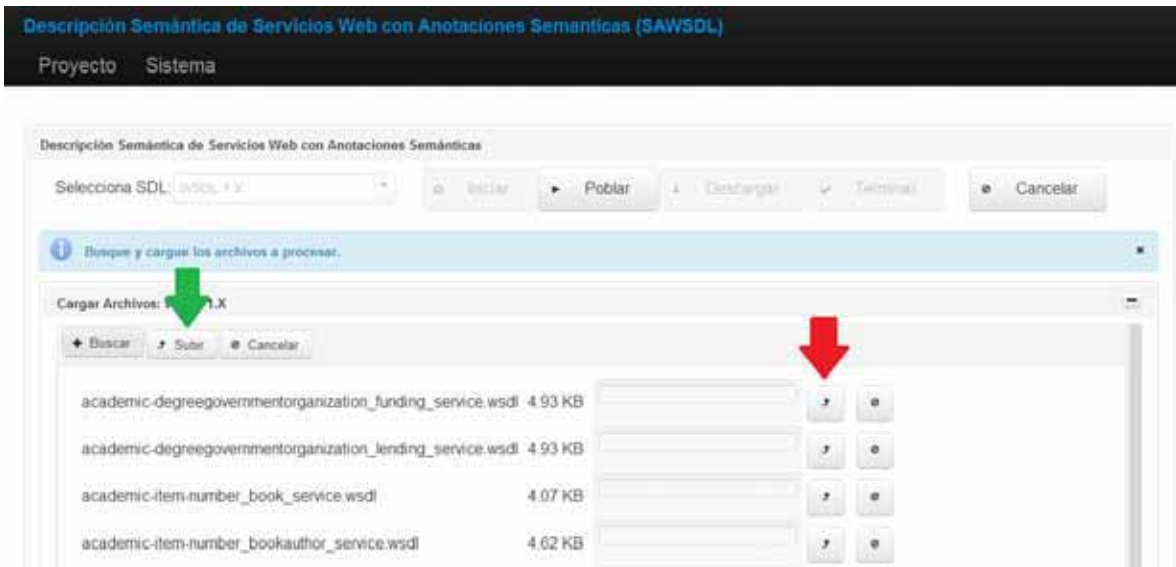
Damos clic en el botón de Buscar y se nos desplegará la ventana para navegar y seleccionar el o los archivos que deseamos subir



Seleccionamos los archivos WSDL 1.1 con los que deseamos trabajar.

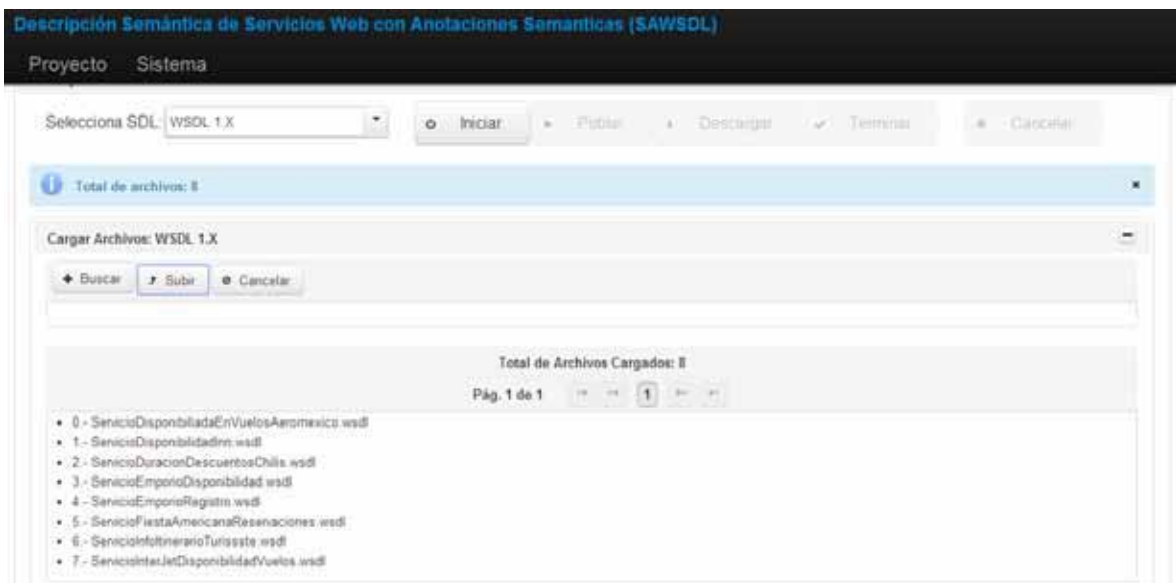
Una vez seleccionados los archivos, el sistema los mostrará con un status de "pendiente", esto con la finalidad de que el usuario pueda revisarlos una vez más antes de subirlos al servidor.

El sistema se vera de la siguiente manera:



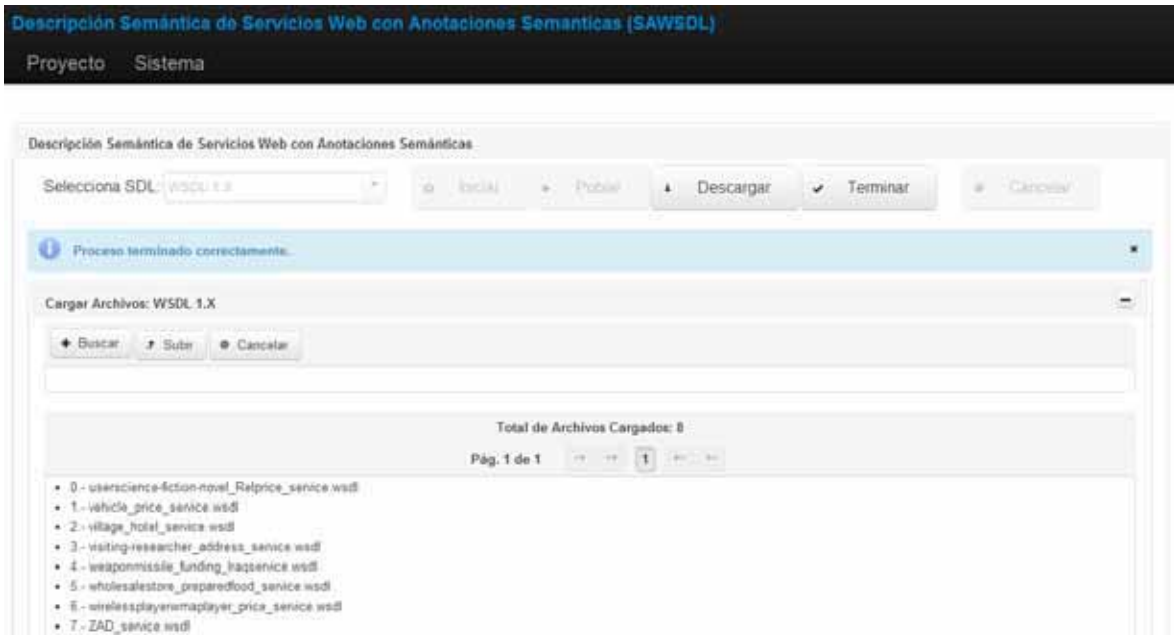
Tenemos la opción de subir uno por uno el archivo dando clic en la flecha roja que se muestra en la imagen de arriba o subir todos los archivos juntos dando clic en el botón que indica la flecha verde.

Si damos clic sobre el botón señalado con la flecha roja el archivo completara el proceso de subida y se agregara a la lista de archivos cargados, si presionamos el botón indicado de la flecha verde se subirán todos y todos los archivos de la lista se agregaran a archivos cargados.

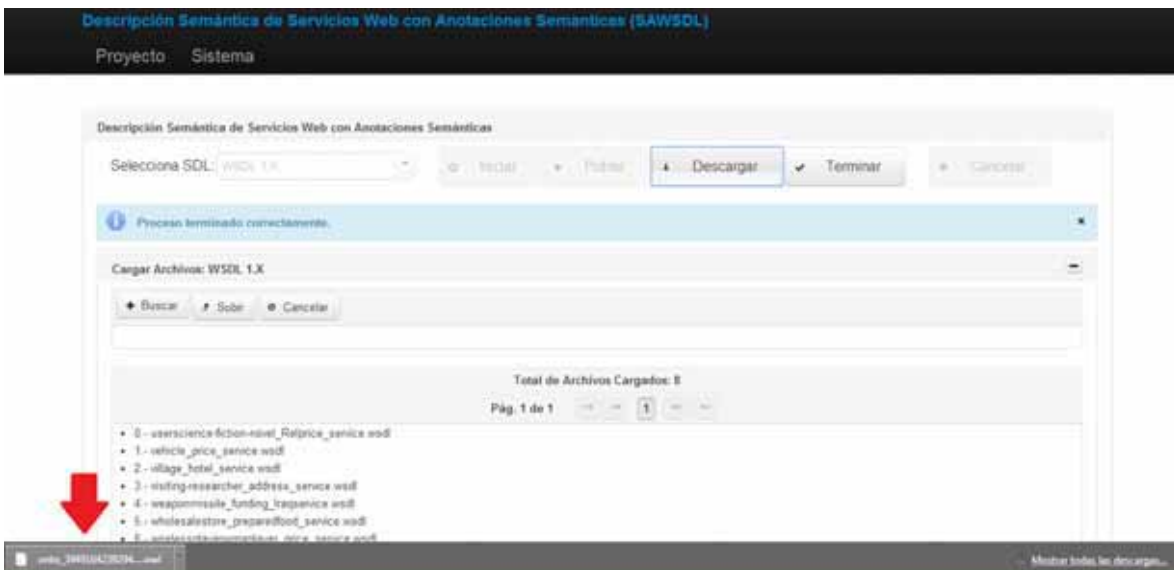


Una vez terminado el proceso de carga, damos clic en el botón Poblar. En caso de que alguno de los archivos cargados no sea WSDL 1.1 marcara un mensaje de error al tratar de usar ese archivo para poblar la ontología.

Al finalizar el poblado de la ontología se mostrará un mensaje avisándonos que el proceso terminó y se activará el botón de Descargar.



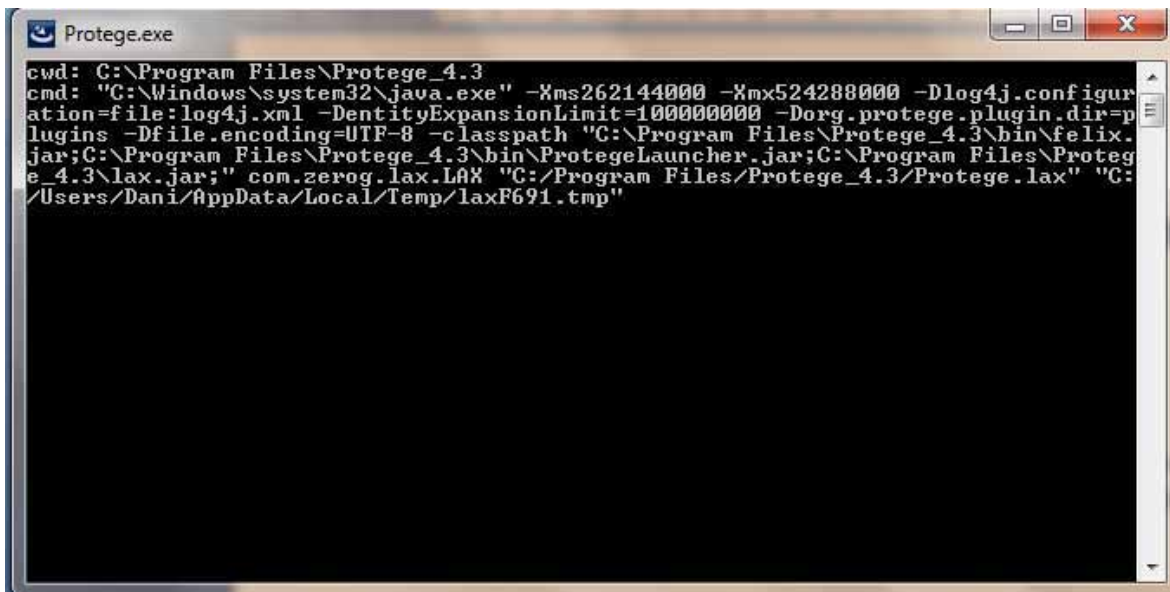
Damos clic en el botón de Descargar y nuestro navegador iniciara la descarga de la ontología.



Para poder tratar el archivo descargado es necesario tener instalado Protégé en cualquiera de sus versiones. Para fines de este manual usaremos la versión 4.3

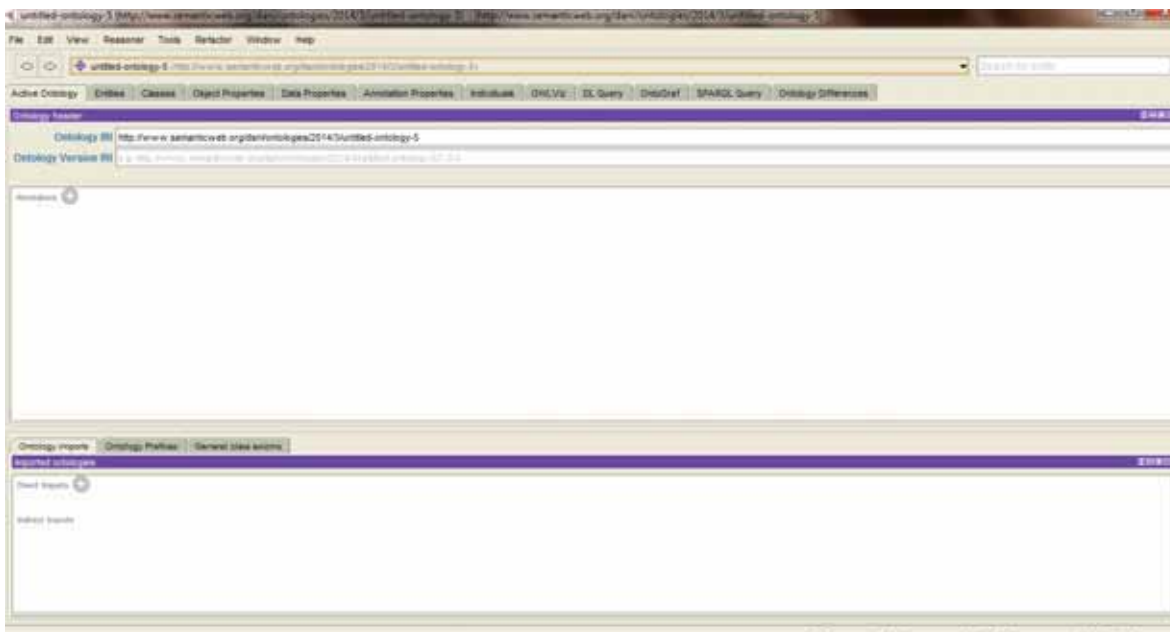
4.- Uso de Protege

El uso de esta herramienta es muy complejo, pero dado que solo exploraremos la ontología descargada será sencillo su uso para este aspecto. Abrimos protege y obtendremos la siguiente ventana:



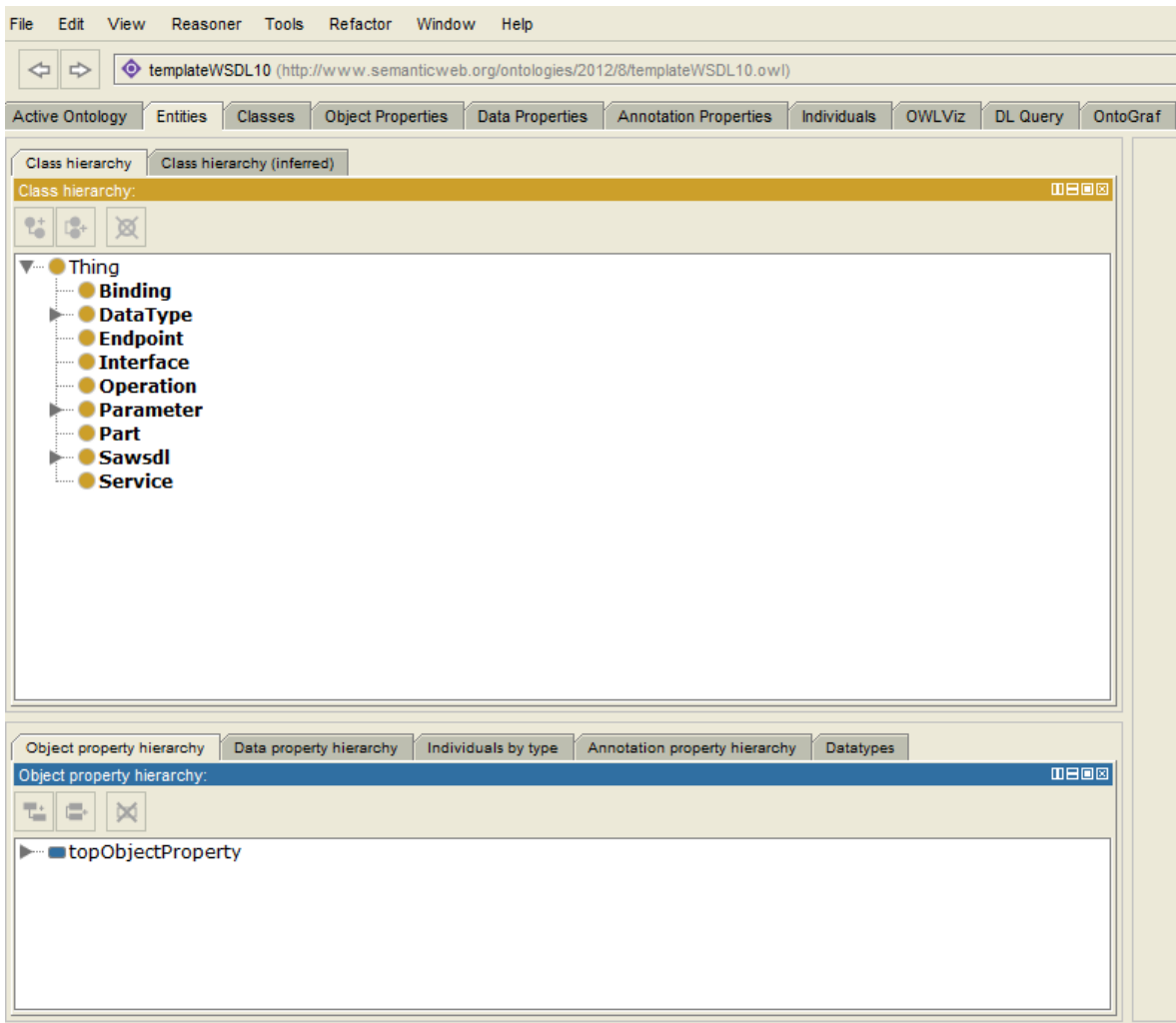
```
Protege.exe
cwd: C:\Program Files\Protege_4.3
cmd: "C:\Windows\system32\java.exe" -Xms262144000 -Xmx524288000 -Dlog4j.configuration=file:log4j.xml -DentityExpansionLimit=1000000000 -Dorg.protege.plugin.dir=plugins -Dfile.encoding=UTF-8 -classpath "C:\Program Files\Protege_4.3\bin\felix.jar;C:\Program Files\Protege_4.3\bin\ProtegeLauncher.jar;C:\Program Files\Protege_4.3\lax.jar;" com.zerog.lax.LAX "C:/Program Files/Protege_4.3/Protege.lax" "C:/Users/Dani/AppData/Local/Temp/laxF691.tmp"
```

Tras unos segundos transcurridos nos lanzara la aplicación que se ve así:

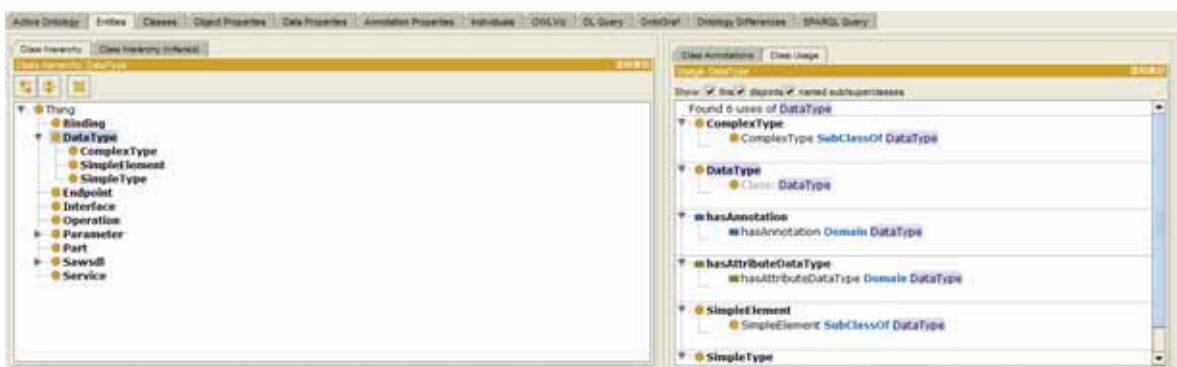


Damos clic en el menú File, Open y buscamos la ontología descargada, damos clic en abrir y nos dirigimos a la pestaña de Entities que se encuentra debajo de los menús principales.

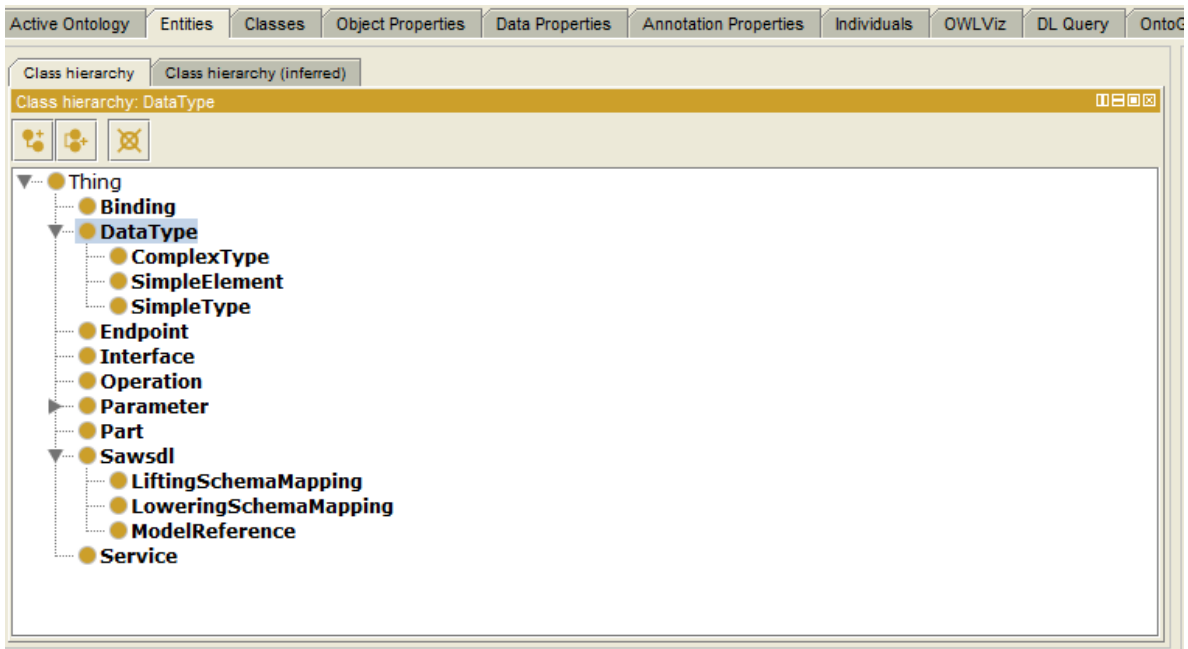
La pestaña Entities nos mostrara el modelo ontológico:



Podemos dar clic sobre cualquiera de los elementos y nos mostrara su descripción



Nos dirigimos al nodo SAWSDL que es el que nos importa, lo desplegamos y podremos ver los tres tipos de anotaciones



Damos clic sobre alguna de las anotaciones y nos mostrara todos los elementos del documento WSDL que la contienen.



Nos indica el tipo de anotación, el valor de la anotación, a que WSDL pertenece y cuál es el origen de consulta para la anotación.

El proceso es idéntico para archivos del tipo WSDL 2.0