Rapid Prototyping of CBR Applications

with the Open Source Tool myCBR

Armin Stahl, Thomas R. Roth-Berghofer

In this paper a novel, freely available tool for rapid prototyping of Case-based Reasoning (CBR) applications is presented. By providing easy to use model generation, data import, similarity modeling, explanation, and testing functionality together with comfortable graphical user interfaces, the tool enables even CBR novices to rapidly create their first CBR applications. Nevertheless, at the same time it ensures enough flexibility to enable expert users to implement advanced applications.

1 Introduction

Compared with other AI approaches, CBR allows reducing the knowledge acquisition and representation effort significantly. Nevertheless, implementing a CBR application from scratch still remains a time consuming software engineering process and requires a lot of specific experience beyond pure programming skills.

Although CBR research has a history of about 20 years now, and in spite of the broad commercial success of CBR applications in recent years, today only few CBR software tools for supporting the development process are available. Software products used for implementing large-scale commercial applications are typically very complex and are connected with high licensing costs. This makes these products little attractive for research, teaching, small commercial projects, or first feasibility studies.

For these purposes, more easily available and less complex CBR tools are required. Unfortunately, such solutions are nearly missing today at all. One exception is the Open Source *JColibri*[1] system, which provides a framework for building CBR systems based on state-of-the-art Software Engineering techniques [3].

In this paper we present the novel Open Source CBR tool *myCBR*[2], developed at the German Research Center for Artificial Intelligence (DFKI). The key motivation for implementing *myCBR* was the need for a compact and easy-to-use tool for building prototype CBR applications in teaching, research, and small industrial projects. Moreover, the tool should be easily extendable in order to facilitate the experimental evaluation of novel algorithms and research results.

The current version of *myCBR* still focuses on the similarity-based retrieval step of the CBR cycle [1], because this is still the core functionality of most CBR applications. While early CBR systems often were based on simple distance metrics, today many CBR applications make use of highly sophisticated, knowledge-intensive similarity measures [8]. The major goal of *myCBR* is to minimize the effort for building CBR applications that require such knowledge-intensive similarity measures. Therefore, it provides comfortable graphical user interfaces for modeling various kinds of attribute-specific similarity measures and for evaluating the resulting retrieval quality. In order to reduce also the effort of the preceding step of defining an appropriate case representation, it includes tools for generating the case representation automatically from existing raw data.

In this article we give an overview of the basic concept and system architecture of *myCBR* and we describe how rapid prototyping of CBR applications is supported by *myCBR*. A more detailed description of myCBR can be found in [9].

2 The myCBR Architecture

The foundation of every CBR system are certain knowledge representation formalisms required to describe the content of the individual CBR knowledge containers, namely the *vocabulary*, the *similarity measure*, the *adaptation knowledge*, and the *case knowledge*. Since knowledge representation is a key issue for most Artificial Intelligence (AI) systems, various software tools for supporting knowledge engineering tasks are already existing today. One of the most popular and widely used systems is certainly the Java-based Open Source ontology editor Protégé[3] with its easily extendable plug-and-play environment.

In order to avoid a reinvention of the wheel, we have chosen Protégé as the modeling platform for *myCBR*. This brings two main advantages: First, the effort for implementing data structures and user interfaces for representing the vocabulary and the case knowledge can be avoided. Second, it allows to add CBR functionality to existing Protégé applications with minimal effort.

The basic architecture of *myCBR* is illustrated in Figure 1. During the development phase of a CBR application, *myCBR* runs as a plug-in within Protégé. This plug-in consists of several modules. The *modeling tools* extend the existing functionality of Protégé for creating domain models and case instances, and add the missing functionality for defining similarity measures. The *retrieval GUI* provides powerful features

[1] http://gaia.fdi.ucm.es/projects/jcolibri
[2] http://www.mycbr-project.net
[3] http://protege.stanford.edu/

Auszug aus: Künstliche Intelligenz, Heft 1/2009, ISSN 0933-1875, BöttcherIT Verlag, Bremen, **www.kuenstliche-intelligenz.de/order**