INTEGRATING CASE BASED REASONING AND EXPLANATION BASED LEARNING IN AN APPRENTICE AGENT

Lei Wang, Tetsuo Sawaragi, Yajie Tian and Yukio Horiguchi
Graduate School of Engineering, Kyoto University, Yoshida Honmachi, Sakyo-ku, Kyoto, Japan

Keywords: Case based reasoning, Explanation based learning, Machine learning.

Abstract: The problem in applications of case based reasoning (CBR) is its utility problem, that is, the cost of retrieving the most appropriate case from the case library for a new given problem and the cost of adapting the retrieved case for solving the new given problem. This paper proposes an approach to solve the utility problem of CBR by integrating CBR and explanation based learning (EBL) from a perspective that emphasizes the function of learning in CBR. In this paper, CBR and EBL are integrated in an apprentice agent, and the application of this apprentice agent in the robotic assembly domain is given as an example.

1 INTRODUCTION

Case-based reasoning (CBR) is a problem solving paradigm that in many respects is fundamentally different from other major AI approaches. Instead of relying solely on general knowledge of a problem domain, or making associations along generalized relationships between problem descriptors and conclusions, CBR is able to utilize the specific knowledge of previously experienced, concrete problem situations (cases). A new problem is solved by finding a similar past case, and reusing it in the new problem situation. A second important difference is that CBR also is an approach to incremental, sustained learning, since a new experience is retained each time a problem has been solved, making it immediately available for future problems (Aamodt and Plaza, 1994). Generally, a CBR cycle is described by the following four processes: 1. ***Retrieve*** the most similar case or cases; 2. ***Reuse*** the information and knowledge in that case to solve the problem; 3. ***Revise*** the proposed solution; 4. ***Retain*** the parts of this experience likely to be useful for future problem solving.

The problem in applications of CBR is its utility problem, that is, the cost of retrieving the most appropriate case from the case library for a new given problem and the cost of adapting (i.e., reusing and revising) the retrieved case for solving the new given problem. Mantaras, et al. (Mantaras, et al., 2006) regard the utility problem as a natural trade-off between the benefits of speed-up knowledge and the cost of its application. In their view, the utility problem in CBR systems is caused by the conflict between: 1. the average savings in adaption effort due to the availability of a particular case, which tends to increase efficiency as the case base grows, and 2. the average retrieval time associated with a given case base size, which tends to decrease efficiency. Moreover, as new cases are added retrieval costs become progressively greater but adaption savings progressively less. Therefore, most researchers on CBR focus on developing new retrieval and adaption methods. There are also researchers who have discovered the importance of maintaining the case library to solve the utility problem of CBR (Iglezakis, Reinartz and Roth-Berghofer, 2004; Wilson and Leake, 2001).

However, in our opinion, learning (i.e., retaining) is very important for CBR with regard to solving its utility problem. This is because the retrieval and adaption costs are not solely depend on the amount of cases, but also rest with the representation forms and contents of the cases. In CBR, learning decides the representation forms of cases and the contents that can be learned from cases. Therefore, our basic idea is to save retrieval and adaption costs by making more efforts on post-processing of cases. The aim of emphasizing learning is to post–process

Wang L., Sawaragi T., Tian Y. and Horiguchi Y. (2010).
INTEGRATING CASE BASED REASONING AND EXPLANATION BASED LEARNING IN AN APPRENTICE AGENT.
In *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence - Artificial Intelligence*, pages 667-670
DOI: 10.5220/0002763306670670

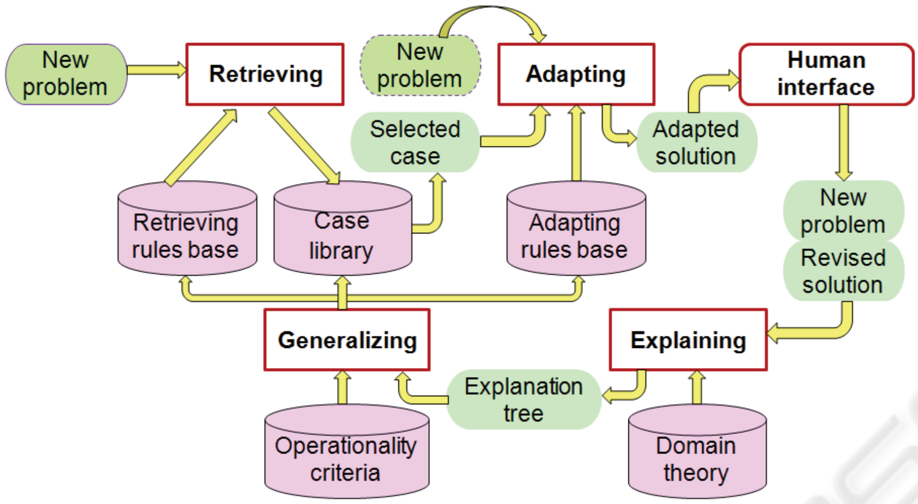

Figure 1: Structure of the apprentice agent.

cases to make them easier to be retrieved and to be adapted for new problems. The method we use to learn cases is explanation based learning (EBL).

EBL is a learning method that can acquire knowledge through observing a single training example with the help of a pre-encoded knowledge base, which is called domain theory (Mitchell, Keller and Kedar-Cabelli, 1986; DeJong and Mooney, 1986; DeJong, 2006). Given a domain theory, a description of the goal concept, an operationality criterion and a training example, the EBL method tries to improve the domain theory in order to obtain a more efficient (operational) definition of the goal concept. EBL has two main steps: 1. to build an explanation justifying why the input example is a positive instance of the goal; and 2. generalizing the explanation as much as possible while the explanation holds. The explanation is the proof tree build by the system during the problem solving process. Therefore, the explanation is generated in a deductive way and its generalization will be correct since deductive methods are truth-preserving. Finally, from the generalized explanation new rules can be generated and stored as part of the domain theory and they can be used for solving further problems.

The idea of EBL is much like that of CBR. Both of them are to acquire knowledge from a single problem-solving example and to reuse the acquired knowledge to solve new problems. The difference between the two is that EBL uses a domain theory to explain why the example is a positive example of the goal concept and generalizes the explanation to form an operational knowledge that can be generally reused for a type of new problems, while CBR doesn't analyze the example, but just directly saves the example as a case into the case library. Therefore, CBR and EBL can be integrated with each other. Armengol, et al. (Armengol, Ontanon and Plaza, 2004; Armengol, 2007) have applied EBL in retrieving appropriate cases in CBR.

In this paper, we integrate EBL into CBR to make an apprentice agent that is applied in the robotic assembly domain. The apprentice agent works as a co-worker of human workers to assist them in their task of teaching robots. The main functions of the apprentice agent are: 1. automatically generating robot programs for new assembly tasks by reusing past learned experiences; 2. providing suggestions and hints for human workers when human workers revise the robot programs or teach robot new assembly skills; 3. acquiring knowledge from revising and teaching demonstrations of human workers. The most distinctive feature of this apprentice agent is that it uses CBR to help human workers in teaching robots new assembly tasks by reusing past experiences and applies EBL to learn assembly knowledge by observing robot teaching demonstrations of human workers. The EBL learning process can be regarded as a post-processing process of cases before retaining them in CBR. Its aim is to reduce the retrieving and adapting cost of cases.

2 THE APPRENTICE AGENT

Figure 1 shows the structure of the apprentice agent. There are five functional modules in the apprentice agent: 1. Retrieving module, 2. Adapting module, 3. Human interface module, 4. Explaining module, and

5. Generalizing module. Each of the modules has its own knowledge base, except for the human interface module. This is because human workers are the knowledge source for the human interface module. Through this module, the apprentice agent interacts with human workers and acquires knowledge from them.

2.1 Retrieving Module

After a new problem (i.e., an assembly task) has been input into the apprentice agent, its retrieving module uses a hierarchical mechanism to retrieve the most appropriate case for the problem.

The hierarchical mechanism works in the following two steps. First, it examines the features of the problem, and matches these features against heads of retrieving rules in the retrieving rules base. If the features satisfy the head of a retrieving rule, then this rule is selected and used to select a class of past cases that corresponding to this rule from the case library. In the robotic assembly domain, the features examined in the first step of the retrieving module are: 1. Type of the target workpiece, 2. Type of the robot tool, 3. Type of assembly operation, and 4. Destination environment. Second, the retrieving module uses a similarity distance calculation algorithm to select the case that is most similar to the new input problem from the class of cases confined by the retrieving rule in the first step. The similarity distance calculation algorithm works by assigning different weights to the specific data in the problem description such as the geometric data of the workpiece and robot tool, assembly operation, and destination environment information. It calculates the similarity distances between the retrieved cases in the first step and the input problem, and selects the case with the shortest similarity distance to the input problem.

Each case in the case library is corresponding to a retrieving rule. This rule is generalized by EBL in the learning process of CBR. Its role is to classify cases in the case library according to their features. In this way, the retrieving cost can be reduced by classifying the cases with retrieving rules.

2.2 Adapting Module

Both the selected case and the new input problem description are input to the adapting module. The adapting module compares the data in the new problem description against those in the problem description of the selected case to determine whether to revise the solution plan in the selected case or to reuse it directly.

In our apprentice agent, a case in the case library is composed of three parts: 1. Primitive problem description of the case, 2. Generalized solution plan, 3. Explanation of the generalized solution plan. In the robotic assembly domain, a primitive problem description is a description of an assembly task. A generalized solution plan is a generalized robot program for the assembly task. A generalized robot program consists of a robot command schema and point parameter deciding methods for points in the robot schema. Explanation of the generalized program provides instructions of the robot schema and point parameter deciding methods to make them be easily understood by human workers.

The adapting module decides whether to revise the selected case by searching available adapting rules with considering differences between the new problem description and the problem description of the selected case. If adapting rules are searched, it will revise the case according to these rules. Otherwise, it won't revise the case but just reuses it directly.

Here, reusing means the adapting module uses the robot program schema and determines point parameters with point parameter deciding methods and data in the new problem description to generate a robot program for the new problem. Revising means the adapting module revises the robot commands or point parameter deciding methods in the generated program according to adapting rules.

The adapting rules are not pre-encoded, but are learned by the EBL process of the apprentice agent. Thus, an adapted case without revision doesn't mean it doesn't need revision, but means there has not been available adapting rules learned for revising it.

2.3 Human Interface Module

The human interface module shows the adapted solution to human workers. Human workers review it and test it in the playback mode of robots. If human workers are not satisfied with its performance or error occurs, they will further revise the adapted solution. Then the human interface module sends the new problem description, the revised solution, together with the adapted solution to the EBL process (i.e., explaining and generalizing modules) of the apprentice agent to acquire assembly knowledge of human workers from them.

2.4 Explaining and Generalizing Modules

Explaining and generalizing modules constitute the EBL process of the apprentice agent. Our past work (Wang, L., Tian, Y. and Sawaragi, T., 2008) has described its detailed working mechanism.

The EBL process both can learn from examples directly given by human workers and can learn from examples generated by the adapting module and further revised by human workers. The learning results include: 1. a retrieving rule for the case, 2. the case, and 3. adapting rules of the case. The adapting rules are learned by comparing the revised solution with the adapted solution.

3 AN EXAMPLE

Figure 2 shows an application example of the apprentice agent. In this example, the apprentice agent assists human workers by generating robot programs for palletizing two rows of blocks into a plate. The blocks in the same row have the same cross-section but different heights. The blocks in the left row have bigger widths (i.e., are thicker) than those in the right row.

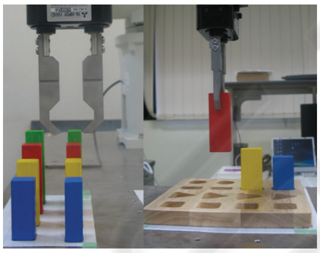

Figure 2: An example: palletizing blocks.

First, human workers teach the robot how to palletize the blue block in the right row. Then the apprentice agent reuses this case successfully in palletizing the rest blocks in the left row.

However, when the case is reused in palletizing the first (i.e., the blue) block in the left row, an error occurs. The robot tool collides with the target blue block. Then human workers revise the adapted solution by inserting a command to slow down the robot speed before the command of closing the robot tool. The revised solution can be executed without errors. The EBL process learns a new case from the revised solution and an adapting rule that if the width of the workpiece is not much smaller than (i.e., >80% of) the open width of the robot tool, then robot should slow down before gripping the workpiece. The apprentice agent reuses the new learned case in palletizing the rest blocks of the left row successfully.

4 CONCLUSIONS

We propose a method that integrates CBR and EBL in an apprentice agent to solve the utility problem of CBR. Its distinctive feature is applying EBL in post-processing an observed case to reduce its reusing and adapting cost. While this apprentice agent can be used for general purposes, in this paper we apply it in the robotic assembly domain.

REFERENCES

Aamodt, A. and Plaza, E., 1994. Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Communications, Vol. 7: 1, pp. 39-59.

Armengol, E., Ontanon, S. and Plaza, E., 2004. Explaining similarity in CBR. ECCBR 2004 Workshop Proceedings.

Armengol, E., 2007. Usages of generalization in case-based reasoning. Lecture Notes In Artificial Intelligence, Vol. 4626. Proceedings of the 7th international conference on Case-Based Reasoning: Case-Based Reasoning Research and Development.

DeJong, G. and Mooney, R., 1986. Explanation-based learning: an alternative view. Machine Learning, Vol.1, pp.145-176.

DeJong, G., 2006. Toward robust real-world inference: a new perspective on explanation-based learning. ECML 2006.

Iglezakis, I., Reinartz, T. and Roth-Berghofer, T., 2004. Maintenance memories: beyond concepts and techniques for case base maintenance. In Proceedings of the Seventh European Conference on Case-Based Reasoning. Berlin: Springer, pp. 227–241.

Mantaras, R., et al., 2006. Retrieval, reuse, revision and retention in case-based reasoning. The Knowledge Engineering Review, Vol.20:3, pp. 215-240.

Mitchell, T., Keller, R. and Kedar-Cabelli, S., 1986. Explanation-based generalization: a unifying view. Machine Learning, Vol.1, pp.47-80.

Wang, L., Tian, Y. and Sawaragi, T., 2008. Explanation-based manipulator learning: acquisition of assembling technique through observation. Proceedings of the 17th World Congress of IFAC, pp.2412-2417.

Wilson, D. and Leake, D., 2001. Maintaining cased-based reasoners: dimensions and directions. Computational Intelligence, Vol.17:2, pp.196–213.