

Tratamiento de la semántica emergente mediante sistemas de agentes basados en conocimiento^{*}

Gonzalo A. Aranda-Corral

Departamento de Ingeniería Informática, Sistemas Informáticos y Automática.
E.P.S La Rábida - Universidad de Huelva.
Crta Huelva-Palos de la Frontera s.n. 21819-Palos de la Frta.
gonzalo.aranda@diesia.uhu.es

Resumen En esta contribución se describe un proyecto de tesis para la integración de las tecnologías de la Web Semántica con la Semántica Emergente, que resulta de la actividad de las redes sociales actuales, mediante la utilización de sistemas multiagente. Estos sistemas multiagente estarán compuestos de agentes basados en conocimiento, de forma que el tratamiento de la semántica sea más adecuado.

Palabras clave: Sistemas multiagente, Web Semántica, Web 2.0, MetaWeb

1. Introducción

Uno de los grandes problemas en el estudio de la semántica emergente en la Web es el difícil encaje entre los dos flujos de conocimiento que se generan actualmente, y previsiblemente en el futuro:

- Por un lado, la Web Semántica provee de tecnologías para asegurar el tratamiento fiable del conocimiento, pero su rango de aplicabilidad está limitado a proyectos ligados a instituciones o a grandes empresas. El grado de sofisticación de las técnicas utilizadas es una barrera para su extensión a otros ámbitos, donde los usuarios no son expertos, y por tanto, pueden desistir de su uso o hacerlo de manera pobre o ineficiente.
- En el otro lado se encuentran las redes sociales en Internet, redes muy activas, generadoras de conocimiento de fiabilidad variable (que puede ir desde la Wikipedia¹ hasta la recomendación social usada en Digg²), y que usan técnicas semánticas muy débiles (*tagging*, taxonomías, microformatos, etc.) que no permiten que el ingente conocimiento que producen sea tratado con fiabilidad.

^{*} Parcialmente financiado por el proyecto TIC-137 del *Plan Andaluz de Investigación* y el proyecto TIN2004-03884 del Ministerio de Educación y Ciencia español, cofinanciado con fondos FEDER.

¹ <http://www.wikipedia.com>

² <http://www.digg.com>

Paradójicamente, el ritmo actual de crecimiento de los proyectos bajo la difusa denominación de Web 2.0 está haciendo económicamente interesante a las empresas que lo gestionan; y esas empresas no adoptan -por criterios de usabilidad y eficiencia- las soluciones que se ofrecen en el primer flujo, la Web Semántica, por lo que la brecha semántica se puede hacer insalvable.

2. Estado del Arte

Actualmente, existen diversas aproximaciones para salvar la brecha que se podrían clasificar, grosso modo, en las siguientes:

- Determinación de qué tipos de lenguajes pueden facilitar la transición entre la semántica emergente y la explícita. Esta aproximación está muy relacionada con los trabajos sobre decidibilidad[7] y complejidad³ de sublenguajes de OWL, y particularmente en su reflejo lógico (lógicas de la descripción), y la adopción de éstos en el ámbito de la Biotecnología, Sistemas de Información Geográfica, etc.
- Extensión moderada de los sistemas de anotación semántica usados en los proyectos Web 2.0. Está claro que de esa aproximación debería ser responsable la propia empresa. Por ejemplo, la extensión de la anotación de *flickr*⁴.
- Fortalecer los mecanismos de representación, y de razonamiento, partiendo de las técnicas semánticas de cada proyecto o sistema utilizado. Por ejemplo, semantizando el blog⁵, las wikis[9], especializando el razonamiento conceptual [6].

Cito dentro de esta aproximación la contribución [4] donde se establece, por primera vez, una conexión lógica para asegurar la adecuación de reparaciones de ontologías mediante representaciones visuales, pues la implementación de una herramienta basada en dicha conexión será parte integrante de la solución que se propone en el presente proyecto.

Y finalmente, es necesario destacar los incipientes proyectos *Metaweb*, proyectos que combinan (o mejor dicho, combinarán) directamente la dimensión social y Web 2.0 con la de la Web Semántica. La *metaweb* (véase fig. 1)⁶ vendría a paliar (al menos en teoría) la brecha utilizando activamente la gestión socializada del conocimiento. En este ámbito destacaríamos Freebase⁷, que ya está a disposición pública para su uso. Mi director de tesis⁸ fue invitado a participar en la fase de testeo de dicho portal.

³ <http://www.w3.org/Submission/owl11-tractable/>

⁴ <http://www.flickr.com>, y la noticia en cuestión se puede leer en <http://www.flickr.com/groups/api/discuss/72157594497877875/>

⁵ Por ejemplo, <http://www.semanticblogging.org/semblog/blog/default/>

⁶ http://novaspivack.typepad.com/nova_spivacks_weblog/metaweb_graph.GIF

⁷ <http://www.freebase.com/>

⁸ <http://www.cs.us.es/~joaquin/blog/index.php/2007/04/11/un-producto-web-semantica-web-20/>

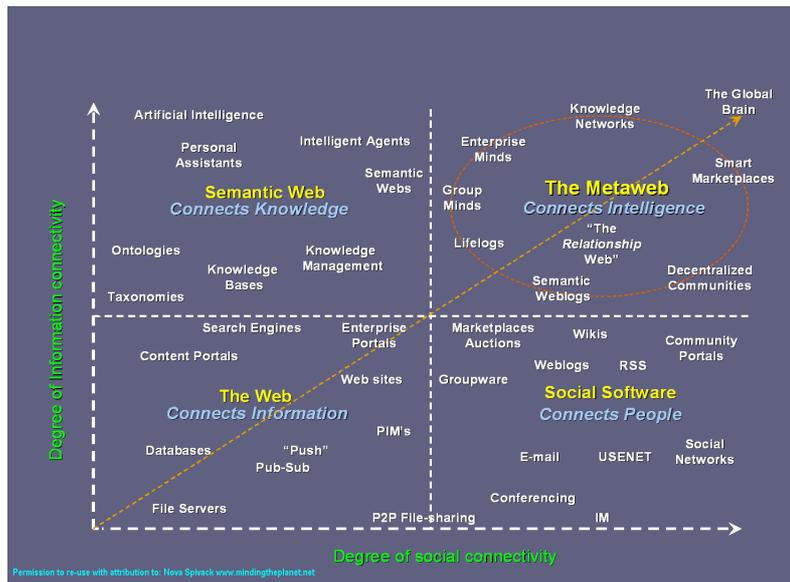


Figura 1. La metaweb y su relación con la Web Semántica y la Web 2.0⁶

3. Contribución esperable del proyecto

La contribución sería un sistema multiagente para el tratamiento de la emergencia semántica en redes sociales y para el enriquecimiento semántico de las ontologías extraídas. Los agentes serían, fundamentalmente, basados en conocimiento para así poder racionalizar y planificar la gestión de las estructuras de conocimiento.

4. Enfoque propuesto

El presente proyecto se basa en la idea de delegar en agentes ciertas tareas relativas al establecimiento de una semántica robusta. Más concretamente:

- Se pretende que la supervisión de las estructuras ontológicas básicas siga bajo supervisión del usuario aunque con cierto grado de contextualización.
- La gestión y la inducción de cambios sería responsabilidad de un Sistema MultiAgente (SMA) que, como capa superior, supervisaría la integridad lógica de la ontología emergente y la podría sintetizar, bajo ciertas condiciones.

De esta forma se podría conseguir una capa de fiabilidad, gestionada por este SMA, en el tratamiento de la semántica emergente.

4.1. Metodología de Investigación

Las fases de este proyecto serían las siguientes:

1. Análisis preliminar, desde el punto de vista de la representación del conocimiento y de la usabilidad, de las facetas semánticas de distintos proyectos Web 2.0.
2. Diseño e implementación de una herramienta para la representación y reparación visual de ontologías ligeras.
3. Desarrollo de la teoría (interpretable sobre el lenguaje FLUX⁹ [8]), del tipo de agentes que tratarán con el conocimiento.
4. Diseño del sistema multiagente asociado a una red social, formado por agentes específicos del apartado anterior.
5. Evaluación y validación.

Actualmente el proyecto se encuentra en la etapa 2. La creación de esta aplicación tendrá como origen de los datos las demostraciones obtenidas de un razonador automático conectado con la ontología, representando gráficamente a los conceptos y a los individuos. Esta representación podrá ser modificada haciendo uso del interfaz gráfico y finalmente, reflejar los cambios dentro de la ontología.

4.2. Descripción de la aplicación

Uno de los requisitos que nos planteamos antes de comenzar el desarrollo de la aplicación fue que los resultados que se obtuvieran fueran realmente usables y exportables a otras plataformas o programas. Por todo ello, se comenzó el desarrollo de ésta como un plug-in de Protégé¹⁰.

Protégé es el editor más extendido y popular de ontologías, gratuito y de código abierto, por lo que nos ofrece la fiabilidad para el acceso a cualquier parte del mismo. Además, la documentación es bastante completa y facilita el desarrollo de extensiones empotrables dentro de la aplicación, tanto para su uso como para su distribución.

Dentro de los diferentes tipos de plug-ins elegimos el *Tab-Widget* (pestaña) por lo que podremos usar toda la zona de trabajo para el interface gráfico de nuestra aplicación.

Una imagen inicial de la aplicación la podemos ver en la figura 2 en la que se nos muestra la ejecución de Protégé con el plug-in incorporado. Disponemos de dos ventanas en la parte izquierda que las vamos a rellenar con las clases, o conceptos, en la parte superior e individuos en la parte inferior.

⁹ <http://www.fluxagent.org/>

¹⁰ <http://protege.stanford.edu>

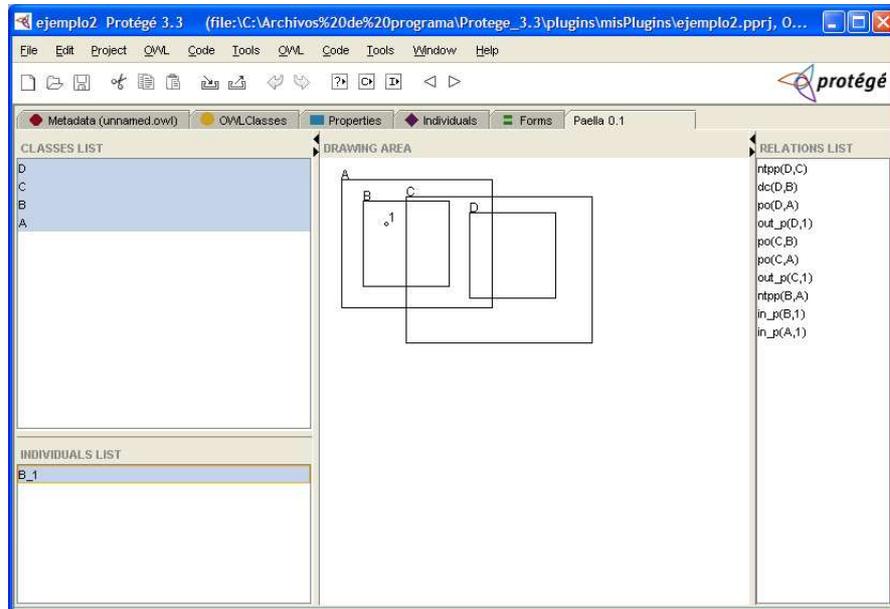


Figura 2. Captura de pantalla de Protégé con Paella

Todos estos datos los obtenemos de la representación interna de la ontología a la que tenemos acceso dentro de la jerarquía de java implementada dentro de Protégé. El objeto que contiene toda esta información es el OWLModel.

El proceso de depuración que queremos implementar con esta herramienta es el descrito esquemáticamente en la figura 3 y que lo describiríamos en 4 pasos principales.

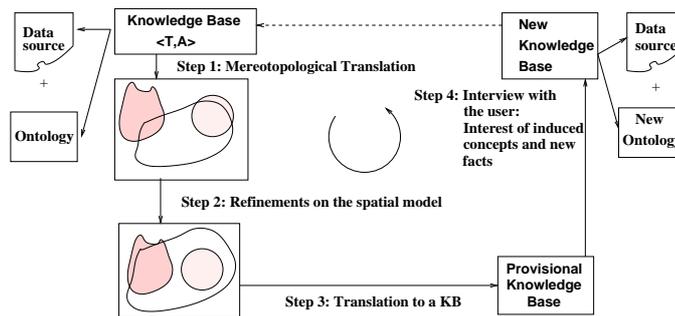


Figura 3. Proceso de depuración

El primer paso sería la “*Traducción Mereotológica*”, proceso por el cual obtendremos una representación gráfica de la base de conocimiento (KB) que queremos estudiar. A partir de esta representación se le permitirá al usuario que modifique esta representación en el paso de “*Refinamiento del modelo espacial*” para dar lugar a una base de conocimiento provisional mediante la traducción, que se realiza en el tercer paso. Una vez tenemos la base de conocimiento provisional, realizamos una serie de preguntas al usuario para la correcta inserción, o integración, de esta base provisional en la KB definitiva y depurada.

Paso 1. En el primer paso creamos un objeto *Teoria* donde almacenamos, en un principio, las clases y los objetos que seleccionamos del modelo, en la ventana correspondiente. Una vez seleccionados éstos, calculamos las relaciones espaciales entre todos los conceptos e individuos, que estarán enmarcadas dentro de RCC8. Las posibles posiciones relativas de RCC8 vienen representadas en la figura 4. El cálculo de estas relaciones lo realizamos mediante preguntas a un razonador

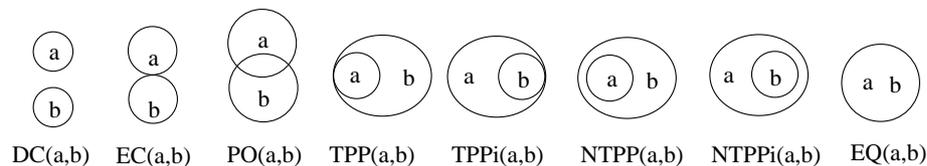


Figura 4. Posiciones relativas dentro de RCC8

externo y que enlazamos usando, también, las clases desarrolladas dentro de Protégé. Este enlace se configura dentro del menú de Protégé.

Preguntamos al razonador y calculamos todas las posiciones relativas de estos elementos y, ya que la ontología es consistente, el dibujo que resulta es representable gráficamente. Almacenamos la lista de posiciones relativas dentro de nuestra *Teoria* para seguir usándola en este primer paso, además de visualizarlas en el panel de la derecha.

Ahora nos quedaría calcular las figuras y las posiciones espaciales correspondientes a esa lista de relaciones. Para este paso nos basaremos en la técnica de programación con restricciones (CHR), y usaremos el lenguaje de programación lógica Prolog para resolverlo. La implementación de Prolog usada es la SWI-Prolog¹¹, elegida por razones similares a las de Protégé, aunque el módulo de restricciones de éste no está completamente depurado y no descartamos migrar a otra implementación de Prolog como GNU-Prolog¹² en un futuro cercano.

Traducimos las relaciones espaciales genéricas de RCC8 a ecuaciones de restricción y las tenemos almacenadas en un fichero aislado, y luego las ecuaciones de las restricciones de nuestro problema en otro fichero temporal que se escribirá

¹¹ <http://www.swi-prolog.org>

¹² <http://www.gnu.org/software/gprolog/>

dinámicamente cada vez que se calculen nuevas relaciones. Finalmente se llama a Prolog para preguntarle cuál es el resultado de esas restricciones, en forma de coordenadas que pintaremos en nuestro área de dibujo. El área de dibujo es el área central blanca de nuestra aplicación

Ya está completado el paso 1 y podemos proceder con la manipulación manual de los datos.

Paso 2. Una vez dibujada la representación de los conceptos e individuos dentro de nuestra área de dibujo, conservando las relaciones, El usuario podrá realizar los movimientos de éstos para que se ajusten a la representación del mundo que quiere el usuario.

Cuando se representa un trozo del mundo debemos de distinguir varias conceptualizaciones o abstracciones (ABS). Es decir, el usuario observa el mundo y se hace un modelo mental (M) de lo que está observando e intenta representarlo haciendo una abstracción de ese modelo (ABS(M)). Esta abstracción no tiene

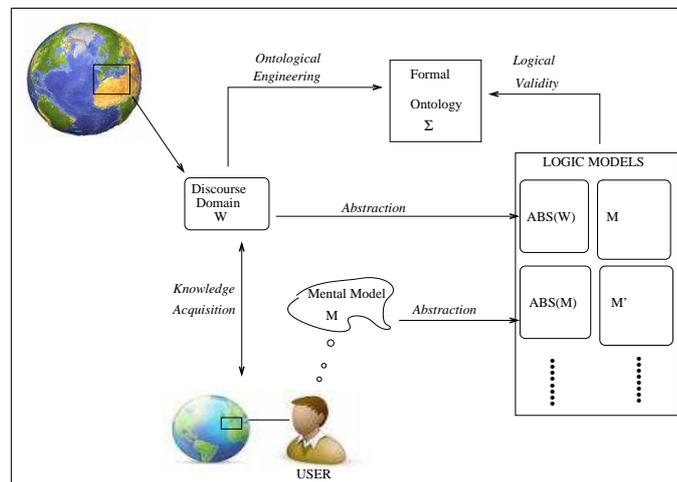


Figura 5. Diferencia entre la abstracción del usuario y el mundo.

por que no corresponder con la abstracción real del mundo (ABS(W)) ya que el usuario puede no ver la realidad completa o simplemente cometa algún tipo de error u omisión a la hora de plasmar su abstracción.

La representación espacial de los conceptos, abstraídos por el usuario (ABS(M)), nos puede ayudar a ver más claramente la relación entre éstos y podemos advertir las diferencias con nuestra idea del modelo.

Si éste detecta alguna diferencia entre lo que era su modelo mental y la abstracción realizada, tendrá dos tipos de movimientos que podrá ejecutar dentro de la aplicación. que son la *traslación* y el *dimensionamiento*. En la traslación cambiamos el objeto de sitio de forma que podemos solaparlo con otro que no lo

hiciese anteriormente o desconectarlo del mismo, e incluso de todos. En el caso de individuos podemos hacer que un individuo pertenezca a una clase moviéndolo dentro del rectángulo, o no sacándolo de él.

El segundo movimiento, dimensionado, consiste en expandir una región para que solape o incluso contenga a una zona, o contraerla para que no se relacione con las otras. Los individuos no tienen posibilidad de redimensionamiento, ya que son puntos.

Todos estos movimientos son libres y se producen pinchando con el botón izquierdo o derecho en la forma de la figura a modificar. El movimiento terminará cuando dejemos de pulsar el botón correspondiente.

Paso 3. Una vez llegados a este punto, la representación espacial que hemos obtenido en la zona de dibujo es la que corresponde con el modelo mental(M) que el usuario quería realizar y ya sólo nos queda grabar esos cambios realizados y plasmarlos dentro de la ontología.

Mediante interpretación de las coordenadas de los elementos representados volvemos a obtener la nueva lista de relaciones RCC8 de posiciones relativas. Esta nueva lista, que almacenamos en nuestra *Teoría*, es la que tomamos como base de conocimiento temporal, que referíamos anteriormente.

Paso 4. Tal como viene detallado en [5] y [1], los arreglos espaciales se traducen en fórmulas de lógicas de descripción, y junto con la propuesta de inserción de individuos y/o áreas para la correcta traducción de la representación.

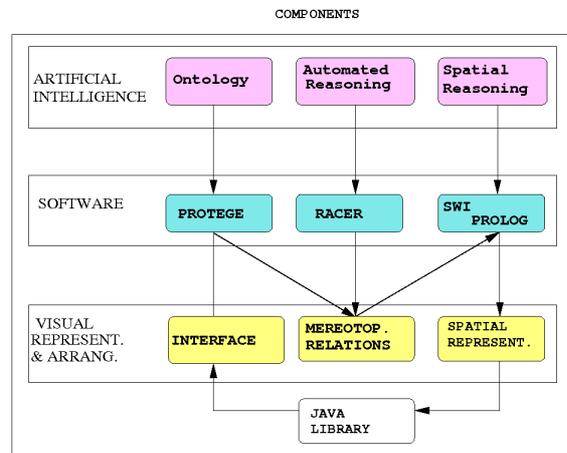


Figura 6. Arquitectura de la aplicación

La figura 6 sería un vistazo general a la arquitectura de la aplicación en estos cuatro primeros pasos actuales.

En estos momentos nos encontramos, concretamente, por la implementación de este punto.

En un futuro la sugerencia de estos arreglos estarán a cargo del sistema multiagentes(SMA), que serán los responsables del mantenimiento de la información correcta y actualizada de nuestra ontología.

Para la implementación del sistema multiagentes nos proponemos integrar la tecnología FLUX, que está basada en Eclipse-Prolog, dentro de la tecnología y plataforma multiagentes JADE. Esto supondrá también un reto tecnológico y que esta fundamentado en los beneficios de poder usar una plataforma que soporte todos los estándares FIPA, además de todas las librerías desarrolladas que nos encapsulan y facilitan la tarea diseñar agentes con capacidades de comunicación, cooperación, movilidad, etc...

5. Comparación con otros similares

En principio, no existen aproximaciones donde se apliquen intensivamente agentes racionales (organizados en un SMA) para la gestión autónoma de la semántica emergente, aunque existen diversas soluciones parciales basadas fundamentalmente en sistemas no agentificados o incluso no agentificables, que son de uso común en la minería de ontologías o en la anotación automática de recursos.

6. Beneficios esperables

La obtención de un SMA de este tipo podría suponer un avance importante en el tratamiento de la semántica emergente, particularmente si se aplicara en organizaciones humanas que comienzan a utilizar las nuevas herramientas de colaboración en red para la gestión de su conocimiento. Por ejemplo, en sistemas de blogs para empresas que por alguna razón, no adoptan plenamente tecnologías de la Web Semántica. En especial, un análisis preliminar parece indicar que incluso se podrían descubrir nuevos servicios que la empresa estaría en condiciones de ofrecer [2].

Por último, aunque no ha sido explícitamente indicado en la metodología (básicamente porque no es un problema relacionado con el tema del simposio ni de la tesis), se pretende obtener, a partir de esta experiencia, una metodología para la construcción de SMA de agentes FLUX que proporcionen soluciones a otro tipo de problemas. De esta forma se reutilizaría la integración JADE[3]-FLUX en otros ámbitos de aplicación (ROBOCUP¹³, resolución multiagente de problemas, juegos, etc.).

¹³ <http://www.robocup.org>

7. Conclusiones y trabajo futuro

En el futuro más inmediato del proyecto, los elementos principales que participarán (a saber, agentes racionales FLUX y el SMA JADE¹⁴) parecen indicados para soportar la implementación del sistema propuesto. Si bien existen dificultades relativas a la programación en un lenguaje de fuentes de algunos de los razonamientos híbridos ontologías/creencias/conocimientos, el campo de aplicación (semánticas débiles asociadas a proyectos Web 2.0) hace viable el establecimiento de una teoría adecuada para soportar la creación de tales agentes.

Referencias

1. J.A. Alonso-Jiménez, J. Borrego-Díaz and A.M.Chávez-González, Ontology cleaning by mereotopological reasoning, *Proc. of DEXA Workshop on Web Semantics(WebS'04)* pp 132-137, IEEE Computer Society (2004).
2. G.A. Aranda-Corral, J.Borrego-Díaz and A.Chávez-González, Learning Services by Formal Concept Analysis, *IADIS Conf. Intelligent Systems and Agents*, Lisboa, Julio de 2007
3. F. Bellifemine, G.Caire, D. Greenwood, Developing multi-agent systems with JADE, *Wiley Series in Agent Technology*, Wiley (2007).
4. J. Borrego-Díaz and A. M. Chávez-González, Visual Ontology Cleaning: Cognitive Principles and Applicability, *3rd European Semantic Web Conf. (ESWC'06)*, LNCS 4011 pp. 317-331 (2006).
5. A.M. Chávez-González, Razonamiento mereotopológico automatizado para la depuración de ontologías, *Tesis*. Universidad de Sevilla, 2005.
6. E. Giunchiglia, F. Giunchiglia y M. Yatskevich, Semantic Matching, *Proc. European Semantic Web Congress 2005*, Lecture Notes in Computer Science n. 3532, pp. 272-289 (2005).
7. I. Horrocks and P. Patel-Schneider, Reducing OWL entailment to description logic satisfiability, *Journal of Web Semantics*, 1(4):345-357, 2004
8. M. Thielscher, Reasoning Robots, *Applied Logic Series*, Springer (2005).
9. M. Völkel, M. Krötzsch, D. Vrandečić, H. Haller, y R. Studer, Semantic Wikipedia, *Proc. 15th Int. Conf. on World Wide Web, WWW 2006*, Edinburgh, Scotland, May 23-26, 2006. May 2006.

¹⁴ <http://jade.tilab.com>