AN EXTENSION OF PROTÉGÉ FOR AN AUTOMATIC FUZZY-ONTOLOGY BUILDING USING CLUSTERING AND FCA

Aloui Amira[1] and Grissa Touzi Amel[2]

[1]Université Tunis El Manar, Ecole Nationale d'Ingénieurs de Tunis, LRSITI, Tunis, Tunisia
[2]Université Tunis El Manar, Faculté des Sciences de Tunis, LIPAH, Tunis, Tunisia

ABSTRACT

The process of building ontology is a very complex and time-consuming process especially when dealing with huge amount of data. Unfortunately current marketed tools are very limited and don't meet all user needs. Indeed, these software build the core of the ontology from initial data that generates a big number of information. In this paper, we aim to resolve these problems by adding an extension to the well known ontology editor Protégé in order to work towards a complete FCA-based framework which resolves the limitation of other tools in building fuzzy-ontology. We will give, in this paper, some details on our semi-automatic collaborative tool called FOD Tab Plug-in which takes into consideration another degree of granularity in the process of generation. In fact, it follows a bottom-up strategy based on conceptual clustering, fuzzy logic and Formal Concept Analysis (FCA) and it defines ontology between classes resulting from a preliminary classification of data and not from the initial large amount of data.

KEYWORDS

Formal Concept Analysis; Fuzzy Logic; Ontology extraction;Protégé-200.

1. INTRODUCTION

The need for applications that are able to process the overload information on a specific field is thriving because of the explosion of the dizzying number of data that continues to increase exponentially, every day. This need has become indispensable for tasks that require an understanding of the semantics of natural language documents or capitalization human expertise in very specific areas.

Ontologies are a very promising way to meet this challenge. But, their manual construction is very expensive and could not be easily reusable. The semi-automatic approaches for building ontologies began to give more and more satisfactory results.

Our goal in this paper is to automate a statistical approach to the extraction of simple ontology from initial data (DB), first by applying a method based on conceptual clustering, fuzzy logic and FCA. After that, we integrate this new approach on protégé 3.5 to avoid manual construction and description of field-specific ontology which is a complex and time-consuming procedure.

We prove in this paper that our approach aims to improve the information retrieval in many specific fields or any other applications in the Semantic Web.

DOI:10.5121/ijcsit.2015.7202

As an original contribution, our work resides in the fact that for the first time, a new tool for building ontology takes into consideration another degree of granularity in the process of this building. Indeed, the proposed tool defines ontology between classes resulting from a preliminary classification of the data and not from the initial large amount of data.

The rest of the paper is formed as follows: Section 2 introduces the basic concepts of ontology and FCA. Section 3 presents the existing approaches and tools; Section 4 presents our methodology for creation the ontology. Section 5 describes our tool for the semi-automatic generation of Fuzzy Ontology of Data Mining and represents the atomization of the approach with its implementation as a plug-in in Protégé 3. We conclude this paper with some perspective and future works.

2. BASIC CONCEPTS

In this second section, we introduce two basic notions of our method for building ontology: Formal concept analysis and Ontology.

2.1 Formal concept analysis (FCA)

The Formal Concept Analysis [8] is defined as bellow:

Definition1. A Formal Context is a triple (G, M, I) where G is a set of objects, M is a set of attributes, and I is a binary relation between G and M (i.e $I \subseteq G \times M$) .
$(g, m) \in I$: to say that object g has attribute m.
Definition2. Formal Concept of context (G, M, I) is a pair (A, B) with $A \subseteq G$, $B \subseteq M$, A' = B, B'=A where if $A \subseteq G$ we define $A' := \{m \in M \mid \forall g \in A : (g, m) \in I\}$ and if $B \subseteq G$ we define $B' := \{g \in G \mid \forall m \in M : (g, m) \in I\}$
Definition3. Sub-Super concept relation:
$(A1,B1) \leq (A2,B2) \Leftrightarrow A1 \subseteq A2$ ($\Leftrightarrow B1 \subseteq B2$ respectively) is defined as :
(A2,B2) is a super-concept of (A1,B1)
A concept is composed by a set of objects which are similar according to the interpretation of attributes. Inclusion relation between the object sets can reflect the sub-super relation between different concepts. The concepts and their relations can construct a concept lattice which will finally be converted to domain ontology; in this ontology only subsumption relation can be extracted rather than other enriched relation. The selection of object and attribute will vary based on different application.

2.2 Ontologies

There is no official definition of what an ontology is. But, almost all approaches share the same core items : concepts, a hierarchical relations, axioms, constraints, functions.,. etc.
We formalize the core in the following way:

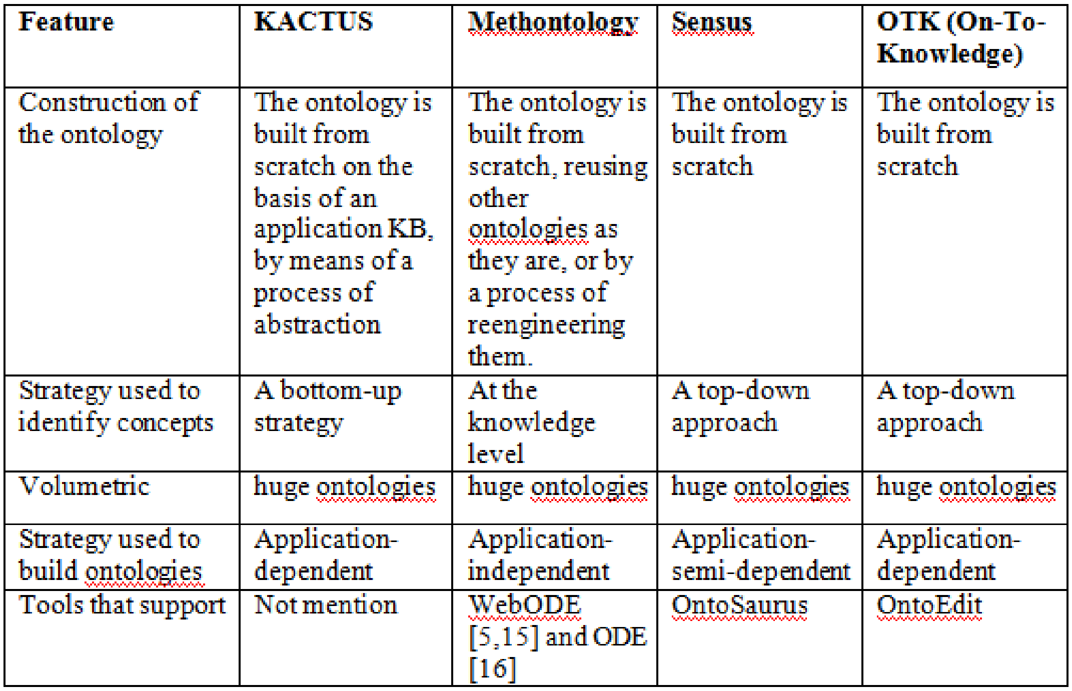
Definition: A (core) ontology is a tuple O = (C, is_a, R, σ) where:

- C is a set of elements called concepts
- is_a is a partial order on C
- R is a set whose elements are relation,
- $\sigma : R \rightarrow C+$ is a function which assigns to each relation name its arity.

Next section of this paper will present the methodologies, tools and languages already available which can help developers in building ontology from scratch.

3. EXISTING ONTOLOGY BUILDING TOOLS AND METHODOLOGIES

3.1 Ontologies Building



Feature	**KACTUS**	**Methontology**	**Sensus**	**OTK (On-To-Knowledge)**
Construction of the ontology	The ontology is built from scratch on the basis of an application KB, by means of a process of abstraction	The ontology is built from scratch, reusing other ontologies as they are, or by a process of reengineering them.	The ontology is built from scratch	The ontology is built from scratch
Strategy used to identify concepts	A bottom-up strategy	At the knowledge level	A top-down approach	A top-down approach
Volumetric	huge ontologies	huge ontologies	huge ontologies	huge ontologies
Strategy used to build ontologies	Application-dependent	Application-independent	Application-semi-dependent	Application-dependent
Tools that support	Not mention	WebODE [5,15] and ODE [16]	OntoSaurus	OntoEdit

Table 1: Ontology building methodologies

A series of approaches have been reported for developing ontologies. We will show, chronologically, in this section a brief history of existing methodologies.

- In 1990, Lenat and Guha published the general steps and some interesting points about the Cyc development [20].
- In 1995, the basis of the experience gathered in developing the Enterprise Ontology [23] and the TOVE (TOronto Virtual Enterprise) project ontology [14] (both in the domain of enterprise modelling), the first guidelines were proposed and later refined in [24,25].
- In 1996, Bernaras et al. [2] presented a method used to build ontology in the domain of electrical networks as part of the Esprit KACTUS [23] project at the 12th European Conference for Artificial Intelligence (ECAI96).
- In 1996, a methodology called METHONTOLOGY [10,11] appeared and discussed in many papers [9, 12].
- In 1997, a new method was proposed for building ontologies based on the SENSUS ontology [21].
- In 2001, as a part of an On-To-Knowledge project a new methodology appeared with the same name [22].

There are many methods and methodologies that have been proposed for other tasks than building ontologies, such as ontology reengineering [6], ontology learning [1,19], ontology evaluation [7,9,10,11,12, 3,4], ontology merging [20], etc.

In this paper, we will focus only on methodologies for building ontology and we will compare in the Table 1 all these methodologies for building ontology.

As a result of a our comparison work , we deduce that none of the approaches presented is fully complete if we compare them with other software engineering. As summarized in Table 1, the most developed approach is METHONTOLOGY. This approach has been recommended by FIPA for the ontology construction task.

Consequently, a great effort will be required for creating a consensuated methodology for ontology construction. Hence, we defined a new approach to automate the generation of Fuzzy Ontology Design (FOD), through the fusion of conceptual clustering, fuzzy logic, and FCA. In our approach, we proposed to generate an ontology taking into consideration another degree of granularity in the process of generation. Indeed, we suggest defining ontology between classes resulting from a preliminary classification of the data. This intends to achieve the objectives of offering better interpretation of the data and minimizing both execution time and space memory by reducing considerably the definition of the ontology.

3.2 Ontology development tools

Table 2: Ontology building tools

Tools	**KACTUS**	**Methontology**	**Sensus**	**OTK**	**WebOnto**
General issues Developers	KSL	ISI (USCr).	SMI	UPM	KMI
Current release and Date	1.0.6.49 Nov 2001	1.9 (Mar 2002)	1.8 (Jul 2002)	2.0 (Mar 2002)	2.3
Software architecture	Client /Server	Client /Server	Standalone	3-tier	Client /Server
Pricing policy	Free	Free	OpenSource Evaluation Version	OpenSource	Free
Extensibility	None	None	Plugins	Plugins	None
Ontology storage	Files	Files	File/ DBMS (JDBC)	File/ DBMS (JDBC)	Files
Imports from languages	Ontolingua IDL KIF	LOOM / DL /ONTO/KIF/ C++	XML/ RDF(S)/XML Schema	XML/ RDF(S) CARIN	OCML
Methodological support	NO	NO	NO	Yes	NO
Automatic classifications	NO	Yes	NO	NO	NO

Over the last years, the number of marketed software for building ontologies has grown exponentially. In Table 2, the most relevant ones are presented. We choose finally to use Protege-2000 as it is an open source, standalone application with an extensible architecture. Protégé offer a library of plug-in that add more functionality to its core which an ontology editor.

We choose to add a plug-in that will build a light-weight ontology from data. Indeed, our proposal is to define ontology between classes resulting from a preliminary classification of the data.

3. OUR METHODOLOGY FOR CREATING ONTOLOGY

We propose a methodology for the construction of ontology, comprising principally the steps recommended by Stanford University for the ontology development.

3.1. Determine the domain and scope of the ontology

Example. Let's describe the domain of the ontology with a relational database table presented by Table 3 containing the list of AGE and SALARY of Employee.

Table 3. Relational database table

	SALARY	**AGE**
t1	800	30
t2	600	35
t3	400	26
t4	900	40
t5	1000	27
t6	500	30

We apply FCA to Age and Salary attributes in order to construct a fuzzy formal context from a database containing uncertainty data.

This step organizes the database records of a specific field in homogeneous clusters having common properties. This step attributes for each cluster a certain number. A membership degree will be attributed for each tuple. This value is included in the interval [0, 1]. Then, in order to keep only the pertinent information, we apply an α-Cut to the set of membership degrees, to replace these last by values 1 and 0 and to deduce the binary reduced formal context. In our example, α-cut (Salary) = 0.3 and, α-cut (Age) = 0.5.

Definition. Alpha-cut We define the cut, noted α-Cut, on the fuzzy context as being the reverse of the number of clusters obtained.

$$\alpha\text{-Cut} = (c)^{-1} \qquad (1)$$

After that, a fuzzy formal concepts will be generated from the fuzzy formal context and organizes the generated concepts as a fuzzy concept lattice to finally generate the fuzzy clusters Lattice.

Definition. A Fuzzy Clusters Lattice (FCL) of a Fuzzy Formal Concept Lattice, consists on a Fuzzy concept lattice where each equivalence class (a node of the lattice) contains
The goal of this phase is to make a certain abstraction on the list of the objects with their degrees of membership in the clusters. This lattice will be used to build a core of ontology.

3.2. Define the classes and the class hierarchy

In this step, we define the classes of the ontology: how is a formal explicit description of classes in a domain (sometimes called concepts). Because both FCA and ontology support formal definitions of concepts, we deduce fuzzy ontology from a fuzzy clusters lattice using the concept hierarchy created by fuzzy conceptual clustering.

Thus, a fuzzy ontology is defined as follows:
Definition (Fuzzy Ontology). A fuzzy ontology Fo consists of four elements (C, AC, R, X), where:

- C represents a set of concepts,
- A^C represents the attribute sets,
- R = (R^T; R^N) represents a set of relationships, which consists of two elements:
    a. R^N is a set of non-taxonomy relationships
    b. R^T is a set of taxonomic relationships.
- Each concept c_i in C represents a set of objects, or instances, of the same kind.
- Each object o_{ij} of a concept c_i can be described by a set of attributes values denoted by $A^C(c_i)$.
- Each relationship $r_i(c_p, c_q,\alpha)$ in R represents a fuzzy association between concepts c_p and c_q, and the instances of such a relationship are pairs of (c_p, c_q) concept objects with confidence α; α is in]0..1].
- Each attribute value of an object or the relationship instance is associated with a fuzzy membership value between [0, 1] implying the uncertain degree of this attribute value or relationship.
- X is a set of axioms. Each axiom in X is a constraint on the concepts and relationships attribute values or a constraint on the relationships between concept objects.

In our approach, we consider the Fuzzy Ontology Lattice as a formal domain-specific ontology. We do a step of mapping to generate our ontology. We consider nodes as concepts. This ontology has all lattice properties, which are useful for ontology sharing, and reasoning.

Then we require a human interpretation to label the extent class name. We can use appropriate names to represent keyword attributes and use them to label the intent class names and the extent class names. In our example, the name of the concept is a concatenation of an attribute and its label linguistics as shown in Table 4.

Linguistic labels, which are fuzzy partitions, will be assigned to the attribute's domain. So, we will define from the beginning (Table 4) some naming conventions for concepts in lattice and then these conventions will make the generated ontology easier to understand. The naming convention we choose are:

- Capitalization and delimiters : to improve the readability of ontology with
- Singular or plural: should be consistent
- Add Prefix and suffix

Table 4. Correspondence between the linguistic labels and their designations

Attribut	**Linguistic labels**	**Designation**
Salary	Low	C1
Salary	Medium	C2
Salary	High	C3
Age	Young	C4
Age	Adult	C5

Nevertheless, taxonomic relationships between concepts are present in the lattice.
Figure 1 proposes a graphical representation of a simple ontology, where lattice nodes A, A1, A2, B are mapped to ontology concepts, Top is the most general concept of the ontology and Bottom to the empty concept.

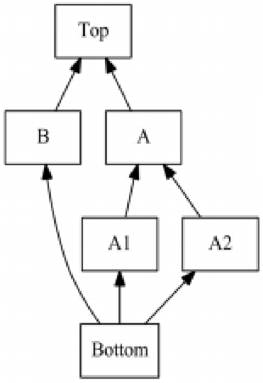


Figure 1. Ontology building using FCA

The process for building ontology starts with assigning the super class ”Thing” to the root of the concept lattice noted TOP. For each concept of the level 1 of the lattice, we create a subclass of the root class: Thing an assign the value 0 as a membership degree to the Thing class. Then from the level 1, we traverse the lattice and for each concept we find its successors (sub-concepts). We repeat this step to each sub-concept to create sub-classes. We assign finally a membership value from each super concept.

3.3. Define the properties of classes

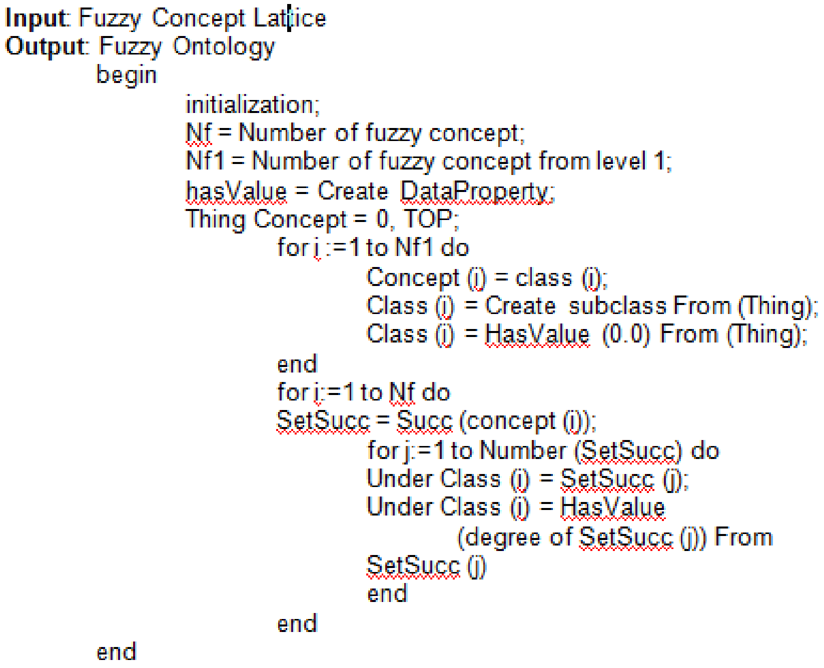
Properties may have certain characteristics, e.g., transitive, functional, symmetric, inverse, cardinality. In our case, properties form hierarchies. The "is_a" construct captures such a relationship. All the properties usually belong to a domain (e.g., Employee). A domain of a property limits the individuals (instances of classes) where the property can be used.

3.4. Define individuals and instances

We generate instances of the extent class. Each instance corresponds to an object in the initial fuzzy context. For example, each instance of the class Employee (which corresponds to an actual employee) will be associated with the appropriate Salary and Age areas.

4. ONTOLOGY BUILDING ALGORITHM

The FOD method for automatically generating a Fuzzy Ontology is illustrated by the algorithm shown in Figure 2. For the definition of the classes, we chose a combined process mix between top-down process. We begin by defining the general concepts then we lays the composition of these.



```
Input: Fuzzy Concept Lattice
Output: Fuzzy Ontology
        begin
                initialization;
                Nf = Number of fuzzy concept;
                Nf1 = Number of fuzzy concept from level 1;
                hasValue = Create DataProperty;
                Thing Concept = 0, TOP;
                        for i :=1 to Nf1 do
                                Concept (i) = class (i);
                                Class (i) = Create subclass From (Thing);
                                Class (i) = HasValue (0.0) From (Thing);
                        end
                        for i:=1 to Nf do
                        SetSucc = Succ (concept (i));
                                for j:=1 to Number (SetSucc) do
                                Under Class (i) = SetSucc (j);
                                Under Class (i) = HasValue
                                        (degree of SetSucc (j)) From
                                SetSucc (j)
                                end
                        end
        end
```

Figure 2. Ontology building algorithm

5. FOD Tab Plug-in Implementation

5.1 Architecture of the FOD Tab Plug-in

In this section, we present the architecture of our tool Fuzzy Ontology Design Tab Plug-In and the process for building fuzzy ontology (Figure 3).

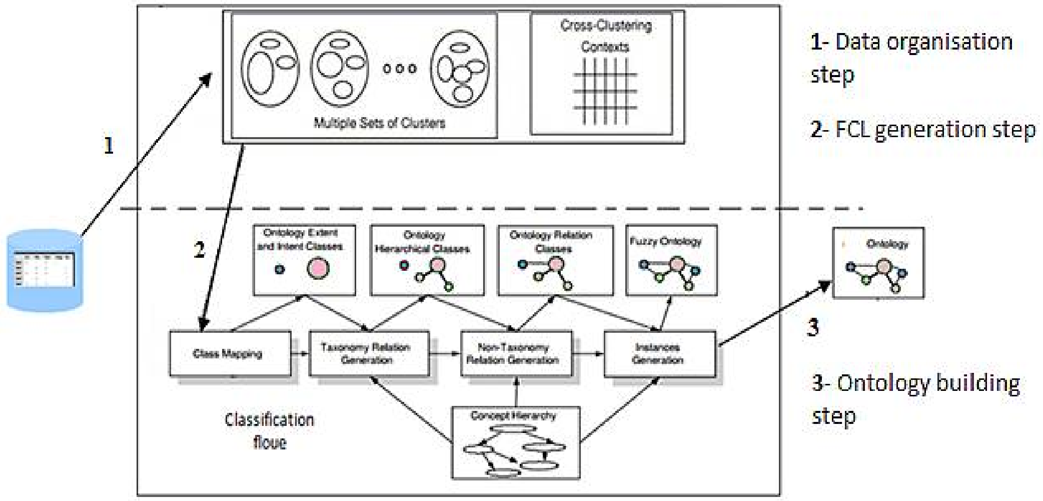
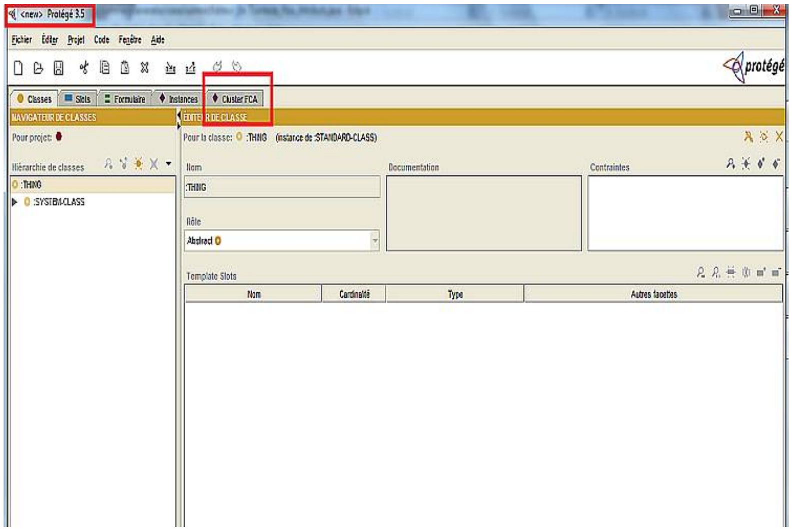

Cross-Clustering Contexts
Multiple Sets of Clusters
1- Data organisation step
2- FCL generation step
1
2
Ontology Extent and Intent Classes
Ontology Hierarchical Classes
Ontology Relation Classes
Fuzzy Ontology
Ontology
Class Mapping
Taxonomy Relation Generation
Non-Taxonomy Relation Generation
Instances Generation
3
3- Ontology building step
Classification floue
Concept Hierarchy

Figure 3. Architecture of FOD Tab plug-In

5.2 Plugin Install

FOD Tab plug-in provides Protégé-3.5 users a novel way to visualize their ontology using the formal concept analysis (FCA) technique. We configure a tab plug-in in Protégé-3.x platform and we visualize a new Tab added in this environment (Figure 4).



Pour la classe: :THING (instance de :STANDARD-CLASS)

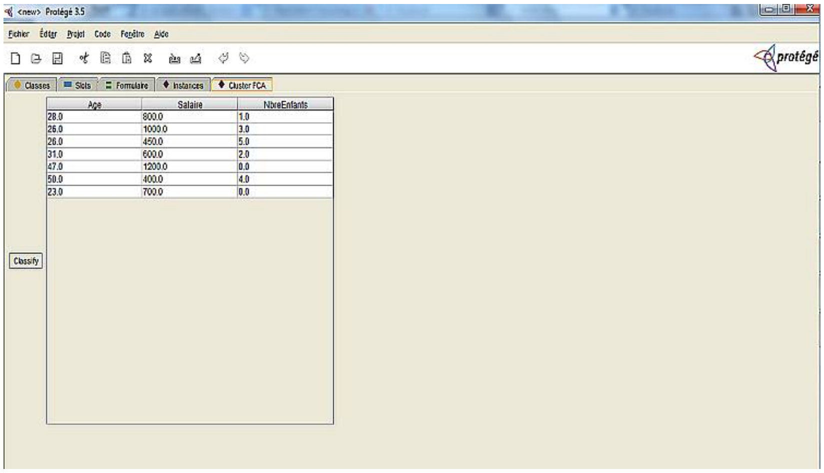
Nom :THING

Rôle Abstract

Figure 4. Install of the FOD Tab plug-in

5.2 Data processing

Figure 5 shows that the user can load the data starting from a text file. If the user select the menu “Open File” a selection windows of the desired file is open for him. We use the same example "Employe" to explain how to use FOD Tab plug-in. After opening the example project, the following information will be shown to the user:

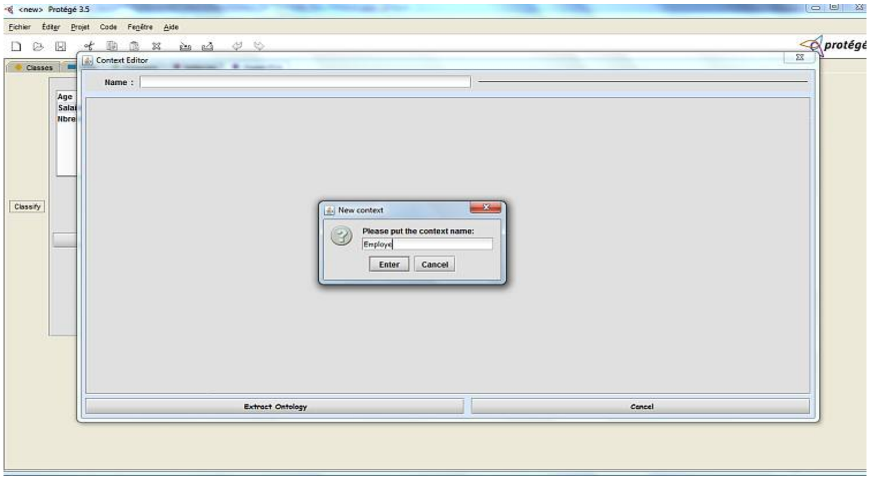


Age	Salaire	NbreEnfants
28.0	800.0	1.0
26.0	1000.0	3.0
26.0	450.0	5.0
31.0	600.0	2.0
47.0	1200.0	0.0
50.0	400.0	4.0
23.0	700.0	0.0

Figure 5. Choice of the data source

5.2 Visualization the clustering result

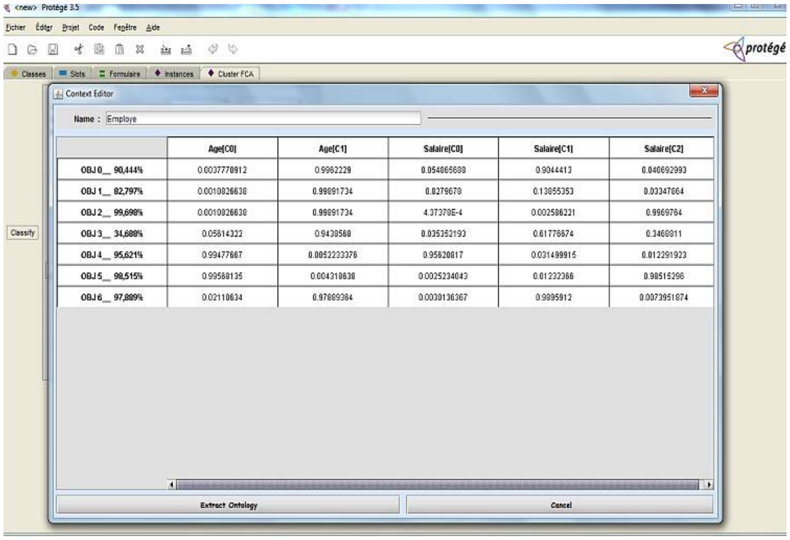
When clicking on Classify, it will have execution of the clustering algorithm with the user-introduced parameters. (Figure 6)



Name :

Please put the context name:
Employe

Figure 6: Define the parameter of clustering



Name : Employe

	Age[C0]	Age[C1]	Salaire[C0]	Salaire[C1]	Salaire[C2]
OBJ 0__ 90,444%	0.0037770912	0.9962229	0.054065688	0.9044413	0.040692993
OBJ 1__ 82,797%	0.0010826638	0.99891734	0.8279678	0.13855353	0.03347864
OBJ 2__ 99,698%	0.0010826638	0.99891734	4.37378E-4	0.002586221	0.9969784
OBJ 3__ 34,688%	0.05614322	0.9438568	0.035352193	0.61776874	0.3468811
OBJ 4__ 95,621%	0.99477667	0.0052233376	0.95620817	0.031499915	0.012291923
OBJ 5__ 98,515%	0.99568135	0.004318638	0.0025234043	0.01232366	0.98515296
OBJ 6__ 97,889%	0.02110634	0.97889384	0.0030136367	0.9895912	0.0073951874

Figure 7. Clustering result

5.4. Visualization of the generated Ontology

When clicking on Generate Ontology, it will have execution of the mapping algorithm from the formal context to the fuzzy ontology (Figure 8).

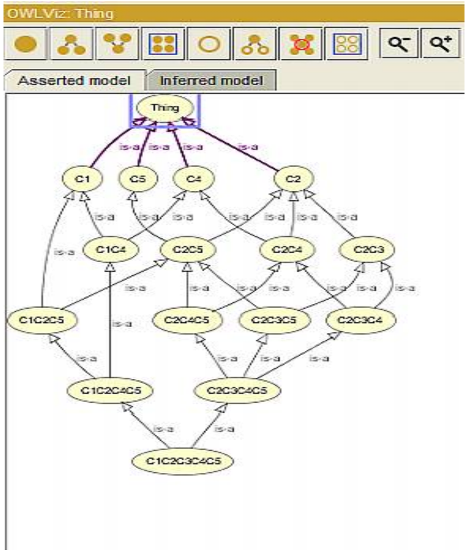


Figure 8. Generated ontology

Many advantages are granted from this generated ontology. In fact, the number of concepts generated is less than the number of classes if we built the ontology with the classic approaches (without clustering). The decrease mainly depends on the number of clusters that we choose at the begining which leads consequently to a considerable reduction in the number of classes composed the ontology. Besides, our approach is applicable to any type / amount of data. Indeed, as part of our support system to build an ontology using the AFC, the experts in each field could reach a mass of acceptable information, Fortunately, formal Concept Analysis helped us to structure and then build reduced ontologies which are able to express user domain easily.

6. CONCLUSION

Many researchers have been working in constructing ontology from many kinds of information sources[13]. But, unfortunately these proposals does not start from any kind of unclassified knowledge. We believe, following our studies, that ontology creation from classified data is an important line of research.

In this sense, our general methodological framework takes into consideration another degree of granularity in the process of generation and extracts ontology between classes resulting from a preliminary classification of the data and, in consequence, the proposed platform, FOD Tab plug-in aims at building ontology from scratch.

This approach promises a fast building of ontologies with reduced manual effort. Our evaluation for ontologies shows that our approach for building ontology using FCA, fuzzy logic and clustering is not only highly efficient but can also achieve better match quality than with a direct building of ontologies with classic approaches.

As futures perspectives of this work, we will focus on flexible interrogation of the generated ontology and how we could use it to improve the future retrieval information from large dataset to yield perfect and real time results with unique power of intelligence in order to interpret finally the best possible solution for the user query.

REFERENCES

[1] Aussenac-Gilles, N., Biebow, B. & Szulman, S., (2000) Revisiting ontology design: a methodology based on corpus analysis, *In EKAW00, 12th International Conference in Knowledge Engineering and Knowledge Management*, Lecture Notes in Artificial Intelligence, vol. 1937, Springer, Berlin, pp. 172–188.

[2] Bernaras, A. , Laresgoiti, I. & Corera, J., (1996) Building and reusing ontologies for electrical network applications, *In ECAI96, Proc. European Conference on Artificial Intelligence*, Budapest, Hungary, pp. 298–302.

[3] Klein, M. & Fensel, D. (2001) Ontology versioning on the Semantic Web, *in: First International Semantic Web Workshop (SWWS01),* Stanford.

[4] Klein, M., Fensel, D., Kiryakov, A. & Ognyanov,D., (2002) Ontology versioning and change detection on the Web, *in: A.Gomez-Perez, V.R. Benjamins (Eds.), 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02*), Lecture Notes in Artificial Intelligence, vol. 2473, Springer, Berlin.

[5] Arpırez, J.C., Corcho, O., Fernandez-Lopez, M., & Gomez-Perez, A., (2001) WebODE: a scalable ontological engineering workbench, *in: First International Conference on Knowledge Capture (KCAP01),* ACM Press, Victoria, pp. 6–13.

[6] Fernandez-Lopez, M. , Gomez-Perez, A., Pazos-Sierra, A. & Pazos-Sierra, J., (1999) Building a chemical ontology using METHONTOLOGY and the ontology design environment, *IEEE Intelligent Systems & their applications*, Vol. 4, No.1, pp.37–46.

[7] Fernandez-Lopez, M., Gomez-Perez, A. Juristo, N., (1997) METHONTOLOGY: From Ontological Art Towards Ontological Engineering, *AAAI Symposium on Ontological Engineering, Stanford.*

[8] Ganter, Bernhard & Wille, Robert, (1999) Formal Concept Analysis: Mathematical Foundation. Springer, Heidelberg.

[9] Gomez-Perez, A., (1998) Knowledge sharing and reuse, J. Liebowitz (Ed.), Handbook of Expert Systems, CRC, New York, Chapter 10.

[10] Gomez-Perez, A., Fernandez-Lopez, M., de Vicente, A., (1996) Towards a Method to Conceptualize Domain Ontologies, *In ECAI96*, *Workshop on Ontological Engineering*, Budapest, pp41–51.

[11] Gomez-Perez A., (1996) A framework to verify knowledge sharing technology, *Expert Systems with Application*, Vol.11, No.4, pp519529

[12] Gomez-Perez, A., (2001) Evaluation of ontologies, *International Journal of Intelligent Systems*, Vol. 16, No. 3, pp.1–10

[13] Manzano-Macho D., Gómez-Pérez A., (2000) A Survey of Ontology Learning Methods and Techniques. Deliverable 1.5. OntoWeb: Ontology-based Information Exchange Management,. Available at www.ontoweb.org.

[14] Gruninger, M. and Fox, M.S., (1995) Methodology for the design and evaluation of ontologies, Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal,.

[15] Corcho, O. , Fernandez-Lopez, M., Gomez-Perez, A., Vicente,O., (2002) WebODE: an integrated workbench for ontology representation, reasoning and exchange, in: A. Gomez-Perez, V.R. Benjamins (Eds.), *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02*), Lecture Notes in Artificial Intelligence, Springer, Berlin, Vol. 2473, pp. 138–153

[16] Blazquez, M., Fernandez-Lopez, M., Garcıa-Pinar, J.M., Gomez-Perez, A., (1998) Building ontologies at the knowledge level using the ontology design environment, in: B.R. Gaines, M.A. Musen (Eds.), *11th International Workshop on Knowledge Acquisition, Modeling and Management (KAW98),* Banff.

[17] Kalfoglou, Y., Robertson, D., (2000) Use of formal ontologies to support error checking in specifications, Studer (Eds.), *In EKAW00, 11th European Workshop on Knowledge Acquisition, Modeling and Management Engineering and Knowledge* Management, Lecture Notes in Artificial Intelligence, Springer, Berlin, Vol. 1937, pp97–112.

[18] Lenat, D.B., & Guha, R.V, (1990) Building large knowledge-based systems*: Representation and inference in the cyc project. Boston,* AddisonWesley.

[19] Kietz, JU., Maedche, A., Volz, R., (2000) A method for semi-automatic ontology acquisition from a corporate intranet, In EKAW00, *Workshop on Ontologies and Texts, CEUR Workshop Proceedings,* Juan-Les-Pins, Vol. 51, No 4, pp14.

[20] Noy, N.F. & Musen, M.A., (2000) PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment, *In AAAI00, 17th National Conference on Artificial Intelligence* , Austin.

[21] Swartout, B., Ramesh, P., Knight, K. , Russ, T., (1997) Toward Distributed Use of Large-Scale Ontologies, *AAAI Symposium on Ontological Engineering*, Stanford.

[22] Staab, S. , Schnurr, H.P., Studer, R. , Sure, Y., (2001) Knowledge processes and ontologies, *IEEE Intelligent Systems*, Vol. 16, No 1, pp26–34.

[23] Uschold, M & King, (1995) Towards a Methodology for Building Ontologies, *In IJCAI95, Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal,.

[24] Uschold, M., (1996) Building Ontologies: Towards A Unified Methodology, Expert Systems, Cambridge.

[25] Uschold, M., Gruninger, M., (1996) Ontologies: Principles methods and applications, The Knowledge Engineering Review , Vol. 2, No11, pp93–155.

Authors

Aloui Amira is a PhD student at the National school of Engineer , Tunisia, Department of LIPH. She received the engineer degree in computer science from the National School of Engineer of Tunis in 2009 and the Master degree in automatic and signal processing from the National School of Engineer of Tunis in 2011.

Amel Grissa Touzi received the diploma of engineering in computer science and PhD in computer science from the Faculty of Sciences of Tunis, Tunisian in 1989 and 1994, respectively.