Visualisation of Case-Base Reasoning for Explanation

Stewart Massie, Susan Craw, and Nirmalie Wiratunga

School of Computing,
The Robert Gordon University,
Aberdeen AB25 1HG, Scotland, UK
{sm|smc|nw}@comp.rgu.ac.uk

Abstract. It is not sufficient for Case Based Reasoning systems to merely provide competent solutions. In complex tasks, such as configuration and design, the user requires an explanation of the solution in order to judge its validity and identify any deficiencies. Providing this explanation is not a straightforward task, particularly in systems using k-nearest neighbour retrieval, because much of the knowledge used to design the system is hidden from the user. This paper presents an approach to explaining the solution reached by a CBR system as well as highlighting differences between the target problem and most similar cases that may help to inform the adaptation process. This is achieved by presenting the best matching cases along with the system solution through a visualisation. The approach is demonstrated on a pharmaceutical tablet formulation problem with a tool called FormuCaseViz. An expert evaluation provides evidence of the potential benefits of our approach.

1 Introduction

Case Based Reasoning (CBR) is experience based problem-solving that mimics the approach often used by humans. One of the many advantages often associated with CBR is its understandability as it can present previous cases to support or explain its conclusions [8]. Recent research provides evidence to support this view that an explanation based on previous experience is more convincing than one based on rules [4].

In contrast to the idea that CBR techniques are understandable, King et al. [7] grade classification algorithms based on the *comprehensibility of the results*. The k-nearest neighbour (k-NN) algorithm, often used in CBR systems, was graded at only 2 out of 5 by users, with only neural network algorithms being graded lower. One reason is that the similarity measure, usually compacted into a single value, hides the knowledge gained during system development and encoded into the design [2]. Making this knowledge more accessible to users is an important challenge; we believe the potential benefits include simplifying the interpretation of results, exposing deficiencies in the reasoning process, and increasing user confidence in the system.

Explanation of CBR solutions is typically based on the single most *similar* case to the new problem, and possibly a similarity value. While this level of explanation might suffice in relatively simple, easily understood domains, it is not sufficient for tasks that are knowledge intensive. Individually the nearest case can provide an explanation. However, as with CBR solutions that can be improved by using several cases to provide

a combined solution, likewise there may be added value in providing an explanation based on several *similar* cases. We believe this is particularly true if the similarities and differences within these cases can be made explicit with the aid of visualisation.

In this paper we attempt to address the apparent contradiction that the CBR paradigm is transparent and understandable, yet the results of k-NN retrieval are not easy to comprehend. The user is expected to accept that the case-base contains representative problems and that the similarity measure used is appropriate to his problem. By hiding its similarity knowledge the system is not providing a satisfactory explanation of the solution and it is difficult for a user to have confidence in the system. We present an approach that makes this underlying knowledge and an explanation of the solution available. We demonstrate our approach on a tablet formulation problem domain. The usefulness of this approach is assessed in an expert user evaluation.

In Section 2 we review recent research on explanation in CBR. Section 3 discusses the information that different users need to explain a solution and to increase their acceptance of CBR systems. The problem domain on which we test our approach is discussed in Section 4. A knowledge-light approach to providing this information for a tablet formulation application is presented in Section 5. In Section 6 the design of the user evaluation is described along with the results obtained. Finally we provide conclusions and recommendation for future work in Section 7.

2 Related Work on Explanation in CBR

CBR systems using decision tree guided retrieval typically provide explanations by highlighting feature values of decision nodes traversed in order to reach the leaf node [4]. This is similar to the methods adopted in rule-based expert systems which often show rule activations [11]. Such rule-based explanation is not possible in systems using only k-NN retrieval because a set of discriminatory features is not identified as part of the algorithm. In these systems a typical approach is to present an explanation in terms of feature value differences between the query and the retrieved case. Cunningham et al. [4] suggest that explanations, expressed in terms of similarity only, can be useful in some domains (e.g. medical decision support) but is inadequate in others. McSherry's [10] approach to explaining solutions is based on identifying features in the target problem that support and oppose the predicted outcome. Discovery of the supporters and opposers of a predicted outcome is based on the conditional probabilities, computed from the cases available at run time, of the observed features in each outcome class.

Hotho et al. [6] provide explanations that are not solely similarity based. In their approach text documents are formed into clusters using a similarity metric and k-means clustering. The importance of the features or words in each cluster are ranked and the most *important* are used to represent the cluster. The relationship between clusters can then be identified using WordNet concept hierarchies. An explanation can now be given which is based not only by the similarity of a document to other members in its cluster, but also on the relationship to other clusters.

Another approach to providing an explanation is through visualisation. McArdle & Wilson [9] present a dynamic visualisation of case-base usage by using a spring based algorithm. The algorithm uses the attraction and repulsion of the *springs* to spread

the cases around a two dimensional graph in an attempt to preserve the n-dimensional distances between cases. This provides more insight into the similarity assessment than the usual single dimensional value. However, the knowledge held within the similarity metric is still hidden. Although this approach is used for supporting the maintenance of large case-bases it could also be adopted to visualise retrieved cases. An alternative approach is the parallel coordinate plot. Falkman [5] uses this approach to develop an information visualisation tool, The Cube, which displays a case-base using three dimensional parallel co-ordinate plots. This approach allows the underlying data to be visualised as well as the similarity metric. We exploit this approach.

3 What Needs to be Explained

Knowledge intensive tasks require a better explanation than simply a proposed solution and a set of retrieved cases. This is particularly true of design problems where the case-base does not contain all possible designs, and the proposed solution is only an initial draft, which may need to be adapted. The domain expert requires additional information and explanations to make the decision making process more transparent and to allow him to judge the validity of the solution. Further information is needed to explain both the CBR process and the proposed solution.

3.1 The CBR Process

Knowledge embedded in the CBR system in the form of stored cases and the similarity measure on which retrieval is based should be visible to the user:

- the case-base is the main knowledge source of a CBR system and usually determines its competence. The user must be able to judge its quality and coverage in order to decide if it is suitable to address current problems. This will allow gaps in the case-base knowledge to be addressed and rectified.
- the retrieval process usually involves a similarity function that compares the cases held in the case-base with the new query. This can be a Euclidean distance function or some domain specific function. The importance of individual features is often identified by feature weighting. The user needs to be able to decide if the similarity function is appropriate and if the importance of features is correctly represented.

This knowledge is often hidden from the user and can result in two effects: the user may accept the hidden knowledge as fact and not question it, or alternatively, confidence in the system may be reduced due to a lack of understanding of the hidden process. Either of these effects may have a negative impact on the acceptability of a CBR system.

3.2 The Solution Itself

In addition to general information about the underlying CBR model being used, local information specific to the current query must be visible to the user. This will allow a judgement to be made on the quality of the proposed solution and provide the relevant information to make manual adaptations. Visible, local information helps identify

deficiencies in this particular problem solving experience (e.g. quality of case-base, similarity function). It can be provided by comparing the new query with either the case-base as a whole or with the most similar cases identified by the similarity function (its nearest neighbours). Local information is required in the following areas:

- **Coverage in the Neighbourhood of the target Problem.** This allows the user to identify whether the case-base coverage is sufficient in the local region for this particular query and allows an area of the problem space to be highlighted. Any deficiencies in coverage can be addressed by adding new relevant cases to the case-base.
- **Similarities & Differences within Best Matching Cases and the Query.** Easily interpretable information is required that allows the user to identify the attribute values that are common to both the query and the best matching cases. More importantly it allows specific attribute value differences to be identified. This is the information needed for adaptation of the proposed solution. The overall similarity scores on which retrieval is based are inadequate for this purpose.

This additional information should be presented in an easily interpretable format that does not swamp the user with detail. We have employed a visualisation approach.

4 Problem Domain and FORMUCASE

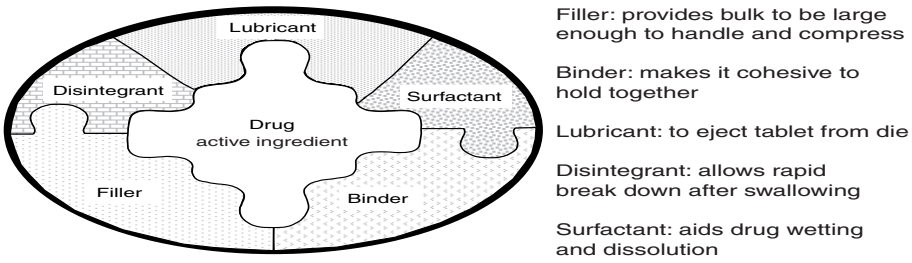

Fig. 1. Tablet formulation problem

FORMUCASE is a CBR system that formulates a tablet for a given dose of a new drug. This involves choosing inert excipients (e.g. Lactose, Maize Starch, etc.) to mix with the new drug so that the tablet can be manufactured in a robust form. In addition to the drug, a tablet consists of five components each with a distinct role; i.e. Filler, Disintegrant, Lubricant, Surfactant, and Binder (see Fig. 1). The formulation task entails identifying a suitable excipient and amount for each chosen component. Each chosen excipient must be suitable for its desired role and be compatible with each other and the drug. A more detailed description of the problem domain is available in [3].

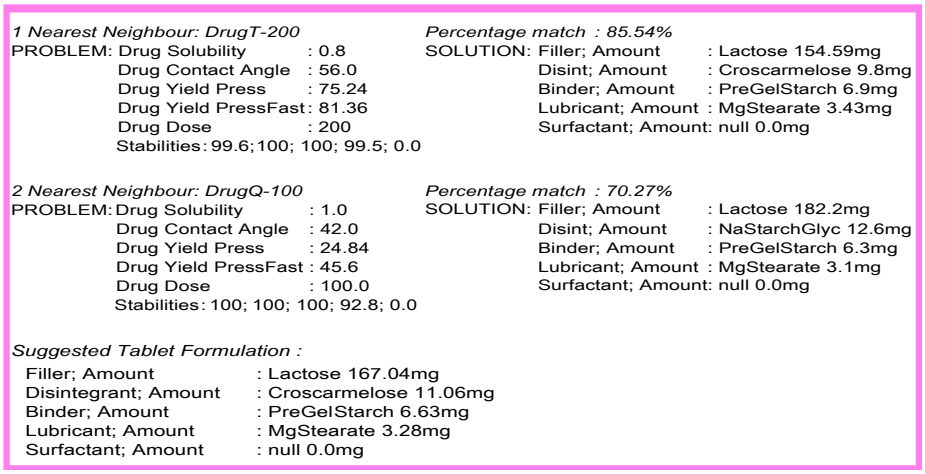

Fig. 2. FormuCase output

Each case has a problem and solution represented by a list of attribute values. The problem attributes consist of five physical properties describing the drug itself and twenty chemical properties which describe how the drug reacts with possible excipients. All these attributes have numerical values. The solution has ten attributes; five with nominal values identifying the excipients used and five numeric values identifying the quantity of each excipient. When formulating a tablet for a new drug the attribute values representing the drug are entered and its nearest neighbours identified using the k-NN algorithm. The multi-component proposed solution to the query is a weighted majority vote of its *k* nearest neighbours to determine excipients and a weighted average for excipient quantities.

The output from FORMUCASE (see Fig. 2) is presented in report format displaying the nearest neighbours, their problem and solution attribute values and their similarity to the new query. The feature values of the proposed solution are then displayed. This retrieve-only system forms the first step in a tablet formulation. Differences between the new test problem and the retrieved cases may indicate the need to refine the predicted solution by manual adaptation.

5 FORMUCASEVIZ

We demonstrate our approach to explanation using visualisation with this tablet formulation problem. Our hypothesis is that the visual version (FORMUCASEVIZ) will help explain the CBR process and increase user confidence in the solution. The problem and solution are displayed in parallel coordinate plots in order to address the issues discussed in Section 3.

A parallel co-ordinate graph's primary advantage over other types of statistical graphs is its ability to display a multi-dimensional vector or case in two dimensions.

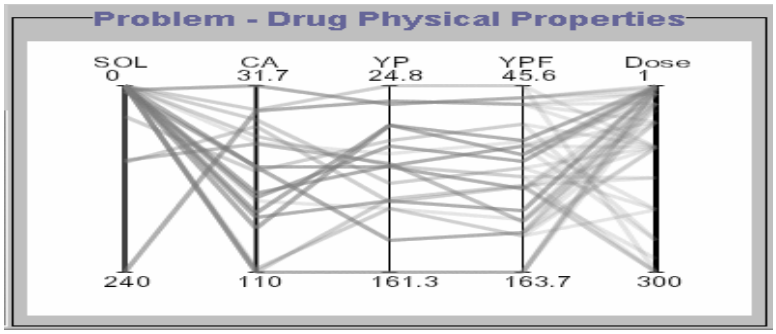

Fig. 3. Parallel co-ordinate plot showing the drug physical properties of a case-base

Fig. 3 shows a plot with five dimensions. Each attribute is represented by a labelled vertical axis. The value of the attribute for each case is plotted along each axis. The points are then connected using horizontal line segments such that each case is represented as an unbroken series of line segments which intersect the vertical axes. Each axis is scaled to a different attribute. The result is a *signature* across n dimensions for each case. Cases with similar data values across all features will share similar signatures. Clusters of like cases can thus be discerned, and associations among features can also be visualised.

The basic layout of the graphical display for the tablet formulation task takes the form of three panels each containing a parallel coordinate graph (see Fig. 4). The top graph contains twenty axes and provides attribute value information for the chemical stabilities of each drug with respect to the excipients commonly used in drug formulation. The lower left graph contains five axes with the drugs physical properties and the lower right graph displays the solution attribute values. Thus the top and lower left panel contain attributes from the problem domain and the lower right graph contains attributes from the solution space.

Loading a case-base results in the vertical axes being drawn and labelled with each attribute's name and minimum and maximum value. The case lines, intersecting the axes, are also shown (see Fig. 3). A visual picture of case-base coverage can now be seen with darker regions representing well covered areas of the problem space and gaps being visible as portions of the axis without case lines. The encoded retrieval knowledge, in the form of feature weights, is represented by the width of each axis. Fig. 3 shows a case-base displayed on the drug physical properties graph. It can be seen that the attributes *SOL* and *Dose* have the highest weights.

We see in Fig. 4 that on entering a new query a black line representing it is drawn on the two problem domain graphs. This provides information on the local coverage provided by the case-base in relation to this particular query. As no solution is yet available there is no black line representing the query in the solution panel.

Fig. 5 shows a solution to a query. The nearest neighbours are identified in the case-base and displayed as coloured dashed lines. The nearest neighbour solutions are also displayed in the solution panel along with the proposed solution for the new query. A

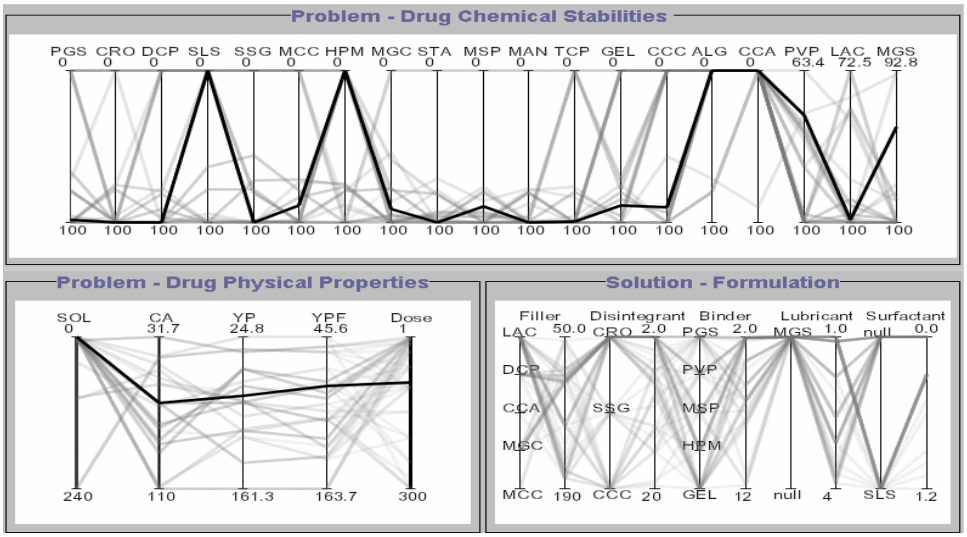

Fig. 4. Output screen of FORMUCASEVIZ with an unsolved problem entered

new axis is added to the drug physical properties problem panel showing the similarity of the query to each of its NN along with labels for each case. This visualisation allows the similarities and differences to be viewed in terms of the real data aiding interpretation of the proposed solution and making the adaptation stage easier. For example, in Fig. 5, it can be seen that the best matching cases disagree on which filler to use. *LAC* is the proposed solution but reference to the chemical stabilities show *DCP* would be a better choice for the new drug as it has a higher chemical stability.

5.1 Ordering the Attributes

The order or arrangement of the attributes is important when using parallel co-ordinate graphs. The arrangement can improve the visualisation by helping to identify trends or correlations within the case-base. Many approaches to multi-dimensional data visualisation arrange the attributes arbitrarily, possibly in the order that they appear in the case representation. We have taken the approach of arranging the attribute axes based on their similarity to each other in order to reduce line crossing on the graph. To achieve this axis arrangement we first use an axes similarity function to identify the pairwise similarities between the axes and then determine an arrangement so that similar axis are placed adjacent to each other.

An obvious way to measure axis similarity is to compare values across the cases. The similarity between axes A_i and A_j is measured using the attribute value similarity across the *cases*, rather than across the attributes as for case similarity. Thus, when case c_k is described by the n-tuple of attribute values $(a_{1k}, \ldots a_{nk})$, the axis similarity from cases $c_1 \ldots c_m$ is defined below where similarity is the inverse Euclidean distance defined for individual (normalised) attribute values.

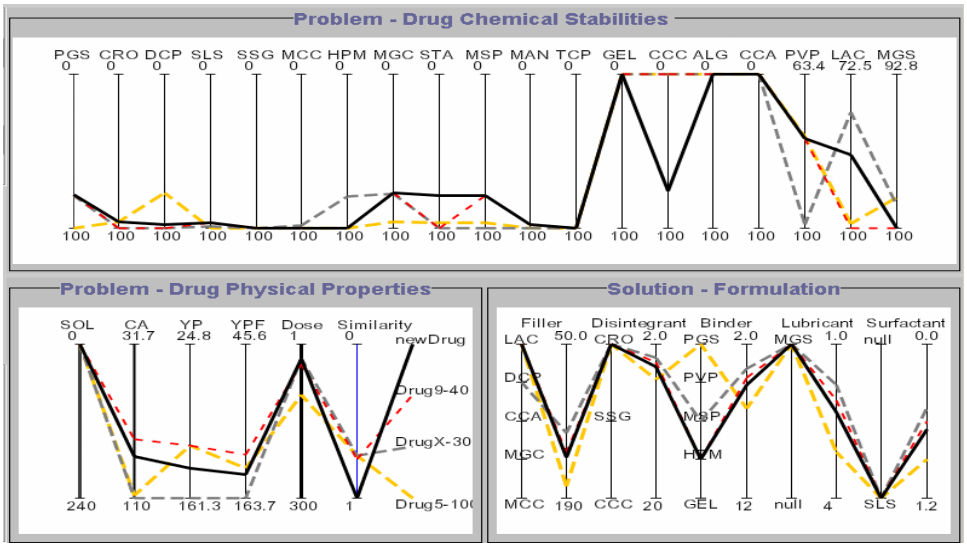

Fig. 5. Output screen of FORMUCASEVIZ with a problem and proposed solution

$$Similarity(A_i, A_j) = \sum_{k=1}^{m} similarity(a_{ik}, a_{jk})$$

Determining a linear arrangement for the axes such that similar axes are placed close to each other is still not straightforward. We adopt the approach of first looking at the pairwise similarity values between the axes and picking the most similar pair. These are placed to the left of the graph. The most similar unallocated axis is placed next to it. This process continues until all the axes have been allocated a position in the graph.

An alternative approach, which may give an optimal arrangement, is to find the order with the minimum total similarity when adjacent axis similarities are summed. However Ankerst et al [1] show that this problem is NP-complete. The use of a genetic algorithm or optimisation approach may be appropriate.

The arrangement of the axes can be carried out from a global or local context. The global arrangement looks at the whole case-base and takes no account of the current query. This approach is best for looking at case-base coverage or when trying to identify trends within the case-base. It also has the advantage that it can be used prior to a query being entered and is more stable as it remains unchanged as each new query is entered. The local arrangement only looks at a portion of the case-base, typically around the new query by only using its nearest neighbours in the calculation of the axis similarities.

FORMUCASEVIZ was implemented with the global arrangement on the two problem domain panels as it was found that the continual rearranging of axes gave problems in interpreting the results. However in other domains the advantages of a local approach may outweigh this disadvantage. No ordering of axes was applied to the solution panel as a fixed order was found to be more easily understood.

6 Evaluating the Explanation

The purpose of the domain expert evaluation was to investigate how well FORMUCASE and FORMUCASEVIZ explain their solution and the process undertaken to arrive at the solution. This was done by looking at how easily the solutions could be interpreted and the confidence the domain expert has in the system's solution.

Two domain experts were given both versions of FORMUCASE, a case-base and three sample problems to solve. The evaluation required the expert to solve three different test problems on the same case-base. The evaluation is carried out first with FORMUCASE and then FORMUCASEVIZ. The experts were asked to fill out a questionnaire, containing thirty questions, that was designed to ascertain their confidence in the system given the tool's ability to explain its reasoning.

While the results of the evaluation cannot be presented in detail here, we summarise our findings by highlighting some of the interesting observations.

- The experts agreed that FORMUCASEVIZ explains the CBR process of generating a solution better than the textual output version.
- There was a reluctance to accept a similarity value alone as a measure of the case-base's competence to answer a specific query. In answer to the question *does the case-base contain similar cases to the query?* with FORMUCASE an *unsure* answer was usually given. In contrast, when presented with the same query on FORMUCASEVIZ a definite and expected answer was always given.
- There was generally more confidence in the solutions provide by FORMUCASEVIZ and it was possible for alternative solutions to be suggested by the expert.
- The evaluators were better able to answer questions requiring them to identify differences within the nearest neighbours and between the query and the neighbours. One evaluator commented *The graphical display is excellent and shows up similarities and differences in a very clear way*.
- Exact numerical values cannot be read from FORMUCASEVIZ as the values have to be interpolated from the axes. This is not ideal with one expert commenting *the absence of easily readable numerical data is a big problem*. This deficiency needs to be addressed.

The positive results from our evaluation suggest that FORMUCASEVIZ provides a useful and more informative explanation of the proposed solution than FORMUCASE.

7 Conclusions and Future Work

A user gains confidence in a system that provides correct results. However confidence is also improved in systems where the decision making process is understood and deficiencies can be identified and resolved. The explanation of results should be a key design criterion in CBR systems.

In this paper we have identified some of the reasons why CBR systems, particularly those using k-NN retrieval, are not as successful as they might be. We have presented an

approach that can address some of these problems using a parallel co-ordinate visualisation of the problem and solution. This approach has been demonstrated on FORMUCASEVIZ in a tablet formulation problem domain in which thirty-five dimensional data is viewed in a single representation. A user evaluation confirmed that this explanation based approach made interpretation of the results easier than the textual version, and better explained the CBR process. The need for exact numerical values to be available on the visualisation was also identified. While we have used tablet formulation in this paper our approach would be applicable across a wide range of CBR problem domains.

Future work will look at providing more local information related directly to the query rather than to the case-base as a whole either by re-ordering all the axes or highlighting specific axis where correlations can be identified. In addition we will look at providing a more dynamic visualisation that allows the user to interact directly with the data, for example to change the query or highlight certain areas of the case-base.

Acknowledgments

We acknowledge the assistance of PROFITS, Bradford University for funding the FORMUCASE demonstrator, providing the tablet formulation data and supplying willing domain experts for user evaluations.

References

1. Ankerst, M., Berchtold, S., Keim, D.A.: Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In Proceedings of IEEE Symposium on Information Visualization, IEEE Computer Society Press (1998) 52–60
2. Bergmann, R.: Experience Management: Foundations, Development Methodology, and Internet-Based Applications. Springer (2002)
3. Craw, S. M., Wiratunga, N., Rowe, R.: Case-based design for tablet formulation. In Proceedings of the 4th European Workshop on Case-Based Reasoning, Springer (1998) 358–369
4. Cunningham, P., Doyle, D., Loughrey, J.: An evaluation of the usefulness of case-based explanation. In Proceedings of the 5th International Conference on Case-Based Reasoning, Springer (2003) 122–130
5. Falkman. G.: The use of a uniform declarative model in 3D visualisation for case-based reasoning. In Proceedings of the 6th European Conference on Case-Based Reasoning, Springer (2002) 103–117
6. Hotho, A., Staab, S., Stumme, G.: Explaining text clustering results using semantic structures. In Principles of Data Mining and Knowledge Discovery 7th European Conference, Springer (2003) 217–228
7. King, R., Feng, C., Sutherland, A.: Statlog: comparison of classification algorithms on large real-world problems. Applied Artificial Intelligence **9-3** (1995) 259–287
8. Leake, D. B.: CBR in context: The present and future. In Leake, D. B. (ed.): Case-Based Reasoning: Experiences, Lessons and Future Directions. MIT Press (1996) 3–30
9. McArdle, G. P., Wilson, D. C.: Visualising case-base usage. In Workshop Proceedings of the 5th International Conference on Case-Based Reasoning, Springer (2003) 105–124
10. McSherry, D.: Explanation in case-based reasoning: an evidential approach. In Proceedings of the 8th UK Workshop on Case-Based Reasoning, (2003) 47–55
11. Southwick, R.: Explaining reasoning: an overview of explanation in knowledge-based systems. Knowledge Engineering Review **6** (1991) 1–19