A Compositional Approach to Solution Adaptation in Case-Based Reasoning and its Application to Tutoring Library

Niloofar Arshadi[1] & Kambiz Badie[2]

[1] Software Engineering Group, Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
arsahdi@ce.sharif.ac.ir
[2] Iran Telecom. Research Center and School of Intelligent Systems, IPM, Tehran, Iran
k_badie@laleh.itrc.ac.ir

Abstract. This paper presents a new approach for compositional adaptation and investigates its applicability for a tutoring library system. Compositional adaptation has been applied in the designed tutoring library system, since many cases at the same time can be similar to the user request, and through this way, the possibility will exist to combine the corresponding solutions (books' chapters in our case) in an efficient way yielding the final solution. Each case in the library itself is represented in terms of a presuggested set of chapters from different books in general. Such an approach to suggesting solution to the user provides a suitable ground to take into account a variety of factors such as current knowledge level and the desired status of knowledge which have not been considered sufficiently in standard library systems.

1 Introduction

Case-Based Reasoning has been used to create numerous applications in a wide range of domains, including prediction, diagnosis, planning, process and quality control, monitoring, classification, configuration and design, decision support, and information retrieval [1,2]. While the theory and practice of case-based reasoning has benefited greatly from recent advances in case representation, similarity assessment, and retrieval, adaptation is still considered as the most difficult step in CBR [1]. The goal of the solution adaptation is to revise the solutions of the past cases such that they could best fit the ongoing situation. A number of approaches have been proposed for solution adaptation in case-based reasoning out of which substitution adaptation, transformational adaptation, derivational (generative) adaptation, and compositional adaptation are mentionable [1,2,3].

This paper presents a new approach for compositional adaptation and investigates its applicability for a tutoring library system. The system solves new problems as follows: first the user request (including keyword, searching area, the current knowledge level of the user and the desired status of knowledge) is presented to the system as the input information. The system then retrieves the most similar cases from the case library, and tries to adapt the corresponding solutions such that it could best fit the ongoing situation. The tutoring library database presents the set of chapters that would satisfy the user request. Finally, if the solution was found to be acceptable/suitable for the user, it would then be added as a new case to the case library.

2 Compositional Adaptation

In our approach to designing of tutoring library, we suggest the compositional adaptation, since many cases at the same time can be similar to the user request, and through this way, the possibility will exist to combine the corresponding solutions (books' chapters in our case) in an efficient way yielding the final solution.

In compositional adaptation, solutions from multiple cases are combined to produce a new composite solution [1,2]. Compositional adaptation can be applied in two distinct situations:

a) When the solution consists of different independent parts, then each of these components can be adapted more or less precisely. This method is effective if there are few conflicts between these components [1]. For example, Prodigy/Analogy constructs a new solution from a set of guiding cases as opposed to a single past case. Here, complex problems may be solved by resolving minor interactions among simpler past cases [9].

b) The solution could not be divided into independent parts, so the solutions in the similar cases should be combined in some way. In Airquap, which is a CBR system for predicting the pollution levels, the solution to the target problem is the mean value of the solutions belonging to the most similar cases in the library [4].

In the case of a tutoring library system, the first method could not be used because the solution could not be divided into independent parts, and since the chapters (or the sections), which are the basic concepts of a book, have discrete characteristics, it is not possible to find numbered mean values for these discrete entities, so, the second method is not applicable, either. However, through combining the similar components in the solutions belonging to the similar cases, the system will be able to generate the final solution that is expected to fit best the user request.

3 Tutoring Library; Basic Concepts

According to the different search aspects defined in [5], when a library user searches for a book in a database, three principle situations might occur:

- The library user knows precisely the specifications of his/her book (book title, authors, publishing year, etc.). So, s/he could find directly what s/he was looking for.
- The library user has forgotten the specifications of his/her book, or the book is not currently available.
- Often, situations occur in which the library user does not know precisely what s/he is looking for. Rather, s/he has some kind of need s/he wants to fulfill and a more or less vague idea of what the solution might look like.

Standard database technology has in particular problems in dealing with the last two situations. Either the customer (or a library user in our case) is overwhelmed with hundreds of offers or s/he is left alone with no solution at all [5].

In our approach to designing the tutoring library database, we have relied on the third principle. So, we can claim that a tutoring library should follow the third principle. In standard library systems, for the user to be able to find his/her desired book, first, s/he should explain his/her request for an expert, and the expert, with respect to the current knowledge level of the user, would propose some books. Finally, to find the book in the library, the user must present the specifications of the book, more or less precisely. For example, a student describes his request as follows: "I need a book about Genetic Algorithms, I do not know anything about it and I want to acquire general knowledge about the subject". By using the tutoring library system there would be no need for the users to refer directly to the experts, and this system can automatically suggest the user an appropriate combination of the chapters (belonging to different books in general). In fact the tutoring library tries to select the chapters in such a manner that it helps the user grasp his/her desired concept in a more efficient way. Since, one single book can hardly achieve this role, and at the same time no single chapter in any selected book may necessarily be useful for the user, it would be desirable to take into account a combination of different chapters from different books. If these chapters are selected properly, their efficiency for covering the whole range of the essential concepts would be remarkable. This issue is, in practice observed by many instructors who advise their students to study certain chapters of different books, when promoting the students' knowledge in a certain area is taken into account.

The tutoring process mentioned here is not directly supported by teachers or librarians' domain knowledge. Instead, the solutions in the cases (to be provided by the teachers or librarians) include this domain knowledge in an implicit way. Let say, if promoting one's knowledge to a certain level calls for studying a number of chapters from different books, one can say that the teacher's domain knowledge is implicitly stored in the way the chapters are selected.

Finally, this fact that the final solution can be a combination of different chapters belonging to different books (in contrast with the concept of the whole book as the solution for the user) provides the facility for tutoring libraries to create something like a virtual book for the user (as a source for upgrading his/her knowledge in a certain field) which consists of a variety of chapters from different real books.

4 The CBR Cycle for the Tutoring Library System

The CBR cycle for the proposed tutoring library system, according to the CBR cycle defined in [6], has the stages as follows:

1. Retrieve the similar cases from case library. Cases that have the same searching area and keyword are considered to be similar.
2. Reuse the information and knowledge in those cases to solve the problem
3. Revise the proposed solution
4. Retain the user request together with the system's proposed solution as a new case to be examined and evaluated furtherly.

It is often difficult to distinguish between the reuse and revise stages, so it is considered as a single solution adaptation stage. Here, a compositional adaptation approach has been used for solution adaptation. Now, we briefly explain each stage.

4.1 Case Representation and Retrieval Similar Case(s)

In CBR terminology, a case usually denotes a problem situation. A previously experienced situation, which has been captured and learned in a way that it can be reused in the solving of future problems, is referred to as a past case [6].

Each case in this system consists of two parts:

a) *Problem description*: In this part, the features of the problem are described. The system receives the related information from the user and produces the corresponding solution based on this information. This part includes two subparts:

- Information about the book: It consists of the searching area and the keyword. For example, "Artificial Intelligence" is a searching area and "The Operators of Genetic Algorithms" is keywords. In the implemented version of the tutoring library system, searching area could be "Artificial Intelligence" or "Software Engineering" and the keywords are selected from a predefined list. This list demonstrates the keywords in a hierarchical tree that is created by consulting the expert teachers or librarians. For the moment, we do not use any method for retrieving the information of the keywords in the chapters, and we just rely on expert intuition for extracting the keywords. We however believe that inducing text processing approaches will enable us to retrieve the keywords information in a more exact manner.
- Information about the user: It contains the current knowledge level of the user about the subject and the desired status of knowledge. The current knowledge level of the user might be preliminary, intermediate or advanced. The desired status of knowledge might be research, teaching, acquiring general knowledge, or developing a new system. It is obvious that, the library's solution for the user may highly depend on both his/her current and the desired status of knowledge. Let's say, if the user's goal is teaching, the desired

solution would be chapters of the textbooks which are sufficiently capable to transfer the basic concepts to the user preferably in terms of examples and exercises. And, if the user's goal is doing research, the desired books should cover theoretical or historical aspects, and may contain mathematical equations or formulas more seriously.

b) *Solution:* In this part, the books' chapters suitable for the user are presented to the user together with the authors and the corresponding publishing years. In other words, after retrieving the most similar cases and adapting the corresponding solutions, the set of suitable chapters would be offered to the user. An important point concerning chapters' selection is that some chapters of a book are highly dependent on some other chapters in the same book. In this case, these groups of chapters should appear together in the final solution.

In our approach, process of retrieval is performed hierarchically. Let say, at the first level, those cases are selected which fully satisfy both the searching area and the keywords included in the user requset. At the second level, the concept of k-nearest neighbour [7] is used to select (out of the cases selected at the first level) k nearest cases whose similarity to the user request from the veiwpoints of current knowledge level and the desired status of knowledge is higher compared to the remaining cases.

To achieve this, for current knowledge level, we considered "0" as the similarity degree between "preliminary" and "advanced", "0.1" as the similarity degree between "preliminary" and "intermediate", or "intermediate" and "advanced", and "0.2" as the similarity degree between the same items. Also, for desired status of knowledge "0" was considered as the similarity degree between the differing items, while "0.1" was considered as the similarity degree between the same items.

4.2 Compositional Adaptation

A new approach to compositional adaptation has been used for proposing suitable solutions to the tutoring library user. The main idea is that if a chapter appears in several similar cases already retrieved, then the possibility that it could be a suitable choice for the user would be high. This is based on the principle that the individual solutions for similar problem situations may structurally share some beneficial components which should appear in the final solution. Although there may exist individual solutions that despite their utilities do not share any component with the other solutions, in our approach we ignore these solutions.

It should also be noted that the degree of similarity between the user request and a retrieved case will influence the structure of the final solution. Therefore, both frequency of appearance of a chapter and the similarity degree will influence the possibility of that chapter to participate in the final solution. In our approach to compositional adaptation, the goal is to determine the appropriateness of each chapter appeared in the solution of each similar case. Following steps should be traced in this respect:

1. The total distance is calculated using the similarity between the retrieved case and the user request. Here, total distance is simply defined to be the

sum of the difference between the current knowledge level of the user and the one stored in the similar case, on one hand, and the difference between the user's desired status of knowledge and the stored in the similar case, on the other hand. Table 1 shows how the distance is determined regarding the current level of knowledge.

Table 1. Scoring the current knowledge level of the user

	Preliminary	Intermediate	Advanced
Preliminary	0	1	2
Intermediate	1	0	1
Advanced	2	1	0

If the desired status of knowledge in both the retrieved case and the user request, is the same, the distance would be zero, otherwise the distance would be one.

2. Let distance_i be the total distance of a similar case_i calculated in the previous step, and n be the number of the cases which are similar to the user request. The normalized distance of a similar case_i , which indicates the amount of utility percentage for the solution, would then be obtained through the following expressions:

$$\text{Temp} = \sum_{i=1}^{n} 1/\ \text{distance}_i \qquad (\mathbf{1})$$

And

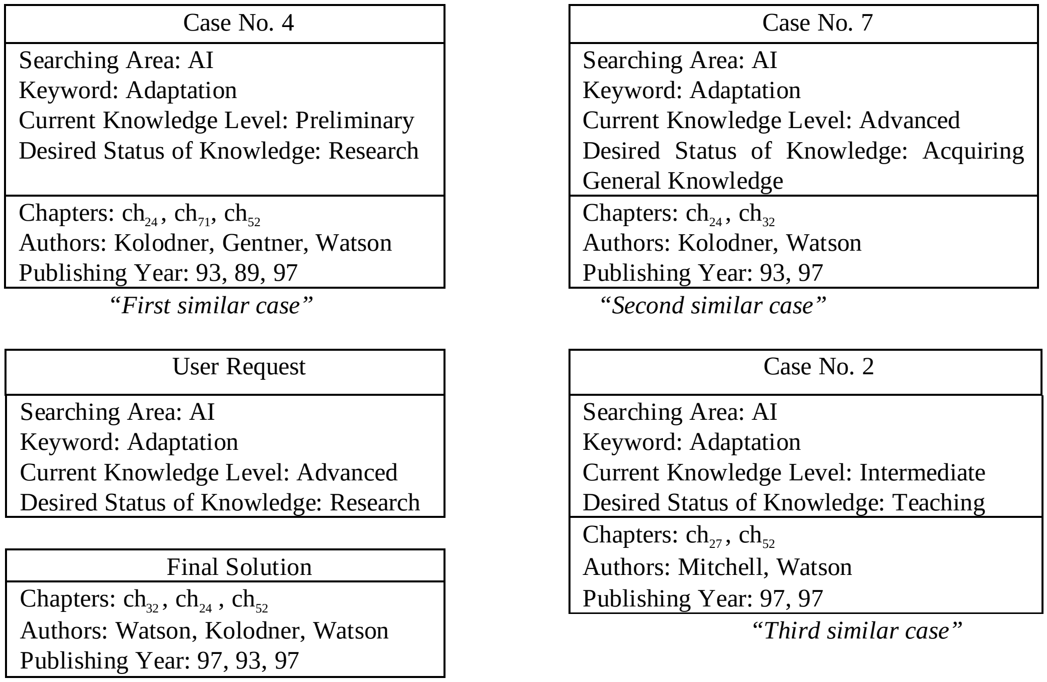
$$\text{Normalized distance case}_i = 1\ /\ (\text{distance}_i * \text{temp}) \qquad (\mathbf{2})$$

3. Let ch_{ij} be the ith chapter in the jth book, and z be the appropriateness degree for each chapter. Then, z would be determined as follows:

```
for k=1 to n do
  if $ch_{ij}$ exists in the solution of similar $case_k$ then
    add the normalized distance $case_k$ to z
```

Finally, those chapters which have an appropriateness degree more than a certain threshold, would be included in the final solution. These chapters would be appeared in the final solution together with the authors’ name and the publishing year of the corresponding books. In the implemented tutoring library system, there is no order among the books' chapters appearing in the final solution. However for the future versions of the system, considering the order can lead to a better selection of the final solution.

Fig. 1 illustrates the final solution to the user request with respect to the three similar cases (No. 2, No. 4, and No. 7) retrieved from the case library. Here, the threshold for appropriateness degree has been considered to be 0.5.



Case No. 4

Searching Area: AI
Keyword: Adaptation
Current Knowledge Level: Preliminary
Desired Status of Knowledge: Research

Chapters: ch_{24}, ch_{71}, ch_{52}
Authors: Kolodner, Gentner, Watson
Publishing Year: 93, 89, 97

"First similar case"

Case No. 7

Searching Area: AI
Keyword: Adaptation
Current Knowledge Level: Advanced
Desired Status of Knowledge: Acquiring General Knowledge

Chapters: ch_{24}, ch_{32}
Authors: Kolodner, Watson
Publishing Year: 93, 97

"Second similar case"

User Request

Searching Area: AI
Keyword: Adaptation
Current Knowledge Level: Advanced
Desired Status of Knowledge: Research

Case No. 2

Searching Area: AI
Keyword: Adaptation
Current Knowledge Level: Intermediate
Desired Status of Knowledge: Teaching

Chapters: ch_{27}, ch_{52}
Authors: Mitchell, Watson
Publishing Year: 97, 97

"Third similar case"

Final Solution

Chapters: ch_{32}, ch_{24}, ch_{52}
Authors: Watson, Kolodner, Watson
Publishing Year: 97, 93, 97

Fig. 1. An example of compositional adaptation in case-based library system

Table 2 and 3 demonstrate the normalized distance of similar cases and their corresponding degree.

Table 2. Normalized distance of similar cases

	Solution	Total distance	Normalized distance
1st similar case	ch_{24}, ch_{32}	1	0.5
2nd similar case	ch_{24}, ch_{71}, ch_{52}	2	0.25
3rd similar case	ch_{27}, ch_{52}	2	0.25

To calculate the appropriateness degree of each chapter, the normalized distance of the similar cases that include the chapters should be added to one another.

Table 3. The appropriateness degree of each chapter

Chapter No.	Appropriateness
ch_{24}	0.75
ch_{32}	0.5
ch_{71}	0.25
ch_{52}	0.5
ch_{27}	0.25

Here, the final solution would be:

Solution to user request = { ch_{32} , ch_{24} , ch_{52} }

We can justify the final solution as follows: since the normalized distance of the first similar case is 0.5 and the threshold is 0.5, its solution (including ch_{24}, ch_{32}) will appear in the final solution. Also, ch_{25} has appeared in two similar cases with the normalized distance 0.25, so this chapter will also appear in the final solution.

It should be noted that, the higher the number of similar cases is, the probability for different books' chapters to participate in the final solution would be higher.

4.3 Retaining Tested Cases

Finally, if the solution was found to be acceptable/suitable for the user, it should be added as a new case to the case library, and at the same time can be stored on the disk to be protected against probable damages. Cases are stored in a case file on the disk, and every time before running the program, its content is loaded into the main memory in order to increase the speed of retrieving similar cases.

5 Implementation and Validation of the Tutoring Library System

The proposed tutoring library system was implemented using Borland C++ under Windows 3.1. According to the method defined in [8], the tests used for validation are as follows:

a) CBR Retrieval Test
b) CBR Adaptation Test
c) Domain Coverage Test

The main concept underlying this validation method is selecting a subset of the cases from the case library and then using this subset as test cases to evaluate the correctness of the system's retrieval and adaptation functions [8]. By considering the results of the above tests, we can claim that the implemented tutoring library system works properly. However to examine and evaluate the system performance, and also to determine the user satisfaction level, a noticeable amount of time in a real library environment is unavoidable. At the moment, we are planning for a suitable library environment to achieve this goal.

6 Concluding Remarks

In this paper, a new approach to compositional adaptation was presented and applied as a methodology for tutoring library.

It was seen that the approach has the capability to make the final solution for the user on the basis of combining a variety of chapters belonging to different cases which are somehow similar to the user request. Each case in the library itself is represented in terms of a presuggested set of chapters from different books in general. Such an approach to suggesting solution to the user provides a suitable ground to take into account a variety of factors such as current knowledge level and the desired status of knowledge which have not been considered sufficiently in standard library systems. Moreover, due to the very dynamicy and complexity which exist in the trend of human knowledge development in general and technological knowledge development in particular, it would be much important for the library information systems to have emergent and dynamic solutions for their users. The tutoring library paradigm, discussed in the paper, have such characteristics.

Finally, It is worth noticing that tutoring library can be a good alternative for virtual education particularly with respect to transferring the particular technical concepts whose performance by human teachers may need intolerable amount of time, energy and systematicness.

6 References

[1] Wilke, W., Bergmann, R., "Techniques and Knowledge Used for Adaptation During Case-Based Problem Solving", Tasks and Methods in Applied Artificial Intelligence, LNAI 1416, Springer-Verlag, pp. 497-505, 1998

[2] Lenz, M. et. al. , "Case-Based Reasoning Technology: From Foundations to Applications", LNAI 1400, Springer-Verlag, 1998

[3] Kolodner, J., "Case Based Reasoning, Morgan Kaufmann Publishers", 1993

[4] Lekkas, G.P., Avouris, N.M., Viras L.G., "CBR in Environmental Monitoring Applications", Applied Artificial Intelligence, vol. 8, pp. 359-376, 1994

[5] Sporl, B.B., Lenz, M., Hubner, A., "Case-Based Reasoning-Survey and Future Directions", Knowledge-Based Systems-Survey and Future Directions, LNAI 1570, Springer-Verlag, pp. 67-89, 1999

[6] Aaomodt, A., Plaza, E., "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches", AI Communications, IOS Press, Vol. 7:1, pp. 39-59, 1994

[7] Watson, I., "Applying Case-Based Reasoning: techniques for enterprise systems", Morgan Kaufmann, Calif., US., 1997

[8] Gonzalez, A.J., Xu, L., Gupta, U.M., "Validation Techniques for Case-Based Reasoning Systems", IEEE Transactions on Systems, Man, and Cybernetics- Part A: Systems and Humans, Vol. 28, No. 4, pp. 465-477, July 1998

[9] Veleso, M.M., "Prodigy/Analogy: Analogical Reasoning in General Problem Solving", Topics in Case-Based Reasoning, LNAI 837, Springer-Verlag, pp. 33-50, 1993