Reuse Cases when Doing Financial Case-Base Reasoning with Respect to Adaptation

Jürgen Hönigl
Institute for Application-Oriented Knowledge Processing
Johannes Kepler University
Linz, Austria
juergen.hoenigl@jku.at

***Abstract*—Case-Based Reasoning (CBR) applies past experience to solve new problems with suitable solutions. This approach presents overloading queries to adapt solutions if necessary. Subpar solutions have to be adapted within a CBR cycle before retaining them to keep a good quality of the case base. Dealing with missing values can be seen as previous step to avoid unnecessary adaptations. Integrate efficient and useful adaptations can be seen as really interesting and challenging task when considering the full CBR methodology. The common CBR principle -similar problems are having similar solutions- can be seen as a rather good point of start when developing an adaptation feature. An adaptation concept and first experience are presented within this paper.**

***Index Terms*—Adaptation, Case-Based Reasoning**

I. Introduction

This paper presents an approach for adaptation of cases. The main goal was achieving a concept, proof the feasibility and get first results which was divided into several sections. Adaptation of cases will be mainly seen as complicated in comparison to retrieve cases because the retrieve step can be clearly divided into different parts such as the case base, a connection between the case base and the CBR system and suitable similarity measures. An adaptation feature depends on the applied domain. For instance, CHEF was using modification rules (change ingredient) and object critics (e.g. cooking time) to modify a cooking recipe namely BEEF-WITH-GREEN-BEANS to BEEF-AND-BROCOLLI. [2] Within the domain regarding this approach, another adaptation process will be used. Similarity measures and queries are suitable for processing different loan applications with numerical and categorial attributes. The similarity value between loan applications can be used for the adaptation when remember the CBR principle that similar problems are related to similar solutions.

Firstly, a brief overview about Case-Based Reasoning will be provided to demonstrate the R^4 model by Aamodt and Plaza. [1] The section Previous Work will briefly introduce associations and similarity measures. Definitions of the case base will be shown which are related to the adaptation of cases. Following two sections are providing the core of this work in progress paper. An adaptation of a case requires consideration of possible missing values which is presented within the next section. Then an adaptation concept will be shown - different queries can achieve different results regarding the precision of relevant and retrieved cases. Then notes regarding first experiments and the concept of evaluation are demonstrated. The manual evaluation by teacher provides a guarantee concerning the quality of the case base. It can be used as a post-condition to the adaptation process. The conclusion and future work are presented at the end.

II. Case-Based Reasoning in a Nutshell

The origin of CBR was given within the research of cognitive science. Schank provides 1982 with his work an approach of Episodic Memory Organization Packets (E-MOPs). [3] CYRUS was a prototype by Kolodner and used meetings and talks by United States of America politician Cyrus Vance to apply E-MOPs to a real scenario. [4] An E-MOP contains a content frame (also known as norm) which stores common information like place, people and subject of a meeting and informations concerning relations to other episodes if necessary. E-MOPs are using a tree-like structure to connect different episodes. In 1994 Agnar Aamodt and Enric Plaza introduced a process model of the CBR cycle which was commonly called the R^4 model. [1] The process involved in this model can be represented by a schematic cycle containing the four R's, namely *Retrieve, Reuse, Revise and Retain*. First, cases are retrieved from the case base which are similar to a new given problem. The old case with a solution will be reused and modified if necessary, an evaluation of the solution will be handled in the Revise step and finally a new case complements the knowledge base in the Retain phase. According to Janet Kolodner a case can be defined as: *"(i) a situation and its goal, (ii) the solution and, sometimes, means of deriving it, (iii) the result of carrying it out, (iv) explanations of results, and (v) lessons that can be learned from the experience."* [5] Anyway, Kolodner also stated that a case can be seen as a *"contextualized piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of the reasoner"*. [5]

In CBR, we distinguish three different approaches: conversational, textual and structural. The conversational approach has the intention to provide solutions for many recurring simple problems. Predefined phrases -such as 'Have you tried to turn it off and on again ?' in first instance- will support a user to obtain a solution. These supporting phrases will be shown in the order of their importance for the given new

 ISBN: 978-1-61208-254-7

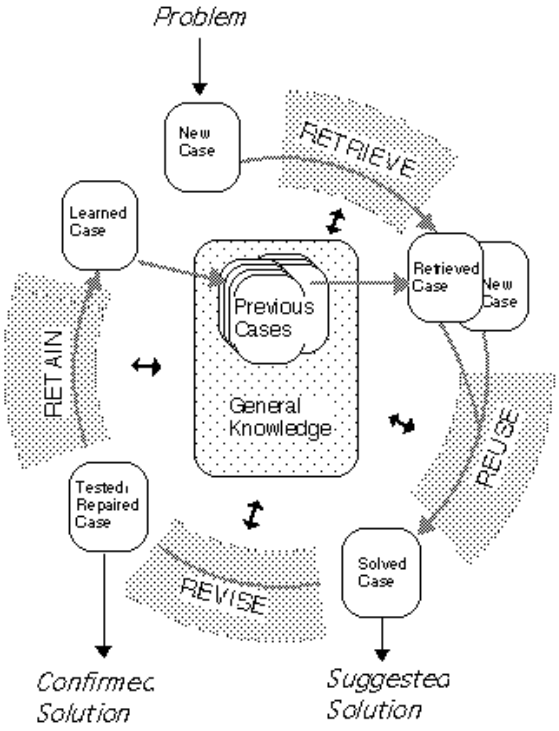

Fig. 1. R^4 model [1]

problem. The case base will be manually organized by the developer, while questions and phrases will be sequentially asked according to a decision tree which must be maintained when adding a new case. The textual CBR approach will be used for many documents which were analyzed concerning their content. The case base should not be greater than a couple of hundred cases, each case containing a short description with three lines. This approach should be aware of synonyms and associations between different terms. The structural CBR approach covers systems which are using a domain model. Therefore, predefined attributes and their representation should be chosen at the beginning of the modeling process. [6]

III. Previous Work

Different issues were researched such as association models and similarity measures. The gained knowledge of the association models were used to model a case base will be regularly used within the retrieve and retain step of the R^4 model. [1] Associations were obtained with the Hotspot algorithm of WEKA (Waikato Environment for Knowledge Analysis). [7] It is a target attribute driven algorithm which clearly presents the associations for a given target in a decision tree like structure. Arguments for this algorithm are the supported segment size of the data, the branching factor (on the top level) and the target attribute. The associations were partially published within [8]. Similarity measures are an ongoing topic and under development. According to the R^4 model they will be mainly used within the retrieve step. [1] Within the following lines they will be briefly described. The distance between two loans regarding the amount can be calculated when using the attribute amount, but to get the nearest cases -within the retrieve step- more attributes has to be considered such as age, purpose and credit history of a customer. Using a similarity measure will be more suitable in comparison to simple distances for these kind of loan cases. Similarity measures which encapsulates support for different attributes were modeled and will be tested due to different aspects such as non-negativity and range, reflexivity and positiveness. Weights will be used in addition if the will improve the functionality of these measures. A few pitfalls were avoided such as the difference between a distance metric and a similarity measure. For instance, distance=0 is equal to similarity=1. Both distance value and similarity value must be greater or equal than zero, but the range of a similarity value ends with 1.

IV. Definition of Problem, Solution and Case

First definitions were made which was a pre-condition for further work regarding the proof of concept. An example for a minor query would be following given problem which contains six attributes which are describing a loan application of a customer.

Problem = {Age, Credit Amount, Credit History, Duration, Income, Purpose}

A query can be extended with an attribute such as guarantors of the debtor. Extending a query towards the prototype will be suitable when the desired data of a customer is available within her or his loan application.

Problem = {Age, Credit Amount, Credit History, Duration, Income, Other Debtors Guarantors,Purpose}

A solution can be abstractly defined with two parts.

Solution = {Cost Factor, Recommendation}

The cost factor can be divided into different elements such as a percentage value of the predicted repayment, an absolute value concerning the amount of an assumed financial loss and a nominal value (e.g. 1 - 5) which describes the cost of this loan. The recommendation can be divided into subparts like a solution quality factor which will be given within the evaluation procedure within the revise step of the R^4 model and a real recommendation regarding the loan query of the customer. In the most efficient representation, the loan recommendation would be a boolean value which will be suitable on the top level for an employee of a financial institute. A case will be a triple of three elements in the minimal form.

Case = {Problem, Solution, Notes}

However, further allocation will be made when developing the prototype. For instance, notes can be used as a relation or as a character large object attribute. These definitions were partially published within [8].

V. Consideration of Missing Values

Unfortunately missing values can affect processing a case within different tasks such as retrieve a case from the case base and reuse a case, for instance. During the work on associations it was obvious that the attribute income was not explicit mentioned within the data definition of the German

 ISBN: 978-1-61208-254-7

credit data set. [9] However, income was chosen as a possible attribute for new problems (or queries) which are submitted to the prototype because newer requests can and should provide this information which can be used for the pre-processing and reasoning. The Oracle Database provides a rather good function namely nvl (null value substitution) but for certain cases an implementation concerning a given domain has to be made. Although the income is missing within the German credit data set but this was not a reason to avoid this attribute within the definitions of a small query for a new given problem. Different strategies can be used to minimize the effect of missing values, for instance attribute income.

1) Substitution - Replace the missing value with an estimation: Generating an estimated value for the attribute income can be done with reasoning from other attributes such as the duration of the current employment and the amount of cash on the account of the customer which would be a rough estimation. An estimation function like this can be improved with additional knowledge provided by other attributes like country, job, age and a sub function which returns a range for a given job for a person within a given region of a country.
2) Overload methods to gain other queries - Using internal another query which can be made with using overloading of functions within the code. If the income is missing, then another query will be used with the same attributes except the attribute which refers to income.
3) Using social networks - Retrieving data by application programming interfaces (APIs) from social networks would be another feature but it would somewhat less than perfect. Many social networks are containing fake profiles or orphaned profiles. An additional preprocessing would be necessary to distinguish between fake profiles and real persons. The APIs of social networks are different and another issues like different e-mail-addresses for a person have to be considered if this kind of support really would be used. Extracting data concerning solvency from social networks was an upcoming issue in approx. 2010 and later but using this kind of approach was too buggy and rather subpar concerning many missing values and assumptions made in another approach during the work on this paper. For instance, an assumption was defined as follows: If the address of a house was evaluated as a real address by a Google Maps API call, then the profile of the social network will be classified as a real person which is comprehensible (if a person has not lied regarding the physical address). Another discussed assumption was to check relations to other human beings within the social network and use the gained knowledge to predict solvency for a person which is not suitable. [10]

VI. To Adapt or not to Adapt

Certain pre-conditions were to resolve before developing an adaptation feature which was enumerated within the previous sections. An awareness about the domain was reached with association models. The definitions of case, problem and solution was the basis for the case base. Similarity measures are an essential part of the adaptation procedure. Removing missing values avoids unnecessary adaptations because these could decrease the quality of the case base and increase the runtime cost of the software application. The solution quality

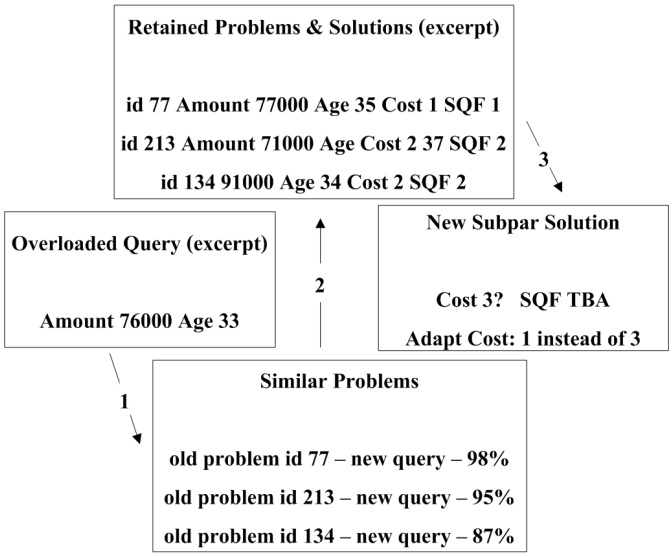

Fig. 2. Adaptation steps

factor (SQF) refers to 'TBA - to be announced' because the evaluation will be made within the revise step, but adaptation will be made earlier in the reuse step within the R^4 model. [1] The main idea will be to use a kind of quality factor for a proposed solution which is given by a user. A comparison between the proposed solution and previous solutions can affect and adapt attributes of a new subpar solution. If a similar solved problem exists according to the used similarity measures and the previous solution was marked with a rather good solution factor within the evaluation of a user, then the previous solution can be partially used as a basis for the adaptation of the new solution. Changing the internal queries towards the case base can provide another solutions, as appropriate, which can be used for a comparison with the suggested new subpar solution and adaptation if suitable. Defining a threshold value concerning the similarity measure regarding the new problem and retained cases (especially alternative solutions within these cases) has to be made.

Firstly, an internal overloaded query has to retrieve another solutions if available. Secondly, a similarity measure, according to the arguments of a modified query, can order the alternative solutions. Thirdly, these retrieved solutions will be used to modify attributes of a subpar solution. Within the reuse step, testing a modified solution can be made again with a similarity measure.

VII. Experiments

When comparing similarity values between a new searched solution and an initial query (also known as problem), it

was clear that a similarity measure with less attributes -in comparison to a former used similarity measure- could impair the quality of the case base. Therefore, a similarity measure should use attributes comparable to the initial given query. Otherwise the evaluation feature of a CBR system will be required. For instance, a similarity measure with only two attributes (age and credit amount) would deliver a subpar similarity result (52 per cent) when comparing a retained case (age 58, credit amount 6143) with a random query (age 27, credit amount 10467). However, interesting alternatives can be found when using a simpler similarity measure. Using weighting of attributes must be carefully considered to keep a good precision of search results within retained cases.

VIII. Evaluation by Teacher

According to Aamodt and Plaza, the assessment of a new solution by a user was defined as evaluation by teacher within the revise step of the R^4 cycle. [1] This concept clearly provides an advantage that probably wrong data or subpar solutions can be modified. The solution can be marked as helpful with a degree from A - Excellent to E - Not helpful. Evaluation by teacher can be seen as improving a CBR software application, but it can not circumvent the adaptation procedure of the reuse step. A manual repair step made by a user would be possible according to the R^4 model by Aamodt and Plaza, but this would not be suitable for every single case if a big volume, velocity and variety of data will occur. Current tendencies such as big data within the future can not be precluded.

IX. Conclusion

Adaptation of cases was the core of this work in progress paper. The adaptation was achieved when overloading queries. There exists a significant difference between different queries regarding the precision of the result which leads to different solutions.

1) The Good - additional data -if not redundant- can be an enrichment for the case base if used in a proper way.
2) the Bad - missing values can hide the actual nearest case.
3) and the Ugly - neglect both adaptation and evaluation would be subpar concerning the final solutions of a case. [11]

Testing and evaluation of new and adapted solutions should be made within the reuse and revise step of the CBR cycle to keep the quality of a case base. Automatically testing of an adapted solution fits to the reuse step, a manual evaluation of an adapted solution fits to the revise step within the CBR cycle.

X. Future Work

Many issues are open like finishing the work on similarity measures, adapt cases to improve solutions, develop an evaluation by teacher component and integrate all of these parts within one prototype. Another interesting point to research will be a deletion strategy to avoid inflating the case base with many (too) similar cases which affects the efficiency of a reasoning process.

The following real world example clearly shows a motivation to model and implement a deletion strategy for both too similar and redundant cases. Boeing has obtained more than eighty million flight hours after ten years which resulted in 23000 troubleshooting reports submitted by SNECMA Services. The maintenance of their engines was supported by a CBR system. At a certain point, they have retained too many similar and redundant cases because a deletion strategy was missing at the begin within their software application. An employee was used to check and remove, if necessary, manual redundant cases *at the rate of 15 cases per hour*. At the end, their system contained 1500 *"clean"* cases. [12]

Acknowledgment

Grateful acknowledgement for proofreading and providing hints to improve the paper go to Yuliya Nebylovych.

References

[1] A. Aamodt and E. Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches," *AI Commun.*, vol. 7, no. 1, pp. 39–59, 1994.
[2] K. Hammond, "Case-Based Planning: A Framework for Planning from Experience." *Cognitive Science*, vol. 14, pp. 385–443, 1990.
[3] R. Schank, "Dynamic Memory: A Theory of Learning in Computers and People," *New York, Cambridge University Press*, 1982.
[4] J. L. Kolodner, "Reconstructive Memory: A Computer Model," *Cognitive Science*, vol. 7, 1983.
[5] R. Bergmann, J. L. Kolodner, and E. Plaza, "Representation in Case-Based Reasoning," *Knowledge Eng. Review*, vol. 20, no. 3, pp. 209–213, 2005.
[6] J. Hönigl, H. Kosorus, and J. Küng, "On Reasoning within Different Domains in the Past, Present and Future," in *23rd Database and Expert Systems Applications (DEXA), 2012. 2nd International Workshop on Information Systems for Situation Awareness and Situation Management - ISSASiM'12*, September 2012.
[7] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The Weka Data Mining Software: An Update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656278
[8] J. Hönigl and Y. Nebylovych, "Building a Financial Case-Based Reasoning Prototype from Scratch with Respect to Credit Lending and Association Models Driven by Knowledge Discovery," *Central & Eastern European Software Engineering Conference in Russia*, November 2012.
[9] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml
[10] M. A. Stetco, "Creditworthiness Analysis using Data Gathered from Social Network Sites using a Supervised Learning Approach," 2012, Master Thesis, Johannes Kepler University, Linz, Austria.
[11] S. Leone, "The Good, the Bad and the Ugly. il buono, il brutto, il cattivo. (original title)," 1966.
[12] R. Bergmann, K. D. Althoff, S. Breen, M. Göker, M. Manago, and S. Wess, *Developing Industrial Case-Based Reasoning Applications: The INRECA Methodology*. Springer Verlag, 2003, vol. Lecture Notes in Artificial Intelligence Berlin, LNAI 1612, Berlin.

 ISBN: 978-1-61208-254-7