Picture Perfect - Visualisation Techniques for Case-Based Reasoning

Barry Smyth, Mark Mullins and **Elizabeth McKenna** [1]

Abstract. Case-based reasoning systems solve new problems by retrieving and adapting the solutions to similar previously solved problems. The success and performance of any case-based reasoning system depends critically on its repository of prior problem solving experiences, the cases in its case-base. It is perhaps surprising then that the case-based reasoning community has only recently begun to investigate new ways of intelligently supporting the authoring (and on-going maintenance) of case-bases. In this paper we describe and evaluate a technique for visualising the cases in a case-base. We argue that such techniques have a vital role to play in helping authors to understand the structure of an evolving case-base and so improve the efficiency of the authoring process and the quality of the resulting case-bases.

1 INTRODUCTION

Case-based reasoning (CBR) systems solve new problems by retrieving and adapting the solutions of similar problems stored as cases in a case-base [9, 15]. The CBR technique has been successfully adapted for a wide variety of tasks (including classification, diagnosis, prediction, planning, and design) across a broad range of domains (for example, fraud detection, property valuation, route planning, and software design).

The success of a case-based reasoner will depend critically on the cases in its case-base, specifically, on the quality of individual cases and the problem solving coverage that they provide. Obviously if vital cases are missing from a case-base then problem solving coverage will be reduced, while too many cases can introduce redundancy into the case-base that ultimately leads to efficiency degradation. In recent years, with the deployment of commercial, large-scale CBR systems, the goal of producing and maintaining an optimal case-base has been brought sharply into focus [13, 17]. In turn the CBR community is beginning to recognise the need for new tools and techniques to support authoring and maintenance processes [11, 17]. We believe that effective visualisation techniques can play an important role in this respect.

Our primary objective in this paper is to explore the potential for improved dialogue and interaction between author and CBR system, by providing the author with access to a visualisation tool capable of accurately visualising the contents and structure of a case-base. We believe that this type of visualisation environment can provide an important and useful function during authoring and maintenance, and may serve as the foundation for a range of new intelligent authoring environments. Using the visualiser the author can perceive the structure of a case-base and the relationships that exists between individual (and groups of) cases. The author can recognise regions of high density (redundancy) and the presence of holes (low coverage) within the case-base, and act accordingly.

The next section surveys related work, focusing on the use of visualisation techniques within case-based reasoning. In Sections 3 and 4 we describe and evaluate our case-base visualisation technique, and finally Section 5 outlines some important applications of the resulting visualisation tool.

2 RELATED WORK

Visualisation techniques are used in artificial intelligence, information retrieval, data analysis, and data mining to help users to discover patterns and trends in complex information spaces that might otherwise be missed [2, 3, 4, 5, 7, 8, 12, 18].

However, to date there has been only limited use of visualisation techniques as part of the CBR problem-solving model. Some systems use graphical visualisations as a way to present case solutions that have a natural graphical form. For example, Macura & Macura [10] describe the MacRad case retrieval system for assisting the diagnostic process in radiology: cases contain medical image scans as part of their solutions structures.

Similarly, Wybo et al. [19] describe PROFIL, a CBR system for decision support in a design domain. Again cases contain a visual solution component - each case includes an annotated image of a given design. In addition, PROFIL uses a visualisation technique to present users with a representation of the cases retrieved for a given query. Cases are plotted on a two-dimensional graph of similarity (to the target query) versus solution quality (of a selected case); the target query is the graph's origin. Thus, the user can perceive the relationship between the target and similar cases. An important limitation of this visualisation technique is that while it preserves the similarity relationship between the fixed target query and retrieved cases, the similarity between the retrieved cases themselves is lost. Dissimilar retrieved cases can appear close on the screen. Therefore, this technique is not useful when it comes to visualising a case-base as a whole, where there is no fixed point of reference (such as a specific target query).

This problem of visualising a complete case-base is addressed by the work of Smyth & McKenna [11, 16]. A visualisation technique is described based on a novel model of competence for case-based reasoning systems. The model makes it possible to identify groups of cases that make shared contributions to overall system competence, and to measure the competence of these groups. The visualisation technique constructs a graphical representation of a case-base by plotting each group on a graph of group competence versus group

[1] Smart Media Institute, Department of Computer Science, University College Dublin, Belfield, Dublin 4, Ireland, email: {firstname.secondname}@ucd.ie

size. Smyth & McKenna describe how the visualisation technique can be put to good effect as an intelligent support tool for case-base authors, by highlighting over-populated and under-populated regions of the case-base. One of the limitations with this technique is that the similarity relationship between cases (or groups of cases) does not translate into on-screen distances. For example, groups may appear close simply because they have similar sizes and competence contributions, but their constituent cases may be unrelated.

We believe that visualisation methods have an important part to play in case-based reasoning by facilitating an improved interaction between system and user. Moreover, given the importance of cases and the case-base in CBR, visualisation techniques that provide a user with a visual representation of the case-base are likely to prove useful in many stages during the CBR process. In the next section we will introduce a new method for visualising case-bases that is specifically designed to model the similarity relationships between cases as on-screen distances.

3 CASE-BASE VISUALISATION

Case-base visualisation is non-trivial. Cases are complex n-dimensional objects (a case is composed of n features), and the similarities between cases represent distances in an n-dimensional feature space. In contrast, our target visualisation space is a two-dimensional computer screen. Thus, the essence of our visualisation problem is how to map n-dimensional cases onto a two-dimensional screen while preserving the similarity relationships between pairs of cases as on-screen distances.

We propose the use of a force-directed graph-drawing algorithm and in the following sections we describe the details of this algorithm and how it has been adapted for use as a case-base visualisation technique.

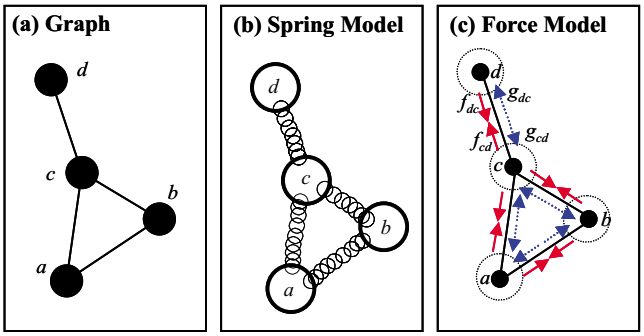

Figure 1. The force-directed graph-drawing algorithm models a graph as a system of rings and springs under attractive and repulsive forces.

3.1 A Force-Directed Graph-Drawing Algorithm

Huang et al. [6] describe a force-directed algorithm for graph drawing that is suitable as the basis for our case-base visualisation technique. The algorithm models a graph, $G = (V, E)$, as a system of steel rings and springs: the vertices are steel rings and the edges are springs. The springs exert an attractive force between connected rings and the steel rings exert a repulsive force (figure 1).

During graph drawing, the position of a vertex is influenced by these attractive and repulsive forces. The graph-drawing algorithm is an iterative process that begins with a random configuration of vertices and proceeds to locate a minimum energy configuration by incrementally adjusting the relative positions of vertices in order to equalise forces. We proceed to outline the basic mechanics of this *spring-embedded* graph-drawing approach. Many of the details have been omitted and the interested reader is referred to [6] and also [1].

3.1.1 *The Force Model*

The total force on a vertex v is given by (1), where f_{uv} is the attractive force exerted on v by the spring between u and v and g_{uv} is the repulsive force exerted between u and each of its neighbouring vertices; note, the set of edges of a vertex v is $N(v)$. The force f_{uv} follows Hooke's law and therefore is proportional to the difference between the distance between u and v and the zero-energy length of the connecting spring. The repulsive force between two nodes follows an inverse square law.

$$f(v) = \sum_{u \in N(v)} f_{uv} + \sum_{u \in V_i} g_{uv} \tag{1}$$

If we denote the Euclidean distance between two points p and q by d(p,q), and suppose that the position of vertex v is denoted by $p_v = (x_v, y_v)$, then, from (1), the x component of the force $f(v)$ on v is $f_x(v)$ and is given by (2); the y component has a similar expression.

$$\begin{aligned} f_x(v) \quad = \quad & \sum_{u \in N(v)} k_{uv}^{(1)} \frac{(d(p_u, p_v) - l_{uv})(x_v - x_u)}{d(p_u, p_v)} \\ & + \sum_{u \in V_i} k_{uv}^{(2)} \frac{(x_u - x_v)}{(d(p_u, p_v))^3} \end{aligned} \tag{2}$$

A number of new parameters are introduced by (2): l_{uv} is the zero-length energy of the spring between u and v, and $k_{uv}^{(1)}$ and $k_{uv}^{(2)}$ are the relative weights of the attractive and repulsive forces, respectively.

3.1.2 *The Animation Model*

As mentioned above, the spring-embedded algorithm iteratively adjusts the positions of all vertices until a minimum energy configuration is found, leaving the vertices in an equilibrium force configuration. This produces a sequence of animation frames $D_1, .., D_n$ such that D_1 displays an initial random configuration of vertices while D_n is the final *equilibrium frame*; each D_i represents a configuration that is closer to equilibrium than D_{i-1}. During an iteration the algorithm moves from D_i to D_{i+1} by computing the appropriate change in the x and y positions of each vertex. In the remainder of this section we show how to compute this change for the x coordinate of v, that is, $\triangle_x(v)$; the computations for the y coordinate are analogous.

By Newton's second law of motion, $f_x(v) = m(v).a_x(v)$, where $m(v)$ is the mass of vertex v and $a_x(v)$ is its acceleration in the x direction, due to a force f. If we assume that each vertex has a mass of one then $f_x(v) = a_x(v)$, and in a few simple steps Huang et al [6] explain how $\triangle_x(v)$ can be written as shown in (3).

$$\triangle_x(v) = \frac{a_x(t_{j-1})}{2} \triangle_t^2 \tag{3}$$

By substituting 3 into $f_x(v,) = a_x(v)$ we get,

$$\triangle_x(v) = \frac{f_x(v)}{2} \triangle_t^2 = C \bullet f_x(v) \tag{4}$$

where $\triangle_t$ is the time period of one animation step and $C = \frac{\triangle_t^2}{2}$; normally we set $\triangle_t$ = 0.5 seconds, so C = 1/200. Finally, we can transform the force model in (2) into the animation model in (5).

As it stands, this animation model can produce jumps in vertex positions rather than smoothly interpolated transitions. Fortunately, a simple solution is to limit the maximum distance a vertex can move in a single iteration according to (6).

$$\frac{\triangle_x(v)}{C} = \sum_{u \in N(v)} k_{uv}^{(1)} \frac{(d(p_u, p_v) - l_{uv})(x_v - x_u)}{d(p_u, p_v)} + \sum_{u \in V_i} k_{uv}^{(2)} \frac{(x_u - x_v)}{(d(p_u, p_v))^3} \quad (5)$$

$$\triangle_x(v) = \begin{array}{lll} -5 & if & \triangle_x(v) \leq -5 \\ \triangle_x(v) & if & -5 \leq \triangle_x(v) \leq 5 \\ +5 & if & 5 \leq \triangle_x(v) \end{array} \quad (6)$$

3.2 Modifications for Case-Base Visualisation

Two modifications are needed to use the above graph-drawing algorithm to visualise case-bases. First, a case-base is a fully connected graph. The cases correspond to the vertices of the graph and the similarity relations between cases correspond to the edges. Usually, it is possible to measure the similarity between each case and every other case and therefore every case can be linked to every other case by an edge. For this reason, there is no benefit in drawing the edges of the case-base graph, so only the graph vertices are drawn.

In the standard graph-drawing algorithm the principal presentation objective is to minimise the number of edges crossings in the graph [6]. This is not important however in our visualisation approach. The edges are not drawn for a start, so there can be no crossings, but more importantly the case-base should be drawn so that the screen distance between two cases corresponds to the similarity between these cases. Cases that are very similar should be drawn close together, and cases that are dissimilar should be drawn far apart. To achieve this we set the zero-energy length of the spring, which represents the relationship between two cases, to be a function of their inverse similarity. Thus, very similar cases will have a low zero-energy spring length to produce a higher attractive force between their corresponding on-screen vertices.

3.3 A Visualisation Example

Figure 2 illustrates a demonstration of the case-base visualiser in action. The figure consists of two screen shots taken during the visualisation of a small 25-case case-base (a small set of cases is used for clarity reasons). 2(a) shows the initial frame, with all cases occupying random screen positions. 2(b) shows the final equilibrium frame (after approximately 100 iterations). The figures have been annotated to indicate the movement of certain cases (C3, C12, C24) during the visualisation process.

The visualiser, implemented in Java and running on a 400MHz Pentium III PC, produces a smooth animation sequence that converges in approximately ten seconds for this 25-case case-base.

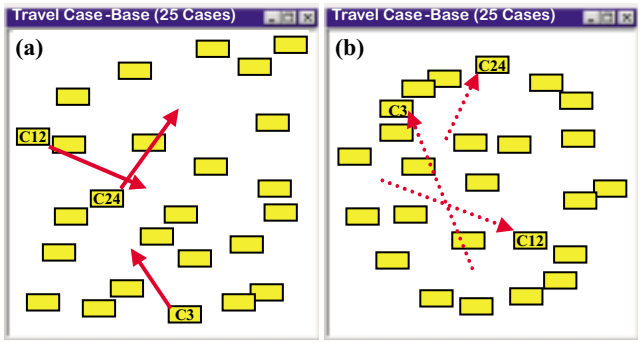

Figure 2. Screen-shots from the visualisation of a 25-case case-base: (a) the initial visualisation frame; (b) the final equilibrium frame

4 EVALUATION

Our main objective is to develop a visualisation technique capable of representing a case-base of cases on a two-dimensional screen, while preserving the similarity relationships between cases as on-screen distances. If these relationships are not reliably preserved, that is, if there is a poor correlation between the similarity of a case-pair and the pair's screen distance, then the utility of the visualisation will be limited. Obviously the transformation from an n-dimensional space to a two-dimensional one will come at a cost. The relationship between similarity and screen distance will be impaired, and in this section we describe a series of experiments to investigate the nature and degree of this impairment.

4.1 Similarity-Distance Correlation

One way to measure the strength of the relationship between case similarities and screen distances is to compute the correlation between the similarity values and the distance values of the case pairs (in the equilibrium frame). For example, in this experiment we use the Pearson's product-momentum correlation coefficient, which measures the degree of linear relationship between the true similarities and screen distances produced during the visualisation.

Method: A publicly available case-base is used as a source of test data. The cases represent package holidays (in terms of 9 features such as holiday type, duration, location, etc.) and are available from the case-base archive at AI-CBR (www.ai-cbr.org). In total the case-base contains 1400 different cases. A standard weighted-sum similarity metric is used to measure case similarities.

In this experiment we investigate the visualisations produced for case-bases of different sizes (10,25,50,100,200,300,400 cases). For each case-base size we randomly produce 20 case-bases from the original 1400 cases. Each case-base is visualised and the correlation coefficient between the similarities and screen distances in the equilibrium frame is calculated. The correlation coefficients are averaged to obtain a mean correlation of each of the different case-base sizes.

Results: The results are plotted in Figure 3 as a graph of mean correlation versus case-base size. The results are very positive, indicating a strong correlation (> 0.7) between similarity and screen distance across the range of case-base sizes tested. As expected the correlation values degrade for increasingly large case-bases - the limited visualisation space means that it becomes more and more difficult to layout large sets of cases in such a way that similarity relationships are preserved. These correlation coefficients are all significant at the 0.001 level and the results indicate that the visualisation technique

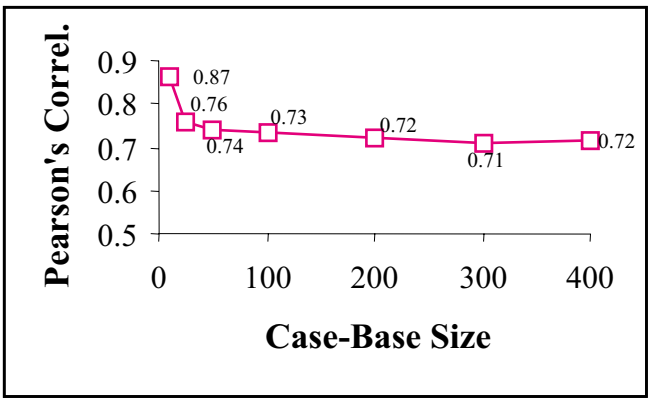

Figure 3. Pearson's correlation coefficient results

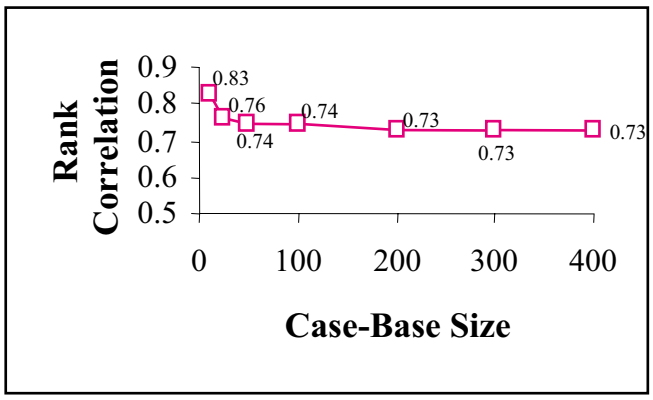

Figure 4. Spearman's rank correlation coefficient results

is capable of producing sufficiently accurate representations of real case-bases containing up to 400 cases.

4.2 Rank Correlation

The correlation coefficient used in the previous experiment is designed to take account of the relative magnitude between case similarities and screen distances. In this experiment we consider an alternative evaluation function that examines the relative ranking of cases according to their similarities and screen distances. We would like our visualisation technique to preserve this relative ordering as far as possible so that, for example, the i^{th} closest pair of cases in terms of screen distance is also the i^{th} closest pair of cases in terms of case similarity.

Method: The experimental method used above is repeated except that instead of calculating Pearson's correlation coefficient we calculate Spearman's rank correlation coefficient, which is designed to explicitly examine the correlation between the relative ranking of paired data points. As before, a mean correlation coefficient is calculated for each case-base size.

Results: The results are plotted in Figure 4 as a graph of the mean rank correlation coefficient versus case-base size. Again the results are very positive, indicating a strong correlation (> 0.7) between similarity and screen distance across the range of case-base sizes tested. The rank correlation values are marginally higher than the Pearson's correlation values - the visualisation method seems to model the relative ranking between case pairs more accurately than the relative magnitude of the relationships between pairs. Once again, as expected, the correlation coefficient falls slowly as the case-base size increases, indicating that the visualisation method scales well with case-base size, and certainly produces accurate visualisations for case-bases of up to 400 cases. These results are significant at the 0.001 level.

4.3 Discussion

In general, the above results are extremely positive. They show that the visualisation technique is generating useful visualisations of large case-bases, in the sense that the similarity relationships between cases are being accurately translated into equivalent on-screen distances.

As it stands there are a number of ways to tweak the visualisation model in order to explore the possibility of producing improved visualisations. The two main tuning parameters available are the constants used in the force model to weight the contribution of the attractive ($k_{uv}^{(1)}$) and repulsive ($k_{uv}^{(2)}$) forces. For the purpose of our experiments we set these values to the defaults recommended by [6], that is, -1/30 and 3, respectively. However, these defaults were originally chosen in [6] to produce a graph visualisation with minimum edge-crossings and, as we explained in Section 3, this is different from our objective, which is to produce a graph visualisation that preserves the similarity between cases. By adjusting these constants it may be possible to improve further the visualisation accuracy. For instance, increasing the weight assigned to the spring-force component should bias the visualisation in favour of similarity preservation. Early experiments confirm this, with correlation increases of over 10% for $k_{uv}^{(1)}$=0.5. Future work will further investigate the impact of $k_{uv}^{(1)}$ and $k_{uv}^{(2)}$ on the final visualisation accuracy.

One final point is worth noting. The ability of the visualisation technique to model similarity on a two dimensional canvas will depend on the area of this canvas: a larger visualisation space offers more opportunities to resolve conflicts between similarity and screen-distances. Our experiments used a 600x600-pixel canvas and improved correlations are possible by increasing this viewing space; again, a complete impact assessment is left for future work.

5 APPLICATIONS

The motivation for this work is the hypothesis that an effective tool for visualising case-bases and the relationship between cases has the potential to revolutionise case-base authoring and maintenance processes, as well as improving the interaction between users and system at problem solving time.

From an authoring and maintenance viewpoint, the visualisation tool will help the user to: (1) perceive the overall structure of an evolving case-base; (2) recognise emerging regions of competence within the case-base, that is, large clusters of cases; (3) recognise potentially redundant regions of the case-base, which contain densely packed clusters of similar cases; (4) recognise potential holes within the case-base indicating regions of poor competence; (5) recognise new regions of competence or exceptional cases that are outliers within the case-base.

We are currently combining this visualisation tool with a computational model of competence for case-based reasoning systems [11, 14, 16]. The competence model can be used to estimate the coverage properties of individual cases and as such can assign coverage

values to particular cases. These values can be used to separate cases according to their competence contributions; for example, redundant cases have low values while important cases have higher values. The model can be used to enhance the visualisation output by annotating case vertices according to their coverage values to provide the author with a instant picture of the structural and coverage properties of a case-base.

Aside from improving the capabilities of authoring and maintenance systems, the visualisation technique also has applications in other parts of the CBR process. For example, it could provide a useful interface for presenting case retrieval results (see also [5]), allowing the user to perceive the relationship between retrieved cases at a glance.

6 CONCLUSIONS

Visualisation offers a powerful means of analysis that can help to uncover patterns and trends in data sets that may be missed by other non-visual methods. We believe that visualisation techniques may hold the key to the next generation of interactive case-based reasoning systems, by facilitating new modes of interaction between system and user, and by supporting more intelligent case-base authoring and maintenance strategies.

We have described and adapted a force-directed graph-drawing technique for use as a case-base visualisation technique - cases are represented as graph vertices and the screen-distances between cases are a proxy for similarity. In addition, we have evaluated the technique on a range of case-bases to show that it successfully preserves the n-dimensional similarity relationships between cases.

Our future work will continue to develop the above visualisation technique. We plan to further adapt the method to meet the specific needs of case-base visualisation, as outlined in Section 4.3. Finally, we are currently integrating the visualisation tool into the CASCADE system, a novel case-based reasoning shell that incorporates an explicit competence model for CBR to provide a case author with a range of innovative support facilities. The current visualisation tool will enhance CASCADE significantly.

Finally, before closing we should point out that in the context of advanced computer graphics research our visualisation technique is not particularly ground-breaking as many researchers have investigated similar graph-drawing techniques prior to this work [6, 1]. However, the work is significant in that it brings these visualisation techniques to the AI and CBR community in the first place. We believe that there is much to be learned and we plan to investigate a range of visualisation techniques in the future that will serve as alternatives to the current two-dimensional graph-drawing method.

REFERENCES

[1] G. Di Battista, P. Eades, R. Tamassia, and I. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice Hall, 1999.

[2] Robert Bosch, Chris Stolte, Diane Tang, John Gerth, Mendel Rosenblum, and Pat Hanrahan, 'The Information Mural: A Technique for Displaying and Navigating Large Information Spaces', *Computer Graphics*, (2000).

[3] M. Devaney and A. Ram, 'Visualization as an Exploratory Tool in Artificial Intelligence', in *Proceedings of the World Multiconference on Systemics, Cybernetics, and Informatics*, (1998).

[4] A.K. Goel, G. Gomez de Silva Garza, S. Garza, N. Grue, JW. Murdock, MM. Recker, and T. Govindaraj, 'Explanatory Interface in Interactive Design Environments', in *Proceedings of the 4th International Conference on AI in Design*, (1996).

[5] MA. Hearst and JO. Pedersen, 'Visualising Information Retrieval Results: A Demonstration of the TileBar Interface', in *Proceedings of Conference on Human Factors in Computing Systems '96*, eds., R. Bilger, S. Guest, and M.J. Tauber. ACM Inc., (1996).

[6] ML. Huang, P. Eades, and J. Wang, 'On-line animated visualisation of huge graphs using a modified spring algorithm', *IEEE Transactions on Computers*, **9**, 623–645, (1998).

[7] D. Jerding and J. Stasko, 'The Information Mural: A Technique for Displaying and Navigating Large Information Spaces', *IEEE Transactions on Visualization and Computer Graphics*, **4(3)**, 257–271, (1998).

[8] D. Keim and G. Wills, 'Visualizing World-Wide Web Search Engine Results', in *Proceedings of the 1999 IEEE Symposium on Information Vizualisation*. IEEE Press, (1999).

[9] J. Kolodner, *Case-Based Reasoning*, Morgan Kaufmann, 1993.

[10] RT. Macura and KJ. Macura, 'MacRad: Radiology Image Resource with a Case-Based Retrieval System.', in *Proceedings of the 1st International Conference on Case-Based Reasoning*, eds., M. Veloso and A. Aamodt, pp. 43–54. Springer Verlag, (1995).

[11] E. McKenna and B. Smyth, 'An Interactive Visualisation Tool for Case-Based Reasoners', *Applied Intelligence: Special Issue on Interactive Case-Based Reasoning*, (2000).

[12] S. Mukherjea and Y. Hara, 'Visualizing World-Wide Web Search Engine Results', in *Proceedings of the International Conference on Information Visualization*. IEEE Press, (1999).

[13] B. Smyth and P. Cunningham, 'The Utility Problem Analysed: A Case-Based Reasoning Perspective', in *Advances in Case-Based Reasoning. Lecture Notes in Artificial Intelligence*, eds., Ian Smith and Boi Faltings, pp. 392–399. Springer Verlag, (1996).

[14] B. Smyth and M.T. Keane, 'Remembering to Forget: A Competence Preserving Case Deletion Policy for CBR Systems', in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, ed., Chris Mellish, pp. 377–382. Morgan Kaufmann, (1995).

[15] B. Smyth and M.T. Keane, 'Adaptation-Guided Retrieval: Questioning the Similarity Assumption in Reasoning', *Artificial Intelligence*, **102**, 249–293, (1998).

[16] B. Smyth and E. McKenna, 'Modelling the Competence of Case-Bases', in *Advances in Case-Based Reasoning. Lecture Notes in Artificial Intelligence*, eds., B. Smyth and P. Cunningham, pp. 208–220. Springer Verlag, (1998).

[17] B. Smyth and E. McKenna, 'Building Compact Competent Case-Bases', in *Case-Based Reasoning Research and Development. Lecture Notes in Artificial Intelligence*, eds., Klaus Dieter Althoff, Ralph Bergmann, and L.Karl Branting, pp. 329–342. Springer Verlag, (1999).

[18] C. Westphal and T. Blaxton, *Data Mining Solutions: Methods and Tools for Real World Problems*, Wiley, 1998.

[19] JL. Wybo, F. Gefraye, and A. Russeil, 'PROFIL: A Decision Support Tool for Metallic Sections Design using a CBR Approach', in *Proceedings of the 1st International Conference on Case-Based Reasoning*, eds., M. Veloso and A. Aamodt. Springer Verlag, (1995).