

Hacia la Recomendación Automática de Patrones de Diseño Ontológico

Tomás Quiñonez¹, Christian Gimenez¹, Laura Cecchi¹, and Pablo Fillottrani^{2,3}
tomas.quinonez@est.fi.uncoma.edu.ar
{christian.gimenez,lcecchi}@fi.uncoma.edu.ar
prf@cs.uns.edu.ar

¹ Grupo de Investigación en Lenguajes e Inteligencia Artificial
Facultad de Informática
UNIVERSIDAD NACIONAL DEL COMAHUE
Neuquén, Argentina

² Laboratorio de I&D en Ingeniería de Software y Sistemas de Información - Departamento de Ciencias e Ingeniería de la Computación
UNIVERSIDAD NACIONAL DEL SUR, Bahía Blanca, Argentina

³ Comisión de Investigaciones Científicas de la provincia de Buenos Aires (CIC), Buenos Aires, Argentina

Resumen En la Ingeniería Ontológica, los modeladores que desean construir una ontología por medio del reuso de patrones poseen poca asistencia en las herramientas de desarrollo. Por ello, se propone una metodología que permita recomendar a los usuarios Patrones de Diseño Ontológico (ODP) para una ontología en etapa de diseño.

La metodología propuesta posee dos etapas: el análisis del patrón para extraer información relevante para detectarlos parcial o totalmente en una ontología, y el análisis de la ontología del usuario para buscar y sugerir patrones.

De esta manera, se presenta una metodología implementable para sugerir patrones a medida que es diseñada una ontología. Así, los modeladores sin conocimiento previo de los ODP, podrán seleccionar patrones ampliamente aceptados mejorando el proceso de desarrollo y la calidad.

Palabras Clave: Ingeniería de Software basada en Conocimiento, Patrones de Diseño Ontológico, Ontologías, Lógicas Descriptivas.

1. Introducción

La Ingeniería Ontológica estudia los métodos y metodologías que guían a los modeladores en el diseño, desarrollo, implementación, mantenimiento, uso y publicación de ontologías [7]. Una posible forma de asistir al modelador es a través del uso de patrones de diseño ontológicos (ODP) [4,5]. Estos patrones son considerados pequeñas ontologías bien definidas y aceptadas por la comunidad, y son usados como modelos o *templates* para ser incorporados a la ontología en desarrollo.

Sin embargo, los enfoques para la Ingeniería Ontológica basados en patrones requieren, por un lado, de la existencia de un conjunto de patrones adecuados y aceptados

en la comunidad para ser reusados. Y por el otro, de metodologías apropiadas que soporten la elicitación de estos patrones y su aplicación en la construcción de nuevos modelos.

Por otra parte, un modelador que desea construir, mantener o validar una ontología a través del reuso de patrones, posee poca asistencia en las herramientas de desarrollo, respecto del uso de estructuras lógicas, que generalmente son poco amigables, haciendo así que las ontologías sean difíciles de comprender [1].

En la literatura, existen pocas herramientas desarrolladas que incluyan soporte para el modelado basado en patrones: CoModIDE [9] y ODPReco [10]. Estas herramientas utilizan un lenguaje gráfico *ad-hoc* basado en grafos para el modelado, lo que implica familiarizarse con este nuevo lenguaje o no proveen soporte gráfico. Asimismo, existen herramientas visuales, pero que no proveen metodologías para el uso de patrones.

Por consiguiente, las tareas de Ingeniería Ontológica en conjunción con la integración de metodologías y buenas prácticas en el uso de patrones, que pueda ser seleccionado en forma amigable de un catálogo, en ambientes gráficos de modelado ontológico, es una arista no explorada en profundidad [6].

En este trabajo, se propone una metodología para sugerir ODP, considerando un modelo ontológico en desarrollo. En particular, se trabaja con Patrones de Contenido, considerando que esta clase de patrones resultan muy útiles en el proceso de modelado [1]. Los Patrones de Contenido se obtendrán de un listado ampliamente aceptado por la comunidad de expertos en el tema [8].

En este sentido, se espera ampliar con la implementación de dicha metodología, el soporte a la Ingeniería Ontológica de las herramientas con ambiente gráfico, como por ejemplo *crowd* [3,2]. De este modo, los modeladores podrán utilizar esta metodología para diseñar sus ontologías, a partir del reuso, extensión e integración de uno o varios patrones del catálogo propuesto.

2. Metodología

La metodología propuesta se diseñó estructurándola como pasos o procedimientos, donde cada uno de ellos recibe y emite productos o información. La Figura 1 presenta el diseño completo, dividido en dos etapas. En dicha imagen, se representa la información de entrada y salida con figuras geométricas ovaladas.

La primer etapa, *Preprocesamiento de Patrones*, tiene como objetivo la extracción de información relevante de los patrones para la detección de los mismos en la ontología del usuario. El *Procesamiento de Nombres* extraerá los nombres de todos los elementos presentes en todos los patrones, y el *Procesamiento de Axiomas* extraerá todos los axiomas de dichos patrones. Esta etapa utilizará una lista de sinónimos para los nombres utilizados en cada uno de los patrones, en caso de que la ontología del usuario utilice algunos de estos sinónimos. Luego, con los nombres y los axiomas extraídos de los patrones, se procede a formar dos grupos de consultas: El primer grupo contendrá todas las consultas de nombres a realizar al razonador; y el segundo grupo contendrá todas las consultas de axiomas presentes en los patrones.

La segunda etapa, *Analizador de Ontologías*, utilizará la información obtenida del submódulo anteriormente mencionado y de la ontología del usuario, para formular con-

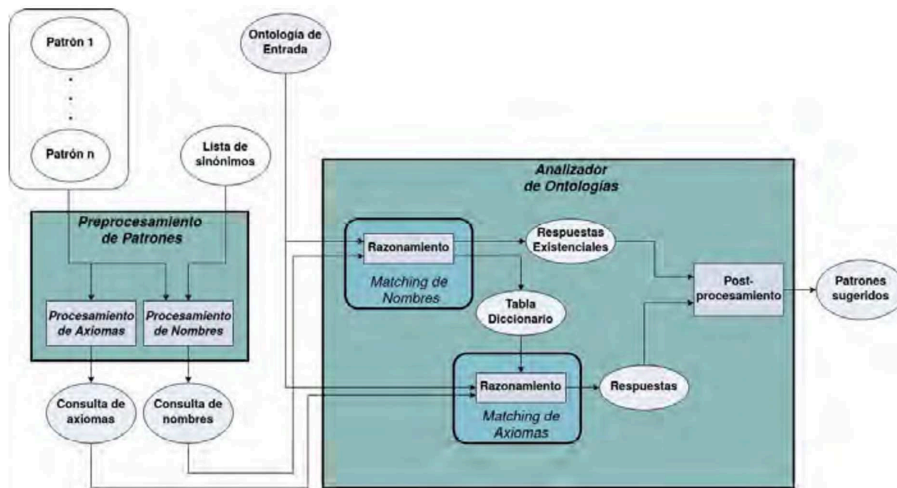


Figura 1. Descripción gráfica de la metodología.

sultas a los razonadores. Inicialmente, se realizará el *Matching de Nombres*, en el que se tomará como entrada las consultas de nombres, junto con la ontología y se le ejecutarán dichas consultas al razonador. Esta tarea realizará una comparación de los nombres (o sinónimos) de los elementos involucrados en la ontología con cada uno de los nombres de los elementos de los patrones para encontrar coincidencias.

A medida que se obtienen los resultados de las consultas, se irán registrando los nombres con sus respectivos sinónimos a la Tabla Diccionario. Esta información será utilizada posteriormente por la tarea *Matching de Axiomas* para verificar que los elementos de un axioma están presentes en la ontología, y para reformular las consultas en base a los sinónimos en caso de ser necesario.

Esta tarea compara los axiomas que existen en la ontología con los axiomas de los patrones. Esto incluye: cardinalidades de las relaciones entre clases, tipos de las relaciones de herencia (disjunta, *covering*), tipos de datos, etc., realizando los cambios necesarios según la Tabla Diccionario poblada durante la tarea *Matching de Nombres*.

Finalmente, luego de este análisis, se hará un postprocesamiento de los resultados obtenidos a fin de determinar los patrones que serán sugeridos al modelador. En este sentido, se realizarán dos tipos de evaluaciones. Primero, se analizará el porcentaje de clases de cada patrón presente en la ontología. Segundo, se analizará la cantidad de nombres de relaciones y de consultas afirmativas de los axiomas del patrón, que están presentes en la ontología. El patrón será sugerido, si los valores obtenidos en ambas evaluaciones supera un umbral particular establecido para cada condición.

Así, la salida de esta etapa será una lista con los patrones sugeridos, que ocurren total o parcialmente en la ontología. Un punto interesante a mencionar es que la complejidad computacional de los algoritmos que implementen la metodología se vislumbra intratable respecto de la cantidad de clases y axiomas.

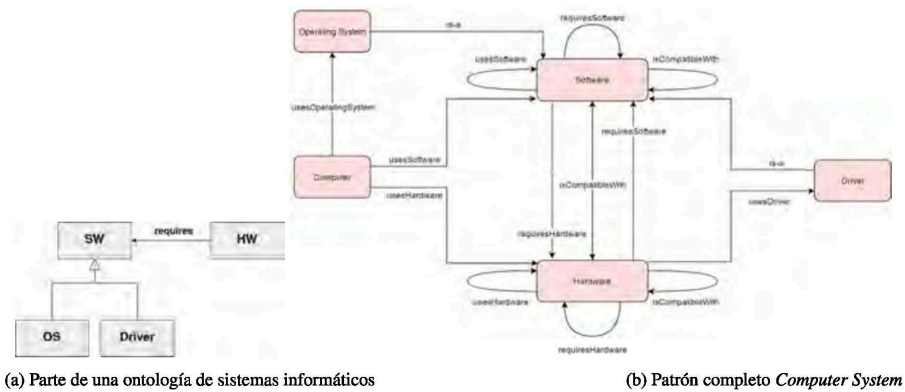


Figura 2. Vista parcial de una ontología y patrón ontológico

3. Ejemplo de aplicación de la Metodología

En la Figura 2 se presenta una vista parcial de una ontología simple, que representa un sistema informático y el patrón *Computer System*.

Inicialmente, la etapa de *Preprocesamiento de Patrones*, el *Procesamiento de Nombres* recorre el patrón *Computer System* y extrae los siguientes nombres: Operating System, Software, Driver, Hardware, requiresSoftware, etc. Por otra parte, selecciona de la Lista de Sinónimos los siguientes nombres: SW, HW, Controlador y OS. El *Procesamiento de Axiomas* extrae los siguientes axiomas: Operating System es subclase de Software, Driver es subclase de Software, Hardware se relaciona con la clase Software a través de la relación requiresSoftware.

A partir de esto, se formulan las consultas que se realizan a los razonadores sobre la ontología del usuario, para detectar la presencia de los elementos del patrón. Las consultas a elaborar son las siguientes: la clase Operating System, ¿es subclase de Software?, la clase Driver, ¿es subclase de Software?, y la clase Hardware, ¿se relaciona con la clase Software a través de la relación requiresSoftware?

Luego del preprocesamiento, el sistema se encuentra listo para analizar ontologías del usuario. En este caso, el usuario ingresa la ontología de la Figura 2 (a) obteniendo una Tabla Diccionario con los valores SW = Software, HW = Hardware, OS = Operative System y Driver = Driver y las Respuestas Existenciales de que dichas clases existen. A continuación, el proceso de *Matching de Axiomas* resuelve las consultas elaboradas previamente. En otras palabras, se confirma que Hardware está en relación requiresSoftware con Software y las consultas de herencia entre Software y sus subclases en esta ontología de entrada. Todas las Respuestas y las Respuestas Existenciales serán Post-procesadas.

En el Post-procesamiento, se realizan las evaluaciones, detectando un 80 % de las clases del patrón en la ontología. Asimismo, el análisis permite concluir que el nombre de la relación requiresSoftware, y tres axiomas del patrón se encuentran presentes en la ontología. Suponiendo un umbral del 50 % para la cantidad de clases en el

patrón detectadas en la ontología y considerando los resultados del análisis, el patrón *Computer System* es sugerido al usuario.

4. Conclusiones y Trabajos Futuros

En este trabajo se presentó el desarrollo de una metodología que recomienda ODP, en la que se consideran Patrones de Contenido obtenidos de un listado ampliamente aceptado. La metodología está dividida en dos etapas. La primer etapa analiza los patrones para obtener una serie de consultas necesarias para detectarlos. En la segunda etapa, recibe la ontología del usuario y sugiere los patrones utilizando las consultas generadas. De esta forma, la metodología permitirá sugerir patrones a medida que el usuario modela su ontología.

Actualmente, se encuentra en desarrollo la implementación de la metodología en una herramienta que puede ser consultada utilizando un *API REST*. Esta decisión fue tomada, teniendo en cuenta que la funcionalidad de la misma no debería depender de una aplicación o software preexistente. Entre nuestros trabajos futuros, se propone una extensión de la herramienta visual Web crowd [3,2], para la Ingeniería Ontológica, permitiendo la utilización de las funcionalidades de dicha API.

Este soporte permitirá a los modeladores sin experiencia en sistemas formales, seleccionar patrones ya aceptados en la comunidad, como base desde donde comenzar sus diseños, acelerando el proceso de desarrollo y mejorando la calidad de sus ontologías al hacerlas más modulares y reusables.

Referencias

1. E. Blomqvist, A. Gangemi, and V. Presutti. Experiments on pattern-based ontology design. In *Proceedings of the fifth international conference on Knowledge capture*, pages 41–48, 2009.
2. Germán Braun, Christian Gimenez, Laura Cecchi, and Pablo Fillottrani. crowd: A Visual Tool for Involving Stakeholders into Ontology Engineering Tasks. *KI - Künstliche Intelligenz*, 2020.
3. Germán Braun, Elsa Estevez, and Pablo Fillottrani. A Reference Architecture for Ontology Engineering Web Environments. *Journal of Computer Science and Technology*, 19(01), Apr. 2019.
4. Aldo Gangemi and Valentina Presutti. Ontology design patterns. In Steffen Staab and Rudi Studer, editors, *Handbook on ontologies*, pages 221–243. Springer, 2009.
5. Pascal Hitzler, Aldo Gangemi, and Krzysztof Janowicz. *Ontology engineering with ontology design patterns: foundations and applications*, volume 25. IOS Press, 2016.
6. Pascal Hitzler and Cogan Shimizu. Modular ontologies as a bridge between human conceptualization and data. In *International Conference on Conceptual Structures*, pages 3–6. Springer, 2018.
7. C. M. Keet. *An Introduction to Ontology Engineering*. University of Cape Town, 2018.
8. NeON Project. Ontology Design Patterns - category proposed contents OP. <http://ontologydesignpatterns.org/> Visitado por última vez el 22 de agosto del 2022.
9. Cogan Shimizu and Karl Hammar. CoModIDE—The Comprehensive Modular Ontology Engineering IDE. In *ISWC 2019 Satellite Tracks*, volume 2456. CEUR-WS, 2019.
10. Maleeha Arif Yasvi and Raghava Mutharaju. ODPReco-A Tool to Recommend Ontology Design Patterns. In *WOP@ ISWC*, pages 71–75, 2019.