Interactive Knowledge Acquisition in Case Based Reasoning

Amélie Cordier[1], Béatrice Fuchs[1], Jean Lieber[2], and Alain Mille[1]

[1]LIRIS CNRS, UMR 5202, Université Lyon 1, INSA Lyon, Université Lyon 2, ECL
43, bd du 11 Novembre 1918, Villeurbanne Cedex, France,
{Amelie.Cordier, Beatrice.Fuchs, Alain.Mille}@liris.cnrs.fr
[2]LORIA (UMR 7503 CNRS–INRIA–Nancy Universities),
BP 239, 54506 Vandœuvre-lès-Nancy, France
Jean.Lieber@loria.fr

Abstract. In Case Based Reasoning (CBR), knowledge acquisition plays an important role as it allows to progressively improve the system's competencies. One of the approaches of knowledge acquisition consists in performing it while the system is used to solve a problem. An advantage of this strategy is that it is not to constraining for the expert: the system exploits its interactions to acquire pieces of knowledge it needs to solve the current problem and takes the opportunity to learn this new knowledge for future use. In this paper, we present two approaches of interactive knowledge acquisition in CBR. Both approaches rely on the exploitation of reasoning failures. Indeed, an interactive learning process aiming at correcting the solution and at learning new knowledge is triggered when a reasoning failure occurs.

1 Introduction

Case-based reasoning (CBR) is a reasoning paradigm which consists in solving new problems by adapting solutions of previously solved problems. This process is supported by various knowledge used to reason on cases. In particular, adaptation knowledge is of major importance: it is used during the retrieval step to retrieve a good source case (e.g. a case easy to adapt) and, of course, during the adaptation step to build the solution to the current problem. Unfortunately, knowledge management in CBR is still a difficult problem.

Recently, several works have addressed the issue of knowledge acquisition by using machine learning techniques. For example in [1] or in [2], adaptation knowledge is automatically extracted from the case base. These methods are efficient to acquire initial knowledge but they require an important effort from the expert: he/she has to deal with a large number of adaptation rules. Furthermore, these methods are not well suited when minor adjustments of some knowledge have to be done locally.

In order to deal with this issue, we propose a complementary approach of knowledge acquisition which exploits interactions between the system and the expert *during* the reasoning sessions. This approach is *opportunistic* because the system makes profit of each opportunity to acquire new knowledge or to update its own knowledge. Reasoning failures constitute such opportunities. Indeed, if a solution proposed by the system is inconsistent, it is most probably because knowledge used to produce it is incorrect or incomplete. Usually, reasoning failures are detected, and often corrected, by the expert during the test-and-repair step of the CBR cycle. Interactive knowledge acquisition takes place during this step. As it is performed during the "normal" use of the system, it is almost transparent for the expert and, therefore, it is not too restrictive. Principles of interactive knowledge acquisition based on reasoning failures exploitation have been used in two projects : FRAKAS , a prototype illustrating domain knowledge acquisition in the field of medical recommendation and IAKA, a more generic framework for adaptation knowledge acquisition.

The remainder of this paper is organized as following. In section 2, we give an overview of the knowledge acquisition issues in CBR and we discuss the tight interconnection between some knowledge containers. Section 3 briefly describes the prototype FRAKAS , the framework IAKA and KAYAK, an application implementing some of the principles of IAKA. Section 4 discuss this work and, finally, section 5 concludes the paper by giving perspectives for IAKA, FRAKAS , and more generally for opportunistic knowledge acquisition.

2 Knowledge acquisition in case based reasoning

Case-based reasoning systems are knowledge-based systems (KBS) which, if we follow Richter's proposition [3], make use of four distinct knowledge sources: domain knowledge, cases, similarity knowledge and adaptation knowledge. But one can have an unified view of the knowledge involved in CBR systems as there exists close relations between the different knowledge containers.

2.1 Relation between similarity and adaptation knowledge

Since Smyth introduced the concept of adaptation-guided retrieval in [4], a great deal of research works studies the relation between similarity and adaptation knowledge, and tools aiming at facilitating adaptation by using relevant knowledge are proposed. Indeed, adaptation is one of the most difficult steps of CBR and therefore any effort to facilitate it is useful. Adaptation-guided retrieval argue that sources cases that are most similar to the target case (i.e. the problem) are not always the easiest to adapt, in particular when the similarity rests on surface features. Retrieval must therefore search not only for similar cases, but especially for easily adaptable cases.

Among the works that take into account the adaptability of a case while retrieval, we can cite [5], that include and «adaptation cost» in its similarity measure and [6], that aims at decreasing the difficulty of adaptation by increasing the similarity between the problems.

These works clearly highlight the strong relation existing between similarity knowledge and adaptation knowledge. More generally, it is not advisable to consider the different stages of the CBR separately and independently from one another, but rather as contributing to a common objective. For example, the retrieval step tends to facilitate adaptation by using an adaptability criteria to select a source case. A case's adaptability must therefore be taken into account in the retrieval step. As similarity and adaptation knowledge are tightly connected, learning adaptation knowledge if of particular importance because it should contribute to better retrieval.

2.2 Acquiring CBR knowledge

Solutions produced by CBR systems may not be satisfactory because of either a lack of sufficient knowledge or imperfectly described knowledge, leading to reasoning failures. Thus, many research work address the learning component in CBR systems along several perspectives.

One of these perspectives characterizes the different knowledge containers targeted by the learning process [3]: case's vocabulary, cases, similarity and solution transformation (i.e. adaptation knowledge). Some approaches consider similarity and adaptation knowledge as distinct and learn them separately [7]. We defend the idea that, ideally, only domain and adaptation knowledge should be learned and similarity knowledge should be deduced from adaptation knowledge.

Another perspective characterizes the knowledge source used by the learning process [8]. Some approaches use the content of the knowledge containers, in particular those who rely on machine-learning or "off-line" techniques in order to explicit knowledge [2; 9; 1]. Other "on-line" approaches, by contrast, aim at acquiring new knowledge that is not already in the system through interactions with the environment [10; 5]. Learning takes place during the use of the system and aims at acquiring domain knowledge. The evaluation of the adapted solution may highlight the fact that it does not meet the requirements of the target problem. In this situation, a reasoning failure occurs and is handled by a learning process. The expert is involved in the identification of faulty knowledge and a repair process is triggered to correct it.

Acquisition of domain knowledge When there is a lack of domain knowledge, the system may infer a solution that is correct with respect to the knowledge base but not with the real world: this constitutes a failure. The historical approach of the CHEF system [11], a case-based planner in the cooking domain, uses a causal model to test an adapted plan and triggers a learning process when a reasoning failure occurs. In case of failure, CHEF generates an explanation to guide the repair of the solution. Then, the learning process sets appropriate indexes in order to avoid a later retrieval of the faulty plan in similar circumstances. The FRAKAS system [12] (briefly described in this paper) is an approach for interactive domain knowledge acquisition. Learning takes place during the use of the system and aims at acquiring domain or adaptation knowledge. The evaluation of the adapted solution may highlight that it does not meet the requirements of the target problem. In this situation, a reasoning failure occurs and is processed by a learning process. The expert is involved in the process of identifying inconsistent parts of the solution which helps to augment the knowledge base. The expert is involved in a simple manner to point out faulty knowledge and he/she may provide a textual explanation of the identified error to support complementary off-line knowledge acquisition.

Acquisition of adaptation knowledge The difficulty of the adaptation step has been subject of numerous research works and has been studied according to several directions: unifying approaches proposing general adaptation models [13]; catalogs of adaptation strategies applicable to several domains [14; 15]; and methods for acquiring adaptation knowledge that try, in a particular domain, to highlight general principles explaining the adaptation process [16]. A distinction is made between different approaches of acquisition of adaptation knowledge: «knowledge light» approaches consist in re-using knowledge available in the system to infer new knowledge while other approaches try to acquire new knowledge by using the interactions between the system and its environment. The former approaches take place outside the problem solving phase, whereas the latter take place during the solving process and therefore present numerous possibilities of interactions with the expert.

The approach presented in [17] can be classified in the first category: it consists in determining pairs of cases and using differences between their attributes to improve adaptation rules. The adaptation rules created are then refined and generalized. In the same light, [1] propose an approach of knowledge learning based on a particular search technique called *frequent pattern extraction*. The main idea is to use the differences between cases taken in pairs. Indeed, these differences can be interpreted as the result of an adaptation effort. It is then possible to deduce some adaptation knowledge. Among the approaches of the second category, we may note that of [18] where knowledge learning takes on several forms. Introspective reasoning gives the systems the possibility to learn new knowledge, for example during the adaptation step [19]. Adaptation knowledge is acquired via a CBR cycle within the main CBR cycle.

One of the drawback of the approaches that aim to use knowledge already available in the system to infer new adaptation knowledge is their limitation to the «vocabulary» of the case-base. They do not allow to infer knowledge that is not «explainable» using the existing knowledge of the application. Furthermore, they only give the expert a minor role which consist in validating the inferred knowledge. On the contrary, approaches that allow the learning of knowledge during the reasoning process provide the possibility of adding new knowledge to the system and the opportunity for the expert to play an actual and active role in the process. We stick to the second approach and our wish is to place the expert at the center of the learning process so that he can simultaneously play an active role in the solution of the problem and in the acquisition of adaptation knowledge.

3 Opportunistic knowledge acquisition

In this section, we describe two works focusing on opportunistic knowledge acquisition. They both rely on the same principle: performing interactive knowledge acquisition by exploiting reasoning failures. The first work, FRAKAS , is a CBR prototype implementing the principles of conservative adaptation and allowing the acquisition of domain knowledge. The second IAKA, is a generic framework aiming at facilitating adaptation knowledge acquisition. KAYAK is a prototype implementing some of the features proposed in IAKA. As it will be discussed in section 4, an ongoing work aims at clarifying the links between these two approaches.

3.1 Interactive domain knowledge acquisition: FRAKAS

FRAKAS (FailuRe Analysis for domain Knowledge AcquiSition) is a CBR system that relies on the principles of conservative adaptation [20] to solve problems. Domain knowledge, represented in propositional logic, is used to perform the adaptation. Interactive domain knowledge acquisition is done by exploiting the adaptation failures. Thanks to an appropriate graphical interface, the expert is able to identify adaptation failures and to highlight faulty knowledge. A process allowing to correct the faulty knowledge is then started. The systems uses the corrections done by the expert to infer new knowledge. More details about FRAKAS can be found in [12].

3.2 Interactive adaptation knowledge acquisition: IAKA

In this section, we present the set of principles to perform interactive adaptation knowledge acquisition defined in IAKA. Next, we briefly present KAYAK, a prototype developed to illustrate IAKA.

Notions, notations and hypothesis of IAKA. According to the classic CBR cycle [21], we assume that the CBR process is composed of four steps: retrieval, adaptation, test-repair and memorization. The aim of a CBR session is to produce a candidate solution to solve a target problem noted $\mathtt{tgt}$ expressed by the expert. During the retrieval step, the system looks for a source case, noted $\mathtt{srce\text{-}case}$ deemed to be used to solve $\mathtt{tgt}$ according to the adaptation-guided retrieval principle [22]. The adaptation step consists in modifying the solution $\mathtt{Sol}(\mathtt{srce})$ of the retrieved source case by applying the relevant adaptation knowledge to produce a candidate solution $\widetilde{\mathtt{Sol}}(\mathtt{tgt})$. During the repair step, the expert validates or corrects the solution $\widetilde{\mathtt{Sol}}(\mathtt{tgt})$ proposed by the system. If the candidate solution is validated, $\widetilde{\mathtt{Sol}}(\mathtt{tgt})$ becomes $\mathtt{Sol}(\mathtt{tgt})$: the problem is solved and a new case $(\mathtt{tgt}, \mathtt{Sol}(\mathtt{tgt}))$ is added to the case base during the memorization step. In the other situation, the system seizes the opportunity to improve its adaptation knowledge and then, proposes a new adapted solution to the expert. Thus, the knowledge acquisition occurs during the adaptation, repair and the memorization steps.

We also make the hypothesis that the adaptation phase can be decomposed in several steps. Each step correspond to an elementary adaptation operation, performed by an adaptation operator $\mathtt{AO}$, to solve a specific adaptation problem. The set of elementary adaptation operators allowing to go from $\mathtt{Sol}(\mathtt{srce})$ to $\widetilde{\mathtt{Sol}}(\mathtt{tgt})$ is called an *adaptation method* noted $\mathtt{AM}$. Thus, we have $\mathtt{AM} = \{\mathtt{AO}_i\},\ i \in \{1, \ldots n\}$ (an adaptation method is a set of n adaptation operators, n is the number of adaptation steps needed to solve $\mathtt{tgt}$). To each case is associated a finite (and not empty) set of adaptation methods: there is one or several ways to adapt a source case.

Retrieval and adaptation. In this framework, the retrieval step is guided by the adaptability: it uses adaptation knowledge to weight the differences between $\mathtt{srce}$ and $\mathtt{tgt}$ in order to estimate a distance $\mathtt{dist}$ which reflects the adaptation difficulty. As the distance is dependent on the adaptation method, it is calculated for each couple $(\mathtt{srce}, \mathtt{AM}(\mathtt{srce})_\mathtt{j})$, with $\mathtt{AM}(\mathtt{srce})_\mathtt{j}$ $(j \in \{1, \ldots m\})$ one of the methods associated to srce-case. The selection of the retrieved case consists in choosing a source case $(\mathtt{srce}, \mathtt{Sol}(\mathtt{srce}))$ and an associated adaptation method $\mathtt{AM}(\mathtt{srce})$. During the adaptation step, the solution $\mathtt{Sol}(\mathtt{srce})$ is reused to build $\widetilde{\mathtt{Sol}}(\mathtt{tgt})$ by applying the adaptation method $\mathtt{AM}(\mathtt{srce})$. The resulting solution, $\widetilde{\mathtt{Sol}}(\mathtt{tgt})$ is then proposed to the expert for validation during the test-and-repair step. Two solutions are then possible:

- The expert judges $\widetilde{\mathtt{Sol}}(\mathtt{tgt})$ satisfactory: the solved target case $(\mathtt{tgt}, \mathtt{Sol}(\mathtt{tgt})) = (\mathtt{tgt}, \widetilde{\mathtt{Sol}}(\mathtt{tgt}))$ is stored in the case base and $\mathtt{AM}(\mathtt{tgt}) = \mathtt{AM}(\mathtt{srce})$ is stored in the adaptation method base during the memorization step.
- The expert judge $\widetilde{\mathtt{Sol}}(\mathtt{tgt})$ not satisfactory, an interaction loop is then activated: it allows to find a solution to tgt and to acquire new knowledge (this process is described in the following paragraph). The cycle adaptation-test-repair continues until a satisfactory solution is accepted by the expert.

The interaction loop. The interaction loop is performed during the test-and-repair phase. It allows the improvement an adaptation method thanks to the correction of its constituting adaptation operators. When a solution is said to be inconsistent by the expert, the system tries to identify and correct the operators responsible for the failure. The failure may come from one or more steps constituting the adaptation. Each step corresponds to the application of a specific adaptation operator. Thus, the system has to identify among the adaptation operators of the adaptation method those that need to be corrected. IAKA's strategy consists in testing separately each one of these operators. For that purpose, the system isolates an adaptation operator and uses it to solve a new problem pb elaborated specifically from srce in order to test it in one step. The solution of this new problem, obtained by adaptation, is then submitted to the expert. The answer of the expert allows to estimate the validity of the adaptation operator. If the expert validates the adaptation performed, the adaptation operator is considered as correct and the system then chooses another operator to test. If the expert does not validate the adaptation, the system asks for the correct solution for pb and modifies the adaptation operator in consequence. The interaction loop ends as soon as an adaptation operator as been updated or when all the operators of the method have been processed by the expert. The figure 1 summarizes this process.

3.3 KAYAK: a prototype for IAKA

KAYAK is a CBR application prototype developed to experiment and validate the principles of the framework IAKA. The application domain of this prototype is the one of the functions of n variables. Solving a problem with KAYAK consists in determining an approximation of the value of the function f knowing the n variables. One of the specificities of KAYAK is that the domain expert is a virtual expert: it is simulated by a couple (f, ε) where f is the function and ε (with $\varepsilon > 0$) is a parameter representing the demand level of the expert. The retrieval step relies on a distance taking into account the adaptation difficulty and the adaptation process is performed thanks to the differential adaptation strategy [23]. The goal of the adaptation step is to estimate the value y^c of the function f, given the values of the variables x_i^c, by adaptation of the solution $\mathtt{Sol}(\mathtt{srce}) = y^s$ of a retrieved case srce-case.

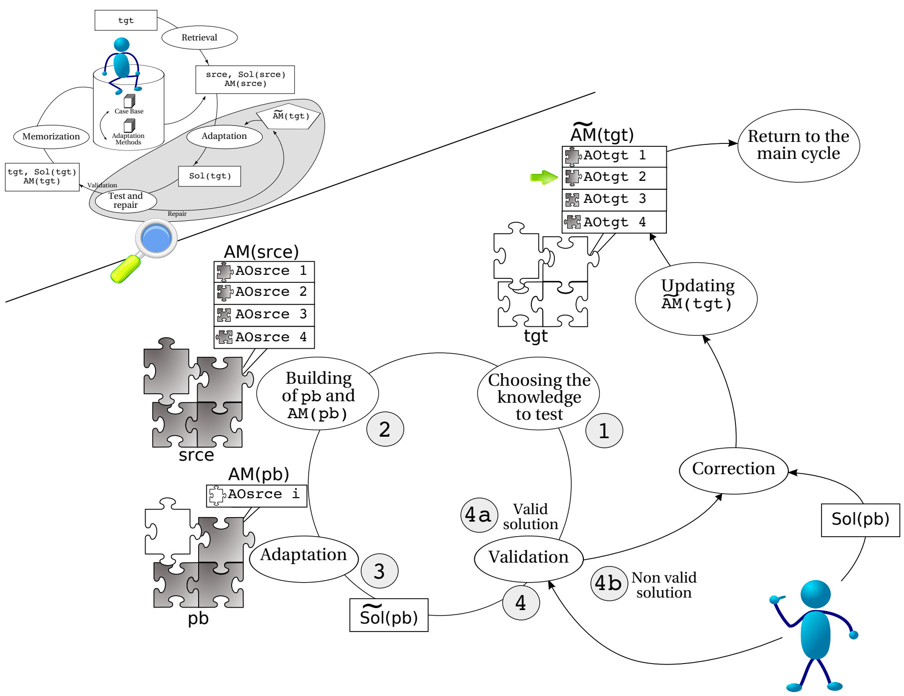

Fig. 1. Interaction loop with the expert. On top left : a reminder of the classic CBR cycle. The main part of the figure describes the interaction loop.

1. Selection of an adaptation operator $\mathtt{AO}(\mathtt{tgt})_\mathtt{i}$ to test among the operators of $\widetilde{\mathtt{AM}}(\mathtt{tgt})$ not tested yet.
2. Building of $\mathtt{pb}$ from $\mathtt{srce}$ by replacing a part of $\mathtt{srce}$ by the part of $\mathtt{tgt}$ justifying the application of $\mathtt{AO}(\mathtt{tgt})_\mathtt{i}$. The adaptation method associated $\mathtt{AM}(\mathtt{pb})$ is constituted of the only adaptation operator $\mathtt{AO}(\mathtt{tgt})_\mathtt{i}$. Thus, we have : $\mathtt{AM}(\mathtt{pb}) = \{\mathtt{AO}(\mathtt{tgt})_\mathtt{i}\}$.
3. The adaptation of $(\mathtt{srce}, \mathtt{Sol}(\mathtt{srce}))$ with the method $\mathtt{AM}(\mathtt{pb})$ to solve $\mathtt{pb}$ gives $\widetilde{\mathtt{Sol}}(\mathtt{pb})$.
4. $\widetilde{\mathtt{Sol}}(\mathtt{pb})$ is presented to the expert who does or does not validate this solution. Two situations are then possible:
   (a) The expert validates $\widetilde{\mathtt{Sol}}(\mathtt{pb})$. Another operator to test will be chosen during the return to the first step.
   (b) The expert does not validate $\widetilde{\mathtt{Sol}}(\mathtt{pb})$.
      i. The system asks to the expert the value of $\mathtt{Sol}(\mathtt{pb})$ and correct $\mathtt{AM}(\mathtt{pb})$ by modifying $\mathtt{AO}(\mathtt{pb})_\mathtt{i}$ (the only one that can be contested).
      ii. $\widetilde{\mathtt{AM}}(\mathtt{tgt})$ is updated by replacing the old $\mathtt{AO}(\mathtt{tgt})_\mathtt{i}$ by the $\mathtt{AO}(\mathtt{tgt})_\mathtt{i}$ newly corrected and a new adaptation is performed to test the impact of the modified adaptation operator.

4 Discussion

Interactive knowledge acquisition is an on-line learning process: it is performed during the problem solving episodes an relies on interactions between the system and its environment. Defining the principles of interactive knowledge acquisition involve a careful analysis of the knowledge to acquire. In our approach, we focus on knowledge involved in the adaptation process for we believe it guides the whole CBR process. Consequently, we focus on domain knowledge and adaptation knowledge. Domain knowledge can be viewed as a set of constraints for the applicability of adaptation knowledge and is thus involved in the adaptation process.

Adaptation has been considered for long as the most important step of the CBR process but also as the most difficult one. Indeed, there exists, among the two main families of adaptation[24], a great variability of adaptation strategies. Though, this variability of the adaptation methods and of the associated knowledge is only apparent and we believe that it is possible to describe the reasoning process and the adaptation knowledge in a sufficient generic way to cover a great deal of the application domains of CBR. Following this principle, in [25] an unified strategy for the adaptation process has been proposed. This differential adaptation strategy relies on a modelization of the discrepancies between problem descriptors and on the definition of adaptation knowledge able to exploit theses discrepancies: adaptation knowledge allows to modify the solution descriptors to reflect the variations of the problems descriptors. This adaptation strategy fits well to domains that can be modeled thanks to numeric descriptors, although it works with symbolic descriptors as well. Nevertheless, it is sometimes difficult to assimilate every adaptation strategy to a differential strategy, in particular when descriptors are symbolic. To reason on symbolic descriptors, other strategies, such as conservative adaptation[20] are better suited.

In this paper, two complementary approaches of "adaptation" knowledge acquisition are proposed. Their combination would allow to benefit of the advantages of each one to support acquisition of symbolic and numeric knowledge and to improve the system's adaptation competencies. The issue of the modelization of the adaptation process and of the interactive knowledge acquisition is of main importance because it is a necessary condition to ensure a coherent knowledge engineering process integrated to the CBR cycle. Indeed, the modelization of adaptation knowledge (considered as the main knowledge of the cycle) decompartmentalize the various steps of the CBR process.

The next perspective for this work is to carry on the development of an integrated CBR tool allowing to solve problems thanks to past experiences and to refine knowledge guiding the reasoning process at the same time. As in KAYAK, this tool will emphasize the close link between the repair step and the memorization step of the CBR cycle and it will be able to acquire knowledge available in its environment. A significant work has to be done to build a tool usable by a human expert and to evaluate it. We also need to investigate the issues that have to be addressed to enlarge the spectrum of application domains for such a system.

5 Conclusion and perspectives

The main idea of this work is to discuss the principles of interactive knowledge acquisition in Case Based Reasoning. Thereby, we have studied the various knowledge used in a CBR system and we have shown why adaptation knowledge is of first importance in such systems. Then, we have described two applications performing interactive knowledge acquisition: FRAKAS and IAKA. These two prototypes are only at a early development stage and open issues are numerous. Perspectives for FRAKAS won't be detailed here as this is done in [12]. Concerning IAKA, a significant work has to be done to perform an evaluation of the prototype. Indeed, even if the prototype implements a full reasoning cycle integrating the interaction loop with the expert and the various steps of the adaptation knowledge acquisition process, an evaluation of the quality and the utility of the acquired knowledge must be done. An other ongoing research issue concerns the improvement of the principles detailed in IAKA. For example, IAKA only handles situations where an adaptation failure comes from a faulty adaptation operator. We still have to investigate situations where knowledge is missing in the knowledge base. To handle such failures, more elaborated interactions with the expert will be necessary. For both systems, we still have to perform real-world experiments. For FRAKAS , this entails work on the ergonomics of the interface to make it easily usable by domain experts.

The development of FRAKAS and IAKA has raised common open research issues. Most of them are linked to the unified view of adaptation we have and to the role played by domain knowledge in such a modelization. At this time, we are investigating the possibility of combining the two approaches in an integrated system centered on knowledge acquisition.

References

1. M. d'Aquin, F. Badra, S. Lafrogne, J. Lieber, A. Napoli, and L. Szathmary. Case Base Mining for Adaptation Knowledge Acquisition. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 750–755, 2007.
2. K. Hanney and M.T. Keane. Learning Adaptation Rules From a Case-Base. In I. Smith and B. Faltings, editors, *Advances in Case-Based Reasoning – Third European Workshop, EWCBR'96*, LNAI 1168, pages 179–192. Springer Verlag, Berlin, 1996.
3. M.M. Richter. The Knowledge Contained in Similarity Measures. Invited Talk of the First International Conference on Case-Based Reasoning, (ICCBR'95), 1995.
4. B. Smyth and M.T. Keane. Adaptation-guided retrieval: Questioning the similarity assumption in reasoning. *Artificial Intelligence*, 102(2):249–293, 1998.
5. D.B. Leake, A. Kinley, and D. Wilson. Case-based similarity assessment: Estimating adaptability from experience. In *Fourteenth National Conference on Artificial Intelligence*, pages 674–679, Menlo Park, CA, 1997. AAAI Press.
6. J. Lieber. Reformulations and adaptation decomposition. In *International Conference on Case-Based Reasoning - ICCBR'99*, Munich, Germany, 1999. LSA, University of Kaiserslautern.
7. S. Craw. Introspective learning to build case-based reasoning knowledge containers. In *Proceedings of the 3rd International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM 03)*, LNAI 2734, pages 1–6, Leipzig, Germany, 2003. Springer.

8. W. Wilke, I. Vollrath, K.D. Althoff, and R. Bergmann. A Framework for Learning Adaptation Knowedge Based on Knowledge Light Approaches. In *Proceedings of the Fifth German Workshop on Case-Based Reasoning*, pages 235–242, 1997.
9. S. Craw, N. Wiratunga, and R.C. Rowe. Learning adaptation knowledge to improve case-based reasoning. *Artificial Intelligence*, 170(16–17):1175–1192, 2006.
10. A. Cordier, B. Fuchs, and A. Mille. Engineering and Learning of Adaptation Knowledge in Case-Based Reasoning. In *Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management (EKAW-06)*, pages 303–317, 2006.
11. K.J. Hammond. Explaining and Repairing Plans That Fail. *Artificial intelligence*, 45(1–2):173–228, 1990.
12. A. Cordier, B. Fuchs, J.Lieber, and A. Mille. Failure Analysis for Domain Knowledge Acquisition in a Knowledge-Intensive CBR System. In *To appear in the proceedings of the 7th International Conference on Case Based Reasoning - ICCBR'07 (to appear)*, 2007.
13. B. Fuchs, J. Lieber, A. Mille, and A. Napoli. An Algorithm for Adaptation in Case-Based Reasoning. In W. Horn, editor, *14th European Conference on Artificial Intelligence - ECAI'2000*, pages 45–49, Berlin, Germany, 2000. IOS Press.
14. J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, 1993.
15. C.K. Riesbeck and R.C. Schank. *Inside Case-Based Reasoning*. Lawrence Erlbaum Associates, 1989.
16. J. Lieber and B. Bresson. Case-based reasoning for breast cancer treatment decision helping. In Enrico Blanzieri and Luigi Portinale, editors, *Advances in Case-Based Reasoning, 5th European Workshop, EWCBR 2000*, volume 1898 of *LNAI*, pages 173–185, Trento, Italy, September 6-9, 2000 2000. Springer.
17. K. Hanney and M.T. Keane. Learning adaptation rules from a case-base. In Lecture Notes In Computer Science, editor, *Proceedings of the Third European Workshop on Advances in Case-Based Reasoning*, 1996.
18. D.B Leake, A. Kinley, and D. Wilson. Multistrategy learning to apply cases for case-based reasoning. In *Third International Workshop on Multistrategy Learning*, pages 155–164, Menlo Park, CA, 1996. AAAI Press.
19. S. Fox and D.B. Leake. Using introspective reasoning to guide index refinement in case-based reasoning. In *Sixteenth Annual Conference of the Cognitive Science Society*, pages 324–329, Atlanta, GA, 1994.
20. J. Lieber. Application of the Revision Theory to Adaptation in Case-Based Reasoning: the Conservative Adaptation. In *To appear in the proceedings of the 7th International Conference on Case Based Reasoning - ICCBR'07 (to appear)*, 2007.
21. A. Aamodt and E. Plaza. Case-based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, 7(1):39–59, 1994.
22. B. Smyth and M.T. Keane. Retrieving Adaptable Cases. In S. Wess, K.-D. Althoff, and M.M. Richter, editors, *Topics in Case-Based Reasoning – First European Workshop (EWCBR'93), Kaiserslautern*, LNAI 837, pages 209–220. Springer, Berlin, 1994.
23. B. Fuchs, J. Lieber, A. Mille, and A. Napoli. A general strategy for adaptation in case-based reasoning. Technical Report RR-LIRIS-2006-016, LIRIS UMR 5205 CNRS/INSA de Lyon/University Claude Bernard Lyon 1/University Lumiere Lyon 2/Ecole Centrale de Lyon, 2006.
24. J.G. Carbonell. Learning by analogy: Formulating and generalizing plans from past experience. In R. S. Michalski and J. G. Carbonell and T. M. Mitchell, editor, *Machine Learning, An Artificial Intelligence Approach*, chapter 5, pages 137–161. 1983.
25. B. Fuchs, J. Lieber, A. Mille, and A. Napoli. An algorithm for adaptation in case-based reasoning. In W. Horn, editor, *14th European Conference on Artificial Intelligence - ECAI'2000*, pages 45–49, Berlin, Germany, 2000. IOS Press.