A Case based Reasoning System based on Domain Ontology for Fault Diagnosis of Steam Turbines

Nadjette Dendani-Hadiby and Mohamed Tarek Khadir

LabGED Laboratory, Computer Science Department,
Badji Mokhtar-Annaba University P-O Box 12, 23000 Annaba, Algeria
hadibi@labged.net, khadir@labged.net

Abstract

Industrial diagnosis is a domain where problems are recurrent and therefore previous documented solution can be successfully reused. Different methodologies may be implemented for a given diagnosis domain, one of the appropriate, appears to be Case-Based Reasoning (CBR). A CBR system is a combination of processes and knowledge called "knowledge containers", that allows to preserve and to exploit previous experiences. Its reasoning power can be improved through the use of general knowledge about the domain in question. CBR systems combining case specific knowledge with general domain knowledge models are called knowledge intensive CBR (KI-CBR). The present work aims to develop a CBR application for fault diagnosis of steam turbines that integrates a domain knowledge modeling in an ontological form. This system is view as a KI-CBR system based on domain ontology, built around jCOLIBRI a well-known framework to design KI-CBR systems.

Keywords: *Domain Ontology, Cases based Reasoning, KI-CBR, Fault diagnosis, jCOLIBRI*

1. Introduction

Fault diagnosis is one of the most regarded maintenance issue 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. One of these techniques is case-based reasoning (CBR) [1].

This choice 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 [2], thus CBR techniques allow to simulate 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. A CBR relies on several containers of knowledge. The source cases are, obviously, among those containers of knowledge, but a lot of systems also use additional knowledge sources as the "domain knowledge". The more correct and accurate the domain knowledge is, the better the CBR system's inferences will be. This additional knowledge 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 [3]. 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.

In CBR, the design integrated systems that combine case specific knowledge with domain ontology 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 [4], an ontology developed as task/method ontology incorporating common CBR terminology and which also allows the integration several domain ontologies.

The presented work aims to design a hybrid knowledge system in the field of fault diagnostic and maintenance field for steam turbines and study how a synergy of ontology and CBR technologies could improve its efficiency.

The paper is organized as follows: Section 2 gives an outline on related work 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 proposed architecture is given in section 4. Section 5 presents the used paradigms and the implementation of the system architecture. We discuss our work in Section 6 and conclude it in the last one.

2. Related Work

The integration of the generic knowledge application in the KI-CBR systems is an important aspect in several projects. In CREEK architecture [5], we find a rather strong coupling between the cases knowledge and those of the domain. Aamodt's approach assumes the existence of a large, multifunctional knowledge base, where frames interconnected through their slots, form a semantic network of concept nodes and relational links. Although the user has an interactive role in knowledge acquisition and maintenance, for example, confirming conclusions or solving explanation conflicts, the system needs an initial densely coupled knowledge model that is not easy to acquire.

Fuchs and Mille [6] 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 [7] 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.

Bichindaritz has demonstrated the use of ontologies for facilitating case structuring and acquisition [8]. Diaz-Agudo and Gonzalez Calero [9] 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 [4], 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. Our approach is very much related with the latter and it fits our architecture which is based on the reuse of past design and implementation of problem solving methods and formalizes the CBR knowledge using a domain –independent CBR ontology CBROnto.

3. Modeling Knowledge

The proposed approach combines AI paradigms previously cited, for instance Ontologies and CBR. Both paradigms contribution in the context of knowledge capitalization and diagnosis for industrial purposes are explained in what follows.

3.1 Ontology

Knowledge capitalization process consists in marking the crucial knowledge (know and know-how) that are necessary to the decision processes. 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 domain ontology [10] where, the following definition has been given “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 have 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.

3.2 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 [1, 14]: retaining a memory of previous problems and their solutions and solving new problems by reference to that knowledge. The principle is inspired from AI researches on problems solving and may 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] (Figure 1) that incorporates the reasoning mechanism and the following steps:

- 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.

Many practical applications, the reuse and revise stages (Figure 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.

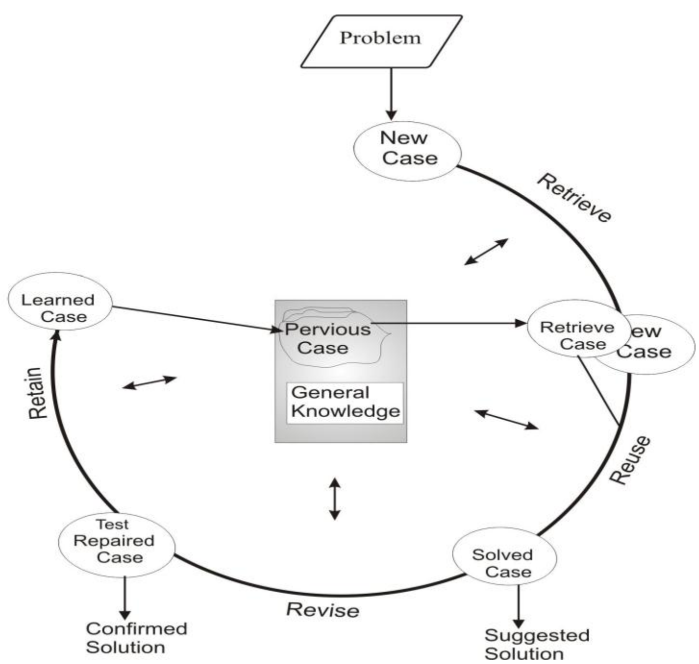

Figure 1. Case based Reasoning Phases [1]

The feasibility of the CBR for decision support where the experience of past situations is reused to manage new situations has been shown in survey on decision making process [16]. In fact, CBR and knowledge management follow the same objective of acquisition and reuse of past experience or knowledge.

3.3 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*.

4. System Architecture

The proposed approach includes the use of ontologies to build models of general domain knowledge. Although in a CBR system the main source of knowledge is the set of previous experiences, our approach to CBR is towards integrated applications that combine case specific knowledge with models of general domain knowledge. The more knowledge is embedded into the system, the more effective it is expected to be. Semantic CBR processes can take advantage of this domain knowledge and obtain more accurate results. In KI-CBR systems, ontologies play an important role [17], as a vocabulary to describe cases, as knowledge structure where cases are located, and as knowledge source allowing the semantic reasoning in the methods of similarity calculation, adaptation and learning.

Based on the latter, a novel ontological form of implementing reasoning process. The proposed architecture is composed of three functional components, domain ontology, CBR application and mediator (Figure 2).

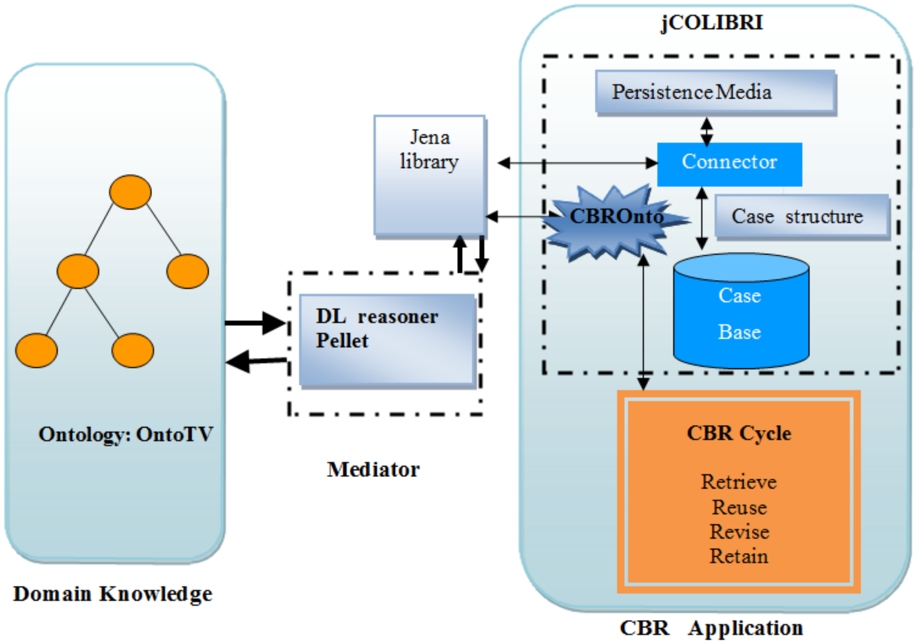

Figure 2. System Architecture

4.1 Domain Knowledge

The domain ontology is implemented to build general knowledge models and which includes vocabularies, concepts and relations for representing all knowledge concerning fault diagnosis of steam turbines equipment and its maintenance aspects.

4.2 CBR Application

This component is used for solving a diagnosis problem and dedicated to the case structure and to the four task of CBR process (Retrieve the most similar case/s, Reuse its/their knowledge to solve the problem, Revise the proposed solution and Retain the experience). The aforementioned tasks are mapped respectively to the following processes: a. Case representation, b. Case matching and retrieval, c. Case adaptation, and, d. Case-base maintenance. The focus is put only on the first three tasks, the last one is not modeled or implemented. As mentioned in section 4 the proposed approach proposes the use of an ontology library to build the domain model for knowledge-rich CBR applications. To take advantage of this domain knowledge, the CBR knowledge needed by the processes, should be expressed in a similar way. Thus, we propose to use an ontology for CBR termed CBRonto [9] that provides the vocabulary for describing the elements involved in CBR processes.

4.2.1. Case Representation: Cases in the case base should be described somehow by mean of the vocabulary provided by the domain model. The issue of case representation involves deciding on the type and the structure of the domain knowledge within cases. To represent the description of a case, specifically, CBR ontology has three classes:

- CBRCASE subsumes the concepts representing the various types of case it can exist in the system.
- CBRDESCRIPTION subsumes the concepts representing the parts of a case (Description, solution).
- CBRINDEX contains the indexing to the structure and content of the case base.

4.2.2. Case Retrieval: When we deal with ontologies, the concept hierarchy influences similarity assessment. Intuitively, it is obvious that the class hierarchy contains knowledge about the objects similarity. There are two main approaches to similarity computation [18]:

- *Classification based retrieval* builds a concept or an individual description using the restrictions specified in the query. This concept/individual is then classified, and finally all its instances/siblings are retrieved.
- *Computational based retrieval* uses numerical similarity functions to assess and order the cases regarding the query. The use of structured representations of cases requires approaches for similarity assessment that allow comparing two differently structured objects, in particular, objects belonging to different object classes.

Similarity measures for structure case representations are often defined by the following general scheme [19]: The goal is to determine the similarity between two objects, i.e., one object representing the case (or a part of it) and one object representing the query (or a part of it). We call this similarity object similarity (or global similarity). The object similarity is determined recursively in a bottom up fashion, i.e., for each simple attribute, a local similarity measure determines the similarity between the two attribute values, and for each relational slot an object similarity measure recursively compares the two related sub-objects. Then the similarity values from the local similarity measures and the object similarity measures, respectively, are aggregated to the object similarity between the objects being compared.

4.2.3. Case Adaptation: Case adaptation plays a fundamental role in the ability of CBR systems to solve new problems. Case adaptation is a knowledge-intensive task and most CBR systems have traditionally relied on an enormous amount of built-in adaptation knowledge in the form of adaptation rules.

Our approach relies on the explicit representation of general terminological knowledge about the domain. That way, certain adaptation knowledge is explicitly represented in the domain knowledge taxonomy, as it indicates, for instance, that individuals that are close in the taxonomy are eventually interchangeable.

We propose to use an ontology based model and an adaptation scheme based mainly on deletions and substitutions [20, 21]. Dependencies within a case are explicitly represented in order to guide the adaptation. If an element is removed the dependent elements are also removed. If an element *e1* is substituted by other element *e2* then the dependent elements are substituted. The search for substitutes is guided by the ontology. Adaptation is required when the retrieved case does not fulfill all the required goals given in the query, or when it solves goals that are different from the ones in the query, deletion and/or substitution adaptation operators is needed. We propose an adaptation mechanism as a process that propagates changes from description to solution items, as follows:

- The list L of items in the solution that need to be adapted is obtained. We find these items following a relation path + a concept.
- Every item in L is substituted by a proper new item. The search of proper substitutes is accomplished as a kind of specialized search which takes advantage of the DL knowledge base organization and applies similarity functions to find a substitute that is similar to the substituted one.
- Substitute items that depend on other items of the solution that have already been adapted.

Specialized search, as described in [22], is a way of finding candidates for substitutions in a case solution, where instructions are given about how to find the needed item.

4.3. Mediator

The Description Logic (DL) Reasoner Pellet [23] is used as the mediator shown in Fig 2, and is responsible on one hand for keeping the consistency of the knowledge base and on the other hand for inferring new knowledge that was not explicitly asserted but can be deduced.

5. Implementation

Protege editor [24] is chosen for the manual generation and modeling of the domain ontology and jCOLIBRI [25] for building CBR application.

5.1. Domain Ontology

Protege is a Java-based open source software tool that can be used to develop knowledge based systems and to create ontologies [26, 27].

We use a domain ontology "Onto-turb" developed in our research laboratory by Djeddi and Khadir [28] built with Protege, used to store the diagnosis cases of the steam turbines equipments. This ontology was constructed from database of a central system at SONALGAZ company and domain experts were asked to validate it. Figure 3 illustrates the obtained ontology that can contain description of classes, properties and instances.

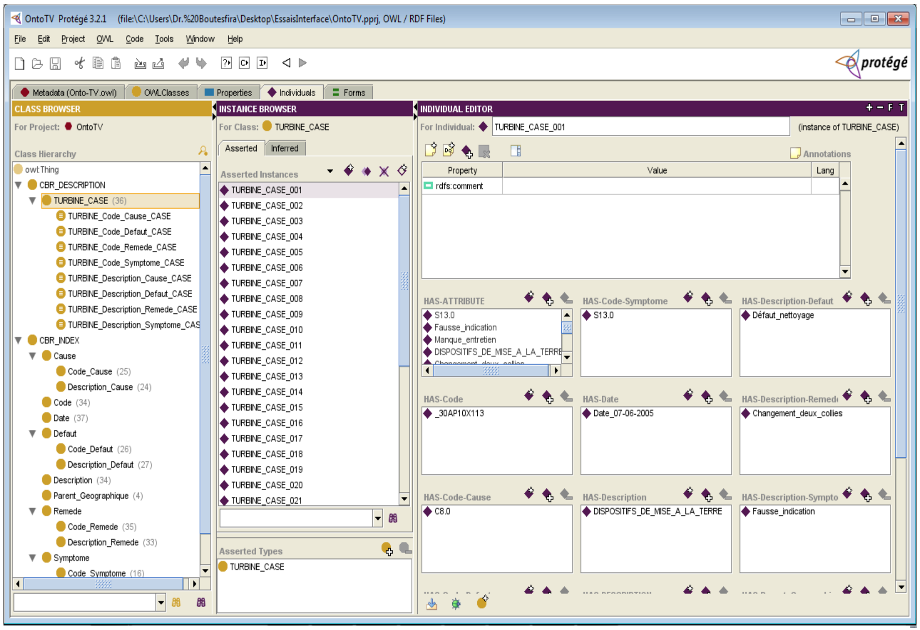

Figure 3. The Classes Hierarchy in "Onto-Turb"

5.2. CBR application

jCOLIBRI is an object-oriented framework in Java for building CBR systems, developed by GAIA group, is used to develop our CBR application. Expert programmers can use java to instantiate the framework, although the easiest way of using its graphical configuration tools. The underlying ideas of CBR can be applied consistently across application domains. However, developing a CBR system is a complex task where many decisions have to be taken. The system designer has to decide among a range of different methods for organizing, retrieving and reusing the knowledge retained in past cases. This process would greatly benefit from the reuse of previously developed CBR systems.

In jCOLIBRI, ontology is not represented as a new source; all concepts of CBR are mapped into classes and interfaces of framework. Classes that represent the ontology concept serve as templates where new CBR types should be added. They also provide the tasks and abstract interface of the methods. The design of the jCOLIBRI framework comprises a hierarchy of Java classes plus a number of XML files and is organized around the following elements [29]:

- Tasks and methods: The tasks supported by the framework and the methods that solve them are all stored in a set of XML files.
- Case-base: Different connectors are defined to support several types of case determination, from the file system to a database.
- Cases: A number of interfaces and classes are included in the framework to provide an abstract representation of cases that support any type of actual case structure.
- Problem solving methods: The actual code that supports the methods included in the framework.

jCOLIBRI version 1 is the first release of the framework. It includes a complete Graphical User Interface (GUI) that guides the user in the design of a CBR system.

The steps for designing a new CBR system in jCOLIBRI are given in what follows:

- **Definition of Case Structures:** A case is composed by three components: description (describes the problem), solution (represents a possible solution approach) and result (reveals if the proposed solution is able to solve the problem). Description and solution are collections of simple or compound attributes, permitting us to build a hierarchical case structure. By using jCOLIBRI GUI users are able to create the case structure defining simple and compound attributes that describe the cases together with their types, weights, similarity measure that is chosen from a library of existing similarity functions and parameters or others can easily be included. This generates a XML file with the structure information. When user has defined the case structure he configures a connector that uses that information for mapping the cases to the chosen persistence media. This mapping is also saved to a XML file. Figure 4 shows the definition of the diagnosis case parameters.

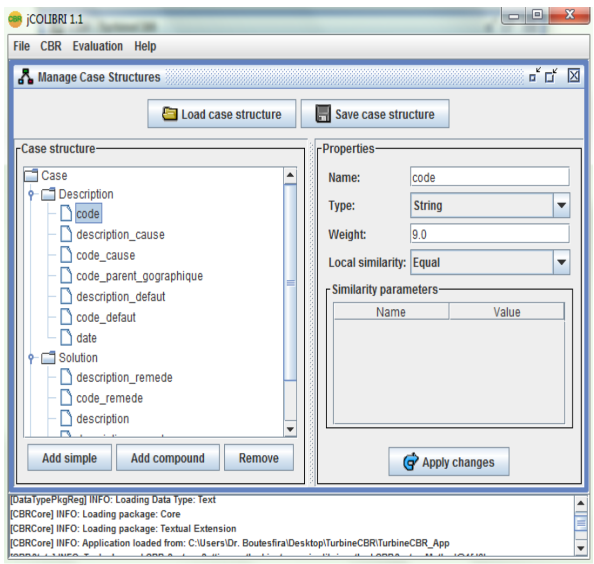

Figure 4. Creation of Case Structure

- **Building the Case-base:** The cases persistence concept is then built around those connectors representing objects that know how to access and retrieve cases from the storage media and return those cases to the CBR system in a uniform way. Therefore connectors provide an abstraction mechanism that allows users to load cases from different storage sources in a transparent way [30]. Defined connectors can work with plain text files, XML files, relational data bases or ontology. However, using ontologies as persistence media means that the slots of the case structure are defined by the concepts and properties of the ontology and the slots-filler are the instances of these concepts. This way, it makes no sense to do a representation of the case structure and then map it (using the connector) with the same case structure contained in the ontology. To solve this problem our pure ontological case structure represents directly the concepts and properties of the ontology using the Java classes exploited for reasoning. With this approach the connector for DLs does not need any configuration file and can load the cases from the ontology using only case structure information. Figure 5 shows how the case structure is mapped with ontology.

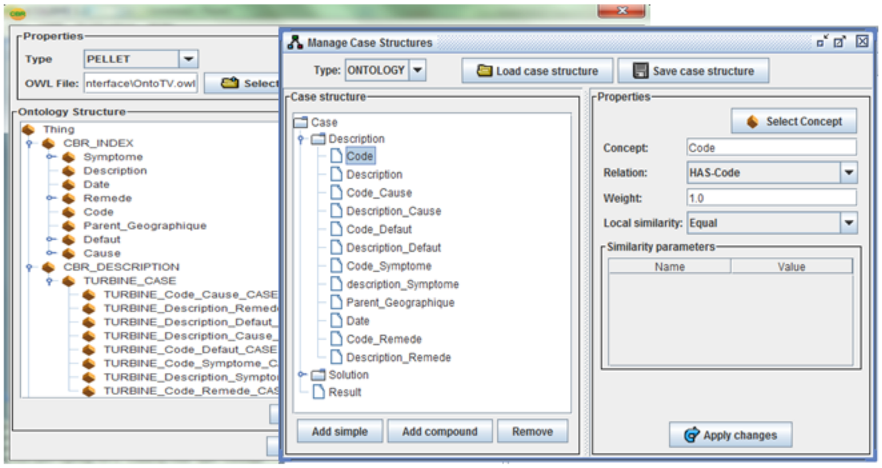

Figure 5. Case Attributes Mapping with Ontology

- **Managing Similarity Measures:** Computational based retrieval approach is used where numerical similarity functions are used to assess and order cases regarding the query. When two cases are compared, the local similarity functions are used to compare simple attribute values. Global similarity functions are linked to compound attributes and are used to gather the similarities of the collected attributes in a unique similarity value. At last, the similarity value of two cases is computed as the similarity of their description concepts. The available similarity measures are listed in a configuration file, and can be managed using the GUI. These functions compute the similarity between the query and the case, and are used to choose the most similar case to the query. There are two types of similarity functions: local functions that compute the similarity between simple attributes and global ones focusing on some sort of average over the local similarities.

  The similarity between query(*q*) and cases (*c*), *Sim (q, c)* is defined as follows:

$$Sim(q,c) = \frac{\sum_{s \in CS}(Sim(q.s, q.c).w_s)}{|CS|} \quad (1)$$

  Or w_s is the weight associated for each attribute ***s***, *CS* is all the simple attributes in q and *c*, $|CS|$ its cardinality , *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. Thus *Sim (q.s, c.s)* is defined as follows:

$$Sim(q.s, c.s) = \begin{cases} 1, & if\ v_{q.s} = v_{c.s} \\ 0, & else \end{cases} \quad (2)$$

  where v***q.s (or*** **vc.*s)*** is the value of this attribute in q(or in c)

- **Configuring the Behavior of the CBR Process:** jCOLIBRI formalizes the CBR knowledge using CBROnto, a knowledge level description of the CBR tasks and a library of reusable PSMs [30]. Configuration of tasks is done in an interactive approach by choosing from a library of reusable methods one that is suitable to solve the selected task. Constraints of the selected task are being tracked during the configuration process so that only applicable methods in the given context are offered to users. In our work, we focus only on the representation, retrieval and adaptation tasks. To achieve the retrieval, adaptation task, similarity functions and adaptation method respectively introduced in the previous point mast be included after used. The CBR application is finished when all the tasks have been configured. Users can test the system from inside the graphical interface. The first task of the CBR system, obtains the query which contains the description of problem (fault), (Figure 6) that is going to be used to retrieve the most similar cases, applying retrieval task. (Figure 7) shows the result of retrieval task , the most similar case is case number 001 with highest similarity 0.25

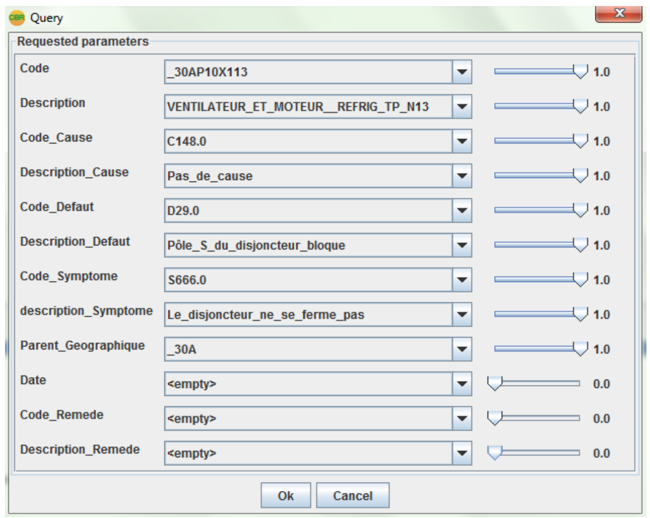

Figure 6. Query Task

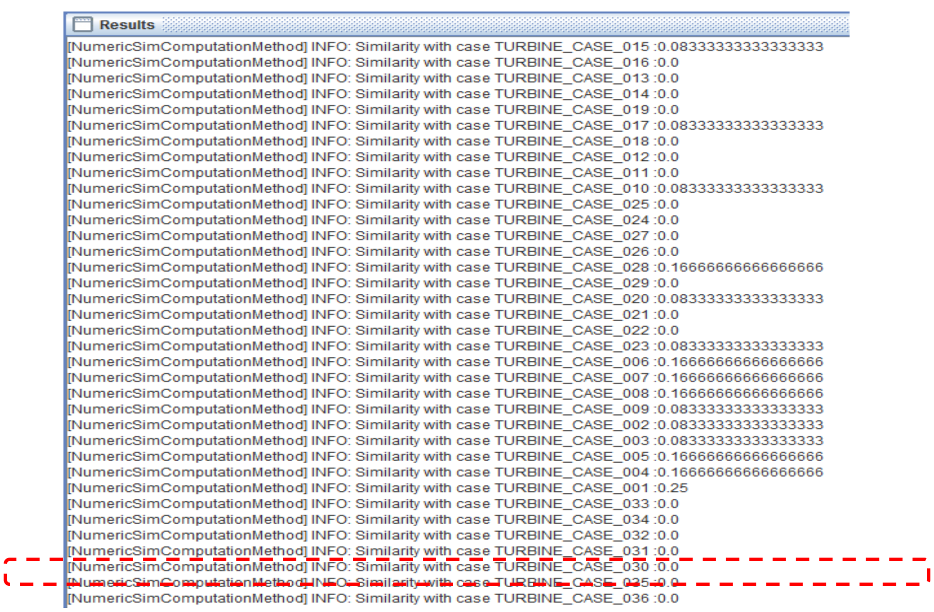

Figure 7. Result of Retrieval Task

6. Results and Discussion

The construction and the design of a CBR system of which the knowledge within is described on ontological form has been presented. 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 have been used. Our motivation for choosing this framework is based on a comparative analysis between it and other frameworks, designed to facilitate the development of CBR applications. jCOLIBRI enhances the other CBR shells: CATCBR, CBR*Tools, IUCBRF, Orenge. jCOLIBRI is and open source framework and their interface layer provides several graphical tools that help users in the configuration of a new CBR system. Another decision criterion for our choice is the easy ontologies integration. jCOLIBRI offers the opportunity to incorporate ontology in the CBR application to use it for case representation and content-based reasoning methods to assess the similarity between them.

For the CBR process we centered our work on three phases of the cycle for instance the elaboration, retrieve and adaptation of the cases based on the a domain ontology (Onto-turb). The main advantage of using a domain ontology in the CBR cycle is not only a concise and powerful description of the case, but also improved the research and facilitate the adaptation phases. The results obtained where validated by a human expert.

A simulation example is given in Fig. 6. The case base contains 36 real case, The Figure 7 shows the result of the retrieval task, returning the most similar case with highest similarity 0.25 Among 36 cases, for instance case number 001 turning to be validated as the most similar case by experts.

In case adaptation task, rules are composed by three parts: identification of the instance to adapt, condition to evaluate for performing the adaptation, and modification of the instance. These rules are not implemented in the chosen method described in Section 4.2.3. Adding those rules, for adapting solutions to the generic adaptation method, may improve substantially the adaptation phase.

7. Conclusion

This paper presents the advantages that ontologies provide when they are used in CBR systems. In the presented work, ontologies are used as:

- The cases/queries definition languages independently of the persistence media we use for cases;
- The knowledge structure where the cases are embedded, so that the persistence media for the case base uses the ontology formalization language; and
- The knowledge source to get semantic reasoning processes, like retrieval, similarity and adaptation.

The main innovation is that we use ontology to represent all this knowledge and DLs-like formalism to reason.

The usage of such KI-CBR for diagnosis purposes on an industrial application, demonstrates the power of integrating conceptualized domain knowledge in a decision solving problem approach such as CBR. The obtained results are promising and deserve to be tested on a larger case base.

References

[1] A. Aamodt and E. Plaza, "Case Based Reasoning: Foundational issues, methodological variations and system approaches", AI communications, IOS press, vol. 7, no. 1, **(1994)**, pp. 39-59.

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

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

[4] B. D-Agudo and P. G-Calero, "CBROnto: a task/method ontology for CBR". In S. Haller and G. Simmons, editors, Procs. of the15th International FLAIRS'02 Conference. AAAI Press, **(2002)**.

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

[6] B. Fuchs and A. Mille, "Une modélisation au niveau connaissance du raisonnement à partir de cas", in L'HARMATTAN, Ed., Ingénierie des connaissances, **(2005).**

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

[8] I. Bichindaritz, Thesis: "Case Based Reasoning Meets the Semantic Web in Biology and Medecine", Advances in Case Based Reasoning, 7th European Conference, LNAI, Spain, **(2004)**.

[9] B. D-Agudo and P. G-Calero, "An architecture for knowledge intensive CBR systems", Advances in Case-Based Reasoning". (EWCBR'00). Springer-Verlag, Berlin Heidelberg New York, **(2000).**

[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, vol. 5, no. 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, V. R. Benjamins and D. Fensel, "Knowledge engineering: Principles and methods", Data and Knowledge Engineering, vol. 25, no. 1-2, **(1998)**, pp. 161-197.

[14] J. Aha, "The omnipresence of case-based reasoning", in science and application. Knowledge –Based Systems, vol. 11, no. 5-6, **(1998)**, pp. 261-273.

[15] B. Fuchs, J. Lieber, A. Mille and 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, vol. 30, **(2006)**, pp. 223-232.

[17] J. R-Garcia, B. D-Agudo, P. G-Calero and A. Sanchez, "Ontology based CBR with jCOLIBRI", In Procs. ot the 26th SGAI Int. Conf. (AI-2006), **(2006)**, pp. 149–162, UK, Springer.

[18] B. D-Agudo and P. A. G-Calero, "Knowledge intensive CBR through ontologies", In Procs of the UK CBR Workshop. **(2001).**

[19] R. Bergmann and A. Stahl, "Similarity measures for object oriented case representations", In Advances in Case-Based Reasoning (EWCBR'98), **(1998)**, pp. 25–36. Springer.

[20] P. A. G-Calero, M. G-Albarrán, and B. D-Agudo, "A substitution-based adaptation model", In Challenges for CBR - Proc. of the ICCBR'99 Workshops. University of Kaiserslautern, **(1999)**.

[21] J. A. R-García, B. D-Agudo and P. A. G-Calero, "A distributed cbr framework through semantic web services", In Research and Development in Intelligent Systems XXII (SGAI 2005), **(2005)**, pp. 88–101, Springer.

[22] J. Kolodner, "Case-Based Reasoning", Morgan Kaufmann, San Mateo, **(1993)**.

[23] E. Sirin, B. C. Parsia, B. C. Grau, A. Kalyanpur and Y. Katz, "Pellet: A practical OWL-DL reasoned", Journal of Web Semantics, vol. 5, no. 2, **(2007)**.

[24] Protege, Ontology Editor and Knowledge Acquisition System, http://protege.stanford.edu/, **(2009)**.

[25] J. R-Garcia, B. D-Agudo, P. G-Calero and A. Sanchez, "Building cbr systems with jcolibri", Special Issue on Experimental Software and Toolkits of the Journal Science of Computer Programming, vol. 69, no. 1-3, **(2007)**.

[26] J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubezy, H. Eriksson, N. F. Noy and S. W. Tu, “The evolution of protege: an environment for knowledge-based systems development”, International Journal of Human-Computer Studies, vol. 58, no. 1, **(2003)**, pp. 89-123.

[27] H. Knublauch, R. W. Fergerson, N. F. Noy and M. A. Musen, “The protege owl plugin: An open development environment for semantic web applications”, Semantic Web - Iswc 2004, Proceedings, vol. 3298, **(2004)**, pp. 229-243.

[28] O. Djeddi and M. T.Khadir “Degnin and realisation of aligning ontology algorithm in OWL”, Master thesis at Badji Mokhtar university Annaba Algeria, **(2009)**.

[29] J. J. B-Tomás, J. A. G-Calero and B. D-Agudo, “JCOLIBRI: An Object-Oriented Framework for Building CBR Systems”, in Advances in Case-Based Reasoning, Lecture Notes in Computer Science, Springer Berlin/ Heidelberg, vol. 3155, **(2004)**, pp. 32–46.

[30] J. A. R-García, A. Sánchez, B. D-Agudo and P. A. G-Calero, “jCOLIBRI 1.0 in a nutshell. A software tool for designing CBR systems”, in M Petridis, editor, Proccedings of the 10th UK Workshop on Case Based Reasoning, CMS Press, University of Greenwich, **(2005)**, pp. 20–28.

Authors

DENDANI-HADIBY Nadjette

DENDANI Nadjette was born on the 2th of May 1972 in Annaba from Algeria; she graduated from the University of Badji Mokhtar Annaba, Algeria, with a state Engineering degree in Computer Science, in June 1996. She is a doctoral student since 2005, and joined the department of computer science in her original university of Badji Mokhtar Annaba in 2006 as an assistant teacher. Her main research interest is the study of AI approaches for acquiring and reasoning about knowledge and interested by the case-based reasoning (CBR) applied to the diagnosis domain.

KHADIR Mohamed Tarek

Pr. Khadir Med Tarek was born on the 5th of July 1972 in Annaba Algeria.After succeeding in the baccalaureat, majoring in Maths and Technology in 1989, he graduated from the university of Badji Mokhtar Annaba, Algeria, with a state Engineering degree in Electronics Majoring in Control, in 1995. After two years' work in the computer industry, he undertook an M.Eng. at Dublin City University, Ireland Graduating with First class honors in 1998. Tarek Khadir received a Ph.D. degree from National University of Ireland, Maynooth, Ireland in 2002. He then continued with this institution as a post-doctoral researcher until September 2003 when he joined the department of computer science at my university of origin, university Badji Mokhtar Annaba, Algeria, as a senior lecturer. He succeeded in obtaining the HDR (Habilitation to Direct Research) in January 2005, and promoted full Professor in 2010.