# Integration of Resources for Corporate Memory using Semantic Technologies

Erik Alarcón-Zamora
Department of Electric Engineering
Autonomous Metropolitan University
Iztapalapa, Mexico
Email: cbi2113802469@xanum.uam.mx

R. Carolina Medina-Ramírez
Department of Electric Engineering
Autonomous Metropolitan University
Iztapalapa, Mexico
Email: cmed@xanum.uam.mx

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

***Abstract*—In this paper, we present a framework supported by semantic technologies for integration of the resources of a corporate memory. We describe the methodology adopted for integration of resources as well as a prototype which shows viability of the semantic approach.**

## I. INTRODUCTION

An organization has a wide variety of information resources, such as staff of the organization, databases, electronic documents, reports, books, presentations, videos, etc.. These resources represent knowledge of the organization: products, research, production processes, operational solutions, workflows, objectives, goals, among others. This knowledge is called corporate memory (CM) or organizational memory (OM) and is defined as "*explicit representation, consistent and persistent knowledge in an organization*" [1]. A memory is important for persons enrolled (members) or people interested in the organization, because it allows them to access, share, exchange and reuse knowledge. Given the importance of memory, you need knowledge management of the same, in order to have the following advantages: better informed staff, better communication in the organization, better decision making, a support tool for learning , a knowledge base persistent and accessible, a tool to search, retrieval and knowledge sharing, to name a few.

Semantic technologies (TS) [2] are *a set of methodologies, languages, applications, tools and standards to provide or obtain the meaning of the information* [1]. These technologies add a *abstraction layer* on sources of information, so that the automatic processes to access, process, reason, combine, reuse and share information. The benefits of using semantic technologies are: catch the vision of contexts, adapting to the changing nature of knowledge, considering the distributed nature of knowledge and users, integrate information from different sources, modeling information in a format standard, use a flexible data model, remove ambiguities in the model, inferences about knowledge, develop generic applications, deploy at lower cost and greater interaction of the experts in the domain.

Semantic technologies can represent and manage knowledge in a corporate memory. In particular, let you do the integration process (search and retrieval) significant information resources in a corporate memory. To achieve this integration, you must carry out the following activities: *1) model the knowledge of resources in a standard format, 2) exploit the implicit knowledge of resources and describe vocabulary (concepts and relationships) from memory*, and *3) search and retrieve information on resources, to answer a given question.*

A corporate memory has multiple information resources and for limiting them to a manageable set, we analyze for detecting priority use cases. This article describes two basic use cases that can be employed in any corporate memory:

1) ***Competency Cartography***: is the search and recovery of significant information of individuals, from the personal and professional characteristics of the same. Some of these professional features are: *skills* (ability to work in a team, leadership skills, etc.), *language skills* (read English, write in Spanish, speaking in French, etc..), *knowledge in the areas of corporate memory domain* (eg operating systems, cognitive radios, etc..), among others.
2) ***Search Digital Resources***: involves the search and retrieval of meaningful information from documents and media files from the content thereof. Some of the search parameters of these resources are: *the author* extension (pdf, doc, wav, etc.), *the issues is the resource* (eg operating systems, ontologies, cognitive radios, etc.), among others.

In this article, represents and integrates corporate memory of *Area Networks & Telecommunications (R&T) of the Autonomous Metropolitan University (UAM)*. The resources of this report represent knowledge of teacher-researchers of R&T. In particular, represent their research resources, collaborations, projects, courses and topics of interest.

This article is organized as follows: section II presents our methodology for semantic integration of resources in a corporate memory. This section is divided into six subsections. Subsection II-A generally describing the methodology and the three general stages of it (representation, development and consultation of the knowledge of resources). Subsection **??** shows the architecture for semantic integration. Subsection II-C describes the resource description framework (RDF) to represent (model) explicit knowledge of resources. Subsection II-D describes axioms and the way to exploit implicit knowledge of resources. Subsection II-E explains query language to

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

interrogate knowledge in the model. Subsection II-F describes the purpose of the prototype for semantic integration. In Section III describes the tests and results (performance and quality of responses) that were made to triplestore Jena [2] and the model for the *networks & telecommunications*. Finally, conclusions about the semantic integration of resources and the experimental results are presented in section IV.

## II. Semantic Integration Resources

Semantic Integration Resource (SIR) is the process of search and retrieval of information significant existing information resources (documents) that are resident on a storage medium. It is based on the use of semantic technologies (ST) [3]. The purpose of this integration is to retrieve documents related to answer a question asked by a user. In this paper, integration SIR is effected in a corporate memory (CM), because this SIR considers some important features of a corporate memory as: explosive growth of resources, heterogeneity in format, content and structure of resources, ambiguities in the information, evolution of knowledge about resources (add, delete, or amend), among others. The main users in the ISR are the experts and people linked to the domain of the CM. In the following subsections, we describe our proposed integration and architecture of this, a knowledge representation of resources, how to exploit knowledge of resources, searching for information, as well as an interface prototype.

### A. Proposal

We propose a methodology for developing *semantic integration of resources* in a *corporate memory*. This methodology was developed considering two use cases, but can be extended to other use cases. This methodology can be used in any CM, for example, Biomedical, Networks and Telecommunications, Chemistry, Biology, Physics, among others, since these are composed of information resources that can be integrated to answer user questions. To carry out the SIR, our methodology involves three general stages:

1) Knowledge Representation: consists in modeling the explicit knowledge of resources in a standard format.
2) Knowledge Exploitation: consists in using inference rules to materialize implicit knowledge.
3) Information Query: involves interrogating the knowledge model from a user a question and respond with information about the resources.

### B. Architecture

The proposed architecture for semantic integration is a solution, which on the one hand, represents and exploits knowledge about the resources of a corporate memory in a model, on the other hand, allows the integration (search and retrieval of information) on the model knowledge. This architecture is generic and can be implemented in any corporate memory associated with a domain. Figure 1 shows our architecture, in which several components are developed in one of three stages of the SIR.

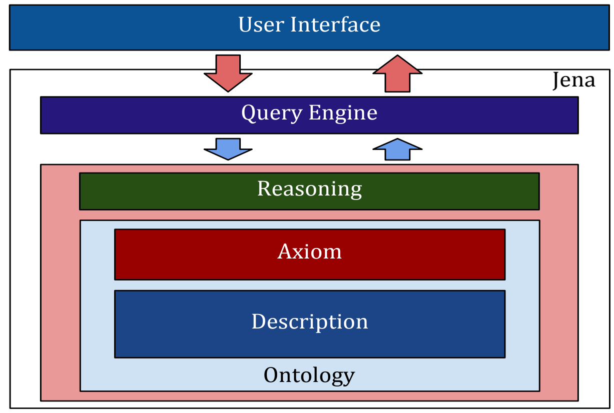


Figure 1. General architecture for semantic integration of resources on a corporate memory

### C. Knowledge Representation

In semantic technologies, it has the framework RDF (Resource Descrption Framework[3]) for knowledge representation of resources in a standard format [4]. The first step to represent the RDF Framework, is established to each resource corporate memory one Resource Unique Identifier [5] (URI), in order that every resource has a unique name and there is no ambiguity between them. For example, the resource *Towards a semantic search* is assigned the following name: *http://arte.izt.uam.mx/ontologies/digiResourceRyT.owl#Hacia_busqueda_semantica*.

Within the framework RDF, resources are represented by their properties (basic characteristics or metadata) and the values assigned to them; this representation is also known as resource description. Each property (title, author and source language) has a unique identifier (URI) as a name. In writing of the name, a verb is written in third person and present tense ('Is', 'is','known', among others) before metadata. For example, the property *title* has the following URI: *http://arte.izt.uam.mx/ontologies/digiResourceRyT.owl#has-title*.

To reduce the size of the identifier (URI) of resources and properties, is replaced sequence of characters from *http://www* to the symbol # for a certain prefix. In our case study the prefix *sirp* translates *http://arte.izt.uam.mx/ontologies/digiResourceRyT.owl#*. For example, the property *has title* is written *sirp: has-title*. In semantic technologies, other prefixes assigned to other vocabularies, such as *rdf, xsd, rdfs and owl*.

The resource properties and their values are taken to a standard format, which in this framework is called triple [4]. The set of all RDF triples on the resources of a domain, is called assertive component (ABox) [6]. A triple is formed by a subject, a predicate and an object; subject is identifier of a resource being described, predicate is identifier of a property described, and the object is either a literal (string, integer) or other resource identifier. An important property for a resource is *assignment*, which links to the resource with a particular class. In the case study, digital resources in the ***CM of N&T*** are classified as: Article, Technical Report, Website, Thesis, Book,

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

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

Slideshow, Image, Audio and Video. In the representation of this *assignment* in the form of RDF triple; subject is a resource, predicate is the name of the assignment (belonging) in the RDF vocabulary is defined as *rdf:type*, and the object is a identifier of a class. For example, the description *Towards a Semantic Search is an article*, is written in triple as follows: sirp:Hacia_busqueda_semantica rdf:type sirp:Article.

There are different way of writing triples from the manual, writing literals and identifiers in a text editor, to automated, using tools to map the information to triples [7], [8], [9], as: OntoMat, MnM, GATE, KIM, Media and RDF123 Aktive. In this paper, the tool we choose is the triplestore *Apache Jena* [10], because it provides reading, processing and writing triples in different formats (XML/RDF, N-triples, Turtle), and this triplestore can be used with the programming language *JAVA*. A triplestore is a system that has two basic functions: the storage and access of RDF triples. There are other triplestore [11], [12] as: Stardog, 4store, OWLIM, Sesame, among others.

### *D. Knowledge Explosion*

At this point, the RDF framework or model to represent explicit knowledge of the resources (data model). This model can be enriched with introduction of *axioms* [13] which supplement, extend, renew and adapt knowledge of resources. To identify the axioms, we analyze the classes/properties and ask questions about them: what classes can be grouped under another class, which properties are grouped under other property, which are disjoint classes, which are synonymous or equivalents classes, what classes or class-literal relates a property, which properties are equivalents (synonymous), property is symmetric, transitive or reflexive, just to mention a few questions.For our case study, we use the following axioms:

**Subclass and Subproperty**, allow establishing hierarchies among classes (Figure 2) and properties (Figure 3) respectively.

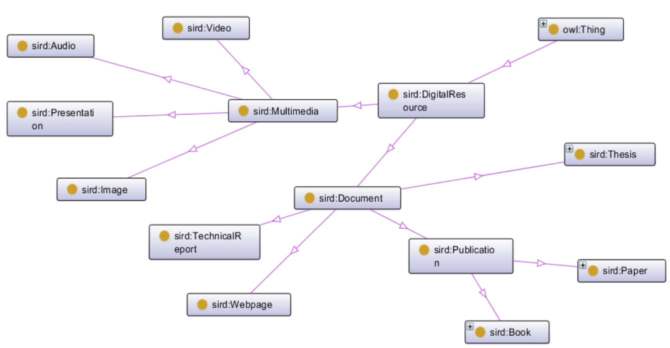


Figure 2. Class Hierarchy of digital resources in the corporate memory (N&T) views with protégé.

**Domain and Range**, to set what classes or class-literal, must relate a property (Figure 4.)

There are other axioms, such as: equivalent classes, disjoint classes, property symmetric, transitive property, just to mention. The features and examples of these axioms can find literature [14], [6].

In the same way that the descriptions, the axioms are represented as triples and semantic technologies, standard

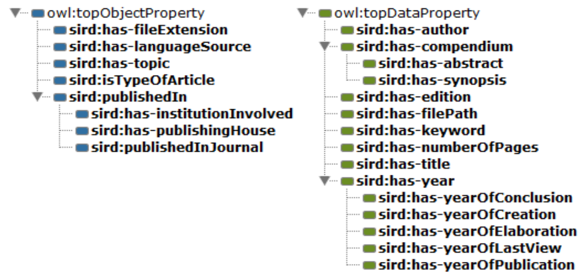


Figure 3. Properties Hierarchy of digital resources in the corporate memory (N&T) views with protégé.

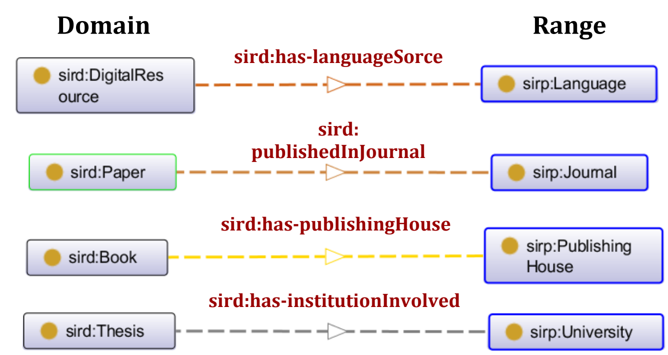


Figure 4. Domain and range of properties of digital resources in the corporate memory (N&T) views with protégé.

languages for writing axioms are: RDF Schema (RDF(S) [4]) and Web Ontology Language (OWL[5]). There are different tools for writing axioms in the form of triple with appropriate vocabulary, such as [15], [16]: protégé, powl, SWOOP, TopBrain, and even to generate triple Jena axioms. In this paper, was elected protégé tool because it is a platform that provides a user friendly interface [17], allowing it the creation, manipulation and visualization of axioms. In addition, this tool can save axioms in different formats (XML/RDF, Manchester, Turtle, OWL/XML).

The set of axioms that enrich the model is called terminological component (TBox) and semantic technologies, knowledge model consisting of TBox and ABox, is called ontology [18]. In this proposal, the use cases are independent, therefore, it was decided that each of these has its own ontology. On the other hand, a common goal in both use cases is to link resources with the themes of corporate memory domain. For this, we propose a third ontology which has vocabulary of domain. In our case study, the ontology is the vocabulary of Networks and Telecommunications (N&T) that developed from another ontology ODARyT [19]. Our ontology vocabulary (ODARyT4sir) consists of 303 items organized into four main branches: Distributed Systems, Networking and Telecommunication, Digital Communication Systems and Semantic Web, and for each concept has its definition. Figure 5, shows the four main branches of ODARyT4sir.

An ontology has triples on explicit knowledge (resource

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

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

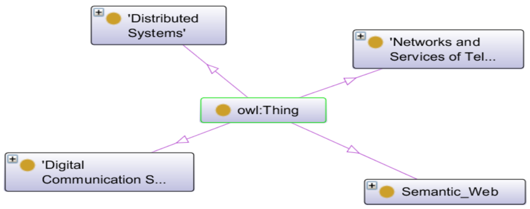


Figure 5. Ontology with the vocabulary of Networks and Telecommunications (ODARyT4sir) (N&T) views with protégé

descriptions) and implicit knowledge (axioms), but the model does not have explicit triples of implicit knowledge. To materialization of this implicit knowledge, uses a reasoner (inference engine) which is a program for inferring facts or associations from existing knowledge (axioms and properties) [20]. For example, we have an ABox with the following description: recourse ”Towards a semantic search” is a paper, and TBox with the following axiom: Paper class is a subclass of the Document class, using a reasoner with this ABox and TBox , materializes (inferred) the following description: recourse *Towards a semantic search* is a document. There are different reasoners, such as [20], [21]: Pellet, fact++, racerPro, to name a few. In this paper, we used the reasoner which has by default Jena, because it can be invoked from Java+Jena, supports our ontology axioms (RDF(S) and OWL) and does not require a previous compilation or configuration for use. Furthermore, a reasoner is a tool to validate the model of knowledge, because to find contradictions or ambiguities. In our case study, knowledge models are consistent.

Our ontology of *digital resources* models knowledge of 1330 digital resources. In this model, there are 691 resources which have explicitly assignment triple (rdf: type) to one of the nine basic classes: Durable, Technical Report, Website, Thesis, Book, Audio, Video, Image and Presentation. Meanwhile, the other 639 resources through inference, materializes this triple. Figure 6 shows the ontology as venn diagram where the circles are the kinds of Digital Resources and the points are the resources. Meanwhile, Figure 7 shows cardinality of Venn diagram.

### *E. Query Information*

Ontologies are sources of knowledge in form of triples and to integrate information from them, three things are needed: *1) a question in natural language, 2) a query from triple and 3) a query engine.*

On the first point, through analysis of use cases are identified and written in natural language the main questions that users can do the model. In our case study, we obtained 10 questions for finding digital resources, but to exemplify the information query, shows one of these: (1) What documents are used to teach a course in P2P Systems?

On the second point, you should use a query language to search and query the triples. In semantic technologies, the language SPARQL[6] [22] is specification to query, retrieve and modify information of RDF triples. This language is based

[6]W3C,“SPARQL 1.1 Overview,” Available: http://www.w3.org/TR/sparql11-overview/

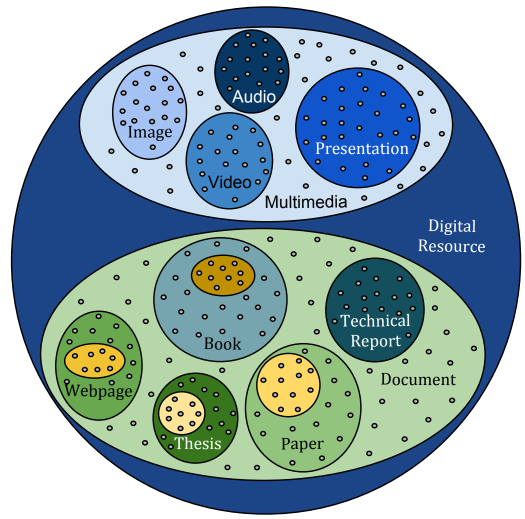


Figure 6. Venn diagram of ontology of digital resources by class view.

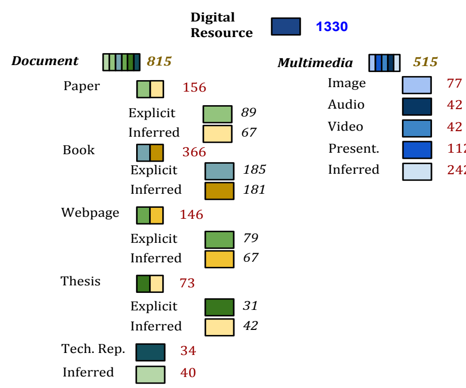


Figure 7. Cardinality of digital resources.

on the use of search patterns to compare triples model and variables for information retrieval. A search pattern is similar to a triple, but unlike the latter, subject, property or object can be a variable. In a SPARQL query, there are two clauses: SELECT and WHERE. The SELECT clause sets out *outcome variables* and WHERE clause sets out *comparison patterns*. In our example, the question (1) is written as SPARQL query as follows:

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

For third point a SPARQL query engine is a program that can respond to user queries. The basic function of this engine is a SPARQL query interprets, compares triples patterns and model, retrieves information associated with variables that are in SELECT clause and returns information to user. A query engine can be found in triplestore. In particular, Jena has a query engine (ARQ) that supports SPARQL language. This engine ARQ to retrieve results of a query and display on the screen in form of a table or to process these results with JAVA.

### *F. Prototype (interface User)*

The semantic integration of resources (SIR) using a triplestore, not a task that any user can do, since it must be familiar with the triplestore and elements of semantic technologies; in particular SPARQL query language and RDF triples. We propose an interface for interaction, transparent and user friendly with triplestore, this interface has the following characteristics:

- Browsing through the basic information resource for use case.
- Resource specific searches for use case.
- Publication of results of search and navigation in a visual format enjoyable.
- Mapping the question to a SPARQL query.
- Invoking triplestore (load, inference, search) and providing at same the query.

In particular, our prototype is a web application interface that works with the triplestore jena, this prototype was implemented in Java and provides a set of servlets that are on a Tomcat server. The prototype provides the following visual interfaces: Browsing through people, Navigation through documents, Navigation through multimedia resources, Advanced people search, Advanced search documents, Multimedia resources advanced search, Advanced search any resource. Figure 8 shows the interface ***Navigation through people***.

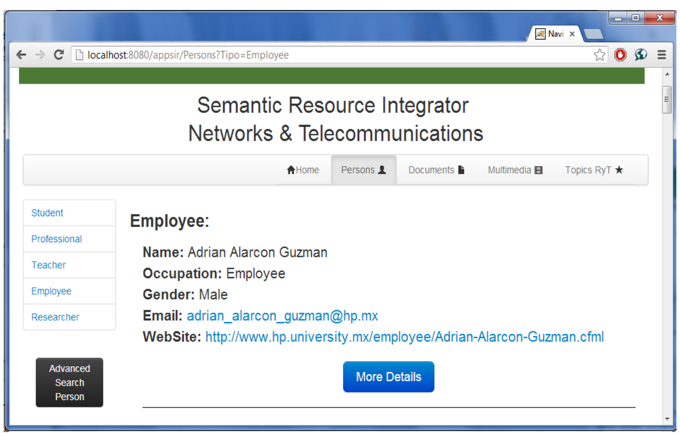


Figure 8. Visual interface navigation through people.

## III. EXPERIMENTATION

In the semantic integration of resources, we have two evaluation criteria for triplestore Jena: *1) performance and 2) the results for a model with inference (reasoner ontology) without inference (explicit descriptions)*. For performance evaluation, we wanted to see if Jena performs adequately with the amount of data that we expect to handle, while for the evaluation of results, we want to see if the results returned by Jena are those who answer our questions.

The performance evaluation involves taking the mean time from model charge, until recovery of results of a query. While the evaluation of results, was to compare information retrieved from a SPARQL query with the resources that are known answer the question (previous analysis). For these two assessments, there was a program in java+jena which repeated $n$ times the processing time for a model and a given query, and in each iteration returns the values of the query. The three input parameters of this program are: the number of repetitions, type of model *with explicit data or a reasoner and an ontology* and SPARQL query. Meanwhile, the three output parameters are: average processing time of the consultation, the number of resources that responds the search engine and query responses.

In this paper, we established following parameters for the program: the number of iterations $n$ was placed at 20, the models are: 1) the Abox digital resources of N&T and 2) the ontology digital resources and vocabulary of N&T, and 3) questions are shown in Table I. This table also shows the number of digital resources that respond to the questions.

Table I. QUESTIONS IN NATURAL LANGUAGE AND AMOUNT OF RESOURCES THAT RESPOND TO THEM

| Id. question | Question | No. of resources |
|---|---|---|
| Q1 | What are the titles, routes, extension, language, of all digital resources in N&T? | 1330. |
| Q2 | What books are on some topics Distributed Systems? | 103 |
| Q3 | What resources were published in the UAM? | 18 |
| Q4 | What documents are to give a course in P2P Systems? | 31 |
| Q5 | What media resources are greater than 2009? | 119 |
| Q6 | What documents are about Ontology? | 30 |
| Q7 | What resources were published in a scientific journal? | 156 |
| Q8 | What resources are in content the words “linked data”? | 159 |
| Q9 | What documents in English and over 2000 are authored by Erik Alarcon Zamora? | 2 |
| Q10 | What thesis of Samuel Hernandez Maza? | 4 |

This program was run on a computer with an Intel Core i7 at 2.3GHz with 8Gb of RAM and 8 processing cores. This test was run using Java 1.7 with integrated development environment Eclipse and Apache Jena 2.7.4 on Windows 7 (64bit). We run the program twice for each question from the list (Table I). The first run, model type is ABox, while in the second run, the model is obtained from the reasoner and ontology. In our case study, the ontology of digital resources has 1330 digital resources and the following amounts of triples: Abox has 20429 and TBox has 107. While vocabulary of N&T (ODARyT4sir) has 303 concepts and the TBox has 1115 triples. Through inference process and combining both ontologies, we have a total of 38661 triples. The results of measurements of the average time and the number of responses for each query is shown in Table II.

Table II shows amount of resources that answer a SPARQL query and to verify that the information in a query answer the question, we did an analysis of the resources that answer each of the questions. Then manually compared information recovered from each SPARQL query with the responses of respective question. In our tests, all results of SPARQL queries are digital resources that we wanted for our questions.

Jena performance is good (less than 1 second) when explicit knowledge is questioned because the query engine directly interrogates the ABox triples. In contrast, Jena consumes

Table II. AVERAGE PROCESSING TIME AND AMOUNT OF RESOURCES THAT MATCH A QUERY.

| Id. question | Model (ABox) | | Model (Reasoner+Ontology) | |
|---|---|---|---|---|
| | Average Time (ms) | No. of Resources | Average Time (ms) | No. of Resources |
| 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 |

more time when model is result of inference, because a reasoner invests time in inference process and if this materializes the triples, then engine compares more triple and invest more time. We think, though Jena does not have a good performance with the use of reasoning, this can be optimized through the use of parallel programming because Jena works with JAVA and can be used processing threads; activity that is outside the scope of this paper.

With respect to the evaluation of information retrieval in two models. On one hand, if the model is the ABox and query is on descriptions, then all query results are expected responses of a question, but if this query is about explicit and implicit knowledge, then several results of question will not be recovered by the ARQ engine. On the other hand, if the model is obtained by the inference in an ontology and the query is about the explicit or implicit, then the ARQ engine is able to recover the expected information for the question. In this way we conclude that despite investing time for inference, we obtained satisfactory results for a user's question.

## IV. CONCLUSION

A corporate memory is a source of knowledge to an organization, if this knowledge is represented properly, then the integration of resources will have better results. Our methodology is a generic solution for integrating resource semantics (SIR) which can be implanted in any corporate memory. This methodology is based on two use cases, but is not limited to these, if required adding or changing use cases proposed, then the methodology can be extended to these. We chose semantic technologies for SIR, because it allows: 1) to model, develop and consult knowledge about resources, 2) represent knowledge in a standard format, 3) use tools for developing semantic applications, 4) use and share multiple vocabularies, 5) use applications to exploit the knowledge, among other advantages. Our methodology for integration (SIR) is based on three basic points: representation, exploitation and consultation of knowledge. In addition to this integration, we developed a prototype for user-friendly and transparent interaction of users with the SIR.

In the performance test of Jena for consultation process, the average time is less than one second when it queries the ABox. But, the time increases (1-3 seconds) when querying a model that was obtained by inference in an ontology. Although the performance of Jena is acceptable for inference model, we believe that using parallel computing operations, these times can be reduced. In the evaluation of information retrieval, using a model that materializes triples, then the results given by Jena are answering the questions we ask. In this way we believe that despite investing time in inference, the results are obtained that satisfies a user's question.

## REFERENCES


[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] 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: 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

[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

[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-ramÃrez, 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