

Exploitation of ontology by the Jena semantic API

Sabria Hadj Tayeb¹, Myriam Noureddine²

^{1,2} Department of Computer Science, Faculty of Mathematics and Computer Sciences,
University of Science and Technology USTO-MB, Oran, Algeria

¹ s.hadjtayeb@gmail.com

² myriam.noureddine@univ-usto.dz

Abstract—Ontology's are currently among the most talked in knowledge engineering models to define structured vocabularies, gathering useful concepts in a domain and their relationships used to organize, share information unambiguously. Once created, the ontology can serve or be used directly by the user, it is necessary to represent using a formalism that will acquire its importance once integrated.

This paper presents the exploitation of ontology by the Jena semantic (Application Programming Interface) for a classification of relations in enterprise. Our approach is to read an ontology previously created via the Jena API using the Eclipse environment, in order to extract the desired information. The approach is applied to a local enterprise and the result is given in the form of relations. These relationships will be classified according to their kind in order to improve the performance of the concerned enterprise.

Index Terms— Ontology, Jena API, relationships, classification.

I. INTRODUCTION

To address problems the heterogeneity due to the development and disparity data, approaches have emerged and are thriving as ontology's. Ontologies are intended to improve the use of information resources by positioning themselves as a representation model consisting of concepts linked by relations structured hierarchically. They allow for example to organize knowledge according to the area in question, to detect inconsistencies as well as to improve or optimize relationships. After their creation, the user can take full advantage of ontologies only if they are exploited through the use of tools.

This paper presents the exploitation of ontology by the Jena Semantic API for classification of relations in enterprise. In the second section, we first briefly give the notion of ontology. We chain by presenting tools, focusing on the Jena API to exploit ontology; then we present our-approach. In the next section, the application of the process on the local enterprise of transformation wiredrawing steel is proposed. The result will list all information contained in the ontology (classes, relationships,

etc.) or to filter them according to specific needs. Then, we assign on each extracted relationship nature and weight to classify in order to improve business performance.

II. PROPOSED APPROACH FOR THE EXPLOITATION OF ONTOLOGY

A. Exploitation of ontology

1) *The concept of ontology*: According to Gruber [1], an ontology is a formal, explicit specification of a conceptualization. In general, ontology's describe the structure and semantics of data. They enable users to organize information to concepts, attributes and relationships [2], according to a domain ontology allows a meta-knowledge representation. Ontology's therefore provide a common vocabulary defining the meaning of terms and relations between them.

2) *Tools for the ontology exploitation*: There are few tools dedicated to ontology exploitation, the most used are the Jena [3] API and Jade platform [4].

Jena is an API (Application Programming Interface) for Java applications of the Semantic Web. This API can read and manipulate ontologies described in the RDFS model (Schema Resource Description Framework) or in the Ontology Web Language (OWL). It can use also to apply some inference mechanisms.

Jade (Java Agent Development Framework) is a multi-platform agents. The exploitation of ontology using Jade results in a process suitable for mapping Jade agents. This process is tedious and it requires a domain expert.

B. The proposed approach

1) *Ontology building*: The preliminary step is the construction of ontology. For building an ontology, we have adopted the Protege environment. Protege [5] is a free Java tool ; it is

produced and made available by the laboratory Stanford Medical Informatic. It plays a crucial role in assisting the developer of the ontology when building the ontology through a graphical modular environment. We have worked with versions 3.4 and 4.1 Beta.

2) *Ontology reading*: Knowing that the Jena API is a framework designed to exploit ontology's, we adopted this tool to read the ontology in order to extract all relationships.

This API includes a set of programming tools [6] using the Java programming language. Jena has object classes to represent graphs; these classes are named:

- *Resource (Subject)*: A resource is an entity that we refer to. It can even be a web page, a link, a particular user identity.
- *Properties (Predicate)*: A properties specified the characteristics of the resource.
- *Object (literal)*: A literal value represent data or resource.

3) *Classification of relationships* : We assign to each extracted relationship a nature of type and a weight to be able to class these relations for improved performance. In the context to enhance the performance of the enterprise, we associate the concept of nature with respect to the notion of efficiency and effectiveness [7] [8].

The Efficiency relates to the following questions: What material, human and financial resources have been mobilized effectively? The effects achieved are they in line with all the resources used? Could we have achieved the same results at a lower cost?

The Effectiveness measures whether the expected achievements were implemented.

Depending on the nature of the relationship (efficiency / effectiveness), we assign a weight for each considered relation. The final result will allow to classify the relationships in order to improve the performance of the concerned enterprise.

The Figure.1 illustrates our approach.

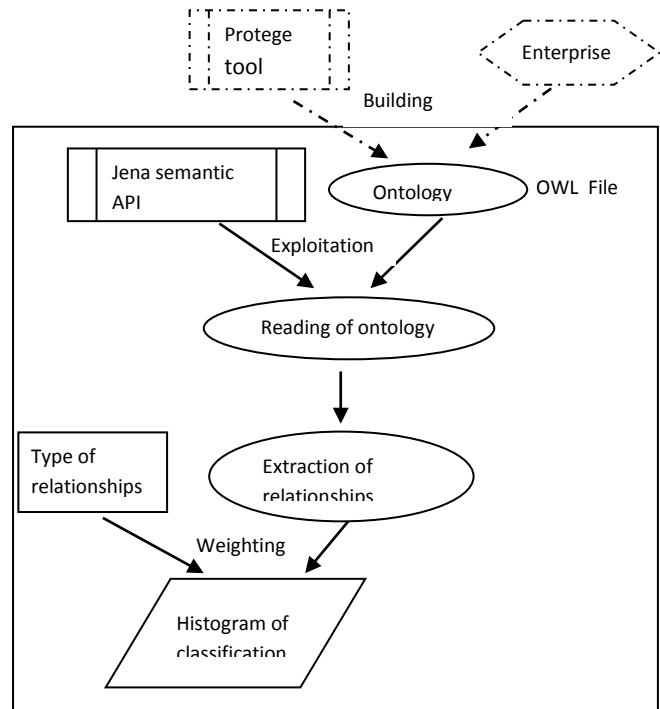


Figure.1. Proposed approach

III. APPLICATION TO ALGERIAN ENTERPRISE PRODUCTION

A. Specification of the enterprise

We apply the approach on a local manufacturing enterprise, involved in the transformation wiredrawing steel. The field of activity of the subsidiary is the production and sale of welded mesh panels, in rolls, concrete reinforcement and lattice girders [9].

The structural hierarchy of the enterprise consists of six departments (*commercial, , supply, maintenance, accounting, means and technical staff*) and two services (*quality control and industrial safety*).

The enterprise interacts with the outside world by nine entities: *customer, supplier, bank, EPO (port company), newspaper, court, wilaya, contributions and reservations*.

B. The adopted ontology

The first step in our approach is the ontology building. We consider the ontology for the interactions of the enterprise with the outside

world. This previously created ontology contains 29 classes including 9 relations with the outside world.

The nine relationships are:

- *R_secu* between Industrial safety service and wilaya.
- *R_appro* between Service of management stock and supplier.
- *R_achat* between the Sales service and EPO/ newspaper / bank / supplier.
- *R_compta* between Accounting service and customer / supplier/ bank.
- *R_manage* between Chief of management quality and customer/ supplier.
- *Appel_J* between Legal businesses and customer / supplier / court.

- *Ramene* between Sales force and the customer.
- *Reserve* between the Service of general means and reservations.
- *Cotise* between the Chief of social relation and contributions

After creation, we generate an OWL file and the figure 2 shows the hierarchy of classes and all interactions via the plugin OntoViz [10].

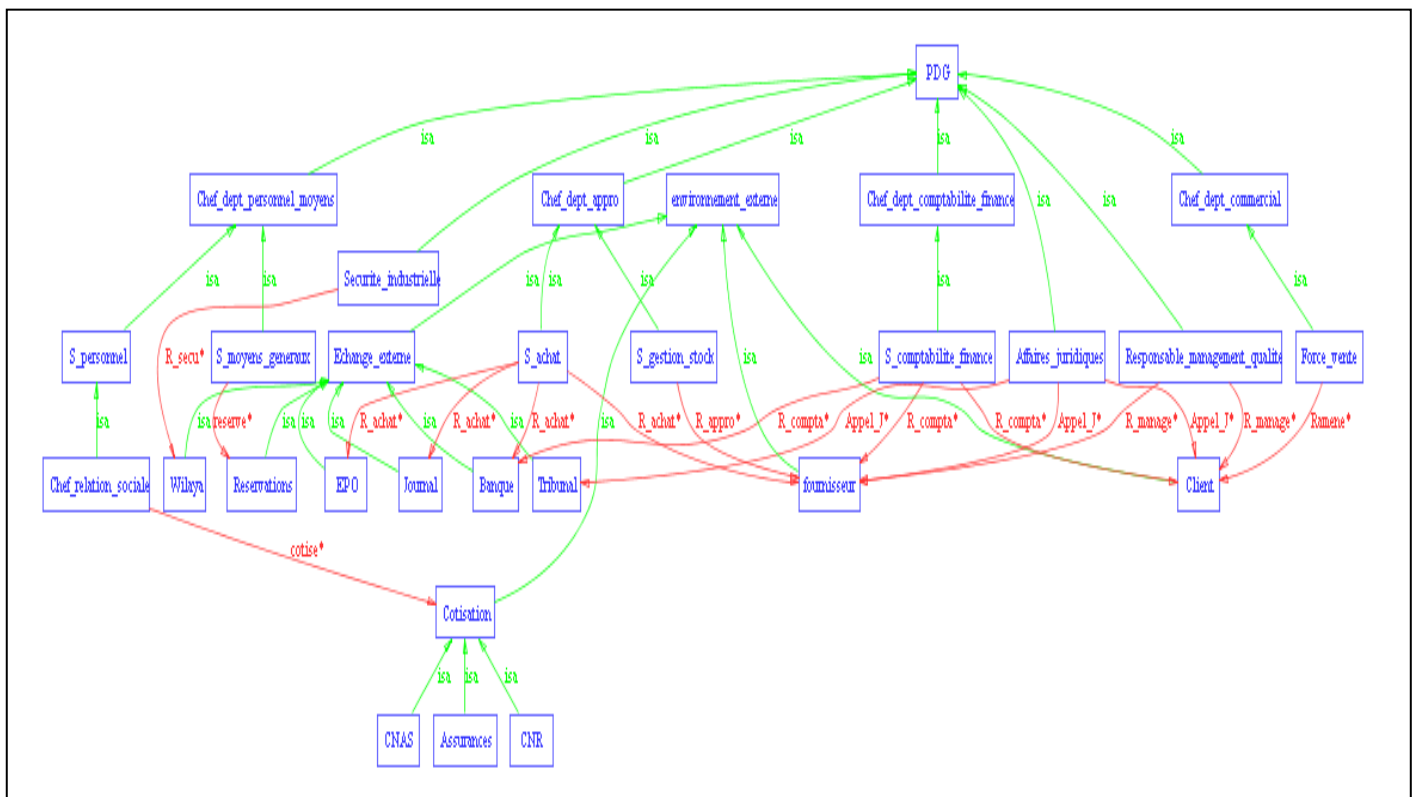


Figure.2. Interactions of the enterprise with the outside world ability to generate all existing owl file or set filters based on the three criteria outlined above: *subject / property / object*.

With:

The red color shows the hierarchy of classes (*isa*).
The blue color shows the class name.
The green color shows the existing relationships between the enterprise and the outside world.

C. Application of the approach

After integration of the Jena API in Eclipse programming environment, we read the OWL file in order to extract relationships.

We obtain a resulting set of Relations *R*:
 $R = \{R_1, R_2, \dots, R_n\}$, where each R_i is related as a record with three arguments as the three criteria: (*S, P, O*) with
- *S*: Subject referring to a class or a relationship.
- *P*: Property referring the type of subject.

This criterion can take three choices as $P = P [t, d, r]$ with

- *t*: type (relationship or class)
- *d*: domain (input entity of the relationship)
- *r*: range (output entity of the relationship).
- *O*: An object that is a class.

To view the principle of our approach for extracting relationships, we make two experiments.

1) *First experiment*: For a first practice, no filter (criteria) has been introduced, so we get all the classes and existing relationships in our ontology. This result highlights all the classes and relations of the ontology. (Figure.3).

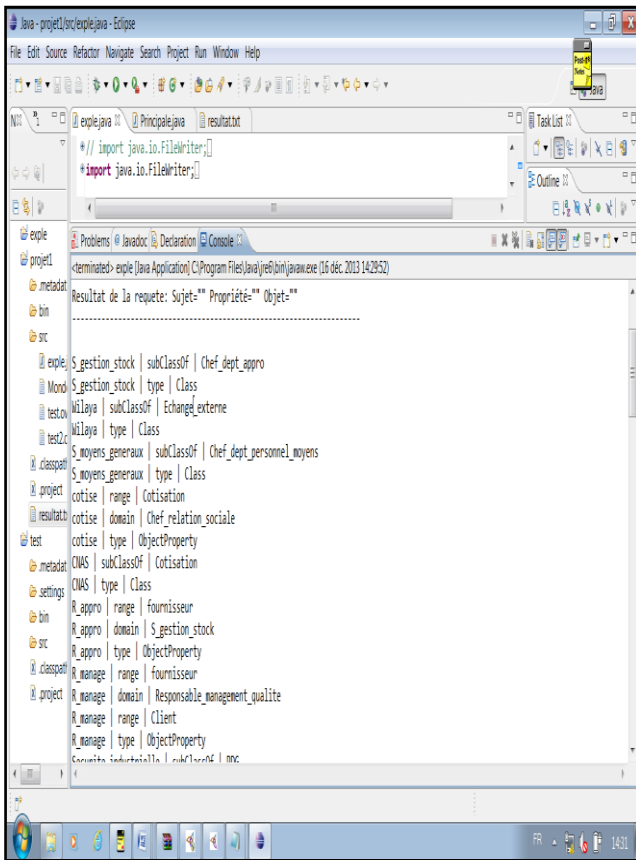


Figure.3. Reading and extract all informations of the ontology

For example, all the information of the relationship *R_appro* are displayed following the records:

- R_appro | range | supplier
- R_appro | domain | management stock service
- R_appro | Type | objetProperty

For clarity, we provide the ability to filter this result.

2) *Second experiment*: Based on the three criteria Jena (*S*, *P*, *O*), several combinations can be made according to one, two or three filters.

Our aim is to extract all relationships, so we choose as a filter criterion *P* (property). In this filter, there are three proposals on the choice of *P*: *t* (type), *d* (domain) or *r* (range). By choosing the domain, we have as output correspondence relations and classes of the form:

Relationship / domain / name of the class.

For example, the figure 4 lists for each relationship the input classes (domains).

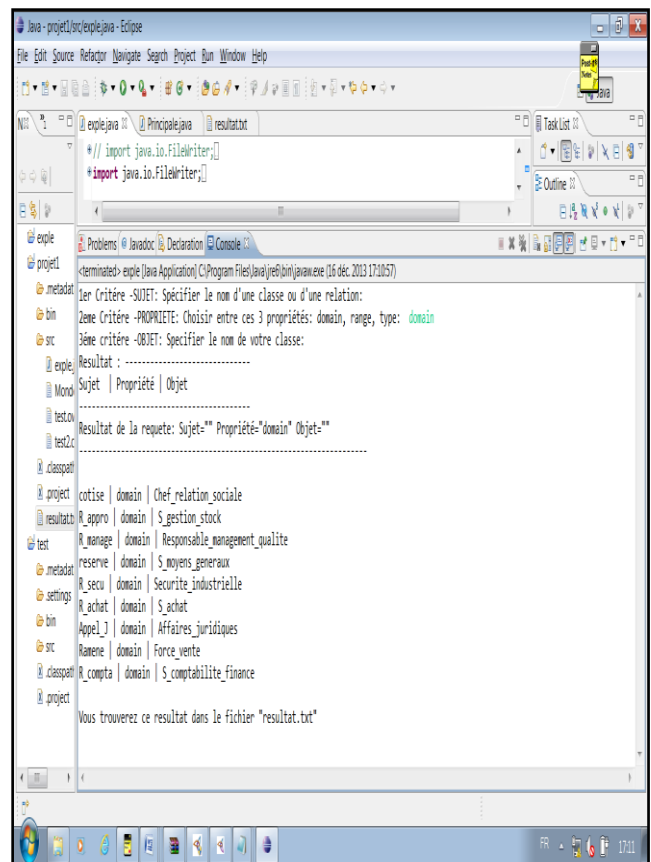


Figure.4. Generation of entities entered for relationships

So the relationship *R_appro* is refined to give: R_appro | domain | management stock service

This result illustrates the generation of input entities for each relationship of the ontology. Following the same process, we can generate output entities by replacing domain by range.

These experiments allowed us to exploit an ontology and extract information according to our needs.

D. Classification of relationships

In order to improve the enterprise performances, we classify the extracted relationships along a nature that is efficiency and effectiveness, through a respective weight of 1 and 2.

Since the considered enterprise is a manufacturing company, we assign the highest weight (weight = 2) for effectiveness. The final weighted relationship is obtained par the product of the weight by the number of relations. Thus, these relationships are ranked in a graph following an histogram form (Figure. 5).

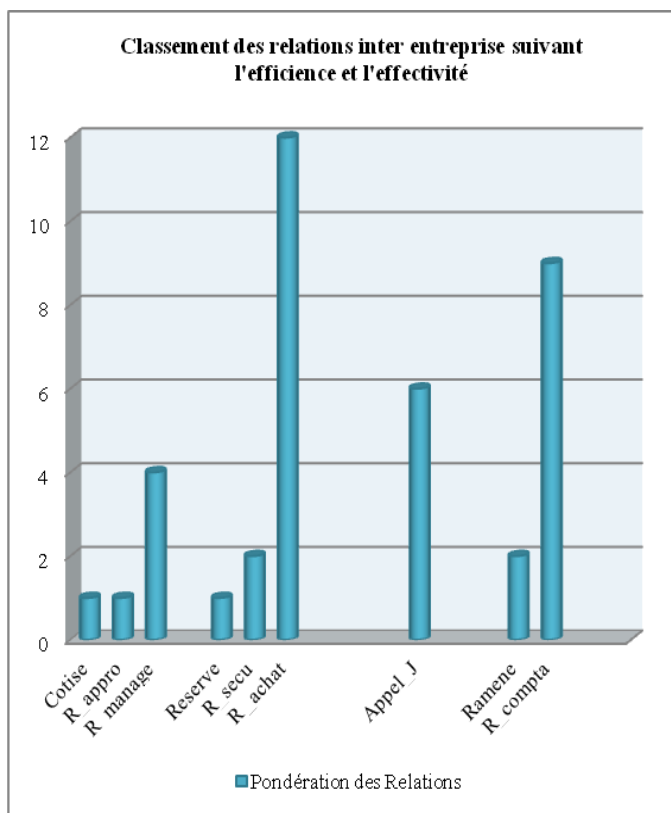


Figure.5. Classification of relationships on that the external relationship *purchasing* (R_Achat) is the most important and must be treated in priority.

Indeed in the manufacturing context, the relationships with the outside world are fundamental to ensure efficient production. For example, The accounting relationship is paramount to finalize the purchasing and also the relationship of legal businesses (Appel_J) for solving conflicts.

IV. CONCLUSION

We have presented in this article the exploitation of ontology by the Jena API for a classification of relations. We discussed the notion of ontology and the Jena API while detailing the proposed approach.

Our aim was to exploit ontologies already created to extract various information for later exploitation. After studying the few existing tools, our choice is leaning with the Jena API since this framework provides a number of Java classes dedicated to handling ontologies which are described in the Ontology Web Language.

The case study was done on a real example of an enterprise of transformation wirerawing steel. Following this study, we consider the ontology for the interactions of the enterprise with the outside world. So, we can read this previously created ontology and after that, we generated the information according to our needs like interactions. Extracted relations were classified according to their relative weight compared to the concept of efficiency and effectiveness in the context of production. The obtained rank allows giving priority relationships in this enterprise.

These preliminary results will be the subject of further studies to improve the enterprise performance.

REFERENCES

- [1] T. Gruber. "A translation approach to portable ontology specifications", Knowledge Acquisition, 5(2), pp.199–220, June 1993.
- [2] F.Fürst. "Contribution à l'ingénierie des ontologies : une méthode et un outil d'opérationnalisation", Ph.D Thesis, EPU-Nancy (France), November 2004.
- [3] The official website of JENA : <http://jena.sourceforge.net/>
- [4] The official website of JADE: <http://jade.tilab.com>
- [5] Tutorial of the tool protégé <http://protege.stanford.edu/>
- [6] H. Rajagopal. "JENA: A java API for ontology management", IBM Corporation, Colorado Software Summit, 23-28 October 2005.
- [7] V.Bronet. "Amélioration de la performance industrielle à partir d'un processus référent –Déploiement inter entreprises de bonne pratique", Ph.D. thesis, University of Savoie, September 2006.
- [8] V.Galdemar, L.Gilles, and M.O.Simon. "Performance, efficacité, efficience : les critères d'évaluation des politiques sociales sont-ils pertinents ? ", CREDOC, Working Paper Number 209, December 2012.
- [9] S. Hadj Tayeb and M. Noureddine. "Modélisation des relations d'entreprise par les ontologies. Étude de cas ", 8th International Conference Integrated Design and Production CPI'2013, CD-136, Tlemcen (Algeria), 21-23 October 2013.
- [10] Tutorial of du plugin OntoViz : <http://protegewiki.stanford.edu/wiki/OntoViz>