International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 6 – No. 1, September 2013 – www.ijais.org

OntoEval
Assessment Tool for OWL Ontology based Application

Bekka Fatiha
Computer Science Department
University Mohamed El-Bachir El-Ibrahimi
Bordj Bou Arreridj, Algeria

Maache Salah
Computer Science Department
University Mohamed El-Bachir El-Ibrahimi
Bordj Bou Arreridj, Algeria

Djaghloul Younes, Ph.D
Research & Development Engineer
CRP Henri Tudor
Luxembourg, Luxembourg

ABSTRACT
Ontologies are increasingly used in various fields such as knowledge management, information extraction, and the semantic web. Ontology evaluation is the problem of assessing a given ontology from the point of view of a particular criterion of application, typically in order to determine which of several ontologies would best suit a particular purpose. In this paper, we propose a pricing model that allows using a number of different ontologies and assessing their performance on specific issues. We suggest an approach for evaluating the ontology based on a golden standard guided by a question generator. Following this scheme, we present the results of an experiment to test and gradually increase the performance of ontologies.

Keywords
Ontology, ontology evaluation, knowledge representation, semantic web.

1. INTRODUCTION
In the field of knowledge management, ontologies are playing an important role and it's a research area of great importance. Different research areas are well defined, such as: languages representations, reasoning techniques, formal logic, and ontology engineering. The latter tries to make the process of creating ontology comparable to that of software development "software engineering".[1]Several issues are addressed in this context include processes and methods of ontology development, maintenance of ontologies, managing the evolution of ontologies, formal and human evaluation of ontologies. In this last area, the main goal is to propose mechanisms for evaluation of ontology in relation to its area rather than merely semantic validation according to the rules of validity of the language used or the associated logic description.

2. ONTOLOGY ASSESSMENT
As an ontology can be built humanly (experts, resource bases), or automatically by data-mining techniques, human evaluation provided an opportunity for experts to refine the ontology stating whether the axioms of the ontology. We found that even if the proposed ontology is valid, there may be cases where the developer does not understand the expert or resource bases. More critical in the case of the automatic generation of ontology, the technique used has not necessarily proposed an accurate ontology. The research question is: How can we evaluate ontology and how to support this automatically in an ontological engineering environment? Then how to allow an expert to evaluate the generated version ensuring that its changes will not break the consistency and validity of the ontology? Two constraints realize this task even more difficult: a) experts do not necessarily know the OWL or the logic behind b) experts can break the consistency and validity of the ontology if they have complete access to the ontology. In this paper, our work focused to propose an OWL ontology evaluation system and it is in order to make the validation of ontologies simple. The main objectives of this work are: 1) present a design for a prototype that allows the creation and manipulation of a model of ontology evaluation. 2) Present the technical aspects used for the implementation of the prototype and see a real application of our approach in a case study.

3. ONTOEVAL

3.1 Platforms of the approach
Two methods of evaluating ontologies are studied: the first is "Task Based Approach" with this approach; we can examine the ontology compared to their three basic levels:
1) Vocabulary, 2) Taxonomy and 3) Semantic relations. As these levels are also subject to different respective learning approaches, we propose that the common notion of error rates, such as found in word- or concept-error rates, known from previous work, suffices for each level of evaluation. In a task-based evaluation the results should show the following shortcomings: *insertion errors* indicating superfluous concepts, isa- and semantic relations, *deletion errors* indicating missing concepts, isa- and semantic relations, and *substitution errors* indicating off-target or ambiguous concepts, isa- and semantic relations. [2]. The second method studied in this work is an assessment of the OWL ontology approach system "Ontology Evaluation based on Question Generation from Model". The system creates questions from ontological elements to evaluate subsequently expert answers form the bases for the refinement process. Intelligent management of the sequence of issues will propose to optimize this process. [3] This will be integrated into the first part is based on the assessment tasks. The contribution of this paper is to propose a hybrid method that combines the two methods mentioned previously.

3.2 Critical & Objectives
Several approaches are proposed in the evaluation of ontology framework in the semantic web, such as semantic evaluation

International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 6 – No. 1, September 2013 – www.ijais.org

or assessment of the context or more based assessment data but many of them were not actually implemented. In the evaluation approach Task Based certain techniques remain difficult specifically implements the task [2].The objective of the proposed approach in the context of this paper is that we get a proper ontology (full vocabulary) by comparison to a reference (gold standard) and reliable (meets the requirements of the field) because it is evaluated by a domain expert. In order to achieve a proper proposal, must identify needs and target goals. It seemed important to identify a number of elements important in the assessment model:

• Support evaluation techniques based on the application for the purpose is the performance results based on existing ontologies(GoldStandard).

• The extension of this architecture for semantic integration issues based on a questions generator. Each list of questions contributes to the proposal of the solution overall evaluation using the feedback mechanism.

• To meet these needs, we propose to add the generator component issue that is implemented in order to enrich the ontology. For example we propose a biological ontology developed in 2010 but until 2013 it confirmed there's a lot of research and innovations in the field. With the feedback generated by the expert we arrive at a more efficient use rich ontology then.

3.3 OntoEval architecture

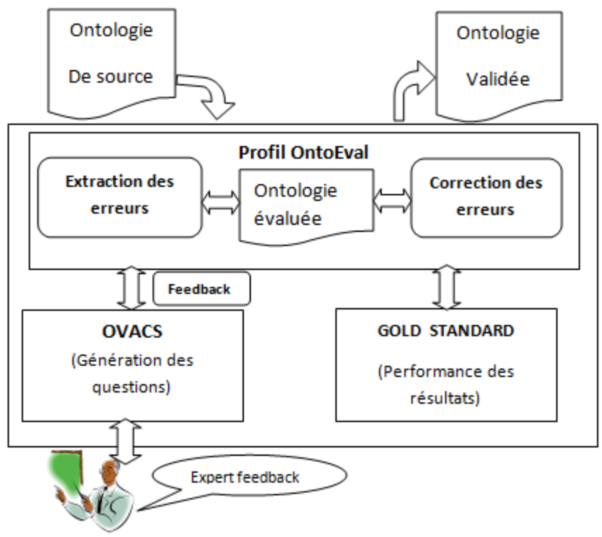

Fig. 1: OntoEval architecture

From the figure above, we note that the OntoEval model consists mainly of three modules, namely: the Golden standard, Builder issue, and Profile OntoEval.

1. Golden standard component is responsible for testing the contents of the ontology by applying the Task-Based approach, concepts and superfluous or missing properties are extracted. Recall that the Golden standard is responsible for performance results in the current platform.

2. The OVACS component is used to implement the compliance of the ontology with the recommendations of the expert, for example: even if there are redundant concepts how to remove them? And if there are missing concepts and adding them we will not necessarily integrate them properly, then the feedback from the expert valid instances, properties, subclasses, domain and range. It offers a questionnaire to be validated according to one expert, the feedback of the expert will be processed according to types of errors described below. (Table.1)

3. The OntoEval profile is the management interface of the proposed model. Based on ontology, this component allows the extraction and correction of semantics and processing requests errors. Other components can not directly access the profile; all requests are handled by the manager profile. The latter is based on a proper API that provides an update of the OWL ontology. In the next sections, we will describe each component separately, showing its internal structure and the dynamic aspect of scripts.

3.4 Similarity calculation

In many application areas, sequential patterns are extracted from the data to identify correlations. Evaluation of sequential patterns extracted to measure the proximity of these reasons and consequently identify consistency or inconsistency discovered by patterns extracted behavior. However, the evaluation of sequential patterns extracted is a difficult task because of the unique characteristics of sequential patterns. For many applications it is necessary to evaluate the proximity of sequential patterns. These comparisons are based on a similarity measure is one of the central concepts of data mining [4]. For a meaningful comparison, the similarity measure should be adapted to the characteristics of data. It also needs to scale application in another area. To evaluate the relationships between concepts defined by different ontologies, most measures assess this connection in terms of similarity. According to Resnik, the semantic similarity is an evaluation of the semantic relationship between two concepts with the objective to obtain an estimate of how close the meaning of these concepts. Most implementations use tables to one or two dimensions to store the objects being compared. In this case, it is the vectors that are used to pre-allocated and reused on different tracks of the algorithm on the treated objects. [5].

3.5 Extracting error

The extraction component errors are responsible for the task of semantic evaluation of ontological resources, namely:

• *Insertion error ()*: are the unnecessary concepts, properties and instances in ontology by comparing a reference and feedback by experts.
• *Deletion error ()*: are the concepts, properties and instances missing in the ontology compared to Golden Standard and feedback by experts.
• *Substitution error ()*: are the concepts, properties and instances placed differently in the ontology compared to standard Golden and test the expert properties.
• *Taxonomy error ()*: this is the feedback from the expert on the test *subclass of*. The answers of the expert will be processed as follows:

International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 6 – No. 1, September 2013 – www.ijais.org

• Response = yes, then no errors and no action to perform.
• Answer = no, the expert provided feedback that contains the correct answer; the error will be extracted following question.
• Answer = I do not know, (when using a computer term, the expert does not know necessarily OWL) the feedback must also contain a reliable answer for example I do not understand the question or I have no such information, in both cases the developer checks out the error in the feedback.

Level	Insertion	Deletion	Substitution	Taxonomy
1	irreverent concepts	omitted concepts	ambiguous concepts	Inconsistant Concepts
2	isa too coarse	isa too fine	isa too polygamous	isa too taxonomic
3	irreverent relations	missing relations	indirect relations	Inconsistant Relations

Table. 1: Overview of the errors on the respective levels

3.6 Correcting error

The last step is to correct errors and / or anomalies. Actions to be taken are: add, remove, replace, and change.
• *Add ():* missing concepts and relationships.
• *Delete ():* the concepts and relationships superfluous, which have no influence on the ontology.
• *Change ():* the concepts and relationships that are not in their proper location and / or ambiguous.
• *Replace ():* a relationship and / or a concept by another.

4. ONTOLOGY ENGINEERING

As part of our work we opted for the method MENTHONTLOY [6], [7] and more particularly the steps Support Activities and Technical Activities.

4.1 Conceptualization

In this phase, the objective is to provide conceptual ontologies called Generation “generation” and Family “Famille”. They are based on the following intermediate representations: a glossary of terms, the taxonomy of concepts, the diagram of binary relations, the dictionary of concepts and tables of binary relations. The ontology Generation is implemented in the evaluation framework as well as gold standard. The ontology Famille has been proposed in the context of comparison, it is a simple and clear presentation on a small family.

4.1.1 Construction of dictionary concepts

The glossary of terms lists is all the useful and potentially used terms in the field of study with a description. To identify all the terms, we have exploited the work done in this area. Most related to the description of resources is identified from the web. The Table 2 presents an extract from the list of concepts.

Nom de concept	Terme en anglais
Personne	Person
Homme	Man
Femme	Woman
Grand parent	Grand parent
Parent	Parent
Progéniture	Offspring
Sibling	Sibling
Sexe	Gender

Table. 2: Extract from the list of concepts

4.1.2 Taxonomy of concepts

The hierarchy of concepts will merge the concepts and have modularity in the domain knowledge. Different taxonomies of concepts are presented in the following figures.

Fig. 2 : Generation Taxonomy

International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 6 – No. 1, September 2013 – www.ijais.org

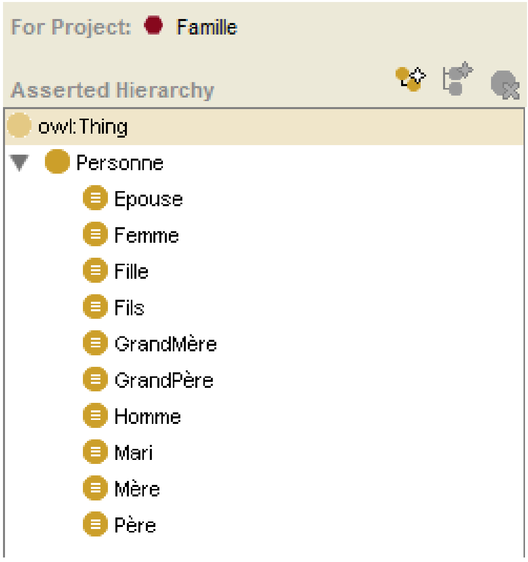

Fig. 3 : Famille Taxonomy

4.1.3 Description of Object Properties

The properties exposed here form themselves a taxonomy that can be hierarchy[8], take a simple example of object property Has Child "A enfant" is general it can be specialized depending on the type of classes even Parent or Grand-Parent and the hierarchy may continue after the specialization of each relationship. Fig 4 and Fig5 show an extract from the list of Object Properties of ontologies Generation and Family respectively.

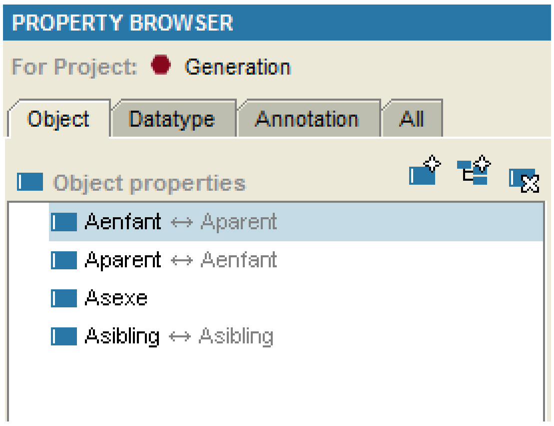

Fig. 4: Extract from list of Object Properties Generation

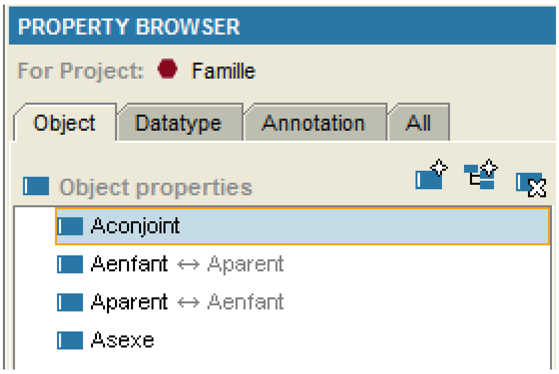

Fig. 5: Extract from list of Object Properties Generation

For each property must be defined the Domain and Range and Property Characteristics is Transitive, Symmetric, Functional, or Inverse Functional. The figure 6 below shows an example of Object Property from ontology Generation witch is Has Gender "A sexe" link between Person "Personne" (Domain) and Gender "Sexe" (Range) with characteristic Functional.

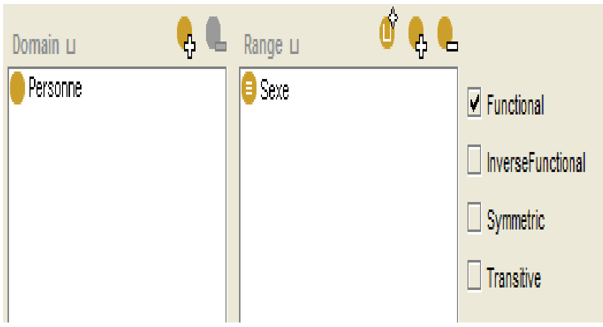

Fig. 6: Domain and Range and Property Characteristics of the Object Property "A sexe".

4.1.4 DataType Properties

After defining the classes and their Object Properties of our ontologies, we must describe the internal structure of classes. An excerpt from the list of properties DataType is shown in Fig 7.

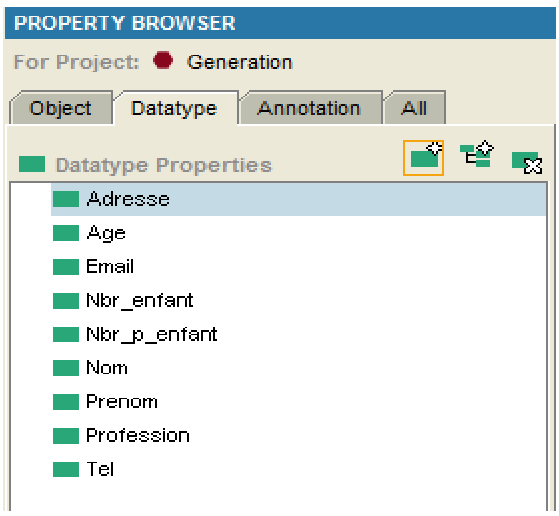

Fig. 7: Extract from the list of DataType.

4.1.5 Individuals

The last step is to create instances of classes in the hierarchy. Define an individual instance of a class requires 1) choosing a class; 2) creating an individual instance of this class; and 3) enter property values. For example MICHOU, SAM, NABILA are individuals of class Person "Personne", an extract from the list of individuals is shown in Fig 8.

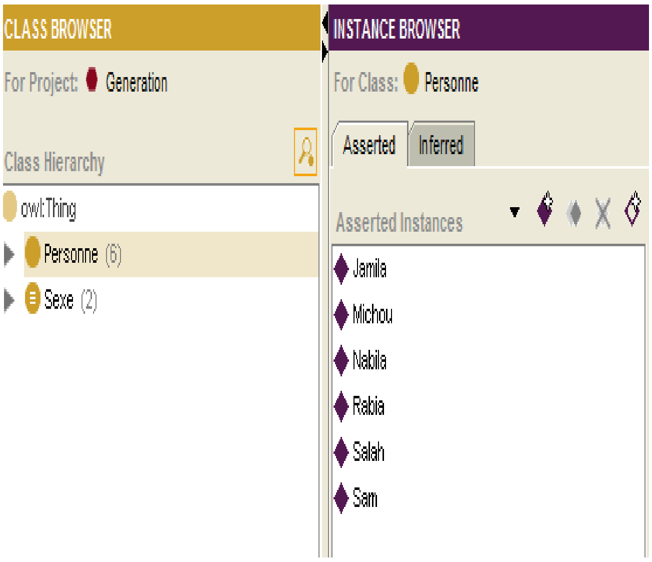

Fig. 8: Extract from the list of Individuals from Generation.

4.2 Implementation

One of the major decisions in the process of developing ontologies is to choose the language of representation. Our ontologies are designed for use by applications that need to process the content of information instead of just presenting information to humans. For this reason the developed Ontologies are implemented using Web Ontology Language (OWL) [9] that facilitates greater machine interpretability of web content than supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics.

Other reason OWL is the most recent development in standard Ontology languages endorsed and recommended by the World Wide Web Consortium (W3C) to promote the Semantic Web vision. We build the ontologies using Protégé-3.4.4 as ontology editing environments.Protege3.4.4 (http://protege.stanford.edu/) is a free, open source ontology editor and knowledge-base framework. The Protégé platform supports two main ways of modeling ontologies via the Protégé-Frames and Protégé-OWL editors. Protégé ontologies can be exported into a variety of formats including RDF(S), OWL, and XML Schema. Protégé is based on Java, is extensible, and provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development. [10].

4.3 Evaluation of ontologies

We use PELLET 1.5.2 inference engine to test our ontologies. It is designed for reasoning on description logics and accepts as input an OWL file. The main services offered by PELLET 1.5.2 are: consistency checking (satisfiability, coherence) and the classification test (subsumption).

4.3.1 Consistency Checking

The test of consistency provided by PELLET is performed based on the class description (conditions). It ensures that no definition of a class conflicts with another (the absence of conflicting classes). The result of the test is shown in Fig 9 and Fig 10.

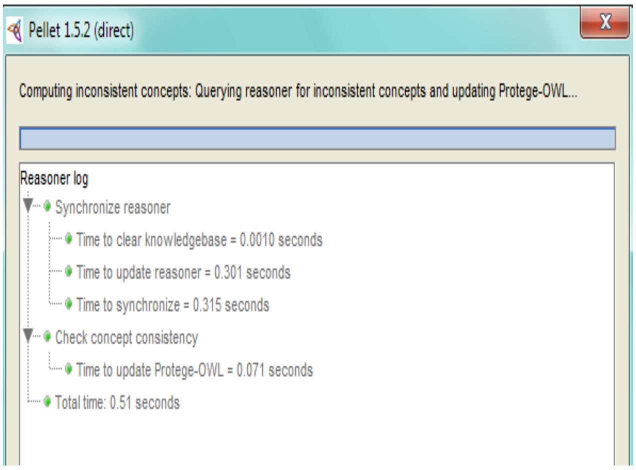

Fig. 9: The test of consistency results of Generation.

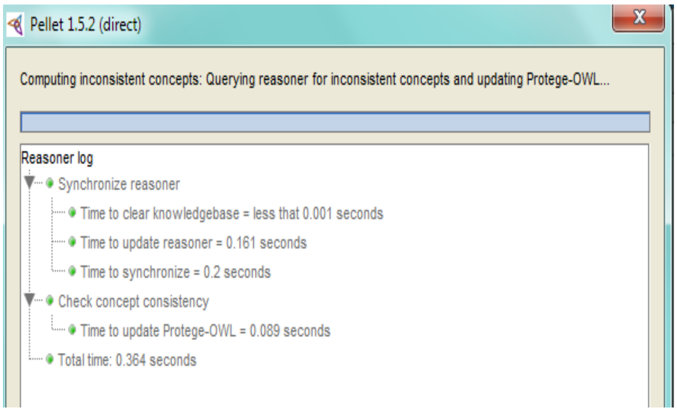

Fig. 10: The test of consistency results of Generation.

4.3.2 Classification Test

The classification test to check if a class is a subclass of another class or not. Once the classification test is performed on the class hierarchy containing the logical expressions, it is possible for the classifier to infer a new hierarchy «inferred ontology class hierarchy» which is a hierarchy where classes are classified according to the relation super class / subclasses. The result of this test is shown graphically by Protégé-OWL in Fig 11 and Fig 12.

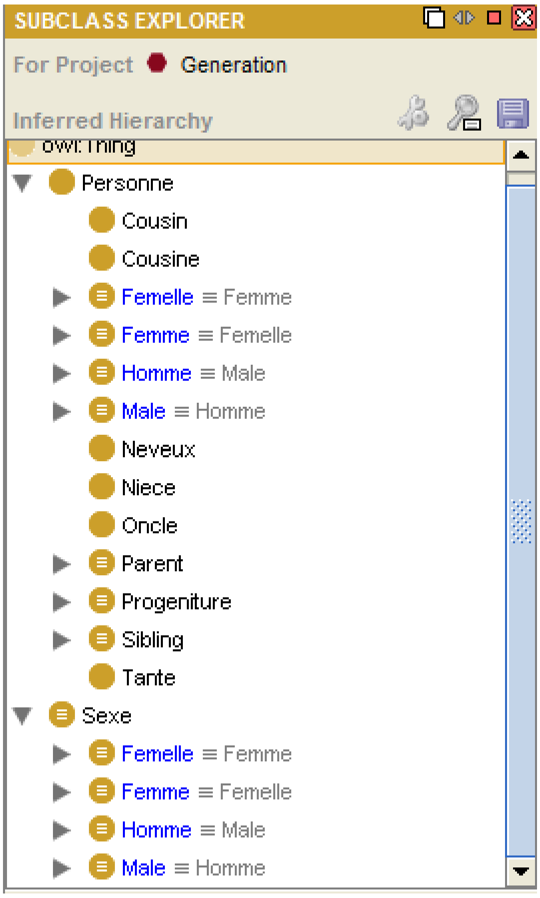

Fig. 11: The test of Taxonomy results of Generation.

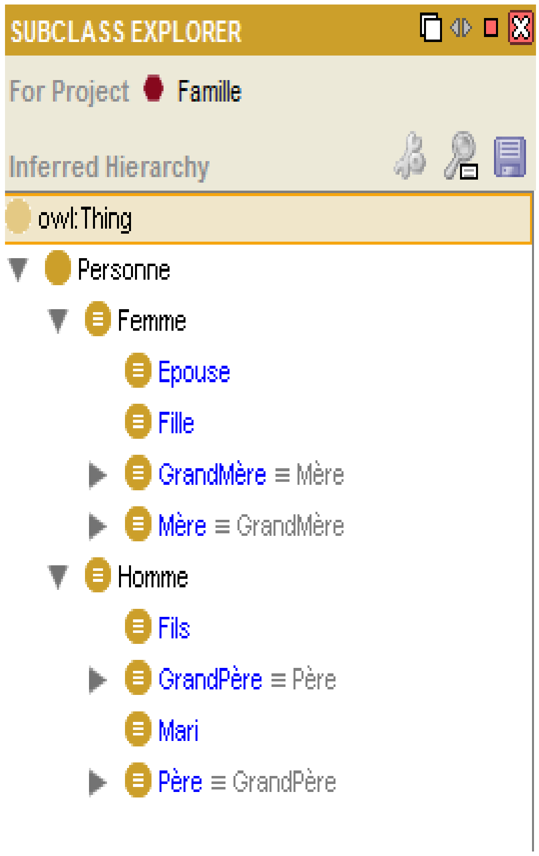

Fig. 12: The test of Taxonomy results of Family.

5. ONTOEVAL IMPLEMENTATION

The transition between the phases of modeling and theoretical reflection to the final code of the application is the culmination of the work. It is very important to make a wise choice of technologies to use and programming techniques most appropriate to the context. This implementation also allows to validate the proposed applying it in an appropriate case study contribution. In this section, we focus on the implementation of the approach in a case study.

5.1 OntoEval Interface

Fig. 13: OntoEval API

The main features of the application are:

• The use of an ontology Generation as extraction and evaluation errors infrastructure. Based on the golden standard approach, our approach offers the possibility of navigation between different ontologies has treated significantly to know the possibilities of research on these ontologies.

• The extraction of the mistakes we propose module allows for an evaluation based on performance results system. This solves the problems of semantic heterogeneity while allowing a constant evaluation of the ontology.

• The mechanism that helps ontoeval different users (experts and developers) in their choice of ontologies to assess and helps refine the proposed semantic model in this work. For this we have developed a special application called OntoEval (figure13). The latter is an implementation of our theoretical contribution. To understand the overall operation of the application, we give a sketch of a single study which encompasses the main operations of the application.

5.2 Extraction errors algorithm

The similarity calculation is based on the proposed model. The elements necessary for the calculation are the vectors of objects. In our case, we made a request satisfaction based on three parameters: the golden standard, the ontology to evaluate and question lists. For each ontology, we propose the following values: *insertion error rate*, *deletion error rate*, *substitution error rate*, and they are immediately removed from the ontology. *Calculer_erreur ()* is the method used. The latter invokes the golden standard to return the value of the extracted errors.

6. EXPERIMENT RESULTS

The error values will be used in order to refine the initial ontology. It has to invoking the error correction module (Section 3). Extraction an error is to remove all objects that do not meet the criteria of evaluation module namely the golden standard (implemented as part of the implementation ontoeval) and the generator question (implemented in the near future). The values are extracted by applying the methods mentioned. The final list will be sent to the graphical user interface for reuse.

7. CONCLUSION

In this paper, we propose an evaluation protocol of semantic classes to address the problems cited in the main issue of this work. Our proposal aims to introduce a new evaluation method based on the comparison with a golden standard and guided by a generator of questions to a domain expert. The originality of our work lies in having coupled metric studied (Task Based Approach and OVACS.) In addition to the effort devoted to the state of the art and classification research in this area, our work makes its own contribution, which can be summarized in three aspects:

The proposal of semantic profile for evaluation based on the concept of evaluation based on the application, the profile provides a unified view for the evaluation, it provides a rich description that is based on the semantic power description logics and OWL. The profile is created through a process of building ontology adapted based on a collaborative approach; The extension of the architecture ONTOEVAL with the questions generator. We have proposed the addition of generator issues that ensures care treatments associated with OVACS component method; This component provides two major services, namely, extraction errors in the ontology, and the correction of defects by a domain expert resulting eventually enriching the ontology.

Future works in this area should focus in particular on the evaluation of the ontology automatically, which is necessary for the proper approach to develop automated processing ontology for a number of technical problems condition, such as ontology learning, population, evolution, and so on.

8. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of the template especially Mr. Salah Maache that accompanied my first steps in the world of research. A special thank to Mr. Djaghloul Younes, who gives me advice and support.

9. REFERENCES

[1] JANEZ BRANK, MARKO GROBELNIK, DUNJA MLADENIĆ. “A survey of ontology evaluation techniques”. Department of Knowledge Technologies Joseph Stefan Institute.2005.

[2] PORZEL, R., MALAKA, R. “A task-based approach for ontology evaluation”. ECAI 2004 Workshop Ont. Learning and Population.

[3] YOUNES DJAGHLOUL, MURIEL FOULONNEAU, RAYNALD JADOUL, NABIL ZARY. “From informal knowledge to High Quality e-Assessment process: The OAT Semantic Approach”. Center of public research Henri Tudor.2012.

[4] MOEN, P. “Attributs, Event Sequence, and Event Type Similarity Notions for Data Mining”. Ph. D. thesis, University of Helsinki, Finland, 2000.

[5] RESNIK. “Semantic Similarity in a Taxonomie: An information-Based measure and its application to problems of ambiguity in natural langage”. Journal of Artificial intelligence, 1999.

[6] FERNANDEZ M., A. GOMEZ-PEREZ, J. PAZOS, AND A. PAZOS. “Building a chemical ontology using methontology and the ontology design environment”. IEEE Intelligent Systems Applications, 1999.

[7] GOMÈZ-PÉREZ A. “Ontological Engineering: A state of the art”. Expert Update, Volume 2 (3), Pages 33-43. 1999.

[8] HASSINA BENSEFIA. “ONTOLOGIES EN INTELLIGENCE ARTIFICIELLE». UNIVERSITÉ DE MOHAMED EL-BACHIR EL-IBRAHIMI. 2012.

[9] M. USCHOLD. “ONTOLOGIES: Principles, Methods and Applications”. Knowledge Engineering Review. 1996.

[10] MAACHE SALAH, TOUAHRIA MOHAMED. “DEVELOPING ONTOLOGY FOR FINANCIAL INVESTMENTS: ALGERIA CASE STUDY», INTERNATIONAL JOURNAL OF COMPUTER APPLICATIONS,(0975-8887),VOLUME 24-N°4,JUNE 2011.