

Using Explanations to guide Adaptation

Béatrice Fuchs, Alain Mille, Benoît Chiron
 CPE-Lyon, LISA , 31 place Bellecour, 69 002 Lyon, France.
 e-mail : {bf,am,chiron}@cpe.ipl.fr, am@cpe.ipl.fr

Abstract

We present the adaptation process in a CBR application for decision support in the domain of industrial supervision. Our approach uses explanations to approximate relations between a problem description and its solution, and the adaptation process is guided by these explanations (a more detailed presentation has been done in [4]).

1 CBR and adaptation

In problem solving, CBR avoids complex inferences by reusing past problem solving episodes. We hope that if the current problem is similar to a previous one, then the solution of the past problem should be reusable for the current problem with a low adaptation cost.

As target and source generally always have some differences, the objective of the adaptation process is tending to focalize on these differences and to reflect them on the solution of the new problem. An adaptation process is necessary when the current problem is not strictly identical to a past one. The adaptation process is difficult to define and is the key step of the CBR process, because the quality of the obtained solution depends on it.

We present here a CBR application in the domain of industrial supervision where the adaptation process has the main role. First, we present a formalization of the CBR process, and we use it to describe our approach shortly. Then, we present the supervision domain, and what consist of supervision cases. Then, we summarize the retrieval step to see how retrieval and adaptation are connected. Finally we describe the adaptation process and adaptation knowledge.

1.1 Terminology

In order to have a precise adaptation concept, we need to formalize the CBR process. A good formalization has been given in [9] for analogical reasoning, that has been adapted for CBR in [6] and we will reuse.

In an intuitive way the CBR tries to approximate a complex reasoning with simple heuristics by completing a partially correct solution (fig. 1).

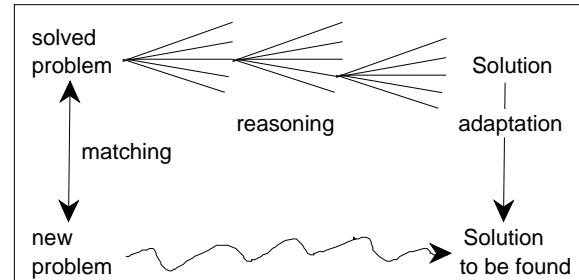


Figure 1: The problem solving process is approximated in CBR by the use of adaptation heuristics.

The reasoning process that yields the solution of a problem is unknown, but we suppose that there exists relations (known or not) between problem and solution elements. If we can know what are the elements of the problem on which the solution depends, we can hope that the elements of the problem will be predictive of the solution. If we don't know the relations between problem and solution, because there is a lack of knowledge or because the required inference is too complex, we can try to discover on what the solution elements depend in an old case, and try to apply, in a current case, an heuristic in terms of these dependencies to approximate the solution.

A is a problem to solve and B its solution. A *target* (A_0, B_0) is a problem to solve, and a *source* (A_i, B_i) is a solved problem.

There exists a **dependency** β between A and B if A is implied in the deduction of B by the relation β associated to U. (A, β, B) is a dependency description.

In the CBR process, (fig. 2) there is a matching process between problem descriptions. We call this matching α : $(A_i, \beta_i, B_i)\alpha(A_0, \beta_0, B_0)$.

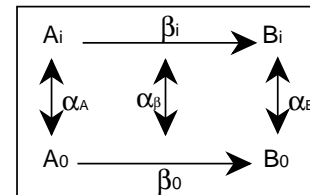


Figure 2: The relations between problem and solution of source and target problems in CBR systems

The use of *points of view* facilitates the reasoning process

by focusing on a subset of features or relations that are particularly relevant for the goal to solve.

The CBR process is generally composed of the following steps :

1. Build the problem A_0 to determine important information for the current problem to solve, and take into account the point of view. This can be achieved by different kinds of inference.
2. Identify possible sources which amounts to find a set of (A_i, β_i, B_i) depending on A_0 . The source's choice for the current problem is made by choosing first the problem description A_0 , and then by ordering and selecting the best sources to reuse.
3. The knowledge transfer from B_i to B_0 is the adaptation process itself.
4. The evaluation of the inference can be made in different ways : if a domain theory is available, but not used for reasoning due to a lack of efficiency, it can be used to evaluate the correctness of the obtained solution. If no domain theory is available, the only way to evaluate the solution is to test it in the real world.

Most systems modify the organization of the case base to take into account the occurred failures : they update indexes in order to avoid future failures. Recent approaches ([2]) use an explicit model of the CBR process to diagnose reasoning failures by introspection.

Michel Py ([9]) cites two reasoning strategies. We have added the third one. These strategies can be mixed in the CBR process to enhance the reasoning :

- a *dependency relations driven strategy* : $\beta_i \alpha_\beta \beta_0$. When dependency relations are known, they can be used to guide the matching process. In [1] a first analysis of a new problem is used to retrieve past problem solving episodes having the same initial reasoning decisions.
- a *description elements driven strategy* : A_i, A_0, B_i, B_0 . As dependency relations are generally not known, most CBR systems begin to retrieve previous cases by using a subset of problem descriptors of the new problem A_0 . In fact they are based on the assumption that these elements are relevant, i.e. somewhat the solution of the problem depends on these elements. The approach of S. Fox ([2]) is original because it tries to detect indexing failures after the adaptation step by using B_0 to search cases with a similar solution. The obtained cases are compared to the previously chosen one in order to validate and possibly refine indexing.
- a *matching driven strategy* : $\alpha_A, \alpha_\beta, \alpha_B$. When a set of cases has been retrieved, a case must be chosen among them. Generally a similarity measure is assessed and based on a matching between problem descriptions α_A .

The transfer of the solution of the old case in the adaptation step is done according to the matching between problem descriptions. This strategy is often used when the adaptation process focuses on discrepancies to perform modifications. The approach described in [10] uses α_A at retrieval in order to predict adaptation requirements. Other approaches [8] use α_A in order to find similar differences in previous adaptation episodes and to help choosing a good adaptation strategy.

1.2 CBR for helping the operator

In the context of decision support applied to industrial supervision, the reasoning process has been presented in [4].

We present a more general adaptation process here. A case is a rich and complex structure with multi-valued attributes representing operational knowledge tightly coupled to domain knowledge. The roles of the different elements are not always explicit, and their settings require to navigate within domain knowledge.

The operator has the initiative to ask for assistance when he needs help to manage the current situation.

2 Decision Support for Industrial Supervision

Industrial supervision covers a wide range of activities, from simple control feedbacks of some parts of a system to whole supervision of a complex and heterogeneous system. In our work we address the management of an unusual situation, where the operator has the main role.

In case of troubleshoot the operator must handle the situation and try to recover a normal operation. The operator has to navigate in the hypermedia information system to look for the needed information to manage the situation. Human-machine interface and tools play a growing role because they give a perception of the current situation.

The PADIM project aims at integrating multimedia and hypermedia facilities in order to allow non-linear dialogues with the operators to help him in his search for the best supervision strategy. A presentation of the project can be found in [4].

2.1 Domain Knowledge

Only few surface knowledge is used and whenever possible collected from existing databases. It has to do with the topology, functions and structures of the system to be supervised, that is the *supervision object* concept ([7]) it's a generic term which means an object to focus on, a goal of the supervision. Supervision objects are represented in hyperdocuments. This knowledge is tightly coupled to supervision cases and widely used to perform adaptation.

A Knowledge representation system has been implemented to allow the representation of knowledge and to provide easier interactions with the environment (that

is the information system), and dig into automatically databases ([3]).

Some well-known situations are also modelled. CBR allows to acquire new supervision strategies that are encountered at operating stage and that were not previously modelled. This allows to acquire and share experiences among operators.

2.2 Supervision cases

An atomic supervision case (A,B) is a simple *supervision episode* i.e., a description of a starting *supervision situation* and a *supervision environment* to manage it (figure 3).

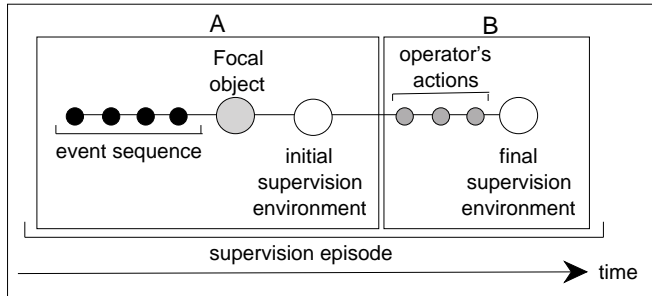


Figure 3: A supervision episode is composed of a situation description (A) and a solution (B). The situation description contains an event sequence, the focal object taken by the operator and the initial supervision environment. The solution is another supervision environment that has been built and maybe modified by the operator.

A starting supervision situation (A) is described by the sequence of already-occured events and by the corresponding supervision contexts. The events sequence includes what occurred in the system and the actions done by the operator. A kind of screening is used in order to eliminate redundant events, and keep only high level events.

A supervision environment describes the *system's state*, the *operator's state*, the *operating context* and is viewed through hypermedia documents.

The whole system is characterized by a small number of states as : normal operation, degraded operation, breakdown stop, maintenance stop, etc...

The operator's state determines a user profile and operating conditions. (newuser / experimented, night / day, team / alone, ...). The operating context is the set of supervision objects represented through hyperdocuments currently on screens. Each supervision object is characterized by its current state.

The solution (B) is a new supervision environment, i.e. a set of hyperdocuments that are relevant for the current situation, and that are organized in an hypermedia interface (using the NeXT'IM hypermedia supervision system). Every supervision object plays an implicit role in the current situation, that has to be discovered.

Supervision cases encode additional information relative to the relevance of the solution elements to the context

(explanations, $\hat{\beta}$), and conclusions of the adaptation process that are used for learning to refine the retrieval and matching processes.

2.3 Building the problem description

First, a supervision case is built by the reasoner when the operator asks for assistance. The operator has to choose a point of view on the system, that is, the main supervision object. This point of view is used to retrieve cases in a discrimination network.

2.4 Case Retrieval

First, the retrieval process uses the general context and the point of view taken by the operator to select potential cases to examine. Two similarity measures are then used : a first one compares event sequences and a second one compares the involved concepts. The matching of concepts (α_A) is used as a basis for adaptation (figure 4).

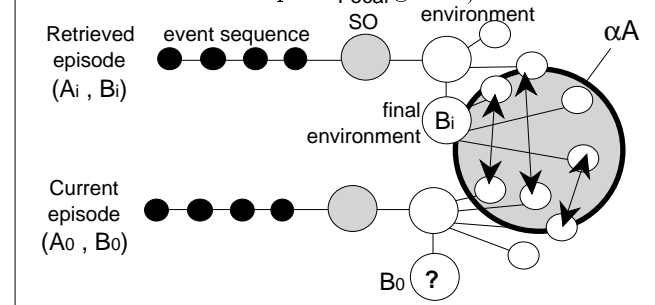


Figure 4: The matching process between supervision objects of the supervision environments in source and target cases.

Facing a problem, we have to build a sub-model for the current situation, i.e. a sub-graph of the conceptual graph of domain knowledge. Thus, we have to analyze the relations between the problem description and the domain knowledge in order to build such a sub-graph. Supervision objects are characterized by their relations with problem elements and domain knowledge.

2.5 Case Adaptation

The problem can be expressed in the following way : A description of the current situation being given, what is the most relevant information to be proposed ? In industrial supervision for example, the situation (A) is described with an event sequence that has led to the current situation, the supervision tools and hyperdocuments currently viewed by the operator, and the point of view the operator wants to have on the system. The solution (B) is a set of hyperdocuments that are relevant for the current situation, and that are organized in an hypermedia interface (using the NeXT'IM hypermedia supervision system). Every information is named *supervision object* and has a role according to the current situation, that has to be discovered.

The adaptation process is composed of two steps:

1. Firstly, the situation is analysed in order to determine which information is relevant to the current situation and must be represented in hypermedia interfaces.
2. Secondly the obtained information is gathered in hypermedia documents to be presented to the operator. The hyperdocuments are either existing ones or adapted from existing ones. Some rules guide interfaces construction.

The adaptation process begins by setting dependencies $\hat{\beta}$ between solution elements and the descriptors of the situation in the retrieved cases. Dependencies are paths linking solution elements to other elements that are present in the sub-model previously built (figure 5). For example, "Pressure PR1 of container R1 is explained by the occurrence of the event pressure too low on the container R1". Real dependencies are not known but we try to approximate them using $\hat{\beta}$ that approximate β . Most CBR systems use (explicitly or not) a kind of $\hat{\beta}$ in order to apply adaptation heuristics (that is adaptation knowledge) to approximate the real dependency.

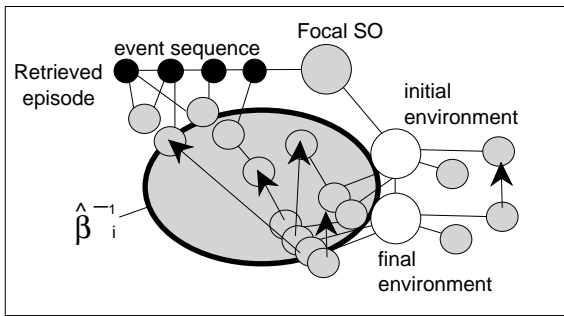


Figure 5: Explanation of the elements of the solution in the retrieved case with elements of the problem description

We call explanation this dependency because it explains or justifies the presence of the object in the solution. We can notice that usually multiple explanations are possible. The explanations are approximations of the relation β_i between elements of B_i and A_i . They are mapped into A_0 to determine the solution elements B_0 (figure 6). The mapping of explanations is in fact a matching process α_β between explanations.

Then, solution elements of B_0 are organized in a set of interface objects that are shown to the operator. Some reorganizations may be necessary if ergonomic constraints are not verified.

Some explanations may not match in the source and target case, and explanation adaptations are required sometimes.

The use of multiple cases allows to choose an explanation among a set of possible ones in order to enforce its plausibility. It allows to do case-based substitution and enhance

performance. Cases are chosen to have the best coverage of the current case with the smallest possible number of retrieved cases.

A general search strategy begins first with causal relations (i.e. events), and if it does not succeed, other kinds of relations are used such as part-of, etc ...

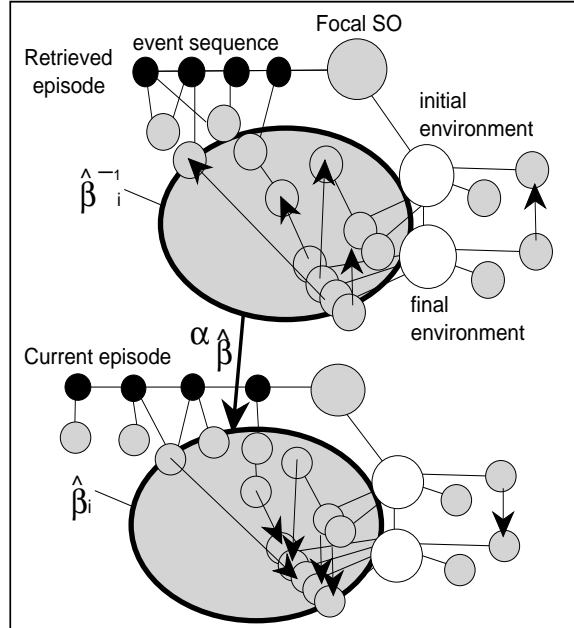


Figure 6: Mapping the explanations in the current case to determine supervision objects that will constitute the new supervision environment

2.6 Adaptation Knowledge

Adaptation Knowledge is a general plan composed of several steps. Steps can be decomposed in sub-steps depending on the adaptation tasks that have to be done. Strategic adaptation knowledge guide the different steps, and more specific knowledge yields basic transformations and substitutions. A unifying framework is discussed in [8].

3 Adaptation and learning

As seen before, reasoning failures may occur, and the adaptation process will be used to enhance the retrieval and matching process.

Events are filtered according to their importance level. This importance level can be further refined by counting the number of times an event is used to explain information of case solutions. Then, important events, that is events with a high importance level can be used to retrieve cases and help the operator to determine what would be the best point of view. Finally, the event similarity measure can be refined by weighting events.

4 Conclusion

In CBR systems, case adaptation is the kernel of the process. Its complexity depends on the domain, the kind of area of application, and the differences between current and past cases.

The complexity of adaptation implies the acquisition of adaptation knowledge. Thus, the knowledge acquisition problem that sometimes justify the use of the CBR paradigm is raising again. The use of learning techniques as sketched in [5] allows to learn adaptation using a CBR process. We think that learning is the only way to cope with adaptation because this enhances the quality and the flexibility of the process.

The approach of B. Smyth ([10], [11]) avoids complex adaptations by an attempt to evaluate the cost of the adaptation process at the retrieval step. The reused case is then chosen according to this cost. Though this approach is interesting, is the criteria realistic ? i.e. is the cost of the application of an adaptation knowledge predictable ? Furthermore, when the CBR system is novice, the system must be creative. We must not forget that adaptation is absolutely necessary because retrieved cases are never matching exactly the current one, and there is no warranty that good indices have been used for retrieval.

It is obvious that when the case-base contains few cases, the probability to find no adaptable case will be important, and it will be necessary to do lots of modifications to reuse a case. These must be reused and eventually enhance the process by refining it for a further better behaviour.

All current approaches use adaptation knowledge, but there are uncertainties concerning the control aspect of the adaptation process, and no approach is unifying it. Our approach can be used in decision support but is probably not usable in other domains with specific constraints. Nevertheless, it is clear that the achievement of a goal under constraints guide the need for adaptation.

CBR uses different reasoning, learning modes and techniques such as classification, explanation, ... and we think that a general reasoning framework integrating CBR must take advantage of existing reasoning tasks. A CBR system should be designed by reusing generic reasoning methods. These are the main keys to facilitate the definition of specific CBR applications by refinement of a general opened framework.

References

- [1] Jaime G. Carbonell, *Derivational Analogy : A theory of reconstructive Problem solving and expertise acquisition*, volume 2, 371–391, Morgan Kaufmann, 1986.
- [2] Susan Fox and David B. Leake, ‘Using introspective reasoning to guide index refinement in case-based reasoning’, in *Proceedings of the sixteenth annual conference of the cognitive science society*, (1994).
- [3] Béatrice Fuchs and Alain Mille, ‘Objective c extensions for knowledge representation’, in *Object-Oriented Computing in the Natural Sciences, OOCNS’95 : Second Multidisciplinary International Workshop of Scientific Computing Environments*, IMAG, Grenoble, France, (21–24 November 1995).
- [4] Béatrice Fuchs, Alain Mille, and Benoît Chiron, ‘Operator decision aiding by adaptation of supervision strategies’, in *Proceedings of the First International Conference on Case-Based Reasoning - ICCBR-95*, pp. 23–32, Sesimba, Portugal, (23–26 Octobre 1995). Lecture notes in Artificial Intelligence, Springer Verlag.
- [5] David B. Leake, ‘Learning adaptation strategies by introspective reasoning about memory search’, in *Proceedings of the AAAI Workshop on Case based reasoning*, ed., AAAI Press, pp. 57–63, MenloPark, CA, (1993).
- [6] Alain Mille, ‘Modèle conceptuel du raisonnement basé sur les cas’, in *4ème séminaire Français sur le raisonnement à partir de cas*, ed., Isabelle Bichindaritz, Université René Descartes, Paris V, (10-11 Avril 1995).
- [7] Alain Mille, *Raisonnement basé sur l’expérience pour coopérer à la prise de décision, Un nouveau paradigme en supervision industrielle*, Thèse de doctorat de l’université Jean Monnet, Saint-Etienne, Décembre 1995.
- [8] Alain Mille, Béatrice Fuchs, and Olivier Herbeaux, ‘A unifying framework for adaptation in case-based reasoning’, in *workshop on Adaptation in Case-Based Reasoning, 12th European Conference on Artificial Intelligence, ECAI-1996*, Budapest, (1996).
- [9] Michel Py, ‘Un modèle conceptuel du raisonnement par analogie’, *Revue d’intelligence artificielle*, 7(1), (Mars 1994).
- [10] Barry Smyth and Mark T. Keane, ‘Retrieving adaptable cases. the role of adaptation knowledge in case retrieval’, in *Topics in Case-Based Reasoning*, eds., M. M. Richter, S. Wess, K.-D. Althoff, and F. Maurer, pp. 76–81, University of Kaiserslautern, (November 1993).
- [11] Barry Smyth and Mark T. Keane, ‘Retrieval and adaptation in déjà vu, a case-based reasoning system for software design’, in *Adaptation of Knowledge for Reuse : A 1995 AAAI Fall Symposium*, pp. 228–240. AAAI Press, (1995).