Knowledge Modeling with the Open Source Tool myCBR

Kerstin Bach[1], Christian Sauer[2], Klaus Dieter Althoff[3], and Thomas Roth-Berghofer[2]

[1] Verdande Technology AS
Trondheim, Norway
http://www.verdandetechnology.com
[2] School of Computing and Technology
University of West London, United Kingdom
http://www.uwl.ac.uk
[3] Competence Center Case-Based Reasoning (CC CBR)
German Research Centre for Artificial Intelligence, Kaiserslautern, Germany
http://www.dfki.de/web/competence/cccbr

Abstract. Building knowledge intensive Case-Based Reasoning applications requires tools that support this on-going process between domain experts and knowledge engineers. In this paper we will introduce how the open source tool *myCBR 3* allows for flexible knowledge elicitation and formalisation form CBR and non CBR experts. We detail on *myCBR 3*'s versatile approach to similarity modelling and will give an overview of the Knowledge Engineering workbench, providing the tools for the modelling process. We underline our presentation with three case studies of knowledge modelling for technical diagnosis and recommendation systems using *myCBR 3*.

1 Introduction

Case-Based Reasoning (CBR) is a methodology introduced by Riesbeck and Schank [13] and Kolodner [8] who derived its basic principles from cognitive science. They describe how humans manage and reuse their experience described in episodes. Aamodt and Plaza [2] introduce a basic model for developing CBR applications. It consists of four processes: Retrieve, Reuse, Revise and Retain. The CBR process requires cases that consist of problem and solution description. Problem descriptions are usually attributes values describing a problematic or critical situation while the solution contains information on how to solve the given problem. In the retrieve phase, the attributes describing a problem are matched against cases in a case base. The best n cases are returned. In order to match a given situation these cases can be adapted (Reuse). In the revision phase, reused cases are verified before they are retained.

CBR systems always carry out the retrieve phase which is characterized by a similarity-based comparison of features, while the remaining phases can omitted. Richter [12] introduced to model of four knowledge containers describe the required knowledge within a CBR system:

- *Vocabulary* defining the range of allowed values for attributes. For numeric values this is usually the value range (minimum, maximum) while for symbolic values this can be a list of values.
- *Similarity Measures* defining the relationship between attribute values in form of a similarty assignments. Similarity measures can be formulas like the hamming distance for numeric values or reference tables for symbolic values.
- *Adaptation Knowledge* is knowledge describing how cases can be adapted in the reuse step, often represented as rules.
- *Cases* are instances describing situations that have happened and are worth capturing in order to be reused. They instantiate attributes describing the problematic situation as well as a solution description. Their degree of formalization can vary.

Developing CBR systems requires a systematic development of knowledge models by defining the requirements and building the models itself. *myCBR 3*[4] is an open source tool targeting at developing customized knowledge models with an emphasis on vocabulary and similarity measure development. *myCBR 3* is an open-source similarity-based retrieval tool and software development kit (SDK). With *myCBR 3* 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 3* SDK[3]. Case-based product recommender systems are just one example of similarity-based retrieval applications.

In the remaining of this paper we will give an overview of other CBR tools and applications (section 2) as well as showcase the functionalities of *myCBR 3* (section 3). In section 4 we will show how *myCBR 3* has been applied in different CBR projects while the final section will sum up the paper and give an outlook on future work on the tool.

2 Related Research

Freely available CBR tools are for instance FreeCBR, jCOLIBRI or eXiT*CBR, which will be briefly discussed in this section. FreeCBR[5] is a rather simple CBR engine, which allows the realization of basic CBR features. However, it does not cover features like case revision or retention and more individualized knowledge models, or comprehensive global and local similarity measures, are not applicable either. Further, it still requires quite some effort to apply it to a high variety of tasks. jCOLIBRI started from a task oriented framework also covering distributed reasoning [10], recently jCOLIBRI Studio [11] for more comprehensive support of building CBR knowledge has been introduced. Up to today jCOLIBRI includes more machine learning and semantic web features while *myCBR 3* focused on the knowledge required in the knowledge containers.

[4] http://www.mycbr-project.net
[5] `http://freecbr.sourceforge.net/`

COLIBRI is another platform for developing Case-Based Reasoning (CBR) CBR software. COLIBRI's main goal, opposed to *myCBR 3*, is to provide the infrastructure required to develop new CBR systems and its associated software components, rather than a CBR knowledge model. COLIBRI is designed to offer a collaborative environment. It is an open platform where users can contribute with different designs or components of CBR systems, which will be reused by other users. Subsequently many of the components available have been developed by third-party research groups and contributed to the platform to be shared with the community.

As a platform, COLIBRI offers a well-defined architecture for designing CBR systems. COLIBRI also provides a reference implementation of that architecture: the jCOLIBRI framework. jCOLIBRI is a white-box tool that permits system designers to have total control of the internal details of the software. The platform also includes graphical development tools to aid users in the development of CBR systems. These tools are enclosed in the COLIBRI Studio IDE and generate applications that use the components provided by jCOLIBRI.

Furthermore, creating individualized case representations and especially flexible similarity measures is the strength of *myCBR 3*. eXiT*CBR has also its roots in machine learning applications and is specialized for medical diagnosis tasks [9]. It has recently been extended in order to cope with more than one case base. In comparison to *myCBR 3*, the ideas behind the methodology also differ, since we are focusing on the knowledge container model rather than the machine-learning-related tasks. The integration of Drools in an existing framework for executing rules on a given corpus has been introduced by Hanft et al. [7]. In this paper Drools has been integrated in an existing OSGi environment. The approach presented here required a more comprehensive customization since *myCBR 3* was not embedded in OSGi and the requirements for the rules differed in terms of usable knowledge and modification of cases.

In industry, most prominent CBR tools or CBR related technologies are used by empolis in the on SMILA[6] based Information Access Suite[7] as well as by Verdande Technology in DrillEdge [8] [6] . The Information Access Suite has been applied in various help-desk scenario applications as well as in document management while DrillEdge focuses on predictive analytics in oil well drilling. Both companies run proprietary implementations based on academic software - CBR-Works [17] and Creek [1] respectively.

3 Knowledge Engineering in myCBR

myCBR 3 is an open-source similarity-based retrieval tool and software development kit (SDK)[18]. With *myCBR 3* 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 3* SDK.

[6] `https://www.eclipse.org/smila/`
[7] `http://www.empolis.com`
[8] `http://www.verdandetechnology.com`

Case-based product recommender systems are just one example of similarity-based retrieval applications.

The *myCBR 3* Workbench provides powerful GUIs for modelling knowledge-intensive similarity measures. The Workbench also provides task-oriented configurations for modelling your knowledge model, information extraction, and case base handling. Within the Workbench a similarity-based retrieval functionality is available for knowledge model testing. Editing a knowledge model is facilitated by the ability to use structured object-oriented case representations, including helpful taxonomy editors as well as case import via CSV files.

The *myCBR 3* Software development Kit (SDK) offers a simple-to-use data model on which applications can easily be built. The retrieval process as well as the case loading, even from considerably large case bases, are fast and thus allow for seamless use in applications built on top of a *myCBR 3* knowledge model.

Within *myCBR 3* each attribute can have several similarity measures. This feature allows for experimenting and trying out different similarity measures to record variations. As you can select an appropriate similarity measure at runtime via the API, you can easily accommodate for different situations or different types of users.

The *myCBR 3* Workbench is implemented as using the Rich Client Platform (RCP) of Eclipse and offers two different views to edit either knowledge models or case bases. In this section we will focus on the modelling view as shown in 1.

The conceptual idea behind the modelling view is that first a case structure is created, followed by the definition of the vocabulary and the creation of individual local similarity measures for each attribute description (eg. CCM in 1) followed by the global similarity measure for a concept description (Car in 1).

The modelling view of the *myCBR 3* Workbench (see figure 1) is showing the case structure (left), available similarity measures (left bottom) and their definition (center). Modelling the similarity in the Workbench takes place on the attribute level for local similarity measures and the concept level for global similarity measures.

3.1 Building a Vocabulary

The vocabulary in *myCBR 3* consists of concepts and attributes. 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. In the current version *myCBR 3* also allows for the import of vocabulary items, e.g. concepts and attributes, from CSV files as well as from Linked (Open) Data (LOD) sources.

Datatypes An attribute description can have one of the following data types: Double, Integer, String, Date and Symbol. When attributes are defined, the data types and value ranges are given with initial default values and can be set to the desired values in the GUI.

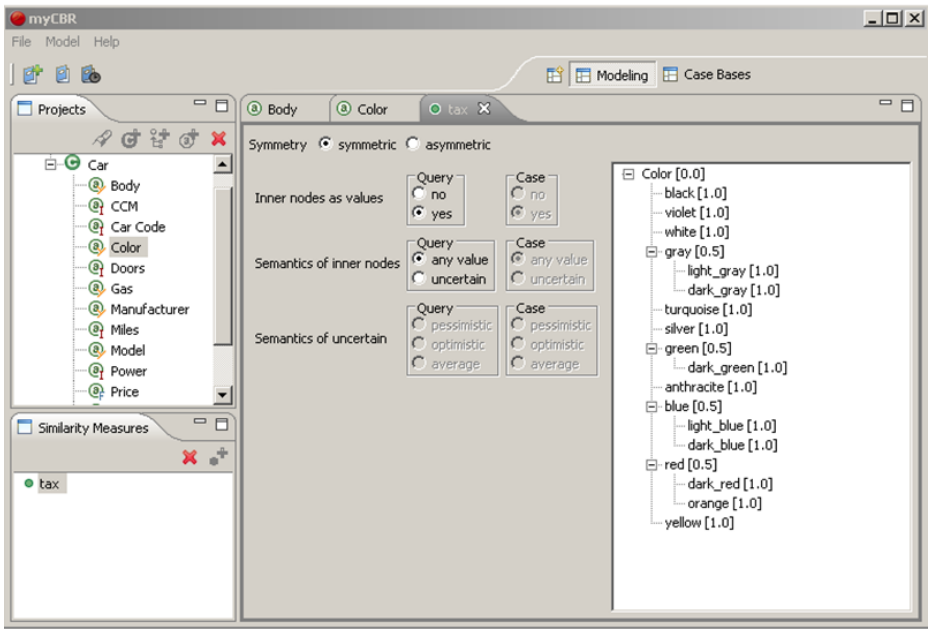

Fig. 1. Example view of the knowledge model view in the *myCBR 3* workbench

3.2 Building Similarity Measures

The Workbench provides graphically supported modelling of similarity functions that support their definition. As an attribute description can have more than one similarity measure experimenting with knowledge modelling approaches is facilitated. For numerical data it is providing predefined distance (or similarity) functions along with predefined similarity behaviour (constant, single step or polynomial similarity decrease). For symbolic values, *myCBR 3* Workbench provides table functions and taxonomy functions. A table function allows defining 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. For each of the similarity measures, as well as for the global similarity measure(s) a specific, versatile editor GUI is provided.

4 Case Study

4.1 Creating Knowledge from Unstructured Documents

This approach has been developed in machine diagnosis based on experiential knowledge from engineers[4]. Most vehicle companies provide service after delivering their machines to customers. During the warranty period they are able

to collect data about how and when problems occurred. They take this data for improving vehicles in different ways: collected data can go back in the product development process, it can be used for improving diagnostic systems to repair them at dealerships or in the factory and also educating service technicians repairing vehicles abroad. This is extremely important if vehicles cannot easily be taken back to factory, e.g. services for aircrafts or trucks.

Such machine manufacturers collect information about machine problems that are submitted by technicians containing machine data, observations and sometimes further correspondence between an engineer or analyst with the technician at the machine. In the end, these discussions usually come up with a solution - however, the solution is normally neither highlighted nor formalized and the topics and details discussed highly depend on the technician and what the Customer Support asks for. That is the reason why cases that are stored for collecting Customer Support information can not directly be used for CBR. Therefore we will differentiate between Customer Support Cases (CS Cases) and CBR Cases. CS Cases contain original information collected during a discussion between Customer Support and the field technician, while a CBR Case contains only relevant information to execute a similarity based search on the cases. The CBR cases content represents each CS Case, but contains machine understandable information.

For building the vocabulary, we extracted all nouns and organized them in attribute values, which were directly imported into *myCBR 3* and from there discussed with the experts. Especially the given taxonomies provided great feedback, because we were discussing both, the terms as well as their relationship. Further, the workbench provided great feedback in explaining CBR because the information the CBR engine uses gets visible. Experts can see local and global similarity measures as well as they can adjust weightings. After 4 sessions with the experts we had a status where the case formats and vocabulary was ready to be deployed in a prototype.

Throughout the project we kept using the workbench when refining case formats as well as similarity measure until the experts themselves started looking into the knowledge models themselves.

On the application's backend, we used the *myCBR 3* SDK to develop a web-based application that searches for similar customer cases after entering all available machine data and observations. Because of the modularity, we were able to deploy updated knowledge models smoothly into the application.

4.2 Knowledge Formalisation for Audio Engineering

A case study on creating a case-based workflow recommendation system for audio engineering support was performed in 2013 [14]. In this study the approach to formalise the special vocabulary used in audio engineering, consisting of vague descriptors for timbres, amounts and directions, was developed. The study introduced CBR as a methodology to amend the problem of formalising the vagueness of terms and the variance of emotions invoked by the same sound in different humans. It was further detailed that the researchers opted for the use of CBR

due to CBR's ability to process fuzzy and incomplete queries and the ability to choose between grades of similarity of retrieved results to emulate the vagueness. The relations between timbres, amounts and effects, were modelled into the local similarity measures of the initial CBR knowledge model as they compose the overall problem description part of what was later used as a case in the resulting CBR engine.

A challenge during this case study was encountered in the form of the task of finding an optimal grade of abstraction for the frequency levels in audio engineering within the CBR knowledge model. This was of importance as in any knowledge formalisation task, one is facing the trade-off between an over engineered, too specific knowledge model and the danger of knowledge loss by employing too much abstraction e.g. choosing the abstraction levels too high. The challenge was met by the researchers by choosing two additional abstraction levels of frequency segments for the timbre descriptors[14].

The next knowledge modelling step consisted of determining the best value ranges for the numerical attributes which were to be integrated into the initial knowledge model. After discussing this approach with the domain experts, the researchers agreed to use two ways to represent *amounts* in the knowledge model. The first way used a percentage approach, ranging from 0 to 100% and the second way used a symbolic approach. The symbolic approach was chosen because the domain experts mentioned that from their experience the use of descriptors for amounts, such as *'a slight bit'* or *'a touch'* were by far more common in audio mixing sessions then a request like *'make it 17% more airy'*. So the researchers integrated, next to the simple and precise numerical approach, a taxonomy of amount descriptors into the initial knowledge model. The taxonomy was ordered based on the amount the symbol described, starting from the root, describing the highest amount down to the leaf symbols describing synonyms of smallest amounts.

The researchers used the *myCBR 3* Workbench to swiftly transfer their initial elicited knowledge model into a structured CBR knowledge model. Figure 2 provides an insight in the modelling of the local similarity measure for timbre descriptors. The first figure shows the taxonomic modelling on the left and a section from the same similarity measure being modelled in a comparative symbolic table on the right.

Within *myCBR 3* the researchers had the choice between a taxonomic and a comparative table approach. Considering the versatile use of taxonomies in structural CBR[5] the researchers initially opted for the use of taxonomies. Yet regarding the complex similarity relationships between the elicited timbre descriptors the researchers also investigated whether a comparative table approach for modelling the similarities of the timbre descriptors. Experiments to establish the performance and accuracy of both approaches yielded no significant difference in the performance of the similarity measures but taxonomies were found to be more easily and intuitively elicited from the audio engineer experts.

After the initial knowledge model was created the researchers performed a number of retrieval experiments using the *myCBR 3* built in retrieval test-

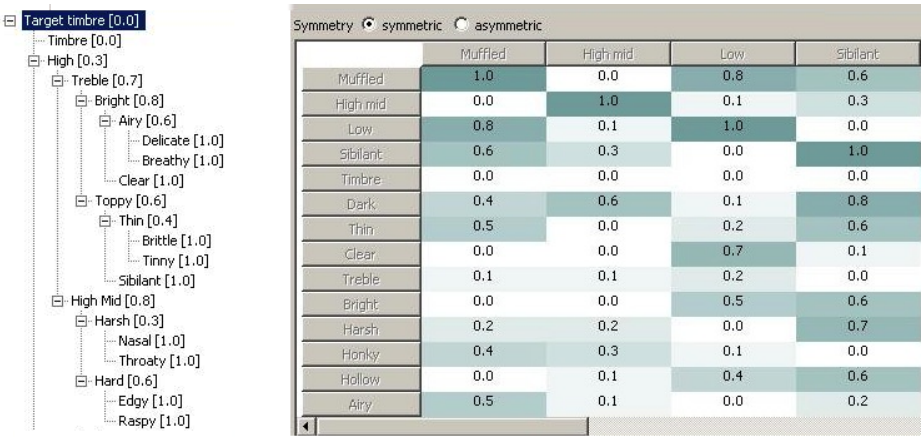

Fig. 2. Timbre descriptor taxonomy and comparative table

ing facilities. The goal of these tests was to refine the initial knowledge model, specifically the similarity measures for the timbre descriptors. Additionally the researchers used the feedback from domain experts to streamline the case structure to the most important attributes. This streamlining was performed within a live system and the researchers were able to directly integrate the streamlined CBR engine into their *Audio Advisor* application thanks to *myCBR 3*'s flexible API.

4.3 Knowledge Formalisation for Hydrometalurgy Gold Ore Processing

In this case study a twofold approach to elicitating and formalising knowledge in the domain of hydrometallurgical processing of gold ore was researched. The study demonstrated processes of formalising hydrometallurgy experts knowledge into two different CBR knowledge models. The first knowledge model was than employed in the *Auric Adviser* workflow recommender software [16].

As the study employed CBR as the reasoning component for the *Auric Adviser* software the researchers had to formalise the gathered knowledge into the knowledge representation structure used for CBR, namely the four knowledge containers; Vocabulary, Cases, Similarity Measures, and Adaption knowledge. Based on the knowledge gathered from the domain experts the researchers created an initial knowledge model and distributed the knowledge into the 4 knowledge containers of CBR in the following way: The vocabulary consisted of 53 attributes, mainly describing the ore and mineralogical aspects of an ore deposit. With regard to the data types used, the researchers used 16 symbolic, 26 floating point, 6 boolean and 5 integer value attributes. The symbolic attributes described minerals and physical characteristics of minerals and gold particles, such as their distribution in a carrier mineral. Further symbols were elicited to

describe the climate and additional contexts a mining operation can be located in, like for example the topography.

The cases were distinctive mainly with regard to the mineralogical context of the mined ore. Thus the researchers created 5 cases describing refractory arsenopyritic ores, 5 describing free milling gold ores, 2 on silver rich ores, 6 cases on refractory ores containing iron sulphides, 4 on copper rich ores and one each on antimony sulphide rich ores, telluride ore and carbonaceous ore.

	ongCrystalBoundari	4longCrystalDefects	loidalParticlesInSolu	BetweenGrains	SolidSolution	rainEnclosedinMiner	Free
ongCrystalBoundari	1.0	0.8	0.5	0.9	0.7	0.2	0.6
4longCrystalDefects	0.8	1.0	0.5	0.9	0.7	0.2	0.6
loidalParticlesInSolu	0.5	0.5	1.0	0.5	0.7	0.2	0.5
BetweenGrains	0.9	0.9	0.5	1.0	0.8	0.2	0.6
SolidSolution	0.7	0.7	0.7	0.8	1.0	0.2	0.7
rainEnclosedinMiner	0.2	0.2	0.2	0.2	0.2	1.0	0.4
Free	0.6	0.6	0.5	0.6	0.7	0.4	1.0
PartialExposed	0.5	0.5	0.5	0.5	0.5	0.2	0.4

Fig. 3. Example of a similarity measure for the gold distribution within an ore

To compute the similarity of a query, composed of prospective data, and a workflow case, the researchers modelled a series of similarity measures. For which the researchers had the choice between comparative tables, taxonomies and integer or floating point functions. For their initial knowledge model the researchers mainly relied on comparative tables.

The study's approach included the idea to model as much of the complex knowledge present in the domain of ore refinement into the similarity measures as possible. This was based on the assumption that the similarity based retrieval approach provided by the use of CBR would allow to capture and counter most of the vagueness still associated with the selection of the optimal process in the hydrometallurgical treatment of refractory ores domain. For example, it was possible to model into the similarity measures such facts as that the ore does not need any more treatment if it contains gold grains greater than 15 micro meters in diameter. Such facts are easy to integrate into the similarity measure and thus are operational (having an effect) in the knowledge model. The researchers deemed this capability of the similarity measures to capture and represent such 'odd' behaviours of the knowledge model very important. The study assumes also that these 'odd' facts or bits of knowledge are hard to capture by rules, and thus has ultimately kept another, rule-based approach of modelling the hydrometallurgical domain knowledge, IntelliGold, from succeeding on a broad scale [19].

For the global similarity measure of the cases the researchers used a weighted sum of the attributes local similarities. This allowed for the easy and obvious emphasise of important attributes, such as for example ' Clay Present', as the presence of clay forbids a selection of hydrometallurgical treatments. As the study mainly aiming for case retrieval, the need for adaptation knowledge was

minor. Therefore the researchers did not formalised any adaption knowledge. The retrieval results achieved with the first knowledge model was described as satisfying in accuracy and applicability by domain experts.

5 Conclusion and Future Work

In this paper we have presented the approach to knowledge formalisation within *myCBR 3*. *myCBR 3* emphasised the fact that *myCBR 3* is a very versatile tool to create CBR knowledge models with a particular versatile suit of editors for similarity modelling. Within the case studies presented it became apparent that the GUI's offered by *myCBR 3* are intuitive to domain experts, particularly with regard to let domain experts without prior knowledge of CBR use them to model their domain knowledge.

Furthermore, also based on experiences from the case studies, we demonstrated that *myCBR 3* allows for on-going knowledge model improvement, even in a running application. This fact allows also for knowledge maintenance and refinement in live CBR applications and also enables developers to follow the rapid prototyping approach in their projects. As shown in previous a research cooperation with COLIBRI, as well as in a research cooperation on similarity of event sequences, *myCBR 3* is particular versatile for similarity measure based knowledge modelling. Furthermore *myCBR 3* is also easily extendable with regard to its SDK and API to cater for any kind of new similarity measures.

For future work we are currently reviewing prototype implementations of additional features for *myCBR 3*. These additional features comprise the ability of automatic extraction of vocabulary items and similarity measures from web community data, the incorporation of drools for the generation of adaption knowledge and the incorporation of case acquisition from databases. Furthermore we are currently finishing the work on the next release of *myCBR 3*, reaching version 3.1. We are also in the process of integrating a mobile version of *myCBR 3*, catering for the needs of android application, such as fast access to assets in a future version of *myCBR 3* [15].

References

1. Aamodt, A.: Knowledge-intensive case-based reasoning in creek. In: Funk, P., Gonzlez-Calero, P.A. (eds.) Proceedings of the ECCBR 2004. LNCS, vol. 3155, pp. 1–15. Springer (2004)
2. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. Artificial Intelligence Communications, 7(1), 39–59 (1994)
3. Bach, K., Althoff, K.D.: Developing case-based reasoning applications using mycbr 3. In: Agudo, B.D., Watson, I. (eds.) Case-Based Reasoning Research and Development, LNCS, vol. 7466, pp. 17–31. Springer Berlin Heidelberg (2012)
4. Bach, K., Althoff, K.D., Newo, R., Stahl, A.: A case-based reasoning approach for providing machine diagnosis from service reports. In: Ram, A., Wiratunga, N.

(eds.) Case-Based Reasoning Research and Development (Procs. of the 19th International Conference on Case-Based Reasoning). vol. 6880, pp. 363–377. Springer Verlag, Berlin Heidelberg (2011)
5. Bergmann, R.: Experience Management: Foundations, Development Methodology, and Internet-Based Applications, Lecture Notes in Computer Science, vol. 2432. Springer (2002)
6. Gundersen, O.E., Sørmo, F., Aamodt, A., Skalle, P.: A real-time decision support system for high cost oil-well drilling operations. AAAI Publications, Twenty-Fourth IAAI Conference (2012)
7. Hanft, A., Schäfer, O., Althoff, K.D.: Integration of drools into an osgi-based bpm-platform for cbr. In: Agudo, B.D., Cordier, A. (eds.) ICCBR-2011 Workshop Proceedings: Process-Oriented CBR (2011)
8. Kolodner, J.: Case-based reasoning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
9. López, B., Pous, C., Gay, P., Pla, A., Sanz, J., Brunet, J.: exit*cbr: A framework for case-based medical diagnosis development and experimentation. Artif. Intell. Med. 51(2), 81–91 (Feb 2011)
10. Recio-García, J.A., Díaz-Agudo, B., González-Calero, P.A.: A distributed cbr framework through semantic web services. In: Bramer, M., Coenen, F., Allen, T. (eds.) Research and Development in Intelligent Systems XXII (Proc. of AI 2005). pp. 88–101. Springer (December 2005)
11. Recio-García, J.A., Díaz-Agudo, B., González-Calero, P.A.: Template based design in colibri studio. In: Proceedings of the Process-oriented Case-Based Reasning Workshop at ICCBR'11. pp. 101–110 (2011)
12. Richter, M.M.: Introduction. In: Lenz, M., Bartsch-Spörl, B., Burkhard, H.D., Wess, S. (eds.) Case-Based Reasoning Technology – From Foundations to Applications. LNAI 1400, Springer-Verlag, Berlin (1998)
13. Riesbeck, C.K., Schank, R.C.: Inside case-based reasoning. Lawrence Erlbaum Associates, Pubs., Hillsdale, N.J. (1989)
14. Sauer, C., Roth-Berghofer, T., Auricchio, N., Proctor, S.: Recommending audio mixing workflows. In: Case-Based Reasoning Research and Development, pp. 299–313. Springer (2013)
15. Sauer, C.S., Hundt, A., Roth-Berghofer, T.: Explanation-aware design of mobile mycbr-based applications. In: Case-Based Reasoning Research and Development, pp. 399–413. Springer (2012)
16. Sauer, C.S., Rintala, L., Roth-Berghofer, T.: Knowledge formalisation for hydrometallurgical gold ore processing. In: Research and Development in Intelligent Systems XXX, pp. 291–304. Springer (2013)
17. Schulz, S.: Cbr-works - a state-of-the-art shell for case-based application building. In: Proceedings of the 7th German Workshop on Case-Based Reasoning, GWCBR'99, Wrzburg. pp. 3–5. Springer-Verlag (1999)
18. 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)
19. Torres, V.M., Chaves, A.P., Meech, J.A.: Intelligold-an expert system for gold plant process design. Cybernetics & Systems 31(5), 591–610 (2000)