Domesticating Bead:
Adapting an Information Visualization System to a Financial Institution

Dominique Brodbeck, Matthew Chalmers, Aran Lunzer, Pamela Cotture

Ubilab, UBS, Bahnhofstr. 45, 8021 Zürich, Switzerland
{dominique.brodbeck,matthew.chalmers}@ubs.com

Abstract

The Bead visualization system employs a fast algorithm for laying out high-dimensional data in a low-dimensional space, and a number of features added to 3D visualizations to improve imageability. Here we describe recent work on both aspects of the system, in particular a generalization of the data types laid out and the implementation of imageability features in a 2D visualization tool. The variety of data analyzed in a financial institution such as UBS, and the ubiquity of spreadsheets as a medium for analysis, led us to extend our layout tools to handle data in a generic spreadsheet format. We describe the metrics of similarity used for this data type, and give examples of layouts of sets of records of financial trades. Conservatism and scepticism with regard to 3D visualization, along with the lack of functionality of widely available 3D web browsers, led to the development of a 2D visualization tool with refinements of a number of our imageability features.

1 Introduction

Bead is a visualization system which has evolved over a number of years in both the layout and presentation of 'maps' of multidimensional data. Layouts are built by means of physically-based force models [5], and we have developed techniques intended to enhance map imageability i.e. legibility and the model of how past searches, selections etc. relate to each other within the layout [6].

In looking for opportunities to apply our work within the bank, we saw several situations where our tools could help. The monthly meeting of the foreign exchange trading group centered on a long succession of scatterplots and bar charts on overhead projector slides, comparing each of many attributes or dimensions against each other. This hampered the business goal: gaining an overview of the month's trading activity and setting guidelines for the next month's trading. The manager of this group expressed his dissatisfaction about not getting a feeling for the overall relationships in the data and a general lack of connection between detail and context. More recently we have been working with the manager of a new financial product, responsible for its adaptation to patterns of customer activity and the marketing by different parts of the bank. Despite having available a database of accounts and their properties, she finds the boolean query tools too limiting when trying to interactively explore and interpret her data. Given these observations, we began to reassess Bead and how we could adapt and extend it to fit into the bank environment and to help in these situations.

Up to now the representation of Bead layouts has been realized using a 3D landscape metaphor. One major issue in information design is the problem of occlusion and clutter which obscure useful detail and structure [20]. We have addressed this problem by developing various static and dynamic imageability features, such as selectively 'popping up' detail based on stochastic sampling of the field of view. Another area of our interest has been the concept of computational wear as introduced by Hill et al [10]. They point out that in analogy to physical wear, usage information is a by-product of normal user activity and requires no extra effort to collect. When fed back in useful ways into the interface it may highlight aspects of phenomena which are important for understanding and task performance but would not otherwise be available to users.

In the Bead system users express their interests and activities either explicitly (search, selection, etc.) or implicitly (position, movement, field of view, etc.) as in the spatial model of interaction of Fahlén et al [7]. By logging these activities and thus making them persistent we can later use them in the representation to enrich information display. Possible applications of making patterns of use evident through this historical or potential information [21] include the dynamic adaptation of the imageability features and the support of collaborative sharing of information. This latter topic is at the heart of collaborative filtering, for example the Tapestry system [8], where ratings of Usenet articles and mail messages were pooled to form recommendations shared amongst users. More generally, our interest in usage data is part of the trend towards having human activity central to the design of information systems, as has been discussed in theoretical terms in [19].

Trying to transfer this technology into the productive parts of our corporation had various implications most notably the shift of the representation of layouts from a 3D virtual environment to a more conventional 2D map display. 2D maps have the advantage of being simpler to navigate and that users are less likely to get lost or disoriented. People in the business and financial world are generally familiar with 2D graphical representations like scatterplots, etc. and the acceptance of such a tool will thus be greater.

Our experience has shown that while professionals in a financial institution acknowledge the potential of new technologies they are under considerable pressure with their everyday work and can not afford to take excessive risks and experiment with unproven technology [4].

With this in mind we developed a suite of tools which satisfies the requirements outlined above and still allows us to refine and advance our novel techniques. These tools make use of a mixture of readily available shrink-wrapped applications like Excel [17] and Spotfire [12] and standard Web technologies like Java and VRML to produce the best combination of extensibility, platform independence, familiarity and ease of use.

In the following sections we describe the implementation of these tools and the design criteria employed.

2 The Layout Algorithm

There are many ways to visualize multivariate data [22]. Among them is the dimensional reduction of the data to a low-dimensional space which can then be visualized using 2D- or 3D-graphics. The dimensional reduction can be done with different techniques such as principal components analysis or multi-dimensional scaling and has been applied for example to collections of text documents [23]. Another technique is the self-organizing map [14] which is an unsupervised learning method to reduce multi-dimensional data to 2D feature maps and has been applied to various data types, including textual data [15].

As described in [5] our layout algorithm uses multi-dimensional scaling techniques for dimensional reduction of multivariate data. The algorithm is based on a spring model where all the data points are mutually connected by springs whose rest distances are proportional to the respective similarity of the data points in high-dimensional space. The algorithm starts out with a random arrangement of the data points in low-dimensional space. Then the system is set free and left to relax with the effect that distant data points which are similar in high-dimensional space are pulled together and close but dissimilar ones are pushed apart by the spring forces. After a number of iterations the system typically stabilizes, resulting in a layout of the data in low-dimensional space where strongly correlated dimensions are blended together, and relationships and clustering of the data in high-dimensional space are preserved.

2.1 Data Types

Our past implementations of the layout algorithm have used specialized data types such as textual documents and time series. In order to make this tool easily applicable to a wider range of problems we shifted our focus to a very general data type which is commonly used and exported by spreadsheets and similar applications: a list of comma separated values (CSV). Each line or record in the list represents a data point described by a set of attributes. There are two types of attribute: numerical/continuous (floating point or integer values like interest rates, prices, etc.) and categorical/discrete attributes (alphanumeric strings like currency, customer name, etc.).

For this data type we can define a generalized similarity metric which is parameterisable and thus lends itself to easy customization and experimentation. The similarity metric or desired distance d_{ij} between two points P_i and P_j is defined in two steps.

To reduce the effect of outliers and take account of the different numerical ranges, the values of all numerical attributes are normalized to two standard deviations of their means. The differences of these normalized numerical values p_{in} (where p_{in} is the value of the n-th numerical attribute of data point P_i and n iterates over all the numerical attributes) are then summed.

$$d'_{ij} = \sum_n w_n (p_{in} - p_{jn})^2$$

In a second step we look at the values of the categorical attributes p_{ic}, where c iterates over all the categorical attributes:

$$d_{ij} = d'_{ij} \prod_c f(p_{ic}, p_{jc}, w_c)$$

$$\text{and } f(a, b, w) = \begin{cases} 0.75 + (1 - w)\,0.25 & \text{for } a = b \\ 1 & \text{otherwise.} \end{cases}$$

The weights $w_n, w_c \in [0 \ldots 1]$ are used to adjust the influence of individual attributes. This assumes that there is always at least one continuous attribute and thus d'_{ij} is non-zero. In the adverse case, d'_{ij} is set to a positive constant value.

In brief, we add all the differences between the values of the numerical attributes and then reduce this distance by 25% for each of the categorical attributes whose values are identical. The adjustable weights for each of the attributes are the parameters of the similarity metric and allow the use of specific domain knowledge to tune the metric.

The multiplicative contribution of the categorical attributes has experimentally been shown to be a good choice for the financial data sets under our investigation.

For data sets with a large number of categorical attributes however, this method could lead to an overemphasis being placed on the categorical attributes. Therefore an additive approach (e.g. the distance is increased by 1 if the values of a categorical attribute are not equal) to mix them with the continuous attributes might be more reasonable in this situation.

2.2 Example

A typical example of financial information is fixed income trading data. Each trade which is executed is recorded as a data point with a mixture of numerical and categorical attributes such as time stamp, currency, yield, amount, customer id, days to maturity, etc. Applying our algorithm with the generalized similarity metric to this data leads to a layout as depicted in Figure 1. Data points are colored according to currency. At this point we would like to acknowledge the problem of using color to effectively distinguish more than a few different values. We have made no effort so far to address this issue but are aware of work being done in this area [9].

Among overall trends for yield, roughly increasing from top to bottom, and amount, roughly decreasing from left to right, we observe the formation of a few dominant clusters. Next to these trends, their primary shape and position are determined by the currency attribute, and their fine structure by a combination of the remaining attributes such as customer id or customer segment, which are correlated with currency in different ways. Positioned around these main clusters we find scattered data points which were placed outside of the dominant clusters even though they sometimes share many of their properties. Further analysis turns out that there are several different reasons for outliers in this particular data set.

The lonely dark green data point at the very bottom of the display for instance is an ordinary Swiss Franc bond whose attributes are comparatively similar to the other Swiss Franc trades in the dark green main cluster, except for its yield of 35%, which is clearly an error and is the reason for its being pushed to the bottom of the map. Since at least part of this data is keyed in manually there is always a potential for data entry errors. These errors promptly become visible in the layouts which can thus become a valuable tool for quickly spotting and separating them from real anomalies. Another reason for outliers are data inconsistencies such as products for which there is no yield and so this attribute is assigned an arbitrary value. This is evidenced by the group of data points in the lower right corner of the display in Figure 1. These are derivative products (futures, etc.) which don't quite fit the data model which was designed for ordinary bond trades.

Data quality and consistency is an important issue and we believe that the combination of these layout techniques together with exploratory visualization tools, such as the visualization tool described in the next section of this paper, are well suited to attack this problem. They allow users to quickly spot anomalies and relationships in their data and help with the interpretation of these features by making the context in the overall layout explicit.

2.3 Implementation

Our first implementation of the layout algorithm which worked with the CSV data type and used this general metric, was done inside Microsoft Excel using the Visual Basic language. The advantage of this approach is the tight coupling between the data source, the layout engine and the analysis tools. On the downside however, Excel has limitations in terms of interactive graphics and user interface capabilities which are needed to enable exploratory data analysis through visualization.

The second implementation of the layout algorithm takes a more general modular approach. It is written using the Java language and thus has the immediate advantage of being platform independent. This is very useful in making these techniques spread to different areas inside the bank. Through an object-oriented architecture based on design patterns we can conveniently divide the problem into three flexible modules which are able to support parallel object hierarchies and thus handle the CSV and arbitrary other data types: the similarity metric or distance function, the container for the data objects and the layout engine.

3 Visualization of Layouts

In [6], the primary tool for visualization of Bead layouts was a shared 3D virtual environment. The focus of this work was adding imageability features to the visualization, in an effort to make the visualization more easily explored, navigated and remembered. Clusters, usage discs, pop-up titles and topics offered useful detail and abstractions over the visualized data but avoided excessive image clutter.

We began to present and discuss our work with colleagues inside the bank, with a view to transferring some of our visualization technology to them. We thus hoped to gain feedback for ourselves while aiding their analyses. We first translated some of our models to VRML, and tried to regenerate the interaction facilities which had been the focal point of our earlier work. Unfortunately, while we could construct static 3D scenes that appeared similar to our earlier VR models, the browsers and APIs available did not yet offer sufficient functionality and performance to support useful dynamic imageability features.

At the same time we grew increasingly sceptical about the use and value of three-dimensional structures in information visualization. We think that 2D structures are more familiar to most people because, with the possible exception of jet pilots in a dog fight, most of our everyday information and navigation tasks are two-dimensional in nature – even though they are set in a three-dimensional world. We therefore prefer 2D information structures viewed in

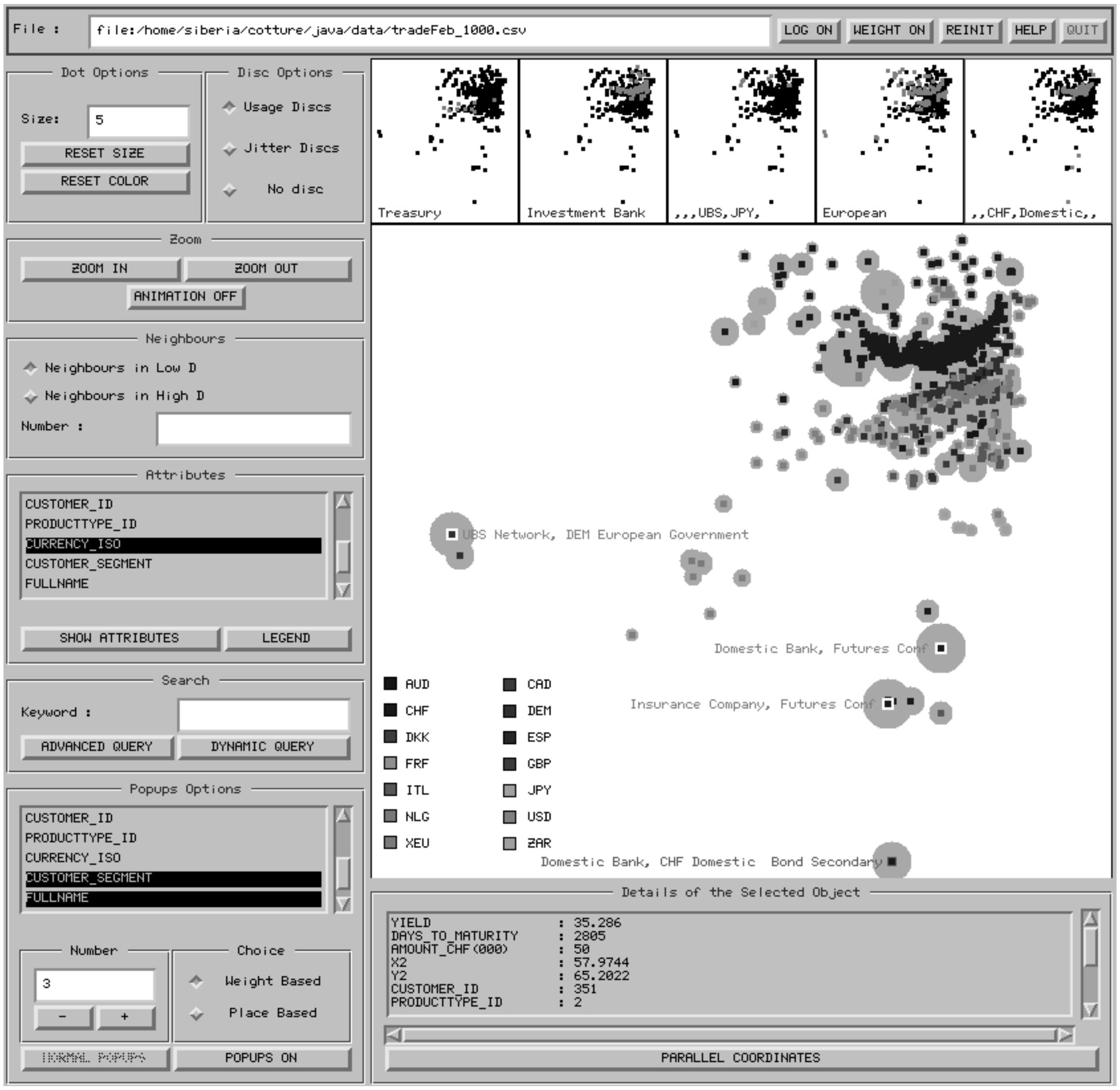

Figure 1. An example of the visualization tool's display. The main canvas shows a layout of fixed income trading data, while the control panel to the left offers facilities for, from top to bottom: controlling the size of individual objects in the display, zooming in and out (although mouse buttons can also be used for this), display of neighbors, object coloration based on data attributes, queries, and pop-up controls. At the bottom of the display is a panel used to show details of an object selected with the mouse and a button to open the parallel coordinates window. At the top of the display is a series of sequential world-in-miniature maps representing the history of searches.

either 2D (e.g. scatter plot) or 3D (e.g. information landscape) over 3D structures (e.g. point clouds) as the latter typically invite clutter, occlusion and the lost-in-cyberspace syndrome, and we thus decided to concentrate on 2D visualizations.

We had already been using a commercial scatterplot-based visualizer to show 2D visualizations to people in the bank. This tool was Spotfire, formerly known as IVEE [1]. Spotfire offers powerful interaction facilities based on sliders and dynamic queries, offering direct coupling to the display.

We noticed, however, that Spotfire had imageability problems. Spotfire offers an option of only showing detailed labels or names of objects when there are a small number of objects (e.g. 20 or less) visible on the screen. If all labels are shown for large numbers of objects then the display is usually illegible. On the other hand, Spotfire is intended to be particularly useful for larger numbers of objects. Its policy on detail makes such visualizations either relatively sparse and under-detailed or cluttered with too much information. Also comparing all the different attributes of two points is not well supported as users have to first click on one data point to get its full list of details, remember all these values, click on the second data point and then mentally compare this list to the previous set of values.

A general characteristic of multi-dimensional scaling techniques is that due to the nature of the layout process, the axes of the resulting two-dimensional space have no inherent meaning. Patterns and relationships in a layout have to be interpreted in view of the global high-dimensional similarities and do not allow direct quantitative statements about data attributes as in conventional scatter plots.

This may lead to confusion with novice users. We tried to avoid this problem by externalizing the layout process and showing an animation of the multi-dimensional scaling algorithm as an introduction, so people could see how a map was iteratively being built. By doing this and by consistently referring to the layouts as 'maps' and not showing any axes at all we could moderate this issue. Indeed, of the roughly 30 people we showed our tool to so far, we only received a couple of questions about what the axis labels are, and were able to resolve the confusion quite easily. These experiences however show the need for visualization tools and techniques which are specifically geared towards interpretation of such layouts and is one of the reasons why we found off-the-shelf applications to be inadequate.

We saw our imageability features as a way to resolve some of these problems. Since Spotfire was a closed system at the time of our evaluation and could not be extended, we decided to develop a 2D visualizer which would offer similar basic data manipulation facilities to Spotfire, allow experimentation with imageability features and other techniques to assist interpretation of layouts and, by being written in Java, be easy to run on a variety of platforms. An example visualization is shown in Figure 1, and the following subsections describe details of this tool.

3.1 Pop-ups

A button on the main control panel of the tool toggles the display of pop-up detail. As the user zooms and pans around the display, a small number of objects currently within the field of view are randomly chosen to be highlighted and have a number of attributes written out beside them on the main display. Every second, a new object is chosen from the set of visible objects and popped up in this way. Simultaneously, the oldest pop-up is removed. The particular attributes to be displayed are chosen from a list on the main control panel. The number of pop-ups simultaneously visible can be chosen also. We do not make an effort to ensure that pop-ups don't overlap because this happens infrequently enough that it has not been found to be a problem.

With our earlier 3D pop-ups, a random pixel was chosen and the nearest object chosen for display. This favored closer objects because perspective made them be drawn with a greater number of pixels. In the 2D case all objects are the same distance and apparent size, and we decided to allow pop-up sampling either to be random or to be weighted by relative usage frequency. These frequencies are represented graphically by the usage discs described in the following subsection. By integrating usage discs and pop-ups, as shown in Figure 1, we favor more active areas of the data set, a feature found to be useful for novice users [13].

We have experimented with two techniques for weight-based sampling. The first approach is much the same as with our 3D pop-ups. A pixel is chosen at random, the nearest disc found, and the associated object is chosen for the next pop-up. This tends to distribute pop-ups over the display, favoring more isolated or peripheral objects. The second approach uses the order of the array of objects alone. We total the weights for all objects, choose a random number x between zero and this total, and then go through the objects in array index order. If x is less than the current object's weight then we choose it for the next pop-up. Otherwise we subtract the weight from x, and go on to the next object. This technique tends to concentrate pop-ups in areas of high layout density.

The previously mentioned lack of a perspective field of view in the 2D display has as a consequence that it is not obvious which implicit information should be used to steer the pop-up of detail. While the above two approaches try to address this problem, experience with users has shown that the process is too random and that a more active use of pop-ups should produce better results. This is evidenced for example by the fact that the integration of pop-ups with search hits resulting from a query – as described in section 3.3 – was well received by users. Another idea is to base the display of detail on the location of the mouse pointer. This could then be used to indicate general areas in which more detail is desired. In our future work we will collect more user feedback on these various approaches to find the right mixture of implicit and explicit steering of the stochastic pop-up process.

3.2 Usage Data

As with our earlier work, user activity is logged and accumulated to form relative frequencies of use for each object. We have extended our previous approach [6], combining different activities by assigning them different weights. The intention is to give more weight to those activities which have involved more attention. With each such activity, relative use frequencies and disc radii are adjusted, and discs redrawn accordingly. Currently the activities and weights used are as follows:

- selection: 1.0
- search hit: 1.0 / n
- neighbor: 1.0 / n
- pop-up: 0.5 / n
- view: 0.01

The x / n division tries to take account of the size of hit sets, assuming that attention per object is less when there are many search hits or neighbors [18]. Selection with a mouse is given most weight. When this happens full details for all its data attributes are written out in a panel at the bottom of the display. Word or attribute searches, described below, are next in the weighting order, before neighbor searches. Once an object has been selected with the mouse its closest neighbors – either in 2D space or the original high-D space – can be highlighted by using another section of the control panel. When an object is popped up for the first time, a little weight is given to it. Finally, each time we zoom in, all objects visible in the new view have a small weight added.

3.3 Searches

We have two search facilities. On the main panel is a simple text field which triggers a search through all categorical attributes for a string. In our earlier work we noticed that displaying the results of such searches can lead to image clutter. When search hits are close to each other, and detail on all hits is shown simultaneously, then the names or titles of objects tend to overwrite and occlude each other. We decided to integrate searches with the pop-up process so as to avoid this problem.

When a search is initiated using the control panel, each member of the set of matching objects is highlighted. Existing pop-ups are removed and the set of matching objects is used as the population used for sampling new pop-ups. Sampling continues on the set of visible matching objects until either 30 seconds have passed, another search is initiated, or a button has been pressed which triggers a return to normal pop-ups.

An additional panel for advanced queries allows one to limit searches to individual attributes or perform compound searches with different queries for chosen attributes. It also allows selection of the colors used in showing matching and non-matching objects. Particular colors can be chosen, or the original object coloration based on data attributes can be used.

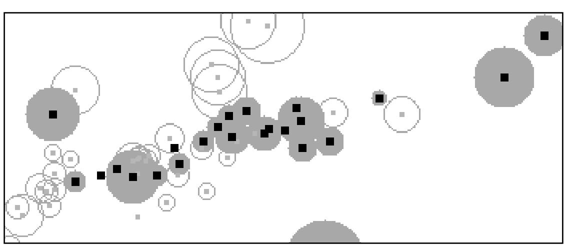

Figure 2. By using 'ghosting', objects which do not match a query can be shown along with matches. By diminishing object size and making usage discs hollow, non-matches continue to serve a contextual role in display of search results.

The history of string searches is made visible by representing them as a series of world-in-miniature displays. For each search a small version of the map with the search results highlighted in green, is shown at the top of the main display area (Figure 1). Successive searches result in the previous maps being pushed to the left while the current search result is inserted on the right side of the row of maps. The history of searches is thus preserved and made visible, allowing users to visually combine them and express more complex questions.

The second search facility is controlled through a separate panel which allows specification of search criteria for each attribute. Specification is done using a set of range sliders, allowing dynamic queries of the map. One slider is set up for each attribute in the CSV file, so that ranges of selected values are AND'd together to form a compound query. Non-matching objects, as shown in Figure 2, have their squares shrunken and usage discs filled with background white to produce a 'ghosting' effect.

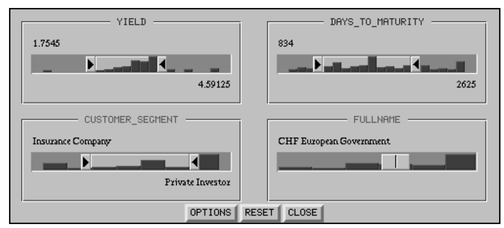

Figure 3. Specification of search criteria using range sliders. The slider widgets are combined with a display of histograms showing the distribution of values for the respective attributes. For categorical attributes there is a choice of range sliders (as shown for CUSTOMER_SEGMENT) or steppers (FULLNAME).

The slider widgets are combined with a display of the histograms showing the distribution of values for each of the attributes (Figure 3). This combination offers additional visual feedback and aids with the formulation of the query and the interpretation of its results.

3.4 Layout Quality

Building a layout is an iterative and non-linear process which tries to lay out multivariate data into a two-dimensional space while preserving the high-dimensional relationships between the data points. The optimization process may not be able to reflect the high-dimensional configuration of similarities accurately in two dimensions. This may result, for example, in some objects' positions not being stable but rather oscillating between different locations.

There are two features which help users assess the quality of a layout, 'jitter discs' and neighbor display.

The layout shown in the visualization tool represents a static snapshot taken after a certain number of iterations met the termination criteria. One way to visualize the dynamic nature of the optimization process is to animate it [2]. For large data sets however this method has the disadvantage of being either slow, if done in real-time, or requiring a lot of space if all the positions have to be stored for each iteration. In order to capture the dynamics present at the final stage of the layout process we record a reference position for all the data points at a time t_s - n , where t_s is the time when the snapshot is taken, and then accumulate the distances between these reference positions and the positions of the data points during the successive n iterations. We then compute the maximum and average separation of each data point from its reference position and use these values to characterize the dynamics that governed the final stages of the layout process. These values are visualized as 'jitter discs' and thus allow users to visually judge the stability of the layout (Figure 4). If the majority of objects have large discs and circles, it means that the layout is far from stability and some more iterations are necessary. If only few objects have large discs then these can be interpreted as data points for which no single low-dimensional configuration exists. The 'jitter discs' can be shown alternatively with the usage discs and are switched on and off in a section at the top of the control panel.

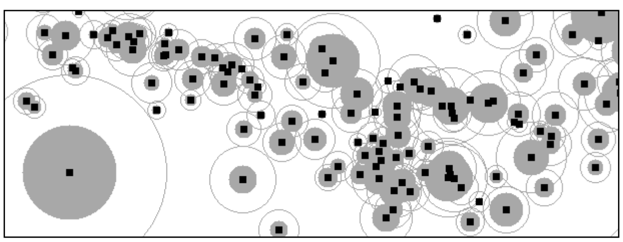

Figure 4. 'Jitter discs' are a measure of the dynamics present at the final stage of the optimization process and help with judging the quality of a layout. The discs represent the average and the circles the maximum separation of data points from their reference positions recorded at an earlier iteration of the layout process.

Another section of the control panel controls the display of the neighbors of a selected object. Once an object has been selected with the mouse, we can highlight a specified number of its neighbors in the layout i.e. the closest objects in the map. We can also show the closest objects according to the high-dimensional similarity metric that was used to generate the map. In this way we can obtain a feel for the correspondence between these two types of neighbor. This has been useful in assessing our layout algorithms, but it also gives users a cue as to the local consistency of objects' attributes within a localized area of the layout.

3.5 Comparisons

While the imageability features introduced so far help with legibility of the information displays and allow the investigation of details in the context of the overall structures, they don't support comparisons among details. The importance of this functionality is increased by the fact that as opposed to standard scatter plots, the axes of the layouts don't have an explicit meaning and thus the positions of individual objects don't allow quantitative interpretations.

We chose the method of parallel coordinates [11] in combination with brushing and linking [3] to support this functionality. With this technique n-dimensional data points are visualized as polygonal lines across a set of parallel axes and can directly be compared quantitatively. Users can select points in the layout and assign different colors in the parallel coordinate display. This allows for example the comparison of outliers against data points in a reference cluster and thus help interpretation of which particular attributes caused a point to be an outlier in the layout. The parallel coordinate display is linked with the layout display and selected points in either of the panels are highlighted in the other to maintain a coherent overall view.

4 Conclusion and Future Work

We have outlined our approach for adapting an experimental information visualization system to the needs of the business environment in a financial institution while still being able to include novel techniques and research ideas during the process. Utilizing standard technology to ensure platform independence and acceptance, the main focus of this work has been the generalization of our layout algorithm and the adaptation of imageability features from a 3D VR environment to a 2D viewer.

The generalization of the layout algorithm allows investigation of varied data sets with no additional development effort and its modular implementation enables easy experimentation and adaptation to different problems. Our future work will take advantage of this flexibility by extending the layout system to manage dynamic data and by investigating the integration of usage data into the layout process.

Another area which needs more attention is the characteristics of the generalized similarity metric, in particular how aspects like weights, the ratio of discrete versus continuous attributes, the number of dimensions, the number of data points, etc. affect layouts, and how users can be supported in experimenting with these parameters and tuning the metric [16]. It will only be possible to satisfactorily answer these questions however if we can gain access to problems along with the specific domain knowledge needed to judge the results.

On the visualization side we have shifted our focus from 3D landscape to 2D map representations. The imageability features of our earlier 3D work are all concerned with user awareness and legibility of detail. In adapting these features to the 2D case we have explored various advances and refinements such as the integration of searches and queries with the stochastic pop-up process, and widening of the range of user activity factors affecting usage logs, and an acknowledgment of their commonness. We will continue to look at usage data and to develop visualization features which specifically support the interpretation of layouts.

The original collaboration with the group from inside the bank which sparked off this work has been discontinued because of their section going through a major restructuring period. We therefore did not reach a conclusion on whether our tools and techniques solve their particular business problem. The intermediate results and experiences however were encouraging enough to set directions and to seek contacts with other groups. We are currently engaged in various collaborations which will allow us to continue receiving feedback about the significance of our work.

In summary we are trying to shape our work in a way to combine its relevance to conservative corporate users with the novelty of research in the lab. This may require the use of different technology but the problems remain the same – imageability, expression of queries, navigation, exploration and making data meaningful. We see our work as becoming decreasingly centered on experimenting with technology and more concerned with issues of perception, meaning and activity.

Acknowledgments

We would like to thank the student interns Patrick Pfister and Jonas Kurth for their contributions during their stay at Ubilab. We are also thankful for feedback received from our colleagues in the bank Susan Farris, Miro Tafra and Sassan Zaker. Special thanks to Dirk Riehle and Markus Stadler for their helpful comments and suggestions.

References

1. Ahlberg C., Wistrand E. IVEE: An Information Visualization & Exploration Environment. Proc. *IEEE Information Visualization ‘95*, Atlanta, Georgia USA, Oct. 30-31, 1995, pp. 66-73.
2. Bentley C., Ward M. Animating Multidimensional Scaling to Visualize N-Dimensional Data Sets. *Proc. IEEE Information Visualization ‘96,* San Francisco, Oct. 28-29, 1996, pp. 72-73.
3. Buja A., McDonald J.A., Michalak J., Stuetzle W. Interactive Data Visualization using Focusing and Linking. *Proc. IEEE Visualization ‘91*, San Diego, October 22-25, 1991, pp. 156-163.
4. Chalmers M. Pearls, Swine and Sows’ Ears: Interface Research inside a Multinational Bank. *Proc. 14th British National Conference on Databases*, Edinburgh, July 1996, published as: *Advances in Databases*, R. Morrison & J. Kennedy (eds.), Springer LNCS 1094, pp. 222-229, 1996.
5. Chalmers M. A Linear Iteration Time Layout Algorithm for Visualising High-Dimensional Data. *Proc IEEE Visualization ‘96*, San Francisco, Oct. 27-Nov. 1, 1996, pp. 127-132.
6. Chalmers M., Ingram R., Pfranger C. Adding Imageability Features to Information Displays. *Proc. ACM UIST ‘96*, Seattle, Washington USA, Nov. 6-8, 1996, pp. 33-39.
7. Fahlén L., Brown C., Ståhl O., Carlsson C. A Space Based Model for User Interaction in Shared Synthetic Environments. *Proc. INTERCHI ‘93*, Amsterdam, April 1993, pp. 43-48.
8. Goldberg D., Nichols D., Oki B.M., Terry D. Using Collaborative Filtering to Weave an Information Tapestry. *Comm. ACM 35*, 12 (Dec. 1992), pp. 61-70.
9. Healey C. Choosing Effective Colours for Data Visualization. *Proc. IEEE Visualization ‘96*, San Francisco, October 27 - November 1, 1996, pp. 263-270.
10. Hill W., Hollan J., Wroblewski D., McCandless T. Edit Wear and Read Wear. *Proc. ACM CHI ‘92*, Monterey, May 1992, pp. 3-9.
11. Inselberg A., Dimsdale B. Parallel Coordinates: A Tool for Visualizing Multidimensional Geometry. *Proc. IEEE Visualization ‘90*, 1990, pp. 361-378.
12. IVEE Development AB. *http://www.ivee.com*
13. Kleiboemer A.J., Lazear M.B., Pedersen J.O. Tailoring a Retrieval System for Naive Users. *Proc. 5th Symposium for Document Analysis and Information Retrieval (SDAIR96)*, Las Vegas, April 1996, pp. 209-216.
14. Kohonen T. Self-Organizing Maps. *Springer Series in Information Sciences*, Vol. 30, 1996.
15. Lin X., Soergel D., Marchionini G. A Self-Organizing Semantic Map for Information Retrieval. *Proc. SIGIR ‘91*, published as a special issue of *SIGIR Forum*, ACM, October 1991, pp. 262-269.
16. Lunzer A. Reconnaissance Support for Juggling Multiple Processing Options. *Proc UIST ‘94*, Marina del Rey CA, November 2-4, 1994, pp. 27-28.
17. Microsoft Corporation. *http://www.microsoft.com*
18. Mariani J., Prinz W. From Multi-User to Shared Object Systems: Awareness about Co-Workers in Cooperation Support Object Databases. *GI-Tagung: Informatik-Wirtschaft-Gesellschaft*, H. Reichel (ed.), Springer, 1993, pp. 476-481.
19. Nardi B. (ed.) Context & Consciousness: Activity Theory and Human-Computer Interaction, *MIT Press*, 1996.
20. Tufte E. Envisioning Information. *Graphics Press*, 1990.
21. Tweedie L. Characterizing Interactive Externalizations, *Proc. ACM CHI ‘97*, Atlanta, April 1997, pp. 375-382.
22. Ward M. XmdvTool: Integrating Multiple Methods for Visualizing Multivariate Data. *Proc. Visualization ‘94*, Washington D.C., October 17-21, 1994, pp. 326-331.
23. Wise J., Thomas J., Pennock K., Lantrip D., Pottier M., Schur A., Crow V. Visualizing the Non-Visual: Spatial Analysis and Interaction with Information From Text Documents. *Proc. IEEE Information Visualization ‘95*, Atlanta, October 30-31, 1995, pp. 51-58.