

Una introducción a la Web semántica

Miller Gómez Mora*

José Nelson Pérez Castillo**

Resumen

Se han desarrollado aplicaciones exitosas en ambientes distribuidos abiertos aplicando tecnologías de la Web semántica, lo que ha conllevado a su potenciación en la solución a problemas, en la cual la computación tradicional ha logrado resultados limitados. En este documento se hace una revisión de la concepción del desarrollo de la Web semántica para luego estudiar las tecnologías más relevantes que la hacen posible, como es XML, RDF y las ontologías. Además, se describe las principales características de Protégé, una herramienta para el desarrollo de ontologías. Asimismo, se aborda el tema de agentes inteligentes, que puede ser de gran importancia en el proceso automático para el descubrimiento, composición y orquestación de servicios Web semánticos.

Palabras claves: Web semántica, ontología, Agentes Inteligentes, XML, RDF.

Abstract

Successful applications in opened distributed environment have been developed applying technologies of the Semantic Web, which has entailed to its involution in the solution of problems where the traditional computation has obtained limited results. In this document is made a revision of the conception of the development of the Semantic Web and study the technologies more excellent than they make it possible, as XML, RDF and Ontologies. In addition, it describes the main characteristics of Protege, a tool for the development of Ontologies. Also, the subject of intelligen agents is approached, which can be of big importance in the automatic process for the discovery, composition and choreographic of Semantic Web Services.

Keywords: Semantic Web, Ontology, Intelligent Agents, XML, RDF.

* Ingeniero de sistemas. Especialista en Teleinformática. Candidato a Magíster en Ciencias de Información y las Comunicaciones (Universidad Distrital). Profesor asistente Universidad Distrital. Correo electrónico: mgomez@udistrital.edu.co.

** Ingeniero de Sistemas. Magíster en Teleinformática. PhD en Informática (Universidad de Oviedo). Profesor Asociado Universidad Distrital. Correo electrónico: jperez@udistrital.edu.co.

1. Introducción

Se puede observar cómo al codificar las páginas Web actuales, por ejemplo con HTML¹, éstas tienen muy poco sentido para las máquinas. Así al mirar el código fuente de una página Web se puede encontrar el siguiente código:

Estudiando la Web Semántica

Al ser interpretado por un computador a través de un navegador (*browser*)² aparecerá un texto en negrilla y cursiva como este:

1.1 Estudiando la Web semántica

Esto es todo lo que pueden hacer los computadores con las páginas HTML. Sin embargo, se pueden construir índices con las palabras que aparecen en las páginas Web. Así, cuando un usuario envía una pregunta a través de un motor de búsqueda³, lo que hace dicho buscador es comparar las palabras de la pregunta con las de su índice.

1 HTML es un lenguaje de marcado basado en el SGML (Standard Generalized Markup Language), al que se le agregaron vínculos de hipertexto. Cada browser presenta la página escrita en HTML de acuerdo con su estructura de marcado.

2 El navegador puede considerarse como una interfaz de usuario universal. Dentro de sus funciones están la petición de las páginas Web, la representación adecuada de sus contenidos y la gestión de los posibles errores que se puedan producir.

3 Un motor de búsqueda es un conjunto de programas coordinados que se encargan de visitar cada uno de los sitios que integran la Web, empleando los propios hipervínculos contenidos en las páginas Web para buscar y leer otros sitios, crear un enorme índice de todas las páginas que han sido leídas y registradas, llamado comúnmente catálogo. Como ejemplos se tienen Altavista y Google.

Esto hace pensar que la asombrosa eficiencia de los motores de búsqueda actuales tiene también sus límites, por ejemplo, si un estudiante de programación de computadores quiere saber la fecha de aparición de la primera versión Java⁴ accede, por ejemplo, a GOOGLE y digita la pregunta: "java history". Esto hará que GOOGLE compare las palabras de la pregunta con los de su índice, si encuentra un documento que tenga "history" y "java" lo devolverá como respuesta, en caso contrario no sucederá nada.

Puede que se necesite realizar varias consultas y leer varios documentos antes de llegar a la respuesta deseada. Con estas limitaciones, la búsqueda en Internet puede ser algo tediosa y frustrante, esto debido, como ya se mencionó, a la falta de capacidad de las representaciones en que se basa la Web actual para expresar significados [2]. La Web semántica quiere solucionar esto.

2. La Web semántica

Se trata de una corriente promovida por el propio inventor de la Web y presidente del consorcio W3C⁵, Tim Berners-Lee, cuyo fin último es lograr que las máquinas puedan entender y, por tanto, utilizar los contenidos y servicios disponibles en la red, sin importar el idioma en que se encuentren escritos [1]. Para cumplir con este objetivo, es necesario contar con las tecnologías Web que

4 Java fue diseñado en 1990 por James Gosling, de Sun Microsystems, como software para dispositivos electrónicos de consumo. Curiosamente, todo este lenguaje fue diseñado antes de que diese comienzo la era World Wide Web, puesto que fue diseñado para dispositivos electrónicos como calculadoras, microondas y la televisión interactiva.

5 El W3C creado en 1994 tiene como misión guiar la Web hacia su máximo potencial, a través del desarrollo de protocolos y pautas que aseguren el crecimiento futuro de la misma. El W3C celebró su sexta semana técnica plenaria anual del 27 de febrero al 3 de marzo de 2006 en Cannes-Mandelieu, Francia.

permitan introducir descripciones explícitas sobre el significado de los recursos disponibles en la Red. Estas tecnologías incluyen lenguajes para la representación de ontologías, *parsers*, lenguajes de consulta, entornos de desarrollo, módulos de gestión para el almacenamiento, acceso y actualización de ontologías, módulos de visualización, conversión de ontologías, y otras herramientas y librerías. Existen muchos ejemplos de estas tecnologías, sin embargo, en este documento estudiaremos las relacionadas con XML, RDF y la creación de ontologías.

2.1 XML⁶

Es un lenguaje de marcado basado en SGML⁷, que permite estructurar datos y documentos en forma de árboles de etiquetas con atributos. XML permite diseñar lenguajes de marcado muy estructurados y específicos, en los cuales en lugar de etiquetas como `` e `<i>`, podrían ser `<titulo>`, `<autor>`, `<lugar>`, `<fecha>`, etc. Por lo anterior, y otras razones que adelante se explicarán, XML se considera un metalenguaje, puesto que permite definir lenguajes específicos, es decir, conjuntos de etiquetas determinados para cada área de conocimiento.

XML es un primer paso hacia una representación explícita de los datos y la estructura de los contenidos de la Web, separada de la presentación en HTML. En definitiva, es un lenguaje con multitud de capacidades a ni-

vel sintáctico, pero no aporta ningún mecanismo para tratar la información a nivel semántico, ya que su modelo de datos consiste en un árbol que no distingue entre objetos y relaciones, ni tiene noción de jerarquías de clases. Por esto los documentos escritos en XML es necesario validarlos mediante Dtd (Document Type Definitions) o XML schemas [3].

XML, por sí mismo, no tiene ninguna característica especial que lo convierta en un lenguaje idóneo para la representación de ontologías, aunque sí ofrece una forma simple y poderosa para especificar la sintaxis de un lenguaje de especificación de ontologías [4]. Además, XML es un lenguaje ideal para ser usado como lenguaje de transporte o de intercambio de información, ya que permite explotar al máximo las capacidades del WWW. Sin embargo, su importancia estriba en que permite encapsular instrucciones de programación, ya sea VBScript o Java applets, compilar controles ActiveX e incluso lenguajes antiguos como Cobol.

Otro elemento para tener en cuenta asociado con XML son los metadatos⁸. A continuación se dará una explicación a este término: si las páginas Web contaran con metadatos del tipo `<titulo>`, `<autor>`, `<tema>`, `<fecha de publicación>`, etc., los usuarios podrían hacer preguntas más precisas a los motores de búsqueda, pero los metadatos actuales no tienen ni semántica, ni sintaxis,

6 XML representa una primera aproximación a la Web semántica y aunque no está expresamente pensado para definir ontologías, es el estándar más extendido hoy día en las aplicaciones de esta línea.

7 SGML fue desarrollado en sus orígenes (1969) por Charles Goldfarb con el nombre de *General Markup Language* (GML), y tenía como objetivo facilitar el intercambio de documentos en IBM.

8 Los metadatos son datos altamente estructurados que describen información (el contenido, la calidad y la condición) y otras características de los datos. Los metadatos pueden ser almacenados dentro de una base de datos con una referencia al documento completo o ser incluidos en un encabezado dentro del propio texto. En el contexto Web, los metadatos como índices y URL son formados y almacenados, principalmente para la implementación de motores de búsqueda.

ni están unificados bajo una norma común que los agrupe. La solución recomendada por W3C es utilizar RDF (Resource Description Framework) [12].

2.2 RDF⁹

Es un lenguaje de etiquetado creado mediante sintaxis XML que define un modelo de datos para describir recursos mediante conjuntos de tres elementos (tríos), constituido cada uno de ellos por un sujeto, un verbo (objeto) y un predicado. Estos tríos pueden escribirse usando etiquetas de XML. RDF especifica una gramática lógica para que los autores de páginas Web puedan describir las propiedades semánticas de los documentos con una notación estándar y común para cualquier tipo de metadatos.

RDF utiliza los URI¹⁰ (Universal Resource Identifier) para dirigir, pueden incluso describir las cosas que no están en el Web mientras tengan una URI. Se prevé que los RDF describirán servicios y la funcionalidad de los dispositivos con el propósito de descubrir servicios. La esencia de RDF es, pues, un modelo formal para la representación de las propiedades y los valores de esas propiedades. Una de las representaciones clásicas del modelo RDF es la de un grafo dirigido y etiquetado. RDF usa los *namespaces* XML para calificar los recursos. Esto previene colisiones de nombre de elemento que podrían suscitarse cuando se com-

binan múltiples vocabularios. Los *namespaces* también se usan para identificar el esquema RDF.

Para la declaración de esquemas RDF se utiliza el lenguaje RDF-Schema. El modelo de datos de RDF está basado *en frames* y provee un mecanismo para la definición de relaciones entre propiedades y recursos. La *RDF Schema Specification* es una extensión de RDF que proporciona adicionales primitivas, enriquece el modelo básico y proporciona un vocabulario para RDF, que se asume tiene una cierta semántica. Asimismo, permite a los diseñadores especificar una jerarquía explícita de clases de recursos y propiedades que describen estas clases, junto con las restricciones sobre las combinaciones permitidas de clases, propiedades y valores.

En este contexto un problema clave es alcanzar un entendimiento entre las partes que han de participar en la construcción y explotación de la Web (usuarios y desarrolladores). De igual manera, los sistemas informáticos deben disponer de un mecanismo que permita descubrir diferentes significados dependiendo de la base de datos a la que se conecten. En este caso nos interesa encontrar una forma de representar, semántica común en estructuras diferentes. La solución a este problema se puede hallar mediante la utilización de ontologías.

2.3 Ontologías

Las ontologías se usaron por primera vez en informática en el campo de la Inteligencia Artificial para facilitar el intercambio y reutilización del conocimiento. A partir de los años noventa se convirtieron en un campo de investigación muy popular y muchas comunidades de investigadores de diferentes campos relacionados con la Inteligencia

9 En 1999 se publicó la primera versión de RDF, un lenguaje para la definición de ontologías y metadatos en la web. RDF es hoy el estándar más popular y extendido en la comunidad de la web semántica.

10 Cada sujeto y cada predicado son identificados por el URI (Universal Resource Identifier), dentro de estos el más difundido es el subconjunto URL.

Artificial, como la ingeniería del conocimiento, el procesamiento de lenguaje natural o la representación del conocimiento, las estudiaron [5].

Gruber define una ontología como "a formal explicit specification of a shared conceptualization" [7]. El producto de esta conceptualización, es llamado una ontología, que es un catálogo o conjunto de tipos de cosas que se asumen existen en un dominio de interés D, desde la perspectiva de una persona que usa un lenguaje L con el propósito de hablar sobre D [8]. Los tipos en la ontología representan los predicados, palabras con sentido, conceptos y tipos de relaciones del lenguaje L que son usadas para discutir temas en el dominio D.

Es decir, una ontología es una jerarquía de conceptos con atributos y relaciones que define una terminología bajo algún consenso para definir redes semánticas en áreas de conocimiento interrelacionadas. Esto se puede hacer gracias a que las ontologías tienen una forma de establecer el alcance de sus atributos y comportamiento por medio de reglas, permitiendo que la ontología deduzca o, por lo menos, proponga a qué clase o categoría puede pertenecer cada nuevo concepto que se ingresa.

Se distinguen tres tipos fundamentales de ontologías [17]:

1. Ontologías de un dominio, en las que se representa el conocimiento especializado pertinente de un dominio o subdominio, como la medicina, las aplicaciones militares, el comercio electrónico o la educación virtual.
2. Ontologías genéricas, en las que se representan conceptos generales y funcionales del conocimiento como las es-

tructuras parte/todo, la cuantificación, los procesos o los tipos de objetos.

3. Ontologías de representación, en las que se especifican las conceptualizaciones que subyacen a los formalismos de representación del conocimiento, por lo que también se denominan meta-ontologías (*meta-level o top-level ontologies*).

A estos tres tipos, se añaden las ontologías que han sido creadas para una actividad o tarea específica denominadas *task ontologies*, como por ejemplo, la venta de productos o el diagnóstico de una enfermedad, y las ontologías creadas para una aplicación específica. Para la creación de ontologías existen diferentes enfoques, basados en la experiencia de expertos y el desarrollo cooperativo [6]. El aspecto social para el desarrollo de las nuevas ontologías es muy importante, ya que existen diferencias en las comunidades que componen el dominio, en el que se crea la ontología [4]. La tendencia en la evolución de las ontologías es desarrollar múltiples ontologías locales, para luego, si es necesario, lograr acuerdos entre los miembros de las comunidades o formalizar las equivalencias entre las ontologías.

Utilizando esquemas RDF es posible crear ontologías, pero presenta deficiencias. Por ejemplo, usando RDF no se puede especificar tipos de datos para los nodos literales y no tiene soporte alguno a axiomas lógicos. El W3C ha desarrollado OWL¹¹, que se pretende

¹¹ En julio de 2002, el grupo de trabajo *WenOnto* del W3C publica el primer borrador de OWL (*Ontology Web Language*), lenguaje derivado de *Daml + OIL* y cimentado sobre *RDF schemas*. OWL es un lenguaje de marcado para la publicación de ontologías en la WWW y tiene como objetivo facilitar un modelo de marcado, construido sobre RDF y codificado en XML que permita representar ontologías, a partir de

sea el lenguaje estándar para ontologías en la Web. OWL cuenta con tres versiones; en realidad, se trata de versiones con distintas limitantes: OWL Lite, OWL DL y OWL Full. La única versión que garantiza computabilidad y decidibilidad es OWL Lite. La complejidad y capacidad de expresión se eleva hasta llegar a OWL Full, en el que la garantía -antes señalada- pierde veracidad. Al igual que en los lenguajes formales, en OWL las variantes tienen una relación contenedora, es decir, toda ontología OWL Lite es OWL DL, toda ontología OWL DL es OWL Full.

Escribir en lenguajes como RDF y OWL resulta sumamente difícil y propenso a errores [9]. Afortunadamente, se pueden utilizar entornos gráficos para visualizar y construir ontologías de forma más razonable, como Kaon, webODE o Protégé. De todas ellas Protégé, desarrollada por el Stanford Medical Informatics de la Universidad de Stanford, es la herramienta de construcción de ontologías que más usuarios tiene actualmente. Con ella, fácilmente se pueden crear clases y jerarquías, declarar propiedades para las clases, crear instancias e introducir valores, todo ello, en un entorno de menús, botones, cuadros de diálogos y representaciones gráficas fáciles de usar [11].

A continuación se presenta una descripción general de esta herramienta

Nombre: Protégé-2000

Versión y fecha de lanzamiento de la versión: Release 1.9. 8 de agosto de 2003.

un vocabulario más amplio y una sintaxis más fuerte que la que permite RDF. Por este motivo OWL puede ser utilizado para representar de forma explícita el significado de términos pertenecientes a un vocabulario y definir las relaciones que existen entre ellos.

Licencia: El código fuente es gratis bajo la licencia de código abierto Mozilla Public License (MPL)

Desarrolladores: Stanford Medical Informatics. The Stanford University School of Medicine- California

Información complementaria: <http://protege.stanford.edu/index.html>

Descripción: Protégé-2000 es una herramienta de software integrado que permite a los desarrolladores crear sistemas basados en el conocimiento. Las aplicaciones desarrolladas con Protégé-2000 se usan para la resolución de problemas y la toma de decisiones en un dominio determinado. El usuario puede realizar las siguientes operaciones:

1. Creación de una ontología.
2. Optimización de datos de entrada de formularios.
3. Inserción de datos.

Es también una plataforma que puede ser extendida con elementos gráficos como tablas, diagramas, componentes de animación, para acceder a otros sistemas basados en el conocimiento embebidos en otras aplicaciones. También puede ser usado como librería por otras aplicaciones para acceder y mostrar bases del conocimiento.

Arquitectura software: Standalone

Extensibilidad: A través del uso de plugins.

Gestión de backups: No.

Acceso a la base del conocimiento vía Web a través de un navegador: Protégé-2000 no proporciona soporte directo para esto; pero se podría hacer a través de servlets. Protégé-2000 puede ser ejecutado a través de un applet.

Almacenamiento de la ontología:

1. Ficheros de texto standard.
2. RDF Schema.
3. Base de datos JDBC: Un proyecto de Protégé 2000 se puede almacenar en cualquier base de datos JDBC compatible.

Almacenamiento en base de datos: Protégé-2000 soporta cualquier base de datos con un driver JDBC 1.0, en la práctica esto implica la mayor parte de las bases de datos relacionales entre las que se han probado: Oracle, MySQL, Microsoft SQL Server y Microsoft Access.

Características del modelado: concepto de herencia múltiple y jerarquías de relación (pero con una única clase por instancia); metaclasses; soporte de especificación de instancias; axiomas con restricciones Prolog, FLogic, OIL y el lenguaje de axioma de Protégé (PAL) mediante plug-ins.

Lenguaje base usado para codificar la ontología: OKBC Open Knowledge Base Connectivity (OKBC) es un API para acceder a la base del conocimiento almacenados en sistemas de representación del conocimiento (KRS).

Plataformas donde puede ejecutarse: Protégé-2000 funciona en cualquier plataforma que soporte la versión 1.3 del JDK. Esto incluye todas las versiones de 32 bits de MS Windows (95/98/NT/2000), las versiones más usuales de Unix incluyendo Linux, Solaris, AIX, y HPUNIX, y sobre el Mac OS X v 10.1. Protégé-2000 no funciona sobre Windows 3.1 ni sobre el sistema operativo Mac "Classic" ya que dichas plataformas no soportan la versión 1.3 del JDK.

Formatos de importación/exportación de la ontología: RDF(S); XML Schema; RDB

schema a través del plug-in Data Genie; (DAML+OIL backend debido a 4Q'02 desde SRI).

Interfaz gráfica: visualización de clases y propiedades globales mediante el plug-in GraphViz y vistas de graficos anidados con el plug-in de edición Jambalaya.

Comprobación automática de consistencia (sintáctica, referencial y/o lógica) de la ontología: existen *plug ins* para añadir y chequear los axiomas que imponen restricciones: PAL, FaCT. Soporte multiusuario, es decir, desarrollo concurrente de la ontología: Varios usuarios pueden leer de la misma base de datos y hacer cambios que no afecten a los demás. Sin embargo, no hay soporte para varios usuarios intentando modificar el mismo elemento del conocimiento base. El soporte multiusuario está en fase de desarrollo.

Soporte para la comparación y fusión de ontologías creadas por aplicaciones independientes: semi-automatizado mediante Anchor-PROMPT.

Soporte léxico (capacidad de referencia léxica de los elementos de la ontología (por ejemplo sinónimos) y el procesado léxico de contenido (búsqueda/filtrado de términos de la ontología): WordNet plug-in; emparejamiento de caracteres especiales (sólo el API).

Información adicional: soporta la metodología CommonKADS.

Contacto: protégé-help@smi.stanford.edu.
 Compatibilidad con otros sistemas: sí.

Proceso para crear una ontología: el proceso de creación de una ontología se realiza llevando a cabo los siguientes pasos:

1. Creación de un proyecto: se inicia Protégé asegurándose que se escoge la opción "New Project" y está seleccionada "Standard Text Files", se guarda seleccionando nombres para el proyecto (archivo con extensión .pprj), archivo que va a contener las clases (extensión .pont) así como las instancias (extensión .pins).
2. Creación de clases y subclases: por defecto aparece seleccionada la pestaña de "Classes". Todas las clases que se vayan a crear van a estar subordinadas de la clase: `THING`. Seleccionando la letra C, se accede al menú de creación de una clase, el nombre por defecto lo proporciona el sistema, pero se cambia inmediatamente. Para crear una subclase se sigue el mismo proceso, pero seleccionando siempre la clase de la que cuelga.
3. Creación de slots (atributos de las clases), se selecciona la clase primero y se despliega el formulario de "Template Slots" tras pulsar sobre la C, aparece el menú para introducir los datos del slot.
4. Creación de instancias: Las instancias son los datos reales del conocimiento base. Es conveniente no introducir instancias hasta no tener completamente creada la estructura de clases y atributos. Se selecciona la pestaña de "Instancias" y tras elegir la clase se escoge la opción C del panel "Direct instances".

Una vez definida la ontología se almacena en la estructura de directorios previamente indicada los ficheros extensión RDF y RDF. El primero contiene los datos, es decir, las instancias mientras que el segundo contiene toda la estructura de la jerarquía de clases de la ontología.

Utilidad / facilidad de uso: Protégé2000 es muy intuitivo y fácil de usar. La creación

de ontologías se hace a través de distintos menús y los datos se introducen mediante formularios.

Existen ya varias ontologías funcionando en Internet: una de ellas es AKTiveSpace1 [13]. Si bien sigue siendo un experimento, muestra la capacidad que pueden tener las ontologías. Se trata de una ontología que busca presentar información acerca de los investigadores de las ciencias de la computación que existen en las diversas regiones de Inglaterra¹².

Algunos ejemplos de interoperabilidad utilizando la tecnología propuesta para la Web semántica son:

1. El proyecto NDLTD3 en el cual un consorcio de universidades, bibliotecas y otras instituciones mantienen un sistema de tesis electrónicas.
2. El Proyecto RODA [14]: dentro de éste se analizan herramientas de anotación de autor o semántica y herramientas de anotación externas que permiten compartir el conocimiento entre diferentes usuarios que trabajan en red.
3. MusicBrainz [18] que se centra en la descripción de música.
4. RSS 1.0 describe canales de televisión.
5. FOAF para describir personas.
6. Dublín Core utilizado para describir documentos.

¹² Es posible acceder a su información por medio de búsquedas geográficas sobre el territorio inglés y a través del área de especialidad de la que se tenga interés. Además de la interfaz que provee, es posible echar una mirada al código fuente y observar las relaciones por medio de las cuales se realizan las inferencias.

7. RdfGeo que describe lugares y mapas.
8. RdfCal para la descripción de eventos y reuniones.
9. ChefMoz para describir restaurantes.

En forma general, se busca que la Web esté formada por una red de nodos tipificados e interconectados mediante clases y relaciones definidas por una ontología compartida por los diferentes autores. Se debe aclarar que la adopción de ontologías comunes es clave para que todos los que participen en la Web semántica, desarrollando o usando recursos, puedan trabajar en forma autónoma con la garantía de que las piezas encajen. Para lograr este objetivo se debe crear programas de computadora que trabajen en la Web y que actúen de una manera autónoma por largos periodos de tiempo, a estos programas se les denomina agentes.

3. Agentes Inteligentes

Al igual que ocurre con la propia definición de la IA, se puede encontrar en la literatura un gran número de definiciones del concepto de agente, sin que ninguna de ellas haya sido plenamente aceptada por la comunidad científica, siendo quizás la más simple la de Russell [16], que considera un agente como una entidad que percibe y actúa sobre un entorno.

En la mayoría de las ocasiones, los agentes no son desarrollados de forma independiente, sino como entidades que constituyen un sistema que se le denomina multi-agente [19]. En este caso los agentes deben o pueden interactuar entre ellos. Las interacciones más habituales, como informar o consultar a otros agentes, permiten a los agentes hablar entre ellos, tener en cuenta lo que realiza cada uno de ellos y razonar acerca del papel jugado por los diferentes agentes que cons-

tituyen el sistema. La comunicación entre agentes se realiza por medio de un lenguaje de comunicación de agentes ACL¹³ (Agent Communication Language).

A continuación, se revisarán algunas de las características que en la literatura se suelen atribuir a los agentes en mayor o menor grado para resolver problemas particulares y que han sido descritos por diferentes autores:

1. Continuidad Temporal: se considera un agente un proceso sin fin, ejecutándose continuamente y desarrollando su función.
2. Autonomía: un agente es completamente autónomo si es capaz de actuar basándose en su experiencia. El agente es capaz de adaptarse aunque el entorno cambie severamente. Por otra parte, una definición menos estricta de autonomía sería cuando el agente percibe el entorno.
3. Sociabilidad: este atributo permite a un agente comunicar con otros agentes o incluso con otras entidades.
4. Racionalidad: el agente siempre realiza "lo correcto" a partir de los datos que percibe del entorno.
5. Reactividad: un agente actúa como resultado de cambios en su entorno. En este caso, un agente percibe el entorno y esos cambios dirigen el comportamiento del agente.
6. Pro-actividad: un agente es pro-activo cuando es capaz de controlar sus propios objetivos a pesar de cambios en el entorno.

¹³ El ACL se codifica en forma de texto con el objetivo de garantizar la comunicación entre plataformas que utilicen diferentes infraestructuras de red.

7. Adaptabilidad: está relacionado con el aprendizaje que un agente es capaz de realizar y si puede cambiar su comportamiento basándose en ese aprendizaje.
8. Movilidad: capacidad de un agente de trasladarse a través de una red telemática.
9. Veracidad: asunción de que un agente no comunica información falsa a propósito.
10. Benevolencia: asunción de que un agente está dispuesto a ayudar a otros agentes, si esto no entra en conflicto con sus propios objetivos.

No existe un consenso sobre el grado de importancia de cada una de estas propiedades para un agente. Sin embargo, se puede afirmar que estas propiedades son las que distinguen a los agentes de simples programas. Además, no podemos olvidar que estos agentes pueden operar en un ambiente Web siempre y cuando a la información disponible en la Web se le agregue un significado semántico en formatos que puedan ser leídos, tanto por dichos agentes como por los humanos. Para este reto es importante contar con un proceso automático para el descubrimiento, composición y orquestación de servicios Web semánticos. En un futuro documento este tema será abordado con toda profundidad.

Cabe indicar que, como siempre ocurre cuando aparece una nueva tecnología, no debemos pensar que todo son agentes y lo que había antes no es válido, más aún, cuando la Web semántica es una evolución de la Web actual y deberán convivir por largo tiempo.

Conclusiones

Para la creación de ontologías es importante tener en cuenta, en la etapa de di-

seño, cuál es el nivel de expresión que se requiere y si vale la pena tomar el riesgo de generar ontologías que no garanticen ser computables. Para esto se debe pensar cuál es el nivel de expresión requerida. Sólo por dar un ejemplo, se debe recordar que la cardinalidad en RDF solamente puede ser 1, mientras que en OWL no hay un límite establecido.

La Web no sólo proporciona acceso a contenidos, sino que también ofrece interacción y servicios, por ejemplo reservar un asiento en un vuelo, comprar un libro, hacer transacciones bancarias, etc. Los servicios Web semánticos son una línea importante de la Web semántica, que propone describir no sólo información, sino también definir ontologías de funcionalidad y procedimientos para describir servicios Web. Estas descripciones procesables por máquinas mediante agentes permitirán automatizar el descubrimiento, la composición y la ejecución de servicios, así como la comunicación entre unos y otros. Además, la capacidad de la Web semántica de dar significado a la información almacenada de tal manera que pueda ser consultada y procesada, provee grandes oportunidades para la gestión de las instituciones, asignando o solicitando en tiempo real, recursos disponibles en la Web.

Sin embargo, aún queda mucho trabajo por hacer. Se necesita crear más y mejor tecnología e infraestructura al igual que desarrollar aplicaciones reales que pongan en práctica los principios de la Web semántica, que pueblen la Web con ontologías y que hagan que la Web semántica adquiera la masa crítica imprescindible para hacerla realidad. Hasta que esto no suceda no se podrán crear agentes inteligentes interoperativos y, mientras tanto, como es lógico,

un usuario podría preguntarse qué ventajas tiene anotar sus páginas Web mediante ontologías si no existen agentes que justifiquen este esfuerzo¹⁴.

Referencias bibliográficas

- [1] Berners-Lee, T., Hendler, J., Lasilla, O. (2001). "The Semantic Web: A New Form of web Content that Meaningful Possibilities". *Scientific American*. s.d., Mayo.
- [2] Codina, L. (2003). "La web semántica: Una visión crítica". *El profesional de la información*. Vol. 12. No. 2. Marzo-abril.
- [3] Cover, R. (1998). "XML and Semantic Transparency". Disponible en: <http://xml.coverpages.org/xmlAndSemantics.html>. (03-15-06).
- [4] Clyde, W., Holsapple, Joshi, K. (2002.) "A Collaborative Approach to Ontology Design". *Communications of the ACM*. Febrero.
- [5] Ding, Y. (2002). "Ontology: the Enabler for the Semantic Web". Disponible en: <http://www.cs.vu.nl/ying/ontology.pdf> (03-25-06).
- [6] Fridman, N., McGuinness, D. (2001). "Ontology Development 101: A guide to Crating your First Ontology". *Stanford Knowledge Systems Laboratory*. Marzo.
- [7] Gruber, T.R. (1993). "Toward Principles for the Design of Ontologies Used for Knowledge Sharing". *Formal Ontology in Conceptual Analysis and Knowledge Representation*. s.d.: Kluwer Academic Publishers.
- [8] Guarino, N. (1998). "Formal Ontology in Information Systems". Amsterdam: IOS Press. Junio.
- [9] Harmelen, F., McGuinness, D. 2002. "Feature Synopsis for Owl Lite on OWL". W3C working draft. Disponible en: www.w3c.org/TR/owl-features/ (04-05-06).
- [10] Hendler, J. (2002). "Agentes and the Semantic Web". *IEEE Intelligent Systems*. Marzo-abril.
- [11] Horridge, M., Knublauch, H., Rector, A., Stevens, R. Wroe, C. (2004). "A Practical Guide to Building OWL Ontologies Using the Protégé-OWL Plugin and CO-ODE Tools". 27 de agosto.
- [12] Lasilla, O., Swick, R. (1999). "Resource Description Framework (RDF): Model an Syntax Specification". Disponible en: www.w3c.org/TR/1999/REC-rdf-syntax-19990222. (03-25-06).
- [13] Proyecto AKTive Space. (s.d.). Disponible en: <http://triplestore.aktors.org/demo/AKTiveSpace/> (04-10-06).
- [14] Proyecto RODA (*Red de Conocimiento Descentralizada a través de Anotaciones*). (2004). Disponible en: <http://roda.ibit.org> (04-10-06).

14 Este fenómeno, denominado por Hendler [10] como "el problema del huevo y la gallina", es similar al que dio en los primeros días de la Web: ¿para qué publicar si había tan pocos usuarios?, o ¿para qué desarrollar navegadores si no había páginas Web?

- [15] Ratnakar, V, Gil, Y. (2002). "A Comparison of (Semantic) Markup Languages". *15th International FLAIRS Conference. Special Track of Semantic Web*. Mayo.
- [16] Russell, S., Norvig, P. (1995). *Artificial Intelligence. A Modern Approach*. USA: Ed. Prentice Hall, Traducido al español en 1996.
- [17] Studer, R., Benjamin, R., Fensel, D. (1998). "Knowledge Engineering Principles and Methods". *Data And Knowledge Engineering*. s.d.
- [18] Swartz, A. (2002). "MusicBrainz: A Semantic Web Service". *IEEE Intelligent Systems*. Enero.
- [19] Wooldridge, M. (1999). "Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence". *The MIT Press*. s.d.

