Science of Computer Programming 79 (2014) 126–145

Contents lists available at ScienceDirect

Science of Computer Programming

journal homepage: www.elsevier.com/locate/scico

jCOLIBRI2: A framework for building Case-based reasoning systems☆

Juan A. Recio-García *, Pedro A. González-Calero, Belén Díaz-Agudo

Facultad de Informática, Universidad Complutense de Madrid, Spain

ARTICLE INFO

Article history:
Received 24 January 2011
Received in revised form 9 April 2012
Accepted 10 April 2012
Available online 24 April 2012

Keywords:
Case-based reasoning
Framework
jCOLIBRI

ABSTRACT

This paper describes the jCOLIBRI2 framework for building Case-based reasoning (CBR) systems. CBR is a mature subfield of artificial intelligence based on the reuse of previous problem solutions – cases – to solve new ones. However, up until now, it lacked a reference toolkit for developing such systems. jCOLIBRI2 aims to become that toolkit and to foster the collaboration among research groups. This software is the result of the experience collected over several years of framework development and evolution. This experience is explained in the paper, together with a description of the specialized CBR tools that can be implemented with jCOLIBRI: CBR with textual cases, recommenders, knowledge/data intensive applications or distributed architectures.

1. Introduction

Case-based reasoning is a subfield of Artificial Intelligence rooted in the works of Roger Schank in the early 80s, on dynamic memory and the central role that the recall of earlier episodes (cases) and scripts (situation patterns) has in problem solving and learning [1]. In spite of its maturity as a field of research, in 2003 when we started to develop jCOLIBRI [2], there was no open source tool that would serve as a reference implementation for the common techniques and algorithms developed and refined in the CBR community over the years. Our goal was to fill this gap by developing an open source framework that could fulfil several goals: to provide a readily useable implementation of common techniques in CBR useful for academic environments; to foster collaboration among research groups by providing an extensible framework which their prototypes can use and extend; to provide a toolset for assembling a CBR system without even writing a line of code. This last goal consists of facilitating the experimentation with different configurations of a CBR system, and enabling the use of the framework for those with a good understanding of the CBR processes but reluctant to program in Java.

As reported by Díaz-Agudo et al. [3], in 2007, with jCOLIBRI1 we had been quite successful in fulfilling two of the three initial goals. The framework was becoming popular as a programming resource in graduate courses in Case-based reasoning, and high-level tools resulted in a significant improvement in the process of configuring CBR systems for research purposes. Nevertheless, the framework was barely attracting any contribution from other research groups. Some colleagues reported that the framework architecture was too biased towards high-level configuration tools, which imposed some artificial constraints on the design of the framework. For example, in order to facilitate the definition of the case structure through a GUI, jCOLIBRI1 included its own type system in parallel with the basic types included in Java. Using terminology from the software reuse community, we had built a successful black-box framework which failed to succeed as a white-box one. jCOLIBRI2 was conceived to solve this problem.

jCOLIBRI2 was designed from the bottom up. First we concentrated on developing a white-box framework that facilitated source code reuse and promotes the integration of a third-party code. Furthermore, by adopting standard Java technologies

☆ Supported by Spanish Ministry of Science & Education (TIN2009-13692-C03-03) and Madrid Education Council and UCM (Group 910494).
* Corresponding author. Tel.: +34 913947599; fax: +34 913947547.
E-mail addresses: jareciog@fdi.ucm.es (J.A. Recio-García), pedro@sip.ucm.es (P.A. González-Calero), belend@sip.ucm.es (B. Díaz-Agudo).

0167-6423/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.scico.2012.04.002

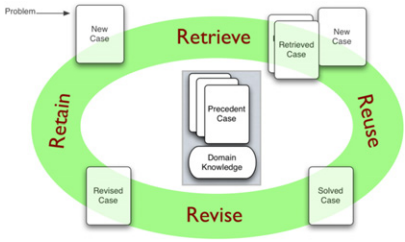

Fig. 1. The CBR Cycle (adapted from Aamodt and Plaza [6]).

for enterprise applications, we extended the scope of the framework which is now useful both for academic and real world applications.

This paper describes the architecture, components and features of jCOLIBRI2. Our goal is to outline the problems found in jCOLIBRI1 to help framework developers avoid them; consequently, to explain how that experience drove us to the new architecture of its successor. Once the architecture of the framework is described, we provide an overview of the components supplied for building CBR systems. Finally, we also explain the capabilities of the framework for building specialized CBR applications.

The rest of the paper runs as follows. The next section briefly introduces the main ideas of Case-based reasoning. Section 3 details lessons learnt from developing jCOLIBRI1 and describes the new architecture proposed for jCOLIBRI2 along with the main components in the framework. Section 4 describes further complements that extend the basic functionality provided by the framework. Section 5 explains how to obtain, install and use the jCOLIBRI2 framework. Section 6 describes some use cases of the framework in third-party large applications. Related work is presented in Section 7 and, finally, Section 8 concludes the paper and provides some hints about future work.

2. Case-based reasoning

Case-based reasoning is a paradigm for combining problem-solving and learning that has become one of the most successful applied subfields of artificial intelligence in recent years [4]. CBR is based on the intuition that problems tend to recur, so that new problems are often similar to previously encountered problems and, therefore, past solutions may be of use in the current situation [5]. When applied to classification problems, CBR, and more specifically instance-based learning, it is considered a lazy learning approach where instead of generating some kind of abstract representation of the set of training examples, it uses those training examples in the neighbourhood of the problem example (k Nearest Neighbours [5]) to determine its class.

Fig. 1 shows a high-level description of a generic CBR system as described by Aamodt and Plaza [6]. In its most general form a CBR system is composed of 4 consecutive processes that, given a collection of precedent cases, some kind of domain knowledge, and a problem to solve or an example to classify:

1. retrieve the most similar cases from the precedent ones;
2. reuse the knowledge in the cases retrieved to propose a solution or classification for the new one;
3. revise the solution proposed by some means external to the system; and
4. learn from the problem-solving or classification episode by retaining the new problem, along with its solution and revision.

Within this general schema we can identify different families of CBR systems, such as:

- Textual CBR, where the cases are given as text in natural language;
- Knowledge-intensive CBR, where a rich domain model is available and thus the system requires a smaller case base;
- Data-intensive CBR, where precedent cases are the main source of information with no domain knowledge available; or
- Distributed CBR, where different agents collaborate to reach conclusions based on their particular case bases.

jCOLIBRI2 includes source code components for supporting such a range of CBR approaches.

CBR, both as a problem-solving paradigm and machine learning approach, is best suited for complex domains where knowledge is scarce. As a problem-solving paradigm, CBR can be used in those domains where a deep domain model is not available but we have a collection of past problem-solving episodes. As a learning technique, CBR can be used in those domains where the number of available training examples, given the number of attributes used to describe the examples, is too small to obtain a useful generalization of the training examples by applying some non-lazy learning technique.

Its ability to cope with complex domains explains why CBR has been successfully applied to a number of real world problems such as diagnosis applications [7], planning [8], medical applications [9], law [10], e-learning [11], knowledge management [12], image processing [13], or recommender systems [14].

For a detailed description of these and other CBR-related aspects, we direct the interested reader to a number of CBR books [15], [5], the special issue of the Knowledge Engineering Review (vol. 20:3) dedicated to reviewing the state of the