

Definiendo Familias de Productos Utilizando OWL y SWRL

Nicolas, Padula

Departamento Ingeniería en Sistemas de Información, Universidad Tecnológica Nacional – Facultad Regional Santa Fe (Lavaise 610, Santa Fe)

INGAR – Instituto de Desarrollo y Diseño (CONICET-UTN) (Avellaneda 3657, Santa Fe)
nicolasvpadula@gmail.com

Resumen - La creciente demanda de variedad por parte del mercado obliga a las empresas a replantear sus esquemas productivos para mantener su competitividad. Un diseño y modelado adecuado de familias de productos en función de su estructura, en conjunto con la integración semántica de los sistemas que gestionan información de estructura de productos, pueden proporcionar grandes beneficios a las organizaciones productivas. Este trabajo plantea la definición de un modelo de familias de productos para un catálogo de notebooks utilizando una ontología extraída a partir de la base de datos de un sistema ERP.

Palabras Clave: Familias de productos, BOM, Ontologías, Web Semántica, Razonamiento

Defining Product Families Using OWL and SWRL

Abstract - The increasing demand for variety from the market forces companies to rethink their production plans in order to maintain competitiveness. An adequate product family design and modeling based on product structure, alongside a semantic integration of the systems that manage product structure information, can provide great benefits to productive organizations. This work proposes the definition of a product family model for a notebook catalogue, using an ontology extracted from a ERP system's database.

Keywords: Product family, BOM, Ontologies, Semantic Web, Reasoning

Introducción

La necesidad de satisfacer una demanda creciente por productos cada vez más específicos por parte del mercado ha llevado a las organizaciones a reevaluar sus estrategias de negocio con el fin de mantener su competitividad. Las organizaciones apuntan a lograr una diversificación de su oferta que satisfaga las necesidades de los distintos segmentos del mercado sin perder eficiencia en la producción (Jiao et al., 2007; Berry y Pakes, 2007).

Sin embargo, ambos objetivos entran en conflicto entre sí. Una mayor diversidad de productos conlleva de forma inherente un aumento en la complejidad de producción, y por lo tanto, un aumento del costo. A su vez, una mayor estandarización de los productos resultaría en una incapacidad de satisfacer las demandas más específicas. Una solución eficaz a este conflicto es el diseño de familias de productos (*Product*

Family Design) (Marion et al., 2008; Park y Simpson, 2008).

Una familia de productos se define como un conjunto de productos que comparten características y componentes en común, con variaciones únicas entre sí, para satisfacer necesidades distintas (Meyer y Lehnerd, 1997). El conjunto común es conocido como la plataforma del producto (Simpson et al., 2006), la plataforma es el núcleo común de las familias de producto. Son conjuntos de componentes, tecnologías, subsistemas, y procesos que forman una estructura común de la cual se pueden derivar conjuntos de productos (Li et al., 2007).

Una plataforma correctamente diseñada permite la manufactura eficiente de los componentes comunes de una familia (Fellini et al., 2006), aumenta la flexibilidad y reduce la complejidad y puede incrementar la reusabilidad y adaptabilidad del esquema productivo (Simpson et al., 2006). El diseño de plataformas se enfoca en maximizar las partes comunes (*commonality*),

mediante la identificación de los aspectos a ser compartidos entre los productos de una familia, y las distintas configuraciones de estos aspectos entre los individuos.

Un diseño adecuado de una familia de productos debe reconciliar aspectos opuestos: Los aspectos externos, que provienen de las necesidades de los clientes y el mercado, y generalmente apuntan a una mayor variedad de funciones y productos; y por otro lado los aspectos internos a la organización, referidos a la factibilidad técnica y económica de la producción, la complejidad logística y de manufactura, y la gestión misma de la organización, todos aspectos que se ven beneficiados ante la estandarización e integración de procesos y componentes.

Debido a la necesidad de encontrar un equilibrio óptimo entre partes comunes y diversidad, y a la cantidad de factores y variables a considerar, el diseño de familias de producto es un problema esencialmente complejo.

Como se expuso anteriormente, existe un gran abanico de criterios con los que se pueden definir familias de productos. Una correcta definición de familias de productos en función de la estructura de los mismos es de especial importancia para la optimización de los procesos de manufactura, sobre todo en ambientes industriales que implementen una cadena de suministro integrada, algo usual en el entorno competitivo actual (Panetto et al., 2007).

La estructura de productos o BOM (Bill of Materials) define las relaciones de composición entre productos. Una BOM puede representarse como un grafo dirigido acíclico en el cual un arco desde *a* hacia *b* indica que *a* compone a *b*, y el peso del arco representa la cantidad de *a* usada para producir *b*.

La información de la estructura de productos suele almacenarse de manera dispersa en sistemas que, al integrarse las cadenas de suministro, resultan ser heterogéneos entre sí. (Vdovjak et al., 2006). Para subsanar el problema de interoperabilidad de estos sistemas heterogéneos, se plantea la necesidad de una integración a nivel semántico (Horrocks et al., 2001). Desde hace unos años las ontologías son vistas como una solución a este problema de integración semántica (Berners-Lee et al., 2001).

Ontologías y tecnologías semánticas

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.
- **Propiedades:** Relaciones a través de las cuales los individuos pueden asociarse.
- **Clases:** Conjuntos de individuos. Se pueden definir formalmente requisitos de pertenencia a una clase dada. Los individuos que cumplan dichos requisitos pertenecerán al conjunto denotado por la clase.

Las ontologías se codifican en lenguajes formales, generalmente basados en lógica de primer orden o lógica descriptiva. Gracias a esto es posible inferir nuevo conocimiento a partir de los axiomas presentes en una ontología mediante el uso de sistemas denominados razonadores. 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).

Para lograr una mayor capacidad expresiva, OWL puede complementarse usando SWRL (Semantic Web Rule Language), un lenguaje de reglas basado en lógica de Horn, recomendado por el W3C para su uso en conjunto con OWL. SWRL permite especificar reglas del tipo *A implica B*, donde A y B son uno o más predicados unidos por el AND lógico. Los predicados en SWRL pueden ser unarios o binarios, los cuales representan clases y propiedades de OWL, respectivamente.

El gran campo de aplicación de las ontologías para la gestión e integración de conocimiento, la mejora de la interoperabilidad, y el enriquecimiento semántico, hacen de las mismas de gran utilidad para la conceptualización, y soporte de modelos de familias de producto en ambientes industriales.

Este trabajo propone una metodología para la definición de un modelo de familias de productos basado en la estructura de los mismos, utilizando tecnologías semánticas, así como también un caso de estudio aplicando dicha metodología.

Se decidió partir de una situación usual en muchas organizaciones productivas, en la cual la información de los productos se encuentra almacenada en un sistema ERP tradicional, es decir, en el cual cada variante de producto se almacena como un producto distinto. Esto se traduce en una situación de duplicación de datos,

lo cual trae aparejado una mayor dificultad de mantenimiento. Para esto se usó el sistema de código abierto OpenERP. El desarrollo de la ontología se llevó a cabo utilizando el editor Protege. El catálogo de notebooks y componentes a modelar fue tomado de la sección "Product Configuration" del sitio web SPLOT (<http://www.splot-research.org>).

Metodología

La metodología seguida puede dividirse en dos etapas: La etapa de extracción o transformación, en la cual se realiza un proceso de mapeo de la base de datos relacional de OpenERP para llegar a una ontología, y la etapa de definición de familias de producto, en la cual se conceptualizan y definen las familias partiendo de la ontología base obtenida en la primera etapa.

Extracción de la ontología a partir de la base de datos de OpenERP

Esta etapa puede a su vez dividirse en 4 sub-etapas como ilustra la Figura 1. Los pasos seguidos en esta etapa fueron abordados en detalle en trabajos anteriores (Padula, 2014a; Padula 2014b) de manera que de aquí en adelante se detallarán solo los aspectos considerados esenciales.

Elaboración del modelo conceptual

Como primer paso se decidió llevar a cabo un análisis del funcionamiento y la estructura de OpenERP para comprender como gestiona la información referida a la estructura de productos. Para esto se analizaron las tablas correspondientes de la base de datos del sistema. Si bien inicialmente el análisis se realizó

considerando el módulo *product_variant_multi*, que permite una gestión básica de familias de productos, posteriormente se decidió descartar esto y tener en cuenta solo el modelo plano de la estructura de productos, puesto que el común de las organizaciones no posee gestión de variantes de ningún tipo.

Mapeo de la base de datos relacional a una ontología

El siguiente paso es la transformación de la base de datos del sistema a formatos más cercanos a los utilizados en el ámbito 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:

- Una tabla de la base de datos se transforma en una clase.
- Un campo de una tabla se transforma en una propiedad de datos de la clase que representa dicha tabla.
- Un campo 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 mapeada 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 base de datos relacionales como grafos RDF virtuales. Mediante reglas de mapeo similares a las anteriores, D2RQ permite obtener dos salidas: una ontología estructural, que refleja el esquema de la base de datos, y un

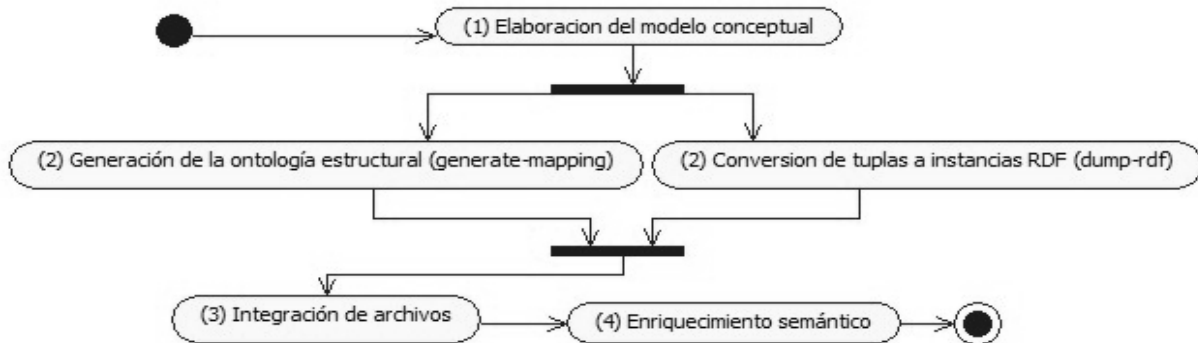


Figura 1. Metodología aplicada

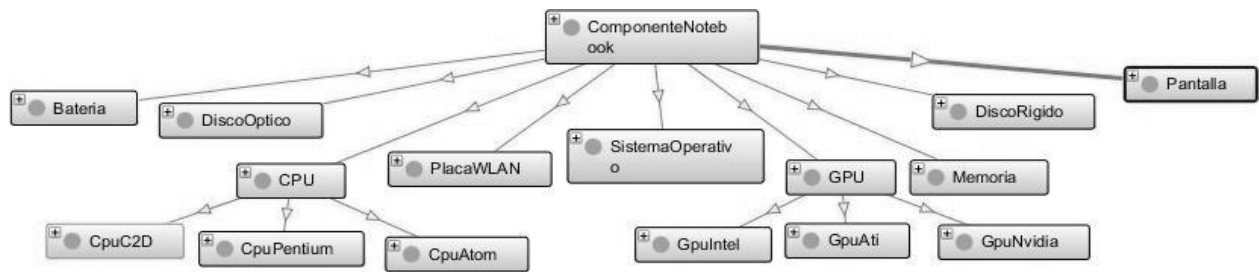


Figura 2. Vista parcial de la jerarquía de *ComponenteNotebook*.

archivo de instancias RDF que contiene la información de los registros de la base de datos.

Puesto que las instancias RDF no poseen un valor semántico significativo, para obtener un modelo que considere tanto instancias como información estructural, es necesario llevar a cabo un proceso de integración de ambos archivos.

Integración de la ontología estructural con el archivo de instancias

El proceso de integración de los dos archivos obtenidos en el paso anterior se llevó a cabo mediante la manipulación de los archivos como texto plano a través de expresiones regulares. De esta forma se realizaron las siguientes modificaciones:

- 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 espacio de nombres.

Enriquecimiento semántico de la ontología obtenida

El objetivo de este paso es especificar formalmente distintos aspectos del dominio que no fueron considerados en la etapa anterior. Esto consiste en:

- Formalización de las disyunciones entre clases.
- Formalización de las características de las propiedades. En este contexto todas las propiedades son consideradas funcionales porque modelan restricciones de integridad referencial, y porque en una base de datos normalizada no pueden existir atributos multivaluados. Para las propiedades de objeto también se tomó en cuenta la asimetría y la irreflexividad de las mismas.

- Definición de propiedades inversas.
- Formalización de axiomas para la conversión a clases definidas (clases que establecen condiciones necesarias y suficientes que los individuos deben satisfacer para pertenecer a dicha clase).

Definición de familias de productos

La etapa de transformación provee una ontología base que representa el dominio de la información de la estructura de productos. La etapa de definición de familias de productos tiene como objetivo extender esta ontología para dar soporte a un modelo de familias de productos para el caso de estudio planteado. Esta ontología permitirá clasificar, mediante el uso de un razonador, individuos en distintas familias en función de sus relaciones de composición.

El razonador usado a lo largo de este trabajo es Pellet, dado que es el único que da soporte a los tipos de datos mapeados desde la base de datos, además de contar un amplio soporte de los constructores de OWL, y un amplio uso y respaldo por parte de la comunidad.

Como se mencionó en la introducción, diseñar adecuadamente es una actividad compleja que requiere la consideración de múltiples factores. El caso de estudio planteado parte de una familia de productos existente (un catálogo de notebooks), de manera que la conceptualización de las familias es sencilla y puede extraerse directamente de la observación del catálogo.

Teniendo esto en cuenta, la etapa de definición de familias de productos se centra en la aplicación e implementación de tecnologías semánticas para dar soporte a las familias de productos ya definidas en el catálogo.

El enfoque de gestión de variantes que se aplica en este trabajo es el de *BOM Genérico* (GBOM) (Van Veen y Wortmann, 1992). El mismo consiste en definir estructuras de productos genéricos, con componentes genéricos, los

cuales se reemplazarán por productos y componentes concretos al momento de especificar una variante puntual. Cada componente genérico representa el conjunto de productos concretos que pertenece a la familia denotada por el mismo. Dicho componente genérico puede ser reemplazado por cualquier producto concreto, y cada reemplazo equivale a una variante distinta de la familia representada por el producto genérico.

Considerando este enfoque, se procedió a la definición de las familias en la ontología. En primera instancia se definieron las clases que representan a las familias y subfamilias de componentes, partiendo del listado de componentes de SPLOT. Todo esto se agrupó bajo una clase general denominada *ComponenteNotebook*. En la Figura 2 puede observarse una vista parcial de la taxonomía descripta.

El paso siguiente es agregar a los individuos que representen a los componentes los axiomas lógicos que especifiquen que dicho individuo pertenece a una determinada clase. Por simplicidad, este proceso se realizó de manera manual, aunque podría considerarse una solución automática, basada en, por ejemplo, la categoría en la que estén almacenados los productos dentro de OpenERP.

El siguiente paso es definir las GBOMs de las notebooks. Para esto se procedió partiendo de la familia más general a las más específicas, comenzando por definir la BOM de una notebook (*BomNotebook*). Una BOM (*mrp_bom*) será una *BomNotebook* si la misma contiene los componentes propios de una notebook (CPU, GPU, Disco Rígido, etc). Debe notarse que los componentes opcionales (por ejemplo, los dispositivos ópticos) son especificados en las subclases de las familias que los contengan, ya que sería incorrecto decir que toda notebook tiene dispositivo óptico. La definición de la clase se puede realizar utilizando directamente las propiedades definidas en la etapa de transformación, pero por razones de semántica, mantenibilidad, y rendimiento, se decidió definir nuevas propiedades que representen la relación de composición entre una BOM y un Componente. Para esto se definió la relación *bomTieneComponente*, la cual establece que un componente forma parte de una estructura dada. Esta relación se instanció a través de la regla SWRL que se muestra en la Figura 3.

```
ComponenteNotebook(?C),
mrp_bom_product_id(?BC, ?C),
mrp_bom_bom_inv_id(?B, ?BC) ->
bomTieneComponente(?B, ?C)
```

Figura 3. Regla para la instanciación de la propiedad *bomTieneComponente*

Esta regla puede leerse como “si C es un *ComponenteNotebook*, y C forma parte de una *mrp_bom* B, entonces B tiene como componente a C”. Partiendo de esta relación, se pueden definir relaciones análogas para las distintas subclases de *ComponenteNotebook* las cuales pueden ser instanciadas con reglas SWRL similares.

La Figura 4 muestra la regla usada para instanciar la relación *bomTieneCpu*. Esta regla puede leerse como “si C es un *CPU* y B es una *mrp_bom* que tiene como componente a C, entonces B tiene un *CPU*”.

Una vez definidas estas reglas y relaciones, se puede definir las familias y subfamilias de notebooks presentes en el catálogo. Las familias se definieron, en primera instancia según la línea de notebooks (Inspiron, Studio, Studio XPS o Netbook), luego cada familia fue subdividida acorde al tamaño de pantalla. Las distintas familias y subfamilias tendrán distintas configuraciones posibles de componentes, es decir, distintos GBOMs.

```
CPU(?C), bomTieneComponente(?B, ?C) ->
bomTieneCpu(?B, ?C)
```

Figura 4. Regla para la instanciación de la relación *bomTieneCpu*.

Las definiciones realizadas restringen el rango de las propiedades que un individuo debe satisfacer para pertenecer a una clase dada, esto se denomina *closure axiom* o axioma de cierre. El axioma de cierre establece que un individuo pertenecerá a la clase en cuestión si tiene una propiedad determinada (cuantificador existencial *some*), y esa propiedad tiene como rango únicamente una clase dada (cuantificador universal *only*). Teniendo en cuenta esto, es necesario recordar que el razonamiento en las ontologías se basa en el concepto de *Open World*

Assumption, es decir, se supone que puede haber más información de la que esta explicitada. Por lo tanto, la única forma de expresar que algo es falso, es hacerlo explícitamente mediante algún axioma.

Debido a esto, los individuos de la ontología no pueden satisfacer la restricción universal del axioma de cierre, a menos que se defina explícitamente que las propiedades que poseen son las únicas que existen. Esto tiene sentido a nivel conceptual, si se tiene en cuenta que esta ontología y todos sus individuos fueron mapeados inicialmente desde una base de datos relacional, es decir, de un modelo de conocimiento basado en *Closed World Assumption* (la información disponible es la única que existe).

Para llevar a cabo esto se utilizaron reglas SWRL de la forma: $(\text{propiedad some Clase})(?X) \rightarrow (\text{propiedad only Clase})(?X)$. En términos coloquiales: “Si X pertenece al conjunto de elementos que se relacionan con Clase a través de propiedad, entonces también pertenece al conjunto de elementos que se relacionan a través de propiedad **únicamente** con Clase”. Se definieron reglas de este tipo para todas las propiedades y clases utilizadas en la definición de familias de *BomNotebook* y sus subfamilias. Puede observarse un ejemplo de este tipo de reglas en la Figura 5.

```
{(bomTieneCpu some
(CpuC2DT or CpuCeleron or CpuPentium))(?X)
-> (bomTieneCpu only
(CpuC2DT or CpuCeleron or CpuPentium))(?X)}
```

Figura 5. Ejemplo de regla que restringe las propiedades de los individuos.

Hasta este punto, la clasificación se hace sobre la clase *mrp_bom*, en otras palabras, lo que se está clasificando son estructuras de producto, y no productos en sí. Para obtener una clasificación de productos, se puede partir de la siguiente noción: “Una notebook es todo producto que tenga la BOM de una notebook”. Esto puede ser expresado de tres formas distintas (aunque lógicamente equivalentes), con diferencias de rendimiento que serán expuestas en la sección de resultados.

La primera forma (*Razonamiento Directo*) es la más usual. Consiste en agregar axiomas a las clases que representen las familias de notebooks para convertirlas en clases definidas, y luego utilizar un razonador para realizar la clasificación.

En la segunda forma (*Razonamiento Segmentado*), primero se utiliza el razonador para clasificar las clases referidas a las BOMs, luego se almacenan todas las inferencias obtenidas, se agregan los axiomas mencionados anteriormente, y se vuelve a clasificar la ontología. Por simplicidad, la segmentación se realizó manualmente utilizando Protege, pero en el ámbito de un sistema de información basado en ontologías, el almacenamiento de inferencias y la carga dinámica de axiomas puede implementarse haciendo uso de herramientas para desarrollo basado en ontologías.

La tercera forma es mediante el uso de reglas SWRL para realizar la clasificación. Puede verse un ejemplo de estas reglas en la Figura 6. Esta regla se puede interpretar como “Si B es una *BomNotebook*, y B es la BOM de un producto P, entonces P es una *NotebookDell*”.

Se realizaron pruebas con el mismo hardware y la misma cantidad de individuos para comparar el tiempo que tarda cada solución en realizar la clasificación de la ontología. Las pruebas se realizaron en una PC de gama media (CPU Intel i5 con 4 *threads* de procesamiento y 4gb de RAM), clasificando 506 individuos, incluyendo todos los componentes del catálogo, y tres notebooks específicas cargadas. Además, se incluyeron varios individuos que no entran en la clasificación de notebooks.

Los resultados obtenidos se presentan en la siguiente sección.

```
BomNotebook(?B), mrp_bom_product_id(?B, ?P)
-> NotebookDell(?P)
```

Figura 6. Regla que establece que el individuo P pertenece a la clase *NotebookDell*.

Resultados

De los resultados parciales de la etapa de transformación se detallarán aquellos considerados importantes para la comprensión del resto del trabajo, obviando los que estén asociados a manejo de archivos RDF a bajo nivel o al funcionamiento interno de OpenERP, aspectos que fueron abordados en trabajos anteriores (Padula, 2014a; Padula, 2014b).

Al finalizar la etapa de transformación, se obtiene una ontología base a partir de la cual pueden definirse las familias de productos contempladas en el catálogo de SPLOT. La

Figura 7 presenta una vista parcial de ésta ontología

A partir de esta ontología base, se obtuvo una ontología que provee soporte a las familias de productos del catálogo mediante la aplicación de los pasos de la etapa de definición de familias de productos. En la Figura 8 se presenta una vista parcial de las clases más relevantes de la ontología de familias de producto.

Una vez definidas las familias según las tres alternativas presentadas en la sección anterior, se puede utilizar la ontología para clasificar individuos haciendo uso de un razonador, logrando deducir a que familia pertenecen en función de su estructura. Si bien las tres soluciones propuestas expresan lo mismo en términos de lógica, hay una gran diferencia de rendimiento a la hora de razonar, como se aprecia en la Tabla 1.

Tabla 1. Rendimiento de las soluciones propuestas

Solución	Tiempo de clasificación [ms]
Razonamiento Directo	∞ (No finaliza)
Razonamiento Segmentado	170888
OWL+SWRL	25795

Discusión

Una posible razón por la que el Razonamiento Segmentado resulte en una mejora respecto al Razonamiento Directo es que la “pre-clasificación” no solo ahorre tiempo de CPU, sino que además resulte en un menor consumo de memoria RAM, un problema común en gran cantidad de herramientas para el desarrollo de sistemas basados en ontologías (Martínez y Sosa, 2014; Martínez et al., 2014).

Una variación de rendimiento tan significativa bajo condiciones parecidas no es un detalle menor. El razonamiento semántico es un

problema complejo y, a pesar de que existen propuestas de predicción del rendimiento (Kang et al., 2012; Kang et al., 2014) o de análisis del rendimiento para la optimización del mismo (*profiling*) (Wang y Parsia, 2007), en muchas ocasiones el desarrollador de la ontología tiene que valerse fundamentalmente de su propia intuición para poder realizar una depuración y optimización de la misma. Esto muestra que existe una discrepancia entre las posibilidades teóricas y las herramientas disponibles para el usuario final, a la hora de desarrollar sistemas de información basados en ontologías.

El trabajo a futuro estará enfocado en la realización de pruebas de rendimiento más intensivas (con mayor cantidad de individuos) para verificar la escalabilidad de las soluciones propuestas. Por otro lado, se intentará generalizar este modelo para cualquier conjunto de productos, considerando características más avanzadas para la configuración de las variantes, como las restricciones de incompatibilidad/obligatoriedad, de manera de poder obtener un mecanismo de configuración que permita excluir variantes inválidas (Olsen et al., 1997; Hegge, 1995) o las estructuras de producto híbridas (basadas en composición y descomposición) (Vegetti, 2007), manteniendo el foco en la aplicabilidad del mismo y en su capacidad de integración semántica.

Conclusiones

Con este trabajo se logró definir un modelo de familia de productos para un conjunto particular y acotado de productos y componentes, partiendo de una ontología base obtenida de un proceso de transformación de una base de datos relacional. Además, se propusieron y analizaron diversas implementaciones con distintos rendimientos, y se consideraron distintas líneas de investigación a futuro que resultan prometedoras.

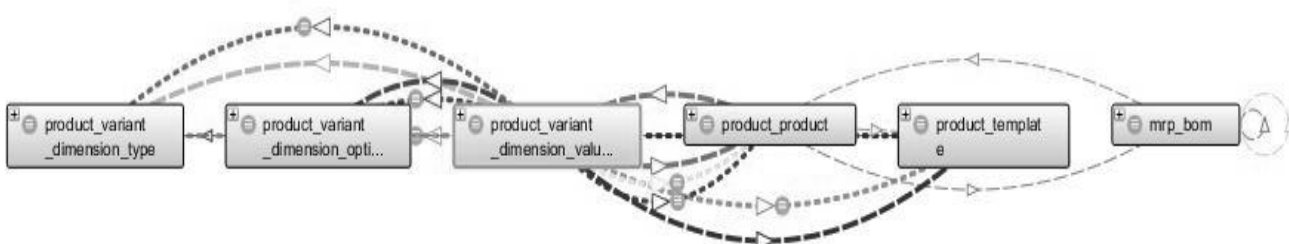


Figura 7. Vista parcial de la ontología obtenida en la etapa de transformación.

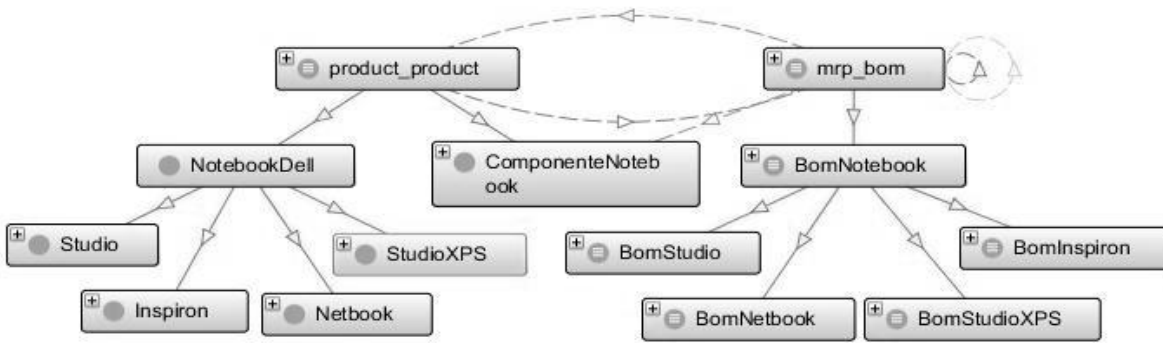


Figura 8. Taxonomía de las familias de productos.

Agradecimientos

Se agradece el apoyo brindado por la Universidad Tecnológica Nacional (PID 25-O156).

Referencias

Jiao, J., Simpson, T., Siddique, Z. "Product family design and platform-based product development: a state-of-the-art review", *Journal of Intelligent Manufacturing*, 18(1), 5-29. (2007).

Berry, S., Pakes, A. "The Pure Characteristics Demand Model", *International Economic Review*; 48(4), 1193-1225. (2007).

Marion, T. J., Thevenot, H. J., Simpson, T. W. "A cost-based methodology for evaluating product platform commonality sourcing decisions with two examples". *International Journal of Production Research*, 45(22), 5285-5308. (2008).

Park, J., Simpson, T. W. "Toward an activity-based costing system for product families and product platforms in the early stages of development". *International Journal of Production Research*, 45(1), 99-130. (2008).

Meyer, M., Lehnerd, A. P. "The power of product platforms – building value and cost leadership", Simon and Schuster, EEUU, (1997).

Simpson, T., Siddique, Z., Jiao, J. "Product Platform and Product Family Design", Springer Science+ Business Media, EEUU, (2006).

Li, L., Huang, G. Q., Newman, S. T. "Interweaving genetic programming and genetic algorithm for structural and parametric optimization in adaptive platform product customization". *Robotics and Computer-Integrated Manufacturing*, 23(6), 650-658. (2007).

Fellini, R., Kokkolaras, M., Papalambros, P. Y. "Quantitative platform selection in optimal design

of product families, with application to automotive engine design", *Journal of Engineering Design*, 17(5), 429-446. (2006).

Panetto, H., Dassisti, M., Tursi, A. "ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment". *Advanced Engineering Informatics*, 36(2), 334-348. (2007).

Vdovjak R., Houben, G. J., Stuchenschmidt, H., Aerts, A. *Semantic Web and Peer-to-Peer Decentralized Management and Exchange of Knowledge and Information*, Springer-Verlag Berlin Heidelberg, Berlin, Alemania, 41-58, (2006).

Horrocks, I., Fengsel, D., Broekstra, J., Decker, S., Erdmann, M., Goble, C., van Harmelen, F., Klein, M., Staab, S., Studer, R., Motta, E. "The Ontology Inference Layer OIL". (2001).

Berners-Lee, T., Hendler, J., Lassila, O. "The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities", *Scientific American*; 284(5), 1-5. (2001).

Padula, N. "Aplicación de tecnologías semánticas para la integración de información de estructura de productos almacenada en sistemas ERP"., *Actas de las Jornadas de Jóvenes Investigadores Tecnológicos 2014*. Rosario, Argentina, 6 de Noviembre, (2014a).

Padula, N. "Extracción de una ontología OWL a partir de un sistema ERP"., *Actas del 2º Congreso Nacional de Ingeniería Informática/Sistemas de Información*. San Luis, Argentina, 13-14 de Noviembre, (2014b).

Spanos, D., Stavrou, P., Mitrou, N. "Bringing Relational Databases into the Semantic Web: A Survey". *Semantic Web Journal*. 3(2), 169-209, (2010).

Van Veen, E. A., Wortmann, J. C. "Generative Bill of Materials Processing Systems"., *Production Planning & Control*, 3(3), 314-326. (1992).

Martínez, A., Sosa, S., Reynares, E., Caliusco, M. L. "Implementación de Sistemas de Información Basados en Ontologías: Análisis de Tecnologías"., *Actas del 2° Congreso Nacional de Ingeniería Informática/Sistemas de Información*. San Luis, Argentina, 13-14 de Noviembre, 1134-1138, (2014).

Martínez, J. A., Sosa, S. G. "Implementación de tecnologías Triplestore para soporte de sistemas basados en ontologías.", *Actas de las Jornadas de Jóvenes Investigadores Tecnológicos 2014*, Rosario, Argentina, 6 de Noviembre, (2014).

Kang, Y. B., Li, Y. F., Krishnaswamy, S. "Predicting Reasoning Performance Using Ontology Metrics"., *Proceedings of the 11th International conference on The Semantic Web*, Boston, EEUU, 11-15 de Noviembre, 198-214. (2012).

Kang, Y. B., Pan, J. Z., Krishnaswamy, S., Sawangphol, W., Li, Y. F. "How Long Will It Take? Accurate Prediction of Ontology Reasoning Performance", *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, Quebec, Canada, 27-31. (2014).

Wang, T. D., Parsia, B. "Ontology Performance Profiling and Model Examination: First Steps". *The Semantic Web, ISWC 2007 + ASWC 2007*, Proceedings. Busan, Korea, 11-15 de Noviembre, 595-608. (2007).

Olsen, K. A., Saetre, P., Thorstenson, A. "A Procedure-Oriented Generic Bill of Materials"., *Computers and Industrial Engineering*, 32(1), 29-45. (1997).

Hegge, H. "Intelligent Product Family Descriptions for Business Applications: production control software based upon generic bills-of-material in an assemble-to-order/make-to-stock environment", (Tesis Doctoral), Technische Universiteit Eindhoven. Eindhoven, Holanda. (1995).

Staab, S., Studer, R. "Handbook on Ontologies". Springer Science & Business Media, EEUU, (2009).

Vegetti, M. "Un modelo integrado para la representación de productos con estructuras complejas". (Tesis Doctoral), Universidad Tecnológica Nacional - Facultad Regional Santa Fe, Santa Fe, Argentina, (2007).