Navigating Through Case Base Competence

Maarten Grachten, F. Alejandro García, and Josep Lluís Arcos

IIIA, Artificial Intelligence Research Institute, CSIC, Spanish Council
for Scientific Research, Campus UAB, 08193 Bellaterra, Catalonia, Spain
`{maarten, fgarcia, arcos}@iiia.csic.es`
`http://www.iiia.csic.es`

Abstract. The development of large-scale case-based reasoning systems has increased the necessity of providing tools for analyzing the case base structure. In this paper we present a hierarchical competence model approach based on the solution qualities. Using this hierarchical approach we propose a new method for visualizing case base competence and understanding the way a CBR system behaves in different parts of the problem space. The visualization method has been used in the *Tempo-Express* system, a CBR system for applying expressivity-aware tempo transformations to recordings of musical performances.

1 Introduction

The development of large-scale case-based reasoning systems has increased the necessity of providing tools for analysing the case base structure and its relation with the similarity measures used in the retrieval phase [1,2]. These tools may be used either in the design stage or in the maintenance stage of the CBR systems.

Reinartz and Iglezakis [3] proposed a collection of properties for monitoring the quality of a CBR system. Moreover, they defined a collection of modify operators on cases for improving the quality of the case base. Their proposal is focused on syntactical measures and tries to avoid domain-specific measures.

The competence model introduced by Smyth et al. [4] is a nice contribution of the analysis of case base structure by assessing the local competence contributions of cases and their interactions. The competence model proposes the use of a *Solves* relation between cases (being either true or false for a given pair of cases). This interpretation of the *Solves* concept (being either true or false) is obvious for classification problems but may be inappropriate for other tasks such as design or configuration. In these tasks it seems more natural to define *Solves* as a *function* (indicating the quality of the solution) rather than a relation. In this paper we present the concept of an hierarchical competence model that is based on such a function and allows for a finer analysis of the case base structure.

We believe that, with increasing complexity of CBR systems, the analysis of the case base structure becomes a hard task without the support of tools capable of accurately visualizing the complex case base structure. The navigation through the case base space may play an important role for understanding the similarity relationships between cases and the quality of the contribution of a given case to the solution of other problems.

H. Muñoz-Avila and F. Ricci (Eds.): ICCBR 2005, LNCS 3620, pp. 282–295, 2005.

Previous work on visualizing the case base structure includes the PROFIL system [5] and the *Picture Perfect* tool [6]. The PROFIL system is a CBR decision support tool for metallic sections design that provides a visualization tool for relating target problem with the collection of retrieved cases. Cases are plotted on a two-dimensional plane where the first dimension represents the similarity of the cases with the target case and the second dimension represents the solution quality. Nevertheless, the visualization is problem centered and only preserves the similarity relationship between the target problem and each case. That is, the similarity relationship among the retrieved cases is lost.

The *Picture Perfect* tool [6] provides an alternative two-dimensional plot where the similarity relationships among all the cases of the case base is preserved. A force-directed graph-drawing algorithm is used for preserving the similarity relationships among cases. The algorithm is an iterative algorithm that uses the case similarities as force vectors. The drawback of the approach is that the quality of solutions is not visualized.

Using the competence model analysis, Smyth et al. [4] proposed a case-authoring tool for visualizing the competence of an evolving case base and help the application designers to identify redundant cases for deletion and useful new cases for addition. Nevertheless, the visualization tool is focused on showing the relationship between the competence group sizes and their coverage.

We propose a new visualization method for case base competence based on the solution qualities. This method allows us not only to draw 'competence islands' in an 'unsolved ocean', but rather to draw the complete surfaces, with hills and valleys.

With respect to the mapping, this poses some new problems. In complex CBR systems, it is usually impossible to find a mapping of the cases to the two-dimensional plane that preserves the case distances. When the distortion is too high, it is impossible to draw a competence map using straight-forward 2D multidimensional scaling (the competence groups would not appear as separated regions). Therefore we propose an alternative way of mapping the cases to the two-dimensional plane, that uses both case distance information and hierarchical competence information.

The paper is organized as follows: In section 2 the competence model is summarized and extended. In section 3 we present a new technique for visualizing competence surfaces using the notion of hierarchical competence groups presented in section 2. In section 4 we exemplify and report the use of the visualization technique in the *TempoExpress* system, a CBR system for applying expressivity-aware tempo transformations to recordings of musical performances. The paper ends with a discussion of the results, and the planned future work.

2 Computation of Case Base Competence

Competence groups were defined by Smyth and McKenna [4] as a proposal for an effective model of case base global competence measure that assesses the local competence contributions of cases and their interactions. Competence groups are defined from the notions of *coverage* and *reachability*. The coverage set of a case c_i is defined as the set of all target problems that can be solved using c_i. The reachability set of a target problem is defined as the set of all cases that can be used to solve it. Formally:

$$CoverageSet(c_i \in CB) = \{c_j \in CB : Solves(c_i, c_j)\} \tag{1}$$

$$ReachabilitySet(c_i \in CB) = \{c_j \in CB : Solves(c_j, c_i)\} \tag{2}$$

where the *Solves* predicate has to be defined for the CBR system under inspection.

From the coverage and reachability definitions, Smyth and McKenna define a *Related Set* of a case c_i as the union of its coverage and reachability sets. Then, a set of cases $G \subseteq CB$ is a competence group if and only if:

$$\begin{aligned} &\forall c_i \in G, \exists c_j \in G - \{c_i\} : SharedCoverage(c_i, c_j) \\ &\wedge \forall c_i \in G, \nexists c_j \in CB - G : SharedCoverage(c_i, c_j) \end{aligned} \tag{3}$$

where two cases have a *SharedCoverage* when their related sets have a non empty intersection.

Using the notion of competence groups, the case base can be organized with a set of case clusters that do not interact from a competence viewpoint.

The use of a *Solves* predicate is possibly a good indicator in analytical tasks (see [7] for a conceptual distinction of CBR tasks). In analytical tasks there is a limited number of solutions and solutions are simple, non-aggregate entities (classification/diagnosis is a typical analytical task). Nevertheless, in synthetic tasks—where the solutions have a composite structure, and as a result the number of possible solutions is usually very large (a typical example of a synthetic task is structural design)—modeling *Solves* as a binary predicate on cases of the case base CB (a subset of $CB \times CB$) is not satisfactory. CBR systems for solving synthetic tasks can be viewed as systems that locally approximate a complex target function. In that context, it is more natural to conceive of the *Solves* notion as a function of type $CB \times CB \rightarrow [0, 1]$ that assesses the quality of the solution. Thus, in synthetic tasks we will say that the solution generated from a case c_j for a target problem c_i is of quality γ.

Then, we can extend the definitions of coverage and reachability in the following way:

$$CoverageSet_\gamma(c_i \in CB) = \{c_j \in CB : \gamma \leq Solves(c_i, c_j)\} \tag{4}$$

$$ReachabilitySet_\gamma(c_i \in CB) = \{c_j \in CB : \gamma \leq Solves(c_j, c_i)\} \tag{5}$$

where γ can take values in the interval [0,1].

Using the above equations (4) and (5), the competence groups defined in a given case base may vary depending on the threshold value used for γ. Then, defining a collection of γ-cuts a hierarchical competence model of the case base can be constructed.

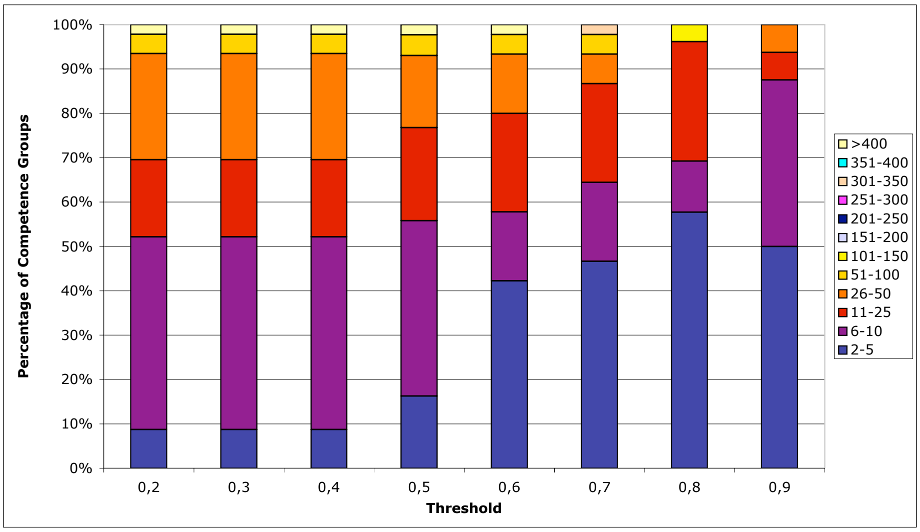

Fig. 1. Competence group sizes in *TempoExpress* for different γ values

This hierarchical competence model allows a finer analysis of the case base competence. The first analysis we can perform is the study of the changes on the sizes of the competence groups when we relax the quality criterion γ.

Figure 1 shows the effect varying the quality threshold in the *TempoExpress* system (see section 4). For a quality threshold of 0.9 (right side column) 50 % of the competence groups are formed by at most 5 cases and 36 % of the competence groups are formed by collections between 6 and 10 cases. On the other side, for a quality threshold of 0.2 there are competence groups with over 400 cases — more than a quarter of the whole case base size.

Given a hierarchical competence model of the case base, it is interesting to analyse the correlation between the case similarities and the quality of the solutions they provide. For this purpose, in the next section we will describe a technique for visualizing the case base taking into account this relationship.

3 Mapping Competence to the Plane

After obtaining the competence partitioning of the cases base at several threshold levels, we can map the cases to a plane, in order to visualize the hierarchical structure of the partitioning. Ideally, the cases would be mapped to the plane so that their mutual euclidean distances are proportional to their real distances. But as mentioned before, there is no guarantee that high dimensional data can be faithfully mapped to a two-dimensional plane. As a consequence, the positioning of the cases according to their real distances do not necessarily provide a good separation of the competence groups. Therefore, a method is needed to alter the case positioning in such a way that the competence groups at each threshold level are spatially separated, preferably with minimal distortion of the real case distances. In this section we propose an algorithm for finding a mapping in the two-dimensional plane that satisfies our requirements. The method is similar to the visualization technique employed by Smyth et al. [6], in the sense that it starts with a random positioning of the cases in the plane and iteratively changes the positions to have the euclidean distances in the plane approach the real distances between the cases. Our method however is more elaborate to accommodate for the additional requirements that are involved to draw the hierarchical competence groups.

The input to the mapping algorithm is the hierarchical structure of competence groups, together with a distance matrix D containing the distances between all pairs of available cases. Rather than considering the competence groups as sets of cases (which they are really), we consider them as nodes in a tree. Nodes at the lowest level in the tree (i.e. with the highest γ-threshold) have as children the cases that are in the corresponding competence groups. But nodes at higher levels have nodes as children rather than cases. To position the set of nodes at a particular γ-level, it is necessary to know something about the way the children of those nodes are arranged. This implies a bottom up traversal of the tree, positioning the nodes level by level.

The first step is thus to position the cases of each node at the lowest level independently. For every bottom-level node, the positioning of its cases is guided by a single (soft) constraint:

- the euclidean distance between two cases in the plane should be equal to the target distance d^t(as defined in D) between the cases

In an iterative process a random positioning of the cases is repeatedly altered to satisfy this constraint as good as possible. If no more progress can be made, the iteration is stopped. The resulting positioning is saved. Note that at this point the cases in the nodes are only positioned internally to the node, not with respect to the cases in the other nodes at the same level. But since we calculated the node internal positionings, we can now compute the positioning of the nodes with respect to each other. To do that, we calculate two values for each node n: the *centroid* and *width*:

$$Centroid(n) = \frac{1}{N} \sum_{c \in Children(n)} p_c = \frac{1}{N} \sum_{c \in Children(n)} \langle x_c, y_c \rangle \tag{6}$$

$$Width(n) = \max_{c \in Children(n)} d(\langle x_c, y_c \rangle, Centroid(n)) \tag{7}$$

where N is the number of children of node n and $p_c = \langle x_c, y_c \rangle$ is the position of case c in the plane. The centroid is the center of gravity of the positioning of the children and the width of the node is the euclidean distance between the centroid and the child furthest away from the centroid.

The positioning of the nodes with respect to each other is then guided by two (soft) constraints:

1. the distance between the centroids of two nodes should not be smaller than the sum of the widths of the two nodes.
2. the euclidean distance between the centroids of two nodes should be equal to the target distance between two nodes.

The target distance between two nodes n_i, and n_j is defined simply as the average target distance between the cases of the corresponding competence groups $CG(n_i)$, and $CG(n_j)$:

$$d^t(n_i, n_j) = \sum_{c \in CG(n_i)} \sum_{c' \in CG(n_j)} d^t(c, c') \tag{8}$$

In the same way as the cases were positioned, the nodes are positioned by iteratively adapting a random positioning to satisfy the constraints. At each iteration, the two constraints are used to calculate two new positionings from the previous positioning. The two positionings are combined linearly to obtain the final positioning for that iteration. As before, when no further improvements can be made to the positioning, the iteration is stopped, the positions of each node are saved, and the process is repeated for the parent nodes.

When the tree has been traversed from bottom to top in this way, we have a positioning for each node in the tree (the position of the root node was not derived but is set to the origin). But note that the positions that were computed for the cases in the initial stage were not updated after computing the positions of the parent nodes. So the final stage is to traverse the tree again in a top down manner to propagate the parent positions down to the children. So for every child n its position p_n is updated as follows:

$$p_n = p_n + p_{Parent(n)} - Centroid(Parent(n))$$

The resulting positioning of cases in the two-dimensional plane will reflect the real distances between the cases as good as possible while at the same time preventing overlap between competence groups at the same γ-level.

3.1 Analysis of Various Competence Scenarios

The mapping obtained in this way provides valuable information about the way a CBR system behaves in different parts of the problem space. Some typical scenarios have been plotted in figure 2. The figure shows the contours of the competence groups for the complete range of γ-values. Dark colors represent regions with low competence without cases (or low competence cases) and light colors represent regions with high competence.

Figure 2(a) shows a part of the problem space where there are many cases that form a single high competence group, without low competence cases (i.e. even with a high solution quality threshold, the cases have shared coverage with each other). This means that in such a region, a case can be solved well even if there is not a very nearby case. Another situation is shown in figure 2(b), which is also a well covered region, but it is composed of separated high competence sub

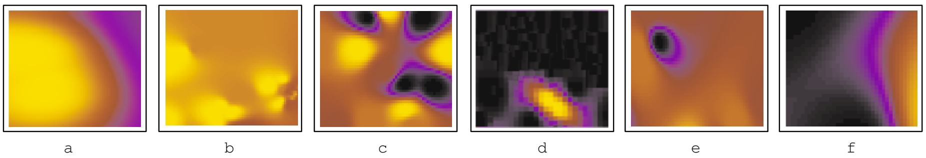

Fig. 2. Some typical competence scenarios

regions. So although cases can be solved well here, the target solutions are not the same for every part of the region. Figure 2(c) shows a situation where high competence and low competence regions are mixed. This means that even though quite similar problems can be retrieved from the case base, they may not provide a good solution for the target problem. In figure 2(d) a region is shown that has only a single dense competence group in an otherwise low competence area. In this scenario, it is probable that the region needs many more cases to provide good competence. The opposite is shown in figure 2(e), where a predominantly high competence region contains single dense low competence area, implying that although the cases in the region can be generally be solved well, there are some similar cases that are hard to solve, and cannot either be used to solve other cases in the region. Finally, figure 2(f) shows a region with low competence. This may indicate that there are either no cases at all in this region, or the cases in this region all have low competence.

4 Experimentation

We have applied the techniques described in this paper in the *TempoExpress* system [8]. *TempoExpress* is a CBR system for applying expressivity-aware tempo transformations to monophonic audio recordings of musical performances. *TempoExpress* has a rich description of the musical expressivity of the performances, that includes not only timing deviations of performed score notes, but also represents more rigorous kinds of expressivity such as note ornamentation, and note consolidation/fragmentation. Within the tempo transformation process, the expressivity of the performance is adjusted in such a way that the result sounds expressively natural for the new tempo. A case base of previously performed melodies is used to infer the appropriate expressivity.

A case is represented as a complex structure embodying three different kinds of knowledge: (1) the representation of the musical score (notes and chords), (2) the musical model of the score (automatically inferred from the score using Narmour's Implication/Realization model and Lerdahl and Jackendoff's Generative Theory of Tonal Music as background musical knowledge [9,10]), and (3) a collection of annotated performances. These annotated performances are acquired automatically from the recordings using a technique explained in detail in [11].

For the case base design, several saxophone performances were recorded from 5 jazz standards, each one consisting of 4–5 distinct phrases. The performances were played by a professional performer, at 9–14 different tempos per phrase.

From this, the initial case base was constructed, containing 20 scores of musical phrases, each with about 11 annotated performances (in total more than 5.000 performed notes).

When a new problem has to be solved in *TempoExpress*—i.e. an input phrase performance that must be transformed to another tempo—the problem is solved stepwise by decomposing the input phrase into segments. These segments are sequences of consecutive notes of around five notes and usually correspond to the musical motifs that constitute the musical phrase. The solution for each input melody segment is constructed from the most similar melody segments in the case base.

We have analyzed the *TempoExpress* case base composed of 1310 cases. A case consists of a phrase performance at a particular tempo (the input tempo) and a number representing the desired output tempo. Because an output performance is generated segmentwise for the input case, segments from various

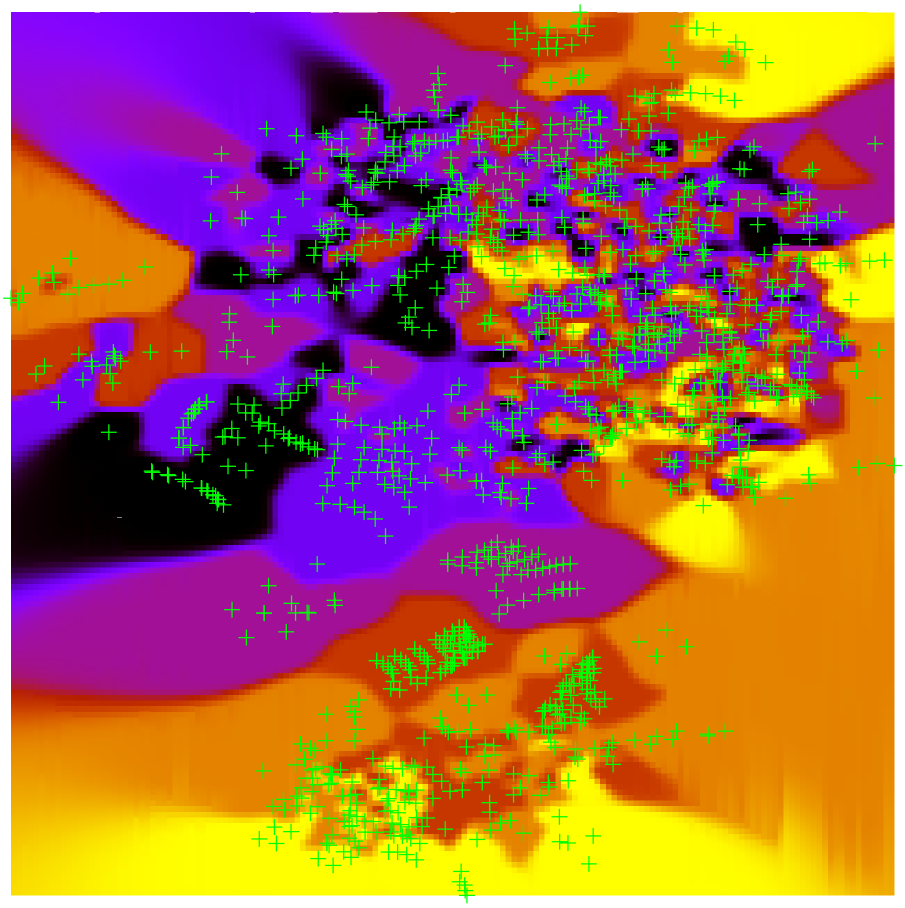

Fig. 3. Contour representation of the competence of *TempoExpress*

retrieved cases are usually involved in the solution of the different segments of the problem. As a consequence, a case may provide only a partial solution to the problem. To represent this relation, we define a solution function as follows:

$$Solves(c_i, c_j) = \frac{||SolvedNotes(c_i, c_j)||}{||Notes(c_j)||} \tag{9}$$

Where $SolvedNotes(c_i, c_j)$ are the notes in the melodic phrase of c_j that were provided with a solution (an expressive interpretation) from the retrieved case c_i (whether a solution for a note can be provided depends on whether segments can be convincingly matched between the input and retrieved phrases). $Notes(c_j)$ is the complete sequence of notes in the melodic phrase of c_j. Rather than representing the true $Solves$ relation, this is a confidence measure for the solution that serves as an approximation. Roughly speaking, the confidence measure is proportional the amount of solution information that could be transferred from the retrieved solutions to the current problem.

The distance function between cases is a linear combination of the pairwise distance of the three case components: phrase, input tempo, and output tempo. The phrase distance is measured as the edit distance between abstract sequential representations of the phrases (using the Implication/Realization model [9], see [12] for details).

With the $Solves$ function as defined above, we computed competence groups at ten different quality threshold values (see section 2). Using the case distance function explained above, the resulting hierarchical competence structure was mapped to the two-dimensional plane, following the method described in section 3. The results are shown as a contour plot in figure 3, and as a 3D surface in figure 4. In both figures, low competence regions are represented by darker colors and high competence regions are represented by lighter colors. In the 3D plot, valleys and hills correspond to low and high competence regions respectively. In the contour plot, the cases are plotted on top of the map as plus signs[1] (the colored map was derived from the scattered case information using *gnuplot*'s *dgrid3d* function).

Viewing the contour map at a glance, some comments can be made. The map shows a rather non-homogeneous distribution of cases and competence areas. The lattice-like positioning of some groups of cases (mostly in the lower part of the figure) reflects the fact that the case distance takes into account the input and output tempos of the cases (phrase performances are available at regularly spaced tempos). It makes sense that each of these lattice structures tends to have a single competence level, since the cases within the structures are various tempo transformation tasks of the same phrase, and the major factor determining whether a case is hard to solve is the phrase (i.e. whether the phrase consists of melodic fragments for which examples are known). Note also that the larger single-colored areas at the edges of the figure should be interpreted with some

---

[1] The plus signs on some shades may be hard to see when printed in black and white. It is recommended to inspect the pdf version of this document which contains colored graphics. Feel free to contact the authors to obtain an electronic copy.

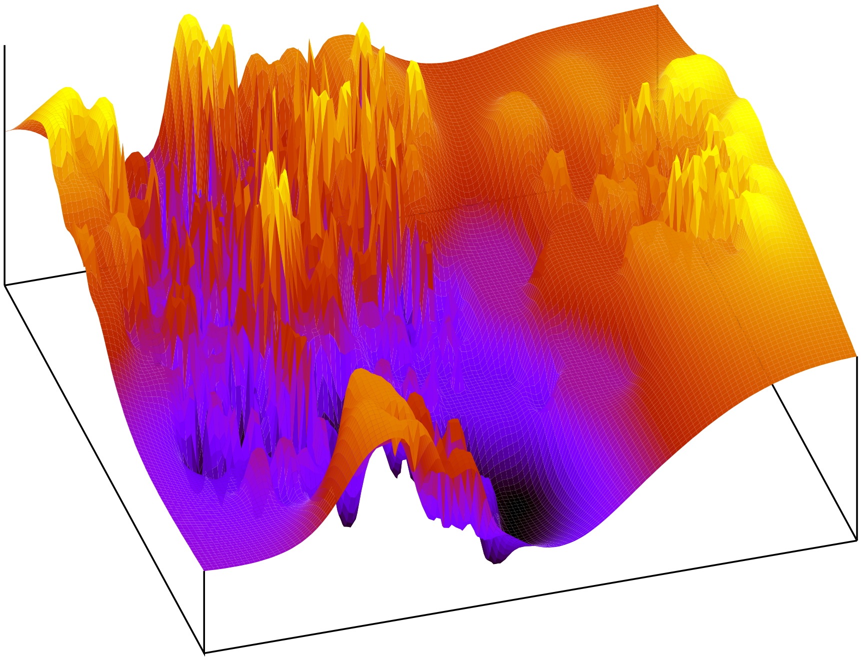

Fig. 4. 3D surface representation of the competence of *TempoExpress*

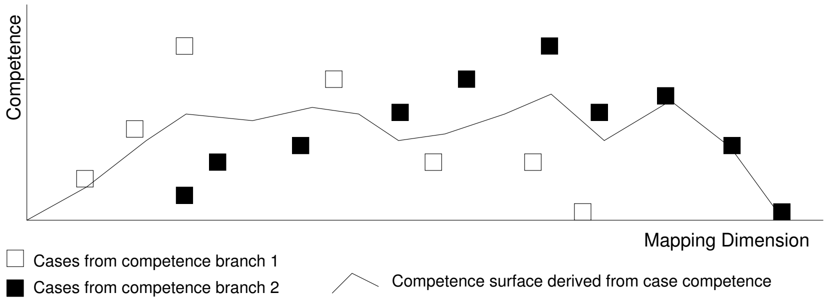

Fig. 5. Problems of non-ideal mapping: Branches from the hierarchical competence structure overlap and the competence surface does not accurately reflect the true competence of the cases

care, since they are unpopulated and the competence estimates mainly result from far-reaching extrapolations of the competence of the nearest-by cases.

The contour map shows roughly three distinct areas within the problem space. Firstly, in the upperleft quadrant of the map there is a coherent set of problems for which no good solution could be constructed (conform scenario *f*, section 3.1). Secondly, in the lower part of the map, there is another rather pop-

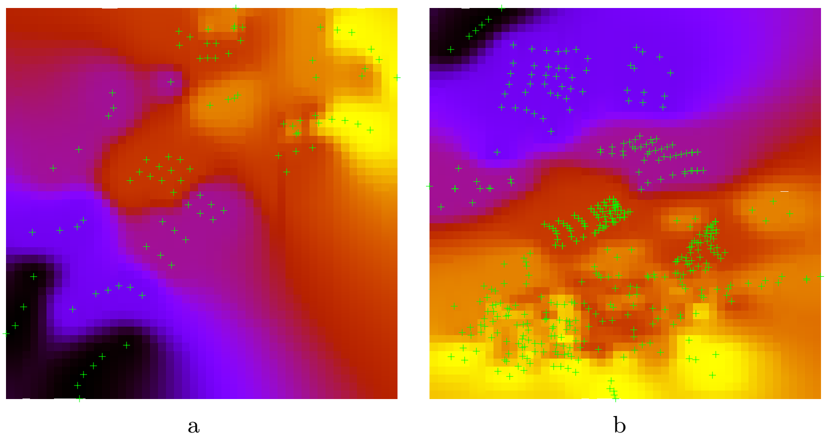

Fig. 6. Two competence branches in isolation

ulated area for which generally good solutions are found (conform scenario *a*/*b*, section 3.1). Lastly, there is a mixed competence region in the upperright quadrant, that shows scattered high and low competence groups (conform scenario *c*, section 3.1).

A disadvantage of the visualization technique is that the final positioning of the cases is only an approximation of a map that satisfies the constraints of faithful case distances and non-overlapping groups (since usually there is no map that completely satisfies both constraints at the same time). It thus sometimes happens that in the global competence map (figure 3), the shade indicating the competence is an average of partially overlapping competence branches of the hierarchical competence structure. Figure 5 shows this situation schematically for two competence branches mapped on a single dimension. In order to get a better impression of individual competence branches, it is therefore useful to view them in isolation.

The competence tree of the *TempoExpress* case base turned out to consist of 46 competence branches just below the root of the tree. In figure 6, two of such branches are shown. Note that the competence distribution of in these maps is less complex than the distribution of the global map. Apart from the fact that the number of cases is smaller, the relation between the positioning of the cases and their competence is more comprehensible. Figure 6(a), for example shows a pattern of steadily increasing competence from the lowerleft to the upperright corner. Inspection of the individual cases showed that cases clustered at a particular competence level tended to have the same musical phrase. A clear relation between competence and input or output tempo was not found. Additionally, note that the case-distance for this particular subset could be mapped to a single dimension, since the cases are positioned roughly on a straight line.

Figure 6(b) shows another, relatively large branch. Since this branch is well separated spatially, it is easy to locate it in the global contour map. It corresponds to the lower part of the map, that was identified earlier as the major high competence area of the map. As before, particularly for the lower competence levels, a clustering of cases in various competence levels can be noticed, that turns out to be correlated with the musical phrase of the case. There are some phrases, like *Body And Soul [phrase B2]* (Green), and *Like Someone In Love [phrase B2]* (Van Heusen/Burke), that appear in the low competence regions of both branches. Since cases pertaining to those phrases tend to be in low competence areas, regardless of the input and output tempos of the cases, an obvious conclusion is that the case base lacks musical material sufficiently similar to those phrase, and therefore no good tempo transformations can be constructed for those phrases.

On the other hand, the phrases *Like Someone In Love [phrase A1]*, and *Up Jumped Spring [phrase A1]* (Hubbard) occur on the high competence region of both branches. The latter case proves that even distinct phrases can have shared coverage, and that a solution to one can be helpful to construct a solution of the other (this is possible, since the final solution is constructed from *parts* of other solutions).

5 Conclusions

We believe that in the design and maintenance of complex CBR systems, the use of tools for analyzing the case base structure become indispensable. Moreover, these analysis tools must be capable of accurately visualizing the complex case base structure in a way that the system designers/users may improve the performance of the CBR system.

In this paper we presented a hierarchical competence model approach, that extends the existing competence model allowing a finer analysis of the case base structure, particularly for CBR systems that perform *synthetic tasks*. Using this hierarchical approach we have proposed a new visualization method for case base competence based on the solution qualities. This method allows us not only to draw 'competence islands' in an 'unsolved ocean', but rather to draw the complete surfaces. The mapping obtained using the proposed method provides valuable information about the way a CBR system behaves in different parts of the problem space. Moreover, some typical competence surfaces have been identified and described.

We wish to add a measure that indicates the faithfulness of the two-dimensional mapping. This is indispensable, since a rigorous reduction in data dimensionality inherently comes with distortion. Especially if more detailed information can be provided about the fidelity/distortion at various regions in the map, this may facilitate the interpretation of the visualized data.

The visualization method has been used for analyzing the case base of the *TempoExpress* system, a CBR system for applying expressivity-aware tempo transformations to recordings of musical performances. Although currently competence maps were only shown as 'snapshot' images, we believe that the approach is very suitable for an interactive case base visualisation tool, where the user can for example zoom in on certain competence areas, or view the effect of raising/lowering the solution-quality threshold on the average case characteristics for a particular competence group.

We plan to use visualization technique presented here in the *T-Air* system, a case-based reasoning application developed for aiding engineers in the design of gas treatment plants [13].

Acknowledgments

This research has been supported by the Spanish project TIC 2003-07776-C2-02 "CBR-ProMusic: Content-based Music Processing using CBR", EU-FEDER funds, and a FPI fellowship.

References

1. Smyth, B., Keane, M.T.: Remenbering to forget: A competence-preserving case delection policy for case-based reasoning systems. In: Proceedings of IJCAI-95. (1995) 377–382
2. Wilson, D.C., Leake, D.B.: Maintaining case-based reasoners: Dimensions and directions. Computational Intelligence **17** (2001) 196–213
3. Reinartz, T., Iglezakis, I., Roth-Berghofer, T.: Review and restore for case-based maintenance. Computational Intelligence **17** (2001) 214–234
4. Smyth, B., McKenna, E.: Competence models and the maintenance problem. Computational Intelligence **17** (2001) 235–249
5. Wybo, J.L., Geffraye, F., Russeil, A.: PROFIL: a decision support tool for metallic sections design using a cbr approach. In Veloso, M., Aamodt, A., eds.: Proceedings of the First International Conference on Case-Based Reasoning (ICCBR-95). Number 1010 in Lecture Notes in Artificial Intelligence. Springer-Verlag (1995) 33–42
6. Smyth, B., Mullins, M., McKenna, E.: Picture perfect - visualization techniques for case-based reasoning. In Horn, W., ed.: ECAI 2000. 14th European Conference on Artificial Intelligence, IOS Press (2000) 65–69
7. Plaza, E., Arcos, J.L.: Constructive adaptation. In Craw, S., Preece, A., eds.: Advances in Case-Based Reasoning. Number 2416 in Lecture Notes in Artificial Intelligence. Springer-Verlag (2002) 306–320
8. Grachten, M., Arcos, J.L., de Mántaras, R.L.: Evolutionary optimization of music performance annotation. In: CMMR 2004. Lecture Notes in Computer Science, Springer (2004)
9. Narmour, E.: The Analysis and cognition of basic melodic structures : the implication-realization model. University of Chicago Press (1990)
10. Lerdahl, F., Jackendoff, R.: An overview of hierarchical structure in music. In Schwanaver, S.M., Levitt, D.A., eds.: Machine Models of Music. The MIT Press (1993) 289–312 Reproduced from Music Perception.
11. Arcos, J.L., Grachten, M., de Mántaras, R.L.: Extracting performer's behaviors to annotate cases in a CBR system for musical tempo transformations. In Ashley, K.D., Bridge, D.G., eds.: Proceedings of the Fifth International Conference on Case-Based Reasoning (ICCBR-03). Number 2689 in Lecture Notes in Artificial Intelligence. Springer-Verlag (2003) 20–34

12. Grachten, M., Arcos, J.L.: Using the Implication/Realization Model for Measuring Melodic Similarity. In: Proceedings of the 16th European Conference on Artificial Intelligence, ECAI 2004, IOS Press (2004)
13. Arcos, J.L.: T-air: A case-based reasoning system for designing chemical absorption plants. In Aha, D.W., Watson, I., eds.: Case-Based Reasoning Research and Development. Number 2080 in Lecture Notes in Artificial Intelligence. Springer-Verlag (2001) 576–588