Using Knowledge Containers to Model a Framework for Learning Adaptation Knowledge

Wolfgang Wilke, Ivo Vollrath, Ralph Bergmann
University of Kaiserslautern
Centre for Learning Systems and Applications (LSA)
Department of Computer Science
P.O. Box 3049, D-67653 Kaiserslautern, Germany
e-Mail: {wilke, vollrath, bergmann}@informatik.uni-kl.de

Abstract. In this paper we present a framework for learning adaptation knowledge which knowledge light approaches for case-based reasoning (CBR) systems. Knowledge light means that these approaches use already acquired knowledge inside the CBR system. Therefore we describe the sources of knowledge inside a CBR system along the different knowledge containers. After that we present our framework in terms of these knowledge containers. Further we apply our framework in a case study to one knowledge light approach for learning adaptation knowledge. After that we point on some issues which should be addressed during the design or the use of such algorithms for learning adaptation knowledge. From our point of view many of these issues should be the topic of further research. Finally we close with a short discussion and an outlook to further work.

1 Introduction

Until now there are only few investigations in learning adaptation knowledge. Some approaches for learning adaptation knowledge can be found in DIAL (David Leake, 1993; Leake, 1995b; Leake, 1995a) or CHEF (Hammond, 1986; Hammond, 1989). These systems use knowledge intensive derivational analogy approaches (Carbonell, 1986; Veloso and Carbonell, 1993) to learn adaptation knowledge. Knowledge intensive means that these approaches require a lot of background and problem solving knowledge. For example in DIAL and CHEF adaptation strategies for special problem fields are acquired based on general domain knowledge. So a reduction of the knowledge acquisition cost is not necessarily the case because the knowledge engineering effort might be costly.

In this work we want to focus on what we call knowledge light approaches for learning adaptation knowledge1. Knowledge light means that these al-

1 In the following we always use the term learning adaptation knowledge for knowledge light approaches
gorithms don’t presume a lot of knowledge acquisition work before learning; they use already acquired knowledge inside the system for learning adaptation knowledge.

The learning of parameters used during adaptation from already acquired knowledge is an example for such an approach. An example for that is the learning of the best $k$ for $k$-NN retrieval. This is used in the feature weights learning algorithms VSM (Lowe, 1995) and also in $k-NN_{vmn}$ (Wettischereck and Aha, 1995).

CARMA (Hastings et al., 1995) learn featural adaptation weights with a hill climbing algorithm from the casebase. There is the possibility to learn global adaptation weights or local weights for each prototypical case. The weights are used during adaptation for the determination of the influence of each feature on the target value.

First more complex work in this area was done by (Hanney, 1996; Hanney and Keane, 1996). They use an inductive learning algorithm to extract adaptation knowledge from (the cases in) the casebase.

In this paper we first categorize the knowledge inside a CBR-System with *knowledge containers* first described by (Richter, 1995). Based upon this, we sketch a framework for learning adaptation knowledge from these containers. This could be seen as a starting point for a guideline for the design of adaptation learning algorithms as well as an early starting point for a methodology for knowledge modeling for adaptation learning approaches. Since we focus here on knowledge light approaches, the knowledge elicitation is almost done and the effort here is mainly in the knowledge modeling task.

The learning of parameters used during adaptation from already acquired knowledge is an example for such an approach. An example for that is the learning of the best $k$ for $k$-NN retrieval. This is used in the feature weights learning algorithms VSM (Lowe, 1995) and also in $k-NN_{vmn}$ (Wettischereck and Aha, 1995).

2 Different Sources of Knowledge in a CBR System

Richter (Richter, 1995; Althoff K.-D. and Richter M. M. and Wilke W.,) described four containers in which a CBR system can store knowledge. Knowledge means here domain knowledge as well as problem solving knowledge which describe the “method of application” of the domain knowledge inside the container. The four containers mentioned are:

- The *vocabulary* used to describe the domain,
the case base,

- the similarity measure used for retrieval and

- the solution transformation used during the adaptation.

In general, each container is able to hold all the available knowledge but this is not advisable. The first three containers include compiled knowledge. With "compile time" we mean the development time before actual problem solving, and "compilation" is taken in a very general sense including human knowledge engineering activities. The case base consists of case specific knowledge that is interpreted at run time, i.e. during the process of problem solving. For compiled knowledge, especially if manually compiled by human beings, the acquisition and maintenance task is as difficult as for knowledge-based systems in general. However, for interpreted knowledge the acquisition and maintenance task is potentially easier because it requires updating of the case base only\(^2\).

Part of the attractiveness of CBR comes from the flexibility to pragmatically decide which container includes which knowledge and therefore to choose the degree of compilation versus case interpretation. When developing a CBR system a general aim should be to manually compile as little knowledge as possible and as much as absolutely necessary. More precisely, a CBR system developer has to decide how the knowledge is distributed to the different containers depending on the availability of knowledge and the engineering effort.

In case there is not enough knowledge available to fill one container as requested, there is a need for knowledge transformation from some containers to others. An example for such a transformation can be found in (Globig and Weß, 1993) where an improvement of the similarity measure is learned by knowledge transfer from the case base into the similarity container. If knowledge is transferred into the adaptation container this is a knowledge light approach for learning adaptation knowledge because the knowledge used is already available and coded in some of the other containers.

We will focus here on learning adaptation knowledge by knowledge-transfer from other containers into the adaptation container or improving the adaptation container itself with already acquired adaptation knowledge.

We will provide a general framework that makes it possible to compare and evaluate different approaches to this problem. When developing such

\(^2\) Adding new cases to the case base could also cause a need for re-engineering some of the compiled knowledge, but this should not happen very often during maintaining a CBR system
a learning algorithm, several design decisions have to be made. Some problems and possible solutions concerning these decisions should be discussed at the workshop.

3 A Framework for Learning Adaptation Knowledge

In this chapter we want to describe our framework for learning adaptation knowledge in the light of the discussed knowledge containers.

3.1 How to Learn Adaptation Knowledge from Knowledge Containers

Figure 1 is an abstract view to the process of learning adaptation knowledge with an inductive algorithm. The sources of knowledge are the previous described knowledge containers: vocabulary, similarity measure, the case base and the adaptation container. This knowledge is transformed into adaptation knowledge using a learning algorithm.

Here we focus on inductive algorithms because they generate general knowledge from examples. Often a CBR system consists of examples because the available model of the real world problem is incomplete. First, there must be a selection of knowledge from the containers to learn from. Depending on the kind of selected knowledge and the inductive algorithm used, this data has to be preprocessed into a suitable representation. The result is a set of examples. Every example is characterized by a set of attributes\(^\text{3}\). These attributes are derived from the knowledge in the described containers. Finally it is necessary to integrate the output of the

\(^{3}\) Attributes are the descriptions of the learning examples for the inductive algorithm
learning process with the old adaptation knowledge to form an adaptation container with the improved knowledge.

3.2 Case Study

Let us now apply our framework to the rule learning approach already mentioned in the introduction. This approach towards learning adaptation knowledge has been described by Hanney and Keane (Hanney and Keane, 1996). Their algorithm builds pairs of cases and uses the feature differences of these case pairs to build adaptation rules. We will briefly describe this algorithm in terms of our framework:

- The *preprocessor* builds pairs from all possible cases and extends them by noting the feature differences and the target difference of these cases. Information from the containers 'case base' and 'vocabulary' is needed. Hanney and Keane also suggest to constrain the case pairs by limiting their number or by taking advantage of the similarity measure to select pairs suitable for learning.

- The *example* input for the learning algorithm are the case pairs computed by the preprocessor.

- In a first step the *learning algorithm* builds adaptation rules by taking a case pair's feature differences as preconditions and the target difference as the conclusion. These rules are subsequently refined and generalized to extend the coverage of the rule base. For complexity reasons, these generalizations are only performed at adaptation time. Hanney and Keane also describe how the learning of rules may be constrained and guided by explicitly known domain knowledge which has been manually compiled into the adaptation container before the automatic rule learning takes place. If this kind of information is to be exploited, the preprocessor must also use the adaptation container as a source of input.

- The dynamic part of the *adaptation container* consists of a set of adaptation rules along with some additional information such as confidence ratings for each rule. These confidence ratings indicate the reliability of a rule's information. They are calculated by the learning algorithm based on the degree of generalization that has been applied to generate the associated rule. The algorithm that controls the rule application and the strategy for resolving conflicts is the non-dynamic part of the adaptation container (see (Hanney, 1996; Hanney and Keane, 1996) for details). They also describe an approach
for integrating the learned rules into already known rules. Learning is here the determination and the generalization of the rules and the improvement of the confidence ratings.

4 Discussion and Further Directions

The main point of this paper is to provide a starting point for a framework for learning adaptation knowledge with knowledge light approaches. However, we think that complex adaptation knowledge is hard to acquire with such techniques. Nevertheless, there are a number of problems where such an approach might be useful like classification tasks with numerical or symbolic targets. Also the learning of parameters for more sophisticated adaptation solutions might be useful with such learning algorithms.

1. The learning of parameters used during adaptation from already acquired knowledge is an example for such an approach. One example for that is the learning of the best \( k \) for \( k \)-NN retrieval. This is used in the feature weights learning algorithms VSM (Lowe, 1995) and also in \( k - NN_{vsm} \) (Wettschereck and Aha, 1995).

2. First more complex work in this area was also done by (Hanney, 1996; Hanney and Keane, 1996). They use an inductive learning algorithm to extract adaptation knowledge from the case base. Their algorithm builds pairs of cases and uses the feature differences of these case pairs to build adaptation rules.

Acknowledgments

The authors would like to thank Prof. Michael M. Richter for the helpful discussions and suggestions. This work was funded by the Commission of the European Communities (ESPRIT contract P22196, the INRECA II project: Information and Knowledge Reengineering for Reasoning from Cases) with the partners: AcknoSoft (prime contractor, France), Daimler Benz (Germany), techno (Germany), Irish Medical Systems (Ireland) and the University of Kaiserslautern (Germany).
Bibliography


