A unifying framework for Adaptation in Case-Based Reasoning

Alain MILLE, Béatrice FUCHS and Olivier HERBEAUX
CPE-Lyon, LISA , 31 place Bellecour, 69 002 Lyon, France. e-mail : {am,bf,oh}@cpe.ipl.fr

1 Introduction

The CBR team of the LISA is involved in several applied research projects based on the CBR paradigm. These applications use adaptation to solve the specific problems they face. So, we have capitalized some experience about how can be expressed and formalized adaptation processes. The bibliography on the subject is quite important but demonstrates a lake of formalism. At most, there exists some classifications about different types of adaptation. A session of DARPA conference 89 was devoted to adaptation strategies [1] [5] [10] [12] [14] [25] with a lot of examples but no generalization nor formalism. Nevertheless there was a first attempt of [14] to propose standard strategies to adapt explanations.
Transformational Analogy and Derivational Analogy proposed by [4] were first general methods generally accepted. Janet Kolodner [15] presented several types of adaptation in her book with an attempt to stress on which kind of information is needed to process adaptation. We have summarized this classification in [17]. Several French researchers have indirectly presented adaptation as a problem of knowledge transfer in Analogical Reasoning (Reasoning based on examples) [3], [21], [6] and other important contributions have been made about different facets of adaptation such as :

- Memory Search oriented adaptation by David Leake [16, 2]
- Adaptability driven adaptation by Padraig Cunningham, Barry Smith and Mark Keane [22] [23](Trinity College, Dublin)
- Adaptation guided by constraints expressions by Ian Smith and Boi Faltings [24] [13], etc.

Our proposal is based on an analysis of these previous works and on our proper experience.

We have developed (or are developing) several projects which involve adaptation as a central point. First, we present three of these projects, stressing about adaptation process, then we develop a unifying framework to represent adaptation as generic tasks at different levels of abstraction of knowledge, and finally we propose some ideas for implementations of the framework within a CASE system.

ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W. Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

2 Examples of CBR systems using adaptation

2.1 Application 1 : Interactive programming of a digestion process (microwave devices)

[20] This application is an industrial one. The problem to solve is the design of a digesting program. A digesting program is described in figure 1 :

STEPS	1	2	I	N
PRODUCT	H2SO4	HNO3		HCL
VOLUME	5	5		7
SPEED	2	4		5
HEATING	-	65		-
TIME	4	10		3
DRYING	NO	YES		NO

Figure 1. A digesting program for a digester

Microwave ovens are sold to laboratories that have to digest various samples of material in order to prepare them for different types of analysis (analysis by spectrometer for example). Ovens are fully automated. Automatic product injections, heating control, and time scheduling of the different steps are programmed. So, a digestion can be seen as a kind of recipe, or a kind of plan at a higher level of abstraction. An oven is sold with its "cooking-book" and the CBR system helps to build new programs from known ones. Similarity follows the semantic of the problem and is based on a rough estimation of the composition of the sample, the type of analyzer and the weight of the sample. Adaptation of the target case (the case we want to design) is made from several source cases (and possibly one of these cases can be a build-in prototype). The final choice of source cases is decided by the user beyond the five first best cases and a prototype case. The process of adaptation is summarized in the diagrams (figure 2 and figure 3) presented in Appendice. The framework to guide adaptation is a very similar one at two different levels of abstraction of the problem. First, a kind of plan is elicited from the source cases and is adapted to the new context, and then the different steps of the plan are adapted. Such a hierarchical processing of adaptation is presented in [24].
The user can repair directly the result of the adaptation. Validation of a new case is made under the responsibility of an

expert who has to verify that the digestion had well prepared the sample for the analysis.
Analysis and design of the application has been aided by Remind (just a little bit!) and Prolog prototypes. The final application is written in C++ under Windows environment. Similarity computing and adaptation process are developed as separated DLL. Cases are managed in a database.

2.2 Experience Based Reasoning in coo-operative decision helping

[18] [19] [9] In this application, the problem to solve is : How to find a good way to manage a new situation through industrial supervision tools ? Driven by an industrial company and funded by the French Minister of Industry, the project has permitted to propose a modelization of the supervision task for the designing stage and a cooperative decision helping system for the operating stage. Operators' experience is capitalized in order to retrieve past supervision situation to adapt to a new context. Adaptation is guided by explanation as developed in an other paper [8]. The best case is retrieved on the basis of evenemential and conceptual similarities in cooperation with the operator. First step is to find which kind of explanations existed in the retrieved case. Explanations are semantic links between items present in the solution and items present in the description of the case. Similar explanations are searched from the description of the new case. When established, these explanations are used to find in memory possible items that are linked to them and that obey to the known context. A representation adaptation is processed for each item according to heuristics.
In other words, the adaptation process is the following :

- eliciting explanations in the retrieved case,
- eliciting similar explanations in the new case,
- eliciting solutions items of the new case that fit elicited explanations and know constraints
- for each item, eliciting the best representation according to associated heuristics.

The proposed solution can be repaired by the operator and a validation process (expert driven) allows to keep the case as a new experience.
An extension of Objective C has been developed [7] to represent the whole system through frames. The prototype is developed in Nextstep environment.

2.3 Adapting Adaptation Cases to design new products

[11] The third application is specifically adaptation oriented. It concerns the production of industrial rubber. Each product is characterized by its mechanical, physical and chemical properties. A product is described by the different ingredients used and by the associated process to get it with the wanted properties. Moreover, as the development engineers used previous product descriptions as guides to a new one, the documentation noticed what product was used and how it was modified to get the new one. So, adaptation was the main task of the engineers. The application is developed in order to :

- manage products with their descriptors as product-cases,
- retrieve previous products on a semantic similarity concerning their properties,
- manage adaptations "sessions" as adaptation-cases,
- link product-cases and adaptation-cases through the gaps of properties reduced by the adaptation plan,
- provide the easier adaptation-case to use for a new product from a close previous one. Each adaptation-item of the adaptation-case is presented with the deviation to reduce and the method to do it.

The engineer can complete the adaptation which will be stored as a new adaptation-case. At a higher level of abstraction, the adaptation process can be described by the following steps :

- eliciting a type of adaptation-plan from the nearest known product,
- choosing the easiest adaptation-plans according to their ability to reduce the gaps between source and target cases (from the point of view of their properties).
- keeping adaptation items from different adaptation-plans (if necessary)

So, the choice of adaptation is memory-driven and that approach seems very close to Leake's one [2] The prototype of the application is pc-based and is developed with the standard Microsoft development tools. Knowledge is managed through a database.

3 A general model of adaptation

Applications presented above are very different from each others. Is it possible to find a general model of adaptation that fits each one ? In problem-solving, (but is there anything else than problem solving in CBR applications ?), it seems that a general framework for adaptation can be established on very simple ideas. First, as sketched in 4, any solution can be described as an ordered (or not ordered) list of ITEMS. An ITEM of solution, in turn can be described in the same way. At each level of description, ITEMS have different semantics. The lowest level is constituted of ATOMIC ITEMS described by a label and its associated values. The ultimate adaptation concerns the value of an item. Modifying an ATOMIC ITEM consists to substitute the source value by a target value. Modifying a list of ITEMs consists to substitute, to add or to delete ITEMS in the list. To generalize a step further, we can see adding and deleting as special cases of substituting. If we name $\gamma(\mathrm{Is}, \mathrm{It}, \delta(\mathrm{Cs}, \mathrm{Ct}))$ the function of substitution of an item(Source) Is by an item(Target) It guided by the function δ of reduction of difference of the context(Source) Cs to the context(Target) Ct, then addition function is $\gamma(\mathrm{Nil}, \mathrm{It}, \delta(\mathrm{Cs}, \mathrm{Ct}))$ and the suppression function is $\gamma(\mathrm{It}, \mathrm{Nil}, \delta(\mathrm{Cs}, \mathrm{Ct}))$ where Nil is the Null ITEM. That approach is illustrated in Appendice, figure 5.
So, the adaptation process can be presented as a recursive process of substitution at each level of granularity of the solution. Substitutions at each level can be independent or not. If they are not independent these substitutions have to be considered as only one big substitution at the higher level. The complexity of such a substitution can be important [24] but can, in turn be considered as a new problem-solving situation, and adaptation cases can be used to make it easier[11, 16]. The

substitution methods depend of the type of item, and has to be applied in order to reduce a function which evaluates the level of discrepancy between the Source Item with the context of the target case.

4 Discussion : Toward generic methods in Adaptation

In order to develop CBR applications needing an adaptation process, we think that main tasks are:

- describing Cases (solutions) as sketched above, eliciting ITEMS.
- for each kind of ITEM, eliciting the substitution method (possibly generic) with its associated function of discrepancy reduction (possibly generic)
- elaborating the similarity measure based on an adaptability estimation (guided by the elicitation of discrepancy reduction ?).
- preparing a learning strategy guided by the such an adaptability "measure".

Some "generic" substitution functions can be listed :

- concept substitution guided by explanation similarity ,
- value substitution guided by interpolation,
- value substitution guided by constraint satisfaction,
- concept substitution guided by memory search,
- etc..

So, a general framework for adaptation can be the starting point of a new generation of CBR tools, specifically oriented on problem solving. We are working on a more precise expression of the sketched framework in order to be able to discuss on concrete material with other researchers.

REFERENCES

[1] Ralph BARLETTA, 'Case adaptation in autoclave layout design', in *Workshop on case-based Reasoning, DARPA 89*, pp. 203–207, (1989).

[2] David B.LEAKE, 'Representing self-knowledge for introspection about memory search', in *Proceedings of the AAAI Spring Symposium on Representing Mental States and Mechanisms*, p. 5, (1995).

[3] Louis BOURELLY and Joel SAVELLI, 'A propos des ambiguités du raisonnement analogique', *Techniques et Sciences Informatiques*, **11**(6), 39–65, (1992).

[4] Jaime G. CARBONELL, 'Derivational analogy: A theory of reconstructive problem solving and expertise acquisition', in *Readings in Knowledge Acquisition and Learning*, 727–738, G Buchanan et David Wilkins, (1986).

[5] Tim CONVERSE, Kristian HAMMOND, and Mitchell MARKS, 'Learning modification rules from expectation failure', in *Workshop on case-based Reasoning, DARPA 89*, pp. 110–114, (1989).

[6] Daniel COULON, JF BOISVIEUX, Louis BOURELLY, Laurent BRUNEAU, Eugene CHOURAQUI, Jean-Marie DAVID, Cheng-Ren LU, Michel PY, Joel SAVELLI, Brigitte SEROUSSI, and Christel VRAIN, 'Le raisonnement par analogie en intelligence artificielle, formalisation, applications', in *Actes des troisièmes journées nationales du PRC-GDR-IA*, pp. 45–88, (1990).

[7] Béatrice FUCHS and Alain MILLE, 'Objective c extensions for knowledge representation.', in *OOCNS'95 : Second Multidisciplinary International Workshop of Scientific Environments, Imag, Grenoble, France*, (1995).

[8] Béatrice FUCHS and Alain MILLE, 'Using explanations to guide adaptation'. proposed as a contribution to the Workshop on Case Based Reasoning - ECAI96, this issue, 1996.

[9] Béatrice FUCHS, Alain MILLE, and Benoit CHIRON, 'Operator decision aiding by adaptation of supervision strategies', *Lecture Notes in Artificial Intelligence vol 1010, First International Conference, ICCBR95, Sesimbra, Portugal*, 23–32, (1995).

[10] Dedre GENTNER, 'Finding the needle: Accessing and reasoning from prior cases', in *Workshop on case-based Reasoning, DARPA 89*, pp. 137–142, (1989).

[11] Olivier HERBEAUX, 'Rapport interne sur l'avancement du projet interep'. Rapport confidentiel, 1995.

[12] Thomas R HINRICHS, 'Strategies for adaptation and recovery in design problem solver', in *Workshop on case-based Reasoning, DARPA 89*, pp. 115–118, (1995).

[13] Kefeng HUA, Boi FALTINGS, and Ian SMITH, 'Cadre : Case based geometric design'. to be published in AI in Engineering, 1996.

[14] Alex KASS, 'Strategies for adapting explanations', in *Workshop on case-based Reasoning, DARPA 89*, pp. 119–123, (1989).

[15] Janet KOLODNER, *Case Based Reasoning*, Morgan Kaufman Publishers, 1993.

[16] David B. LEAKE, 'Learning adaptation strategies by introspective reasoning about memory search', in *AAAI93 Workshop on Case-Based Reasoning*, pp. 57–63, (1993).

[17] Alain MILLE, 'Modèle conceptuel du raisonnement basé sur les cas', in *4ème Séminaire Franais sur le Raisonnement Partir de Cas, Paris*, pp. 40–52, (1995).

[18] Alain MILLE, *Raisonnement basé sur l'expérience pour coopérer la prise de décision. Un nouveau paradigme en Supervision Industrielle ?*, Ph.D. dissertation, CPE-Lyon et Univesité Jean Monnet, Saint-Etienne, France, 1995.

[19] Alain MILLE, Jean Marc CHARTRES, Eric NIEL, Béatrice FUCHS, and Benoit CHIRON, 'Intelligent workstation for immediate decision helping in process supervision.', *European Symposium on Computer Aided Process Engineering 5, Bled, Slovenia*, (1995).

[20] Alain MILLE, Jean-Louis DI-MARTINO, and André MICHEL, 'Adaptation : the key-point in case based reasoning. case study : Digester programming helping', in *Workshop on practical developments strategies for industrial strength Case Based Reasoning applications, 16th International Conference on Artificial Intelligence, IJCAI'95, Montreal, Canada*, (1995).

[21] Michel PY, 'Un modèle conceptuel de raisonnement par analogie', *Revue d'Intelligence Artificielle*, **8**(1), 63–99, (1994).

[22] Barry SMITH and Pádraig CUNNINGHAM, 'Complexity of adaptation in real world case-based reasoning systems', in *The sixth irish conference on Artificial Intelligence and Cognitive Science*, pp. 228–240, (1993).

[23] Barry SMITH and Mark T. KEANE, 'Retrieving adaptable cases: the role of adaptation knowledge in case retrieval', in *EWCBR 93, Otzenhauzen, Germany*, pp. 76–81, (1993).

[24] Ian SMITH, Ruth STALKER, and Claudio LOTTAZ, 'Interactive case-based spatial composition'. submitted to AID 96, Stanford University, 1996.

[25] Roy M. TURNER, 'Case-based and schema-based reasoning for problem solving', in *Workshop on case-based Reasoning, DARPA 89*, pp. 341–344, (1989).

APPENDIX

In the digestion application, the first adaptation concerns the plan level of the solution, as sketched in the following diagram :

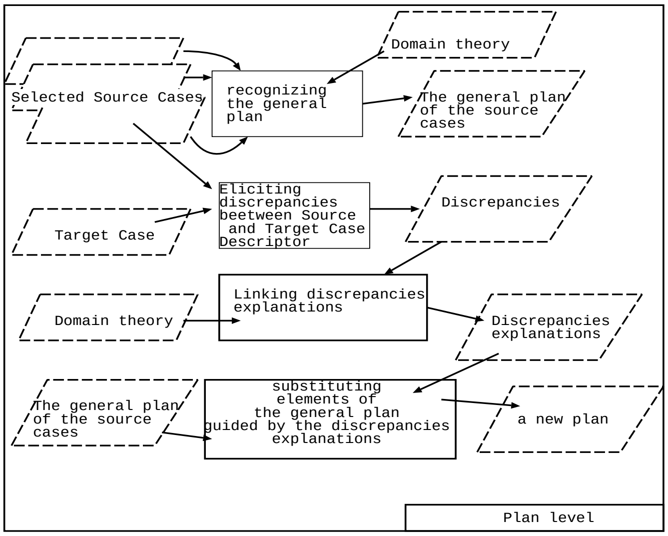

Figure 2. Adaptation process for a digesting program: the plan level

On the basis of the adapted plan, different steps are adapted according to simple rules of substitution (interpolation and constraint satisfaction) :

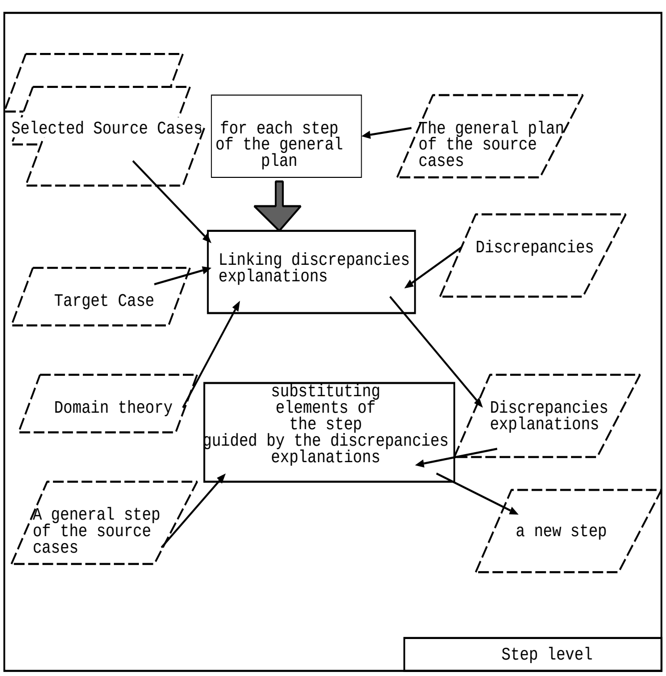

Figure 3. Adaptation process for a digesting program: the step level

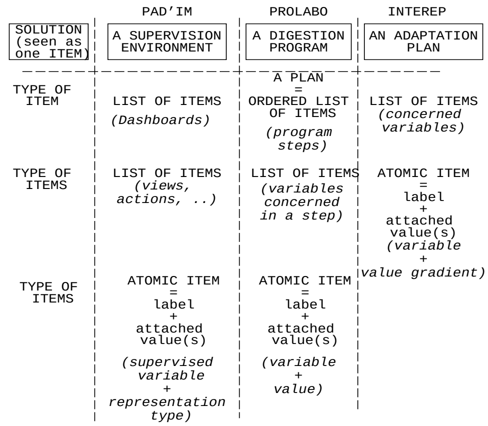

Figure 4. General description of a solution in problem-solving context

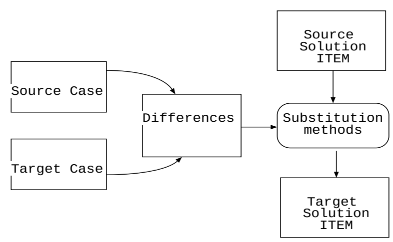

Figure 5. General substitution process