Revista de Ingeniería
ISSN: 0121-4993
reingeri@uniandes.edu.co
Universidad de Los Andes
Colombia

Zapata, Carlos Mario; Giraldo, Gloria Lucía; Portilla, Byron; Gómez, Duvan; Naranjo, Marcela; Carmona, Patricia
Aproximación a una ontología para lenguajes de modelado gráfico
Revista de Ingeniería, núm. 29, mayo, 2009, pp. 16-25
Universidad de Los Andes
Bogotá, Colombia

Aproximación a una ontología para lenguajes de modelado gráfico

Approach to a Graphical Modeling Language Ontology

Recibido 8 de septiembre de 2008, modificado 23 de febrero de 2009, aprobado 27 de marzo de 2009.

Carlos Mario Zapata

Phd. Profesor Asociado, Grupo de Investigación en Lenguajes Computacionales, Escuela de Sistemas, Universidad Nacional de Colombia, Sede Medellín. Medellín, Colombia.

cmzapata@unalmed.edu.co ✉

Gloria Lucía Giraldo

Doctora en Informática. Profesora Asistente, Grupo de Investigación en Lenguajes Computacionales, Escuela de Sistemas, Universidad Nacional de Colombia, Sede Medellín. Medellín, Colombia.

glgiraldog@unalmed.edu.co ✉

Byron Portilla

Estudiante de Maestría en Ingeniería de Sistemas, Escuela de Sistemas, Universidad Nacional de Colombia, Sede Medellín. Medellín, Colombia.

beportillar@unalmed.edu.co ✉

Duvan Gómez

Estudiante de Maestría en Ingeniería de Sistemas, Escuela de Sistemas, Universidad Nacional de Colombia, Sede Medellín. Medellín, Colombia.

dagomezbe@unalmed.edu.co ✉

Marcela Naranjo

Estudiante de Maestría en Ingeniería de Sistemas, Escuela de Sistemas, Universidad Nacional de Colombia, Sede Medellín. Medellín, Colombia.

emnaranj@unalmed.edu.co ✉

Patricia Carmona

Estudiante de Especialización en Ingeniería de Sistemas, Escuela de Sistemas, Universidad Nacional de Colombia, Sede Medellín. Medellín, Colombia.

gpcarmonar@unalmed.edu.co ✉

PALABRAS CLAVES

Diagrama, lenguaje de modelado gráfico, lenguajes de modelado, modelo, ontología, SysML, UML, WebML.

RESUMEN

UML, SysML y WebML son lenguajes de modelado gráfico (LMG) similares que no se pueden interpretar conjuntamente, pues tienen diferencias en tipos de modelos y diagramas. En la literatura se encuentran técnicas que estudian las características de algunos LMG, pero se aplican sobre lenguajes particulares, sin considerar sus características comunes. En este artículo se propone el diseño e implementación de una ontología que resuma los principales conceptos y relaciones de los LMG, utilizando una metodología creada en la Universidad de Stanford. La ontología desarrollada responde 35 preguntas de competencia, de las cuales algunas se ejemplifican en el artículo.

KEY WORDS

Diagram, graphical modeling languages, model, modeling languages, ontology, SysML, UML, WebML.

ABSTRACT

UML, SysML, and WebML are graphical modeling languages (GML). Despite their similarities, these languages can not be jointly interpreted, since they exhibit different kinds of models and diagrams. Some studies for examining the features of some GML are proposed in the state of the art, but applied to individual languages, avoiding the common features among such languages. In this paper, we propose an ontology design and implementation for summarizing GML concepts and relations. We use a methodology created in the Stanford University. The developed ontology can successfully answer 35 competence questions, some of them exemplified in this paper.

INTRODUCCIÓN

Los lenguajes de modelado, tanto los gráficos como los textuales, permiten la representación del comportamiento y la estructura de los sistemas del mundo real. Los lenguajes de modelado gráfico (LMG) utilizan modelos visuales y diagramas que realizan esa representación de manera precisa, entendible y económica [1], lo que facilita su uso en las herramientas CASE convencionales. Diversas organizaciones, como el *Object Management Group*—OMG (que se puede ubicar en http://www.omg.org), proponen lenguajes de modelado gráfico orientados a cubrir varias situaciones de acuerdo con sus perspectivas. Es así como cobran importancia UML [2, 3], SysML [4, 5] y WebML [6, 7], que, a través de modelos y diagramas, permiten tomar la información suministrada por los interesados y representarla efectivamente. Sin embargo, debido a la multiplicidad de modelos y diagramas que utilizan estos lenguajes, su interpretación en conjunto es compleja. Por lo tanto, la unificación de los conceptos generales que fundamentan este tipo de lenguajes permite aproximar los diagramas y modelos a los analistas, con el fin de ayudarles en la toma de decisión sobre cuál lenguaje utilizar en el momento de modelar un sistema. De igual manera, es importante lograr el consenso de la comunidad de Ingeniería de Software alrededor de un vocabulario común en estos temas.

Para dar solución a estos problemas, existen diferentes aproximaciones tales como MOF, una especificación que representa la estructura de UML [2, 3]. De igual manera, existen ontologías, como la de Genilloud y Frank [8] y la de Milton y Kazmierczak [9], cuyos alcances son puntuales y tratan de dirimir esta complejidad, pero sin un cubrimiento general.

En este artículo se propone el diseño e implementación de una ontología (empleando una metodología creada en la Universidad de Stanford), a partir de la evaluación de los conceptos y relaciones propios de los lenguajes de modelado gráfico. Para la identificación de las relaciones y equivalencias entre sus diagramas y modelos, se delimita el alcance a tres de estos lenguajes: UML, SysML y WebML. La similitud entre los diagramas propios de cada lenguaje permite establecer equivalencias y posibles relaciones entre ellos.

La estructura de este artículo es la siguiente: en la sección "Los lenguajes de modelado gráfico", se describe el campo de acción y los conceptos involucrados con la ontología a desarrollar; en la sección "Métodos utilizados para representar lenguajes de modelado gráfico", se revisan los trabajos existentes en cuanto a ontologías para lenguajes de modelado gráfico y se establece un punto de vista crítico de ellas; en la sección "Diseño e implementación de una ontología para lenguajes de modelado gráfico", se describe la metodología utilizada para el diseño y elaboración de la ontología propuesta; las secciones finales son "Resultados" y "Conclusiones y trabajo futuro".

LOS LENGUAJES DE MODELADO GRÁFICO

En la ingeniería de software, un soporte para los grupos de desarrollo es la utilización de herramientas que permitan describir y comunicar, de forma estandarizada y consistente, el sistema al cual se pretende llegar. Para ello, se utilizan los lenguajes de modelado, que son lenguajes artificiales usados para expresar información o conocimiento en una estructura definida por un conjunto de reglas consistentes [10]. Estos tipos de lenguajes se clasifican en dos: lenguajes de modelado textual y lenguajes de modelado gráfico. Estos últimos se refieren a un conjunto de símbolos gráficos y reglas, utilizados para transmitir la información al usuario a través de diagramas [11]. Los LMG son el objeto de estudio y análisis del presente artículo.

Cada uno de los LMG cuenta con una serie de modelos y diagramas para representar un sistema. Un modelo contiene los métodos o procesos asociados a los lenguajes de modelado, que describen un sistema a través de diferentes tipos de diagramas [12]. Estas descripciones se presentan de forma visual. UML [2, 3], SysML [4, 5] y WebML [6, 7] presentan los modelos: de estructura, de comportamiento, de hipertexto

y de interacción. A su vez, cada modelo se compone de varios diagramas para representar un sistema a partir de sus características particulares.

UML

El *Unified Modeling Language* es un lenguaje estándar para modelar la estructura y el comportamiento de una aplicación. Además, modela procesos de negocio y estructuras de datos, con el objetivo de visualizarlos, especificarlos, construirlos y documentarlos [2, 3]. Los diagramas de UML se dividen en tres categorías: estructura (clases, componentes, estructura compuesta, despliegue, objetos y paquetes), comportamiento (actividades, casos de uso y máquina de estados) e interacción, un subgrupo de los de comportamiento (secuencias, comunicación, vista de interacción y tiempos). Se considera este lenguaje como el estándar *de facto* para modelar aplicaciones de software orientadas a objetos [3].

SysML

El *System Modeling Lenguage* es un lenguaje de modelado basado en UML y sirve para especificar, analizar, diseñar y verificar sistemas complejos, que pueden incluir hardware y software [4,5]. Este lenguaje utiliza fundamentos semánticos que le permiten modelar requisitos, comportamiento, estructura y parámetros de un sistema. Comparando SysML 1.0 con UML 2.0, se aprecia que SysML presenta tres diagramas nuevos: *Requirements, Parametrics* y *Allocation*. Los diagramas de actividades y de bloque, ya existentes en UML 2.0, se reutilizan y extienden en SysML. Además, los diagramas de máquinas de estado, interacciones y casos de uso se reutilizan sin modificaciones en SysML [4]. SysML se diferencia de UML porque contiene extensiones y diagramas adicionales, que involucran el software y el hardware.

WebML

El *Web Modeling Language* apoya las actividades de diseño de sitios Web a través de conceptos, gráficos, formalismos, especificaciones y diseño de procesos, soportados por herramientas de diseño visual [6,7]. WebML posee cuatro tipos de modelos: estructural, que es similar al diagrama de clases de UML; de hipertexto, que se divide a su vez en composición y navegación; de presentación, que es el único textual pues se presenta en forma de sentencias en XML (*Extensible Markup Language*); y de navegación.

MÉTODOS UTILIZADOS PARA REPRESENTAR LENGUAJES DE MODELADO GRÁFICO

Las técnicas propuestas para definir las características de los lenguajes de modelado gráfico estudian independientemente cada uno de ellos, con el propósito satisfacer necesidades específicas. En la revisión de la literatura no se encontró una técnica que hiciera posible el estudio unificado de varios lenguajes de modelado gráfico, para encontrar equivalencias y diferencias.

Entre las técnicas encontradas está la especificación MOF (*Meta-Object Facility*), descrita en [13]. MOF representa, mediante clases, relaciones, tipos de datos y paquetes, la descripción del lenguaje UML [2,3]. Los metamodelos instanciados de MOF se emplean para definir cuáles deben ser los componentes utilizados en cada diagrama de UML y cómo debe ser la forma de relacionarlos. Esto garantiza la consistencia y compatibilidad con los servicios necesarios para transportar los diagramas resultantes de una herramienta de diagramación a otra. Aunque MOF es útil para la definición del metamodelo de UML (y podría ser útil para definir otros lenguajes de modelado), no facilita la comparación entre ellos. Por lo tanto, aunque ayuda en la correcta construcción de los diagramas para modelar los requisitos, no contribuye en la selección adecuada del lenguaje de modelado.

Otra técnica consiste en describir la estructura de un lenguaje de modelado a través de ontologías, las cuales representan el conocimiento mediante conceptos, relaciones, funciones, instancias y axiomas, como se expresa en [14]. Sin embargo, allí sólo se reconoce la importancia de las ontologías en este contexto, pero no se propone la estructura de una ontología particu-

lar que posibilite la comparación entre los lenguajes de modelado gráfico.

La ontología *Chisholm* [9] se utiliza para el análisis conceptual de los lenguajes de modelado de datos. Esta ontología conforma una teoría unificadora de estos modelos, porque incluye los diagramas: entidad-relación, de datos funcionales, de datos semánticos y la técnica de modelado de objetos. Empero, esta ontología no contempla la totalidad de las características de los LMG, pues se orienta casi de manera exclusiva a los datos, asociados con modelos de tipo estructural.

Por otro lado, en [8] se utiliza el modelo de referencia RM-ODP (*Reference Model for Open Distributed Processing*) como una ontología de UML, la cual plantea un vocabulario y un marco semántico común a todos los usuarios de este lenguaje de modelado. El problema identificado en [8] consiste en que el estándar de UML confunde a los usuarios, debido a que utiliza varias definiciones para un mismo concepto y responde a este problema de ambigüedad de vocabulario de manera parcial. Además, sólo contempla la definición de un vocabulario común para UML, segmentando de esta manera los demás LMG.

En la Tabla 1 se presenta un cuadro comparativo en el que se establecen aspectos positivos y negativos de los diversos métodos utilizados para la definición de lenguajes de modelado gráfico.

DISEÑO E IMPLEMENTACIÓN DE UNA ONTOLOGÍA PARA LENGUAJES DE MODELADO GRÁFICO

A partir del análisis de los métodos en la sección "Métodos utilizados para representar lenguajes de modelado gráfico" y teniendo en cuenta sus aspectos negativos, los cuales se compendian en la Tabla 1, se propone una ontología para superar estas limitaciones que, a la vez, sirva como soporte importante para que los grupos de desarrollo puedan elegir el lenguaje de modelado gráfico que más se adapte al producto que están elaborando.

Para evitar la omisión de conceptos importantes, la ontología utiliza un conjunto de información relacionada con los lenguajes de modelado gráfico UML, SysML y WebML, tres de los más representativos en la actualidad. Esta información se estructura de acuerdo con la composición de los lenguajes de modelado gráfico, con el fin de determinar equivalencias entre ellos.

El proceso de estructuración de la información asociada a cada uno de los LMG se realizó a través de una serie de pasos, que van desde la formulación de preguntas de competencia hasta la definición del manejo y utilización de los conceptos asociados a la composición de cada diagrama perteneciente a los LMG. Estos pasos forman parte de la metodología defini-

Métodos para la definición de los LMG	Aspectos positivos	Aspectos negativos
Especificación MOF	Captura la diversidad del estándar de modelado para integrar diferentes tipos de modelos y metadatos, e intercambiarlos entre diferentes herramientas. Permite definir lenguajes de modelado a partir de metamodelos.	Permite definir un lenguaje de modelado (UML), mas no permite la comparación entre varios lenguajes de modelado.
Ontología *Chisholm*	Tiene la potencia necesaria para ser una teoría unificadora de los modelos de datos y crea una ontología de lenguaje de modelado de datos. Asimismo, disminuye la heterogeneidad semántica de los modelos de datos.	Sólo se utiliza para los lenguajes de modelado de datos.
Ontología RM-ODP	Propone una ontología que disminuye la heterogeneidad semántica para el vocabulario de UML.	Sólo se presenta para UML. La solución no cubre la totalidad de los LMG. No se describe cómo utilizar la RM-ODP como una ontología.

Tabla 1. Comparación de algunas técnicas de definición de lenguajes de modelado.

da por Noy y McGuinness [15] de la Universidad de Stanford. Para la implementación de la ontología, se seleccionó la herramienta Protégé® de la misma universidad [16].

En la metodología empleada, un paso importante es la elaboración de las preguntas de competencia, útiles para determinar el alcance de la ontología. En esta etapa, se definieron 35, tales como: ¿Qué modelos componen un lenguaje de modelado gráfico? ¿Qué diagramas constituyen un modelo de comportamiento? ¿Qué componentes tiene un diagrama de casos de uso? ¿Qué componentes son comunes al diagrama de clases y al diagrama de objetos de UML? ¿En qué diagramas se presenta la relación de herencia? En la definición de estas preguntas se contó con la información de veinte proyectos correspondientes a la línea de profundización en Ingeniería de Software de la Universidad Nacional de Colombia, sede Medellín. Éstas son algunas de las que pueden surgir en las diferentes fases del ciclo de vida de una aplicación de software, en las cuales se emplean los diferentes modelos, diagramas y componentes de los lenguajes de modelado gráfico. Además, sirven para definir el alcance de la ontología y orientan la selección del conjunto de términos que se van a utilizar en ella.

Una vez identificados los términos que hacen parte de la ontología, se procede a definir la jerarquía de clases, mediante la cual se identifica la dependencia de los términos asociados a los lenguajes de modelado gráfico, a partir de las relaciones que se pueden deducir de la sintaxis de tales lenguajes. Como consecuencia del estudio de los términos y las relaciones entre ellos, resultaron tres clases: *Componente*, *Modelo* y *Diagrama*, las cuales cubren los conceptos representativos de los lenguajes de modelado gráfico y se consideraron suficientes para caracterizarlos. La clase *Componente* tiene, a su vez, dos subclases: *Elemento* y *Relación*. De igual manera, la clase *Modelo* posee las subclases *Modelo de Estructura* y *Modelo de Comportamiento*, que a su vez se divide en: *Modelo de Interacción* y *Modelo de Hipertexto*. Esta estructura inicial se presenta en la Figura 1,

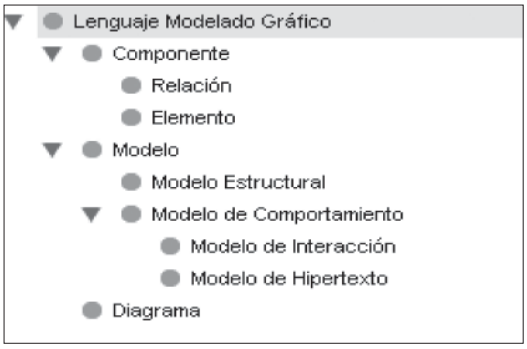

Figura 1. Jerarquía de clases para la ontología propuesta para los lenguajes de modelado gráfico.

elaborada a partir de la ontología implementada en Protégé®.

Definidas las clases y subclases, se asignan las propiedades o *slots* correspondientes a cada una de éstas y se consideran las relaciones de generalización que se puedan aprovechar. Se definió la propiedad *Nombre*, común para todas las clases y subclases, que se utiliza como etiqueta para identificar todas las instancias de las mismas. Además de esa propiedad, se asignan otras propiedades específicas a cada clase. Así, a la clase *Componente* se asocia la propiedad *NombreDiagrama* para relacionar las instancias de ésta con las instancias de la clase *Diagrama*; a la clase *Modelo*, se le asigna la propiedad *Lenguaje* para que sus instancias se relacionen con las de la clase *Lenguaje de Modelado Gráfico*; finalmente, a la clase *Diagrama* se asocian las propiedades *NombreComponente*, *Lenguaje* y *NombreModelo* para, de forma similar a los casos anteriores, relacionar sus instancias con las de las clases *Componente, Lenguaje de Modelado Gráfico* y *Modelo* respectivamente. A manera de ejemplo, la Figura 2 muestra los slots asociados a la clase *Diagrama*.

Con el propósito de crear relaciones diferentes a las de la jerarquía entre clases, se definen los *slots* de tipo *"instance of"*. Así, por ejemplo, para la clase *Diagrama* se definió el *slot Lenguaje* de tipo instancia de la clase *Lenguaje de modelado gráfico* y de cardinalidad múltiple (para expresar que uno o varios LMG pueden defi-

nir un diagrama). Por ejemplo, el diagrama "Casos de uso" se define tanto para UML como para SysML. De la misma manera, se definieron los *slots NombreModelo* y *NombreComponente* de tipo instancia de las clases *Modelo y Componente*, respectivamente. De esta manera, se expresa que un diagrama se puede asociar a uno o varios modelos. Por ejemplo, el diagrama de colaboración o comunicación se asocia tanto al modelo de interacción como al de comportamiento. También, se expresa que un diagrama puede tener una o varias componentes. Por ejemplo, el diagrama de Actividades de UML posee, entre sus componentes, las siguientes: actor, relación *include*, relación *extend*, etc. Ésta es la forma en que se pueden definir equivalencias estructurales entre los diferentes lenguajes de modelado gráfico.

En el caso de la ontología propuesta para los LMG, se definieron en total 114 instancias, que corresponden a los componentes (relaciones y elementos), diagramas y modelos asociados a UML, WebML y SysML. A cada clase se le crearon sus respectivas instancias y, además, se crearon relaciones con instancias de otras clases. Por ejemplo, en la Figura 3 se observa cómo se relacionan las instancias de la clase *Elemento* con las de la clase *Diagrama*, para expresar que un elemento como *Actor* está presente en los *diagramas de Secuencias, de Estructura compuesta y de Casos de Uso*.

RESULTADOS

La metodología empleada para el diseño e implementación de la ontología para LMG, exige que se valide dicho diseño, al verificar que la ontología sí responda a las preguntas de competencia que se definieron inicialmente. Se definieron 35 consultas (*queries*), a las cuales la ontología respondió de manera satisfactoria

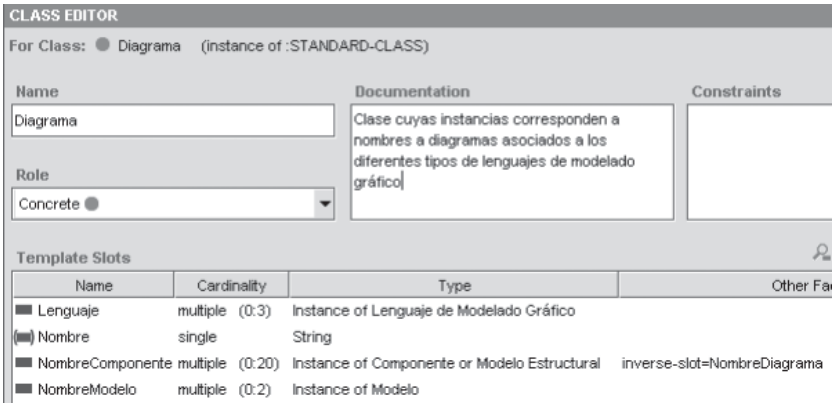

Figura 2. Propiedades asociadas a la clase *Diagrama*.

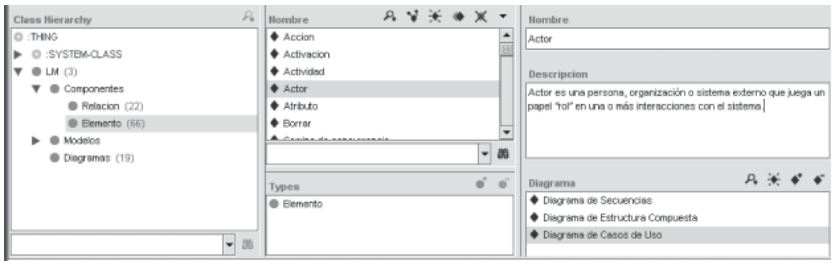

Figura 3. Relaciones, a nivel de instancia, de la clase *Elemento* con la clase *Diagrama*.

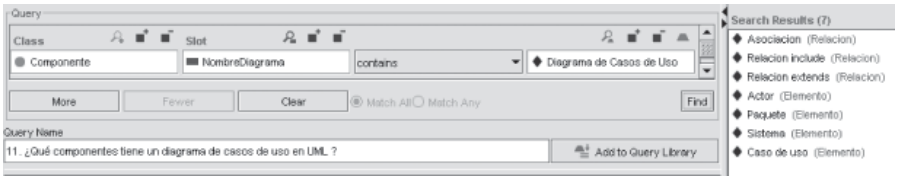

Figura 4. Resultado de la consulta 1.

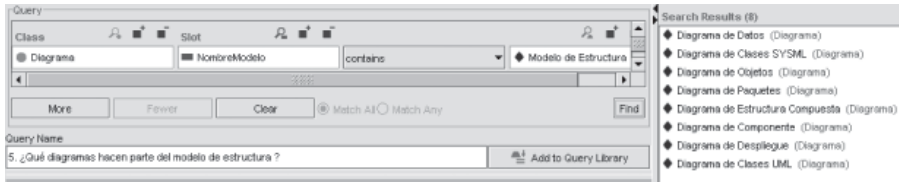

Figura 5. Resultado de la consulta 2.

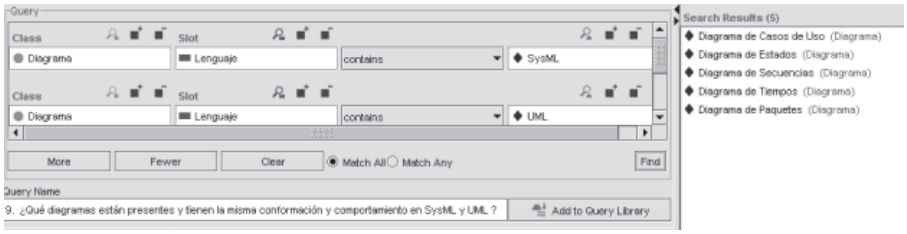

Figura 6. Resultado de la consulta 3.

y entregó los resultados esperados. Tres analistas expertos revisaron los resultados y concluyeron que las respuestas eran las adecuadas para las preguntas de competencia planteadas. De esta manera, se verificó el cumplimiento del objetivo inicial propuesto: elaborar una ontología que constituya una herramienta de selección de LGM orientada a los grupos de desarrollo de aplicaciones de software. Esta ontología logra un cubrimiento amplio del vocabulario relativo a dichos lenguajes. En esta sección se presentan algunas *queries* (escritas en Protégé®) correspondientes a las preguntas de competencia y se muestran las respuestas que arrojó la ontología diseñada e implementada.

Query 1: ¿Qué componentes tiene un diagrama de casos de uso en UML?

En Protégé®, la expresión para esta consulta es: *Clase* = Componente, *Slot* = Diagrama y *Contains* = Diagrama de Casos de uso. La Figura 4 muestra el resultado bajo el título *Search Results*.

Query 2: ¿Qué diagramas hacen parte del modelo de estructura?

En Protégé®, la expresión para esta consulta es: *Clase* = Diagrama, *Slot* = Modelo, y *Contains* = Modelo de Estructura. La respuesta se muestra en la Figura 5.

Query 3: ¿Qué diagramas son equivalentes (están presentes y tienen el mismo comportamiento y estructura) en los lenguajes de modelado SysML y UML?

En Protégé®, la expresión para esta consulta es: *Clase* = Diagrama, *Slot* = Lenguaje y *Contains* = SysML, *And Clase* = Diagrama, *Slot* = Lenguaje y *Contains* = UML. La Figura 6 muestra el resultado de esta consulta. Este tipo de consultas permite resolver y definir equivalencias entre los lenguajes de modelado.

CONCLUSIONES Y TRABAJO FUTURO

En este artículo se propuso el diseño y la implementación en Protégé® de una ontología sobre lenguajes de modelado gráfico, con el propósito de permitir a los grupos de desarrollo de software conocer, comprender, estructurar, identificar y obtener equivalencias sobre este tipo de lenguajes. La ontología propuesta puede determinar, por ejemplo, cuáles son los diagramas comunes y cuáles los modelos utilizados en diferentes tipos de lenguaje. Para la construcción de la ontología, se seleccionaron las características de tres de los lenguajes de modelado gráfico más representativos: UML, SysML y WebML.

La ontología construida constituye una herramienta útil en la comprensión y análisis de diagramas de los LMG y, a su vez, sirve de soporte en la toma de decisiones por parte de los grupos de desarrollo acerca de qué lenguaje utilizar, dadas las características de los diagramas que hacen parte de su sintaxis.

La ontología propuesta es dinámica y evolutiva, es decir, se puede fácilmente extender con otros lenguajes de modelado gráfico y se puede actualizar a medida que las versiones de los LMG aparezcan.

Cada lenguaje de modelado tiene una serie de características que lo hacen independiente de otros. Esas características incluyen algunos diagramas y/o modelos que suprimen o complementan otros. No obstante, en cualquier caso, su composición se orienta a contar con una parte estructural, una parte de interacción y otra de comportamiento, que hacen posible la organización de la información, de tal forma que pueda ser viable una comparación entre los diferentes lenguajes y la identificación de sus posibles relaciones.

Como trabajo futuro, y gracias a que la estructura de la ontología propuesta es flexible, se propone incrementar la base de conocimientos de la ontología, incluyendo otros lenguajes de modelado gráfico.

De igual manera, se pretende seguir con la descripción de cada una de las instancias representadas en la ontología, de acuerdo a la evolución (versiones) de cada uno de estos lenguajes.

REFERENCIAS BIBLIOGRÁFICAS

[1] R. Moody, W. Cooper and W. Fleming.
"Modeling and Simulation of Information Systems using Graphical Languages". Proceedings of the IEEE 1991 National Aerospace and Electronics. Conference NAECON, Dayton, Mayo de 1991. Disponible: http://ieeexplore.ieee.org/Xplore/guesthome.jsp

[2] M. Fowler.
UML Distilled: A brief guide to the Standard Object Modeling Language. 3rd edition. Addison-Wesley, Reading, 2004.

[3] G. Booch, J. Rumbaugh. and I. Jacobson.
The Unified Modeling Language User Guide. Addison-Wesley, Reading, 1998.

[4] L. Balmelli.
"An Overview of the Systems Modeling Language for Products and Systems Development". *Journal of Object Technology*. JOT, Vol. 6, No. 6, Julio-Agosto 2007, pp. 149-177.

[5] M. Hause.
"The SysML Modelling Language". Fifth European Systems Engineering Conference, Cheltenham, Glos. UK, Septiembre de 2006. Disponible: www.omgsysml.org/The_SysML_Modelling_Language.pdf.

[6] S. Ceri, P. Fraternali and A. Bongio.
"Web modeling language (WebML): a modeling language for designing Web sites". *Computer networks—the international journal of computer and telecommunications networking.* Vol. 33, Junio 2000, pp. 137–157.

[7] M. Bronloilla, S. Comai, P. Fratenali and M. Matara.
"Designing Web Applications with WebML and WebRatio". En: *Web Engineering: Modelling and Implementing Web Applications*. Springer, London, 2007, pp. 221–261.

[8] G. Genilloud and W. Frank.
"Use Case Concepts using a Clear, Consistent, Concise Ontology". *Journal of Object technology*. Vol. 4, No. 6, Special Issue: Use Case Modeling at UML, 2005, pp. 95-107.

[9] S. Milton, and E. Kazmierczak.
"An ontology of data modelling languages: a study using a common-sense realistic ontology". *Journal of Database Management*. Vol. 15, No. 2, 2004, pp. 19-38.

[10] D. Harel and B. Rumpe.
Modeling languages: syntax, semantics and all that stuff, Part I: the basic stuff. Technical report MCS00-16, Mathematics & Computer Science, Weizmann Institute of Science, 2000.

[11] He. Xiao, S. Zhiyi and L, Weizhong.
"A metamodel for the notation of graphical modeling languages". *31st Annual International Computer Software and Applications COMPSAC 2007*, Vol. 1, Julio.2007, pp. 219–224.

[12] P. Kruchten.
Rational Unified Process: An Introduction. Addison-Wesley, Reading, 1999.

[13] "Meta-Object Facility Home Page. Objet management group". OMG. Fecha de Consulta: Agosto 4 de 2008. Disponible: http://www.omg.org/mof

[14] G. Barchini.
"El rol de las ontologías en los SI". *Revista Ingeniería Informática*. Vol. 14, Mayo 2007, pp. 1–12.

[15] N. Noy and D. McGuinness.
Ontology Development 101: A Guide to Creating Your First Ontology. Technical Report KSL-01-05, Knowledge Systems Laboratory, Stanford University, March 2001.

[16] "The Protégé Ontology Editor and Knowledge Acquisition System. Stanford". Stanford University. Fecha de Consulta: 4 de Agosto de 2008. Disponible: http://protege.stanford.edu