

Tipo de artículo: Artículo original
Temática: Sistemas de información y administración de datos.
Recibido: 25/05/2021 | Aceptado: 28/06/2021

Administrando Ontologías Grandes y Altamente Expresivas Mediante OWLAPI^a

Managing large and highly expressive ontologies through OWLAPI

Manuel Enrique Puebla Martínez ^{1*} <https://orcid.org/0000-0001-6008-0370>

Alfredo Simón Cuevas ² <https://orcid.org/0000-0002-6776-9434>

José Manuel Perea Ortega ³ <https://orcid.org/0000-0002-7929-3963>

Yanet Espinal Martín ⁴ <https://orcid.org/0000-0001-9186-8304>

¹ Universidad de las Ciencias Informáticas. Apartamento 23-106. Carretera a San Antonio de los Baños, Km 2 ½, reparto Torrens, municipio Boyeros, La Habana, Cuba. CP: 19370. mpuebla12@gmail.com

² Universidad Tecnológica de La Habana José Antonio Echeverría, CUJAE. D. asimon@ceis.cujae.edu.cu

³ Universidad de Extremadura (España). jmperea@unex.es

⁴ Universidad de las Ciencias Informáticas. yanete@uci.cu

*Autor para la correspondencia. (mpuebla12@gmail.com)

RESUMEN

La mayoría de las herramientas de código abierto para administrar y razonar sobre ontologías grandes requieren cargar previamente toda la ontología en la memoria principal para realizar operaciones de inferencia. Esto es un inconveniente cuando los recursos de hardware son limitados y se requiere administrar ontologías geográficas con millones de datos (instancias), debido a la alta complejidad espacial y temporal de los algoritmos utilizados por los razonadores basados en memoria. Este artículo presenta una extensión de OWLAPI para administrar y razonar sobre grandes ontologías geográficas en cualquier perfil de OWL2. La contribución principal es la reducción del uso de la memoria principal por parte de OWLAPI, pues solo una parte de la ontología es cargada en memoria principal y no todo el archivo OWL. Dos grandes geo-ontologías de diferentes áreas geográficas se gestionaron utilizando la solución propuesta con fines de demostración. Los resultados obtenidos demuestran la viabilidad de la propuesta para administrar grandes ontologías geográficas en OWL2 mediante el uso de OWLAPI y recursos de hardware limitados.

Palabras clave: Ontologías geográficas grandes; Representación del conocimiento; Razonamiento sobre grandes ontologías; OWLAPI; OWL2.

ABSTRACT

Most of the existing open-source tools to manage and reason on large ontologies require previously loading the entire ontology in main memory to perform inference operations. This is a drawback when limited hardware resources are available and it is required to manage geographic ontologies with millions of data (instances), due to the high spatial and temporal complexity of the algorithms used by memory-based reasoners. This paper presents an extension of the OWLAPI to manage and reason on large geographic ontologies in any OWL2 profile. The main contribution is the reduction of the main memory usage by the OWLAPI, thus allowing loading only part of the ontology and not the entire OWL file. Two large geo-ontologies from different geographical areas were managed by using the proposed solution for

demonstration purposes. The obtained results demonstrate the viability of the proposal to manage large geographic ontologies in OWL2 by using OWLAPI and limited hardware resources.

Keywords: Large geographic ontologies; Knowledge representation; Reasoning on large ontologies; OWLAPI; OWL2.

Introducción

El uso de ontologías es cada vez más requerido por aquellos sistemas computacionales que necesitan procesar la semántica asociada al contenido. Este es el caso de los Sistemas de Recuperación de Información Geográfica (SRIG), aplicaciones que requieren representar y describir semánticamente datos espaciales de zonas geográficas del mundo real (ciudades, países, etc.). En este escenario se hace imprescindible utilizar ontologías de gran tamaño que ofrezcan la posibilidad de describir el significado de los datos espaciales y sus relaciones, faciliten la generación de nuevo conocimiento espacial y permitan mejorar la toma de decisiones (Ulutaş Karakol et al., 2018). El uso efectivo de las ontologías no solo requiere de la codificación del conocimiento representado en un lenguaje formal, sino también de un soporte computacional adecuado para su administración y la realización de procesos de inferencia sobre el conocimiento formalmente representado. En este sentido han surgido los sistemas DBBO (*Databases Based on Ontologies*) y los sistemas OBDB (Ontology Based Data Bases), muy parecidos entre sí, pero con sus diferencias en cuanto a las características de la fuente de datos que administran: bases de datos relacionales para los DBBO y ontologías para los OBDB.

Los sistemas DBBO también conocidos como OBDA (*Ontology-Based Data Access*), se han convertido en un popular paradigma para acceder a una o varias fuentes de datos mediante el uso de ontologías (Nikolaou et al., 2019), aprovechando las ontologías para incrementar su capacidad semántica de administrar información. En los sistemas DBBO los usuarios acceden a los datos a través de una capa conceptual (abstracción de los aspectos específicos relacionados con las fuentes de datos) que proporciona un cómodo

vocabulario de consulta. Esta capa se representa generalmente mediante una ontología en RDF^b u OWL^c y se conecta a las bases de datos relacionales subyacentes utilizando asignaciones R2RML^d. Cuando se realiza una consulta SPARQL sobre la ontología, el sistema DBBO explora las asignaciones representadas para recuperar elementos de las fuentes de datos y construir las respuestas (Calvanese et al., 2016). Una de las limitaciones que presentan actualmente los sistemas DBBO tiene que ver con la imposibilidad de hacer cambios en la BDR a través de la ontología debido, generalmente, a la existencia de asignaciones complejas y arbitrarias entre la ontología y la BDR. Este problema se conoce como el “*view update problema*” (Franconi y Guagliardo, 2012) y se produce porque, debido a las asignaciones, no existe una manera única de propagar una actualización específica del nivel de la ontología a la base de datos subyacente. Otra limitación de los sistemas DBBO es que no están diseñados para soportar ontologías en cualquier perfil OWL2, debido a que su función principal es acceder a grandes BDR. Por este motivo utilizan lenguajes que proporcionan un menor poder de expresividad, como es el caso de OWL2-QL. Algunos ejemplos de sistemas DBBO son GraphDB (Güting, 1994), OntoDB (Dehainsala, Pierra y Bellatreche, 2007), Optique (Calvanese et al., 2013), RDFox (Nenov et al., 2015), Ontop (Calvanese, Rezk y Xiao, 2016), OptiqueVQS (Soylu et al., 2018), Squerall (Mami M.N et al., 2019).

Por otra parte, los sistemas OBDB, permiten almacenar y consultar ontologías con una gran cantidad de instancias. Una ventaja de estos sistemas tiene que ver con su capacidad para utilizar algunas de las funcionalidades que proporcionan los Sistemas de Gestión de Bases de Datos Relacionales (RDBMS, siglas en inglés), tales como rendimiento en las consultas, almacenamiento eficiente de los datos, administración de transacciones, entre otras. Ejemplos de estos sistemas son: *Database-Based Sesame* (Lantica Software LLC., 2017), DLDB-OWL (Pan y Heflin, 2003), OWLIM (Kiryakov, Ognyanov y Manov, 2005), InstanceStore (Horrocks et al., 2004), Minerva (Zhou et al., 2006a), DBOWL (García y Montes, 2012), OntoMinD (Al-Jadir, Parent y Spaccapietra, 2010), OwlOntDB (Faruqui y MacCaull, 2013) y FGOLD (Bakhtouchi, 2015).

Son varios los autores que reconocen la existencia de limitaciones a la hora de combinar administración de grandes volúmenes de información y poder especificar determinadas características de los datos, como ocurre en el caso del dominio geográfico (Faruqui y MacCaull, 2013), (Gu y Zhang, 2015), (Maria Keet, 2018), (Eine, Jurisch y Quint, 2017), (Koutsomitropoulos y Kalou, 2017). Una posible solución al problema

planteado anteriormente pasa por utilizar lenguajes ontológicos con un mayor poder de expresividad (por ejemplo OWL, Ontology Web Language), la cual es requerida en este dominio para poder inferir de forma automática un mayor número de relaciones espaciales (Faruqui y MacCaull, 2013), (Jerry R. Hobbs y Feng Pan, 2017).

Una limitación relacionada con la gestión de grandes volúmenes de información está relacionada con los propios editores de ontologías. La mayor parte de los editores disponibles actualmente (*Protégé*^e es de los más utilizados, así como los razonadores *Pellet*, *FaCT++*, *HermiT*, *TrOWL*, *RACER*, entre otros), requieren cargar previamente toda la ontología en memoria principal para poder llevar a cabo tareas de administración y ejecutar mecanismos de inferencia, no permitiendo su gestión en memoria externa cuando la capacidad en memoria principal es limitada (Dentler et al., 2011), (Alatrish, 2013). Este aspecto constituye un problema cuando se gestionan ontologías geográficas con millones de datos (instancias) y se tienen recursos hardware limitados debido a la elevada complejidad espacial y temporal de los algoritmos utilizados por los razonadores basados en memoria principal (Faruqui y MacCaull, 2013). En este sentido OWLAPI^f es una API (*Application Programming Interface*) de alto nivel muy utilizada para trabajar con ontologías codificadas en OWL que ofrece mecanismos para la manipulación de estructuras ontológicas y el uso de motores de razonamiento. Según varios autores (Horridge y Bechhofer, 2011) (Bechhofer y Matentzoglou, 2018), su mayor limitante está en la necesidad de cargar el fichero OWL en memoria principal, siendo esto una inconveniente importante cuando el tamaño de la ontología es similar al de la memoria principal disponible en el computador.

Por otro lado existe un compromiso entre la expresividad lógica de una ontología y la complejidad computacional: cuanto más expresivo es el lenguaje de la ontología, mayor es su complejidad computacional (Dentler et al., 2011), (Zhou et al., 2006b). Como consecuencia, la mayoría de herramientas ontológicas capaces de trabajar con memoria externa y grandes volúmenes de datos optan por disminuir la expresividad del lenguaje para garantizar tiempos computacionales aceptables.

Teniendo en cuenta este escenario, el diseño de un modelo de representación basado en la integración de ontologías y Bases de Datos Relacionales (BDR) está resultando ser una alternativa prometedora (Aguar, Falbo y Souza, 2018), (Jean et al., 2012) ya que, por un lado, la ontología podría describir la semántica asociada al modelo de datos utilizado en una BDR y, por otro lado, una BDR podría ser utilizada para

almacenar parte del conocimiento que normalmente se representa en la ontología como, por ejemplo, las instancias.

En este trabajo se desarrolló un modelo de representación de conocimiento en el dominio geográfico y se propone un mecanismo de administración y razonamiento aplicable a dicho modelo. La propuesta se basa en los principios de un sistema OBDB y se dirige fundamentalmente a la gestión de ontologías geográficas de gran tamaño, ya que soluciones actuales como OWLAPI no ofrecen soporte cuando todo el conocimiento de la ontología no se puede almacenar en memoria principal debido a limitaciones hardware. Las principales contribuciones del enfoque propuesto son:

1. Utilización de un esquema de representación de conocimiento compartido entre la ontología y una BDR, incluyendo mecanismos de administración y razonamiento necesarios para este tipo de esquema. Como novedad este esquema permite hacer cambios en la BDR a través de la ontología, aspecto que no contemplan los sistemas DBBO actuales debido a que estos solo permiten realizar operaciones de consulta sobre la BDR.
2. Disminución del uso de memoria principal al permitir cargar solo una parte de la ontología y no todo el fichero OWL, consiguiendo así facilitar la administración y el razonamiento sobre ontologías de gran tamaño en el dominio geográfico cuando los recursos hardware son limitados.
3. Soporte de ontologías escritas en cualquier perfil OWL2. Esta es otra limitación de los sistemas DBBO actuales debido a que su objetivo principal es el acceso a grandes BDR, optando por utilizar lenguajes con menor poder de expresividad que OWL2.

Métodos o Metodología Computacional

La solución propuesta pretende extender OWLAPI en dos dimensiones fundamentales para trabajar con ontologías de gran tamaño: por un lado, la administración de la ontología y por otro el razonamiento o inferencia, incrementando así las potencialidades de dicha herramienta. Esta solución se basa en los principios de un sistema OBDB donde la estructuración y almacenamiento de la conceptualización a

formalizar en la ontología y los datos son compartidos entre el fichero OWL y una BDR. En la Fig. 1 se ilustra de forma gráfica la asignación de responsabilidades entre OWL API y la BDR en el momento de gestionar los componentes de la ontología.

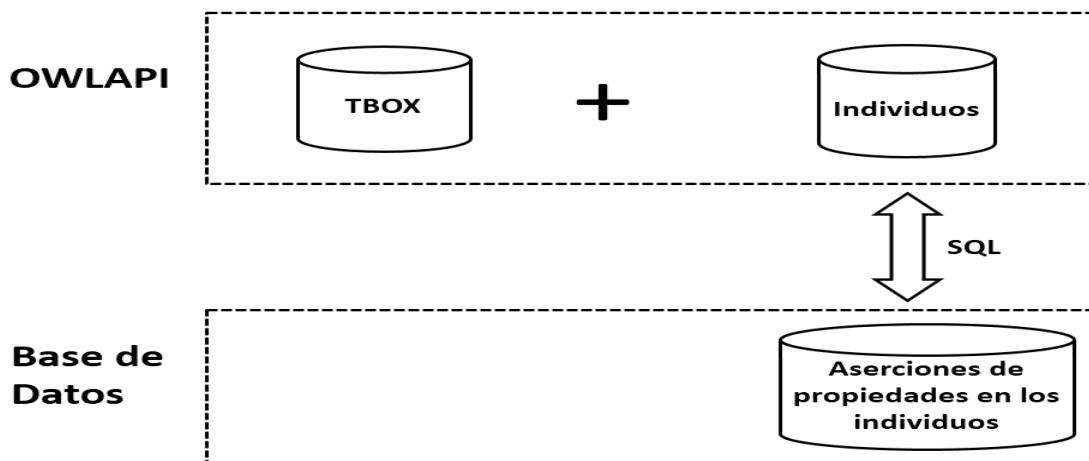


Fig. 1 – Asignación de responsabilidades entre OWLAPI y la Base de Datos.

Administración de la ontología

En la solución propuesta se decidió extraer del fichero OWL gestionado por OWL API algunos datos pertenecientes al ABOX (*Assertion box, data*) y mantener íntegramente el TBOX (*Terminological box, schema*) de la ontología, de forma similar a los sistemas OBDB analizados. Esto se hizo con el objetivo de poder utilizar los razonadores en memoria interna en algunas tareas específicas y así mantener la capacidad de razonamiento sobre la ontología. Los razonadores existentes están diseñados para realizar procesos de inferencia solo sobre lo incluido en el fichero OWL de la ontología, fundamentalmente sobre el TBOX. Los datos pertenecientes al ABOX que se decidieron almacenar en la BDR fueron principalmente dos: las relaciones de objeto expresadas entre pares de individuos (*Object Property Assertion Axioms*) y la asignación de valores a las propiedades de datos (*Data Property Assertion Axioms*).

Inicialmente, debido a la simplicidad estructural de la información a almacenar, se decidió utilizar simples ficheros como medio de almacenamiento en memoria externa. Sin embargo, al hacer varias pruebas los

autores se percataron de que el tiempo de acceso a la información era demasiado elevado y la frecuencia con la que se debía acceder a los mismos también, sobre todo si consideramos que el tamaño del fichero sobrepasó los 15 GB para la región de Marianao. Por tal motivo, se optó por utilizar una base de datos relacional.

Entre utilizar un modelo de Base de Datos Relacional (BDR) y No Relacional (BDNR) se decidió por el primero porque las BDNR no son adecuadas cuando existen relaciones entre los datos. En esta investigación, aunque los individuos no se almacenaron en la BDR, sí se plantea esta opción como mejora en el futuro, por lo que se decidió apostar desde un principio por utilizar una BDR. Dejar los individuos en el fichero OWL tiene la ventaja de reutilizar toda la eficiencia de OWLAPI en relación a los individuos y sus clases, almacenarlos en la BDR implicaba el esfuerzo de implementar toda esa lógica desde cero. El modelo físico de la BDR utilizada en la solución propuesta se ilustra en la Fig. 2.

Para la tabla “*DataProperty*”, los tres últimos campos conforman una restricción de campo único, es decir, no pueden existir dos tuplas con los mismos valores en estos campos. Lo mismo sucede con los tres últimos campos de la tabla “*ObjectProperty*”. La comunicación entre el fichero OWL y la información que se almacena en la base de datos se realiza a través de consultas SQL. Por ejemplo, si se necesita conocer todos los literales de un individuo para una propiedad de datos en particular, se ejecuta una consulta de selección que dado el nombre del individuo y el nombre de la propiedad de datos seleccione todos los literales.

En la solución desarrollada se implementaron los mecanismos necesarios para que las instancias de propiedades pudieran ser integradas nuevamente en el fichero OWL en su forma tradicional, siempre y cuando la memoria interna del hardware utilizado soportara dicho crecimiento.

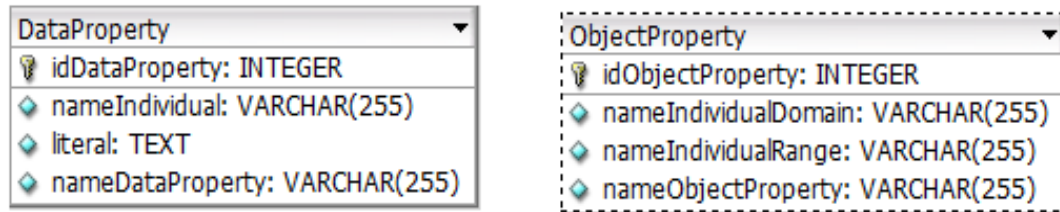


Fig. 2 – Modelo físico de la base de datos diseñada.

Razonamiento sobre la ontología owl2 usando OWLAPI

Una vez cargada la ontología en OWLAPI fue necesario desarrollar algunas operaciones que exigían el uso de razonadores en la manera tradicional, lo cual continuaba siendo una limitación a pesar de las modificaciones realizadas. En este sentido se implementaron diferentes operaciones necesarias sin utilizar razonadores y apoyándose únicamente en recursos facilitados por OWLAPI. Estas operaciones, a diferencia de las desarrolladas por un razonador profesional sobre una ontología estándar, no validan la totalidad de las situaciones presentes en el expresivo lenguaje OWL2, pero permitieron dar soporte a las necesidades que planteaba la gestión de una ontología en el dominio geográfico. El total de operaciones implementadas sin utilizar razonadores se enumeran a continuación:

1. Buscar las subclases directas e indirectas de una clase.
2. Identificar los individuos directos e indirectos de una clase, teniendo en consideración la equivalencia entre clases y entre individuos.
3. Identificar las “clases topes”, es decir, las clases que no son subclases de ninguna clase.
4. Identificar todas las clases equivalentes a una clase, considerando que la relación de equivalencia es transitiva.
5. Identificar el conjunto de superclases de una clase.
6. Obtener el conjunto de individuos que están relacionados con otro a través de una propiedad de objeto, teniendo en consideración las características de la propiedad de objeto expresada en OWL2: simetría, reflexividad, funcional, transitividad, inversa, equivalencia.
7. Obtener el conjunto de literales asociados a una propiedad de datos para un individuo en particular.
8. Similitud entre individuos pertenecientes al dominio geográfico.

En las Tablas 1 y 2 se muestra el algoritmo definido para inferir las subclases directas o indirectas de una clase. La Tabla 2 muestra un procedimiento auxiliar que apoya al mostrado en la Tabla 1.

Tabla 1 - Algoritmo para las subclases directas de una clase.

Entrada: Una clase (owlClass) y un valor lógico (direct) que indica si se buscarán las subclases directas o no.

Salida: Conjunto de subclases de owlClass.

- 1 **Inicio**
- 2 Crear un conjunto vacío de clases: “owlClasses”
- 3 **Retornar** SubClasesDirectasAuxiliar(owlClass, owlClasses, direct)
- 4 **Fin**

La solución propuesta soporta cualquier perfil OWL2 y permite la gestión de datos geográficos, modelándolos en la ontología como una cadena de texto en el formato WKT⁸. La gestión de la ontología se hace mayormente a través de OWL API, mientras que solo las aserciones de propiedades son administradas desde la BDR.

Tabla 2- Procedimiento auxiliar “subclasesdirectasauxiliar”.

Entrada: Una clase llamada OwlClass, lista de subconceptos encontrados: listResult y un valor lógico que indica si se buscarán los subconceptos directos o no: direct.

Salida: Conjunto de subconceptos.

- 1 **Inicio**
- 2 **Si** (OwlClass es distinto **null**)
- 3 **Inicio**
- 4 Guardar en “listAux” todas los sub conceptos directos de “OwlClass”
- 5 **Para todo** objeto perteneciente a “listAux” **hacer:**
- 6 Adicionar a “listResult” el objeto
- 7 **Si** (direct == falso), es decir, si piden los conceptos indirectos
- 8 **Inicio**
- 9 **Para cada** objeto perteneciente a listAux **hacer:**
- 10 SubClasesDirectasAuxiliar(objeto, listResult, direct)
- 11 **Fin**
- 12 **Fin**
- 13 **Retornar** listResult
- 14 **Fin**

Resultados y discusión

Para demostrar la viabilidad de la solución propuesta se administraron dos ontologías geográficas de gran tamaño (OntoMarianao y OntoAcapulco)^h, que fueron previamente generadas utilizando el método reportado en (Puebla-Martínez et al., 2018). Con un hardware estándar en cuanto a capacidad de cómputo y memoria, estas ontologías no podrían haberse gestionado con herramientas actuales como OWL API, Protégé, JENA o HERMIT, entre otras, debido al tamaño de las ontologías y al elevado uso de memoria principal por parte de las herramientas antes mencionadas. Sin embargo, la solución propuesta en esta investigación permitió reducir el consumo de memoria principal y garantizó las operaciones de razonamiento básicas en una ontología del dominio geográfico. Los experimentos llevados a cabo se realizaron sobre un hardware con una CPU Mobile DualCore Intel Core i5-2430M, 2800 MHz (28 x 100) y 4 GB de memoria principal.

En la Tabla 3 se muestran las características de la ontología OntoMarianao, que está relacionada con la zona geográfica correspondiente al municipio de Marianao (Cuba). Los datos de OntoAcapulco no se exponen por cuestiones de espacio, pero se han puesto a disposición de la comunidad científica los ficheros OWL (sin las aserciones de propiedades) y la base de datos con las aserciones de propiedades de ambas ontologías geográficas, para que los investigadores interesados pueden disponer de ellas.

Los datos mostrados en la Tabla 3 evidencian la viabilidad de la propuesta, pues una ontología con esas características no podría haber sido administrada por OWL API (o herramientas similares) en una computadora con un hardware estándar como el utilizado en la experimentación llevada a cabo. Los dos primeros datos que aparecen en negritas en la tabla 3 (fila 5 y 6) son almacenados en la base de datos. Noten el tamaño del respaldo de la base de datos (última fila), a pesar de todas las técnicas de compactación de la información que utilizan los gestores de bases de datos tradicionales. Ese volumen de información para una ontología no es posible administrarlo con las herramientas tradicionales.

Por otro lado, aunque no es el principal objetivo de este trabajo, la Tabla 4 muestra los resultados obtenidos para la ontología de Marianao en algunas de las funcionalidades de inferencia desarrolladas. La columna “Operación” hace referencia al número de la operación de razonamiento mostrada en la Sección 3. La información recuperada para cada consulta realizada demuestra también la viabilidad de las operaciones de razonamiento desarrolladas a pesar del tamaño de la ontología y el alto nivel de expresividad utilizado.

Es válido resaltar que, sin la solución de segmentación que se propone en este artículo, no hubiera sido posible obtener las dos ontologías de experimentación que aquí se abordan: OntoMarianao y OntoAcapulco. De las siete operaciones mostradas en la tabla 4, cuatro de ellas se pueden implementar usando solo OWLAPI (Q1, Q2, Q3 y Q6) y en tres se necesita utilizar el enfoque de segmentación propuesto (Q4, Q5 y Q7). No obstante, es válido una aclaración, normalmente para implementar las operaciones Q1, Q2, Q3 y Q6, se utilizan razonadores que infieren información a partir de lo expresado en la ontología. Pero en este caso no es posible debido al gran tamaño de la ontología, por lo que fue necesario implementar funcionalidades similares a las que desarrollan los propios razonadores, ya explicadas en la sección III.

Tabla 3- Características de la ontología geográfica del municipio Marianao.

Componentes ontológicos	Ontología
Clases	722
Clases equivalentes	12
Relaciones taxonómicas o subclases	954
Individuos	4.665
Relaciones de objeto entre pares de individuos	76.398.888
Propiedades de datos con valor en individuos	102.755
Pares de Individuos similares	13.321
Anotaciones	728
Tamaño del fichero OWL	6.118 KB
Tamaño del respaldo de la base de datos en PostgreSQL	592.334 KB

Tabla 4 - Operaciones realizadas sobre ontomarianao.

Id	Consulta	Información recuperada	Operación
Q1	Subclases directas e indirectas de la clase “aeroway”	1) “aerodrome”. 2) “aiport”. En la ontología la clase “aiport” es subclase directa de “aerodrome” y esta a su vez subclase directa de “aeroway”.	1
Q2	Instancias directas e indirectas de la clase “aeroway”	1) “OSM-24047147-26-08”. 2) “OSM-252369011-Hangar”. 3) “OSM-252369012-Hangar”. 4) “OSM-259849768-Aerodromo-Ciudad-Libertad”. 5) “Geo-8554333-Ciudad-Libertad-Airport”. 6) “DB-osm_new_aeroways-2”. 7) “DB-osm_new_aeroways-3” Además de lo expuesto en Q1, tenga en consideración que la clase “aeroway” es equivalente a “osm_new_aeroways”. Los 4 primeros resultados son instancias directas de “aeroway”. El quinto es instancia directa de “aiport” y las dos últimas instancias directas de “osm_new_aeroways”	2 y 4
Q3	Clases Topes en la ontología.	1) “SpatialObject”	3
Q4	Calles que cruzan la localidad de “Los Pocitos”	36 resultados. Aquí solo seis: 1) Avenida 51. 2) 130. 3) 116. 4) 128. 5) 120. 6) 118...	6
Q5	Objetos espaciales en “Los Pocitos”	1) Arroyo Bañabuey. 2) Policlínico Docente 27 de noviembre. 3) Plaza de Marianao. 4) Los-Pocitos. 5) Plaza-de-Marianao ...	6
Q6	Superclases de “Administrative Boundary”	1) “boundary”. 2) “GeonameConcept”. 3) “SpatialObject” La clase “AdministrativeBoundary” es subclase de “boundary” y de “GeonameConcept”, las cuales a su vez son subclases de “SpatialObject”	5
Q7	Pares de Individuos similares	OSM-3917885242-Cocosolo y Geo-3563920-Coco Solo. OSM-4201289738-Alturas de Belén y Geo-3567530-Belén	8

Conclusiones

En este trabajo se desarrolló una extensión de OWLAPI que permite administrar ontologías de gran tamaño utilizando un hardware con recursos estándar. La solución propuesta soporta el lenguaje ontológico OWL2, con lo que se eleva el nivel de expresividad que se puede utilizar al desarrollar operaciones de inferencia sobre la ontología. Este aspecto es fundamental cuando se administran ontologías del dominio geográfico, pues permite incluir algunas capacidades de razonamiento sobre el ABOX de la ontología que son difíciles desarrollar con otros lenguajes menos expresivos. Además de la extensión de OWL API se desarrollaron varias operaciones de razonamiento básicas en una ontología del dominio geográfico sin utilizar otros razonadores existentes.

Para demostrar la viabilidad de la solución propuesta se administraron dos geo-ontologías de gran tamaño correspondientes a diferentes áreas geográficas. Las características de las ontologías administradas en cuanto a clases, instancias, relaciones ontológicas o tamaño en memoria confirmaron el éxito de la solución propuesta. Además de reducir el consumo de memoria principal, se garantizó el funcionamiento de las operaciones de razonamiento básicas desarrolladas, siempre bajo el marco de experimentación utilizado. Por último, se comprobó que las ontologías utilizadas en la experimentación no podrían haberse administrado con herramientas actuales como OWL API utilizando el mismo marco de trabajo, debido al elevado uso de memoria principal que requieren.

El resultado obtenido puede ser aplicado a grandes ontologías de otros dominios de información, lo que aumentaría las posibilidades de aplicación y el impacto de la investigación.

Como trabajo futuro se propone aumentar las capacidades de razonamiento desarrolladas, así como almacenar los individuos de la ontología en la base de datos relacional para disminuir aún más los requisitos de memoria principal.

Agradecimientos

Los autores desean agradecer de manera especial los criterios y recomendaciones aportadas por el profesor Dr.C. Ignazio Palmisano de la Universidad de Manchester, así como al profesor Dr.C. Diego Calvanese de la Facultad de Ciencia de la Computación de la Universidad de Bozen-Bolzano, por sus intercambios relacionados con la herramienta Ontop y los sistemas DBBO.

Referencias

- Aguiar, C.Z. De, Falbo, R. De A. Y Souza, V.E.S., 2018. Ontological Representation of Relational Databases. *Ontobras 2018. Xi Seminar on Ontology Research in Brazil* [En Línea]. São Paulo, Brazil, Ceur Workshop Proceedings, Pp. 140-151. Disponible En: <Http://Ceur-Ws.Org/Vol-2228/Paper9.Pdf>.
- Alatrish, E., 2013. Comparison Some of Ontology Editors. *International Scientific Journal of Management Information Systems* [En Línea], Vol. 8, No. 2, Pp. 018-024. Issn 1452-774x. Disponible En: <Https://Www.Ingentaconnect.Com/Content/Doaj/1452774x/2013/00000008/00000002/Art00001>.
- Al-Jadir, L., Parent, C. Y Spaccapietra, S., 2010. Reasoning with Large Ontologies Stored In Relational Databases: The Ontomind Approach. *Data Knowl. Eng.*, Vol. 69, No. 11, Pp. 1158-1180. Issn 0169-023x. Doi 10.1016/J.Datak.2010.07.006.
- Bakhtouchi, A., 2015. Fgold: A Flexible and Graphical Ontology-Based Database Ready for Use. *International Journal on Human Machine Interaction* [En Línea], Vol. 2, No. 2, Pp. 32-49. Issn 2356-556x. Disponible En: <Http://Ijhmi.Asdfjournals.Com/Index.Php/Ijhmi/Article/View/20>.
- Bechhofer, S. Y Matentzoglou, N., 2018. *The Owl Api: An Introduction* [En Línea]. 2018. S.L.: University Of Manchester. Disponible En: <Http://Syllabus.Cs.Manchester.Ac.Uk/Pgt/2018/Comp62342/Slides/Week4-Owlapi.Pdf>.
- Calvanese, D., B, C., S, K.-E., Kontchakov, R. Y Lanti, D., 2016. Ontop: Answering Sparql Queries over Relational Databases. *Semantic Web – Interoperability, Usability, Applicability* [En Línea], Vol. Preprint, N.º Preprint, Pp. 1-17. Issn Ssn Print 1570-0844 And Issn Online 2210-4968. Disponible En: <Http://Dx.Doi.Org/10.3233/Sw-160217>.

- Calvanese, D., Giese, M., Haase, P., Horrocks, I., Hubauer, T., Ioannidis, Y., Jiménez-Ruiz, E., Kharlamov, E., Kllapi, H., Klüwer, J., Koubarakis, M., Lamparter, S., Möller, R., Neuenstadt, C., Nordtveit, T., Özcep, Ö. Rodríguez-Muro, M., Roshchin, M., Savo, F., Schmidt, M., Soylu, A., Waaler, A. Y Zheleznyakov, D., 2013. Optique: Obda Solution for Big Data. En: P. Cimiano, M. Fernández, V. Lopez, S. Schlobach Y J. Völker (Eds.), *the Semantic Web: Eswc 2013 Satellite Events*. Berlin, Heidelberg: Springer Berlin Heidelberg, Pp. 293-295. Isbn 978-3-642-41242-4. Doi https://doi.org/10.1007/978-3-642-41242-4_48.
- Calvanese, D., Rezk, M. Y Xiao, G., 2016. *Iswc 2015 Tutorial \Textbackslashtextbar -Ontop-* [En Línea]. S.L.: S.N. Disponible En: <http://ontop.inf.unibz.it/iswc-2015-tutorial/>.
- Dehainsala, H., Pierra, G. Y Bellatreche, L., 2007. Ontodb: An Ontology-Based Database for Data Intensive Applications. En: R. Kotagiri, P.R. Krishna, M. Mohania Y E. Nantajeewarawat (Eds.), *Advances In Databases: Concepts, Systems And Applications: 12th International Conference On Database Systems For Advanced Applications, Dasfaa 2007, Bangkok, Thailand, April 9-12, 2007. Proceedings* [En Línea]. Berlin, Heidelberg: Springer Berlin Heidelberg, Pp. 497-508. Isbn 978-3-540-71703-4. Disponible En: http://dx.doi.org/10.1007/978-3-540-71703-4_43.
- Dentler, K., Cornet, R., Ten Teije, A. Y De Keizer, N., 2011. Comparison of Reasoners for Large Ontologies in the Owl 2 El Profile. *Journal Semantic Web* [En Línea], Vol. 2, No. 2, Pp. 71-87. Issn 1570-0844. Doi 10.3233/Sw-2011-0034. Disponible En: <http://dx.doi.org/10.3233/Sw-2011-0034>.
- Eine, B., Jurisch, M. Y Quint, W., 2017. Ontology-Based Big Data Management. *Systems* [En Línea], Vol. 5, No. 3, Pp. 45. Issn 2079-8954. Doi 10.3390/Systems5030045. Disponible En: <https://www.mdpi.com/2079-8954/5/3/45>.
- Faruqui, R.U. Y Maccaull, W., 2013. Owlontdb: A Scalable Reasoning System for Owl 2 RI Ontologies with Large Aboxes. En: J. Weber Y I. Perseil (Eds.), *Foundations of Health Information Engineering and Systems* [En Línea]. Berlin, Heidelberg: Springer Berlin Heidelberg, Pp. 105-123. Isbn 978-3-642-39088-3. Doi https://doi.org/10.1007/978-3-642-39088-3_7. Disponible En: https://link.springer.com/chapter/10.1007%2f978-3-642-39088-3_7#citeas.
- Franconi, E. Y Guagliardo, P., 2012. The View Update Problem Revisited. *Corr* [En Línea], Vol. Abs/1211.3016. Disponible En: <http://arxiv.org/abs/1211.3016>.

- García, M. Del M.R. Y Montes, J.F.A., 2012. Evaluating Dbowl: A Non-Materializing Owl Reasoner Based On Relational Database Technology. *Proceedings of the 1st International Workshop on Owl Reasoner Evaluation (Ore-2012)*. Manchester, Uk: Ceur-Ws.Org,
- Gu, Z. Y Zhang, S., 2015. Querying Large and Expressive Biomedical Ontologies. *2015 Ieee 17th International Conference On High Performance Computing And Communications, 2015 Ieee 7th International Symposium On Cyberspace Safety And Security, And 2015 Ieee 12th International Conference On Embedded Software And Systems*. S.L.: S.N., Pp. 491-496.
- Gütting, R.H., 1994. Graphdb: Modeling and Querying Graphs in Databases. *Proceedings of the 20th International Conference on Very Large Data Bases* [En Línea]. San Francisco, Ca, Usa: Morgan Kaufmann Publishers Inc., Pp. 297-308. Isbn 1-55860-153-8. Disponible En: [Http://Dl.Acm.Org/Citation.Cfm?Id=645920.672980](http://dl.acm.org/citation.cfm?id=645920.672980).
- Horridge, M. Y Bechhofer, S., 2011. The Owl Api: A Java Api For Owl Ontologies. *Semantic Web* [En Línea], Vol. 2, No. 1, Pp. 11-21. Issn 1570-0844. Disponible En: [Http://Dl.Acm.Org/Citation.Cfm?Id=2019470.2019471](http://dl.acm.org/citation.cfm?id=2019470.2019471).
- Horrocks, I., Li, L., Turi, D., Bechhofer, S., Haarslev, V. Y Möller, R., 2004. The Instance Store: Dl Reasoning with Large Numbers of Individuals. En: V. Haarslev Y R. Möller (Eds.), *Description Logics*. S.L.: Ceur-Ws.Org, Ceur Workshop Proceedings,
- Jean, S., Bellatreche, L., Fokou, G., Baron, M. Y Khouri, S., 2012. Ontodbench: Novel-Benchmarking System for Ontology-Based Databases. En: R. Meersman, H. Panetto, T. Dillon, S. Rinderle-Ma, P. Dadam, X. Zhou, S. Pearson, A. Ferscha, S. Bergamaschi Y I.F. Cruz (Eds.), *On The Move To Meaningful Internet Systems: Otm 2012* [En Línea]. Berlin, Heidelberg: Springer Berlin Heidelberg, Pp. 897-914. Isbn 978-3-642-33615-7. Doi [Https://Doi.Org/10.1007/978-3-642-33615-7_32](https://doi.org/10.1007/978-3-642-33615-7_32). Disponible En: [Https://Link.Springer.Com/Chapter/10.1007/978-3-642-33615-7_32#Citeas](https://link.springer.com/chapter/10.1007/978-3-642-33615-7_32#citeas).
- Jerry R. Hobbs Y Feng Pan, 2017. *Time Ontology in Owl* [En Línea]. 19 Octubre 2017. S.L.: W3c & Ogc. Disponible En: [Https://Www.W3.Org/Tr/Owl-Time/](https://www.w3.org/Tr/Owl-Time/).
- Kiryakov, A., Ognyanov, D. Y Manov, D., 2005. Owlrim – A Pragmatic Semantic Repository for Owl. En: M. Dean, Y. Guo, W. Jun, R. Kaschek, S. Krishnaswamy, Z. Pan Y Q.Z. Sheng (Eds.), *Web Information Systems Engineering – Wise 2005 Workshops: Wise 2005 International Workshops, New York, Ny, Usa*

- November 20-22, 2005. *Proceedings* [En Línea]. Berlin, Heidelberg: Springer Berlin Heidelberg, Pp. 182-192. Isbn 978-3-540-32287-0. Disponible En: [Http://Dx.Doi.Org/10.1007/11581116_19](http://Dx.Doi.Org/10.1007/11581116_19).
- Koutsomitropoulos, D.A. Y Kalou, A.K., 2017. A Standards-Based Ontology and Support for Big Data Analytics in the Insurance Industry. *Ict Express* [En Línea], Vol. 3, No. 2, Pp. 57-61. Issn 2405-9595. Doi [Https://Doi.Org/10.1016/J.Icte.2017.05.007](https://Doi.Org/10.1016/J.Icte.2017.05.007). Disponible En: [Http://Www.Sciencedirect.Com/Science/Article/Pii/S2405959517300875](http://Www.Sciencedirect.Com/Science/Article/Pii/S2405959517300875).
- Lantica Software Llc., 2017. *Sesame Database Manager* [En Línea]. S.L.: Lantica Software. [Consulta: 11 marzo 2017]. Disponible En: [Http://Www.Sesamedatabase.Com/](http://Www.Sesamedatabase.Com/).
- Mami M.N, Graux D., Scerri S, Jabeen H, Y Lehmann J, 2019. Squerall: Virtual Ontology-Based Access to Heterogeneous and Large Data Sources. *The Semantic Web – Iswc 2019* [En Línea]. Auckland, New Zealand: Springer International Publishing, The Lecture Notes In Computer Science, Pp. 229-245. Isbn Print Isbn 978-3-030-30795-0, Online Isbn 978-3-030-30796-7. Disponible En: [Https://Doi.Org/10.1007/978-3-030-30796-7_15](https://Doi.Org/10.1007/978-3-030-30796-7_15).
- Maria Keet, 2018. *An Introduction to Ontology Engineering* [En Línea]. 1. S.L.: S.N. Isbn 1-84890-295-6. Disponible En: [Https://People.Cs.Uct.Ac.Za/~Mkeet/Files/Oebook.Pdf](https://People.Cs.Uct.Ac.Za/~Mkeet/Files/Oebook.Pdf).
- Nenov, Y., Piro, R., Motik, B., Horrocks, I., Wu, Z. Y Banerjee, J., 2015. Rdfox: A Highly-Scalable Rdf Store. En: M. Arenas, O. Corcho, E. Simperl, M. Strohmaier, M. D’aquino, K. Srinivas, P. Groth, M. Dumontier, J. Heflin, K. Thirunarayan Y S. Staab (Eds.), *The Semantic Web - Iswc 2015* [En Línea]. Cham: Springer International Publishing, Pp. 3-20. Isbn 978-3-319-25010-6. Doi [Https://Doi.Org/10.1007/978-3-319-25010-6_1](https://Doi.Org/10.1007/978-3-319-25010-6_1). Disponible En: [Https://Link.Springer.Com/Chapter/10.1007/978-3-319-25010-6_1#Citeas](https://Link.Springer.Com/Chapter/10.1007/978-3-319-25010-6_1#Citeas).
- Nikolaou, C., Kostylev, E.V., Konstantinidis, G., Kaminski, M., Grau, B.C. Y Horrocks, I., 2019. Foundations of Ontology-Based Data Access under Bag Semantics. *Artificial Intelligence* [En Línea], Vol. 274, Pp. 91-132. Issn 0004-3702. Doi [Https://Doi.Org/10.1016/J.Artint.2019.02.003](https://Doi.Org/10.1016/J.Artint.2019.02.003). Disponible En: [Http://Www.Sciencedirect.Com/Science/Article/Pii/S0004370219300426](http://Www.Sciencedirect.Com/Science/Article/Pii/S0004370219300426).
- Pan, Z. Y Heflin, J., 2003. Dldb: Extending Relational Databases to Support Semantic Web Queries. *In Psssl* [En Línea]. S.L.: S.N., Pp. 109-113. Disponible En: [Http://Swat.Cse.Lehigh.Edu/Pubs/Pan04a.Pdf](http://Swat.Cse.Lehigh.Edu/Pubs/Pan04a.Pdf).
- Puebla-Martínez, M.E., Perea-Ortega, J.M., Simón-Cuevas, A. Y Romero, F.P., 2018. Automatic Expansion of Spatial Ontologies for Geographic Information Retrieval. En: J. Medina, M. Ojeda-Aciego, J.L.

Verdegay, D.A. Pelta, I.P. Cabrera, B. Bouchon-Meunier Y R.R. Yager (Eds.), *Information Processing And Management Of Uncertainty In Knowledge-Based Systems. Theory and Foundations*. [En Línea]. 1. Cham: Springer International Publishing, Communications in Computer and Information Science. Pp. 659-670. Isbn 978-3-319-91476-3. Disponible En: <https://www.springer.com/us/book/9783319914756>.

Soylu, A., Kharlamov, E., Zheleznyakov, D., Jimenez-Ruiz, E., Giese, M., Skjæveland, M.G., Hovland, D., Schlatter, R., Brandt, S., Lie, H. Y Horrocks, I., 2018. Optiquevqs: A Visual Query System over Ontologies for Industry. *Semantic Web*, Vol. 9, No. 5, Pp. 627-660. Issn 2210-4968. Doi 10.3233/Sw-180293.

Ulutaş Karakol, D., Kara, G., Yä±Lmaz, C. Y Cömert, Ç. 2018. Semantic Linking Spatial Rdf Data To The Web Data Sources. *Isprs - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* [En Línea], Vol. Xlii-4, Pp. 639-645. Doi 10.5194/Isprs-Archives-Xlii-4-639-2018. Disponible En: <https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/Xlii-4/639/2018/>.

Zhou, J., Ma, L., Liu, Q., Zhang, L., Yu, Y. Y Pan, Y., 2006a. Minerva: A Scalable Owl Ontology Storage and Inference System. En: R. Mizoguchi, Z. Shi Y F. Giunchiglia (Eds.), *The Semantic Web – Aswc 2006: First Asian Semantic Web Conference, Beijing, China, September 3-7, 2006. Proceedings* [En Línea]. Berlin, Heidelberg: Springer Berlin Heidelberg, Pp. 429-443. Isbn 978-3-540-38331-4. Disponible En: http://dx.doi.org/10.1007/11836025_42.

Zhou, J., Ma, L., Liu, Q., Zhang, L., Yu, Y. Y Pan, Y., 2006b. Minerva: A Scalable Owl Ontology Storage And Inference System. En: R. Mizoguchi, Z. Shi Y F. Giunchiglia (Eds.), *The Semantic Web – Aswc 2006: First Asian Semantic Web Conference, Beijing, China, September 3-7, 2006. Proceedings* [En Línea]. Berlin, Heidelberg: Springer Berlin Heidelberg, Pp. 429-443. Isbn 978-3-540-38331-4. Disponible En: http://dx.doi.org/10.1007/11836025_42.

Conflicto de interés

El autor autoriza la distribución y uso de su artículo.

Contribuciones de los autores

1. Conceptualización: Manuel Enrique Puebla Martínez, Alfredo Simón Cuevas y José Manuel Perea Ortega.
2. Curación de datos: Manuel Enrique Puebla Martínez, Alfredo Simón Cuevas y José Manuel Perea Ortega.
3. Análisis formal: Manuel Enrique Puebla Martínez, Alfredo Simón Cuevas y José Manuel Perea Ortega.
4. Investigación: Manuel Enrique Puebla Martínez, Alfredo Simón Cuevas y José Manuel Perea Ortega.
5. Metodología: Manuel Enrique Puebla Martínez, Alfredo Simón Cuevas y José Manuel Perea Ortega.
6. Administración del proyecto: Manuel Enrique Puebla Martínez, Alfredo Simón Cuevas y José Manuel Perea Ortega.
7. Recursos: Manuel Enrique Puebla Martínez, Alfredo Simón Cuevas, José Manuel Perea Ortega y Yanet Espinal Martín.
8. Software: Manuel Enrique Puebla Martínez y Yanet Espinal Martín.
9. Supervisión: Manuel Enrique Puebla Martínez, Alfredo Simón Cuevas, José Manuel Perea Ortega y Yanet Espinal Martín.
10. Validación: Manuel Enrique Puebla Martínez, Alfredo Simón Cuevas, José Manuel Perea Ortega y Yanet Espinal Martín.
11. Visualización: Manuel Enrique Puebla Martínez, Alfredo Simón Cuevas, José Manuel Perea Ortega y Yanet Espinal Martín.
12. Redacción – borrador original –: Manuel Enrique Puebla Martínez.
13. Redacción – revisión y edición: Manuel Enrique Puebla Martínez, Alfredo Simón Cuevas, José Manuel Perea Ortega y Yanet Espinal Martín.

Notas:

^a <http://owlapi.sourceforge.net/>

^b <https://www.w3.org/RDF/>

^c <https://www.w3.org/OWL/>

^d <https://www.w3.org/TR/r2rml/>

^e <https://protege.stanford.edu>

^f <http://owlapi.sourceforge.net/>

^g <https://www.opengeospatial.org/standards/wkt-crs>

^h https://drive.google.com/drive/folders/1kCTdE7_gLxecZIW12dXXS6kPWGVUJhV_?usp=sharing

ⁱ CPU DualCore Intel Core i5-2430M, 2800 MHz (28 x 100), 4 GB RAM