Agentes Inteligentes y Web Semántica: Hacia la Verbalización de un Subconjunto de UML en una Herramienta Gráfica Web

Matías Garrido[1] Germán Braun[1,2,3] Sandra Roger[1]

email: {roger,german.braun}@fi.uncoma.edu.ar

[1]Grupo de Investigación en Lenguajes e Inteligencia Artificial
Departamento de Teoría de la Computación - Facultad de Informática
UNIVERSIDAD NACIONAL DEL COMAHUE

[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

[3]Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)

RESUMEN

El proyecto de investigación Agentes Inteligentes y Web Semántica, financiado por la Universidad Nacional del Comahue, tiene como objetivo general la generación de conocimiento especializado en el área de agentes inteligentes y en lo referente a la representación y el uso del conocimiento en sistemas computacionales basados en la Web, es decir, lo que se ha llamado la Web Semántica.

El objetivo general del trabajo de investigación es la extensión de una herramienta de modelado ontológico, denominada crowd, mediante la verbalización de un subconjunto del lenguaje de modelado conceptual UML. Esta integración permitirá generar especificaciones en Lenguaje Natural a partir de un diagrama de clases.

Esta línea de investigación se desarrolla en forma colaborativa entre docentes-investigadores de la Universidad Nacional del Comahue y de la Universidad Nacional del Sur, en el marco de proyectos de investigación financiados por las universidades antes mencionadas.

Palabras Clave: Verbalización, Generalización de Lenguaje Natural, UML, Ontologías.

CONTEXTO

Este trabajo está parcialmente financiado por la Universidad Nacional del Comahue, en el contexto del proyecto de investigación Agentes Inteligentes y Web Semántica (04/F014), por la Universidad Nacional del Sur a través del proyecto de investigación Integración de Información y Servicios en la Web (24/N027) y por el Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), en el contexto de una beca interna doctoral. Los proyectos de investigación tienen una duración de cuatro años y la beca doctoral con duración de 5 años, finalizando en abril de 2019.

1. INTRODUCCIÓN

La fase de requerimientos es la más problemática en un proceso de desarrollo de software [1]. Estos problemas incluyen las dificultades de elicitar correctamente los requerimientos del usuario, en su entendimiento y en la transformación de estos mismos dentro de un modelo computacional que puede ser semiformal, usualmente referidas a una notación grafica tal como los modelos Orientados a Objetos (OO) [2] los cuales usan el Lenguaje Unificado de Modelado (UML) [3] o lenguajes de

especificación formal tales como Vienna Development Method (VDM) [4] o Z [5].

La verbalización es el proceso de escribir la semántica capturada en una teoría lógica (relaciones entre entidades y sus restricciones) en sentencias en lenguaje natural.

En la fase de análisis del desarrollo de los sistemas de información, es importante que el esquema conceptual sea validado por el experto del dominio, para asegurar que el esquema modela con precisión los aspectos relevantes del dominio del negocio.

Una manera efectiva de facilitar esta validación es verbalizar el esquema en un lenguaje fácilmente comprensible por el experto del dominio, quien puede no contar con el conocimiento técnico adecuado. Dicha verbalización también puede ser usada como una manera de integrar a los usuarios en los procesos de chequeo de consistencia cuando los cambios son realizados en el diseño o en la implementación. Como consecuencia, esta brecha de comunicación entre los modeladores y los expertos en el dominio es minimizada.

En este sentido, existen algunas investigaciones a tener en cuenta que se relacionan con sistemas de Generación de Lenguaje Natural (GLN) [6]. Uno de ellos, es el sistema ModEx (Model Explainer) [7], que genera lenguaje natural desde descripciones de modelos de software OO. Sin embargo, ModEx no verifica semánticamente la salida final y asume que dicha verificación es realizada por los usuarios, comparando el diagrama con el sistema de especificación generado mediante la GLN. En general, el sistema funciona con éxito para los modelos que cumplen con las suposiciones de como las clases y las relaciones deben ser llamadas.

Con el fin de extender las capacidades del sistema anterior y considerando una convención de nomenclaturas más amplia, surgió GeNLangUML [8]. Se trata de un sistema que propone la generación de una especificación de lengua je natural en inglés a partir de diagramas de clases UML. Utiliza WordNet, una ontología lingüística, para realizar el análisis sintáctico de los nombres de entrada y la verificación de las sentencias generadas. La validación de GenLangUML fue realizada extrayendo convenciones de nomenclaturas de libros académicos. Sin embargo, esto podría impactar negativamente en su aplicación en la industria, debido a que las organizaciones podrían tener diferentes convenciones para definir sus diagramas de clases.

Hay enfoques establecidos con respecto a la verbalización multilingüe. Uno de ellos es DogmaModeller [9], basado sobre ORM 1 (Object-Role Modeling) [10], para la generación automática de verbalizaciones en lenguaje pseudo-natural. Es una herramienta de ingeniería de ontologías basada en los principios de ORM e implementa totalmente la verbalización multilingüe. Para cada uno de estos idiomas de salida, DogmaModeller posee un template que contiene estructuras determinadas por la sintaxis para cada tipo de restricción de ORM. Cada estructura contiene etiquetas para referenciar a los tipos de objetos y los roles que forman parte de dichas restricciones. Por otro lado, DogmaModeller es extensible al poder crear nuevos templates de verbalización para otros idiomas. Sin embargo, para hacer que las sentencias verbalizadas sean gramaticalmente correctas en cualquier lenguaje natural, se requiere un tratamiento más complejo a través de un análisis morfológico automatizado para cada idioma. Este es un área de investigación activa en NLG.

Otra herramienta de software de verbalización automatizada, que soporta modelos ORM de segunda generación (ORM2) es NORMA (Natural ORM Architect) [11]. En esta herramienta, la verbalización de elementos individuales en el modelo ORM principal se genera utilizando una transformación XSLT aplicada a un archivo XML. En este proceso se identifican rápidamente diferentes patrones de verbalización y posteriormente se describe como deben combinarse las frases para producir la verbalización en un inglés legible.

En este contexto, el objetivo del presente trabajo es extender la arquitectura de nuestra herramienta gráfica de modelado ontológico crowd [12, 13], para soportar la verbalización multilingüe de un subconjunto del metamodelo

de UML referido a los diagramas de clases. crowd es un prototipo cliente-servidor, actualmente en desarrollo por nuestros grupos de investigación, en respuesta a la complejidad inherente al modelado conceptual y ontológico, además de explotar las bondades de los sistemas basados en Lógicas Descriptivas (DL) [14].

La estructura del presente trabajo es la siguiente. En la sección 2 presentamos los objetivos de los proyectos de investigación en los que se enmarca este trabajo y describimos la línea de investigación actual. En la sección 3 indicamos algunos resultados obtenidos y trabajos futuros. Finalmente, comentamos aspectos referentes a la formación de recursos humanos en esta temática.

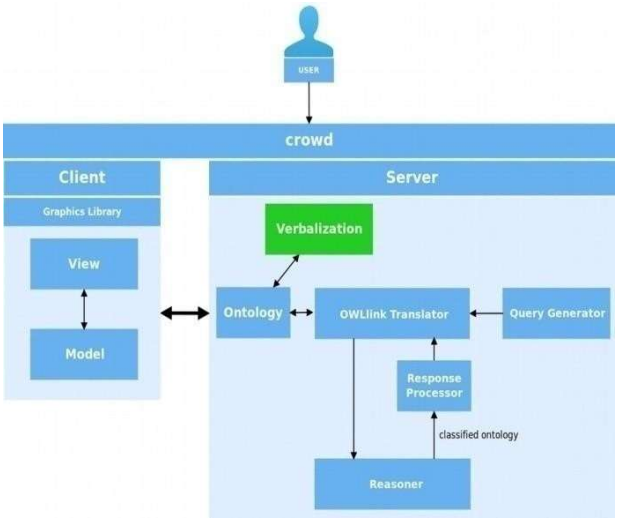

Figura 1. Arquitectura de crowd

2. LÍNEA DE INVESTIGACIÓNY DESARROLLO

El proyecto de investigación Agentes Inteligentes y Web Semántica tiene como objetivo general generar conocimiento especializado en el área de agentes inteligentes y en lo referente a la representación y el uso del conocimiento en sistemas computacionales basados en la web, es decir lo que se ha llamado la Web Semántica.

Por otro lado, en el proyecto de investigación Integración de Información y Servicios en la Web se propone investigar y desarrollar metodologías y herramientas que favorezcan la interoperabilidad semántica de información y de servicios en la Web, fundamentados en los últimos avances en el área de lenguajes de representación del conocimiento, ontologías y modelado conceptual.

Ambos proyectos confluyen en la línea de investigación de este trabajo, en la que se explora entre otros, sobre temas afines a la Representación del Conocimiento, las Lógicas Descriptivas, [15], las Ontologías, la Ingeniería de Software basada en Conocimiento y la Ingeniería de Conocimiento.

En los trabajos [12, 13] se presentó la arquitectura inicial de crowd (ver Figura 1) y un prototipo implementado que permite, en primer instancia, determinar la consistencia de un modelo gráfico representando una ontología. El front-end permite al usuario modelar de forma gráfica usando diagramas de clases UML, mientras que el back-end trabaja del lado del servidor con un razonador capaz de inferir posibles restricciones implícitas en los modelos. Los módulos en el servidor traducen el modelo inicial en uno lógico basado en DL, como propone [16]. La comunicación entre el cliente y el servidor es a través del protocolo OWLlink [17].

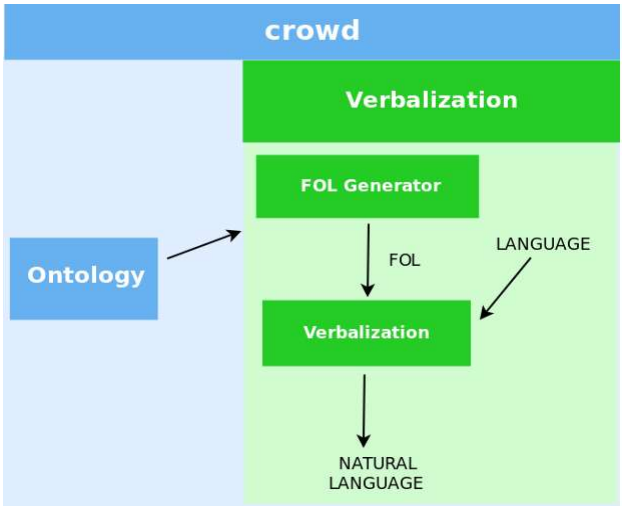

Figura 2. Arquitectura de Verbalización

La Figura 2 muestra la arquitectura básica del proceso de verbalización planteada e incorporada a la arquitectura mostrada anteriormente. La entrada a dicho proceso es

un subconjunto del metamodelo de UML, que es traducido a un lenguaje intermedio basado en lógica de primer orden (FOL), por el módulo FOL generator. La verbalización es multilingüe, por lo cual el usuario puede elegir el idioma para traducir las sentencias FOL. En una primera etapa, los idiomas de traducción serán inglés y español y, posteriormente, planeamos extender el conjunto de idiomas posibles.

3. RESULTADOS OBTENIDOS Y TRABAJOS FUTUROS

Inicialmente, se diseñó una primera versión de la arquitectura cliente-servidor, incluyendo entre otros módulos la generación de consultas, librerías gráficas y un traductor para OWLlink.

Con el fin de extender esta herramienta para validación de requerimientos del usuario, se incorporó un módulo de verbalización. El mismo fue ideado con el objetivo de ser multilingüe y, para facilitar esta traducción, se decidió utilizar una representación intermedia en lógica de primer orden de los diagramas de clases UML. Finalmente, para abordar la complejidad inherente a la generación de las sentencias en el idioma destino elegido, el proceso de verbalización procederá con la creación de patrones de escritura generales, basados sobre algunas herramientas lingüísticas existentes [18, 19, 20].

Actualmente, nos encontramos en la fase de diseño de los módulos de verbalización en crowd, y próximos a iniciar la implementación de los mismos. Asimismo, se pretende estudiar diferentes técnicas para la validación de nuestro prototipo.

4. FORMACIÓN DE RECURSOS HUMANOS

Durante la realización de este sistema se espera lograr, como mínimo, la culminación de 2 tesis de grado dirigidas y/o codirigidas por los integrantes del proyecto. Uno de los autores de este trabajo está inscripto en el Doctorado en Ciencias de la Computación en la Universidad Nacional del Sur (beca interna doctoral CONICET).

Finalmente, es constante la búsqueda hacia la consolidación como investigadores de los miembros más recientes del grupo.

BIBLIOGRAFÍA

[1] Michael Christel and Kyo Kang. Issues in requirements elicitation. Technical Report CMU/SEI-92-TR-012, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1992.

[2] Grady Booch, Robert Maksimchuk, Michael Engle, Bobbi Young, Jim Conallen, and Kelli Houston. Object-oriented Analysis and Design with Applications, Third Edition. Addison-Wesley Professional, third edition, 2007.

[3] Grady Booch, James Rumbaugh, and Ivar Jacobson. Unified Modeling Language User Guide. Addison-Wesley Professional, 2005.

[4] Jones, Cliff B. Systematic Software Development using VDM. Prentice-Hall, Up-per Saddle River and NJ 07458 and USA, 1990.

[5] J. M. Spivey. The Z Notation: A Reference Manual. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[6] Ehud Reiter and Robert Dale. Building Natural Language Generation Systems. Cambridge University Press, New York, NY, USA, 2000.

[7] Benoit Lavoie, Owen Rambow, and Ehud Reiter. The modelexplainer, 1996.

[8] Farid Meziane, Nikos Athanasakis, and Sophia Ananiadou. Generating natural language specifications from UML class diagrams. Requir. Eng., 13(1):1–18, 2008.

[9] Mustafa Jarrar. Towards Methodological Principles for Ontology Engineering. PhD thesis, Vrije Universiteit Brussel, Brussels, 5 2005.

[10] Terry Halpin and Tony Morgan. Information Modeling and Relational Databases. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edition, 2008.

[11] Matthew Curland and Terry A. Halpin. The NORMA software tool for ORM 2. In Information Systems Evolution - CAiSE Forum 2010, Hammamet, Tunisia, June 7-9, 2010, Selected Extended Papers, pages 190–204, 2010.

[12] Christian Gimenez, Germán Braun, Laura Cecchi, and Pablo Fillottrani. Una Arquitectura Cliente-Servidor para Modelado Conceptual Asistido por Razonamiento Automático. In XVIII Workshop de Investigadores en Ciencias de la Computación, 2016.

[13] Christian Gimenez, Germán Braun, Laura Cecchi, and Laura Fillottrani. crowd: A Tool for Conceptual Modelling assisted by Automated Reasoning - Preliminary Report. In Proc. of the 2nd Simposio Argentino de Ontologías y sus Aplicaciones (SAOA) colocated at Jornadas Argentinas de Informática (JAIIO) - to appear, 2016.

[14] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. The Description Logic Handbook: theory, implementation, and applications. 2003.

[15] Diego Calvanese, Maurizio Lenzerini, and Daniele Nardi. Description logics for conceptual data modeling. In Logics for Databases and Information Systems, pages 229–263. Kluwer, 1998.

[16] Daniela Berardi, Diego Calvanese, and Giuseppe De Giacomo. Reasoning on UML class diagrams. Artif. Intell., 168(1-2):70–118, 2005.

[17] Thorsten Liebig, Marko Luther, Olaf Noppens, and Michael Wessel. Owllink. Semantic Web, 2(1):23–32, 2011.

[18] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky . The Stanford CoreNLP natural language processing toolkit. In Association for Computational Linguistics (ACL) System Demonstrations, 2014.

[19] Edward Loper and Steven Bird. Nltk: The natural language toolkit. In Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1. Association for Computational Linguistics, 2002.

[20] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, Adam Funk, Angus Roberts, Danica Damljanovic, Thomas Heitz, Mark A. Greenwood, Horacio Saggion, Johann Petrak, Yaoyong Li, and Wim Peters. Text Processing with GATE (Version 6). 2011.