Building Case-based Reasoning Applications with *myCBR* and *COLIBRI Studio*

Thomas Roth-Berghofer[1], Juan Antonio Recio García[2], Christian Severin Sauer[1], Kerstin Bach[3], Klaus-Dieter Althoff[3], Belen Díaz Agudo[2], and Pedro A. González Calero[2]

[1] School of Computing and Technology, University of West London, Ealing, London, UK
[2] Department of Software Engineering and Artificial Intelligence, Universidad Complutense de Madrid, Spain
[3] Department of Computer Science, University of Hildesheim, Germany

{christian.sauer|thomas.roth-berghofer}@uwl.ac.uk, belend|pedro|jareciog@fdi.ucm.es, {bach|althoff}@uni-hildesheim.de

Abstract. *myCBR* and *COLIBRI Studio* are two well-established open-source frameworks for building case-based reasoning (CBR) applications, though they follow different approaches and support different phases of the CBR application development. In a nutshell: Where *myCBR* supports its users in developing a knowledge model for representing cases, it more or less leaves the software developers alone when they try to develop an application that uses the generated knowledge model. *COLIBRI Studio*, on the other hand, is focused in the development of applications that use that knowledge model. As soon as you have a knowledge model *COLIBRI Studio* offers templates for a variety of application types and supports in generating its source code. This paper explains the strengths and weaknesses of both frameworks regarding the rapid development of CBR applications. It also shows how to use both of them in conjunction.

1 Motivation

The development of appropriate technologies to enable direct and efficient access to relevant information is one of the key challenges for the knowledge society. Traditional database query languages are optimised to reconcile large volume data stocks with great efficiency based on exactly specified queries. Today, many application scenarios demand more advanced technologies that offer intelligent support to the user when searching for information.

In e-Commerce, for example, the user searching for a product is seldom in a position to formulate an appropriate, exact specification for a query.

Frequently, either the necessary background knowledge about the product offering is lacking or, over or under specified database queries lead to empty or unmanageable results, which is of little help in finding a suitable available product. Intelligent product recommendation systems present an alternative which, even with relatively vague desires and needs on the part of the user, can recommend suitable, target oriented products.

Case-based reasoning is a paradigm for combining problem-solving and learning that has become one of the most successful applied subfields of AI in recent years. It does not rely solely on general knowledge of a problem domain, or making associations along generalised relationships between problem descriptors and conclusions. CBR is able to utilise the specific knowledge of previously experienced, concrete problem situations (cases). A new problem is solved by finding a similar past case, and by reusing its solution in the new problem situation [15]. Now that CBR is a mature and established technology two necessities have became critical: the availability of tools to build CBR systems, and the accumulated practical experience of applying CBR techniques to real-world problems. Many researchers in the field agree about the increasing necessity to formalize this kind of reasoning, define application analysis methodologies, and provide design and implementation assistance with software engineering tools [2,8,4,7].

myCBR is an open-source similarity-based retrieval tool and software development kit (SDK). With *myCBR Workbench*, you can model and test highly sophisticated, knowledge-intensive similarity measures in a powerful GUI and easily integrate them into your own applications using the *myCBR* SDK. *COLIBRI Studio* is the implementation of the top layer of the COLIBRI platform. It provides the visual builder tools required to generate CBR systems without dealing directly with the source code. *COLIBRI Studio* is integrated into the popular Eclipse IDE.

2 Philosophies behind *myCBR* and *COLIBRI Studio*

The development of even a quite simple CBR application already involves a number of steps, such as collecting case and background knowledge, modelling a suitable case representation, defining an accurate similarity measure, implementing retrieval functionality, and implementing user interfaces. Compared to other AI approaches, CBR allows to reduce the effort required for knowledge acquisition and representation significantly, which is certainly one of the major reasons for the commercial success of CBR applications. Nevertheless, implementing a CBR application from

scratch 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 over 20 years, 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.

2.1 *myCBR Workbench* and SDK

Key motivation for implementing *myCBR* was the need for a compact and easy-to-use tool for building prototypical CBR applications in teaching, research, and small industrial projects with low effort. Therefore, *myCBR Workbench* provides comfortable graphical user interfaces for modelling 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, *myCBR Workbench* includes tools for generating the case representation automatically from existing raw data. The accompanying Software Development Kit (SDK) allows for integration into other applications and extension to specific requirements such as additional similarity calculations.

myCBR's foundation can be traced back to several European research projects, namely INRECA, INRECA II, and WEBSELL. All of them were funded by the European Strategic Program on Research in Information Technology (ESPRIT).

In the INRECA project – Induction and Reasoning from Cases (1993-1996) – first tools and methods for developing, validating, and maintaining decision support systems were developed (see, for example, [1]). Among those tools were a descriptive model editor for interactively defining classes, objects, attributes, and relations, and a case manager used this information to build automatically a 'questionnaire' for collecting cases.

The INRECA II project – Information and Knowledge Reengineering for Reasoning from Cases (1996-1999) – not only continued working on those tools but also developed an experience-based methodology and a set of tools that supports the methodology for guiding CBR application development, validation, and maintenance. The INRECA II methodology [2] is composed of a set of steps, guidelines and recommendations that allows a primarily non-CBR user to build, to validate, to maintain, and to scale up a CBR application. These steps include the initial understanding of the CBR technique, the building and the maintenance of a CBR application, and the evaluation, qualification and acceptance process of the CBR system being built. INRECA II showed the applicability

of this CBR methodology to a variety of application tasks such as case-based help desk support, intelligent catalogue search, and maintenance of complex technical equipment. The project also resulted in a set of tools for building integrated case-based reasoning applications. The tools represented a significant commercial opportunity for the industrial partners of the project. Furthermore, INRECA II enhanced CBR technology in the areas of solution adaptation.

The WEBSELL project – Intelligent Sales Assistants for the World Wide Web (1998-2000) – developed Intelligent Sales Support technology and products for the World Wide Web (see, for example, [3]). These products support web shoppers in two aspects of the sales process neglected before on the Web; helping the customer to select or configure the product that meets her needs and supporting the customer in navigating through the space of possible product alternatives.

2.2 The *COLIBRI* Platform

COLIBRI is a platform for developing Case-Based Reasoning (CBR) software. Its main goal is to provide the infrastructure required to develop new CBR systems and its associated software components. *COLIBRI* is designed to offer a collaborative environment where users could share their efforts in implementing CBR applications. It is an open platform where users can contribute with different designs or components that will be reused by other users.

As a platform, *COLIBRI* offers a well defined architecture for designing CBR systems, and a reference implementation of that architecture: the *jCOLIBRI* framework. jCOLIBRI is a white-box tool that permits programmer users to have total control of the internal details of the software and lacks of any kind of visual interface. On the other hand, the platform also includes the graphical development tools to aid users in the development of CBR systems. These tools are enclosed in the *COLIBRI Studio* IDE and generates applications that use the components provided by *jCOLIBRI*.

COLIBRI supports the development of a wide collection of CBR systems: standard CBR systems [5], textual CBR [12], knowledge-intensive [13], data-intensive [6], recommender systems [9], and distributed CBR applications [10]. It also includes evaluation, maintenance and case-base visualization tools. Many of the components available have been developed by third-party research groups and contributed to the platform to be shared with the community.

The underlying idea of the development process proposed in *COLIBRI* is reuse (something inherent in Case-Based Reasoning) of both system designs and their components. The development process defines activities, roles, resources and tools that reduce the development time compared to the effort required to implement CBR systems from scratch. Additionally, it promotes the repeatability of the results reported by other researchers as it simplifies generation of existing CBR systems.

The platform provides a catalogue of already implemented CBR systems represented as workflows, and lets users select the most suitable system and adapt it to the concrete requirements of the target application. These workflows are called *templates* and comprise CBR system designs which specify behaviour but do not explicitly define functional details by means of task. This development process includes several activities regarding the generation, publication, interchange, retrieval and reuse of templates and components. These activities are carried out by different actors and supported by the tools integrated in *COLIBRI Studio*.

3 Building structural knowledge models with *myCBR Workbench*

myCBR offers a workbench that supports the creation and maintenance of the vocabulary and knowledge models for CBR systems. The *myCBR Workbench* is implemented as using the Rich Client Platform (RCP) of Eclipse. *myCBR Workbench* offers two different views - knowledge modeling and case bases. In this section we will focus on the modeling view (see Figure 1).

The conceptual idea behind the modeling view is that first, a case structure is created, followed by the definition of the vocabulary and then the creation of individual local similarity measures for each attribute description (eg. CCM in Figure 1) and the global similarity measure for a concept description (Car in Figure 1). A *concept description* can contain one or more attribute descriptions as well as attributes referencing concepts, which allows the user creating object-oriented case representations. An *attribute description* can be one of the following data types: Boolean, Double, Integer, Interval, String, and Symbol. For each data type *myCBR* provides a similarity functions that support their definition. For prototyping applications, one attribute description can have more than one similarity measure. Among the available functions can be switched by activating them.

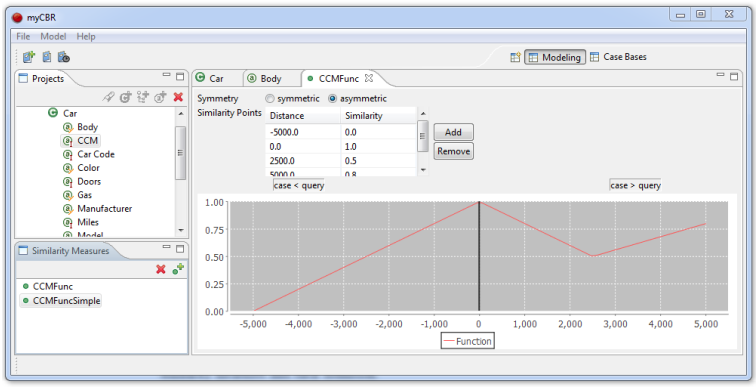

Fig. 1. The Modeling view of *myCBR Workbench* showing the case structure (left), available similarity measures (left bottom) and their definition (center).

Modeling the similarity in *myCBR Workbench* takes place on the attribute level (local similarity measures) and the concept level (global similarity measures). When attributes are defined, the data types and value ranges are given. The *myCBR Workbench* then provides graphically supported modeling: For numerical data it is providing predefined distance functions along with predefined similarity behavior (constant, single step or polynomial similarity decrease). For symbolic values, *myCBR Workbench* provides table functions and taxonomy functions. A table function allows to define for each value pair the similarity value, while a taxonomy subsumes similarity values for subsets of values. Depending on the size of a vocabulary, table similarity measures are hard to maintain and taxonomies allow an easier overview. For symbolic values, also set similarities are provided in order to compare multiple value pairs [14].

4 Building CBR applications with *COLIBRI Studio*

COLIBRI Studio provides the visual builder tools required to generate CBR systems without dealing directly with the source code. It is built on top of the *jCOLIBRI* framework and enables the composition of its CBR components. COLIBRI Studio is integrated into the popular Eclipse IDE. This way it takes advantage of the facilities provided to manage projects and Java source code. Next we briefly describe the tools included in *COLIBRI Studio*.

Project Management. *COLIBRI Studio* takes advantage of the project management features provided by the Eclipse platform. The integration with the Eclipse platform enables the execution and management of the Java source code.

Case Designer. The Case Designer tool is used to define the structure of the cases. In *jCOLIBRI*, cases contain a Description, Solution, Result and Justification components. Each component is composed of several attributes, that can be also simple or compound.

Case Base Selector. Configures the in-memory organization (linear lists, k-d trees, ...) of the case-base once cases are loaded from the persistance media.

Database and plain text connector configurator. Configures how to load cases from text files or databases.

Similarity Defines the similarity configuration of the CBR system. Compound attributes are configured with a global similarity function, whereas simple attributes use local similarity functions. These functions are automatically obtained from the *jCOLIBRI* framework or imported from *myCBR*.

These tools are used to configure the building blocks of a CBR system in *COLIBRI Studio* however, the main activity is consists on creating applications from existing templates. Next we outline this process.

4.1 Template-based design

Development of CBR systems in *COLIBRI Studio* follows a case-based approach where a user *retrieves* a design from a library (case base) of previously designed templates and, if needed, *adapts* it by adding, removing or substituting components in the template selected until the tools are able to generate the source code of the application. This process is named Template-Based Design.

Each template is composed of several task that define the workflow of the system. The functionality defined by each task has to be implemented by using one method of the *jCOLIBRI* framework. In case that *jCOLIBRI* will not provide the desired method, users can program it directly in Java (using Eclipse's facilities) and integrate it in any template.

To aid users in the retrieval of a template that fits her requirements, *COLIBRI Studio* includes a wizard that implements a recommender system of templates. This recommender follows an approach named "retrieval by trying" where templates are associated to a real implementation that allows users to test the behaviour of the CBR system once implemented.

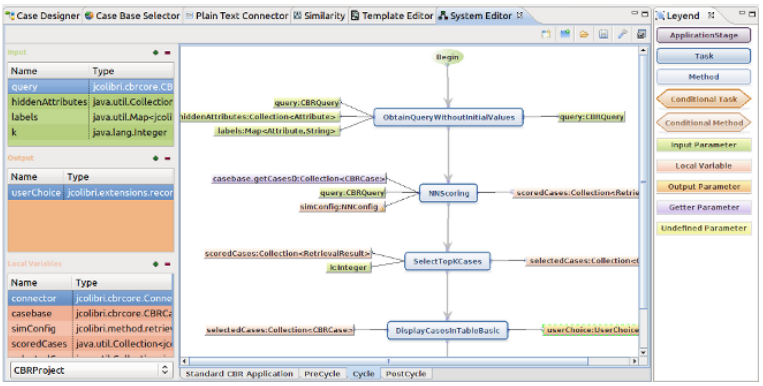

Fig. 2. Template adaptation *COLIBRI Studio*

Once a template has been selected it is adapted by selecting a suitable component for each task. The main tool in *COLIBRI Studio* implements this second stage of the Template-based design by enabling the configuration of templates with components available in the *jCOLIBRI* framework. This tool is shown in Figure 2.

5 Using *myCBR* knowledge models with *COLIBRI Studio*

The *myCBR*Workbench offers a versatile GUI to rapidly prototype a CBR knowledge model of almost any given domain. The weight of the development tools available within the *myCBR* is aimed at the development of cases for structured CBR as well as on the ability to encode as much domain knowledge into the similarity measures of the knowledge model as possible. Starting from the knowledge gathered within a domain to be modelled the knowledge engineer (KE) can define the main concepts and their attributes to form up an initial case structure. *myCBR*offers the possibility to model concept hierarchies to model for example 'part of' relationships of sub-concepts within a domain. The attributes available now are of the types integer number, floating point number, double number and symbol attributes. After assigning the desired attributes of a concept and their value ranges as well as minimum and maximum values allowed the KE can start to model the similarity measures for local similarities for each attribute. *myCBR*offers a range of graphical interfaces to offer the KE the most comfortable approach to model complex similarity measures. For a detailed insight into how these tools can be

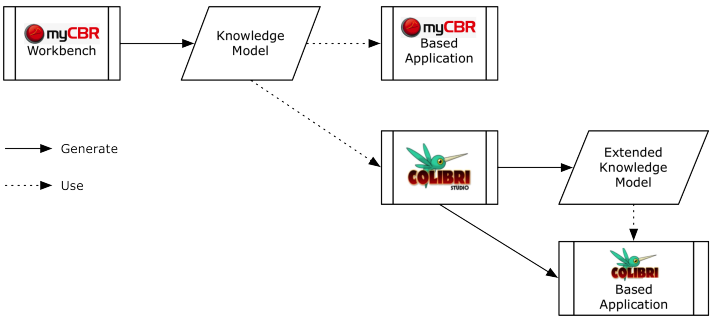

Fig. 3. Basic interplay of *myCBR Workbench*/SDK with *COLIBRI Studio*

used to integrate domain knowledge into the knowledge model we must, for reasons of space within this paper, refer the reader to the documentation of the *myCBR*Software. The documentation is available at this URL: http://mycbr-project.net/tutorials.html. After modelling the local similarity measures *myCBR*also offers a host of options to model the global similarity measures to determine the overall similarity of two cases. After the initial modelling of the domain into a first usable knowledge model the KE is able to test her model within the *myCBR*workbench. To test her knowledge model the KE can run retrieval tests within the workbench and thus evaluate the performance as well as the accuracy of her knowledge model and the cases captured/gathered so far.

Once the knowledge model is ready, it can be integrated in an application generated with the *myCBR* SDK or exported to *COLIBRI Studio* to take advantage of its development capabilities. *COLIBRI Studio*can import those knowledge models created with *myCBR* and integrate them in the template-based design. The importation of the case structure transforms the conceptual model of *myCBR* in java clases that can be integrated in the applications managed by *COLIBRI Studio*. On the other hand, similarity functions are also transformed into java methods that can be used by the k-NN methods in *COLIBRI Studio*. This way, developers can generate complex applications that reuse the experience stored in the templates of the system. *COLIBRI Studio* also provides facilities to evaluate and debug the generated system.

This joint development process is illustrated in Figure 3 and can be summarized as follows:

1. Knowledge models are created using *myCBR*.
   – Case structure.

   – Similarity metrics.
2. The knowledge model is tested and optimized using the retrieval capabilities of *myCBR*.
3. Knowledge model is exported to *COLIBRI Studio*.
4. *COLIBRI Studio*generates a fully functional CBR system that integrates the knowledge model created in *myCBR*.

6 Case study

To illustrate the joint development process using *myCBR* and *COLIBRI Studio* we will use a simple example consisting of a basic CBR application about cars.

myCBR allows to define the representation of the cases by defining their attributes and value restrictions. Then, similarity metrics can be defined for every attribute. In our example, this definition is shown in Figure 4. We can observe that a `Car` is defined by several attributes such as `Body`, `Color`, `Miles`, etc. Associated to every attribute there is a similarity function (*SMF* in the Figure) that will compare the attribute values taken from the query and the case being compared. *myCBR* provides several tools to define complex similarity functions. Additionally, each attribute has a *weight* that indicates its importance in the global concept `Car`. These weights are used in the weighted average that obtains a global similarity value between the query and a given case.

Once the case representation and the similarity metrics have been configured *myCBR* includes a tool to perform the k Nearest Neighbour (kNN) retrieval. This tool can be used to optimise the performance of the retrieval process by modifying the weights and similarity metrics.

Using *myCBR* we have created the knowledge model that defines the case structure and retrieval process. In order to create a ready to use application, we have to export that knowledge model to *COLIBRI Studio*. *COLIBRI Studio*has its own tool for defining case structures and similarity metrics that are able to import those knowledge models created by *myCBR*. These tools are shown in Figure 4. Once these tools have imported the case structure and similarity configuration, the CBR system can be generated using the Template-Based Design process supported by *COLIBRI Studio*and explained in Section 4.

7 Conclusion and future work

Case-based reasoning (CBR) is a mature and established AI technology where two necessities have become critical: the availability of tools to build

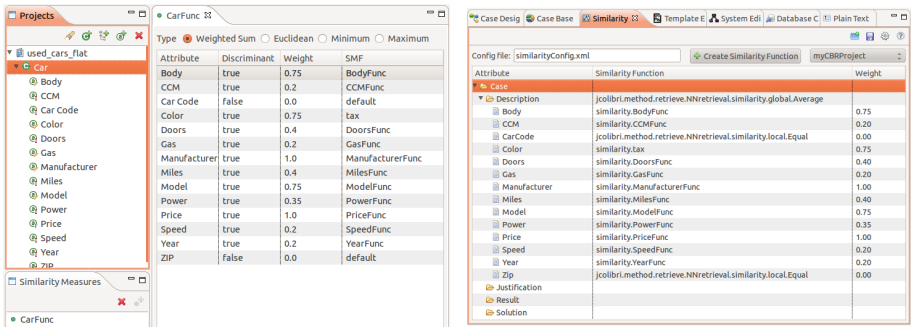

Fig. 4. Comparison of the case representations in *myCBR*(left) and *COLIBRI Studio*(right)

CBR systems, and the accumulated practical experience of applying CBR techniques to real-world problems [11].

The authors of this paper have done a joint effort to identify the main features of the tools that up to date are being used for the CBR community.

The main feature of *COLIBRI* is that it is a collaborative environment where users share their effort in implementing CBR applications. It benefits from the reuse of previously developed CBR systems and provides a catalogue of already implemented systems, that are reused in a "CBR way". *myCBR* on the other hand offers a workbench to create the knowledge models for CBR systems, like the case structure and similarity measures.

We have illustrated how to use both tools in a simple application on the car domain. Using *myCBR* we have created the knowledge model that defines the case structure and retrieval process. This knowledge model is exported and used in *COLIBRI Studio* to create a running CBR application based on the previously defined system templates.

This work is a preliminary collaboration approach of the two teams meanwhile the underlying philosophy of each tool is taken into account. As future work we will provide an integrated, joint interface from which all the functionalities of both tools can be used.

References

1. Bergmann, R.: Highlights of the european INRECA projects. In: Aha, D.W., Watson, I. (eds.) Case-Based Reasoning Research and Development: Proceedings of the

Fourth International Conference on Case-Based Reasoning, ICCBR 2001, Vancouver, Canada. pp. 1–15. Springer-Verlag, Berlin (2001)
2. Bergmann, R., Althoff, K.D., Breen, S., Göker, M., Manago, M., Traphöner, R., Wess, S.: Developing Industrial Case-Based Reasoning Applications: The INRECA Methodology. LNAI 1612, Springer-Verlag, Berlin, second edn. (2003)
3. Cunningham, P., Bergmann, R., Schmitt, S., Traphöner, R., Breen, S., Smyth, B., Weber, R., von Wangenheim, C.G.: Intelligent support for online sales: The websell experience. In: Proceedings of the Workshop Program at the Fourth International Conference on Case-Based Reasoning (ICCBR 2001) (2001)
4. Díaz-Agudo, B., González-Calero, P.A.: An architecture for knowledge intensive CBR systems. In: Blanzieri, E., Portinale, L. (eds.) EWCBR. Lecture Notes in Computer Science, vol. 1898, pp. 37–48. Springer (2000)
5. Díaz-Agudo, B., González-Calero, P.A., Recio-García, J.A., Sánchez, A.: Building cbr systems with jcolibri. Special Issue on Experimental Software and Toolkits of the Journal Science of Computer Programming 69(1-3), 68–75 (2007)
6. Fornells, A., Recio-García, J.A., Díaz-Agudo, B., Golobardes, E., Fornells, E.: Integration of a methodology for cluster-based retrieval in jcolibri. In: McGinty, L., Wilson, D.C. (eds.) ICCBR. Lecture Notes in Computer Science, vol. 5650, pp. 418–433. Springer (2009)
7. Funk, P., González-Calero, P.A. (eds.): Advances in Case-Based Reasoning, 7th European Conference, ECCBR 2004, Madrid, Spain, August 30 - September 2, 2004, Proceedings, Lecture Notes in Computer Science, vol. 3155. Springer (2004)
8. Jaczynski, M., Trousse, B.: An object-oriented framework for the design and the implementation of case-based reasoners. In: Proceedings of the 6th German Workshop on Case-Based Reasoning (1998)
9. Recio-García, J.A., Díaz-Agudo, B., González-Calero, P.A.: Prototyping Recommender Systems in jCOLIBRI. In: Proceedings of the 2008 ACM conference on Recommender Systems. pp. 243–250. ACM (2008)
10. Recio-García, J.A., Díaz-Agudo, B., González-Sanz, S., Quijano-Sánchez, L.: Distributed deliberative recommender systems. T. Computational Collective Intelligence 1, 121–142 (2010)
11. Recio-García, J.A., Díaz-Agudo, B., Sánchez, A., González-Calero, P.A.: Lessons learnt in the development of a cbr framework. In: Petridis, M. (ed.) Proccedings of the 11th UK Workshop on Case Based Reasoning. pp. 60–71. CMS Press,University of Greenwich (2006)
12. Recio-García, J.A., Díaz-Agudo, B., Gómez-Martín, M.A., Wiratunga, N.: Extending jCOLIBRI for textual CBR. In: Procs of the 6th Int. Conf. on CBR, ICCBR 2005. LNAI, vol. 3620, pp. 421–435. Springer, US (2005)
13. Recio-García, J.A., Díaz-Agudo, B., González-Calero, P.A., Sánchez-Ruiz-Granados, A.: Ontology based cbr with jcolibri. In: Ellis, R., Allen, T., Tuson, A. (eds.) Procs. of the Twenty-sixth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence. pp. 149–162. Springer, Cambridge, United Kingdom (December 2006)
14. Stahl, A., Roth-Berghofer, T.R.: Rapid prototyping of CBR applications with the open source tool myCBR. In: Proceedings of the 9th European conference on Advances in Case-Based Reasoning. pp. 615–629. Springer-Verlag, Heidelberg (2008)
15. The CBR community: Case-Based Reasoning wiki. Online (2012), `http://cbrwiki.fdi.ucm.es`