

Construcción de ontologías OWL

Héctor Arturo Flórez Fernández¹

Resumen

En la actualidad todas las organizaciones deberían estar a la vanguardia de la tecnología para ser competitivas, por lo cual deben tener la capacidad de ofrecer servicios de acuerdo a los requerimientos de los usuarios y a las condiciones de su entorno. La web actual ha venido presentando varios inconvenientes en cuestión de conceptualización de la información publicada en Internet y es por eso que se pretende evolucionar a lo que se denomina la web semántica. Para la aplicación de la web semántica es indispensable mencionar el elemento principal, que es la ontología. El presente artículo proporciona una descripción completa para la elaboración de ontologías OWL utilizando la herramienta Protégé.

Palabras clave

Ontología, web semántica, servicio web semántico, Protégé, OWL (Ontology Web Language), Racer OWLViz, OntoViz.

¹ Ingeniero Electrónico de la Universidad El Bosque de Bogotá, ingeniero de sistemas de la Universidad El Bosque de Bogotá, maestro de ciencias en Ciencias de la Información y las Comunicaciones de la Universidad Distrital Francisco José de Caldas, docente investigador de la Universidad Konrad Lorenz, docente de la Universidad Distrital Francisco José de Caldas de Bogotá, hectorarturo@yahoo.com.

Abstract

Nowadays, all the organizations should be at the vanguard of technology to be competitive, thus they must have the capacity to offer services according to the users' requirements and the conditions of their surroundings. The present web has presented several disadvantages regarding the conceptualization of the information published in the Internet, and that is why it is pretended to evolve to a denominated semantic web. For the application of the semantic web it is necessary to mention the main element: ontology. The present article provides a complete description for the elaboration of OWL ontologies using the development tool Protégé.

Keywords

Ontology, semantic web, semantic web service, Protégé, OWL (Ontology Web Language), Racer, OWLViz, OntoViz.

1. Introducción

La web semántica se prepara para dar solución a los problemas de conceptualización de información de la web actual. El solo uso de los servicios web trae grandes ventajas y la capacidad de poder combinar varios servicios en procesos web es uno de los aspectos clave para maximizar las potencialidades individuales de las organizaciones a través de la integración de datos y de aplicaciones, lo cual es algo que en general se conoce como composición o colaboración de servicios.

Sin embargo el éxito de los procesos se basa en la conceptualización de la información, que es donde entra a participar la web semántica, donde se intenta complementar el conjunto actual de protocolos de los servicios agregando una descripción semántica tanto a los servicios como a las diferentes interacciones entre ellos, para que éstos puedan ser objeto de

análisis, razonamiento y manipulación automática por parte de agentes software.

El elemento principal para la elaboración de un sistema basado en web semántica es la ontología, la cual provee todos los elementos de lógica que permiten conceptualizar el negocio y obtener información dadas las bases de conocimiento que el sistema construye.

2. Web semántica

La aparición de Internet se sitúa en 1989, cuando Tim Berners-Lee presentó su proyecto de World Wide Web en el CERN (Suiza), con las características esenciales que perduran en estos días. Berners-Lee completó en 1990 el primer servidor web y el primer cliente (Berners-Lee, 1996), y un año más tarde publicó el primer borrador de las especificaciones de HTML y http (Berners-Lee, 1997).

Desde que se lanzó al mercado el primer navegador de dominio público de Internet en 1993, se marcó el momento en que la web se dio a conocer al mundo, extendiéndose primero en universidades y laboratorios y, en cuestión de meses, al público en general, iniciando el que sería su vertiginoso crecimiento (Castells, 2003). A partir de este momento el crecimiento de la información que se manipula en la web se ha multiplicado y ha generado la necesidad de desarrollar sistemas capaces de buscar información que sea útil para los usuarios, pero la mayoría de estos sistemas se basan en búsquedas por palabras clave, lo que puede generar respuestas que no corresponden a las necesidades de los usuarios.

Organizaciones como el w3c (Consortio World Wide Web) han destinado investigaciones y recursos para desarrollar tecnologías y especificaciones tendientes a mejorar las técnicas de búsqueda, la integración de servicios y otros recursos disponibles en la web.

La integración de la semántica en la web ha permitido agregar notaciones semánticas que harán que los sistemas puedan interactuar automáticamente, lo cual beneficiará bastante a los usuarios. Tim Berners-Lee tenía una visión sobre la web que conocemos. Él pensó en una red de recursos que nos permitiera programar agentes que navegaran la infinitud de sitios pudiendo obtener la información que necesitamos sin tener que indicarle de dónde obtenerla o qué significado debe tener cada recurso, transformando finalmente esa información a un formato que sea fácilmente entendible por nosotros. Esa web, que aún se encuentra en una fase de desarrollo, es lo que se conoce como la web semántica (Berners-Lee, Hendler y Lassila, 2001).

La web semántica se basa en dos conceptos fundamentales (Castells, P., 2003):

- La descripción del significado que tienen los contenidos en la web.
- La manipulación automática de estos significados.

La descripción del significado requiere conceptos ligados a:

- La semántica, entendida como significado procesable por máquinas.
- Los metadatos, como contenedores de información semántica sobre los datos.
- Las ontologías, conjunto de términos y relaciones entre ellos que describen un dominio de aplicación concreto.

3. Ontologías

Una ontología es una jerarquía de conceptos con atributos y relaciones que tiene una terminología consensuada para definir redes semánticas de unidades de información interrelacionadas. Una ontología proporciona un vocabulario de clases y relaciones para describir un dominio.

Las ontologías son usadas para capturar conocimiento sobre algún dominio de interés. Una ontología describe los conceptos del dominio así como las relaciones que se dan entre éstos. Existen diferentes lenguajes para ontologías que proveen diversas facilidades; el más reciente estándar dado por la w3c es el owl (Guarino, 1998).

Una ontología debe ser capaz de proveer un buen entendimiento del dominio que representa. Esto incluye relaciones de términos y conceptos, sus definiciones o significados, sus relaciones con cada uno de ellos y las características del dominio. Esto ayuda en la solución de las diferencias entre los modelos y el dominio usado por las fuentes disponibles, lo que se hace mapeando los datos disponibles de todos los recursos para todos los

dominios desde el modelo local usado por la fuente para el modelo especificado por medio de la ontología (Klein, et al., 2002, pp. 95-135).

Las ontologías proveen una comprensión compartida y concensuada del conocimiento de un dominio que puede ser comunicada entre personas y sistemas heterogéneos. Fueron desarrolladas en el área de Inteligencia Artificial (IA) para facilitar el intercambio y reuso del conocimiento (Guarino, 1998).

En referencia a las ontologías, se describe el modelo ontológico OIL, concebido para la web. Algunos problemas surgen cuando se trata con grandes cantidades de información semiestructuradas. Los actuales buscadores basados en palabras clave suelen devolver información irrelevante que usa una cierta palabra con un significado diferente del que se pretende en la búsqueda, y pierden información cuando no reconocen palabras diferentes pero con el mismo significado que la buscada.

Actualmente se requiere lectura humana para extraer información relevante de un origen, porque los agentes automáticos no tienen el sentido común requerido para reconocer dicha información en representación textual. Mantener orígenes textuales débilmente estructurados representa una tarea difícil y consumidora de tiempo cuando tales orígenes aumentan considerablemente de tamaño. Mantener esas colecciones consistentes y al día requiere de representación interpretable por computador, de semántica que ayude a detectar anomalías automáticamente. La utilidad de sitios web adaptativos, que permitan su reconfiguración dinámica de acuerdo al perfil del usuario u otros aspectos relevantes, requiere una representación computable de la semántica involucrada.

3.1. OWL (Ontology Web Language)

OWL es uno de los lenguajes recomendados por la organización W3C que incorpora lecciones aprendidas en el diseño y aplicación del lenguaje DAML+OIL. Al igual que los anteriores lenguajes es usado cuando la información contenida en los documentos necesita ser procesada por las aplicaciones. Este lenguaje provee un mayor conjunto de primitivas para representar el significado de los elementos y sus relaciones con otros elementos en una ontología.

Ontology Web Language es un lenguaje que permite definir las ontologías. OWL está especificado en 6 documentos del W3C:

- OWL Overview.
- OWL Semantics and Abstract Syntax.
- OWL Use Cases and Requirements.
- OWL Test Cases.
- OWL Guide.
- OWL Reference.

OWL se puede formular en RDF, por lo que se suele considerar una extensión de éste. Además incluye toda la capacidad expresiva de RDF y la extiende con la posibilidad de utilizar expresiones lógicas. Permite la creación de clases y permite atribuir ciertas propiedades a las relaciones, como cardinalidad, simetría, transitividad o relaciones inversas (Knublauch, 2005).

OWL tiene los siguientes niveles:

- OWL Lite: Tiene restricciones simples (cardinalidad sólo 0 o 1), facilita compatibilidad con otros modelos/paradigmas, facilita desarrollo de herramientas de autor. Tiene un razonamiento eficiente, facilita desarrollo de herramientas, compatibilidad con otros modelos. Es el menos

expresivo, se usa en situaciones donde sólo existe una simple jerarquía de clases y donde se necesitan pocas restricciones.

- **OWL DL (Description Logics):** Es de término medio en cuanto a la expresividad; está basado en lógica descriptiva, la cual es una lógica basada en primer orden; es favorable para el razonamiento automático, para la clasificación de jerarquías y para detectar las inconsistencias en las ontologías. Limita la expresividad intentando conseguir decidibilidad y máxima expresividad manteniendo completitud y computabilidad. Usa restricciones para asegurar la existencia de un procedimiento de razonamiento decidible. No puede haber restricciones de cardinalidad (locales ni globales) en propiedades transitivas, ni sus inversas, ni sus superpropiedades. No pueden faltar definiciones. Los axiomas de desigualdad sólo se pueden aplicar a instancias con nombre.
- **OWL Full:** Es el más expresivo, se usa en situaciones donde es más importante una extrema expresividad que la capacidad de razonamiento. Realiza unión de sintaxis OWL y RDF (sin restricciones).

Los niveles de OWL frente a RDF explican que OWL Lite está “contenido” en OWL DL y éste está contenido en OWL Full. OWL se expresa con primitivas RDF, de manera que todo documento válido OWL es un documento válido RDF. RDF está “contenido” en OWL Full (este último es una extensión de RDF) de manera que un documento válido RDF es un documento válido OWL Full.

RDF no está “contenido” ni en OWL DL ni OWL Lite, estos dos son extensiones de una versión restringida de RDF, de modo que no todo documento válido RDF es un documento válido en OWL DL u OWL Lite.

4. Protégé

Protégé es una herramienta open source desarrollada en Stanford Medical Informatics. Como muchas otras herramientas de modelamiento, Protégé se encuentra separada entre una parte de modelo y una parte de vista. El modelo de Protégé es el mecanismo de representación interna de ontologías y bases de conocimiento. Los componentes de la vista de Protégé proveen al usuario una interfase para visualizar y manipular el modelo (“Getting Started With Protégé”, 2003).

El modelo de Protégé está basado en un simple metamodelo comparable con sistemas orientados a objetos y sistemas basados en marcos. Éstos básicamente pueden representar ontologías que consisten en clases, propiedades, características de propiedades e instancias. La ventaja de Protégé es que su metamodelo es una ontología Protégé. La estructura del metamodelo de Protégé habilita fácilmente la extensión y adaptación a otras representaciones.

Usando las vistas de la interfase de usuario de Protégé, los diseñadores de ontologías básicamente crean clases y les asignan propiedades. Usando la función *resulting ontologies*, Protégé es capaz de generar automáticamente interfaces de usuario que soporten la creación de instancias. En cada clase de la ontología el sistema crea un formulario con componentes de edición para cada propiedad de la clase.

Protégé actualmente puede ser usado para cargar, editar y salvar ontologías en varios formatos como CLIPS, RDF, XML, UML, OWL. Las ontologías tienen un papel importante en las aplicaciones de web semántica. Las extensiones de Protégé pueden beneficiar desde los servicios generales proveídos de la plataforma núcleo, hasta un mecanismo de

evento, las capacidades de deshacer y los mecanismos de plugin.

Con el plugin OWL de Protégé se puede permitir que mucha gente edite la misma ontología al mismo tiempo. Protégé también provee una base de datos altamente escalable, lo que permite que los usuarios creen ontologías con muchas clases (Knublauch, 2005).

El plugin OWL es una extensión compleja que puede ser usada para editar archivos OWL y bases de datos. El plugin incluye una colección de características para los componentes denominados *widgets* y provee acceso a servicios basado en clasificación, chequeo consistente y pruebas.

Protégé OWL es una herramienta que hace posible la descripción de conceptos y provee otras facilidades como por ejemplo: tiene un enriquecido conjunto de operadores (*and*, *or* y la negación); está basado sobre un modelo lógico que garantiza que los conceptos estén bien descritos; se pueden describir conceptos complejos, y el modelo lógico permite usar un razonador que chequea la consistencia descrita en la ontología garantizando que la jerarquía esté correcta.

5. Construcción de ontología OWL

Para crear una ontología OWL mediante Protégé es necesario adquirir el plugin OWL, el cual viene incluido en las últimas versiones de Protégé.

Una vez iniciado Protégé, es necesario crear una ontología OWL. Para ilustrar este documento, se plantea una ontología que conceptualice un modelo universitario en donde se contemplan elementos como estudiantes, docentes, cursos, programas, títulos y proyectos, entre otros.

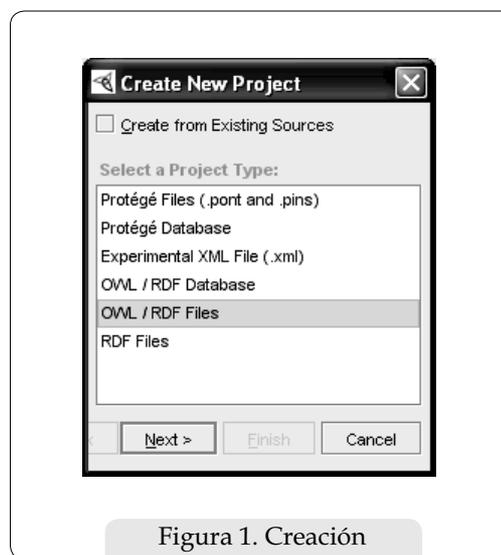


Figura 1. Creación de ontología OWL.

Una ontología OWL está compuesta de individuos, propiedades y clases. Los individuos representan objetos del dominio de interés y son también conocidos como instancias. Las propiedades son relaciones binarias sobre los individuos y pueden ser inversas, transitivas o simétricas. Las clases OWL se entienden como conjuntos que contienen individuos y pueden ser organizadas dentro de una jerarquía de clases y subclases conocida como taxonomía. Las clases también son conocidas como conceptos, pues son una representación concreta de éstos (Horridge, 2004).

5.1. Clases

Al iniciar una ontología OWL existe una clase inicial llamada "owl:Thing" que contendrá la ontología. Para crear una clase se debe usar el ícono correspondiente en la ventana de explorador de clases.

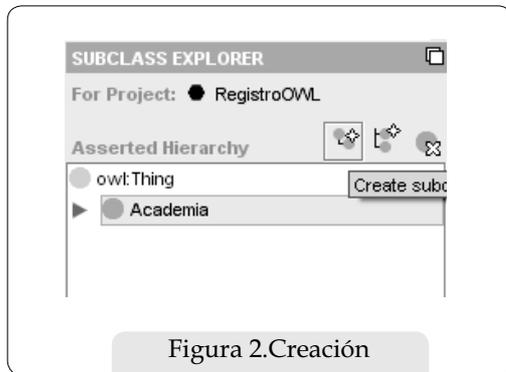


Figura 2. Creación de clase.

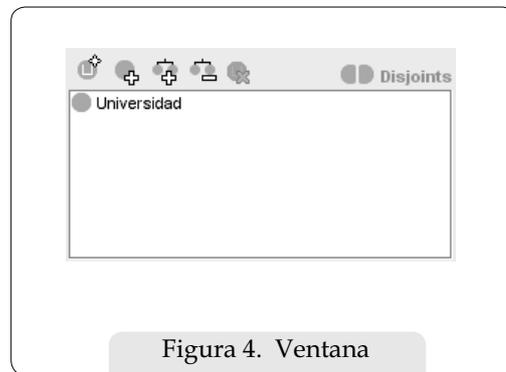


Figura 4. Ventana de clases disjuntas.

Clases disjuntas

Después de adicionar varias clases en la jerarquía de la ontología, se requiere establecer clases disjuntas, que indican que un objeto o individuo no puede ser instancia de más de una de estas clases que se establecieron disjuntas.

Para especificar que ciertas clases son disjuntas es necesario usar la ventana "Disjoints". Por ejemplo, si se tiene la jerarquía mostrada en la siguiente figura, en donde se presenta una clase llamada "Persona" y otra llamada "Universidad", es necesario especificar que son distintas. Para ello, se adiciona la clase correspondiente por la ventana mencionada.

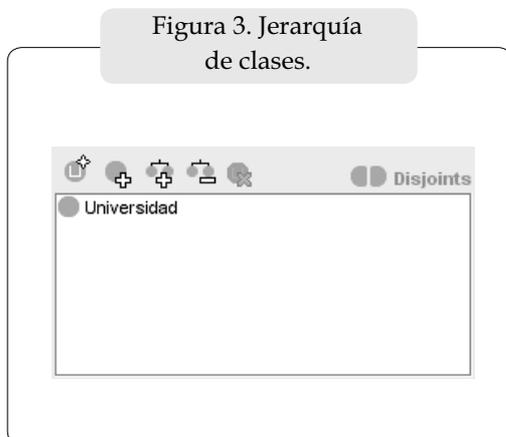


Figura 3. Jerarquía de clases.

5.2 Propiedades OWL

Las propiedades OWL representan relaciones entre dos objetos o individuos. Existen dos tipos de propiedades en OWL: "ObjectProperties", que permite relacionar un individuo con otro, y "DatatypeProperties", que relaciona un individuo con un XML Schema Datatype value o un literal RDF (Horridge, 2004).

Existe una ventana que permite adicionar propiedades de los dos tipos mencionados. Los elementos que debe tener un ObjectProperty son nombre, dominio y rango, donde el dominio hace referencia a la clase o clases iniciales y el rango a la clase o clases finales que relaciona dicha propiedad.

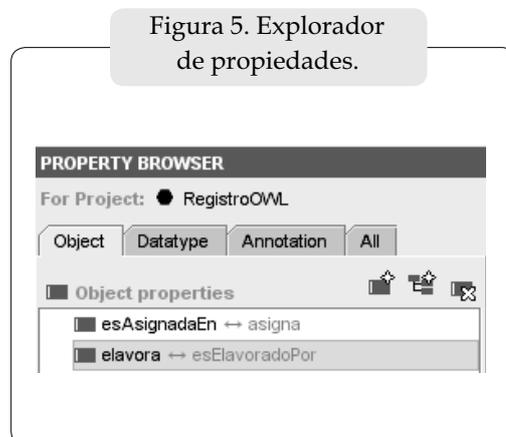


Figura 5. Explorador de propiedades.

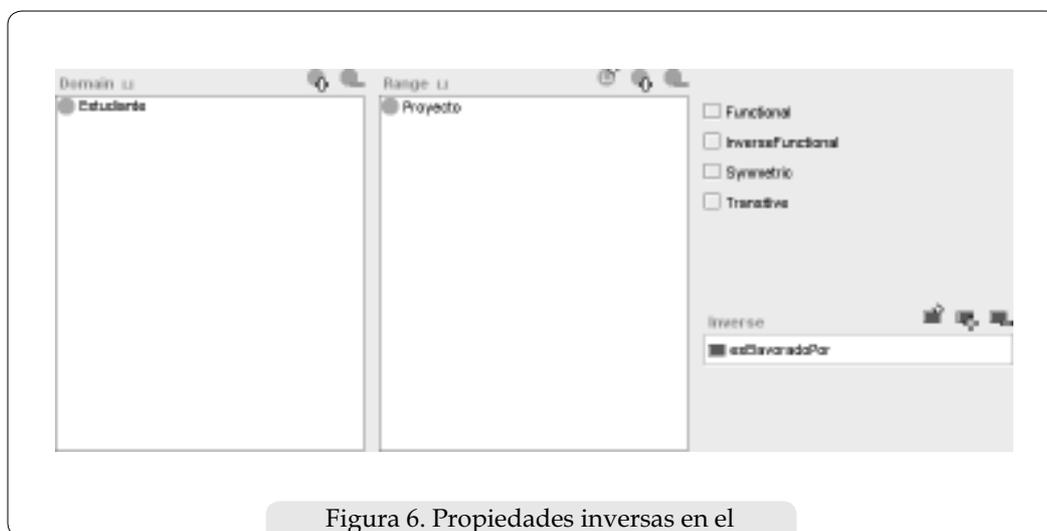


Figura 6. Propiedades inversas en el explorador de propiedades.

Propiedades inversas

Cada ObjectProperty debe tener su correspondiente propiedad inversa. Si una propiedad enlaza un objeto A con otro B, entonces la propiedad inversa enlaza el objeto B con el A. La propiedad inversa puede ser creada con el campo correspondiente en el explorador de propiedades mostrado en la figura 6.

Características de propiedades owl

OWL permite que el significado de las propiedades sea enriquecido con las características de las propiedades, que son:

- **Propiedades funcionales:** Si una propiedad es funcional dado un objeto, puede estar a lo sumo un objeto relacionado con otro mediante esta propiedad. Por ejemplo, si se tienen tres objetos que son A, B y C y se tiene una propiedad funcional tienePadre, entonces se podrían asociar los objetos A y B por medio de la propiedad y daría como resultado A tienePadre B. Igualmente se podrían asociar los objetos A y C por medio de la propiedad

y daría como resultado A tienePadre C. Como tienePadre es propiedad funcional, se concluye que B y C son el mismo objeto.

- **Propiedades funcionales inversas:** Si una propiedad es funcional inversa, indica que puede estar a lo sumo un objeto relacionado con otro mediante esta propiedad de forma inversa a la propiedad funcional original. Por ejemplo, si se tienen tres objetos que son A, B y C y se tiene una propiedad funcional esPadreDe, entonces se podría asociar el objeto B y A por medio de la propiedad y daría como resultado B esPadreDe A. Igualmente se podrían asociar los objetos C y A por medio de la propiedad y daría como resultado C esPadreDe A. Como esPadreDe es propiedad funcional inversa, se concluye que B y C son el mismo objeto.
- **Propiedades transitivas:** Si una propiedad es transitiva y relaciona dos objetos A y B, y además hay una propiedad que relaciona al objeto B con otro C, entonces se puede inferir que el objeto A está relacionado con el objeto C mediante la propiedad transitiva.

- Propiedades simétricas: Si una propiedad P es simétrica y la propiedad relaciona a los objetos A y B , entonces el objeto B es relacionado por medio de la propiedad P con el objeto A .

Dominios y rangos

Toda propiedad de objeto o "ObjectProperty" debe tener un dominio y un rango especificados; las propiedades enlazan objetos del dominio a objetos del rango, donde el dominio y el rango son clases definidas en la ontología.

5.3 Definición y descripción de clases

Una vez creadas varias propiedades, se pueden utilizar para definir y describir el comportamiento de las clases ("Getting Started with Protégé", 2003).

Restricciones de propiedades

Las propiedades son utilizadas para crear restricciones en las clases en una ontología OWL. Usualmente el nombre de la propiedad debería sugerir las restricciones impuestas a los objetos de la clase. Las restricciones OWL se presentan en las siguientes tres categorías:

- Restricciones de cuantificación.
- Restricciones de cardinalidad.
- Restricciones de valor.

Las restricciones de cuantificación se componen de los siguientes elementos:

- Cuantificador existencial (\exists), el cual permite indicar la existencia de al menos un objeto.
- Cuantificador universal (\forall), el cual permite indicar la existencia de todos los objetos.

Dada la ontología que se muestra en la siguiente figura, que representa la conceptualización del modelo universitario, se pueden plantear diferentes restricciones haciendo uso de propiedades previamente definidas y aplicando cuantificadores existenciales.

Figura 7. Ontología de modelo universitario.

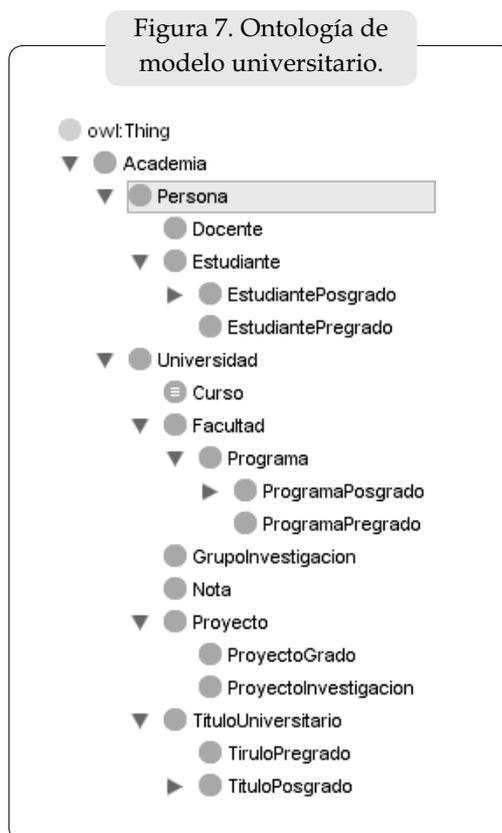
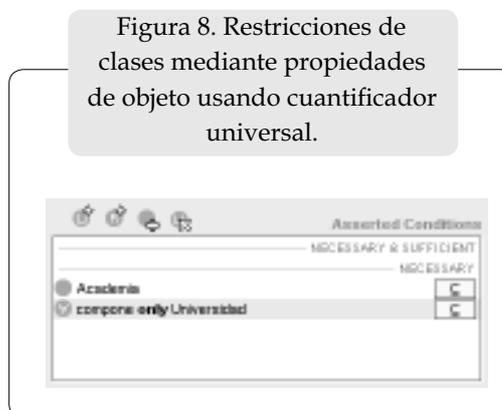
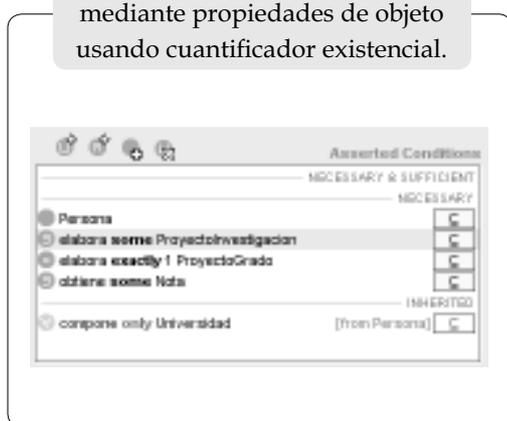


Figura 8. Restricciones de clases mediante propiedades de objeto usando cuantificador universal.



En la figura anterior, dado que tiene seleccionada la clase “Persona”, la restricción indica que todas las personas componen la universidad.

Figura 9. Restricciones de clases mediante propiedades de objeto usando cuantificador existencial.



En la figura anterior, dado que tiene seleccionada la clase “Estudiante”, las restricciones indican que:

- Cada estudiante elabora al menos un proyecto de investigación.
- Cada estudiante elabora exactamente un proyecto de grado.
- Cada estudiante obtiene al menos una nota, debido a que compone al menos un curso.
- Finalmente se encuentra una restricción heredada de la clase “Persona” que indica que todos los estudiantes componen la universidad.

5.4. Razonamiento de la ontología

Para realizar el razonamiento de una ontología es necesario adquirir una herramienta que actúe como razonador para Protégé. Existe una herramienta llamada Racer que se puede conectar con Protégé para el análisis y razonamiento de las ontologías.

Razonador Racer

Para iniciar Racer se ejecuta el programa RacerPro. Aparece luego una ventana (figura 10) que indica que se está ejecutando para una dirección IP y un puerto previamente configurados. (Por defecto se encuentra en dirección IP: localhost y Puerto 8088.)

Figura 10. Inicio del razonador RacerPro.



Es necesario configurar en Protégé la dirección en que se encuentra ejecutando el razonador RacerPro. Para ello se ingresa a la ventana OWL Preferencias y se establece la dirección configurada en RacerPro.

Figura 11. Configuración del razonador en Protégé

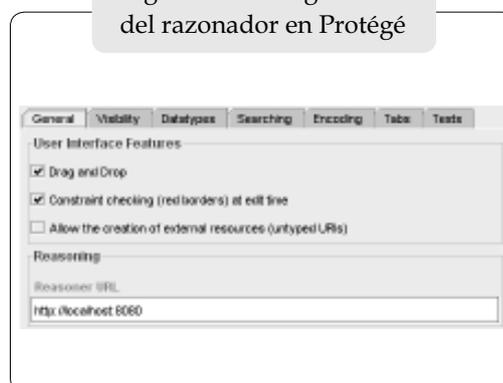




Figura 12. Barra de herramientas de Protégé.

Racer se utiliza para tres actividades principales que son:

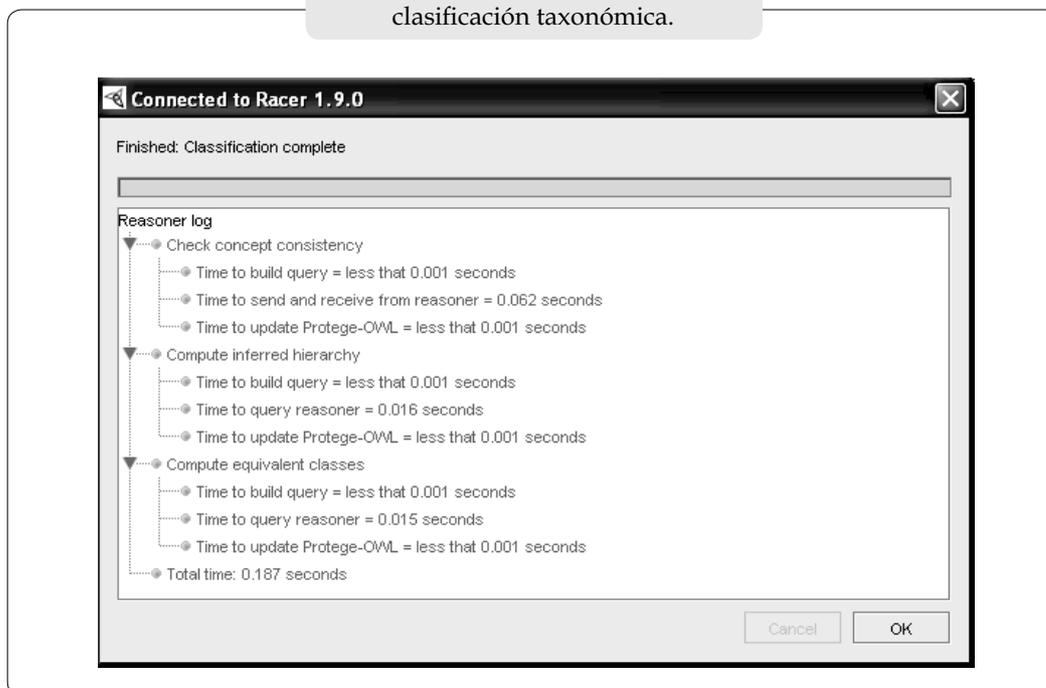
- Verificación de consistencia: Permite controlar que exista consistencia entre las propiedades de objetos y restricciones asignadas a las clases.
- Clasificación taxonómica: Encuentra la jerarquía de inferencia de la ontología basada en la jerarquía de aserción, la cual es la clasificación que se obtiene en la construcción de la ontología.

- Computación de tipos inferidos: Permite computar los tipos de objetos dentro de la ontología.

Estas funciones se ejecutan mediante íconos en la barra de herramientas de Protégé, como se ve en la figura 12.

El ícono central es el que permite hacer la clasificación taxonómica que es la operación más importante de Racer. Al realizar esta operación, Protégé genera una jerarquía de inferencia en la ontología.

Figura 13. Resultados de ejecución de clasificación taxonómica.



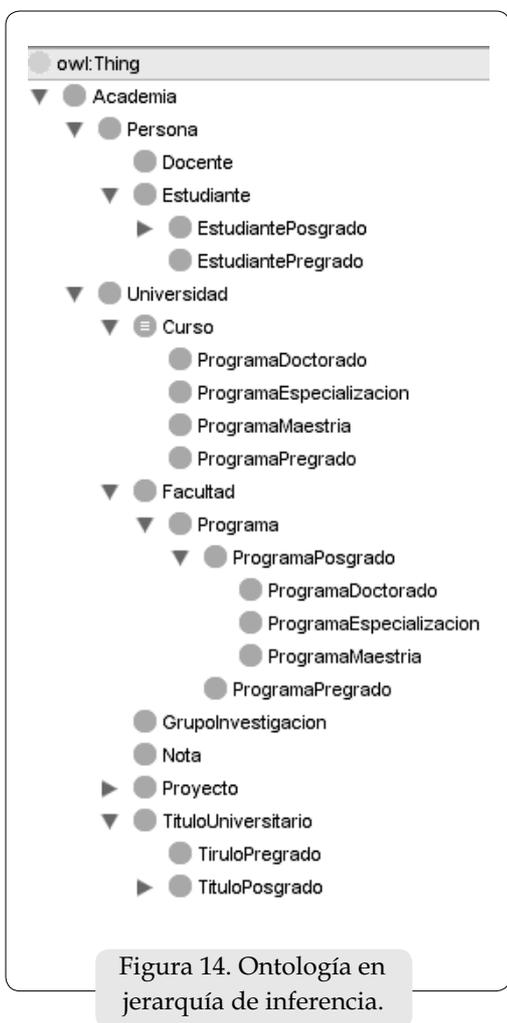


Figura 14. Ontología en jerarquía de inferencia.

La ontología en la jerarquía de inferencia generada por la clasificación taxonómica difiere de la ontología en jerarquía de inserción presentada en la figura 7 en la adición de subclases en la clase "Curso" como resultado de la inferencia basada en las propiedades de objeto y restricciones aplicadas a las clases de la ontología.

5.5 owlviz

owlviz es una herramienta de Protégé que se instala mediante un plugin. Para hacer uso de este plugin es necesario instalar una

herramienta que permite realizar componentes gráficos de la ontología denominada Graphviz. Una vez instalados owlviz y Graphviz, se genera una pestaña en el entorno de Protégé denominada owlviz. En ella existe una barra de herramientas que provee diferentes funcionalidades al plugin. En esta barra se encuentra un ícono denominado options, al cual es necesario acceder para indicar la ruta de la aplicación Graphviz (Knu- blauch, 2005).

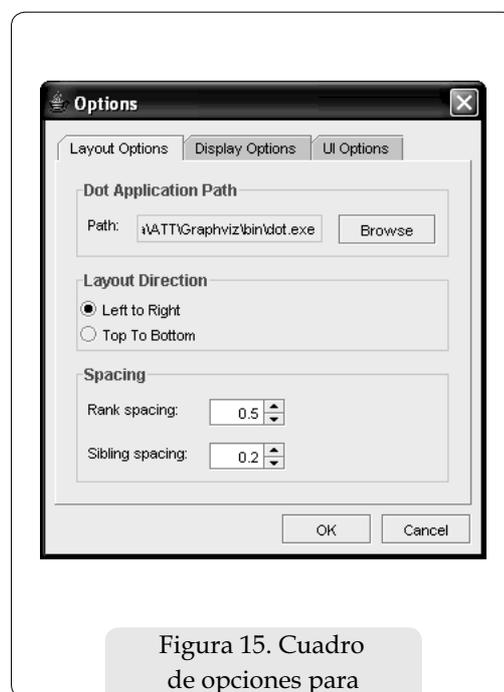


Figura 15. Cuadro de opciones para la configuración de Graphviz.

Una vez se encuentren todos los elementos instalados, se puede generar la ontología en forma grafica con la jerarquía de aserción o la jerarquía de inferencia.

de nuevas tecnologías de información como la inteligencia artificial, la minería de datos, las bases de conocimiento, etc.

Una vez elaborada la ontología haciendo uso de herramientas como Protégé, se procede a generar la taxonomía que dicha ontología representa con ayuda de la herramienta RacerPro. Esta taxonomía se basa en propiedades OWL planteadas en la ontología diseñada. Este proceso permite observar los resultados en cuanto a las inferencias generadas por el modelo que describe el dominio del problema. Partiendo de dichos resultados, se procede a realizar la implementación del modelo descrito con ayuda de una plataforma tecnológica que permita la construcción de servicios web. Estos servicios, al ser construidos bajo un modelo semántico, se convierten en servicios web semánticos.

Sin embargo, el proceso de implantación de dicho modelo conlleva la necesidad de hacer uso de tecnologías y paradigmas existentes, con menor nivel de abstracción que las ontologías. Al realizar este paso, que se vuelve necesario en el proceso de ejecución de un proyecto, se puede pensar que se simplifica la potencia que los modelos de web semántica ofrecen.

Este enfoque puede llevar a la siguiente pregunta: ¿Es posible, con las técnicas utilizadas en otros paradigmas, lograr la semántica proporcionada por las ontologías?

Para que las aplicaciones implementen toda la semántica propuesta por una ontología puede ser necesaria la construcción de una plataforma de interpretación semántica de dicha ontología. En otras palabras, se hace necesario generar un lenguaje de muy alto nivel que permita codificar de manera directa la semántica proporcionada del dominio representado. Considerando este punto

de vista, se podría pensar que las características que ofrece OWL no son suficientes para la implantación de sistemas web semánticos que proporcionen la abstracción completa de la lógica del negocio del modelo usando los lenguajes de desarrollo de software existentes en la actualidad.

7. Conclusiones

- La web semántica se puede considerar una extensión de la web de la actualidad que permite manejar nuevas tecnologías para mejorar el uso y acceso de los recursos de la web haciendo uso de conceptos como OWL y WSDL (Web Service Definition Language), los cuales soportan un lenguaje ontológico para definir y clasificar conceptos en un dominio. Éstos se pueden implementar mediante servicios web.
- Una ontología es una jerarquía de conceptos con atributos y relaciones que crea una clasificación para definir la semántica de unidades de información interrelacionadas. Las ontologías proveen una comprensión de un dominio que puede ser comunicada entre personas y sistemas heterogéneos.
- Protégé es una herramienta de desarrollo de ontologías que permite la construcción de ontologías OWL teniendo en cuenta todas las características del dominio. Por medio de Protégé la elaboración de propiedades y restricciones permite generar inferencia sobre el dominio, generando conceptualización y conocimiento del mismo.
- Racer permite realizar de forma rápida la verificación de una ontología teniendo en cuenta elementos de consistencia, además de realizar la clasificación taxonómica denominada jerarquía de inferencia, derivada del modelo construido en la ontología denominado jerarquía de

aserción. Con base en estos elementos se hace comprensible la importancia de la ontología dentro del concepto de la web semántica.

- owlviz y OntoViz son herramientas de Protégé que permiten realizar representaciones gráficas de la ontología ya sea en modelo de aserción o de inferencia.

Bibliografía

Berners-Lee, T. (1996). The World Wide Web: Past, Present and Future. Extraído del World Wide Web: <http://www.w3.org/People/Berners-Lee/1996/ppf.html>

– (1997). The World Wide Web: A very short personal history. Extraído del World Wide Web: <http://www.w3.org/People/Berners-Lee/ShortHistory.html>

–, Hendler, J. y Lassila, O. (2001). The Semantic Web: A New Form of Web Content that is Meaningful to Computers Will Unleash a Revolution of New Possibilities. Extraído del World Wide Web: http://www.scientificamerican.com/print_version.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21

Castells, P. (2003). La web semántica. Madrid: Escuela Politécnica Superior, Universidad Autónoma de Madrid. Extraído del World Wide Web: <http://www.ii.uam.es/~castells/publications/castells-uclm03.pdf>

Dinos, J. (2002). Arquitectura de un sistema basado en agentes para la recuperación de metadatos RDF con base a una ontología de documentos. Universidad de

Puerto Rico. Extraído del World Wide Web: grad.uprm.edu/tesis/dinosrojas.pdf.

Getting Started with Protégé (2003). Extraído del World Wide Web: <http://protege.stanford.edu/doc/tutorial/index.html>

Guarino, N. (1998). Formal Ontology and Information Systems. In the Proceedings of Formal Ontology in Information Systems. Extraído del World Wide Web: www.loa-cnr.it/Papers/FOIS98.pdf

Horridge, M. (2004). *A Practical Guide to Build owl Ontologies*.

Klein, M., et al. (2002). Ontologies and Schema Languages on the Web, en *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, pp. 95 - 135.

Knublauch, H. (2005). The Protégé owl Plugin: An Open Development Environment for Semantic Web Applications. Stanford University. Extraído del World Wide Web: www.scf.usc.edu/~csci586/ppt-2005/bhavin.ppt.

Infografía

w3c (Consortio World Wide Web): <http://www.w3.org/>

w3c España: <http://www.w3c.es>

Organización Web Semántica: <http://www.semanticweb.org/>

Organización XML: http://www.xml.org/xml/resources_cover.shtml

Sitio web de Protégé: protege.stanford.edu/plugins/owl/

Sitio web de Racer: <http://www.racer-systems.com/products/tools/protege.phtml>