

Un sistema semántico para la gestión de información de egresados

A system for semantic information management of graduated

Ingrid Durley Torres Ing., Jaime A. Guzmán Luna, Ph. D. & Jovani A. Jiménez Builes Ph. D.

Grupo SINTELWEB, Universidad Nacional de Colombia, sede Medellín, Colombia

{idtorresp, jaguzman, jajimen1}@unal.edu.co

Recibido para revisión 04 de junio de 2010, aceptado 28 de octubre de 2011, versión final 28 de junio de 2011

Resumen — Este artículo propone la introducción de las tecnologías emergentes basadas en las tecnologías de la web semántica, combinadas con sistemas Multiagentes, técnicas de recuperación información y procesos de razonamiento, con el fin de automatizar y facilitar, los procesos de identificación, extracción, almacenamiento y recuperación de información del personal egresado asociado a una institución educativa, orientada a obtener su acreditación. El sistema Implementa para ello un sistema compuesto por cuatro elementos: un modulo de conocimiento y tres agentes (Crawler, Wrapper, Semántico e Interfaz).

Palabras claves — Multiagentes, información de egresados, almacenamiento, recuperación de información, web semántica

Abstract — This article proposes the introduction of emerging technologies based on semantics, combined with multi-agent systems, information retrieval techniques and reasoning processes, to automate and facilitate the processes of identification, extraction, storage and retrieval of personal graduate information associated with an educational institution, oriented to obtain accreditation. Thereby implementing a system composed of four elements: a knowledge module and three agents (Crawler, Wrapper, Semantics and Interface).

Key word — multi-agent, graduate information, storage, information retrieval, semantic web.

I. INTRODUCTION

La Acreditación es un testimonio que da el Estado sobre la calidad de un programa o institución con base en un proceso previo de evaluación en el cual intervienen la institución, las comunidades académicas y el Consejo Nacional de Acreditación [1]. Bajo esta perspectiva, la universidad Nacional, cuenta con varios sistemas de información que apoyan y complementan la constante búsqueda en el mejoramiento de

sus requisitos de calidad educativa. Entre los que se registra el sistema de gestión de información de egresados, el cual permite a la universidad hacer seguimiento continuo a sus graduados con el fin de establecer lazos de comunicación permanente. Actualmente, este sistema es operado de manera manual, por parte del personal administrativo de la universidad.

En un intento por trasladar la información y los sistemas tecnológicos actualmente asociados a los procesos de acreditación de la Universidad Nacional sede Medellín, a sistemas automáticos que proveen una más fácil administración de la información; el presente trabajo propone el diseño e implementación de un sistema semántico de recuperación de información de egresados, denominado SEGIR (por sus siglas en inglés –*SEmantic Graduated Information Retrieval*-), basado en una ontología OWL [2], una arquitectura multiagente [3, 4], el lenguaje de consulta SPARQL [5, 6] y el razonador Pellet[7,8].

Con el objetivo de dar una visión más detallada del sistema, el documento se organiza como sigue: en la siguiente sección se describe la arquitectura SEGIR, abordado de manera independiente cada uno de los módulos componentes: La ontología OGIR (por sus siglas en inglés –*Ontology Graduated Information Retrieval*-), el Agente Crawler, el Agente Wrapper, el Agente Semántico y finalmente el Agente Interfaz. El siguiente aparte recopila el proceso de comunicación entre los agentes, del cual ha sido definido a través de otra ontología y finalmente, en la sección IV se presentan las conclusiones y trabajo futuro asociados con el presente desarrollo.

II. ARQUITECTURA SEGIR

En esta sección se presenta una breve descripción de la arquitectura del sistema multiagente semántico propuesto *SEGIR* (Ver Figura 1):

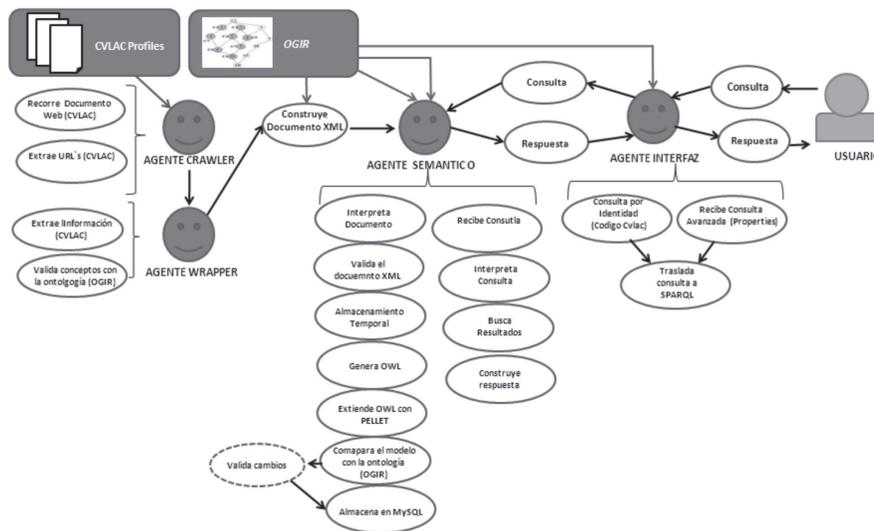


Figura 1. Arquitectura SEGIR

(i) En primera instancia, se construye una base de conocimiento, representada en una ontología abstracta (sin instancias), denominada *OGIR*. Esta ontología constituye el vocabulario necesario para especificar el dominio de conocimiento de los egresados; el cual compartirán cada uno de los agentes del sistema *SEGIR*; (ii) El segundo elemento, se encuentra representado por un conjunto heterogéneo de información distribuida en las diferentes páginas web de Colciencias (*CVLac Profile*) [9], las cuales son identificadas, recorridas y recuperadas a partir de la intermediación constante de uno de los agentes *SEGIR* (*Crawler*) [10]. (iii) el tercer elemento es precisamente, el *Agente Crawler*, el cual es el encargado de identificar y recorrer las URL's, relacionadas con las páginas CVLAC, de un egresado. (iv) En este nivel, se encuentra el *Agente Wrapper* [11], el cual se ocupa de extraer la información de cada CVLAC, para ello debe referirse únicamente a la ontología *OGIR*. De esta manera logra instanciar la ontología, con la información específicamente encontrada por el agente para cada egresado. Esta información, es registrada finalmente, en un documento XML [12], que será transferido al siguiente agente. (v) el quinto elemento, es el *Agente Semántico*, el cual se ocupará de retomar cada documento XML para interpretarlo y recolectarlo en una estructura de almacenamiento persistente modelada en *MYSQL* [13] y en *OWL* [2]; además de trabajar con la información concreta, este permitirá extender la ontología con nuevos elementos, producto de un proceso de inferencia (basado en las relaciones no explícitas), a cargo del razonador *Pellet*. Lo anterior permite extrapolar las consultas y precisar e identificar los resultados de la misma con un mayor grado de precisión. (vi) Finalmente, aparece el *Agente Interfaz*, cuya única finalidad, se encuentra representada en su intermediación entre un usuario humano y los demás agentes del sistema.

En los apartes siguientes se abordará cada elemento de la arquitectura con una mayor profundidad en los detalles, a fin de dar una mejor claridad en los procesos.

A. Ontología de Egresados (*OGIR*)

El diseño e implementación de la ontología “*OGIR*”, permitirá generar en *OWL*, un vocabulario unificado incluyendo las relaciones existentes entre cada uno de los conceptos que especifican el dominio de información relacionado con los egresados de una institución académica universitaria.

La forma de especificar conceptos, estuvo directamente relacionada con el conocimiento propio de la producción investigativa de los egresados en general. Identificando, en esta línea elementos como: (i) Datos personales comunes en la identificación de cada egresado (*Personal Data*), (ii) Programa curricular (en este caso facultad, carrera y sede de la universidad Nacional) al que se asocia cada egresado (*Curriculum Program*), (iii) premios obtenidos en su trayectoria (*Award*), proyectos con los cuales ha estado vinculado el egresado (*Project*), las publicaciones asociadas a la productividad (*Publication*), entre otros. Adicional a los conceptos mencionados, también se incorporaron otros conceptos (conjunto de clases y propiedades), originadas desde dos estándares: (i) *FOAF* [14], el cual constituyó uno de los estándares para la representación semántica de la información personal de un individuo y (ii) *Dublin-Core* [15], el cual corresponde a un conjunto de metadatos usados para describir los recursos de un sistema de información.

En el orden de citación, del primero se extrajeron conceptos como *Person*, usados para representar a un egresado como una persona y *Company*, que representaría toda la información (como atributos) de una compañía con la cual un egresado, pudo establecer una relación laboral. Del segundo, en cambio, se extrajeron propiedades como *Title* y *Date*, ambas asociadas a las publicaciones y/o proyectos. Las propiedades en general, tanto las generadas, como las importadas de ambos estándares (*FOAF* y *Dublin-Core*), buscan asignar las características propias a cada concepto, en la fase de instanciación, la cual

consiste en dar valores concretos a cada elemento abstracto de la ontología, ya sea este un concepto o un atributo; Lo cual hace posible asociar, a cada egresado con su respectiva producción académica, investigativa y demás datos personales.

Como valor agregado al conocimiento explícito representado por las estructuras expuestas, se añadieron un conjunto de características que permiten generar inferencias adicionales sobre la información, usando para ello elementos propios del OWL (tales como *owl:equivalentClass*, *owl:inverseOf*, entre otras), sobre los cuales es posible realizar procesos de inferencias, que permitan a partir de axiomas generar nuevas propiedades o conceptos, sin que estos hayan sido previamente descritos. Por citar un ejemplo es posible establecer la relación de equivalencia entre *curriculum program* y *academic program*.

En la Figura 2. es presentada una porción de la jerarquía de conceptos propios de la ontología OGIR.

```

xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns="http://xue.unalmed.edu.co/~jaguzman/SRIW/recuperacion/OGIR/graduates2010.rdf#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:OGIR="http://xue.unalmed.edu.co/~jaguzman/SRIW/recuperacion/OGIR/rdfs/OGIRSchema#"
xml:base="http://xue.unalmed.edu.co/~jaguzman/SRIW/recuperacion/OGIR/graduates2010.rdf#"
<owl:Ontology rdf:about="">
  <owl:imports rdf:resource="http://xue.unalmed.edu.co/~jaguzman/SRIW/recuperacion/OGIR/rdfs/OGIRSchema#" />
</owl:Ontology>

<rdf:Description rdf:about="#graduate01">
  <OGIR:has_identity>123456789</OGIR:has_identity>
  <OGIR:has_dategraduate>2000-11-31</OGIR:has_dategraduate>
  <OGIR:has_PersonalData>
    <rdf:Description rdf:about="#personaldata01">
    </OGIR:has_PersonalData>
  <OGIR:isAssociatedCompany>
    <rdf:Description rdf:about="#Company01">
    </OGIR:isAssociatedCompany>
  <OGIR:isAssociatedCurriculumProgram>
    <rdf:Description rdf:about="#CurriculumProgram01">
    </OGIR:isAssociatedCurriculumProgram>
  </OGIR:has_PersonalData>
  </OGIR:isAssociatedCompany>
  </OGIR:isAssociatedCurriculumProgram>
  </OGIR:has_dategraduate>
  </OGIR:has_identity>
</rdf:Description>

```

Figura 2. Porción de la ontología OGIR

La ontología OGIR es entonces construida como un medio para unificar el conocimiento y desambiguar la diversidad sintáctica y semántica de los términos usados en cualquier especificación de información de egresados; es decir, cuando uno de los agentes del sistema quiera referirse a la compañía o empresa en la que un egresado trabaja o ha trabajado lo hará mediante el concepto “company”.

B. Páginas Web (Cvlac)

Actualmente, existe mucha información asociada al personal egresado especialmente registrada en páginas web; algunas de las cuales se relacionan con empresas, grupos sociales o con su perfil profesional. El objetivo entonces es que *SEGIR*, obtenga y actualice la información de manera automática y constante, de

algunas de esas páginas. Para ello, se ha optado por identificar aquellas URL’s asociadas a los perfiles de COLCIENCIAS (comúnmente conocidas como Cvlac). La razón de tal selección, se debe a tres principales motivos: (i) la seriedad de la información registrada, (ii) el tipo de datos consignado, los cuales relacionan la hoja de vida de los asociados, principalmente clasificados como investigadores o personal vinculado a la academia. La información registrada incluye temas personales como nivel de estudio, empresas con las que ha estado vinculado, cargos, publicaciones, distinciones, entre otras. (iii) Cada Cvlac, asocia a su vez otras URL que apuntan a otros Cvlac’s de otros potenciales egresados de la misma Universidad, lo que facilita el proceso de automatización.

C. Agente Crawler

El *Agente Crawler*, actúa como un gestor para el recorrido y la recuperación de las URL’s. La funcionalidad es la siguiente: el agente recibe al correr el sistema por primera vez, una URL “base” o “semilla”, asociada a un Cvlac de un egresado en particular; este agente, recorre entonces toda esa página (Cvlac) y en su recorrido identifica y recupera otras URL’s (también Cvlac’s) asociadas a otros egresados. Lo anterior, permite acceder las páginas web enlazadas con la URL “base”, obteniendo y recuperando otras URL que concuerden con el dominio de egresados definido. Este proceso se repite indeterminadamente, hasta que el sistema reciba una orden directa del administrador o hasta que la aplicación sea desinstalada. Si el sistema, repite algunas URL, este de igual manera comunica al *Agente Wrapper* esta información, a fin de que sea construido un nuevo documento XML sin importar que ya haya sido previamente almacenada. Esto debido a que en el lapso de tiempo entre uno y otro recorrido del agente *Wrapper* pueden existir actualizaciones a la información registrada, razón por la cual una URL no puede ser desechada.

D. Agente Wrapper

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<siep:Graduated>
  <foaf:firstName>SANTOS MARTINEZ, ADRIANA</foaf:firstName>
  <dc:coverage>Colombiana</dc:coverage>
  <siep:studies>
    <siep:study>
      <siep:type>Doctorado</siep:type>
      <siep:institution>Universidad de Guadalajara</siep:institution>
      <dc:title>Doctorado en Ciencias</dc:title>
      <dc:date>Febrero de 2008 - de </dc:date>
    </siep:study>
    <siep:study>
      <siep:type>Maestria/Magister</siep:type>
      <siep:institution>Universidad Nacional De Colombia - Sede Bogotá</siep:institution>
      <dc:title>Maestria en Biología Marina</dc:title>
      <dc:date>de 1986 - de 1989</dc:date>
    </siep:study>
    <siep:study>
      <siep:type>Pregrado/Universitario</siep:type>
      <siep:institution>Universidad Pedagógica Nacional - U.P.N.</siep:institution>
      <dc:title>Licenciatura En Biología</dc:title>
      <dc:date>de 1981 - de 1985</dc:date>
    </siep:study>
  </siep:studies>
  <siep:experience>
    <siep:experience>
      <siep:company>Universidad Nacional De Colombia - Sede San Andrés</siep:company>
      <dc:date>Mayo de 1997 Diciembre de 2007</dc:date>
    </siep:experience>
    <siep:experience>
      <siep:company>Universidad Del Magdalena - Unimagdalena</siep:company>

```

Figura 3. Porción del documento XML con datos de un egresado

Este agente es el encargado de extraer toda la información, de las URL's identificadas por el *Agente Crawler* (una a la vez). Sin embargo, la información extraída, debe estar orientada a instanciar la ontología *OGIR*, a partir de la información concreta registrada en el Cvlac del egresado. Por último, construye un documento XML, que corresponde a toda la información de un egresado particular. En Figura 3, se muestra una porción del documento XML construido a partir de la ontología *OGIR* instanciada con los datos del egresado "Adriana Santos Martínez".

E. Agente Semántico

El documento XML construido por el *Wrapper*, es ahora recibido por el *Agente Semántico*. A fin de permitir la interpretación coherente e inequívoca de los mensajes intercambiados entre los agentes, el Agente Semántico, también accesa a la ontología *OGIR*, con lo cual puede validar semánticamente el contenido de un mensaje, interpretarlo, localizar y recuperar información específica contenida en él.

Este agente, posee un comportamiento reactivo, es decir, permanece en un estado de espera de mensajes externos de solicitud para realizar sus tareas particulares, estas tareas, se encuentran divididas en dos grupos: i) tareas de consulta y ii) tareas de almacenamiento. El primer grupo de tareas, está orientado a la ejecución de consultas SPARQL, el cual corresponde a un lenguaje consultivo para RDF [17] que actúa paralelo al estándar SQL de base de datos. El agente, entonces debe realizar el razonamiento semántico al interior de la base de conocimiento a la vez que se debe ocupar de identificar y retornar los resultados para la respectiva consulta. El segundo grupo de tareas tienen como propósito el almacenamiento y actualización de información del egresado dentro de la base de conocimiento.

Para llevar a cabo las tareas planteadas en el diseño del *Agente Semántico*, fueron considerados tres módulos básicos, los cuales permiten la realización de las tareas de consulta y almacenamiento de manera concurrente; los módulos son: i) El módulo constructor de ontología: el cual actúa cuando el agente recibe una solicitud de almacenamiento, interpreta la información XML recibida en el documento remitido por el *Agente Crawler*, para ello utiliza el componente traductor el cual a través de la lectura de algunas reglas de traducción programadas en permite llevar tal información desde el vocabulario de origen, a la especificación particular de la ontología de egresados, esto le permite al agente ser de propósito general, así, a través de la respectiva configuración el agente puede almacenar información de los egresados a partir de diferentes fuentes (agentes recuperadores, buscadores, entre otros). ii) módulo gestor de la base de conocimiento: Una vez construida la instancia de la ontología del egresado, el módulo gestor de base de conocimiento almacena de manera persistente dicha ontología, dentro de una base de datos, indexando tal ontología con el número de identificación del

egresado (este corresponderá al número de identificación de la URL asociado al Cvlac de tal egresado). iii) módulo motor de consulta: para el caso en el cual el mensaje recibido este orientado a una solicitud de consulta, es recuperada la instancia de la ontología correspondiente al identificador del egresado asociado a la consulta realizada, esta instancia, previa ejecución de la consulta SPARQL es procesada por el componente de inferencia, el cual a través de algunas reglas programadas con JENA[16], permite identificar las propiedades OWL, a fin de generar una ontología extendida con conocimiento adicional inferido, la implementación de este proceso de razonamiento, es realizado a través del razonador PELLET. Tal ontología extendida, es posteriormente, al componente ejecutor el cual toma la consulta realizada y la aplica sobre ella (la ontología). Finalmente, el componente intérprete de resultados, toma los resultados obtenidos a partir de la consulta y con ellos construye un documento RDF [17], el cual está conformado por los recursos y propiedades inmersos dentro del conjunto de resultados, de la consulta. Dicho documento, es remitido al agente interfaz, a fin de que esté presente apropiadamente los resultados al usuario.

F. Agente Interfaz

El agente interfaz se encarga de la comunicación directa con el usuario. Este agente opera de la siguiente manera: (i) recibe una consulta formulada en función de algunos datos específicos del egresado (conocidos como *properties* dentro de la ontología *OGIR*) tales datos pueden corresponder a: identificación (código de indexación), Primer nombre, apellidos, fecha de graduación, etc. La formulación de estas consultas son realizadas por el usuario a través de la selección de alguna de las dos opciones: búsqueda por un perfil particular (código de indexación) o búsqueda por un filtro. La primera opción será ejecutada cuando el usuario tenga conocimiento del índice de indexación de los datos de un egresado en particular, en cuyo caso el sistema retornara toda la información almacenada de dicho usuario. La segunda opción, en cambio, da libertad al usuario de seleccionar de un conjunto de datos (todos ellos representados como "*Properties*" dentro de cada ontología), aquellos de su interés; bajo esta modalidad el usuario podrá indagar por todos los egresados graduados en una fecha específica, o de una carrera o sede en particular, o quizás preguntar por un egresado específico (digitando el nombre y el apellido), entre muchas otras. Es importante sin embargo aclarar, que ambos tipos de consultas son guiados por el sistema, a fin de que el usuario únicamente ingrese el valor específico a consultar. Esto contribuye a dar una visión más clara al usuario respecto a lo que él puede consultar; a la vez que facilita, el siguiente paso, el cual consiste en realizar la traducción de la consulta a SPARQL, siendo este paso ejecutado siempre por el agente interfaz, a fin de que la consulta pueda ser trasladada al agente semántico, quien finalmente retornara los resultados correspondientes. Una vez obtenidos los resultados de la consulta, el agente interfaz, se encargará de recibirlos y mostrarlos organizadamente al usuario.

En la Figura 4, el usuario ha decidido indagar por aquel egresado cuyo primer nombre (en *Properties: firstName*) es Jaime Alberto y su apellido es Guzmán Luna (en *Properties: lastName*) para luego ser trasladadas por el agente interfaz a SPARQL. Con el fin de trasladar tal consulta al Agente Semántico, quien finalmente retornara los resultados (según definición funcional definida en el apartado anterior).

Figura 4. Interfaz de formulación de consulta de un egresado, a través de la opción de filtro (nombre y apellido)

II. MODELO DE COMUNICACIONES

El uso de un vocabulario común (ontología), permite la interpretación coherente e inequívoca de los mensajes intercambiados entre los agentes que comparten tal vocabulario; de este modo, los agentes pueden validar semánticamente un mensaje, interpretarlo, localizar y recuperar información específica contenida en él.

En este orden de ideas, se ha implementado, una ontología que considerando el proceso funcional de cada agente, logra recopilar el conjunto de conceptos asociados a cada tarea. Bajo esta perspectiva, es posible considerar los siguientes tipos de mensaje: i) Documento XML representa el documento, como tal construido por el Agente Wrapper, el predicado para identificar tal elemento es *Document*; ii) Para hacer referencia al contenido de dicho documento, los agentes lo interpretaran a través del predicado *Content*; iii) consulta: representado por el predicado *Query*, el cual a través de la expresión SPARQL asociada, representa la consulta a realizar, dentro de la base de conocimiento. iv) resultado: el cual será interpretado a través del predicado *ResultSet* y funcionalmente permite enviar la ontología respuesta a una consulta realizada. Finalmente v) almacenamiento: identificado a través del predicado *Store*, representa una solicitud almacenamiento de la ontología de egresado dentro de la base de conocimiento. Estos mensajes en consecuencia son asociados a los comportamientos propios del agente los cuales son detallados en las secciones anteriores.

La Figura 5, presenta apartes de la implementación de la ontología de comunicación entre agentes.

```

@author SINTELWEB
*/
public class SEGIROntology extends Ontology {
// Ontology Name
public static final String ONTOLOGY_NAME = "SEGIROntology+";
//Vocabulary of the ontology that will handle agents
public static final String DOCUMENT = "Document";
public static final String DOCUMENT_Content = "Content";

public static final String QUERY = "Query";
public static final String QUERY_EXP = "QueryExpression";
public static final String QUERY_ID = "QueryId";

public static final String RESULTSET = "ResultSet";
public static final String QUERY_SLN = "QuerySolution";

public static final String GRADUATE_PROF = "GraduateProfile";
public static final String GRADUATE_DOC = "document";

public static final String STORE = "Store";
public static final String STORE_DOCUMENT = "document";

public static final String REQUEST = "Request";
public static final String REQ_QUERY = "query";

public static final String RESPONSE = "Response";
public static final String RES_QUERY = "resultset";

// Instancia de la ontología (que será instanciada)
private static Ontology instancia = new SEGIROntology();

```

Figura 5. Apartes de la implementación de la ontología de comunicación entre agentes

III. CONCLUSIONES

El modelo propuesto aunque tiene como propósito específico, el proceso de acreditación de universidad Nacional de Colombia sede Medellín, presenta varias ventajas competitivas: (i) la ontología desarrollada representa de manera genérica el conocimiento asociado a cualquier tipo de egresado, razón por la cual se considera viable su reutilización en otros sistemas semánticos y otras instituciones educativas que requieran gestionar la información de sus propios egresados. (ii) de igual manera, el modelo multiagente propuesto, está orientado a tratar los procesos de identificación, extracción, gestión semántica y presentación de cualquier tipo de información asociada a un documento XML, basado en la ontología previamente creada. Lo anterior, gracias a la especificación genérica de una ontología de comunicación propia a las tareas de negociación entre agentes. (iv) El sistema puede ser extendido para incluir en su búsqueda otras páginas web (*facebook*), además de las ya consideradas (*cvlac*). Esta posibilidad puede ser lograda únicamente a través de la reconfiguración de uno solo de los agentes del sistema; (iv) SEGIR, es compatible con las tecnologías Web, razón por la cual puede ser publicado y accesado en ese ambiente web.

AGRADECIMIENTOS

Este trabajo presenta algunos resultados preliminares asociados al proyecto de tesis de maestría titulado “Modelo de Planificación Reactiva para la Composición Dinámica de Rutas de Aprendizaje en Ambientes Virtuales Inciertos y Heterogéneos”.

REFERENCIAS

- [1] Consejo Nacional de Acreditación (CNA), Lineamientos para la acreditación institucional. ISSN:0122-7874. Disponible en: http://www.cna.gov.co/1741/articles-186359_lineamientos_3.pdf. Consultado Octubre 2011. pp. 59-71. Noviembre de 2006.
- [2] OWL, Web Ontology Language (OWL). W3C Recommendation 10 Febrero 2004. Disponible en: <http://www.w3.org/2004/OWL/>. Consultado febrero 2011.
- [3] Bellifemine F. Y. and Willmott S., Developing multiagent systems with JADE. Pp. 89-106. 2003.
- [4] Wooldrige , M., An Introduction to MutiAgent Systems. Second Edition. ISBN:978-0-470-51946-2. Torkuay U.K. 2009.
- [5] SPARQL., Disponible en: <http://www.w3.org/2001/sw/wiki/SPARQL>. Consultado enero 2011.
- [6] Dodds, L., Introducing SPARQL: Querying the Semantic Web. November 16, pp. 167-194. 2005. Disponible en: <http://www.xml.com/pub/a/2005/11/16/introducing-sparql-querying-semantic-web-tutorial.html>.
- [7] Parsia, B. and Sirirn, E., Pellet: An OWL DL Reasoner. In Proceedings of the International Workshop on Description Logics. 2004.
- [8] Pellet, Disponible en: <http://semanticweb.org/wiki/Pellet>. Consultado en febrero 2011.
- [9] SCIENTI, Disponible en: <http://www.colciencias.gov.co/scienti/>. Consultado febrero 2011
- [10] De Maele, F. V., Spyns, P. and Meersman, R.. An Ontology-Based Crawler for the Semantic Web. OTM 2008 Workshop. LNCS. 5333/2008.Spring Verlang. Berlin, Heidelberg, pp. 48-55. 2008.
- [11] L. Liu, Pu, C. and Wei, H., XWRAP: An XML-enabled Wrapper construction System for Web Information Sources. pp. 13-22. 2000.
- [12] XML: Disponible en: <http://www.xmlnews.org>. Consultado Enero de 2011.
- [13] MySQL 5.0 Reference Manual. Copyright © 1997, 2011, Oracle and/or its affiliates. Disponible en: <http://dev.mysql.com/doc/refman/5.0/es/index.html>. Consultado en: Octubre de 2011.
- [14] FOAF. The Friend of a Friend (FOAF) project. Disponible en : <http://www.foaf-project.org/docs>. Consultado Febrero 2011.
- [15] Dublin-Core.Disponible en. <http://es.dublincore.org/>. Consultado Febrero 2011.
- [16] JENA. <http://jena.sourceforge.net/>. 2006. Consultado febrero 2011
- [17] RDF. "RDF Semantics". W3C Recommendation 10 February 2004. Disponible en: <http://www.w3.org/TR/rdf-nt/>. Consultado enero 2011