Presented at CMV2003, International Conference on Coordinated & Multiple Views in Exploratory Visualization, 15 July 2003, London UK

Design Study: Using Multiple Coordinated Views to Analyze Geo-referenced High-dimensional Datasets

Dominique Brodbeck
Macrofocus GmbH
dominique.brodbeck@macrofocus.com

Luc Girardin
Macrofocus GmbH
luc.girardin@macrofocus.com

Abstract

We developed an interactive visualization system that supports analysis and exploration of a large number of indicators that characterize the attractivity of cities in Switzerland. The application is available as a companion to a paper-based publication. This system is embedded in a conceptual framework of best practices in information visualization that we developed over the course of various past projects. It consists of several coordinated views that are tightly integrated, and that successfully reveal the data in its complexity. We present the components of this system, describe design decisions on how to integrate them into an application, and provide details on innovative refinements of various standard information visualization techniques.

1. Introduction

Every year, the business newspaper l'Agefi conducts a study about the attractiveness of Swiss cities [1]. The aim is to establish a qualitative ranking of the 110 Swiss cities with more than 10,000 inhabitants on the basis of 19 distinct criteria. The criteria are taken from the major domains (economic, social, and environmental aspects) of the concept of sustainable development, defined during the United Nations Conference on Environment and Development, which took place in Rio in 1992. This concept emphasizes the important role of the local communities to translate the concept of sustainable development into concrete measures.

The ranking aims at being a barometer of their general health condition, without being able to take into account all the elements influencing the functioning of cities. The selected criteria are the following:

- economic criteria: unemployment, construction, taxes, financial situation, debt per capita, tourism, public investments;
- social criteria: working population, youth population, health, culture and sport, social security, secondary education, public transport;
- environmental criteria: energy, spring water, environmental protection, green spaces, air quality;
- plus a number of additional characteristic attributes (surface area, population), as well as the total ranking score over the past four years.

While the three domains have not been weighted, the collected data have been normalized to a value per inhabitant, in order to keep the proportionality between cities of significantly different dimensions and structural conditions.

Up to now, the results of the study were presented in the newspaper in the form of a simple top ten ranking, accompanied by editorial interpretations and stories, plus a busy double-page print-out of a big table containing all the numerical results. This way of presenting large amounts of data is surprisingly common in the publishing industry, and can also be encountered in domains such as finance.

We convinced the editors that this is not a very effective approach to publishing such data. As a remedy, we created an interactive visualization application in the form of a companion to the study, which can be ordered on CD-ROM by the readers of the newspaper.

Over the past few years, we created various (research and commercial) interactive visualization systems [2, 3, 4] and developed a conceptual framework of best practices based on the experience gained. We used the opportunity that the above study presents, and created an application that follows these rules and showcases our approach.

2. Conceptual framework

From the development of interactive visualization applications during the last few years, a set of rules emerged that seem to be important to build systems that deliver on the promises of information visualization. These

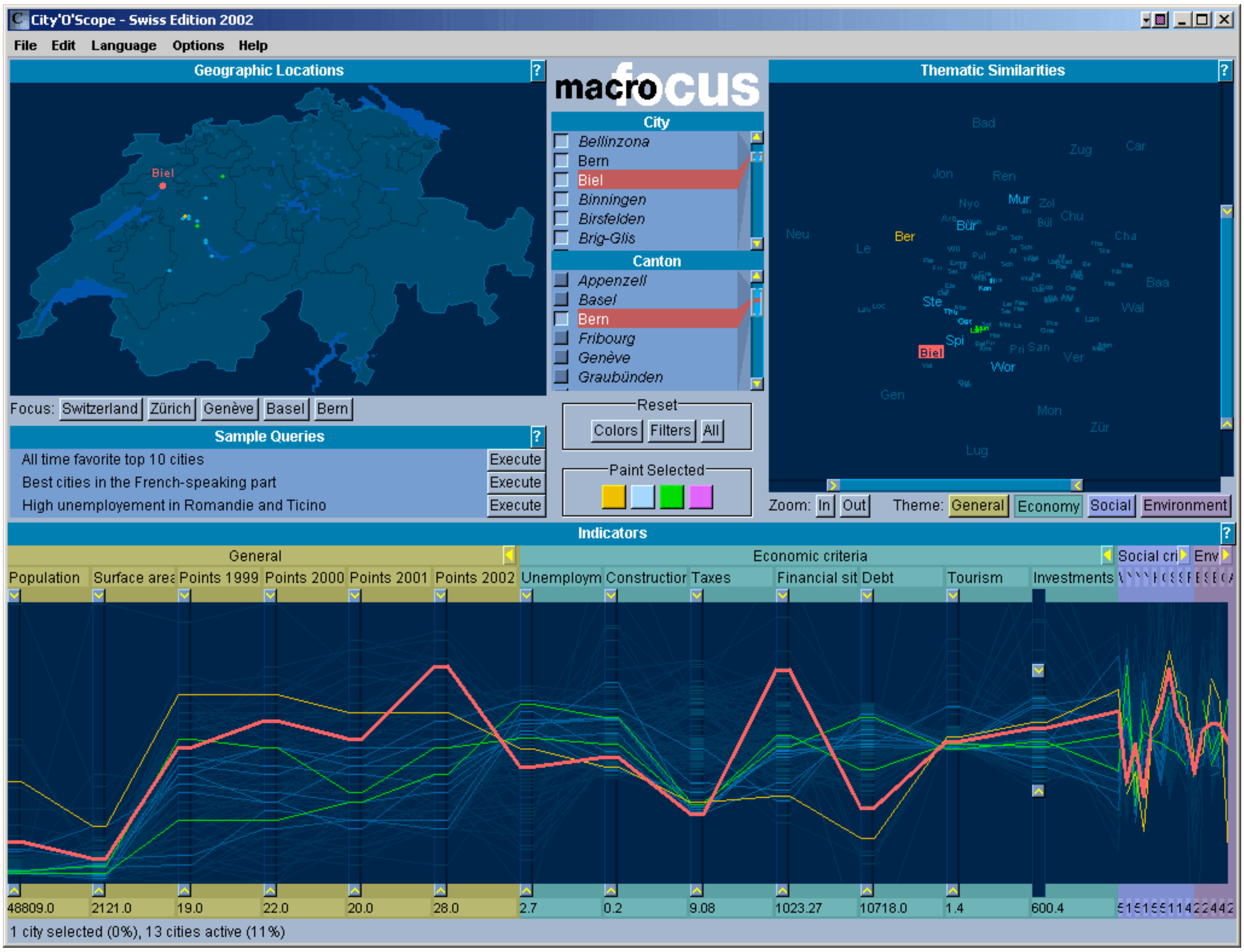

Figure 1: City'O'Scope and its multiple coordinated views: geographic map (upper left), lists (upper center), similarity map (upper right), and the parallel coordinates view (bottom).

rules form the basis of a conceptual framework that we use as a foundation for building new applications. Some of the main rules are:

- integrated systems: the system should provide all the essential tools and views in a single integrated frame to preserve spatial continuity;
- high interactivity: the system should provide immediate feedback for all actions to preserve temporal continuity and to encourage exploration;
- different views: the system should provide different views onto the same data to emphasize different aspects and perspectives;
- tightly linked views: the views should be tightly coupled so that changes in one view are reflected in the others;
- information design: data should be shown with clarity and precision and the overall look and feel should create a pleasurable user experience.

These rules are in accordance with recent research on coordinated multiple views [5]. Our successful commercial implementations confirm the benefits outlined in [6]: improved user performance, discovery of unforeseen relationships, and integration by interaction in addition to integration by visual design.

3. City'O'Scope

The interactive visualization system that we developed to explore and analyze the data characterizing the cities - called City'O'Scope - follows the principles outlined in our conceptual framework. It is composed of several linked views that present the multiple facets of the data. In the following subsections, we will briefly describe each of these views and the combination of information visualization techniques that they use. We will rationalize the design decisions that we took as well as outline the lessons learned. The system has been fully implemented with the Java development platform.

3.1 Geographic map

The cities under study are geographically distributed in a rather inhomogeneous way. The information density in certain areas of the geographic map is much higher than in others. We therefore chose a distortion-oriented presentation technique for the geographic map viewer [7]. The distortion is implemented as a Cartesian fisheye view [8], but uses a central constant-magnification focus region as seen in bi-focal displays [9] or the perspective wall [10]. The transfer function is shown in Figure 2.

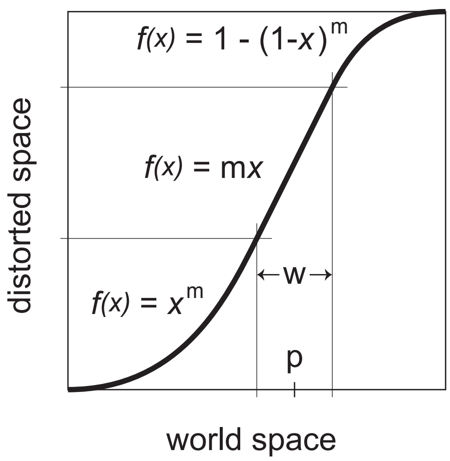

Figure 2: Transfer function of fisheye view with central constant magnification region. Distortion is controlled by three parameters: focus position (p), focus width (w) and magnification factor (m).

This combination has the advantage of having a non-distorted focus region with a visually smooth transition into the peripheral highly distorted regions. Implementation is straightforward with basic mathematics and well defined parameters (focus position, width and magnification).

The fisheye view allows preservation of context while focusing, and is in this case not disorienting because of the familiar shape of geographic features. For quick navigation, the map viewer is equipped with a number of predefined foci that set the appropriate zooming parameters for the most important geographic regions (Figure 3).

Figure 3: A fisheye view of the world focused on Europe. Highlighted objects in the periphery remain visible.

3.2 Similarity map

Spatialized views are useful metaphors because concepts about space are easily accessible to human cognition [11]. We exploit this by mapping the objects in our database onto a plane, based on their similarity. This similarity map helps to gain an overview of the global relationships between objects: cities are placed on a map so that similar cities are located close to each other and dissimilar ones far apart. Similarity is defined by taking into account all of the relevant attributes of a city for a particular theme.

We provide three different similarity maps based on the different types of criteria characterizing the cities (economic, social, environment), and one for the general attributes. The maps are created using a multidimensional scaling technique based on a forced-based layout algorithm [12]. The switching from one map to another is smoothly animated to help understand their differences and avoid the problem of change blindness [13, 14].

Labelling dense information displays is an important problem [15]. For the similarity maps, we use the first three letters of the name of a city as a short form. Since place names are usually discriminative at the beginning with a range of similar endings (at least for German) this is appropriate for our application. To display the labels, we use an algorithm that varies label size according to the local density of the objects to be labelled. Local density is approximated by the minimum separation between neighboring objects, using an elliptical instead of a circular Euclidean metric, to take into account that labels are wider than tall. The full name of an object is displayed when hovering over it with the mouse pointer (Figure 4).

Navigation is supported by semantic zooming. When the “Zoom In“ button is pressed, the window is zoomed

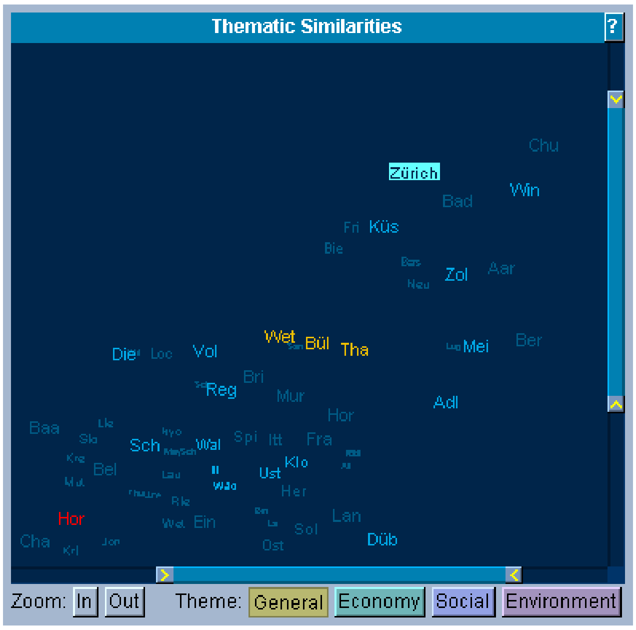

Figure 4: The similarity map showing each city with its abbreviation. Hovering the mouse over the city pops up the full name.

automatically to the smallest area that shows only non-filtered cities. Pressing the button again zooms to the smallest area that shows only selected cities. The same mechanism is applied in the reverse direction.

3.3 Collapsible parallel coordinates and range sliders

The parallel coordinates [16] view shows one axis for each attribute or criterion that characterizes the cities. Connecting the actual values for one specific city on all the axes leads to a polygonal line that forms a visual representation of the characteristics of this city. Differences between cities can easily be spotted by visually comparing the two polygonal lines representing them.

Choosing appropriate ranges, scales, and groupings is central to creating a readable parallel coordinates plot. We use three different strategies to define the setup of the axes (Figure 1). For attributes that are directly comparable and logically related, we use the same scale and group them together (e.g. ranking 1999, 2000, 2001, 2002). Attributes that are logically related, but whose numerical values are not comparable, are plotted centered around their median value and their axes scaled to a common multiple of the standard deviation, so that their statistical properties become comparable (e.g. economic criteria). This causes extreme outliers to be plotted out of range, but the two lines connecting the data point to the preceding and succeeding axes provide enough visual cues to perceive their position. These attributes are sorted according to importance, or any other order that is imposed naturally by the data. The remaining attributes finally, are simply plotted on an axis bounded by their minimum and maximum values (e.g. population).

The range of an attribute can be specified by moving the handles at the top and bottom of a range slider. The range sliders are embedded within the parallel coordinate plot. Cities whose value for that attribute falls outside of the specified range are filtered and can not be selected anymore. A combination of range sliders can be used to dynamically formulate queries [17] such as: which city has low unemployment AND medium rates for taxes?

Because of the large number of attributes, displaying them all at the same time as classical parallel coordinates, would render this method useless. To tackle this problem, we added the possibility of collapsing groups of attributes into a more compact representation. We first experimented with a two stage approach, where the parallel coordinates are first collapsed into a compact list of range sliders, removing the connecting lines, and in a second step into a single aggregating attribute (see Figure 5).

This proved to be too complicated however for this particular application. The final solution has just one collapsed state, where the range sliders are removed and the coordinate axes simply compressed in the horizontal direction, limited by a certain minimum separation. The result is a compact visual representation of a whole group of attributes (see Figure 6).

Next to the problem of handling a large number of axes, parallel coordinates also have the problem of displaying a large number of objects, represented by the polygonal lines. After drawing a certain number of lines on top of each other, the drawing space becomes saturated and simply looks like one homogeneously colored solid area, hiding the underlying structures. Solutions to this problem that have been suggested are hierarchical approximations, either by using wavelets [18], or in XmdvTool through the use of clustering and partitioning [19]. These methods are designed for large datasets (> 10^4 objects). The problem outlined above already appears well below these numbers however. In contrast to these computationally intensive approaches, we therefore chose to use a density-based representation such as the one introduced in [20]. This approach also has the advantage that objects are represented, and can be manipulated, individually.

Transparency is supported in modern graphics libraries through the concept of an alpha channel, with compositing rules for combining source and destination pixels to achieve blending and transparency effects with graphics and images. The Mondrian system [21] for example makes use of the alpha channel mechanism of Java to alleviate overplotting problems. The general nature of these mecha-

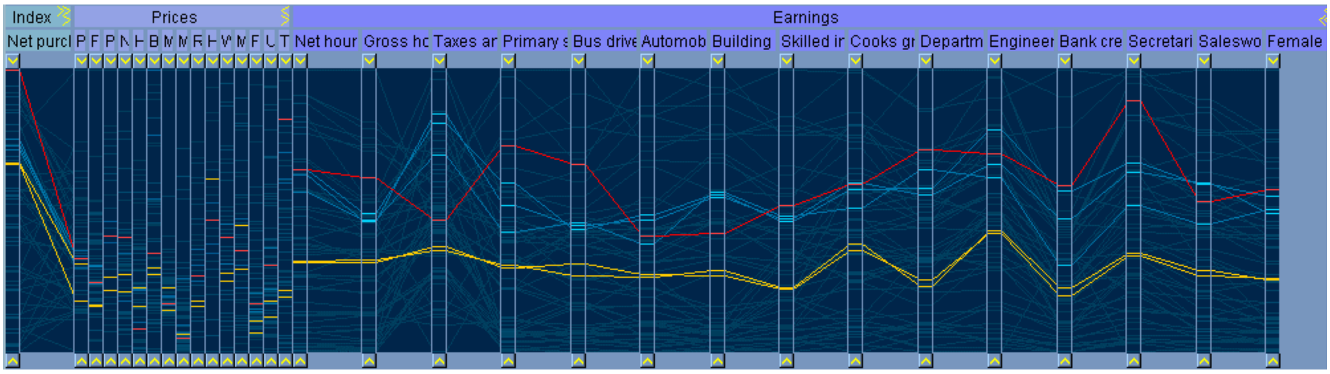

Figure 5: An early version of the parallel coordinates view with three states (from left to right): completely collapsed (Indices), half-collapsed (Prices), and fully expanded (Earnings).

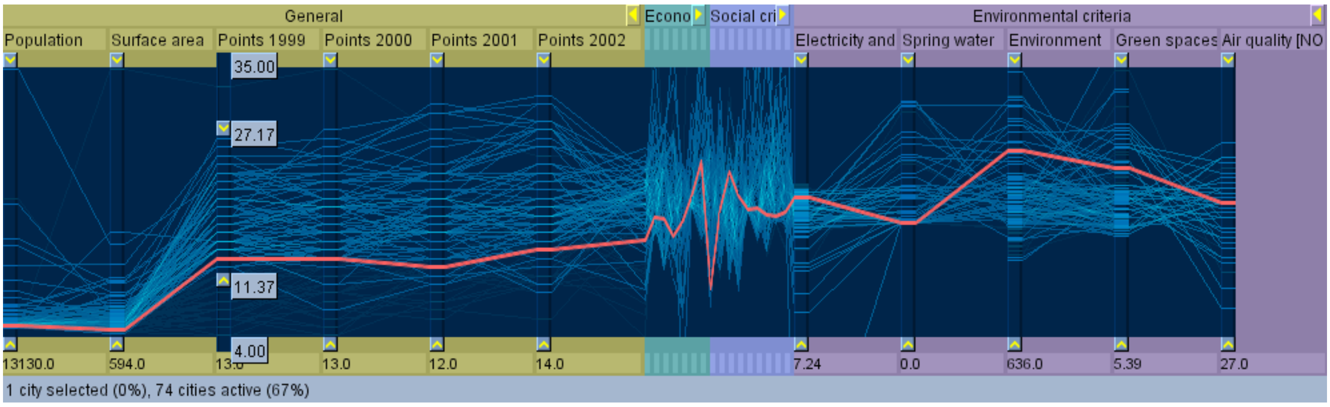

Figure 6: The parallel coordinates view with two groups of attributes collapsed into a compact representation (center). When the mouse pointer is hovering over an axis, its full and filtered ranges are displayed in little pop-up windows.

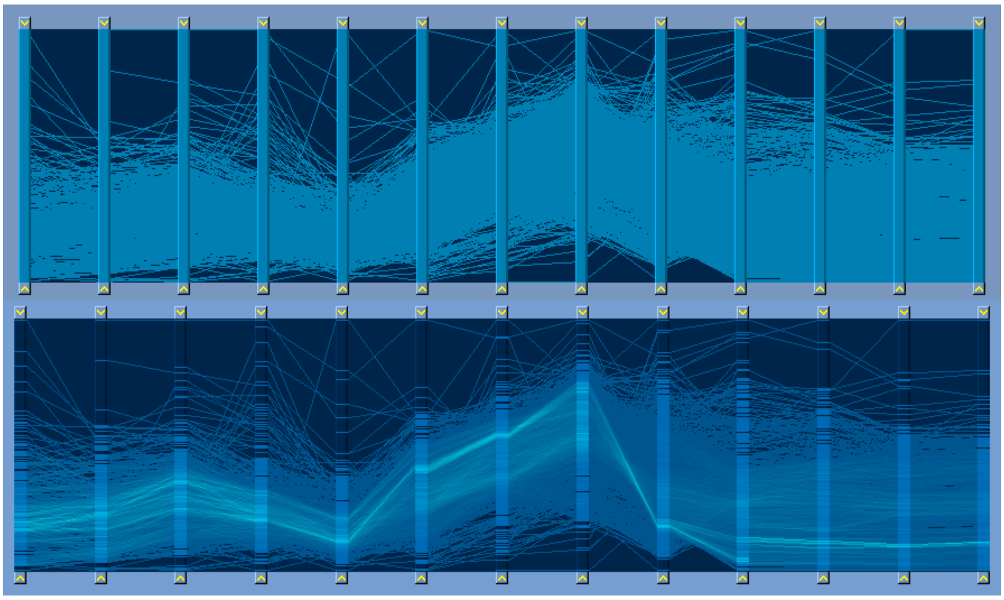

Figure 7: Standard parallel coordinate plot exhibiting saturation from overplotting (top). The density-based representation preserves the underlying structures in the data (bottom).

nisms however, leads to low performance for sufficiently complex graphics, and they require hardware acceleration for real-time use. To create a density-based representation for the particular case of parallel coordinates, we implemented an optimized simple accumulation buffer algorithm, where we record for each pixel how many times it is overdrawn by a line, and then scale its brightness relative to the overall maximum. The accumulated values are then mapped directly to a perceptually linear color map. The result clearly improves the problem of saturation (see Figure 7) and the performance scales to a reasonable number of objects (up to 10^4 in our implementation).

Another problem of parallel coordinates arises from the fact, that objects with similar values for a particular attribute are represented by lines that cross the respective axis at quite different angles, depending on the values of the attributes to the left and right. This makes it difficult to spot clusters of values along a particular axis. To alleviate this problem, we added a short horizontal line as a connector between the entry and exit points of a line on an axis. This improves the readability of the density distribution of values along a particular attribute axis (see Figure 8).

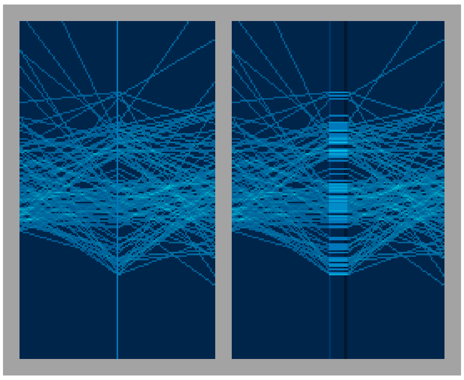

Figure 8: Adding a short horizontal line improves the perception of the distribution of values along a particular attribute axis.

3.4 List views

Realistic multi-dimensional data sets, such as the one under investigation, usually contain a mixture of numerical and categorical attributes. The latter are typically represented as a sequential list of items. As soon as these lists are longer than a few items, they suffer heavily from what we call the keyhole effect, where only a small slice of the data is visible at any one time, and users have to shift their point of view to see other limited slices. As a solution, we used the vertical scroll bar to embed the visible section of the list in its own context. Using the space inside the slider to augment representation of and interaction with long lists of data has been suggested before. In contrast to these approaches, we do not provide additional interaction modes [22], or dynamically reconfigure the contents of the list depending on a query [23]. We maintain the well-known concept of a simple slider, but augment it with a world-in-miniature view in the background. This view shows the whole range of the list, the location of selected items within that range, and the relation of the visible items to the overall space (see Figure 9).

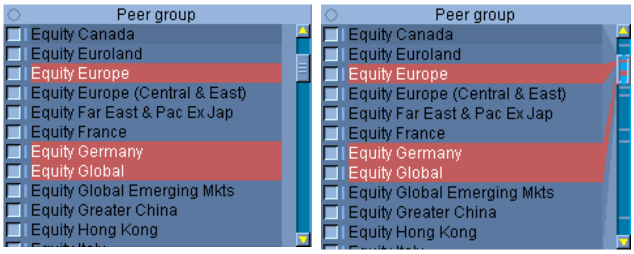

Figure 9: A list with a regular scrollbar (left) and our augmented version with a world-in-miniature view within the slider (right).

Navigation is supported by semantic scrolling. Clicking with the alternate mouse button on the up or down stepper, advances the list to the next selected item. This way, one can easily step through the selected objects. Our list component has the particularity of being bidirectional, i.e. that one or multiple items can be selected in the list and that the selection will be reflected in the other views, but also that selection in another view is reflected in the list.

3.5 Coordination and interaction

When offering multiple views onto the same data to show the different aspects, coordination between the views becomes important. In user studies [24] it has been clearly shown, that coordination is essential, when users need access to details in addition to getting the overview.

A common coordination technique is brushing and linking [25], where users can select objects in one view and the corresponding objects in all the other views are also automatically selected. Over the course of using and building many applications that support this technique, we developed a refined approach. We distinguish three modes of brushing and linking interaction that are coordinated among all the views described in the previous sections:

- **probing**: this mode is used to view more details about an object (e.g. full city names instead of abbreviations) and to get an understanding of the relationships between the different views. Probing is a transient operation. Moving the mouse pointer over an object, highlights that object (e.g. bright blue elements in Figure 3 and Figure 4). As soon as the mouse pointer is moved away, the highlighting disappears. It is used like a flashlight that examines a dark room;

- **selecting**: this mode is used to mark objects that are of short-term interest, in order to further examine or perform operations on them (e.g see averages of attribute values for groups of cities). Clicking on an object selects it and marks it in red (see red elements in Figure 1). If a selected object is filtered (see below), then it becomes deselected;
- **painting**: this mode is used to mark objects that are of long-term interest, in order to use them as references for comparisons (e.g compare a capital with the other cities of a county). Selected objects can be painted by pressing one of the predefined color buttons in the center of the window (e.g. yellow and green elements in Figure 1). Objects remain painted until they are reset explicitly.

The second technique that is coordinated between the views is filtering. Objects that are filtered in one view become inactive, and can not be probed or selected anymore in any of the views. To maintain the overall context of the system, their “ghosts” remain visible, in a way that is appropriate for each view (e.g. grayed out, font italicized).

Four different views with coordinated interaction among them, create a rich system that can communicate many different aspects of the underlying data. To highlight some of the salient insights that can be gained from this data, we provided a number of sample queries (see Figure 1) that capture a particular state of all the views, and that can be replayed by pressing the respective button. Since we wanted to keep this application as simple as possible, we did not provide the possibility of creating new custom snapshots, but this would certainly be desirable for a more general system.

4. Conclusion

Due to the nature of this project, the end-users of the application are not known to us. They simply buy and download the software through an e-commerce channel, or through a mail-in advertisement in the newspaper. Formal requirements and user feedback are therefore not straightforward to collect. Users’ tasks are likely to vary considerably. Design decisions were thus taken based on discussion with the editors of the study, and through our own experience with using a similar application in numerous public demonstrations over time. We will attempt to get more direct user feedback in the future. The challenge will be to measure the effectiveness of the application, when the user population is so diverse and tasks not well defined.

The emerging field on information visualization has produced many innovative techniques that help humans access and understand large amounts of complex information. Taking these techniques out of the research domain and creating effective information visualization applications has remained difficult however. In this paper we outlined the lessons that we learned from creating such applications and presented an example application based on these principles. We described a number of refinements to standard information visualization techniques that help in creating applications for real-world problems.

5. References

[1] L'Agefi (www.agefi.ch), in collaboration with the Valais University of Applied Sciences (Haute Ecole Valaisanne HEV)

[2] Dominique Brodbeck, Matthew Chalmers, Aran Lunzer, Pamela Cotture. *“Domesticating Bead: Adapting an Information Visualization System to a Financial Institution.”* In John Dill, Nahum Gershon (editors). In Proceedings of Information Visualization '97, Phoenix AZ, October 1997, pp. 73-80.

[3] Luc Girardin, Dominique Brodbeck. *“A Visual Approach for Monitoring Logs.”* In Proceedings of the 12th System Administration Conference (LISA '98), Boston, MA, December 1998, pp. 299-308.

[4] Dominique Brodbeck, Luc Girardin. *“Atom'O'Scope: Interactive visualization of a nanotechnology experiment.”* 3rd CODATA Euro-American Workshop on Information Visualization, Presentation and Design, Paris, July 10-11, 2002.

[5] Michelle Q. Wang Baldonado, Allison Woodruff, Allan Kuchinsky. *"Guidelines for using multiple views in information visualization."* In Proc. ACM Advanced Visual Interfaces `00, (May 2000).

[6] Chris North, Ben Shneiderman. *"A taxonomy of multiple window coordinations."* University of Maryland, College Park, Dept. of Comp. Science Tech. Report #CS-TR-3854, (1997).

[7] Y.K. Leung, M.D. Apperley. *“A Review and Taxonomy of Distortion-Oriented Presentation Techniques.“* ACM Transactions of Computer-Human Interaction, Vol 1, No. 2. June 1994, pp. 126-160.

[8] Manojit Sarkar and Marc H. Brown. *“Graphical fisheye views.“* Communications of the ACM, 37, 12, July 1994.

[9] Apperley, M.D., Tzavaras, I. and Spence, R. “*A Bifocal Display Technique for Data Presentation.*” In: Proceedings of Eurographics'82, Conference of the European Association for Computer Graphics., 1982, pp. 27-43.

[10] Mackinlay, J. D., Robertson, G. G. and Card, S. K. “*The Perspective Wall: Detail and context smoothly integrated.*” In Proc. of CHI'91 (New Orleans, LA), ACM Press, 1991, pp. 173-179.

[11] Sara I. Fabrikant. *“Spatialized Browsing in Large Data Archives.“* Transactions in GIS, 4(1), 2000, pp. 65-78.

[12] Matthew Chalmers. *“A Linear Iteration Time Layout Algorithm for Visualising High-Dimensional Data.”* Proc. IEEE Visualization 96, San Francisco, October 1996.

[13] Ronald A. Rensink, J. Kevin O'Regan, James J. Clark. *"To See or Not to See: The Need for Attention to Perceive Changes in Scenes."* Psychological Science, 8, pp. 368-373.

[14] Lucy T. Nowell, Elizabeth G. Hetzler, Ted Tanasse. *"Change Blindness in Information Visualization: A Case Study."* In Proc. IEEE Symposium on Information Visualization 2001 (INFOVIS'01), pp. 15-22.

[15] Jia Li, Catherine Plaisant, Ben Shneiderman. *"Data Object and Label Placement for Information Abundant Visualizations."* In Proceedings, Workshop on New Paradigms in Information Visualization and Manipulation (NPIV'98), HCIL Technical Report No. 98-05 (December 1998).

[16] Alfred Inselberg. *"N-dimensional graphics, part I -- lines and hyperplanes."* Technical Report G320-2711, IBM Los Angeles Scientific Center, Los Angeles, 1981.

[17] Christopher Ahlberg, Ben Shneiderman. *"Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays."* Proc. ACM CHI'94 Conference, (1994), pp. 313-317.

[18] Pak Chung Wong, R. Daniel Bergeron. *"Multiresolution Multidimensional Wavelet Brushing."* In Proceedings of IEEE Visualization '96, October 1996, ACM Press, New York, NY, pp. 141-148.

[19] Ying-Huey Fua, Matthew O. Ward, Elke A. Rundensteiner. *"Hierarchical Parallel Coordinates for Exploration of Large Datasets."* In Proc. of IEEE Visualization '99, October 1999, San Francisco CA, IEEE Computer Society Press, pp. 43-50.

[20] Edward J. Wegman, Qiang Luo. *"High Dimensional Clustering Using Parallel Coordinates and the Grand Tour."* Computing Science and Statistics, Vol. 28, 1997, p. 361-368.

[21] Martin Theus. *"Interactive Data Visualization using Mondrian"*. Journal of Statistical Software, Volume 7, Issue 11, 2002.

[22] Stephen G. Eick. *"Data Visualization Sliders."* In Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'94), ACM Press, November 1994, pp. 119-120.

[23] Toshiyuki Masui. *"LensBar - Visualization for Browsing and Filtering Large Lists of Data."* In Proceedings of IEEE Symposium on Information Visualization (InfoVis'98) (October 1998), pp. 113-120.

[24] Chris North, Ben Shneiderman. *"Snap-together Visualization: Can Users Construct and Operate Coordinated Visualizations?"* Int. J. Human-Computer Studies (2000) 53, pp. 715-739.

[25] R. Becker, W. Cleveland. *"Brushing Scatterplots."* Technometrics, 29 (1987), pp. 127-142.