

# 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, personal de la organización, bases de datos, documentos electrónicos, reportes, libros, presentaciones, videos, etc. 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) o memoria organizacional (MO) y se define como "*la representación explícita, consistente y persistente del conocimiento en una organización*" [1]. Una memoria es importante para las personas adscritas (miembros) o personas 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, con la finalidad de tener las siguientes ventajas: un personal mejor informado, una mejor comunicación en la organización, mejores tomas de decisión, una herramienta de apoyo para el aprendizaje, una base de conocimiento persistente y accesible, un instrumento para búsqueda, recuperación e intercambio de conocimiento, por mencionar algunas.

Las tecnologías semánticas (TS) [2] son *un conjunto de metodologías, lenguajes, aplicaciones, herramientas y estándares para suministrar u obtener el significado de la información*<sup>1</sup>. Estas tecnologías agregan una *capa de abstracción* sobre las fuentes de información, para que los procesos automáticos puedan acceder, procesar, razonar, combinar, reutilizar y compartir la información. Los beneficios de emplear las tecnologías semánticas son los siguientes: 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 las personas expertas 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, etc.), *habilidades lingüísticas* (leer en inglés, escribir en español, hablar en francés, etc.), *conocimientos en los temas del dominio de la memoria corporativa* (por ejemplo: sistemas operativos, radios cognitivos, etc.), 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 estos recursos son: *el autor*, *la extensión* (pdf, doc, wav, etc.), *los temas que trata el recurso* (por ejemplo: sistemas operativos, ontologías, radios cognitivos, etc.), entre otros.

En este artículo, se representa e integra la memoria cor-

<sup>1</sup>L. Feigenbaum, "Semantic Web vs. Semantic Technologies," Available: <http://www.cambridgesemantics.com/semantic-university/semantic-web-vs-semantic-technologies>

porativa del área de Redes y Telecomunicaciones (RyT) de la Universidad Autónoma Metropolitana (UAM). Los recursos de esta memoria representan el conocimiento de los profesores-investigadores del área RyT. En concreto, los recursos representan sus investigaciones, colaboraciones, proyectos, cursos y temas de interés.

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 la arquitectura para la integración semántica. 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<sup>2</sup> y al modelo para el área de redes y telecomunicaciones. Finalmente, las conclusiones sobre la integración semántica de los recursos 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 (TS) [3]. 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 usuarios principales en esta 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, la metodología se puede emplear en cualquier MC, por ejemplo, Biomédica, Redes y Telecomunicaciones, Química, Biología, Física, entre otras, ya que éstas están compuestas por recursos de información que pueden ser integrados para responder las preguntas de los usuarios. Para llevar a cabo la ISR, 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 propuesta para la integración semántica es una solución que, por un lado, representa y explota el conocimiento sobre los recursos de una memoria corporativa en un modelo, y por otro lado, permite la integración (búsqueda y recuperación de información) del modelo de conocimiento. Esta arquitectura es genérica y se puede implementar en cualquier memoria corporativa asociada a un dominio. La Figura 1 muestra la arquitectura, en la cual varios de los componentes son desarrollados en alguna de las tres etapas de esta ISR.

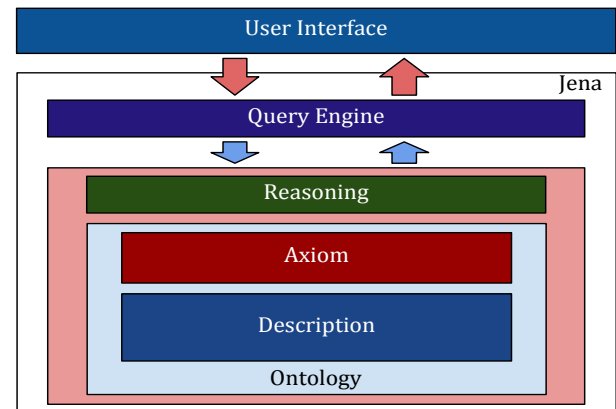


Figura 1. Arquitectura general para la Integración Semántica de Recurso en una Memoria Corporativa.

### C. Representación del Conocimiento

Las tecnologías semánticas proponen al marco de trabajo RDF (Resource Description Framework<sup>3</sup>) para la representación del conocimiento de los recursos en un formato estándar [4]. El primer paso en esta representación es establecer un identificador único (URI) [5] 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](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 también se conoce como descripción de los recursos. Cada propiedad (título, autor y lenguaje fuente) tiene un identificador único (URI) como

<sup>2</sup>The Apache Software Foundation, “Apache Jena,” Available: <http://jena.apache.org/>

<sup>3</sup>W3C, “RDF 1.1 Concepts and Abstract Syntax,” Available: <http://www.w3.org/TR/rdf11-concepts/>

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 prestablecidos 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 triple [4]. El conjunto de todos los triples RDF sobre los recursos de un dominio, se denomina componente asertivo (ABox) [6]. Un triple 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. Los recursos suelen agruparse en clases. Para decir que cierto recurso pertenece a una clase se utiliza el triple: **sirp:recurso** *rdf:type* **sirp:Clase**

Hay diferentes manera de escribir triples, desde lo manual escribiendo literales e identificadores en un editor de texto, hasta lo automatizado utilizando herramientas que mapean la información a triples RDF [7], [8], [9], ejemplos de estas herramientas son: OntoMat, MnM, GATE, KIM, Aktive Media y RDF123. La herramienta que elegimos para administrar triples RDF es el triplestore Apache Jena [10], porque proporciona lectura, procesamiento y escritura de triples 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 los triples RDF. Existen otros triplestore [11], [12], por ejemplo: Stardog, 4store, OWLIM, Sesame, entre otros.

Hemos descrito cómo el marco RDF permite representar o modelar el conocimiento explícito de los recursos (modelo de datos). Este modelo de datos se puede enriquecer con la introducción de *axiomas* [13], 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 se pueden agrupar en otra clase, qué propiedades se agrupan en otra propiedad, qué clases son disjuntas, qué clases son sinónimos o equivalentes, qué clases o clase-literal relaciona una propiedad, qué propiedades son equivalentes (sinónimos), la propiedad es una relación simétrica, transitiva, reflexiva, sólo por mencionar algunas preguntas. Para nuestro caso de estudio, empleamos los siguientes axiomas:

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

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

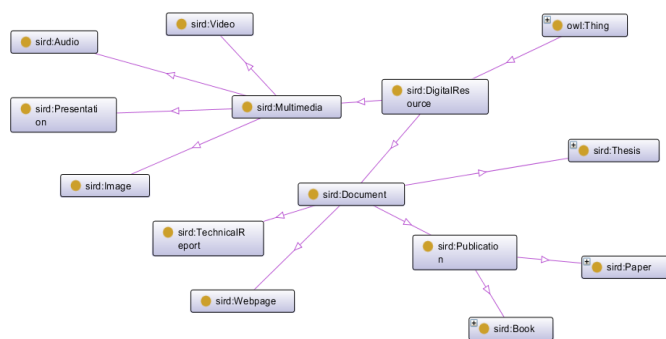


Figura 2. Jerarquía de Clases sobre los recursos digitales de la memoria corporativa (RyT) vistas con protégé.

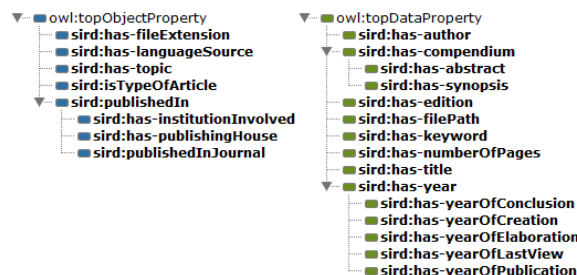


Figura 3. Jerarquía de Propiedades de los recursos digitales en la memoria corporativa (RyT) vistas con protégé.

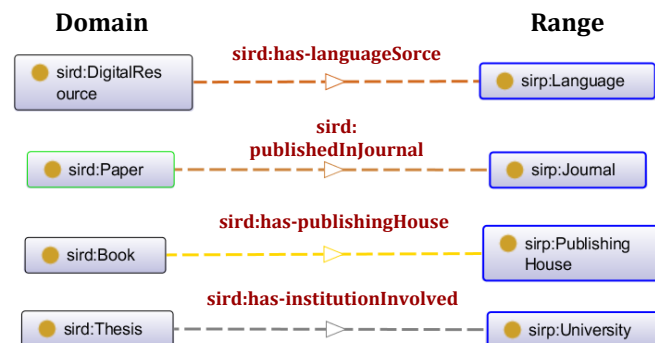


Figura 4. Dominio y Rango de las propiedades de los recursos digitales en la memoria corporativa (RyT) vistos con protégé.

Existen otros axiomas, como son Clases equivalentes, Clases disjuntas, Propiedad Simétrica, Propiedad Transitiva, solo por mencionar. Las funcionalidades y ejemplos de estos axiomas se pueden encontrar en la literatura [6], [14].

Los axiomas también se representan en forma de triples y los lenguajes estándar para escribir axiomas son: RDF Schema (RDF(S))<sup>4</sup> y el Web Ontology Language (OWL)<sup>5</sup>. Existen distintas herramientas para escribir los axiomas en forma de triple con el vocabulario adecuado [15], [16], por ejemplo: protégé, pOWL, SWOOP y TopBrain. La herramienta que nosotros elegimos es protégé, porque es una plataforma que proporciona un interfaz agradable al usuario [17],

<sup>4</sup>W3C, “RDF Vocabulary Description Language 1.0: RDF Schema,” Available: <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>

<sup>5</sup>W3C, "OWL 2 Web Ontology Language Structural Specification and Functional Style Syntax," Available: <http://www.w3.org/TR/owl2-syntax/>

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 [18]. 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 ambos 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 [19]. Nuestra ontología vocabulario (ODARyT4sir) está constituida por 303 conceptos organizados en cuatro ramas principales: Sistemas Distribuidos, Redes y Servicios de Telecomunicaciones, Sistemas de Comunicación Digital y Web Semántica, y para cada concepto se tiene su definición. La Figura 5 muestran las cuatro ramas principales de ODARyT4sir.

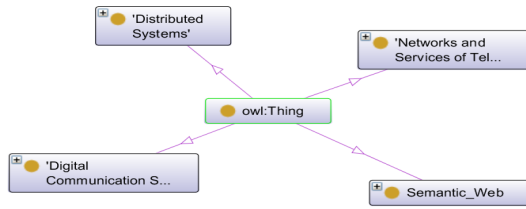


Figura 5. Ontología con el vocabulario de Redes y Telecomunicaciones (ODARyT4sir) vistas con protégé.

Una ontología tiene triples sobre el conocimiento explícito (descripciones de recursos) y el conocimiento implícito (axiomas). Es posible explotar el conocimiento implícito capturado 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) [20]. 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 [20], [21], por ejemplo: Pellet, Fact++, racerPro, 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. En nuestro caso de estudio, los modelos de conocimiento son consistentes, es decir, no tienen contradicciones.

Nuestra ontología de recursos digitales modela el conocimiento de 1330 recursos digitales. En este modelo, hay 691 recursos que tienen explícitamente el triple de asignación (rdf:type) a alguna de las nueve clases básicas: Artículo, Re-

porte Técnico, Página Web, Tesis, Libro, Audio, Video, Imagen y Presentación. Mientras, los otros 639 recursos capturan la pertinencia a alguna clase de manera implícita. La Figura 6 muestra la ontología vista como diagrama de Venn, en el cual, los círculos son las clases de los Recursos Digitales y los puntos son los recursos. Mientras, la Figura 7 muestra la cardinalidad del diagrama de Venn.

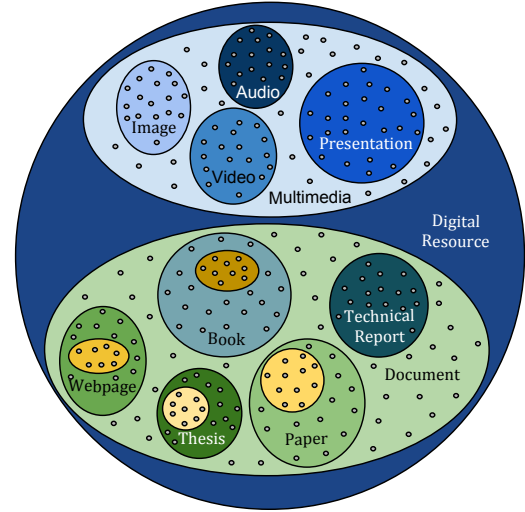


Figura 6. Diagrama de Venn de la ontología de recursos digitales.

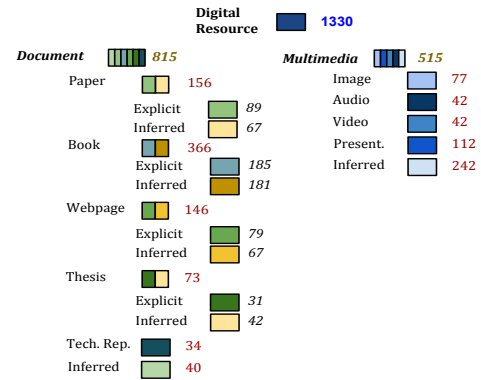


Figura 7. Cardinalidad de recursos digitales.

### E. Consulta de información

Las ontologías son fuentes de conocimiento en forma de triples, para buscar y recuperar (integración) la información de éstas, 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, mediante el análisis de los casos de uso, se identifican y escriben en lenguaje natural las principales preguntas que pueden hacer los usuarios al modelo. En nuestro caso de estudio, se obtuvieron 10 preguntas (Tabla I) para la búsqueda de recursos digitales, pero para ejemplificar la consulta de información, se muestra una de éstas: (1)... ¿Qué documentos sirven para dar un curso de Sistemas P2P?



En el segundo punto, las tecnologías semánticas establecen al lenguaje SPARQL<sup>6</sup> [22] como la especificación para consultar, recuperar y modificar la información de triples RDF. Este lenguaje se basa en el uso de patrones de búsqueda para comparar con los triples del modelo y variables para la recuperación de información. Un patrón de búsqueda es parecido a un triple, pero a diferencia de este último, el sujeto, la propiedad o el objeto pueden ser una variable. En una consulta SPARQL, hay dos cláusulas SELECT y WHERE, la cláusula SELECT enuncia las *variables resultado* y la cláusula WHERE enuncia los *patrones de comparación*. En nuestro ejemplo, la pregunta (1) se escribe como consulta SPARQL de la siguiente manera.

```
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 que permite responder las consultas de los usuarios. La función básica de este motor es: interpreta una consulta SPARQL, compara los patrones con los triples 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 un motor de consulta (ARQ) que soporta el lenguaje SPARQL. Este motor ARQ permite recuperar 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 los triples RDF. Nosotros proponemos una interfaz para la interacción transparente y amigable del usuario con el triplestore; esta interfaz tiene las siguientes características:

- Navegación a través de la información básica de los recursos, por caso de uso.
- Búsquedas específicas de los recursos, por caso de uso.
- Publicación de los resultados de las búsquedas y navegación en un formato visual agradable.
- Mapear la pregunta a una consulta SPARQL.
- Invocar al triplestore (carga, inferencia, búsqueda) y proporcionarle la consulta.

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 8 muestra la interfaz web de *navegación a través de las personas*.

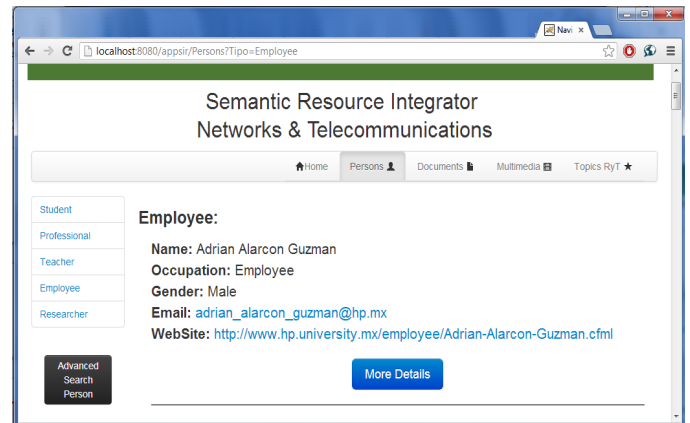


Figura 8. Interfaz visual de navegación a través de personas.

### 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, responden nuestras preguntas adecuadamente.

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 también se muestran el número de recursos digitales que responden a 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

<sup>6</sup>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

procesamiento. Esta prueba se ejecutó usando Java 1.7, con el entorno de desarrollo integrado Eclipse y 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 triples: ABox tiene 20429 y TBox tiene 107. Mientras el vocabulario de RyT (ODARyT4sir) tiene 303 conceptos y en el TBox tiene 1115 triples. Mediante el proceso de inferencia y combinando ambas ontologías se tiene un total de 38661 triples. 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 triples 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 triples, disminuyan los tiempos de consulta y permitan el uso de un razonador.

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. Ribiére, "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 monogeneous information retrieval," in *Computer and Network Technology (ICCNT), 2010 Second International Conference on*, 2010, pp. 591–594.
- [3] J. Rilling, R. Witte, D. Gasevic, and J. Pan, "Semantic technologies in system maintenance (stsm 2008)," in *Program Comprehension, 2008. ICPC 2008. The 16th IEEE International Conference on*, 2008, pp. 279–282.
- [4] 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.
- [5] T. Berners-Lee, R. T. Fielding, and L. Masinter, "Uniform resource identifier (URI): Generic syntax," Internet Requests for Comment, RFC Editor, RFC 3986, Jan. 2005. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc3986.txt>

- [6] M. Krötzsch, F. Simančík, and I. Horrocks, "A description logic primer," *CoRR*, vol. abs/1201.4089, 2012. [Online]. Available: <http://arxiv.org/abs/1201.4089>
- [7] O. Corcho, "Ontology based document annotation&#58; trends and open research problems," *Int. J. Metadata Semant. Ontologies*, vol. 1, no. 1, pp. 47–57, Jan. 2006. [Online]. Available: <http://dx.doi.org/10.1504/IJMSO.2006.008769>
- [8] 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 Semant.*, vol. 4, no. 1, pp. 14–28, Jan. 2006. [Online]. Available: <http://dx.doi.org/10.1016/j.websem.2005.10.002>
- [9] L. Han, T. Finin, C. Parr, J. Sachs, and A. Joshi, "Rdf123: From spreadsheets to rdf," in *Proceedings of the 7th International Conference on The Semantic Web*, ser. ISWC '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 451–466. [Online]. Available: [http://dx.doi.org/10.1007/978-3-540-88564-1\\_29](http://dx.doi.org/10.1007/978-3-540-88564-1_29)
- [10] B. McBride, "Jena: a semantic web toolkit," *Internet Computing, IEEE*, vol. 6, no. 6, pp. 55–59, 2002.
- [11] S. Dietze, S. Sanchez-Alonso, H. Ebner, H. Yu, D. Giordano, I. Marenzi, and B. P. Nunes, "Interlinking educational resources and the web of data - a survey of challenges and approaches," *Emerald Program: electronic Library and Information Systems*, vol. 47, no. 1, 2013.
- [12] H. Mühleisen, T. Walther, and R. Tolksdorf, "A survey on self-organized semantic storage," *IJWIS*, vol. 7, no. 3, pp. 205–222, 2011. [Online]. Available: <http://dblp.uni-trier.de/db/journals/ijwis/ijwis7.html#MuhleisenWT11>
- [13] 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. [Online]. Available: <download/2007/BaHS07a.pdf>
- [14] 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>
- [15] 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.
- [16] 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. [Online]. Available: <http://thor.info.uaic.ro/~busaco/publications/articles/web-ontology-tool-survey.pdf>
- [17] 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, S. . McIlraith, D. Plexousakis, and r. a. n. k. van, Harmelen, Eds. 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](http://dx.doi.org/10.1007/978-3-540-30475-3_17)
- [18] 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>
- [19] A. B. Rios-alvarado, R. C. Medina-ramírez, and M. Iztapalapa, "A semantic web approach to represent and retrieve information in a corporate memory," 2009.
- [20] S. Singh and R. Karwayun, "A comparative study of inference engines," in *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*, 2010, pp. 53–57.
- [21] 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>
- [22] J. Pérez, M. Arenas, and C. Gutierrez, "Semantics and complexity of sparql," *ACM Trans. Database Syst.*, vol. 34, no. 3, pp. 16:1–16:45, Sep. 2009. [Online]. Available: <http://doi.acm.org/10.1145/1567274.1567278>