Use a domain ontology in CBR systems for fault diagnosis

N.Dendani
Laboratoire LabGED
Computer Science Department,
UniversityBadjiMokhtar,Annaba.
n_dendani@yahoo.fr

Med.Khadir
Laboratoire LabGED
Computer Science Department,
UniversityBadjiMokhtar,Annaba.
khadir@labged.net

S.Guessoum
Laboratoire LRS
Computer Science Department,
UniversityBadjiMokhtar, Annaba.
souadguessoum@yahoo.fr

***Abstract*— The maintenance of industrial systems is crucial for productivity, products quality and supplied services. Numerous computer systems are therefore developed for the task and must, in most cases, collaborate with each other. In the light of this statement, our work aims at realizing a system which consists in gathering the knowledge and the know-how in the field of fault diagnosis for steam turbines, by the construction of domain ontology. In order to better exploit the ontology and reason using its classes, sub-classes and instances a Case Based Reasoning (CBR) paradigm is chosen, as it offers an ideal solution for diagnostic of real application systems. The paper, therefore, presents a current work which has for objective to develop a CBR application for fault diagnosis based on ontology, by using the API JColibri.**

Keywords- Fault diagnosis, domain Ontology, Cases based Reasoning , JColibri.

I. INTRODUCTION

Fault diagnostic became an important preoccupation in most industrial sites. The difficulty is bound at first to the complexity and the increasing variety of components, equipments, machines, processes requiring a significant knowledge. The difficulty comes also from the unavailability of experimented technicians "domain experts" to take care of all maintenance activities.

Companies perceived the importance of fault coverage in case of faults that are not settled (adjusted) in the planned time causing system inactivity and consequently production drop and costs increase. Indeed, the diagnosis is an intelligent act which is hardly programmable with classic techniques. Several studies have been conducted for the development of fault diagnosis methods based on Artificial Intelligence (AI) methods and techniques.

The presented work aims to design a knowledge system in the field of fault diagnostic and maintenance field for steam turbines. This development thus, is based on knowledge modeling which is given by a representation model in the form of domain ontology. Ontologies proved their power in knowledge representation of industrial maintenance, as an example the PROTEUS platform [1].

Furthermore they allow clarification of data semantics and describe the field concepts regardless of all applications where they could be used. The formal ontological aspect allows reasoning abilities, either to verify the consistency of information, or to infer new knowledge. The consensual nature of ontologies permits to represent in the same manner concepts, in all systems of a community of practice. To exploit well this ontology and allow reasoning, we used the case based reasoning approach (CBR) [2] which offers another alternative to implement intelligent diagnostic systems for real applications. This alternative is motivated by the idea that an industrial expert intervenes to diagnose a fault, he tries to remember past experiences (experiments) of fault observed in similar situations which can lead to similar results [3], thus CBR techniques allow to replace the expert reasoning. It also offers the re-utilization of past experience facilitating knowledge acquisition and process sharing, creating the opportunity of learning from experiences.

Even though any Case-Based Reasoning (CBR) system relies on a set of previous specific experiences, its reasoning power can be improved through the use of general knowledge about the domain. In the CBR the design of build integrated systems that combine case specific knowledge with models of general domain knowledge is offered by COLIBRI (Cases and Ontology Libraries Integration for Building Reasoning Infrastructures), an environment to assist during the design of knowledge intensive CBR (KI-CBR) systems. The core of the COLIBRI architecture is CBROnto, an ontology developed as task/method ontology incorporating common CBR terminology and which also allows the integration several domain ontologies.

The paper is organized as follows: Section 2 gives the state of the art; describing systems which integrates case specific knowledge with models of general domain knowledge. The ontology concept, CBR and domain of application are given in the Section 3. The description of the developed approach for the representation of the domain model, the case model and their implementation as well as an explanation of the use of the principles of CBR for the fault diagnosis in steam turbines are given in section 4. We conclude our work in the Section 5.

II. STATE OF THE ART

The integration of the application generic knowledge in the KI-CBR systems is an important aspect in several projects. In CREEK architecture [4], we find a rather strong coupling between the cases knowledge and those of the domain. The cases are immersing in a generic model of the domain represented by a semantic network. Fuchs and Mille [5] have proposed a modeling of the CBR at the knowledge level. They have distinguished four knowledge models: the conceptual model of the domain describing the concepts use to describe the domain ontology independently of the reasoning; the case model which separates the case in 'problem, solution', and track of reasoning; the tasks reasoning models which include a model of specification and other one of tasks decomposition and; reasoning supports model. D'Aquin and al. [6] worked on the integration of the CBR in semantic Web. For that purpose, they have proposed an extension of OWL (Ontology Web Language) allowing representing the adaptation knowledge of the CBR. The expression of domain and cases knowledge in OWL allowed them to add to the CBR system the appropriate reasoning capacities of OWL by exploiting, for example, the subsumption and the instantiation. Diaz-Agudo and Gonzalez Calero [7] proposed an architecture independent from the domain which helps to integrate ontologies in CBR applications. Their approach consists in building integrated systems which combine cases specific knowledge with generic models of the domain knowledge. They presented CBROnto [8], as task / method ontology which supplies the necessary vocabulary to describe implied elements in the CBR processes, and which also allows to integrate various domain ontologies. CBROnto was later reused by jCOLIBRI, an object-oriented framework built in Java, rather powerful for the construction of CBR systems [9]. JCOLIBRI separates the management of the cases bases in two aspects: the obstinacy and the organization in memory, what allows having various storage media of case (files text / XML, ontology, etc.) accessible via specific connectors.

III. MODELLING KNOWLEDGE

A. *Ontology*

Knowledge capitalization process consist in marking the crucial knowledge (know and know-how) that are necessary to the processes of decision. So it's important to identify; then to formalize and model the explicit knowledge in order to memorize them. One of the proposed methods is the construction of the ontology [10]. The following definition has been given to the ontology in [10] "to make ontology, is to decide of the individuals who exist, the concepts and properties that characterize them and the relations that link them".

Gruber [11] defines the ontology as:"An ontology is an explicit specification of a conceptualization". Since then, a number of definitions for an ontological construction has been given. In 1997 Borst [12] added the terms shared and formal to Gruber's definition giving: "An ontology is a formal specification of a shared conceptualization". One year later both definitions were merged into one [13], giving: "An ontology is a formal, explicit specification of a shared conceptualization". The type of an ontology is closely related to its conceptualization objects such as: knowledge representation, high level, generic, domain and application. In our case the developed ontology is of domain type, as it contains a number of concepts and a certain vocabulary that defines a targeted domain i.e., the steam turbine and its maintenance aspects

B. *Case based reasoning (CBR)*

The processes that make up case-based reasoning can be seen as a reflection of a particular type of human reasoning. In many situations, the problems that human beings encounter are solved with a human equivalent of CBR. When a person encounters a new situation or problem, he or she will often refer to a past experience of a similar problem. This previous experience may be one that they have had or one that another person has experienced. If the experience originates from another person, the case will have been added to the (human) memory through either an oral or a written account of that experience. The idea of CBR is intuitively appealing because it is similar to human problem solving behavior. Therefore, CBR involves reasoning from prior examples [2][14]: retaining a memory of previous problems and their solutions and solving new problems by reference to that knowledge. Descended of the research in artificial intelligence on the problems resolution, this principle of resolution can be described as follows [15]: Generally, a case-based reasoner will be presented with a problem, either by a user or by a program or system. The case-based reasoner then searches its memory of past cases (called the case base) and attempts to find a case that has the same problem specification as the case under analysis. If the reasoner cannot find an identical case in its case base, it will attempt to find a case or multiple cases that most closely match the current case. In situations where a previous identical case is retrieved, assuming that its solution was successful, it can be offered as a solution to the current problem. In the more likely situation that the case retrieved is not identical to the current case, an adaptation phase occurs. During adaptation, differences between the current and retrieved cases are first identified and then the solution associated with the case retrieved is modified, taking these differences into account. The solution returned in response to the current problem specification may then be tried in the appropriate domain setting.

At the highest level of abstraction, a case-based reasoning system can be viewed as a black box represented often by a cycle[15],[16] (Fig.1) that incorporates the reasoning mechanism and the following external facets:

- The input specification or problem case.
- The output that defines a suggested solution to the problem.
- The memory of past cases, the case base, that are referenced by the reasoning mechanism.

In many practical applications, the reuse and revise stages (Fig.1) are sometimes difficult to distinguish, and several researchers use a single adaptation stage that replaces and combines them. However, adaptation in CBR systems is still an open question because it is a complicated process that tries to manipulate case solutions [16]. Usually, this requires the development of a causal model between the problem space (i.e., the problem specification) and the solution space (i.e., the solution features) of the related cases.

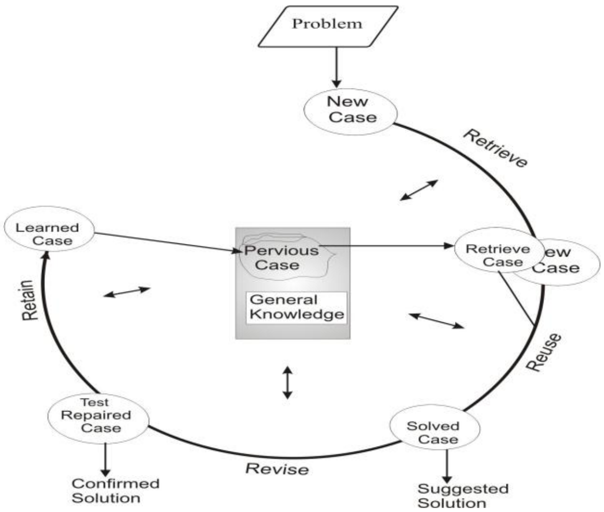

Figure 1. Case based reasoning phases

The feasibility of the CBR for the decision support where the experience of past situations is reused to manage new situations, has been shown in the survey of the decision making process [16]. The deepening of this mechanism (CBR) brings us to see behind a knowledge management process. In fact, the CBR and the knowledge management follow the same objective of acquisition and reuse of experience or knowledge.

C. Colibri/CBROnto

COLIBRI helps to design KI-CBR systems that combine specific cases with various knowledge types and reasoning methods. The major problem associated with the knowledge intensive approach to CBR is the so called knowledge acquisition bottleneck. Diaz-Agudo Gonzalez Calero [7] Proposed an approach of knowledge acquisition based on reusing knowledge from an ontology library to create complex, multirelational knowledge structures to support the CBR processes.

KI-CBR system should be able to take advantage of the acquired domain knowledge. COLIBRI views KI-CBR systems as consisting of collaborating knowledge components, and distinguishes different types of knowledge[17]. *Ontologies* describe the structure and vocabulary of the Domain Knowledge that refers to the actual collection of statements about the domain. Tasks correspond to the goals that must be achieved. PSMs capture the problem-solving behavior required to perform the goals of a task. And Inferences describe the primitive reasoning steps in the problem solving process.

COLIBRI uses CBROnto as a unifying framework that structures and organizes different types of knowledge in KI-CBR systems according to the role that each one plays. CBROnto captures CBR semantically important terms, includes CBR dependent but domain-independent terms, and aims to unify case specific and general domain knowledge representational needs.

D. Steam turbines

Steam turbines are mechanical devices using superheated steam power, and convert it into useful mechanical work. In the studied case, the mechanical work produced is used for electrical production. The steam is created by a boiler, where pure water passes through a series of tubes and then boils under high pressure to become superheated steam. The heat in the firebox is normally provided by burning fossil fuel (*e.g.* coal, fuel oil, or natural gas as in the studied case). The superheated steam leaving the boiler then enters the steam turbine throttle, where it powers the turbine and connected generator to make electricity. After the steam expands through the turbine, it exits the back end where it is cooled and condensed back to water in the surface condenser. This condensate is then returned to the boiler through high-pressure feed pumps for reuse. Heat from the condensing steam is normally rejected from the condenser to a body of water; in the studied case sea water is used. Because of the importance of the steam turbines in the process of the economic development, maintenance operation of these equipments is of a fundamental importance. It permits to reduce the inactivity time of equipments that is *very expensive.*

IV. THE ARCHITECTURE OF OUR SYSTEM

In this section we describe our system architecture where we used the API jCOLIBRI [18] as the base for the architecture. More of specifically our system [18] is based on the extension of ontology jCOLIBRI where we have included different variations. Our system includes mechanisms to retrieve, reuse, revise and retain cases and it is designed to be easily extended with new components.

A. Domain model

This model represents the domain knowledge in the ontology form. In the KI-CBR systems, the ontologies play an important role [19], as vocabulary to describe the cases, as knowledge structure where the cases are located, and as knowledge source allowing the semantic reasoning in the methods of similarity calculation .

In this work, we use a domain ontology Onto-turb developed by Djeddi[20] built with Protege [21], used to store the diagnosis cases of the steam turbines equipments for a central system established by the following concepts:

The class "centrale" is cut in five subclass representing the edges of the “centrale”:

00-Equipement, 30-Groupe3, 40-Groupe4, 80-Poste HT, 90-Génie Civils. These last ones are cut in turn in subclasses representing the systems.

For the needs of the application, we added to the ontology hierarchy two other class, the class "Equipment" which is going to represent all the equipments, and the class "System" which represents all the systems (Fig.2).

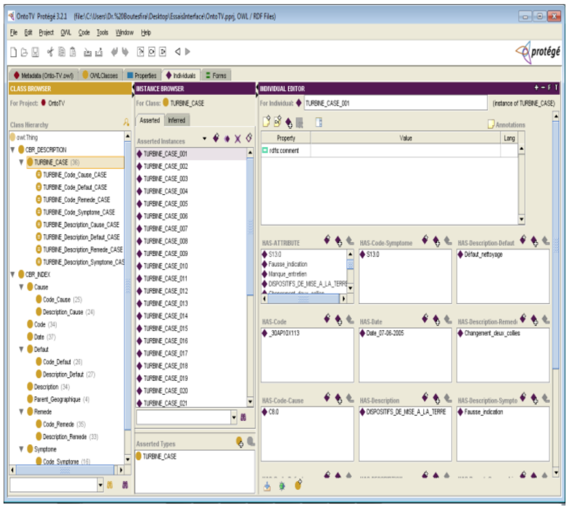

Figure 2. The classes hierarchy in "Onto-Turb"

B. *Case model*

A case in our system is a diagnosis case. it is generally described by a couple (problem, solution). According to our model, the description parts correspond to the problem part, and the causes part corresponds to the solution part. To improve the communication between the case base and the domain model, our case model is represented by means of an ontology which integrates the domain model. We are inspired in it of the jCOLIBRI approach. This ontology contains the roots concepts following ones:

- CBR-CASE : it subsumes the concepts representing the various types of case it can exist in the system.
- CBR-DESCRIPTION : it subsumes the concepts representing the parts of a case (Description, solution).
- CBR-INDEX : it allows to integrate the concepts of the domain model.

The cases are then represented by ontology instances and they thus have two attributes types:

- Simple attributes corresponding to data-type properties of the ontology which take simple values, i.e. string, integer, real, etc.
- Complex attributes corresponding to instances of the ontology.

The creation of CBR application by jCOLIBRI takes place in several stages:

1. The definition of the case structure in which the user explains to the system which are the data which he uses to represent the cases. The structure is saved under the XML file for the needs of the application modules (Fig.3).

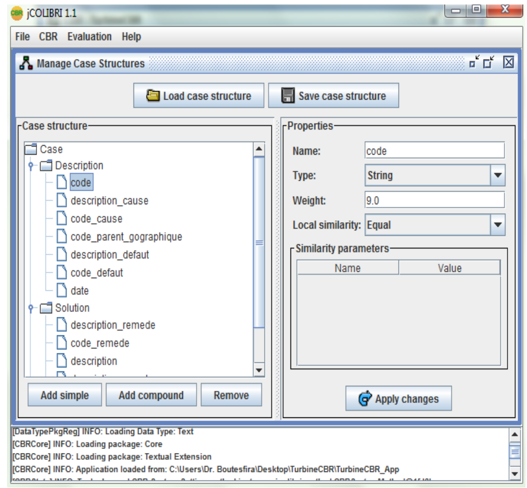

Figure 3. Creation of the case structure

2. The definition of the Connector which the user explains how are stored the data previously described. It is consist here of describing the used file format (basic file SQL, simple file text) to allow the jCOLIBRI modules for interfacing with the data file (Fig.4).

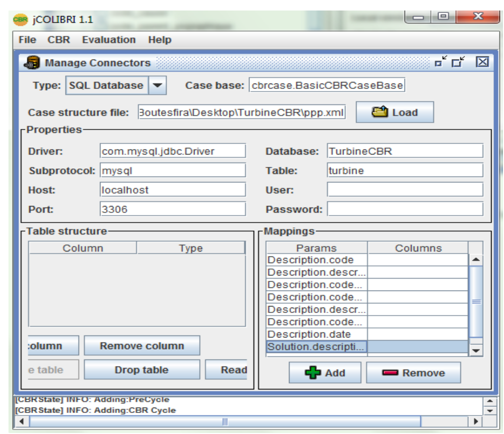

Figure 4. Creation of a connector

3. The definition of new methods if need for using in jCOLIBRI.

The effective instantiation of 3 data processing phases (Fig.5):

- The pretreatment of the data including the load of the data and their forms.
- The data processing it is the core of the application, applying the principle of CBR.
- The post-treatment it is the saving of the data in a persistent layer.

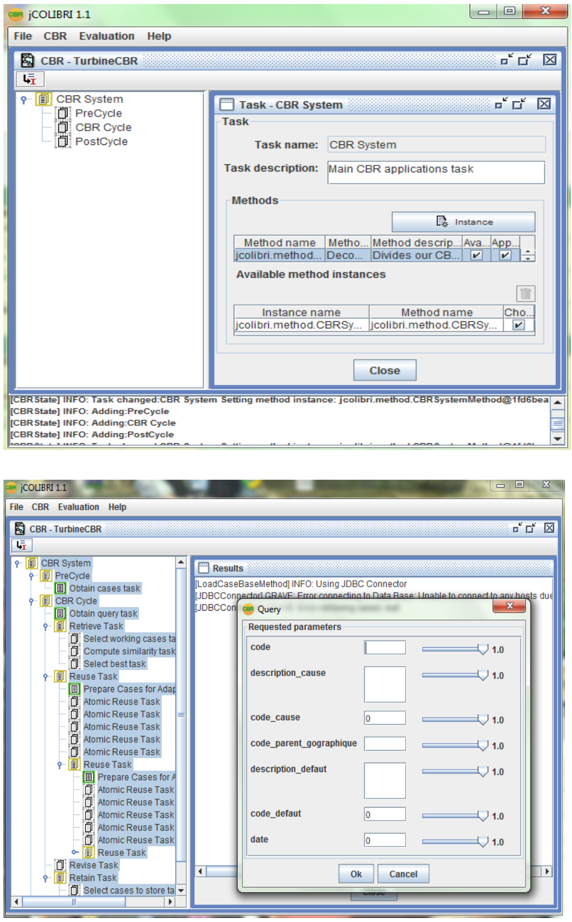

Figure 5. Configuration the CBR Application

C. CBR process

In our study we focused two phases of CBR cycle: describe the new case and retrieve sources cases, the other phases are in course of development and will be the object of a paper to come.

The cases are described by concepts of the domain model. Once the target case is elaborated we proceed to the retrieve of the cases sources. In this phase, similarity measures are used to get back the similar cases with a target case. Generally, with the object-oriented structures of the cases, the similarity measures follow the "local-global" principle [21], [22] which says: " The purpose is to determine the similarity between two objects, an object representing the Case source (or a part of the case) and another object representing the case targets (or a part of the case). This similarity is called the global similarity, and is calculated by a recursive way; i.e. for every simple attribute, a measure of local similarity determines the Similarity between the two values of attribute. On the other hand, for every complex attribute, a global similarity measure is used. Finally, the values of the local and global similarity are included, of recursive way, to give the global similarity of the two compared objects. From an ontological point of view, the calculation of similarity between two concepts of the ontology can be divided into two constituents [21],[19]: a concept based similarity (or similarity intra-class) which depends on the level of the concepts in the ontology, and a slot based similarity (or similarity inter-class) which depends on common attributes values of the compared objects.

SIMILARITY MEASURE

The attributes of a target case still have no same importance in the calculation of similarity. So, it is important to allow the user to associate with every attribute certain weight. In our work, the weights can be attributed to two levels different from the target case:

- The simple attributes can have one of three fashions of calculation of similarity:
  - IGNORE: the attribute has no importance.
  - EXACT: it allows verifying the equality Strict of the attribute values.
  - NUMERIC: It is applicable in only Numeric attributes. More Two values are near one of the other one, more they are similar.
- The complex attributes have a weight in the interval [0,1]. Besides, every simple attribute of a complex attribute can have one of three modes of calculation (IGNORE, EXACT, NUMERIC).

We are going to explain, thereafter, the used similarity measures.

where Q = $\{q_i : 1 \le i \le n, n \in N^*\}$ a targets case for which we look for similar cases, or *qi* is a simple either complex attribute, and is $\Omega = \{C_j : 1 \le j \le k, k \in N^*\}$ the base of case, or $C_j = \{C_{jl} : 1 \le l \le m_j, m_j \in N^*\}$ The similarity was based concept, *simcpt*, is defined as follows: For every complex attribute, $q \in Q$ et $c \in C$,

$$sim_{cpt}(q,c) = w_q * \frac{2 * \mathrm{MAX}\left(prof\left(LCS(q,c)\right)\right)}{prof(q) + prof(c)} \quad (1)$$

Where w_q is the weight associated in *q*, *Prof* is the depth of a concept (or of instance) in the ontology, and *LCS* is “least Common subsume” of two instances. In a particular case, when *q* and *c* represent the same instance, we have:

Prof (LCS (q, c)) = Prof (q).

The based slot similarity, sim_{slt}, is defined as follows:

$$sim_{slt}(q,c) = \frac{\sum_{s \in CS} \mathrm{sim}(q.s, c.s)}{|CS|} \qquad (2)$$

Or *CS* is all the simple attributes in common enter *q* and *c*(CommonSlots), $|CS|$ its cardinality is, *q.s* (or *c.s*) represent the simple attribute of *q* (or of *c*), and *sim (q.s, c.s)* is the similarity between these two attributes. For the moment, We consider only the first two modes (IGNORE, EXACT), and thus *Sim (q.s, c.s)* is defined as follows:

$$\mathrm{sim}(q.s, c.s) = \begin{cases} 1 & if\,(w_{q.s} = exact) \wedge (v_{q.s} = v_{c.s}) \\ 0 & else \end{cases} \qquad (3)$$

Where *wq.s* is the mode associated with the attribute *q.s*, and *vq.s* is the value of this attribute in *q*. The global similarity measure between two complex attributes, *q* and *c*, is defined by the following formula [23]:

$$sim(q,c) = (1-\alpha) * sim_{cpt}(q,c) + \alpha * sim_{slt}(q,c) \qquad (4)$$

α is an experience parameter.

V. CONCLUSION

We presented in this article the construction and the design of a CBR system of which the knowledge within is described on ontological form. For it we used the API Jcolibri which uses CBROnto a task / method ontology of which supplies the necessary vocabulary to describe elements implied in the CBR process, and which also allows integrating various domain ontologies. Our ontology of domain Onto-Turb built with Protege used to store the fault diagnosis cases for steam turbines. For the CBR process we centered our work on two phases of the cycle for instance the elaboration and retrieve of the cases. At present, we are developing the other phases of the process and studying the possible links with the ontology domain representing the system knowledge.

REFERENCES

[1] I.Rasovska, B.Chebel-Morello, et N. Zerhouni,(2005). Process of s-maintenance: decision support system for maintenance intervention, Emerging Technologies and Factory Automation, *ETFA 2005,10th IEEE, pp.8-15. (references)*

[2] A. Aamodt, E. Plaza. "Case Based Reasoning: Foundational issues, methodological variations, and system approaches". *AI communications,* IOS press, Vol.7: 1(1994), pp39-59

[3] J.Lieber."Contribution à la conception de systèmes de raisonnement à partir de cas" , HàDR de l'université Henri Poincaré –Nancy1, janvier 2008.

[4] A. Aamodt."Explanation-Driven Case-Based Reasoning ''. Lecture Notes In *Computer Science,* (1994) *pp274-274.*

[5] B.Fuchs & A.Mille. Une modélisation au niveau connaissance du raisonnement à partir de cas. In L'HARMATTAN, Ed., Ingénierie des connaissances (2005).

[6] M.D'aquin, J.Lieber & A.Napoli. Artificial Intelligence: Methodology, Systems, and Applications, Volume 4183/2006 of Lecture Notes in *Computer Science*, chapter Case-Based Reasoning Within Semantic Web Technologies. Springer Berlin / Heidelberg

[7] B.Diaz-Agudo & P.Gonzalez-Calero . An architecture for knowledge intensive CBR systems.Advances in Case-Based Reasoning–(EWCBR'00). Springer-Verlag, Berlin Heidelberg New York (2000).

[8] B.Diaz-Agudo & P.Gonzalez-calero. CBROnto : a task/method ontology for CBR. Procs. Of the 15th International FLAIRS, 2, 101–106 , (2002).

[9] B.Diaz-Agudo , P.Gonzalez-Calero, & A.Sanchez-Ruizgranados (2007). Building CBR systems with jcolibri. Science of Computer Programming, 69(1-3), 68–75. Special issue on Experimental Software and Toolkits.

[10] J. Charlet ,"L'ingénierie des connaissances, entre science de l'information et science de gestion "(2004). *Rapport de recherche SIC 805*, http://archivesic.ccsd.cnrs.fr.

[11] T.R.Gruber, "A translation approach to portable ontology specifications". Knowledge Acquisit, 5(2),1993.

[12] W.N. Borst, "Construction of engineering ontologies for knowledge sharing and reuse". CTIT Ph.D. Thesis Series No. 97-14, (1997), Universiteit Twente.

[13] R.Studer, , Benjamins, V.R., and Fensel, D. Knowledge engineering: Principles and methods, Data and Knowledge Engineering Vol 25(1-2), 1998, Pages 161-197

[14] J. Aha, "The omnipresence of case-based reasoning" in *science and application*. Knowledge –Based Systems 11 (5-6), (1998), pp 261-273

[15] B.Fuchs, J. Lieber, A. Mille, A. Napoli. "A general strategy for adaptation in case based reasoning ", Technical report RR-Liris-2006-016 , LIRIS /CNRS/INSA (2006).

[16] A. Mille, "From case-based reasoning to traces-based reasoning", Elsevier, Annual reviews in control 30, (2006), pp223-232

[17] Van Heijst, G.; Schreiber, A.; and Wielinga, B. Using explicit ontologies in knowledge based systems development. *International Journal of Human and Computer Studies* 46(2/3),(1997).

[18] J.Recio-Garcia, B.Diaz-Agudo, P.Gonzalez-calero & A.Sanchez: Building cbr systems with jcolibri. special Issue on Experimental Software and Toolkits of the *Journal Science of Computer Programming 69(1-3)* (2007)

[19] J.Recio-Garcia, B.Diaz-Agudo, P.Gonzalez-calero & A.Sanchez Ontology based CBR with jCOLIBRI. Applications and Innovations in *Intelligent Systems, 14,* (2006), *149–162.*

[20] O.Djeddi Conception et realization d'un algorithme d'alignement d'ontologie en OWL these de Master à l' université Badji Mokhtar annaba Algerie,2009

[21] http://protege.stanford.edu

[22] Bergmann & Stahl. Similarity measures for object-oriented case representations. In Proceedings of the European Workshop on Case-Based Reasoning, EWCBR'98.

[23] M. Richter "Case-Based Reasoning on Images and Signals", volume 73/2008 of Studies in Computational Intelligence, chapter Similarity, p. 25–90. Springer Berlin / Heidelberg.

[24] K.Zhang, J.Tang, M.Hong, J.LI & W. WEI," Weighted Ontology-Based Search Exploiting Semantic Similarity". Lecture Notes In Computer Science, (2006),3841,498.