

# Modelado conceptual de gramáticas formales y máquinas abstractas utilizando ontologías

**Cardenas, Marina Elizabeth - Marciszack, Marcelo Martín  
Castillo, Julio Javier - Vázquez, Juan Carlos**

**Dpto. de Ingeniería en Sistemas de Información  
Universidad Tecnológica Nacional, Facultad Regional Córdoba**

## **Abstract**

*El presente trabajo describe el desarrollo de un modelo de representación conceptual de un dominio específico, en este caso teoría de gramáticas formales y máquinas abstractas, a través del modelado ontológico, una reciente rama de la inteligencia artificial.*

*Este desarrollo es parte de un proyecto que nos permitirá comparar entre sí las diferentes metodologías y herramientas que suelen utilizarse en la etapa de relevamiento de requerimientos del proceso de desarrollo de software, y por otro lado, permitirá establecer el grado de correspondencia entre la conceptualización de las Máquinas y Gramáticas, ya que al existir un isomorfismo entre ambos dominios, éste debería continuar en los esquemas conceptuales resultantes de la aplicación de cada una de las metodologías comparadas.*

## **Palabras Clave**

Ontología, maquinas abstractas, gramáticas formales, modelo conceptual, captura de requerimientos, Protégé 2000.

## **Introducción**

Una ontología es un sistema de representación del conocimiento acerca de un dominio o ámbito específico, con el fin de obtener una representación formal de los conceptos que contiene y de las relaciones que existen entre los mismos.

Además, una ontología se construye en relación a un contexto de utilización especificando una conceptualización, por lo que cada ontología incorpora un punto de vista.

Todas las conceptualizaciones (definiciones, categorizaciones, jerarquías, propiedades, herencia, etc.) de una ontología pueden ser procesables e interpretadas por una computadora o un ser humano.

Un concepto muy asociado a este sistema de representación es la Web Semántica, la cual tiene como objetivo

principal la creación de un medio universal para el intercambio de información basado en representaciones del significado de los recursos que se encuentran en la Web, de una manera inteligible para las máquinas. Para que esto pueda llevarse a cabo, se necesita que el conocimiento de la web esté representado de forma que sea legible por las computadoras, esté consensuado, y sea reutilizable y es por ello que las ontologías proporcionan la vía para representarlo.

Para el diseño del modelo ontológico del presente, se establece como dominios a modelar, las Gramáticas Formales y Máquinas Abstractas. La elección de estos dominios tiene un doble propósito: por un lado nos permitirá comparar entre sí las diferentes metodologías y herramientas de la etapa de relevamiento de requerimientos del proceso de desarrollo de software y por otro lado, permitirá establecer el grado de correspondencia entre la conceptualización de las Máquinas y Gramáticas, ya que al existir un isomorfismo entre ambos dominios, éste debería continuar en los esquemas conceptuales resultantes de la aplicación de cada una de las metodologías comparadas.

## **Elementos de las ontologías**

Las ontologías proporcionan un vocabulario común de un área y definen, a diferentes niveles de formalismo, el significado de los términos y relaciones entre ellos.

El conocimiento en ontologías se formaliza principalmente usando cinco

tipos de componentes: *conceptos*, *relaciones*, *funciones*, *axiomas* e *instancias*.

Los *conceptos*, entidades o clases en la ontología se suelen organizar en taxonomías. Se suele usar tanto el término clases como conceptos. Un concepto puede ser algo sobre lo que se dice algo y, por lo tanto, también podría ser la descripción de una tarea, función, acción, estrategia, proceso de razonamiento, etc.

Las *relaciones* representan un tipo de interacción entre los conceptos del dominio. Como ejemplos clásicos de relaciones binarias podemos mencionar: “subclase de” y “conectado a”.

Las *funciones* son un tipo especial de relaciones en las que el *n*-ésimo elemento de la relación es único para los “*n*-1” precedentes.

Los *axiomas* son expresiones que son siempre ciertas. Pueden ser incluidas en una ontología con muchos propósitos, tales como definir el significado de los componentes ontológicos, definir restricciones complejas sobre los valores de los atributos, argumentos de relaciones, etc. verificando la corrección de la información especificada en la ontología o deduciendo nueva información.

Las *instancias* se usan para representar elementos específicos de la ontología.

### **Herramienta de modelado de ontologías: Protégé**

Las ontologías requieren de un lenguaje lógico y formal para ser expresadas.

En la inteligencia artificial se han desarrollado numerosos lenguajes para este fin, algunos basados en la lógica de predicados y otros basados en frames (taxonomías de clases y atributos), que tienen un mayor poder expresivo, pero menor poder de inferencia; e incluso existen lenguajes orientados al razonamiento.

Todos estos lenguajes han servido para desarrollar otros lenguajes aplicables

a la Web. Es por ello que para el desarrollo de este proyecto se ha optado por utilizar la herramienta Protégé que implementa el lenguaje OWL (**O**ntology **W**eb **L**anguage) para el modelado de ontologías basadas en Frames. La misma ha sido desarrollada por la Universidad de Stanford y se utiliza para el desarrollo de Ontologías y Sistemas basados en el conocimiento por medio de una interfaz de usuario que facilita la creación de estructuras de frames con clases, slots e instancias de una forma integrada.

Protégé permite:

- Modelar una ontología de las clases que describen un tema particular.
- Creación de una herramienta de adquisición de conocimiento para recoger conocimiento.
- Entrar en casos específicos de datos y de la creación de una base de conocimiento.
- La ejecución de usos.

### **Metodología**

Debido a la existencia de diversas metodologías de desarrollo de ontologías, se ha optado por la descrita en “*Ontology Development 101: A Guide to Creating Your First Ontology*” (por sus autores N. F. Noy y D. L. McGuinness) ya que se adapta mejor a los requerimientos de modelado, desde el punto de vista de la simplicidad y completitud de la misma.

Antes de proceder a la explicación de dicha metodología, es preciso definir una serie de reglas que ayudarán a tomar decisiones de diseño y modelado:

1. *No existe una forma correcta y única de modelar un dominio*, es por ello que la mejor solución casi siempre depende de la aplicación que se le dará a la ontología una vez modelada.

2. *El desarrollo de ontologías es un proceso necesariamente iterativo*. La ontología inicial evoluciona y se refina a través de las sucesivas iteraciones.

3. *Los conceptos en la ontología deben ser cercanos a los objetos (ya sean*

físicos o lógicos) y relaciones en el dominio de interés.

Tomando como base lo dicho anteriormente, a continuación se explicará la metodología adoptada para realizar el modelado de la ontología:

**Paso 1.** *Determinar el dominio y alcance de la ontología.*

Este paso ayuda a determinar el alcance del modelo, definiendo el ámbito de aplicación de la ontología con respecto a su dominio específico. Para ello se requiere la formulación de preguntas de competencia que ayudarán a limitar el alcance. Estas preguntas servirán después como prueba de control de calidad: ¿La ontología contiene suficiente información para responder esos tipos de preguntas? ¿Las respuestas requieren un nivel particular de detalle o representación de un área particular? Las preguntas de competencia son solamente un bosquejo y no necesitan ser exhaustivas.

**Paso 2.** *Considerar la reutilización de ontologías existentes.*

Reusar ontologías existentes puede ser un requerimiento si nuestro sistema necesita interactuar con otras aplicaciones que ya se han dedicado a ontologías particulares o vocabularios controlados. Muchas ontologías ya están disponibles en forma electrónica y pueden ser importadas dentro un entorno de desarrollo de ontologías.

**Paso 3.** *Enumerar términos importantes para la ontología*

En este paso es necesario escribir una lista con todos los términos que potencialmente servirán para el modelo de la ontología, estos términos son extraídos de la descripción del dominio de aplicación.

**Paso 4.** *Definir las clases y la jerarquía de clases.*

Hay varios posibles enfoques para desarrollar una jerarquía de clases:

- *Un proceso de desarrollo top-down* comienza con la definición de los conceptos mas generales en el dominio la

subsecuente especialización de los conceptos.

- *Un proceso de desarrollo bottom-up* comienza con la definición de las clases mas específicas, las hojas de la jerarquía, con el subsecuente agrupamiento de esas clases en conceptos mas generales.

- Un *proceso de desarrollo* combinado es el resultado de una combinación de los enfoques top-down y bottom-up: primero definimos los conceptos más sobresalientes y luego los generalizamos y especializamos apropiadamente.

Ninguno de esos tres métodos es inherentemente mejor que cualquiera de los otros. El enfoque a tomar depende fuertemente de la visión personal del dominio.

**Paso 5.** *Definir las propiedades de las clases: slots.*

Las clases aisladas no proveerán suficiente información para responder las preguntas de competencia del Paso 1. Una vez que hemos definido algunas de las clases, debemos describir la estructura interna de los conceptos.

Ya hemos seleccionado clases de la lista de términos creada en el Paso 3. La mayoría de los términos restantes son muy probablemente propiedades de esas clases.

En general, hay varios tipos de propiedades de objeto que pueden llegar a ser slots en una ontología:

- Propiedades “intrínsecas”.
- Propiedades “extrínsecas”.
- partes, si el objeto es estructurado; pueden ser “partes” físicas y abstractas.
- relaciones con otros individuos; éstas son las relaciones entre miembros individuales de una clase y otros ítems.

**Paso 6.** *Definir las facetas de los slots.*

Los slots pueden tener diferentes facetas que describen el tipo de valor, valores admitidos, el número de los valores (cardinalidad), y otras características de los valores que los slots pueden tomar.

**Paso 7.** *Crear instancias.*

El último paso consiste en crear instancias individuales de clases en la jerarquía. La definición de una instancia individual de una clase requiere (1) elegir una clase, (2) crear una instancia individual de la clase y (3) rellenar los valores del slot.

## Resultados

Debido a que las gramáticas proporcionan las reglas utilizadas en la generación de las cadenas de los lenguajes, es inmediata la relación entre estas Gramáticas Formales y las Maquinas Abstractas capaces de aceptarlos. Es así que los lenguajes son el puente que vincula ambos conceptos.

En el modelo ontológico obtenido a partir de la metodología descrita en la sección anterior, se puede observar que se ha representado este tipo de relaciones con el objetivo de determinar el isomorfismo entre los conceptos mencionados anteriormente.

Utilizando Protégé 2000 para el modelado de la ontología diseñada en el presente trabajo, se procedió a la construcción de la jerarquía de clases conceptuales vinculadas al dominio, tal como se muestra en la Figura 1.

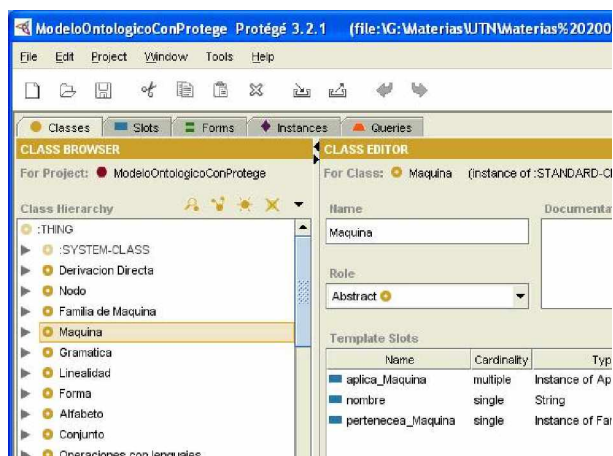


Figura 1. Modelo de clases en Protégé 2000

Como se muestra en la Figura 2, subsecuentemente se incorporaron los Slots a las clases conjuntamente con las facetas asociadas a cada uno. Las relaciones jerárquicas de herencia se pueden visualizar al hacer clic sobre una clase Padre (o también llamadas clases base).

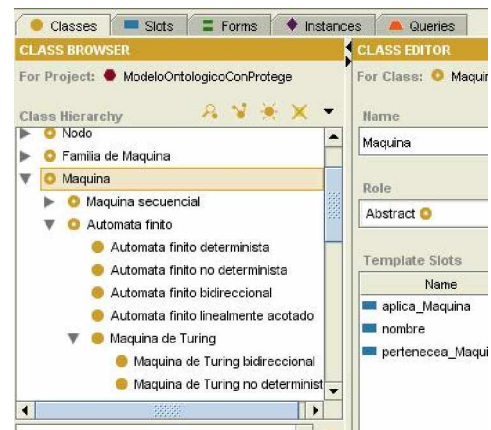


Figura 2. Taxonomía de clases en Protégé

Aquellas relaciones que no son de herencia entre los conceptos, son representadas a través de slots del tipo Instance, a los cuales se les define en su faceta, el tipo de clase con la cual se relacionan. Una vez que se realizó la carga del modelo completo, se crearon nuevas instancias para algunas clases con el objeto de efectuar pruebas. Entre las instancias que se crearon, corresponden a la clase Aplicación la cual representa las aplicaciones que tienen las Maquinas Abstractas en diferentes áreas. La Figura 3 permite observar el proceso de creación de instancias al modelo.

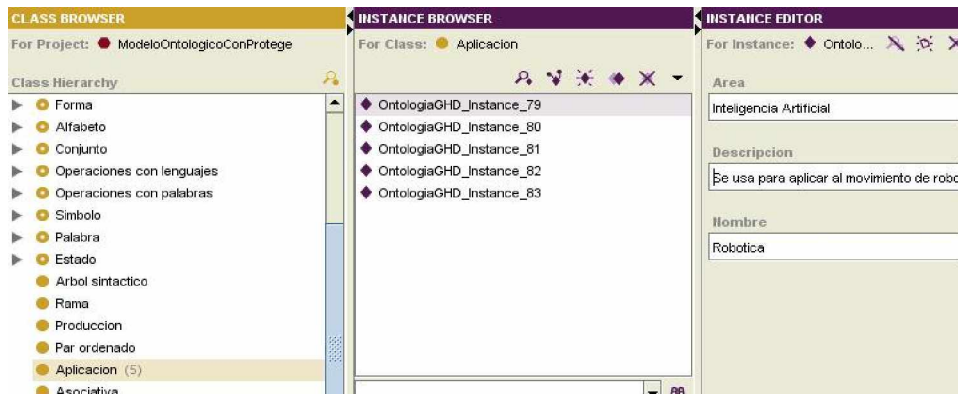


Figura 3. Creación de Instancias con Protégé 2000

Se puede observar en la figura que la instancia *OntologiaGHD\_Instance\_79* seleccionada en el Instance Browser corresponde a la clase *Aplicacion*, seleccionada en el Class Browser. En el Instance Editor se puede observar los campos que solicitan el ingreso de los datos para cada slot identificado con su nombre.

Una vez que se finalizó con la carga de instancias, se efectuó una prueba para evaluar la respuesta del modelo ontológico, para ello se ingresó una consulta en la pestaña de la ventana principal llamada Queries.

La consulta se puede visualizar en la Figura 4, y su resultado exitoso aparece en el borde derecho de la ventana. Se prevé en trabajos futuros, la utilización de programas razonadores de ontologías para poder inferir nuevo conocimiento a partir el modelo ontológico.

## Conclusión y trabajos futuros

En base al modelo ontologico obtenido, se prevé la ampliación y

profundización de los conceptos representados, continuando con el proceso iterativo que caracteriza a la metodología utilizada para su implementación. A partir de ello, se incorporará y refinará el dominio del problema que delimita el alcance de la ontología y se construirá un conjunto de queries de prueba del modelo ontologico para comprobar su performance. Una vez obtenido el modelo ontologico final, se procederá a compararlo con los obtenidos de otras metodologías y herramientas tales como las siguientes: LEL (Léxico Extendido del Lenguaje) , Escenarios y Tarjetas CRC, utilizando como herramienta de descripción al BMW (Baseline Mentor Workbench); casos de uso obtenidos a partir de la metodología RUP/UML (Rational Unified Process) con la utilización de Rational; y un modelo de objetos y diagramas de clase obtenido a partir de una definición Ad-hoc.

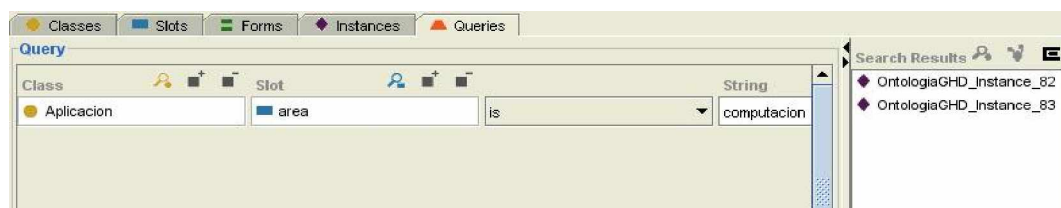


Figura 4. Creación de Queries con Protégé 2000

## Referencias

- [1] Introducción a Protégé. Universidad Politécnica de Valencia. 2003. Homepage: [http://personales.upv.es/ccarrasc/extdoc/p1\\_1\\_parte.pdf](http://personales.upv.es/ccarrasc/extdoc/p1_1_parte.pdf)
- [2] W3C. OWL Web Ontology Language. Overview. <http://www.w3.org/TR/owl-features/>  
[]OWL.Homepage: <http://www.hipertexto.info/documentos/owl.htm>
- [3] Ontologías 2.Ontologías en acción. Protégé – OWL. Alberto Barrón Cedeño (2005). Homepage: <http://theory.lcs.mit.edu/~rajsbaum/cursos/web/ontologias2.pdf>
- [4] Tesis Doctoral. (2005) Un Entorno para la Extracción Incremental de Conocimiento desde Texto en Lenguaje Natural. Rafael Valencia García
- [5] Noy, Natalya F., McGuinness, Deborah L.(2005). Ontology Development 101: A Guide to Creating Your First Ontology. Stanford University, Stanford.

## Datos de Contacto

*Marina E. Cardenas. UTN FRC. E-mail: [angelaesmeralda@gmail.com](mailto:angelaesmeralda@gmail.com)*

*Marcelo M. Marciszack. UTN FRC. E-mail: [marciszack@gmail.com](mailto:marciszack@gmail.com)*

*Julio Javier Castillo. UTN FRC. E-mail: [jotacastillo@gmail.com](mailto:jotacastillo@gmail.com)*

*Juan Carlos Vázquez. UTN FRC. E-mail: [jcvazquez@gmail.com](mailto:jcvazquez@gmail.com)*