Computer-Project-Ontology

Construction, Validation and Choice of Knowledge Base

Raja Hanafi
National School of Computer Science
University of Manouba
Manouba, Tunisia
e-mail : rajarajahanafi@yahoo.com

Lassad Mejri
Faculty of Science of Bizerte
Carthage University
Bizerte, Tunisia
e-mail **:** Mejrilassad@gmail.com

Henda Hajjami Ben Ghezala
National School of Computer
University of Manouba,
Manouba, Tunisia
e-mail : hhbg.hhbg@gmail.com

Abstract—**The ontology design has developed within the framework of approaches of acquisition and capitalization of knowledge. In this case, we are talking about the design of the domain ontology which models the knowledge of a particular domain whose bounded terms are specified, mostly coming from controlled vocabularies. This ontology makes it possible not to encroach on another field of expertise. This has the advantage of being reused for applications designed within a defined domain. In this paper, we propose domain ontology(C-P-Onto: Computer-Project-Ontology) to represent knowledge in the field of computers projects. "Protégé" tool is the most popular and widely used tool for ontology development. Thus, we use this device for developing, validating and questioning the proposed ontology. In order to test it in a real field, the "HAL" will be applied as our knowledge base.**

Keywords-Computer project; Knowledge Capitalisation; Domain Ontology; Ontology Test; Ontology Validation.

I. INTRODUCTION

Recently, project management has been well established in companies, but the project's success in terms of quality, costs and deadlines is still difficult to reach. There are many failures, and one of them is the lack of capitalization of feedback during projects: learn lessons and reuse knowledge or acquired skills. The problem here is that these experiences are not always available in companies. This can be explained by the absence of the concept of capitalization, the lack of structuring of their experiments, or the outflow of experts [1].

During the realization of the projects, particularly the computer ones, project leader encountered many problems during the design phase. In the aim of solving these difficulties, designers either contact the old experts of the company or they look for similar projects in the market [2].

However, this process is not always efficient because it can be a waste of time, an exceeding of deadlines and a rise of cost.

In this context, we will propose a capitalization approach of memory knowledge of computer design projects. This approach presents a supporting decision in the project management phase of the design phase. Our decision-support process will not only help structure, formalize and capitalize knowledge, but also provide a dashboard in the form of indicators, information and a guide for the project leader.

In this paper, we will introduce our approach by focusing on its modeling part which is defined and explained by the proposition of domain ontology. This ontology describes all the concepts and relations associated to the computer project management term. After examining the coherence and the consistency of the proposed ontology, we will move to the process of creating a knowledge base in order to confirm our tests on a real level.

This paper is composed of four major sections. The first Section introduces the main works existing in the literature as well as a comparative study. Then, we describe our proposed approach architecture. The second Section involves the presentation the notion of ontology and the construction of the domain ontology. The third Section presents our analysis and reveals our knowledge base (HAL). The last and the four Section summarizes the main points mentioned in this article and open new horizons for future works.

II. PROPOSED APPROACH

We introduce in the following sub-section some of the most important contributions and capitalization models related to our research study.

A. Literature Review

1) Description Of The Models : In the literatures, various works addressed the project memory models which aim at the capitalization of knowledge and the construction of project memory.

IN [3] Ermine described the knowledge management processes. The proposed processes are based on a model that is called “margerite model”. These processes can be internal or external. What interests us is the internal process of capitalization and sharing of knowledge within the company.

Harani [4] presents a design assistance tool whose main objective is the capitalization of knowledge involved in the design of a product for reuse.

Bekhti in [5] proposed a dynamic project definition and reuse process named DyPKM. This approach is based on a method that provides a structured trace of a project memory

ISBN: 978-1-61208-827-3

containing the context in which the design takes place and the logic resolution.

Zacklad [6] propose a groupware "MEMO-net" using the DIPA problem solving method for capitalization and knowledge management in design projects. This groupware is a tool that has two modules (design and diagnostic) that allows a project group to solve problems encountered during the design (capitalization of the design logic) and to preserve the characteristics related to such a product.

Serrano [7] proposed a global system of capitalization of knowledge allowing the actors of the company to exploit the important mass of information. This system also makes it possible to capitalize events in the field of Open Source Intelligence (OSI) based on the Web Lab platform.

2) Comparative Study Of The Studied Approaches : After we have studied these different approaches, we decided to propose our own classification (Table I). This comparative study is based on a set of criteria, namely:

- Simplicity of the method: This criteria means that the models must be used in an easy way and without the intervention of any other methods.
- Resource: Includes the data representing the constraints to be considered and the data of the project organization**.** The resource used in our contribution must be the memory project.
- Application domain: This criterion gives a global vision on the field of application. In our research study, we focus on the field of computer projects.
- Use of case based reasoning (CBR): We have introduced this criterion because we believe that it is crucial to use the CBR in the learning part.
- Capitalization level**:** This criterion is proposed in order to check the importance of the conception level. We try to determine, for each model, the level or the part concerned by the capitalization (context, design, realization, etc).

TABLE I. COMPARATIVE STUDY OF APPROACHES

Model	Simplicity of the method	Resource	Application domain	Use of (CBR)	Capitalization level
Ermine's process [3]	Complex (marguerite model)	Corporate memory	Area of economy	No	Design
Zacklad ‘s model [6]	Complex Collective Software (DIPA)	Diverse	For all design projects	No	Conception and context
Serrano [7]	Global + wave (weblab platform)	Open source (blog, internet, site,etc)	Field of defense	No	Event
Harani Model [4]	Simple help tool	Company knowledge	Computer, mechanical, industrial	No	Design + Feature
Bekhti model [5]	Simple process	Project memory	Design project (all areas)	No	Context + design rational

Based on our comparative study we will define in the next section our approach to capitalize knowledge of project memory. This approach aims to provide decision support in project management from the design phase to the implementation.

B. Towards a knowledge Capitalization Approach

Our goal is to present an approach that helps the leader to deal with its new project by referring to the experiences and knowledge which are stored in a project memory. This section introduces the architecture of our approach and in particular the modeling part which is composed of three models: the project class model, the project model and the rational design model. The architecture of our approach contained three main parts (Fig.1):

- The offline process: It is from modelization (models + ontology) to the project excavation. This part starts with the proposal of the models to identify and to classify projects.
- An online process: It is from the acquisition of new project until the project learning. This part presents CBR reasoning cycle, which are development, remembering, adaptation, revision, validation and learning.
- A base case: It contains all the instances of the ontology, projects, project classes, problems, solutions and suggestion.

In the following subsections we will start with the offline process description. We will explain the proposed models namely project model, project class model and rational design model, and then we will introduce the proposed ontology.

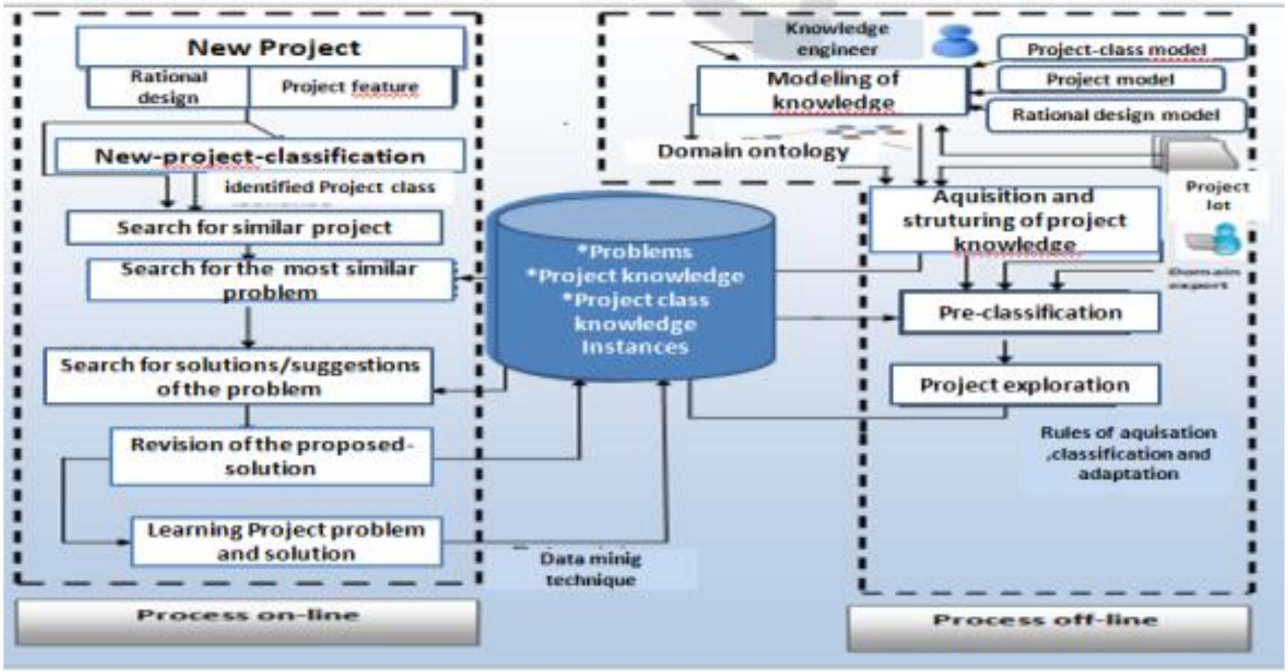

New Project
Rational design
Project feature
New-project-classification
identified Project class
Search for similar project
Search for the most similar problem
Search for solutions/suggestions of the problem
Revision of the proposed-solution
Learning Project problem and solution
*Problems
*Project knowledge
*Project class knowledge Instances
Data minig technique
Knowledge engineer
Modeling of knowledge
Project-class model
Project model
Rational design model
Domain ontology
Aquisition and struturing of project knowledge
Project lot
Domain expert
Pre-classification
Project exploration
Rules of aquisation ,classification and adaptation
Process on-line
Process off-line

Figure 1. Architecture of the proposed approach.

1) The Project Class Model : In the same organization, we can distinguish different classes of computer project such as security, software engineering, imaging, data base, artificial intelligence, etc. It is in this context, that we propose this model to allow the leader to classify, from the beginning, the project. This process can be done by specifying the project knowledge, its resolution method such as Scrum [8] and Pert [9], its reasoning rule and its architecture.

The Project class model (Fig.2) is composed of three elements:

- Project class: This element is composed of two lists: a list of projects belonging to the same class, and a list of common denominators such as rules and keywords.
- Project class knowledge: All the knowledge related to the project class in question are associated to all the rules used in the reasoning phase for this type of project class.
- Point of view: This component presents the method of conducting project class and the type of architecture used.

2) The project model: We have proposed this model to identify the project itself. When the user is in front of a new project he will first determine the characteristics of each project. These will be used as indexes to select similar ones. The proposed project model (Fig. 2) has three dimensions. The choice of components of this model is inspired from the composition of the project memory:

- Project identifier: This pillar gives general information about the project. It includes the project name, abstract, project team.
- Project features: This component reflects all the characteristics that a project can have during its realization. Among these characteristics, we can quote the size, scope, cost, time, complexity, type, team project, scheduling, etc.
- Deliverable: This class is composed of two sub-classes:

-Type of deliverable: It can be a service, a product (software, hardware), etc.
-Rational design: Contains the list of problems, suggestions and solutions for each computer project.

3) Rational Design Model : Once the project has been identified, the user must see the logical design part to distinguish similar problems and select solutions. For the purpose of presenting this part, we have suggested the design logic model explained below.

The design rationale, in the project memory, consists of modeling the process of decision-making through all the elements characterizing this process. These elements are the problem objects, suggestions, and participants [10]. This model (Fig.2) is presented using three essential components:

- Problem list: Each problem is described by its name, its textual description and its attributes.

- Suggestion list: Before reaching the final solution the designers have proposed a set of suggestion.
- Solution list: For each problem there are one or more solutions that are defined (text) and argued (arguments).

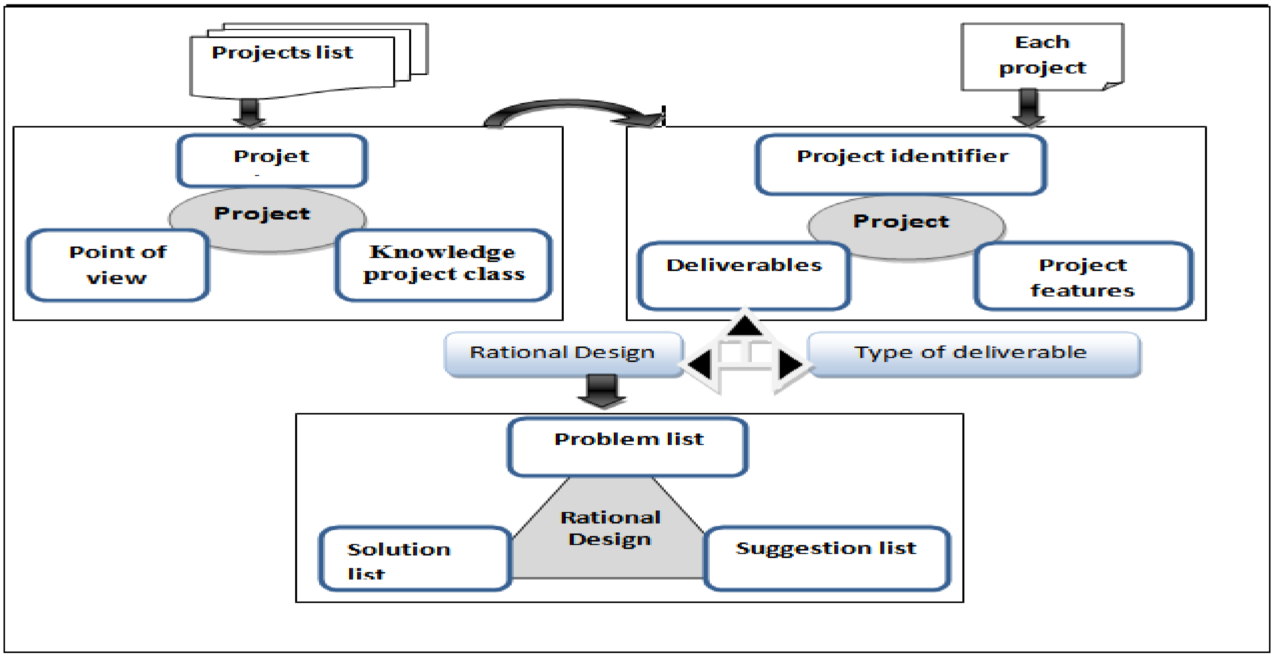


Figure 2. Knowledge Modelisation (Proposed Models)

By studying the components and elements of the proposed models we have noticed that the field of information projects contains a huge quantity of concepts and terms which relate to each other. It is in this context that we decided to present the important mass of this knowledge with one of the techniques of knowledge representation. In this research study, we will propose domain ontology relative to the notion "computer project" in order to present concepts composing this domain and relations between them.

III. Ontology

Using ontology and other related knowledge has also become very important for storing, and managing of huge amount of research data [11]. Ontology is essentially defined by a set of business concepts and relationships. The instantiation of these different concepts gives birth to a new case to study in the future (new project, new thesis).

A. Components Of Ontology

To describe a domain with ontology, knowledge of this domain should be defined by the following five components [13]:

- Concepts: (concepts also called class) representing the meaning of a field of information, whether by the metadata of a namespace or the elements of a given domain of knowledge.
- Relations also called properties: It translates the associations existing between the concepts. These relationships allow us to see the structuring of concepts, the ones compared to the others.
- Function: Presents special case of relations, of which an element of the relation can be defined according to the preceding elements.
- The axioms: Also called rules are used to describe assertions of the ontology in order to define the meaning of the components of the ontology.
- Instance or individual: Constituting the extensional definition of ontology; these objects convey knowledge about the domain of the problem.

A. The Methodology Of Ontology Construction

The construction of ontology is a difficult task requiring the implementation of an elaborate process to extract the knowledge of a domain, manipulated by computer systems and interpreted by human being. There are many methods of ontology construction. Different types of ontology construction approaches are distinguished according to the support on which they are based: from texts, dictionaries, knowledge bases, semi-structured diagrams, relational diagrams, etc. In what follows we present some methods of ontology construction.

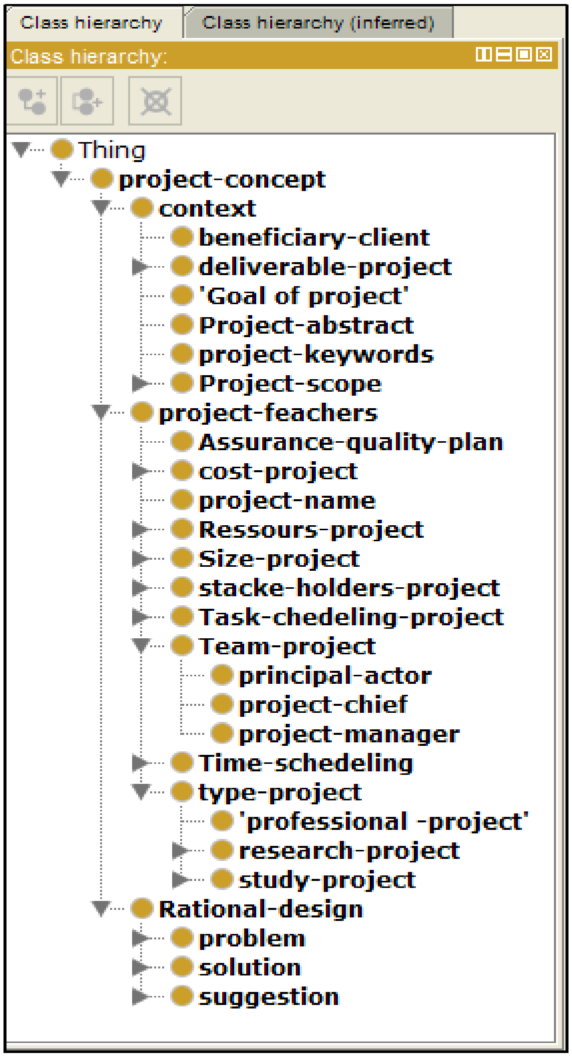
- The Text To Into [14] methodology is an application for extracting ontologies from corpora or web documents and it also allows the reuse of existing ontologies.
- The Onto Builder methodology [15] allows us to build ontology from web resources.
- METHONTOLOGY [16] is a structured method to build ontologies from scratch. It is based on the experience acquired in developing ontology in a special domain.
- KACTUS [17] designed to be applied in more general settings. This methodology, which aims to reuse existing ontologies, is interesting since it avoids building an ontology from scratch.

By studying these four methods, we have proposed a new method for the construction of our ontology. Our methodology is based on METHONTOLOGY methodology. It is a method of building ontology from scratch which is related to computer projects domain experiences. To apply the proposed methodology, we will follow these three steps:

1. Choice of the relevant terms of the field, favoring the semantic normalization and specify the relations between the different terms.
2. Formalization of knowledge and the construction of a referential ontology.
3. Evaluation, testing, validation and documentation of the proposed domain ontology. In our situation, we used the ontology editor "Protégé" [18] to formalize our ontology.

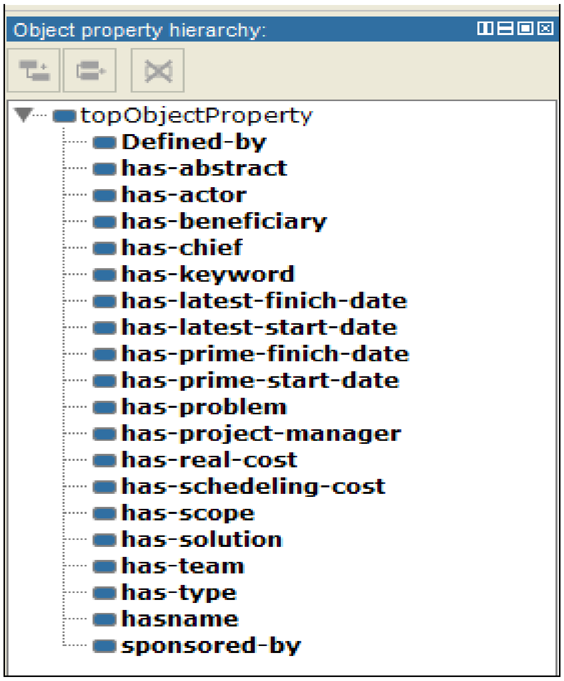
C. Basic Steps For Building Ontologies

- Step 1 'classes and class hierarchy': The first step as illustrated in Fig. 3 gives the computer project and management project related classes or concepts. All the concepts shown in the figure are focusing on the project-concept, project-team-project-features and rational-design.
- Step 2 'object properties of ontology (Fig. 4) ': We define it according to relationship which we want to add between classes.
- Step 3 'data properties of ontology': In this step we display data properties of proposed ontology which show the relationship between individuals.
- Step 4: In this step we add the details of the instances, relations, classes and properties. These details present the definition, description and the type of each element.
- Step 5 'the axioms of ontology': Axioms are used to describe the relationship between classes, attributes and individuals.
- Step 6 'the instance of ontology': Defining the instance (individual), first one should select the right class, and then create its instances for the class. The final instantiation of this ontology (individuals + instances) is actually the new case on which our reasoning is based (Fig. 6). They help to establish a common vocabulary to describe the case, or the model knowledge needed to index and organize the event. We have advanced our thesis research topic to instantiate our proposed ontology.
- Step 7 'the reasoning of ontology': To have a consistent ontology and ready to be properly interrogated and without contraction we carried out a reasoning using the automatic reasoner of the “Protégé tool”.



- Thing
  - project-concept
    - context
      - beneficiary-client
      - deliverable-project
      - 'Goal of project'
      - Project-abstract
      - project-keywords
      - Project-scope
    - project-feachers
      - Assurance-quality-plan
      - cost-project
      - project-name
      - Ressours-project
      - Size-project
      - stacke-holders-project
      - Task-chedeling-project
      - Team-project
        - principal-actor
        - project-chief
        - project-manager
      - Time-schedeling
      - type-project
        - 'professional -project'
        - research-project
        - study-project
    - Rational-design
      - problem
      - solution
      - suggestion

Figure 3. Classes of the Proposed Ontology.



Object property hierarchy:

- topObjectProperty
  - Defined-by
  - has-abstract
  - has-actor
  - has-beneficiary
  - has-chief
  - has-keyword
  - has-latest-finich-date
  - has-latest-start-date
  - has-prime-finich-date
  - has-prime-start-date
  - has-problem
  - has-project-manager
  - has-real-cost
  - has-schedeling-cost
  - has-scope
  - has-solution
  - has-team
  - has-type
  - hasname
  - sponsored-by

Figure 4. Property and Relationship.

IV. RESULT AND ANALYSIS

In order to validate our ontology (especially the choice of concepts and relations between concepts) we carried out two types of validations:

- A first validation made by a professional who is an expert in the computer field. He is a computer project manager in a company located in France.
- A second validation was made technically by the standard tool "Protégé"[18]. This validation is done using three tests: consistency test, coherence test and query test.

In this section, we will describe the main tests used to verify the coherence and the consistency of the proposed ontology.

A. Knowledge Base

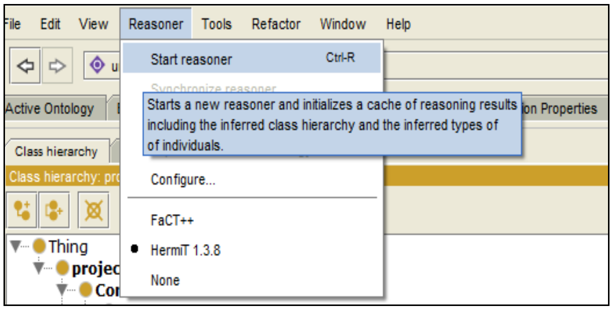
The first goal in this part of the research study is to create a knowledge-base that contains all instances of the concept defined by the proposed ontology. To achieve this goal, we decided to work with real examples of computer research projects. It is in this context that we decided to test our ontology with the help of the dozen end-of-study projects that we have supervised.

The second objective is to be able to question our ontology. This step is also due to two other sub-phases: The validation of the coherence and the consistency of the created ontology. It is in this context that in the following sub-sections we will present our procedures to carry out these tests.

B. Coherence test :Reasoner tab

The great advantage of using Protégé is the possibility of checking whether the ontology created does not contain contradictory definitions. From the Reasoner menu, we can select FaCT++ or HermiT, then select Reasoner or Start Reasoner to classify the active ontology. We can also select Reasoner or Synchronize Reasoner to classify again at any time.

Once we validate that our ontology is classified by selecting the entities tab and then the "Class hierarchy (inferred)" tab that appears in the "Class hierarchy" view. It should contain classes that sub-class Thing. Once we have validated that our ontology is classified, you can execute a query using Dl-Query tab.

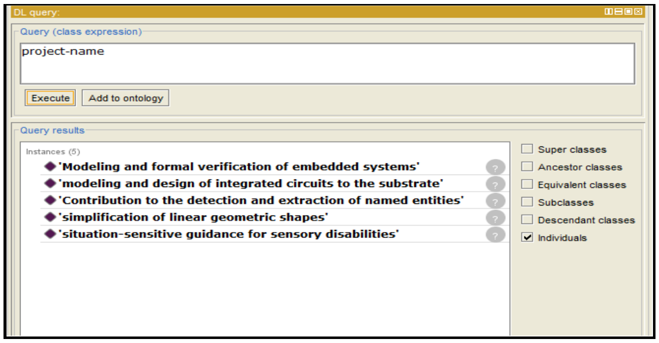


Starts a new reasoner and initializes a cache of reasoning results including the inferred class hierarchy and the inferred types of of individuals.

Figure 5. Coherence Test (Ontology ' s Reasoning)

C. Consistency test : Dl-Query tab

We have utilized the DL Query to check the consistency of the ontology hierarchy. The DL-Query tab provides a powerful and easy-to-use feature for searching a classified ontology. So we can only execute a query on a classified ontology. Before attempting to execute a query, we should run a classifier: Using the DL, we can have as a result a list of super classes, subclasses, instances or direct subclasses of a class expression. In our case we have to display all the projects (fig.6) which are instances of the class "project_name". The complete display (without failure) of the results of the query requested shows that our ontology is well classified.



DL query:

Query (class expression)

project-name

Query results

Instances (5)

- 'Modeling and formal verification of embedded systems'
- 'modeling and design of integrated circuits to the substrate'
- 'Contribution to the detection and extraction of named entities'
- 'simplification of linear geometric shapes'
- 'situation-sensitive guidance for sensory disabilities'

□ Super classes
□ Ancestor classes
□ Equivalent classes
□ Subclasses
□ Descendant classes
☑ Individuals

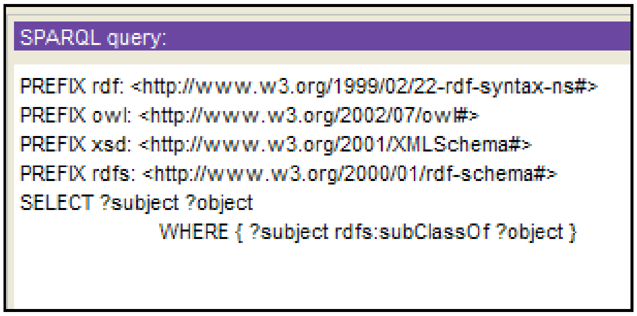
Figure 6. Example of Test With Dl-Query.

D. Query an ontology via a sparql query Request:

After we have tested our ontology we can now request it by several methods: either we load it into an RDF database like JENA [19], Sesame [20], Stardog [21], etc, or we can simply use the SPARQL-query (Fig.7) option automatically integrated in the "Protégé tools" [21].

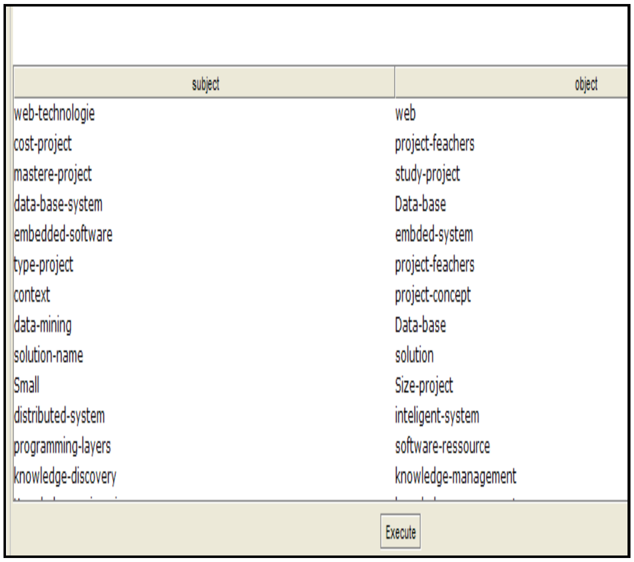
Here we used the SPARQL-query to request our ontology. We have launched an example of a query in SPARQL-query that allows us to display all the classes and subclasses of the created ontology. The result is displayed on two columns "subject and project" as indicated in the request (Fig. 8).

These results are still modest and weak. The application of SPARQL-query does not allow us to display, for example, all individuals of such a class with a given condition. It is in this context that we are going to orient our future work on the interrogation of ontology using API such as JENA and Sesame. In addition, even if our ontology is consistent and well classified but the shortcomings of interrogation by SPARQL-query shows the weakness of inferences especially at the instances level.



```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?subject ?object
            WHERE { ?subject rdfs:subClassOf ?object }
```

Figure 7. Example of a Sparql Query is Request).



subject	object
web-technologie	web
cost-project	project-feachers
mastere-project	study-project
data-base-system	Data-base
embedded-software	embded-system
type-project	project-feachers
context	project-concept
data-mining	Data-base
solution-name	solution
Small	Size-project
distributed-system	inteligent-system
programming-layers	software-ressource
knowledge-discovery	knowledge-management

Figure 8. Example of a Sparql Query result.

V. Conclusion and Futures Works

The main objective of this research study was to propose a domain ontology that helps to present a computer project field (concept and relation). Moreover, we described multiple methodologies in the construction of ontology and we ended this section by proposing a method of construction of a domain ontology based on METHONTOLOGY methodology.

Given the importance of the information and knowledge of the website of publication and archiving, we chose to apply it to feed our knowledge base. For future works, we will focus on interrogating by utilizing the API "JENA"[19]. Also, we will try to complete the on-line process of our approach to apply the concept of case-based reasoning (CBR).

References

[1] J.Stal-Le Cardinal, J.-L.Giordano and G.Turré,“ Les Retours d'expérience du Projet, Réduire les Risques, Augmenter les Performances Collectives (From The Project Reduce Risks, Increase Collective Performance)”, https://hal.archives-ouvertes.fr/hal-01482335, 2014.

[2] B.Francois, “ La Capitalisation Des Connaissances Dans un Contexte de Projet (The Capitalization of Knowledge in A Project Context)”, Thesis submitted as part of the project management master's program, University of Quebec at Rimouski, 2014.

[3] J-LErmine, “Knowledge Management(Gestion de Connaissance)”,HeLavoisier, p.166, https://hal.archives-ouvertes.fr/hal-00997696/file/. pdf,2003.

[4] Y.Harrani, “A Multi-model Approach For the Capitalization of Knowledge In The Field of Design”,Thesis, specialty in Industrial Engineering, 1997.

[5] S.Bekhti, “A Dynamic Process For Definig And Reusing Project Memories" ,University of Technology Troyes, France,2003.

[6] M.Lewkowicz and M.Zacklad, “Using Problem-Solving Models to Design Efficient Cooperative Knowledge-Management Systems Based on Formalization and Traceability of Argumentation”, EKAW '00: Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management, Pages 288–295, 2000.

[7] L.Serrano, “Vers Une Capitalisation Des Connaissances Orientée Utilisateur: Extraction et Structuration Automatiques d'informations à Partir de Sources Ouvertes“: (Towards user-oriented knowledge capitalization: Automatic Extraction and Structuring of Information From Open Sources), Thesis, 2014.

[8] M. Sliger, “Agile Project Management with Scrum”, paper presented at PMI® Global Congress2011—North America, Dallas, TX. Newtown Square, PA: Project Management Institute, 2011.

[9] N.Mahfouf and B.Ramdhani, “Planification et ordonnancement d'un projet avec des moyens limités au sein d'ENGTP”, (“Planning and Scheduling of a Project with limited Means Within ENGTP”), thesis , 2014.

[10] S.I.Hyder, J.Ansari, M.S.Ramish and M.Y.T.Fasih, “Emerging Role of Ontology Based Repository in Business Management Research”, Journal of Organizational Knowledge Management, p.17, 2017.

ISBN: 978-1-61208-827-3

[11] B.Chabot,“Ontologies-Pourquoi-Quoi Comment(Ontologies:Why?What?How?), Published on October 26, 2017.

[12] J. Busse, B. Humm, C. Lubbert and F. Moelter, “Actually, What Does Ontology Mean? A Term Coined by Philosophy in the Light of Different Scientific Disciplines”, Journal of Computing and Information Technology, vol. 1, no. CIT 23, p. 29–41, 2015.

[13] J.Chaumier, “Les Ontologies Antécédents, Aspects Techniques et Limites (Ontologies History, Technical Aspects And Limits)“, Documentalist-Information Sciences », Vol. 44, pages 81 to 83 ISSN 0012-4508, 2007.

[14] A.Maedche, E.aedche, S.Staab, “The TEXT-TO-ONTO Ontology Learning Environment”, a Software Demonstration at ICCS-2000 Eight International Conference on Conceptual Structures, 2000.

[15] H.Roitman, A.Gal, “OntoBuilder: Fully Automatic Extraction and Consolidation of Ontologies from Web Sources Using Sequence Semantics”, Technion, Israel Institute of Technology Technion City, 2004.

[16] M.Fernandez,A.Gómez-Pérez,N.Juristo,“From Ontological Art Towards Ontological Engineering”, Acte of AAAI, 1997.

[17] G.Schreiber, B-J.Wielinga, W.Jansweijer, “The kactus view of the ’o’ word.”, IJCAI’1995, Workshop on Basic Ontological Issues in Knowledge, 1995.

[18] V.Giudicelli, “Ontologies et l’éditeur Protégé - Application à la Formalisation des Concepts de Description d’IMGT-ONTOLOGY (Ontologies And The Protected Editor - Application To The Formalization Of IMGT-ONTOLOGY Description concepts)”, 2010,

[19] https://jena.apache.org/documentation/ontology/accessed August 06, 2018.

[20] https://websemantique.developpez.com/tutoriels/francart/debuter-avec-sesame/, March 25, 2018.

[21] About: Stardog, http://dbpedia.org/page/Stardog, March 22, 2018.