Semantic Representation of Experiments

Master MIASHS WIC

2019

| Supervised project |

Project Tutor: Jérôme Euzenat
Academic Tutor: Manuel Atencia

Students:

Avae Jimmy
Couret Robin

Table des matières

INTRODUCTION

This report is the description of work realized in TER (Study and Research Work) project whose theme is the semantic representation of experiments. It is a part of the WIC (Web Computing and knowledge) formation taught Université Grenoble Alpes. The project was proposed by mOeX team working in INRIA laboratory. We will begin to describe the thematic, technology and the experiment to introduce the issues. Then we will develop our response to this problematic, to conclude with the ameliorations.

Context

The description of experiments is in line with the semantic web. Firstly, we propose to explain this concept. To begin with, the observations are that in the research workflow there are collections of information. They are produced, storage in different forms and manipulated in a specific goal. The set of data represents a base of knowledge. There are different ways to request a part of information: manual research with human memory and exploration, information retrieval methods, organization in the relational databases followed by SQL request. Those solutions can be useful, but with that simple structure, the data can't be really analyzed by computers. However, it is possible to represent this knowledge with intelligence, it is about to structuring the data with their conceptual relations. Here is introduces the concept of ontology, define as*: "it encompasses a representation, formal naming and definition of the categories, properties and relations between the concepts, data and entities that substantiate one, many or all domains of discourse." (Wikipedia, Ontology) Also the* interest of ontology uses is to give a computer access at the knowledge. Indeed, with a semantic representation the computer can know the link between information and it can clever process the data.

Technologies

To operate the semantic Web needs technologies. Therefore, for the purposes of developing a semantic computing standard are developed. The mains are: specific data model to organize elements of data (Resource Description Framework - RDF), language to represent knowledge (Web Ontology Language - OWL) and a query language to RDF databases (SPARQL).
RDF uses a concept of triples to represent data, that means that each expression has the form subject–predicate–object, the subject represents a resource, predicate it is a relationship between subject and object.
OWL is used to describe ontology in RDF with different axioms.
*"Individuals in the OWL 2 syntax represent actual objects from the domain. Classes can be understood as sets of individuals. Datatypes are entities that refer to sets of data values. Object properties connect pairs of individuals. "(*W3C, 2019)

To store RDF data, there is a specific storage technology nait ismed triplestore:

"A triplestore or RDF store is a purpose-built database for the storage and retrieval of triples through semantic queries." (Wikipedia,Triplestore)

The use of a triplet store has many advantages. There is no predefined schema, so it is easy to modify the data model. RDF can be imported or exported using standard N-Triples formats. Users are not subject to a specific implementation and can therefore change providers without problems. Triplestores can be queried using the SPARQL and can easily handle complex queries. It is easy to share data through the use of URIs. When combined with ontologies that formally define the objects and their relationship types, triplestores support inferencing that enables the discovery of implicit facts and relationships.

Experiment and workflow

With that as background, let's now look how this concept is part of project. Actually, the description of experiments is an important part of experimentation workflow. These experiments are an evolution of the knowledge of a set of agents through game simulation. This knowledge is measured and described in reports in the form of html pages. To ensure the validity of the research work these experiments to be reproducible. Thus *"the expected benefit of a such description is to be able to manipulate the descriptions of experiments in order to compare or verify that they measure well the assumptions that they are intended to test." (Euzenat, J.)* Therefore, the main purpose of the project is to describe experiment as web semantics.

Having examined the subject, to develop a project we need collaborative effort between team and customer to meet final user needs. The first process is to study the end user and his workflow.

- **End user:** He is a MOEX team researcher using LazyLavender, an agent simulation game, to perform experiments. He is familiar with the software and uses a wiki to publish his results. This wiki allows him to find the experiments done previously.

- **Initial user story:** To make an experiment the scientific builds a hypothesis. To do that he works with old experiments. In this case it is with a wiki (https://gforge.inria.fr/plugins/mediawiki/wiki/lazylav/index.php/), where the information is disordered and not structured. After he designs the experiment then executes it. Finally, a set of data representing the experiment is created.

With information about workflow and for the purpose of integrate semantic in it, the development choices are to translate experiment into RDF and the possibility to query in a triplestore. It seems that a variety of tools need to be developed to this end. First an ontology which describe experiment, a translator which transform legacy data in knowledge graph and the graph implementation at the triplestore.

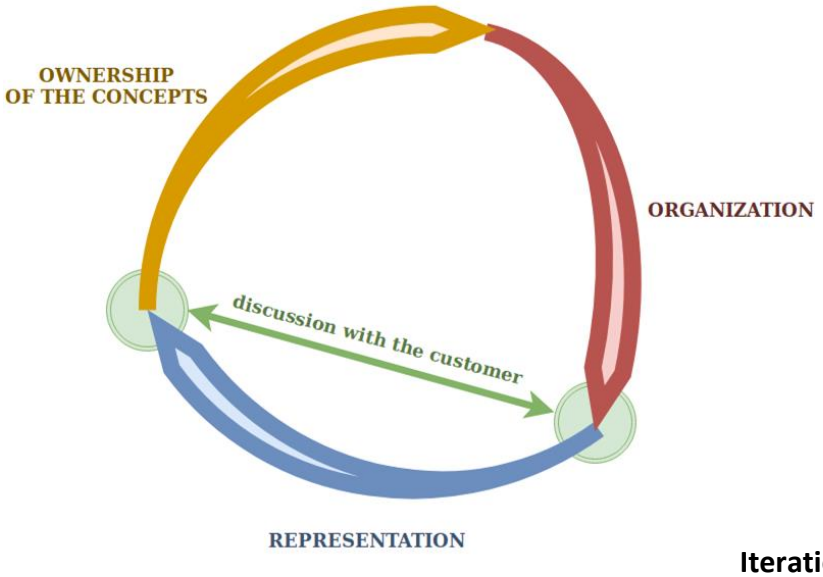
The experiment ontology

Specifications

After having studied context and from needs assessment we conclude that specification about the experiment ontology:
"It is a question of modelling two bash scripts and their metadata in RDF form. This requires an exploration of existing ontologies and a work of abstraction, structuring and prioritization of the content of the experience." (Translate from specification document)

Development

Develop an ontology is a long process. We used a different step (below) to develop the Experiment Ontology, the modelling is a work with multiple iterations with trial and error strategy.



Iteration process

1. **Understanding of the concepts** of semantics concepts: we extract and classified the concepts relating to the experiment's reports, it is a human heuristic approach.
2. **Organization** of concepts: we make relation existing between concept, and we construct a taxonomy.
3. **Knowledge representation:** we use graphic visualization or Web Ontology Language (OWL2) recommended by the W3C, in this step we integrate the description logic.

We work with successive customer feedback. It is important to develop a product with agility. The different modifications in the ontology are related by discussion with the client.

Three times can be distinct in the ontology development. The approach time where the semantic and experiment concepts are manipulated for first time with faults and approximation.

The EXPO time where an existent ontology is used, then abandoned later. And the final time where the main concepts of experiments are well-known and the development of ontology is completed. The roadmap below describes with details the multi-iterative process.

RoadMap of ontology development

In the first iteration, we need a long step to understanding the concepts.
Syntheses of the first acquisition:

- Tools list (link to Bitbucket)
- Comprehension notes : (link to Bitbucket)

Part 1 :

Iteration (link to Bitbucket)	**Comments**
Iteration 1 : OWL	Very simple, first approach.
Iteration 2 : OWL Iteration 2 : Visual	Extracting concept about experiment reports. Very bad Logical Description: no properties used.
Iteration 3 : Board visual	Previous ontology with properties. Wrong use properties

Part 2 :

At this stage the customer wants use an existent ontology, we select the only ontology which describe completely an experiment.

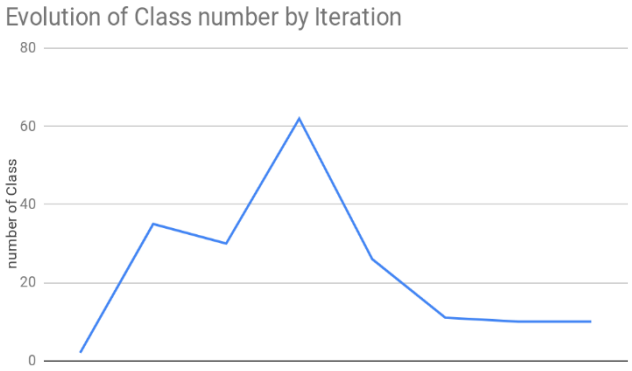
"The aim of EXPO is to abstract out the fundamental concepts in formalizing experiments that are domain independent."
"The vision of 'e-Science' is to publish online both papers and all of the data and metadata from a scientific experiment for posterity; so that all results can be repeated and compared with other related experiments." (Soldatova, 2006)

Iteration 4: OWL	Select the concept and relation interesting in EXPO to represent an experiment.
Iteration 5 : OWL Iteration 5: Visual	Hybrid Ontology between our modeling and EXPO.
Iteration 6 : OWL Iteration 7 : OWL	Simplification, deleting object.

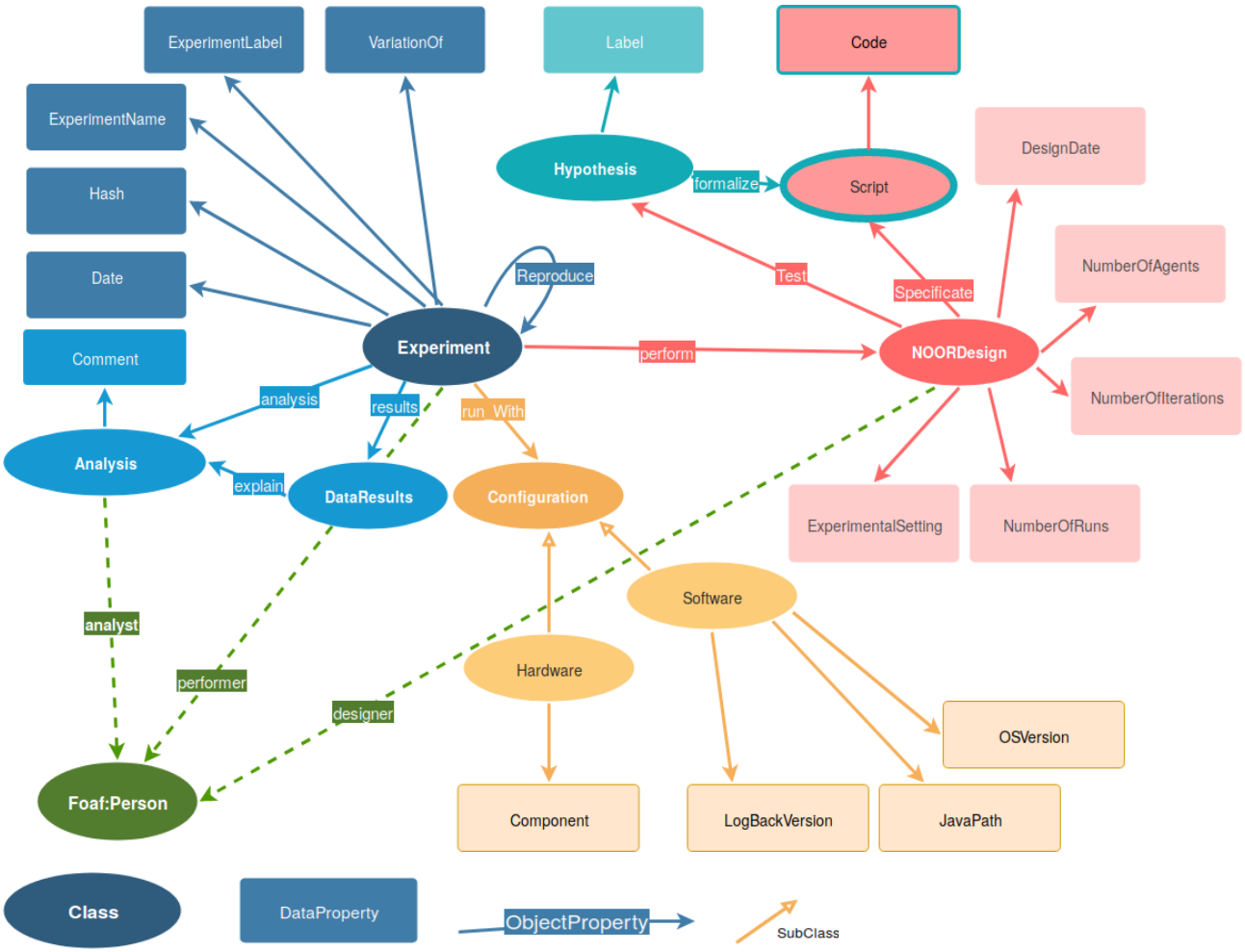
Part 3 :

After a meeting with the customer, he analyses with precision the EXPO ontology and he disapproves the philosophy of modeling. His analysis is that the kind of the use Object Properties (has_Part and has_Attribute) is incorrect.

Iteration 8 : OWL	Use basic concept and relation
Iteration 9 : OWL	Simplification
Iteration 10 : OWL Visual : below	Complete and final Ontology



The curve describing the evolution of number Class is interesting to represent the development. In fact, at the beginning we have just few poor objects, at the middle we have too exhaustive logic description and at the end we have precise and relevant class and relation.



Ontology describing Lazy Lavender experiment.

This visual graph is efficient to describe the ontology. The different Class represent the main concepts content in a Lazy Lavender experiment. The data properties are information content in experiment report file. They are jointed around concept. And the Object Property describe the relation between concepts. Finally, all Lazy Lavender experiment can be represented by this model.

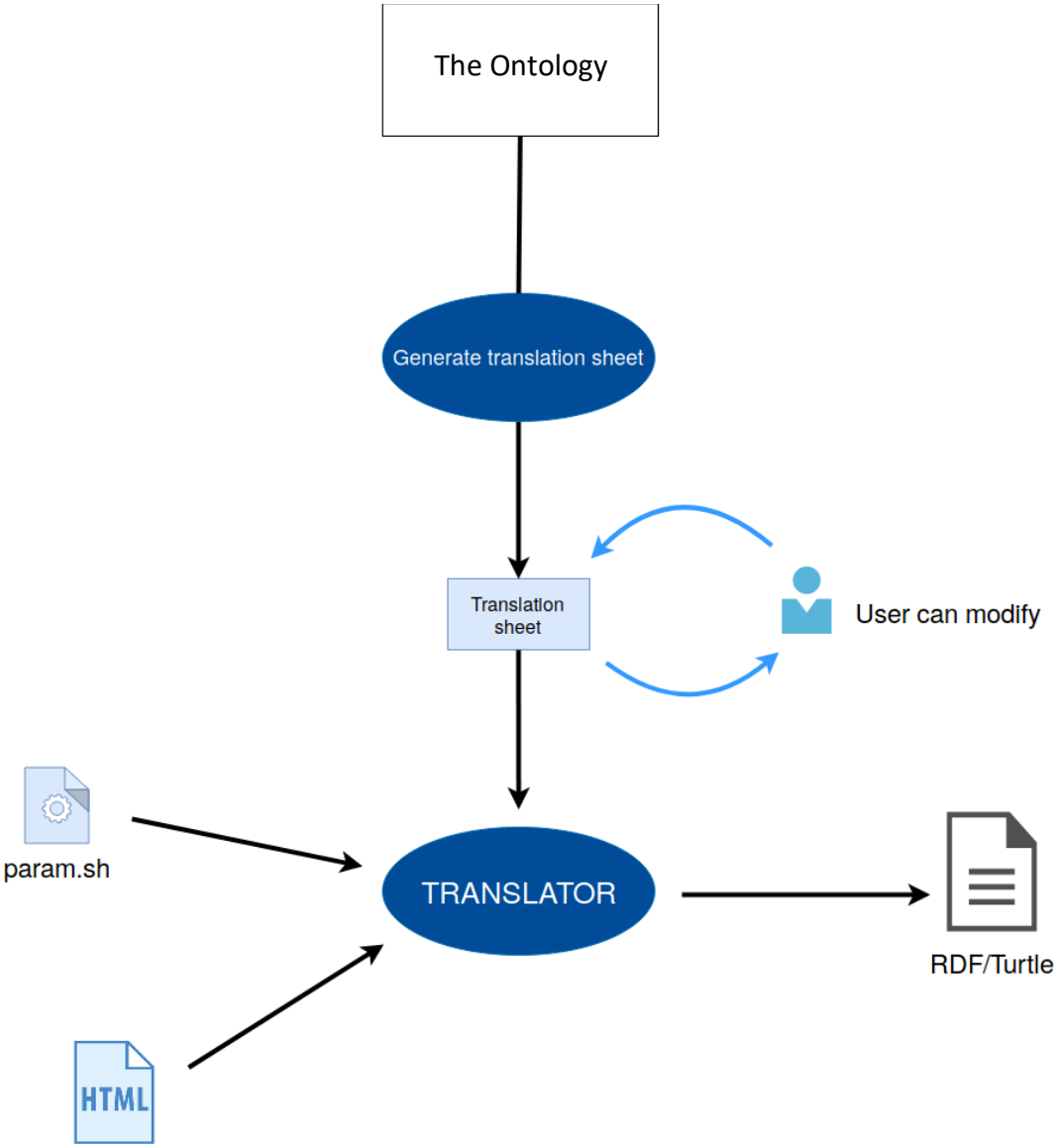
It is possible to observe how that ontology is applicated: (see annex: Turtle)

Possible improvement

A possible evolution of our ontology is the description of "scripts" with semantic. Each experiment is executed by a script. The script loads the parameters that can perform actions for each operation (delete replace refine add addjoin refadd) ...

For this purpose, we have studied a design pattern ontology (See annex: design pattern)
This ontology could describe these actions, performed for our operations. https://sparontologies.github.io/pwo/current/pwo.html

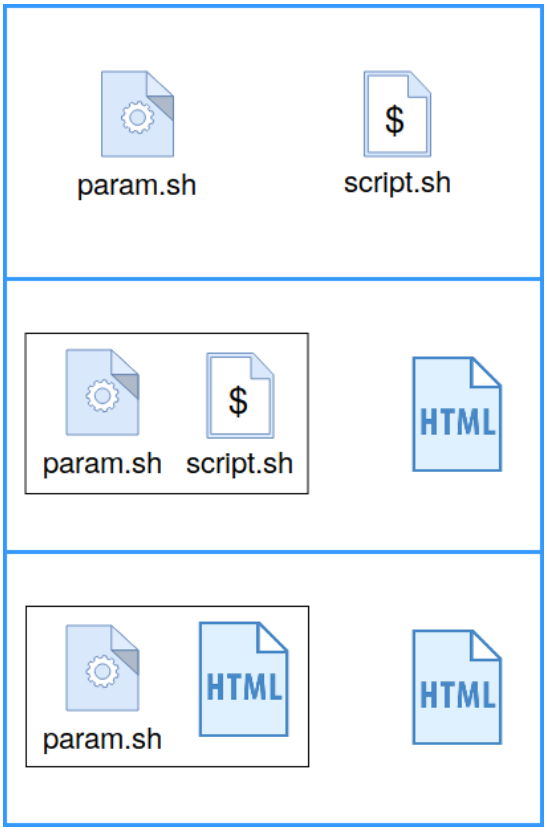
The ontology is used to describe knowledge; however, the raw data need to be transformed to represented by knowledge graph. For that we developed a translator using a Java OWL API, i.e. a program which parses HTML and Bash data to generate a turtle data file describe in RDF format, and with a logic description in OWL2 file.



General Schema of Translator

Different parsing

To cover all the experiences, we have to adapt the parser according to the different formats of the reports. Here are the different formats:



For the first iteration our translator takes a params file (see annex: Params.sh) and a script file as input. These files were generated from the most recent experiments, but did not cover all of them.

This is why in the second iteration our translator took in more params and script files, the html pages themselves. These pages are generated during the execution of each experiment, they cover all of them. The parsing was done separately as shown in the diagram on the left.

During the last iteration, we can choose between either to translate the data via the html files or to mix them with those of the params. The script file does not contain additional data from the html file (see annex: html) so we take it into account more.

Schema of the different inputs

Association of Data

The match between our ontology and the parsed data needs a specific text file. This one, called “Association File” make a junction between OWL axiom and data. At the left of the file, there are the data or object’s properties - generated automatically by the translator, at the other side there are the corresponding name of the data - added by humans.

```
http://www.inria.fr/moex/ExperimentOntology#label=Hypothesis|Hypotheses|HYPOTHESIS
http://www.inria.fr/moex/ExperimentOntology#javaPath=JPATH
http://www.inria.fr/moex/ExperimentOntology#logbackVersion=Full log
http://www.inria.fr/moex/ExperimentOntology#OSVersion=Execution environment|OSVERS
http://www.inria.fr/moex/ExperimentOntology#code=Script|Command line
```

The links are fakes.

Data Output

The translator generates turtle files. Turtle is an alternative to RDF/XML, it is generally recognized as being more readable and easier to edit.

```
xpd:Experiment_20140204-NOOR a owl:NamedIndividual , xp:Experiment ;
    xp:performer xpd:Person_JEuz ;
    xp:performs xpd:NOORDesign_20140204-NOOR ;
    xp:reproduces xpd:Experiment_20140204-NOOR ;
    xp:results xpd:DataResults_20140204-NOOR ;
    xp:variationOf xpd:Experiment_20140204-NOOR ;
    xp:experimentationDate "2014-02-04" ;
    xp:hash "769936317dba7a3a3d7294155ef333e12f5d905a" .
```

Turtle Example

As we can see above, the translation was well done. We can see different triplets and read data such as the title of the experiment. These generated files are stored in a specific directory.
From all these generated files we will be able to import them into a triplestore and make requests. The benefits of having our data in a triplestore are many and varied.

Triple Store

The choice of triplestore

We were recommended to choose Jena Fuseki server as a triplestore. “Apache Jena Fuseki is a SPARQL server. It can run as an operating system service, as a Java web application (WAR file), and as a standalone server. It provides security and has a user interface for server monitoring and administration.” (Apache Jena Fuseki, 2019)

SPARQL queries

Thanks to the triplestore and the sparql requests we can retrieve information about our experiences easily.
All experiences derived from a specific experience:

```
SELECT ?Experiment ?DerivedExperiment
WHERE {
<http://localhost:3030/Tests1/data#Experiment_20180601-NOOR> xp:experimentLabel ?Experiment.
?xp1  xp:variationOf <http://localhost:3030/Tests1/data#Experiment_20180601-NOOR>.
?xp1  xp:experimentLabel ?DerivedExperiment.
}
```

Experiment	DerivedExperiment
"20180601-NOOR"	"2018-09-15-NOOR"
"20180601-NOOR"	"2018-08-29-NOOR"
"20180601-NOOR"	"2018-08-28-NOOR"

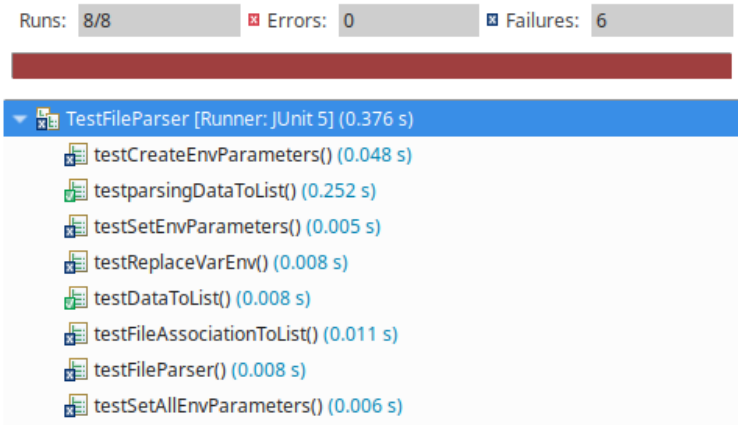
Example of queries and its result (see more queries in annex: query)

This is one of the many interesting queries possible, you can see more in (See annex: query). We performed some tests and measurements on our data for proven / verified their utilities and reliability.

Test and reports tests

Unit Tests

A unit test is used to test the correct operation of a specific part of a program. It makes sure that the behavior of an application is correct. We performed unit tests on only part of our translator. For this we use junit to do these tests.



Example launching tests

Our tests were very simple, here our method created a list by reading a file. The test consists in initializing different cases such as nonexistent files, correct files, incorrect files and checking if the list is created.

```
assertTrue("Fichier Inexistant", isFile );// isFile.True Le fichier existe
assertTrue("Nom Fichier null", paramFileName != null );// paramFileName is null
FileParser fileparser = new FileParser( new File(paramFileName));
List<DataParsed> listParam = new ArrayList<DataParsed>();
List<DataParsed> listParam_null = new ArrayList<DataParsed>();
listParam = fileparser.dataToList();
assertTrue("Fichier rend liste vide", !listParam.isEmpty() );// listParam vide
assertTrue("Fichier rend liste null", listParam_null != null );// listParam vide
```

Example of Unit Test (dataTolist method)

Acceptance testing

To complete these incomplete unit tests, we check the data consistency with results of sparql queries.
For example, we query the date of experiments, two results of them:

"2014-02-25"

"`dat-e -ld"

We can observe their validity, if it was well translated and parsed. We test all data properties of the ontology on RDF data. We report these results in a graph. (next page)

Data consistency

Absciss: number of experiments well translated/parsed
Ordinate: data properties

The graph is divided in two population. The first one represents the data from html and parameter file parsing the second one to the parameter file parsing. We see that some data are specific to our parameter file.
We realize that most of the data is consistent, the results meet with our expectation.

CONCLUSION

Presentation of NEW work's client

This project allows to improve the main process in the workflow of experimentation, i.e. the building of new hypothesis and new experiments. The human work is computerized, the access at old data to think new tests are now possible with a smart database. We have easy access to information. This information is formalized according to the ontology that we realized. Moreover, this formalization is done automatically thanks to our translator. We can imagine a saving of time and cost for our users to be able to access these experiences via a triplestore. This new process able to the client to retrieve all information about all of his experiments. Per example he can compare the analysis between all the variations of an experiment.

Feedback

We encounter some problems in the development, interactive work with the customer have advantages but also problems. The change of specification in the time is difficult in the code development, the design of application evolves slower that the expectations.
We learn that flexibility and anticipation are important.

We worked a few weeks with EXPO ontology (see ontology development) and finally it wasn't a proper choice. So, we have chosen to stop to develop this solution despite the time invested. Take the correct decision-making can be difficult when we invest in a solution, but sometimes it is necessary.

We worked in a research context, the problem is that not everything is normalized in semantic web and the practice to describe an Ontology is different between Protege (editing software), the Java OWL API and the different research team. It is a real difficulty for our because we haven't the skills to make decisions.
However, it is interesting to understand the first process in a technology development the different point of view in the scientific community. This project has allowed us to develop skills in web semantic: culture of semantic concepts, modeling and OWL and RDF description language.

We developed solutions to answer at the mains customer's requests. And this work we have allowed us to develop many skills in project management, modulization, coding and scientific culture.

References

Apache Jena Fuseki. (2019, June 19). *Apache Jena Fuseki*. From https://jena.apache.org/documentation/fuseki2/

Brickley, D. G. (2014, February 25). *Schema rdf 1.1.* From https://www.w3.org/TR/rdf-schema/

Euzenat, J. (s.d.). Ter Subject Presentation of experiments.

Inria. (s.d.). *Lazylav*. From http://lazylav.gforge.inria.fr

Soldatova, L. N. (2006). *An ontology of scientific experiments. Journal of the Royal Society, Interface.*

W3C. (2019, December 11). *OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax*. From W3C: https://www.w3.org/TR/owl2-syntax/

Wikipedia. (2019, June 19). From Web Ontology Language: http://fr.wikipedia.org/w/index.php?title=Web_Ontology_Language&oldid=158268644.

Wikipedia. (2019, June 19). *Ontology (information science)*. From Wikipedia: https://en.wikipedia.org/w/index.php?title=Ontology_(information_science)&oldid=901558404

Wikipedia. (2019, June 19). *Semantic Web*. From Wikipedia: https://en.wikipedia.org/w/index.php?title=Semantic_Web&oldid=901173627

Wikipedia. (2019, June 19). *Triplestore*. From Wikipedia: https://en.wikipedia.org/w/index.php?title=Triplestore&oldid=883772758