Uso de Tecnologías Semánticas para la Integración de Recursos de Información en una Memoria Corporativa

Erik Alarcón-Zamora
Departamento de Ingeniería Eléctrica
Universidad Autónoma Metropolitana
Iztapalapa, México
Email: cbi2113802469@xanum.uam.mx

R. Carolina Medina-Ramírez
Departamento de Ingeniería Eléctrica
Universidad Autónoma Metropolitana
Iztapalapa, México
Email: cmed@xanum.uam.mx

Héctor Pérez-Urbina
Clark & Parsia, LLC
Washington, USA
Email: hector@clarkparsia.com

***Resumen*—En este artículo se presenta un marco de trabajo apoyado en las tecnologías semánticas para la integración de recursos de una memoria corporativa. Se describe la metodología adoptada para la integración de recursos, así como un prototipo que muestra la viabilidad del enfoque semántico.**

I. Introducción

Una organización tiene una amplia variedad de recursos de información, por ejemplo, personas, bases de datos, documentos, reportes, presentaciones, vídeos, entre otros. Estos recursos representan el conocimiento de la organización: productos, investigaciones, procesos de producción, soluciones operacionales, flujos de trabajo, objetivos, metas, entre otros. Este conocimiento se denomina memoria corporativa (MC) y se define como "*la representación explícita, consistente y persistente del conocimiento en una organización*" [1]. Una memoria es importante para las personas adscritas o interesadas en la organización, porque ésta les permite acceder, compartir, intercambiar y reutilizar el conocimiento. Dada la importancia de una memoria, es necesario una gestión del conocimiento de la misma, para tener las siguientes ventajas: un personal mejor informado, mayor comunicación en la organización, una herramienta para el aprendizaje, una base de conocimiento persistente, un instrumento para búsqueda, recuperación e intercambio de conocimiento, por mencionar algunas.

Las tecnologías semánticas [2] son *un conjunto de metodologías, lenguajes, aplicaciones, herramientas y estándares para suministrar u obtener el significado de la información*[1]. Éstas agregan una *capa de abstracción* en las fuentes de información, para que los procesos automáticos puedan acceder, procesar, razonar, combinar, reutilizar y compartir la información. Los beneficios de emplear estas tecnologías son: captar la visión de contextos particulares, adaptarse a la naturaleza cambiante del conocimiento, considerar la naturaleza distribuida del conocimiento y de los usuarios, integrar desde distintas fuentes de información, modelar la información en un formato estándar, utilizar un modelo de datos flexible, eliminar ambigüedades en el modelo, inferir sobre el conocimiento, desarrollar aplicaciones genéricas, implementar a menor costo y mayor interacción de los expertos en el dominio.

Las tecnologías semánticas permiten representar y gestionar el conocimiento en una memoria corporativa. En particular, permiten hacer el proceso de integración (búsqueda y recuperación) de información significativa de los recursos en una memoria corporativa. Para lograr esta integración, se deben efectuar las siguientes actividades: *1) modelar el conocimiento de los recursos en un formato estándar, 2) explotar el conocimiento implícito de los recursos y describir el vocabulario (conceptos y relaciones) de la memoria*, y *3) buscar y recuperar la información sobre los recursos, para responder una pregunta dada.*

Una memoria corporativa tiene múltiples recursos de información y para limitar éstos a un conjunto manejable, se hace un análisis para detectar los casos de uso prioritarios. Este artículo describe dos casos de uso básicos que se pueden emplear en cualquier memoria corporativa:

1. ***Cartografía de Competencias***: consiste en la búsqueda y recuperación de información significativa de las personas, a partir de las características personales y profesionales de las mismas. Algunas de estas características profesionales son: *competencias profesionales* (capacidad de trabajar en equipo, habilidad de liderazgo, por mencionar algunas), *habilidades lingüísticas* (leer en inglés, escribir en español, hablar en francés), *conocimientos en los temas del dominio de la memoria corporativa* (sistemas operativos, radios cognitivos, por mencionar algunos.), entre otras.
2. ***Búsqueda de Recursos Digitales***: consiste en la búsqueda y recuperación de información significativa de los documentos y archivos multimedia a partir del contenido de los mismos. Algunos de los parámetros de búsqueda de éstos son: *el autor, la extensión* (pdf, doc, wav, etc.), *los temas que trata el recurso* (sistemas operativos, ontologías, radios cognitivos, por mencionar algunos), entre otros.

[1]Lee Feigenbaum Bio, "Semantic Web vs. Semantic Technologies," Available: http://www.cambridgesemantics.com/semantic-university/semantic-web-vs-semantic-technologies

En este artículo, se representa e integra la memoria corporativa del *área de Redes y Telecomunicaciones (RyT) de la Universidad Autónoma Metropolitana (UAM)*. Los recursos de esta memoria representan las investigaciones, colaboraciones, proyectos, cursos y temas de interés de los profesores-investigadores del área RyT.

Este artículo se organiza de la siguiente manera: la sección II presenta nuestra metodología para la integración semántica de recursos en una memoria corporativa. Esta sección se divide en seis subsecciones. La subsección A describe de manera general la metodología y las tres etapas generales de la misma (representación, explotación y consulta del conocimiento de los recursos). La subsección B muestra nuestra arquitectura. La subsección C describe el marco de descripción de recursos (RDF) para representar (modelar) el conocimiento explícito de los recursos. La subsección D describe los axiomas y la manera de explotar el conocimiento implícito de los recursos. La subsección E explica el lenguaje de consulta, para interrogar el conocimiento en el modelo. La subsección F describe la finalidad del prototipo para la integración semántica. En la sección III se describen la pruebas y resultados (desempeño y calidad de las respuestas) que se hicieron al triplestore Jena[2] y a nuestro modelo. Finalmente, las conclusiones sobre la integración semántica y sobre los resultados de la experimentación se presentan en la sección IV.

II. Integración Semántica de Recursos

La Integración Semántica de Recursos (ISR) *es el proceso de búsqueda y recuperación significativa de información existente en los recursos de información (documentos) que están residentes en algún medio de almacenamiento.* Se basa en el uso de tecnologías semánticas. La finalidad de esta integración es recuperar documentos vinculados que respondan a una pregunta hecha por un usuario. En este artículo, la integración ISR se efectúa en una memoria corporativa (MC), porque esta ISR considera algunas características importantes de una memoria corporativa como: el crecimiento explosivo de recursos, heterogeneidad en formato, contenido y estructura de los recursos, ambigüedades en la información, evolución del conocimiento en los recursos (agregar, eliminar, modificar o renovar), entre otras. Los principales usuarios en la ISR son los expertos y personas que se vinculan al dominio de la MC.

A. Propuesta

Nuestra propuesta es una metodología para desarrollar la *integración semántica de recursos* (ISR) en una *memoria corporativa* (MC). Esta metodología considera dos casos de uso, sin embargo ésta puede extenderse a otros casos de uso. También esta metodología se puede emplear en cualquier MC, por ejemplo, Biomédica, Química, Biología, entre otras, ya que éstas están compuestas por recursos de información que pueden ser integrados, para responder las preguntas de los usuarios. Nuestra metodología contempla tres etapas generales:

1. Representación del conocimiento en los recursos: consiste en modelar el conocimiento explícito de los recursos en un formato estándar.
2. Explotación del conocimiento sobre los recursos: consiste en emplear reglas de inferencia para explotar el conocimiento implícito.
3. Consulta de información sobre los recursos: consiste en interrogar al modelo de conocimiento a partir de una pregunta de un usuario y responder con información sobre los recursos.

B. Arquitectura

La arquitectura que presentamos en la Figura 1, se diseñó con base en el modelo de tres capas (nivel usuario, nivel negocio, nivel de datos):

- En el nivel de usuario, se tiene un conjunto de servlets y páginas web estáticas que proporcionan la interfaz visual para que los usuarios hagan sus preguntas y visualicen las respuestas. Las páginas estáticas son un conjunto de formularios para que el usuario escriba la información que desea buscar. Mientras, los servlets en este nivel son los mecanismos para visualizar la información.
- En el nivel de negocios, un servlet transforma la información del formulario a consultas SPARQL e invoca al triplestore. Este último se encarga de hacer las siguientes actividades: 1) solicitar y cargar la ontología 2) hacer inferencia a partir de la ontología y un razonador, y 3) consultar la información en el modelo a partir del motor de búsqueda y la consulta en SPARQL.
- En el nivel de datos (conocimiento), la ontología es la base de conocimiento sobre los recursos; las descripciones son la representación del conocimiento explícito, mientras los axiomas permiten aprovechar el conocimiento implícito.

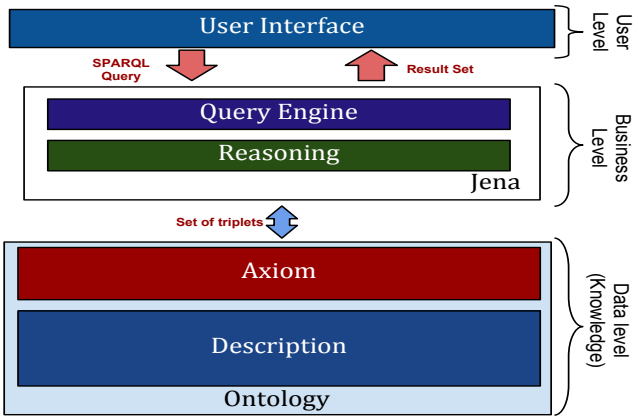

Figura 1. Arquitectura general para la Integración Semántica de Recurso en una Memoria Corporativa.

Las herramientas para desarrollar los componentes de la arquitectura, se describen en las subsecciones C, D, E y F.

[2]The Apache Software Foundation, "Apache Jena," Available: http://jena.apache.org/

C. *Representación del Conocimiento*

Las tecnologías semánticas proponen al marco de trabajo RDF[3] (Resource Description Framework) para la representación del conocimiento de los recursos en un formato estándar [3]. El primer paso en esta representación es establecer un identificador único (URI) para cada uno de los recursos en la memoria corporativa, con la finalidad que no exista ambigüedad entre éstos. Por ejemplo, al recurso "*Hacia una búsqueda semántica*" se le asigna la siguiente URI: *http://arte.izt.uam.mx/ontologies/digiResourceRyT.owl#Hacia_busqueda_semantica*

El siguiente paso es representar a los recursos mediante sus propiedades (características básicas o metadatos) y los valores asignados a ellas; esta representación se conoce como descripción de los recursos. Cada propiedad (título, autor y lenguaje fuente) tiene un identificador único (URI) como nombre. En la escritura de este nombre, se escribe un verbo en tercera persona y tiempo presente ('tiene', 'es', 'conoce', entre otros) antes del metadato. Por ejemplo, la propiedad *título* tiene la siguiente URI: *http://arte.izt.uam.mx/ontologies/digiResourceRyT.owl#tiene-titulo*.

Para reducir el tamaño del identificador (URI) de los recursos y las propiedades, se sustituye la secuencia de caracteres desde "*http://www*" hasta el símbolo "*#*" por un prefijo. En nuestro caso de estudio, el prefijo *sirp* se traduce en *http://arte.izt.uam.mx/ontologies/digiResourceRyT.owl#*. Por ejemplo, la propiedad *título* se escribe *sirp:tiene-titulo*. Existen otros prefijos preestablecidos para otros vocabularios, por ejemplo: *rdf, xsd, rdfs y owl*.

Las propiedades de los recursos y sus respectivos valores se llevan a un formato estándar, que en este marco se denomina terna (triple) [3]. El conjunto de todas las ternas RDF sobre los recursos de un dominio, se denomina componente asertivo (ABox) [4]. Una terna se forma por un sujeto, un predicado y un objeto; el sujeto es el identificador del recurso que se describe, el predicado es el identificador de la propiedad, y el objeto es o bien una Literal (cadena, entero), o bien el identificador de otro recurso. Un ejemplo de terna se muestra a continuación. Los recursos suelen agruparse en clases. Para decir que cierto recurso pertenece a una clase se utiliza la terna: **sirp:recurso *rdf:type* sirp:Clase**. En la representación del conocimiento, se pueden emplear vocabularios preestablecidos para las características de los recursos, como el dublin core. Sin embargo, este último no proporciona algunos metadatos que se requieren para describir los recursos persona.

Hay diferentes manera de escribir las ternas, desde lo manual escribiendo literales e identificadores en un editor de texto, hasta lo automatizado utilizando herramientas que mapean la información a ternas RDF [5] ejemplos de estas herramientas son: GATE[4], RDF123[5], entre otras. La herramienta que elegimos para administrar las ternas RDF, es el triplestore Apache Jena [6], porque proporciona lectura, procesamiento y escritura de ternas en distintos formatos (XML/RDF, N-triples, Turtle), además este triplestore se puede utilizar con el lenguaje de programación *Java*. Un triplestore es un sistema que tiene dos funcionalidades básicas: el almacenamiento y acceso de las ternas RDF. Existen otros triplestore [7], por ejemplo: Stardog[6], Sesame[7], entre otros.

D. *Explotación del Conocimiento*

Hemos descrito cómo el marco RDF permite modelar el conocimiento explícito de los recursos (modelo de datos). Este modelo de datos se enriquece con la introducción de *axiomas* [8], que permiten completar, extender, renovar y adaptar el conocimiento de los recursos. Para identificar estos axiomas, se analizan las clases/propiedades y se hacen preguntas sobre éstas: qué clases/propiedades se pueden agrupar en otra clase/propiedad respectivamente, qué clases y propiedades son sinónimos, qué clases o clase-literal relaciona una propiedad, la propiedad es una relación simétrica, transitiva, reflexiva, sólo por mencionar algunas preguntas. En nuestro caso de estudio, empleamos los siguientes axiomas:

Subclase y Subpropiedad permiten establecer jerarquías entre las clases (Figura 2) y propiedades (Figura 3) respectivamente.

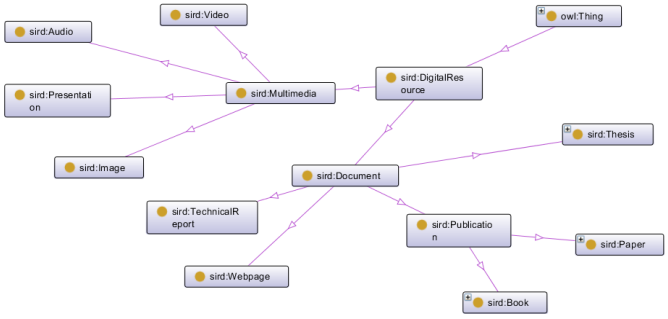

Figura 2. Jerarquía de Clases sobre los recursos digitales de la memoria corporativa (RyT) vistas con protégé.

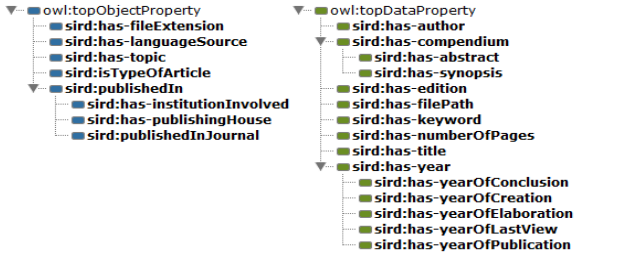

Figura 3. Jerarquía de Propiedades de los recursos digitales en la memoria corporativa (RyT) vistas con protégé.

Dominio y Rango permite establecer qué clases o clase-literal, debe relacionar una propiedad (Figura 4).

Existen otros axiomas, como son Clases equivalentes, Clases disjuntas, Propiedad Simétrica, sólo por mencionar.

[3]W3C, "RDF 1.1 Concepts and Abstract Syntax," Available: http://www.w3.org/TR/rdf11-concepts/

[4]University of Sheffield, "GATE," Available: http://gate.ac.uk/projects.html

[5]Ebiquity, "RDF123," Available: http://ebiquity.umbc.edu/project/html/id/82/RDF123

[6]Clark & Parsia LLC, "Stardog," Available: http://stardog.com/docs/

[7]Aduna, "Sesame," Available: http://www.openrdf.org/about.jsp

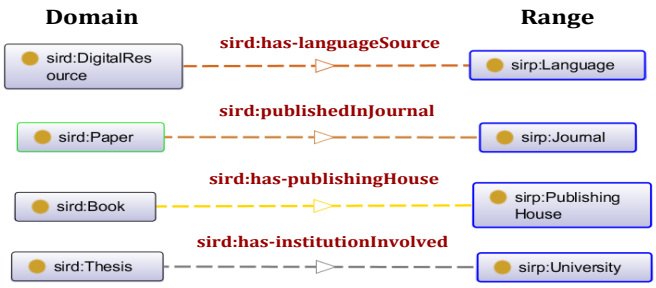

Figura 4. Dominio y Rango de las propiedades de los recursos digitales en la memoria corporativa (RyT) vistos con protégé.

Las funcionalidades y ejemplos de éstos se pueden encontrar en la literatura [4], [9].

Los axiomas se representan en forma de ternas y los lenguajes estándar para escribir axiomas son: RDF Schema[8] (RDF(S)) y el Web Ontology Language[9] (OWL). Existen distintas herramientas para escribir los axiomas en forma de terna con el vocabulario adecuado [10], [11], por ejemplo: protégé[10], TopBraid[11], por mencionar algunas. La herramienta que nosotros elegimos es protégé, porque es una plataforma que proporciona un interfaz agradable al usuario [12], permitiendo a éste la creación, manipulación y visualización de los axiomas. Además esta herramienta permite guardar los axiomas en diferentes formatos (XML/RDF, Manchester, Turtle, OWL/XML).

El conjunto de axiomas que enriquecen el modelo, se denomina componente terminológico (TBox). El modelo de conocimientos conformado por TBox y ABox, se denomina ontología [13]. En esta propuesta, los dos casos de uso son independientes, por lo tanto, se decidió que cada uno de éstos tenga su propia ontología. Un objetivo en común para los casos de uso es vincular los recursos con los temas del dominio de la memoria corporativa. Para ello, se propone una tercera ontología que tiene el vocabulario del dominio. En nuestro caso de estudio, la ontología es el vocabulario de Redes y Telecomunicaciones (RyT) que se desarrolló a partir de otra ontología ODARyT [14]. Nuestra ontología vocabulario (ODARyT4sir) está constituida por 303 que están organizados en cuatro ramas principales: *Distributed Systems, Networking and Telecommunication, Digital Communication Systems* y *Semantic Web*.

Una ontología tiene ternas sobre el conocimiento explícito (descripciones de recursos) y el conocimiento implícito (axiomas). Es posible explotar el conocimiento implícito en una ontología a través de un razonador, que es un programa para inferir hechos o asociaciones a partir del conocimiento existente (axiomas y propiedades) [15]. Por ejemplo, se tiene un ABox con la siguiente descripción: el recurso "*Hacia una búsqueda semántica*" es un *artículo* y un TBox con el siguiente axioma: *la clase Artículo es subclase de la clase Documento*; empleando un razonador con este ABox y TBox, se infiere la siguiente descripción: el recurso "*Hacia una búsqueda semántica*" es un *documento*. Hay distintos razonadores [15], [16], por ejemplo: Pellet[12], Fact++[13], por mencionar algunos. En este artículo, se utilizó el razonador que trae por default Jena, porque éste se puede invocar desde Java+Jena, soporta los axiomas de nuestra ontología (RDF(S) y OWL) y no requiere una configuración o compilación previa para utilizarse. Por otro lado, un razonador es una herramienta para validar el modelo de conocimiento, porque permite encontrar contradicciones o ambigüedades. Nosotros verificamos la consistencia de nuestros modelos, mediante un programa en Java y Jena que verificar si el modelo tiene o no contradicciones. En nuestros modelos el programa no encontró contracciones.

Nuestra ontología de *recursos digitales* modela el conocimiento de 1330 recursos digitales. En este modelo, 691 recursos tienen explícitamente la terna de asignación (rdf:type) a una de las nueve clases básicas: Artículo, Reporte Técnico, Página Web, Tesis, Libro, Audio, Vídeo, Imagen y Presentación. Mientras, los otros 639 recursos por inferencia pertenecen a alguna de las clases básicas. La Figura 5 muestra la ontología vista como diagrama de Venn; los círculos son las clases de los Recursos Digitales y los puntos son los recursos.

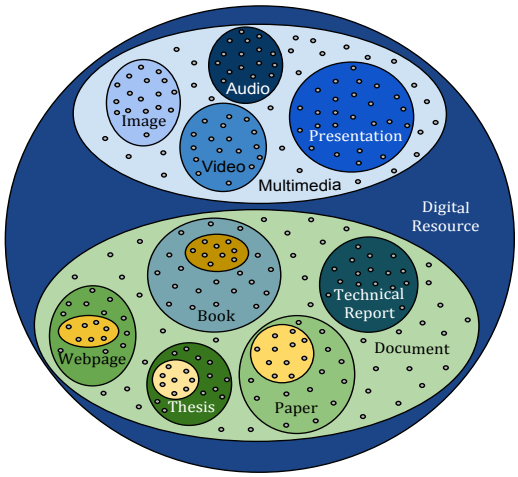

Figura 5. Diagrama de Venn de la ontología de recursos digitales.

E. *Consulta de información*

Para hacer búsqueda y recuperación de la información en una ontología, es necesario tres cosas: *1) una pregunta en lenguaje natural, 2) una consulta basada en triples y 3) un motor de consulta.*

En el primer punto, del análisis de los casos de uso, se identifican y escriben en lenguaje natural las preguntas que los usuarios quieren hacer al modelo. En nuestro caso de estudio, se obtuvieron 10 preguntas (Tabla I) para la búsqueda

[8]W3C, "RDF Vocabulary Description Language 1.0: RDF Schema," Available: http://www.w3.org/TR/2004/REC-rdf-schema-20040210/

[9]W3C, "OWL 2 Web Ontology Language Structural Specification and Functional Style Syntax," Available: http://www.w3.org/TR/owl2-syntax/

[10]Stanford Center, "Protégé," Available: http://protege.stanford.edu/

[11]TopQuadrant, "TopBraid Composer," Available: http://www.topquadrant.com/products/TB_Composer.html

[12]Clark & Parsia LLC, "Pellet: OWL 2 Reasoner for Java," Available: http://clarkparsia.com/pellet/

[13]Dmitry Tsarkov, "Fact++," Available: http://owl.man.ac.uk/factplusplus/

de recursos digitales. Una de estas preguntas es: (1)... ¿Qué documentos sirven para dar un curso de Sistemas P2P?

En el segundo punto, las tecnologías semánticas establecen al lenguaje SPARQL[14] como especificación para consultar, recuperar y modificar la información de ternas RDF. Este lenguaje se basa en patrones de búsqueda, que son comparados con las ternas del modelo. Un patrón de búsqueda es parecido a una terna, pero a diferencia de ésta, el sujeto, la propiedad o el objeto pueden ser una variable. En una consulta SPARQL, hay dos cláusulas: la cláusula SELECT enuncia las *variables resultado* y la cláusula WHERE enuncia los *patrones de comparación*. La consulta SPARQL para la pregunta (1) es:

```
SELECT ?title ?path
WHERE
{?x rdf:type sird:Document;
   sird:has-topic redes:Peer_to_Peer_System;
   sird:has-title ?title;
   sird:has-filePath ?path.}
```

Un *motor de consulta SPARQL* es el programa para responder las consultas de los usuarios. La función básica de éste es: interpreta una consulta SPARQL, compara los patrones con las ternas del modelo, recupera la información asociadas a las variables que están en la cláusula SELECT y regresa la información al usuario. Un motor de consulta se puede encontrar en un triplestore. En particular, Jena posee el motor de consulta (ARQ). Este motor recupera los resultados de una consulta para mostrarlos en pantalla en forma de una tabla o para procesar estos resultados con Java.

F. Prototipo (interfaz de usuario)

La integración semántica de recursos (ISR) utilizando un triplestore, no es una tarea que cualquier usuario puede hacer, ya que éste debe estar familiarizado con el triplestore, el lenguaje de consulta SPARQL y las ternas RDF. Nosotros proponemos una interfaz para la interacción transparente y amigable del usuario con el triplestore; esta interfaz tiene las siguientes características: *1)permitir a los usuarios, navegar a través de la información básica de los recursos, 2)permitir a los usuarios, hacer búsquedas específicas de los recursos, 3)publicar los resultados de la búsqueda y navegación en un formato visual agradable, 4)mapear la pregunta a la respectiva consulta SPARQL, así como 5) invocar al triplestore (carga, inferencia, búsqueda) y proporcionar a éste la consulta SPARQL.*

Nuestro prototipo de interfaz es una aplicación web que trabaja con el triplestore Jena, este prototipo se implementó con Java y proporciona un conjunto de servlets que están en un servidor Tomcat. El prototipo proporciona las siguientes interfaces visuales: navegación a través de las personas, navegación a través de los documentos, navegación a través de los recursos multimedia, búsqueda avanzada de personas, búsqueda avanzada de documentos, búsqueda avanzada de recursos multimedia, búsqueda avanzada sobre cualquier recurso. La Figura 6 muestra la interfaz web de ***navegación a través de las personas***.

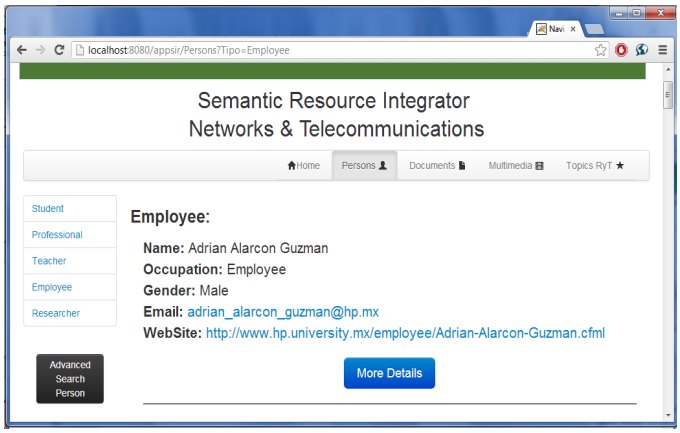

Figura 6. Interfaz visual de navegación a través de personas.

III. EXPERIMENTACIÓN

En la integración semántica de recursos, se tienen dos criterios de evaluación para el triplestore Jena: *1) desempeño y 2) cantidad de resultados para un modelo con inferencia (ontología con razonador) y sin inferencia (descripciones explícitas)*. Para evaluar el desempeño de Jena, queríamos ver el rendimiento de ésta con la cantidad de datos que esperamos manejar. En la evaluación de resultados, queríamos ver si los resultados devueltos por Jena, son los que responden nuestras preguntas.

La evaluación del desempeño, consiste en tomar el tiempo promedio desde que el modelo se carga hasta la recuperación de los resultados de una consulta. Mientras la evaluación de la calidad de resultados, consiste en comparar la información recuperada de una consulta SPARQL con los recursos que se saben responden la pregunta (un análisis previo). Para hacer esto, se hizo un programa en Java+Jena que repite ***n*** veces el cálculo del tiempo de procesamiento para un modelo y una consulta dada, además en cada iteración regresa los valores de la consulta. Los tres parámetros de entrada de este programa son: el número de repeticiones, el tipo de modelo *con datos explícitos o con un razonador y una ontología* y la consulta en SPARQL. Mientras los tres parámetros de salida son: el tiempo promedio de procesamiento de la consulta, el número de recursos que responde el motor de búsqueda y las respuestas de la consulta. Nosotros establecimos los siguientes parámetros para el programa: el número de iteraciones ***n*** se colocó en 20, los modelos son: *1) el ABox de recursos digitales de RyT y 2) la ontología recursos digitales y el vocabulario de RyT, y 3) un conjunto de preguntas básicas, que se enuncian en la Tabla I.* En esta tabla se muestra el número de recursos digitales que responden a cada una de las preguntas.

Este programa se corrió en una computadora con un procesador Intel Core I7 a 2.3GHz con 8Gb en RAM y 8 núcleos de procesamiento. Esta prueba se ejecutó usando Java 1.7, con el entorno de desarrollo integrado Eclipse y

[14]W3C, "SPARQL 1.1 Overview," Available: http://www.w3.org/TR/sparql11-overview/

Tabla I. PREGUNTAS EN LENGUAJE NATURAL Y CANTIDAD DE RECURSOS QUE RESPONDEN A ÉSTAS.

Id. Consulta	Pregunta	No. de Recursos
Q1	¿Cuáles son los títulos, rutas, extensión, idioma de todos los recursos digitales de RyT?	1330
Q2	¿Cuáles libros tratan sobre algunos temas de Sistemas Distribuidos?	103
Q3	¿Qué recursos fueron publicados por la UAM?	18
Q4	¿Qué documentos son para dar un curso de Sistemas P2P?	31
Q5	¿Qué recursos multimedia son mayores al año 2009?	119
Q6	¿Cuáles documentos tratan sobre Ontologías?	30
Q7	¿Qué recursos fueron publicados en una Revista científica?	156
Q8	¿Qué recursos tienen en su contenido las palabras "linked data"?	159
Q9	¿Cuáles documentos en inglés y mayores al año 2000 son de autoría de Erik Alarcón Zamora?	2
Q10	¿Cuáles la tesis de Samuel Hernández Maza?	4

Apache Jena 2.7.4 en Windows 7 (64bits). Nosotros corrimos el programa dos veces para cada pregunta de las lista (Tabla I). En la primera corrida, el modelo es el ABox, mientras en la segunda corrida el modelo es del razonador y la ontología. En nuestro caso de estudio, la ontología de recursos digitales tiene 1330 recursos digitales y las siguientes cantidades de ternas: ABox tiene 20429 y TBox tiene 107. Mientras el vocabulario de RyT (ODARyT4sir) tiene 303 conceptos y en el TBox tiene 1115 ternas. Mediante el proceso de inferencia y combinando ambas ontologías se tiene un total de 38661 ternas. Los resultados de las mediciones del tiempo promedio y la cantidad de respuestas para cada consulta se muestran en la Tabla II.

Tabla II. TIEMPO PROMEDIO DE PROCESAMIENTO Y CANTIDAD DE RECURSOS QUE RESPONDEN UNA CONSULTA.

Id. Consulta	Modelo (ABox)		Modelo (Razonador+Ontología)	
	Tiempo promedio (ms)	No. Recursos	Tiempo promedio (ms)	No. Recursos
Q1	12	1330/1330	138	1330/1330
Q2	10	0/103	194	103/103
Q3	8	18/18	406	18/18
Q4	28	15/31	129	31/31
Q5	7	66/119	157	119/119
Q6	9	15/30	4016	30/30
Q7	12	156/156	3520	156/156
Q8	16	159/159	3472	159/159
Q9	42	0/2	3451	2/2
Q10	13	3/4	3312	4/4

En la Tabla II, se muestra la cantidad de recursos que responden una consulta SPARQL y para verificar que la información de la consulta responde la pregunta, se hizo un análisis de los recursos que responden cada una de las preguntas. Después, se compara manualmente la información recuperada de cada consulta SPARQL con las respuestas de la respectiva pregunta.

El desempeño de Jena es bueno (menor a 1 segundo) cuando se interroga al conocimiento explícito, porque el motor de consulta interroga directamente los ternas del ABox. En contraste, Jena consume más tiempo cuando el modelo es resultado de la inferencia, porque un razonador invierte tiempo en el proceso de inferencia. Si bien Jena no tiene un adecuado desempeño mediante el uso de un razonador, en un futuro podemos probar otras herramientas que procesan mayor cantidad de ternas, disminuyan los tiempos de consulta y permitan el uso de un razonador. Ahora bien, para la evaluación de la calidad de los resultados se tiene lo siguiente: si no se emplea un razonador, varios resultados son omitidos durante el proceso de consulta, ya que algunos recursos no poseen cierta información explícita. Pero, cuando se emplea un razonador, no se pierden respuestas para las consultas, porque éste hace evidente el conocimiento implícito.

IV. CONCLUSIONES

Una memoria corporativa es una fuente de conocimiento para una organización, si este conocimiento se representa adecuadamente, entonces la integración de los recursos tendrá mejores resultados. Nuestra metodología es una solución genérica para la integración semántica de recursos (ISR). Estas tecnologías semánticas nos permitieron: 1) modelar, explotar y consultar al conocimiento sobre los recursos, 2) representar el conocimiento en un formato estándar, 3) utilizar herramientas para desarrollar aplicaciones semánticas, 4) utilizar y compartir varios vocabularios, 5) utilizar aplicaciones para explotar el conocimiento, entre otras ventajas.

En la prueba de desempeño de Jena para el proceso de consulta, el tiempo promedio es menor a un segundo cuando se hace consultas al ABox. Pero, el tiempo se incrementa (1 a 3 segundos) cuando se consulta un modelo por inferencia en una ontología. Aunque el desempeño de Jena es aceptable para el modelo con inferencia, nosotros creemos que mediante operaciones de cómputo paralelo o utilizando otro triplestore, se pueden reducir estos tiempos. En la evaluación de recuperación de la información, si se utiliza un razonador, entonces hay una mayor posibilidad que los resultados de Jena sean los que respondan las preguntas de los usuarios. Por tanto, a pesar de invertir tiempo en la inferencia, se obtienen resultados más satisfactorios para los usuarios.

REFERENCIAS

[1] R. Dieng, O. Corby, A. Giboin, and M. Ribière, "Methods and Tools for Corporate Knowledge Management," INRIA, Tech. Rep. RR-3485, Sep. 1998. [Online]. Available: http://hal.inria.fr/inria-00073203

[2] S. Alfred, A. Arpah, L. H. S. Lim, and K. K. S. Sarinder, "Semantic technology: An efficient approach to monogenean information retrieval," in *Computer and Network Technology (ICCNT), 2010 Second International Conference on*, 2010, pp. 591–594.

[3] S. Bouzid, C. Cauvet, and J. Pinaton, "A survey of semantic web standards to representing knowledge in problem solving situations," in *Information Retrieval Knowledge Management (CAMP), 2012 International Conference on*, 2012, pp. 121–125.

[4] M. Krötzsch, F. Simančík, and I. Horrocks, "A description logic primer," *Computing Research Repository (CoRR)*, vol. abs/1201.4089, 2012. [Online]. Available: http://arxiv.org/abs/1201.4089

[5] V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta, and F. Ciravegna, "Semantic annotation for knowledge management: Requirements and a survey of the state of the art," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 4, no. 1, pp. 14–28, Jan. 2006. [Online]. Available: http://dx.doi.org/10.1016/j.websem.2005.10.002

[6] B. McBride, "Jena: a semantic web toolkit," *Internet Computing, IEEE*, vol. 6, no. 6, pp. 55–59, 2002.

[7] H. Mühleisen, T. Walther, and R. Tolksdorf, "A survey on self-organized semantic storage." *International Journal of Web Information Systems (IJWIS)*, vol. 7, no. 3, pp. 205–222, 2011.

[8] F. Baader, I. Horrocks, and U. Sattler, "Description Logics," in *Handbook of Knowledge Representation*, F. van Harmelen, V. Lifschitz, and B. Porter, Eds. Elsevier, 2008, ch. 3, pp. 135–180.

[9] M. Horridge, H. Knublauch, A. Rector, R. Stevens, and C. Wroe, "A Practical Guide To Building OWL Ontologies Using The Protege-OWL Plugin and CO-ODE Tools Edition 1.0," Aug. 2004. [Online]. Available: http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf

[10] T. Aruna, K. Saranya, and C. Bhandari, "A survey on ontology evaluation tools," in *Process Automation, Control and Computing (PACC), 2011 International Conference on*, 2011, pp. 1–5.

[11] S. C. Buraga, L. Cojocaru, and O. Nichifor, "Survey on Web Ontology Editing Tools," *Buletinul Stiintific al Universitatii Politehnica din Timisoara*, pp. 1–6, 2006.

[12] H. Knublauch, R. W. Fergerson, N. F. Noy, and M. A. Musen, "The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications," in *The Semantic Web - ISWC 2004*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin / Heidelberg, 2004, vol. 3298, ch. 17, pp. 229–243. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-30475-3_17

[13] L. Gandon, Fabien, "Ontology Engineering: a Survey and a Return on Experience," INRIA, Tech. Rep. RR-4396, Mar. 2002. [Online]. Available: http://hal.inria.fr/inria-00072192

[14] A. B. Rios-Alvarado, R. C. M. Ramírez, and R. Marcelín-Jiménez, "A semantic web approach to represent and retrieve information in a corporate memory," in *OWL: Experiences and Directions Workshop (OWLED)*, 2009.

[15] S. Singh and R. Karwayun, "A comparative study of inference engines," in *Proceedings of the 2010 Seventh International Conference on Information Technology: New Generations*, ser. ITNG '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 53–57. [Online]. Available: http://dx.doi.org/10.1109/ITNG.2010.198

[16] R. B. Mishra and S. Kumar, "Semantic web reasoners and languages," *Artif. Intell. Rev.*, vol. 35, no. 4, pp. 339–368, Apr. 2011. [Online]. Available: http://dx.doi.org/10.1007/s10462-010-9197-3