



Flexible knowledge representation and new similarity measure: Application on case based reasoning for waste treatment

Philippe Chazara, Stéphane Négny, Ludovic Montastruc

► To cite this version:

Philippe Chazara, Stéphane Négny, Ludovic Montastruc. Flexible knowledge representation and new similarity measure: Application on case based reasoning for waste treatment. Expert Systems with Applications, 2016, 58, pp.143-154. 10.1016/j.eswa.2016.03.014 . hal-01900381

HAL Id: hal-01900381

<https://hal.science/hal-01900381>

Submitted on 22 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Open Archive Toulouse Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of some Toulouse researchers and makes it freely available over the web where possible.

This is an author's version published in: <http://oatao.univ-toulouse.fr/20505>

Official URL: <https://doi.org/10.1016/j.eswa.2016.03.014>

To cite this version:

Chazara, Philippe and Negny, Stéphane and Montastruc, Ludovic Flexible knowledge representation and new similarity measure: Application on case based reasoning for waste treatment. (2016) Expert Systems with Applications, 58. 143-154. ISSN 0957-4174

Any correspondance concerning this service should be sent to the repository administrator:
tech-oatao@listes-diff.inp-toulouse.fr

Flexible knowledge representation and new similarity measure: Application on case based reasoning for waste treatment

Philippe Chazara*, Stéphane Negny, Ludovic Montastruc

Universite de Toulouse, Laboratoire de Genie Chimique UMR CNRS/INPT/UPS 5503, BP 34038, 4 allée Emile Monso, Toulouse 31030 Cedex 4, France

A B S T R A C T

In Case Based Reasoning the representation of a case and the similarity measures are two difficult steps in the conception of a system. Often, these steps are developed to resolve one kind of problem. However, in some of them such as recovery treatment processes generation, it is necessary for the system to be able to modify and adapt the representation of a case and the similarity measures with respect of the context and also the kind of solutions proposed. In this paper, authors introduce a new method to represent cases with a flexibility based on a structure in a connectionist model. This flexibility is needed due to the complexity of cases, the number of possible options and to ensure the durability of the system. In a second main contribution, authors introduce a method for the selection of source cases using abstraction, conceptualisation and inference mechanisms. Finally, authors test their system in a CBR developed on SWI-Prolog with different problems. The CBR is applied to find new recovery processes and try to estimate the new upgraded product generated.

Keywords:

Case-based reasoning
Similarity measures
Ontology
Dynamic cases
Flexible case representation
Waste management

1. Introduction

The problem of waste and in particular the problem of waste management has increased sharply during the last decades, producing three kinds of effects. First, the problem of waste treatment is becoming more and more important due to the quantity produced with the increase of human population size and consumption. Second, the prices of some raw materials are growing sharply due to the phenomenon of depletion. It becomes more and more difficult to find new sources and their exploitation costs enhance. Third, the treatment of waste can have a strategic dimension. Actually, it can reduce the raw material dependency for some countries, it can develop new industries and create new jobs. But currently, waste is considered as a pollution source for environment and as a costly burden for companies because of the loss of material and the waste treatment. Consequently, it is necessary to propose new recovery processes and new ways to manage waste. However, some elements induce limitations. First, contrary to a new product, a waste has not essence by definition. Therefore, the first question is to find one or more essences for it. The

second question is how to transform a waste into new valuable products. To solve these questions, authors propose to use an artificial intelligence system, and more particularly case based reasoning (CBR). CBR is relevant for this kind of problems because it allows solving problems without a clearly defined knowledge of the process needed for the resolution. The reasoning can rely on a vast number of cases, with their precise description of previous solved problems and their associated solutions (Cordier, Mascaret, Mille, 2009). Secondly, in the domain of waste treatment, cases may contain different information: valorisation processes and essences for the new created objects. In the literature, case based reasoning systems are used in different waste treatment problems and in processes research. For example, López-Arévalo, Bañares Alcántara, Aldea, Rodríguez-Martínez, and Jiménez (2007) describe a tool based on CBR for the generation of process alternatives. Yang and Chen (2011) propose a classical CBR retrieve method used for Eco-innovation Kuo (2010) gives an example of CBR used to determine a recyclable index of some components. Liu and Yu (2009) use CBR for problems linked to environmental topic. Zeid, M. Gupta, and Bardasz (1997) propose a model dedicated to disassembling problems.

As detailed in Section 2, CBR method is decomposed in different steps: *Retrieval, Adaptation, Memorisation or Learning* as explained by Aamodt and Plaza (1994) and Napoli, Lieber, and Curien (1996), similarity measure is one key cornerstone of a CBR system and of

* Corresponding author. Tel.: +330534323663.

E-mail addresses: philippe.chazara@ensiacet.fr (P. Chazara), stephane.negny@ensiacet.fr (S. Negny), ludovic.montastruc@ensiacet.fr (L. Montastruc).

the *Retrieval* part in particular. This measure allows finding close and relevant cases to solve the new problem. Therefore, with our goal to reuse the knowledge related to recovery methods for new waste valorisation, it is important to propose new approaches for this step respecting the constraints imposed by this category of problems.

On this topic, authors tackle several problems related to the similarity question. The first one is how to represent a case and more particularly for the domain of application, how to represent a waste. There are many kinds of waste and they need different representations. Moreover, the domains of waste and waste treatment have an important dynamic. Indeed, these domains change quickly i.e. the composition of waste, or the waste treatment processes evolve over time. To take into account these points, it is necessary to develop a flexible case representation to ensure a precise description of problem, knowledge reuse, CBR system efficiency and durability. Another consequence of these points is that the problems of waste cannot be considered as routine problems. However, CBR systems are developed to resolve only routine problems i.e. problems which are very similar. Consequently, a system used for these kinds of problems need to go beyond this limitation by the introduction of flexibility. Another point is how to take into account that there are different possibilities of valorisation for a same waste. For example, in the case of used tyres, they can be burnt to produce energy, reused as tyres, transformed by crunching into material for different kinds of new object, transformed by fermentation to produce syngaz. For each solution, the same description parameters are not selected: for some solutions is the chemical composition; for other ones is the form or the functionality, for other ones mechanical properties. Therefore, as showed by [Lieber \(2002\)](#), problems and their solutions depend on their use. As a consequence, authors think that problem representation and similarity measure depend on the solution or the kind of solution targeted.

In this paper, authors propose to explain their methods for representing knowledge and cases, and for selecting relevant cases. These methods try to take into account the solution and therefore to adapt the similarity measure in function to the important parameters according to a kind of solution. Moreover, these methods do not produce a metric value of distance or similarity measure but, it determines if a case is similar to the current problem or not, i.e. if the case can be used to generate an original solution for the problem. Contrary to [Perner \(2003\)](#), the method is not based on graphs, and it does not use threshold or other metric value, but it is based on logical deductions. In conclusion, the major contributions of this paper are the following:

- The introduction of a flexible representation for knowledge.
- A dynamic construction of cases, which allows going beyond the limitation of routine problems.
- A new method for similarity measure, without calculation and with a limited need of knowledge.

In the remainder of this paper, the [Section 2](#) explains some elements about CBR systems and develops some ideas for the realisation of each step finding in the literature. In [Section 3](#), the proposed flexible representation of a case is described and more specifically the management of the knowledge is explained. Then, the core of this method is introduced with the presentation of the main assumptions, and the retrieve part is described step by step in [Section 4](#). The [Section 5](#) highlights the method capabilities through a case study, where some tests have been realised to assess the proposed method. [Section 6](#) issues opinions about the positive points and the limitations of the method, and underlines some difficulties met during its implementation. Finally, [Section 7](#) draws conclusions and sum-

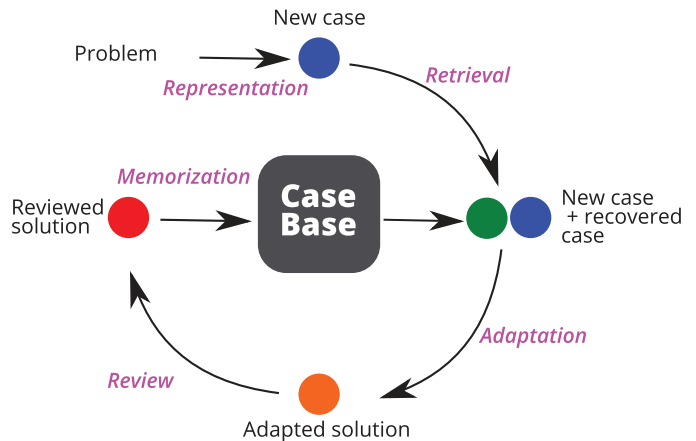


Fig. 1. Steps in classical CBR.

marises the presented work, and proposes different perspectives to improve it.

2. Case-based reasoning : different related steps

As explained in the introduction, a CBR system is based on different steps (each of them decomposes in sub processes not detail here) ([Reyes, Negny, Robles, & Le Lann, 2015](#)) ([Fig. 1](#)).

However, the realisation of one step impacts all the CBR's processes. The representation of the knowledge or cases impacts the sub-processes in the retrieval step, for example the similarity measure or the mapping phase. Therefore, it is necessary to represent knowledge by taking into account that retrieval step uses it, i.e. the definition of all the sub-processes depends on the choice of a kind of representation. Finally, the last sub-process of this retrieval step of the CBR is the selection of the relevant case in order to revise its solution to match to the target case requirements. One mechanism used is the analogy. [Cornuéjols \(1996\)](#) has studied the fundamental of this mechanism. He defined analogical reasoning as the way to find the expression which allows passing from a previous problem to its solution and to apply it to a new target case. Here too, the representation of cases is important.

In traditional CBR, the knowledge is often represented as a set of spaces. [Napoli et al. \(1996\)](#) explain that there is a space for the problem and another one for their solution. [Mougouie and Bergmann \(2002\)](#) define a query in CBR system as a point in these spaces. Therefore, each point of these spaces has to be represented with a common method. [Kokinov \(1994\)](#) explains that a cognitive mechanism is based on representation, memorisation. In CBR and in general for all artificial intelligent systems, representation is only a partial description of the reality. As a consequence, [Mougouie and Bergmann \(2002\)](#) explain that a query is only partially described. For Peschl, it is an interpretation of the world which allows the construction of a behaviour ([Peschl & Riegler, 1999](#)). Under this idea, [Amaief and Lu \(2013\)](#) link an ontology to a CBR system to facilitate the understanding of a situation and the retrieve step. This interpretation is very important in the resolution phase as Richard highlights because a modification of the interpretation can improve the efficiency of solving methods ([Richard, 1979](#)). Finally, representation can be symbolic, based on connexions ([Kokinov, 1994](#)), defined as vector features, or complex as semantic network ([Branting & Aha, 1995](#)). Whatever, the manner to represent knowledge, it is a reduction of the reality. But, the choice of the representation approach impacts the similarity measure step. For example, [Branting and Aha \(1995\)](#) and [Garey and Johnson \(2002\)](#) explain that the utilisation of semantic network for the representation of cases in CBR causes that the mapping step is

a NP-complex task. Napoli et al. (1996) work with an object based representation allowing a classification of cases.

The *Similarity* step in CBR tries to find the most similar case to a new problem. Similarity is a fundamental part of the CBR (Rifqi, 2010), and it measures if two things shared some common elements (Nesme & Hidalgo, 2013). In the literature, it is possible to find that similar cases research step starts with a description of the case and sometimes by a mapping step. This process is defined as the identification of the relationships between the elements describing two cases, as suggested by Markman and Gentner (1993) which are NP-hard or NP-complete problems (Sorlin & Solnon, 2005). For example, Falkenhainer, Forbus, and Gentner (1989) describe the structure-mapping engineering and explain the mapping result as the correspondence between the source case and the target case which can be improved by a set of analogical inferences. In addition, McFee and Lanckriet (2011) highlight the question of similarity between different kinds of item. The authors propose a method to integrate heterogeneous data into a single unified similarity space and to consider some similarity comparison as a direct graph. Usually, similarity measure evaluates the distance between the target case and a source one (Richter, 1993). More generally, it can be defined as the task to find the closed point to a target one (Mougouie & Bergmann, 2002). In the literature for similarity calculation between two cases, it is possible to find plenty of methods. For example, Bisson (2000) proposes to estimate the similarity by the effort required to transform one case into the other one. Other methods compare each elements one by one once the mapping is realised. Avramenko and Kraslawski (2006) give three kinds of similarity measures for CBR in process engineering. These kinds are *Quantitative distance*, *Hierarchical tree* and *Qualitative comparison* which allow giving a distance measure or a similarity measure as a number. *Similarity* in CBR can be applied to concepts studying the position of one concept to another one in a taxonomy structure (Wu & Palmer, 1994). Then, similarity could be the inverse of distance, however there are many definitions of distance (Bisson, 2000). Mougouie and Bergmann (2002) propose two methods based on optimisation. Armaghan and Renaud (2012) suggest a retrieve step based on the use of a multi-criteria selection to improve this step. Therefore, some conclusions of previous researches lead to the idea that similarity depends on the study case. Rifqi (2010) explains that similarity depends on the general context of the domain and Goldstone and Barsalou (1998) highlight that it can also depend on the conditions of the study. Montani (2011) shows the importance of the context in CBR system which can help to reduce the retrieval search, to revise conclusions, and to adapt knowledge and strategies. In the same logic, Leake studies the possibility to adapt the similarity measure to the context (Leake, Kinley, & Wilson, 1996). Indeed, some distance measures are based on knowledge integrated in the system during the development step. It shows that similarity measure needs an additional knowledge which comes from the kind of problems solved by the CBR. In the same idea, Xiong (2011) proposes a system based on fuzzy rules which are learned by the system using genetic algorithm on a case database. This system allows the adaptation of the selection and the integration.

Furthermore, another question is to know how the data have to be saved in the system. In other words, how the information is structured in the CBR system. To reuse cases, it is important to organise them under a structure facilitating the research and therefore the application of the similarity measure. Different approaches are detailed in the literature, for example, Díaz-Agudo and González-Calero (2001b) use Galois Lattice for a CBR system. For these authors, this method offers the possibility for the system to answer to different demands. Branting and Aha (1995) propose to use stratified case based reasoning, using abstraction of case in

a hierarchical structure. Napoli et al. (1996) study the retrieval and adaptation steps of CBR under the same data organisation. As well as for cases structuring, the organisation of cases in the data base is important. Usually, the organisation is based on a concept hierarchy which contains nodes ordered by relation as “is_a” (Gennari, Langley, & Fisher, 1989). In lattice theory, the organisation includes binary relations as “is a part of” or “is contained in” (Birkhoff, 1940). The result of this kind of organisation of concepts is called a taxonomy. However, for Díaz-Agudo and González-Calero, the classification process based on taxonomy structure needs to anticipate the questions submitted to the system (Díaz-Agudo & González-Calero, 2001a).

To organise the data structure with the aim to simplify the retrieval step, an approach is to generalise cases. This idea is not new and some authors explain that it is present since the beginning of CBR (Bareiss, 2014; Kolodner, 2014). The use of generalised cases gives many advantages such as the possibility to use them when the problems do not represent structure allowing to be partially ordered (Napoli et al., 1996). General cases can be defined as a case globally described and therefore it can incorporate different cases. For Mougouie and Bergmann, a generalised case cover a part of the CBR knowledge and they define it as a subset of representation case (Mougouie & Bergmann, 2002). In the same idea, Díaz-Agudo and Gonzales-Calero group cases with shared properties (Díaz-Agudo & González-Calero, 2001b). It is also possible to find an analogy with the clustering method. Gennari et al. explain that conceptual clustering permits understanding the world and making predictions (Gennari et al., 1989).

Therefore, the next points to take into account are how to realise these generalised cases and how to organise them. Different approaches can be found in literature. For example, Díaz-Agudo and González-Calero (2001a) think that ontologies can be useful to design knowledge intensive CBR, to reduce the knowledge acquisition and they use Formal Concept Analysis to produce the concept lattice. A method for the resolution based on known cases is the analogy. It is defined as a mapping of knowledge from a base and a target and can be used in reasoning (Falkenhainer et al., 1989). According them, analogy allows generalizing cases in an abstract one. The implementation of an analogical reasoning depends on the knowledge representation. For example, Cornuéjols (1996) explains that this process is based on the comparison between two graphs when a case is described as a network structure. However, Bunke and Riesen (2011) observe a lack of method in the recognition patterns with graphs which highlights the complexity of the task.

3. Flexible case representation

As explained in the previous section, there are two major kinds of representation: the classic feature values description and another one based on connections as graph or semantic network. The first one allows simplifying the similarity process because it avoids a random mapping phase. Each feature value is fixed for each case. This description defines a priori the representation and therefore a part of the interpretation of the reality and it limits the kind of elements which can be described. However, for the aim application domain, it is important to have the flexibility to represent different elements and to enable a most complete description of the cases. Indeed, a waste can take several ways of description which are not common. This last point is important because the description has to adapt itself to the kind of solutions. Therefore, authors chose to use a kind of network structure to describe the knowledge. Authors define two levels of description for the knowledge. The higher is composed by two elements: *states* and *relations*. In this level, it is possible to compare *states* to nodes and *relations* to edges of a graph. Our model is not based

on graph theory, but it can be represented by graphs. In a higher representation, there is the description of the links between these states. Therefore, in our CBR system, authors do not define case in the knowledge structure. The second level is the detailed description of a state. These points will be explained in the following parts.

3.1. Representation of the state

The *state* is a description of an element. In our system, a state is represented with a network structure. This structure is based on connections linked to objects (for example *rubber*) or concepts (for example *metals*) and some parameters. These parameters permit including quantities as for example the number of objects contained in another one, values associated to units or value ranges allowing the introduction of a kind of fuzzy logic. Therefore, it is not a binary relation but a predicate which represents a fact (Falkenhainer et al., 1989).

Definition 1. A connection between concepts or objects is defined as: `def(State, Relation, Object1, Numeric value 1, Numeric value 2, Unit, Object 2)`

A state represents a situation and therefore, it can represent different things. For example, in the case study, a state defines a waste or a set of wastes. However, it is possible to describe other things such as human situation or conceptual situation. For example, describing the situation between a team with its members and other elements needed.

An object is defined only by its relations with concepts as in an ontology. The name of an object is important only to ensure the cohesion in a state description. Therefore, in a same state, it is important to ensure that a name of an object is always used for the same thing. Consequently, the definition of an object is its relations with other objects or concepts as in ontologies.

Finally, all the concepts are linked in a taxonomy, which is a limited ontology. Therefore, our model of representation of state is based on connections. In a global view, it is possible to consider that each state is linked in a huge network and, therefore, that each state is linked to other states.

Another element is the introduction of *global definition*. A *global definition* is a set of properties constituting the structure which are sharing by all objects. For example if *table* is defined with a global definition containing these following elements, *is in wood*, *has four feet*, each object respecting these properties is, by definition, a *table*.

Definition 2. A *global definition* is a structure containing a minimal set of properties defining a type of object.

It is a major point of our methodology because it defines the similarity. In other words, if a description of a state (*state_1*) satisfied a *global definition* of another state (*state_2*) then this state (*state_1*) can be considered as equal to the second one (*state_2*).

Example 1. `def(state1,is_composed_,tyre,_,_,rubber).`
`def(state1,is_composed_,tyre,_,_,metal).`
`def(state1,is_composed_,tyre,_,_,fiber).`
`def(state1,has_the_form_of,tyre,_,_,torus).`
`def(state1,has_the_color,tyre,_,_,black).`
`def(state2,is_composed_,tyre_granule,_,_,rubber).`
`def(state2,is_composed_,tyre_granule,_,_,metal).`
`def(state2,is_composed_,tyre_granule,_,_,fiber).`
`def(state2,has_the_form_of,tyre_granule,_,_,granule).`
`def(state2,has_the_color,tyre_granule,_,_,black).`
`def(state3,is_composed_,tyre_powder,_,_,rubber).`

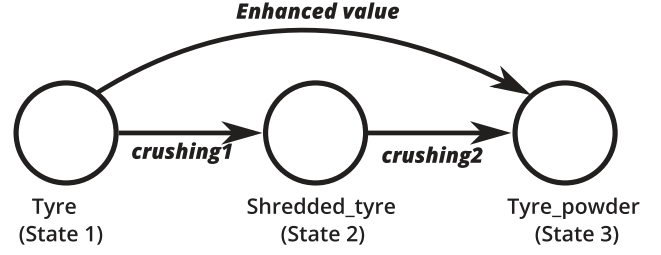


Fig. 2. Example of connections between states : following example 1.

```

def(state3,is_composed_,tyre_powder,_,_,metal).
def(state3,is_composed_,tyre_powder,_,_,fiber).
def(state3,has_the_form_of,tyre_powder,_,_,powder).
def(state3,has_the_color,tyre_powder,_,_,black).

```

where *tyre*, *tyre_granule* and *tyre_powder* are objects, *rubber*, *metal*, *fiber*, *black*, *torus*, *granule* and *powder* are concepts and *is_composed*, *has_the_form_of* and *has_the_color* are relations between objects and concepts. This is a simple example showing different states describing a process like in our case study.

A *state* is a *global definition* in our system, but a *global definition* is not necessary a *state*. Therefore, the description of a case can be different depending on the user and his interpretation of the reality. Indeed, a state described an object or concept in the real world which is linked to other states by relations whereas a global definition can be the description of a state or the description of an abstract object belonging to the reasoning world of the system. These abstract objects are types of representation of state but they do not represent the concrete object.

3.2. The link, an element composing the case

This part deals with the connections between two states, that is to say the *link*. It represents different kinds of relations. Indeed, a relation can describe a fact or an action between two states. For example, if there is a state describing a father, and another one describing his son, a relation *is_son_of* can link the two states. Therefore, it is possible to consider this kind of relation as an extension of the description of a state. However, the main differences come from the role of this relation in the CBR system. Indeed, if there are two states, each state can play the role of problem or solution during the resolution process. Therefore, the choice of the description, i.e. if facts are described by different states or only one depends on the kind of problem submitted to the CBR system. Consequently, the model proposed can be used for several problems. Its specification depends on the representation of the information during the learning process. A relation can also describe the result of a process or a transformation. The difference with the first relation described is the time. With the first relation, father and son, the two states can exist in the same time i.e. is a static fact. However, in the second kind of relation, a state will exist after another one. This reasoning is in accordance with human mind. But, in the CBR system, the relation is defined differently. It distinguishes relations representing facts and others representing results of processes. For example, if there is *tyre*, “enhanced value” *tyre_powder*, it is possible to describe this relation by two states, one for *tyre*, one for *tyre_powder* and a link *enhanced_value* as in an ontology or concept map (Fig. 2). With the same idea a product *A* can be linked to a product *B* by *transformation_1* and it means that *A* is transformed into *B* by *transformation_1*. The advantage of this structure of knowledge representation is the possibility to link a state to some other states. As a consequence, it allows expressing all the possible representations for a state under different

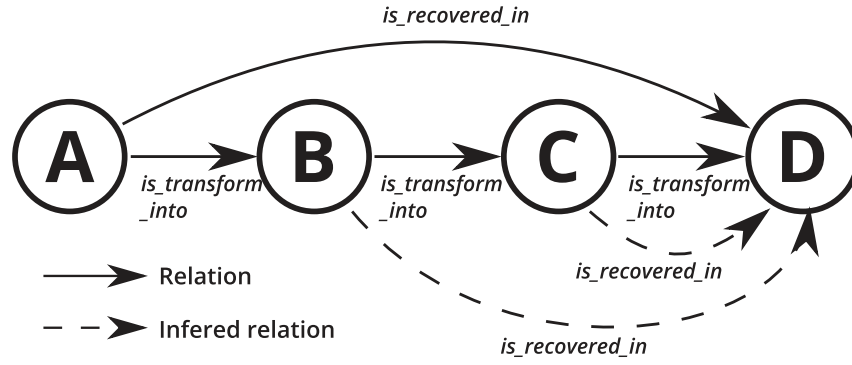


Fig. 3. Example of enhance number of possible cases using inference mechanisms.

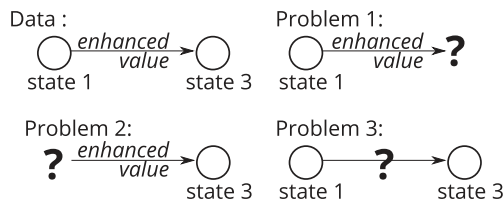


Fig. 4. Example of dynamic cases.

meanings. In our CBR system this concept allows describing different possibilities of waste treatments or processes from an element or a set of elements. It also permits building different levels of relations. For example, a process can be defined between two states, and other processes including other states can describe the same knowledge under a lower level of representation as in Fig. 3. This capacity of representation is the basis of our method to revise solutions.

3.3. Dynamic case

The last point is the description of a case. By definition, a case in CBR is the couple *problem/solution*. However, in some CBR systems there are two spaces in the knowledge base as explain in the Section 2. In our system, information needs to be sometimes used as problems, and sometimes as solution. Therefore, authors propose to define a case, i.e. a couple problem/solution as a knowledge structure composed by two states and a link.

Definition 3. A case is defined as a set composed by two states and a link where one state represents an initial state (the object of the question), the link is what is wanted (the verb of our question) and the last state represents the solution. Only two of these three elements are necessary to constitute the problem.

For example in (Fig. 4) there is the data: *state1 (tyre) → enhanced_value → state3 (tyre powder)*. From this data, it is possible to infer 3 problems:

- What is the final state to enhance_value of a tyre ?
- What is the initial state of the tyre before to enhance its value ?
- How to reach the state 3 (tyre_powder) from the state 1 (tyre) ?

To realise this part, some inference mechanisms are used which encapsulate states as part of the solution and others as a part of the problem. Authors call this *Dynamic cases*.

Definition 4. In a CBR system, a case is dynamic when the identification of the problem part and the solution part is realised during each Retrieve step. That is to say, the system does not store cases but only knowledge on states and their relations. Moreover, this knowledge with some mechanisms produce cases corresponding to the current problem as the need arises.

This mechanism has several advantages. One of them is the possibility to exploit more information of the knowledge than in a classical division of the spaces. Another one is the possibility to use information (states or links) as part of a problem and as a part of the solution during the same resolution process i.e. for the same problem. This capacity of the mechanism is the basis of the revise step in our system (not detailed in this paper). Indeed, this step is based on the decomposition of problem into sub-problems, which allows adapting the solution using different cases and not only one.

4. Case retrieval and similarity

In the literature, there are many examples describing the retrieval step in CBR. In addition to a similarity measure, there is often a mechanism to try to identify the most similar case by limiting the exploration with for example, filters or indexation techniques. This part explains how the process of selection of similar cases occurs as well as the different mechanisms which allow reducing the time of research. The presented methodology is divided into two phases. The first one explains how the knowledge is stored and processed in order to apply the research step. The second one describes the research algorithm. The major difference between the two parts of the methodology is their runtime. Indeed, the first part is realised as a learning step i.e. during the introduction of new knowledge. On the contrary, the research step occurs during the resolution process. Therefore, the realisation of these two parts can be separated in time and in processes.

4.1. Pre-phase: learning phase

This part of the process is realised during the introduction of new knowledge in the system. It can take place during the initialisation of the CBR system or during its utilisation thanks to the **retain** step. It is possible to divide this process into three parts. The first one requires the intervention of the expert in knowledge management and the two others are automatically operated by the system.

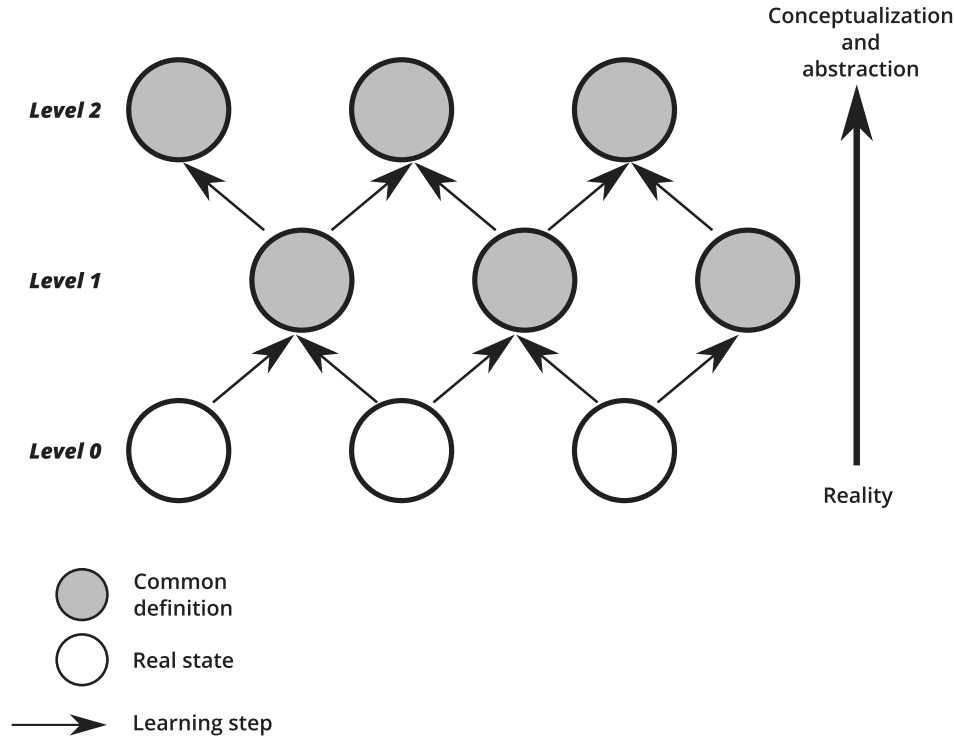


Fig. 5. Creation and storage of common definitions in the learning step.

4.1.1. Pre-phase step 1: introduction of knowledge

The learning phase starts when a user introduces new knowledge containing a set of states linked by relations as explained in Section 3. To reduce knowledge engineering efforts, the user previously describes each state, the relations between states and he completes the taxonomies if new concepts are introduced. Finally, this knowledge is integrated in the system sharing the same semantic network.

4.1.2. Pre-phase step 2: enhance relations with inferences

Once new knowledge is introduced in the system, the learning step starts. The system starts to enhance the number of relations between the states. This process is based on the use of taxonomies for the relations and allows generalizing these relations i.e. the process realises a conceptualisation of the new knowledge focused on its relations. Inference mechanisms are used for this task with different rules introduced during the initial conception of the CBR system. There are two kinds of rules. The aim of the first category is to conceptualise relations. These rules permit replacing a relation by another one defined in a higher level of the taxonomy. The aim of the second one is to make inferences on the relations in the input knowledge (Fig. 3). In other words, when a relation between two states is defined with other relations, authors propose to include these relations as a part of the first one. This mechanism is relevant because it increases the number of possible cases.

For example, if there is *A is_transform into B*, *B is_transform into C*, *C is_transform into D*, and *A is_recovered_in D*, in some kinds of problems, it is possible to affirm that *B is_recovered_in D* and *C is_recovered_in D*. In the same way, if a relation is defined in a taxonomy, it is possible to enhance the relation of the state with more conceptual links. For example if *A is_fixed_with_glue to B*, and in a taxonomy there is ***fixed_with_glue is fixed*** therefore authors propose to infer that *A is fixed to B*.

4.1.3. Pre-phase step 3: completion of common definition structures

Once new knowledge is introduced and its relations have been inferred, the CBR system prepares the research step with a phase of "learning". This phase is a kind of indexation and conceptualisation of the information as it is possible to find in the literature (Section 2). However, there are many differences with the traditional methods. The conceptualisation is not focused on case as for Bichindaritz (2008) because cases do not exist in the system (they are inferred), but in states. More precisely, for each relation (the both directions are possible) between states (inferred or not), states are grouped together in a *common definition structure* which represents each kind of resolvable problem (Section 3). A *common definition structure* is composed by different level of states. In the lower one, there are the states introduced in the system. Therefore, the lower level represents the reality. When a new state is introduced in the structure, the system will create a *common definition* as result of the combination of this state and the states existing in the lower level.

Definition 5. A common definition is a state arising from two states "origins" and it is a global definition applicable to these two origins. If an object satisfies the properties of this common definition then this object can satisfy the properties of one origin but is not sure. Conversely, if it does not satisfy the properties, it will not satisfy the properties of ether origins.

The mechanism creates a second level composed only by *common definitions*. Then, the mechanism works with the new elements created in this level and it generates new *common definitions* in a higher level and so forth (Fig. 5). Some system's parameters permit defining the rate of mixing for each level.

A *common definition* is generated using abstraction mechanism and conceptualisation mechanism. The first one allows deleting properties, i.e. relations as defined in part 3.1, which are not present in the two original states. The second one, the

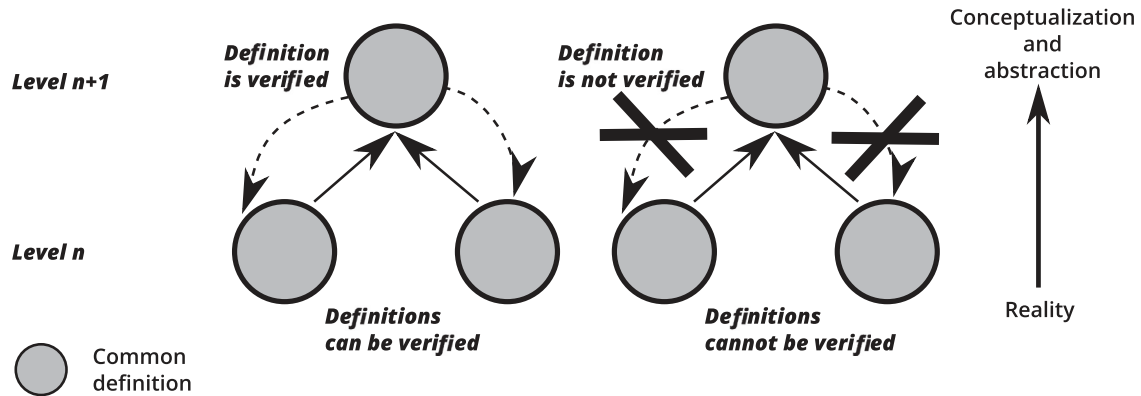


Fig. 6. Mechanism of research.

conceptualisation, is based on the use of taxonomies of concepts. If the two original states have the same property but defined at different levels of conceptualisation (for concept, relation or both), this mechanism creates or selects the property in the higher level. Therefore, this new property satisfies the two original ones. For example if there are the two original properties: *is in copper*, *is in led*, the property *is in metal* satisfies the two original ones if, in a taxonomy, copper and led are defined as metal (copper is_a metal and led is_a metal). For numbers, the fuzzy logic is used with range value which contains the two original values.

The structure obtained is not a cluster because there is no root. Indeed, for a combination of two states, the mechanism can produce different *common states* if they contain different objects. In this case, a kind of matching process is realised and it generates all the mapping possibilities. All the combinations with objects are possible if the two matched objects shared at least one common property. However, in the structure, implausible *common states* are not saved in higher levels because they do not shared properties with the others. Therefore, the more a level is higher, the more the *common states* contained properties shared by all the states. Then, two processes are supported by the structure. First, there is an indexing which organises states. Second, there is a filter and weighting system. For the relation, higher levels contain only the main properties and the more a property is present in higher level, more is important.

In conclusion, in the learning phase some knowledge structures are enriched by the new state and by the creation of common definitions and there is one structure by relations (links) originating in this state. The level zero of this structure is the reality. The higher the level is, the more the conceptualisation and abstraction degree is important and the levels are composed by states with very large definition.

4.2. Retrieve phase: research of similar states

This part of the process is realised during the resolution of a problem. It is based on the use of *common definition structures* created or completed in the learning phase. It is composed of three steps.

4.2.1. Retrieve phase step 1: selection of the common definition structure

As the problem is defined as the combination of a state and a relation, the first step of the retrieve part is to select the common definition structure corresponding to the relation and the type of the submitted problem, (Fig. 4).

4.2.2. Retrieve phase step 2: evaluation of states

The next step is to check if the problem's state satisfied the common definitions present in the selected structure. The research mechanism converts the definition of these states into rules where objects are converted into variables. Then, it starts to check if the rules are applicable to the problem's state, i.e. if the problem's state contains all the properties contained in the rules with the same level of conceptualisation or a lower one. The mechanism begins with the common definitions of the higher level of the structure. If a common definition is satisfied, it continues with the original ones. If it is not satisfied, the mechanism checks another common definition in the higher level until there is no more (Fig. 6). The mechanism stops the exploration when a common definition from the lower level is verified (and it continues with another one from the higher level) or when all the common definitions from the higher level were tested.

During this phase, each verified state is stored in a list with its associated level.

4.2.3. Retrieve phase step 3: selection of the most similar state

Once the exploration is finished, an ascending sort on the level is realised with the stored elements. Logically and following the description of common definitions, if there are verified states coming from the level zero (lower level), that means that the current problem can be defined as solved problems existing in the data base. Therefore, they can be considered as similar to the current problem and used to solve it. If there is no state coming from the level zero, the system will select the state with the lower level. It is not a real state but a state generated during the learning phase. However, authors propose to define it as the most similar one to the current problem and to use it to solve it. Here, it is possible to measure the similarity.

Definition 6. In a common definition structure, the more a state is verified with a lower level, the more similar it is to the current one.

Finally, if there are several verified states in the lower level, the system can apply different policies. As this system can not determine which one is the most similar (all are in the same level), the system can randomly select one or proposes each ones as a possible solution.

5. Case study: recovery treatment

Authors implement the method previously described in a CBR system dedicated to generate new recovery processes for wastes

treatment. The idea is to use known recovery process for a waste or a type of waste to propose new ones for other different wastes. The aim of the CBR is not to define each unit process with all the parameters and to give solutions ready for use, but to give the main steps of a new process and to try to estimate the final product and what will be its functions or potential applications. Therefore, authors developed a CBR based on logical programming paradigm with the SWI-Prolog implementation, which is a free software ¹ and it comes for Linux². SWI-Prolog extends Prolog language (Wielemaker, 2014).

5.1. Data

The data used comes from different known recovery processes for 6 types of waste. The selected wastes and their solution share some common steps and generally some relations allowing the use of CBR system. The 6 kinds of waste are:

- Wastes composed of polypropylene
- CRT television composed of elements containing glass, metal, plastic elements
- Neon tube composed of elements containing glass, metal, chemical compounds, gas
- Glass bottle
- Wastes composed of aluminium
- Car battery

For example, the definitions of some wastes introduced in the system are the following:

```
(def(compose,bottle'cap',',',polypropylene),
def(has,bottle'cap',',',metal),
def(size,bottle'cap',3,',cm,'),
def(form,bottle'cap',',',tube)
),
```

or

```
(def(has,neon'tube',',',glass'tube),
def(composed,glass'tube',',',glass),
def(has,neon'tube',',',powder'PhM),
def(composed,powder'PhM',',',phosphorus),
def(composed,powder'PhM',',',mercury),
def(has,neon'tube',',',piece'metal),
def(composed,piece'metal',',',metal),
def(form,glass'tube',',',tube)
),
```

Each waste can have several recovery processes, furthermore each process can be divided into other sub-processes in function of separation steps. To complete the knowledge base of the system, authors include taxonomies on operations and concepts. In this system, a taxonomy can be completed progressively depending on the concepts used in the knowledge base. The taxonomy on operations is a tree structure ordering processes in families and sub-families. With the same idea, taxonomies on concepts are divided into two structures. One concerns the component where it is possible to find concepts as glass, metal, aluminium, etc. The other one is about geometry and allows describing forms and architecture of objects. Each transformation step is modelled

in the system as a link and each product or intermediate product is defined as a state. Moreover, authors define relation between states creating a crude approximation of the process. These links are included in the taxonomy. For example, a waste is connected to the end product with a link defining the whole recovery process.

5.2. Experiments and results

All the data are introduced into the CBR system and the learning phase is launched. Authors only present a fragment of the database obtained. It contains a huge number of information, therefore, only three kinds of fragment are presented below:

- Fragment of Def, which describe the definition of states (the information is condensed).
- Fragment of taxonomies.
- Fragment of Relations, which describe the relations between two states.

Fragment of Def database:

```
:- dynamic def/7.
```

```
def([2], taille, bouchon, 3, ', cm, ').
def([3], taille, pare'choc, 2, ', m, ').
def([4], contient'trace, broyat'polypropylene'sale, ',
    ↳ ', ', metal).
def([7], compose, poudrette'polypropylene'humide, ', ',
    ↳ ', polypropylene).
def([7], taille, poudrette'polypropylene'humide, 1, 3,
    ↳ mm, ').
def([7], forme, poudrette'polypropylene'humide, ', ',
    ↳ ', poudrette).
def([7], contient'trace, poudrette'polypropylene'humide
    ↳ ', ', ', eau).
def([8], compose, poudrette'polypropylene, ', ',
    ↳ polypropylene).
def([8], taille, poudrette'polypropylene, 1, 3, mm, ').
def([8], forme, poudrette'polypropylene, ', ',
    ↳ poudrette).
def([9], contient, tv, ', ', tube'cathodique).
def([10, 9], compose, tube'verre, ', ', verre'plomb)
    ↳ .
def([13], compose, broyat'plastique, ', ', plastique
    ↳ ).
def([13], forme, broyat'plastique, ', ', broyat).
def([14], compose, residu, ', ', salete).
def([14], forme, residu, ', ', broyat).
def([16], compose, verre'broye'sale, ', ', verre).
def([17], compose, verre'broye, ', ', verre).
def([19], compose, verre'plomb'broye'sale, ', ',
    ↳ verre'plomb).
[...]
```

¹ <http://www.swi-prolog.org/license.html>.

² <http://www.fraber.de/university/prolog/comparison.html>.

Fragment of taxonomy database:

```
:- dynamic ontologie/2.

ontologie(transformation(broyage'bouchon), traitement).
ontologie(transformation(broyage'parechoc), traitement)
    ↪ .
ontologie(transformation(broyage'fin), traitement).
ontologie(transformation(broyage'2), traitement).
ontologie(transformation(broyage'3), traitement).
ontologie(transformation(broyage'4), traitement).
ontologie(transformation(broyage'5), traitement).
ontologie(transformation(broyage'bouteille), traitement)
    ↪ .
ontologie(transformation(broyage'alu), traitement).
ontologie(transformation(broyage'batterie), traitement)
    ↪ .
ontologie(transformation(flotation), traitement).
ontologie(transformation(tri'a), traitement).
ontologie(transformation(tri'b), traitement).
ontologie(transformation(separation'poudre'a),
    ↪ traitement).
ontologie(transformation(separation'poudre'b),
    ↪ traitement).
ontologie(transformation(separation'chimique'a),
    ↪ traitement).
ontologie(transformation(separation'chimique'b),
    ↪ traitement).
ontologie(transformation(tri2), traitement).
ontologie(transformation(separation'metal'a),
    ↪ traitement).
ontologie(transformation(separation'metal'b),
    ↪ traitement).
ontologie(transformation(separation'verre), traitement)
    ↪ .
ontologie(transformation(separation'oxyde'plomb),
    ↪ traitement).
[...]
```

Fragment of relation database:

```
:- dynamic relation/3.

relation(traitement, 2, 6).
relation(traitement, 3, 6).
relation(pretraitement, 6, 8).
relation(revalorisation(matiere), 2, 8).
relation(revalorisation(matiere), 3, 8).
relation(pretraitement, 9, 11).
relation(traitement, 11, 13).
relation(traitement, 11, 14).
relation(pretraitement, 9, 15).
relation(pretraitement, 9, 18).
relation(traitement, 15, 16).
relation(traitement, 18, 19).
relation(pretraitement, 16, 17).
relation(pretraitement, 19, 20).
relation(revalorisation(matiere), 9, 13).
relation(revalorisation(matiere), 9, 14).
relation(revalorisation(matiere), 9, 17).
relation(revalorisation(matiere), 9, 20).
relation(traitement, 21, 25).
relation(traitement, 21, 26).
relation(traitement, 21, 28).
[...]
```

Then, different questions are submitted to the CBR system to try to assess the efficiency and the capabilities of the proposed methods. The first element tested is the learning phase. During this phase, the system produced some structures containing common definitions as expected. It shows that the mechanism is realisable. First, all the states composing the different processes linked with

```
Resolution -->2
2,enhance_value(material),8
---2,treatment,6
-----2,transformation(crushing_cap),4
-----4,transformation(separation),5
-----5,transformation(crushing),6
---6,pretreatment,8
-----6,transformation(washing),7
-----7,transformation(drying),8
true.
```

Fig. 7. Example of result, where each number represents a state.

different connexions had been inferred. Authors have filled the knowledge base with 45 states and the inference mechanisms generated 137 different *real* possible relations. Then, it creates some structures with around 5000 common definitions. Then the CBR is used to solve some problems describing new states and the desired link. In the first request someone demands solved problems introduced in the system as data. All queries work well and, during the tests, the reuse step gives relevant answers when there are no differences between the desired problem and a known solved problems. In a second phase, the system is tested with problems where inference mechanisms are required, for example, by submitting problems composed by intermediate product and general definition of the process (the link originating from the waste and finishing to the new product). One example is illustrated in Fig. 7 where a concrete problem is solved:

1. The waste which is a small material composed by polypropylene (state 2) is found to be recovered into polypropylene powder ready for use (state 8) via two main processes defined as treatment and pretreatment. Treatment groups correspond to material modification operations and pretreatment is conditioning.
2. The treatment phase is identified by three operations: crushing_cap which corresponds to an operation of crushing little pieces in polypropylene, separation of non-polypropylene material and another phase of crushing to obtain a powder of polypropylene with some impurities.
3. The pretreatment phase which consists in washing the powder obtained and to dry it.

Here too, for all the tests the system gives relevant answers that means each step of the process and an estimation of each states representing intermediate products work correctly. After these experiments to validate a simple reuse mechanism, the system have to solve two categories of states which were not in the database.

The first one is describing states, copies of existing states, but where properties were modified with new concepts deriving from the original one. For example, if there is a property in the original state as *A is_fixed_with conceptA*, authors define a new state with the following properties *A is_fixed_with conceptB* where **conceptB is_a conceptA** in the taxonomy. The mechanism of conceptualisation is tested and, more precisely, the system's capacity to affirm that two elements which are not in the same level of taxonomy description can be similar. For example:

1. *A is_fixed_with conceptA*
2. *A is_fixed_with conceptB*
3. **conceptB is_a conceptA**

it is possible to evaluate the following assertions:

Role A		Role B
2 is true	⇒	1 is true
2 is false	⇏	1 is false
1 is true	⇏	2 is true
1 is false	⇒	2 is false

```

Resolution -->17610
No solution found
Propositions : return known similar case
How many elements do you want ?5.
Similar case of level :1 case number :95 coming from
    ↪ :[2,15]
Similar case of level :1 case number :96 coming from
    ↪ :[2,18]
Similar case of level :1 case number :173 coming from
    ↪ :[3,15]
Similar case of level :1 case number :174 coming from
    ↪ :[3,18]
Similar case of level :1 case number :179 coming from
    ↪ :[3,29]

true.

```

Fig. 8. Example of fail return in the resolution process.

Therefore, the system tries to determine if a property of the current state satisfied a property of a known state, the current state plays the *Role A* and the known state the *Role B*. The system tries to establish that the current state has the properties to satisfy the definition of the known state. This is why authors call this kind of description *Global definition*. Here again, as expected the mechanism was validated and good results were found in accordance with the theory.

The second one is to increase the number of properties in the state compared to the original one. The idea tested here is to ensure that the abstraction mechanism works correctly and therefore, that the interpretation of the state by the user does not impact the global resolution if it satisfies the minimal description defined by the *Global definition*. For this test, authors create some states as copies of states coming from the knowledge base. Then authors add some properties to them to ensure that the generated states are not exact copies of existing ones. For example, **state_1** has the following definition:

1. A *relation_1* Concept1
2. A *relation_2* Concept2
3. A *relation_3* Concept3

A new **state_2** can be created with the following definition:

1. A *relation_1* Concept1
2. A *relation_2* Concept2
3. A *relation_3* Concept3
4. A *relation_4* Concept4
5. A *relation_5* Concept5

The first definition is included in the second one. Let define Def_1 the set of properties describing **state_1** and Def_2 the set of properties describing **state_2**. The following assertions can be established:

Def_1	\subset	Def_2
Def_1 is true	\Rightarrow	Def_2 is true
Def_1 is false	\Rightarrow	Def_2 is false
Def_2 is false	\Rightarrow	Def_1 is false
Def_2 is true	\Rightarrow	Def_1 is true

Therefore, the mechanism of comparison verifies that the definition of the current state is equal or including the definition of the state coming from the data base and after that all properties of this last state are verified by the current one. During the test where several problems have been submitted, all solutions have been found showing that the mechanism gives relevant results and

therefore that a more complete description than the original one does not stop the resolution step.

The third test is the opposite of the second one. The idea is to propose to the system to solve states for which, compared to the states in knowledge base, some properties are missing. For the example of the second test, roles are reversed. This test tries to assess the answers of the system where the definition of the current state is incomplete to verify a known solved state. Therefore, authors submit different states deleting different parts of the definition. For all the tests, no known state coming from the level 0 has been found. Therefore, the system gives the most similar states found (Fig. 8).

These results can be explained because known states found are *Common definitions* from the level 1 or upper. However, the system describes in the previous part, returns the origins (real states) of the common definition with the lower level. Some of them are randomly selected and the solving process (not described in this paper) used them to estimate the generated product. The generated results are different. Some of them correspond to the original solution. However, others are unexpected and propose processes not very compatible with the original state. But, all the returned solutions are logic compared to the description of the current state. Finally, authors conclude that the proposed system works under the logic of *Global definition* (the fact that a Global definition is the minimal description of an object).

6. Discussion

The method detailed in this paper allows realizing some steps of a CBR system designed to generate new recovery processes for waste. In a first time, it permits describing the knowledge under two levels. The state represents a situation or a thing with a model based on connections using relations, concepts and numerical properties. This representation allows a flexible description of a situation and it allows representing a very wide variety of situation. The second level is the network composed by states and links or relations. It represents the relations between states and it is the level of problem resolution.

The proposed method does not store knowledge in a space for problems and another for solutions. There is only one containing the network of states and links. The CBR's cases are generated by a set of inference mechanisms which define a part as problem and another as solution. This method allows modifying the status of a set of knowledge (problem or solution) during the same problem resolution.

The application of inference mechanisms permits enhancing the knowledge stored with use of taxonomies. The relations and concepts describing states or relations between states are more conceptualised. Therefore, it increases the possibilities to retrieve a similar case.

The proposed retrieve method is not common for traditional CBR system. Indeed, it does not measure a distance between two points of the knowledge space and does not use any weight or similarity values. The method is based on the assumption that a state encompasses the minimal set of properties needed to describe a situation. Therefore, a situation verifying all the properties of a state is considered as describing the same state. The classical CBR systems provide a minimum range set of properties. These sets describe problems and position them to each other. Almost, they are limited to the minimal set which allows doing this positioning used by the Retrieve step. However, the proposed description method links a set of properties to a major concept represented by a state, that is to say, a set describes a state which is a concept as an object or a waste. Therefore, the method tries to identify objects or concepts, whereas classical CBRs try to position the problems retrieve to each other. The retrieve method is based on this assumption. Structures of knowledge containing states and more conceptual and abstract states are building for each relation during the learning phase. These structures allow reducing the time of research, filtering the important properties for the relation and weighting them. Therefore, the method permits a retrieve CBR's step based on logical deduction.

The result of this method is the acquisition of a list of states ordered in decreasing similarity values. The inference mechanisms allow enhancing the flexibility of the research and permit considering as equal elements in different levels of description and enhancing the creativity of the CBR system by the realisation of original combinations of objects or concepts. These original combinations are a logical consequence of the mechanisms described in Section 5.2, i.e., a state or a part of a state can be considered as similar to a more conceptual or more abstract objet. Therefore, solutions which are not known to resolve the current problem can be applied to this state. As an example, a glass bottle can be considered as glass material and the glass material's solutions can be applied to the glass bottle. However, a glass bottle can be considered as a container like a wood box or a flower pot for example. Therefore some solutions applied to a box wood or to a flower pot can be used to resolve the glass bottle problem. Authors think that these kinds of reasoning can lead to creative processes.

In addition, a comparison between some methods found in the literature and the proposed method can be summarised. The main distinction is the fact that there is not a problem space and a solution space in the exposed method. The majority of CBR systems are based on this distinction which reduce the possibility of this kind of system. Moreover, the presented method contains some common elements with traditional CBR. As explain in the Section 2, Amaïef and Lu (2013) use an ontology to improve the comprehension of a submitted problem and its representation. However, in the proposed method, the ontology contains all the information and it is the support of the generation of case. Therefore, the ontology evolves during the time. Another comparison is the similarity measure. The proposed measure is not based on a distance measure with the use of mathematical formula as in most of the CBR systems but if the submitted problem is a part of a real group of case or more conceptual groups. As for Xiong (2011), the method adapts by the creation of new groups, common definitions, and by the introduction of new concepts in the taxonomies. Finally, the flexibility to describe a case is a major distinction with a main part of CBR system presented in the literature review.

However, the proposed method has some limitations. As the knowledge is described by states and links, the method imposes

that a problem can be described under this form. In the system a state has to describe a static situation in the intellectual approach of the problem resolution. In other words, a state has to represent a step in this resolution.

Another limitation is the main assumption of this method, i.e. the capacity to describe a situation with a minimal set of properties taking into account that all situations with these properties will be considered as similar. Whereas, it increases the creativity of the problem resolution, this point is also a limitation because it can consider equal two different situations because the description of the state is not strict enough and therefore it can lead to inconsistent association.

The use of taxonomies can also be a limitation in the CBR system. A taxonomy is a data structure where different concepts are organised in a hierarchical structure. However, this hierarchical structure determines an interpretation of the reality. This interpretation impacts and limits all the mechanisms using these taxonomies during the problem resolution phase and therefore, this phase is oriented to follow this interpretation. In other words, the use of taxonomies reduces the quantity of solution generated and requires a sharp knowledge on the application domain. In addition, this method needs to be able to create these taxonomies.

Finally, the realisation of this method raises different problems. To realise the retrieve part, the CBR system builds knowledge structure containing states and common definitions which are combination of the properties of these states. In addition, the combination of two states can produce different common definition allowing a kind of creativity and the inference mechanisms increase the number of possible combinations. Therefore, the number of common definition grows exponentially with the introduction of new states. In our application, for a number of 45 states described and 137 possible cases generated by inferences, the system during the learning phase produced around 5000 *Common definitions*. The structure's parameters, as the number of slices or the rate of mixing, are not optimally defined. The consequences of this is the tremendous computational time of the learning step. Also, this original number of possible usable cases is small (147) but the trajectories described share part of solutions or some common states. This sharing brings to light the possibilities of this presented method by a possible recombination of solutions and producing a creative process. Nevertheless, this method will be tested with a significant number of cases when a second version of this method will be developed.

Another difficult point is the possible random selection of the most similar known state. In fact, if under the logic of the system there is no doubt, it appears to be important to develop a good policy of selection depending of the resolving method. Thus, it can be interesting to select every known states from the lower level if the number of possible combinations is not important in the resolution process. On the contrary, the random selection can produce non-deterministic solving process and some good solutions may be lost.

Finally, another difficulty can appear if the state is described with a lot of slices. For example if a state is described where an object is composed by other objects defined with other objects or concepts, there are no problems during the research step. However, once a source case is selected from another level than the level 0, the mechanism of adaptation has to resolve a random mapping process because the satisfied common definition was done with the abstraction of some properties.

7. Conclusion and outlook

This paper deals with the similarity measures in CBR and also with the representation and memorisation of knowledge. Cases are not described with the classical feature-value representation, but

it proposes to describe them with a network structure. In the proposed method, knowledge is stored so that it enables the generation of dynamic case and the application of inference mechanisms. These mechanisms permit increasing the flexibility of the system's logic and therefore to give many original solutions. It also allows weighting the importance of some properties taking into account the context of the problem but also the kind of solution. To reach this goal two definitions of concepts are introduced which are the base of this method. Moreover, it presents a structure composed by common definitions which plays the role of indexation mechanism and filter. All these points enable to design a flexible CBR which can be used with very different kinds of problems. However, two limitations are identified. Firstly, to be able to describe the knowledge under the structure of *state - relation - state* where the properties of the situation or the object are contained in state. The second is to have several states linked with the same relation to provide the necessary elements to generate the structure composed by *common definitions*.

One way to improve this method is to reduce the number of *common definitions* generated by the system to not increase exponentially the time of the learning step. Another point should be the introduction of full ontology and not one limited to a taxonomy structure. Therefore, more developed inference mechanisms should be introduced to increase the possibilities of the system without reducing its flexibility.

References

- Aamodt, A., & Plaza, E. (1994). Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI communications*, 7(1), 39–59.
- Amailef, K., & Lu, J. (2013). Ontology-supported case-based reasoning approach for intelligent m-government emergency response services. *Decision Support Systems*, 55(1), 79–97. doi:10.1016/j.dss.2012.12.034.
- Armaghan, N., & Renaud, J. (2012). An application of multi-criteria decision aids models for case-based reasoning. *Information Sciences*, 210, 55–66. doi:10.1016/j.ins.2012.04.033.
- Avramenko, Y., & Kraslawski, A. (2006). Similarity concept for case-based design in process engineering. *Computers & Chemical Engineering*, 30(3), 548–557. doi:10.1016/j.compchemeng.2005.10.011.
- Bareiss, R. (2014). *Exemplar-based knowledge acquisition: a unified approach to concept representation, classification, and learning*. Academic Press.
- Bichindaritz, I. (2008). Memory structures and organization in case-based reasoning. In D. P. Perner (Ed.), *Case-based reasoning on images and signals number 73*. In *Studies in Computational Intelligence* (pp. 175–194). Springer Berlin Heidelberg.
- Birkhoff, G. (1940). *Lattice theory*. American Mathematical Soc.
- Bisson, G. (2000). La similarité : une notion symbolique/numérique.
- Branting, L. K., & Aha, D. W. (1995). Stratified case-based reasoning: reusing hierarchical problem solving episodes. In *IJCAI* (pp. 384–390). Citeseer.
- Bunke, H., & Riesen, K. (2011). Recent advances in graph-based pattern recognition with applications in document analysis, 44(5), 1057–1067. doi:10.1016/j.patcog.2010.11.015.
- Cordier, A., Mascaret, B., & Mille, A. (2009). Extending case-based reasoning with traces. In *Grand challenges for reasoning from experiences, workshop at ijcai: vol. 9* (p. 31).
- Cornuéjols, A. (1996). Analogie, principe d'économie et complexité algorithmique. *Actes des 11èmes Journées Françaises de l'Apprentissage*.
- Díaz-Agudo, B., & González-Calero, P. A. (2001a). Classification based retrieval using formal concept analysis. In D. W. Aha, & I. Watson (Eds.), *Case-based reasoning research and development number 2080*. In *Lecture Notes in Computer Science* (pp. 173–188). Springer Berlin Heidelberg.
- Díaz-Agudo, B., & González-Calero, P. A. (2001b). Formal concept analysis as a support technique for CBR. *Knowledge-Based Systems*, 14(3), 163–171. doi:10.1016/S0950-7051(01)00093-4.
- Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The structure-mapping engine: algorithm and examples. *Artificial Intelligence*, 41(1), 1–63. doi:10.1016/0004-3702(89)90077-5.
- Garey, M. R., & Johnson, D. S. (2002). Computers and intractability.
- Gennari, J. H., Langley, P., & Fisher, D. (1989). Models of incremental concept formation. *Artificial Intelligence*, 40(1), 11–61. doi:10.1016/0004-3702(89)90046-5.
- Goldstone, R. L., & Barsalou, L. W. (1998). Reuniting perception and conception. *Cognition*, 65(2), 231–262. doi:10.1016/S0010-0277(97)00047-4.
- Kokinov, B. (1994). A hybrid model of reasoning by analogy. *Advances in connectionist and neural computation theory*, 2, 247–318.
- Kolodner, J. L. (2014). *Retrieval and organizational strategies in conceptual memory (PLE: Memory): a computer model*. Psychology Press.
- Kuo, T. C. (2010). Combination of case-based reasoning and analytical hierarchy process for providing intelligent decision support for product recycling strategies. *Expert Systems with Applications*, 37(8), 5558–5563. doi:10.1016/j.eswa.2010.02.057.
- Leake, D., Kinley, A., & Wilson, D. (1996). Linking adaptation and similarity learning. In *Proceedings of the eighteenth annual conference of the cognitive science society*.
- Lieber, J. (2002). Strong, fuzzy and smooth hierarchical classification for case-based problem solving. In *ECAI* (pp. 81–85).
- Liu, K. F.-R., & Yu, C.-W. (2009). Integrating case-based and fuzzy reasoning to qualitatively predict risk in an environmental impact assessment review. *Environmental Modelling & Software*, 24(10), 1241–1251. doi:10.1016/j.envsoft.2009.04.005.
- López-Arévalo, I., Bañares Alcántara, R., Aldea, A., Rodríguez-Martínez, A., & Jiménez, L. (2007). Generation of process alternatives using abstract models and case-based reasoning. *Computers & Chemical Engineering*, 31(8), 902–918. doi:10.1016/j.compchemeng.2006.08.011.
- Markman, A. B., & Gentner, D. (1993). Structural alignment during similarity comparisons. *Cognitive Psychology*, 25(4), 431–467.
- McFee, B., & Lanckriet, G. (2011). Learning multi-modal similarity. *The Journal of Machine Learning Research*, 12, 491–523.
- Montani, S. (2011). How to use contextual knowledge in medical case-based reasoning systems: a survey on very recent trends. *Artificial Intelligence in Medicine*, 51(2), 125–131. doi:10.1016/j.artmed.2010.09.004.
- Mouguouie, B., & Bergmann, R. (2002). Similarity assessment for generalized cases by optimization methods. In S. Craw, & A. Preece (Eds.), *Advances in case-based reasoning number 2416*. In *Lecture Notes in Computer Science* (pp. 249–263). Springer Berlin Heidelberg.
- Napoli, A., Lieber, J., & Curien, R. (1996). Classification-based problem-solving in case-based reasoning. In I. Smith, & B. Faltings (Eds.), *Advances in case-based reasoning number 1168*. In *Lecture Notes in Computer Science* (pp. 295–308). Springer Berlin Heidelberg.
- Nesme, M., & Hidalgo, P. (2013). PhiloSophie. definitions : ressemblance et analogie.
- Perner, P. (2003). Incremental learning of retrieval knowledge in a case-based reasoning system. In *Case-based reasoning research and development* (pp. 422–436). Springer.
- Peschl, M. F., & Riegler, A. (1999). Does representation need reality, 9–17.
- Reyes, E. R., Negny, S., Robles, G. C., & Le Lann, J. (2015). Improvement of online adaptation knowledge acquisition and reuse in case-based reasoning: application to process engineering design. *Engineering Applications of Artificial Intelligence*, 41, 1–16.
- Richard, A. (1979). *La résolution de problèmes*. Université Laval. Ph.D. thesis.
- Richter, M. M. (1993). *Classification and learning of similarity measures*. Springer.
- Rifqi, M. (2010). *Mesures de similarité, raisonnement et modélisation de l'utilisateur*.
- Sorlin, S., & Solnon, C. (2005). Reactive tabu search for measuring graph similarity. In *Graph-based representations in pattern recognition* (pp. 172–182). Springer.
- Wielemaker, J. (2014). Swi-prolog version 7 extensions. In *Workshop on implementation of constraint and logic programming systems and logic-based methods in programming environments 2014* (p. 109).
- Wu, Z., & Palmer, M. (1994). Verbs Semantics and Lexical Selection. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*. In *ACL '94* (pp. 133–138). Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/981732.981751.
- Xiong, N. (2011). Learning fuzzy rules for similarity assessment in case-based reasoning. *Expert Systems with Applications*, 38(9), 10780–10786. doi:10.1016/j.eswa.2011.01.151.
- Yang, C. J., & Chen, J. L. (2011). Accelerating preliminary eco-innovation design for products that integrates case-based reasoning and TRIZ method. *Journal of Cleaner Production*, 19(9), 998–1006. doi:10.1016/j.jclepro.2011.01.014.
- Zeid, I., M. Gupta, S., & Bardasz, T. (1997). A case-based reasoning approach to planning for disassembly. *Journal of Intelligent Manufacturing*, 8(2), 97–106.