OATAO is an open access repository that collects the work of 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/20626

Official URL: http://doi.org/10.1016/j.engappai.2015.01.015

To cite this version:

Roldan Reyes, Eduardo and Negny, Stéphane and Cortes Robles, G. and Le Lann, Jean-Marc *Improvement of online adaptation knowledge acquisition and reuse in case-based reasoning: Application to process engineering design.* (2015) Engineering Applications of Artificial Intelligence, 41. 1-16. ISSN 0952-1976

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

Improvement of online adaptation knowledge acquisition and reuse in case-based reasoning: Application to process engineering design

E. Roldan Reyes [a], S. Negny [a,*], G. Cortes Robles [b], J.M. Le Lann [a]

[a] *Université de Toulouse, INPT, UPS, CNRS Laboratoire de Génie Chimique, ENSIACET, 4 Allée Emile Monso, BP 44362 F-31030, Toulouse Cedex 04, France*
[b] *Instituto Tecnologico de Orizaba, Av Oriente No. 852, 94320 Orizaba, Veracruz, Mexico*

ABSTRACT

Despite various publications in the area during the last few years, the adaptation step is still a crucial phase for a relevant and reasonable Case Based Reasoning system. Furthermore, the online acquisition of the new adaptation knowledge is of particular interest as it enables the progressive improvement of the system while reducing the knowledge engineering effort without constraints for the expert. Therefore this paper presents a new interactive method for adaptation knowledge elicitation, acquisition and reuse, thanks to a modification of the traditional CBR cycle. Moreover to improve adaptation knowledge reuse, a test procedure is also implemented to help the user in the adaptation step and its diagnosis during adaptation failure. A study on the quality and usefulness of the new knowledge acquired is also driven.

As our Knowledge Based Systems (KBS) is more focused on preliminary design, and more particularly in the field of process engineering, we need to unify in the same method two types of knowledge: contextual and general. To realize this, this article proposes the integration of the Constraint Satisfaction Problem (based on general knowledge) approach into the Case Based Reasoning (based on contextual knowledge) process to improve the case representation and the adaptation of past experiences. To highlight its capability, the proposed approach is illustrated through a case study dedicated to the design of an industrial mixing device.

1. Introduction

Preliminary design in the industrial domain is a complex and decisive phase in the design process. In economic terms, Douglas (1998) has shown that the cost of this phase represents between 10% and 20% of the entire project cost but decisions taken during this stage impact 80–90% of the total cost. In process engineering (more particularly focused in this study) the total cost saving in industrial application ranges from 20% to 60% according to Harmsen (1999). Consequently, this design task has experienced significant improvement to new computer-aided design methods and tools whose contributions have led to rapid development taking into account quality, safety, operability, economic and environmental performances.

Among the new approaches to address this phase, Knowledge Based Systems (KBS) offer many possibilities and potentialities to support design decisions. Effective knowledge acquisition, reuse and valorization are increasingly important assets for firms in order to provide competitive advantages. Furthermore, KBS propose useful and original solutions without imposing limits to creativity as underlined by Cortes Robles et al. (2009) and Schimtt et al. (1997). KBS intend to rapidly integrate the new scientific knowledge coming from the fast pace of technological evolutions, and to provide users with knowledge access. Indeed, in industrial practices to reduce significantly the design time and cost, it is common to start an activity relying on a previously solved experience, and then to modify and adapt it to match the new requirements. Consequently, KBS are suitable for numerous industrial activities like preliminary design because it avoids starting from a scratch since some choices are neither to do nor to question. Thus the control of the knowledge is a necessity (i) to realize (design and manage process), (ii) to decide, (iii) to create new knowledge, (iv) to preserve the knowledge capital of an organization, and (v) to impel innovation. As the environment and the activities evolve rapidly, one of our main challenges is to propose a system that includes a phase to update the knowledge stored, but also to improve the confidence and quality of the knowledge monitored during activities. Nevertheless, the elaboration of KBS is still a difficult and extensive task, while approaches have been proposed to overcome these issues. Scientific expectations are mainly in knowledge representation, modeling, reuse and maintenance because they are tremendous knowledge engineering tasks.

However, the requirements evolve and improvements of current advanced KBS are mandatory to meet the current context

needs such as more agility and reactivity. Besides these needs, for design applications, KBS must also enhance their dynamic dimension to encourage rapid and flexible responses to some choices and to spread their impacts in the rest of the design process. This dynamic aspect in the adaptation phase of the KBS is a key milestone for knowledge reuse and continuous improvement of the performance of the system. Thus one goal of this study is to propose a KBS meeting these requirements of dynamic.

This paper is more focused on process engineering, which is the part of engineering that deals with processes that convert raw materials into more useful or/and valuable products through several transformations, under economic, environmental, safety, and energy constraints. A chemical process can be decomposed into individual sub-processes called unit operations: chemical reactors, separators, mixers, heat exchangers, etc. Due to the new industrial context, this discipline has undergone significant changes that strongly affect the design phase: the design and production of specialty products with high added value, introduction of numerous innovations on multi-functional units to ensure process intensification, and so on. As a result, a huge amount of new knowledge was and is still created. Optimization and heuristic approaches were the traditional methods to address the process design issues. For the former, we have a mathematical representation of the problem with the formulation of a multi-criteria objective function. However, the drawbacks of this approach are as follows: a huge computational effort, the difficulty to include uncertainties and ill-defined problem. The most important disadvantage is probably that the solution is closely dependent on the initial set of possible alternatives represented under the form of a superstructure. Consequently, it depends on the knowledge of the design team and not on the whole knowledge available. For the latter, process engineer has many heuristics for the traditional design problem, but for the new multi-functional units they are still to be created. Furthermore, as noticed by Li and Kraslawski (2004) their major limitations are their impossibility to manage the interactions between different design levels and the difficulty to handle multi-objective problems. This is due to the sequential nature of this approach.

Due to both the limitations of traditional methods and the mutation of the industrial context, there is a need to find new efficient approaches to capitalize the new implicit and explicit design knowledge. As a consequence, different KBS have emerged in process design based on methods such as Conflict Based Approaches and Case Based Reasoning (CBR). The first ones are based on modified TRIZ methods and tools to make them more easily applicable in the process engineering domain like in the studies of Li et al. (2003) and Negny et al. (2012). These approaches are more focused on the phase of the research of new concepts. CBR is also suitable because numerous design problems become recurrent and these experiences can be easily reused. Their applications to assist in design decisions have been studied and improved for process design in the last decades as demonstrated in Negny et al., (2010). But CBR suffers from three major drawbacks. The first two are knowledge elicitation and case adaptation. These drawbacks are commonly encountered in numerous CBR systems as proved by Chebel-Morello et al. (2013), who explained that the time of knowledge workers dedicated to these phases is, respectively, 37.7% and 45.9% of their total time. The third drawback is more specific to the application of CBR in design, where two categories of knowledge, i.e. contextual (corresponding to past experiences) and general (corresponding to rules, constraints, etc.), must be combined to support a wide range of design decisions on the one hand, and to improve the quality of the solution on the other. Unfortunately, CBR systems only aim to encompass contextual knowledge. Thus the challenge of this work is to raise the level of maturity of KBS for process engineering design; as a consequence, the objective of this work is twofold:

– From the process engineering design point of view, the aim is to improve the current CBR systems, which are mainly focused on the system to design (unit operation or the process) but not on design method but also to include the dynamic aspect. Moreover, the proposition concerns an approach that combines the two kinds of knowledge, previously cited.
– From the knowledge management point of view, the goal is to minimize the knowledge elicitation effort during the adaptation phase. Another important objective is to evaluate the quality and usefulness of the acquired adaptation knowledge in order to increase the skills of the CBR system.

Concerning the first point, among Artificial Intelligence (AI) approaches to capitalize and reuse knowledge Constraint Satisfaction Problem (CSP) has also been successfully applied in various activities and more particularly in design applications. CBR and CSP rely, respectively, on contextual and general knowledge. Due to this complementarity, this paper proposes coupling these two approaches, to address the adaptation problem in CBR. The main motivation is to achieve a synergy that produces a better knowledge exchange, capitalization and reuse.

For the dynamic aspect, several issues must be solved, with different approaches proposed in the literature. Karray et al. (2014) suggested using a trace based system whose goal is to extract new knowledge rules about transitions and activities in the maintenance process. Traces are considered as knowledge containers. This interesting approach is well suited for very dynamic and reactive system as in the maintenance field, but in the domain of design the time constants are lower. In another approach Craw (2009) transforms the traditional CBR into an agile one. In accordance with this work and with the work of Cordier et al. (2007), the traditional CBR cycle is modified to introduce an interactive process with the expert in the reuse step in order to create an online knowledge acquisition, but also to add agility to our KBS.

Concerning the second point to develop our adaptation method, we were interested in the different approaches proposed in the literature. Adaptation in CBR has been widely studied in the 1990's (Smyth and Keane, 1996; Pu and Parvis, 1995, 1997; Voss, 1997; Hanks and Weld, 1995; Craw et al., 2006), but no general models have emerged. Since then, this CBR step has received little attention as confirmed by the analysis of the research theme in the CBR literature realized by Greene et al. (2008). However, the recent evolutions on differential adaptation proposed by Fuchs et al. (2014) seem to give promising ways for an operational formalization of adaptation, while it is currently limited to numerical problems. The main idea is that small variations between problems are related to variations between solutions as in differential calculus. More generally, Chebel-Morello et al. (2013) have classified the main strategies to deal with the adaptation problem into three categories: (i) Adaptation Knowledge Acquisition that aims to obtain adaptation knowledge and to model them through general methods and techniques; Lieber et al. (2004) and Lieber (2007) provide a comparison and an overview on this strategy. (ii) Specific adaptation strategies depending on the application domain or on the case study. (iii) General adaptation methods independent of the application. For instance the method based on the dependency between problem and solution descriptors is the most advanced and used: Fuchs et al. (2000) for computer configuration, Chebel-Morello et al. (2013) for diagnostic. As one motivation of this paper is to improve the efficiency and accuracy of a CBR system for process engineering design, the adaptation method proposed is based on adaptation knowledge acquisition

strategy. Indeed, specific adaptation strategy could be deployed for each kind of unit operation (reactor, distillation column, heat exchanger, etc.) but we lose in generality and the creation of a specific method for each type of unit operation would be a tremendous effort. The method based on dependency aims to establish the direction and the strength of the relationships between problem and solution features. Unfortunately, in process engineering design the dependencies are impossible to establish immutably because they depend on many factors, e.g. the operating conditions, the occurring phenomenon that can be neglected at a scale and be overriding at another scale. Furthermore, the strengths of the links are difficult to establish due to the strong linearity of the phenomenon that occurs in a chemical process, their dependences on the operating conditions and on the chemical components in the mixture.

The remainder of this paper is organized as follows. In the next section, the backgrounds of CBR, adaptation knowledge acquisition and the knowledge elicitation issues are described. In Section 3 we discuss the interest for combining CBR and CSP and propose a framework to capture expert knowledge. Section 4 describes the methodology and highlights the tool capabilities through a case study dealing with the configuration of an industrial chemical mixer. Before drawing conclusions, in Section 5 some tests are realized to quantify the reusability of the adaptation knowledge acquired.

2. Backgrounds

2.1. Case based reasoning

The main feature of CBR is its ability to emulate human reasoning for solving new problems by remembering past experiences. The general principle according to Schank (1994) is that similar problems have similar solutions. In CBR, past experiences are stored as cases; each one encloses the description of a problem (source problem) and its associated solution (source solution). A new problem, namely the target problem, can be solved by retrieving the most similar cases and relying on the source solutions. Various models have been developed in order to provide a systematic way to perform CBR. The R^5 model, illustrated in Fig. 1, which is an expanded version of the model of Aamod and

Fig. 1. CBR cycle.

Plaza (1994), is now commonly implemented in many practical CBR systems.

- During Representation, the problem is described with a predefined framework. Depending on CBR goals, problems and solutions can be represented according to two major approaches: textual (e.g. simple binary or text files) or structural (i.e. based on attributes and predefined values).
- Retrieval is the process of matching and selecting from the case memory, one or more cases that can be reused to solve the target problem. Here, the main issue is the similarity measurement; a recent overview in similarity measures is given in Qi et al. (2011). Similarity estimation often relies on a mathematical distance between problems, inferring that retrieval distance is proportional to the adaptation effort. But several authors like Massi et al. (2007) or Smyth and Keane (1998) argue that the most similar case is not necessarily the easier to adapt or the most relevant to solve the target problem. Indeed, retrieval based on similarity can generate an incapacity for CBR to solve a problem or worse in some cases to propose an inadequate solution. Consequently, they introduce a new criterion, i.e. case adaptability, to evaluate the adaptation effort in order to improve the retrieval performance.
- In most practical approaches, the Reuse step is quite simple: the source solution without any modification is proposed. But most of the time the retrieved solutions must be adjusted to withdraw the discrepancies between the target and source problems and to fit the target problem requirements as illustrated by Maher and Pu (1997). This leads to one of the most important, problematic and challenging subjects in CBR: adaptation. Many authors like Leake et al. (1996), Cordier et al. (2007), and Smyth and Keane (1996) have underlined that adaptation adds the intelligence to what would be simple patterns or tendencies.
- In the Revise step, the proposed solution must be tested to validate its adequacy and relevance with respect to the target problem, or to consider what actions are to be taken to withdraw the remaining discrepancies.
- Once a satisfactory solution is reached, the current problem and its solution form a new case that can be retained in the case base, only if it brings a real added value to the CBR system. This new learned case increases the CBR system's effectiveness by enlarging its coverage of the problem and solution spaces.

The five steps detailed above represent the essential components to build up a CBR system, even if it has other important issues whose significance is crucial; case acquisition process, case base structuration and indexation.

CBR provides a set of particular advantages concerning the design activity: reducing the knowledge acquisition task, its ability to support long-term learning, its capacity for reasoning with incomplete or imprecise data, its vicinity with human reasoning and its ability to create and to maintain a computer decision support tool. Nevertheless, it is necessary to provide an important number of cases to have significant results. Unfortunately, this is rare in preliminary design. To overcome this drawback, the CBR system must contain an efficient adaptation module.

2.2. Adaptation knowledge acquisition

Despite that the knowledge stored in source cases gathers a huge part of the problem solving expertise, the adaptation knowledge acquisition to achieve the solution refinement can be demanding. It consists in modeling and storing an adaptation process performed by an expert in order to capitalize this new knowledge. Adaptation patterns are stored in the case base to exploit them in future adaptation episodes. Few studies on adaptation knowledge acquisition have been conducted, but after
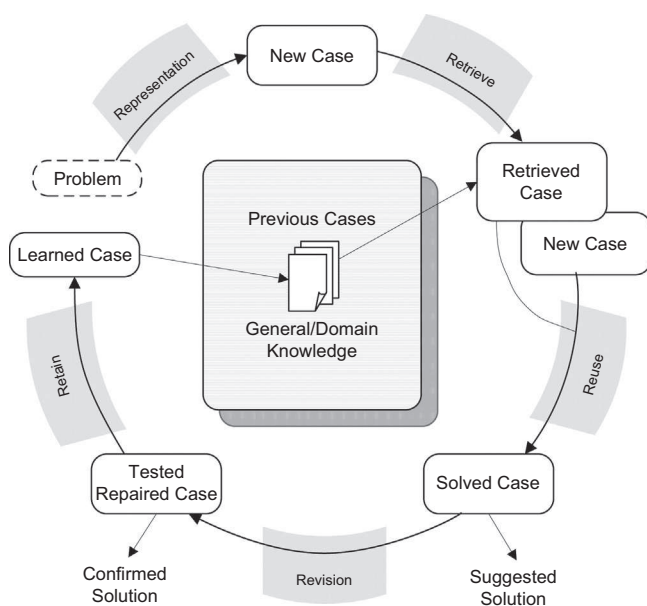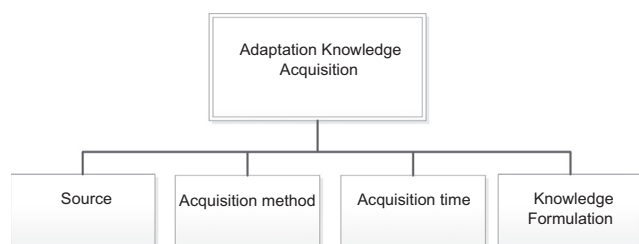
Fig. 2. Taxonomy of the different characteristics for adaptation knowledge acquisition.

reviewing the literature on the subject we can identify four principal characteristics, as illustrated in Fig. 2.

For knowledge extraction, two types of sources are identified: external or internal to the CBR system. Internal adaptation knowledge can be extracted from the differences between the cases stored in the case memory. Here the main assumption is that the case base is considered as a representative sample of differences between problems that could be encountered during adaptation. For instance, Craw et al. (2006) have proposed an introspective learning approach where cases provide a source from which representative adaptation knowledge can be extracted. This approach is easy to operate and implement, but it does not allow inferring explicable knowledge. Furthermore, it remains the question of confidence we can have to the knowledge extracted. Among the external sources the most obvious is the expert, who can be solicited to formulate the new necessary knowledge. Recently, the World Wide Web has emerged as a new category of external sources which offers access to various knowledge bases, for example the WebAdapt system presented by Leake and Powell (2007). There is also a research activity based on the semantic web languages as in the work of Fidjeland (2006).

The methods for knowledge acquisition may differ depending on the acquisition mode: manual, automatic or semi-automatic. In the manual mode, experts are interviewed during specific adaptation knowledge acquisition sessions to explain how they solve problems. This way to proceed needs a significant work in knowledge engineering, leading to very complex and time-consuming tasks. In return, the extracted knowledge is accurate with a high degree of confidence. The automatic alternatives consist in extracting adaptation knowledge from data sets. For instance, machine-learning or data mining techniques are often used to produce heuristics or to develop automatic processes. While they are easy to operate, the knowledge generated is difficult to understand, to exploit and to reuse due to its intrinsic quality. Furthermore a memory gathering many cases is required in order to avoid inaccuracies and approximations on heuristics. But as explained before, in preliminary design, it is unusual to have such a vast case base. Semi-automatic approaches combine both previous ones, i.e. some knowledge elements are generated automatically and then they are submitted to an expert for validation.

The adaptation knowledge can be acquired online or offline. Offline acquisition can be time consuming. Online acquisition takes advantage of an adaptation episode to solicit punctually the expert. This approach is motivated by the goal to reduce the effort of knowledge engineering. Nevertheless, the major drawback is the number of iterations that the adaptation can necessitate. Badra et al. (2009) graphically sum up these characteristics in Fig. 3. As a consequence, it would be more effective to have an online sub-process to check the solution and to acquire expert knowledge in particular when it is necessary to repair the solution or when an adaptation failure occurs. After corrections, the expert knowledge is updated and added to the adaptation knowledge container.

Despite various formalisms such as constraints, adaptation cases with recursive CBR like in Jarmulak et al. (2001), rules are commonly

used for knowledge formulation. Unfortunately, they are not appropriated for process engineering design even with the current improvements, for instance by adding a measure of confidence D'Acquin et al. (2004). Indeed, rule-based systems have shown some limitations, such as their difficulty to manage and to update rules, their ineffectiveness due to the complexity and non-linearity of the phenomenon that occur in chemical processes.

Regarding the scope of application and goals of our CBR system, we decide to acquire adaptation knowledge with an expert in order to have great confidence in the knowledge; we also use an interactive approach (online and manual acquisition) to reduce knowledge engineering effort. Consequently, to elaborate our CBR system, we must address the adaptation knowledge elicitation issue.

2.3. Knowledge elicitation

In AI, elicitation allows one to formulate the expert reasoning in an inference engine, thus giving the possibility to artificially reproduce the situation analysis and the decision making. In knowledge management (KM), the goal of elicitation is to help the expert formalize his knowledge in order to save and share it. Here elicitation aims to transform tacit knowledge in knowledge as explicit as possible and therefore easier to transmit. Elicitation is often essential to organize and ensure the sharing of knowledge. In our approach, we try to couple both visions of elicitation: capitalization of the expert reasoning as in AI but also facilitating adaptation knowledge formulation. Concerning knowledge formalism, the adaptation operators are retained because they allow one to decompose the adaptation knowledge into individual containers, to make them more easily reusable, but also to facilitate knowledge maintenance.

In the proposed approach, the whole additional knowledge needed is captured in the form of an adaptation method, which encloses all the changes that affect the source solution. As in the CBR paradigm the knowledge included in the adaptation stage is less expensive than the knowledge required to build a solution from scratch; let's assume that changes can be made by a small finite number of successive elementary steps. As proposed by Cordier et al. (2007) each elementary step corresponds to a single adaptation operation traduced by an Adaptation Operator (AO). These operators symbolize the actions that the expert carries out on the source solution to obtain a satisfactory solution. An adaption method i (denoted AM^i) is composed of a finite set of m^i successive AO.

$$AM^i = \left\{ AO_j^i \right\} \text{ with } j \in \left\{ 1, \ldots, m^i \right\} \; m^i \text{can be variable from}$$
one method to another.

The decomposition into a finite list of successive AO allows having an accurate and sharp description of the modifications. Furthermore after each AO is created, the expert has the possibility to add a comment to explain the interest of this operator and thus to improve confidence in the adaptation knowledge capitalized. The combination of available information in a case, the AO and the expert's comments produce a knowledge episode, which is often easier to analyze, more credible and consequently easier to reuse. Besides, for each single AO it will be possible to distinguish if it is rather to include in the framework of the general knowledge domain or rather in the framework of the specific modifications to the problem studied. This distinction enables one to facilitate the knowledge maintenance.

As explained before, the manual knowledge acquisition is retained because we do not have enough data to automatize the acquisition process. On the one hand, manual acquisition produces more confidence in the knowledge, but on the other hand, it requires tremendous effort. This drawback can be partially removed by taking advantage of an adaptation episode to acquire knowledge online by requesting punctually the expert. Unfortunately, the traditional CBR
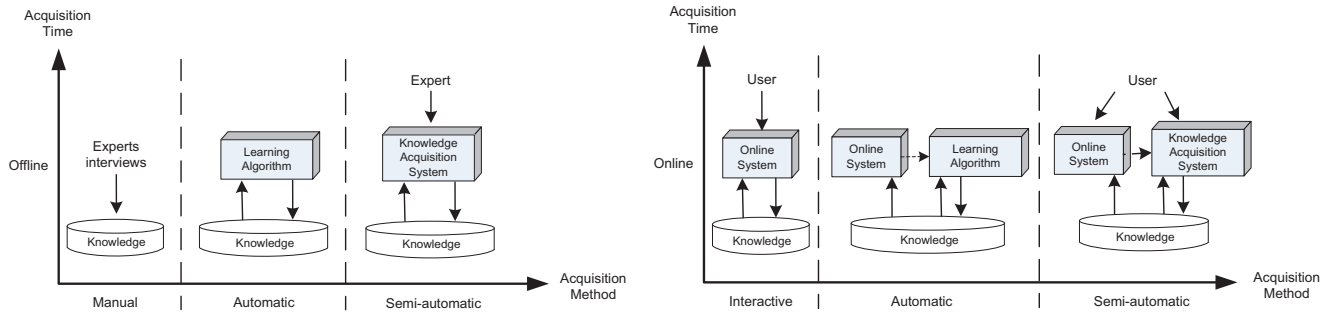
Fig. 3. Acquisition times for the adaptation knowledge (Badra et al., 2009).

Table 1
CBR and CSP feature's comparison.

CBR	CSP
Operational method.	Heuristic method.
Use of tacit and contextualized knowledge.	Use of explicit, general and formalized knowledge.
Analogical reasoning in a domain with small quantity of knowledge.	Constraint-based reasoning, relying on a deep knowledge on the activity.
Possibility of reaching solutions to complex problems even when the application domain is not well known by the user.	Possibility of reaching all solutions to complex problems. The user must understand the problem situation.
Ability to produce solution rapidly.	No theoretical warranty on finding solutions.
Thanks to its memory, CBR can offer solutions without the need to start from scratch.	Because of the lack of memory, it requires the user to make a detailed analysis of the CSP model in case of failure.
Flexibility in knowledge modeling. The information contained in the cases is not necessarily conditioned by a particular formalism.	Richness it offers in terms of modeling because constraint expressions are not limited to mathematical relationships.
The limitation to solve similar problems and therefore no guarantee to find optimal solutions.	It can find the optimum by introducing an objective function.
The effectiveness of such a system is highly related to the coverage of the problem and solutions spaces.	The effectiveness lies in the proper modeling and the performance of the algorithms for resolution.
It will always find a source case more or less similar to the target problem and therefore such a system is unable to establish when a problem has no solution.	The possibility of controlling the number of solutions. It can also establish that a problem does not have solution.
User interaction is possible if an appropriate adaptation strategy is chosen.	No interaction during the problem solving.

cycle does not provide sufficient interactivity for this online acquisition. In practice, Cordier et al. (2007) have proposed modifying the traditional CBR cycle to introduce a new interaction loop at the adaptation step to create this interactivity. This interaction loop is twofold: to facilitate adaptation knowledge acquisition and to quickly visualize the consequences of each decision.

For our approach, the general framework proposed by Cordier et al. (2007) has several important drawbacks: (i) it uses an adaptation method based on dependencies and differential adaptation which is not relevant for us as explained before, (ii) all the adaptation process is automatized, in their prototype there is a virtual expert who can not only automatically detect the adaptation errors but also correct them with the good value, (iii) their method and AO are limited to numerical values for features, this is too restrictive especially when we want to acquire knowledge related to the design method, (iv) there is no decision support system to choose an appropriate adaptation method when several relevant methods are proposed to adapt the source case, (v) there is nothing on the usefulness and the quality of the acquired knowledge, and (vi) there is no link with the next CBR step, i.e. maintenance. Consequently relying on this previous work, the next step for our approach is to propose a new formalism for the AO but also a new way to manage the new acquired knowledge. The CSP is used in the CBR to reach the different objectives of our approach.

3. CBR-CSP coupling to support design

3.1. Constraint satisfaction problem

In CSP, the knowledge is explicitly expressed through a mathematical model composed of variables, definition domains and constraints such as logical relations, mathematical expressions or domains of validity. Constraints express the authorized and/or forbidden combinations of values for the variables. The CSP approach offers a natural way for representing problems. The new problem is submitted to the knowledge model via the variables, and then a reasoning process is led through constraints to progressively restrict the domains by retaining only consistent values. The designer interacts with the constitutive elements of the model to add his successive and progressive decisions which are propagated through the constraints. This process is repeated until one (or several) solution that respects all the constraints is reached. The two main mathematical solving techniques are filtering and search. CSP provides many advanced algorithms with a limited computational cost to deal with highly combinatorial problems. The major disadvantage of this approach is that it requires a huge effort to identify, extract, interpret and formalize knowledge and to build the reasoning model. This implies a sharp and deep understanding on the activity. However, the systems managing constraints have the advantage of quickly providing original solutions, to establish when a problem does not have one or to find all the possible solutions.

3.2. The reasons of the coupling

At first sight the two approaches seem contradictory because CBR assumes that there is not enough explicit knowledge and therefore past experiences are used, whereas CSP requires a full understanding of the concrete domain. The first step of the proposal is to drive a deep analysis of these two approaches to establish some potential cooperations. A detailed analysis of both reasoning paradigms is given in Table 1.

Based on this comparison, the integration of the CSP reasoning process in the CBR offers several benefits such as: (1) to improve and make more accurate case representation, (2) to develop a more systematic and efficient retrieval mechanism, (3) to provide

a way to adapt solutions, (4) to propose a strategy to reduce problem complexity thanks to constraints propagation, and (5) to efficiently manage design preferences (Sqalli et al., 1999; Ho et al., 2010; Galushka and Patterson, 2006). Similarly, the CBR offers several advantages to the CSP solving process, among them the most important are: (1) the ability to propose CSP models without building them from scratch, (2) to complete models when it remains fuzziness or incompleteness on the domain knowledge, (3) to reuse an earlier experience to improve the resolution process, and maybe the most important, (4) to add a learning method and a memory to the CSP process (Sqalli et al., 1999; Roldan et al., 2011; Wang and Liao, 1997).

3.3. Related works integrating CBR and CSP

In the literature, several authors present a synergy between CBR and CSP; for instance the CBR system JULIA proposed by Hinrichs (1992) was one of the first to incorporate the CSP approach in CBR for kitchen recipes. In the work of Vidotto et al. (2007), CSP was used as a tool to analyze dynamic combinatorial problems in restaurant management. A similar integration is described in the CADRE system of Hua et al. (1996) in the field of architectural design, where the CSP approach and the rules of production of topological knowledge are integrated to the CBR. The CBR system of Pu and Parvis (1995, 1997), which formalizes the adaptation process with constraints, was applied to the configuration domain. The same approach was used by Ruet and Geneste (2002) in order to assist the design of plant operations, where the CSP technique is used to guide the adaptation phase of the retrieved solution. Several other approaches integrating CBR and CSP were developed, pointing out the effectiveness of this coupling; a detailed state of the art was carried out by Sqalli et al. (1999). In their work, the coupling is applied to compensate incompleteness and incorrectness of CSP models in network control protocols for diagnoses interoperability. More recently, Inakoshi et al. (2011) have proposed a framework for product configuration using CBR to generate criterion by estimating the user preferences on products. Then the CSP module classifies the resulting configurations by strictly preserving the user requests and definitions. Lopez (2003) also use the CSP formalism to represent cases in CBR to reach a solution for scheduling problems. The adaptation is led thanks to the CSP techniques. In this approach, only adapted cases that have led to a satisfactory solution are stored in the base. This reduces the effort during the base maintenance and keeps it within a reasonable size. Neagu (2005) proposes a generic platform for the case adaptation and presents new algorithms for an interchangeability of the CSP parts in order to facilitate the adaptation in CBR. He validates his approach with planning, scheduling and configuration problems. In the same way, Medjdoub (2009) couples the two approaches to deal with the adaptation problem in the architectural design domain. The system substitutes the parts of a retrieved solution which do not correspond to the new requirements, and starting from the incomplete solution, the system identifies the inconsistent parts and solves them separately and successively through the CSP algorithm, reducing considerably the complexity of the problem. Wang et al. (2009) proposed an algorithm that integrates the two approaches applied to "online" product configuration: the user inputs his request then the CBR recovers a similar case, which is modified according to conditional constraints. Recently Vareilles et al. (2012) have summed up these previous couplings into four possibilities:

- To validate the knowledge stored either in one or in the other. Cases are used to validate or invalidate the constraints of a CSP. Conversely, a CSP can be used to qualify the consistency of a CBR case with respect to its model of constraints.

- To modify or create knowledge in the CSP. The analysis of a case of CBR can add constraints to the CSP. Similarly, the CSP can be used to complete CBR cases when some features are missing.
- To mix them sequentially. The CBR allows extracting the most similar cases and then CSP to perform adaptation. In the other possible sequence, the CSP permits one to limit the possible values for the features while maintaining the consistency and then CBR allows one to retrieve the most similar case in order to adapt it.
- To combine general and contextual knowledge. In their proposal they combine the two approaches in a simultaneous and iterative manner according to the availability of knowledge by taking into account contextual constraints. The general knowledge prunes the solution space and contextual knowledge gives more accurate advices to the user. They applied their approach in the field of maintenance.

3.4. Discussion

According to Vareilles et al. (2012), only the last two possibilities really deal with knowledge processing. The other ones were more focused on knowledge validation or completeness, which is out of the scope of our study. Like in Lopez (2003), in our proposal, CBR is used to find the appropriate CSP model which is then adapted. Purvis (1998) has demonstrated that CSP techniques help formalize and treat the process of adaptation but also to systematize and give flexibility to the CBR. This way to proceed is particularly suitable when we have domain-specific CSP models. However, with the retained combination of both approaches, there are some important drawbacks:

- In the current approaches, it is always assumed that the adaptation is a human process and not a processed one. But to better assist designers, this phase must rely on knowledge processing.
- The knowledge enrichment, thanks to the adaptation phase, is not correctly exploited. Indeed, after the model modifications, we must be able to distinguish between specific knowledge only valid for the problem studied and the general knowledge. Of course the former must be stored for a future reuse and to improve the CBR system skills. As they are specific, they are not candidate to be included in the original CSP model. Inversely, the general knowledge could be included in the model to improve it.
- Successful adaptations are stored as new case in the base, leading to a case base size that increases sharply. Even if it is mandatory to store the new modifications, it would be more valuable to find an alternative that requires a lower storage space while allowing easy maintenance.
- More interaction is needed to add agility. During adaptation, the system should offer to the designer the opportunity to see and evaluate the consequences of his choices and more particularly the progressive reduction of the solutions space.
- The expert is the only person who interacts with the knowledge model for knowledge maintenance and evolution (CSP model evolution). But, most of the time, it is a time consuming human process, which needs to be simplified and partially automated. Consequently, the proposed approach must take into account the requirements on knowledge acquisition and storing while remaining compatible with the next step of the CBR cycle.

For these reasons the integration of another knowledge acquisition paradigm detailed in Section 4 is proposed. It would allow improving the interaction with the expert and the confidence in the knowledge stored in the system.
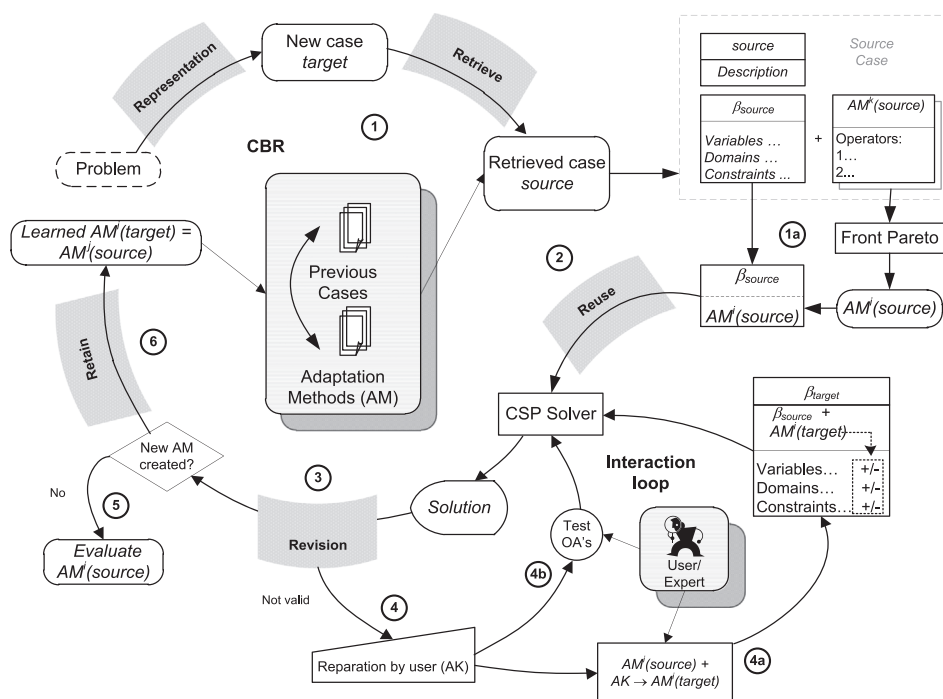
Fig. 4. CBR adaptation process.

4. Methodology

In this section, the whole methodology is presented. It gathers all the elements detailed and discussed in the two previous sections: case representation, the AM, the AO, the created inter-active loop, and the test procedure for knowledge reuse. It also encompasses all the proposals and new contributions to provide an answer to the drawbacks identified previously. Fig. 4 illustrates the adaptation process integrated in the new CBR cycle. The following sections describe and exemplify the various steps.

4.1. Case representation

4.1.1. Problem and solution descriptions

As it is broadly accepted, the information targeted in a case can be decomposed into three parts: the problem description, its associated solution, and some explanations and justifications. The new problem is described with an attribute-value representa-tion where some feature values are used as input data for the CSP solving. In our system, the solution part is composed of a CSP model. First this choice is in accordance with our KBS objective to easily update or add new knowledge to ensure viability and sustainability of the system. Another argument is that we prefer to focus the capitalization on the design method, rather than on the solution, because the method is often richer in knowledge, more generic and more easily transposable. Furthermore, for some systems, it is also more advantageous because technology can evolve rapidly and become obsolete. Finally the CSP model was also chosen due to its ability to reach a coherent solution, even when the problem statement is incomplete and complex, which is often the case in preliminary design.

4.1.2. Example of CSP

The case study is focused on an industrial mixer which is an important unit of operation for creating favorable hydrodynamic conditions for heat and mass transfer. It is required at many stages of a chemical process: from the storage of raw materials, through the reaction process, or to put phases in contact. Depending on the mixer configuration several hydro mechanical phenomenon can occur: homogenization, heat and/or mass transfer, suspension, dispersion, emulsifications, etc. These phenomena are used in chemical or physical operations like chemical reaction, extraction, absorption, desorption, dissolution, crystallization, etc. A bad technological mixing configuration or a wrong sizing can lead to erroneous features or outputs with disastrous consequences for the whole chemical process. Mixing is used in numerous industrial domains, for instance: chemistry, pharmaceutical, food industries, and fine chemistry. Among all the various possibilities for a mixing system, this example is limited to mechanical systems by rotation because most of the stirring processes use these technologies.

The goal of the CSP is to find the different configurations for a mixer according to desired operating conditions. As a conse-quence, the user must specify some input data before to run the model. They are related to the characterization of the phases, physical data, the type of operation, the hydrodynamic character-istics aimed, and if there is or not a thermal exchange. It is important to notice that in preliminary design, information is often poorly and ill-defined but only an order of magnitude or even qualitative information can be specified. As a consequence, problem features filling is flexible enough to accept one specific value, an interval of possible values or qualitative information (High, Medium, Low). In Annex 1, all the data required for the problem description are detailed.

The mechanical agitation device is assumed to be composed of three parts: the agitation system, the impeller and finally the vessel. As the technological choices for such a device are numer-ous, they require a detailed description. The agitation shaft is described with two variables: the power of the agitation engine and the type of agitation shaft. For the impellers, designers must choose among a wide variety (one variable) sorted by the generated liquid flux (axial, radial or tangential flow), they must also be precise on the position of this impeller (two variables). To describe the geometry of the vessel, five variables are mandatory:

geometry of the vessel, the presence (the size) and the position of the baffles, and finally the presence (the type) and the size of the heat exchange system. Each variable is associated with a definition domain, Annex 2.

All the constraints cannot be presented but they can be classified into three categories:

– conditional constraints: a constraint may or may not be present in the model, depending on the operating conditions. For example for an adiabatic operation the variables and constraints dealing with the heat exchange system are automatically eliminated.
– continuous constraint: implicit formulation with an analytic formula
  For example the constraints on the power of the engine or the turbine (depending on the type of impeller).

$$P_{engine} \geq 2P_{turbine}$$

  where $P_{turbine} = \rho * \omega^3 * f(pos) * (D_t)^5$,   (1)

  ρ is the=density of the mixture, ω the=rotation speed, *pos* the turbine position, and D_t the=turbine diameter.
- Discrete constraints: explicit enumeration of the possible combinations expressed in intention. For example for the selection of an impeller several conditions have to be considered: for homogenizing miscible liquids the impeller just acts as a generator of movement, to ensure mass transfer between phases the impeller acts as a promoter element, when required it can improve drops formation of the dispersed phase in the continuous phase, to put solid in suspension in a liquid the role of the impeller is twofold: lifting the particles from the bottom of the tank and keeping them in suspension. This non-exhaustive list demonstrates the importance for the choice of the appropriated impeller.

4.1.3. Adaptation operators and methods

Every time, a retrieved CSP model is adapted, all the modifications are stored as one adaptation method. The adaptation method (AM) becomes the means through which expert knowledge is acquired and may be capitalized. As an AM comes from a root CSP model, the third part of the case representation encompasses the list of AM linked to the retrieved case. The storage of AM instead of a complete case permits one to keep the case base in a reasonable size but also to eventually store failure.

As each model is formulated as a CSP, the possible actions are centered on the three constitutive elements of the model: variables, domains and constraints. Consequently, the AO corresponds to the following possible actions:

● Add: a new variable, a new constraint or a new value in a domain.
● Change: the domain of a variable, the formulation of a constraint, one value in a domain.
● Delete: a variable, a constraint, a value in a domain.

These modifications generate different levels of difficulty. Some of them are very obvious like adding a value in a domain, but others are more complex, e.g. add a new variable. Indeed to keep the model coherence, a definition domain and constraints must be added or modified to include the new variable. Some tests are implemented in order to ensure this coherence. The retained formalism allows one to easily modify these operators and to introduce problem specificities such as expert requirements, specifications, etc. Thus, for each AO, it is possible to distinguish if the modification relies on domain specific design knowledge or a
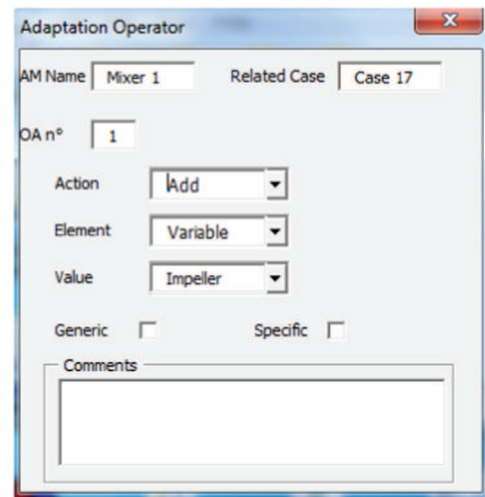

Fig. 5. AO description.

requirement specific to the problem faced. Fig. 5 illustrates how AO are defined.

Moreover to preserve model reusability, some metrics are used to qualify the AM (detailed in the next section). The AM that does not meet a minimum threshold on these metrics will not be taken into account in the maintenance step.

4.2. Retrieval and reuse

4.2.1. Choice of an adaptation method

Once the target problem is described, the CBR system retrieves a similar source case according to the user inputs and requirements. This source case encloses a CSP model called *βsource* and the adaptation methods linked to the CSP model, named *AM (source)* (point 1 in Fig. 4). One of our proposals is to improve adaptation knowledge reusability by measuring the performance of the AM to support the user choice. Indeed, each *AM(source)* performance is evaluated through four indicators: two were inspired from the approach of Vernat (2004) and the last two were specifically created for the evaluation of design models:

1- Parsimony: is the ability to obtain a solution with a minimum number of changes in the model;
2- Accuracy: defines the number of possible solutions calculated by this model;
3- Rate of comments: all the AO belonging to the framework of the general knowledge must be commented to explain the reason of this model modification. Even if they are not mandatory, comments on AO dedicated to specific modification improve this criterion but to a lesser extent. The comments improve AO reusability and knowledge quality;
4- Cyclomatic complexity: indicates the complexity of a program by measuring the number of linearly independent paths, computed using the control flow graph of the program.

The goal of these criteria is to qualify the modifications, and thus to limit the intake of adaptation knowledge to what is strictly necessary for its improvement. These indicators lead to reflect the nature, relevance and interest of the adaptation knowledge introduced and thus to improve its quality and preserve its reusability. With these indicators, we not only propose a decision support for the user but also we try to anticipate the step of maintenance (not implemented yet) of the adapted knowledge. Indeed, *AM(source),* which do not reach a minimum threshold on the metrics, are primarily analyzed to understand the root cause of the problem for either removing them
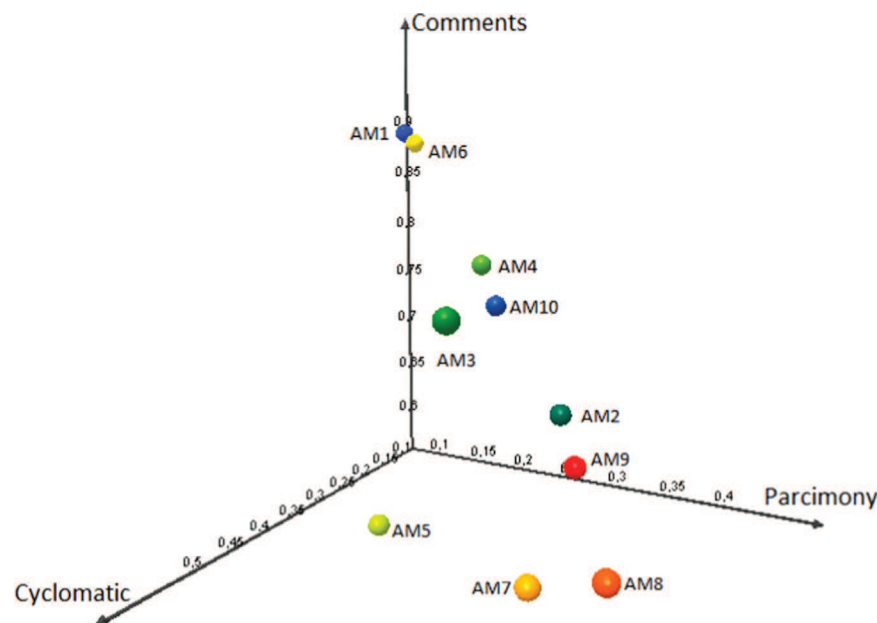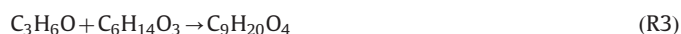
Fig. 6. Selection of the AM method.

from the base, or to improve their formulation to make knowledge more exploitable (when this adaptation knowledge is mandatory but in the current form it is unusable). Since these criteria can be difficult to prioritize, the Pareto front technique is envisaged to assist the user in the selection of an AM, by offering the best compromise between them (step 1a in Fig. 4). After AM selection, the *βsource+AM(source)* model is solved, step 2 in Fig. 4. The calculated solution is displayed to the user for validation. If it is satisfactory the process continues with step 5, only if the *AM* has not undergone changes otherwise with step 6. However if the solution is not satisfactory then the adaptation loop is activated.

4.2.2. Case study: problem description, retrieved solution and AM selection

The case study will serve as a recurring example to illustrate each phase of the methodology. The target problem concerns the production of propylene glycol in a chemical reactor. Propylene glycol is used as chemical feedstock for the production of unsaturated polyester resins and also as a humectant, as a solvent, as a preservative in food or in tobacco products and introduced in personal care products. In our case study, propylene glycol is produced as a solvent for pharmaceuticals, including oral, injectable and topical formulations. Industrially, propylene glycol is produced by the hydrolysis of the propylene oxide (R1). Unfortunately, there are two other secondary chemical reactions (R2 and R3). These reactions are highly exothermic. Furthermore, the temperature inside the reactor must be lower than 45 °C for security reason. Moreover, the greater the temperature is, the more the secondary products are produced. Besides to meet design requirements, the generic model requires some input parameters from either the needs of the designer, the functional requirements specification, or preliminary calculations. In the case study, the initial parameters to be provided by the user are: the phases to mix and their properties (e.g. viscosity, density, solubility, pressure, etc.), type of operation (dispersion), the desired hydrodynamic characteristics (turbulence) and in this case study the presence of a thermal system due to the high thermal flux to transfer at the reactor wall to respect the security constraint.

$$C_3H_6O + H_2O \rightarrow C_3H_8O_2 \tag{R1}$$

$$C_3H_6O + C_3H_8O_2 \rightarrow C_6H_{14}O_3 \tag{R2}$$

$$C_3H_6O + C_6H_{14}O_3 \rightarrow C_9H_{20}O_4 \tag{R3}$$

The retrieved *βsource* deals with a generic model for mixer configuration, for which the formulation is detailed in Section 4.1.2. This case contains also ten different adaptation methods. As the goal is to find a relevant mixer configuration, in a first step we may find the widest set of possible configurations and then progressively reduce it. In these conditions the choice is based on the three indicators: parsimony, rate of comments, and cyclomatic complexity. In the Pareto front provided by the decision system, the upper portion of the curve, on Fig. 6, suggests that a good AM may exist in this zone for the three indicators. The two remaining AM1 and AM6 are very close; thus, it is difficult to decide. We can notice that AM 3, 5 and 7 are out of the front and the other five are dominated. The two retained AM are edited and the AO are analyzed in details to make the final choice. The discarded method has few generic constraints and too many specific constraints to the case previously solved such as: constraint to impose the type of vessel and constraints on very specific flow conditions. The retained adaptation method, i.e. AM 1, is composed of the seven following successive adaptation operators:

> AO 1 -Generic- Add a value in the domain TypeM (mobile type).
> AO 2 -Generic-Add a type of stirring shaft (motor).
> AO 3 -Generic- Add a constraint on position 1: focus on the high vertical position as the most used.
> AO 4 -Specific- Impose a flat bottom and a cylindrical vessel. For this constraint, the designer had imposed to reuse equipment already present in the workshop.
> AO 5 -Generic- Add a constraint on the power of the motor. Required constraint, because of the balance between the mixture characteristics and the energy cost of the operation.
> AO 6 -Specific- Add a constraint for non-selection of the type screw for mobile agitation. This is often a very specific type for some categories of fluids.
> AO 7 -Specific- Remove the "Off-center" and "Tilted" values in the domain of position horizontal position. Even if they provide hydrodynamic conditions to prevent vortex, these two positions of the impeller increase dramatically the power of the motor and increase the mechanical stress on the driving shaft.

After the resolution of the CSP model corrected with the retained adaptation methods, among the ten mixer configurations found, none satisfies all the problem requirements. Consequently, the adaptation must be refined through the activation of the adaptation loop.

4.3. Adaptation loop

4.3.1. Loop description

The difficulty with "on line" knowledge acquisition is establishing the least restrictive interactions with users while trying to get enough information to learn knowledge. Moreover, the diagnosis of the reasons of adaptation failure must be made. In the approaches proposed in the literature, the only source of adaptation failure considered comes from none adapted AO. But it is too restrictive; we must also include the possibility that the adaptation knowledge is missing in the knowledge base. Furthermore, in our approach the expert is real and not virtual. With a virtual expert, the system can easily diagnose and correct the data or the reasons of the error, but this is not always the case with human expert. Consequently, a more sophisticated user interaction would be necessary to diagnose the causes of failure and to determine the necessary repairs.

A process is triggered to correct the unsatisfactory solutions with two possibilities:

1- The user modifies himself the model by changing or creating AO, sub-process 4a in Fig. 4. The new set of AO becomes a new AM. Note that these changes complement the *AM(source)*. Indeed, the new AM is generated on the basis of the *AM(source)* used in step 2 or from a previous adaptation cycle. Then, the user returns to step 3, to validate the new model. The loop is activated as many times as necessary until a solution is reached.

2- The user chooses to check each AO from the *AM(source),* sub-process 4b on Fig. 4. The AO test procedure is activated to help him establish the reasons of the failure. Indeed, it is not always obvious to identify the AO that impede a suitable solution. This procedure tests successively and separately the various AO, in order to identify and correct the faulty operators with respect to the problem requirements, Fig. 7.

If after testing and correcting the faulty AO, the solution remains not suitable, the diagnosis then turns to a lack of adaptation knowledge in the base. To refine the diagnosis and to try to identify the missing knowledge, the user indicates the variables that do not meet its requirements, then all the constitutive elements of the final model (definition domain, AO, constraints) that involve these variables are extracted and submitted to user analysis. At this stage, there are two possibilities:

- A solution is found. The necessary adjustments are done with expert knowledge or by trial and error methods if the expert cannot precisely identify the reasons of the failure (or does not have the necessary knowledge do remove it).
- The problem remains unsolved. The failure of adaptation could be stored in specific base gathering all the adaptation failures with the first development of diagnosis.

Test procedure:

1- Choose an AO.
2- Construct the corrected CSP model: retrieved model coupled with the set of AO already tested and preserved.
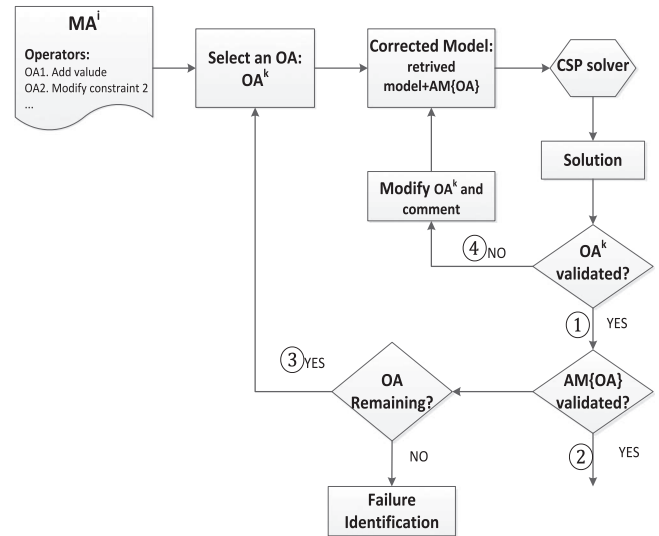3- Resolution of the corrected model.

Fig. 7. Test procedure.

4- The intermediate solution is submitted to the expert for validation:
   - 4-1 The intermediate solution is validated ①, besides it meets the problem requirements, then we can go to the retain step of the CBR cycle ②.
   - 4-2 The intermediate solution is validated ①, but it does meet the problem requirements, then go to step 1③ to choose another AO and to update the AM.
   - 4-3 The intermediate solution is not validated ④, modification of the tested operator until a solution is reached (a comment can be added to explain the correction). If all the AO had been tested, we must research the reason of the failure thanks to the variables and elements extraction.

Besides interactivity and knowledge engineering effort reduction, the proposed adaptation loop has various new advantages: (i) to identify the reasons of the failure, (ii) to evaluate the consequences of some modifications on the remaining design choices, and (iii) to propose a global resolution that encompasses all the modifications rather than including them independently. Indeed, the additional knowledge are made in sequence, but in process engineering design, all the elements are strongly connected, so we cannot afford to have such an approach. Instead, we traditionally prefer a global resolution.

4.3.2. Case study: modification of the AM

Before the application of the adaptation method and the resolution of the CSP model, the expert starts with analysis of AM 1. He decides to keep AO 1 and 2 because they extend the definition domain of some variables, non-exhaustiveness in the definition of the initial problem. Operator 3 is also kept because it represents a generic constraint which is regularly applied in the research of mixer configuration. The AO 4 is removed because it is too specific to the initial problem and it is not valid in the current problem, especially because the majority of vessels have a hemispherical bottom in order to ease flow and drain. The constraint in AO 5 is retained; however the maximum motor power has been shifted to a higher value to fit with the faced problem. While they are specific operators, AO 6 and 7 are also maintained, especially the latter, which is consistent with the constraint of AO 5. The former is always available, because screws are used to mix liquid with very high viscosity such as paste. But in the case study, the viscosity remains in the classical range of fluid with low viscosity. To deal with the security constraint not included

yet, an additional constraint is added on the thermal device. Since the problem requires a large heat evacuation through the vessel walls, it is better to choose devices with double lagging or half casing which are more effective for thermal withdrawal.

The resulting model is then solved providing six different mixer configurations, but none of them fills all the needs. Consequently the loop with the expert is activated. The test procedure enables to identify AO 5 as a faulty adaptation operator, first by changing the constraint on the power of the motor and then decrease the overly optimistic value. The analysis of the result after this modification on the motor energy saving shows that the whole remaining mixer configurations have a hollow shaft that needs less energy for rotating. Consequently, we can be more accurate on the constraint and instead to limit the energy consumption, AO 5 is modified to impose a hollow shaft for the variable "Type of Agitation Shaft". This example not only highlights the utility of the test procedure but also demonstrates that it can also be useful to be more accurate on the adaptation knowledge. However after testing the whole set of AO, the proposed solutions do not meet all the initial requirements. In this case we are faced with a lack of knowledge in the base; thus the diagnosis procedure is now activated. Always with the goal to save energy, elements focusing on the motor power and the type of shaft are extracted as they are the main root causes of energy consumption. With the detailed analysis of the remaining solutions and the extracted elements, we remark that there exists the possibility of having a shaft with bottom bearing. The latter generates a friction inside the mixture and therefore a loss of energy. Moreover after a deeper research on the presence of the bottom bearing, we also find that the friction generates pollution of the liquid by small metal chips. As the desired product will be used in pharmaceuticals, this pollution must be removed. Consequently a constraint is added specifying that bottom bearing cannot be used with the following comment "bottom bearing generates pollution by metal chips and increases energy consumption". Finally the model is now validated and it gives three possible mixer configurations, illustrated in Fig. 8, that must be evaluated in the next design stage.

4.4. Case evaluation and storage

When a satisfactory solution is found, then an evaluation can be made using the criteria already explained above, with the opportunity to add comments on the AM. The goal of these comments is to explain the context and objectives of the AM. For the moment, their usefulness is limited to give indications to inform the user for a possible reuse. But, as the number of AM linked to a case would go increasing, we hope to include the comments in the similarity measure to extract both $\beta source$ and $AM(source)$. For this we will propose a new similarity measure based on a semantic analysis. Currently, the evaluation of the AM is done in step 5 of the process. It is important to notice that the indicators are automatically calculated; therefore, they do not involve the user judgment that can be imbued with subjectivity. The subjective vision that a person could have on his own knowledge would affect his judgment and make the

knowledge more difficult to understand and reuse. The same remark could be done on the quality of knowledge added.

The last step of the process, i.e. step 6, deals with the storage of the new adaptation method and its association with the retrieved model if relevant for the CBR system. As we distinguish specific and generic AO, AM only composed of specific modifications, i.e. only valid for the case study, are not automatically stored in the case base. Inversely, AM that gathers more than one AO that enhances the general knowledge is proposed for storage. Currently, this step is relatively simple since it is the expert, during a maintenance session, who chooses or not to include the adaptation method in the case base. The expert can rely on the previous indicators to support his choice. Even if in our approach we introduce some requirements in order to facilitate the case base maintenance, this step deserves further study, but it remains one of the perspectives of this work.

For the case study, the AM is stored with a comment that highlights its two principal objectives: avoid mixture pollution and evacuate or bring a large heat flux to the mixture. Even if in our example we only evacuate heat, the same thermal subsystem can also be used to bring a large amount of energy. This precision is important because it widens the scope of the AM and expands the knowledge base. The AM is then evaluated and stored in the base as it encompasses generic AO and adds new knowledge to the KBS as explained before.

4.5. Discussion

The proposed methodology represents the foundation for an interactive interface, which offers a good initial solution method based on the already solved cases. In this work several evolutions were introduced to present a general formalization of the adaptation phase in the CBR cycle:

- The coupling of general and contextual knowledge, thanks to the use of CSP for the description of case in the CBR. However, Vareilles et al. (2012) have proposed using them not successively but iteratively to reach a deeper coupling. We have not implemented such an iterative approach because it cannot afford to reach the other objectives of our KBS, i.e. adaptation knowledge capitalization, and knowledge maintenance. In addition to the sustainability of knowledge, the proposed approach makes it easier to manage the consistency between the different design choices. The major drawback of our approach is that it is not completely generic because it relies on the assumption that the problem must be formulated with a CSP model. Furthermore even if the CSP model can be updated or it avoids restarting to model the problem from scratch, the problem of the initial model with its tremendous tasks for knowledge extraction and formalization remains unsolved.
- With the implementation of the interaction loop, the online expert knowledge acquisition reduces the knowledge engineering effort compared to offline processes through a guided process to
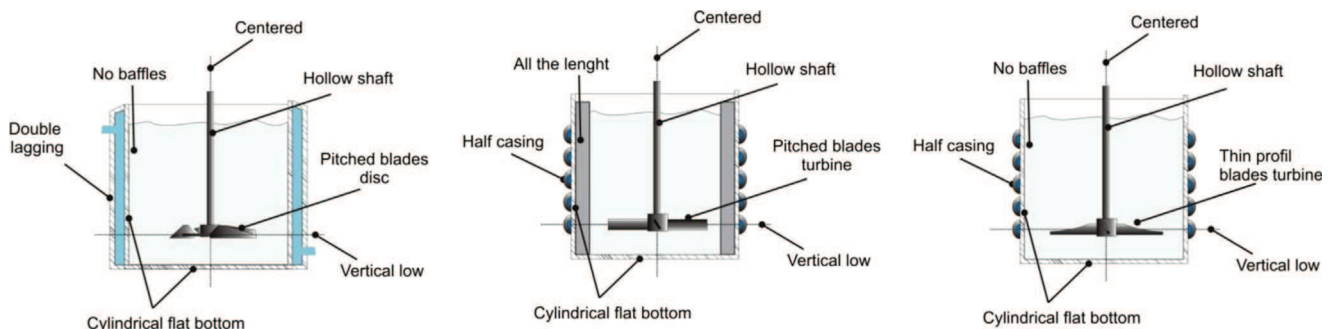

Fig. 8. Three configurations for the mixer problem.

diagnose failure, to correct or ameliorate directly the proposed solution. It also improves the relevance and confidence of the knowledge stored. Moreover, this loop provides interactivity to the system, which is not present in the adaptation methods proposed in the literature. After each modification, the user can run the model and can assess the consequences of his technological choices on the possible solutions or simply on the definition domains of the variables. It offers the ability to quickly detect constraints that may lead to design failures.

– Supporting adaptation with a CSP approach fits very well with CBR language and with design activities because specific requirements on a problem can be easily integrated to a generic model. Unfortunately, representing a case as a CSP is not always possible for all the industrial activities. Consequently, the whole method cannot be used but some parts could be easily transferred to other adaptation methods, for instance the loop and the AO. Indeed, the only assumption with AO is that the adaption knowledge can be decomposed into a finite set of elementary operators, which is often the case regardless of the application scope of the CBR. It just remains to make the AO description compatible with case representation and with the adaptation knowledge acquisition. Moreover, the advantage to decompose the adaptation method into successive single adaptation operators is that it allows one to have a detailed and deep knowledge acquisition. When an adaptation failure occurs, it also facilitates the detection between a faulty operator or the lack of adaption knowledge in the base.

– From the perspective to facilitate adaptation knowledge formalization and reuse, adaptation methods are linked to cases stored in the case base. In consequence, adaptation is not considered as a human process but rather as a process that can be automatized. But the identification of the most relevant adaptation method remains awkward despite criteria to support the choice. Indeed the criteria focus on the performances of the adapted model and not on its claimed reasons of existence or its objectives.

– The approach with adaptation operators could ameliorate the maintenance step. Currently maintenance is often considered as a human process. With our approach it can be semi-automated. Indeed as we distinguish specific and general operators in the adaptation method, only the latter are considered in the maintenance step. Then when an adaptation operator is common to several adaptation methods, it could be extracted and proposed to the expert for validation and to eventually integrate it in the initial CSP model. The performance model indicators could also be used to maintain the CSP model. Another important component of the maintenance appears when a number of adaptation methods are connected to one case. In this configuration, the initial CSP no longer meets the users requirements; thus the CSP model must be improved. Another question arises: should we divide the case in several cases or not? All these points are open questions because the maintenance step is still a perspective of this work.

5. Knowledge reusability

In KBS, adding knowledge raises inevitably the question of the quality and usefulness of the adaptation knowledge acquired. Concerning the former, it was measured through the previous four indicators and ensured thanks to the case base maintenance. But nothing is done to demonstrate that this acquired knowledge is reusable and it improves the competency of the KBS. In this section we want to demonstrate this through a series of tests. Unfortunately, we are aware that for each subsection the number of problems tested is not sufficient to draw relevant conclusions, but once again in preliminary design it is difficult to have a large

number of target problems. However, these comparative tests can give first indications of knowledge utility.

As the goal of this part is to demonstrate the adaptation knowledge reusability, we have imagined three different tests. The first one puts in highlight the influence of the quality of the knowledge acquired on its reuse and on the efficiency of the CBR system to solve new problems. The second series of tests is to solve problems twice: a first time without using the knowledge acquired and the second time by using the knowledge acquired. The goal is to prove that the way to manage this knowledge allows one to easily identify, extract and reuse it, but also to demonstrate that the CBR system raises its capacity to solve problem and its knowledge quality after each new solving episode (if the knowledge is stored in the memory). The last series of tests is to show the benefit of knowledge maintenance and the relevance of the criteria retained to extract AM.

5.1. Global test

The first test consists in measuring the number of tries necessary to solve the problem. Twenty-five problems were solved by the same user. The retained indicator measures the number of modifications of the global model ($\beta source + AM(source)$) necessary to reach a satisfactory solution. We do not retain the number of time the adaptation loop is activated because during one loop activation several AO can be modified. In the adaptation loop, each single adaptation operation is counted as one iteration, i.e. each unit modification of an AO or each new AO added. We also measure the failure rate.

The results are presented in Fig. 9, where the number of problems solved is plotted against the number of iterations needed (treated by ranges). Inside each range of iterations we distinguish four classes of AM on the basis of global criterion (GC) calculated with the four evaluation metrics:

$$GC = \frac{1}{\sum_1^4 w_i}(w_1 * Parsimony + w_2 * Accuracy + w_3 * Comments + w_4 * Cyclomatic) \qquad (2)$$

The global criterion ranges from 1 for the most relevant AM to 0 for the worst AM. For the study we use the same weight for each metrics. Indeed, we try to vary the weights and to make a sensibility analysis to see the relative importance of each criterion but we do not obtain significant results, perhaps because of the low number of problems tested.

The first comment in Fig. 9 is the relatively low rate of failure (12%) even if it seems to be slightly overestimated because of the
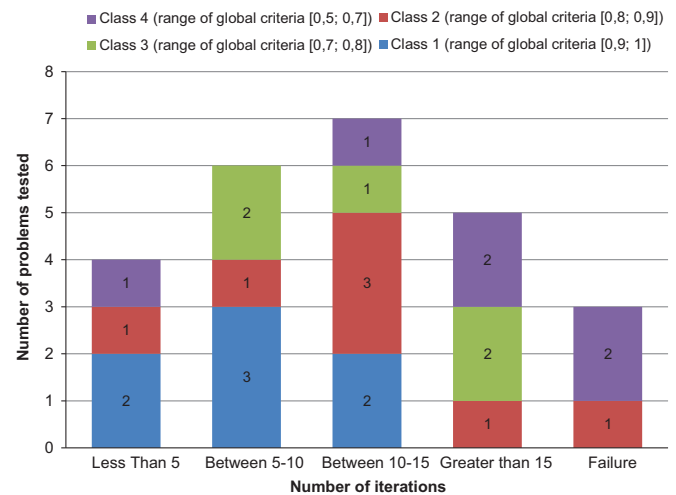

Fig. 9. Global test results.

proximity of the problems treated. Indeed, during the retrieval phase the lower value of the similarity function for the 25 problems is 0.68 (in our CBR the similarity function ranges between 0 and 1). However the low failure rate indicates that the adaptation knowledge is reused. Another remark is that the quality of the AM, measured with the global criteria, has a strong influence on the number of iterations necessary to adapt a case. The greater the quality is, the lower the number of iterations. The level of expertise of the user may also have an influence on the number of iterations but we cannot realize this test as we only have two levels of expertise.

In Fig. 9, two specific results focus our attention. In the first column we have a test with a class 4 AM but it requires less than 5 iterations to be adapted. When the AM is reviewed in details, it appears that it has a low global criterion because it is composed of numerous AO and most of them are specific. But one specific and one generic AO were well formulated and had great relevance for the faced problem. This raises the question of the granularity of the analysis. Currently, we modify AM but it will be more efficient to use AO. This would allow one to reuse AO from different AM and thus to build a method more dedicated to the adaptation problem rather than modifying an existing AM. This point is discussed in the conclusion. The second noteworthy test is in the subset of failure. There is a class 2 problem in this subset, which leads to the conclusion that when the adaptation knowledge is missing, the quality of the method does not have a real impact on the failure. It is important to notice that thanks to the diagnosis procedure we avoid the failure for two problems tested.

5.2. Comparative tests

The purpose is to test the utility of the learned adaptation knowledge. In this test a series of very similar problems were successively solved storing every time the adaptation knowledge. For each new problem addressed, the adaptation knowledge of the previous resolutions is available. In the second time, the same series of problem is solved again but without using the adaptation knowledge previously acquired. Here again the test indicator is the number of iterations, i.e. the number of corrections of the global model. For this test, twelve problems dealing with a problem of mixer configuration were treated. Indeed, to be sure that the acquired knowledge will be potentially reused we need to solve similar problems.

Fig. 10 illustrate the results for both series of tests. First by comparing the two figures we can conclude that the acquired knowledge is reused because the number of iterations for solving a problem decreases when it is stored. Indeed the mean number of iterations drops from 11 to 9. The second observation is that the overall quality of the knowledge acquired is enhanced. We can notice that there are more AM in classes 1 and 2 and less in the last two classes. The

conclusion of this test is that the purposes of the methodology are reached, namely the reuse of the adaptation knowledge acquired, the increase of the competence of the KBS through the successive resolutions, and the reduced knowledge engineering effort as the new knowledge added can immediately be available. Despite the quality and usefulness ameliorations, it still remains missing adaptation knowledge because there is always a failure.

5.3. Local tests

The local tests are often realized except for one strategy. It consists of a first resolution of a series of problems by storing the adaptation knowledge necessary but without storing the solution. Then the same problems are solved again to verify if the adaptation knowledge acquired during the first resolution is exploited. In our approach, this strategy cannot be applied in its original form because the solution part of our case encompasses the adaptation knowledge and it cannot be dissociated. Consequently, this two-step method is used to test the relevance of the four criteria retained to evaluate the AM. A series of twenty-four problems was solved and once the AM is established they were stored and evaluated (in fact we use the same series of problems as in subsection 5.1, but for one problem the AM is not stored because it does not improve the competency of the KBS). Then the same series of problems was solved again to see the position of the AM in the Pareto front. First, after the first resolution, the AM were stored as they were created, i.e. without maintenance. Fig. 11 shows the rank of the AM in the Pareto front during the second resolution. The set of AM linked to a case is ranked according the decreasing values of the global score calculated with formula (2). The AM of 17 problems are classified in the first three methods and 20 in the first four. The proposed criteria to evaluate the quality of AM and the tool to support the AM selection seem relevant even if some problems remain for which the acquired knowledge is accessible with difficulty. As the AM are found mainly in the first ranks, this also demonstrates the usefulness of the acquired knowledge.

The same series of tests was realized but after performing a maintenance operation on the AM and their AO. Indeed, as we have shown in the case study corrections on AO or AM can improve the relevance of the method and enlarge their scope. As this maintenance is done by an expert, it results in an increase of the global score. Diagram 2 in Fig. 11 gives the results of this second test. We can clearly see the benefit of the maintenance step on the quality and the usefulness of the adaptation knowledge acquired.

6. Conclusions

In this article, we first presented a new methodology for the adaptation phase in CBR. This methodology is more focused on
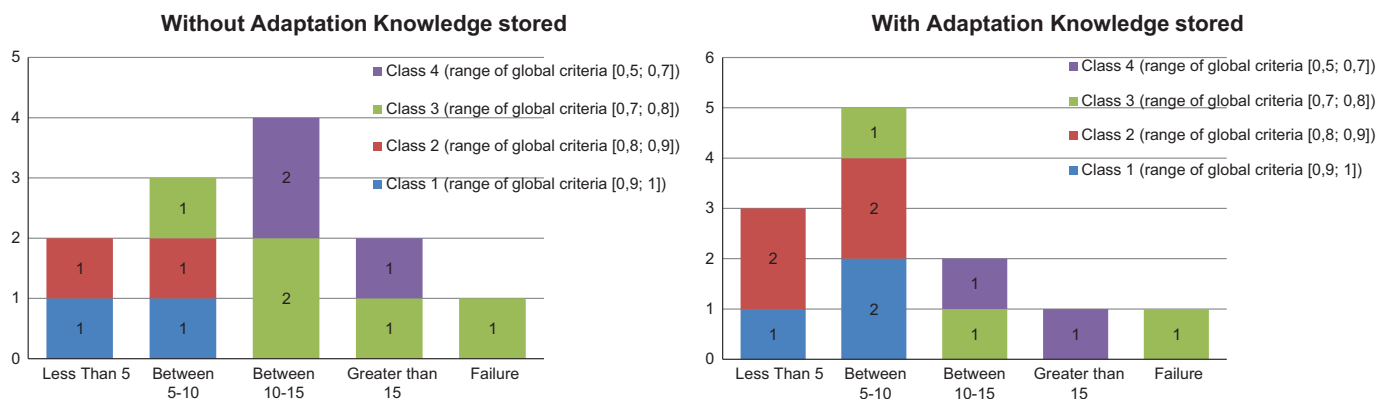

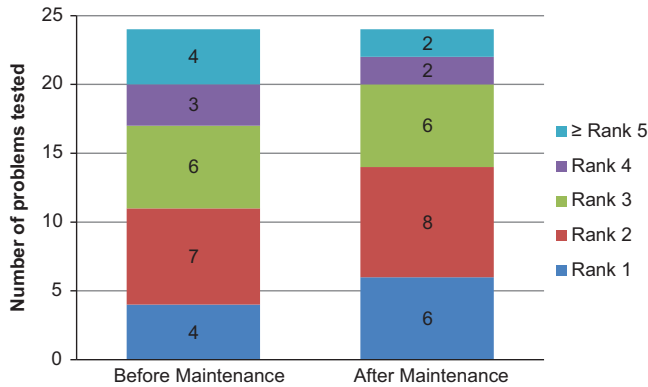**Fig. 10.** Comparative tests results.

Fig. 11. Local tests results.

design and more particularly in the process engineering domain, but some parts of this methodology can be easily transferred to other domains and activities. The preliminary process design is a complex domain where past experiences are often reused for solving new problems. Among the possible methods for knowledge capitalization and reuse, CBR is an effective approach that finds a solution by retrieving past experiences and CSP methods have been applied in the resolution of complex combinatorial problems in the domains of configuration and design.

The primary motivations for combining these two approaches in process design are:

- To provide an excellent representation for a problem with the CSP formalism that facilitates its treatment by the CBR, including retrieval, adaptation and maintenance phases; meanwhile, it enables combining of two categories of knowledge: contextual and general.
- To use the powerful CSP algorithms for solving very complex problems.
- To increase the efficiency of the CSP models by reusing the past experiences especially in the cases where modeling is difficult or almost impossible to obtain.

The major contributions of this article, we believe, are (i) the modification of the adaptation loop which adds interactivity in CBR and permits one to acquire online expert knowledge to reduce the knowledge engineering effort by a timely solicitation, (ii) the decomposition of the adaption knowledge into elementary knowledge containers to facilitate knowledge elicitation, reuse and maintenance, (iii) a decision support system to search and identify the relevant adaptation knowledge thanks to the proposition of evaluation criteria, and (iv) a diagnosis process to identify the root cause of failures. Some tests were also performed to evaluate the quality and the usefulness of the adaptation knowledge acquired. Indeed in numerous CBR systems, new knowledge is acquired to improve the skills of the system but nothing was done to establish the relevance of this new added knowledge.

In the future, we will pay attention to these five main perspectives:

- The similarity measurement can be improved by introducing the AM retrieval in the calculation as explained in Section 4.4. This can be done thanks to the proposal of a similarity measure relying on a semantic analysis.
- The way the adaptation knowledge would be managed and maintained is still an open question. Even if case base maintenance requirements were anticipated and introduced in the methodology, this step is perfectible. For instance, it could be improved with machine learning approach in order to extract new knowledge from the adaptation methods encompassed in the case base.
- Currently the reuse of the adaptation knowledge is done at the level of the AM. We can imagine going further, rather than using existing AM we can build a specific AM to the problem addressed. As at the deeper level the knowledge container is the AO, we could extract AO from different AM and thus elaborate a more targeted AM. The difficulty is not the research and extraction of suitable AO but the construction of a coherent AM. Indeed, the grouping of AO belonging to different AM may lead to incoherent and unusable AM and would generate adaptation failure. The way to ensure model coherence is not obvious and needs further development.
- Another perspective concerns the failure. As explained before, when an adaption failure occurs the problem is stored in a dedicated case base. The goal would be to exploit these failures to raise the issue at the stage of research and to build a link with Conflict Based Approach tools. We can also include this information at the early steps of the CBR, for example at step 1a or 2 of our cycle to alert the user that the problem cannot be solved with the current knowledge stored in the base.

The phase of tests can be continued for instance by increasing the number of problems treated in the series or by integrating the level of expertise of the user in the parameters to test.

Annex 1. Input data for problem description

		Feature	Value Type	Example
Mixture description		Phase State	Semantic	Liquid, Liquid–Solid…
Phase description	Liquid	Density Viscosity	Numerical Numerical or semantic	1000 kg/m³ 1 Pa/s, high…
	Solid	Type of fluid Density Wettability	Semantic Numerical Semantic	Newtonian… 2000 kg/m³ High, Medium, Low
		Solubility	Numerical or Semantic	300 g/L, Low…
		Mean Diameter	Numerical	10 mm
	Gas	Flow rate Solubility	Numerical Numerical or Semantic	10 kg/s 200 g/L, High
		Pressure	Numerical or Semantic	1 Pa, Low
		Inert phase	Semantic	Yes or no
Type of Operation		Type of application	Semantic	Liquid-Gas
		Physical Characteristic	Semantic	Suspension, Dispersion
		Chemical Characteristic	Semantic	Dissolution, Absorption, Fermentation
Hydrodynamics Characteristics		Shear	Semantic	High, Medium, Low

Turbulence	Semantic	High, Medium, Low
Pumping	Semantic	High, Medium, Low
Thermal Device	Thermal	Numerical or Semantic	10 KW, High, Medium…

Annex 2. Variables and Definition Domains

See Annex Table A1.

Table A1
Variables and domains for a mixer.

Variables	Domains
Vessel	Cylindrical flat bottom, Cylindrical conical bottom, Hemispherical, square…
Baffle	None, upper part, lower part, all the length…
Size of the baffle	Range of real number depending on the two previous variables
Thermal device	None, Heating coil, Double lagging, Half casing, Half casing with coil…
Size of the thermal devices	Range of real number depending of the previous variable
Type Impellers	Marine propeller, Hydrofoils, flat blades, pitched blades, Rushton turbine…
Vertical position	Vertical High, Vertical low, Lateral
Horizontal position	Off center, Centered, Tilted
Motors	Power, Speed
Shaft	Solid Shaft, Hollow Shaft, Shaft with bottom bearing

References

Aamod, A., Plaza, E., 1994. Case-based reasoning: foundational issues, methodological variations, and systems approaches. AI Commun. 7, 39–59.

Badra, F., Cordier, A., Lieber, J., 2009. Opportunistic adaptation knowledge discovery. Lect. Notes Art. Int. 5650, 60–74.

Chebel-Morello, B., Haouchine, M.K., Zerhouni, N., 2013. Reutilization of diagnostic cases by adaptation of knowledge models. Eng. Appl. Artif. Intell. 26 (10), 2559–2573.

Cordier, A., Fuchs, B., Lieber, J., Mille, A., 2007. Acquisition interactive des connaissances d'adaptation intégrée aux sessions de raisonnement à partir de cas. 15ème Atelier de RàPC. pp. 71–84.

Cortes-Robles, G., Negny, S., Le Lann, J.M., 2009. Case based reasoning and TRIZ: a coupling for Innovative conception in chemical engineering. Chem. Eng. Proc. 4, 239–249.

Craw, S. 2009. Agile case based reasoning: a grand challenge towards opportunistic reasoning from experiences. IJCAI Workshop, pp. 33–39.

Craw, S., Wiratunga, N., Rowe, R.C., 2006. Learning adaptation knowledge to improve case based reasoning. Artif. Intell. 170, 1175–1192.

D'Acquin M., Brachais S., Lieber J., Napoli A., 2004. Vers une acquisition automatique de connaissances d'adaptation par examen de la base de cas.- Une approche fondée sur des techniques d'extraction de connaissances dans des bases de données, 12ème atelier raisonnement à partir de cas, RàPC'04.

Douglas, J.M., 1998. Conceptual Design of Chemical Processes. McGraw-Hill, New York.

Fidjeland, M.K., 2006. Distributed Knowledge in Case-Based Reasoning: Knowledge Sharing and Reuse within the Semantic Web. Trondheim.

Fuchs, B., Lieber, J., Mille, J., Napoli, A., 2000. (ECAI00). An algorithm adaptation case based reasoning, 45–49.

Fuchs, B., Lieber, J., Mille, J., Napoli, A., 2014. Differential adaptation: an operational approach to adaptation for solving numercial problem with CBR. Knowl. Based Syst. 68, 103–114.

Galushka, M., Patterson, D., 2006. Intelligent index selection for case-based reasoning. Knowl. Based Syst. 19 (8), 625–638.

Greene, D., Freyne, J., Smyth, B., Cunningham, P., 2008. (ECCBR LNAI). In: Althoff, K.-D., et al. (Eds.), An analysis of research themes in the CBR conference literature, 5239. Springer, Berlin, pp. 18–43.

Hanks, S., Weld, D.S., 1995. A domain independent algorithm for plan adaptation. J. Artif. Intell. Res. 2, 319–360.

Harmsen, G.J., 1999. (FOCAPD99). Industrially applied process synthesis method creates synergy between economy and sustainability.

Hinrichs, T.R., 1992. Problem Solving in OpenWorlds: A case study in Design. Lawrence Erlbaum Associates, Hillsdale.

Ho, K.T.V., Sebastian, P., Quirante, T., 2010. Multiobjective optimization of the design of two-stage flash evaporators: Part 1. Int. J. Thermal Sci. 49 (2), 2453–2458.

Hua, K., Faltings, B., Smith, I., 1996. CADRE: case-based geometric design. Artif. Intell. Eng. 10, 171–183.

Inakoshi, H., Okamoto, S., Ohta Y., Yugami, N., 2011. Effective decision support for product configuration by using CBR. In: Proceedings of the 4th International Conference on Case-Based Reasoning.

Jarmulak J., Craw S., Rowe R., 2001. Using case base data to learn adaptation knowledge for design. In: Proceedings of the 17th IJCAI'01, pp. 1011–1016.

Karray, M., Chebel-Morello, B., Zerhouni, N., 2014. PETRA: Process Evolution using a Trace based system on maintenance platform. Knowl. Based Syst. 68, 21–39.

Leake, D., Kinley, A., Wilson, D., 1996. Acquiring case-adaptation knowledge: a hybrid approach, Proceedings of the 13th National Conference on Artificial Intelligence, Menlo Park, CA. AAAI Press, Menlo Park, CA, pp. 684–689.

Leake, D., Powell, J., 2007. Mining large-scale knowledge sources for case adaptation knowledge. Lect. Notes Comp. Sci. 4626, 209–223.

Li, X., Kraslawski, A., 2004. Conceptual process synthesis: past and current trends. Chem. Eng. Proc. 43, 589–600.

Li, X.N., Rong, B.G., Kraslawski, A., Nyström, L., 2003. A conflict Based approach for process synthesis with wastes minimization. Comp. Chem. Eng. 14, 209–214.

Lieber, J., 2007. (ICCBR'07). Application of the revision theory to adaptation in case based reasoning: the conservative adaptation, 239–253.

Lieber, J., D'Aquin, M., Brachais, S., Napoli, A., 2004. (RàPC'04). A comparative study of Adaptation Knowledge Acquisition works for case based reasoning (In French), 53–60.

López, B., 2003. Holiday scheduling for city visitors. Inf. Commun. Technol. Tourism, 252–260.

Maher, M.L., Pu, P., 1997. Introduction to the Issues and Applications of Case-Based Reasoning in Design. In: Maher, M.L., Pu, P. (Eds.), Issues and Applications of Case-Based Reasoning in Design. LEA Publishers, London, pp. 1–9.

Massi, S., Craw, S., Wiratunga, N., 2007. When Similar Problems Don't Have Similar Solutions (ICCBR 2007). In: Weber, R.O., Richter, M.M. (Eds.), Lect. Notes Art. Intell., 4626; 2007, pp. 92–106.

Medjdoub, B., 2009. Constraint-based adaptation for complex space configuration in building services. J. Inform. Tech. Constr 14, 724–735.

Neagu, N., 2005. Constraint Satisfaction Techniques for Agent-Based Reasoning. Whitestein Series in Software Agent Technologies and Autonomic Computing, Berlin.

Negny, S., Belaud, J.P., Cortes Robles, G., Roldan Reyes, E., Barragan Ferrer, J., 2012. Toward an eco-innovative method based on a better use of resources: application to chemical process preliminary design. J. Clean. Prod. 32, 101–113.

Negny, S., Riesco, H., Le Lann, J.M., 2010. Effective retrieval and new indexing method for case based reasoning: application in chemical process design. Eng. Appl. Art. Int. 23, 880–894.

Pu, P., Purvis, L., 1995. Adaptation Using Constraint Satisfaction Techniques. Lect. Notes Comp. Sci. 1010, 289–300.

Pu, P., Purvis, L., 1997. Formalizing the adaptation process for Case-Based Design. In: Maher, M.L., Pu, P. (Eds.), Issues and Applications of Case-Based Reasoning in Design. LEA Publishers, London, pp. 221–239.

Purvis, L., 1998. Synergy and commonality in case based and constraint based reasoning. In: AAAI Spring Symposium on Multimodal Reasoning. pp. 80–84.

Qi, J., Hue, J., Peng, Y., Wang, W., Zhan, Z., 2011. AGFSM: an new FSM based on adapted Gaussian membership in case retrieval model for customer-driven design. Expert Syst. Appl. 38, 894–905.

Roldan, E., Negny, S., Le Lann, J.M., Cortes, G., 2011. Modified case based reasoning cycle for expert knowledge acquisition during process design. Comput. Aided Chem. Eng. 29, 296–300.

Ruet, M., Geneste, L., 2002. Search and adaptation in a fuzzy object oriented case base. Lect. Notes Comp. Sci 2416, 350–364.

Schank, R.C., 1994. Dynamic Memory Revisited. Cambridge University Press.

Schmitt, G., Dave, B., Shih, S., 1997. Case-based architectural design: the experience of CADRE. In: Maher, M.L., Pu, P. (Eds.), Issues and Applications of Case-Based Reasoning in Design. LEA Publishers, London, pp. 241–260.

Smyth, B., Keane, M.T., 1996. Using adaptation knowledge to retrieve and adapt design cases. Knowl.-Based Syst. 9 (2), 127–135.

Smyth, B., Keane, M.T., 1998. CBR and the Similarity Assumption. J. Art. Int. 102, 249–293.

Sqalli, M.H., Purvis, L., Freuder, E.C., 1999. Survey of applications integrating constraint satisfaction and case-based reasoning. In: Proceedings of the First International Conference and Exhibition on the Practical Application of Constraint Technologies and Logic Programming, London.

Vareilles, E., Aldanondo, M., Codet-de-Boisse, A., Coudert, T., 2012. How to take into account general and contextual knowledge for interactive aiding design: towards the coupling of CSP and CBR approaches. Eng. Appl. Art. Int. 25, 31–47.

Vernat, Y., 2004. Formalizing and qualifying models for constraint-based embodiment design (Ph.D. thesis). Univ of Bordeaux.

Vidotto, A., Brown, K., Beck, J., 2007. Managing restaurant tables using constraints. Knowl.-Based Syst. 20 (2), 160–169.

Voss, A., 1997. Case reusing systems – survey, framework and guidelines. Knowl. Eng. Rev. 12 (1), 59–89.

Wang, H., Liao, L., 1997. A framework of constraint-based modeling for cooperative decision systems. Knowl.-Based Syst. 10 (2), 111–120.

Wang, X., Zheng, J., Zeng, Q., 2009. A design of product collaborative online configuration model. Lect. Notes Comput. Sci. 5738, 359–366.