Knowledge-Intensive Decision Support System for Manufacturing Equipment Maintenance

Djamila Bouhalouan[1,2], Bakhta Nachet[1], Abdelkader Adla[1]
[1]Department of Computer Science
Oran 1 University
Oran, Algeria

[2]Department of Mathematics and Informatics
University Center of Ain Temouchent
Ain Temouchent, Algeria

Journal of Digital
Information Management

ABSTRACT: *To ensure continuous production in industrial plants, the high valued manufacturing eqipments should be kept in good working conditions. This brings plants to search for means to control and reduce equipment failures. When faults emerge in plants, appropriate actions for fault diagnosis and reparation must be executed promptly and effectively to prevent large costs due to breakdowns. To provide reliable and effective maintenance support, the aid of advanced decision support technology utilizing previous repair experience is of crucial importance for the expert operators as it provides them valuable troubleshooting clues for new faults. Artificial intelligence (AI) technology, particularly, knowledge-based approach is promising for this domain. It captures efficiency of problem solving expertise from the domain experts; guides the expert operators in rapid fault detection and maintenance. This paper focuses on the design and development of a Knowledge-Intensive Decision Support System (KI-DSS) for Manufacturing Equipment Maintenance in industrial plants to support better maintenance decision and improve maintenance efficiency. With integration of casebased Reasoning and ontology, the Ki-DSS not only carries out data matching retrieval, but also performs semantic associated data access which is important for intelligent knowledge retrieval in decision support system. A case is executed to illustrate the use of the proposed KI-DSS to show the feasibility of our ap proach and the benefit of the ontology support.*

Categories and Subject Descriptors
[H.3] Information Storage And Retrieval; [I.2] Artificial Intelligence

General Terms: Group Memory, Information Search, Decision Support Systems, Case-Based Reasoning

Keywords: Decision Support Systems, Knowledge-Intensive DSS, Case-Based Reasoning, Ontology, Owl, Maintenance, Diagnosis

Received: 19 October 2019, Revised 3 March 2020, Accepted 14 April 2020

Review Metrics: Review Scale: 0/6, Review Score: 4.25, Inter-reviewer consistency: 78.5%

DOI: 10.6025/jdim/2020/18/3/85-98

1. Introduction

Capitalizing knowledge is of a considerable contribution during organizational decision making activities which are often critical and repetitive (Adam, 2012). The development of a shared memory that stores the knowledge of expert members and their experience invoked in prior solutions can be useful for expert decision makers engaged in similar problem solving activities and will clearly assist them (Ackerman & Halverson, 2004). For an expert decision maker, it would be easier to reuse solutions and resolution schema corresponding to similar problems that have worked in the past than to analyze and solve the problems in scratch. Therefore, mechanisms to capture the experiential knowledge of experts can be of significant value to the organization in general, and the decision makers in particular.

A knowledge based approach shows significant promise for improving the effectiveness of maintenance activities support. It allows accumulating, organizing, storing, and sharing knowledge coming from past experience (Aamodt & Plaza, 1994). Among existing AI technology, Case-based reasoning (CBR) as an alternative reasoning paradigm and computational problem solving method has increasingly attracted more and more attention and grown in importance for businesses and academics over the past few years. The main principle of CBR is: *similar problems have similar solutions.* But, existing CBR systems lack semantic understanding, which is important for intelligent knowledge retrieval in knowledge-based systems (Guo et Al., 2012). To overcome this drawback, ontology technology is an ideal selection for realizing knowledge-based systems because ontology has not only powerful ability of knowledge representation, but also good semantic understanding. However, the explicit use of ontology based reasoning to support repetitive problem solving activities has received less attention.

To develop a such effective systems, two issues are critical: the first is how to find an effective method for case representation, which ensures domain knowledge can be acquired in an accurate easy manner, thus laying a good foundation for case retrieval; the next is how to find an appropriate method for case retrieval, which assures the right knowledge can be retrieved to solve a specific problem when a new task takes place. The system can not only carry out data matching retrieval, but also perform semantic associated data access, and improve the traditional keyword-based search. Through the semantic search capability, the hidden, but previously defined relations among data and concepts could be shown and represented if needed. These relations allow the user to understand the knowledge behind the stored data. Semantic technology could lead to the improvement in knowledge extraction, dissemination and management. Moreover, the use of common and unified domain ontology can improve the problem solving process where most of the decisions are dependent on individual experiences and domain knowledge of relevant managerial personnel.

The objective of this paper is to construct a maintenance intelligent knowledge-based system that can leverage the support of semantics. We suggest that the integration of ontology and CBR within a knowledge-based system is likely to provide additional information processing support. Ontology is used as a means to acquire domain knowledge and construct a case-base and use ontological semantic retrieval method as the case retrieval. Besides the case base, the system uses ontology (domain and task ontology related to the combustion machines to be maintained. This system will allow a more efficient searching in the case base by exploiting the semantic relations which exist between the cases. The system uses the semantic relations existing between the concepts within each of the ontology.

We apply our approach in equipment maintenance domain. Equipment maintenance deals with the detection and isolation of abnormal events. It consists of interpreting the current state of the machine from sensor readings and process knowledge. Equipment maintenance is of crucial importance in terms of safety and also of economics, because of the influence of abnormal events in yield and quality of products.

The remaining part of the paper is organized as follows. First, we outline the related work on knowledge systems using ontology and case-based reasoning in section II. Next, we detail our approach to develop the Knowledge-Intensive Decision Support System. In section IV, we present a case study related to maintenance in industrial plants and we describe an example session to deal with breakdowns. Finally, we give some clues on future work before concluding.

2. Related Work

Knowledge management encompasses various practices of managing knowledge such as knowledge generation, capture, sharing, and application. Within these practices, effective sharing and use of knowledge depends – to a large extent – on the organization's ability to create and manage its knowledge. This knowledge can be described as the way organizations store it from the past to support present activities (Gallupe, 2001) (Alavi & Leidner, 2001) (Maier & Hadrich, 2011) (Tan & Gallupe, 2008).

Knowledge management and Case-Based Reasoning are two intertwined topics. Case-based reasoning is a problem solving paradigm that in many respects is fundamentally different from other major AI approaches. Instead of relying solely on general knowledge of a problem domain, or making associations along generalized relationships between problem descriptors and conclusions, the case-based reasoning formalism was proposed as a way of storing human experiences and retrieving stored cases similar to the current item through a process of analogical search (Lamontagne & Plaza, 2014). It draws its knowledge from a reasonably large set of cases contained in the case library of past problems and by adapting their solutions solves new problems rather than only from a set of rules. Furthermore, case-based reasoning systems are claimed to "learn" through addition of further significant cases to the case-base and by forms of abstraction which may then be applied to this collection of cases (Richter & Weber, 2013).

Reasoning by re-using past cases is a powerful and frequently applied way to solve problems for humans. However, one of the drawbacks of CBR is the lack of flexibility of the knowledge representation. Indeed, the structure of the case is considered as constraining and strict which does not allow dealing with a carried out experiment in its semantic context, really limiting the performances of the system. As a way to deal with needs, ontology technology is an ideal selection for realizing knowledge-based decision support systems because ontology has not only powerful ability of knowledge representation, but also good semantic understanding (Ming et al., 2020) . Ontologies provide a semantic based approach to explicitly represent information in a computable manner so that information can be automatically processed and integrated. Ontology also provides shared understanding of a domain to overcome differences in terminology from various sources (Antoniou & Harmelen, 2004). The integration of an ontology-based model and CBR within a knowledge-based system has its advantages in: (1) Facilitating knowledge sharing by providing a formal specification of the semantics for context information; (2) Supporting for logic reasoning, referring to the capability of inferring new context information based on the defined classes and properties; (3) Enabling knowledge reuse by use of existing and mature ontology libraries without starting from scratch; (4) Having the stronger ability for expressing complex context information.

Several studies have given empirical evidence for the dominating role of ontologies integrated with specific, previously experienced situations (what we call cases) in human problem solving. Park and his colleagues (Park et al., 2007) propose an ontology-based fuzzy CBR support system for ship's collision avoidance to prevent the cumbersome tasks of creating a new solution each time a new situation is encountered. A case-based decision support system applied to loan evaluation is developed in (Benmessaoud & Adla, 2019). The approach uses AHP method to select important features and fuzzy sets technique to measure similarity between cases. A proposal presented in (Garrido et al., 2008) aimed at knowledge reuse, during the decision activities by means of interwoven concepts from the knowledge management research. In (Wang et al., 2009), the constructed decision support CBR prototype system of marketing strategy contains more than 600 cases. The evaluation shows that with the support of semantics, they can not only carry out data matching retrieval, but also perform semantic associated data access. Kobti & Chen (2010) construct domain ontology of mold design and propose an ontologybased search model to improve the traditional keyword-based search for the mold design domain. In (Rockwell et al. 2009), the authors proposed an approach based on the integration of three techniques: a CBR-personalized retrieval mechanism designed to provide a user with an optimum itinerary that meets his personal needs and preferences; a semantic web rule language considered to provide the system with enhanced semantic capabilities and support personalized case representation; and a user-oriented ontology used as source of knowledge to extract pertinent information about stakeholder's preferences and needs. A case-based reasoning (CBR) system for the Semantic Web is presented in (Gaillard et al., 2014). It implements a generic case-based inference mechanism in which adaptation consists in retrieving similar cases and in replacing some features of these cases in order to obtain one or more solutions for a given query. A knowledge-based approach to support decision making in human resource management is proposed in (Zhukova et al., 2014). The appropriate support of decision making is implemented using case-based reasoning and ontology. The problems of knowledge and case representation are considered, as well as the algorithm of case retrieval.

Many research efforts for decision modelling and support have systematically applied CBR technology to the field of maintenance (Bumblauskas et al., 2017) (Shana et al., 2019) (Maa et al., 2020). However, there is no complete method that would define how to model decisions in ontologies, and a few isolated cases in which an established decision making method was used in ontology for a specific domain, and often the reasoning procedure is based only on domain ontology.

In our approach, we consider particularly the case where the reasoning process is enriched by exploring ontology. Thereby the purpose is to retrieve and provide a set of possible solutions relating to source case showing the semantic relations between them. Afterwards, it is the duty of the decision-maker, according to his/her expertise, to opt for the decision which will seem to him appropriate to the target problem. An important goal of our work was to structure decision model in such a way that the problem solution can be obtained by reasoning upon three ontologies (domain, task, and decision). The ontologies with reasoning support can be used in the function of a case base reasoning system.

3. The Knowledge-Intensive Decision Support System

Decision making is mostly used as a multi-participant process with high level of interactivity. In (Adla et al., 2007) the paradigm of distributed decision-support systems is considered where several decision-makers must reach a common decision. An integrated framework based on a distributed architecture (DCI-DSS) is developed where each decision-maker uses a specific cooperative intelligent decision support system. The networked decisionmakers (DM) work together to solve a particular problem although they might neither be present at the same time in the same place nor constitute a permanent organization.

In (Adla et al., 2011), a toolkit for GDSS facilitators is proposed that was integrated in the proposed architecture for distributed GDSS (Adla et al., 2007). Based on a model of the decision making processes, group facilitation tasks are automated to increase the ability of facilitator to monitor and control the group decision meeting process. Having a model of the decision making process built into the system should enable intelligent decisional guidance. The resulting structure creates a model-based problem solving environment.

The aim of our work is to integrate a knowledge-based tool within this Group Decision Support System structure in order to combine a knowledge-based method (Case-based reasoning and Ontology) and the model-based reasoning, and thus enable hybrid reasoning (Figure 1). In

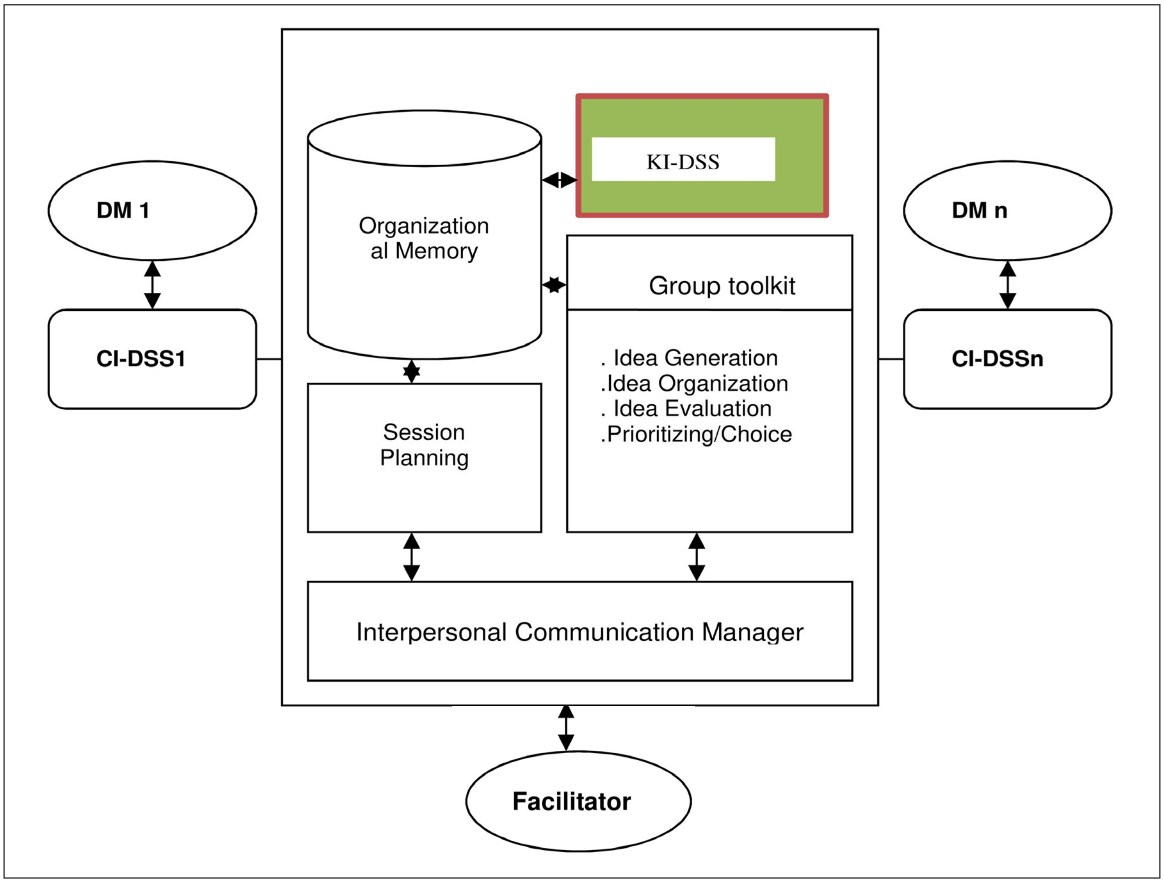

Figure 1. DCI-DSS Architecture

this way, when the knowledge-based method fails to provide a solution (in the case of a new situation or problem), the model-based method is executed to solve the problem and retain the solution as a new case for future use.

The proposed system called KI-DSS will assist the actors involved in maintenance session by offering them a set of decisions for the new problem. The actor situates each offered solution in its semantic context and then chooses a particular one based on his expertise. The integration of such a knowledge-intensive reasoning tool within the GDSS is likely to provide additional information processing support. KI-DSS enables indexing and retrieving cases to propose whole or part of them as solution(s) to a new problem. This increases knowledge reuse functionalities through cases and their components.

A problem-solving approach adopting CBR is used to solve a new problem (target case) by remembering a previous similar situation (source case) and by reusing information and knowledge of that situation. The effectiveness of this approach is further improved by the application of ontologies as a mechanism for reasoning about the domain concepts and dealing with the inconsistencies that can arise in the applied vocabulary when multiple decision makers are involved. Thus, our approach to knowledge-intensive system is towards integrated problems that combine case specific knowledge with models of general domain knowledge. The more knowledge is embedded into the system, the more effective is expected to be. Knowledge-intensive processes can take advantage of this domain knowledge and obtain more accurate results.

3.1. The Case Base

Knowledge representation is essential in building a knowledge-based system since on this presentation depends the effectiveness and the fastness of the system case retrieving mechanism. It is therefore necessary to well identify information to be stored in each case and to choose the more efficient representation scheme of this information. A case is a contextualized piece of knowledge representing an experience. The information encoded about the past experiences, depends on the domain of application as well as on the goal for which the cases are used. Case indexing and storage are an important aspect in designing efficient knowledge-based decision support systems in that, it should reflect the conceptual view of what is represented in the case and take into account the indices that characterize the case.

Knowledge considered in our knowledge-based decision support system is represented by cases and ontologies. The case base is composed of all the structured cases which will be explored during retrieving step (recall stage). Every case consists of a breakdown problem already experienced and solved. A case represents a diagnosis experience, and thus consists of two main parts: a problem part describing the failure, and a solution part. Each part

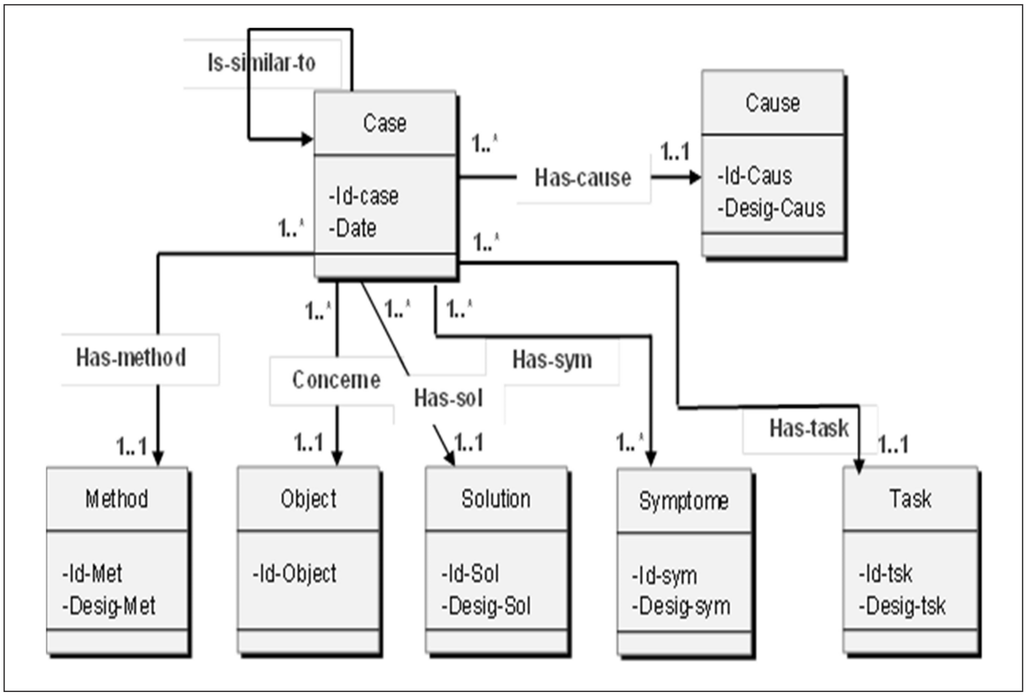

Figure 2. UML Class Diagram of the Case Base

is represented by a set of simple or complex descriptors among which some are defined in an ontology.

Problem part: the task to be solved;

Solution part: the solution, the problem solving method used and, the object concerned by recommended solution.

The case base is a finite set of source cases (S), denoted by $CB = \{S1, S2,... Sn\}$ where a source case $Si = (PbS, Sol[PbS]) / PbS$ is the source problem part and $Sol\ [PbS]$ is the solution part of the source case. $Sol\ [PbS] = \{[A1/V1], [A2/V2], ... [An/Vn]\}$ where $[Ai/Vi]$ means [Attribute/Value]. The case base is created manually with typical failures. The target case (C) is denoted by $C = (PbC, Sol\ [PbC])$. The target case problem part (PbC) is the structural representation of the new failure to be repaired; the solution of the target case ($Sol[PbC]$) is the structural representation of the resolution of this failure. The attributes of the PbC are filled upon an analysis phase of the problem parameters introduced by the user. This analysis involves the task ontology and allows structurally representing the new problem to be solved. Initially, the attributes of the $Sol(PbC)$ part remain empty. Figure 2 shows the UML classes diagram relating to the modelling of the case base. The descriptors are entries to the ontologies (e.g. Id-Task, Id-Symp and Id-Cause are entries for the task ontology; the descriptors Id-Object is an entry for the domain ontology, and the descriptor Id-Solution-Id is an entry for the application ontology).

3.2. Ontology Modelling

The ontology development methodology is usually composed of several strategies on defining classes and class hierarchy, defining properties and naming considerations (Sure et al., 2009) (Cormicana & Yub, 2019). The ontology is built based on documentation resources as all the potential decisions that might be made by the decision makers are listed in an appropriate documentation. Similarly, the description of the equipment to be maintained is get from specific documentation while the specification of the task ontology is built with the support of an expert in industrial maintenance.

The ontology is composed of two parts related to equipment domain and maintenance task. Figure 3(a) presents UML class diagram of the domain ontology and Figure 3(b) presents UML class diagram of the task ontology. The equipment domain part consists of a specification of the concepts relating to the equipment to maintain as well as the relations between these concepts. These relationships are principally aggregation and composition ones between the equipment components. The task ontology describes all the maintenance problems related to the equipment in terms of task, symptom, cause and solution concepts, and the relations between them.

The ontology is created using Protégé before their generation in OWL format (Hitzler et al., 2012). Figure 4 illustrates a partial view of the ontology. An operationalized ontology is expressed in an operational language and endowed with operational semantics. In this sense the ontology operationalization consists of a computer specification of all the operations made on concepts in an operational language. The use of an operational ontology assumes its representation in an operational but also formal language, i.e. providing reasoning mechanisms appropriate to the targeted knowledge manipulations. To do this, we used the NetBeans developing environment associated to Java language (Schildt, 2014). Furthermore, we used the Jena framework (Jena, 2019) to manage the ontology. Jena provides a programming environment for RDF, RDFS (Pan, 2009) and OWL as well as a query engine allowing SPARQL queries execution (Simple Protocol And RDF Query Language) (Prud'hommeaux et al., 2008) which is a RDF query language.

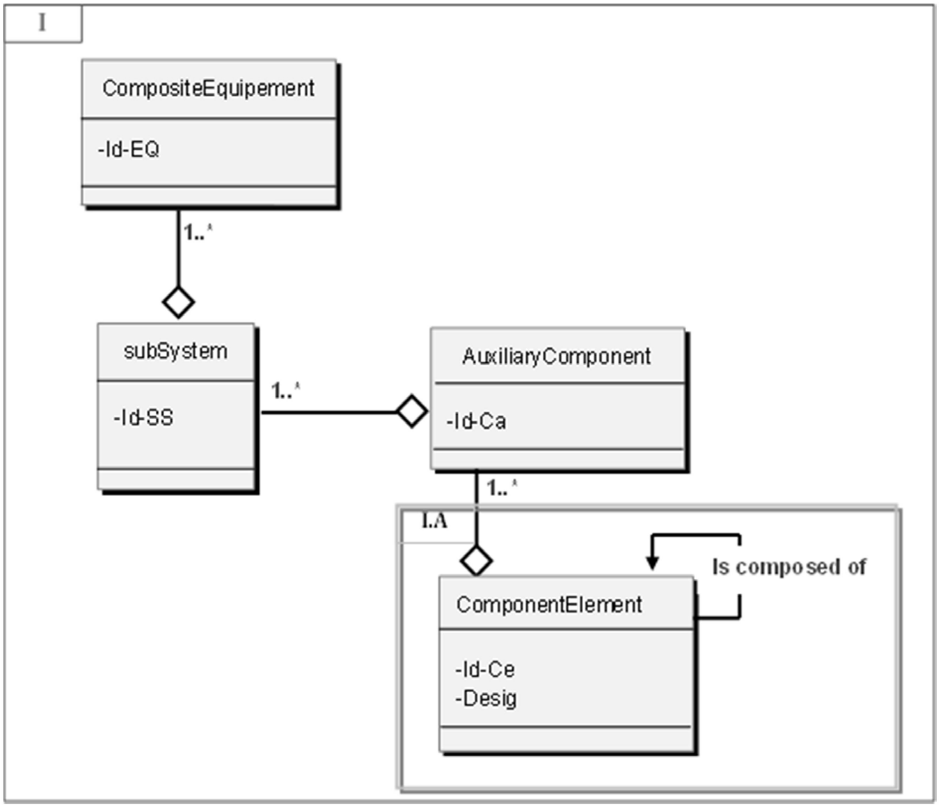

Figure 3(a). Conceptual model of the domain ontology

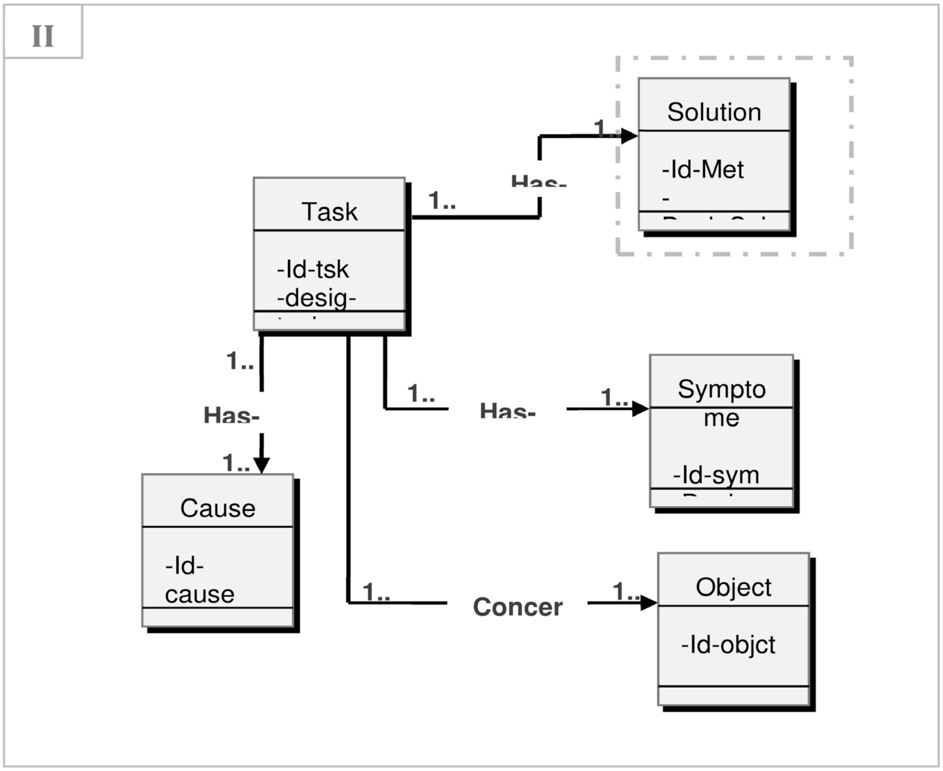

Figure 3(b). Conceptual model of the task ontology

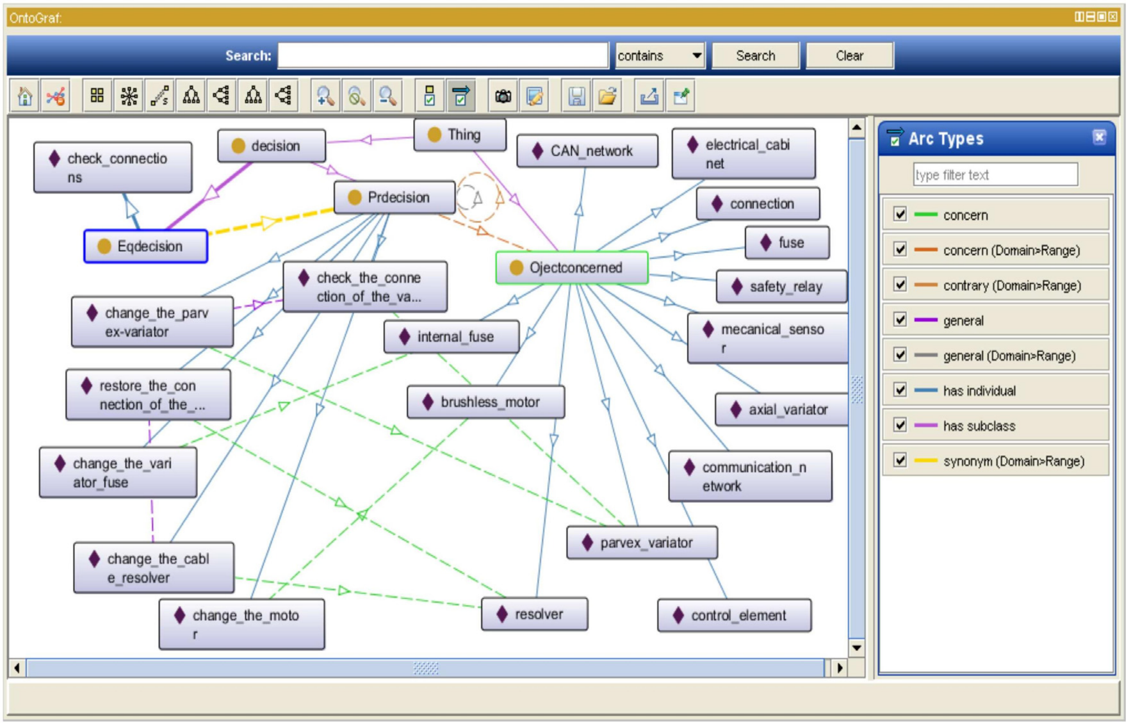

Figure 4. Partial view of the decision ontology

```
<owl:NamedIndividual rdf:about=”http://www.basedecas.org/ontologycases# 3">
<rdf:type rdf:resource=”http:// www.basedecas.org/ontologycases#Cases”/>
<rdf:type rdf:resource=”http:// www.basedecas.org/ontologycases#Cases”/>
<has-as-task rdf:resource=”http:// www.basedecas.org/ontologycases#T3"/>
<has-as-cause rdf:resource=”http:// www.basedecas.org/ontologycases# Failure to turn
up the variator”/>
<has-as-method rdf:resource=”http://www.basedecas.org/ontologycases#M1"/>
<concerns rdf:resource=”http://www.basedecas.org/ontologycases# Internal fuse”/>
<has-as-solution rdf:resource=”http://www.basedecas.org/ontologycases#change internal
fuse”/>
<possesses rdf:resource=”http://www.basedecas.org/ontologycases#display variator off”/
>
<possesses rdf:resource=”http://www.basedecas.org/ontologycases# Failure to turn up
the variator, the machine
is shut down”/>
<is-similar-tordf:resource=”http://www.basedecas.org/ontologycases#15"/>
</owl:NamedIndividual>
```

Table 1. Example of the source case Task T3

OWL language (Hitzler et al., 2012) is used to represent the case base (Table 1). This would allow managing the case base as a knowledge base upon which inferences may be made. It is possible to define semantic relations between cases as for instance the transitive relation “is-similar-to” which relates the source cases already identified as being similar. Furthermore, as the remained knowledge (i.e. the ontologies) is also expressed in OWL, this would allow having to some extent compatibility between languages formalizing the different knowledge manipulated by the system, as well as, the knowledge operating tools such as SPARQL.

3.3. The Reasoning Process

The proposed case based system should reflect human knowledge by storing data about previous significant events as “cases” within a computerized system. In this regard, the system uses the case base to retrieve similar cases to the problem to be solved. But, when the retrieving process fails or the cases retrieved are not satisfactory for the decision maker, the system uses ontologies. It makes use semantic relations between concepts within the same ontology to derive other solutions to the problem.

By making use of the ontology, the system derives more specific or more general decisions than those initially retrieved by the system. It can also set the solution relating to the equipment by visualizing the concerned component. Then, it uses the ontology to set the involved component relating to the neighboring ones or to the component in which it’s comprised. Similarly other case descriptors may be used as entries to the ontology to enlarge or reduce solution space. When a solution is retained, then tested and validated, it is stored in the case base as a new case (with all its descriptors). The reasoning process consists of the following steps (Figure 5):

Problem description: The decision maker describes the problem to be solved. This description can be made of different ways: by providing the task to be solved, the observed symptoms, or the faulty object, etc.

Retrieving: It consists to search in the case base and retrieve similar cases to the problem to be solved. Here, we consider the usual local and global similarities measures to retrieve similar cases to the targeted problem. Once the target problem is introduced, the goal of this step is to recall the source case that is most similar to the target case, applying two measures of similarity (local, global) between the target problem and the source problem.

• **Local Similarity:** This similarity measure is computed between the value of an attribute in target problem (PbC) and the value of the same attribute in source problem (PbS). It is evaluated in two different ways depending on whether the attribute is simple or complex.

• **Simple Attribute:** The simple attribute in our case is the symptom attribute which has a unique value. The local similarity between the attributes is equal to 1 if the values of the two attributes are equal (see equation (1)), else it is equal to 0.

$$Sim_i(s_i, c_i) = \begin{cases} 1 & pour \quad ci = si \\ 0 & pour \quad ci \neq si \end{cases} \qquad (1)$$

C_i: Value of the attribute i in PbC part.

S_i: Value of the same attribute i PbS part.

• **Complex Attributes:** The complex attribute in our case is the symptom attribute which has a list of values. The

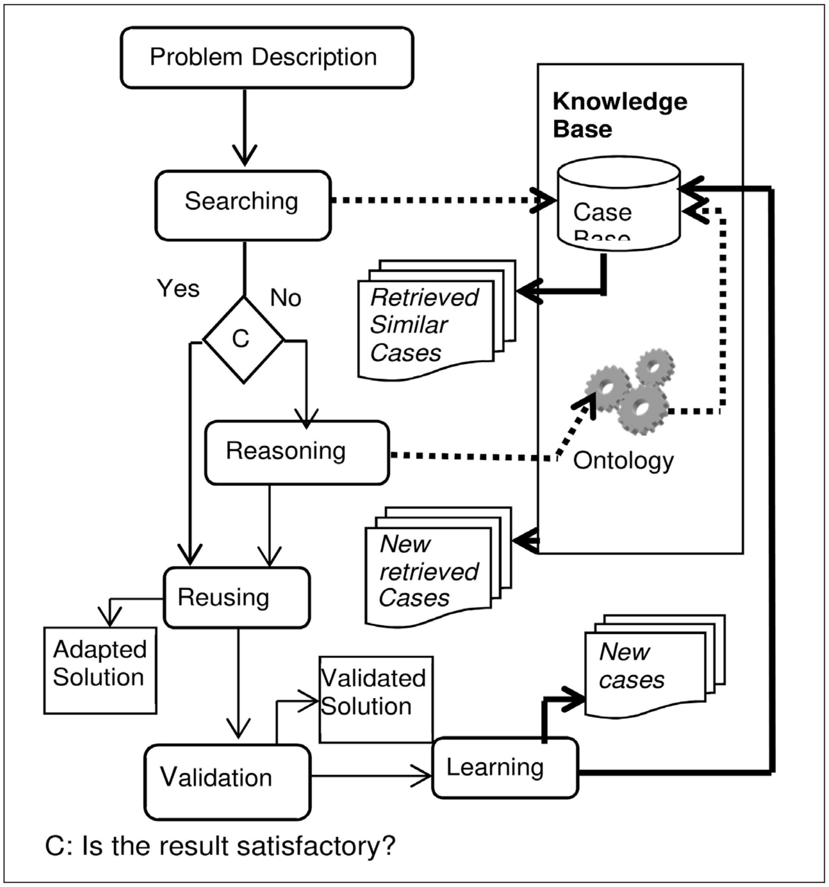

Figure 5. Ontology-based CBR Process

local similarity between the values of the symptom attribute of the PbC and the values of the same symptom attribute of the PbS is given by equation 2 below:

$$Sim\,(si,\, ci) = \frac{ni * 2}{n1 + n2} \qquad (2)$$

ni: Number of equal values in both target and source problem for the i attribute.

$n1$: Number of values relating to the attribute in the source problem.

$n2$: Number of values relating to the attribute in the target problem.

• **Global Similarity:** This similarity measure is calculated between a set of attributes in the PbC and the same set of attributes in the PbS, it corresponds to the mean of the local similarities and its value is in the range [0,1]. It is defined by the similarity function below:

$$SIM\,(S,\, C) = \frac{1}{n} + \Sigma_{i=1}^{n}\, simi\,(si,\, ci) \qquad (3)$$

C: target case.

S: source case.

n: number of attributes in which the local similarity has been calculated.

$Sim_i\,(s_i,\, c_i)$: value of local similarity for attribute i.

To search a PbS, the most similar to the PbC. For this, we apply the measure of local then global similarities. In the global similarity measure, a similarity threshold is determined at 0.5. If $SIM\ (S,\ C) <0.5$ the source case is considered negligible (not similar) else if $SIM\,(S,\,C) \geq 0.5$ the source case is considered important (partially similar). In case $SIM\,(S,\,C) = 1$ the source case is considered perfectly similar.

Reasoning: If a perfectly similar case is not found, we move on ontology-based reasoning in order to search other source cases. For each of the initially retrieved source cases, we use semantic relations between the source cases to derive other source cases semantically close to similar cases retrieved. These if they exist are presented to the user. The latter, if the suggested solutions are satisfactory, the process of recalling is stopped and we pass to the next phase of the reasoning process. Furthermore The ontology may be used to enlarge or to reduce the solution search space. According to the object of widening, the ontology is used. For example when the object of widening is a task or a symptom, the task ontology part is used; when the object of widening is a faulty component, the equipment domain ontology part is explored, but when the object of enlargement is the problem solution then the decision ontology part is used.

Validation: Once the decision is made, executed and validated the process will skip to the next step.

Learning: The new case is added to the case base. It referees to all the similar source cases if exist.

The reasoning step is useful as it allows revealing semantic knowledge from ontologies between the different parameters of the problem to be solved. Given a problem to be solved, this would allow: 1) Converging to the semantically close case in the case base, or 2) Retrieving first a structurally close case from the case base then, according to the case descriptors, exploiting ontologies in order to derive other possible solutions to the problem. The decision-maker will choose among the suggested solutions that he considers being the most appropriate one to the problem.

4. Case Study

In industrial plants, the presence of abnormal events generates risks. To avoid such risks, safety barriers are set up. However, barriers may not work properly, and thus abnormal events may arise. In such a case, industrial technicians intervene to diagnose the failures basing mostly on past failure experiences occurred in similar situations.

The application concerns EMETAL; an Algerian company which offers a wide range of sheet bending machines built for years of operation at full capacity. The machines are manufactured for the automotive industry, outdoor and indoor furniture, supermarkets and household electrical appliances, etc. The machines are numerically controlled tools that bend and fold sheets and tubes, precision and industrial sheet metal, steel sheet and strip. The controllers perform a variety of functions including protecting the machine from damage by performing an automated shutdown when dangerous conditions are detected and archiving sensor data.

As of 2016 there are thousands of these machines in use by EMETAL's customers national wide. EMETAL has contracts to service more than a thousand of machines and that number has been growing by hundreds every year. Different types of machines are manufactured in this company. Each machine is identified with two parameters: the number of axes and the diameter of the tube to be bent. The FS Series machine is a folding machine with three axes: feeding, folding and folding. This machine can bend wires with diameters up to 7 mm. The FS Series machine is a combination of 13 different components. The Parvex dimmer in the switch cabinet is a particularly critical component. It consists of a set of drives.

A statistical study was carried out on the failures encountered in the company. Some of the failures are common between the different machines. In most cases, the failure is due to the electrical problem. The Parvex dimmer is the component the most affected by the failures. The duration of repairing can last up to a few weeks for complex problems.

A failure may be defined as an abnormal change in the characteristics of a system which gives rise to undesirable performance. Equivalently, the performance change could be due to deterioration or malfunction. Hence, the technicians should know the cause(s) of the change. In such cases the diagnosis tool should be able to identify the cause(s).

The goals are to improve machine and system reliability, reduced machine operating/maintenance costs, and produce the greatest possible sustained availability from the machine. For the company, it is a matter of better formalizing the experiences of the experts on maintenance, using reasoning from experience. The knowledge intensive approach is to be used to automate the data review, hypothesis generation, and hypothesis confirmation of this process whenever possible and assist the user when it does not have confidence in a single cause.

4.1 Example Session

We experiment our system on a case of bending machine breakdown. The handling of a machine breakdown consists of three main steps: discerning defects while the machine is functioning, diagnosing defects, and proposing one or several appropriate actions of repair.

Currently, the machine maintenance process was as follows. When a machine has broken down or a malfunction of a machine is detected, the on-site operator will call the company and will send a message. The breakdown will be assigned to a company technician for analysis. The technicians take into account information provided by the customer, will access the data from the machine, review key values, draw on their previous know-how and experience, create a hypothesis about the breakdown cause, create plots specific to the breakdown type hypothesized, confirm the cause of the breakdown as best as can be done using the available data, then call the site to provide assistance and confirm the breakdown cause. Each repairing is recorded. The saved records are not well formalized and thus unexploited. These sheets should be used as a basis of experience for use in future repairs.

We are in the context where typically incidents are not entirely identical to each other (some symptoms are not observed) but the knowledge of past incidents enables decision makers to recognize a similar situation and tailor their strategies by taking a course of action that experience has shown is effective and successful. This can happen when there is failure at some sensors so that lights or alarms cannot be triggered. The search in the case base can then be disoriented.

In order to contribute, we use a case base of reference structured cases. Each case consists of a breakdown problem already experienced and solved. It represents a diagnosis experience, and thus consists of two main parts: a problem part describing the failure, and a solution part (Table 2). Major symptoms are given that may be automatically signposted to the operator by means of triggered-off alarms. and pointed out on the board (control room (Table 3).

If a “variator breakdown” occurs, an alarm is triggered-off and the following symptoms are automatically signposted to the operator: “Symp1 (Symp1: impossible to power up the variator), Symp18 (No information on the drive displayed). The corresponding case has the following structure (Table 4):

We do search for and retrieve similar cases to the problem occurred. First, the task ontology is used to infer the features lacking (Cause) in the target problem structure (Table 5).

Secondly, we calculate the local similarities measures related to the attributes: task, cause, symptom of the new problem and those of source cases. Then, the global similarity is computed (Table 6).

Id- T	Task	Cause	Method	Object	Solution
1	T1	Resolver break failure	M3	Resolver cable	Check the resolver cable
2	T2	Breaking of one or more wires of the encoder	M2	Encoder	Changing the encoder
3	T3	Failure to power up the variator	M1	Internal fuse	Change the internal fuse
4	T4	Internal drive failure	M4	Drive	Changing the drive
5	T5	High temperature	M6	Radiator	Clean the radiator
6	T6	Excessive engine speed	M5	Engine	Changing engine
7	T7	Drive power incident	M1	Internal fuse	Changing the internal fuse
…	…	…	…	...	...

Table 2. The case base

Id-T	Symptoms
1	Symp 1 (Failure to power up the machine, the variator is shut down)
1	Symp 3 (Fault 2 appears on the axis drive display)
2	Symp 2 (No information at the encoder display)
3	Symp 1(Failure to power up the machine, the variator is shut down)
3	Symp 4 (At the time of automatic start the orientation head moves downwards)
3	Symp 10 (Fault 4 appears on the display of the axis drive)
3	Symp 18 (No information displayed)
3	Symp 20 (Electrical axis incident, drive power incident)
4	Symp1 (Failure to turn up the machine, the variator is shut down)
4	Symp 20 (Electrical axis incident, drive power incident)
4	Symp 21 (Absence of "READY" (Power E1)) (Symp displayed on the computer screen)
5	Symp 7 (Fault 4 appears on the display of the axis drive)
5	Symp 20 (Electrical axis incident, drive power incident)
5	Symp 21 (absence of "READY" (Power E1)) (Symp displayed on the computer screen)
6	Symp 9 (High engine temperature)
6	Symp 18 (No information displayed)
6	Symp 1 (Failure to power up the machine, the variator is shut down)
…	…

Table 3. List of symptoms

New Problem	**Task**	Variator failure
	Cause	
	symptoms	Impossible to power up the variator
		No information on the drive displayed
New Problem Solution	**Method**	
	Objet	
	Solution	

Table 4. The case base

New Problem	**Task**	Variator failure
	Cause	Drive power failure
	symptoms	Impossible to power on the machine
		No information on the drive displayed
New Problem Solution	**Method**	
	Object	
	Solution	

Table 5. The case base

Case	Attributes			Global Similarity
	Task	Cause	Symptom	
T1	0	0	0.5	0.16
T2				
T3	1	1	0.5	0.83
T4	0	1	0.4	0.46

Table 6. The case base

Similar source case	**Id-T**	3
	Task	T3
	Cause	Drive power problem
	symptoms	Symp1
		Symp2
Similar source case solution	**Method**	M1
	Object	Internal Fuse
	Solution	Changing the fuse

Table 7. The case base

Regarding the results got from this step, a source case (source problem) T3 is given similar to the new task with a higher similarity (0.83) (Table 7).

If the user judges that the result is satisfactory and d would accept the proposal, he will apply the solution to the target case (target problem).

4.2 Discussion

We adopted two grades of retrieval strategy in case retrieval step. The first strategy is ontological semantic retrieval, which can enlighten and inspire the user by retrieving relevant cases as sources of inspiration, and thus infer a new structure from the initial incomplete one. Semantic retrieval is a qualitative search which provides a set of cases. Whilst the second grade retrieval strategy which is quantitative retrieval is conducted based on similarity measurement for getting the most similar case for reuse. The second grade retrieval strategy mainly measures similarities between structure features. CBR can provide high-quality decision-making, but it will fail when lack of applicative cases. Based on the logical reasoning ability of ontology, the deficiency of CBR is likely made up, and the performance of the proposed KI-DSS system enhanced, compared to existing CBR systems.

5. Conclusion

In this paper, we focused on the design and development of a Knowledge-Intensive Decision Support System (KIDSS) to support better decision and improve equipment maintenance efficiency in industrial plants. We integrated case-based reasoning and ontology within KI-DSS. The effectiveness of CBR can be further improved by the application of ontologies as a mechanism for reasoning about the domain concepts and dealing with the inconsistencies that can arise in the applied vocabulary when multiple decision makers are involved. Moreover, the reuse of ontologies from a library also benefits from their reliability and consistency.

We claim that this combination is useful for the design of KI-DSS and strengthens its reasoning process as it allows the knowledge engineer to use knowledge already acquired, conceptualized and implemented in a formal language; reducing considerably the knowledge acquisition bottleneck. We believe that our approach is useful in several aspects. First, it enables to formalize the case base in OWL what allows managing it as a knowledge base. Indeed, by exploiting the semantic relations within the case base, it is possible to derive new knowledge from those stored. Also, as a result of memorizing a source case base with its descriptors, the ontologies exploration will allow deriving new knowledge which will serve for a new research cycle in case base.

As future work, we aim to collect much information to use quantitative clues for performance evaluation of our approach. Further experiments are to be organized to evaluate the performance of our system with fault coverage rate, diagnosis effectiveness ratio, and other quantitative information.

References

[1] Adam, F. (2012). 20 years of decision making and decision support research published by the Journal of Decision Systems. *Journal of Decision Systems*, 21 (2) 93-99.

[2] Ackerman, M. S., Halverson, C. A. (2004). Organizational Memory as Objects, Processes, and Trajectories: An Examination of Organizational Memory in Use. *Computer Supported Cooperative Work (CSCW)*, 13, 155–189.

[3] Aamodt, A., Plaza, E. (1994). Case-based reasoning: foundation issues, methodological variations and system approaches. *Artificial Intelligence Communications,* 7, 39-59.

[4] Guo, Y., Hu, J., Peng, Y. (2012). A CBR system for injection mould design based on ontology: A case study. *Computer-Aided Design*, 44, 496–508.

[5] Gallupe, B. (2001). Knowledge management systems: surveying the landscape. *International Journal of Management Reviews*, 3 (1) 61-77.

[6] Alavi, M., Leidner, D. E. (2001). Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. *Management Information Systems (MIS),* 1 (10) 107-136.

[7] Maier, R., Hadrich, T. (2011). Knowledge Management Systems. *Encyclopedia of Knowledge Management*, Second Edition, IGI Global, 779-790.

[8] Tan, F. B., Gallupe, R. B. (2008). Global information management research: current status and future directions. In Tan, FB (ed.): Global Information Technologies: Concepts, Methodologies, Tools, and Applications, 3571-3584, IGI Global.

[9] Lamontagne, L., Plaza, E. (2014). Case-Based Reasoning Research and Development. LNCS 8765, Springer International Publishing Switzerland.

[10] Richter, M. R., Weber, R. O. (2013). Case-Based Reasoning: A Textbook. Springer-Verlag Berlin Heidelberg.

[11] Ming, Z., Sharma, G., Allen, J. K., Mistree, F. (2020). An Ontology for Representing Knowledge of Decision Interactions in Decision-Based Design. *Computers in Industry*, 114, 103145.

[12] Antoniou, G., Harmelen, F. (2004). A Semantic Web. Primer, MIT.

[13] Park, G., Benedictos, R. M., Lee, C., Wang, M. H. (2007). Ontology-Based Fuzzy-CBR Support System for Ship's Collision Avoidance. *In: Proc. of International Conference on Machine Learning and Cybernetics (ICMLC.2007)*, pages 1845-1850, Hong Kong, (August).

[14] Benmessaoud, N., Adla, A. (2019). Intelligent Semantic Case Based Reasoning System for Fault Diagnosis. *Journal of Digital Information Management (JDIM)*, 17 (2) 75-86.

[15] Garrido, J. L., Hurtado, M. V., Noguera, M., Zurita, J. M. (2008). Using a CBR approach based on ontologies for recommendation and reuse of knowledge sharing in decision making. *In: Proc. of 8th International Conference on Hybrid Intelligent Systems (HIS 2008)*, pages 837–42, Barcelona, (September).

[16] Wang, D., Xiang, Y., Zou, G., Zhang, B. (2009). *Research on Ontology-Based Case Indexing in CBR. In: Proc. of International Conference on Artificial Intelligence and Computational Intelligence (AICI.2009)*, pages 238-241, Shanghai, (January).

[17] Kobti, Z., Chen, D (2010). A domain ontology model for mold design automation, *Canadian AI,* 6085, 336–339.

[18] J. Rockwell, J., Grosse, I. R., Krishnamurty, S., Wileden, J. C. (2009). A Decision Support Ontology for collaborative decision making in engineering design. In Proc. of *2009 International Symposium on Collaborative Technologies and Systems (CTS 2009)*, pages 1-9, Baltimore, (May).

[19] Gaillard, E., Infante-Blanco, L., Lieber, J., Nauer, E. (2014) Tuuurbine: A Generic CBR Engine over RDFS. *In*: Lamontagne L., Plaza E. (eds) Case-Based Reasoning Research and Development. (ICCBR 2014), Lecture Notes in Computer Science, 8765, Springer, Cham.

[20] Zhukova, I., Kultsova, M., Navrotsky, M., Dvoryankin, A. (2014). Intelligent Support of Decision Making in Human Resource Management Using Case-Based Reasoning and Ontology. *In*: Kravets A., Shcherbakov M., Kultsova M., Iijima T. (eds) Knowledge-Based Software Engineering, Communications in Computer and Information Science, 466. Springer, Cham.

[21] Bumblauskas, D., Gemmill, D., Igou, A., Anzengruber, J. (2017). Smart Maintenance Decision Support Systems (SMDSS) based on corporate big data analytics. *Expert Systems with Applications*, 90, 303–317.

[22] Shana, W., Dongbob, L., Gaoc, J., Jinga, L (2019). A knowledge based machine tool maintenance planning system using case-based reasoning techniques. *Robotics and Computer Integrated Manufacturing*, 58, 80–96.

[23] Maa, Z., Rena, Y., Xianga, X., Turk, Z. (2020). Data-driven decision-making for equipment maintenance. *Automation in Construction*, 112, 1-17.

[24] Adla, A., Soubie, JL., Zaraté, P (2007). A cooperative Intelligent Decision Support System for Boilers Combustion Management based on a Distributed Architecture. *Journal of Decision Systems (JDS)*, 16 (2) 241-263.

[25] Adla, A., Zarate, P., Soubie, J. L. (2011). A Proposal of ToolKit for GDSS Facilitators. *Group Decision and Negotiation (GDN)*, 20, 57-77.

[26] Sure, Y., Staab, S., Studer, R. (2009). Ontology Engineering Methodology. *In*: Staab S., Studer R. (eds) Handbook on Ontologies. International Handbooks on Information Systems, Springer, Berlin, Heidelberg.

[27] Cormicana, K., Yub, M (2019). Ontology-based

systems engineering: A state-of-the-art review. Computers in Industry 111, 148–171.

[28] Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P. F., Rudolph, S (2012). OWL 2 web ontology language: document overview, W3C Recommendation, http://www.w3.org/TR/owl2-overview/.

[29] Schildt, H. (2014). Java: The Complete Reference, McGraw-Hill Education Group.

[30] Jena, (2019). A free and open source Java framework for building Semantic Web and Linked Data applications. HP Labs, ena.apache.org.

[31] Pan, J. Z. (2009). Resource Description Framework. In: Staab S., Studer R. (eds) Handbook on Ontologies. International Handbooks on Information Systems, Springer, Berlin, Heidelberg.

[32] Prud'hommeaux, E., Seaborne, A. (2008). SPARQL Query Language for RDF. W3C, http://www.w3.org/TR/rdfsparql-query/