



Identificación del Trabajo	
Área:	Electrónica, informática y comunicaciones
Categoría:	Alumno
Regional:	Facultad Regional Santa Fe

Aplicación de tecnologías semánticas para la integración de información de estructura de productos almacenadas en sistemas ERP

Nicolás PADULA

INGAR – Instituto de Desarrollo y Diseño (Avellaneda 3657, Santa Fe)

Departamento Sistemas (Lavaise 610, Santa Fe), Facultad Regional Santa Fe, UTN

E-mail de contacto: nicolasvpadula@gmail.com

Este trabajo ha sido realizado bajo la dirección del Dr. Horacio Leone y la Dra. Marcela Vegetti, en el marco del proyecto "Aplicación de Tecnologías de la Web Semántica para la Administración de Información de Productos Durante su Ciclo de Vida".

Resumen

Los avances de las tecnologías de la Información y las comunicaciones han posibilitado a las industrias la adopción de nuevas formas de organizarse y hacer negocios. Sin embargo, impone también un conjunto de inconvenientes a superar relacionados con la integración semántica de sistemas distribuidos. Las ontologías y la Web Semántica son vistas desde hace algunos años como una solución a la interoperabilidad semántica. Según este enfoque, la integración de diversas fuentes de información se logra a través de un vocabulario común definido por una ontología, pero manteniendo la gestión de los datos de manera local en cada nodo de información. Como un paso hacia la definición de este vocabulario, el presente trabajo propone la definición de una ontología a partir del modelo conceptual de una herramienta ERP de código abierto.

Palabras Claves: Ontologías; Web semántica; Interoperabilidad; BOM

1. Introducción y Objetivos

Una forma de afrontar los cambios frecuentes que las empresas enfrentan actualmente en sus entornos de negocios es integrar sus cadenas de suministro. Esta integración debe ser precedida por la integración de la información de productos (Panetto et al., 2012). Tradicionalmente la información de productos se encuentra distribuida en diferentes sistemas como el ERP (Enterprise Resource Planning), PDM (Product Data Management) y más recientemente los sistemas PLM (Product Lifecycle Management).

Actualmente, al integrarse las organizaciones productivas en cadenas de suministros extendidas, los mencionados sistemas conforman una arquitectura del tipo repositorios independientes (Vdovjak et al., 2006). Como una solución al problema de



Identificación del Trabajo	
Área:	Electrónica, informática y comunicaciones
Categoría:	Alumno
Regional:	Facultad Regional Santa Fe

interoperabilidad de estos repositorios heterogéneos, Horrocks et al. (2001) plantean la necesidad de una integración inteligente con tres niveles: tecnológico, sintáctico y semántico. Desde hace ya varios años, Internet y las tecnologías Web proveen una respuesta a los dos primeros tipos de integración. En tanto, las ontologías y las tecnologías de la Web Semántica (Berners-Lee et al., 2001), están siendo vistas como herramientas para alcanzar la integración a nivel semántico.

1.1. *Web Semántica, Interoperabilidad Semántica y Ontologías*

Berners-Lee et al. (2001) plantean la Web Semántica como el próximo paso en la evolución de la web, asociándole la capacidad de dotar a los datos de semántica que pueda ser interpretada por agentes, facilitando así la reusabilidad y la integración de conocimiento a nivel global. Esto desemboca en el desafío de la Interoperabilidad Semántica. La Interoperabilidad Semántica es la posibilidad de los sistemas de intercambiar conceptos con un significado compartido, sin importar que lenguajes o estructuras se usen para definirlos.

Si se consideran dos bases de datos para almacenar información de libros, una perteneciente a una biblioteca de una universidad y otra a una librería (Euzenat y Shvaiko, 2013), las mismas comparten determinados conceptos (los libros) pero al servir distintos propósitos muy probablemente la estructura de las mismas difiera (una puede categorizar los libros empleando categorías comerciales, mientras que la otra puede clasificarlos utilizando categorías literarias o académicas), además, incluso así se estén refiriendo a un mismo concepto, podría darse la situación que se usen términos distintos para denotar los mismos conceptos (una puede hablar de *escritor* mientras que la otra puede hablar de *autor*). Si estas dos organizaciones quieren interactuar, necesariamente habrá que llevar a cabo un proceso para garantizar la interoperabilidad semántica. Es aquí donde entran en juego herramientas como las ontologías, para facilitar la integración de sistemas. Las ontologías capturan los conceptos relevantes de un dominio, así como las relaciones entre los mismos, proveyendo una interpretación semántica de estos conceptos.

Una ontología es una “especificación formal y explícita de una conceptualización compartida” (Staab y Studer, 2009), es decir, una forma de modelar un determinado dominio. Esta especificación se lleva a cabo a través de tres componentes:

- **Individuos:** Objetos de interés en el dominio de discurso.
- **Propiedades:** Relaciones a través de las cuales los individuos pueden asociarse. Las mismas pueden presentar distintas características (reflexividad, transitividad, etc.).
- **Clases:** Conjuntos de individuos. Se pueden definir formalmente requisitos (que se especifican restringiendo propiedades) de pertenencia a una clase dada. Los individuos que cumplan dichos requisitos pertenecerán al conjunto denotado por la clase. Las clases se organizan en taxonomías.

Las ontologías se codifican en lenguajes formales, basados generalmente en lógica de primer orden o algún subconjunto de ésta, como la lógica descriptiva. Debido a la utilización de estos lenguajes es posible, mediante el uso de razonadores ("*reasoners*"), inferir nuevo conocimiento a partir de los axiomas presentes en una ontología. En particular, en el ámbito de la Web Semántica el lenguaje estándar designado por el W3C (World Wide Web Consortium) es OWL (Web Ontology Language), este se basa en tecnologías como RDF (Resource Description Framework) y RDFS (RDF Schema).

Sin embargo, no basta desarrollar una ontología para cada organización para solucionar el problema de interoperabilidad. Es por esto que se debe realizar un proceso de alineamiento entre las dos ontologías (Euzenat y Shvaiko, 2013) que representan el conocimiento de los dominios a integrar. El alineamiento es el proceso de descubrir y explicitar formalmente equivalencias entre clases (conceptos) de dos ontologías distintas.

1.2. Información sobre la Estructura de Productos en sistemas ERP

Los sistemas ERP son ampliamente usados por organizaciones productivas de moderada o gran envergadura. Una de las funcionalidades que provee este tipo de sistemas es el almacenamiento y administración de la información la estructura de productos. La estructura de productos es la información referida a los componentes necesarios para la manufactura de un producto dado.

La mencionada estructura, denominada Bill of Materials (BOM) puede representarse mediante un grafo dirigido acíclico en el que un arco dirigido de *a* hacia *b* indica que *a* es usado para manufacturar *b*. Los productos manufacturados también pueden ser usados para producir otros productos, es decir, pueden ser Ensamblados Intermedios, a diferencia de la Materia Prima, las cuales generalmente no se producen dentro de la organización sino que se adquieren a terceros. Todo arco de *a* hacia *b* debe además contener la información del número de unidades de *a* que se requieren para producir una unidad del producto *b*, además debe incluir la información acerca de la unidad de medida en que está expresada (Vegetti, 2007).

La Figura 1 ilustra los componentes necesarios para producir P1, y como los distintos ensamblados pueden, a su vez, requerir determinados componentes para su producción, a esto se lo conoce como BOM multinivel.

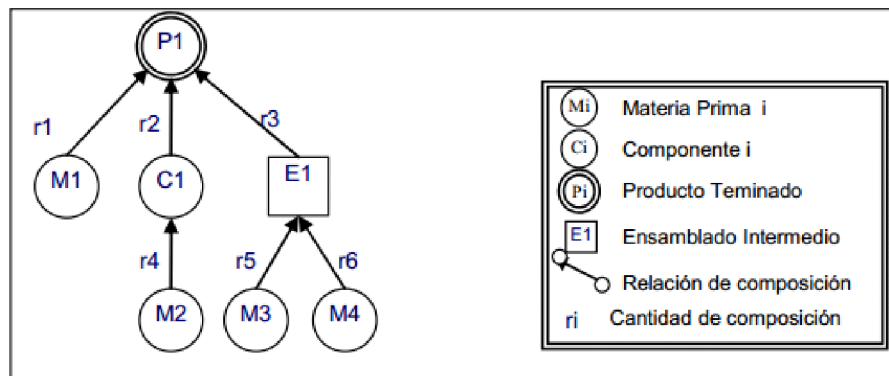


Figura 1. Representación gráfica de un BOM multinivel.

Muchas organizaciones manufacturan un gran número de productos muy similares con ligeras variaciones, y muchos sistemas gestionan dichos productos y sus BOMs asociadas como productos distintos, lo que puede derivar en problemas como duplicación de la información e inconsistencias, entre otros.

1.3. Objetivos

Como se menciona al inicio, la problemática de la interoperabilidad semántica supone un desafío fundamental para el desarrollo de la Web Semántica. El objetivo de este proyecto es lograr hacer interoperar distintos sistemas (que gestionan la información de los BOM de manera distinta). Para esto, se llevará a cabo la confección de ontologías locales a partir de estos modelos, y posteriormente un proceso de alineamiento entre estas ontologías. Este trabajo constituye la primera etapa del proyecto.

En este trabajo se propone una metodología para la utilización de D2RQ y Protegé para obtener de manera semi-automática una ontología OWL a partir de la base de datos de OpenERP, un sistema ERP de código abierto.

2. Metodología

La Fig. 2 resume los pasos de la metodología seguida, los cuales se introducen brevemente en esta sección.

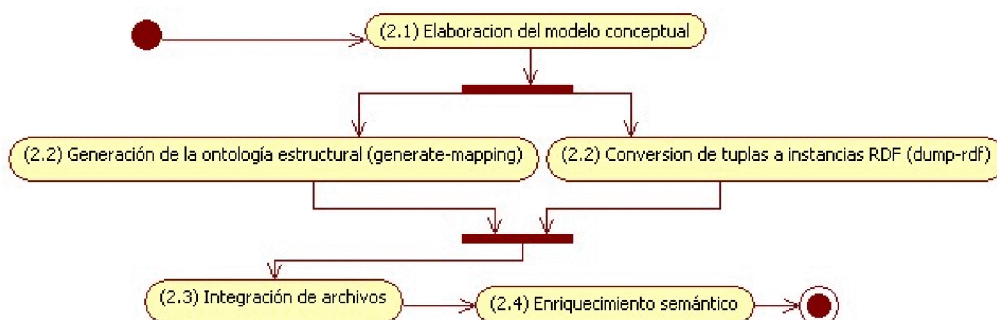


Figura 2. Diagrama de actividad de la metodología a seguir.

2.1. Análisis de OpenERP y elaboración de un modelo conceptual

Como primer paso se decidió llevar a cabo un análisis de OpenERP para dilucidar como gestiona la información referida a la estructura de productos. Para esto se analizaron las tablas correspondientes de la base de datos del sistema. Cabe destacar que OpenERP no soporta variantes de productos de manera nativa, sino mediante la instalación del modulo *product_variant_multi*. Una vez instalado dicho modulo, se procedió al análisis de la estructura de la bases de datos, obteniéndose como resultado un modelo conceptual que será descrito en la sección 3.

2.2. Uso de D2RQ como herramienta de mapeo de Base de Datos relacional a ontologías

La siguiente etapa es la transformación, o mapeo, de la base de datos del sistema a formatos más cercanos a los utilizados en las aplicaciones de la web semántica. Muchas herramientas de transformación de modelo relacional a ontología se basan en un enfoque básico (Spanos et al., 2010) de transformación a partir de las siguientes reglas, identificadas como *Direct Mapping* por el W3C (Arenas et al., 2011):

- Una tabla de la base de datos se transforma en una clase.
- Un campo de una tabla se transforma en una propiedad *datatype* (es decir, que relaciona un individuo con un objeto literal perteneciente a un tipo de datos

definido, equivalente al tipo de datos del campo mapeado) de la clase que representa dicha tabla.

- Un atributo que cuente con una restricción de clave foránea se transforma en una propiedad de objeto (propiedad que relaciona un individuo de la clase que representa la tabla, con un individuo de la clase que corresponde a la tabla a la cual la clave foránea hace referencia).

Por su amplio uso y respaldo de la comunidad se eligió D2RQ como herramienta para llevar a cabo el mapeo. D2RQ es un sistema que permite acceder a bases de datos relacionales como grafos RDF virtuales (Cyganiak et al., 2012). D2RQ realiza las transformaciones en función de reglas de mapeo definidas en un *mapping-file*, el mismo se encuentra codificado en *D2RQ Mapping Language*. Se utilizó el módulo *generate-mapping* para obtener automáticamente un archivo de reglas a partir del subconjunto de tablas de la base de datos que son relevantes al manejo de BOMs. El sistema permite que esta generación automática se haga mediante Direct Mapping, o mediante un algoritmo propio de D2RQ. Ésta última opción detecta ciertos artefactos propios del modelo relacional (como las tablas que se usan para representar relaciones m-a-n), que no representan conceptos del dominio, y no deberían ser mapeados como clases. El mapeo realizado por D2RQ transforma estas tablas intermedias en propiedades de objeto.

El módulo *generate-mapping* tiene dos posibles salidas: La primera salida es un archivo ("*mapping-file*") intermedio, para usarse a la hora de acceder a la base de datos como un grafo RDF; la segunda salida no produce un archivo intermedio, sino que directamente genera un vocabulario RDFS que puede ser interpretado como una ontología estructural ("*schema ontology*"), la cual refleja la estructura de la base de datos. Ésta ontología es producto de las transformaciones antes mencionadas, donde cada propiedad cuenta además con dominio y rango definidos. La ontología estructural no incluye el contenido de la base de datos (las tuplas).

Para obtener la transformación de las tuplas se debe hacer uso del módulo *dump-rdf*, el mismo, a partir de las reglas de transformación definidas en un *mapping-file* dado, devuelve un grafo RDF en el cual cada individuo representa una tupla. Estos individuos están codificados como recursos RDF, y si bien poseen todos los valores de la tupla, con sus propiedades correspondientes, las mismas son interpretadas como *annotations*, en otras palabras, no poseen valor semántico (a diferencia de la ontología estructural obtenida previamente), por ende, para tener un modelo que considere tanto instancias como información estructural es necesario llevar a cabo un proceso de integración de ambos archivos.

2.3 Integración del esquema de la ontología estructural con el archivo de instancias

En la etapa anterior se llevó a cabo la transformación de la información de un modelo relacional a formatos más cercanos a los adoptados para la Web Semántica, sin embargo, las salidas obtenidas no son óptimas para las aplicaciones semánticas deseadas, por lo tanto, se llevó a cabo un proceso de refinamiento de las salidas obtenidas en la etapa anterior.

Como primer paso antes de abordar otros aspectos del refinamiento, es prioritario integrar el archivo de instancias RDF con la ontología estructural, para esto se utilizaron herramientas de procesamiento de textos que hacen uso de expresiones regulares para realizar las siguientes manipulaciones a los archivos:



Identificación del Trabajo	
Área:	Electrónica, informática y comunicaciones
Categoría:	Alumno
Regional:	Facultad Regional Santa Fe

- Eliminación de prefijos, declaraciones de propiedades y declaraciones de clases en el archivo de instancias.
- Concatenación del archivo resultante a la ontología estructural, unificación de todas las tripletas bajo un mismo URI (Uniform Resource Identifier) base.

2.4. Enriquecimiento semántico de la ontología obtenida

El objetivo de esta etapa es especificar formalmente distintos aspectos del dominio que no fueron considerados en la etapa de transformación. El primero de estos aspectos son las características de las propiedades.

Se dice que una propiedad es funcional si dado un individuo, puede haber como máximo un individuo que esté relacionado al primero mediante la propiedad dada. Esta característica resulta útil para modelar las relaciones de integridad referencial de las claves foráneas, y también para expresar que en una base de datos normalizada no pueden existir atributos multivaluados. Teniendo esto en mente, todas las propiedades *datatype* se definieron como funcionales. También se definieron como funcionales todas las propiedades de objeto que representan relaciones 1-a-1 o 1-a-n. Para las propiedades de objeto se consideraron también las propiedades de asimetría e irreflexividad. Resulta útil también definir propiedades inversas a algunas propiedades significativas del dominio, tarea que resulta sencilla si se hace uso de Protegé y un *reasoner*.

El siguiente paso a realizar fue la conversión a clases definidas. Una clase es definida cuando, si un individuo satisface un cierto conjunto de restricciones especificadas en la clase, es *suficiente* para garantizar que pertenece a dicha clase. Las restricciones de una clase definida deben ser aquellas que estén definidas sobre propiedades que representan campos obligatorios en una tabla dada, de manera que los campos que puedan o no estar presentes, se definan en las subclases de la clase definida (es decir, en casos particulares).

3. Resultados y Discusión

Como resultado de la etapa 2.1 se obtuvo el modelo conceptual que se presenta en la Figura 3. En OpenERP, un producto necesariamente tiene una plantilla o "*template*", dicho template especifica las maneras en que los productos pueden variar, a través de sus *Tipos de Dimensión (TD)*, por ejemplo, una plantilla *Automóvil* puede tener un TD *Color*. Cada TD puede tener distintos valores u *Opciones*, por ejemplo, el TD *Color* puede tener como Opciones los valores *Negro*, *Rojo* y *Azul*. Donde cada *template* que posea *Color* como TD podrá tener productos con los valores Negro, Rojo y Azul. Por su parte, un *Producto* es la instancia (o variación) de un *Template*, y cada producto está asociado a una combinación <TD, Opción>, llamada Variante. Nótese que las Opciones están asociadas a un TD, y este puede estar vinculado a distintos *templates*.

OpenERP gestiona las BOMs a nivel de variante de producto, es decir, el producto *Automóvil Azul* tendrá una BOM individual, distinta de *Automóvil Negro*, sin importar que pertenezcan al mismo template, y que varíen solo en el color.

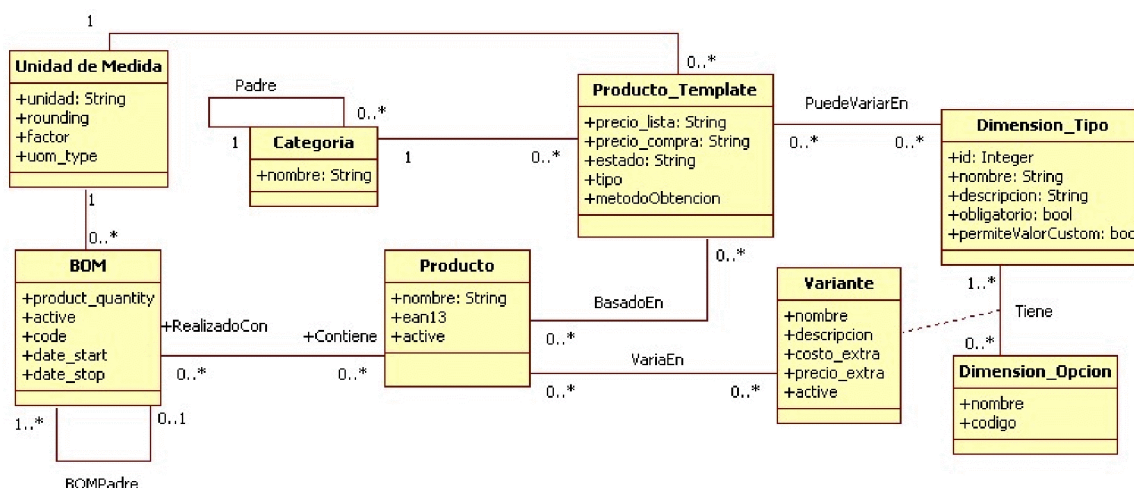


Figura 3. Diagrama de clases obtenido.

La relación BOM-BOM representa que una BOM está definida como mononivel, es decir, una BOM describiendo un producto dado puede tener asociada una BOM padre (de nivel inferior) (Vegetti, 2007). Por otro lado, la clase *Variante* representa la combinación de un TD con su Opción asociada.

La etapa 2.2 produjo dos archivos, cada uno de los cuales, de manera independiente, capturan distintos aspectos de la base de datos (estructural y de instancia). La etapa 2.3 unifica ambos archivos, permitiendo que las propiedades de las instancias no sean interpretadas como *annotations*, sino como propiedades OWL con valor semántico. La etapa 2.3 provee un punto de partida para empezar a generar valor semántico.

Si se analizan los resultados obtenidos en cada etapa, se puede observar que no hay una mejora sustancial (en términos de capacidad semántica) sino hasta el final de la etapa 2.4. La etapa 2.2 acerca la estructura de la base de datos y los contenidos de la tabla a formatos más fácilmente adaptables a la Web Semántica, y la etapa 2.3 produce una única ontología, pero la misma no es más que un reflejo de la base de datos. Es la etapa 2.4 la cual comienza a generar un valor semántico mayor al provisto por el modelo relacional. La Figura 4, presenta una vista parcial de la ontología obtenida como resultado del paso 2.4.

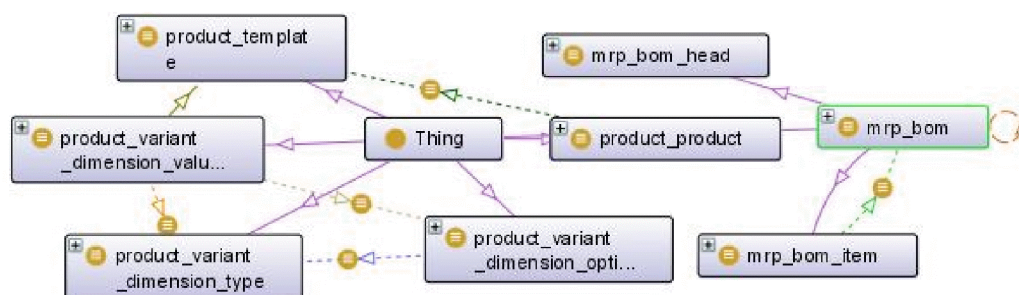


Figura 4. Visualización parcial de la ontología obtenida en la etapa 2.4



Identificación del Trabajo	
Área:	Electrónica, informática y comunicaciones
Categoría:	Alumno
Regional:	Facultad Regional Santa Fe

4. Conclusiones y trabajo a futuro

En función de lo expresado en la sección 3, se puede afirmar que hoy en día, la generación de un modelo semántico a partir de uno relacional posee una complejidad inherente que solo puede ser resuelta por alguien con conocimiento del dominio, y que las herramientas automatizadas apenas pueden extraer un mínimo de información semántica.

Con este trabajo se ha alcanzado un modelo semántico de un primer sistema. El trabajo a futuro consistirá en el posterior refinamiento de este modelo, y en la reiteración de estas etapas para obtener una ontología a partir de otro sistema, teniendo por objetivo final la aplicación de técnicas y herramientas de alineamiento para lograr interoperabilidad semántica entre ambos.

Bibliografía

- Arenas, M., Prud'hommeaux, E., Sequeda, J. (2011). A Direct Mapping of Relational Data to RDF. Web: <http://www.w3.org/TR/2011/WD-rdb-direct-mapping-20110324/>.
- Berners-Lee, T., Hendler, J., Lassila, O. (2001). The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American.
- Cyganiak, R. et al. (2012). Web: <http://d2rq.org/>.
- Euzenat, J., Shvaiko, P. (2013). Introduction. En: Ontology Matching. Springer.
- Hitzler, P., Krotzsch, M., Parsia, B., Patel-Schneider, P. F., Rudolph, S. (2012). OWL 2 Web Ontology Language Primer (Second Edition). Web: <http://www.w3.org/TR/2012/REC-owl2-primer-20121211/>.
- Horrocks, I., Fensel, D., Broekstra, J., Decker, S., Erdmann, M., Goble, C., van Harmelen, F., Klein, M., Staab, S., Studer, R., Motta, E. (2001). The Ontology Inference Layer OIL. Technical Report.
- Panetto, H., Dassisti, M., Tursi, A. (2012). ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment. Advanced Engineering Informatics 26.334–348.
- Spanos, D., Stavrou, P., Mitrou, N. (2010). Bringing Relational Databases into the Semantic Web: A Survey. Semantic Web Journal, 13.
- Staab, S., Studer, R. (2009). What is an Ontology? En: Handbook on Ontologies. Springer.
- Vdovjak R., Houben, G.J. Stuchenschmidt, H., Aerts, A. (2006). RDF and Traditional Query Architectures. En: Semantic Web and Peer-to-Peer. Decentralized Management and Exchange of Knowledge and Information. Springer. 41-58.
- Vegetti, M. (2007). Un modelo integrado para la representación de productos con estructuras complejas.