

Knowledge Representation and Reasoning on the Web of Data

D. Teja Santosh

MVGR College of Engineering, Chintalavalasa,
Vizianagaram

P. Srinivasa Rao

MVGR College of Engineering, Chintalavalasa,
Vizianagaram

ABSTRACT

Finding the right information at the right time and in the right format becomes increasingly difficult as more information from myriad producers is made available to increasingly diverse communities of information consumers. The domain of eGovernment is unique because of its enormous challenge to achieve interoperability, given the manifold semantic differences of interpretation. Setting up seamless e-government services requires information integration as well as process integration involving a variety of objects with specific semantics.

One of the information sharing approach- the Semantic Web[7]. One of the opportunities opened by the Semantic Web is the possibility of accessing and reusing knowledge bases available via the Internet in the form of RDF/OWL ontologies.

Keywords

Semantic Web, RDF, OWL, Ontologies.

1. INTRODUCTION

Modern e-Government[3] calls for interoperable information across various departments and communication systems, which ideally interact smoothly to provide citizen services and public access in a single window. In recent times, the usage of Internet has enlarged multi-fold and there is still a large gap between the information requirement and availability, which needs to be filled. The information was captured using heterogeneous applications and recorded in heterogeneous storage environments. Sharing of information requires conversion data from one form to another, which is laborious, time consuming and costly as heterogeneous development and storage environments are being used. Present web-enable search engines such as Google, AltaVista, Yahoo, windows live etc., will only provide access to HTML documents. Moreover, the service provider machines are vulnerable to hackers. These semantic differences are related to a great variety of IT solutions (on local, regional, inter-/national level) which will have to be networked (despite any effort of standardization).

In consequence, some of the key obstacles for networked computer applications in governmental processes and services are those kinds of barriers in which the different meanings of data objects and interfaces[6] cannot be automatically mediated. Hence, a simple and clear-cut standards and specifications are required to achieve interoperability of information and communication systems.

The vision of Sir Tim Berners Lee is to publish open government data (develop a Service Oriented Architecture) that follows the principles of the same.

The paper is divided into six sections which provides emphasis on definition and principles of open government data, provides the architecture of the proposed system, what are proposed algorithms for the proposed system to work effectively, implementation screens, conclusion and future work and references.

2. DEFINITION AND PRINCIPLES OF OPEN GOVERNMENT DATA

The government data that is considered open, modular and scalable is open government data[1]. The government organizations must be able to share[5] and integrate this open data.

2.1 Principles of Open Government data

1. Complete
2. Primary
3. Timely
4. Accessible
5. Machine process able
6. Non-discriminatory
7. Non-proprietary
8. License-free[5]

3. SCHEMATIC DIAGRAM OF PROPOSED SYSTEM

The User Interface Layer

- The User Interface layer is appropriately diverse.
- Existing options at the application layer enable structured data and its controlling vocabularies (ontologies) to drive applications and user interfaces.

The Ontology ('Schema') Layer

- Ontologies are the key structures that provide the horsepower behind these ontology-driven applications.
- Two roles for ontologies are to position existing datasets into an "aboutness" framework and to help guide how the data can be described and related to other data. General vocabularies can support more focused vocabularies through the incorporation of domain-specific ontologies.

The Conversion, Extraction and Authoring Layer

- Information extraction is important because 80% to 85% of all information resides in unstructured text. Metadata tagging through information extraction allows faceting, finding named entities, and inferencing over conceptual relationships[4].

Leveraging Existing Assets and Web Services Interfaces

- The source of most of the information for interoperation at the semantic layer comes from existing assets (like XML DBs, Oracle DBs , et. al.,).
- A Web services framework can provide platform-independent middleware for accessing and exposing structured RDF[8] data, with generic tools driven by underlying data structures. Best practices include using the perspective of the dataset.

4. PROPOSED ALGORITHMS

4.1 Metadata Extraction

The goal of this task is to extract metadata in the strict sense, i.e., data about text, based on the Dublin Core standard. Extracting metadata from a text requires mainly entity recognition, as the object of the relationship is already known

4.2 Filtered Information Retrieval

This task allows constraining terms in IR queries to selected entity types[9] (e.g., documents in which Ford refers to a person, not a company).

4.3 Entity Referencing

This task augments web pages with cross-references from contained entities to external background information.

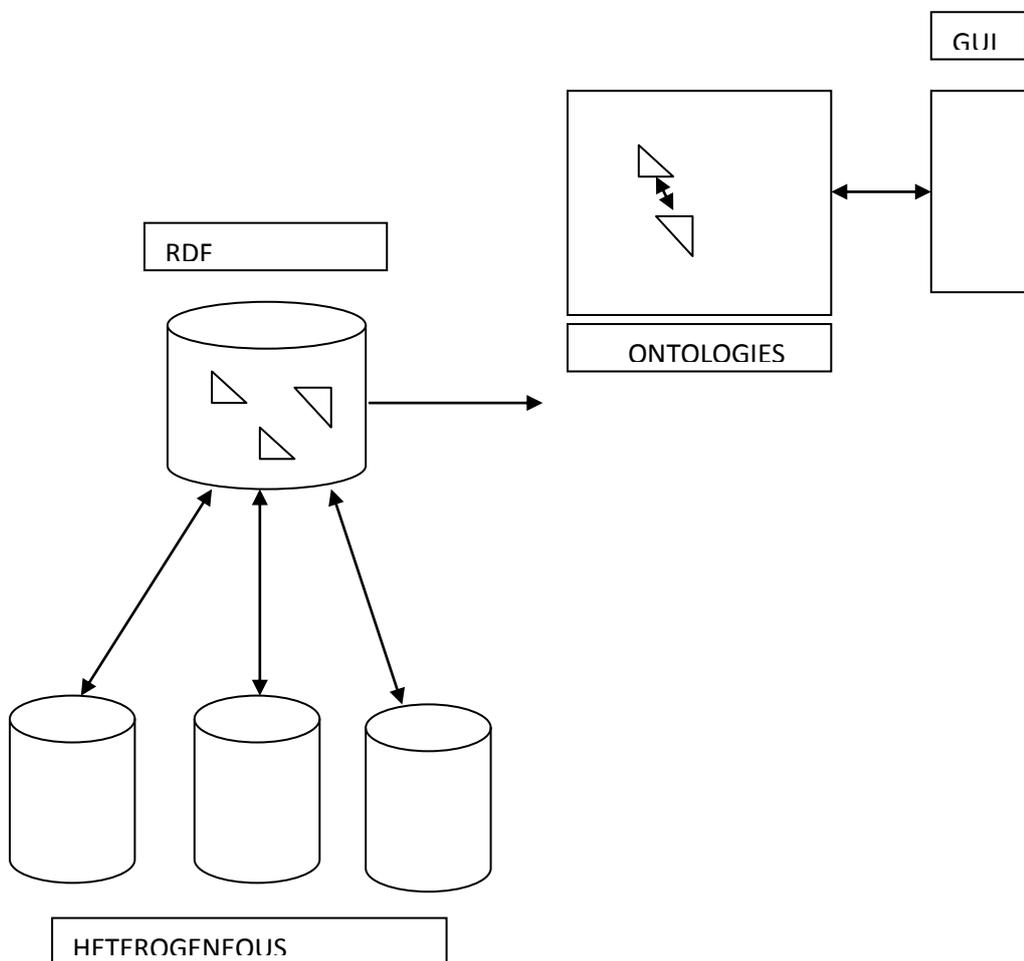


Fig 1: Generation of RDF Triples and Classification of derived Ontologies

4.4 Semantic Highlighting and Indexing

Semantic highlighting marks all entities of a certain type in a text or all co references to a chosen entity. Semantic Indexing shows a hyperlinked index of the names of all persons (or any other entity type) mentioned in a document. An index can also list the subjects or objects of a chosen relationship[10] , e.g., all persons working for company X.

5. IMPLEMENTATION SCREENS

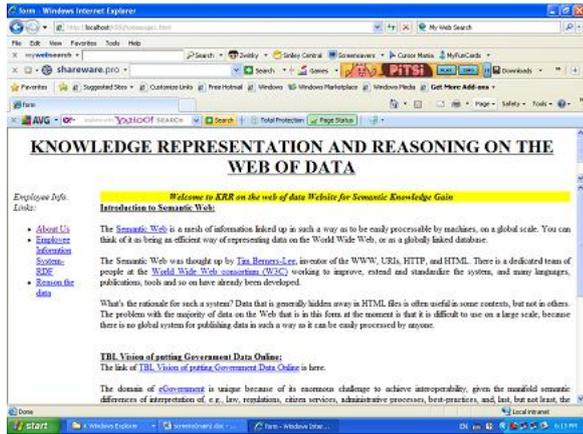


Fig 2: Welcome Page

Figure 2 shows the homepage of the application. Entire application is developed in Dot Net.

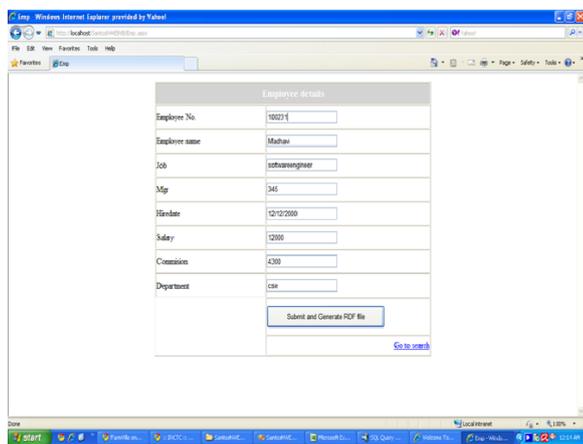


Fig 3: Record Creation Field Data

Figure 3 asks the end user to enter his/her details to create a record in their respective databases.

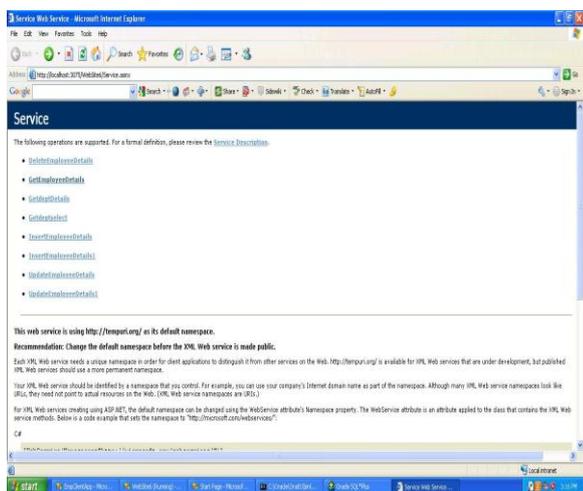


Fig 4: Web Service taking the request

Figure 4 shows the Web Service which is running at the server site containing a number of web methods from which the actual RDF Triples can be generated.

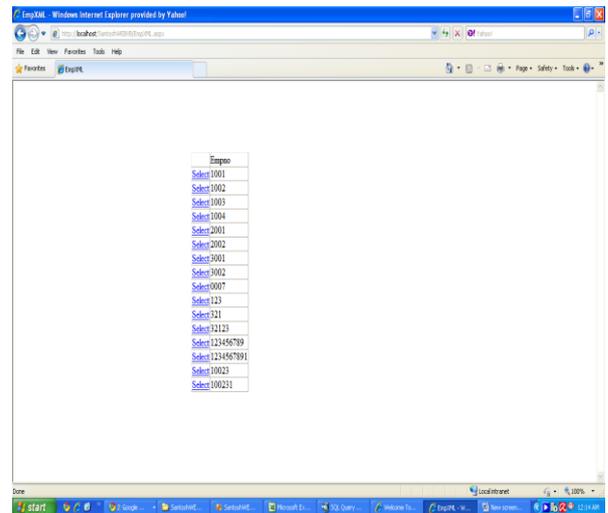


Fig 5: Links of RDF Triples from databases

Figure 5 shows the links obtained from various databases and which are in the final form of RDF Triples.

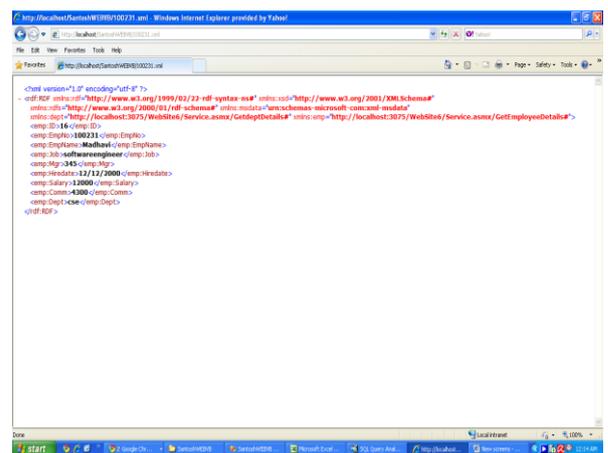


Fig 6: Knowledge Representation in RDF

Figure 6 shows the RDF format of representing knowledge extending normal XML. The Triples (Subject, Predicate and Object) are shown in this RDF.



Fig 7: Searching based on Ontologies

Figure 7 asks the end user for the input to search for the required data. This is achieved by generating Ontologies for the obtained RDF Triples.

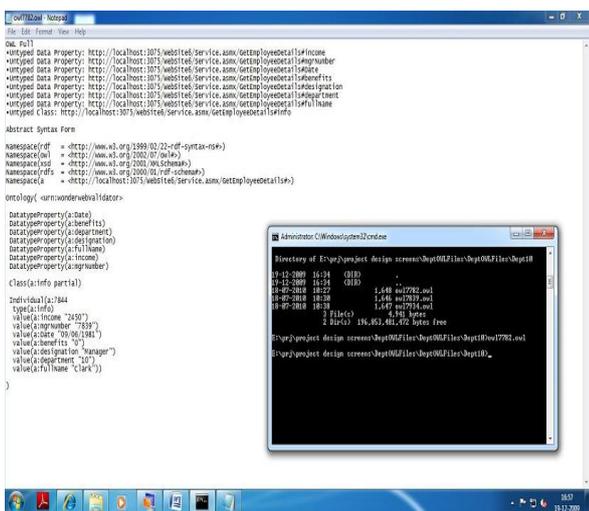


Fig 8: OWL reasoning of data

Figure 8 shows the Ontologies in OWL format that have been reasoned.

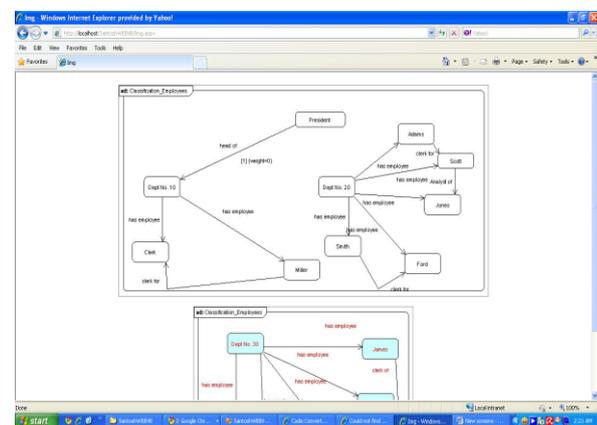


Fig 9: Classification of RDF Triples in picture format

Figure 9 illustrates the classification of obtained Ontologies in a picture fashion.

6. EXPERIMENTAL RESULTS

The experimental results provide the information about the reasoned ontologies from the obtained classifiers. The classifier here is “has_employee”. Conclusions have been derived from the reasoned ontologies which helps in accurate acquisition of knowledge.

Table 1. Department 10 Classifier Table

Department number	Classifier	Employee Name
10	has_employee	Miller
10	has_employee	Clark

Derived (Reasoned) Conclusion is Miller is “clerk for” Clark.

Table 2. Department 20 Classifier Table

Department number	Classifier	Employee Name
20	has_employee	Adams
20	has_employee	Scott
20	has_employee	Jones
20	has_employee	Smith
20	has_employee	Ford

Derived (Reasoned) Conclusions are:

Adams is “clerk for” Scott.

Scott is “analyst to” Jones.

Smith is “clerk for” Ford.

7. CONCLUSION AND FUTURE WORK

Development of a “KILLER APP” that automates the needed only data retrieval which ultimately leads to a SOA Application to the end user who feels the application is a convenient one. Vigorous research and work is going on providing security for the web of data by enforcing rigid policies in parallel to the publishing.

8. REFERENCES

- [1] Tim-Berners Lee, Wendy Hall, James A. Hendler, Kieron O'Hara, Nigel Shadbolt and Daniel J. Weitzner, “A Framework for Web Science”, Foundations and Trends in Web Science, ISBN: 1-933019-33-6 144 pp, September 2006
- [2] Tim Berners-Lee, forward: Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential by Dieter Fensel (Editor), Wolfgang Wahlster, Henry Lieberman, James Hendler, MIT Press, 2002

- [3] Tim Berners-Lee, "WWW: Past, Present, and Future", IEEE, Computer Magazine, Vol. 29, No. 10, Oct. 1996.
- [4] James Hendler, Tim Berners-Lee and Eric Miller, 'Integrating Applications on the Semantic Web', Journal of the Institute of Electrical Engineers of Japan, Vol 122(10), October, 2002, p. 676-680
- [5] Tim Berners-Lee and James Hendler "Publishing on the Semantic Web", Nature, April 26 2001 p. 1023-1025.
- [6] Max L. Wilson, Bill Kules, m.c. schraefel and Ben Shneiderman (2010) "From Keyword Search to Exploration: Designing Future Search Interfaces for the Web", Foundations and Trends® in Web Science: Vol. 2: No 1, pp 1-97.
- [7] Tim Berners-Lee, James Hendler and Ora Lassila, "The Semantic Web", Scientific American, May 2001, p. 29-37.
- [8] World Wide Web Consortium (W3C). (2004a). D. Beckett ed. RDF/XML Syntax Specification (Revised), W3C Recommendation. D. Beckett ed. Available: <http://www.w3.org/TR/REC-rdf-syntax/>
- [9] World Wide Web Consortium (W3C). (2004b) RDF Vocabulary Description Language 1.0: RDF Schema. Available: <http://www.w3.org/TR/rdf-schema/>
- [10] Atre, M., Chaoji, V., Zaki, M. J., Hendler, J. A.: Matrix "Bit"loaded: A Scalable Lightweight Join Query Processor for RDF Data, In: WWW 2010, pp. 41-50. ACM Press(2010)