Explanation Capabilities of the Open Source Case-Based Reasoning Tool *myCBR*

Thomas R. Roth-Berghofer[1,2] and Daniel Bahls[1,2]

[1] Knowledge Management Department,
German Research Center for Artificial Intelligence DFKI GmbH
Trippstadter Straße 122, 67663 Kaiserslautern, Germany

[2] Knowledge-Based Systems Group, Department of Computer Science,
University of Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern

{thomas.roth-berghofer,daniel.bahls}@dfki.de

Abstract. This paper describes the various explanation capabilities of the open source case-based reasoning tool *myCBR*. *myCBR* features *conceptual explanations*, which provide information about concepts of the application domain, *backward explanations*, which explain results of the retrieval process, and *forward explanations*, which support in the modelling of similarity measures. *myCBR* has been developed as a *rapid prototyping* tool with a general purpose interface as well as a *similarity-based retrieval engine* for easy integration in other applications where the explanations can be further adapted to the application's requirements.

Key words: Case-Based Reasoning, explanation

1 Introduction

Ease-of-use as well as approachability of any software system is improved by increasing its understandability, which in turn can be supported by appropriate explanation capabilities [1, 2]. We follow Schank [3] in considering explanations the most common method used by humans to support understanding and decision making. In everyday human-human interactions explanations are an important vehicle to convey information in order to understand one another. Explanations enhance the knowledge of the communication partners in such a way that they accept certain statements. They understand more, allowing them to make informed decisions.

This communication-oriented view leads to the following explanation scenario with the three participants (Figure 1) *originator*, *user*, and *explainer*. The originator provides something to be explained, e.g., the solution to some problem, a technical device, a plan, etc. Here, the originator comprises the modelling tools and the retrieval engines of *myCBR*. The user is the addressee of the explanation, and the explainer provides the explanation. The explainer is interested in transferring the intention of the originator to the user as correctly as possible. The explainer chooses the kind of the explanation [4] and is responsible for the

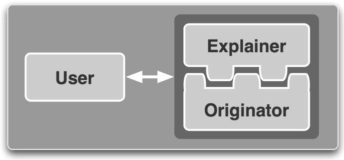

Fig. 1. Participants in explanation scenario [2]

computational aspects of the explanation process. The originator and explainer need to work together rather tightly to improve the communication with the user. The originator needs to provide the appropriate information in order to allow the explainer to construct appropriate explanations.

In order to make use of an information system the user needs at least a basic understanding of the application domain, i.e., the respective terms and concepts. But usually the user is not familiar with all of them. *Conceptual explanations* provide information about concepts of the application domain, linking unknown concepts to already known concepts.

In order to support the communication scenario described above, *myCBR* provides two general kinds of explanations: forward and backward explanations. *Forward explanations* explain indirectly, presenting different ways of optimising a given result and opening up possibilities for the exploratory use of a device or application. *Backward explanations* explain the results of a process and how they were generated.

The rest of the paper is structured as follows: After a brief introduction of *myCBR* in Section 2 the supported kinds of questions are presented in Section 3 followed by corresponding explanations (Section 4). We then describe the integration of the explanations into the user interface of Protégé in Section 5 and the accessible explanation data structures in Section 6. We close the paper with a summary and outlook.

2 The Open-Source Case-Based Reasoning Tool *myCBR*

myCBR[3] is an open-source plug-in for the open-source ontology editor Protégé[4]. Protégé [5] allows to define classes and attributes in an object-oriented way. Furthermore, it manages instances of these classes, which myCBR interprets as cases. So the handling of vocabulary and case base is already provided by Protégé. The myCBR plug-in provides several editors to define similarity measures for an ontology and a retrieval interface for testing.

The main motivation for the development of *myCBR* was the need for a compact and easy-to-use tool for building prototype CBR applications in teaching, research, and small industrial projects with minimal effort [6]. The tool needed

[3] `http://mycbr-project.net`
[4] `http://protege.stanford.edu/`

to be easily extendable to allow the experimental evaluation of new algorithms and recent research results. Many ideas for the implementation of *myCBR* came from *CBR-Works*[5] [7] which is not supported any more.

The current version of *myCBR* focuses on the similarity-based retrieval step of the CBR cycle [8]. A popular example of such retrieval-only systems are case-based product recommender systems [9]. While the first CBR systems were often based on simple distance metrics, today many CBR applications make use of highly sophisticated, knowledge-intensive similarity measures [10].

As the main goal of *myCBR* is to minimise the effort for building CBR applications that require knowledge-intensive similarity measures, *myCBR* 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, it includes tools for generating the case representation automatically from existing raw data.

myCBR provides retrieval mechanisms to find similar cases for a specified query. Both functionalities, modelling and retrieval, are available in separate tabs of the Protégé editor. Eventually, a myCBR model can be integrated into other applications, for which a standalone API is provided. In addition, since a CBR model is used as a background of an application we will later present a third tab, which is used to define conceptual explanations (Section 6.1).

From its conception, *myCBR* was designed with improved communication between the system and the user—knowledge engineer and end-user—in mind. The novice as well as the expert knowledge engineer is supported during the development phase of a *myCBR* project through intelligent support approaches and advanced GUI functionality. A dedicated explanation component provides modelling support information as well as explanations of retrieval results for quicker round trips of designing and testing.

3 Questions for *myCBR*

Explanations are in principle answers to questions. Hence, we formulated the requirements for the explanation support in terms of questions for which we developed explanation schemes (see Section 4). From the many questions, some of which are interesting at modelling time and others at retrieval time, we selected the most often asked questions, i.e., about concepts, about retrieval outcomes, and questions arising during modelling and maintenance.

3.1 Questions about Concepts

Questions about terms and concepts often arise for the end user when he or she is not familiar with the application domain. The user must be familiar with the

[5] CBR-Works was developed at the University of Kaiserslautern in co-operation with empolis GmbH, formerly tecinno GmbH.

terms and concepts used in the application in order to take advantage of it. The respective question is very simple:

a) What is meant by this concept?

Fortunately, this kind of questions can be also answered quite easily, even though additional knowledge is needed.

3.2 Questions about a Retrieval Outcome

A user may be surprised by a particular case's similarity value or she wants to assure herself of the system's quality. Both situations offer opportunities to increase trust into the system.

b) How did the system come to the similarity assessment of a particular case?
c) Which are the most similar aspects of a case? Which are the least?
d) Why are some demands of the query not met by the most similar cases?

The first two questions come to mind when a particular case is under examination. Asking question b) the user wants to retrace the procedure of similarity assessment. Especially when the outcome was surprising, the answer may deliver a justification or reveal a modelling error within the similarity measure. An answer to question c) explains in which way the case is similar to the query.

Another issue which rather concerns the whole ranking than a particular case is the absence of a good solution that meets the constraints in the query satisfyingly (addressed by question d). The user is unhappy with the top ranked cases of the retrieval result. If the estimated similarity of the case does not comply with the utility for the user, the similarity measure is insufficient. The source can then be found by asking questions of type b). If the similarity measure is correct though, a good case is missing in the case base, and the top cases are the best available.

3.3 Questions during Modelling and Maintenance

Forward explanations are intended to assist the knowledge engineer during modelling time, i.e., regarding the knowledge containers vocabulary and case base [11]. They concern the system's future behaviour. This situation here requires an analysis of the system as a whole in its current configuration.

e) Are some problem types underrepresented in the case base?
f) Is there an imbalance of cases in the case base?
g) Which parts of the similarity measure are of high or low relevance?
h) Which symbols are similar to a given symbol?

The first two questions address related issues. The case base may contain cases which appear to be very much the same while some special cases seem to be missing. Imagine a doctor who medicates patients having a cold a lot more often than patients having malaria. He does not need to remember every patient who had a cold. But probably it is useful to remember every patient who had malaria. On the other hand, regarding the used cars domain, it is probably not helpful, if the explainer states that the model lacks cars having a price lower than 1,000 Euro and being manufactured one year ago. In general, the attributes of a model are not statistically independent.

The knowledge engineer tries to approximate the utility of a case with the help of similarity measures. This can be a quite complex task, and getting lost in details can happen quickly. We want to guide the user to the relevant aspects of a similarity measure by answering question g).

The last question addresses the local similarity measure for attributes of symbolic type. The size of a symbol table grows quadratically with the number of allowed values. In this case, the task of keeping this table consistent during maintenance work becomes hard.

4 *myCBR*'s Explanations

Even though the principles of case-based reasoning are easy to understand sometimes a particular retrieval result is not. We distinguish between the explanations for knowledge engineers, which are supposed to assist in modelling and maintaining, and the explanations for the end user, who wants to understand the system's behaviour and the concepts involved.

If the knowledge engineer comes to the conclusion that he encountered a modelling error, several action alternatives are available. He or she can insert, delete or modify existing cases, modify similarity measures or even change the current vocabulary for which each task offers again a variety of alternatives in detail and needs a lot of attention [12]. The choice of action must be founded on the analysis of the knowledge base with respect to the purpose of the application. In such a situation, the explanatory capabilities of a CBR system are certainly limited, because it is not provided with a deeper understanding of the application domain. However, it can give hints, illustrate internal dependencies and highlight certain parts of the model in such a way that the knowledge engineer is enabled to make his decision faster and more confidently.

In the context of case-based reasoning, conceptual explanations are used to explain the vocabulary knowledge container. Backward explanations explain the outcome of a particular retrieval result and provide means for understanding the results of a similarity calculation [13]. Forward explanations are intended for modelling and maintenance assistance, by providing information about the status of the model.

An explanation scheme, here, describes a special procedure to answer a particular kind of question. During the design of an explanation scheme we kept in mind that for an explanation to count as good it needs to be *relevant*, *innovative*, *compact*, *correct*, and *convincing* [14]. We do not consider innovativeness here, because it requires some kind of user model and some facility that records when an explanation has been provided. We intend to use the justification-based explanation support server Redux^{exp} in the future [15].

4.1 Conceptual Explanations

A conceptual explanation is a comprehensive description of a concept. It consists of a definition, some examples and references to further characterisations, for which any kind of medium can be used (e.g., text, images, audio, video). Conceptual explanations are inherently static, because concepts usually do not change. But there are good reasons to consider the context in which the concept is used and the user's personal level of knowledge. However, in this work we regard a conceptual explanation as static regarding the given ontology.

We want to support the end user and the knowledge engineer. For the end user knowledge about the concepts is the most important conceptual knowledge. This kind of knowledge is domain dependent and not yet part of a *myCBR* model. It must be additionally supplied by the knowledge engineer.

An important source for conceptual explanations for the end user is the vocabulary knowledge container of a CBR system [16]. CBR systems do not provide sufficient knowledge to provide satisfying conceptual explanations per se. This also applies for myCBR in its role as the originator in the explanation scenario (see Figure 1). Thus, the vocabulary knowledge container, i.e., the ontology, needs an extension.

Conceptual explanations for the knowledge engineers are about concepts and functionalities of *myCBR* itself, for which help buttons and other documentation are already provided.

4.2 Backward Explanations

Question b) will be answered with the help of a detailed recording of the retrieval process. All local and global similarity measures involved write comments and provisional similarity values to a certain protocol. So, every step of similarity calculation can be traced.

In order to answer Questions c) and d) the notion of aspect must be clarified. Here, an aspect of a query is one single attribute. The similarity of an aspect is then a local similarity value and can be found in the mentioned protocol. Thus, sorting the attributes of a case with respect to their (local) similarity values gives the answer to question c).

4.3 Forward Explanations

Both questions e) and f) ask for some kind of distribution within the case base. For each attribute the value distribution within the case base is used to show the user which attribute values have many representations and which have only a few. In a similar way, a distribution of class instances is set up. Even if the value distribution was uniform for every class and attribute of the vocabulary, the case distribution may not be uniform at all due to interdependencies between attributes. On the other hand, if the value distribution is irregular for some attributes or classes, the case distribution cannot be uniform at the same time. Hence, the questions are only partly answered.

Similarity measures provide a means to compare two objects of a certain domain with each other. Since some comparisons are more frequent than others, the corresponding part of the similarity measure is also of higher relevance than other parts. To answer Question g), the task is to find out the comparison frequency for two arbitrary objects for each attribute domain. This will be referred to as a *relevance distribution* in the following. On the one hand, this depends on the value distribution within the case base. On the other hand, it depends on the value distribution within the submitted queries by the user. This piece of information is not given, because it involves the continuous analysis of user queries[6]. For now we provide a preliminary solution by assuming the value distribution within the case base to be similar to the one within the user queries.

The average number of comparisons between a query value q and a case value c is given by the following formula:

$$freq_{relevance}(q,c) = freq_{casebase}(q) * freq_{casebase}(c)$$

where $freq_{casebase}(x)$ is the number of cases having the attribute value x divided by the size of the case base.

A look at the local similarity measure gives the answer to question h). But keeping track of the contents of a big table becomes more difficult as the number of allowed symbols increases. Trying to understand the origin of this problem, one realises that it cannot only follow from its quadratic size. Another issue is the chaotic arrangement of the symbols regarding the users varying exploration interest. Because the row and column header are in a fixed symbol order, the comparison of some relevant similarities demands high concentration and a good eye. To solve this problem we introduced the concept of dynamic symbol orders where symbols are ordered by ascending or descending similarity in a selected row or column.

5 User Interface Integration

As ease-of-use is of high priority for the development of *myCBR* we aimed at seamlessly integrating explanations into the user interface.

Conceptual and backward Explanations In order to increase transparency and trust in the retrieval process [16], *myCBR* creates an explanation object for each case during similarity calculation.

Figure 2 gives a schematic overview of the Retrieval GUI. The area is divided into several columns. The leftmost column is used for query specification. The others are used to show the retrieval results. The rightmost column lists all cases of the case base ordered by their similarity to the query. The number of columns displayed in between can be configured. The rows of the table are labelled with the names of the classes' attributes.

[6] Collecting and analysing user queries is an important maintenance and quality improvement task (cf. [12])

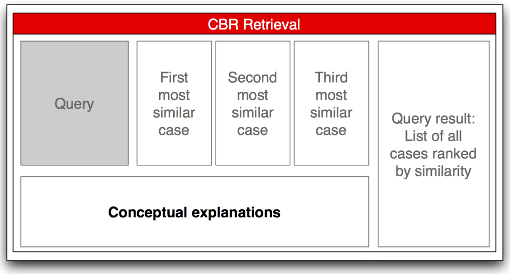

Fig. 2. Schematic overview of retrieval GUI

Conceptual explanations are addressing the end users of the system. They are interested in retrieving the most similar cases to their queries. At the bottom of the retrieval GUI conceptual explanations are shown when the user hovers the mouse over a table cell. Figure 3 shows a snapshot. Retrieval details are

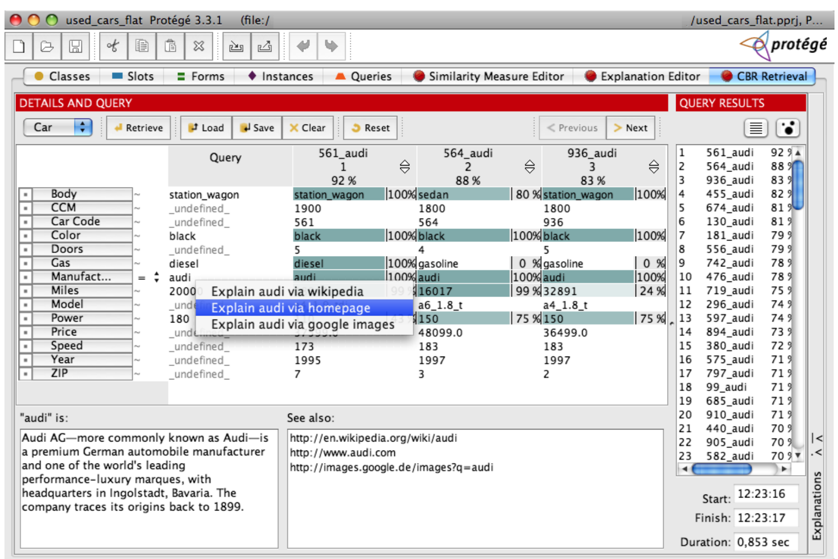

Fig. 3. Retrieval interface with conceptual and backward explanations

presented to the user either as tool tips or in abbreviated form along with the case's attribute value, e.g., the mileage of car offer 561_audi (18,940) is 100% similar to the requested mileage (20,000). Another valuable feature is the option to find the most similar cases with respect to a single attribute by simply clicking on the attribute name (row head). In attribute rich cases one might also want to sort the local similarity values of one case. For this one clicks on the respective case name (column head).

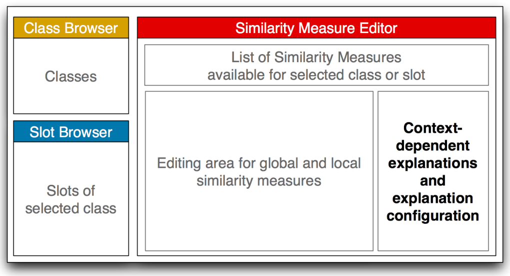

Fig. 4. Schematic overview of similarity measure editing with explanation panel

Forward Explanations While developing a CBR system an important question is whether a similarity measure leads to the appropriate cases for a given query. We distinguish between local and global similarity measures. This is reflected in the GUI (Figure 4). Depending on whether a class or slot is selected, a global or local similarity measure editor is displayed on the right side of the window. A corresponding explanation panel is superimposed on demand.

6 Implementation Issues

The support of explanations requires new components and changes of the current myCBR system. As we mentioned earlier, the explainer must be strongly interlaced with the originator due to reliability and authenticity issues (Figure 1). The explainer needs an insight into the originator's knowledge, and the originator must actively deliver information about its behaviour to the explainer.

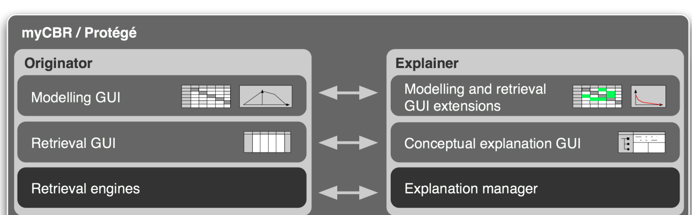

Fig. 5. Detailed components of originator and explainer

Considering the different kinds of explanation, one notices that a central and easily accessible explanation component, which we named *explanation manager*, is needed to generate conceptual and forward explanations. The explanation manager also provides backward explanations using logged similarity calculations. Figure 5 gives an overview of the GUI extensions of the similarity measure editors and the retrieval widget as well as the additional editor adding conceptual knowledge to a *myCBR* model.

6.1 Implementation of Conceptual Explanations

Conceptual explanations do not involve complicated algorithms. The required functionality at its core is a static mapping from concepts to explanations. Figure 6 shows the conceptual explanations editor. On the left hand side a tree

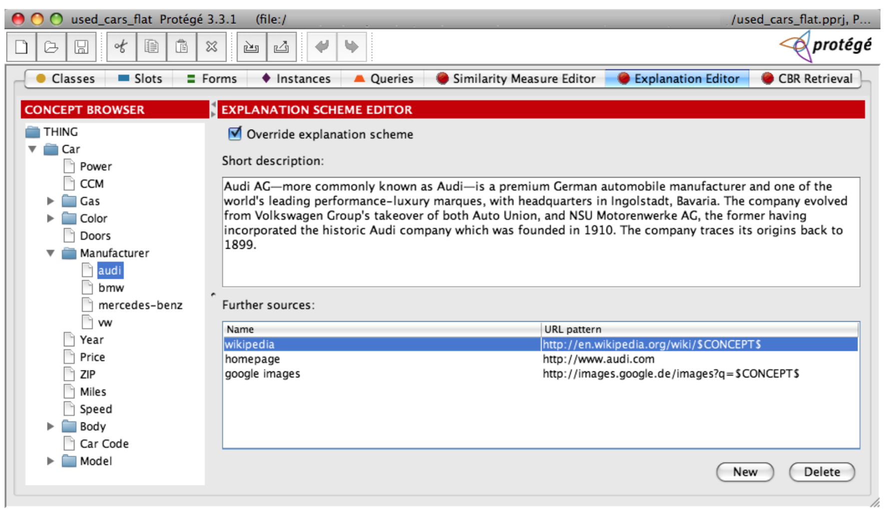

Fig. 6. Editor for conceptual explanations

displays all concepts of the ontology. On the right hand side one can edit its explanation, i.e., a short description and a list of URLs of further documents. For example, manufacturers in the used cars application can be explained via *wikipedia*[7] or *google define*[8].

The conceptual explanations are stored as part of the *myCBR* model, but in a separate file in order to separate it from the reasoning knowledge. When a *myCBR* model is loaded, the explanation manager looks for this file.

[7] `http://wikipedia.org`
[8] `http://www.google.com/help/features.html#definition`

6.2 Implementation of Backward Explanations

Backward explanations are given with the help of a certain kind of retrieval recording. Since we want to log the calculation steps of each global and local similarity measure, the interface $float \leftarrow SIM(query, case)$ which is implemented by all similarity measures is extended to $float \leftarrow SIM(query, case, explanation)$.

Every similarity measure was modified in order to write the resulting similarity value and further comments such as intermediate results to the *explanation* object while the similarity between *query* and *case* is determined. A tree-like structure, mirroring the local-global structure of similarity measures, is used to build the overall backward explanation for a case. The query and case values as well as the responsible similarity measure are tracked also. The retrieval engine is responsible for connecting them appropriately.

6.3 Implementation of Forward Explanations

Statistical information about the value distribution within the case base is required to give forward explanations. Furthermore, this information cannot be gained in tow of another procedure as it was the case for the backward explanations. The case base must be examined in an active way. Certainly, it is not advisable to let each explanatory component calculate its required statistics on its own as soon as an explanation is demanded, because the computation time for each given explanation would be annoyingly high. We decided to use the explanation manager as an intermediate component that holds all statistical information about the case base.

7 Summary

Three different kinds of explanation are delivered by *myCBR*: *Conceptual explanations* describe items of the vocabulary knowledge container to the end user via textual descriptions or via links to additional information, e.g., documents. *Backward explanations* explain the retrieval outcome in relation to a particular query to the end user. And, *forward explanations* assist the knowledge engineer during modelling and maintenance. We formulated questions and corresponding explanation schemes and extended the user interfaces of *myCBR* accordingly to support the explanation presentation.

myCBR is still an ongoing project. Several extensions of the system are planned or are already under development. We encourage other researchers to try out *myCBR* in their own research and teaching projects and to contribute to the further development by implementing their own extensions and experimental modules.

References

1. Roth-Berghofer, T.R.: Explanations and Case-Based Reasoning: Foundational issues. In Funk, P., González-Calero, P.A., eds.: Advances in Case-Based Reasoning, Springer-Verlag (2004) 389–403

2. Roth-Berghofer, T.R., Richter, M.M.: On explanation. Künstliche Intelligenz **22**(2) (2008) 5–7
3. Schank, R.C.: Explanation Patterns: Understanding Mechanically and Creatively. Lawrence Erlbaum Associates, Hillsdale, NJ (1986)
4. Roth-Berghofer, T., Cassens, J., Sørmo, F.: Goals and kinds of explanations in case-based reasoning. In Althoff, K.D., Dengel, A., Bergmann, R., Nick, M., Roth-Berghofer, T., eds.: WM 2005: Professional Knowledge Management, Kaiserslautern, Germany, DFKI GmbH (2005) 264–268
5. Gennari, J.H., Musen, M.A., Fergerson, R.W., Grosso, W.E., Crubézy, M., Eriksson, H., Noy, N.F., Tu, S.W.: The evolution of Protégé an environment for knowledge-based systems development. Int. J. Hum.-Comput. Stud. **58**(1) (2003) 89–123
6. Stahl, A., Roth-Berghofer, T.R.: Rapid prototyping of CBR applications with the open source tool myCBR. In Bergmann, R., Althoff, K.D., eds.: Advances in Case-Based Reasoning, Springer Verlag (2008)
7. Schulz, S.: CBR-Works: A state-of-the-art shell for case-based application building. In Melis, E., ed.: Proceedings of the 7th German Workshop on Case-Based Reasoning, GWCBR'99, Würzburg, Germany, University of Würzburg (1999) 166–175
8. Aamodt, A.: Explanation-driven case-based reasoning. In Stefan Wess, K.D.A., Richter, M., eds.: Topics in Case-Based Reasoning, Berlin, Springer-Verlag (1994)
9. Bridge, D., Göker, M.H., McGinty, L., Smyth, B.: Case-based recommender systems. Knowledge Engineering Review **20**(3) (2006)
10. Stahl, A.: Learning of Knowledge-Intensive Similarity Measures in Case-Based Reasoning. PhD thesis, University of Kaiserslautern (2003)
11. Richter, M.M.: The knowledge contained in similarity measures. Invited Talk at the First International Conference on Case-Based Reasoning, ICCBR'95, Sesimbra, Portugal (1995)
12. Roth-Berghofer, T.R.: Knowledge Maintenance of Case-Based Reasoning Systems – The SIAM Methodology. Volume 262 of Dissertationen zur Künstlichen Intelligenz. Akademische Verlagsgesellschaft Aka GmbH / IOS Press, Berlin, Germany (2003)
13. Bahls, D., Roth-Berghofer, T.: Explanation support for the case-based reasoning tool myCBR. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence. July 22–26, 2007, Vancouver, British Columbia, Canada., The AAAI Press, Menlo Park, California (2007) 1844–1845
14. Swartout, W.R., Moore, J.D.: Explanation in second generation expert systems. In David, J., Krivine, J., Simmons, R., eds.: Second Generation Expert Systems, Berlin, Springer Verlag (1993) 543–585
15. Roth-Berghofer, T.R., Mittag, F.: ReduxExp: A justification-based explanation-support server. Proceedings of AI-2008. The twenty-eighth SGAI international conference on artificial intelligence. In Petridis, M., Coenen, F., Bramer, M., eds.: Research and Development in Intelligent Systems XXV, London, UK, Springer Verlag (2008)
16. Roth-Berghofer, T.R., Cassens, J.: Mapping goals and kinds of explanations to the knowledge containers of case-based reasoning systems. In Muñoz-Avila, H., Ricci, F., eds.: Case-Based Reasoning Research and Development, 6th International Conference on Case-Based Reasoning, ICCBR 2005, Chicago, IL, USA, August 2005, Proceedings. Number 3620 in Lecture Notes in Artificial Intelligence LNAI, Heidelberg, Springer Verlag (2005) 451–464