



UNIVERSIDAD TECNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ESCUELA DE CIENCIAS DE LA COMPUTACIÓN

“SERVICIOS WEB SEMÁNTICOS Y AGENTES INTELIGENTES PARA LA GESTIÓN DEL CONOCIMIENTO”

TESIS DE GRADO PREVIA A LA
OBTENCIÓN DEL TÍTULO DE INGENIERO
EN SISTEMAS INFORMATICOS Y COMPUTACIÓN.

AUTOR:

OSCAR ALDO TAPIA FLORES

DIRECTOR:

Ing. SAMANTA CUEVA CARRIÓN

CODIRECTOR:

Ing. PATRICIO ABAD

Loja- Ecuador

2010



CERTIFICACIÓN

Ing. Samanta Patricia Cueva Carrión.

DIRECTORA DE TESIS

CERTIFICA:

Haber dirigido y supervisado el desarrollo del presente proyecto de tesis previo a la obtención del título de **INGENIERA EN SISTEMAS INFORMÁTICOS Y COMPUTACIÓN**, y una vez que este cumple con todas las exigencias y los requisitos legales establecidos por la Universidad Técnica Particular de Loja, autoriza su presentación para los fines legales pertinentes.

Loja, 26 de Marzo del 2010

Ing. Samanta Cueva Carrión



AUTORÍA

El presente proyecto de tesis con cada una de sus observaciones, análisis, evaluaciones, conclusiones y recomendaciones emitidas, es de absoluta responsabilidad del autor.

Además, es necesario indicar que la información de otros autores empleada en el presente trabajo está debidamente especificada en fuentes de referencia y apartados bibliográficos.

Oscar Tapia Flores



CESIÓN DE DERECHOS

Yo, Oscar Aldo Tapia Flores declaro conocer y aceptar la disposición del Art. 67 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado que se realicen a través, o con el apoyo financiero, académico o institucional (operativo) de la Universidad”.

.....

Oscar Tapia Flores



AGRADECIMIENTO

Dejo testimonio de agradecimiento:

- ✓ A la Universidad Técnica Particular de Loja, por haberme abierto sus puertas y permitirme formar como profesional.
- ✓ Al Ing. Nelson Piedra, Director de la Escuela de Ciencias de la Computación, quien me brindó todo su apoyo y ayuda durante mi vida universitaria.
- ✓ Al Ing. Patricio Abad, por su ayuda para la terminación y culminación de mi proyecto de tesis.
- ✓ De manera especial mi sincero agradecimiento a la Ing. Samanta Cueva, Directora del presente proyecto, por haberme guiado y orientado tan acertadamente en mi práctica profesional.
- ✓ A todas las personas que de distinta manera colaboraron hasta la culminación de éste trabajo.

El Autor



DEDICATORIA

Dedico el fruto de mi esfuerzo; a Dios, por regalarme la vida, la salud y la sabiduría, para hacer realidad este sueño anhelado.

A mí querida esposa y mis bellas hijas, que me han brindado todo su amor y apoyo moral para terminar mi carrera Universitaria.

A mis padres, hermanos y familiares, que con todo su cariño y esfuerzo, me han apoyado para culminar esta etapa de formación.

Oscar



ÍNDICES



ÍNDICE DE CONTENIDOS

CERTIFICACIÓN.....	I
AUTORÍA.....	II
CESIÓN DE DERECHOS.....	III
AGRADECIMIENTO.....	IV
DEDICATORIA.....	V
ÍNDICE.....	VI
Índice de Contenidos.....	VII
Índice de Figuras.....	XI
Índice de Tablas.....	XII
INTRODUCCIÓN.....	1
CAPÍTULO I. INGENIERÍA ONTOLÓGICA.....	3
1.1 Introducción.....	3
1.2 Ontologías.....	4
1.2.1 Definición.....	4
1.2.2 Componentes de una Ontología	6
1.2.2.1 Clases.....	6
1.2.2.2 Conceptos.....	6
1.2.2.3 Atributos.....	6
1.2.2.4 Relaciones.....	6
1.2.2.5 Funciones.....	6
1.2.2.6 Axiomas.....	6
1.2.2.7 Instancias.....	7



1.2.3	Uso de las Ontologías.....	7
1.2.4	Tipos de Ontologías.....	7
1.2.4.1	Clasificación por el ámbito del conocimiento al que se aplican.....	8
1.2.4.2	Clasificación según el tipo de agente al que vayan destinadas.....	8
1.2.4.3	Clasificación por motivación.....	8
1.2.5	Formalismos lógicos y razonamiento con ontologías.....	9
1.2.6	Lenguajes para la representación de ontologías.....	9
1.2.6.1	RDF (Resource Description Framework).....	9
1.2.6.2	RDFS (Resource Description Framework Schema).....	12
1.2.6.3	OWL (Web Ontology Language).....	14
CAPÍTULO II. TECNOLOGÍA DE AGENTES INTELIGENTES.....		17
2.1	Introducción.....	17
2.2	Agente.....	17
2.3	Agentes Inteligentes.....	18
2.4	Agente Semántico.....	20
2.5	Sistemas Multi-agente.....	20
2.6	Negociación en sistemas Multi-agente.....	24
2.7	Metodologías para el diseño de Sistemas Multi-Agentes.....	26
2.7.1	MaSE.....	26
2.7.2	ZEUS.....	27
2.7.3	GAIA.....	27
2.7.4	MAS-CommonKADS.....	28
2.7.5	MOBMAS.....	28
2.7.6	INGENIAS.....	29



2.7.7	Cuadro Resumen.....	31
2.7.8	Análisis y justificación de las metodologías de diseño de SMA.....	33
CAPÍTULO III. SERVICIOS WEB SEMÁNTICOS.....		35
3.1	Introducción.....	35
3.2	La Web Semántica.....	35
3.2.1	Antecedentes.....	35
3.2.2	Fundamentos de la Web Semántica.....	36
3.2.3	Arquitectura de la Web Semántica.....	39
3.2.4	Ventajas y desventajas de la Web Semántica.....	41
3.3	Servicios Web.....	42
3.3.1	Antecedentes.....	42
3.3.2	Conceptualización de los Servicios Web.....	43
3.3.3	Componentes de la arquitectura de los Servicios Web.....	43
3.3.3.1	Web Services Description Language (WSDL).....	46
3.3.3.2	Simple Object Access Protocol (SOAP).....	47
3.3.3.3	Universal Description, Discovery and Integration (UDDI).....	49
3.3.4	Arquitectura orientada a servicios.....	49
3.3.5	Ventajas y Desventajas de los Servicios Web.....	50
3.4	Servicios Web Semánticos.....	51
3.4.1	Antecedentes.....	51
3.4.2	Web Ontology Language for Services (OWL-S).....	52
3.4.3	Web Services Modeling Ontology (WSMO).....	54
3.5	Ventajas y desventajas de los Servicios Web Semánticos.....	57



CAPÍTULO IV. PROPUESTA BASADA EN ONTOLOGÍAS, AGENTES INTELIGENTES Y SERVICIOS WEB SEMÁNTICOS COMO APOYO AL CITTES DE GESTIÓN DEL CONOCIMIENTO DE LA UTPL.....	58
4.1. Introducción.....	58
4.2. Situación actual del CITTES de Gestión de Conocimiento	59
4.3. Requerimientos.....	60
4.4. Servicios del portal web de la UTPL	60
4.5. Propuestas	63
4.5.1. Solución para consulta de fechas y horarios de exámenes para estudiantes de las diferentes modalidades.....	63
4.5.1.1. Descripción del problema.....	63
4.5.1.2. Escenario de pruebas.....	63
4.5.1.3. Propuesta de solución.....	65
4.5.2. Solución para la búsqueda de Video Conferencias.....	68
4.5.2.1. Descripción del problema.....	68
4.5.2.2. Escenario de pruebas.....	68
4.5.2.3. Propuesta de solución.....	69
DISCUSIÓN.....	71
CONCLUSIONES.....	76
RECOMENDACIONES.....	78
ANEXOS.....	79
BIBLIOGRAFÍA.....	82



ÍNDICE DE FIGURAS

Figura 1. Sentencia RDF.....	10
Figura 2. Grafo RDF.....	11
Figura 3. Grafo RDF-S.....	13
Figura 4. Esquema de un Agente Inteligente.....	18
Figura 5. Herramienta de soporte para INGENIAS.....	31
Figura 6. Página HTML (Web actual).....	37
Figura 7. Página web con metadatos.....	38
Figura 8. Evolución de la estructura de la Web tradicional a Web Semántica.....	38
Figura 9. Estructura de la Web Semántica para el W3C.....	39
Figura 10. Evolución de la Web.....	41
Figura 11. Arquitectura de los Servicios Web.....	44
Figura 12. Funcionamiento de los Servicios Web.....	44
Figura 13. Modelo WSDL.....	46
Figura 14. Modelo SOAP.....	48
Figura 15. Estructura de un mensaje SOAP.....	48
Figura 16. Evolución de la Web.....	52
Figura 17. Ontología de Alto Nivel de OWL-S.....	53
Figura 18. Ontología de Alto nivel de WSMO.....	55
Figura 19. Búsqueda con palabra clave “Horarios exámenes modalidad abierta”.....	64
Figura 20. Búsqueda con palabra clave “Horarios abierta”.....	65
Figura 21. Proceso Jerárquico del Servicio Web.....	67
Figura 22. Blog con proceso para búsqueda de fechas y horarios de exámenes.....	68
Figura 23. Búsqueda con palabra clave “video tutorías virtuales”.....	60



ÍNDICE DE TABLAS

Tabla 1. Capacidades de razonamiento con RDF/RDFS.....	15
Tabla 2. Resumen características principales de KQML, KIF y FIPA ACL.....	22
Tabla 3. Resumen características principales de ZEUS, MAS-CommonKADS, GAIA, MaSE, MOBMAS, INGENIAS.....	32
Tabla 4. Cuadro resumen de OWL-S y WSMO.....	56



INTRODUCCIÓN

Uno de los avances más importantes para el intercambio de la información, es la creación del Internet. Con el nacimiento de la WWW (World Wide Web), en el año de 1989, se dió un paso más hacia la consecución de su objetivo que es el de interconectar remotamente varios sistemas software.

En sus orígenes la web fue concebida como un medio para compartir recursos (información). En la actualidad, gracias a los Servicios Web, este ha pasado a ser de un simple repositorio donde se acumula información de cualquier tipo, a una fuente de servicios accesibles desde cualquier lugar del planeta.

El incremento exponencial de la cantidad de datos publicados en la web y el consecuente aumento de la complejidad y el tiempo necesario por parte de usuarios humanos para encontrar la información que precisan en un determinado momento, es lo que dió lugar a lo que se conoce como Web Semántica.

La Web Semántica plantea dotar de estructura y semántica a un sinnúmero de recursos y servicios que provee la web, de forma que se facilite su acceso y explotación no sólo por usuarios sino por agentes software.

En el mundo de los Servicios Web, y de forma similar a lo ocurrido en la web, la cantidad cada vez mayor de servicios disponibles hace inviable en tiempo y eficiencia que sea un usuario humano el que determine el servicio o servicios necesarios para satisfacer una necesidad concreta, surgiendo de este modo los Servicios Web Semánticos.

El problema a resolver es la inexistencia de información procesable automáticamente por máquinas. Y la solución es equivalente, a saber, incluir información expresada formalmente y que permita a los sistemas informáticos entender en cierto modo, el contenido de los documentos. Con este propósito se hace uso de la Ingeniería Ontológica, la misma que encamina hacia la mejora del acceso a la información.

Cuando la información ha sido descrita por medio de ontologías, los sistemas informáticos quedan habilitados para acceder a esta información de forma automática sin la intervención de usuarios humanos, es allí donde aparecen en escena los agentes inteligentes, los cuales pueden definirse como entidades software encargadas de acceder a la información de los servicios y ejecutarlos en lugar de usuarios humanos.



El propósito de este proyecto es analizar y profundizar el estudio de las ontologías, agentes inteligentes y los servicios web semánticos, de forma que interactúen de manera cooperativa aprovechando al máximo las posibilidades que estas tecnologías ofrecen. El componente central y estratégico de la propuesta son las ontologías, por medio de las cuales los agentes se comunican e interactúan entre sí, sin la intervención de agentes humanos. Para cumplir este objetivo, se deben cumplir los siguientes objetivos específicos.

- Analizar el estado del arte de la Ingeniería Ontológica.
- Analizar y describir la tecnología de Agentes Inteligentes.
- Analizar y describir los Servicios Web Semánticos.
- Propuesta basada en Ontologías, Agentes Inteligentes y Servicios Web Semánticos como apoyo al CITTES de Gestión del Conocimiento de la UTPL.

La investigación realizada en el presente trabajo de fin de carrera, servirá como punto de partida para futuras investigaciones de los servicios web semánticos y de las nuevas tecnologías y herramientas relacionadas con la misma.



CAPÍTULO I: INGENIERÍA ONTOLÓGICA

1.1. Introducción

La web semántica nace de la inteligencia artificial y de las tecnologías web, proponiendo nuevas técnicas para representar el conocimiento, facilitando de esta manera su localización e integración de todo tipo de información a través de la web.

Uno de los puntales primordiales para introducir el conocimiento semántico es la noción y estudio de las ontologías como herramienta clave para determinar un entendimiento entre usuario y programas que participan de este conocimiento común.

Las ontologías fueron creadas con la finalidad de representar de manera formal y consensuada especificaciones de conceptos que proveen un conocimiento compartido y común del dominio, como información semántica procesable por las máquinas e interoperable a través de agentes. Estas proveen una comprensión compartida y consensuada del conocimiento de un dominio que puede ser comunicada entre personas y sistemas heterogéneos.

En el presente capítulo, se describe el estado actual de la ingeniería ontológica. Para determinar cuál es el papel principal del conocimiento en la solución de problemas, es necesario considerar cuidadosamente como efectuar el almacenamiento del mismo, es decir representándolo mediante un dominio y relaciones, de modo que los problemas sean más fáciles de resolver.

En primer lugar, se define el concepto de ontología a través de algunas de las definiciones según varios autores contemporáneos, y luego una definición aceptada del término en el ámbito informático, y más concretamente, de la representación del conocimiento.

Seguidamente se enumera los distintos tipos de ontologías, clasificándolos tanto por el conocimiento que contienen, como por la motivación de las mismas.

Luego se describe los principales formalismos lógicos que construyen la base de los lenguajes ontológicos actuales.

Finalmente, se introducen algunos de los lenguajes ontológicos definidos hasta la fecha, poniendo especial énfasis en el lenguaje OWL (Ontology Web Language) "*Lenguaje de*



Marcado para publicar y compartir datos usando ontologías en la WWW¹", que se ha elegido como un estándar para esta investigación.

1.2. Ontologías

1.2.1. Definición

El pilar fundamental dentro de la arquitectura de la web semántica, son las ontologías, las mismas que ayudan a representar el conocimiento. Las ontologías fueron desarrolladas en el área de inteligencia artificial para facilitar el intercambio y la reutilización del conocimiento.

Para determinar una definición apropiada para ontología, se analizan varias definiciones citadas por algunos autores, las mismas que ayudarán a determinar la apropiada para iniciar con el estudio de la ingeniería ontológica.

Primeramente se toma una definición descrita por el diccionario de la Real Academia de la Lengua Española que define a la ontología, como la parte de la metafísica que trata del ser en general, y de sus propiedades transcendentales. Esta definición se utiliza dentro del estudio de la rama de la filosofía. A continuación se presentan las definiciones de ontología en informática, con más aceptación sobre las que se basa este tema de investigación.

La primera definición que se cita, es la siguiente: **[1]**

"Una ontología es una especificación explícita de una conceptualización"

Esta definición considera que una *conceptualización* está compuesta por objetos, conceptos y otras entidades que existen en una determinada área, y las relaciones que se dan entre ellos. De un modo más abstracto una conceptualización se puede definir como una interpretación estructurada de una parte del mundo que usan los seres humanos para pensar y comunicar sobre ella. Por *explícita*, se entiende que los conceptos y las restricciones se definen de forma explícita.

Seguidamente se tiene otra definición que argumenta lo siguiente: **[2]**

"Una ontología es una especificación formal de una conceptualización compartida"

¹ World Wide Web.



En este contexto se refiere a la necesidad de disponer de ontologías comprensibles por las aplicaciones informáticas. Por otro lado, *compartida*, enfatiza la necesidad de consenso en la conceptualización, refiriéndose al tipo de conocimiento contenido en las ontologías, esto es conocimiento consensuado y no privado.

Finalmente se juntan y analizan las dos definiciones antes mencionadas y se establece que una ontología es una especificación formal y explícita de una conceptualización compartida. Por *formal*, quiere decir que la ontología debe ser interpretada por una computadora. *Explícita*, si el tipo de concepto usado y las restricciones para su uso, son explícitamente definidos. *Conceptualización*, se refiere a un modelo abstracto de algún fenómeno en el mundo real proveniente de haber identificado los conceptos relevantes de dicho fenómeno. *Compartida*, una ontología debe capturar el compartimiento consensuado, no el privado de un individuo sino aceptado por un grupo.

Existen muchos autores que dan algunos conceptos sobre lo que representa una ontología, por lo que se puede comprobar que en la actualidad no existe una definición consensuada del término, muy al contrario incluso los autores de las definiciones más reconocidas y aceptadas se replantean las mismas.

Una ontología no es una base de datos ni un programa (porque tiene sus propios formatos internos), no es una conceptualización (porque no es una especificación, es sólo una vista) ni una tabla de contenidos (aunque una taxonomía si es una ontología).

El término ontología en informática, hace referencia a la formulación de un exhaustivo y riguroso "*esquema conceptual*"² dentro de un dominio dado, con la finalidad de facilitar la comunicación y la compartición de la información entre diferentes sistemas. Aunque toma su nombre por analogía, esta es la diferencia con el significado filosófico de la palabra ontología.

Las ontologías son especificaciones formales de cómo representar, objetos, conceptos y otras entidades que se asumen que existen en un área de interés, así como las relaciones que mantienen entre sí. Un uso común tecnológico actual del concepto de ontología, en este sentido, se lo encuentra en la inteligencia artificial y la representación del conocimiento.

² Método de enseñanza con la cual se puede llegar a representar un tema específico.



1.2.2. Componentes de una Ontología

El conocimiento en ontologías, se formaliza principalmente usando siete tipos de componentes: [3]

1.2.2.1. Clases: Una clase, es una declaración o abstracción de un objeto, lo que significa que una clase es la definición de un objeto. Cuando se programa un objeto y se definen sus características y funcionalidades, realmente se programa una clase. Las clases en la ontología se suelen organizar en taxonomías. En todo caso, cabe destacar que ontología y taxonomía son dos elementos diferentes, aunque algunas veces la noción de ontología se diluye en el sentido que las taxonomías se consideran ontologías completas.

1.2.2.2. Conceptos: Los conceptos son ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento etc.

1.2.2.3. Atributos: Los atributos representan la estructura interna de los conceptos. Atendiendo a su origen, los atributos se clasifican en específicos y heredados. Los atributos *específicos* son aquellos que son propios del concepto al que pertenecen, mientras que los *heredados* vienen dados por las relaciones taxonómicas en las que el concepto desempeña el rol del hijo y, por tanto, heredan los atributos del padre.

1.2.2.4. Relaciones: Las relaciones representan un tipo de interacción entre los conceptos del dominio. Se definen formalmente como cualquier subconjunto de un producto de n conjuntos, esto es: $R: C_1 \times C_2 \times \dots \times C_n$. Entre los distintos tipos de relaciones posibles, se encuentran las relaciones taxonómicas ("es_un") y las lógicas o parte nómicas ("parte_de") como relaciones binarias más destacadas.

1.2.2.5. Funciones: 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. Formalmente, definimos las funciones (F) como $F: C_1 \times C_2 \times \dots \times C_{n-1} \rightarrow C_n$. Como ejemplos, se pueden mencionar las funciones "madre de" y "precio de un auto usado".

1.2.2.6. Axiomas: 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.

1.2.2.7. Instancias: Las instancias son las ocurrencias en el mundo real de los conceptos. En una instancia todos los atributos del concepto tienen asignado un valor concreto.

Es indispensable en la actualidad que los diferentes ordenadores en todo el mundo tengan acceso a conjuntos estructurados de información, además de conjuntos de reglas de inferencia que puedan utilizar para realizar el razonamiento automático. Esto es posible mediante la utilización de ontologías, que son una parte fundamental como se mencionó anteriormente, de la Web Semántica, permitiendo a las personas y a las máquinas su comunicación de manera eficiente.

1.2.3. Uso de las Ontologías

Las ontologías se usan para favorecer la comunicación entre personas, organizaciones y aplicaciones, porque proporcionan una comprensión común de un dominio, de modo que se eliminan confusiones conceptuales y terminológicas.

Con las ontologías, los usuarios organizarán la información de manera que los “agentes de software”³, podrán interpretar el significado y, por tanto, podrán buscar e integrar datos mucho mejor que en la actualidad. Gracias al conocimiento almacenado en las ontologías, las aplicaciones podrán extraer automáticamente datos de las páginas web, procesarlos y sacar conclusiones de ellas, así como tomar decisiones y negociar con otros agentes o personas. Por ejemplo, un agente inteligente que busque un vino que satisfaga las preferencias de un usuario, usará las ontologías vinícolas para elegir el vino (color, sabor, olor, embotellado), y empleará las ontologías empresariales para encargarlo a alguna tienda y regatear en el precio (siempre que se pueda). [4]

1.2.4. Tipos de Ontologías

Existen diferentes clasificaciones de tipos de ontologías para el cual se siguen principalmente tres criterios:

³ Agente de Software, es una parte del software que actúa para un usuario u otro programa como agente. El agente tiene la autoridad de decidir cuando una acción es apropiada. (<http://www.alegsa.com.ar/Dic/agente%20de%20software.php>)



- El ámbito del conocimiento al que se aplican
- El tipo de agente al que vayan destinadas.
- La motivación de la ontología.

A continuación, se enumeran los distintos tipos de ontologías atendiendo al conocimiento que contienen: [5]

1.2.4.1. Clasificación por el ámbito del conocimiento al que se aplican

Ontologías generales: Son las de nivel más alto y las que describen conceptos generales.

Ontologías de dominio: Describen el vocabulario de un dominio concreto.

Ontologías específicas: Describen los conceptos para un campo limitado y concreto.

1.2.4.2. Clasificación según el tipo de agente al que vayan destinadas

Ontologías lingüísticas: Vinculados a aspectos lingüísticos a aspectos semánticos y sintácticos.

Ontologías no lingüísticas: Destinados a ser utilizados por agentes inteligentes.

Ontologías mixtas: Son aquellas que combinan las características de las anteriores.

1.2.4.3. Clasificación por motivación

Ontologías para la representación del conocimiento: Permiten especificar las conceptualizaciones que subyacen a los formalismos de representación del conocimiento, por lo que también se denominan *meta-ontologías*.

Ontologías genéricas: Definen conceptos generales y fundacionales del conocimiento como las estructuras parte/todo, la cuantificación, los procesos o los tipos de objetos. Estas ontologías son reutilizables en diferentes dominios.

Ontologías del dominio: Permiten representar el conocimiento especializado pertinente de un dominio o subdominio, como la medicina, las aplicaciones militares, la cardiología, etc.

Ontologías de aplicación: Están ligadas al desarrollo de una aplicación concreta. Tales ontologías cubren los aspectos relacionados con aplicaciones particulares. Típicamente estas ontologías toman conceptos de ontologías del dominio y genéricas, así como métodos específicos para realizar la tarea, por lo que frecuentemente no pueden ser reutilizadas.



1.2.5. Formalismos lógicos y razonamiento con ontologías

Para que los programas software sean capaces de realizar procesos de inferencia sobre la información almacenada en las ontologías, éstas deben de fundamentarse en algún formalismo lógico.

“La utilización de estos formalismos lógicos, además de proporcionar la base necesaria para el uso automatizado de la información, ofrece la posibilidad de realizar procesos de razonamiento que conducen a la adquisición de nuevo conocimiento a partir del ya existente. Así, se puede decir que una lógica define un lenguaje formal para expresar conocimiento acerca de algún dominio y las proposiciones de ser derivadas de este conocimiento, es decir, fija el conjunto de primitivas que pueden emplearse para modelar el mundo, esto es, la *sintaxis*. De la misma forma, una lógica establece la *semántica* formal de cada proposición en el lenguaje correspondiente (valor de verdad de cada sentencia respecto a cada mundo posible) y está equipada con un *cálculo*, es decir, un procedimiento formal (computable) para derivar nuevos hechos (descritos en el lenguaje correspondiente), a partir de un conjunto de proposiciones dado en el lenguaje”. [2]

1.2.6. Lenguajes para la representación de Ontologías

Aunque las ontologías se han utilizado durante muchos años en diversos campos de la investigación, se han convertido en la tecnología estándar para la representación del conocimiento tras el surgimiento de la web semántica: Antes de esto, ya se habían ideado numerosos lenguajes basados en distintos formalismos lógicos para la representación de ontologías.

Para el presente trabajo investigativo, se estudiará a continuación con más detalle, las características de los lenguajes RDF, RDFS y OWL, los mismos que aparecieron por encima de la sintaxis proporcionada por XML, y que resultan un modelo de datos simple que permite identificar recursos y sus relaciones con otros recursos.

1.2.6.1. RDF (Resource Description Framework)

RDF es un lenguaje para representar información sobre los recursos en la WWW. El mismo que es el indicado para representar metadatos sobre recursos web, tales como el título, autor, modificaciones de los datos de un sitio web, copyright y otras licencias de



información sobre documentos web, así como la disponibilidad para algunos recursos compartidos.

RDF está basado en la idea de identificar los recursos en la web usando los URI⁴, y describiendo los recursos en términos de propiedades simples y valores. [6] Por ejemplo, se identificará un recurso a través de la forma como se ha accedido al siguiente sitio, **http://www.powernet.net.ec/servicios/index.html**, este ejemplo identifica una página HTML en un servidor con nombre de DNS⁵ **www.powernet.net.ec** y accedida vía protocolo http.

RDF permite a los agentes software compartir e intercambiar conocimiento, y está basado en un modelo de datos simple, el cual es independiente de la forma como se escriba (sintaxis). [6]

El modelo de datos RDF está basado en *tripletas*, cada una de las cuales está formada por: Sujeto, predicado y objeto.

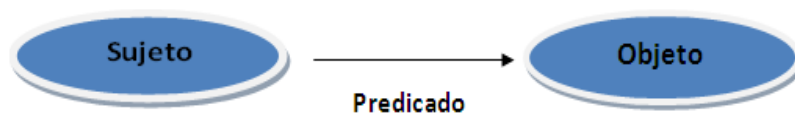


Fig. 1. Sentencia RDF.

Fuente: <http://www.w3c.es/Presentaciones/2005/1018-WebSemanticaREBIUN-MA/>

- ✓ **Sujeto:** Identifica al recurso que la sentencia describe.
- ✓ **Predicado:** Personifica una propiedad del sujeto.
- ✓ **Objeto:** Especifica el valor para la propiedad del sujeto.

En el siguiente ejemplo, se muestra una serie de declaraciones o sentencias: Hay una persona identificada por **http://www.powernet.net.ec/personal/listado.pdf**, cuyo nombre es *Oscar Tapia*, cuya dirección de correo electrónico es **oscar@powernet.net.ec**, y cuyo cargo es "Gerente." que podría representarse como el grafo de la siguiente figura:

⁴ Universal Resource Identifiers

⁵ Domain Name System

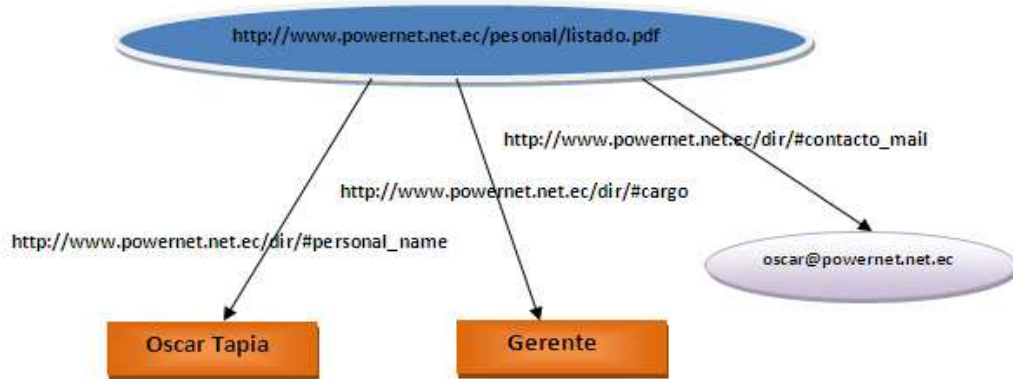


Fig. 2. Grafo RDF.

La figura anterior, ilustra que RDF usa URI para identificar:

- ✓ Individuos, ejemplo, **Oscar Tapia**, identificado por **http://www.powernet.net.ec/personal/listado.pdf**
- ✓ Clases o cosas, ejemplo, **Gerente**, identificado por **http://www.powernet.net.ec/dir/#personal_name**
- ✓ Propiedades de estas cosas, ejemplo, **mail**, identificado por **http://www.powernet.net.ec/dir/#contacto_mail**
- ✓ Valores de estas propiedades, ejemplo, **oscar@powernet.net.ec** como el valor de la propiedad **mailbox**

Existen algunas diferencias entre XML y RDF, por ejemplo XML es un lenguaje que se utiliza para modelar datos, en tanto que RDF es un lenguaje para especificar metadatos: los esquemas de la XML tienen una interpretación sintáctica, restringiendo de esta manera el conjunto de documentos XML que pueden elaborarse, en cambio que RDF posee principalmente una interpretación semántica, el cual sirve para modelar o construir conocimiento.

RDF también posee algunas limitaciones de las cuales hay que tener muy en cuenta al momento de trabajar con el mismo, por ejemplo tienen problemas con ambigüedades en la definición de sus elementos, así como la complejidad de ajustarse a documentos XML.

En resumen, RDF aporta una forma de expresar enunciados simples acerca de recursos usando propiedades y valores.



1.2.6.2. RDFS (Resource Description Framework Schema)

RDF es un lenguaje de propósito general para representar información en la web. La representación RDFS, describe cómo usar RDF para describir vocabularios RDF. En particular, el propósito de RDFS es proveer de un vocabulario XML a través del cual expresar clases y sus relaciones, al tiempo que se definen propiedades y se asocian las mismas con clases. De este modo, RDFS facilita mejoras en los procesos de búsqueda y permite hacer inferencias. Esta capa no sólo ofrece una descripción de los datos, sino también cierta información semántica.

RDF provee una forma para expresar enunciados simples a cerca de los recursos usando propiedades y valores. Sin embargo, la comunidad de usuarios de RDF se percató que se necesitaba indicar algunas veces que lo que estaban describiendo eran tipos o clases específicas de recursos. RDF por sí sólo, no proporciona tal vocabulario, por lo que las clases y propiedades se describen en RDF Schema. [7]

RDF Schema no proporciona un vocabulario sobre aplicaciones orientadas a clases, sino que provee de mecanismos para especificar que tales clases y propiedades son parte de un vocabulario y de cómo se espera su relación. También permite definir a los recursos como instancias de una o más clases. Además permite que las clases puedan ser organizadas en forma jerárquica.

Por lo tanto, RDF Schema extiende a RDF para incluir un amplio vocabulario con un significado adicional.

Las clases definidas por RDFS, pueden representar casi cualquier categoría de cosas, como páginas web, gente, tipos de documentos, conceptos abstractos, etc. Las clases son descritas por recursos **rdfs:Class** y **rdf:Resource**, y por propiedades como **rdf:type** y **rdfs:subClassOf**.

Los recursos que pertenecen a una clase son llamados *instancias*. Así que en RDF Schema, una clase es cualquier recurso que tenga un tipo (**rdf:type**) como propiedad cuyo valor sea un recurso **rdfs:Class**.

Entenderemos por la etiqueta "**rdf:type**" que todo recurso que la ocupe deberá tener las características que distinguen al recurso que se encuentre del lado derecho de la etiqueta. Veamos un ejemplo tomado de GARCIA (2003). [8]



Supongamos que queremos describir la clase Vehículo Motorizado con URIref como **ex:MotorVehicle** (usamos ex como <http://www.example.org/schemas/vehicles>), así que la terna: **ex:MotorVehicle rdf:type rdfs:Class** representa tal clase. La propiedad **rdf:type** indica que tal recurso es una instancia de una clase. Si queremos describir que el recurso **ex:companyCar** es un vehículo motorizado, debemos escribir la siguiente terna:

ex:companyCar rdf:type ex:MotorVehicle

Es decir, la terna anterior es un ejemplo de una instancia de la clase **MotorVehicle** (vehículo motorizado).

Se describen clases adicionales representando varios tipos adicionales de vehículo motorizado. Estas clases las podemos representar de la misma forma descrita antes:

- **ex:Van rdf:type rdfs:Class**
- **ex:Truck rdf:type rdfs:Class**

Para indicar su relación con la clase **MotorVehicle** usamos:

- **ex:Van rdfs:subClassOf ex:MotorVehicle**
- **ex:Truck rdfs:subClassOf ex:MotorVehicle**

Con esto establecemos que Van y Truck son subclases (con **rdfs:subClassOf**) de **MotorVehicle**.

Lo explicado anteriormente se ilustra en la siguiente figura:

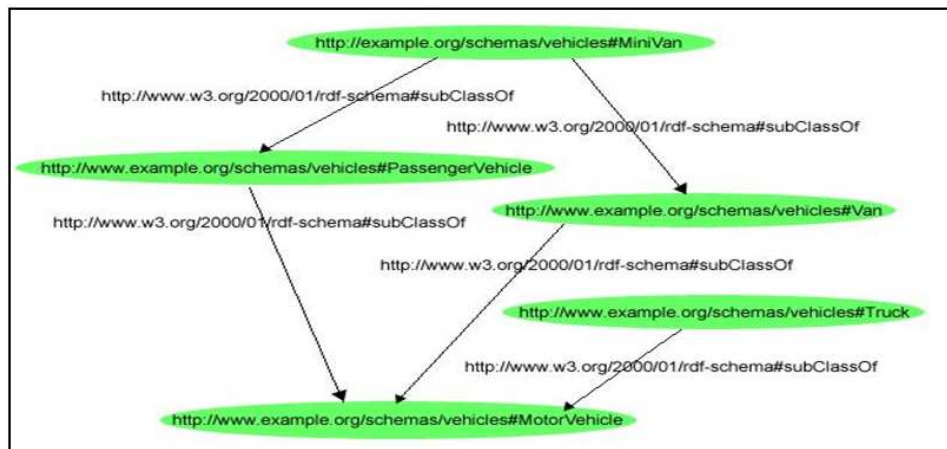


Fig. 3. Grafo RDF-S.

Fuente: http://www.matem.unam.mx/~grecia/semantic_web/rdf.html



Lo que se ha modelado en la figura anterior, da como resultado que un vehículo Van, un vehículo Truck y un vehículo de pasajeros, son vehículos motorizados. Además de que un MiniVan es un Van y por lo tanto es un vehículo motorizado.

1.2.6.3. OWL (Web Ontology Language)

OWL es el acrónimo del Inglés Web Ontology Language o lenguaje de ontologías de web, un lenguaje de marcado para publicar y compartir datos usando ontologías en la WWW, OWL está diseñado para ser usado en aplicaciones que necesitan procesar el contenido de la información, en lugar de únicamente representar información para los humanos. OWL facilita un mejor mecanismo de interpretación del contenido web, que los mecanismos admitidos por XML, RDF y RDFS, proporcionando vocabulario adicional junto con la semántica formal.

La Web Semántica es una visión del futuro de la web, donde la información está dando un significado explícito, permitiendo que las máquinas puedan procesar automáticamente e integrar la información disponible en la web. La Web Semántica, se basará en la capacidad de XML para definir esquemas de etiquetas a medida y en la aproximación flexible de RDF para representar datos. El primer nivel requerido por encima de RDF para la Web Semántica, es un lenguaje de ontologías que pueda describir formalmente el significado de la terminología usada en los documentos Web. [2]

OWL ha sido diseñada para cubrir esta necesidad de un lenguaje de ontologías web. OWL forma parte de un conjunto creciente de recomendaciones del W3C relacionadas con la Web Semántica.

A continuación se enumeran algunas de las posibilidades adicionales que proporciona el vocabulario de OWL:

- Definición de clases mediante restricciones sobre propiedades, valores o cardinalidad, por ejemplo restricción, características.
- Definición de clases mediante operaciones booleanas sobre otras clases: Intersección, unión y complemento.
- Relaciones entre clases, por ejemplo inclusión, disyunción, equivalencia.
- Propiedades de las relaciones, por ejemplo inversa, simétrica, transitiva.
- Cardinalidad, por ejemplo, "únicamente una"



- Igualdad y desigualdad de clases, por ejemplo equivalencia, diferencia.
- Clases enumeradas.

Estas características introducidas en el lenguaje, si bien mejora el poder expresivo del lenguaje, también limitan la capacidad de los razonadores para inferir nuevo conocimiento a partir de unas sentencias dadas. En particular, no se pueden garantizar ni la completitud computacional, esto es, que el razonador encuentre todas las conclusiones válidas, ni la divisibilidad computacional, esto es, que se obtenga la respuesta para cualquier entrada en un periodo de tiempo finito.

OWL es muy similar a RDFS, esto es, en servir de vocabulario XML para definir clases, propiedades y relaciones. En comparación con RDFS, OWL permite expresar relaciones mucho más ricas y, por tanto, disponer de capacidades de inferencias mejoradas. Las capacidades de razonamiento que soporta OWL y que extienden lo indicado para RDF/RDFS se presentan en la siguiente tabla:

Transitividad Owl:TransitiveProperty	(rdf:type progenitor owl:TransitiveProperty) (progenitor Blas Francisco) (progenitor Francisco Maria) → (progenitor Blas Maria)
Propiedades inversas Owl:inverseOf	(owl:inversorOf padreDe tienePadre) (padreDe francisco Maria) → (tienePadre Maria Francisco)
Herencia de disyunción Owl:disjointWith	(owl:disjointWith Planta Animal) (rdfs:subClassOf Mamifero Animal) → (owl:disjointWith Planta Mamifero)
Disyunción y Complementario Owl:complementOf	(owl:complementOf Animal NoAnimal) (rdfs:subClassOf Mamifero Animal) → (owl:disjointWith Mamifero Animal)
Pertenencia a clases	Si x pertenece a A y A es una subclase de B, entonces x pertenece a B.
Equivalencia de clases	Si A es equivalente a B y B es equivalente a C, entonces A es equivalente a C.
Consistencia	Si x pertenece a A y a B pero A y B son disjuntas, Entonces se ha producido un error.
Clasificación	Si para pertenecer a una clase A es necesario cumplir ciertas propiedades y x cumple dichas propiedades, entonces x pertenece a A.

Tabla 1. Capacidades de Razonamiento con OWL.

Elaboración: A Semantic Web Primer.

Fuente: García, 2007



OWL, es un lenguaje poseedor de un mecanismo para elaborar o desarrollar vocabularios específicos que asociado con recursos, proporciona un lenguaje para definir ontologías bien estructuradas que puedan ser utilizadas en diferentes sistemas computacionales, las mismas que representan un sector de conocimiento en particular que van a ser usadas por los usuarios y aplicaciones que necesitan compartir cualquier tipo de información en particular.



CAPÍTULO II: TECNOLOGÍA DE AGENTES INTELIGENTES

2.1. Introducción

Como sucede en toda nueva rama del conocimiento, existe bastante confusión en cuanto al vocabulario de las diferentes tecnologías semánticas, en este capítulo se explica el concepto de Agente, Agente Inteligente y Agente Semántico, términos que aparecen casi siempre en el estudio de la Web Semántica, además de destacar sus semejanzas y diferencias para lo cual se utilizan algunas definiciones y ejemplos.

Además en el presente capítulo, se analizan los Sistemas Multi-Agente: Su concepto, desarrollo y evolución; también se revisan las metodologías para el diseño de los Sistemas Multi-Agente y las plataformas que permiten el desarrollo de este tipo de sistemas de acuerdo con los estándares establecidos.

2.2. Agente

Según la Real Academia Española, un agente se define como, *"Una persona o máquina que actúa con poder o voluntad de una persona o de otra persona o máquina"*. [9]

Agente es un sistema capaz de percibir cambios que se dan a su alrededor, además representa la información sobre el estado actual de sus objetivos y actúa basándose en experiencias pasadas.

Existen diferentes tipos de agentes, como los físicos, que pueden ser: Personas, animales, robots, autómatas; y los agentes de software tales como: La Web, Intranets, que son considerados como entornos virtuales y que son la base para nuestra investigación.

Al igual que ocurre con la definición de Inteligencia Artificial "*Rama de la ciencia informática dedicada al desarrollo de agentes racionales no vivos*" [10], se pueden encontrar en la literatura gran cantidad de definiciones sobre el concepto de *agente*, sin que ninguna de ellas haya sido plenamente aceptada por la comunidad científica, siendo quizás la más simple la de RUSELL, que considera un agente, como una entidad que percibe y actúa sobre un entorno, como se muestra en la siguiente figura:

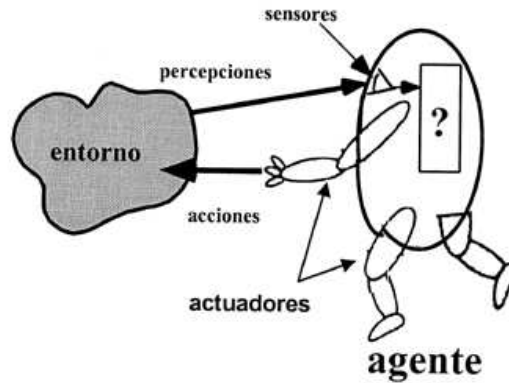


Fig. 4. Esquema de un Agente Inteligente.

Fuente: <http://www.ati.es/novatica/2000/145/vjulia-145.pdf>

Esta definición determina a los agentes por los atributos que poseen, que luego definen su comportamiento para resolver un problema determinado en el entorno en el cual interactúan.

Como ya se analizó para el caso de las ontologías en el capítulo 1 apartado 1.2, no existe una definición universalmente aceptada del término “*agente*”, ni acuerdo en cuanto a las propiedades que este tipo de entidades debe de presentar.

Una de las definiciones más utilizadas de agente es la siguiente: [11]

"Un agente es cualquier cosa, capaz de percibir su medio ambiente con la ayuda de sensores y actuar en este medio utilizando actuadores".

Esta definición se centra en el componente físico del término y en su interacción con el mundo que lo rodea.

Una de las principales características de un agente, es que ayuda a los usuarios para escuchar el problema, buscar en diversos medios y finalmente dar una respuesta al mismo. Este agente puede residir tanto en la red como en el ordenador del usuario, y además, puede conocer los fallos para tratar de proponer soluciones a los mismos antes de que sucedan.

2.3. Agentes Inteligentes

Para que un agente llegue a ser denominado como *inteligente*, debe satisfacer un conjunto mínimo de propiedades que se destacan a continuación: [12]



Persistencia, Autonomía, Reactividad, Pro actividad y Habilidad Social.

Persistencia: Un agente se ejecuta continuamente, no cuando el usuario lo decide, sino que actúa por sí mismo y decide cuando dejar de ejecutar alguna acción. Los agentes siempre permanecen activos: Se ejecutan en un bucle infinito y observan su entorno, actualizan su estado y determinan que acciones tomar.

Autonomía: Un agente inteligente puede tomar decisiones sin la intervención humana o de otros agentes. Es decir, que para cumplir un agente sus objetivos, este debe controlar sus propias acciones y decisiones.

Reactividad: Los agentes inteligentes son capaces de percibir su entorno y responder en un período de tiempo adecuado, a cambios que ocurren en el mismo, para satisfacer los objetivos para el cual fueron diseñados.

Pro actividad: Los agentes inteligentes no actúan dirigidos solamente por los sucesos que se producen en su entorno, sino que también toman iniciativas para lograr sus objetivos, es decir, los agentes reconocen y utilizan las oportunidades que se les presentan. Las iniciativas pueden proceder de experiencias pasadas, pues los agentes aprenden del pasado.

Habilidad Social: Los agentes inteligentes son capaces de interactuar con otros agentes, tanto los de tipo software como posiblemente con los de tipo humano, para satisfacer sus objetivos de diseño.

Existen trece propiedades que, además de albergar las ya mencionadas (autonomía, reactividad, etc.), incluyen entre otras, la continuidad temporal (que un agente funciona sin cesar), la capacidad de aprendizaje, el razonamiento (como mecanismo de toma de decisiones) y la cooperación (comunicación e interacción entre agentes para conseguir un objetivo común).

Analizando las diferentes definiciones y teorías de varios autores, acerca de lo que es un agente inteligente, se determinó que la siguiente definición, es la apropiada para el desarrollo del presente trabajo:

“Agente Inteligente es una entidad software situada en un lugar concreto, apto de expresar un comportamiento autónomo, reactivo y proactivo sobre el mismo, preparado



para interactuar con otros agentes, con el fin de satisfacer determinados objetivos establecidos por un ente ya sea software o humano, representado por el agente.”

Se concluye en este apartado diciendo que un agente inteligente es capaz de entender lo que se le pide, además comprende el contenido de los sitios web, validando si lo encontrado corresponde a lo pedido y deduciendo la nueva información encontrada sobre la ya obtenida.

2.4. Agente Semántico

Los agentes semánticos, son agentes que usan tecnologías de la web semántica como RDF/RDFS, OWL, ontologías, las mismas que ya fueron estudiadas en el primer capítulo. Existen agentes semánticos que operan sobre dominios muy específicos, como intranets de empresas, y algunos agentes semánticos que actúan como buscadores de información o como asistentes personales.

Para mostrar las posibilidades de los agentes semánticos, se considera el ejemplo que se encuentra detallado en el Anexo 1. Este ejemplo de agente semántico, da la pauta para el estudio profundo de la web semántica el cual como se ha visto, es de primordial importancia para satisfacer las necesidades de búsqueda en la web, este estudio se profundizará más adelante con mayor énfasis en el capítulo 3.

2.5. Sistemas Multi-agente [13]

Para comenzar estudiando los Sistemas Multi-Agente, se debe tomar en cuenta que en la mayoría de los casos, los agentes, se encuentran en entornos que contienen otros agentes, constituyendo de este modo un Sistema Multi-Agente llamado por sus siglas como SMA, esto es, un sistema formado por un grupo de agentes que pueden interactuar. Una definición de SMA más elaborada dice lo siguiente:

“Un SMA, es un sistema informático formado por un grupo de agentes que interactúan entre sí utilizando protocolos y lenguajes de comunicación de alto nivel, para resolver problemas que están más allá de las capacidades individuales o del conocimiento de cada uno de los componentes que resuelven problemas.” [14]

Como se dijo anteriormente, un grupo de agentes individuales forma un SMA, el mismo que hace surgir la necesidad de disponer de un mecanismo para coordinar dicho grupo de agentes, así mismo de un lenguaje para permitir la comunicación entre estos agentes.



Estos mecanismos de coordinación, se distinguen dependiendo de los objetivos que tienen estos agentes, por ejemplo, si tienen objetivos comunes, estos cooperan entre sí, y los otros agentes que son los conflictivos, los mismos que precisan de mecanismos de negociación.

De forma similar a la clasificación anterior, se distingue entre sistemas distribuidos de resolución de problemas, que son los constituidos por agentes diseñados explícitamente para conseguir de forma cooperativa un objetivo dado, y sistemas abiertos, donde confluyen agentes elaborados por distintos desarrolladores y que poseen posiblemente objetivos totalmente diferentes.

La comunicación entre los agentes, es una parte muy importante y primordial, para lo cual se han elaborado diversos lenguajes. Estos lenguajes fueron diseñados bajo la influencia de la teoría de los actos del habla. Esta teoría está basada, fundamentalmente, en la existencia de los denominados actos del habla (por ejemplo, solicitar, informar, prometer). Por lo tanto, las conversaciones se basan en los actos del habla y facilitan el potencial necesario para la coordinación, la cooperación y la negociación.

Coordinación.- Se puede definir, como la manera en que los agentes se comportan individual y socialmente para que se satisfagan los objetivos personales y los globales.

Cooperación.- Puede verse como la actuación coordinada entre agentes de tal manera que nos colaboran, interesada o desinteresadamente, en la resolución de tareas de otros.

Negociación.- Es un proceso mediante el cual un grupo de agentes llegan a un acuerdo mutuamente aceptable sobre algún asunto. Se indican más detalles acerca de la negociación y sus posibilidades en la sección 2.6.

Existen gran cantidad de lenguajes de comunicación entre los agentes, entre los más destacados tenemos: KQML⁶, KIF⁷ y FIPA-ACL⁸.

KQML se ocupa de definir un formato común para los mensajes sin meterse en el contenido de los mismos. Cada mensaje en KQML está formado por un acto del habla,

⁶ KQML Knowledge Query and Manipulation Language

⁷ KIF Knowledge Interchange Format

⁸ FIPA Agent Communication Language



que define el tipo de mensaje y un conjunto de parámetros (p.ej. contenido, remitente, receptor). KIF es un lenguaje que permite representar conocimiento sobre un determinado dominio de discurso. KIF es usado, principalmente, para formar la parte del contenido de mensajes KQML. FIPA-ACL es similar a KQML, en tanto que define el formato de los mensajes sin preocuparse ni implicar el uso de un lenguaje específico para el contenido de los mensajes. En el siguiente cuadro, constan las características principales de estos lenguajes.

KQML	KIF	FIPA-ACL
<ul style="list-style-type: none"> • Lenguaje de consulta. • Protocolo de Intercambio de Información. • Sintaxis similar a la de LISP. • Los argumentos (palabras claves) deben estar ordenados. • La semántica del mensaje se define por los campos: <i>Content, language y ontology</i>. • Envuelve un mensaje en una estructura entendible por agentes. • Para entender un mensaje un agente receptor debe entender este lenguaje. 	<ul style="list-style-type: none"> • Lenguaje para el intercambio de conocimiento. • Es legible para máquinas. • Basado en cálculo de predicado de primer orden. • Se define a través de Ontologías (conceptos, atributos, relaciones). • Utiliza un lenguaje natural. • Sintaxis similar a LISP y Schema. • Permite representar meta-conocimiento. • Words, son los objetos fundamentales del lenguaje (variables, constantes, operadores) • Utiliza operadores lógicos como: <i>Negación, disyunción, unión</i>. 	<ul style="list-style-type: none"> • Similar a KQML • Es fiable y confiable. • Es capaz de enviar un mensaje, codificarlo para su transmisión como una secuencia de bytes. • No se preocupa por el lenguaje del contenido de los mensajes.

Tabla 2. Resumen características principales de KQML, KIF y FIPA ACL.

Elaboración: Oscar A. Tapia F.

Las principales ventajas en la utilización de un SMA son:

- ✓ *Fiabilidad.*- Cuando un elemento del sistema deja de funcionar, no implica que también es resto lo haga. Por lo que son más robustos y tolerantes a fallos.
- ✓ *Modularidad.*- Permite una programación más estructurada, por lo que reduce la complejidad de la programación con unidades más pequeñas.
- ✓ *Adaptabilidad.*- Los agentes tienen la habilidad de reconfigurarse de forma que se adapten a fallos y cambios.



- ✓ *Concurrencia.*-Los agentes son capaces de razonar y realizar tareas en paralelo.
- ✓ *Flexibilidad.*- Los SMA pueden añadir y eliminar agentes dinámicamente.

Seguidamente se identifica seis retos principales que se debe tomar en cuenta a la hora de diseñar e implementar SMA: **[2]**

- Formular, describir y descomponer problemas y asignar tareas a agentes individuales.
- Hacer que agentes heterogéneos, sean capaces de comunicarse e interaccionar.
- Hacer que los agentes individuales, actúen de forma coherente para realizar acciones y tomar decisiones.
- Hacer que los agentes individuales, razonen sobre las acciones, planes, y el conocimiento de otros agentes.
- Reconocer y reconciliar intenciones y objetivos conflictivos entre agentes coordinados.
- Alcanzar una visión más ingenieril de los sistemas de inteligencia artificial distribuida, empleando metodologías para el desarrollo de SMA.

Los Sistemas multi-agente además de poseer numerosas ventajas como se explicó anteriormente, también posee inconvenientes. Uno de los cuales es la inexistencia de aplicaciones reales desarrolladas que exploten el potencial tanto de la tecnología de agentes, como de los SMA.

La comunicación entre agentes situados en servidores remotos, es una de las mayores limitantes, esto se da cuando se quiere enviar un mensaje, el mismo que tiene que traspasar diferentes fronteras para poder llegar a su destino final, esta limitante es debido a que las plataformas multi-agente hacen uso de protocolos que no están estandarizados o son incompatibles, en la gran mayoría de los casos los mensajes enviados, no llegan a su destino, debido a que son rechazados por firewalls instalados en los sistemas de otras compañías. Los problemas asociados al uso de protocolos no estándar en tecnologías como CORBA, RMI o DCOM, dio origen a los Servicios Web detallado más adelante, que



hace uso de protocolos estándar como HTTP, que evita los problemas antes mencionados.

Se concluye diciendo que los SMA, aprovechan las bondades que ofrecen las tecnologías de los servicios web semánticos para llevar a cabo las tareas de descubrimiento, selección y composición de servicios web, interactuando con varios agentes de acuerdo a la organización social, cooperación, coordinación, negociación y control que cada uno de ellos aporta para una comunicación eficiente, y de esta manera dar solución a los requerimientos de los diferentes usuarios.

2.6. Negociación en sistemas Multi-Agente

Para que los mecanismos de cooperación y coordinación sean exitosos en un sistema multi-agente, se define, el concepto de negociación y se indican cuáles son los componentes necesarios para llevar a cabo, de forma satisfactoria, un proceso de negociación en un SMA.

Existen varios procesos de negociación que tienen como resultado la modificación o confirmación de las creencias de cada agente involucrado, en lo relacionado con los demás agentes y con el mundo en el que se desenvuelven.

Existen diferentes tipos de negociación, como por ejemplo las humanas, que ayuda a alcanzar decisiones conjuntas, siendo investigado por mucho tiempo y dando como resultado procesos de negociación automatizados. Esta negociación humana es relativamente lenta debido a esto no se obtienen los resultados requeridos. Existe otro mecanismo de negociación que es el automático, que es de primordial para nuestro estudio ya que es llevado por sistemas computarizados como son los agentes software.

Es por eso que se ha tomado una definición de negociación que dice lo siguiente:

“Negociación es el medio por el cual los agentes se comunican y se comprometen a alcanzar acuerdos mutuamente beneficiosos”. [15]

Queda claro que en los procesos de negociación, se debe de buscar, un acuerdo que satisfaga a todas las partes en la negociación. Es posible llegar a una situación en la que una negociación no llegue a un final satisfactorio, por el hecho de que alguna de las partes no esté de acuerdo con ninguna de las propuestas realizadas. Sin embargo, en la



mayoría de los casos, ésta es la situación menos deseada y, por tanto, siempre se busca llegar a buen término el proceso de negociación por medio de concesiones de las partes.

A continuación se identifican dos componentes básicos que debe existir en todo sistema de negociación automática:

Protocolo de negociación.- Este protocolo es el que define en su totalidad las reglas por las que se maneja el proceso de negociación automático, esto es, las circunstancias bajo las cuales la interacción entre los agentes tiene.

Para que una negociación sea exitosa es necesario un protocolo que facilite y en lo posible garantice la convergencia de ideas a una solución común. Un protocolo establece un conjunto de pasos que debe seguir un proceso de negociación, así como las posibles respuestas de un agente, a las acciones de otro agente.

A continuación se describe un ejemplo simple de protocolo de negociación entre dos agentes **A** y **B**:

- 1) **A** lanza una propuesta.
- 2) **B** evalúa esta propuesta y determina si es buena para él.
- 3) Si la propuesta de **A** satisface a **B**, el proceso termina. De lo contrario, **B** debe lanzar una contrapropuesta con sus argumentos.
- 4) Si **A** considera que los argumentos de **B** son buenos, el proceso termina. Si no se logran poner de acuerdo, se necesita la intervención de un tercer agente que tenga en cuenta los puntos de vista de ambos y tome una decisión

Estrategia de negociación.- Es la especificación de la serie de acciones, ofertas y respuestas que el agente proyecta hacer durante la negociación.

Dada la limitación de que no existe una solución para todo tipo de dominio, sino que el desarrollador o diseñador, dependiendo de las condiciones en el proceso de negociación, aplicada a cada momento, tiene que determinar el mecanismo de negociación para cada interacción, para que mediante el uso de estrategias asociadas con el manejo de ontologías, codifiquen protocolos de negociación, de forma que sean los agentes, quienes decidan que protocolo y estrategia utilizar al momento de llevar a cabo una interacción.



2.7. Metodologías para el diseño de Sistemas Multi-Agentes

A continuación se presenta un resumen de lo que proponen las metodologías actuales tales como: INGENIAS, ZEUS, MAS-CommonKADS, GAIA, MaSE, MOBMAS. Ante el impedimento de analizar y estudiar todas y cada una de las existentes, se ha optado por elegir un conjunto significativo cercano a las prácticas de ingeniería del software orientado a agentes. Esta selección atiende a la presencia de un proceso de desarrollo, el combinar diferentes vistas para describir el sistema, e incorporar elementos asociados al área de los agentes. Como resultado, se tienen seis metodologías relevantes. A continuación se estudian las aportaciones de cada una de las metodologías mencionadas anteriormente y se escogerá la mejor que servirá como punto de partida para nuestro estudio dentro de la web semántica.

2.7.1. MaSE [2]

Conocida por sus siglas en inglés (Multi-agent Systems Software Engineering), tiene como punto de partida el paradigma orientado a objetos, asumiendo que un agente es una especialización de un objeto capaz de coordinarse con otros agentes a través de conversaciones y de actuar por iniciativa propia, para conseguir metas individuales y del sistema. AgentTool es la herramienta que acompaña a MaSE, que habilita la ejecución de la mayoría de los pasos de los que consta el proceso de desarrollo, especificado en MaSE. Este proceso está dividido en dos fases, la fase de análisis y la fase de diseño.

En la fase de *análisis*, los pasos a realizar consisten en: *La captura de los objetivos, captura de los casos de uso y el refinamiento de los roles.*

La fase de diseño está compuesta por cuatro pasos: *Crear clases de agentes, construir conversaciones, ensamblar clases de agentes y realizar el diseño del sistema.* AgentTool, además, permite generar código automáticamente a partir de la especificación del sistema. La generación de código es totalmente independiente del lenguaje de programación que se utiliza, ya que se realiza recorriendo las estructuras de datos generadas en la especificación, y generando como salida un texto.



2.7.2. ZEUS

ZEUS es una metodología muy similar a AgentTool y MaSE. Desde sus inicios y apariciones, se ha convertido en referencia de cómo debería ser una herramienta para el desarrollo de sistemas Multi-agente. ZEUS propone un desarrollo en cuatro etapas:

- Análisis del dominio.
- Diseño de los agentes.
- Realización de los agentes.
- Soporte en tiempo de ejecución.

La metodología ZEUS viene acompañada de una herramienta que abarca y soporta las etapas de realización de los agentes, y de soporte en tiempo de ejecución.

ZEUS es conceptualmente superior a MaSE, debido a que incorpora más elementos relativos a la tecnología de agentes. Sin embargo, metodológicamente MaSE es más poderosa, ya que se acerca más a las prácticas de la ingeniería convencional. Hasta la aparición de MaSE, ZEUS era considerado, como una herramienta referente, ahora ya se encuentra más información determinando que MaSE tiene mayor referencia y prioridad. Conceptualmente ZEUS es mayor a MaSE, está por decir que ZEUS está más orientado, a la ampliación de tecnología de agentes y MaSE mas orientada a aplicaciones en ingeniería convencional.

2.7.3. GAIA

GAIA es una metodología para el diseño de sistemas basados en agentes cuyo objetivo es obtener un sistema que maximice alguna medida de calidad global.

En GAIA se entiende que el objetivo del análisis es conseguir comprender el sistema y su estructura sin referenciar ningún aspecto de implementación. Esto se consigue a través de la idea de *organización*. Una organización en GAIA es una colección de roles, los cuales mantienen ciertas relaciones con otros y toman parte en patrones institucionalizados de interacción con otros roles. Los roles agrupan cuatro aspectos: responsabilidades del agente, los recursos que se le permite utilizar, las tareas asociadas e interacciones.

GAIA propone trabajar inicialmente con un análisis a alto nivel. En este análisis se usan dos modelos, *el modelo de roles* para identificar los roles clave en el sistema junto con



sus propiedades definitorias y el *modelo de interacciones* que define las interacciones mediante una referencia a un modelo institucionalizado de intercambio de mensajes. Tras esta etapa, se entraría en lo que GAIA considera diseño a alto nivel. El objetivo de este diseño es generar tres modelos: el *modelo de agentes* que define los tipos de agente que existen, cuántas instancias de cada tipo y qué papeles juega cada agente, el *modelo de servicios* que identifica los *servicios* asociados a cada rol, y un *modelo de conocidos*, que define los enlaces de comunicaciones que existen entre los agentes.

A partir de aquí, los autores de GAIA proponen aplicar técnicas clásicas de diseño orientado a objetos. Sin embargo, GAIA declara que queda fuera de su ámbito. Esta metodología sólo busca especificar cómo una sociedad de agentes colabora para alcanzar los objetivos del sistema, y qué se requiere de cada uno para lograr esto último.

2.7.4. MAS-CommonKADs

Esta tecnología extiende la metodología para el desarrollo de sistemas expertos CommonKADS, aplicando ideas de metodologías orientadas a objetos para su aplicación a la producción de SMA. En particular, se extiende los modelos de CommonKADS para tener en cuenta la posibilidad de que dos o más componentes del sistema interactúen. Son siete los modelos disponibles: Agente, tareas, experiencia, coordinación, comunicación, organización y diseño. MAS-CommonKADS plantea el desarrollo de SMA integrado con un ciclo de vida de software en espiral dirigido por riesgos.

La metodología CommonKADS gira alrededor del modelo de experiencia, y está pensada para desarrollar sistemas expertos que interactúen con el usuario. Tiene dos agentes básicos que son el usuario y el sistema.

2.7.5. MOBMAS

MOBMAS es una metodología para el diseño de SMA basado en ontologías. Estas ontologías han sido ampliamente reconocidas por sus beneficios significativos para la interoperabilidad, la reutilización, actividades de desarrollo del SMA tales como el análisis de sistemas y modelos de conocimiento agente, y el funcionamiento del SMA como el agente de la comunicación y el razonamiento. Reconociendo estos beneficios, hace posible que la metodología MOBMAS, sirva para identificar y aplicar las diversas formas en que las ontologías pueden ser utilizadas en el proceso de desarrollo del SMA. Explotando esta metodología es factible el desarrollo de productos SMA con diferentes



dosificaciones. Estos son los puntos fuertes incluyendo los beneficios de la ontología antes mencionados, y los beneficios adicionales descubiertos por MOBMAS, por ejemplo, apoyo para la verificación, validación, extensión, mantenimiento y fiabilidad. En comparación con numerosas metodologías orientadas a agentes, MOBMAS es el primero que de manera explícita y extensamente investiga las diversas ventajas potenciales de las ontologías en el desarrollo del SMA. Otra contribución importante de MOBMAS al campo de la AOSE⁹ es su capacidad para hacer frente a todas las preocupaciones clave del desarrollo del SMA en un marco metodológico.

MOBMAS proporciona soporte para una lista completa de los requisitos metodológicos, que son importantes para el análisis y diseño del agente. MOBMAS apoya los requisitos metodológicos identificados al combinar las fortalezas de las metodologías existentes, es decir, mediante la reutilización y la mejora de las distintas técnicas fuertes y definiciones de modelo de las metodologías existentes en su caso, y proponiendo nuevas técnicas y definiciones de modelo cuando sea necesario.

Para concluir, el proceso de desarrollo MOBMAS consta de tres actividades de investigación secuencial. La primera actividad es identificar y validar una lista de requisitos metodológicos para una Ingeniería de Software Orientada a Agentes, metodología mencionada anteriormente. La segunda actividad de investigación que desarrollada MOBMAS, es la especificación de un proceso de desarrollo, un conjunto de técnicas y un conjunto de definiciones de modelo para el apoyo de otras metodologías. La última actividad de la investigación evaluada y refinada por MOBMAS mediante la recopilación de opiniones de expertos sobre esta metodología, utilizando la misma en la aplicación y conducción de un análisis de las características de la metodología. [16]

2.7.6. INGENIAS

Para empezar con esta metodología, se debe recalcar que INGENIAS es parte de MESSAGE¹⁰, la misma que es una pieza muy importante en el estudio de los agentes software; como muestra de esto, INGENIAS aplica las ideas de MESSAGE al desarrollo

⁹ Agent Oriented Software Engineering

¹⁰ MESSAGE (Methodology for Engineering Systems of Software Agents). Proyecto que desarrolla una metodología del software para agentes inteligentes, donde se mezclan ideas del agente y programas informáticos, buscando lo mejor de ambos y creando la metodología que se necesita.



de sistemas en tiempo real. La razón por la cual se hace referencia primero a MESSAGE antes que INGENIAS, es que INGENIAS es una evolución de las ideas de MESSAGE.

INGENIAS es una metodología de desarrollo de SMA que proporciona un conjunto de métodos y herramientas para desarrollar dicho tipo de sistemas. El método de desarrollo de SMA propuesto en INGENIAS, concibe el SMA como la representación computacional de un conjunto de modelos. Cada uno de estos modelos muestra una visión parcial del SMA: Los agentes que lo componen, las interacciones que existen entre ellos, cómo se organizan para proporcionar la funcionalidad del sistema, qué información es relevante en el dominio y cómo es el entorno en el que se ubica el sistema a desarrollar. Para presentar estas perspectivas, INGENIAS define un conjunto de meta-modelos, esto es, una descripción de alto nivel de qué elementos tiene un modelo: Entidades, relaciones y restricciones, que comprenden toda la información que se debe tener en cuenta al especificar un SMA.

Para la construcción de los meta-modelos, que son la base primordial de INGENIAS, se integran resultados de investigación en forma de entidades y relaciones entre entidades. La instanciación de éstos meta-modelos, emiten unos diagramas, que son modelos similares a los que se usa en UML, pero existe una diferencia ya que estos diagramas se han creado para definir exclusivamente un sistema multi-agente.

Una de las características principales de INGENIAS, es que tiene como respaldo una aplicación denominada INGENIAS IDE, la cual es utilizada para presentar un modelado visual, la misma que almacena la especificación del sistema utilizando XML.

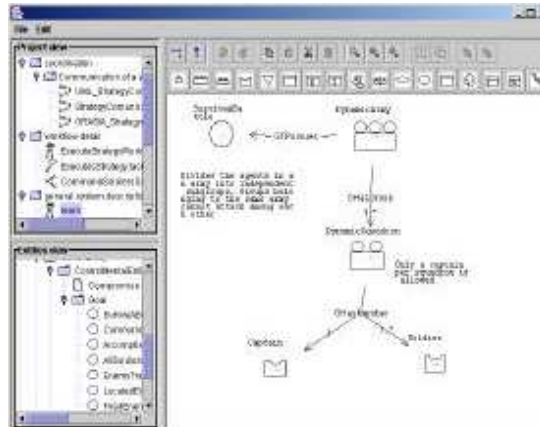


Fig. 5. Herramienta de soporte para INGENIAS.

Fuente: <http://ingenias.sourceforge.net>

Esta herramienta genera un código relleno plantillas de código con información de la especificación, y genera la documentación de una manera similar a la del código que parte de plantillas de documentación que se completan utilizando la información de los modelos.

Existe otra herramienta de modelo visual, el IDK (INGENIAS Development Kit), la misma que hace posible la construcción de todos los modelos que se precisan, de acuerdo con la metodología e incluye un generador automático de código capaz de devolver código JADE ¹¹.

2.7.7. Cuadro resumen

En el presente cuadro resumen, se muestran las características principales de cada una de las metodologías que se ha analizado, para escoger la mejor según el requerimiento que se tenga.

¹¹ JADE (Java Agent Development Framework). Plataforma multi-agente que genera código para un entorno a partir de diagramas elaborados. <http://jade.tilab.com>



Tabla 3. Resumen características principales de ZEUS, MAS-CommonKADS , GAIA, MaSE, MOBMAS, INGENIAS.

Elaboración: Oscar A. Tapia F.

ZEUS	MAS-CommonKADS	GAIA	MaSE	MOBMAS	INGENIAS
<p>Herramienta y metodología de desarrollo de SMA.</p> <p>Basado en una arquitectura concreta.</p> <p>Utiliza entornos visuales.</p> <p>Utiliza Ontologías, Agentes y Relaciones.</p> <p>Propone 4 etapas:</p> <ul style="list-style-type: none"> • Análisis de dominio. • Diseño de Agentes. • Realización de Agentes. • Soporte en tiempo de ejecución <p>Conceptualmente superior a MaSE.</p> <p>Orientado a la aplicación de tecnología de agentes.</p> <p>Soportado por una herramienta.</p>	<p>Sistema construido bajo tecnología Orientado a Objetos.</p> <p>Define el modelo computacional del agente.</p> <p>Utiliza Sistemas expertos.</p> <p>Gira alrededor de un modelo de experiencia.</p> <p>Propone 7 modelos:</p> <ul style="list-style-type: none"> • Agente, Tareas. • Experiencia. • Coordinación. • Comunicación. • Organización. • Diseño. <p>No posee herramientas de soporte.</p> <p>Utiliza metodologías de diseño de protocolo.</p> <p>Utiliza Ontologías.</p>	<p>Metodologías para diseñar sistemas basados en Agentes.</p> <p>SMA como conjunto de entidades que interactúan.</p> <p>Independiente de la arquitectura.</p> <p>Utiliza un diseño de Alto Nivel.</p> <p>Genera Modelos, Agentes, Servicios.</p> <p>Consta de 2 etapas:</p> <ul style="list-style-type: none"> • Análisis de Alto Nivel. • Diseño de Alto Nivel. <p>Utiliza la Organización.</p>	<p>Sistema construido bajo tecnología Orientado a Objetos.</p> <p>Basado en una arquitectura concreta.</p> <p>Orientado a las prácticas de Ingeniería convencional.</p> <p>Basado en RUP.</p> <p>Propone 3 etapas:</p> <ul style="list-style-type: none"> • Capturar los Objetivos. • Capturar los casos de Uso. • Refinar Roles. <p>Utiliza Diagramas de Clases de Agentes.</p> <p>Se aprende rápido, pero tiene problemas con las dependencias entre diagramas.</p> <p>Genera código automáticamente.</p> <p>Soportado por la herramienta agentTool.</p>	<p>Posee un soporte extensivo del uso de Ontologías en el diseño de SMA.</p> <p>Consta de 5 etapas.</p> <ul style="list-style-type: none"> • Análisis. • Diseño de organización de SMA. • Diseño Interno de los Agentes. • Diseño de interacción de los Agentes. • Diseño de la Arquitectura. <p>Proporciona soporte para una lista completa de requisitos metodológicos para el análisis y diseño del agente</p> <p>Contribuye al AOSE.</p> <p>Carece de una evaluación completa de sus capacidades a partir de su aplicación en entornos reales y complejos.</p>	<p>Define un conjunto de Meta-modelos.</p> <p>Basado en RUP.</p> <p>Los meta-modelos se presentan de forma separada:</p> <p>Integra resultados de investigación de forma de entidades y sus relaciones.</p> <p>El proceso de generación de código es más flexible que en ZEUS y MaSE.</p> <p>Posee una herramienta IDK(Ingenias Development Kit)</p> <p>Utiliza Ontologías, protocolos y agentes.</p>



2.7.8. Análisis y justificación de las metodologías de diseño de SMA

A continuación se analizan las metodologías mencionadas anteriormente para determinar cuál es la mejor opción para el diseño de SMA.

Se empieza haciendo una comparación entre ZEUS y MaSE, ante la similitud de enfoques, se hace una breve comparativa. Conceptualmente, ZEUS es superior a MaSE, si bien ZEUS está más orientado a la aplicación de tecnología de agentes, MaSE se orienta más a las prácticas de ingeniería convencional.

MAS-CommonKADS es la metodología más cercana a las líneas principales de INGENIAS. Incorpora la idea de proceso de ingeniería en el sentido de Pressman, y describe con bastante detalle cómo se debe definir el sistema teniendo en cuenta las dependencias entre los modelos.

Con respecto a la metodología GAIA, la principal crítica que se puede hacer es que se queda a un nivel de abstracción demasiado alto. Según los autores, con ello se consigue desacoplar GAIA, de las distintas soluciones de implementación de agentes. Sin embargo, la utilidad de este desacoplamiento queda por demostrar. Dado el nivel de abstracción en que se queda, es de esperar que el esfuerzo a invertir para pasar de una especificación GAIA hasta su implementación, sea alto.

La elección de una metodología no es tan simple, debido a que cada metodología, por el bagaje de sus creadores, se especializa en áreas concretas. Por ejemplo, si se está acostumbrado a trabajar con sistemas basados en conocimiento con la metodología CommonKADS, lo lógico es que se elija MASCommonKADS. Por otro lado si la experiencia del usuario está en el área de los objetos, la recomendación sería de MaSE. Pero, si se está interesado en un enfoque más orientado a agentes, puede seleccionar ZEUS, INGENIAS, BDI o GAIA. Y si lo que se quiere es tener un soporte de herramientas, la lista de metodologías válidas se reduce considerablemente: ZEUS, MaSE, INGENIAS. En los casos en que se requiera un proceso de desarrollo robusto, detallado y ensayado en desarrollos reales, la recomendación sería MASCommonKADS o INGENIAS.

La elección de una metodología u otra, como se dijo anteriormente, no es una tarea sencilla y, en general, depende del dominio de aplicación y de las características del sistema final que se pretenda construir. En muchas ocasiones, es apropiado disponer de



una herramienta que dé soporte a la metodología y permita la elaboración de los diagramas y modelos propuestos por la metodología de una manera más eficiente.

El estudio de la aplicación de las metodologías, depende de la formación que tenga su usuario final. Esto quiere decir que son interpretadas en función de los intereses de quien las aplica. De momento, no hay soluciones que fuercen un uso concreto.



CAPÍTULO III: SERVICIOS WEB SEMÁNTICOS

3.1. Introducción

Los Servicios Web Semánticos son la evolución de la popular tecnología de los Servicios Web, a los que se les aplica los conceptos introducidos por la Web Semántica. En el presente capítulo se describen primeramente, las partes constituyentes de la tecnología de los Servicios Web Semánticos, esto es, la Web Semántica y los Servicios de Web. A continuación, se presenta una descripción detallada de las principales características de esta tecnología y se enumeran las distintas aproximaciones propuestas al W3C para alcanzar un estándar. Finalmente, se discuten algunas de las carencias y problemas de esta tecnología que, en cierto modo, justifica la necesidad de integrar los servicios web semánticos con tecnología de agentes.

3.2. La Web Semántica

3.2.1. Antecedentes

La WWW, fue concebida por Tim Berners-Lee en el año de 1989, a partir de un proyecto elaborado por el CERN¹², la misma que cambió radicalmente el modo en que la gente recopila información y accede a la misma. Hoy en día, la WWW, se ha convertido en un gigantesco repositorio de información en continuo crecimiento, y al que se puede acceder desde cualquier punto con el único requisito de disponer de un navegador Web.

En la actualidad, millones de páginas se encuentran indexadas por los distintos buscadores existentes, de forma que la búsqueda de un dato concreto, se ha convertido en uno de los mayores cuellos de botella de la WWW. El ejemplo típico que se utiliza para presentar este problema y demostrar las carencias de la Web actual es el de la búsqueda de información concerniente a la isla de java. Si en un buscador cualquiera, como por ejemplo el Google, escribimos la consulta “java” nos aparecen cientos de miles de páginas que nada tienen que ver con lo que el usuario está buscando realmente, la isla de java, sino que la mayoría se referirán al lenguaje de programación JavaTM. Este ejemplo, sin embargo, no demuestra las verdaderas carencias de la WWW ni de las tecnologías detrás de ésta, en particular el lenguaje HTML¹³, ya que el usuario solo tendría que refinar

¹² Consejo Europeo para la Investigación Nuclear

¹³ HTML (Hyper Text MarkupLenguaje). Lenguaje de marcas hipertextuales.

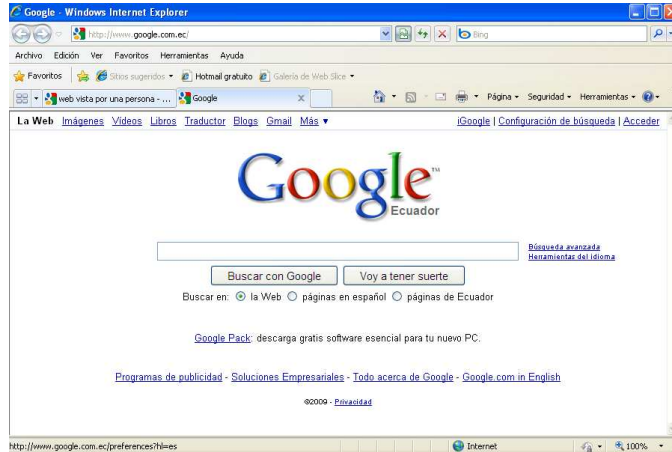


la búsqueda con la consulta “la isla de java” y, ahora sí, encontraría todos aquellos documentos y páginas Web en los que puede estar interesado. El problema surge cuando se trata de obtener información sobre una relación concreta entre dos conceptos. Por ejemplo, si se realiza la consulta “artículos sobre Tim Berners-Lee” muchos de los resultados que se obtendrán serán enlaces a artículos escritos por Tim Berners-Lee. De forma resumida, la principal limitación de la Web actual, es el hecho de que las tecnologías que la conforman no son capaces de capturar, o representar finalmente, la semántica del contenido presentado. La Web Semántica surgió con el propósito de restringir estos efectos nocivos causados por el crecimiento del número de páginas publicadas en la WWW.

3.2.2. Fundamentos de la Web Semántica

El término Web Semántica fue presentado al dominio público tras la publicación del artículo “The Semantic Web” aparecido en “Scientific American” en mayo del 2001, y del que fueron coautores Tim Berners-Lee, James Hendler y Ora Lassila. [17] En este artículo se establecen diversos escenarios imaginarios en los que agentes software son capaces de realizar numerosas tareas, accediendo al contenido de diferentes páginas de la WWW. Los autores señalan que, para que este escenario sea factible, debería cambiar la manera de representar el contenido dentro de la web, que hasta ahora ha sido diseñado para que los seres humanos puedan leerlo, para incluir una semántica bien definida que permitiese a componentes software acceder al mismo.

Tim Berners-Lee, en un texto citado por Francisco García, afirma lo siguiente: *La Web Semántica incrementará estructura al contenido lleno de significado de las páginas Web, creando un entorno donde distintos agentes software, transitando de página en página en la web, puedan llevar a cabo fácilmente tareas sofisticadas para los usuarios.* Por lo tanto, la Web Semántica se basa, en añadir metadatos semánticos a los datos publicados en la WWW, de forma que las entidades software puedan procesar el contenido de la información de forma similar, como lo hacen los humanos. Al hablar de metadatos, se refiere a ofrecer datos, de forma que el significado de los datos es capturado. En la siguiente figura, se muestra como es una página Web en la actualidad, vista desde dos perspectivas, tanto de un usuario como de un computador.



Sitio Web visto por un usuario o persona.

```
<!doctype html><html onmousemove="google&google.fade&google.fade()"><head><meta http-equiv="content-type"
content="text/html; charset=UTF-8"><title>Google/</title><script>window.google=
{kEi:"TOY7S7mnKSSVeMnbkM4I",kEXFI:"17259,17311,17315,23129",kCSI:
(e:"17259,17311,17315,23129",ei:"TOY7S7mnKSSVeMnbkM4I",kHL:"es",time:function(){return(new Date).getTime
()},log:function(b,d,c){var a=new Image,e=google,g=e.lc,f=e.li;a.onerror=(a.onload=(a.onabort=function(){
delete g[f]}));g[f]=a;c||"/gen_204?atyp=1&cm="b"&cad="ad"&sz="google.time();a.src=c;e.li=f+1;lc:
[],li:0};
window.google.sn="webhp";window.google.timers=(load:(t:(start:(new Date).getTime())));try
(window.google.pt=window.external&window.external.pageT);catch(u){}window.google.jert_kill=1;
var _gjlwLocation=function _gjlw() {var b=_gjlw.href.indexOf("#");if(b>=0){var a=_gjlw.href.substring
(b);if(/(^[^#]*)/ .test(a)&&a.indexOf("#")===a.length/0)}&&$.test(a)};_gjlw.replace
("/search?ea.replace(/(^[^#]*)/g,")-"&cad=";return 0;function _gjp() {
(window._gjlw.hash&&window._gjlw())&&setTimeout(_gjp,800)};
window._gjp && _gjp()</script><style>td{line-height:.8em}.gac_m td{line-height:1.7px;form{margin-
bottom:20px;tbody,td,a,p,.h{font-family:arial,sans-serif}.h{color:#36c;font-size:20px}.q{color:#00c}.ts td
{padding:0}.ts{border-collapse:collapse}em{font-weight:bold;font-style:normal}.lst{font:17px arial,sans-
serif;margin-bottom:.2em;vertical-align:bottom}input{font-family:inherit}.lsb,.gac_sb{font-
size:15px;height:1.85em!important;margin:.2em;overflow:visible;padding:0 20px;}
#fctr,#ghead,#pmoctr,#sbl,#tba,#tbe,.fade{background:#fff}#gbar{float:left;height:22px}.gbb,.gbd{border-
top:1px solid #c9d7f1;font-size:1px}.gbb{height:0;position:absolute;top:24px;width:100%}#gbs,.gbbm
{background:#fff;left:0;position:absolute;text-align:left;visibility:hidden;z-index:1000}.gbbm{border:1px
solid;border-color:#c9d7f1 #36c #36c #a2b872;z-index:1001}#guser{padding-bottom:7px !important;text-
align:right}#gbar,#guser{font-size:13px;padding-top:1px !important}.gb1,.gb3,.gb3L,.gb3F{zoom:1;margin-
right:.5em}.gb2,.gb2L,.gb2F{display:block;padding:.2em .5em}.gb1,.a.gb2,.a.gb3,.a.gb4{color:#00c !
important}.gb2,.gb2L,.gb2F,.gb3,.gb3F{text-decoration:none}.a.gb2:hover{background:#36c;color:#fff !
important}</style><noscript><script>#fctr,#ghead,#pmoctr,#sbl,#tba,#tbe,.fade,.gbb{opacity:1;filter:alpha
(opacity=100)}</script></script><script>google.y={};google.x=function(e,g){google.y[e.id]=[e,g];return
false};if(!window.google)window.google={};window.google.crx={};window.google.cri=0;window.clk=function
(d,e,f,j,k,l,m){if(document.images){var a=encodeURIComponent(!escape,b=new
Image,g=window.google.cri++;window.google.crx[g]=b;b.onerror=(b.onload=(b.onabort=function(){delete
```

Sitio Web visto por un computador.

Fig. 6. Página HTML (Web actual).

Fuente: www.google.com.ec

En un ejemplo particular, investigando en la web el horario de consulta de un fisioterapeuta, una entidad software puede realizar esta búsqueda basándose en palabras clave, que identifiquen palabras como “fisioterapéutico”, “hora de consulta”, etc., pero tendrá problemas para identificar al personal y distinguir entre los fisioterapeutas y la secretaria. La Web Semántica, propone modificar la forma en que se presentan los contenidos en la Web de modo de que no solo se indique información para formatear el contenido, sino que también se incluya información que describa este contenido. Siguiendo el ejemplo anterior, el código quedaría como se presenta en la figura 7. Esta



presentación, mediante metadatos hace mucho más sencillo para las máquinas el procesamiento de la información.

```

<empresa>
  <nombre>Centro Fisioterapeutico Ingenio</nombre>
  <tratamientoOfrecido>Fisioterapia</tratamientoOfrecido>
  <personal>
    <terapeuta>Felicia Fernandez</terapeuta>
    <terapeuta>Juan Sanchez</terapeuta>
    <secretaria>Carmen Rodriguez</secretaria>
  </personal>
  <horario>
  ...
</horario>
</empresa>
    
```

Fig. 7. Página Web con metadatos.
Fuente: W3C Oficina Española. [18]

En la figura 8, se puede comprobar la evolución que ha tenido lugar en cuanto a la estructuración de los contenidos en la Web tradicional y en la Web Semántica. Si la conexión entre recursos en la Web tradicional se produce a través de enlaces únicamente entendibles por usuarios humanos, la nueva estructura de la Web Semántica se basa en relaciones semánticas que son capaces de interpretar tanto humanos como entidades software.

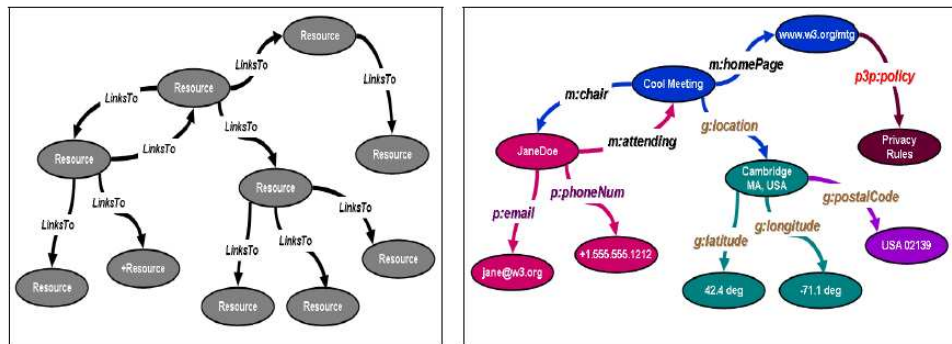


Fig. 8. Evolución de la estructura de la Web tradicional a Web Semántica.

Fuente: <http://www.w3c.es/Presentaciones/2008/0220-semanticBPMS-MA/img/OldWebLinks.png>.

La Web Semántica es un concepto donde todos los datos o documentos sobre la World Wide Web, tienen algún grado de significado relacionado con ellos. Un beneficio general, por ejemplo, es que se podía encontrar lo que se esté buscando, más rápidamente y con



una mejor relación. La idea fue propuesta hace unos años por Tim Berners - Lee y sus colegas (Jim Hendler y Ora Lassila) que se dieron cuenta de que el World Wide Web era limitado en su eficacia porque todos los documentos en la web, todavía necesitaban la interpretación humana si se desea hacer algo útil con ellos. [19]

La Web Semántica se basa en la utilización de ontologías como mecanismo para la representación del conocimiento. Las ontologías se utilizan para definir un vocabulario que describa las relaciones entre diferentes términos, permitiendo a los sistemas computarizados y a los humanos, interpretar su contenido flexiblemente y sin ambigüedades. En conclusión, una ontología contiene una lista finita de términos y relaciones, los mismos que denotan conceptos como por ejemplo: Clases de objetos. En la sección 1.2, se encuentra más detallada la definición de ontología.

3.2.3. Arquitectura de la Web Semántica

El estudio y evolución de la Web Semántica, ha hecho que la misma sea dada en forma de capas, sobre las cuales se ha producido la aparición de otras capas sobre las que ya se encuentran o existen. El W3C¹⁴, muestra la estructura de la Web Semántica mediante la siguiente figura:

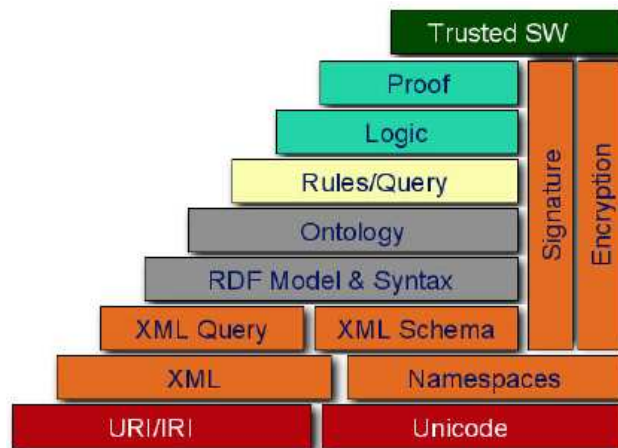


Fig. 9. Estructura de la Web Semántica para el W3C.

Fuente: <http://www.daedalus.es/tecnologias-web/web-semantica/>. [20]

¹⁴ W3C. Es un consorcio internacional donde organizaciones miembro, personal a tiempo completo, y el público en general, trabajan conjuntamente para desarrollar estándares Web.



A continuación se presentan las características de las principales capas de la estructura expuesta por el W3C.

- XML, lenguaje de marcado de propósito general, aporta la sintaxis superficial para los documentos estructurados, pero sin dotarles de ninguna restricción sobre el significado.
- XML Schema, es un lenguaje para definir la estructura de los documentos XML.
- RDF, modelo de datos para los recursos y las relaciones que se pueden establecer entre ellas, dota de una semántica básica para este modelo de datos, que se representa mediante XML.
- RDF Schema, vocabulario para describir las propiedades y las clases de los recursos RDF, dotadas de una semántica para establecer jerarquías de generalización entre dichas ontologías.
- OWL, finalmente añade más vocabulario para describir propiedades y clases.
- SPARQL, es un lenguaje para formular consultas sobre datos en formato RDF.

En la siguiente figura se muestra como ha ido evolucionando la web en estos últimos años, y lo que nos vendrá para el 2010, desde el punto de vista de las tecnologías que se le han incorporado para incrementar su potencialidad.



Fig. 10. Evolución de la Web.

Fuente: <http://noesis.usal.es> [21]

3.2.4. Ventajas y desventajas de la Web Semántica

Una de las principales ventajas de la Web Semántica, es el desarrollo de nuevas aplicaciones con esquemas de datos comunes y el incremento de las transacciones entre empresas por comercio electrónico.

Otra ventaja es la eficiencia en la búsqueda de la información, ya que ahorra muchísimo tiempo en el procesamiento de la misma, ya que la mayor parte de las tareas a realizarse, son hechas por componentes software que son automatizados y que no requieren en muchos casos de la intervención humana.

No se depende de un solo servicio para obtener cualquier tipo de información, sino que esta puede estar distribuida en varios sitios, y por ende puede ser juntada con un tercero.

La Web Semántica dota a la web tradicional, de un significado y por lo tanto de más semántica, por lo que se pueden obtener soluciones a muchos problemas habituales en la búsqueda de la información gracias a la utilización de una infraestructura común, mediante la cual se hace más fácil compartir transferir y procesar información.



Una de las principales desventajas es la reestructuración de la información que ya esta subida al internet, ya que deben acoplarse al nuevo formato que propone la nueva Web, para que los computadores puedan interpretar de manera eficiente y exacta la información.

Otro de los inconvenientes de la web semántica, son los efectos que puede producir en relación a la privacidad y censura. Actualmente las técnicas de análisis de textos utilizadas por ejemplo por los gobiernos para controlar los contenidos de la Web son vulnerables a simples modificaciones de palabras o al uso de imágenes en lugar de texto, consiguiendo de esta manera limitar la forma de censurar dicha información.

3.3. Servicios Web

Un servicio web es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma.¹⁵

Para la presente investigación se propone la siguiente definición de Servicios Web, dada por el W3C. **[22]**

“Un Servicio Web, es un software designado para resistir interacciones interoperables de máquina a máquina sobre una red de ordenadores. Un Servicio Web tiene una interfaz descrita en un formato procesable por el ordenador. Otros sistemas pueden interaccionar con los Servicios Web de la manera prescrita en su descripción usando mensajes SOAP, generalmente transmitidos mediante el uso de protocolo HTTP con una serialización en XML en conjunción con otros estándares relacionados con la Web.”

3.3.1. Antecedentes [2]

Los Servicios Web, no son una novedad en la actualidad, ya que en los años 70, es donde empieza su origen con el surgimiento del Internet y la aplicación de la computación.

Por esos mismos años la computación tiene sus orígenes con dos tecnologías, como son los minicomputadores y las redes entre los mismos, como son Ethernet e Internet. Estas tecnologías trabajan conjuntamente para llevar a cabo algunas tareas, las entidades software en un sistema distribuido, no operan en el mismo entorno de procesamiento, de

¹⁵ <http://es.wikipedia.org/wiki/Servicioweb>



forma que deben de comunicarse a través de protocolos hardware o software sobre una red de comunicaciones.

Los sistemas de objetos distribuidos, son sistemas en los que la semántica de la inicialización de un objeto y la de la invocación de los métodos, se expone a sistemas remotos por medio de mecanismos estándares o propietarios que permiten intermediar en las peticiones que salen de los límites del sistema.

3.3.2. Conceptualización de Servicios Web

Los Servicios Web son componentes que ejecutan procesos y funciones de negocio significativas, con una interfaz claramente definida, y accesible a través de la web, basado en intercambio de documentos electrónicos en formato XML y que pueden ser combinados entre sí. Es una unidad lógica de un negocio en particular que proporciona datos y servicios para otras aplicaciones.

Los Servicios Web utilizan protocolos estándares para lograr una perfecta comunicación e integración multi-plataforma.

3.3.3. Componentes de la arquitectura de los Servicios Web

Uno de los principales problemas que se tiene con la tecnología de los servicios web, es la limitación en cuanto a la interoperabilidad. Así mientras algunas tecnologías se restringían a algunas plataformas concretas, el resto no eran flexibles en cuanto al lenguaje de programación que están diseñadas.

En general el uso de algunas soluciones, suponía un problema en la mayoría de los casos, debido a que se imposibilitaba la interacción entre los elementos elaborados de acuerdo con las distintas soluciones y, lo que es peor, su uso global estaba fuertemente restringido por los firewalls o cortafuegos que bloquean los puertos por donde se entablaba la comunicación entre estos componentes. Es por eso que se planteó encontrar un conjunto de estándares que fueran aceptados mundialmente, para construir una tecnología que fuese más acorde a la visión de Internet. XML y HTTP son la base sobre la que se ha desarrollado la tecnología de los Servicios Web.

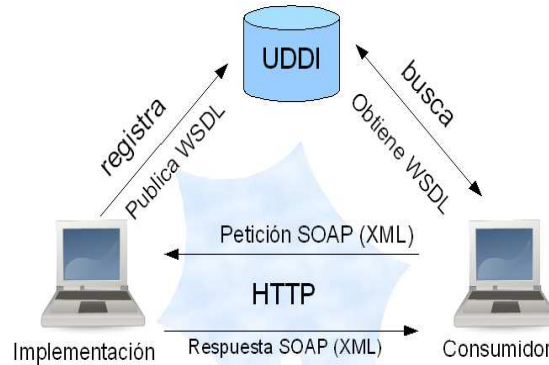


Fig. 11. Arquitectura de los Servicios Web.

Fuente: <http://www.di.uniovi.es/%7Elabra/cursos/Web20/>

En la siguiente figura se muestra cómo interactúan entre si un conjunto de Servicios Web, donde se explica cuál es el papel tanto del usuario como el servicio Web que se solicite.

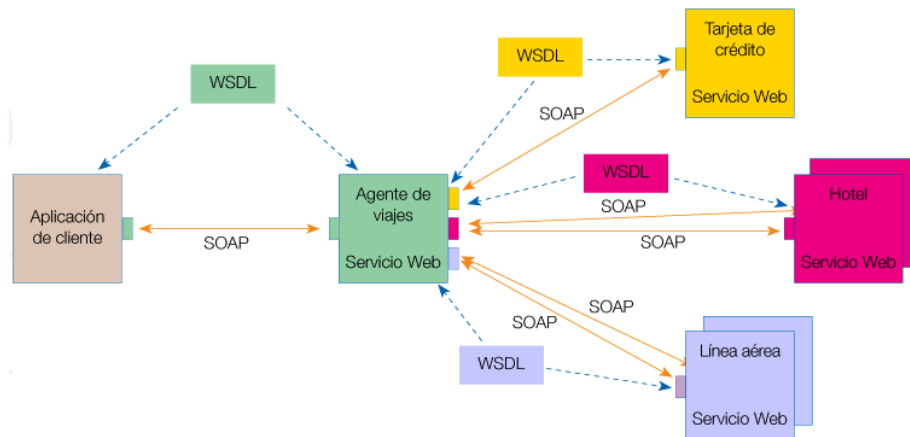


Fig. 12. Funcionamiento de los Servicios Web.

Fuente: <http://www.w3c.es>

Según el gráfico de la figura 12, un usuario o cliente mediante una aplicación, solicita toda la información posible de una tour a Israel, esto lo hace mediante una petición a una agencia de viajes que brinda este tipo de servicio, a través del internet, la misma que ofrece al usuario la información requerida. La agencia de viajes para dar al cliente la información que necesita, solicita la información a otros recursos o Servicios Web, en relación con hoteles, vuelos, horarios, etc. La agencia obtiene la información de estos recursos, lo que la convierte a su vez en clientes de esos otros Servicios Web que le van



a proporcionar todo tipo de información sobre hoteles y líneas aéreas. Y para terminar, el usuario realizará el pago a través de la agencia de viajes que servirá de intermediario entre el usuario y el Servicio Web que tramitará el pago del tour.

Para realizar todo este proceso, intervienen algunas tecnologías que hacen posible la circulación de la información requerida por el usuario, por un lado SOAP¹⁶, que es un protocolo basado en XML, que permite la conexión entre varios dispositivos y que tienen la capacidad de transmitir información compleja. Por otro lado WSML¹⁷, que hace posible que un servicio y un cliente, establezcan un acuerdo en lo que se refiere a los detalles de transporte de los mensajes y su contenido, donde representa como una manera de contrato entre los mismos. Más adelante se estudia con más detalle estos estándares.

A continuación se describe, los cinco pasos principales que constituye el modus operandi de los servicios web, como se representa en la figura 11. **[2]** Es necesario primeramente, que la entidad que desea ofrecer un servicio, añada una descripción del mismo en un registro UDDI, esto es, registre el servicio para su posterior utilización. Esto debe ocurrir para todos los servicios, de todas las organizaciones que se deseen que sean accesibles en todo el mundo. El siguiente paso, una vez que la descripción de todos los servicios se encuentra almacenada en los registros UDDI accesibles por internet, consiste en que los clientes consumidores de servicios busquen aquellos que les sean necesarios para conseguir un objetivo. El proceso de búsqueda lo realiza el propio registro UDDI, que recibe como entrada la funcionalidad esperada del servicio y debe devolver todos aquellos servicios que cumplan con esa funcionalidad. El registro de servicios devuelve al cliente, como respuesta de una petición, una lista con los ficheros de descripción de los servicios, esto es en WSDL cuyas capacidades cubran las necesidades del usuario o cliente.

Una vez que el cliente consumidor del servicio, dispone de los ficheros WSDL de todos aquellos servicios a los que el cliente puede acceder para alcanzar un objetivo, aquel debe determinar cual servicio, de entre los de la lista, es el más apropiado de acuerdo con las preferencias del cliente. Cuando el cliente ha seleccionado el servicio más apropiado a ejecutar, realiza la petición al proveedor del servicio a través de un mensaje SOAP. El servicio es entonces ejecutado utilizando los parámetros de entrada indicados en el

¹⁶ Protocolo Simple de Acceso a Objetos

¹⁷ Lenguaje de descripción de Servicios Web



mensaje SOAP, y el resultado de la ejecución se devuelve al cliente en forma de otro mensaje SOAP, terminando así el proceso de invocación de un Servicio Web.

Seguidamente se muestra una descripción de los elementos que constituyen el eje principal de la tecnología de los Servicios Web.

3.3.3.1. Web Services Description Language (WSDL)

El lenguaje de descripción de Servicios Web, permite especificar en un formato o modelo XML todas las operaciones y tipos de datos de un Servicio Web. WSDL describe la interfaz pública a los servicios Web. Basándose en XML, describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligán después al protocolo concreto de red y al formato del mensaje.

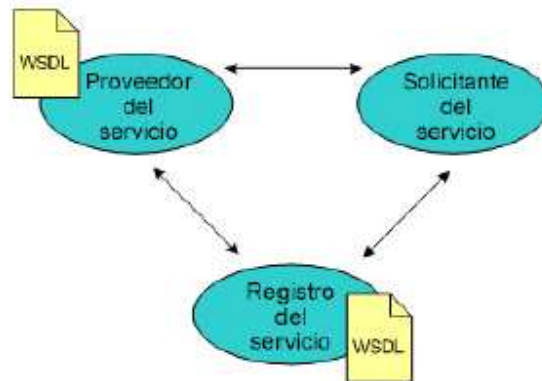


Fig. 13. Modelo WSDL.

Fuente: AGUIRRE (2007)

Un documento WSDL consta de varias partes que son:¹⁸

Tipos de datos <types>. Define los tipos de datos usados en los mensajes definidos en esquemas XML.

Mensajes <message>. Define los elementos del mensaje. Las partes pueden ser de cualquiera de los tipos definidos en la sección anterior.

¹⁸ <http://es.wikipedia.org>



Tipos de Puerto <portType>. Define las operaciones permitidas y los mensajes intercambiados en el servicio.

Bindings <binding>. Se especifican los protocolos de comunicación usados

Servicios <service>. Conjunto de puertos y dirección de los mismos.

Una descripción WSDL está estructurada en dos etapas fundamentales, una concreta y una abstracta. En el nivel abstracto, WSDL describe un Servicio Web en términos de los mensajes que éste envía y recibe. Los mensajes están descritos independientemente del formato utilizado, empleándose para ello un sistema de tipos como XML Schema. Una operación ('operation') asocia un patrón de intercambio de mensajes a uno o más mensajes. Un patrón de intercambio de mensajes ('message exchange pattern') identifica la secuencia y cardinalidad de los mensajes enviados y/o recibidos, así como a quién se envían y/o de quién se recibe. Una interfaz ('interface') agrupa las operaciones sin concretar un formato específico. En el nivel concreto, una vinculación ('binding') especifica los detalles relativos al formato para una o más interfaces. Por su parte, un punto final ('endpoint') asocia una dirección de red con una vinculación. Finalmente, un servicio ('service') agrupa los puntos finales que implementan una interfaz común.

3.3.3.2. Simple Object Access Protocol (SOAP)

SOAP es un protocolo basado en XML para el intercambio de información de una forma descentralizada sobre entornos distribuidos. SOAP define un mecanismo para el paso de instrucciones, comandos y parámetros entre clientes y servidores. Utiliza tecnologías XML para definir un marco de intercambio de mensajes extensible que permite trabajar sobre múltiples pilas de protocolos.

El protocolo SOAP funciona sobre HTTP, por lo que el mismo puede trabajar con Firewalls y Proxys ya existentes. SOAP se encuentra estandarizado por el W3C.

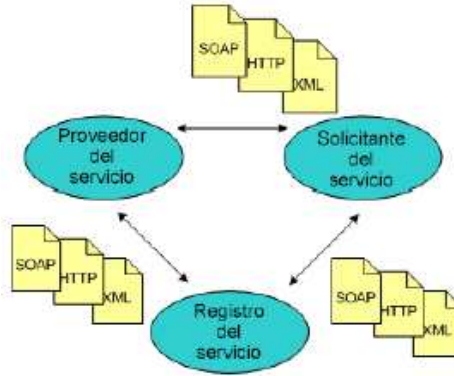


Fig. 14. Modelo SOAP.

Fuente: AGUIRRE (2007)

En la figura 14, se enumeran las partes de que consta un mensaje SOAP:

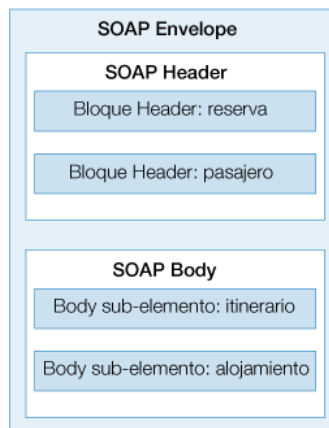


Fig. 15. Estructura de un mensaje SOAP.

Fuente: <http://www.w3c.es/Divulgacion/GuiasBreves/images/soap11.png>

Un mensaje SOAP contiene un elemento contenedor llamado **envelope** que contiene: Nombre, espacio de nombres, atributos y de forma opcional un elemento **header** o cabecera del mensaje, y un elemento esencial llamado **body** o llamado también cuerpo del mensaje. **Header**, proporciona un mecanismo para extender un mensaje SOAP de manera modular y descentralizada. Aquí es donde se incluye la información de control distinta a la carga útil de la aplicación que está contenida en el cuerpo del mensaje. Los elementos descendientes directos de **header**, son denominados **header blocks** o bloques de cabecera, que representan una agrupación lógica de datos y que pueden estar



destinados a nodos intermedios específicos en el camino entre el emisor y el receptor del mensaje. El elemento **body**, es obligatorio dentro de **envelope**, y proporciona un mecanismo para transmitir información relativa a la aplicación a receptor SOAP final.

3.3.3.3. Universal Description, Discovery and Integration (UDDI)

UDDI es una iniciativa de varias empresas, que ofrece un servicio gratuito para registrar y buscar servicios web. [23] Es un registro público diseñado para almacenar estructuradamente todo tipo de información de una empresa y de sus servicios. UDDI es un protocolo que proporciona una plataforma estándar que permite a las diferentes compañías y a las aplicaciones, encontrar de forma rápida, fácil y dinámica Servicios Web a través de la Web.

El registro de un negocio en UDDI se basa en estándares como HTTP y XML, y contienen las siguientes partes: [24]

- Páginas blancas, que registran dirección, contacto y otros identificadores conocidos.
- Páginas amarillas, categorización industrial basada en taxonomías.
- Páginas verdes, información técnica sobre los servicios que aportan las propias empresas.

UDDI es uno de los estándares básicos de los Servicios Web, cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL, en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros.

3.3.4. Arquitectura orientada a servicios

SOA (Arquitectura Orientada a Servicios) es un concepto de arquitectura de software que permite la utilización de servicios para dar soporte a los requisitos del negocio. A continuación, se muestra la definición de SOA propuesta por el consorcio OASIS¹⁹:

"Un paradigma para organizar y utilizar capacidades distribuidas que pueden encontrarse bajo el control de diferentes dominios de propiedad, proporciona un mecanismo uniforme

¹⁹OASIS: Organization for the Advancement of Structured Information Standards. (<http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf>)



para ofrecer, descubrir, interaccionar con y usar las capacidades para producir los efectos deseados de forma consistente con precondiciones y expectativas medibles". [25]

La aplicación de Servicios Web para el desarrollo SOA, es conveniente cuando la aplicación a desarrollar cumple los siguientes requisitos: debe de operar sobre Internet, donde la fiabilidad y velocidad no puede ser garantizada; donde no hay posibilidad de gestionar el despliegue de la aplicación de forma que todos los proveedores y solicitantes son mejorados de forma instantánea; donde los componentes del sistema distribuido se ejecutan en diferentes plataformas y productos propietarios; donde una aplicación existente necesite ser expuesta para su utilización a través de una red de comunicaciones y pueda ser 'envuelta' por un Servicio Web.

3.3.5. Ventajas y desventajas de los Servicios Web

Una de las principales ventajas que tienen la utilización de Servicios Web, es la integración de varios negocios con terceros, debido a la utilización de protocolos estándar como es HTTP, con esto se permite el paso de mensajes entre servicios o servidores sin necesidad de preocuparse por los firewalls que puedan poseer otras organizaciones.

Los Servicios Web integran plataformas, al poder emplearse desde cualquier programa, independientemente del sistema operativo que estén utilizando (Windows, Linux, etc.). Además aportan interoperabilidad entre aplicaciones software dependientemente de su implementación, del lenguaje de programación utilizado y de la plataforma en que estén instaladas.

La mayor limitante de la tecnología de los Servicios Web es su rendimiento. Comparado con otros modelos de computación distribuida, la prestación de esta solución es baja. Este problema viene ocasionado por el uso de XML como lenguaje para la codificación de mensajes. En particular, el problema radica en la necesidad de pasear y componer los ficheros XML, lo cual requiere mayor capacidad computacional y hace que las aplicaciones se ejecuten más lentamente.

Una de las desventajas de los Servicios Web es que la tecnología actual (UDDI, WSDL y SOAP) no proporciona los medios para conseguir un dinamismo utilizando software independiente y reutilizable.



3.4. Servicios Web Semánticos

3.4.1. Antecedentes

La WWW es una plataforma que permite a los usuarios de forma remota, interactuar y solicitar un servicio que algún proveedor ofrezca en la red. Pero para que esta interacción funcione, deben existir unos mecanismos de comunicación estándares entre diferentes aplicaciones, los mismos que deben poder interactuar entre sí para presentar la información de manera dinámica al usuario. Para ello es necesario contar con una arquitectura de referencia estándar que haga posible la interoperabilidad y extensibilidad entre las distintas aplicaciones y que permita su combinación para realizar operaciones complejas.

Los Servicios Web, estudiados en el apartado anterior, permiten que las aplicaciones compartan información y que además invoquen funciones de otras aplicaciones independientemente de cómo se hayan creado dichas aplicaciones e independientemente del sistema operativo o plataforma en que se ejecuten y de los dispositivos utilizados en el acceso. Los Servicios Web, aunque sean independientes entre sí, pueden vincularse para realizar una tarea específica.

A los Servicios Web tradicionales se incorporan componentes semánticos que permiten automatizar el descubrimiento, composición, invocación e interoperabilidad de Servicios Web.

Para remediar las limitantes mencionados en el estudio de los Servicios Web, y utilizando conceptos surgidos con la aparición de la Web Semántica, surgió lo que se denomina como Servicios Web Semánticos, donde se explica su apareamiento en la siguiente figura:

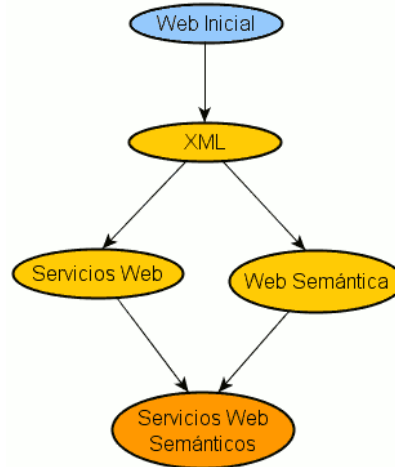


Fig. 16. Evolución de la Web.

Fuente: www.w3c.es/Presentaciones/2006/0420-servicios-JA/

En la figura anterior, se define como ha ido evolucionando la Web, aplicación que de manera conjunta utiliza conceptos de Servicios Web y de Web Semántica para crear Servicios Web Semánticos. Esta tecnología, es un nuevo paso adelante en la evolución de la Web, que consiste en describir a los Servicios Web con contenido semántico, de forma que el descubrimiento de servicios, su composición e invocación se puedan realizar de forma automática por parte de entidades software (por ejemplo, mediante agentes inteligentes) capaces de procesar la información semántica disponible.

Para poder añadir información semántica a los Servicios Web, se debe hacer uso de ontologías como OWL-S y WSMO que definen a los denominados Servicios Web Semánticos. A continuación se presenta un breve estudio de estas dos tecnologías mencionadas.

3.4.2. Web Ontology Language for Services (OWL-S)

OWL-S se fundamenta en la definición de ontologías escritas en OWL, que permiten la descripción de Servicios Web Semánticos en diferentes niveles de abstracción. Según este enfoque, la anotación semántica trata de dar respuesta a tres cuestiones esenciales para cualquier servicio web: qué ofrece el servicio, cómo funciona el servicio y cómo se interactúa con él. Para ello, OWL-S emplea tres secciones: *profile*, *model* y *grounding*.

[26]

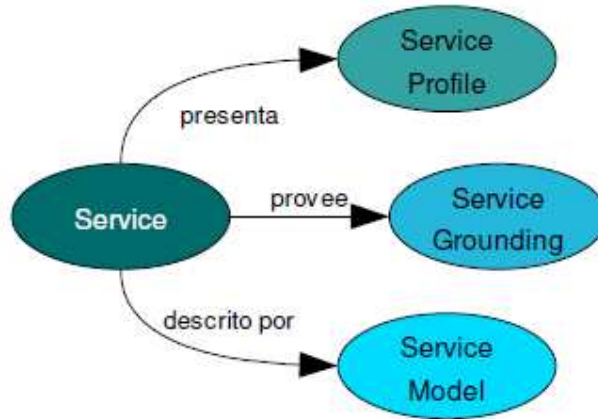


Fig. 17. Ontología de Alto Nivel de OWL-S.

Fuente: OWL-S 2004.

En la figura anterior se presenta la ontología de alto nivel de OWL-S para describir un servicio. En esencia esta ontología contiene los elementos fundamentales que caracterizan un servicio y que permite describir las capacidades que sustenta un servicio. Esta ontología contiene tres tipos de conocimiento fundamentales, como se mencionó anteriormente:

ServiceProfile, utilizado para publicitar el servicio. Se definen propiedades del servicio como nombre, descripción e información de contacto sobre los autores, publicadores, etc. Además de la información puramente identificativa, en el *ServiceProfile* se describe funcionalmente el servicio en base a las entradas, salidas, precondiciones y efectos, comúnmente conocidos como (*Inputs, Outputs, Preconditions* y *Effects*).

ServiceModel, como se usa el servicio. El modelo de un servicio es utilizado por agentes de búsqueda hasta de cuatro formas distintas: primeramente para analizar detalladamente si el servicio cumple los requisitos buscados, luego para componer descripciones de múltiples servicios, seguidamente para coordinar clientes durante la ejecución de los servicios, y por último para monitorizar la ejecución del servicio.

ServiceGrounding, como interaccionar con el servicio. Especifica cómo se interactúa con el servicio llegando a describir para ello el protocolo que lo implementa, el formato de los mensajes, serialización, transporte y direccionamiento (máquina y puerto) para ejecutarlo. Es, por tanto, el eslabón clave para completar la descripción semántica de un servicio y



establecer la correspondencia con éste. Esta correspondencia a nivel de ejecución del servicio se realiza siempre por cada proceso atómico.

Una característica a destacar en OWL-S es la distinción que hace entre servicios "atómicos" y servicios "compuestos". Los *servicios atómicos* son aquellos donde un solo programa accesible por la Web, sensor o dispositivo es invocado por un mensaje de solicitud, realiza su tarea y, posiblemente, produce una única respuesta a la solicitud. Por su parte, los *servicios compuestos* son aquellos que están constituidos por múltiples servicios primitivos y pueden requerir una interacción extendida o conversación entre el solicitante, y que el conjunto de servicios que están siendo utilizados.

3.4.3. Web Services Modeling Ontology (WSMO)

Dentro de WSMF (Web Services Modelling Framework) (Anexo 2), se propone una ontología llamada WSMO, esta propuesta está relacionada con la descripción semántica de los Servicios Web que cronológicamente siguió a OWL-S. De forma similar a la de los autores de OWL-S, WSMO proporciona un marco de trabajo conceptual y un lenguaje formal para describir de forma semántica todos y cada uno de los aspectos relevantes relacionados con los Servicios Web, facilitando de esta forma la automatización de tareas tales como el descubrimiento, la combinación y la invocación de servicios electrónicos sobre la Web.

En WSMO, un Servicio Web se define como una entidad computacional capaz, una vez invocada, de satisfacer el objetivo de un usuario.

Teniendo como base los conceptos identificados en WSMF, WSMO proporcionan una especificación ontológica para los elementos que conforman el núcleo de los servicios Web Semánticos. A diferencia de OWL-S, el lenguaje de ontologías utilizado con este propósito en WSMO es WSML. En la siguiente figura se muestra la arquitectura de WSMO. [2]

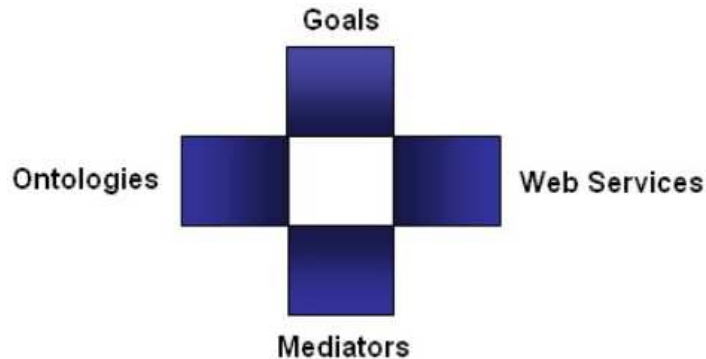


Fig. 18. Ontología de Alto Nivel WSMO

Fuente: <http://www.w3.org/>

Ontologías.- Proporciona la terminología empleada por el servicio, permitiendo la definición de funciones y reglas sin la necesidad de extender el lenguaje. Las ontologías definen una terminología común concertada indicando conceptos, relaciones entre conceptos y axiomas. Los axiomas se utilizan para capturar las propiedades semánticas de las relaciones y los conceptos.

Servicio Web.- Especifica los aspectos funcionales del servicio, la descripción del servicio está formada por sus capacidades y sus interfaces que especifican su comunicación. Se distingue, además, el concepto de servicio Web, entendido como entidad computacional capaz de satisfacer un objetivo de usuario, del concepto servicio, entendido como el valor real proporcionado por la invocación del servicio Web.

Objetivo.- Los objetivos describen aspectos relacionados con los deseos de los usuarios sobre la funcionalidad requerida. Se expresan en términos de la ontología. Los objetivos pueden ser descripciones de servicios Web que, potencialmente, satisfarían los deseos del usuario.

Mediadores.- Describen los elementos encargados de gestionar los problemas de interoperabilidad entre los elementos anteriores o conectarlos para ofrecer servicios más complejos.

A continuación se presenta un cuadro resumen con las principales características de estos lenguajes que se han descrito anteriormente.



OWL-S	WSMO
<ul style="list-style-type: none"> • Arquitectura no definida • La terminología del consumidor y proveedor debe ser la misma. • Para el proveedor y consumidor, no existe una separación explícita en la forma de definir cada uno, el Perfil es usado tanto para definir deseo del cliente como Servicio ofrecido. • Carece de Semántica formal. • Basado en Ontologías. • Permite una sola forma de interactuar con el servicio. • La invocación a los servicios es mediante WSDL. • Combina diferentes semánticas y notaciones, incluso áreas donde se deja la semántica abierta. • Descubre, invoca, compone y monitoriza recursos Web. 	<ul style="list-style-type: none"> • Arquitectura claramente definida • La terminología del proveedor y consumidor puede ser diferente, resolviéndose a través de mediadores. • Visiblemente se separa la representación del consumidor a través del Objetivo y del Proveedor por medio del Servicio Web. • No tienen definido la invocación del Servicio. • Proporciona una familia de lenguajes: WSML-Core, WSML-DL, WSML-Flight , WSML-Rule y WSML-Full • Compatible con la Web. • Basado en Ontologías

Tabla 4. Cuadro resumen de OWL-S y WSMO

Elaborado por: Oscar Tapia

Una de las diferencias entre las tecnologías antes mencionadas, es que OWL-S describe la funcionalidad del servicio (Service Profile) pero no separa de lo que un usuario quiere de la funcionalidad que un servicio puede proporcionar; en cambio WSMO diferencia entre el objetivo y descripción de Servicio de Web, lo que significa desacoplar los puntos de vista del proveedor del servicio y del solicitante. Un aspecto clave en WSMO es el uso de mediadores para resolver los problemas de interoperabilidad entre servicios Web en distintos aspectos (terminología utilizada, patrones de intercambio de mensajes, y relación entre los objetivos de un cliente y las capacidades que los servicios pueden ofrecer). En definitiva es preciso trabajar bajo el marco definido por WSMO partiendo del grado de desarrollo de OWL-S.



3.5. Ventajas y desventajas de los Servicios Web Semánticos

Los Servicios Web Semánticos, funcionan independientemente del sistema operativo que se use, esto quiere decir que esta tecnología es multiplataforma, inclusive facilitando el acceso a la información requerida desde cualquier dispositivo.

Los requerimientos de hardware, no son muy exigentes debido a que sólo es necesaria la utilización de un navegador.

Una de las desventajas de los Servicios Web Semánticos, es que la información privada, generalmente no se sabe en manos de quien va a caer, ni el uso que pueden dar los mismos a la información solicitada.

Desde un punto de vista más integral, y teniendo en cuenta las posibilidades ofrecidas por todas y cada una de las aproximaciones existentes en la actualidad para la provisión de Servicios Web Semánticos, los agentes deben ser capaces de realizar de forma autónoma las tareas de descubrimiento, selección, composición, ejecución y monitorización de los servicios, adaptándose en cada momento, a los cambios que se pueden producir en el entorno debido al dinamismo propio de este tipo de sistemas.



CAPÍTULO IV: PROPUESTA BASADA EN ONTOLOGÍAS, AGENTES INTELIGENTES Y SERVICIOS WEB SEMÁNTICOS COMO APOYO AL CITTES DE GESTIÓN DEL CONOCIMIENTO DE LA UTPL

4.1. Introducción

La Universidad Técnica Particular de Loja, tiene a su cargo varios CITTES²⁰, de los cuales muchos de ellos están encaminados al desarrollo de proyectos que involucran directamente a los profesionales en formación. Uno de estos CITTES, es el de Gestión del Conocimiento, el mismo que tiene como visión, ser referente latinoamericano de gestión del conocimiento, sociedad de la información y tecnología web en el ámbito universitario para compartir y crear conocimiento en beneficio de nuestra sociedad.

El éxito del internet se ha dado por diferentes factores, los mismos que también han originado problemas, sobre todo con la sobrecarga de información, variedad de fuentes y problemas consiguientes de interoperabilidad. El análisis y estudio de la Web Semántica ayuda a resolver estos problemas al permitir a los usuarios delegar tareas en herramientas de software. Incorporando semántica a la web, el software es capaz de procesar contenidos, razonar con ellos, combinarlos y realizar deducciones lógicas para resolver problemas automáticamente. La habilidad automática es el resultado de la aplicación de técnicas de inteligencia artificial, las cuales hacen uso de los agentes inteligentes para perfeccionar de manera óptima las búsquedas, agregando razonamiento y toma de decisión a los Servicios Web que almacenan altos contenidos.

Con el propósito de dar un carácter especializado a los contenidos temáticos, es indispensable hacer uso de las ontologías para la realización del mismo. El concepto de ontología está basado en la filosofía y recientemente se utiliza en informática para definir vocabularios que las máquinas puedan entender y que sean especificados con la suficiente precisión como para permitir diferenciar términos y referenciarlos de mejor manera.

²⁰ CITTES (Centros de Investigación Transferencia de Tecnología, Extensión y Servicios). Tienen como función específica la investigación y la extensión o servicio a la sociedad, además de contribuir al autofinanciamiento de la UTPL.



Las ontologías incluyen definiciones de conceptos básicos relacionados con un dominio, así como las relaciones entre ellos, de tal forma que los computadores pueden codificar el conocimiento y también el conocimiento extendido, haciendo reutilizable el mismo. Integrando todos los aspectos anteriores, las organizaciones y entidades, cuentan con herramientas tecnológicas eficientes de comunicación y organización de información, de tal forma que el conocimiento organizacional podrá ser gestionado, categorizado, retroalimentado y ampliamente divulgado.

En el presente capítulo, se analiza, el uso de las Ontologías, los Agentes Inteligentes y los Servicios Web Semánticos, como apoyo para el CITTES de Gestión de Conocimiento y se facilitan dos propuestas para desarrollarlas en futuras investigaciones o trabajos de fin de carrera, para dos problemas específicos como son: la consulta fechas de horarios de exámenes y video conferencias.

4.2. Situación actual del CITTES de Gestión de Conocimiento

Las instituciones de nivel superior en la actualidad trabajan con firmeza en torno al conocimiento, su gestión, y la tendencia en la colaboración y la apertura. En este contexto la UTPL creó un centro de investigación que se dedica en forma prioritaria a la gestión del conocimiento, que actualmente existe en cada dependencia. Para llevar a cabo este propósito, el CITTES de Gestión del Conocimiento consta de un equipo multidisciplinario que ha venido desarrollando proyectos investigativos o de gestión de la información en las diversas unidades que posee la Universidad.

Para establecer gestión del conocimiento, es necesario aprovechar y potenciar el recurso humano, la infraestructura, la tecnología disponible, el material educativo, productivo y de investigación; y el que se deriva de las relaciones con socios estratégicos, instituciones afines, universidades, etc.

Los objetivos que está cumpliendo actualmente el CITTES de Gestión de Conocimiento son los siguientes:²¹

- Ser un soporte de gestión y tecnología para la Universidad que permita crear sociedades de conocimiento que potencien el capital intelectual.

²¹ <http://www.utpl.edu.ec/gcblog/presentacion/>



- Facilitar la sincronía entre los requerimientos de las personas y la disponibilidad de información para crear un entorno de conocimiento.
- Fomentar un trabajo colaborativo en que los componentes de la organización compartan y difundan el conocimiento que poseen.
- Explotar continuamente el conocimiento para desarrollar y fortalecer relaciones con nuevos productos y servicios.
- Investigar las tendencias en modelos, recursos, tecnologías orientadas a gestión del conocimiento

4.3. Requerimientos

Actualmente la UTPL genera grandes cantidades de información muy valiosa para todo el personal que pertenece a la universidad como para los estudiantes de las diferentes modalidades. A continuación se enumeran algunos de los principales requerimientos que tiene el CITTES de Gestión del Conocimiento: (Según encuesta realizada a estudiantes e integrantes del CITTES de Gestión del Conocimiento) (Anexo 3)

- Encontrar diferentes vías de accesibilidad y confiabilidad de la información que circula en toda la Universidad.
- Buscar nuevas herramientas y tecnologías Web 2.0 para mejorar los servicios que ofrece el portal Web de la UTPL que a futuro debe incluir Servicios web Semánticos.
- Definir un modelo estratégico que pretenda utilizar la potencia de las tecnologías Web Semántica.
- Utilización de Servicios Web para simplificar y facilitar el acceso a la funcionalidad de búsqueda semántica, revisión y mantenimiento de información.
- Implementación de aplicaciones de Web Semántica.
- Mayor difusión de los objetivos del CITTES.

4.4. Servicios del portal web de la UTPL

Actualmente el portal de la UTPL brinda a la comunidad universitaria una serie de servicios los cuales se listan a continuación:



- Consulta de fechas y horarios de exámenes para estudiantes de las diferentes modalidades.
- Consulta de notas.
- Directorio telefónico de los docentes de la UTPL.
- Biblioteca Virtual.
- Servicios al estudiante.
- Información académica como matriculas, oferta de pre y postgrado.
- Entorno virtual de aprendizaje EVA.
- Aulas virtuales.

Cada uno de estos servicios genera información importante para el internauta, pero la congestión de ellos hace que exista demasiado flujo de información que hace desinformar o no obtener lo que el usuario desea.

Existen algunos inconvenientes en lo que respecta a la búsqueda de información exacta dentro de la web, hacer uso de una Web Semántica bien definida, permitirá que los inconvenientes que se mencionan a continuación sean solucionados.

- La gran cantidad de información que existe en el Internet²², como por ejemplo libros, publicaciones, videos, documentos, etc., están situados en páginas que no están vinculados específicamente con la universidad, y que no cuentan con una plataforma entendible y compatible para un buscador específico, hace que al instante de buscar cierta información, el resultado no sea el requerido por el usuario.
- Lo ideal sería almacenar toda la información en un solo portal web, pero como esto no es factible debido a los requerimientos que se necesiten para levantar la misma, resultaría imposible tanto en hardware como en tecnologías, la implementación de este ideal.
- El tratamiento de la información no permite realizar deducciones sobre ella, de manera que pueda ser encontrada como resultado, información que no estuviera explícitamente detallada.

²² www.intergraphicsdesigns.com/blog/2009/01/%C2%BFcuantas-paginas-web-hay-en-internet/



- La incompatibilidad de las tecnologías en servidores ubicados en cualquier lugar del mundo, hace imposible la comunicación de las mismas, haciendo imposible obtener o recopilar información de dichos servidores.

Con esta problemática se ha hecho participe del estudio del estado del arte de las tres tecnologías (Ontologías, Tecnología de Agentes y Servicios Web Semánticos) que ayudarán a evaluar y facilitar las metodologías adecuadas para resolver los problemas antes mencionados, especialmente en la problemática de gestionar la información de manera eficiente, como lo es al momento de buscar fechas y horarios de exámenes para los estudiantes que estudian en las diferentes modalidades que la universidad tiene a su cargo, como también la búsqueda de video conferencias específicas para un estudiante. Actualmente los recursos que se encuentran en la web, no se pueden encontrar con facilidad al momento de hacer una búsqueda específica, es por eso que es necesaria, la creación de un Servicio Web para que toda la información requerida pueda encontrarse mediante una búsqueda semántica dentro del portal de la UTPL.

El portal de la UTPL, está cada día más adaptándose, a las nuevas tecnologías de la información y especialmente a las demandas de los usuarios, cada vez más encaminados al manejo y búsqueda de información específica, para lo cual se propone la construcción de una arquitectura compuesta por un sistema multi-agente con tecnología Web Semántica, concebida para dar acceso a un requerimiento de información específica desde cualquier ordenador situado en cualquier lugar del mundo.

La infraestructura que se propone serviría para encontrar explícitamente, cualquier tipo de información (conocimiento), producido o generado por los diferentes departamentos que levantan información en el portal Web de la UTPL.

Como se mencionó anteriormente, el estudio de las ontologías, agentes inteligentes y servicios web semánticos, son la base fundamental de esta propuesta, las mismas que mediante su análisis y evaluación, se podrán determinar las tecnologías que nos ayudarán a buscar, evaluar y encontrar la información de otros portales web, donde no existe compatibilidad entre los estándares de comunicación existentes en la actualidad.



4.5. Propuestas

Para la mención de esta propuesta se tomarán algunos de los servicios que brinda el portal de la UTPL, y que son de vital importancia para los internautas, ya sea por la cantidad y/o calidad de la información que generan.

4.5.1. Solución para consulta de fechas y horarios de exámenes para estudiantes de las diferentes modalidades.

4.5.1.1. Descripción del problema

La gestión del conocimiento en las distintas organizaciones, se auxilia principalmente en diferentes tipos de herramientas que permiten representar el conocimiento o información en una forma dinámica.

Resumiendo a la problemática del servicio, los estudiantes tanto de la modalidad presencial o a distancia, antes de dar sus evaluaciones respectivas, buscan las fechas y horarios de sus exámenes, ya sea telefónicamente, en la página web de la UTPL, o a su vez enviando un correo electrónico al administrador solicitando dicha información. Estos dos últimos métodos, hacen que la información que reciben a veces no sea la correcta, o también no sé de contestación inmediata a la misma, y si se realizan búsquedas dentro del portal Web de la UTPL, el resultado esperado, se lo encuentra en un tiempo que para el usuario no es satisfactoria.

4.5.1.2. Escenario de pruebas

Para demostrar lo anteriormente dicho en la problemática, se ingresó al portal web de la UTPL (<http://www.utpl.edu.ec/search/>), en donde se realizó una búsqueda de los horarios de exámenes para la modalidad de estudios a distancia, poniendo primeramente como palabra clave "Horarios Exámenes Modalidad Abierta", el cual, no arrojó ningún resultado como se muestra en la figura 19.



Fig. 19. Búsqueda con palabra clave “Horarios exámenes modalidad abierta”.

Fuente: <http://www.utpl.edu.ec/search/node/horarios%20exámenes%20modalidad%20abierto>

Seguidamente se introdujo la palabra clave “horarios abierta” en donde la búsqueda arrojó algunos resultados, dando como derivación que ninguno de ellos nos llevó directamente a la información requerida por el usuario o estudiante, como se muestra en la figura 20. Para tener una idea de los lineamientos de un tema en especial, se tuvo que ingresar a algunos de los links encontrados y luego de leer minuciosamente algunos de ellos, se puede encontrar lineamientos acerca de un tema de tesis que al estudiante le puede interesar. Cabe destacar que el proceso de búsqueda que actualmente lleva a cabo el portal web de la UTPL, es mediante palabras clave, la misma que no da un resultado exacto sobre la información que el estudiante solicita.



Fig. 20. Búsqueda con palabra clave “Horarios abierta”.

Fuente: <http://www.utpl.edu.ec/search/node/horarios%20%20abierta>

4.5.1.3. Propuesta de solución

Para poder trabajar con Web Semántica y poder resolver la problemática descrita anteriormente, es necesaria primeramente la creación de una o varias ontologías, esto ayudará a describir explícitamente los conceptos y recursos del dominio. Para la creación de la ontología, es necesario seguir los siguientes pasos: [27] En primer lugar determinar el dominio y alcance de la ontología, es decir determinar para que se va a utilizar la ontología, seguidamente considerar la utilización de ontologías existentes (reutilización), luego enumerar o escribir una lista de los conceptos y propiedades importantes que se va a manejar en la ontología, a continuación definir los conceptos y su jerarquía, determinar lo tipos de propiedades de las clases, definir relaciones y finalmente la creación de instancias individuales. La creación de esta ontología, es la base tecnológica fundamental para la interoperabilidad semántica para resolver el problema propuesto.

Con la finalidad de estructurar y representar el conocimiento de las ontologías creadas siguiendo los pasos antes mencionados, se necesitan lenguajes de marcado específicos. Actualmente, mediante anotaciones RDF, RDF-Schema se pueden representar algunas facetas sobre conceptos de un dominio y permite, mediante relaciones taxonómicas, crear una jerarquía de conceptos. Pero se necesitan lenguajes de marcado (basados en RDF)



con mayor expresividad y capacidad de razonamiento para representar los conocimientos que contienen las ontologías. El lenguaje con gran capacidad expresiva que está emergiendo como un estándar para realizar anotaciones de ontologías en web es DAML (OWL-S)²³.

Para la consulta de fechas y horarios de exámenes de los estudiantes, se propone elaborar una jerarquía de conceptos relacionados, almacenados en RDF donde se registren cada uno de los horarios y fechas de los exámenes ordenados y categorizados por áreas.

Por otro lado, se necesitan agentes inteligentes y una aplicación web que exploten este conocimiento anotado en los RDF. Estos agentes de conocimientos web serán capaces de interpretar los esquemas ontológicos y axiomas de servicios web, mantendrán la consistencia de las instancias que se inserten en las páginas web siguiendo los esquemas ontológicos definidos, realizarán una búsqueda con inferencias utilizando los axiomas situados en los esquemas, y podrán realizar ligaduras de los árboles taxonómicos de varias ontologías.

Para potenciar el uso de ontologías en la web, y para que se acoplen a los servicios, se necesitan aplicaciones específicas de búsqueda de ontologías, como OntoAgent²⁴, que indique a los usuarios las ontologías existentes y sus características para poder utilizarlas en su sistema.

Así mismo es necesaria una aplicación con un Agente Inteligente que vaya respondiendo automáticamente los correos que tengan como objetivo la obtención de algún horario y fecha específica. Para este servicio se tendría que desarrollar una aplicación web, la misma que puede ser en el lenguaje php, ya que es el lenguaje que utiliza el CITES de Gestión del Conocimiento. Esta aplicación tendrá que filtrar los correos que lleguen al Call Center que tengan como tema alguna petición de información de las fechas y horarios, una vez localizados estos correos, la aplicación web se enlazará con el agente inteligente que interprete los parámetros obtenidos por el correo con los esquemas ontológicos de los RDF para obtener la información necesaria.

²³DAML. Ontología para la descripción de un Servicio Web, permite a un agente software localizar, invocar, interoperar y monitorizar dicho servicio.

²⁴ OntoAgent . Aplicación específica de búsqueda de Ontologías en la Web. Indica a los usuarios las ontologías ya existentes y sus características para poder utilizarlas en su sistema.



A continuación se presenta un esquema de la solución antes mencionada.



Fig. 21. Proceso Jerárquico del Servicio Web.

Desarrollado por: Oscar Tapia F.

En la figura anterior, se indica el proceso jerárquico que se sigue para la obtención de la información por parte del internauta. Primeramente la información debe ser almacenada en los RDF previamente diseñados, seguidamente para su análisis, este necesita de un lenguaje intérprete de RDF (DAML), que sirve para una correcta comunicación con el agente inteligente (OntoAgent), el cual analiza la petición del internauta en los RDF, y devuelve la información requerida. Para aplicar lo antes mencionado, es necesario el desarrollo de una aplicación web (blog²⁵), en el cual el estudiante se registraría una primera vez, y conforme vaya pasando los ciclos filtre información relacionada a su ciclo actual. La herramienta Web consumirá toda la información de las ontologías para luego ser publicadas en un solo frontal.

En la siguiente figura, se acopla el blog para la propuesta de consulta de fechas y horario de exámenes.

²⁵ Sitio Web o bitácora periódicamente actualizado que recopila cronológicamente textos o artículos de varios autores, apareciendo primero el más reciente. (Citado por www.wikipedia.com)



Fig. 22. Blog con proceso para búsqueda de fechas y horarios de exámenes.

Desarrollado por: Oscar Tapia F.

4.5.2. Solución para la búsqueda de video conferencias

4.5.2.1. Descripción del problema

Resumiendo a la problemática del servicio, los docentes de la modalidad abierta de la UTPL periódicamente realizan grabaciones de las tutorías virtuales de las materias que imparten, estos videos son almacenados en el portal de youtube, en donde la cuenta de la UTPL tiene cientos de videos de las diferentes asignaturas. Existe el problema que el estudiante de una carrera por ejemplo Ingeniería en Sistemas, busque un video específico como “tutorial de estadística” y como resultado le salgan videos de estadística de las diferentes carreras que posee la UTPL, esto causa cierto malestar por parte del estudiante ya que no puede obtener la información que requiere.

Debido a que los recursos que se encuentran en la web, no se encuentran con facilidad al momento de hacer una búsqueda específica, es necesaria la creación de un Servicio Web para que toda la información requerida pueda encontrarse mediante una búsqueda semántica dentro del portal de la UTPL.

4.5.2.2. Escenario de pruebas

Para demostrar lo que se ha mencionado en la definición del problema, se ingresó igualmente como en la problemática anterior, al buscador del portal web de la UTPL (<http://www.utpl.edu.ec/search/>), donde se realizó una búsqueda de videos de tutorías virtuales poniendo como palabra clave “video tutorías virtuales”, el mismo que no arrojó ningún resultado en su búsqueda como se muestra en la siguiente figura:



Fig. 23. Búsqueda con palabra clave “video tutorías virtuales”.

Fuente: <http://www.utpl.edu.ec/search/node/video%20tutorias%20virtuales>

4.5.2.3. Propuesta de solución

Para mejorar la búsqueda de los recursos existentes en el portal web de la UTPL como videos, documentos, pdfs, libros etc., es necesaria la utilización de un mecanismo que utilice un Servicio Web con una semántica bien definida para que dichos documentos que se quieren encontrar, se actualicen periódicamente y al momento de encontrar dicha información, la misma sea lo más exacta y precisa. Para lo cual se puede crear un Servicio Web con todos los recursos o información que se pretende acceder, o a su vez disponer de una de las siguientes soluciones disponibles actualmente. ISYS²⁶ y RSS²⁷.

ISYS es un Servicio Web que se integra como motor de búsqueda en un servidor de una empresa en particular, puede leer archivos pdf y fuentes de datos SQL convirtiéndolos dinámicamente en HTML, disponiendo de un agente inteligente que constantemente controla la base de conocimiento escogida por el usuario.

Para resolver el problema de este apartado, es necesaria la utilización de un vocabulario RDF basado en XML que permita la catalogación de información precisa adaptada especialmente a la preferencia de los usuarios, la familia de formatos RSS es la que nos permite realizar lo mencionado anteriormente. Los archivos RSS contienen metadatos

²⁶ <http://www.isysdev.com/products/web.html>

²⁷ RSS es una familia de formatos de fuentes Web codificados en XML. Se utiliza para suministrar a suscriptores de información actualizada frecuentemente. RSS sirve para facilitar el acceso a la información Web que se actualiza con regularidad, logrando que el usuario recupere al momento las novedades producidas en la información de su interés.



sobre fuentes de información especificadas por los usuarios, cuya función principal es avisar a los mismos, que los recursos que ellos han seleccionado para formar parte de esa RSS, han cambiado sin necesidad de comprobar directamente la página, es decir, notifican de forma automática cualquier cambio que se realice en esos recursos de interés seleccionados.

Si no se maneja ninguna de las dos soluciones mencionadas anteriormente, se debe luego de desarrollar el Servicio Web, diseñar y crear una Ontología para este dominio, con todos sus conceptos y relaciones bien definidas, seguidamente describir al Servicio Web con contenido semántico, utilizando WSMO para este objetivo, y finalmente utilizar un Agente Inteligente como en la propuesta del apartado 4.5.1, para recuperar la información que se encuentra almacenada y anotada semánticamente dentro del Servicio Web.

Si la información que se pretende buscar se encuentra en distintos departamentos que contienen diferentes bases de datos, es necesario llevar a cabo una integración de los procesos internos mediante el Servicio Web descrito, esto se puede realizar a través de la utilización de la tecnología SOAP que es un protocolo escogido para el intercambio de datos levantada en formato XML, WSDL para la descripción del Servicio Web y UDDI para la publicación de los mismos.



Discusión

Si bien el concepto de ontología, podría, haberse incluido como parte del estudio de los servicios web semánticos, se ha optado por separarlo en una sección con el fin de enfatizar la importancia de este elemento en la solución y cumplimiento del objetivo del presente trabajo investigativo. Una de las principales características de las ontologías, es que permiten trabajar con conceptos, al momento de utilizar sistemas de recuperación de información, no como antes o actualmente se lo hace que es mediante palabras clave. Estos sistemas describen el contenido de los repositorios de datos independientemente de la representación sintáctica de los mismos, posibilitando de esta manera su integración semántica.

Cabe mencionar que una ontología puede crearse para un dominio en particular, por ejemplo si necesito crear una ontología para la biblioteca de la UTPL, la misma no me serviría por ejemplo para la biblioteca de la UNL, aunque tuviesen la misma estructura ambas bibliotecas, son dos casos completamente distintos, debido a que si por ejemplo en un buscador, yo solicito datos de "Biblioteca UTPL", obviamente me dará resultados relacionados con la misma y no con biblioteca UNL, esto se debe a que cada caso contienen una ontología propia. Por lo que se concluye diciendo que una ontología de forma general no puede abarcar ambos casos, como es el caso particular de las bibliotecas, aunque se parezcan demasiado.

Para poder explotar la web semántica, se necesitan lenguajes de marcado apropiados que representen el conocimiento de las ontologías. Mediante los lenguajes RDF, RDF-S, se puede representar algunas de las facetas sobre conceptos de un dominio en particular, permitiendo mediante relaciones taxonómicas, crear una jerarquía de conceptos. Esto se puede lograr a través de lenguajes de mayor expresividad y capacidad de razonamiento como lo es OWL, ya que mediante este lenguaje con gran capacidad de expresividad, se puede representar el conocimiento que contiene una ontología en particular.

La posibilidad de acceso a cualquier tipo de información en la actualidad es demasiado alta, los agentes inteligentes son en la actualidad una buena alternativa para ahorrar tiempo en las búsquedas de información e incluso para análisis y atención a diversas peticiones.



La definición de agente inteligente que se aplica en este trabajo es la siguiente:

“Agente Inteligente es una entidad software situada en un lugar específico, capaz de expresar un comportamiento autónomo, reactivo y proactivo sobre este lugar, preparado para interactuar con otros agentes, con el fin de satisfacer determinados objetivos establecidos por un ente, ya sea software o humano, que es representado por el agente”.

Un ejemplo de un agente inteligente que opera actualmente en la web, es Google Alerts, que enfoca su búsqueda a documentos, noticias, videos, blogs, etc., en donde puede configurarse para que el propio agente pueda enviar una alerta o la búsqueda específica al correo electrónico del usuario, esto cada vez que el sistema encuentre lo solicitado. Se cree que en los próximos años, el internet estará invadido de agentes inteligentes virtuales que estarán constantemente trabajando para satisfacer los requerimientos para el cual fueron creados.

El mayor problema de la tecnología de agentes, es la utilización de protocolos de comunicación no estándar, el uso de estos protocolos impide la posibilidad de que dos agentes situados en distintos lugares puedan comunicarse con éxito si entre ellos existen cortafuegos o firewalls. El uso del protocolo HTTP para la comunicación en el mundo de los Servicios Web permite a los servicios una comunicación fluida y eficiente, salvo el escollo que se mencionó anteriormente.

Hay que tener en cuenta la dificultad que constituye la elaboración de un Sistema Multi-Agente, y la necesidad, por tanto de metodologías para guiar el proceso de desarrollo. La elección de una metodología no es tan simple, debido a que cada metodología, por el bagaje de sus creadores, se especializa en áreas concretas. Por ejemplo, si se está acostumbrado a trabajar con sistemas basados en conocimiento con la metodología CommonKADS, lo lógico es que se elija MASCommonKADS. Por otro lado si la experiencia del usuario está en el área de los objetos, la recomendación sería de MaSE. Pero, si se está interesado en un enfoque más orientado a agentes, puede seleccionar ZEUS, INGENIAS, BDI o GAIA. Y si lo que se quiere es tener un soporte de herramientas, la lista de metodologías válidas se reduce considerablemente: ZEUS, MaSE, INGENIAS. En los casos en que se requiera un proceso de desarrollo robusto, detallado y ensayado en desarrollos reales, la recomendación sería MASCommonKADS o INGENIAS.



Los Servicios Web Semánticos surgieron de la combinación de la Web Semántica y de los Servicios Web, para llegar a comprender los inconvenientes de esta tecnología, fue necesario conocer los detalles relativos a sus elementos constituyentes. Así, primeramente se introdujo al estudio de la Web Semántica, donde sus antecedentes indican el motivo del porque la web se ha convertido en un repositorio gigantesco de información.

Cabe mencionar que el problema de las soluciones previas a la tecnología de los Servicios Web, es la limitación en la interoperabilidad, para lo cual la solución radica en encontrar un conjunto de estándares que fueran aceptados en todo el mundo, para diseñar y construir una tecnología que fuese más acorde a la visión del Internet.

Una de las mayores ventajas que se determinó en la tecnología de los Servicios Web, es la utilización de protocolos de transporte estándar como HTTP. Esta medida permite el paso de mensajes entre servicios sin necesidad de preocuparse por los sistemas de seguridad y, en particular, los cortafuegos de las distintas organizaciones. En general, el uso de protocolos estándar facilita la interoperabilidad entre plataformas de distintos fabricantes. Otra ventaja es la independencia con respecto a la implementación, esto aporta flexibilidad al sistema, de forma que se puede modificar la implementación del servicio sin necesidad de actualizar su descripción. También, los Servicios Web, permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos, puedan ser combinados fácilmente para proveer servicios integrados.

Los servicios Web Semánticos, se definen como la aplicación conjunta de conceptos de Servicios Web y de Web Semántica para crear Servicios Web Inteligentes. Esta tecnología consiste en describir a los Servicios Web con contenido semántico.

Por otra parte los Servicios Web Semánticos surgieron con el objetivo de permitir la utilización de los servicios por parte de los sistemas software, sin necesidad de la intervención humana. La descripción semántica de las capacidades de los servicios, permite a entidades software determinar las posibilidades de dicho servicio de forma similar a como lo podría hacer un humano, de manera que los procesos de descubrimiento, selección, composición, invocación y monitorización de tales servicios, se puedan hacer de manera automática.



Para terminar con el análisis de los Servicios Web Semánticos, se puede decir que los mismos, desde sus inicios, se han ligado con los agentes software (agentes inteligentes), que conjuntamente pueden procesar las descripciones semánticas que requieran los servicios. Partiendo de esto, se plantea primeramente, el estudio de plataformas de trabajo para la integración de Tecnología de Agentes y Servicios Web Semánticos, y para futuros trabajos de fin de carrera, el desarrollo de una nueva herramienta que contenga los beneficios de la integración de estas dos tecnologías.

Como se ya mencionó anteriormente uno de los principales inconvenientes que posee la web actual y que es de conocimiento de todos los internautas, es la poca precisión al momento de buscar un tipo de información en particular, se ha puesto énfasis para proponer las herramientas y tecnologías necesarias que cumplan con las expectativas del usuario y sobretodo solucione los problemas al momento de buscar información en el portal de la UTPL.

Para las dos propuestas se han dividido en tres partes que son, la definición del problema, la esencia de las pruebas y la propuesta. Se ha determinado que tanto para la búsqueda de fechas de horarios de exámenes y video conferencias dentro del portal de la UTPL, los resultados con palabras clave son insuficientes para llenar las expectativas del usuario, por lo que es necesario tomar las medidas necesarias para la búsqueda de tecnologías que ayuden a solucionar los problemas descritos.

Para la primera propuesta se hace necesaria la creación de ontologías, que ayuda a describir los conceptos y recursos del dominio, esta es la base fundamental para la interoperabilidad semántica de los datos buscados. Mediante anotaciones RDF, se estructurará y representará el conocimiento de las ontologías mediante relaciones taxonómicas y la creación de una jerarquía de conceptos. Para lo cual se adoptó la utilización de un lenguaje de marcado basado en RDF como lo es DAML, que es una ontología para la descripción de un servicio web que permite a un agente software localizar, invocar, interoperar y monitorizar dicho servicio: Posteriormente se utiliza OntoAgent que sirve para indicar a los usuarios las ontologías existentes y sus características para poder ser utilizadas en su sistema.



Es muy necesaria la creación de un blog para que el estudiante se registre, para que mediante la propuesta mencionada, consuma toda la información de las ontologías para luego ser publicada en un solo frontal.

En lo que respecta a la segunda propuesta, se vio necesario la utilización de un vocabulario RDF basado en XML que permita la catalogación de la información. A través de RSS, que son archivos que contienen metadatos sobre fuentes de información especificadas por los usuarios, se puede avisar a los mismos que los recursos que ellos han seleccionado, forman parte de esa RSS.



Conclusiones

- La utilización de ontologías como punto de partida para el desarrollo de aplicaciones basadas en web semántica, permite el intercambio de información entre programas, través de una jerarquía de clases, permitiendo la relación con otras, simplificando la unificación de distintas representaciones y facilitando la comunicación entre las personas que utilizan dichos servicios.
- Tanto la tecnología de agentes como los servicios web semánticos, pueden utilizarse de forma separada para el desarrollo de aplicaciones, debido a que los agentes se benefician de propiedades tales como autonomía, flexibilidad, aprendizaje, inteligencia, etc., y los servicios web semánticos proporcionan un alto grado de interoperabilidad entre plataformas, sistemas operativos y lenguajes de programación.
- El mayor problema de la tecnología de agentes y que, en mayor medida, ha restringido su aplicación a entornos meramente de investigación, es la utilización de protocolos de comunicación no estándares, impidiendo la comunicación entre agentes situados en distintas localizaciones, debido a que entre ellos existe e interponen diferentes cortafuegos o firewalls.
- La capacidad semántica de los servicios web, permite a entidades software determinar las posibilidades de dicho servicio de forma similar a como lo podría hacer un ente humano, de manera que los procesos de descubrimiento, selección, composición, invocación y monitorización de los mismos, se puedan hacer de forma automática.
- Utilizar web semántica, ayudará a encontrar los resultados más adecuados en cualquier tipo de búsqueda realizada por los usuarios de la red global, debido a que todo el contenido de las páginas que se encuentran subidas al internet, estarán anotadas semánticamente, esto se da a través del uso de motores de búsqueda que ofrecen resultados acordes a una ontología donde se encuentra dicha información.
- Para encontrar la eficiencia en la búsqueda de fechas de horarios de exámenes, video conferencias o cualquier otro tipo de documentación, por parte de los usuarios o estudiantes, es necesario llevar a cabo una integración de los procesos internos a través de un servicio web, llevado a cabo mediante la utilización de la tecnología



SOAP que es un protocolo escogido para el intercambio de datos levantada en formato XML, WSDL para la descripción del servicio web y UDDI para la publicación de los mismos.



Recomendaciones

- Se recomienda el estudio y desarrollo de aplicaciones que hagan uso de la tecnología de agentes y de servicios web semánticos, para la elaboración de sistemas que puedan tratar con el dinamismo de la web, al tiempo que se pueda beneficiar de características como la autonomía, el aprendizaje y el razonamiento.
- Las diferencias conceptuales entre agentes inteligentes y servicios web semánticos, justifican la necesidad de proponer un nuevo trabajo de fin de carrera para la búsqueda de tecnologías y herramientas que ayuden a resolver el inconveniente de integrar los mismos.
- Para la implementación y desarrollo de la propuesta para la búsqueda de fechas y horarios de exámenes para estudiantes de las diferentes modalidades, se recomienda elaborar una jerarquía de elementos bien diseñada, manteniendo un orden en la representación de la información, permitiendo mejorar el rendimiento del intérprete para que la ontología consuma los recursos y optimice el tiempo de envío de la información solicitada por el usuario o estudiante.
- Usar ontologías en el desarrollo de nuevas aplicaciones para los servicios que actualmente posee la UTPL, de manera especial con lo que respecta a Biblioteca Virtual, debido al impacto que posee el mismo, tanto para el acceso a la información requerida, como para el tiempo de búsqueda necesaria, para encontrar la documentación solicitada por el usuario o estudiante.
- En el caso que se desee implementar un servicio web dentro de la universidad, se debe tener bien claro, que servicios web de la UTPL se desea acoplar como ontología, ya que dependiendo del tipo de servicio que se desee implementar, se escogerá la metodología y el proceso de desarrollo adecuado.



ANEXOS

Anexo 1

Agente Semántico

“Pedro Gómez ha puesto en marcha una tienda de lámparas en su barrio. Se acerca la hora de hacer el primer pago fraccionado del IRPF (impuesto) y no sabe qué impreso usar ni cómo calcular el rendimiento neto empresarial. Preocupado por hacer mal las cosas y recibir luego una simpática carta de Hacienda, Pedro ha decidido usar un agente semántico para encontrar un asesor contable y fiscal. Para ello, se conecta a la Web semántica y elige un agente personal que parece fiable.

Escribe en un cuadro de texto "Busco asesor contable y fiscal (cerca de la calle Arboleda, en Soria). Quiero que sea barato y bueno", y el agente se pone en marcha. Primero obtiene una lista de todos los asesores que están a menos de dos kilómetros de la calle Arboleda; luego, para comprobar la calidad de cada uno, investiga si tienen reclamaciones de clientes, quejas ante los colegios profesionales o denuncias. A continuación, busca los que tienen tarifas bajas. Finalmente, selecciona a dos asesores y se los propone a Pedro.

A éste no le gusta ninguno de los dos. Uno no es del todo barato que Pedro desearía, y el otro no puede atenderle hasta dentro de dos semanas. Decepcionado con los resultados, Pedro escribe a su agente semántico: "Mira otras opciones". El agente repite el trabajo que había hecho, pero no encuentra más asesores que cumplan las condiciones del usuario. Entonces prepara una lista de los asesores cercanos a la calle Arboleda y de buena reputación, independientemente de que sean caros o baratos; acto seguido, contacta con los agentes semánticos de esos asesores para intentar negociar una reducción de las tarifas que aplican. Sólo uno acepta la reducción. Tras una rápida negociación entre agentes, la reducción se fija en un 20% (con la condición de que el cliente pagará cada trimestre de asesoría por anticipado y al contado).

El agente muestra a Pedro los datos del asesor elegido y la tarifa que ha quedado establecida. Pedro desconfía de la tarifa, porque le parece demasiado barata, y le pregunta a su agente: ¿Cómo has negociado la tarifa? ¿Cómo sabes que ese asesor es bueno? El agente, dotado de paciencia infinita, le proporciona pruebas de todo. Contento por ahorrarse un dinero, Pedro cierra su agente personal y se levanta del ordenador.



Mientras se prepara un café, piensa en la web semántica: "Es buena idea esto de la nueva Web. ¡Y pensar que hace diez años parecía ciencia-ficción! Hasta me están cayendo simpáticos los bichos esos, los agentes".

Anexo 2

WSMF

WSMO está basado en el modelo conceptual propuesto en el "Web Service Modeling Framework" (WSMF), que identifica cuatro elementos fundamentales para describir Servicios Web Semánticos:

1. *Ontologías*: Proporcionan la terminología que será usada por los restantes elementos.
2. *Objetivos*: Representan los deseos de los usuarios o intenciones que deben ser satisfechas por algún servicio Web.
3. *Descripciones de servicios Web*: Define los aspectos funcionales y de comportamiento de un servicio Web.
4. *Mediadores*: Tienen el propósito de gestionar de forma automática los problemas de interoperabilidad que surjan entre los restantes elementos.

Anexo 3

Encuesta

La presente encuesta, ayudará a determinar los requerimientos que tiene el CITES de Gestión del conocimiento, con respecto al manejo, tratamiento y búsqueda de la información que posee actualmente la universidad, y de manera especial a la que se encuentra en el portal de la UTP, y que es manejada y solicitada por los estudiantes.

1. Usted ha utilizado alguna vez el buscador que posee el portal de la UTPL?
Si () No ()
2. Si contestó Si a la pregunta anterior, al momento de recibir una respuesta por parte del buscador, usted encontró:
() La información exacta y precisa?
() Algo parecido a lo que usted solicitó?



() No encontró nada?

3. Piensa usted que el portal de la UTPL, cumple con la necesidad y manejo de la información por parte de los estudiantes y cibernautas?

Si () No ()

Por que

4. Cree usted que la Web Semántica ayudará a mejorar los procesos de búsqueda dentro del portal de la UTPL? , conoce de alguna tecnología en especial que se menciona a continuación?

Si () No ()

OWL () RDF () XML ()

5. Cite algún requerimiento que usted crea necesario implementar dentro del CITTES de Gestión del Conocimiento.

.....
.....



BIBLIOGRAFÍA

- [1] LEIVA, X. (2004): “*Definiciones Ontológicas*”. [Citado 12 mayo 2009] [En Línea]. Disponible en: <http://www.javierleiva.info/definiciones-ontologicas>
- [2] GARCIA, F. (2007): “*Sistema basado en tecnologías del conocimiento para entornos de Servicios Web Semánticos*”. [Citado 12 diciembre 2008] [En Línea]. Disponible en: <http://digitum.um.es/xmlui/bitstream/10201/154/3/GarciaSanchezFrancisco.pdf.txt>
- [3] LAMARCA, M. (2009): “*Ontologías: Hipertexto el nuevo concepto de documento en la cultura de la imagen*”. [Citado 21 enero 2009] [En Línea]. Disponible en: <http://www.hipertexto.info/documentos/ontologias.htm>
- [4] ABIÁN, M. (2005): “*Ontologías: Que son y para qué sirven*”. [Citado 30 enero 2009] [En Línea]. Disponible en: <http://www.wshoy.sidar.org/index.php?2005/12/09/30-ontologias-que-son-y-para-que-sirven>
- [5] RAMIREZ, J. (2006): “*Las Ontologías como herramienta en la Gestión de Conocimiento*”; [Citado enero 2009] [En Línea]. Disponible en: <http://www.congreso-info.cu/UserFiles/File/Info/Info2006/Ponencias/208.pdf>
- [6] RODRIGUEZ, S, (2007): “*Web Semántica: Vista General de RDF*”; [Citado enero 2009] [En Línea]. Disponible en: http://www.tec.url.edu.gt/boletin/URL_05_SIS02.pdf
- [7] REAL ACADEMIA ESPAÑOLA (2009): “*Agente: Definición de Agente*”. [Citado 15 mayo 2009] [En Línea]. Disponible en: <http://www.rae.es>
- [8] GARCIA, G. (2003): “*RDF y RDF Schema*”. [Citado 28 enero del 2009] [En Línea]. Disponible en: http://www.matem.unam.mx/~greicia/semantic_web/rdf.html
- [9] WIKIPEDIA. (2009): “*Inteligencia Artificial. Enciclopedia de libre Contenido*”. [Citado 23 junio 2009] [En Línea]. Disponible en: http://es.wikipedia.org/wiki/Inteligencia_artificial
- [10] MAMAMI, V. (2009): “*Inteligencia Artificial. Agentes Estímulo Respuesta*”. [Citado 23 junio 2009] [En Línea]. Disponible en:



<http://www.slideshare.net/boreasH/inteligencia-artificial-agentes-estimulo-respuesta-victor-mamani-catachuraboreash>

- [11] QUINTERO, A., RODRIGUEZ, S., Y UCRÓS, M. (2007): "*Sistemas Multi-agente: Agentes y Sistemas multi-agente: Integración de Conceptos Básicos*". [Citado 23 junio 2009] [En Línea]. Disponible en: <http://agamenon.uniandes.edu.co/yubarta/agentes/agentes.htm>
- [12] CUESTA, P. (2007): "*Sistema Multi-agente. Planeta Blogs*". [Citado 20 Junio 2009] [En Línea]. Disponible en: <http://sistemasmultiagente.blogspot.com/>
- [13] BEAM, C. Y SEGEV, A. (1997): "*Automated Negotiations: A survey of the State of the Art*". [Citado 20 mayo 2009]. Disponible en la WWW.
- [14] TRAN, QUYNH, N., INFORMATION SYSTEMS. (2005): "*MOBMAS: A methodology for ontology-based multi-agent systems development*". [Citado 14 julio 2009] [En Línea]. Disponible en: <http://unsworks.unsw.edu.au/vital/access/manager/Repository/unsworks:952>
- [15] BERNERS-LEE, T., HENDLER, J., Y LASSILA, O. (2001): "*The Semantic Web. Scientific American*". [Citado 11 Mayo 2009].
- [16] BERNERS-LEE, T. (2009): "*Metadatos: Word Wide Web Consortium*". [Citado 17 mayo 2009] [En Línea]. Disponible en: <http://www.w3c.org>
- [17] BARCELÓ, M., GUZMÁN, G., Y PEREZ, A. (2006): "*Evolución de la Web. La Web Semántica como apoyo a la Gestión del Conocimiento y al Modelado organizacional*". [Citado 15 mayo 2009] [En Línea]. Disponible en: <http://www.inf.udec.cl/~revista/ediciones/edicion12/articulo%2012-4.pdf>
- [18] DAEDALUS (1998): "*Arquitectura de la Web Semántica: Tecnologías Web*". [Citado 20 junio 2009] [En Línea]. Disponible en: <http://www.daedalus.es/tecnologias-web/web-semantica/>
- [19] LEGUÍZAMO, A. (2007): "*Web Semántica: Un acercamiento a la Web Semántica como herramienta para el aprendizaje en línea*". [Citado 12 julio 2009] [En Línea]. Disponible en: <http://noesis.usal.es/Educare/Vanessa.pdf>



- [20] GONZALES, B. (2004): “*UDDI, Servicios Web en plataforma .NET*”. [Citado 14 mayo 2009] [En Línea]. Disponible en: <http://www.desarrolloweb.com/articulos/1589.php>
- [21] WIKIPEDIA. (2009): “*UDDI. Enciclopedia de libre Contenido*”. [Citado 20 junio 2009] [En Línea]. Disponible en: <http://es.wikipedia.org/wiki/UDDI>
- [22] W3C (2008) “*Servicios Web: Guía Breve de Servicios Web*”. [Citado 22 junio 2009] [En Línea]: Disponible en: <http://www.w3c.es/Divulgacion/Guiasbreves/ServiciosWeb>
- [23] DE LA ROSA, J., TRIVIÑO, A., Y ALDANA, J. (2009): “*OWL-S, Servicios Web semánticos*”. [Citado 24 julio 2009] [En Línea]. Disponible en: http://www.mundointernet.es/IMG/pdf/ponencia76_2.pdf
- [24] LLAMAS, C. (2002): “*Sistemas Multiagente: Desarrollando una ontología*”. [Citado 12 diciembre 2009] [En Línea]. Disponible en: <http://www.infor.uva.es/CLLAMAS/MAS/DesOntoSimpl.pdf>
- [25] OASIS (2006): “*Reference Model for Service Oriented Architecture 1.0*” [Citado 20 diciembre 2009] [En Línea]. Disponible en: <http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf>
- [26] WIKIPEDIA. (2009): “*UDDI. Enciclopedia de libre Contenido*”. [Citado 14 noviembre 2009] [En Línea]. Disponible en: <http://es.wikipedia.org/wiki/RSS>
- [27] NATALYA, F. Y DEBORAH, L. (2005). “*Desarrollo de ontologías. Guía para crear la primera ontología*”. [Citado 12 diciembre 2009] [En línea] Disponible en: http://protege.stanford.edu/publications/ontology_development/ontology101-es.pdf