Proceedings of the 14th Annual Conference of JSAI, 2000
Hybrid Reasoning Architecture for Solving the Object Classes Identification’s Problems in the OOExpert System

Romi Satria Wahono  Behrouz H. Far

Abstract: The recent evolution of hybrid architectures for knowledge based systems has resulted in several approaches that combine Rule-Based Reasoning (RBR) with Case-Based Reasoning (CBR) techniques to engender performance improvements over more traditional one-representation architectures. CBR is used in learning and problem-solving systems to solve new problems by recalling and reusing specific knowledge obtained from past experience. RBR systems learn general domain-specific knowledge from a set of training data and represent the knowledge in comprehensible form as if-then rules. Due to their complementary properties, CBR and RBR techniques have been combined in some systems to solve problems to which single technique fails to provide a satisfactory solution. In the knowledge-based systems for examination tasks, advice task and so on, not only rules but also cases are necessary for decision-making. In this paper we present a new hybrid reasoning architecture for integrating both reasoning paradigms for solving the object classes identification’s problems in the object-oriented software design.

1 Introduction

Artificial Intelligence (AI) researchers have embraced a variety of reasoning techniques in their efforts to improve the quality of knowledge-based systems or expert systems. The recent evolution of hybrid architectures for knowledge-based systems has resulted in several approaches that combine RBR with CBR techniques to engender performance improvements over more traditional one-representation architectures [Cercone et al., 1999].

CBR can mean adapting old solutions to meet new demands, using old cases to explain new situations, using old cases to critique new solutions, or reasoning from precedents to interpret a new situation or create an equitable solution to a new problem. RBR learn general domain-specific knowledge from a set of training data and represent the knowledge in comprehensible form as if-then rules.

In our project, we are developing a distributed knowledge-based system that aims to help designers while designing object-oriented software by automating the difficulties and ill-defined tasks in the object model creation process, including identification of objects, relationships, attributes, behaviors, and organization of objects with inheritance.

We formulate design patterns and rules for solving above problems, and store them in the distributed knowledge bases. This system is named OOExpert [Romi et al., 1999].

In this paper we present the hybrid reasoning architecture for integrating both reasoning paradigms for solving the object identification’s problem in the OOExpert.

2 Integration Approaches

The essential characteristics and comparisons between RBR and CBR technique is shown in Table 1. By comparing both techniques we try to figure out the strength and weakness of both techniques.

However, the complementary properties of CBR and RBR can be advantageously integrated to solve some problems to which only one technique fails to provide a satisfactory solution. Generally, RBR and CBR are often used together, where the use of rules is supplemented with the use of cases that determine the scope of the rules. CBR processing can be augmented with RBR when general domain knowledge is required.

Figure 1 shows the architecture of the object identification in the OOExpert by using RBR and CBR integration approach.

The first step constructing an object model is to identify relevant objects from the application do-
Table 1: RBR and CBR Comparisons

<table>
<thead>
<tr>
<th>Problem Area</th>
<th>RBR</th>
<th>CBR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Narrow, well understood, stable domain theory</td>
<td>Wide, poorly understood, unstable domain theory</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Knowledge Representation</th>
<th>Facts and IF-THEN rules</th>
<th>Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Provided</td>
<td>Answers</td>
<td>Procedures</td>
</tr>
<tr>
<td>Explanation By</td>
<td>Trace of fired rules</td>
<td>Procedures</td>
</tr>
<tr>
<td>System Can Learn</td>
<td>No, usually requires manual addition of new rules</td>
<td>Yes, by case acquisition</td>
</tr>
<tr>
<td>When To Use</td>
<td>Well understood, stable, narrow problem area and justification by rule-trace acceptable</td>
<td>Poorly understood problem area with complex structured data that changes slowly with time and justification required</td>
</tr>
<tr>
<td>When Not To Use</td>
<td>Poorly understood problem area that constantly changes</td>
<td>When case data is not available, or if complex adaptation is required, or if an exact optimum answer is required</td>
</tr>
</tbody>
</table>

Figure 1: Architecture of the Object Identification

main. Objects include physical entities, such as houses, employees, and machines, as well as concepts, such as trajectories, seating assignments, and payment schedules. All objects must make sense in the application domain, i.e., confine with the task ontology of the domain. As shown in Figure 1, begin by listing candidate objects found in the written requirements specification of the problem. Objects often respond to nouns. Then the reasoning engine of the OOExpert will process this nouns extraction request by using rules from rule-base and cases (experiences) from case-base. As a result we have tentative objects.

The next step is to eliminate spurious objects. In the RBR, the system will discard unnecessary and incorrect objects according to the following criteria: redundant objects, irrelevant objects, vague objects, attributes, operations, roles, and objects that point at implementation constructs.

In other hand, CBR is based on psychological theories of human cognition. We collect design rules from human experts, and store/index them in the case-base. It rests on the intuition that human expertise does not depend on rules or other formalized structures, but on experiences. Human experts differ from novices in their ability to relate problems to previous ones, to reason based on analogies between current and old problems, to use solutions from old experiences, and to recognize and avoid old errors and failures. Using cases from case-base, we can get another solutions of identifying object, from experiences of human experts.

Using this integration approach, RBR and CBR have been combined in the OOExpert to engender performance improvements and to solve the problems of object identification.

3 Conclusion

CBR is used in learning and problem-solving systems to solve new problems by recalling and reusing specific knowledge obtained from past experience. RBR systems learn general domain-specific knowledge from a set of training data and represent the knowledge in comprehensible form as if-then rules. Due to their complementary properties, CBR and RBR techniques have been integrated in some systems to solve problems to which single technique fails to provide a satisfactory solution, also to engender performance improvements over more traditional one-representation architectures.

In this paper we presented the architecture for integrating both reasoning paradigms and implement it for solving the object identification in the OOExpert.

References


