TECHNISCHE UNIVERSITÄT WIEN
VIENNA UNIVERSITY OF TECHNOLOGY

MASTERARBEIT

A Detailed Comparison of Information Visualization Tools Using a Reference Data Set

Ausgeführt am Institut für
Softwaretechnik und Interaktive Systeme
der Technischen Universität Wien

unter der Betreuung von
Univ.-Prof. Mag. Dr. Silvia Miksch,

eingereicht
an der Technischen Universität Wien,
Fakultät für Informatik

von
Eva Maria Wohlfart
1070 Wien, Kaiserstrasse 92/2/11
Matrikelnummer: 0125128

Wien, November 2007 Eva Maria Wohlfart

1 Abstract

(engl.) Time is a complex dimension, especially when trying to visualize it. In the last ten years a lot of approaches to display and interact with temporal data have been published. They range from linear timeline visualizations to novel ideas employing visual metaphors and even clustering techniques to support the user in exploring large-scale data sets. The diversity of the proposed methods has raised the awareness that a common categorization needs to be defined to efficiently evaluate the usability and interactivity of information visualization tools. Therefore, this work aims at giving a detailed overview of the possibilities and problems of current information visualization tools by applying a recently published categorization. A data set containing air pollution data measured in the years 2002 to 2006 at five different measurement stations in Great Britain is displayed with each of them. To enhance the judgement of the visualization tools, tasks that cover different areas of practical work are defined and carried out. After this practical part the categorization is applied to all of the examined applications. Both the task accomplishment and the use of the categorization are then reflected and occurred problems are described. Possible improvements are pointed out and future research areas are mentioned.

(deut.) Durch die Komplexität der Dimension Zeit ist die Visualisierung temporärer Daten kein einfaches Problem. In den letzten zehn Jahren wurden viele Ansätze, derartige Daten darzustellen und mit ihnen zu interagieren, veröffentlicht. Deren Vielfalt reicht von einfachen linearen Zeitachsen über innovative Ideen, welche visuelle Metaphern zur Darstellung verwenden, bis zu Clustering-Techiken, die den Benutzer bei der Untersuchung besonders umfangreicher Datensätze unterstützen sollen. Die Verschiedenheit der präsentierten Methoden hat das Bewusstsein um den Bedarf einer Kategorisierung, welche einen effizienten Vergleich von Visualisierungsapplikationen bezüglich Benutzerfreundlichkeit und Interaktivität ermöglicht, geschaffen. Daher setzt sich diese Arbeit zum Ziel, durch die Anwendung einer kürzlich publizierten Kategorisierung eine detaillierte Übersicht der Möglichkeiten und Probleme aktueller Visualisierungstools zu präsentieren. Der eingesetzte Datensatz enthält Luftverschmutzungsdaten der Jahre 2002 bis 2006, die an fünf Messstationen in Großbritannien ermittelt wurden. Um die Beurteilung der Applikationen zu erweitern, werden Testaufgaben, die unterschiedliche Bereiche der praktischen Arbeit abdecken, definiert und ausgeführt. Daraufhin wird die Kategorisierung auf alle untersuchten Tools angewendet. Sowohl die Durchführung der Testaufgaben als auch die Verwendung der Kategorisierung werden schließlich reflektiert und aufgetretene Probleme beschrieben. Mögliche Verbesserungen werden aufgezeigt und Bereiche für zukünftige Forschungsarbeit erwähnt.

Table of Contents

2 Introduction

”Time is the most valuable thing a man can spend” [33] - Diogenes put it in a nutshell. Time has always been one of the most important things for mankind, although no one can control the time he/she is given. Because of this fact time is a very complex dimension, not only for those who philosophize about it, but also for the attempt to grasp and visualize it. Many approaches to solve this problem have been published in recent time, creating the need to categorize not only the methods to visualize time-oriented data but also the information visualization tools available on the market. Because of the difficulty of comparing such tools no agreement on a common categorization has been reached. This master’s thesis uses a recently published categorization to perform a detailed comparison of several visualization tools, focusing on the temporal aspects of data. Special emphasis is devoted to the problems and possible improvements of currently available information visualization tools and the evaluation of the employed categorization. During the project, the following research questions should be answered:

Focusing on the visualizations of time-oriented data:

- How are current information visualization systems dealing with time-oriented data?
- How do they support the user in exploring and the data?
- Is there special support for large datasets available?
- Where lie the problems of currently available visualization applications?
- What would be possible improvements?

Focusing on the used categorization:

- Is the categorization applicable and useful?
- Were there any problems using the categorization in practice?
- Are there any recommendations that could enhance the quality or accuracy of the categorization?

3 Related Work

This section reviews literature which is important in the context of the work presented in this thesis. As already mentioned in the previous chapter, a lot of approaches to solve the problem of the visualization of time-oriented data have been published in recent history, but no common categorization to judge the qualities of a visualization has been found yet. Nevertheless, several different points of view both on the visualization of temporal data and on the categorization of such visualizations have been presented, which are of specific interest for this thesis. Especially, the various attempts to find a general taxonomy for information visualizations will be relevant for the evaluation of the used categorization later on.

3.1 Visualizations of Temporal Data

Many approaches for the visualization of time-oriented data have been published in recent years. Due to the complexity of considering many aspects of time, most of them solve only specific problems and conceive the structure of time to be either linear or cyclic. Linear structured visualizations show time either as a continous flow or as the sequence of discrete events. Cyclic approaches often tend to use the metaphor of a circle to express that kind of structuring.

3.1.1 Linear Structured Visualizations

The most common form of time visualization is the timeline together with user interactions. Flow-based approaches follow this idea of time structuring. Other visualizations use metaphor graphics constructed of several parts each representing a different part of the data. Thus, they are capable of conveying further meaning that a simple timeline cannot provide.

Timelines

The approach *Lifelines* visualizes both medical patient records and youth criminal histories of a person [37]. The visualization displays one person and its attributes in an overview. Horizontal regions of the screen represent specific phases of a person's life, with horizontal lines and icons depicting events along a time axis. Color and shape of the icons are used to show their importance and to indicate relationships between events. The offered interaction mechanisms include

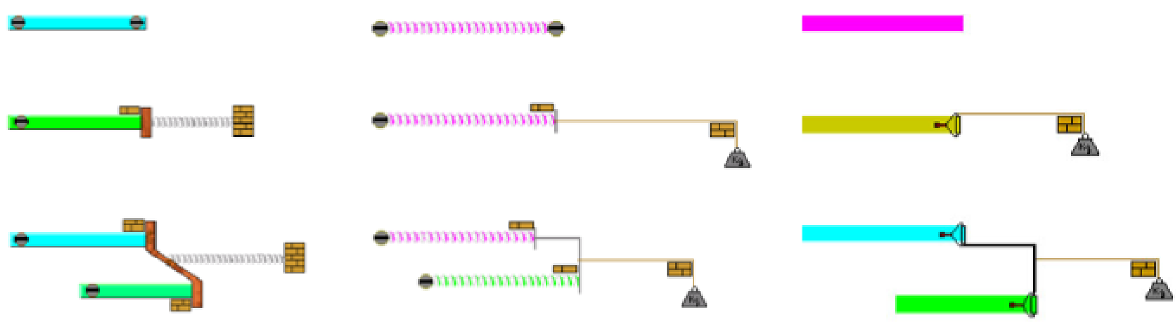


Figure 3.1: **Three different interval representations.** *Elastic bands, springs and paint strips were chosen to graphically depict temporal intervals. Fixed ends are either represented as screws or as plain strips. Movable ends are connected to a weight or a moving mass system. (Chittaro et al. [23]).*

interactive zooming, filtering and details on demand. Additionally, the granularity of the time axis can be changed to the units year, month or day.

One possibility to enhance the basic timeline visualization is the usage of distortion. The *Perspective Wall* is a well-known technique that employs this and therefore makes better use of the display space [38]. The visualization consists of three walls that signify the timeline with events depicted as colored bars on them. The length of the bars symbolizes the length of that event or interval. The left and right walls are distorted and build the contextual information to the focused wall in the middle. User selection of an event on one of the walls leads to a smooth transition that brings the chosen event in focus.

Starting from real-world objects, Chittaro and Combi propose three alternative visual metaphors for timelines: elastic bands, springs and paint strips [23]. In the first metaphor, *elastic bands*, intervals are seen as strips on the time axis. Herein, the familiar bar representation of intervals is kept while introducing new concepts as well. Any end of a strip can be either fixed by screws or attached to a moving mass system. To express interval relationships, e.g., interval A always has to end after interval B, more than one strip can be attached to such a system. The second metaphor replaces the bars by *springs*. Spring ends can be fixed by screws or be connected to a weight by means of a wire. Similar to elastic bands, the possibility of connecting more than one spring to a weight is given. *Paint strips* are either represented plainly without any attached object or with a paint roller associated to any of their ends which is again connected to a weight. Again, a weight can be connected to more than one paint roller simultaneously. Figure 3.1 shows an overview of the three proposed metaphors. User testings suggested a significant preference for paint strips because they allow more direct interpretation of the situation and simpler identification of interval ends.

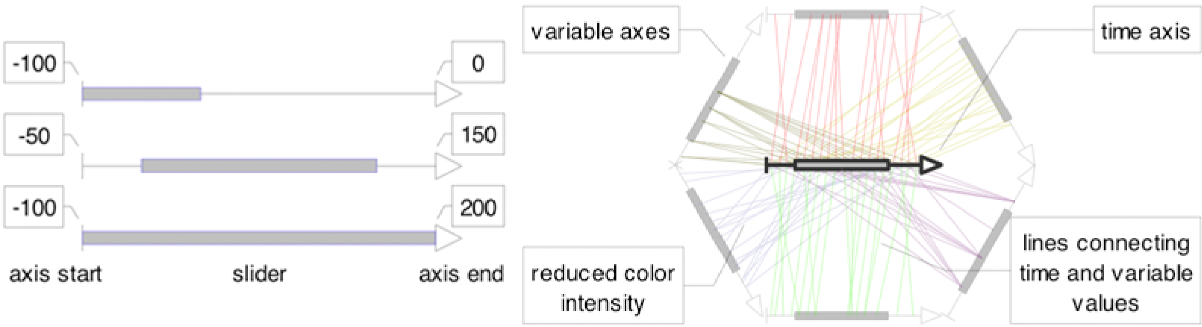


Figure 3.2: **TimeWheel visualization.** *The left-hand side shows the axis concept of the TimeWheel: only the values lying inside the range of a slider placed on that axis are displayed. The slider can be narrowed or widened to change the size of the section of interest. By combining the concept of sliders and axes interactivity is increased. The user is enabled to focus on a section of interest by moving the slider on the axis and choosing the size of the slider. Variable axes that can be rotated are arranged around the centered time axis. (Tominski et al. [40]).*

A completely different placement of variable axes is introduced with the *TimeWheel* technique [40]. Instead of setting up axes in parallel, the time axis is situated in the middle of their visualization and the other variable axes are arranged in a circle around it. Points of the time axis are connected with associated points on the variable axes with lines. The relations between the time values and the values on the surrounding axes are explored most efficiently if they are laid out in parallel. Due to this, the user has the possibility to interactively rotate the ”axes circle” to bring axes of interest into focus. Other axes are faded out to lead the user's attention to the parallel axes. The axes itself use an interesting zooming mechanism: instead of showing all variable values on an axis, only those values lying inside of the range of a slider placed on that axis are shown. Figure 3.2 explains that concept and also shows an example of a timewheel visualization.

Flow-Based Approaches

The Internet offers a relatively new form of interaction: wikis. The key characteristic of those platforms is the opportunity of every reader to become an author. Launched in 2001, Wikipedia is one of those websites. Since then, its number of articles has grown vastly. Even though an archiving system for storing and viewing the changes for every article exists, the examination of version changes still remains cumbersome. A visualization technique called *history flow* [42] has been designed to solve this problem and to improve the analysis of such collaborative systems. The basic idea of this technique is simple: each version of an article is shown as a vertical line, with length proportional to the length of the text. A different color is assigned to every author, and each part of the version line is colored according to its author. When someone

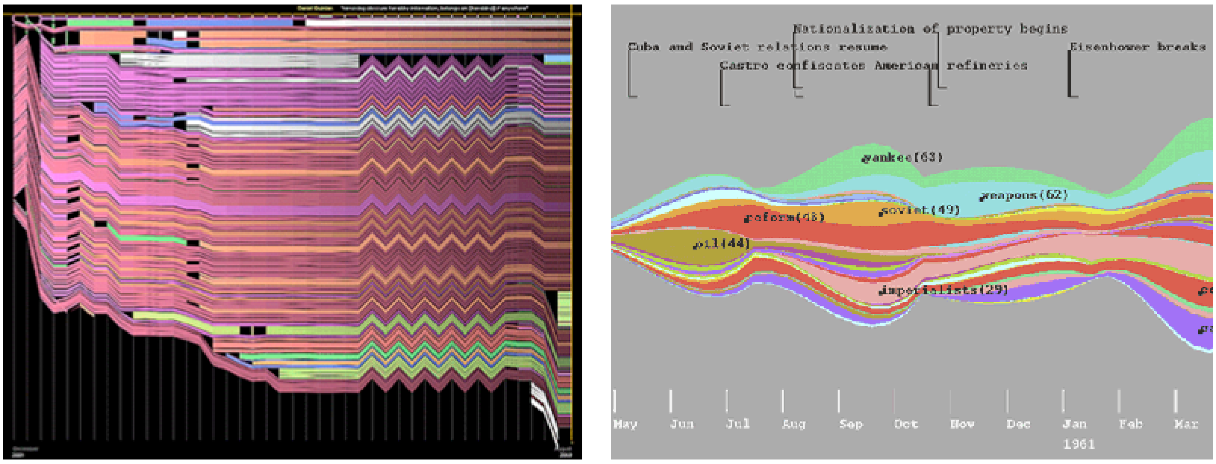


Figure 3.3: **Flow-based time visualizations.** *On the left-hand side, history flow shows editing operations in wikis. The zig-zag-pattern reveals an edit war: one author wanted to add a text passage, while another one continued to delete it. ThemeRiver on the right-hand side visualizes the occurences of themes in newspapers. (Viegas et al. [42]).*

for example deletes a passage and adds a new paragraph, the new version line will be shortened by the length of the deleted text, and the new paragraph will be given the color of its author and inserted at its position, as shown in Figure 3.3. To illustrate the relation of consecutive versions corresponding segments are visually linked.

Another system that handles temporal data as a kind of ”flow” is *ThemeRiver* [28], as its name implies. In contrast to history flow, the visualization is not focused on documents, but on themes. Like seen in Figure 3.3, every theme has an assigned color, and the width of the corresponding current at a point in time in the river represents the strength of the theme. If the occurrence of a theme increases over time, the current widens and vice versa. In addition to the ThemeRiver itself, the prototype visualization contains a time line and labels for theme currents. On top, important events are shown as labels asssociated with a point on the time line. Thus the finding of correlations between the occurence or widening of a theme and subsequent events is simplified. Because of the fact that the underlying data are not continous but discrete the true boundaries of the currents must be interpolated. Therefore, the theme river can only be used as an overview. If the user zooms in farther than the data resolution supports, the visualization loses credibility. A small usability evaluation showed that users find the metaphor of the river easy to understand and more intuitive than a histogram. But they mentioned as well that the tool is less useful for identifying minor trends because narrow currents are hardly visible.

A similar flow is formed by the visualization technique *Wormplots* [41], but starting from a different approach. Based on the well-known scatterplot technique which draws a point for each data value, a scatterplot is drawn for each point in time. Then, point clouds are identified

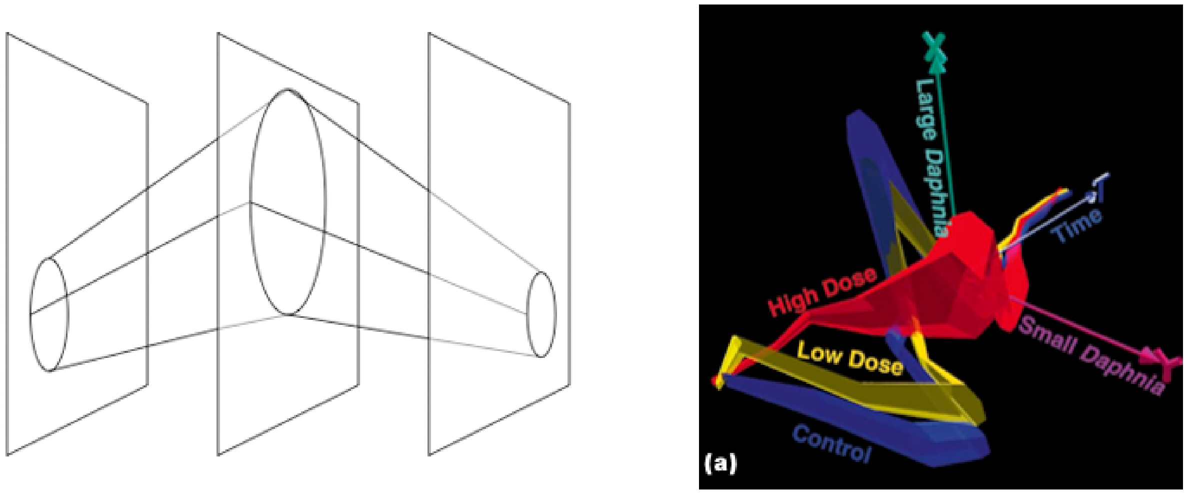


Figure 3.4: **Wormplot visualization.** *The left-hand side shows the principle of connecting corresponding point cloud circles which is followed in the process of the visualization generation. The wormplot visualization on the right-hand side depicts the reaction of aquatic organisms to Jet-A jet fuel. Data was taken from a standardized aquatic microcosm experiment. (Treinish et al. [41]).*

and represented by circles. Corresponding point clouds are traced over time and connected by linear interpolation. Every point cloud is assigned a specific color to simplify differentiation of the clouds. Losing the linkage to the data is avoided by adding a coordinate system with labeled axes to the visualization. Additionally, each ”worm” is associated to an explanatory label drawn in its own color. Therefore, a 3-dimensional view of the data progress is generated which briefly summarizes the data characteristics and helps the user in the process of exploration. Figure 3.4 shows an example of such a wormplot.

Glyph-Based Approaches

Glyph-based approaches show discrete events depicted as glyphs. Glyphs ”refer to graphical objects or symbols that represent data through visual parameters that are either spatial, retinal, or temporal” [24]. A view of time using glyphs is appropriate to visualize software management data, since the versions of software can be seen as discrete snapshots over time. During the management of software projects a huge amount of data is created. Visualizations can help project managers to detect patterns among sets of software artifacts and to find differences between them.

The simplest way of illustrating time-oriented information is a time-series plot - a line chart with time on the x-axis and the value of a variable on the y-axis. The *timewheel glyph* [24] combines several of those charts and rotates them around a circle. Thus, trends can easily be discovered - ”prickly fruits” indicate increasing trends, whereas ”hairy fruits” signify tapering

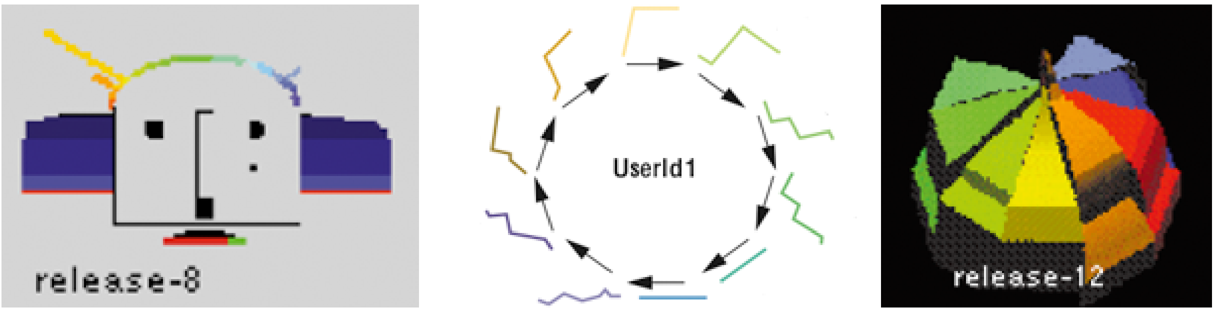


Figure 3.5: **Glyph-based time visualizations.** *From left to right: infobug glyph, timewheel glyph and 3D wheel glyph. The 3D wheel glyph uses a similar metaphor for the visualization as the timewheel glyph, but encodes time in height of the glyph. (Chuah et al. [24]).*

trends. One timewheel-"fruit" is associated with one software release, and a number of releases can be compared by arranging the "fruits" in parallel, compare Figure 3.5.

Similar to the timewheel, the *3D wheel glyph* [24] rotates visualizations of data around a circle, but time is encoded by the height dimension. Each variable is displayed as a slice of a base circle, and the radius of the slice encodes the variable's size. An object with a sharp apex implies an increasing trend, while objects that balloon out indicate tapering trends.

A different approach to visualize software project data is pursued with the *infobug glyph* [24]. Each body part of the infobug - wings, head, tail and body - depicts one important class of software data. For example, the y-axis of the two wings encodes the time and the wings' x-axes represent the number of code lines and the number of errors. Due to the fact that more code lines usually lead to more errors, symmetrical wings are the expected result. Non-symmetrical wings may indicate poor testing of the component or architectural problems of the code.

Patients in intensive care units (ICU) are monitored every second. The created data is combined with manually collected data and stored in a computer-based patient-data management system. Such systems allow a physician to display a complete review of the patient's condition, shown in form of spreadsheets and x-y-plots. This way of visualization is appropriate for extracting a specific value of a parameter at a specific point in time, but not to get a summary of the patient's data or to find the essential changes of a patient's state. A display system called *VIE-VISU* [29] has been developed to ease the fast acquisition of a patient's condition. The system uses a structured metaphor graphic object, namely a glyph, which consists of triangles and rectangles, each symbol's size or color representing a parameter. One object depicts the patient information of one hour and 24 objects are arranged in standard display, showing a whole day.

Approaches Using Other Metaphors

In clinical practice, the medical staff uses standardized health care procedures. Such guidelines contain many variables and conditions to achieve the desired goal. One visualization for such guidelines is *AsbruView* [32]. Three-dimensional running tracks are used as a metaphor for the therapy plans, with the time on the x-axis, depth representing plans on the same level of decomposition and height representing the division of plans into subplans. At the beginning of the track, a "no entrance with exceptions"-sign and a turnpike illustrate preconditions which have to come true to start the plan. If this happens, the patient enters the running track respective plan. A yellow traffic light on top of the plan encodes the suspend-condition which means the plan is halted until the reactivate condition becomes true. Then the traffic light turns green. Red light means abort of the plan. The exact amount of time a plan needs to be carried out is not known in advance, but it always has a minimum and maximum duration, or starting and ending time.

The unusual metaphor of a garden full of flowers is utilized by the *PeopleGarden* [44] visualization. It depicts not only personal histories, but also histories of interactions among users in a virtual environment. Each user is represented by a flower and its messages are rendered as petals of the flower, ordered by the time they were sent. An example of this visualization is shown in Figure 3.6. The flower height stands for the time a user has been at the board. Color is used to differentiate initial posts from replies and to characterize a user's posting behaviour. The arrangement of the flowers as well as the coloring give definite clues of interaction patterns.

3.1.2 Cyclic Structured Visualizations

Temporal data often has an inherent cyclic structure, which can be revealed by the use of visualization techniques. Especially, multivariate data with many values impede the understanding of its inherent structure. Visual representations are able to give better insights and even the possibilities to discover data relationships.

The *circle segment technique* [30] is tailored to the visualization of datasets containing many dimensions. A circle is segmented into the number of dimensions needed to be shown, and the single values of the attributes may be time-ordered. Relationships between the dimensions are easily discovered through closed rings appearing in the circle. Figure 3.7 illustrates the connections between 50 stocks of the Frankfurt Stock Index and also shows that some stocks remained unchanged though all others went through major exchange rate fluctuations.

Carlis and Konstan introduce the use of a planar spiral for the visualization of time-series data [22]. Serial data are explored along the spiral, whereas periodic data are explored along the radii. The simplest case is the visualization of one attribute over time. Like in Figure 3.9, every value refers to a dot according to its size placed on the intersections between laps and spokes of

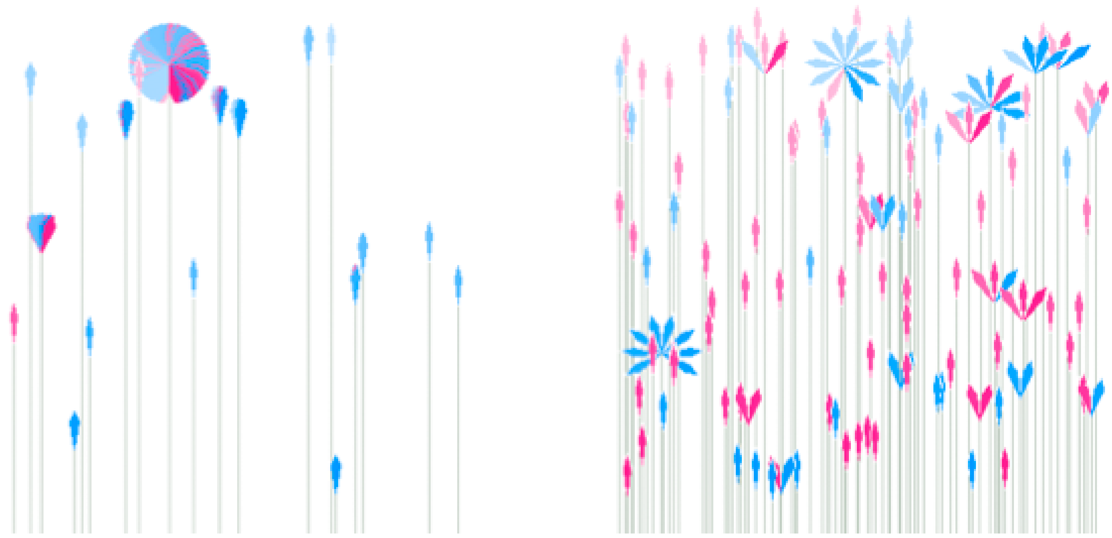


Figure 3.6: **PeopleGarden.** *This visualization uses the metaphor of flowers to depict user behavior in virtual environments. Magenta petals represent initial posts, whereas blue ones visualize replies. The left-hand side shows a group with one dominating voice and a large number of replies. A more democratic group is shown on the right-hand side. (Xiong et al. [44]).*

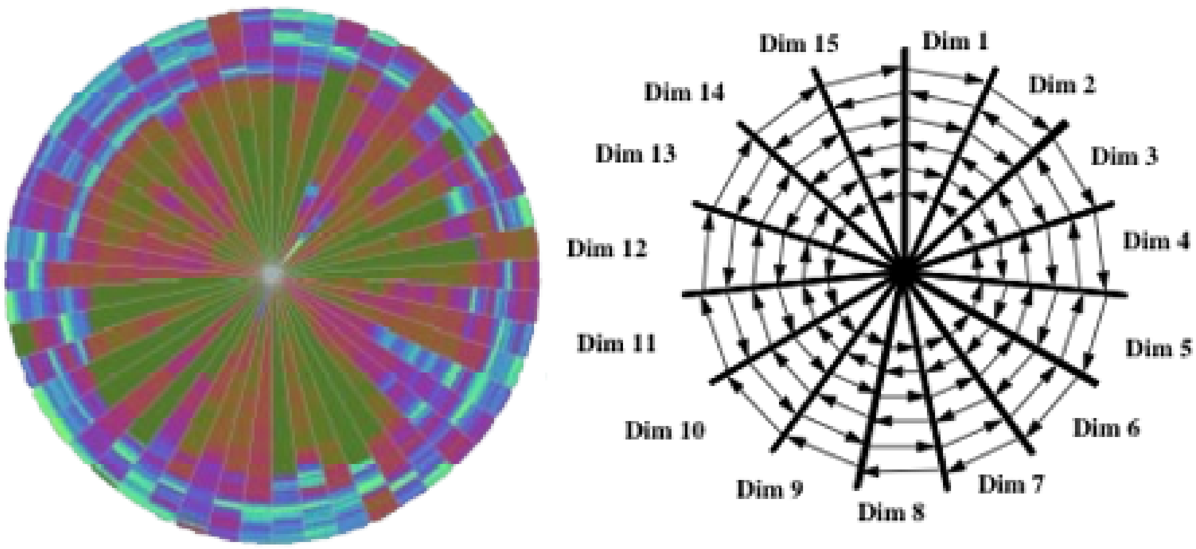


Figure 3.7: **Circle Segment Technique.** *Each segment of the left circle represents one of 50 stocks of the Frankfurt Stock Index. The attribute values are ordered from the inside to the outside of the circle. The right part of the image shows the arrangement of the attributes on the segments of the circle with 15-dimensional data. (Keim [30]).*

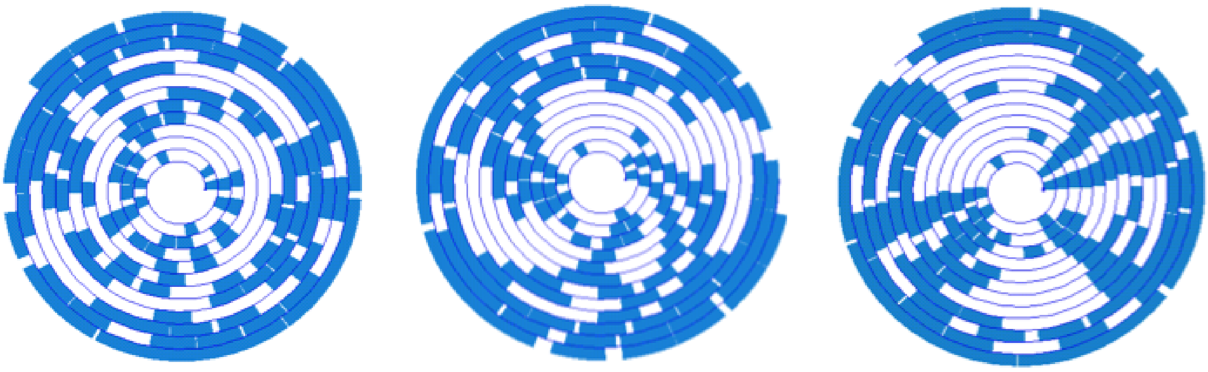


Figure 3.8: **Discovery of Periodicity.** *By tightening the spiral visualization of sound data the periodic character of the data is discovered. From left to right the previously disorganized display turns into straightened stalks. (Carlis et al. [22]).*

the spiral. For the visualization of multivariate data over time an enhancement of the basic spiral is proposed. Instead of placing a dot on the spiral the attribute values appear as spikes similar to a histogram. Each attribute is assigned its own color to ease the distinction. To explore the data characteristics, the user can zoom in, rotate, pan or tilt the spiral and adjust the height scale and width of the "histogram spikes". Interactive spirals enable the user to reveal previously unknown periodicity. By adjusting the lap rate periodic data is discovered when wilted stalks emerge and straighten. Figure 3.8 shows such a process in the visualization of quantized sound data.

3.2 Comparisons of Information Visualization Tools

Considering the evaluation of information visualization tools two main directions can be perceived: papers considering the empirical testing of information visualization tools with subsequent statistical analysis, and more theoretical approaches which aim to find a general taxonomy for visualization systems. These approaches are now presented with special focus on the differences to the categorization applied in this thesis. Even though empirical observations are strictly speaking out of scope of this thesis, they still should not be ignored because they are of importance for the formulation of detailed working tasks to compare the usability of the visualization tools.

3.2.1 Taxonomies and Categorizations

The main concern of Freitas et al. was the definition of criteria for the general evaluation of information visualization techniques [27]. They take three aspects into account: visual representation, interaction mechanisms offered and data usability. Visual representation criteria deal with the density and dimensions of the data displayed, the spatial organization of the individual elements, the codification of information and the transition between consecutive states. Essential interaction mechanisms a visualization should provide include user support for the control of levels of detail, mechanisms for navigating and querying the visualization, e.g., selection, manipulation or searching, and data set reduction techniques, like filtering or clustering of elements. Data usability is associated to three principles: data reliability describes the gap between the gathered data and reality, the original data should not be changed during the visualization process and the data representation should help the users in decision-making.

Pfitzner, Hobbs and Powers take a wider view on information visualization systems in their taxonomic framework [36]. They define five interface design factors: data, task, interactivity, skill level and context. The factor *data* has two dimensions: data type and relational structure. Possible data types are low-level (objects or attributes) or high-level data, often also named meta-data. Relational structure includes five data structures, according to Bertin (1981): linear, circular, ordered tree, un-ordered graph and lattice. Considering the *tasks* a user has to accomplish, they adopt Shneiderman's well-known task by data type taxonomy [39]. According to him, the task domain is divided into seven dimensions:

- Overview: view of the whole data set.
- Zoom: zoom in on individual items of interest.
- Filter: remove unwanted items from the visualization.
- Detail-on-demand: select items or groups and acquiring details when needed.
- Relate: view relationships among items.
- History: user support, e.g., undo/redo, replay and refinement.
- Extract: extract a sub-collection or query criteria.

Interactivity is seen as a continuum between tools that allow direct manipulation and fully automated transitions. This continuum is further broken down into several interaction types, beginning from completely manual interactions, e.g., dragging, over mechanized ways of interacting, e.g., filtering out using sliders to more automated interactions, like using a query language to formulate a query or even completely automatic processes where a program performs undirected to achieve a result. Different *skill* levels require different presentation and interaction techniques. Evaluations have also shown that success for a given user task improves as experience increases. The fifth design factor *context* describes external influences to the user like the user's need or intent, the type of the device used and his/her life experiences. Additionally, the focus is put on the possibilities an information visualization has to depict information: display dimensions, like color, shape, size, orientation and the position of elements, are crucial ways to convey meaning. In addition, the output of a visualization system can be static or dynamic, with animations that use the human visual system's sensitivity to motion being a substantial part. Although not an explicit component of the taxonomy, the importance of grouping items and smooth transitions between visualization states is mentioned.

Amar and Stasko define high-level knowledge tasks that have to be supported by information visualization systems [20]. According to them, these tasks are accomplished to bridge two analytic gaps: the *Rationale Gap* and the *Worldview Gap*. The former stands for the ”gap between perceiving a relationship and actually being able to explain confidence in that relationship and the usefulness of that relationship” [20, p.144]. The latter means the gap between what is shown and what needs to be shown to draw a conclusion for making a decision. Each of the mentioned gaps is associated to three tasks. The tasks to bridge the Rationale Gap are:

- Expose Uncertainty: the uncertainty of data measures and its possible effect should be indicated.

- Concretize Relationships: the system should clearly present what comprises the representation of a relationship.

- Formulate Cause and Effect: possible sources of causation have to be clarified.

The tasks to bridge the Worldview Gap are:

- Determination of Domain Parameters: the user should be supported in gathering knowledge about important domain parameters and their positive or negative connotation.

- Multivariate Explanation: relationships sometimes involve more than three variables or simple transformations. The discovery of such correlations has to be facilitated.

- Confirm Hypotheses: the formulation and verification of user hypotheses should be supported.

By evaluating how well a visualization supports the knowledge tasks they can be used as a framework for a form of heuristic evaluation. Therefore, definite clues for the pragmatic value of a visualization system are provided.

The problem of classifying an information visualization is handled completely different by Wiss and Carr [43]. They construct a classification framework considering three cognitive aspects: attention, abstraction and affordances. Often the user needs to draw *attention* to certain elements of the visualization. The design of the visualization should allow the user to focus with as little cognitive overhead as possible. Visual features, like color, shape or size, help directing the attention. The use of the human capability to register movement is another way to do

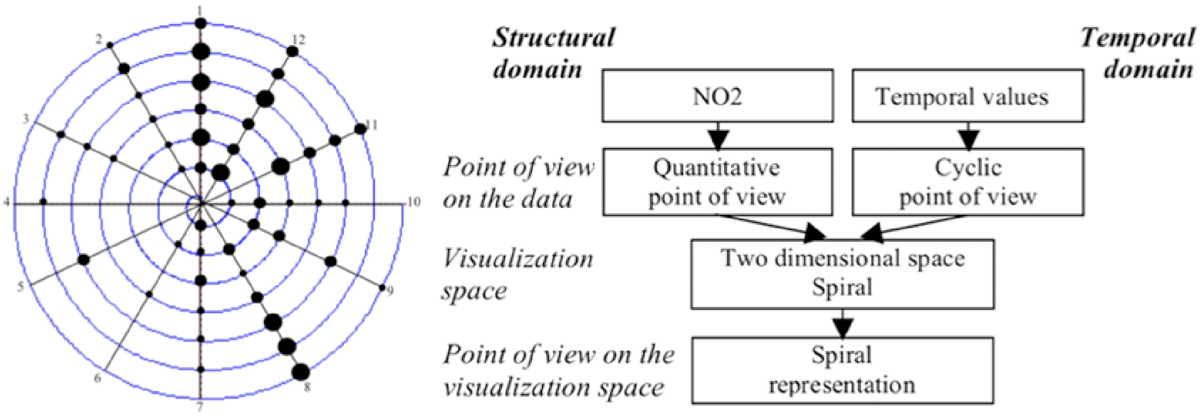


Figure 3.9: **Spiral visualization.** *Every NO2 pollution value is depicted by a dot on the spiral. Rotating the representation of the temporal domain (the spiral) implies a rotation of the structural domain (the dots). Therefore, they share the same visualization space. (Daassi et al. [25]).*

this. Also, employing visual metaphors may enable the user to concentrate on important parts. *Abstraction* can be seen as information hiding to ease the perception of the overall visualization. Filtering, grouping and different levels of detail correspond to this concept. Similarities among objects, such as color, shape or even animation, provide visual cues for abstraction as well, though they do not exactly follow the principle of information hiding. The *affordances* of a visual object are the cues it gives the user to show what can be done with it, e.g., a button affords pushing. Besides general interface components, like buttons or sliders, the growing use of hypermedia has introduced the pointing device symbol to provide affordance too. Experienced users will try to click on every visualization element to find interactions similar to clicking on every image or icon to find a hyperlink. Fifteen information visualization designs were described with respect to this classification framework. Research revealed that attention and abstraction techniques are frequently used in current information visualization designs whereas explicit affordances appear rarely.

One of the rare categorizations dealing with temporal data is the taxonomy by Daassi, Nigay and Fauvet [25]. They refer to Chi's visualization process which follows the visualization from the raw data to the actual view. The process traverses four steps: data, point of view on the data, visualization space and point of view on the visualization space. Adapted for temporal data the four steps are time, point of view on time, time space and point of view on the time space. The step *time* consists of the time values to be visualized. In the second step, *point of view on time*, the representation of the time values is chosen, e.g., which time units are used and whether time is considered linear or cyclic. At the third step, *time space*, a visualization abstraction is defined, e.g., time values observed at unit Hour and expressed in unit system [Hour, Day] may be mapped onto a two-dimensional representation where each unit is mapped to one dimension, or each of the units could be mapped to its own time axis. The fourth step, *point of view on the time space*, contains the final rendering of the temporal domain. Continuing the previous example, each of the two units could now be mapped to a slider to allow coarser or finer examination. Every history contains two parts: gathered time values and mapped values of a given type. The former are called the temporal, the latter the structural domain of the history. The point of contact between the visualization processes of the two domains defines the class the visualization technique belongs to. Figure 3.9 shows a visualization of air pollution data where the fusion happens at the ”visualization space” step.

Chittaro and Combi present seven features for a temporal visualization framework [23]. Instead of observing the visualization process itself, they examine the resulting visualization. They define four basic aspects that have to be considered in visualizing time-dependent data:

- *Time points* are usually depicted as some kind of graphic objects and located with reference to a time axis.
- *Time intervals* tend to be visualized as boxes or lines and, again, their location is referencing a time axis.
- *Temporal relations* are expressed by the relative position among time points and intervals. However, this visualization does not always address the need of precisely describing the relationships among points/intervals, so other metaphors may be used.
- *Logical expressions* connect the represented intervals and time points. In some situations, they might be implicitly related by AND operators, whereas other cases afford the display of more complex conjunctions, e.g., the formulation of database queries.

Further temporal features are dealing with the general structure of the visualization:

- *Indeterminacy* refers to the imprecision often adherent to real-life temporal data. It can lead to uncertainty of the temporal relations associated to the time points and intervals and therefore has to be taken into account in the visualization.
- *Granularity* is related to the different time units temporal information needs to be displayed in. It allows users to explore both an overall and a detail view of the data shown.
- *Temporal views* enable the user to focus on different aspects of temporal information, e.g., temporal extents and temporal relations.

3.2.2 Empirical Analyses of Visualization Tools

Li and North compare two visualization techniques for the exploration of information, namely dynamic query sliders and brushing histograms [34]. A dynamic query slider represents one attribute of the data. By adjusting the slider the range of the attribute can be reduced. Brushing histograms let the user directly select bars in the histogram of an attribute to highlight the corresponding regions in the main visualization. In both cases, the result is shown in real-time in the main visualization window. The information visualization system used is DataMaps, a geographic information visualization tool with census data about the 50 states of the USA. User task areas are single range and multiple ranges (deal with ranges of one attribute), multiple criteria (combine at least two attribute ranges), attribute correlation (discover relationships between the attributes), compare (compare states according to multiple criteria) and evaluate trend (evaluate the trend of a particular state in a global context). 36 persons participated in the test. After the accomplishment of the user tasks a detailed data analysis was done, taking into account user performance time to complete each task, correctness of their answers and user satisfaction ratings.

An evaluation of the effectiveness of visualization types for information retrieval tasks was done by Morse and Lewis [35]. Their the definition of the user tasks followed a user-centered design approach. Visual tasks were chosen from a taxonomy of Zhou and Feiner [45] and reformulated to generalized task statements. These statements where then mapped to specific tasks suitable to the used dataset. Two-term and three-term questions were treated separately and compared. For example, for the task "compare" the two-term question was "Are there more documents that contain ONLY the term Romania or ONLY the term Czechoslovakia?" and the corresponding three-term question was "Are there more documents that contain ONLY the term earthquake or ONLY the term California or ONLY the term death?" The other tasks were associate, correlate, locate, distinguish, rank, categorize, cluster, and identify. Four visualization types were compared and performance was measured by time to completion of each task and the correctness of the given answer.

Four visualizations of hierarchical data were compared by Barlow and Neville [21]. The tasks used to evaluate the visualizations were based on the requirements of the user in a data mining context, which are:

- Ease of interpretation (understand parent-child relationships)
- Comparison of node size
- User preference

Five tasks were chosen to test the ability of the view to communicate the tree topology, which are:

- *Binary or N-Ary* - Has every parent exactly two children or not?
- *Deepest Common Ancestor* - Which node is the deepest common ancestor of two highlighted nodes?

- *Balanced or Unbalanced* - Are the leaves located on the same or the consecutive level?
- *Number of Levels* - How many levels has the tree?
- *Three Largest Leaves* - Which three leaves are the largest?

The participants' performance was measured by the time to complete a task and the correctness of their answers. Additionally, the users ranked the views according to their preference. In a second experiment the visualizations were compared according to their ability to help the user in more complex tasks, like memorizing a before highlighted node or finding the ancestor of the deepest leaves in a given level.

Three complete information visualization systems were tested by Kobsa [31], namely Eureka, InfoZoom and Spotfire. Eureka offers only a single visualization with rows being the objects and columns the dimensions, for example the attributes of the objects. Columns may be sorted by clicking on the category label on top of a column while the other columns rearrange themselves to stay consistent. By moving two columns to the left their entries are grouped. InfoZoom has three views: the wide view presents the whole data in a table format with rows being the attributes and columns the objects, the compressed view packs the current dataset horizontally to fit the window width, and the overview mode where the values in the rows become detached from their objects, which means that they represent values distributions of attributes. By clicking on an attribute value the user zooms in to show only records that contains that value. Besides the scatterplot, Spotfire offers several types of graphics, e.g. histograms or charts. Attribute ranges are selected by sliders, checkboxes or radio buttons. The tasks for the user testing were defined during a brainstorming process based on whether or not they would naturally occur in the analysis of the respective datasets. Biasing effects due to knowledge of one of the systems tested could be foreclosed. Performance measurements were answer correctness and completion time.

4 The Dataset

The Internet was searched for available datasets containing temporal information which can be used for the subsequent tool comparison. After a short phase of exploration using a tool of choice, namely Tableau, the dataset and its contents are now described. The chosen working tasks which are going to be used to help in the comparison of the visualization systems are mentioned too. Special emphasis is given to the temporal information and how it is connected to the other variables.

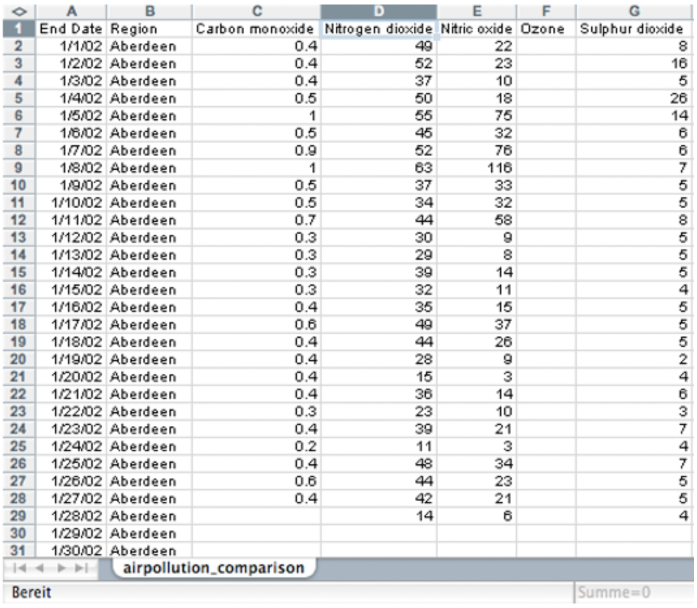
4.1 Dataset Description

The data set that is going to be used to compare the information visualization tools was obtained from the UK National Air Quality Archive[17]. At this website, air pollution data from the present back to 1960 is offered. The user is able to choose between automatic monitoring data either measured hourly, daily or monthly, descriptive statistics including mean and maximum or minimum values, and exceedence statistics, like strategies to reduce air pollution or effects on health. Queries can be interactively chosen and the resulting csv file is sent to the e-mail address provided by the user.

The following measurement sites spread over the United Kingdom are included in the data set, as shown in Figure 4.1: Aberdeen, Birmingham Centre, Glasgow Centre, London Westminster and Plymouth Centre. Air pollutants compared are Carbon monoxide, Nitric oxide, Nitrogen dioxide, Ozone and Sulphur dioxide. The daily mean of each of these pollutants from January the 1st 2002 until December the 31st 2006 is contained in the data set. In addition to that, the status of each measurement site and the unit used in the measurement process is specified. Missing data points are declared by a ”No data” label.

4.2 Working Tasks

The working tasks which should be carried out with each tool are listed now, along with points of special interest during the accomplishment of the tasks. Different user tasks are covered, like for example exploring the data, finding a specific point in time or filtering out items.



	A	B	C	D	E	F	G
1	End Date	Region	Carbon monoxide	Nitrogen dioxide	Nitric oxide	Ozone	Sulphur dioxide
2	1/1/02	Aberdeen	0.4	49	22		8
3	1/2/02	Aberdeen	0.4	52	23		16
4	1/3/02	Aberdeen	0.4	37	10		5
5	1/4/02	Aberdeen	0.5	50	18		26
6	1/5/02	Aberdeen	1	55	75		14
7	1/6/02	Aberdeen	0.5	45	32		6
8	1/7/02	Aberdeen	0.9	52	76		6
9	1/8/02	Aberdeen	1	63	116		7
10	1/9/02	Aberdeen	0.5	37	33		5
11	1/10/02	Aberdeen	0.5	34	32		5
12	1/11/02	Aberdeen	0.7	44	58		8
13	1/12/02	Aberdeen	0.3	30	9		5
14	1/13/02	Aberdeen	0.3	29	8		5
15	1/14/02	Aberdeen	0.3	39	14		5
16	1/15/02	Aberdeen	0.3	32	11		4
17	1/16/02	Aberdeen	0.4	35	15		5
18	1/17/02	Aberdeen	0.6	49	37		5
19	1/18/02	Aberdeen	0.4	44	26		5
20	1/19/02	Aberdeen	0.4	28	9		2
21	1/20/02	Aberdeen	0.4	15	3		4
22	1/21/02	Aberdeen	0.4	36	14		6
23	1/22/02	Aberdeen	0.3	23	10		3
24	1/23/02	Aberdeen	0.4	39	21		7
25	1/24/02	Aberdeen	0.2	11	3		4
26	1/25/02	Aberdeen	0.4	48	34		7
27	1/26/02	Aberdeen	0.6	44	23		5
28	1/27/02	Aberdeen	0.4	42	21		5
29	1/28/02	Aberdeen		14	6		4
30	1/29/02	Aberdeen					
31	1/30/02	Aberdeen					

airpollution_comparison

Bereit Summe=0

Figure 4.1: **The Dataset.** *The collected data of air pollution measurement sites in Aberdeen, Birmingham, Glasgow, London and Plymouth are used to evaluate the information visualization applications. The data have been collected from 2001 to 2006 and include measurements for the pollutants Carbon monoxide, Nitric oxide, Nitrogen dioxide, Ozone and Sulphur dioxide, as seen on the right side. Due to measurement errors or other failures, missing data values appear in the data set. (image courtesy by [5]).*

4.2.1 Low-level tasks

Basic questions concerning the existence of a variable at a point in time and other low-level issues are answered in the completion of these tasks. The handling of temporal primitives and their relationships to the variables in the dataset is an essential part that is examined here [26].

- *1L - Existence - Was a Nitrogen dioxide value measured on the 17th of May 2006 in Birmingham?* - This task describes if a measure value exists at a specific point in time. The user has to identify a point in time and relate it to a measured data value.

- *2L - Location - At which points in time was a value of 19 measured for Sulphur dioxide?* - Starting from a fixed value, the user has to find the temporal locations where this value was measured.

- *3L - Time interval - Which temporal range is covered by the Ozone measurements in Plymouth?* - The measurement duration of Ozone has to be determined in this task.

- *4L - Temporal texture - How often reached the Ozone pollution a value higher than 80 in 2004?* - The user has to count the occurrence of a data value to solve this task. Based on this the perception of patterns is possible.

- *5L - Rate of change - How much did the Sulphur dioxide pollution change between January and July 2003?* - This task deals with the visual recognition of the difference of data values. The observation of the magnitude of change of the Sulphur dioxide measurements at different points in time is required to complete this task.
- *6L - Sequence - In Glasgow, was an Ozone measurement of 10 made just before or just after the value of 170 of Nitroc oxide on the 8th of December 2003?* - Here, the temporal order of data elements has to be determined. The possibility to eliminate unwanted elements is of great importance for solving this task.
- *7L - Synchronization - Was a value of 3 of Nitric oxide ever measured together with a value of 27 of Nitrogen dioxide?* - The user has to detect whether two measured values ever occur at the same point in time or not.

4.2.2 Basic Tasks

These tasks handle only one or two variables at a time. Often the solution can be found by looking at one plot. The main challenge for the user is to find the right visualization that suits the given task.

- *1B - What range has the Carbon monoxide pollution?* - In this simple single range task the user only has to look at one variable only and determine minimum and maximum values.
- *2B - What was the highest Sulphur dioxide value measured in 2002?* - The user has to find the maximum value of one pollutant for 2002.
- *3B - Which measurement site had the highest Nitric oxide pollution in 2004?* - Here, the pollution values of one variable have to be summed up for each measurement site. The highest sum has to be determined.
- *4B - Are there regions with both high Nitric oxide and Nitrogen dioxide values at the same time?* - Two variables are seen in correlation. This can be done, for example, with a scatterplot.
- *5B - Which year had the highest ozone pollution in London?* - The user has to compare the pollution sum for each year and find the highest.

4.2.3 Advanced Tasks

Here, the user has to compare several variables or filter out unwanted items. Some tasks deal with trends over the last five years or correlations between different variables. Specific knowledge of the visualization tool used is needed to find the solutions in acceptable time.

- *1A - Which region has the highest/lowest air pollution? Is the south of Great Britain more polluted as the North, as one would expect?* - The user has to compare pollution sums of all regions to find the region with the highest air pollution.
- *2A - Are there pollutants having high values at specific seasons?* - First, the range of each pollutant is determined to know which values have to be judged as high for each variable. Then, the behavior of each pollutant over the last five years has to be analyzed to find seasons with high values.
- *3A - Are there any correlations between pollutants? Does the concentration of all of them increase together, or are there pollutants that are independent of the others?* - In this task, pollutants are analyzed in relation to each other to find correlations and pollutants that do not increase while others are.
- *4A - Has the general pollution increased in the last five years? Or could some pollutants be reduced?* - The user has to analyze the trend of pollution over the last five years and also look at each pollutant exclusively to answer this question.

5 The Information Visualization Tools

After the practice with every tool and the reading of the related documentation a description of the tools facilities is firstly given. The description includes informations about which data types can be visualized with the tool and which visualizations are used to visualize the data. Furthermore, the interface and its components are described, along with the possible user interactions.

The second part of each tool description deals with the practical knowledge gained during the accomplishment of the previously defined working tasks. Beginning with Tableau, the results have been verified by using a second visualization tool, Spotfire. The collected experiences are used to evaluate the user support in exploring and filtering the data. The differences between the tools in presentation and available user interactions are brought into focus. Special emphasis is also put on the available visualizations for temporal data. Problems of the software tools are pointed out and possible improvements are suggested.

5.1 Tableau

The visual analysis tool Tableau[16] assists the user in the exploration of databases, from small ones like Excel or Access to big ones like Oracle. The user chooses out of a variety of views to display his/her data set. Tableau is designed to allow business people with no technical training to analyze their data efficiently. Users are enabled to build views, connect them with each other, issue queries and filter out or build groups of items. Visualization results can be exported as images and shared with other team members. Tableau's online help is detailed and includes a search function by text or index. Supported input data include, among others, Excel, Access and comma delimited text files, MySQL and PostgreSQL databases and also Oracle and Microsoft SQL Server.

5.1.1 Importing Data

To import data into Tableau, the user has to connect to a data source. This is done either by using the Open dialog in the case of simple files, like comma delimited text files or Excel files, or by opening the Connect option data dialog. Hence, the type of data source is selected and the data are imported. It is possible to create a shortcut to a data source to bypass the connection configuration in the future. When connecting to a relational data source, a custom SQL query can be issued rather than connecting to the entire data source, or multiple tables can be joined.

After the establishment of the data connection, the data source fields are displayed in the data window on the left-hand side of the current working environment, called a workbook.

5.1.2 User Interface Components

The *data window* on the left-hand side organizes the data into categories: dimensions and measures. Dimension fields typically hold discrete qualitative data, like dates or customer names, while measures hold numerical data that can be aggregated, for example data values collected in a weather station. A search function is included in the data window and the hiding of selected fields is possible. In the basic process of generating a view, the user selects a data field from the data view and drags it onto a shelf or directly onto one components of a table. The basic shelves are the *columns and row shelves*. As their names indicate, they create the columns and rows of a table. Dragging a dimension field to one of these shelves leads to the generation of a header which is kind of an axis label and the same done with a measure creates a quantitative axis. Adding more fields to these shelves results in a more and more detailed view of the data. Another shelf is the *page shelf*: dragging a field to it divides the visualization to multiple pages to ease the analysis how a specific field affects the rest of the data in a view. The information is displayed by visual entities, e.g., bars and lines, the so-called marks. Which mark is used is decided according to the data types but can be changed by the user. Navigation tools include selecting, panning, zooming and focusing on a small area while retaining the context of the visualization.

5.1.3 User Interaction

To effectively encode the desired data, not only rows and columns are utilized. Visual features like color, shape and size can be associated with data fields by the user. All these features are represented by shelves in the workbook. Dragging a field on a shelf results in the assignment of this field to the visual feature. When the user adds a field to the *color shelf*, the field name is put in the color text field. Underneath the relation of color to field name is shown and can be changed by clicking on it. Placing a dimension field on the *shape shelf* assigns different shapes to each of the members in the dimension. Measures are converted to a discrete measure when doing this. The *size shelf* works similar to the shape shelf: moving a field to it connects the size attribute with the field values. Continuous fields use ranges to assign sizes to values.

The *level of detail shelf* allows the encoding of additional information in the visualization. Placing a field on it separates the marks in a data view according to the members. By clicking an mark, the corresponding level of detail segment is highlighted, as shown in Figure 5.1. To exclude unwanted data from the view, the *filters shelf* is employed. After placing a field on it the filter dialog box opens automatically. There, ranges and values that should be

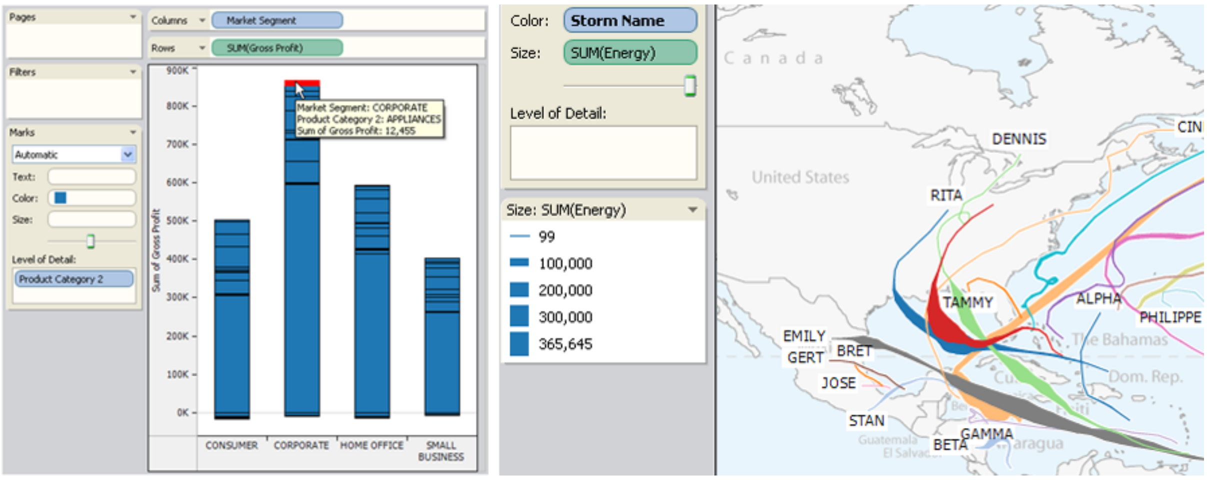


Figure 5.1: **Tableau: Enhancing a view.** *Using the level of detail shelf a mark is partitioned according to field members. Selection of mark reveals the partitioning by red highlighting, as shown on the left-hand side. The actual field values are displayed by a tooltip. Above the bars the row and column shelf are situated. The right-hand side shows a path visualization of storm data from the Atlantic basin. The assignment of line thickness to the strength of the storm can be seen in the middle. (image courtesy: [16]).*

excluded are selected. The arrangement of field members is set with the sort dialog box. Sorting can be chosen to be ascending, descending or completely manual. The connection of sort order and the aggregation of a data field is possible as well.

5.1.4 Special Features

The *path shelf* is a useful tool to visualize data with an inherent temporal relation. It allows the encoding of data by connecting marks using a specific drawing order according to the data values. If a dimension field is a date, the order is given by the date order. Dimensions holding characters are ordered alphabetically. Measures are ordered corresponding to their values. Ordering can be changed by changing the sort order of the data members. Figure 5.1 shows an example of such a path visualization. Whenever a date type is visualized, its granularity can be chosen dynamically.

5.1.5 Evaluation

After reading some documentation about the Tableau environment the defined tasks were carried out the first time. Tableau was chosen because due to the fast communication about getting a trial there was enough time to get familiar with it and therefore to work in an efficient way. Tableau's user interface is clearly structured because all data values and visualization variables, like color and size, are visible all the time. The user always knows if a filter is currently applied to the data, or which color corresponds to which value. The handling of variables and shelves is very intuitive and a lot of decisions are made automatically for the user which lead to meaningful visualizations. The plus and minus signs on the variable symbols are very practical to zoom into the temporal history. During the work, often the whole shelf has to be cleaned to start a new visualization. This has to be done via a second-level menu item in the top menu. Therefore, an always visible button would be very practicable to do this.

Although Tableau should be able to read in delimited text files, some effort had to be made to get the data read in. The first solution that worked was to open the comma delimited file with Microsoft Excel and save it as a .xls file. The input of this file worked fine then. The following sections describe the tasks carried out, the found solutions and problems that occurred during the use of Tableau.

Low-level Tasks

Especially, these tasks were done most comfortably by using Tableau. Time is shown and interpreted in a very intuitive way and only the essential values are displayed. This leads to both a very good overview of the data set and insights into detailed temporal aspects.

1L - Existence - Was a Nitrogen dioxide value measured on the 17th of April 2006 in Birmingham?
This task is simply solved by filtering out the desired date and moving Year and Region to one shelf and Nitrogen dioxide to the other shelf.

2L - Location - At which points in time was a value of 19 measured for Sulphur dioxide?
Again, the year is dragged to one shelf and then split up to show single days. The other shelf holds the Region and the Sulphur dioxide variable. The splitting up by regions is very important to avoid the possibility of overlapping data points. Tableau arranges the visualization in a way that the user sees the desired dates simply at first glance.

3L - Time interval - Which temporal range is covered by the Ozone measurements in Plymouth?
Dragging the year to one shelf and Ozone to the other creates a simple line plot that shows the Ozone progress over the last five years. The unwanted regions are filtered out by using the filter shelf. As Figure 5.2 indicates, the Ozone range starts at the 1st of January 2002 and ends at the 19th of February of 2006.

4L - Temporal texture - How often reached the Ozone pollution a value higher than 80 in 2004?
With the filter shelf all years but 2004 and Ozone values under 80 are removed from the data.

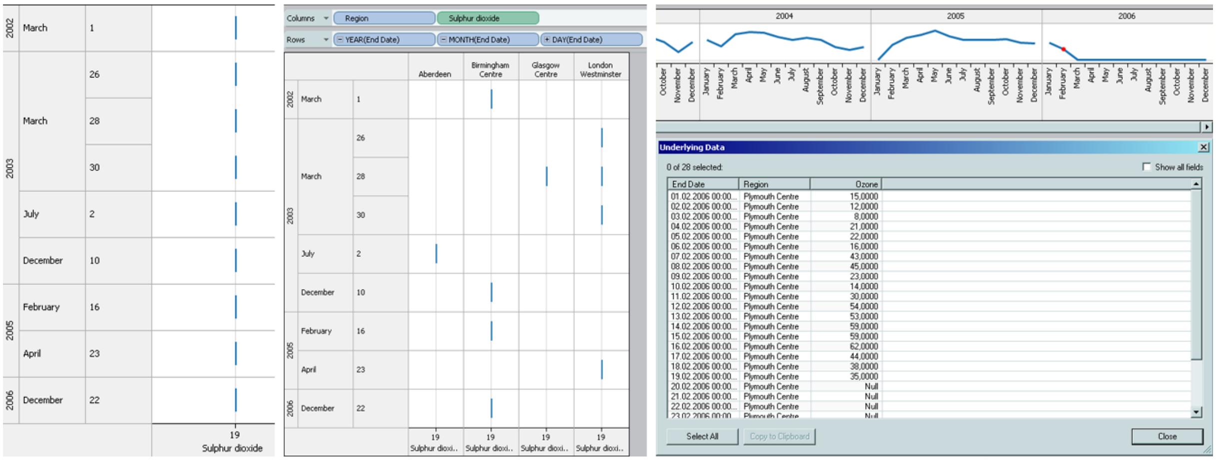


Figure 5.2: **Tableau: Tasks 2L and 3L.** *The left-hand side shows all points in time with Sulphur dioxide measurements of 19. First, as shown on the far left, it seemed as if 9 points are relevant, but a refinement of the visualization by adding the Region variable unveiled that there are 10. The line plot on the right side depicts the Ozone progress from 2002 to 2006. Apparently, the Ozone measurement stops somewhere in February 2006. Using the context menu to retrieve the exact values measured in February reveals that the last measurement was made on the 19th of February. The filtering out of null-values in advance is possible as well. (image courtesy: [16]).*

Then, the year is dragged to one shelf and Ozone to the other. The user has to select count as aggregation function to retrieve the desired result. The solution is determined by clicking on the data point shown or, if the year is split up in months in a line plot, by selecting all the points and looking at the bottom status bar.

5L - Rate of change - How much did the Sulphur dioxide pollution change between January and July 2003?
The user has to drag the year to one shelf and Sulphur dioxide to the other to see a line plot of the Sulphur dioxide history. Then the relevant time range is set in the filter window, shown in Figure 5.3.

6L - Sequence - In Glasgow, was an Ozone measurement of 10 made just before or just after the value of 170 of Nitroc oxide on the 8th of december 2003?
Again the relevant variables are dragged to the shelves and Ozone values other than 10 or not measured in Glasgow are filtered out. In the sequence of measurements shown the user finds the result easily because the date of each point is included in the visualization.

7L - Synchronization - Was a value of 3 of Nitric oxide ever measured together with a value of 27 of Nitrogen dioxide?

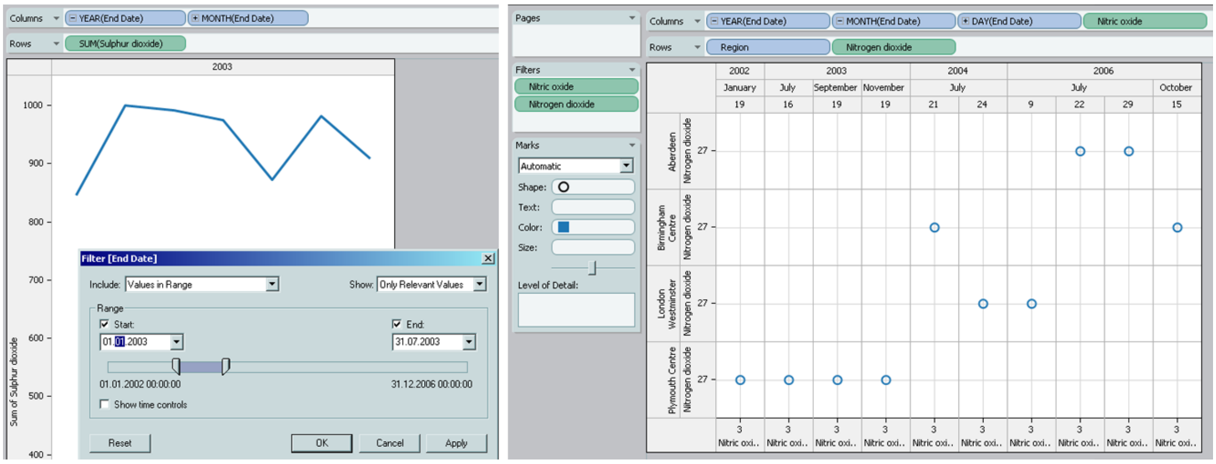


Figure 5.3: **Tableau: Tasks 5L and 7L.** *The left-hand side shows the progress of the Sulphur dioxide pollution from January to July 2003. To do this, the filter window is used. Temporal filtering in Tableau is very convenient because a date can be entered manually, chosen from a calendar or adjusted dynamically with a range filter. Corresponding Nitric oxide values of 3 and Nitrogen dioxide values of 27 are shown on the right. The user sees at first glance on which date and in which region a data point was measured. (image courtesy: [16]).*

Dragging year and Nitric oxide to one and Region and Nitrogen dioxide to the other shelf displays the desired result. The filter shelf is used again to filter out all values but 3 for Nitric oxide and 27 for Nitrogen dioxide. Figure 5.3 shows the resulting visualization.

Basic Tasks

All of the basic tasks could be accomplished within less than five minutes. The simple user interface which utilizes dragging and tooltips eases the handling of the data.

1B - What range has the Carbon monoxide pollution?
This task was simple to accomplish and there were two possibilities to find the solution, as shown in Figure 5.4. The range of a single variable can be found out by right-clicking on a data field in the data window and choosing the menu point "Describe..." in the context menu. This causes a window to open where the characteristics of this column are listed, including the data range. The second way to complete the task was to solve it graphically. By dragging the date field to the column shelf and the carbon monoxide field to the row shelf the measured values are plotted over time. Because the default aggregation function for each measure is the sum, one has to click on the icon in the shelf to show the single values. The range then can be found out by clicking on the peaks and the values at the bottom and examining the tooltip.

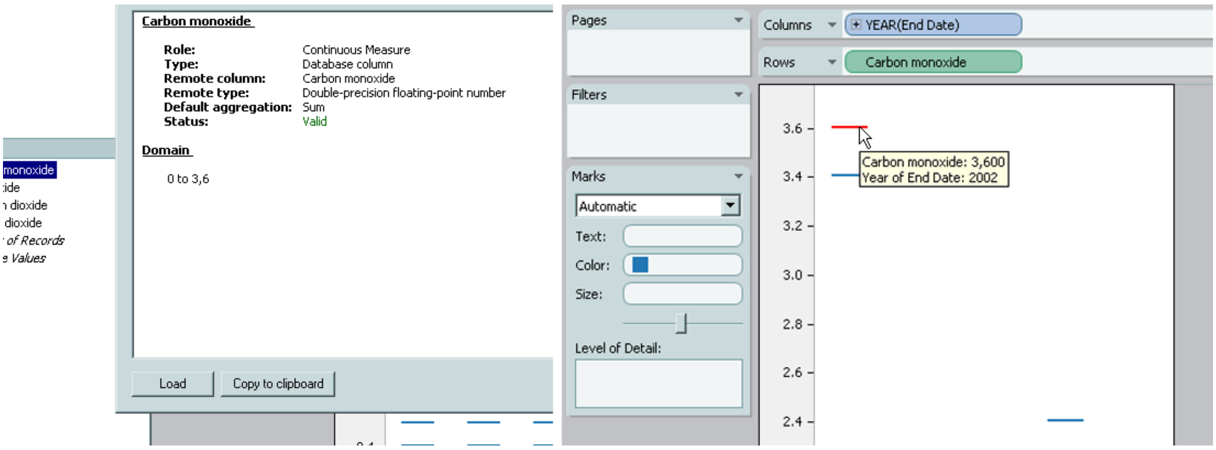


Figure 5.4: **Tableau: Task 1B.** *To find out the range of a variable, two strategies can be used. The left-hand side show the window that describes each variable on a text-basis. By dragging date and carbon monoxide to the shelfs a visualization is generated. Tooltips reveal the exact values drawn. (image courtesy: [16]).*

2B - What was the highest Sulphur dioxide value measured in 2002?
By dragging the sulphur dioxide variable to the row and the date variable to the column shelf a visualization is drawn. The user has to choose the display of single values via context menu. To focus on the year 2002 the date variable is dragged to the filter shelf where 2002 is chosen. Then the highest measured value which is 27 can be seen in the drawing.

3B - Which measurement site had the highest Nitric oxide pollution in 2004?
This task is similar to the latter, only that the sum of nitric oxide is used to show the pollution as whole, and the regions are added to the column shelf to be compared. Also, the filter is set to 2004. The answer is Glasgow which has the highest bar. Figure 5.5 displays the realization of tasks 2B and 3B.

4B - Are there regions with both high Nitric oxide and Nitrogen dioxide values at the same time?
By dragging region and Nitric oxide to one shelf and Nitrogen dioxide to the other the two variables are shown in correlation. In the scatterplots it is easy to see that high dioxide values implicate high oxide values and vice versa. Also, the dioxide doesn't increase as high as oxide. Therefore, the answer is that all regions have Nitric oxide and Nitrogen dioxide values at the same time.

5B - Which year had the highest ozone pollution in London?
Again, dragging to the shelves reveals that 2006 is the year searched for. The filtering is adjusted to eliminate all regions but London. Figure 5.6 depicts this task and the first advanced

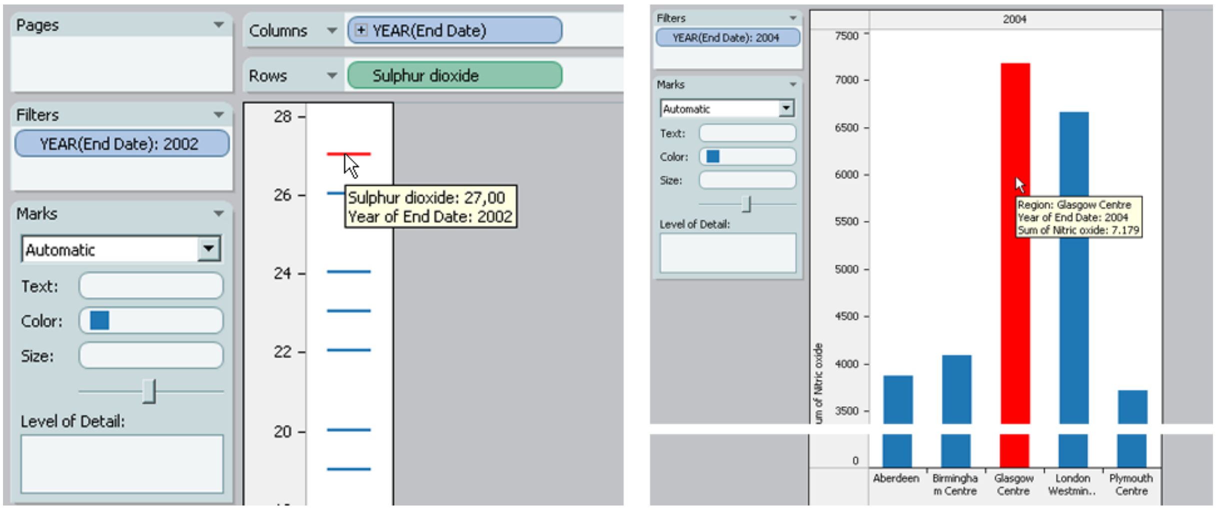


Figure 5.5: **Tableau: Tasks 2B and 3B.** *These tasks involve the use of filters to focus on specific years and the combination of more than one variable on a shelf. By simply dragging variables to the shelves visualizations are drawn and tooltips help in the navigation. (image courtesy: [16]).*

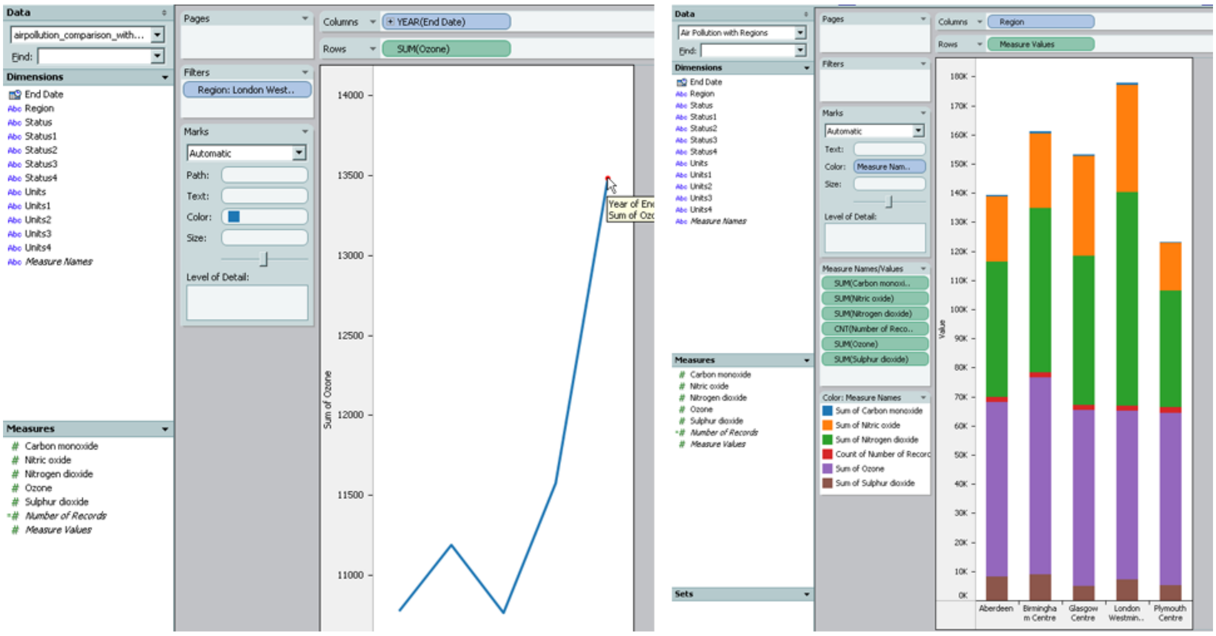


Figure 5.6: **Tableau: Tasks 5B and 1A.** *The left-hand side shows in a simple line graph that 2006 had the highest ozone pollution in London. On the right-hand side, all regions are compared to find those with the highest pollution. London and Birmingham have the highest overall pollution values. Birmingham has a high Ozone pollution, while London has the highest Nitrogen dioxide sum. (image courtesy: [16]).*

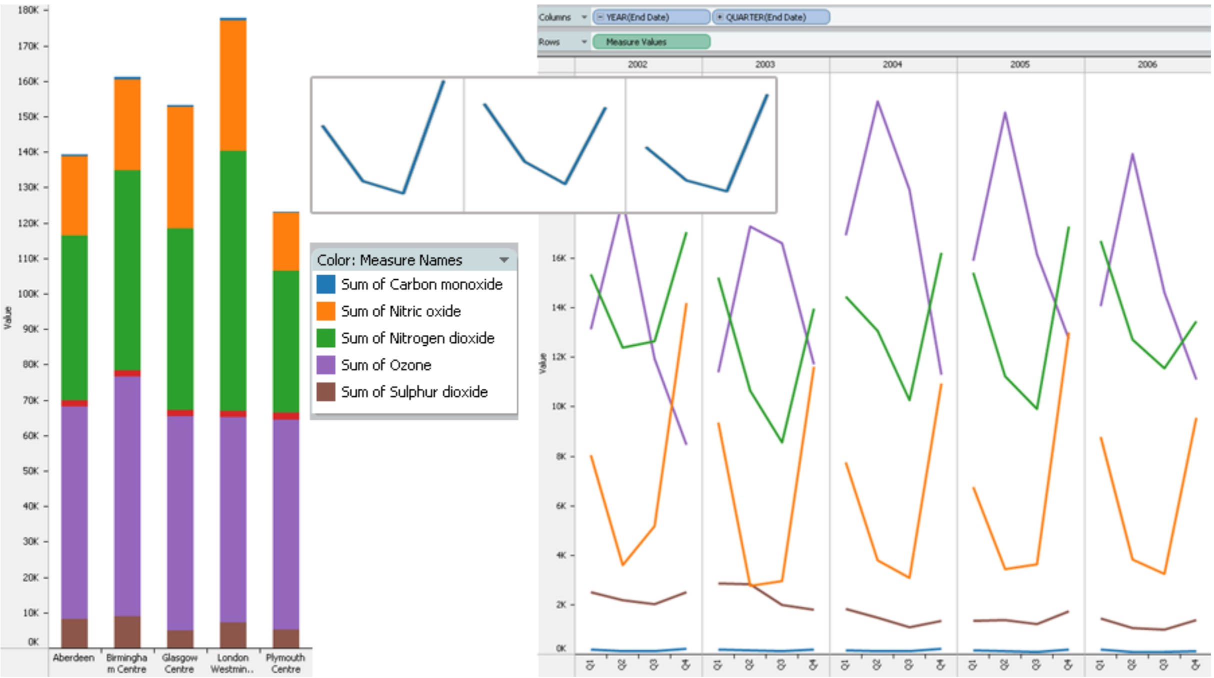


Figure 5.7: **Tableau: Tasks 1A and 2A.** *On the left-hand side the pollution sums for each region are shown. Regions in the center of Great Britain are more polluted than border regions. The right-hand side demonstrates the process of task 2A. First, the distribution of each pollutant for every quarter of the year is shown. Therefore, relationships between season and pollutant values are discovered easily. Because of the low range of the Carbon monoxide pollution a second visualization was used to get a clearer sight on the distribution of this pollutant, displayed in the middle. (image courtesy: [16]).*

task.

Advanced Tasks

1A - Which region has the highest/lowest air pollution? Is the south of Great Britain more polluted as the North, as one would expect?
When reading in data, Tableau generates three data fields respectively variables automatically: they contain the names of all measures, the values of all measures and the number of records. These variables were useful in this task, though the possibility to use them was not obvious at first. By dragging Region and Measure Values to the shelves and applying a coloring according to Measure Names, a visualization is generated that shows the sum of all pollutants for each region, as seen in Figure 5.7. Therefore, the pollution distribution is easily examined. London has the highest air pollution, while Plymouth is the lowest polluted region. Coming to comparing regions the visualization indicates that populous regions are most polluted, as London and Birmingham which lie in the southern center of Great Britain, mostly due to high Ozone and

Nitrogen dioxide values.

2A - Are there pollutants having high values at specific seasons?
Because of the time-dependency of this task the variable End Date has to be dragged to the column shelf and further subdivided according to seasons. The variable Measure Values is positioned on the row shelf, and the same coloring as in the last task is used. Tableau generates a line plot which shows the sums of each pollutant in each quarter of the last five years. Sulphur dioxide seems to have no clear relation to a specific season. Ozone reaches the highest values during the summer, and Nitrogen oxide, Nitrogen dioxide and Carbon monoxide increase during the winter. The distribution of Carbon monoxide was not visible well due to the dimensionality of the other measures, so a second visualization was used to clarify its relation to the seasons of the year. Figure 5.7 depicts this process.

3A - Are there any correlations between pollutants? Does the concentration of all of them increase together, or are there pollutants that are independent of the others?
As seen before in the last task, Carbon monoxide, Nitrogen dioxide and Nitrogen oxide values increase and decrease together, though the Carbon monoxide values values do not reach as high values as the other two pollutants do. Ozone behaves contrarily to them and reaches high values in the summer. Sulphur dioxide seems to be independent of the other pollutants.

4A - Has the general pollution increased in the last five years? Or could some pollutants be reduced?
Dragging the year variable to one shelf and the variables Region and Measure Values to the other one creates a bar visualization of the pollution over the last five years. According to the visualization, Ozone has increased definitely in Aberdeen, but looking at the input file reveals that there are no Ozone values measured from the beginning of 2002 until mid 2003. Nitrogen dioxide and oxide have not changed their values. Sulphur dioxide could be reduced in all regions but London, and Carbon monoxide got less as well. Figure 5.8 displays the realization of this task.

5.2 Spotfire DXP

The analytics platform Spotfire DXP [15] offers a wide variety of visualizations, including scatter plots, bar charts, pie charts, box plots and several others, and even tools to calculate statistical measures. Different interfaces for different tasks are provided to assist the user at the best possible rate. Spotfire DXP supports a variety of databases including OracleClient and SQLClient and also imports data from delimited text files and excel files. Imports are integrated dynamically with the application and the merging of several different data sources is possible

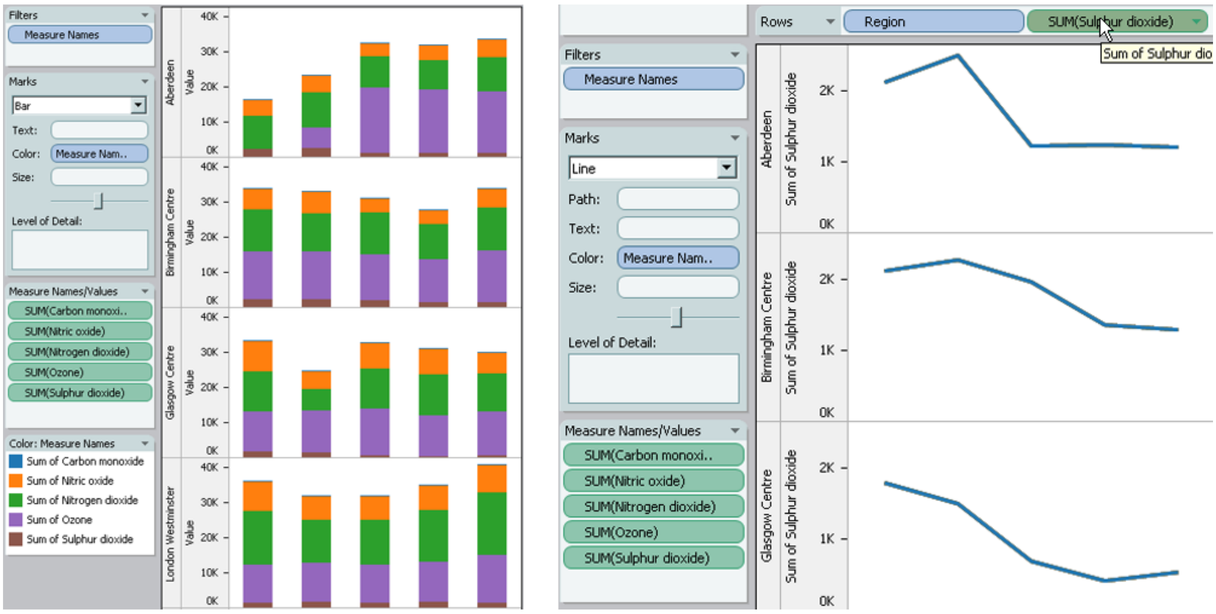


Figure 5.8: **Tableau: Task 3A.** *To find out about the general pollution changes, the visualization from task 1A is split up to one bar for each year and region. The uppermost bar for Aberdeen indicates a dramatic increase in the Ozone pollution between 2002 and 2004, but this seemingly increase is only based on missing measure values. Sulphur dioxide and Carbon monoxide have very low values, so separate visualizations like the line plot on the right are used to get a closer look. The plot reveals that the Sulphur dioxide pollution has decreased in most regions. (image courtesy: [16]).*

as well. Exporting options include powerpoint presentations, image and text files. A detailed user's manual that eases the working process comes with the application.

5.2.1 Importing Data

The import dialog supports the user in getting his/her data into Spotfire. All the columns representing variables are listed, together with the associated data types. Here, if needed, column names and data types are changed by the user, and the skipping of empty rows is activated. If a column should not be contained, it can be excluded by clicking a checkbox. Advanced settings, for example providing a label for missing data values or specifying how many rows to read in, are provided as well.

5.2.2 User Interface Components

Every component of Spotfire's user interface is highly customizable. The positions and sizes of the visualizations can be adjusted and interaction elements containing lists can be re-ordered. If there are a lot of variables and therefore a lot of interaction elements displayed, unwanted elements can be hidden or elements can be grouped. A document consists of several pages which usually include different views on the data. Figure 5.9 shows the main components of Spotfire DXP which are described in the following paragraphs.

In the *main window* all the visualizations coupled to the data are contained. The default visualization, a scatterplot, is drawn when the data are first read in. By adjusting the axes of this scatterplot, which is easily done by selecting the desired variable in an axis dropdown box, called the axis selector, the user gets a quick view of his dataset. By using sliders that are placed on the axes, the user is enabled to zoom into a region of interest. Moving the cursor over a visualized data point causes a tooltip to appear that displays details about the highlighted item.

The *Filter Panel* allows the user to quickly navigate through his/her data and to remove unwanted items from the visualization. There are several filter types, e.g., checkboxes, radio buttons or range sliders, and the type of filter device is chosen by the user according to his needs and his data. When a filter is manipulated, all the linked visualizations are updated dynamically. The left and right box of a range filter are used to change the lower and upper limit of the range of a variable, and moving the currently selected interval is a powerful way to look at different ”slices” of the data. Values on a range filter are always positioned to be distributed on a linear scale, even if the original data values are not.

The *Details-on-Demand Panel* is used to give information about the exact values of a data point or a group of data points. By clicking an item in the visualization or marking several items by drawing a rectangle around them the numerical and textual values they represent are shown. If the selection should be remembered, a tag can be attached to marked items and is shown in the *Tag Panel* afterwards. Storing a snapshot in the course of analysis is done by creating a bookmark.

5.2.3 User Interaction

When a data set is analyzed using multiple visualizations, all of them are coupled and interactive. This means that if the user issues a dynamic query with sliders, the resulting sections are highlighted in all of the visualizations. The same occurs when a section is highlighted by the user. This technique is often called brushing and linking. The drag-and-drop technique is frequently used in Spotfire DXP, for example to adjust visual features, like size and coloring, or to split up a visualization according to the column dragged. So-called drop targets appear when a filter is dragged from the filter panel to the visualization and change the visualization accordingly, as shown in Figure 5.10.

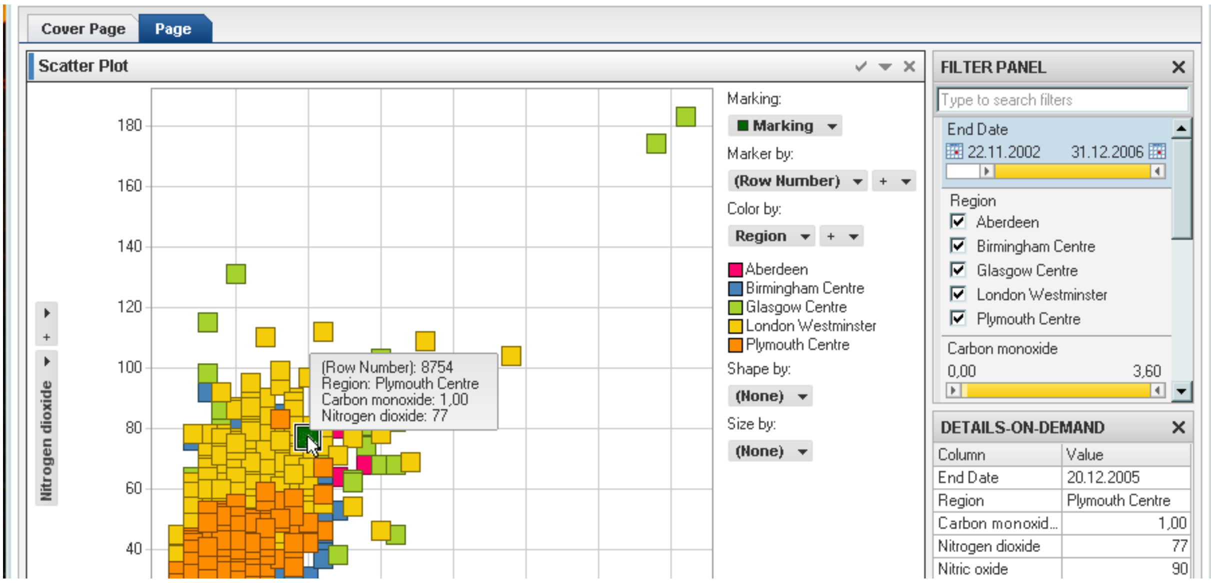


Figure 5.9: **Spotfire DXP.** *The main part of the Spotfire window is covered by one or several visualizations which can be positioned dynamically. On the right-hand side the filter and details-on-demand panels are displayed. The type of filter can be adjusted and by modifying a filter unwanted visualization items are hidden. The details-on-demand panel shows the exact values of a selected data point. The displayed scatterplot is colored according to the region of each data point. Changing color, shape or item size can be done by drop targets (see Figure 5.10) or by choosing the reference variable at the right side of a visualization. (image courtesy: [15]).*

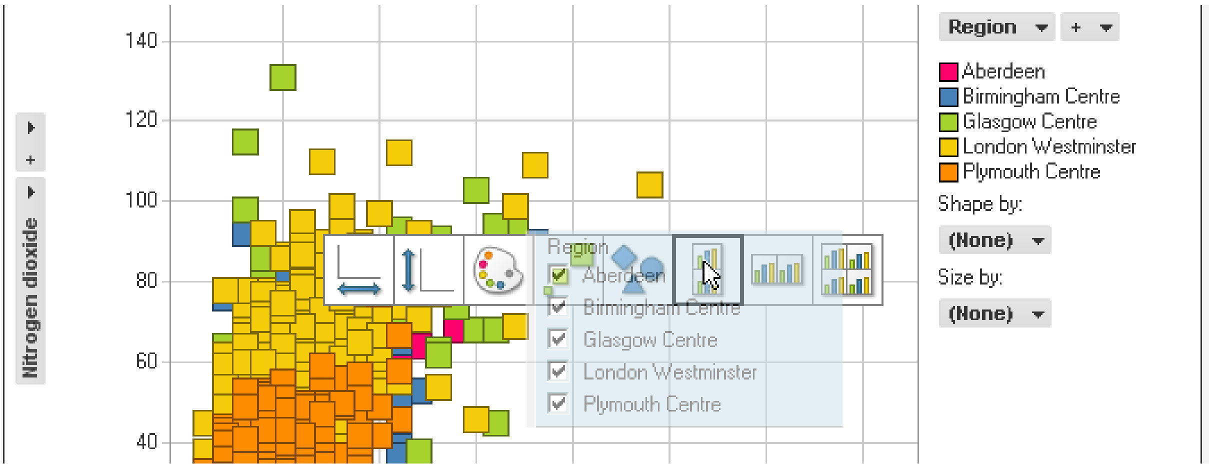


Figure 5.10: **Spotfire DXP - Drop targets.** *Adjusting the properties of visualizations is done via using the panels on the right side or by dragging filters to the visualization. Doing so, available drop targets respective icons appear that represent the possible user operations. These include changing the values for the axes, changing color and size according to a variable's value and splitting up the visualization in several parts. (image courtesy: [15]).*

5.2.4 Special features

Spotfire addresses the needs of a multi-user system: if a DXP file should be shared with other people, their analysis steps can be guided by changing the navigation to a step-by-step system or by placing a link to a relevant tool at a certain point.

Settings for a column are not only editable during the input process but also later on in the column properties window. Columns do not have to consist of static values coming from the input only, storing a dynamically calculated expression is possible as well. The user also has the ability to insert a column that organizes another column in bins.

5.2.5 Evaluation

After a short phase of reading some manual sections and watching introductory videos about Spotfire, the air pollution data was read in. The input of an Excel file did not work though this should be possible according to the manual, but a delimited text file was accepted without any problems. The input process using Spotfire took only about five minutes which was the shortest among all the applications evaluated, and resulted in the scatter plot visualization chosen by default.

Spotfire is a tool which has a wide range of possibilities for the analysis of data. Sometimes it happened that not visible filters were forgotten and led to irritating results, e.g., in bar chart sums, although the user interface is structured intuitively. But this problem may also originate in the minor user experience and disappear when one works with Spotfire more often.

Low-level tasks

1L - Existence - Was a Nitrogen dioxide value measured on the 17th of April 2006 in Birmingham?
By changing the date filter to a text field to only see the measures of one day and typing in the 17th of April 2006 an inserted bar chart shows the measure values. Nitrogen dioxide and the Region have to be selected for the axes of this chart to show the desired result. Clicking on the bar that illustrates the region Birmingham causes the details-on-demand window to show all the values measured on this day.

2L - Location - At which points in time was a value of 19 measured for Sulphur dioxide?
The Sulphur dioxide filter has to be changed to a radio button filter to select the value of 19. An inserted scatterplot with the End Date on one and Region and Sulphur dioxide on the other axis shows the result. Though it seems as if there are 8 points in time, a later on analysis in Tableau revealed that three points overlapped in Spotfire - see also Figure 5.11.

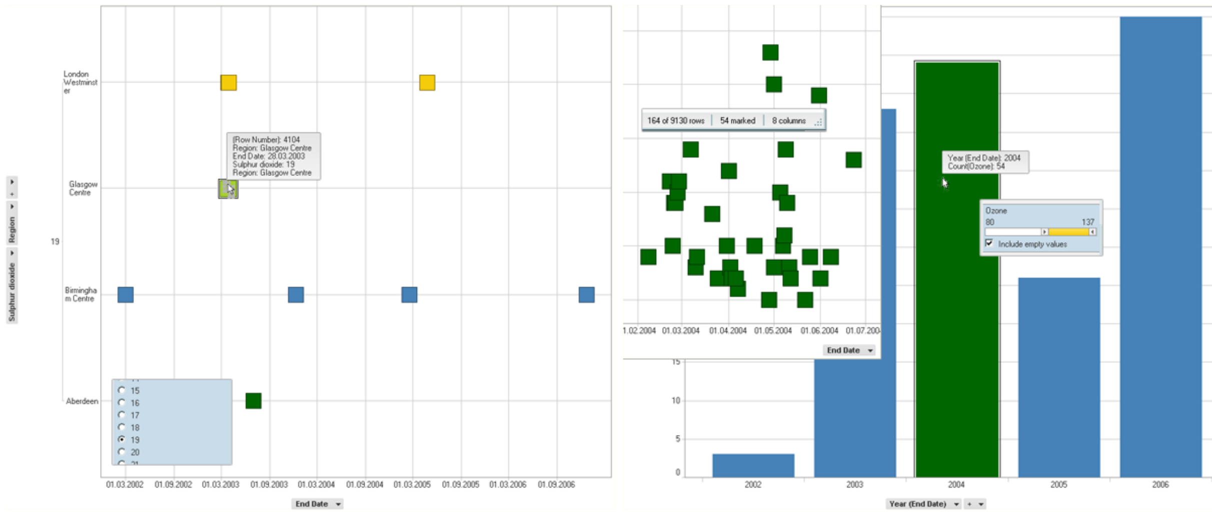


Figure 5.11: **Spotfire DXP: Tasks 2L and 4L.** *The scatterplot on the left-hand side displays all Sulphur dioxide measurements of 19. In this case, the scatterplot led to the wrong conclusion of 8 relevant points due to overlapping of the upper left yellow point. The accomplishment of the same task in Tableau revealed that there are 10 points with measurement 19. To find all points with an Ozone measurement higher than 80 the Ozone filter has to be adjusted. Then, a scatter or bar plot can be used to determine the number of them. Selecting all scatter plot points lets the status bar show how many they are and a tooltip gives information about the bar plot. (image courtesy: [15]).*

3L - Time interval - Which temporal range is covered by the Ozone measurements in Plymouth? Marking the first respective last values in a scatterplot of the Ozone measurements in Plymouth over time lets the details-on-demand window display these points. The user then just has to sort them by the variable End Date to find the beginning and end of the measurement interval - from the 1st of January 2002 until the 19th of February 2006 which means 4 years and 50 days.

4L - Temporal texture - How often reached the Ozone pollution a value higher than 80 in 2004? To accomplish this task, the Ozone and Region filters have to be adjusted. Then, the user has two possibilities: firstly, to select all points in a scatterplot to show the number of points in the bottom status bar. Secondly, to use the aggregation function count in a bar plot, as shown in Figure 5.11.

5L - Rate of change - How much did the Sulphur dioxide pollution change between January and July 2003?
After adjusting the date range filter, a simple line plot answers this task.

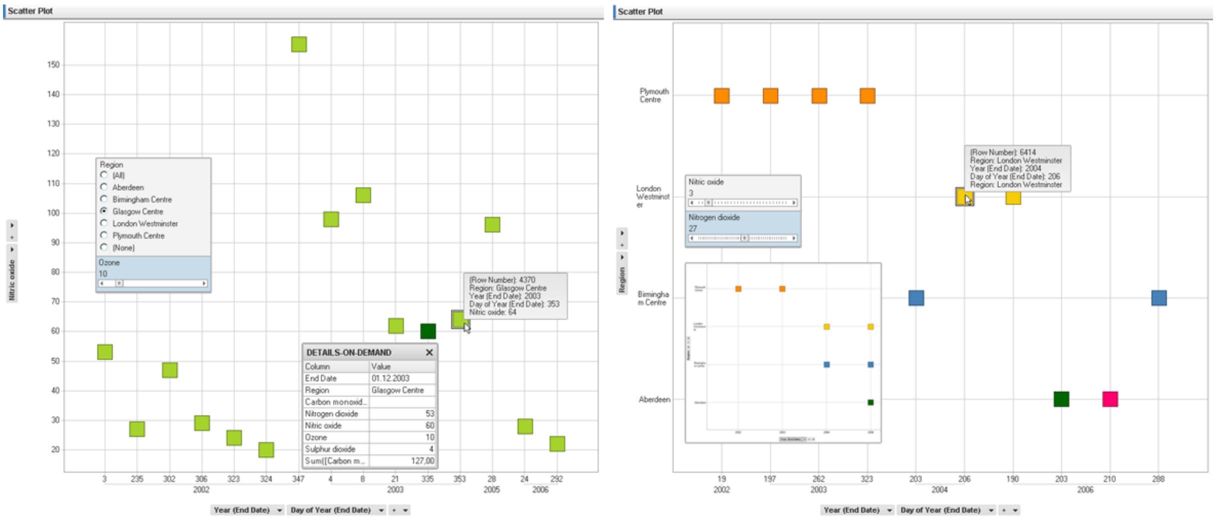


Figure 5.12: **Spotfire DXP: Tasks 6L and 7L.** *On the left-hand side the measurement just before or after the Nitric oxide measurement on the 8th of December is looked up in a scatter plot. First, the filters have to be adjusted according to the task. Then, the detail-on-demand window helps in the discovery of the task solution. The right-hand side shows the points in time where a Nitric oxide value of 3 and a Nitrogen dioxide value of 27 were measured together. In the little window, the visualization is depicted without splitting up the visualization to the single days. Therefore, only 7 points seem to meet the condition. After refining the visualization, the correct 10 points are shown. (image courtesy: [15]).*

6L - Sequence - In Glasgow, was an Ozone measurement of 10 made just before or just after the value of 170 of Nitric oxide on the 8th of December 2003?
For this task the Ozone filter has to be changed to an item filter where one measurement item is chosen easily. After choosing Glasgow as region and the insertion of a scatterplot all Ozone measurements of 10 are shown. Then the nearest Ozone measurement on the 1st of December 2003 is found by clicking on the nearest measurements and examining the exact date in the details-on-demand window, as Figure 5.12 indicates.

7L - Synchronization - Was a value of 3 of Nitric oxide ever measured together with a value of 27 of Nitrogen dioxide?
After changing the Nitric oxide and Nitrogen dioxide filters to item filters a scatterplot shows the relevant points. To avoid misleading results, it is important to split the visualization by regions and year to see how many data points meet the condition. Figure 5.12 explains that otherwise the possibility of overlapping points is given. The correct answer to this task is ten times.

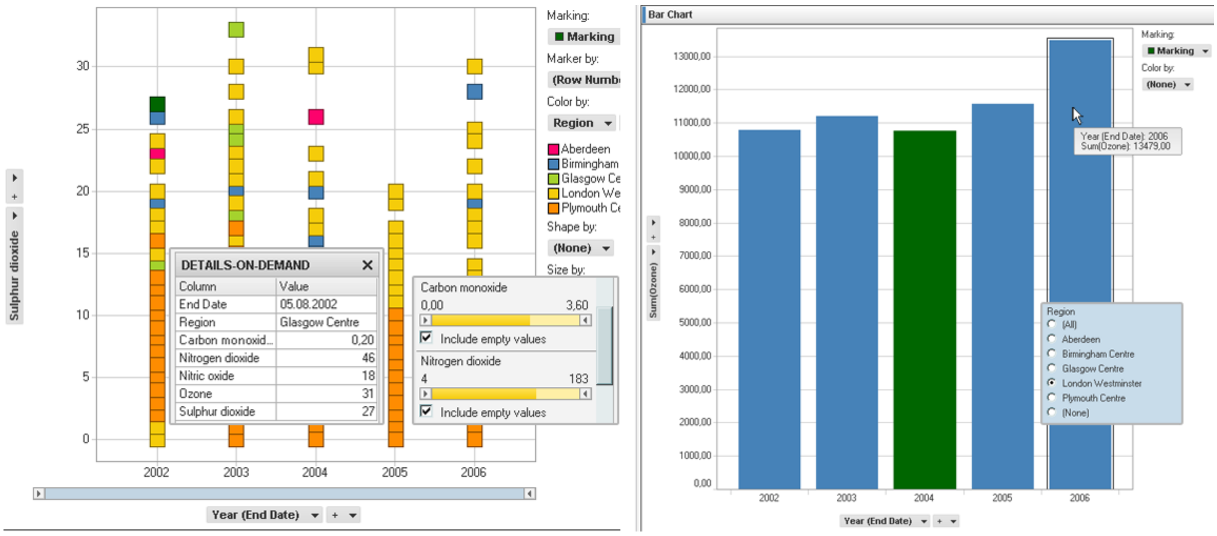


Figure 5.13: **Spotfire DXP: Tasks 1B,2B and 5B.** *The user interface elements on the right-hand side of the Spotfire window offer easily accessible information about the data visualized. The first task is solved by looking at the Carbon monoxide filter and clicking on the highest Sulphur dioxide value in the left visualization reveals all values of this data point in the Details-on-Demand window. On the right-hand side the Ozone pollution in London over the last five years is shown. The region London is selected by using a radio button filter for the Region variable. (image courtesy: [15]).*

Basic tasks

1B - What range has the Carbon monoxide pollution?

This task is carried out fast by looking at the filter panel where all the variables and their ranges are listed.

2B - What was the highest Sulphur dioxide value measured in 2002?

To find out the solution for this task, only the axes of the scatter plot have to be adjusted to use the date and the Sulphur dioxide values. Clicking on the highest value in the year 2002 causes the Details-on-Demand window to show all the values for this data point, as seen in Figure 5.13.

3B - Which measurement site had the highest Nitric oxide pollution in 2004?

For this task the Nitric oxide pollution has to be summed up for the year 2004. Therefore, a different visualization has to be used which was not clear in the first place. But a little trying out led to the right visualization, namely a bar visualization, which allows the user to employ a large number of aggregation functions on the data. The highest bar then indicates that Glasgow had the highest Nitric oxide pollution in 2004.

4B - Are there regions with both high Nitric oxide and Nitrogen dioxide values at the same time?
Choosing the two Nitric pollutants for the scatter plot axes shows them in correlation. By dragging the region filter to the split drop target the visualization is split up to examine each region by itself. During this task it happened the first time that a previously adjusted filter was adjusted which led to a wrong result at first, but was corrected later. The scatter plots clearly indicate that high Nitric oxide values go along with high Nitric dioxide values and vice versa.

5B - Which year had the highest ozone pollution in London?
This task requires building a sum again and therefore, the bar visualization is used. The already familiar process of selecting the relevant axes, namely Year and Ozone, adjusts the visualization according to the user's needs. To eliminate all regions but London the Region filter is conveniently changed to use radio buttons and London is selected, as shown in Figure 5.13.

Advanced tasks

1A - Which region has the highest/lowest air pollution? Is the south of Great Britain more polluted as the North, as one would expect?
To find out about the general pollution a calculated column that sums all the pollutants has to be added to the data values. This column is selected for the y axis of a bar plot. Then the highest and lowest polluted regions in Great Britain can be found out, as shown in Figure 5.14. A coloring by pollutant is not possible here because only a single variable can be chosen for that.

2A - Are there pollutants having high values at specific seasons?
For this task five line charts are used that show the behavior of each pollutant for each month of the last five years. To ease the creation of such five similar visualizations a duplicate visualization function is provided. Figure 5.14 displays part of this visualization.

3A - Are there any correlations between pollutants? Does the concentration of all of them increase together, or are there pollutants that are independent of the others?
This task can be solved by either using a parallel coordinates plot or the visualization from the last task. Figure 5.15 illustrates that the parallel coordinates plot is a bit crowded due to the size of the data set. Some kind of clustering as the xmdv Tool has implemented would be useful here.

4A - Has the general pollution increased in the last five years? Or could some pollutants be reduced?
To look at each pollutant by itself, one bar visualization per pollutant is created. The bars are

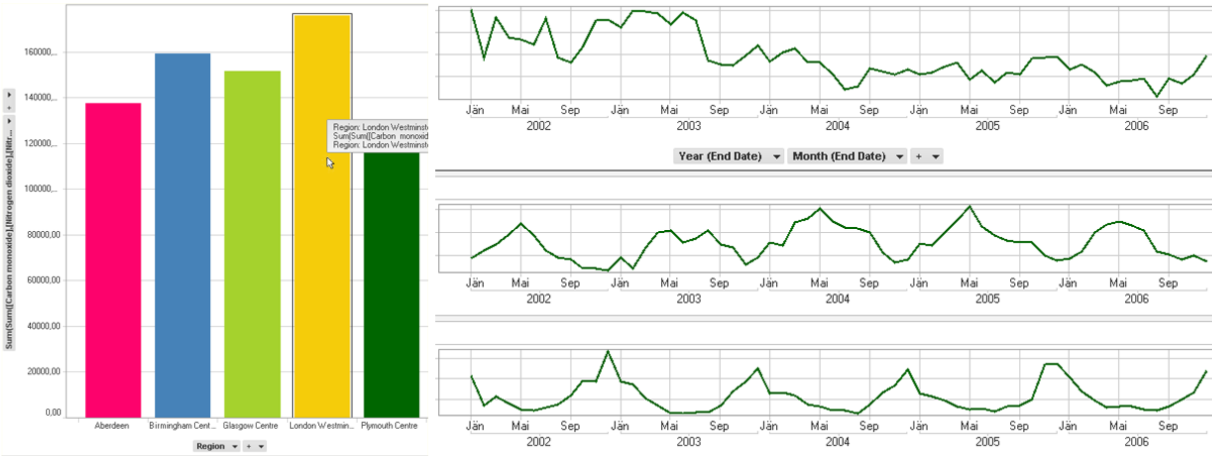


Figure 5.14: **Spotfire DXP: Tasks 1A and 2A.** *Each bar on the left-hand side represents the general pollution over the last five years for one region of Great Britain. The highest pollution values are reached in London, represented by the yellow bar, and the lowest values are measured in Plymouth, the green bar. The line plots on the right-hand side show the trends, from top to bottom, for Suphur dioxide, Ozone and Nitric oxide. Sulphur dioxide clearly got less in the last five years. Ozone always reaches the highest values in May, and Nitric oxide in December. (image courtesy: [15]).*

colored according to the region the values are measured in. Then the five-year-trend for each pollutant is discovered easily. In some cases the Region filter was applied to examine only one region at a time, as demonstrated in Figure 5.15.

5.3 Xmdv Tool

The Xmdv Tool is a freely available software package for the visualization of multivariate data. It is available on all major platforms, including Linux, Mac OS and Windows. The website offers a good introduction to its features and an online help that describes the user interface comes with the application. Only special file formats are supported, but conversion tools for comma-delimited files as well as sample data sets can be downloaded from the website.

5.3.1 Importing Data

When clicking on the ”Open...” menu point in the File menu the input dialog opens. Hence, an .okc file has to be chosen from the file system.

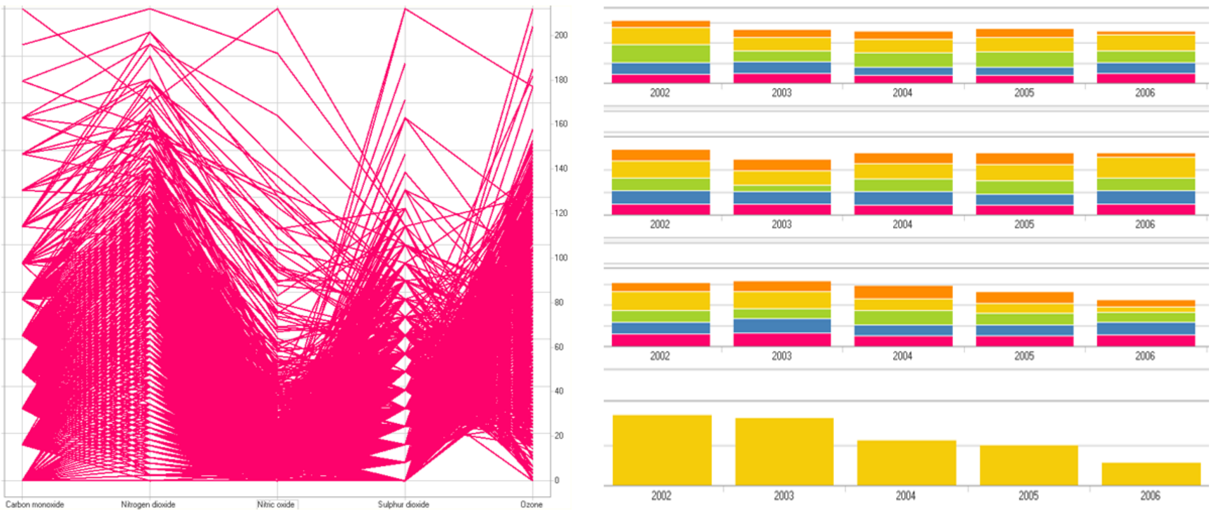


Figure 5.15: **Spotfire DXP: Tasks 3A and 4A.** *Task 3A can be solved by using the visualization of Figure 5.14 or by creating a parallel coordinates plot as shown on the left-hand side. Looking at the Nitrogen oxide and Nitrogen dioxide axes one can see that the connecting lines always have a negative slope, which means that high Nitrogen dioxide values induce also higher Nitrogen oxide values. The right-hand side displays a trend analysis, from top to bottom, of Nitrogen oxide, Nitrogen dioxide and Carbon monoxide. The bars are colored according to the region the value was measured in. Carbon monoxide could be reduced in general, especially in London, as the bottom right yellow bar plot is showing. Nitrogen oxide could be reduced slightly as well, and Nitrogen dioxide stayed about the same. (image courtesy: [15]).*

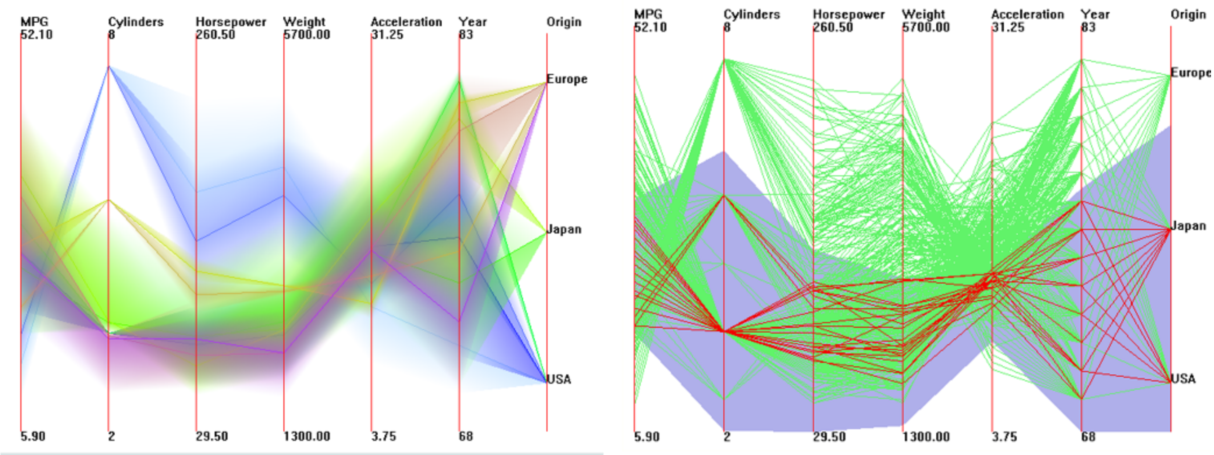


Figure 5.16: **Xmdv Tool: Parallel Coordinates.** *Above two example of a parallel plot visualization are shown. The right one is a conventional visualization with the grey area being the brush to highlight selected lines. A line is colored red if it lies completely inside this area. On the left-hand side, the hierarchical version of the same plot is shown. The lines now represent clusters of multiple lines with the range displayed as bands surrounding them. (image courtesy: [18]).*

5.3.2 User Interface Components

The main window of the Xmdv Tool holds all of the visualizations displaying the data set. The user chooses the visualization shown in it by clicking a button on the right side. Four well-known methods of displaying data are supported, each in a flat and hierarchical way: Scatterplots, Star Glyphs, Parallel Coordinates and Dimensional Stacking. Flat visualizations render one visual entity per data point. Hierarchical visualizations follow the principle of letting a visual entity represent a cluster of data points. Neighboring data points are subsumed in a cluster and organized in a hierarchical structure. Colors are mapped according to cluster proximity and assigned to each cluster. In this manner, relationships among clusters are emphasized. The range of a cluster is indicated by a band whose display is turned on or off by the user, as shown in Figure 5.16. Conventional flat approaches often do not scale well, whereas this approach eases the handling of large data sets.

Several other application windows provide control over the data set. It is possible to get an overview of the mean and variance of each dimension, to reorder the dimensions or reduce the number of dimensions. The positioning of glyphs can be customized, and the applied colors can be changed. A time series animation box lets the user visualize the temporal sequence of the records.

5.3.3 User Interaction

Interaction mechanisms include both standard methods, like panning and zooming, and less common techniques, like distortion. Highlighting a region in a view results in the same action in all other views. The highlighting process is often called brushing, and the connection to other view is referred to as linking.

Flat and hierarchical visualizations are explored by dimension zooming. This distortion operation scales up a subset of the data display while maintaining the context with surrounding data, a key characteristic of distortion techniques. This subset may then be examined as an independent data set. The user interface implements this function by following a mouse click by an enlargement of the clicked area. An example for distortion is also shown in Figure 5.17.

5.3.4 Special features

Navigational and filtering tools allow the user to navigate clusters in hierarchical visualizations until the desired region and level of detail is reached. Therefore, interactive data exploration is made possible. *Structure-based brushing* uses the metaphor of a triangle for the filtering process. By selecting a wedge of the triangle or moving a contour line data points are included or excluded, as shown in Figure 5.17. To ease the mental connection between the main visualiza-

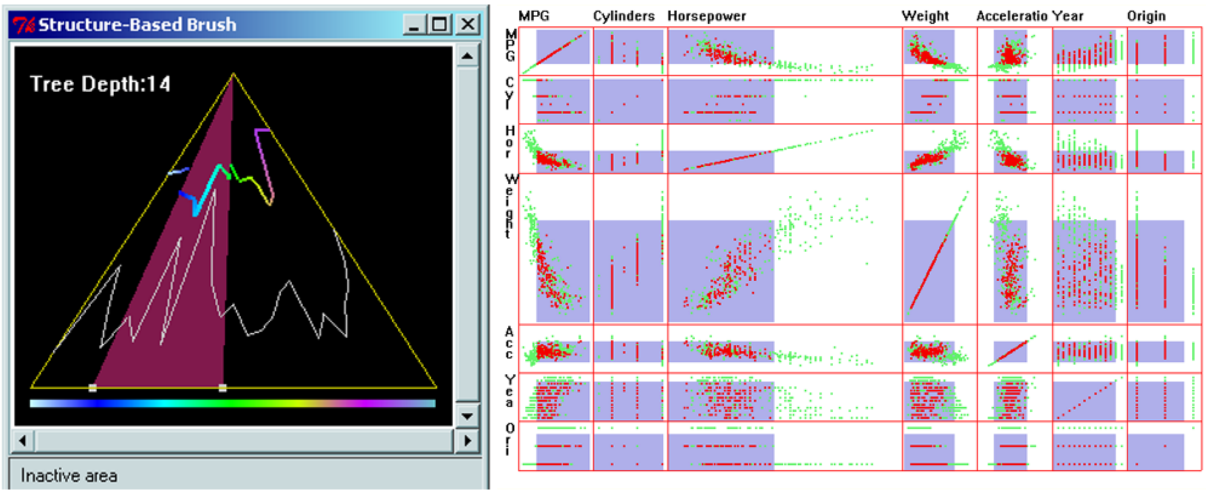


Figure 5.17: **Xmdv Tool: Brushing and Distortion.** *By changing the size of the brushing triangle on the left-hand side data points are filtered out. The contour line in the triangle represents the cluster hierarchy level that is displayed. Color associations are the same in the visualization and in the brush window. The right-hand side shows a flat scatterplot with highlighted red points. It is distorted to focus on a specific region of the visualization. (image courtesy: [18]).*

tion and the structure-based brush window, colors of the contour line correspond to the color of the data points in the main window. Elements within the brushing subset may be examined at a different level of detail or magnified and examined in full view. Brushed and unbrushed regions can be faded out according to the focus of the user. This technique is called dynamic masking. Flat displays can be brushed too.

5.3.5 Evaluation

The Xmdv tool needs a special input format that contains the dimensions of each variable, the number of datapoints and some other describing values in the first lines of the input file. The website provides a conversion tool to convert csv files to this format, but when trying to apply this tool to the air pollution only three columns instead of eight got read in. Maybe the empty entries on the read-in file caused it to be not readable. Therefore, the file had to be arranged manually. The values in the columns must be float numbers, so some additional manual changes including changing the string region values to numbers had to be made to interpret region and date values correctly. No date format is supported, so only the year values were used. Instead of the empty entries in the read in file -1 was filled in.

During the work, the program crashed several times, presumably due to the large amount of data. One main problem for the carried out tasks was the lack of any possibility to apply aggregation functions to the values. The possibility to eliminate currently not needed parallel coordinate axes was helpful and therefore used often. The relations between the variables are not that easy to see as when using other tools, but for the problem of getting to know the structure of a data set it is a good tool. If a real and therefore big data set is used, other tools may result more appropriate.

The concept of hierarchical clustering and the connected interaction metaphors are not quite easy to understand at the first glance, but a powerful tool to handle large data sets. The Xmdv Tool was the only application that had this real abstraction level in a visualization.

Low-level tasks

Tasks 1L,2L,3L,5L and 6L could not be carried out because only the year numbers could be read into the application. The second possibility would be to use a sequential number for each point in time, but then every time a date is looked up the corresponding number would need to be calculated. Figure 5.18 shows the two low-level tasks accomplished.

1L - Existence - Was a Nitrogen dioxide value measured on the 17th of May 2006 in Birmingham?
Not carried out.

2L - Location - At which points in time was a value of 19 measured for Sulphur dioxide?
Not carried out.

3L - Time interval - Which temporal range is covered by the Ozone measurements in Plymouth?
Not carried out.

4L - Temporal texture - How often reached the Ozone pollution a value higher than 80 in 2004?
Brushing leads to the result that 39 points in time had 2004 higher Ozone values than 80 which is not correct. Spotfire and Tableau detected 54 points.

5L - Rate of change - How much did the Sulphur dioxide pollution change between January and July 2003?
Not carried out.

6L - Sequence - In Glasgow, was an Ozone measurement of 10 made before or after the value of 170 of Nitroc oxide on the 8th of december 2003?
Not carried out.

7L - Synchronization - Was a value of 3 of Nitric oxide ever measured together with a value

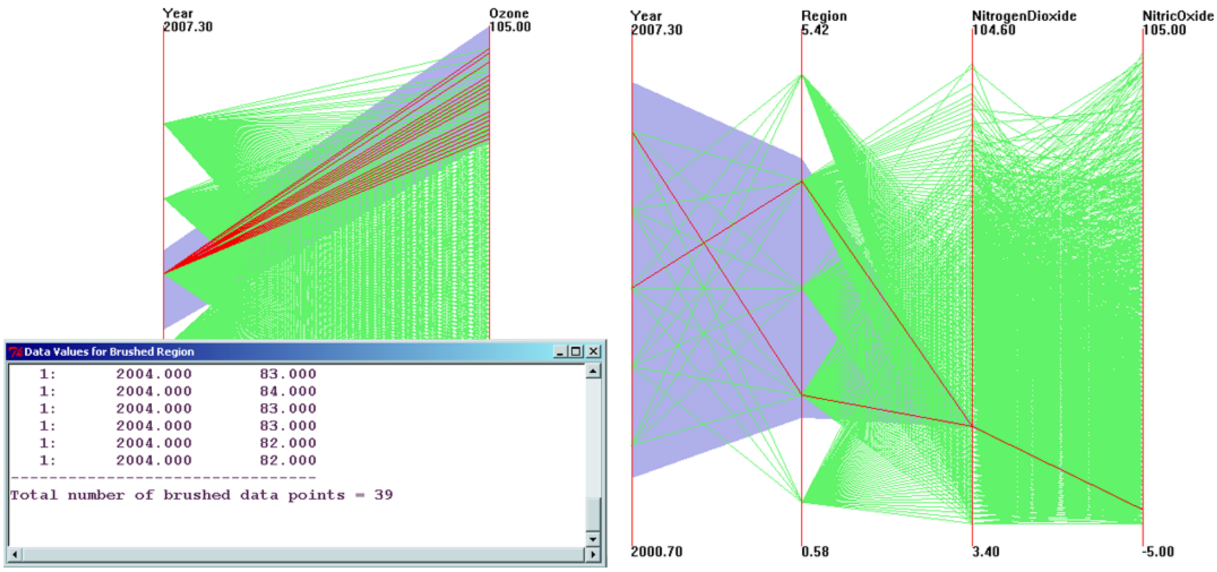


Figure 5.18: **Xmdv Tool: Tasks 4L and 7L.** *On the left-hand side the values higher than 80 of the Ozone pollutant are shown. The result of 39 such points is not correct as proven with other information visualization tools. Task 7L, on the right-hand side, could be accomplished successfully and the two points in time where a Nitric oxide value of 3 and a Nitrogen dioxide value of 27 occur were found. The parallel coordinate filtering by brushing here is not the ideal way of eliminating unwanted items because only one point in time is needed and a filter with a text field would be more helpful. (image courtesy: [18]).*

of 27 of Nitrogen dioxide?
This time, the correct result is retrieved, the coincidence of these two values occurred twice in the last five years.

Basic tasks

1B - What range has the Carbon monoxide pollution?
Looking at the parallel coordinates it is found out that the Carbon monoxide range starts at a value of 0 which is correct, but the upper lines are not drawn correctly. The highest line should meet the Carbon monoxide axis at a value of 3.6, but only a value of about 1.6 is reached. The reason for this could not be found out clearly. A possible explanation would be the clustering that the Xmdv tool employs which possibly eliminates outliers. Figure 5.19 illustrates this task and the tasks 2 and 4.

2B - What was the highest Sulphur dioxide value measured in 2002?
This can be found out simply by brushing in the parallel coordinates or the scatter plot visualization. The exact value is determined by moving the mouse cursor over the desired line and

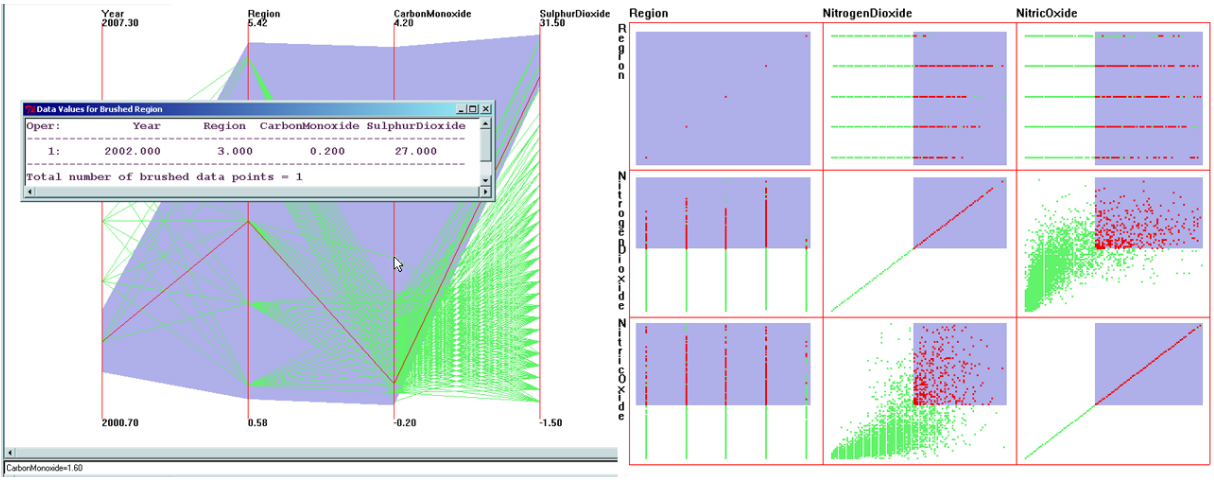


Figure 5.19: **Xmdv Tool: Tasks 1B,2B and 4B.** *Brushing leads to the highest Sulphur dioxide value measured in 2004, as shown on the left-hand side. The highest Carbon monoxide value intersecting the parallel coordinate axis - 1.60 - is determined by moving the mouse cursor over it. This value is not correct, maybe because of Xmdv's clustering processes. The scatter plots on the right-hand side illustrate that high Nitrogen dioxide values mostly come along with high Nitric oxide values. (image courtesy: [18]).*

looking at the data point information at the bottom of the window, or by opening a window that provides all the measured values of this data point.

3B - Which measurement site had the highest Nitric oxide pollution in 2004?
The calculation of a sum is not possible when using the Xmdv tool, so this task could not be solved correctly. What can be done is the discovery of regions with high Nitric oxide measurements in 2004. After a short brushing process the relevant regions are clearly visible both in the parallel coordinates and the scatter plot visualization.

4B - Are there regions with both high Nitric oxide and Nitrogen dioxide values at the same time?
Both the scatter plot and the parallel coordinates visualization show that every region has high values of both measurements at the same time.

5B - Which year had the highest ozone pollution in London?
This task was not possible to be carried out without calculating an aggregated sum value.

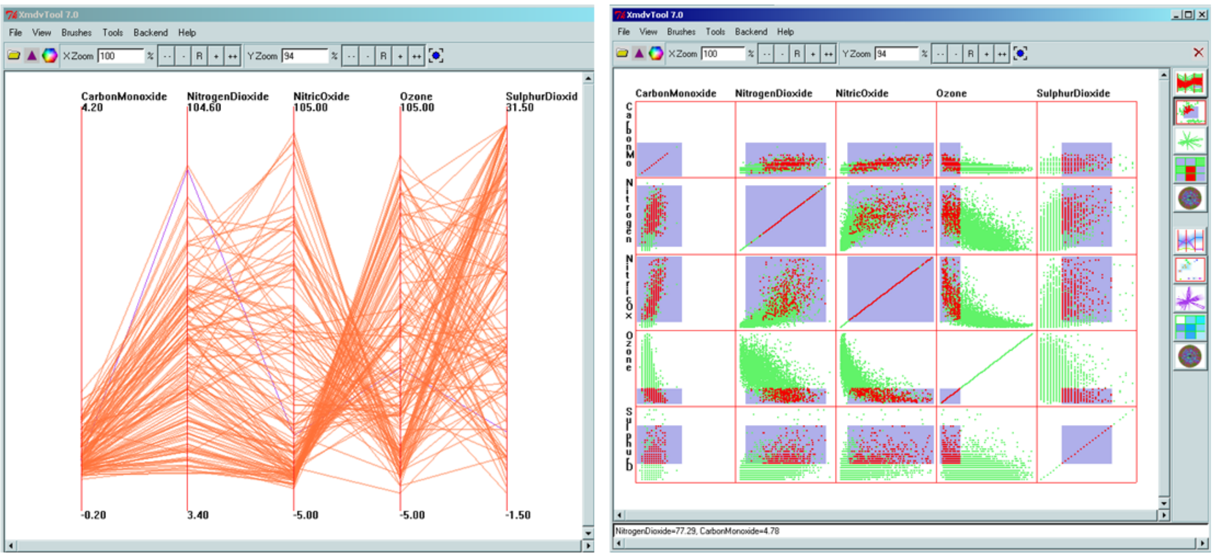


Figure 5.20: **Xmdv Tool: Task 3A.** *Correlations are determined by using the hierarchical parallel coordinate plot and the scatter plot visualization. Crossing lines like those of Nitric Oxide and Ozone indicate an indirect proportional correlation. In the scatter plot visualization on the right-hand side the data points with low Ozone values are selected. Looking at the Nitric Oxide plots, the user sees that mostly high values are highlighted. (image courtesy: [18]).*

Advanced tasks

Tasks 1A, 2A and 4A were not possible to be carried out without using an aggregation function.

3A - Are there any correlations between pollutants? Does the concentration of all of them increase together, or are there pollutants that are independent of the others?
For this task, the hierarchical parallel coordinate plot visualization was used to find correlations. Rearrangement of parallel coordinate axes and filtering out irrelevant lines leads to the looked for result, as shown in Figure 5.20. It is found out that Ozone reaches high values when Nitrogen dioxide, Nitrogen monoxide and Carbon monoxide have low values and vice versa. Sulphur dioxide seems to have no correlations. Because of the impossibility of reading in the dates the relationship to the seasons is lost and the user has no ability to find out at which season a pollutant has high values.

5.4 ILOG Discovery

The visualization tool ILOG Discovery provides access to databases and enables the user to analyze data sets intuitively. A wide variety of views, including parallel coordinates, treemaps, scatter plots, histograms and customized combinations of these views is produced with a few manipulations. When opening a data set the first time, overview visualizations like parallel coordinates and distributed histograms are used to get a first overall glimpse of its distribution and inherent correlations. In parallel plots, correlations are found when lines that join two columns appear to be quite parallels. The distributed histogram visualization distributes all attributes along the horizontal axis according to a main variable's value which is useful for temporal data sets. To focus on the distribution according to two variables, scatter plots are used. Gantt charts ease the focusing on time-dependent information. A detailed user's manual comes with the application and supported input covers delimited text files, text files saved in Excel and Access and the special feature of importing directory information like file names, usage and size directly via opening a directory. Sample data files that help the user to get acquainted with the tool are provided as well.

5.4.1 Importing Data

After opening a table from a file, the columns of the table are shown in the *table inspector* at the top of the main window - see also Figure 5.21. Each column component consists of a range slider to filter out items, information on which visual attribute the column is mapped which can be changed here too and the current value of the highlighted record. By clicking and dragging convenient reordering of columns is possible.

5.4.2 User Interface Components

The *projection inspector* on the right-hand side of the main window allows the user to control the parameters of the visualization in accurate manner. For example, unwanted data columns can be filtered out or the distribution of a column can be chosen to be linear or logarithmic. Associations between data and visual attributes can be changed here too, for example the mapping of a data value to size or color. If the user wants to view the records sorted according to the values of one attribute he/she chooses this attribute in the projection inspector. The *main view* holds one or more visualizations of the data set. The visualizations are dynamically linked, which means highlighting in one view results in highlighted corresponding items in the other views. If a view is not useful at the moment but is needed later on it can be kept as a so-called thumbnail at the bottom of the main window.

5.4.3 User Interaction

Moving the cursor over a visualization element causes it to get highlighted in yellow. If the user clicks on it, it is selected and therefore colored in yellow. Pressing Ctrl or Shift while selecting allows the user to select more than one item. Selections are stored using marks which can be

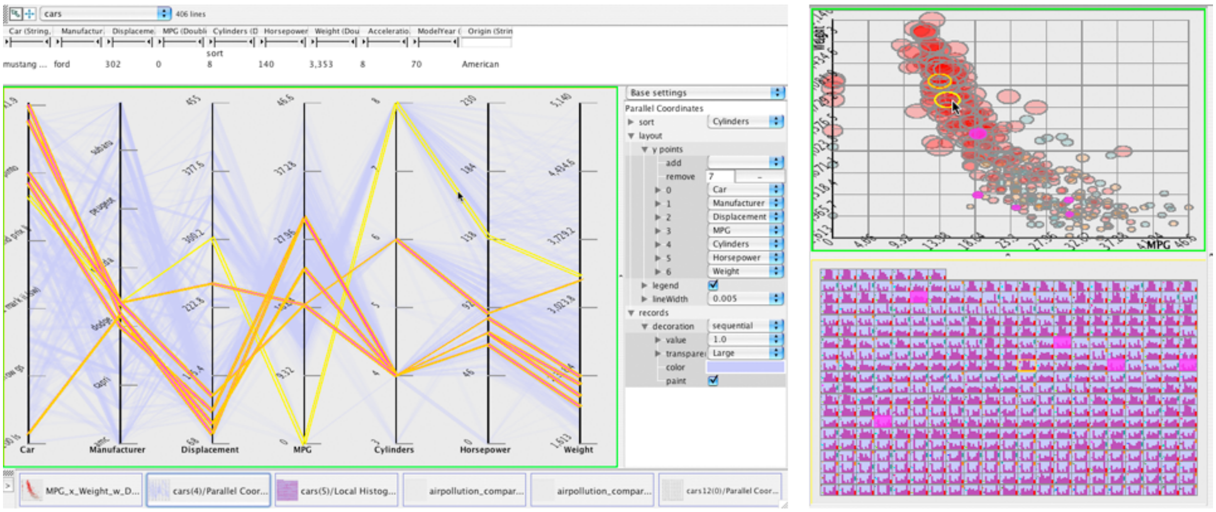


Figure 5.21: **ILOG Discovery.** *The Discovery interface contains a variety of useful tools: at the top all the data columns are shown. By adapting sliders the range of a column's value is shortened. The value of the currently highlighted record, seen here as the yellow line, is displayed too. On the right-hand side the user is enabled to add/remove columns and to change the sorting by using the functions of the projection inspector. At the bottom currently iconified views are displayed for later use. The orange lines indicate selected records. After a selection, these items can be marked. Marked items are linked over all visualizations and shown in all of them in magenta as in the view on the right-hand side. (image courtesy: [6]).*

used to create a subtable and examine it in a different view. Zooming and panning are supported as well.

5.4.4 Special Features

ILOG Discovery enables the user to share a project on an intranet or the internet. In this process a html file that contains an applet that references this project is created, with some editing limitations defined by the publisher.

5.4.5 Evaluation

ILOG offers both a component framework for the development of customized visualization applications and a visualization tool called Discovery. The location of this tool on the website was a bit hidden, but the following downloading process was easy. The trial length of 3 months can be renewed as long as it is not available for sale.

The most cumbersome part was the process of getting the data into the visualization. Direct import did not work so the file had to be opened in Microsoft Excel to make some

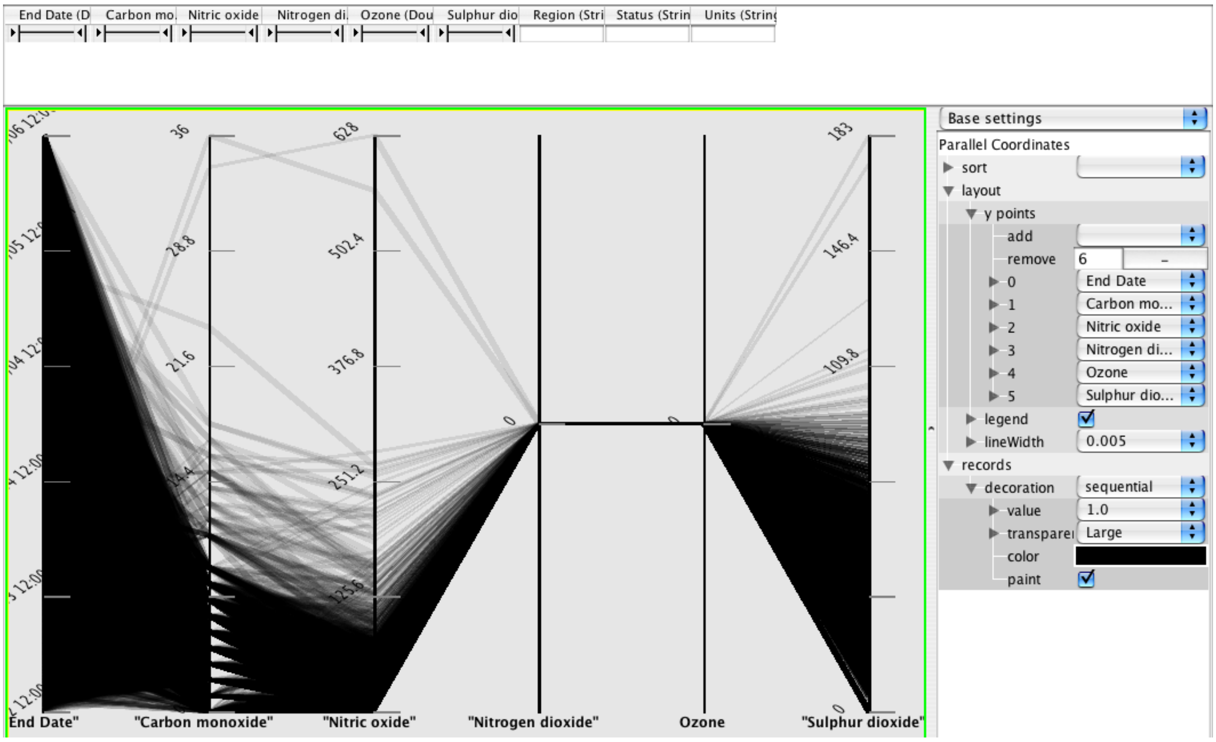


Figure 5.22: **ILOG Discovery: Task 1B.** *After applying some changes to the input file the read in worked but some columns were set to 0. Because of this a reduced number of columns was used in the task accomplishment later on. The range of the Carbon monoxide pollution, discovered by looking at the corresponding axis, seems to be 0 to 36 instead of 0 to 3.6. At this state of work the number read in did not work correctly yet because a locale token had to be added to each number column to interpret each number the right way. (image courtesy: [6]).*

changes to the file and save it as a text file. This process is described in the user's manual. Then the import worked, but every time the whole file was used two of the columns were set to 0 although there were values in the input file, as seen in Figure 5.22. Using a file containing a reduced number of columns resulted in correct read-in values.

The working process was a bit difficult and slow because it seems that ILOG Discovery is not aimed at handling such large input files. The example files coming with the application have a length of about 400 lines compared to the length used in this thesis of 9000 lines. Working with this file resulted in non-responding behavior and the need to force the application to quit several times. Due to this even simple tasks took half an hour to complete because of the needed restarts. The only interaction possible was moving the cursor over the plot and look at the highlighted line and its values. Trying to use the column filters at the top of the user interface caused the application to be stuck. Therefore, only the first two basic tasks were carried out.

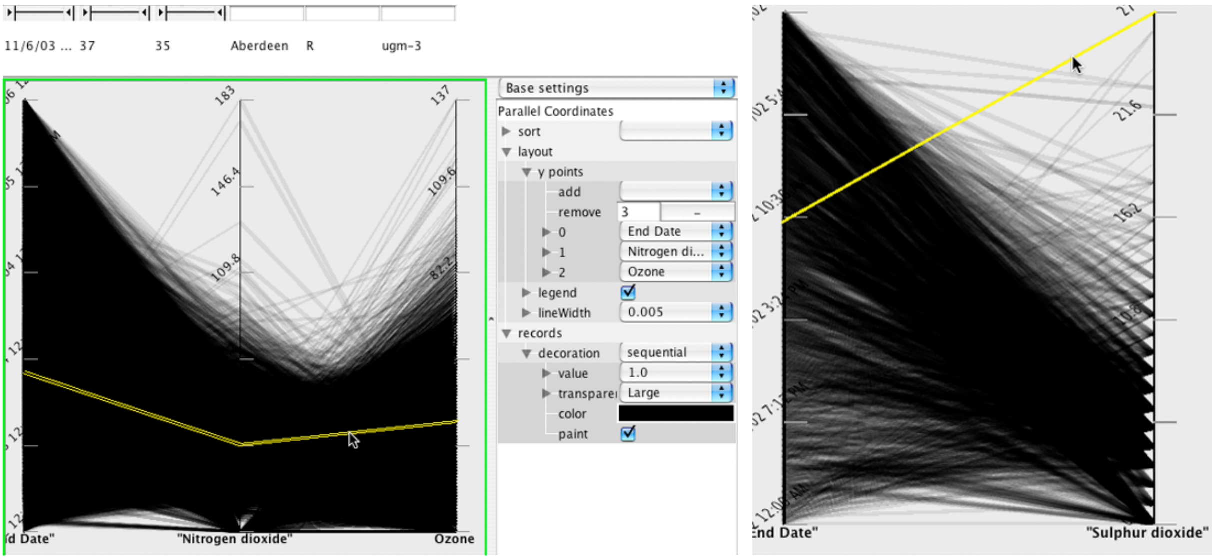


Figure 5.23: **ILOG Discovery: Task 2B.** *Using a reduced number of columns the input file was read in correctly, as seen on the left-hand side. To find out the highest Sulphur dioxide value in 2002 the rows containing other years had to be deleted using Excel because ILOG Discovery had problems managing such a large file. Then, the searched for value was discovered by looking for the line meeting the highest point on the Sulphur dioxide axis. (image courtesy: [6]).*

Basic tasks

1B - What range has the Carbon monoxide pollution?
The values read in were not interpreted the right way first, e.g. 3.6 was interpreted as 36. Looking up the input section of the user's manual revealed that the correct locale has to be set in the input file to overcome this problem. After that, the solution to this first task, namely 0 to 3.6, was easy to see in the parallel coordinates visualization.

2B - What was the highest Sulphur dioxide value measured in 2002?
The simplest way to solve this task would be to use the range sliders to eliminate all years but 2002 and looking for the highest measured value. But trying to do this caused the program to crash and even with only the variables Sulphur dioxide and End Date in the input file it behaved very slowly. Therefore, the only way to accomplish this task was to filter out all years but 2002 with Excel, reading this file in and looking at the highest value drawn in the parallel coordinates. This visualization is also depicted in Figure 5.23.

ILOG Discovery had problems handling the large input file used in the task accomplishment. Working with this file resulted in non-responding behavior and even simple tasks took a long time to complete. Trying to use the column filters at the top of the user interface caused the application to be stuck. Therefore, the other tasks were not carried out.

5.5 CViz

CViz [2] is IBM's visualization tool for the analysis of multivariate data sets. It is available for all major platforms. After reading in data, CViz builds clusters of the input data and assigns each cluster a different color in a scatter plot. The cluster centroids are determined and displayed as icons, using circular icons for primary centroids. Through these primary centroids, axes are drawn. Four types of scatter plot are possible: linear, non-linear, SVD-Centroids and simple. They are distinguished by the number of primary centroids they are based on. The linear plot is based on three centroids, while non-linear is based on two. The SVD plot can be based on any number of centroids greater than two, and the simple plot draws the data along two selected dimensions.

5.5.1 Importing Data

Input files should have the common csv format to be readable by this application. One specially designated column labeled "class" may contain the class values as strings. In the input dialog the filename to be read in is chosen. The user has the possibility to reduce the size of the displayed data by choosing which attributes should be read in. CViz then builds clusters of the input data and highlights them in a scatter plot.

5.5.2 User Interface Components

The *main window* holds a scatter plot. The user is able to choose which type of scatter plot is displayed by selecting from a drop down box. The *non-linear plot* is based upon the distance of each data record to the two primary cluster centroids. This results in the fact that some points in the built-up coordinate system will never contain a data point. The origin, for example, would mean that a data point has zero distance from both centroids which is impossible. In the *linear plot*, each data point is projected onto the plane formed by the three cluster centroids the plot is based on. Therefore, the positions relative to the centroids remain correct, but the plotted positions do not have to be the same as in the original attribute space. When the user selects the *SVD plot*, he/she chooses which clusters should be the basis of the scatter plot. Since a 2D plot lacks the information about which clusters are relatively close to each other in the original attribute space, similarity lines can be drawn to reveal neighboring clusters, as shown in Figure 5.24. The threshold at which clusters will be connected is controlled by the user with a slider. Similar clusters can be joined.

To get more information about each cluster, the cluster statistics window is used. There,

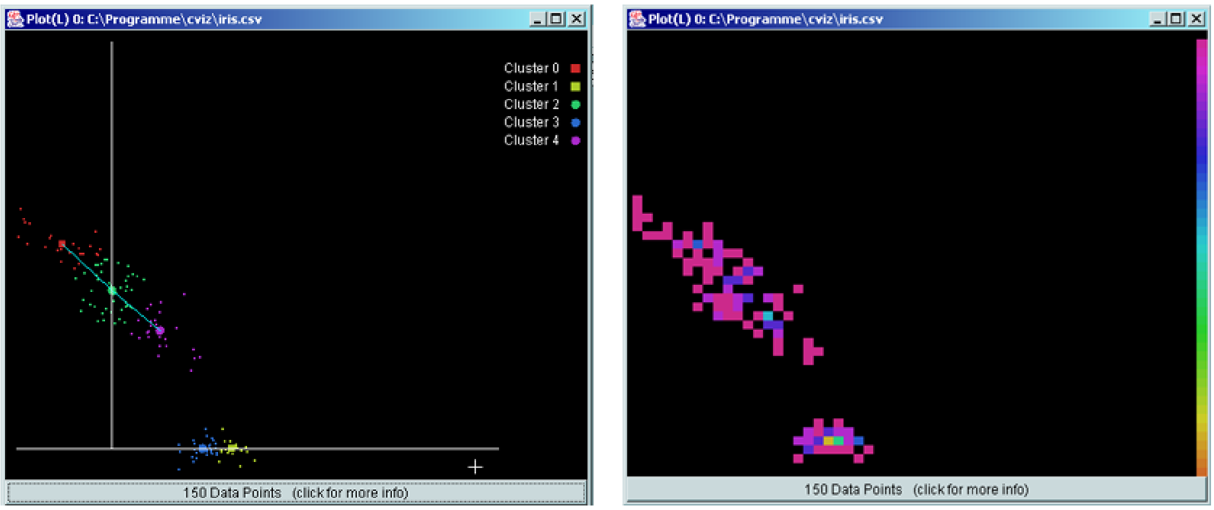


Figure 5.24: **CViz: Scatter plots.** *The main visualization method used in IBM's visualization tool CViz is the scatter plot. Therein clusters are built corresponding to the similarity of data points and emphasized by a specific color. Neighboring clusters are illustrated by cluster connecting lines. The left-hand side shows a density plot which is colored according to the number of points drawn there in the original scatter plot. (image courtesy: [2]).*

the user sees two bars for each attribute, one showing the mean for the whole plot, the other showing the mean in the selected cluster - see also Figure 5.25. Cluster names can be changed in this window according to their characteristics. The distribution of classes within clusters is shown in the class statistics window.

5.5.3 User Interaction

An animation technique called touring is used to help the analyst to quickly view different perspectives on the data and therefore ease the process of exploration. During the animation, the primary centroids are changed in each view, and the clusters move smoothly from one view to the next. Major tour steps can be viewed in a separate window, and the best perspective on the data is determined using the information retention value of each step. Zooming in CViz is the creation of a new scatterplot by excluding a number of data points. By clicking on a cluster centroid, the scatter plot zooms on only that cluster. The possibility of only showing one class or excluding a specific class is given too. If the user wants to zoom in on selected points, he/she drags the mouse over these points in a brushing process.

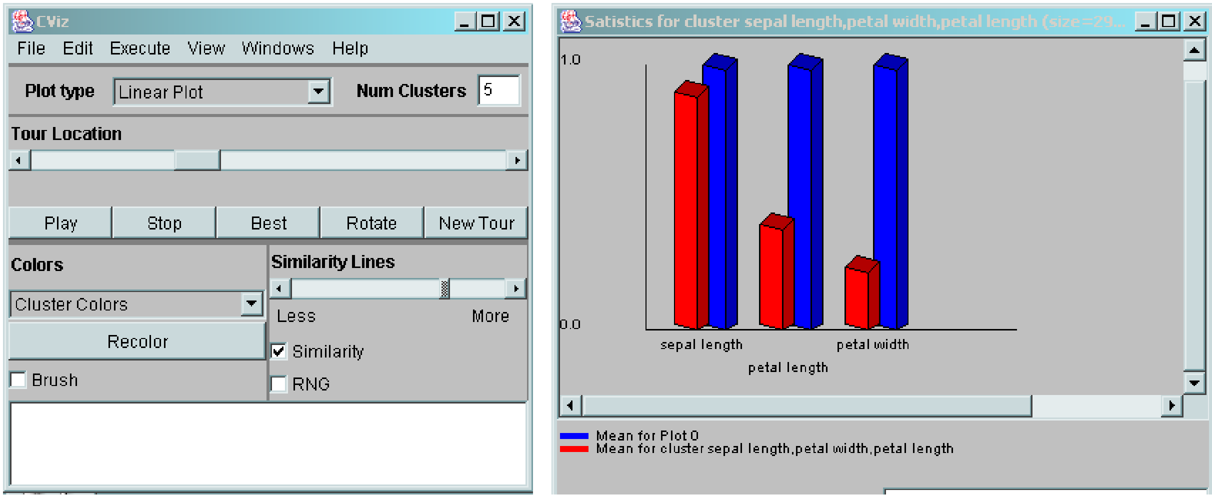


Figure 5.25: **CViz: Interface.** *The CViz interface contains a variety of tools. The user chooses from a number of different plots, explores the visualization by brushing and zooming and views an animation, the so-called tour, to view the data set from different perspectives. The cluster statistics window on the right side is used to obtain more information about each cluster. The colored bars depict the mean for an attribute for the whole plot and in comparison only in the selected cluster. The cluster viewed here has relatively low petal width and lengths and could be named "low petal width and length" in the cluster name text input field. (image courtesy: [2]).*

5.5.4 Special Features

CViz provides a feature to find abnormal data points, so-called outliers. By clicking on a check-box they are highlighted with a circle drawn around them. If a data set is so large that it overwhelms the display resources of the computer, a sampling function is included to reduce the number of points displayed. Regions containing a high density of data points can be magnified in a separate window. The magnification square is moved by the user to interesting points. To overcome the problem of overlapping, a density plot that colors regions according to the number of points drawn there in the original scatter plot is used.

5.5.5 Evaluation

In the first try to read in the csv data file a java exception was thrown and nothing was read in. A possible cause for this would be empty values due to missing measurement values. Therefore, the input file was reformatted to contain values of -1 for missing values. Even then the input of the whole file did not work. CViz has a function to read in only a specified number of lines. Using this function, it was found out that the input of up to 6000 lines was possible but going over that range led to the previously mentioned java exception. The support for CViz was contacted twice but no answer was received. Due to these input problems it was not possible to carry out the defined tasks.

5.6 Summary

The following table contains short information about the tools' differences. The tasks accomplished and their degree of completeness are mentioned. Problems are described shortly to facilitate an at-a-glance comparison of the information visualization applications.

Low-Level Tasks					
	Tool	**accomplished**	**Completeness**	**Easiness**	**Problems**
Task 1L	Tableau	yes	100%	easy	read in to Excel first
	Spotfire	yes	100%	easy	xls read in did not work, csv without problems
	Xmdv Tool	no	0%	-	conversion tool did not work
	Discovery	no	0%	-	program runs too slow
	CViz	no	0%	-	input did not work
Task 2L	Tableau	yes	100%	easy	-
	Spotfire	yes	wrong result	easy	overlapping scatterplot points
	Xmdv Tool	no	0%	-	no date format support
	Discovery	no	0%	-	program runs too slow
	CViz	no	0%	-	input did not work
Task 3L	Tableau	yes	100%	easy	-
	Spotfire	yes	100%	easy	-
	Xmdv Tool	no	0%	-	no date format support
	Discovery	no	0%	-	program runs too slow
	CViz	no	0%	-	input did not work
Task 4L	Tableau	yes	100%	easy	-
	Spotfire	yes	100%	easy	-
	Xmdv Tool	yes	wrong result	easy	incorrect number of brushed points
	Discovery	no	0%	-	program runs too slow
	CViz	no	0%	-	input did not work
Task 5L	Tableau	yes	100%	easy	-
	Spotfire	yes	100%	easy	-
	Xmdv Tool	no	0%	-	no date format support
	Discovery	no	0%	-	program runs too slow
	CViz	no	0%	-	input did not work

	Tool	accomplished	Completeness	Easiness	Problems
Task 6L	Tableau	yes	100%	easy	-
	Spotfire	yes	100%	easy	-
	Xmdv Tool	no	0%	-	no date format support
	Discovery	no	0%	-	program runs too slow
	CViz	no	0%	-	input did not work
Task 7L	Tableau	yes	100%	easy	-
	Spotfire	yes	100%	easy	overlapping points at first irritating
	Xmdv Tool	yes	100%	easy	-
	Discovery	no	0%	-	program runs too slow
	CViz	no	0%	-	input did not work

Basic Tasks					
	Tool	**accomplished**	**Completeness**	**Easiness**	**Problems**
Task 1B	Tableau	yes	100%	easy	-
	Spotfire	yes	100%	very easy	-
	Xmdv Tool	yes	50%	easy	wrong upper margin displayed
	Discovery	yes	100%	easy	locale had to be set for correct read-in, program runs slowly
	CViz	no	0%	-	input did not work
Task 2B	Tableau	yes	100%	easy	-
	Spotfire	yes	100%	easy	-
	Xmdv Tool	yes	100%	easy	-
	Discovery	yes	100%	middle	use of filters causes program to crash
	CViz	no	0%	-	input did not work
Task 3B	Tableau	yes	100%	easy	-
	Spotfire	yes	100%	middle	bar plot had to be used to build sum
	Xmdv Tool	no	30%	easy	no aggregation function, but regions with high values can be found
	Discovery	no	0%	-	program runs too slow
	CViz	no	0%	-	input did not work
Task 4B	Tableau	yes	100%	easy	-
	Spotfire	yes	100%	middle	filter adjustment was forgotten first
	Xmdv Tool	yes	100%	easy	-
	Discovery	no	0%	-	program runs too slow
	CViz	no	0%	-	input did not work
Task 5B	Tableau	yes	100%	easy	-
	Spotfire	yes	100%	easy	-
	Xmdv Tool	no	0%	-	no sum aggregation
	Discovery	no	0%	-	program runs too slow
	CViz	no	0%	-	input did not work

Advanced Tasks					
	Tool	**accomplished**	**Completeness**	**Easiness**	**Problems**
Task 1A	Tableau	yes	100%	middle	use of ”all measurement”-variables not per se obvious
	Spotfire	yes	90%	middle	though not explicitly needed: coloring per pollutant not possible
	Xmdv Tool	no	0%	-	no sum aggregation
	Discovery	no	0%	-	program runs too slow
	CViz	no	0%	-	input did not work
Task 2A	Tableau	yes	100%	easy	-
	Spotfire	yes	100%	easy	-
	Xmdv Tool	no	0%	-	no sum aggregation
	Discovery	no	0%	-	program runs too slow
	CViz	no	0%	-	input did not work
Task 3A	Tableau	yes	100%	easy	-
	Spotfire	yes	100%	easy	-
	Xmdv Tool	partly	50%	middle	no date format support
	Discovery	no	0%	-	program runs too slow
	CViz	no	0%	-	input did not work
Task 4A	Tableau	yes	100%	easy	-
	Spotfire	yes	100%	easy	-
	Xmdv Tool	no	0%	-	no sum aggregation
	Discovery	no	0%	-	program runs too slow
	CViz	no	0%	-	input did not work

6 The Visual Programming Tools

Tools for the creation of visualization networks by combining predefined modules are examined in this chapter. The user interface elements are described along with the possible interactions. Due to the need to create a new visual program for each new task one task was chosen to be carried out with all of the applications.

6.1 OpenDX

OpenDX [11] is the open source software version of IBM's Visualization Data Explorer, an environment for the creation of information visualization applications, called visual programs. By combining several predefined modules the user is enabled to construct visualization processes. By using the right module, both 2D and 3D visualizations can be created. The former include contour lines, vectors and 2D glyphs, the latter comprise isosurfaces, slices, streamlines and 3D glyphs. The customization and new creation of modules is possible too. In the basic visualization process, data are read in, then they are modified by flowing through one module after another. Some modules add new components, others change or remove components of the data. In the last step, the data are rendered including the calculation of lights and shadows and displayed in the Image Window. OpenDX is designed as a client-server model, the graphical user interface being the client and the used data and modules making up the server part. Supported data formats comprise hdf, cdf, array-formatted or delimited text files and images. A detailed user's manual and quick start guide come with the application download and it is available for many platforms, including Windows, MacOS X, various Linux distributions and Solaris.

6.1.1 User interface Components

OpenDX[11] provides four main windows: the Visual Program Editor, several Control Panels, the Image Window and the Data Prompter. They are described in the following sections and two of them are shown in Figure 6.1.

The *Data Prompter* is used to import data and provides several general purpose visualization programs which can visualize a variety of data types. Therefore, the user is able to get a quick look of his data without having to construct a visual program by himself.

The *Visual Program Editor* is divided in two basic sections. On the left, the available modules are displayed as a list of items organized in categories. The so-called canvas makes

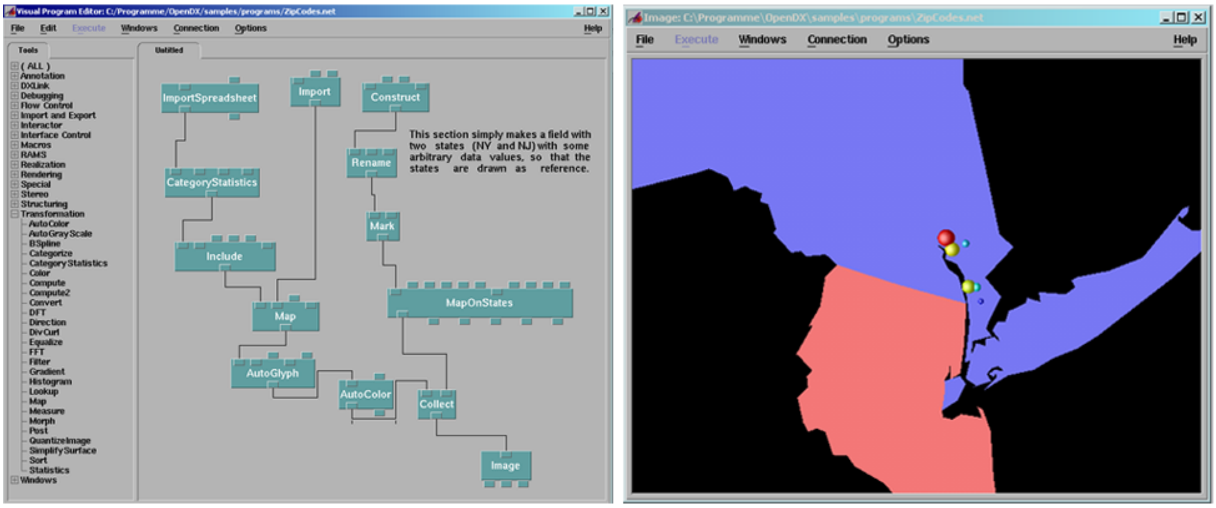


Figure 6.1: **OpenDX: User interface.** *Above the two main windows of the information visualization environment OpenDX are shown. The Visual Program Editor is used to build and execute customized visualization programs. They are rendered in the Image Window which is shown on the right side. Several mechanisms for user interaction like panning and zooming ease the exploration of the visualized data. (image courtesy: [11]).*

up the main part of the window. The visualization network consisting of modules and their connections is shown in it. Tabs at the top and bottom of the modules represent inputs to and outputs from it. By dragging a module from the tool menu it is positioned on the canvas. Connections between modules are established by clicking an output tab and dragging to an input tab of another module. Double-clicking a module opens its configuration dialog box where its parameters are set.

After the start of the module execution the modules are processed and the result is displayed in the *Image Window*. It provides a number of user interaction mechanisms to explore the visualization. The viewing angle can be changed by selecting from seven viewing directions, e.g. top and left. By clicking on a set of small coordinate axes the visualized object is rotated. The rotation is either recalculated after the mouse button release or after each change. In the latter case, the image is replaced by a dot representation of the object, and redisplayed after releasing the mouse button. Panning and zooming are possible too.

Control Panels allow the user to control the inputs to a visual program directly. For example, the colormap editor is used to change the colors applied to the visualized object. Another window, the sequencer, allows the user to control and change animations.

6.1.2 Evaluation

Before starting Open-DX, a X-Server program, namely X-Win32, was installed because Open-DX needs an X-Server running. Although according to the Readme file the X-Server should start automatically when starting Open-DX, X-Win32 did not start. Therefore, X-Win32 was started manually before running Open-DX. The input process led again to problems: missing values in the input file are not supported and the last column has to contain a value to prevent errors. The value -1 was filled in again for missing data values. Like many other applications, Open-DX did not support a date format, so year numbers were used again. Due to the fact that it took about five hours of practice to get one simple example running Open-DX is considered not suitable if the focus lies on the obtainment of answers about a dataset. Especially persons not coming from a technical background may have problems.

Task accomplishment

To find the highest Sulphur dioxide value measured in 2002, a module design similar to an example provided with the software which displays a line plot was used. The required modules were combined and some adaptations were made to fit to the current task. The vertical axis of the line plot was adjusted to show the maximum of the Sulphur dioxide values measured in that year. Furthermore, the horizontal axis labeling was set to a list of year strings. Figure 6.2 shows the module design and the visualization.

6.2 AVS/Express

AVS/Express is a visual programming tool for the display of corporate and scientific data. It offers wizards for the two main tasks of data import and data visualization. Using the wizards, the user only has to choose the dimensionality of the viewing window, the type of the application, the input file and some parameters of the visualization concerning e.g. bounding options and legend settings. Then the visualization viewer is displayed. It can display both two- and three-dimensional data. The second possibility of creating a visual program is the use of the network editor. By combining visual modules and connecting them, the application is created. In the visualization viewer, the user chooses the properties associated with the employed modules and interacts with the created visualization.

AVS/Express is available in two editions: the Developers and the Visualization Edition. These editions share a common architecture and allow the development of new visual modules. The Developers Edition is designed for developers creating applications and generally exposes more of the underlying system architecture. Due to its orientation to end users, some of the functions provided in the Developers Edition are not available in the Visualization Edition.

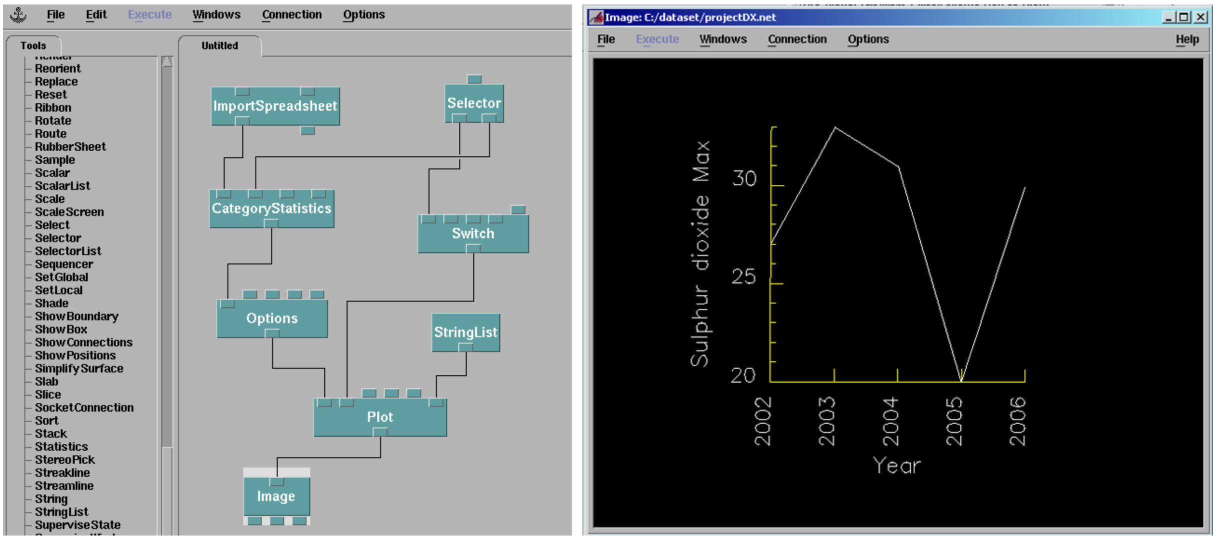


Figure 6.2: **Open-DX: Task 2B.** *On the left-hand side the module design for this task is displayed. The CategoryStatistics module calculates the maximum of the Sulphur dioxide values for every year and forwards the result to the Options module. The StringList module contains the axis labels for the horizontal axis. With the selector module other statistical functions than the maximum can be chosen to be displayed. The right side shows the line plot created by the application. The user sees clearly that 27 was the highest Sulphur dioxide values measured in 2002. (image courtesy: [11]).*

Still, some development on high-level modules is possible. AVS/Express is available on all major platforms including Windows, Linux, Macintosh and Sun and comes with a detailed user's manual and quick start guide. In addition to it, Advanced Visual Systems offers both introductory and advanced tutorials for AVS/Express. Supported input formats include ASCII files with column data, image data, some formats specific to AVS and several databases using a special database kit.

6.2.1 User Interface Components

In the *Network Editor* the visual program is created by combining predefined or newly developed modules. It contains the available modules in the library at its top and an application container positioned below, as shown in Figure 6.3. The user selects the desired module from the library and inserts it into the application container. Doing so, a module object is instantiated and made available for use within a visualization network. Each module has input and output ports represented by colored bars. The user connects these ports by clicking the left mouse button on it. Then, available connections are indicated by highlighting the objects in green and drawing a white connection line between the ports. User interfaces associated with an instantiated module automatically become created at the left of the visualization viewer. For

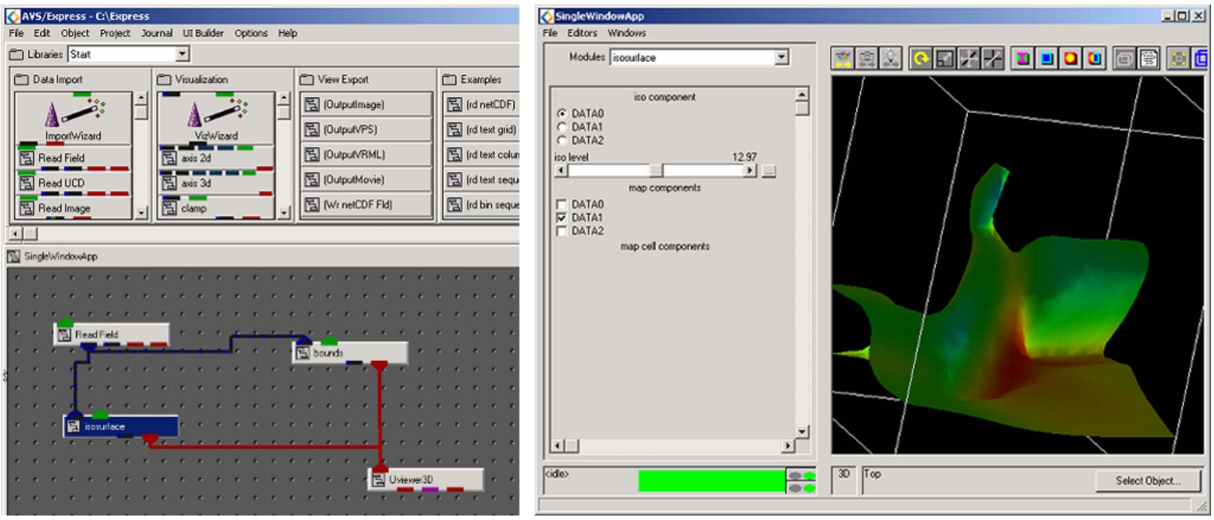


Figure 6.3: **AVS/Express: User interface.** *The left-hand side shows the network editor which is used to combine modules contained in the lobrary to build visual applications. The user selects a module from the top and drags it to the application container below where it is instantiated. Then, the modules are connected by clicking and dragging lines from one module port to another. Once all modules are connected, the visual network is finished. After that, the properties of each module can be set in the associated user interfaces automatically created in the visualization viewer on the right. The user is also enabled to change a wide range of visual properties of each object displayed in the scene. (image courtesy: [1]).*

example, the insertion of an input module causes the creation of some user interface containing the selection of a file name in the visualization viewer.

The *Visualization Viewer* provides a wide range of functionality for the interaction with the displayed data. The user is enabled to pan, zoom and rotate the visualization as well as controlling each object's style and appearance. For example, the surface opacity or the light reflection properties of an object's surface may be changed. Camera and lights in the scene are also highly customizable, e.g., bidirectional lighting and perspective camera view can be chosen. The user interfaces associated with the modules in the visualization network are shown on the left side of the visualization viewer. They allow the dynamical change of each module's properties.

6.2.2 Evaluation

Contacting the company Advanced Visual System and getting a trial license to test AVS/Express took three months. The first try to read in data did not work because of the lack of date support. Again, the year number was used and then the input worked correctly. The other columns contained in the input file were recognized from the beginning which makes AVS/Express the best

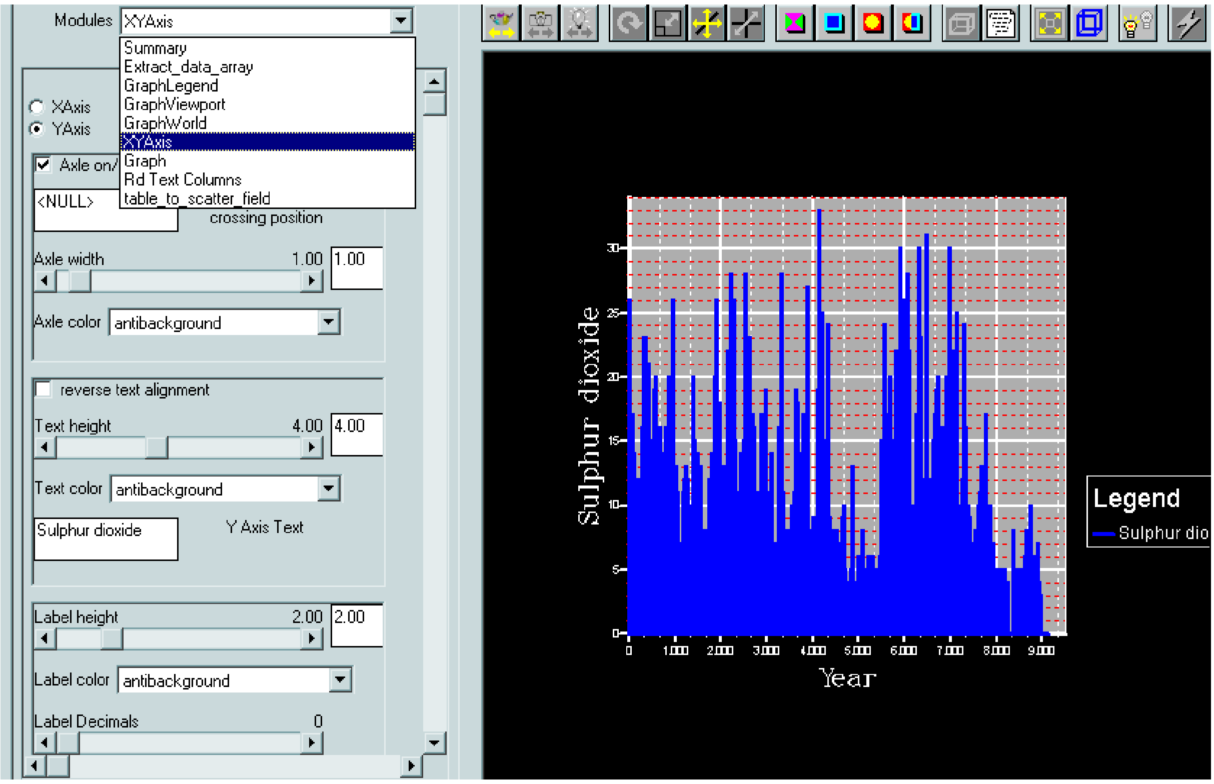


Figure 6.4: **AVS/Express: Task 2B.** *With the drop down box on the top left every module can be selected to adjust its settings. The vertical axis did not display the right labels from the beginning and had to be adjusted to map to the correct values. Although the year column was chosen to be displayed on the horizontal axis it shows the index number of each data point. The support was contacted to get information on this problem but no answer was received. (image courtesy: [1]).*

program regarding input handling. Several examples are coming with the application and with their help a visualization was produced quite fast. AVS/Express has very much documentation which makes it not easy to find the right section for the current problem.

Task accomplishment

The input wizard which guides the steps to read in and display data was used to create a visualization of the input dataset. With the help of the documentation and several examples the Sulphur dioxide values were displayed. The vertical axis had to be adjusted manually to show the correct values. The horizontal axis should plot the year values, but somehow this did not work. The support was contacted to solve this problem, but no answer was received. Figure 6.4 displays the bar plot visualization.

6.3 GeoVista Studio

GeoVista Studio [4] facilitates rapid and programming-free development of visualization applications. It includes 2D and 3D visualizations, interactive query devices and statistical tools. Since JavaBeans are used as programming components it is easily extendible with customized components. Intentionally designed for the analysis of geographical scientific data, it employs several general analysis and exploration tools as well, for example scatter plots and parallel coordinates. The GeoVista website offers a quickstart guide, several usage examples and some other files that help in the process of getting familiar with the programming environment. Several sample designs that can be loaded into the application are provided too.

6.3.1 User Interface Components

Two windows are used to build visual programs: the *DesignBox*, also shown in Figure 6.5, and the *GUIBox*. In the DesignBox the components are connected to build applications. The GUIBox is the interface for the user where the application and its GUI is displayed and where data are read in and examined. The main window contains all the components that can be placed in the DesignBox and the general menu where designs are opened and saved and components can be created or edited. The user chooses a component palette from a dropdown element and then selects the proper bean according to his needs. The context menu is utilized to get quick information about the properties of a component. Each component respectively bean has input and output ”ports” representing a method or action the bean can take. These ports are activated and deactivated by the user in the properties window of each bean. By clicking one output port and dragging the cursor to an input port of another bean a connection is established. Whenever this is done, the corresponding GUI element, if existent, is placed in the GUIBox. After finishing a design the GUI elements are arranged and can be used immediately to load and visualize data. Not every component has to be visible in terms of having a GUI. There are also components dealing with computation or data transformation tasks. The following list shows some components especially important for the use in information visualization.

- *File chooser and Data reader* - These are beans used for reading-in data and transforming it to java data objects for further computation.
- *AttributeList* - A component that allows the user to select a subset of the original list of attributes. The reduced number of attributes then can be included in a new view of the data.
- *PCP Component* - This component draws parallel coordinate plots with the inherent feature of line selection and highlighting.

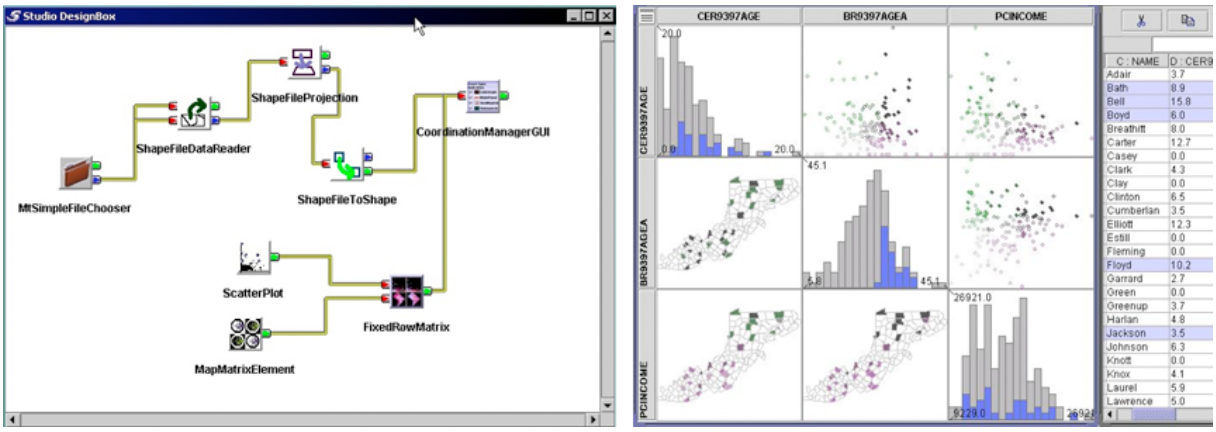


Figure 6.5: **GeoVista Studio.** *JavaBeans are combined in the DesignBox window shown on the left-hand side. By clicking and dragging the cursor between two components connections are established. The right-hand side shows an example of a visualization using the Matrix component that arranges components in a grid-like fashion. Components employed here draw scatter plot ands and histograms. (image courtesy: [4]).*

- *GeoMap* - This component accepts a list of shapes and maps them. As for the PCP component, user selections are possible and highlighted in all plots.

- *Histogram* - There is also a component for histogram drawing.

- *Coordinator* - The Coordinator handles the communication between components. It passes events fired by one component to others listening to these events and ensures that each object uses the same color in all visualizations.

- *Matrices* - Several components have the ability to combine multiple visualizations in one matrix - see also Figure 6.5.

6.3.2 Evaluation

The lack of documentation for the java beans set up in the DesignBox hampered the work with GeoVista Studio. The designs for the creation of visualizations were not very difficult to build up, but it was difficult to find out how to arrange them. Sample data files were positioned a bit hidden on the website and only found after contacting the GeoVista newsgroup. The input file format is not documented as well and had to be determined by examining the input sample files. Because there is no support for a date format the original date was changed to year numbers again. Although the example files at the website were helpful still a considerable amount of initial training is needed to work efficiently with GeoVista Studio.

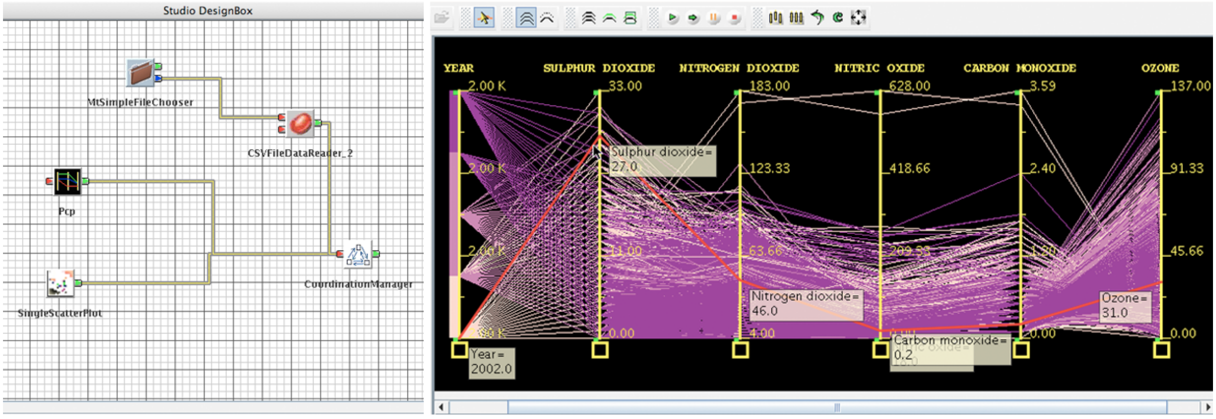


Figure 6.6: **GeoVista Studio: Task 2B.** *The left-hand side shows the arrangement of the java beans to create the visualization in the GUIBox. The Coordination Manager handles the communication between the beans contained in the visual program. MtSimpleFileChooser and CSVFileDataReader are used to read in the data, and SingleScatterPlot and Pcp visualize the data in a scatterplot and a parallel coordinate plot, which is shown on the right. By moving the cursor over the parallel coordinate axes, the relevant line is highlighted in red, and the highest Sulphur dioxide value in 2007 is determined. (image courtesy: [4]).*

Task accomplishment

After adding the relevant beans in the DesignBox, the user interface elements were arranged conveniently. Then the data were read in by clicking on a button. The parallel coordinate visualization generated displays all data points of the last five years. By moving the mouse cursor beginning from the top over the Sulphur dioxide axis the highest measured value in the year 2002 which is 27 is found, as Figure 6.6 shows. During the work the amount of data is noticeable in the slow reaction of the application, but GeoVista never quit and was able to handle the 9000 lines of data.

6.4 IRIS Explorer

The IRIS Explorer[9] is a tool for the creation of customized visualization applications. By combining a variety of available modules, such applications are built by the user. Therefore, the duration of the process of developing and prototyping a custom visualization is shortened. Modules include for example methods for reading in data, processing mathematical functions, displaying read in images and generating a color map for the application to a visualized structure. Created sets of modules are saved as a map and can be used later on. All functions of the IRIS Explorer are also controllable using a scripting language. This enables the developer to run the application automatically which is useful for testing purposes and remote access. The

IRIS Explorer website provides a detailed user manual and frequently asked questions. The following sections describe the parts of the Explorer's user interface.

6.4.1 User Interface Components

The *Librarian window* holds all of the available modules and maps. A set of example maps come with the programming environment which helps to get started. Complete maps are dragged from the Librarian to the Map Editor to be rendered in a separate window. Modules that are used very often can be stacked on the module shelf, an area at the bottom of the Librarian window.

In the *map editor* modules are placed and connected to each other. When dragging a map from the Librarian to the Map Editor, it appears module by module on the screen. The connections between modules are shown as lines and are colored according to the data type that is being passed along them. They illustrate the flow from the input data to the final rendering process. Modules on the left deal with the gathering of input data, and the modules on the far right render the pre-processed data, as shown in Figure 6.7. By creating loops, animations are set up. When rendering a loop, data from one phase of the computation is fed back around to the next phase of the computation. Each loop has to contain a loop controller module that controls and terminates loop iteration. Termination happens if a termination condition evaluates to true. This condition is either checked at the beginning or at the end of each loop iteration, depending on which loop controller module is employed.

By default, modules are shown in a compressed way, called the mini control panel, but by clicking on the buttons at the right top corner of each module, the module expands to the full-scale control panel that contains parameter widgets and their values. After modifying these widgets, the parameters of the rendered visualization are changed and it is re-calculated. The duration of the recalculation depends on the size of the map and the amount of data processed. Maps containing a large number of modules and connecting lines tend to get cluttered because of overlapping "wires". To address this problem, related modules can be collapsed into a group module. It is possible to change the appearance of a control panel and also the calculation that is done within a module using special Editor windows.

The *Render window* displays the visualization resulting from stepping through all of the modules. By clicking and dragging the visualized object is rotated dynamically. A thumbwheel at the bottom right enables the user to zoom in or out. Two picking modes are controlled by buttons at the right of the render window: in the view mode, camera parameters, like position and directions are controlled. The user pick mode allows the selection and manipulation of visualized objects. Three render editors are provided: the material editor, the color editor and the transform sliders window. The material editor is used to set material properties like ambient color, shininess and transparency. With the color editor, the user changes the color properties

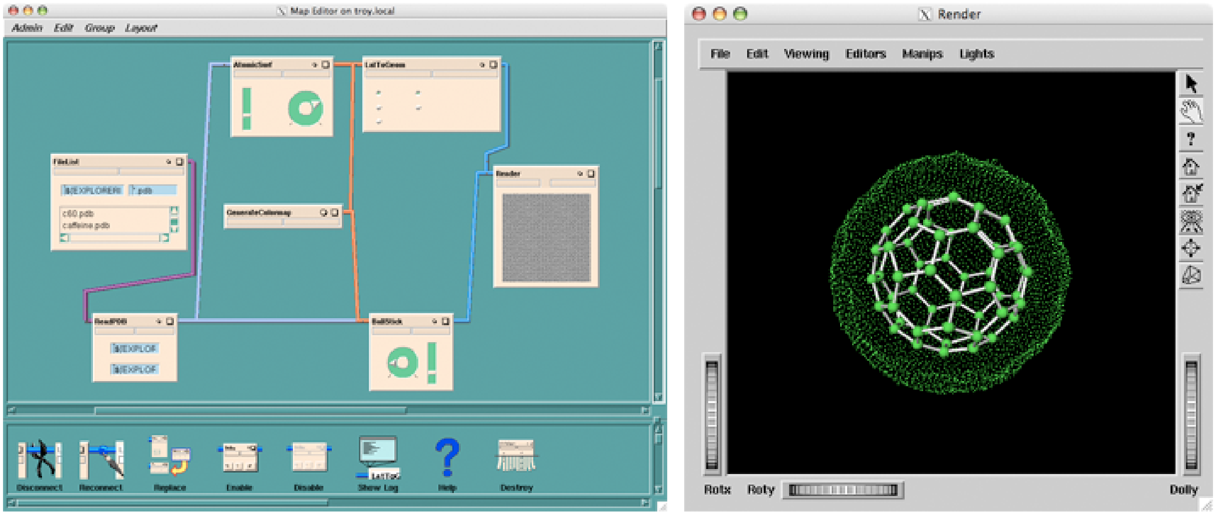


Figure 6.7: **IRIS Explorer Interface Elements.** *The figure above shows an example map provided with the IRIS Explorer that visualizes a chemical data structure. On the left, the map editor containing all modules and their connections is shown. The right-hand side depicts the rendered structure resulting from the calculation. (image courtesy: [9]).*

of the object material. The transform sliders window sets object position, scale and rotation exactly. By clicking with the right mouse button on the background of the render window, the pop-up menu opens. It provides several functionalities: when the user gets lost by zooming in too far, a click on ”View all” brings him back by resetting the camera position. Several draw styles such as wireframe can be chosen and lighting options are given.

6.4.2 Evaluation

Like most of the other applications, IRIS Explorer had read in problems. Every time the input of data was tried, an error message was displayed in the output window and the data was not read in. Although the support was contacted and answered relatively fast, the source for this problem could not be found out. Trying the input of a ten-line-version of the input file, as suggested by the support, did not work as well. Therefore, the tasks could not be accomplished.

7 Other relevant Visualization Applications

Other software packages concerning the visualization of data are covered next. Some of them have not been used in the practical part of this thesis because they serve only a specific purpose, others because they only allow the creation of visualization applications but no end-user interface. But since these applications are well-known and offer good visualization possibilities, they are described in this section.

7.1 DeVise

The DeVise environment [3] is written in java and allows the fast creation of simple visualizations for data exploration. Predefined visualization types include bar chart and scatterplot. In the basic visualization process the user reads in the data from a file or web source after defining the data schema. Then two attributes for x and y axes have to be chosen. To refine the visualization, attributes like color or size can be connected to the value of data attributes. Visual filters specified by the user that narrow the range of an attribute value are applied to filter out unwanted items. These queries can be stored and utilized over and over. Visual links are used to link axes of multiple views which results in synchronously scrolling and zooming actions. DeVise supports the common csv file format and lets the user choose other separators as well. It is available for Linux and Solaris. The DeVise website offers a good introduction to the creation of visualizations and a detailed user manual.

7.2 RefViz

RefViz [14] is a software concerning the analysis and visualization of reference data. It is designed to provide both an overview of all references and in-depth knowledge of interesting spots in the research field. Two main visualizations can be switched in its main window: the galaxy and the matrix view, as shown in Figure 7.1. The galaxy view shows the whole set of references organized according to their content. RefViz determines major and minor topics and subsequently calculates the data visualization containing points for single references and special items for reference groups. Using several query tools, the user is able to search this visualization for a term or highlight those references related to a selected topic. The matrix view expresses the relationship between the assigned reference groups and the major concepts and helps the

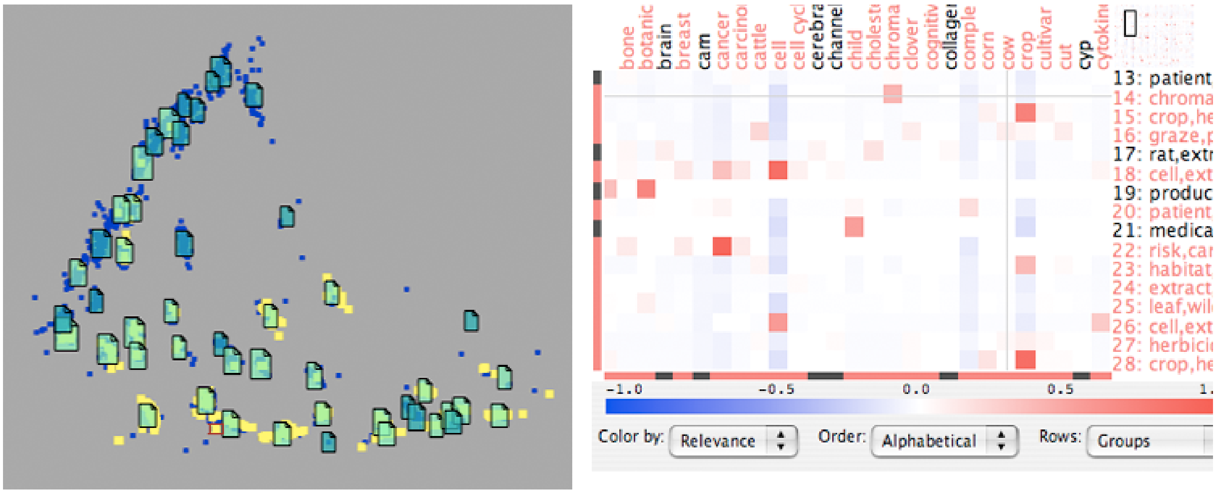


Figure 7.1: **RefViz: Reference visualizations.** *The galaxy view on the left-hand side shows the complete set of references with yellow highlighting due to a search. A single reference is expressed as a point and a reference group is displayed as a paper symbol. Topics and their relationship to reference groups are depicted by the matrix visualization on the right. The color of each cell indicates the strength of the association between a group and a topic. Red means that the column topic is a concept important to the group, and blue signifies that the topic is mentioned rarely or not at all in that group. (image courtesy: [14]).*

user to discover which topics tend to be discussed together in literature. Both views are linked: by selecting a topic in the galaxy view, the related concepts and groups in the matrix view are highlighted too. RefViz has the ability to retrieve reference information from a series of content providers, which improves the collaboration when users are using different software packages for the organization of their reference collection. The import of text documents in various formats is possible as well.

7.3 IN-SPIRE

Similar to the RefViz software, IN-SPIRE [7] enables the user to gain deeper understanding of the relations in big document collections. A set of different visualization tools is available for that task. The theme and galaxy views provide an overview of the whole data set. The location of major and minor themes is assisted by the former, while the latter has query functions and the possibility to highlight groups. The management of outliers is included into the galaxy view to improve the positioning of the documents and overlapping groups are emphasized by a specific color. Figure 7.3 shows an example of such a visualization recalculation. Single documents can be rated regarding their support or refuge of the own hypothesis. The correlation view is used to depict the ratio of common documents between several groups and allows the bundled

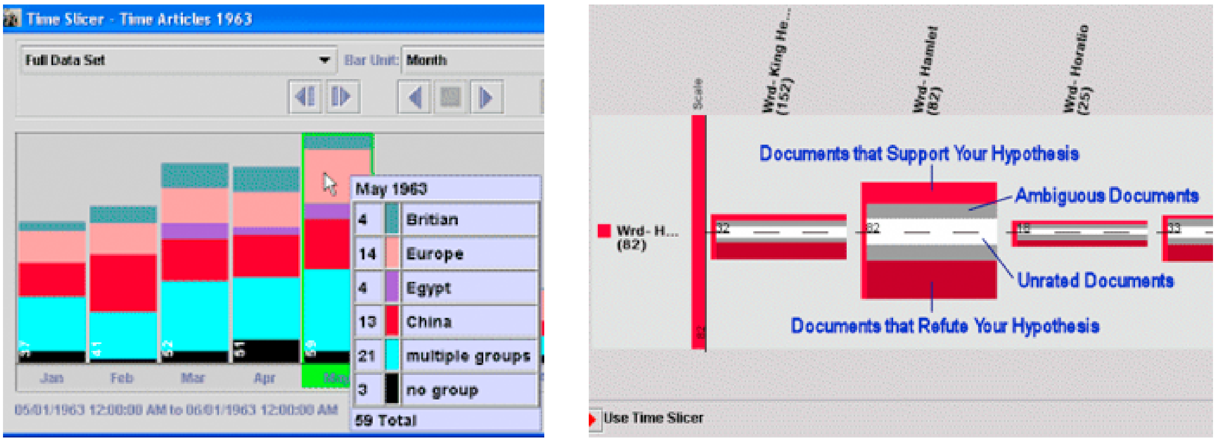


Figure 7.2: **IN-SPIRE reference tool.** *The evolvement of document groups over time is visualized by the timeslicer tool, as shown on the left-hand side. Groups are highlighted in their own color and exploration is assisted by tooltips. The correlation view on the right depicts the ratio of common documents between several groups and therefore helps to see differences and similarities among groups. (image courtesy: [7]).*

visualization of the group rating, as shown in Figure 7.2. Querying comprises simple text queries connected by boolean operations, the search for documents similar to a selected one and the finding of document notes containing a query term. These notes are created by the user and annotate text phrases in the documents. With the triage tool it is possible to connect and save multiple queries and user actions for later application on other data sets or groups. Translation facilities are included in the set of tools. Data sets whose documents contain a date field can be analyzed using the timeslicer tool, a histogram of the number of documents over several time periods. The highlighting of groups in their own color reveals the development of theme groups over time and reminds of the history flow visualization [42]. The documentation on the IN-SPIRE website is good and helps to get started fast.

7.4 Libraries and Toolkits

Although the focus of this thesis lies on the evaluation of complete visualization frameworks, available libraries which can be used to build such frameworks shall also be described. The visualization possibilities offered will be shown and used data structures will be characterized. Features especially important for developers like documentation and examples will be outlined.

7.4.1 Prefuse Toolkit

Prefuse [13] is a freely available toolkit for the development of java-based information visualization applications. It provides both the needed backbone-datastructure and a lot of concrete

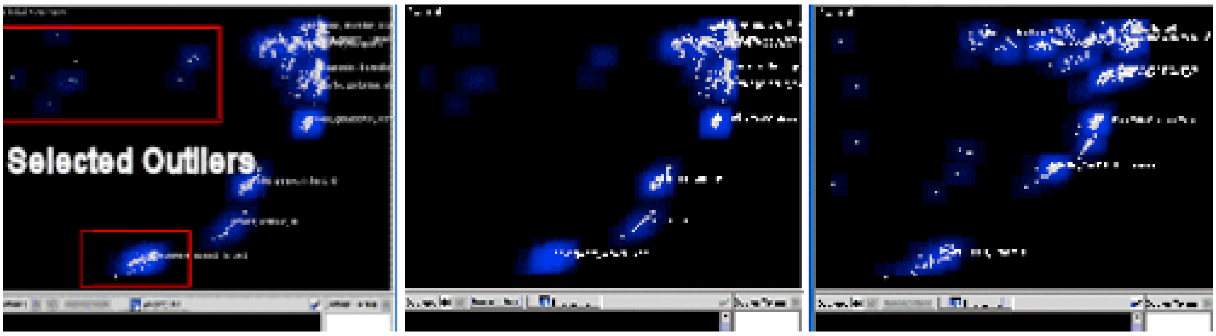


Figure 7.3: **IN-SPIRE: Elimination of outliers** *Sometimes a data set contains documents so different from the others that they skew how the data set is presented as whole in the galaxy view. This case is resolved by moving those outliers to a special panel to be kept accessible but not to interfere with the calculation of the visualization. After that the visualization is recalculated, as shown in the third image. (image courtesy: [7]).*

visual encoding techniques. The data are organized in the familiar structure of tables, graphs and trees. Visualization capabilities cover several layouts for the visualization of trees, the fish-eye distortion technique, zooming, animation and even a gravitational force model to express dynamical relations like repulsion and attraction. Several utilities for the processing of data connect the underlying data model with the visualization elements. Input data are accepted in common formats, for example delimited and comma-separated text files for the input of tables and the xml formats TreeML and GraphML for the input of trees and graphs. The Javadoc API documentation has grown in the last three years and is offered at the Prefuse website. Demos and FAQs are given as well. The user manual at the website provides a good introduction to the toolkit with the explanation of an example application. Unfortunately the manual is not yet finished, and other sections describing data structures and interaction mechanisms are left blank.

7.4.2 Piccolo

Applications using either the java or the c# version of the Piccolo framework [12] are constructed as a scenegraph containing nodes, e.g., a camera node, a text node or an image node. By subclassing and compositing given nodes the programmer is able to build complex visualizations. Zooming and animations are supported and fully integrated into the framework. Based on the scale a node is shown at its visual representation can be changed. This technique is generally called semantic zooming. The arrangement of multiple cameras and the construction of fisheye views is possible when using Piccolo. The framework comes with good documentation: the website provides a detailed description of the visualizing possibilities and a javadoc API documentation is offered as well, though some method entries are still missing. The installation is explained and eclipse project files come with the java version of Piccolo. Several examples of the use of Piccolo are given at the website to ease the process of getting started.

7.4.3 InfoVis Toolkit

The InfoVis software package [8] is fully written in java and builds upon three data structures: tables, trees and graphs. The concept of columns is used internally to store attribute values. This eases the addition of new attribute columns while the application is running. A column's values can be dependent on the values of another column and are employed to specify filtering and sorting. Aggregation classes manage calculations like sum, mean or maximum within a column. Nine visualizations are implemented: scatter plots, time series, parallel coordinates and matrices for tables, node-link diagrams, icicle trees and treemaps for trees and adjacency matrices and node-link diagrams for graphs. Various interaction techniques are supported: fisheye visualizations, zooming, filtering using dynamic queries and dynamic labeling. Fisheye views and dynamic labeling can be enabled and disabled by the user. Visual attributes of a visualization element contain transparency, labels, color and size. The toolkit is distributed with a set of example input files to try and test its facilities. The website holds a preliminary version of the user manual and an uncompleted javadoc API documentation. Available input file formats are covering csv, xml and several other common formats for the reading of data.

7.4.4 Jung

The open-source library Jung[10] (Java Universal Network/Graph) handles visualizations of graph or network data. Therefore, the basic underlying data structure consists of graphs containing edges and vertices. Annotations can be assigned to each vertex by applying a built-in mechanism. Layout algorithms and the use of filtering provide the means to explore the visualized network data. Interaction techniques like tooltips, zooming and viewing lenses assist the user as well. Implementations of graph algorithms, e.g. clustering, optimization or network distances are included in the framework. The library documentation comprises an almost finished user manual and a javadoc API documentation. Several examples come with the distribution of Jung and it supports the Pajek file format for reading and writing and the GraphML format for reading only.

8 Categorization of the Software Tools

To compare the given visualization tools a meaningful categorization is used. This categorization builds upon three criteria: *time*, *data* and *representation*. Every tool is categorized to point out the differences between them.

8.1 Description of the Categorization

To compare the given visualization tools a meaningful categorization is used. This categorization builds upon three criteria: *time*, *data* and *representation* [19]. The time axis can be made up of time points or time intervals. Each composition implies other relationships between the elements in time and influences the validity of data: if an application uses time points, data are only valid at discrete points in time, which should be reflected in the visualization itself. The structure of the time axis can be linear, cyclic or branching. Linear time sees time as an ordered collection of temporal events. A cyclic time axis is composed of a set of recurring temporal elements, e.g., the seasons of the year. Branching time allows for alternative scenarios by offering the possibility to follow a temporal primitive by more than one element. A fourth temporal category is mentioned in addition: multiple perspectives. In branching time only one path of time will actually happen, whereas multiple perspectives facilitate concurrent views on time.

Considering the criterion data, we can distinguish abstract and spatial data. In contrast to spatial data, abstract data have no inherent spatial structure. The number of associated data values is another categorization criterion: multivariate data bears the challenge to detect correlations, whereas univariate data is handled easier. Visualizing the raw data is useful in many scenarios, but complex and large-scale data sets introduce the need to derive data abstractions tailored to the needs of the users. Overview+detail views containing aggregated data values are an example for such data abstractions.

The third criterion deals with the representation of time-oriented data. Static representations show still images of points in time. They enable the user to concentrate on the visualized information and to compare parts of the time axis. Dynamic representations use the physical dimension time to visualize the information and change over time. They are well suited to give an overview to the general development over time, but especially with long time lines, users are often unable to follow all changes in the visual representation. The presentation space can be two- or three-dimensional. 2D Views are often considered as sufficient for data visualization and analysis. 3D Views bring the opportunity to encode more information, but involve problems like occlusion and back faces not seen by the user.

8.2 Categorization Application

After working with every software tool the categorization is applied to any of the used software tools. Familiarity with each tool is important to correctly assign the particular parts of the categorization criteria. Problems during the categorization are kept in mind for the subsequent evaluation. The tools where the data input did not work, namely IRIS Explorer and cviz, are excluded from the categorization.

Tableau

Considering the criterion *time* Tableau is built upon time points but the values can be summed up or combined using other aggregation functions to show intervals, e.g., year, quarter or day. The time axis has a linear structure and the time intervals are seen in a hierarchical way - the granularity of the time axis is adjusted dynamically by the user by clicking on plus and minus signs in the time interval icons. For example, clicking on the plus sign of a year symbol opens its four quarters and automatically splits the connected visualization in the new intervals. Branching is not supported. The *data* context is abstract which means that data inherently connected to a spatial layout can not be visualized. Uni- and multivariate analysis is possible when using Tableau. The user is enabled to visualize the raw data directly or to build abstractions using filters or aggregation functions. The visualization concept of overview and detail is employed in the level of detail shelf. Placing a variable on it separates an element of the visualization according to its members. Clicking on it reveals this partitioning by red highlighting, and the exact values are showed by a tooltip. The data *representation* in Tableau is static and the presentation space used is two-dimensional.

Spotfire

Similar to Tableau, the temporal primitives are considered points in *time* and can be combined with a variety of aggregation functions to view time intervals. Spotfire supports more aggregation functions than Tableau, more than twenty functions are offered. The temporal primitives are arranged in linear fashion and without support for branching or cyclic structures. In general, the visualized *data* come from an abstract context, but the import of geographic data by using an image or an ESRI shape file is possible. Both uni- and multivariate data sets can be analyzed by the application. To avoid cluttered displays, several data abstractions are provided: each variable in the data set has an associated filter which type can be changed dynamically, e.g., from a radio button filter where the user selects one variable value to a range filter where a data range is selected. Aggregation functions can be applied to every variable to determine the overall process of a variable. The user is also enabled to add custom columns to the data set. Columns can be calculated from other data values at the same point in time or combined in so-called binned columns that divide the data in classes. To find the exact values of a selected visualization item the details-on-demand panel is employed. Spotfire uses a static *representation* of the time-oriented data and two-dimensional visualizations.

ILOG Discovery

ILOG Discovery handles temporal primitives as time points and does not support time intervals. *Time* is structured linearly and has no branching structure. The visualized *data* come from a non-spatial context, therefore the frame of reference is specified as abstract according to the categorization. Both uni- and multivariate analysis are possible. The interaction process follows the common concept of overview and detail: first, an overview visualization is selected by the user to become familiar with the data structure. Then, other visualizations can be created to gain deeper insight and to view only parts of the whole data set. These visualizations are viewed in parallel and dynamically linked which means the highlighting of a selected region in all other visualizations. Interactions, like filtering by manipulating sliders, data partitioning using aggregation functions and comparison of these groups are provided. Coming to the criterion *representation*, ILOG does not utilize the physical dimension time to express the structure of temporal data but uses static representations of the data. The presentation space itself is two-dimensional.

Xmdv Tool

Time points represent time-dependent data at the usage of the Xmdv Tool. The temporal structure of the visualizations is linear and neither branching behavior nor time intervals are supported by the application. The viewed *data* are not inherently connected to a spatial structure and therefore considered abstract. Multiple data values are allowed to be associated to a temporal primitive, so uni- and multivariate data can be visualized. Both the data level and the data abstraction level are integrated in the application: the user is enabled to choose between "normal" visualizations and hierarchical visualizations especially suitable for large data sets. These hierarchical visualizations employ the technique of hierarchical clustering to find hidden relationships and connections in the data. No parallel display of overview and detail visualizations is possible. One major drawback of the Xmdv Tool is the lack of aggregation functions which are important for the combination of data values. Several filtering methods are employed, e.g., adjustment of the clustering radius in the hierarchical visualizations and brushing methods in the non-hierarchical ones. The temporal data *representation* can be static or dynamic, animations are included in the application. But due to the lack of a supported date format temporal data points have to include a sequential number to be recognized as a sequence. The dimensionality of the viewing space is two-dimensional.

OpenDX

The visual programming environment OpenDX handles temporal primitives as *time* points and positions them linearly on the time axis. Intervals, cyclic or branching structures are not supported. Since spatial structures can be visualized as well as abstract *data*, the frame of reference is abstract or spatial. Multiple variables are allowed to be associated with a temporal primitive, therefore uni- and multivariate data analysis is possible. In the process of visualizing a data set the focus is given to the raw data which is visualized in the Image Window. The representation of time-oriented data can be static or dynamic depending on the utilized visual modules. The user is enabled to use modules for interaction processes or the sequencer for animated visualizations. Even the dimensionality of the presentation space depends on the modules in the Visual Program Editor. Two- and three-dimensional presentation spaces can be created.

GeoVista Studio

GeoVista Studio focuses on geographic data and information visualization. The temporal primitives are modeled as *time* points arranged linearly without branching or cyclic structures. The *data* visualized can have an abstract or spatial inherent layout. The number of variables is limited by nothing but the speed of the application, therefore uni- and multivariate analysis is possible. No aggregation functions are provided to sum up or combine variables in another way. Range sliders for the filtering of data are available in the Java Beans collection, but no tutorial how to employ them is found on the website which makes their handling quite difficult. In general, emphasis is put on the visualization of the raw data. The *representation* of time-oriented data is static and two-dimensional.

AVS/Express

Temporal primitives in AVS/Express are considered points in *time*. The structure of time is linear, neither cycling nor branching behavior is supported. Both abstract and spatial *data* can be visualized using AVS/Express. Each temporal primitive may be associated to one or multiple data values, therefore both uni- and multivariate analysis are possible. When displaying abstract data, the focus lies on the visualization of the raw data. But coming to data with an inherent relation to some spatial layout, several data abstractions are provided. e.g., slices of volumes, isolines and contours. The user is also enabled to remove noise contained in the data by employing special filters. Depending on the modules used, the *representation* of the temporal data is either static or dynamic. Due to the possibility of visualizing spatial data, the presentation space can be two- or three-dimensional.

8.3 Summary

The following tables contain the results from applying the categorization to every information visualization tool. By examining the findings in parallel the applications can be compared more easily.

Information Visualization Tools					
		Tableau	**Spotfire**	**ILOG Discovery**	**Xmdv Tool**
Representation	*Time dependency*	Static	Static	Static	Static, Dynamic
	Dimensionality	2D	2D	2D	2D
Data	*Lev. of Abstraction*	Data, Abstract	Data, Abstract	Data, Abstract	Data, Abstract
	#Variables	Univariate, Multivariate	Univariate, Multivariate	Univariate, Multivariate	Univariate, Multivariate
	Frame of Refer.	Abstract Spatial	Abstract, Spatial	Abstract	Abstract
Time	*Temp. Primitives*	Linear	Linear	Linear	Linear
	Struct. of time	Time points Time intervals	Time points Time intervals	Time points	Time points

Visual Programming Tools				
		OpenDX	**Geovista Studio**	**AVS/Express**
Repres.	*Time dependency*	Static, Dynamic	Static	Static, Dynamic
	Dimensionality	2D, 3D	2D	2D,3D
Data	*Lev. of Abstraction*	Data	Data	Data, Abstract
	#Variables	Univariate, Multivariate,	Univariate, Multivariate	Univariate, Multivariate
	Frame of Refer.	Abstract, Spatial	Abstract Spatial	Abstract, Spatial
Time	*Temp. Primitives*	Linear	Linear	Linear
	Struct. of time	Time points	Time points	Time points

9 Summary

The previous chapters are now summarized and the main statements are recapitulated. A discussion of the findings taking into account both the used categorization and the examined information visualization tools is included.

9.1 Chapter Summary

This work evaluates several state-of-the-art visualization tools with special focus on the visualization of data with an inherent relationship to the dimension time. Herein a recently published categorization and a task schema is applied. Beginning with a description of possible visualizations of temporal data and other categorizations for visualization applications, a set of tasks is defined to guide the practical work. To get familiar with the content analyzed, the data set containing air pollution data is described. The visualization tools are examined and partitioned in the categories information visualization tools, visual programming tools and other relevant but not categorized applications. The previously defined tasks are carried out with every information visualization tool and the categorization is applied to all of them. In the following, key statements of the main sections of this thesis are pointed out.

9.1.1 Related work

Currently available applications for the visualization of temporal data often solve only specific problems. Every linear visual metaphor used, be it timelines, glyphs or flows, may be employed to display every kind of time-oriented data set. But for the discovery of solutions for a certain problem a more specialized visualization often yields better results, which explains the amount of specialized visualization tools. Applications that enable the user to explore and discover the periodicity in temporal data sets are still rarely available and often sacrifice the display of multiple variables for the possibility to recover their periodical nature. Here, research effort has to be done to better support the user in this task.

Categorizations for visualization applications position themselves in a continuum between the user and the application. User-oriented categorizations take into account psychological aspects like the cognitive effort made by the user, whereas categorizations oriented by the examined application focus on its features and interaction possibilites. The main criteria considered in these categorizations are the *data representation*, the *interactions* a user is enabled to make and the factor *data* which includes e.g. the reliability of the gathered data or the relational structure of the data. User-centered taxonomies also take a look out of the box and bring user-centered considerations into account, like the *skill level* and the user's needs or expectations, though these factors are difficult to measure. Important and often mentioned factors in the way data is represented are smooth transitions between different states of the visualization, the use of display dimensions like size or color, and the way multivariate data is handled. Filtering, grouping and the ability to control the levels of detail of the visualization are considered as crucial interaction features an application should provide.

Taxonomies dealing with the special field of visualizing temporal data are rare. Though they are correlating with general categorizations in many ways, e.g., the control of several levels of detail, some specific considerations have to be made when handling the dimension time. Since time has different granularities, the unit or a unit system for the representation has to be chosen. The decided structure of time can be linear or cyclic, and the visualization elements can be considered time points or intervals. The use of metaphors for the expression of relations between these elements is mentioned as well.

Empirical analyses of visualization applications use predefined tasks to evaluate their practical effect. Tasks are often selected according to their relevance in the practical scientific work. Users have to analyze one or more attributes of the data set, find correlations between at least two variables, compare the values of attributes or find clusters. The performance of the tested applications is generally measured by the time the users needed to complete each task and the correctness of their answers. Some evaluations also take into account more subjective impressions like user preference and rating of the tested tools.

9.1.2 Practical work

Oriented by other empirical analyses of visualization tools tasks are defined at first. They are selected to cover as much of the work in real life as possible. The tasks are differentiated to three categories: low -level tasks covering some essential questions, e.g., the location of a measured value, basic tasks handling only one or two variables at a time and advanced tasks dealing with more complex problems like the determination of trends and the discovery of correlations.

Due to the diversity of the examined applications they are divided into classes: information visualization tools, visual programming tools and other visualization applications. Since this work is focused on information visualization tools, all the tasks are carried out with each of them. To give an impression of the visual programming tools one selected task is performed. The other visualization tools that serve too specific purposes or do not include a graphical user interface because of their toolkit characteristic are described.

The most cumbersome part in the process of visualizing the data set is the data read in.

The data input did not work at the first attempt for any of the applications. The input file had to be changed to avoid empty columns due to missing values, the data type of the columns often was not recognized correctly and some of the applications even required a specialized file format and did not accept a common .csv file. The ability to recognize a date format which is very important for the visualization of temporal data was only existent for Tableau and Spotfire. All other tools would have needed a column containing sequential numbers to recognize the sequential character of the measurements.

The amount of data examined stated a big problem for some of the applications. Crashes occurred often and in the case of ILOG Discovery prevented the accomplishment of the whole set of tasks. The quality of the user support was diverse: some trial licenses could be obtained in the first two weeks after the first contact, and others took nearly three months to acquire. Inquiries in case of problems were answered fast in the case of most of the applications, but sometimes not at all.

Most of the examined information visualization tools use well-known visualization concepts like line plots, parallel coordinate plots, bar plots and scatter plots. The advantage in doing this lies in the familiarity with these concepts, but as seen in the related work section, other visualizations would be possible as well. Animations are very rarely utilized, and the temporal data are treated in general like any other data set containing a linear sequence of values.

The visual programming tools follow mutual principles: from a set of predefined modules the desired modules are chosen and placed on some kind of canvas. After establishing connections between them, the data are visualized in a display window. Because of the need to know which modules have to be connected together to get the desired visualization a good documentation is of great importance which is not available for some applications. To ease the process of creating a visualization, AVS/Express is the only application that offers supporting wizards. Though the visualization of two-dimensional content is possible, most of the visual programming are intent on the display of three-dimensional data. This becomes apparent e.g. in the support for choosing lights in the scene or inserting intersection layers.

Although strictly speaking out of scope of this thesis, visualization tools serving specific purposes and some toolkits for the creation of visualization applications are described as well. RefViz and IN-SPIRE both display document collections employing the concept of overview and detail. A main window shows the ”galaxy” of documents and some supporting windows may be opened to further explore the data using other visualizations. The examined toolkits support common user interactions like zooming or filtering mechanisms. Some of them focus on the visualization of network and graph data, while others take a wider view and offer concrete visualizations like parallel coordinates or treemaps.

9.1.3 Categorization Application

The categorization which is used to compare and evaluate the information visualization tools is described first. It consists of three main criteria: time, data and representation. The first criterion deals with the assumed structure of time in terms of elements in time and their positioning. The structure of the visualized data and the ability of the application to derive abstractions are considered in the criterion data. Finally, representation covers the layout and dynamics of the viewing space.

After the practical part including some initial training and the task accomplishment the categorization is applied to the information visualization and the visual programming tools. The application features according to the categorization are collected and summarized. Correspondent to the field of information visualization, information visualization tools focus on the visualization of two-dimensional and abstract data. Visual programming tools put more emphasis on scientific visualization and therefore rather support dynamic and spatial visualizations.

9.2 Answers to Research Questions

The research questions formulated at the beginning of this work are answered now. Both the visualization concepts employed by the examined tools and the application process of the categorization are reflected and described. Starting from occurred problems, suggestions for possible solutions are specified.

9.2.1 Visualizations of Time-Oriented Data

Because of the orientation to scientific and three-dimensional visualization of the visual programming tools the following research questions refer to the information visualization tools described and evaluated in section 5. The examined applications have diverse characteristics: some of them employ unique concepts that may not be easy to understand at first, but facilitate new insights, and others build on familiar visualizations and focus on supporting the user in the process of finding answers and solutions.

- *How are current InfoVis systems dealing with time-oriented data?*
  All examined applications consider time a linear dimension and do not offer any features that would help the user to discover branching or cyclic temporal structures. This sequential sight of time is reflected in the visualizations available: parallel coordinates, scatter plots and simple bar or line plots are provided by most of the visualization tools. Glyphs are rarely used by the applications, maybe due to their specific nature and the not easily solvable problem of assigning a variable to the best possible representing glyph part. Only

two of the information visualization tools, namely Spotfire and Tableau, recognized the date format correctly and, therefore, let the user interact with the dates contained in the data set. The others treated the data like any other sequential input and simply positioned the data points along an axis.

The control of the granularity of the time axis is realized quite similar in Tableau and Spotfire. The default granularity ”year” can be changed by clicking on the axis that represents the time. After the selection of a level the visualization is readjusted to display the data values according to the new detailedness. The granularity levels possible range from year over quarter to single days. Data values inside an interval are combined by a user-specified aggregation function. These functions are of great importance for the clear comparison of variable values in different intervals.

- *How do they support the user in exploring the data?*
  Most of the current visualization tools offer some kind of slider mechanisms to filter the unwanted data items out of the visualization. Spotfire enables the user to choose from several kind of filters to handle different kinds of variables, and Tableau has a filter window where either a slider or text fields are used to narrow the range of a variable. Brushing techniques are employed by each tool to highlight specific data points. Visualizations viewed in parallel are always linked to each other to maintain the connection between data items in different views.

  When displaying temporal data drawn on some kind of time axis, the control of temporal granularity is crucial. Only two applications, namely Spotfire and Tableau, recognized the data as being temporal and therefore let the user choose the time interval for data display. Tableau has a convenient mechanism known from directory trees of clicking on plus- or minus-signs to increase or decrease the granularity. The granularity levels are predefined and step from year over quarter to single days. Using Spotfire, one clicks on a plus sign on the time axis to add a granularity level. By choosing the desired temporal unit from a dropdown menu the labeling for this unit is added to the time axis.

- *Is there special support for large datasets available?*
  Large data sets pose a big problem to several of the examined visualization applications. Some cannot handle them at all and crash when a big amount of data is read in. Others have performance problems and allow very slow interaction only. But some tools do not have problems at all dealing with large data sets, e.g., Tableau and Spotfire.

  Aggregation functions that enable the user to compare collected data in different intervals are the most common support to ease the exploration of large data sets. However, not all of the evaluated visualization tools have aggregation functions included, and the number of them differs as well. Spotfire offers the biggest number of such functions, about twenty, and Tableau still has about ten different functions available. The Xmdv Tool does not have the possibility to aggregate data values but includes another method of handling large data sets: hierarchical clustering. Cluttered visualizations due to a large number of data points are clarified by combining multiple similar visualization elements to one element and therefore showing basic trends in the data set.

- *Where lie the problems of currently available visualization applications?*
  A smooth input process without any changes to the input file was not available for any of the examined applications. The real-life data set used contained missing values which caused errors during read-in. Some applications were not able to recognize a date format. Therefore, only the number describing the year could be read in. Input file formats were quite diverse as well: besides the common csv format, other proprietary formats were demanded that required adding some lines to the beginning of the file.

  Data sets collected in real-life settings usually comprise a large amount of data. The data set utilized in this work contained about 9000 lines of five measurements at a time. Some of the applications allowed the user to access these data efficiently by saving the project after reading in the data and therefore using some kind of caching. But others had serious problems dealing with such a big amount of data. Problems ranged from definitely slowed down user interaction to complete crash of the application.

  The information visualization tools compared have quite different features and allow more or less user interactions. Once the range of functions increases, the simplicity and clarity of what can be done and how becomes more and more important. Tableau and Spotfire were undoubtedly the applications with the widest user possibilities. In the case of Spotfire the user interface tended to get a little overloaded. Sometimes it was forgotten that somewhere and not visible at the moment a filter has been set and narrowed the data range displayed.

- *What would be possible improvements?*
  The rather innovational character of the field of information visualization is reflected in the problems during the input process. The conversion of the data set to different file formats made the read-in quite cumbersome. Therefore, a standard for formatting input files is needed, especially for the format of temporal data, which was recognized by only two applications. The complying with such a standard would greatly improve the input data handling.

  Some of the examined applications had essential problems dealing with the size of the used data set. The judgement of a visualization tool not only includes the data kinds and formats supported, but also its performance abilities. Therefore, applications should either support real-life data without problems, as in the case of Tableau and Spotfire, or specify the allowed data set size in their documentation if they do not support large data sets.

9.2.2 Categorization Evaluation

The application of the categorization is now reflected. Problems in the use of the categorization that have occurred are described. Possible improvements or changes to enhance the categorization are presented.

- *Is the categorization applicable and useful?*
  The categorization gives a good overview to guide an initial judgement of visualization tools. It is not too detailed which would result in a loss of clarity, and still allows a comparison that points out the most important features of the examined applications. Although information visualization tools and visual programming tools take different approaches in the process of displaying the input data, the application of the categorization is not only possible but also reveals important differences. Therefore, this question clearly has to be answered with yes.

- *Were there any problems using the categorization in practice?*
  Generally, the application of the categorization was done without problems. Sometimes the exact definition of some categorization criterion had to be looked up to assign the correct value, e.g., if the ability to display geographic information results in a spatial frame of reference. But due to the clear and helpful descriptions uncertainties could be clarified fast.

  Looking at the overview table of all compared visualization applications, one may get the impression of no or only a few essential differences between them. Examining for example the differences between Tableau and the Xmdv Tool it is figured out that the Xmdv Tool is not able to display spatial data and does not structure time in intervals, but in return uses animations to express the temporal dimension. By only having this information the Xmdv Tool would be considered to have a little less features than Tableau which barely matches reality. Tableau offers the user more functions to interact and explore the data, for example range and text filters for removing unwanted items which is not possible at all using the Xmdv Tool. As also mentioned in the next section, the addition of the criterion interaction to the categorization should be considered.

- *Are there any changes that could enhance the quality or accuracy of the categorization?*
  The criterion data is divided into the frame of reference, the number of variables and the level of abstraction. The first two points are clearly assignable: the visualization of

abstract or spatial data is easily determined, and the differentiation between uni- and multivariate data visualization is a simple decision. Considering the level of abstraction, one has to consider if an application shows raw data or provides abstractions, e.g., calculated aggregation values or supporting features and events. Here the number and character of abstraction methods available is important to compare visualization tools adequately. Therefore, a further differentiation would be a enhancement to the categorization that nonetheless would diminish its clarity.

The ability to interact with visualizations fundamentally influences the user's process of exploration. As seen in the practical part of this work, a lot of interaction mechanisms are offered by all of the tools, but their realizations differ from each other. Some applications also provide new forms of interactions, e.g., the hierarchical clustering view of the Xmdv tool. Due to the essential contribution of interaction methods to the usability of an application we propose the addition of a fourth category considering the factor *interaction*.

9.3 Final judgement of the applications

Undoubtedly, the evaluated information visualization tools differ in many ways. Each tool provides different ranges and kinds of interaction possibilities that ease the data exploration process. The size of the data set caused problems for ILOG Discovery and some of the visual programming tools as well. The second essential problem, the file input, was shared by all of the applications. Tableau and Spotfire provided relatively fast read-in, and did support and recognize the date format, which was not done by the Xmdv Tool and ILOG Discovery.

The most functions and features are clearly offered by Spotfire and Tableau. The interaction process employed by Tableau that heavily uses the drag-and-drop technique is easily learned and allows access to all important interactions with the visualization. Tableau's user interface provides excellent feedback about which visualization parameters are associated with the variables displayed. This feedback was kind of a problem when using Spotfire: every filter is linked to a variable, and positioned at the right of the user interface. If a lot of variables are contained in the data, the user has to scroll to see the setting of each filter. Sometimes it was forgotten that a filter limited the range of the data displayed and irritating results were produced. But considering the range of functions, Spotfire offers slightly more than Tableau, for example more aggregation functions, different kinds of filters and calculated columns.

ILOG Discovery and the Xmdv Tool basically provide several visualizations for the display of data. The user chooses the visualization he/she wants to use and works with it. Simple interaction mechanisms are offered as well, range filters by ILOG Discovery and brushing by the Xmdv Tool. But the facilities for changing the visualization are relatively small compared to Tableau and Spotfire, for example assigning different colors to different values of a variable is not possible. ILOG Discovery also had serious problems handling the size of the data set, and the Xmdv lacked the aggregation of variable values. Therefore, these tools give a first overview of a data set, but for in-depth and accurate analysis a more professional application like Tableau or Spotfire is more suitable.

10 Conclusions

In this thesis, currently available information visualization tools were compared. The comparison comprehended both a practical and theoretical part: first, predefined tasks were carried out and their results were reported. The different visualization and interaction strategies of the examined tools required the differentiation of two classes: information visualization tools and visual programming tools. In the theoretical part a recently defined categorization was applied to each of the applications.

All of the evaluated applications allowed the visualization of the data in some way. But the specific temporal information contained in the data was correctly recognized by only two visualization tools. The others handled temporal data as a general sequence of values. This illustrates how rarely time-oriented data are considered when designing a visualization application. Due to the frequent relation of scientific data to the dimension time awareness for the specific needs when displaying temporal data has to be raised.

Visualization tools that support date values and treat them correctly as temporal data strictly position them along a linear axis. This view, though obvious and familiar to the human sense of time, limits the exploration and interaction abilities of the user. As already mentioned in the employed categorization, time can also be considered periodic. Especially when dealing with large data sets specific means for the discovery of periodicity would substantially improve the usability. The handling of periodic data seems to be generally underrepresented in both visualization tools and related scientific work and remains as a field for future research.

The available user interactions differ for each visualization application. Simpler tools often use separate windows and dropdown menus to interact with the displayed data. The more functionalities are provided to the user, the more the user's concentration is focused on the visualization itself and its direct surroundings. Tableau and Spotfire both heavily employ dragging and dropping items, but follow different concepts. Spotfire widens the possibilities when dragging by appearing icons, so-called drop targets, that represent different actions. Unlike Spotfire, Tableau keeps it simple and lets the user drop data variables on fixed positions on the screen. Due to the positions of the variables the user gets always visible feedback about what is currently visualized. Using Spotfire, not all filters were visible at a time which sometimes led to confusion. Though this is a subjective impression the structuring of user interfaces of information visualization tools is definitely an important field for further research.

Information visualization and visual programming tools follow different approaches. Using a visual programming application, the user has to think vice versa in some way: to display data, the visualization needed has to be considered first because the right modules to be connected together have to be chosen. Information visualization tools allow the user to read in the data set first and then combine data variables to create visualizations. Some tools, e.g., Tableau, even choose a default visualization for the chosen data to help the user in getting quick results. This allows the user to easily experiment with different views. Therefore, researchers focusing on the exploration of their data set and the discovery of revealed characteristics will choose an information visualization tool rather than a visual programming tool.

11 Technical Data

11.1 Software Version Numbers

Spotfire 2.0
Tableau 3.0
XmdV Tool 7.0
ILOG Discovery Preview Version
CViz 1.04
OpenDX 4.4.4
AVS/Express 7.1.1
GeoVista Studio 1.2
IRIS Explorer 5.2

11.2 Test PC

Intel Pentium 4, CPU 1500 MHz
512 MB SDRAM
NVidia GeForce 2 MX
Windows 2000

Acknowledgements

I would like to thank Silvia Miksch, who aroused my interest for information visualization and encouraged me to concentrate on working in this field. Special thanks go out to Alessio Bertone for his continuous support and constructive criticism during my work.

Additional thanks go out to the students I was allowed to meet at the university, who shared my interest in computer science and sometimes taught me more than some of the professors. I will never forget your friendly welcome to the circle of ”geeky” people and, of course, the programming sessions at nighttime.

I cannot end without thanking my family and friends, on whose constant encouragement and support I have relied throughout my time at the university. I am especially grateful to my brother Michael who provided crucial help with using LaTeX and always knew the answer for every organizational problem.

Bibliography

[1] Avs advanced visual systems website [online]. June 2007. Available from: http://www.avs.com/ [cited 2007-06-24].

[2] Cviz website [online]. June 2007. Available from: http://www.alphaworks.ibm.com/formula/CViz [cited 2007-06-23].

[3] Devise website [online]. June 2007. Available from: http://www.cs.wisc.edu/~devise/ [cited 2007-06-10].

[4] Geovista studio website [online]. June 2007. Available from: http://www.geovistastudio.psu.edu/jsp/index.jsp [cited 2007-06-24].

[5] Hotels in great britain [online]. 2007. Available from: http://www.4hotels.co.uk/ [cited 2007-10-24].

[6] Ilog discovery website [online]. June 2007. Available from: http://www2.ilog.com/preview/Discovery/index.html [cited 2007-06-24].

[7] In-spire website [online]. June 2007. Available from: http://in-spire.pnl.gov/ [cited 2007-06-11].

[8] Infovis toolkit website [online]. June 2007. Available from: http://ivtk.sourceforge.net// [cited 2007-06-10].

[9] Iris explorer website [online]. June 2007. Available from: http://www.nag.com/Welcome_IEC.asp [cited 2007-06-15].

[10] Jung website [online]. June 2007. Available from: http://jung.sourceforge.net/ [cited 2007-06-10].

[11] Opendx website [online]. June 2007. Available from: http://www.opendx.org [cited 2007-06-18].

[12] Piccolo website [online]. June 2007. Available from: http://www.cs.umd.edu/hcil/piccolo/ [cited 2007-06-10].

[13] Prefuse toolkit website [online]. June 2007. Available from: http://prefuse.org/ [cited 2007-06-10].

[14] Refviz website [online]. June 2007. Available from: www.refviz.com [cited 2007-06-10].

[15] Spotfire website [online]. June 2007. Available from: http://www.tibco.com/software/business_intelligence/ [cited 2007-06-13].

[16] Tableau website [online]. June 2007. Available from: `http://www.tableausoftware.com/` [cited 2007-06-15].

[17] Uk national air quality archive [online]. June 2007. Available from: `http://www.airquality.co.uk/archive/data_and_statistics_home.php` [cited 2007-06-23].

[18] Xmdv tool website [online]. June 2007. Available from: `http://davis.wpi.edu/xmdv/` [cited 2007-06-21].

[19] Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, and Christian Tominski. Visualizing time-oriented data - a systematic view. In *Computers and Graphics*, volume 31, pages 401–409. Elsevier, 2007.

[20] Robert Amar and John Stasko. A knowledge task-based framework for design and evaluation of information visualizations. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 143–150. IEEE Computer Society Press, 2004.

[21] Todd Barlow and Padraic Neville. A comparison of 2-d visualizations of herarchies. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 131–138. IEEE Computer Society Press, 2001.

[22] John V. Carlis and Joseph A. Konstan. Interactive visualization of serial periodic data. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, pages 29–38. ACM Press, 1998.

[23] Luca Chittaro and Carlo Combi. Representation of temporal intervals and relations: Information visualization aspects and their evaluation. In *Proceedings of the Eigth International Symposium on Temporal Representation and Reasoning*, pages 13–20. IEEE Computer Society Press, 2001.

[24] Mei C. Chuah and Stephen G. Eick. Information rich glyphs for software management data. *Computer Graphics and Applications, IEEE*, 18(4):24–29, 1998.

[25] Chaouki Daassi, Laurence Nigay, and Marie-Christine Fauvet. A taxonomy of temporal data visualization techniques. *Information-Interaction-Intelligence*, 5:41–63, 2006.

[26] Alan M. Eachren. *How Maps Work: Representation, Visualization, and Design*. Guilford Press, 1995.

[27] Carla M. Dal Sasso Freitas, Paulo R. G. Luzzardi, Ricardo A. Cava, Marco A. A. Winkler, Marcelo S. Pimenta, and Luciana P. Nedel. Evaluating usability of information visualization techniques. In *Proceedings of the 5th Symposium on Human Factors in Computer Systems*, pages 40–51. ACM Press, 2002.

[28] Susan Havre, Beth Hetzler, and Lucy Nowell. Themeriver: In search of trends, patterns, and relationships. In *IEEE Transactions on Visualization and Computer Graphics*, volume 8, pages 9–20. IEEE Computer Society Press, 2002.

[29] Werner Horn, Christian Popow, and Lukas Unterasinger. Metaphor graphics to visualize icu data over time. In *Proceedings of the Intelligent Data Analysis in Medicine and Pharmacology. Workshop Notes of the ECAI-98 Workshop.*, 1998.

[30] Daniel A. Keim. Visual techniques for exploring databases. In *Tutorial Notes: Third International Conference on Knowledge Discovery and Data Mining*, pages 1–121. AAAI Press, 1997.

[31] Alfred Kobsa. An empirical comparison of three commercial information visualization systems. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 123–130. IEEE Computer Society Press, 2001.

[32] Robert Kosara and Silvia Miksch. Asbruview: Capturing complex, time-oriented plans - beyond flow-charts. In *Second Workshop on Thinking with Diagrams*, 1998.

[33] Diogenes Laertius. *Lives of Eminent Philosophers*. early 200s AD.

[34] Qing Li and Chris North. Empirical comparison of dynamic query sliders and brushing histograms. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 147–153. IEEE Computer Society Press, 2003.

[35] E. Morse and M. Lewis. Evaluating visualizations: using a taxonomic guide. *International Journal of Human Computer Studies*, 53:637–662, 2000.

[36] Darius Pfitzner, Vaughan Hobbs, and David Powers. A unified taxonomic framework for information visualization. In *Proceedings of the Asia-Pacific symposium on Information visualization*, pages 57–66. Australian Computer Society, Inc., 2003.

[37] Catherine Plaisant, Brett Milash, Anne Rose, Seth Widoff, and Ben Shneiderman. Lifelines: Visualizing personal histories. In *Proceedings of the ACM CHI96 Conference on Human factors in Computing Systems*, pages 221–227. ACM Press, 1996.

[38] George G. Robertson, Stuart K. Card, and Jock D. Mackinlay. Information visualization using 3d interactive animation. *Communications of the ACM*, 36(4):57–71, 1993.

[39] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336–343. IEEE Computer Society Press, 1996.

[40] Christian Tominski, James Abello, and Heidrun Schumann. Axes-based visualizations with radial layouts. In *Proceedings of the 2004 ACM Symposium on Applied computing*, pages 1242–1247. ACM Press, 2004.

[41] Lloyd Treinish and Deborah Silver. Worm plots. *IEEE Computer Graphics and Applications*, 17:17–20, 1997.

[42] Fernanda B. Viegas, Martin Wattenberg, and Kushal Dave. Studying cooperation and conflict between authors with history flow visualizations. In *Proceedings of the 2004 conference on Human factors in computing systems*, volume 6, pages 575–583. ACM Press, 2004.

[43] Ulrika Wiss and David Carr. A cognitive classification framework for 3-dimensional information visualization. Technical report, Lulea University of Technology, 1998.

[44] Rebecca Xiong and Judith Donath. Peoplegarden: Creating data portraits for users. In *Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology*, pages 37–44. ACM Press, 1999.

[45] M. Zhou and S. K. Feiner. Visual task characterization for automated visual discourse synthesis. In *Proceedings of the CHI*, pages 392–399. ACM Press, 1998.

Curriculum Vitae

Eva Maria Wohlfart
born on May 24th, 1983, Eisenstadt, Austria

Address: 1070 Wien, Kaiserstrasse 92/2/11, Austria
Cell: +43 / 664 / 92 073 92
Mail: eva@ull.at

Education

2005 – 2007	Master in Computer Science, Major Media and Design Vienna University of Technology, Vienna, Austria
2001 – 2005	Bachelor in Computer Science, Major Media and Design Graduation with honors Vienna University of Technology, Vienna, Austria
1993 – 2001	Secondary school with scientific bias, A-levels with distinction Wiener Neustadt, Austria
1989 – 1993	Elementary school in Wiener Neustadt, Austria

Job

since 2006	Web design and Programming sIT Solutions, Vienna, Austria
2002 – 2007	Teaching of introductory java lectures Vienna University of Technology, Vienna, Austria
2003 – 2005	Graphic and Web design (3 months) Narosy KEG, Wiener Neustadt, Austria
2000 – 2001	Organization and management of two youth camps (2 months) Lower Austrian Jungschar, Austria

Hobbies Tap dance, Singing, Jazz dance, Guitar