Eurographics Conference on Visualization (EuroVis) 2013
B. Preim, P. Rheingans, and H. Theisel
(Guest Editors)

Volume 32 (2013), Number 3

VisRuption: Intuitive and Efficient Visualization of Temporal Airline Disruption Data

P. Rosenthal L. Pfeiffer N. H. Müller P. Ohler

Chemnitz University of Technology, Germany

Abstract

The operation of an airline is a very complex task and disruptions to the planned operation can occur on very short notice. Already a small disruption like a delay of some minutes can cost the airline a tremendous amount of money. Hence, it is crucial to proactively control all operations of the airline and efficiently prioritize and handle disruptions. Due to the complex setting and the need for ad hoc decisions this task can only be carried out by human operation controllers. In the field of airline operations control there exists already a vast variety of different software in productive use. We analyze the different approaches from two of the market leaders and identify problematic design choices. We take into account this analysis and develop a set of rules for an intuitive visualization of airline disruption data. Finally, we introduce our tool for visualizing such data which complies to these rules. The visualization enables the user to gain a fast overview over the current problem situation and to intuitively prioritize different problems and problem hierarchies. The efficiency of the design is evaluated with the help of a user study which shows that the new system significantly outperforms the current state of the art.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications—

1. Introduction

Nowadays, large airlines like Delta Air Lines or United-Continental operate more than 700 aircraft and fly to more than 300 destinations with several thousand operated flights per day. Managing the vast amount of resources and processes is a very complex task and makes computer-aided operation irreplaceable. In addition, there is a multitude of problems which can occur every day during airline operation, e. g. bad weather, broken aircraft, or sick employees. Since already small delays can be very expensive [CTA04, Now09], airlines have to react to disruptions expeditiously and resolve them as efficiently as possible. Such crucial and sometimes costly decisions can not be carried out automatically. Thus, nearly every airline operates a department of operations control where a team of operators observes the current problem situation and decides which problems to solve first and how to solve them.

All controllers need many years of experience, particularly in situations of extraordinary problem load like airport or airspace closures, where problems might not be solvable anymore but can only be alleviated. For instance, after the volcanic ash cloud in 2010 some international airlines were able to resume normal operation within some days while other airlines were still not fully operational one week after reopening of airspaces [Mou10]. In such situations the economic severity of operations control gets obvious. For example British Airways reported a daily loss of 20 million British pounds [MHR10].

In this paper we analyze the problem of providing a good visualization for airline disruption data. First, we define the data and the underlying visualization problem. We extract a set of requirements for a good visualization of airline disruption data and review the current state of the art of operations control tools. Afterwards, we introduce the design and interaction mechanisms of our tool VisRuption which follows the design requirements. A user study shows that the proposed approach significantly outperforms the current state of the art.

2. Related Work

Visualization of time-varying and time-oriented data has a long tradition in information visualization. Especially when visualizing "live" data, i. e. data with a direct relation to the current point in time, the data points include a strong link to reality. A comprehensive survey of the state of the art of visualizing time and time-oriented data is given by Aigner et al. [AMST11].

Since there exists a huge amount of different types of time-oriented data, the variety of approaches for visualizing such data is abundant. One main direction is the analysis of time series to detect trends and predict future developments. In this context the LifeLines project [WPS*09] provides a tool to summarize a huge amount of events to detect patterns and time-oriented structures. LifeFlow [WGGP*11] aims in a similar direction by visualizing event sequences in linked views. Buono et al. [BPS*07] follow the same lines and develop a tool which facilitates not only an analysis of time series but also tries to forecast the future behavior in a river plot integrating statistical measures about the prediction. In contrast, our goal is to provide the user with a tool to get a global understanding of the current situation.

For the detailed exploration of industrial production data Luo et al. [LTL*10] present an interactive tool which utilizes a fish-eye view for the time line of processes. This allows a detailed inspection of specific data points in the context of the whole data set. Kosara and Miksch [KM01] show the hierarchical structure of process chains in a temporal tree view. In terms of hierarchical data, Card et al. [CSP*06] provide a tool for visualizing and exploring the changes of one hierarchy over time. In contrast, Burch et al. [BBD08] present an approach which allows a comparison of time lines in the context of their hierarchical relation.

Similar to our problem setting is the work of the Visdom team [WRF*11], dealing with flood disaster management. The authors developed a tool which uses a linked view [Hen98] of the actual flood-simulation space and a hierarchically branched time line to visualize the parallel possible worlds. Piringer et al. [PBB12] propose a design to increase situation awareness for road tunnel surveillance. The system provides intuitive access to respective video data through an overview over the spatio-temporal relations in the tunnel.

To our knowledge, there exists no approach which systematically analyzes airline disruption data and provides a visualization for efficient airline operations control. Some of our design choices are inspired by previous approaches. We translate the ideas to our very specific setting and incorporate them into our tailored tool.

3. Design Process

VisRuption was designed in an iterative process [SMM12] and in close collaboration with industry partners. In the first phase we interviewed experts from Lufthansa Systems AG and 15 international airlines to understand the actual data structure and extract first design requirements. The final set of requirements was obtained during visits to four different control centers, where the work of the controllers was observed and analyzed in several informal interviews. Afterwards, state-of-the-art tools were reviewed and analyzed for strengths and flaws. Combining all available information resulted in a first mock-up which was reviewed and discussed with 21 operation controllers at the Airline Forum 2011.

A first interactive prototype was implemented taking into account this feedback. The prototype was constantly redesigned during several rounds of informal evaluation involving experts from Lufthansa Systems AG and two airlines. VisRuption was finally implemented in Java with the ability to be plugged into the database and infrastructure of NetLine [Luf], an operations control software used by more than 50 airlines and developed by Lufthansa Systems AG.

4. Problem Definition

At each point in time the operational situation of an airline can be considered as a set of problems, i. e. a set of points p in problem space P. Each problem holds a multitude of variables which differ from airline to airline. In our discussions with experts from several airlines we extracted a representative set of the most common and important variables, which also most of the existing tools concentrate on:

Problem ID: This unique number allows a distinct definition and handling of problems.

Parent ID: The ID of the problem which caused the current problem. This is not always the root problem, which is the final cause for the problem, but just the parent of the current problem in the hierarchy.

Severity: This number between 0 and 1 is an airline specific judgment about how severe a specific problem is. It is derived from long term evaluation of problems and their impact on airline customers and yield.

Location: The IATA code [IAT] of the location of the problem.

Problem Time: This is the future point in time when the problem actually occurs if it is not solved. In many cases the problem time is the departure time of a flight, e. g. if a flight is missing staff. However, it is not limited to this, e. g. for problems caused by duty restrictions.

Lead Time: This time implicitly encodes the time period needed for solving the problem. An airline operator has to start solving a problem at latest at its lead time.

Type: The problem type is an airline specific classification of problems into different classes like missing crew, delayed flight, or airport closure.

Furthermore, each problem can hold several additional properties like a detailed description, airline-specific airport or flight measures, and remarks by other departments of the airline. However, those properties are only relevant during closer inspection.

For the design of an effective visualization it is not only important to specify the quality or nature of the problems but also their quantity, i. e. typical sizes of data sets. This quantification is very hard since sizes of airlines can vary in more than two orders of magnitude regarding the number of flights as well as the number of aircraft and employees. Furthermore, the structure of an airline (world-wide operating passenger airline, local carrier, or freight airline) has significant impact on the nature of a typical problem set. Thus, our visualization of the problem situation will be laid out very flexibly to handle sets of problems with very different sizes and time frames. For large airlines, the typical number of problems visible to operators is 500.

One last, but fundamental, feature of the problem set is its temporal character. The whole problem situation can change dramatically from one point in time to the other just by the introduction of a new problem, e. g. the closure of an airport. Besides causing problems to develop and pass, the time dependency of the data also heavily influences prioritization. A very severe problem with still three hours left to be solved can be less important than a problem with low severity and only three minutes left for solving. This interplay of severity and urgency is a major feature of the data and plays a key role for successful airline operations control.

5. Design Requirements

After the data space is specified in terms of dimensions and typical data-set sizes the question of needed capabilities for the desired system arises. Specifically, we are interested in how a typical user would imagine a "perfect" visualization tool. As a result we extracted a list of requirements all relevant experts in general agreed on. The desired design goals of a visualization tool for airline disruption data are:

1. The visualization must provide a good overview over the current overall situation of the airline. This is essential for large airlines or in severe situations with many problems and facilitates operators and coordinators to rapidly detect and proactively react to new developments.
2. An operator must be able to prioritize the given set of problems reliably and fast. Here the main focus is on the correctness of the prioritization, since processing unimportant problems can cause a significant loss of time.
3. Since the hierarchy of the problems is a crucial feature it should be clearly visible in the visualization. Solving a complicated problem that is severe and urgent is inefficient if it is caused by a problem which can be solved much faster.
4. The visualization should support the well-established mechanisms of sorting and filtering problems. Performing such standard routines as well as the solution or introduction of problems should preserve the mental map [MELS95] as best as possible.
5. The visualization must concentrate on a minimal set of primitives to produce an expressive and effective visualization [SJ07] with minimal disturbance of the work flow. All important features should be easily identifiable and all visual elements should have an important meaning. Color should only be used when really needed to highlight very important features and taking into account human visual perception [Tuf90].
6. Data should be visualized in an objective fashion leaving judgment and prioritization to the operator.
7. The visualization should facilitate the operation with new input paradigms like touch displays or gesture recognition. Although current operation controllers interact with keyboard and mouse, it is supposed by collaboration partners that those techniques might get in productive use in the near future.

6. State of the Art

In the previous section we list the main requirements for an effective visualization of airline disruption data. There exists a big market for airline software with several competitors. Most of the tools are very sophisticated and well-optimized for the static part of the airline operation pipeline, i. e. the design of flight plans and generation of pairings. However, the efforts in terms of facilitating an intuitive operations control seem antiquated. In the following we will discuss three representative approaches which are in or were proposed for operational use. These approaches can be seen as state of the art in this field regarding the feedback of airline operators.

6.1. Problem List

The first approach is the problem list or table, as illustrated in Figure 1. All problems are listed with a set of variables in linear fashion. The list can be sorted according to one or more variables and the number of visible problems can be reduced by filtering with respect to user-specified criteria. Problem lists have been the first approaches for visualizing airline disruption data in times where graphics capabilities of computers were limited. The approach is still widely used in many commercial products, e. g. by the NetLine Problem Desk.

The visualization is perfectly adapted to sorting and filtering (Design Requirement 4.). However, getting an overview of the current problem situation (Design Requirement 1.) is nearly impossible when dealing with several hundred problems. Even the excessive use of sorting and filtering can provide only small insights into the overall problem space. This weakness induces a poor support for fast prioritization of problems (Design Requirement 2.). During the prioritization the whole problem list has to be filtered down to a small set of problems that can be compared side-by-side. During this process it can happen that problems which would need to be solved first are filtered out. Such incidents can even occur during the work of experienced operators and lead to suboptimal prioritization.

6.2. Linked View

A second type of problem monitor is the enhancement of the standard problem list by a linked view. One representative of such approaches is illustrated in Figure 2. The second view is typically used to integrate a small overview of the current situation. In the case of the Jeppesen Alert Monitor [Jep] the second view consists of a time line where problems are indicated by small symbols. Selection and filtering can be

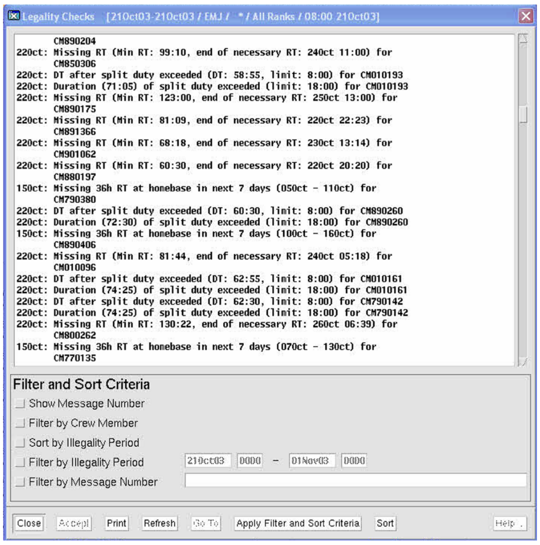


Figure 1: *Problem Desk of the NetLine Crew software by Lufthansa Systems AG [GST05, Hös09]. All current problems are listed in a tabular view which enables sorting and filtering for different dimensions like problem ID or problem time. (Image courtesy of Lufthansa Systems AG)*

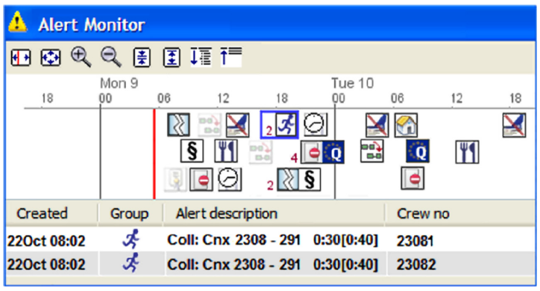


Figure 2: *Alert Monitor of the Crew Tracking Enterprise software by Jeppesen Systems AB. A linked view of problem list and time line with pictographs allows brushing and linking. (Image courtesy of Jeppesen Systems AB)*

done with the help of this additional view, which produces a respective set of problems in the problem list below.

Again, the approach facilitates sorting and filtering. In contrast to a standard problem list, the linked view enables the operator to get a rudimentary overview of the current problem situation. However, the pictograph view does not provide enough information to effectively prioritize problems without the problem list (Design Requirement 2.). Furthermore, the linked view disturbs the user's work flow with the arbitrarily colored pictographs (Design Requirement 5.). Similar to the standard table it is not possible to get an overview of the hierarchical structure of the problems (Design Requirement 3.).

6.3. 3D Problem Desk

A new approach for visualizing airline disruption data has been proposed by Hösel [Hös09]. It relies on visualizing the problem situation by representing the space of problems in 3D and perspective projection to the screen, see Figure 3. One dimension is used for encoding the time while the other two dimensions are used to distinguish between the problems. All airports are circularly arranged in segments on a plane perpendicular to the time axis. The size of the segments in terms of its angles is chosen uniformly such that all airports form a full circle. The width of each segment represents the number of problems which are currently connected to the airport. For each problem a "needle", i. e. a sphere with a line parallel to the time dimension, is placed on top of the airport's segment. The tip of the needle represents the lead time while the head of the needle is placed at the problem time. The hierarchical relation between problems is encoded by connecting lines.

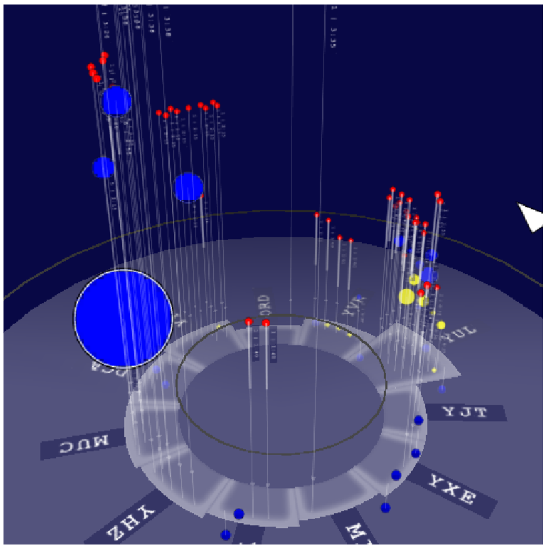


Figure 3: *Three-dimensional Problem Desk developed by Hösel [Hös09]. All airports are circularly arranged in segments. Time is running perpendicular to this disk. The problems at one airport are represented by "needles" above the airport's segment. The size of a segment is determined by the number of problems at the respective airport. Hierarchical connections between the problems are indicated by lines connecting the respective problems.*

Obviously, this approach heavily disturbs the user's work flow with meaningless visual primitives (Design Requirement 5.). The visualization is transferred into a pseudo-3D space without any need, consequently suffering severe perspective problems [TKAM06] like overplotting and low comparability of the problems (Design Requirement 2.). The visualization could be easily transformed to 2D by just unfolding the circle of airports to a line, which would already solve some of the problems. Furthermore, color is used to encode the type for each problem. This can hinder an objective prioritization and result in unwanted preference of problems by human cognition when fast processing is needed [HBE96]. Furthermore, all airports without current problems still capture space without contributing any information.

To our knowledge we are the first to present an approach for visualizing airline disruption data which fulfills all the design requirements from Section 5.

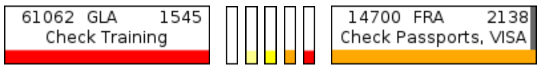
7. General Design

During the design of our visualization application we followed a top-down approach strictly respecting the design requirements from Section 5. Observing the nature of the problem space yields the conclusion that time is the only prominent dimension warranting a specific dimension in visual space. This also coincides with the human expectation of temporal data. In addition, urgency as one of the main features of disruption data is very well recognizable if problems are aligned with a time dimension. Discussions with domain experts revealed two additionally desired properties of the time dimension. First, most people preferred the idea of problems running from top to bottom as time passes by. This alignment is very natural and well-known from games like "Tetris". As a second important property the visualization should facilitate an absolute judgment of time distances, e. g. avoid clipping of time periods even if no problems are existent.

Further dimensions are not needed for data variables and can be used to distinguish or group problems. Here the hierarchical structure of the set of problems is the most prominent feature (Design Requirement 3.). Visualizing such hierarchies has a long tradition in the infovis community and plenty of different approaches exist [SHS11]. To avoid clutter and overplotting (Design Requirement 5.) we use an approach implicitly encoding the hierarchy, similar to sunburst [SZ00] and interring [YWR02]. In contrast to the circular approaches we have to maintain a linear layout like icicle plots [KH81, KL83] to facilitate an easy side-by-side comparison [PW06] of the problems with respect to time. Furthermore, there is no unique root in most cases, i. e. the disruption data set consists of a set of hierarchies. Consequently, we propose a 2D layout for the display of the problems where the second dimension provides room for sorting and filtering (Design Requirement 4.).

The choice of dimensionality together with our goal of minimal disturbance (Design Requirement 5.) implies the use of boxes as visual primitives for problems. This choice also maximizes the usage of screen space. Problem boxes are placed such that the lower border of the box coincides with the problem time, see Figure 5. The lead time of each problem is indicated by a small horizontal marker connected to the problem's box with a vertical line. Thus, the distance on the screen between the box and the "now"-line at the display's bottom directly encodes the time left to solve the problem while the lead time marker shows the point in time when the operator should start solving the problem at the latest.

Each hierarchy is visualized in a separate vertical band with the root node's box as lower border. The boxes for the child nodes are vertically aligned with their parent node's box. If the boxes of a node and its child are overlapping the node's box is drawn above the child's box. This behavior is inspired by the fact that child problems are unlikely to have a higher priority for the operator than their parents. The bands of different hierarchies are placed side by side. See Figure 5 for an illustration of the hierarchy visualization. So far the position of each hierarchy on the x-axis is not fixed. Consequently, the positions can be used for sorting the problems according to any data variable (Design Requirement 4.). Note, that this sorting is always done with respect to the hierarchical structure.



(a) (b) (c)

Figure 4: *Visualization of problems as boxes. (a) The box of a problem consists of three rows. Problem ID, airport code, and problem time are indicated in the first row. The problem type description is shown in the second row. (b) Severity of each problem is color-coded in a small bar at the bottom of the box using five discrete colors from white over yellow to red. (c) Similar problems with equal problem time are grouped into one box. A grey marker on the right side indicates the number of problems.*

Another important feature is the size of the problem boxes. The width of each box implicitly encodes the breadth of the subtree at the respective problem. More precisely, the width of each leaf problem is chosen constant with respect to screen space. The width of each other node's box is chosen such that it equals the sum of the widths of its children. The height of a box should be dependent on the content. In consultation with the domain experts we found that problem ID, airport code, problem time, problem type, and severity should be visible in the boxes of urgent problems. As illustrated in Figure 4 (a), we have decided to use a three-row layout for each problem. The problem ID, airport code, and exact problem time are indicated in the upper row. A short description of the problem type can be found in the second row. The lower row is implemented as a colored bar indicating severity.

Using color maps to encode variables is a very complex but powerful design choice [HBE96, Tuf90]. Intentionally we did not use any color in the design of the visualization to reserve this unique feature for one of the two most important variables, the severity. Having only this colored part per problem will give special attention to the variable and allow

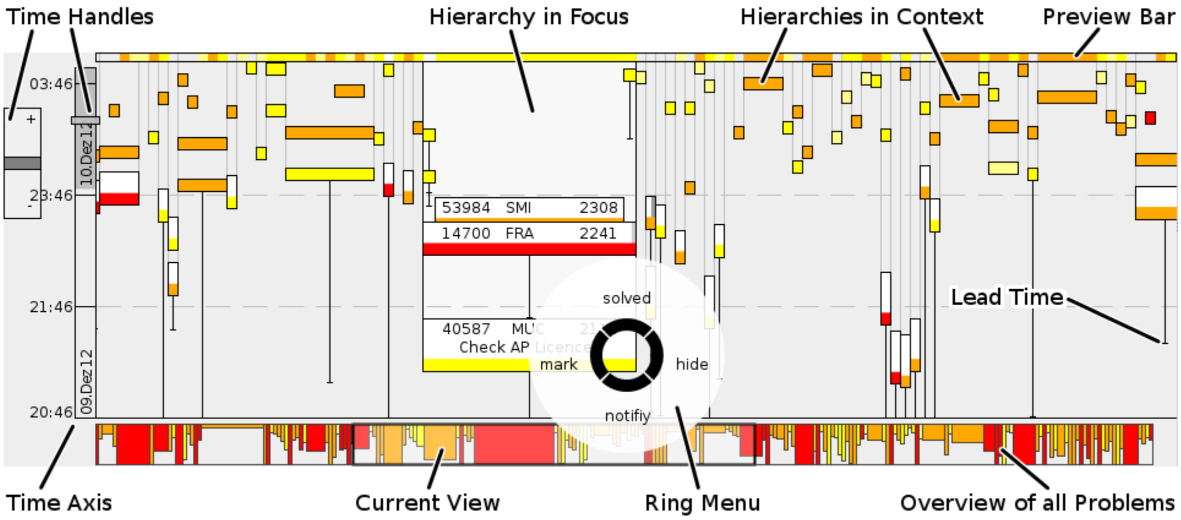


Figure 5: *Annotated screen shot of VisRuption showing a serious problem situation with 1,000 problems. The visible time span is set to seven hours and hierarchies are sorted with respect to airport.*

very fast identification of severe problems. To further enhance the comparability we have chosen a discrete color map following the intuitive comprehension of severity [RTB96]. The scheme ranging from white over yellow to red is illustrated in Figure 4 (b) and was favored by domain experts. Other schemes, e. g. from ColorBrewer [Bre05], could be used as well.

A typical property of disruption data is the fact that several problems in a hierarchy can be very similar, e. g. a delayed flight causes the whole crew to be late at the connecting flight. Such problems have the same problem time and are most likely solvable at once. After several discussions with domain experts we decided to show in such cases only the box of the most severe problem and indicate the number of hidden problems by a grey bar on the right side of the problem's box, see Figure 4 (c). The lightness indicates the number of hidden problems: 80% for one or two, 60% for three to eight, and 40% for more hidden problems.

Many problems are very far away from now in terms of problem time and only act as context for some important root problems. Hence, the displayed information and used screen space for such problems should be minimized. We decided to split the time dimension of VisRuption into three parts. The partition is even with respect to screen space. However, the visualized time periods differ which was inspired by one-dimensional focus+context methods [RC94]. The middle part represents twice the time of the lower part and in the upper part the visualized time span is once more doubled, as illustrated in Figure 5.

All urgent problems, i. e. the ones in the lower part, are visualized with the full set of information. For problems in the middle part the problem type description is omitted. Problems in the upper part are only visualized by their severity bar. See Figure 6 for an illustration of the differently detailed boxes. This successive zoom, similar to the distortion used by Powsner and Tufte [PT94], allows a detailed inspection of problems with high urgency while maintaining an overview over a long period of time.

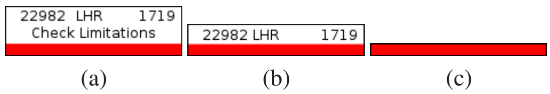


Figure 6: *Problem boxes for the same problem but with different relative points in time: (a) problem time in the lower third of the desk, (b) problem time in the middle third, (c) problem time even further away.*

8. Interaction

Disruption data is temporal and highly dynamic in nature. To maintain the mental map [MELS95] of the problem situation (Design Requirement 4.) it is very important that changes to the problem space or the visualization result in only minimal visual changes. This goal is achieved by animating all visual modifications smoothly. The speed of the animations was determined with the help of domain experts to master the trade-off between efficient work and preservation of the mental map. Changing the sorting criterion can obviously not preserve the mental map and is done without any animation.

Currently, the main interaction devices of crew controllers are mouse and keyboard which we also utilize. However, all paradigms and techniques have been chosen to be easily usable by touch interfaces as well (Design Requirement 7.).

Hovering over a problem shows all details of the subjacent problem(s) in a semitransparent pop-up. Clicking the right mouse button opens a ring menu [GLG00, SKSH01], centered at the mouse pointer's position. Moving the mouse in the direction of a circle segment accentuates the respective segment. Releasing the mouse button selects the respective option. The ring menu when clicking on top of a problem contains problem specific commands, see Figure 5. Clicking right on free space opens the ring menu for sorting and filtering.

When visualizing the whole set of problems of a large airline it is not possible to view all problems in full detail on one screen (Design Requirement 1.). We tackle this problem of information overflow in three ways, respectively illustrated in Figure 5. First, we introduce a focus+context technique for the visible hierarchies. Distortions of the visual space introduced in several non-linear techniques [MRC91, SB94] would not be acceptable due to loss of comparability of problems in time (Design Requirement 2.). Instead a focus+context technique similar to table lens [RC94] is used. The focus region consists of exactly one problem tree. Each problem in focus is drawn in full detail. All other hierarchies are considered as context. All problems in context are drawn without any textual content and shrink to a minimal width. This width is obtained by assigning a width of just eight pixels to leaf problems and propagation to the root of each problem tree. Note, that severity is still prominently visible for context problems to prohibit the missing of severe problems. In addition, the lead time is drawn for context problems only if it is just one hour away from now to reduce display clutter. The change of focus is executed by clicking left on a context problem tree.

Even with the focus+context technique it might happen that not all problem trees fit onto the screen when dealing with severe problem situations. For retaining the overview of problems not visible in the current view (Design Requirements 1. and 6.) we introduce a summary bar at the bottom of the screen, see Figure 5. The bar shows a small box for each hierarchy. The lower border of the box corresponds to the problem time of the respective root problem and the color to the maximum severity of all problems in the hierarchy. A smart slider on top of the summary bar allows changing the position of the current view.

In several cases it might be helpful to also adapt the visible time period to the current problem load. We provide two interaction mechanisms to achieve this. A handle in the upper left corner similar to a track ball can be dragged such that the time frame is continuously adjusted. A more accurate zooming into a specific time period is possible with a handle on the time axis. Dragging the handle and dropping it at a specific point in time selects a time period which corresponds to the new viewable time frame. Especially in the case of close zooming, it can happen that problems are further away than the currently visible time span. Still, it might be beneficial to be aware of them. We visualize such problems by their severity color in a preview bar on top of the window, see Figure 5.

9. User Study

Adequately assessing quality and performance of a system as complex and heavily human-centered as VisRuption necessitates taking respective users' feedback into account. Because of limited resources and against plenty of nondisclosure resistances we managed to conduct a user study comparing VisRuption and NetLine by Lufthansa Systems AG, see Figure 1, with real NetLine users and real data.

9.1. Method

Participants and design

25 attendees of the Airline Forum 2012, one of the premier airline IT user conferences, participated in the study. All of them were presented with both of the systems before completing an electronic questionnaire relating to the systems. The participants were randomly invited to take part in the study (convenience sampling) and randomly assigned to start with either the established NetLine Tool or VisRuption.

From the participants 88% were male and 12% female. The average age of the participants was 39.4 years (SD=8.1) with an average working experience in the field of crew control of 11.8 years (SD=8.1). The educational background was quite evenly distributed ranging from High-School or "other" forms of educational degrees up to a Ph.D. or higher. We used a 1×2 factorial design with the within-subject factors of starting with either the NetLine tool or VisRuption, i. e. each participant did the test with both tools in random order. For each tool, the participants were asked to identify certain problems to answer specific questions.

Procedure

Participants were tested individually at a booth on-site during the conference. Two experimenters welcomed them and led them to one of two computers, which were both equipped with mouse, keyboard, and a 24" screen with FullHD resolution, similar to the workstations in today's typical airline control centers. Afterwards, the participants were instructed on how to use the respective tool, including visualization and interaction paradigms.

In close collaboration with domain experts, we developed a set of five questions that are relevant and representative for the everyday work with a problem desk in operations control. Two versions were generated by changing airports, times, or crew members and their equivalence was proven in a pre-test. The questions were of the following types:

Q1: What is the problem time of the problem with ID x?
Q2: What is the ID of the next problem at airport x?
Q3: How many problems will occur in the next x hours at airport y?

Q4: How many child problems does the problem at airport x and time y cause?
Q5: A problem at which airport initially causes the problem of type x and time y?

The right answers had to be chosen via multiple choice with one target and three distractors each.

Whether the answer was chosen correctly and how much time the participants needed to identify the problem was recorded. Afterwards, they were asked to fill in a questionnaire for demographic information as well as their opinion on the new VisRuption tool compared to the established NetLine software.

Materials

The experimental situation was programmed in Java which allowed a reliable measure of the time as well as an unobstructed multiple-choice test. Both tools operated on the same snapshot of a real-world problem situation provided by Lufthansa Systems AG. The situation originated from an average day of a medium-sized European airline with 20 destinations and 100 problems. The questionnaire in its electronic form was generated using the software LimeSurvey, and the results were interpreted using SPSS 20.

9.2. Results

Basic Analysis

To rule out sequence effects, we calculated a Mann-Whitney-U-Test check, whether it made a difference if participants had to start with the established NetLine or the VisRuption software. The two groups did not differ significantly:

$$\begin{aligned} \text{Correctness}: \quad & U_{\text{NetLine}} = 69.0 & p = 0.672 \\ & U_{\text{VisRuption}} = 71.5 & p = 0.806 \\ \text{Time}: \quad & U_{\text{NetLine}} = 75.0 & p = 0.936 \\ & U_{\text{VisRuption}} = 63.0 & p = 0.467 \end{aligned}$$

For all participants we summed up the individual scores per question to generate a mean-score-equivalent as well as a mean-time for completion. Afterwards, we had a look at the two different scores of being correct and how long the participants needed. Since the assumptions for parametric tests had been violated, nonparametric tests were used. The Wilcoxon-Test showed a significant difference and a strong effect size ($Z = -3.67,\ p < 0.001,\ r = -0.73$) in the correctness of the answers. The mean correctness was 0.4 for NetLine compared to 0.8 for VisRuption. The Wilcoxon-Test also revealed a significant time difference combined with a medium effect size ($Z = -2.68,\ p < 0.01,\ r = -0.54$). Answering each question took on average 155 seconds with the NetLine tool and only 107 seconds with VisRuption. Details can be found in Table 1. The poor correctness values and relatively good times for NetLine at questions **Q3**, **Q4**, and **Q5** can be explained by guessing after some time of examination, as we found out from qualitative feedback of participants.

	$c_{\text{NetL.}}$	$c_{\text{VisR.}}$	$t_{\text{NetL.}}$	$t_{\text{VisR.}}$
Q1	0.56	0.88	197 sec	79 sec
Q2	0.44	0.92	155 sec	65 sec
Q3	0.24	0.68	140 sec	123 sec
Q4	0.08	0.64	105 sec	89 sec
Q5	0.48	0.72	177 sec	178 sec

Table 1: *Average correctness c and time t for each question using the NetLine Problem Desk or VisRuption.*

Correlations

Looking for relationships in the data, we assumed that the score of participants would be higher the longer they were working with the respective software during the test. However, a two-tailed bivariate correlation showed no relationship between time and correctness with either of the tools:

$$\begin{aligned} r_{\text{NetLine}} &= 0.32(23) & p_{\text{NetLine}} > 0.12 \\ r_{\text{VisRuption}} &= 0.14(23) & p_{\text{VisRuption}} > 0.49 \end{aligned}$$

Further analysis with the questionnaire showed a surprising effect: When correlating the experience on working with NetLine and the scoring or time for answering with NetLine, there is no visible effect:

$$\begin{aligned} \text{Correctness}: \quad & r = -0.101(23) & p > 0.63 \\ \text{Time}: \quad & r = 0.35(23) & p > 0.86 \end{aligned}$$

However, there is a significant interaction when asked about experience with operations control software other than NetLine ($r = -0.476(23),\ p < 0.02$), i.e. the longer people had experience with competing operations control tools the lower was their correctness with NetLine.

Questionnaire

The participants answered a questionnaire using 5-point Likert-scales, ranging from (1=NetLine) to (5=VisRuption) whether or not they experienced certain aspects better on either one of the tools. The questions were categorized into four groups, where the first question asked for the general opinion on which approach the participants liked more. While the second group of questions asked for general aspects of the tools, the third group concentrated on the layout. Finally, the fourth group of questions asked which approach could visualize certain data details better. An overview of the results from the questionnaire is given in Table 2.

Discussion

Looking at the results, we found some surprising insights. First, the participants overall preferred working with VisRuption compared to NetLine, visible due to the strong numbers in Table 2. VisRuption is not only more time-efficient

	NetLine 1	2	↔ 3	4	VisRuption 5	Mean	SD
General Opinion	0%	<10%	<20%	<70%	<30%	4.00	0.722
Functionality	0%	<10%	<20%	<70%	<20%	3.92	0.717
Reliability	0%	0%	<30%	<50%	<40%	4.14	0.770
Usability	0%	<10%	<30%	<50%	<30%	3.92	0.812
Efficiency	0%	0%	<10%	<70%	<30%	4.19	0.602
Clarity	0%	<20%	<30%	<40%	<20%	3.54	0.977
Arrangement	<10%	0%	<40%	<50%	<20%	3.71	0.908
Structurization	0%	0%	<30%	<70%	<20%	3.96	0.624
Comprehension	<10%	<10%	<40%	<50%	<20%	3.63	0.970
Colors	0%	<10%	<30%	<50%	<30%	3.87	0.850
Visualization	0%	0%	<20%	<70%	<30%	3.87	0.850
Problem Time	0%	0%	<10%	<30%	<70%	4.58	0.654
Lead Time	0%	0%	<20%	<20%	<70%	4.57	0.728
Problems per Hour	<10%	0%	<10%	<40%	<60%	4.36	1.075
Problems per Airport	0%	0%	<10%	<30%	<70%	4.58	0.654
Hierarchy of Problems	0%	0%	<10%	<20%	<70%	4.67	0.637

0%
<10%
<20%
<30%
<40%
<50%
<60%
<70%

Table 2: *Overview of the results from the questionnaire. A color-coded histogram shows for each question the percentage of answers in the respective category. Additionally, we indicate the mean value and the respective standard deviation.*

(on average 40 seconds faster than working with the NetLine Problem Desk) but also provides a twice as high score of correctness. In addition it was rated very favorably compared to NetLine by the participants in regards to visualization, layout, and personal opinion (Mean=4, SD=0.72).

Furthermore, we found some interesting facts about the problem-visualization of the established NetLine Problem Desk. First, longer work experience with the NetLine System has no impact on either the rate of success or on the time it takes a participant to identify a problem. Even worse seems the information that if someone has already been working with competing software before, working with the NetLine Problem Desk is even more hindered. Since there is a significant difference between the time needed with NetLine and with VisRuption, we assume that the participants were able to interpret the information and get an overview of the situation much faster with our visualization.

All in all the tailored visualization techniques of VisRuption show a high potential in regard to information representation and relaying problems. Although the correctness score of VisRuption is just 80% within the test, two things must be considered: First, the participants had a background of working with the NetLine system and still only managed to produce a correctness of 40% within their experienced environment. And second, the participants had no training beforehand with VisRuption - the reported results were established on first-time-first-use. With proper training and more than just five problems to exercise on, we assume an even better performance afterwards.

10. Conclusions

We have presented a detailed analysis of the problem setting and the requirements to the visualization of airline disruption data. Several flaws were discussed with respect to the current state of the art. Afterwards, we have introduced the tool VisRuption, which strictly follows the design requirements and was developed in close collaboration with domain experts. Finally, we discussed the practical performance of VisRuption on the basis of a user study carried out with domain experts.

Acknowledgments

We would like to thank Lufthansa Systems AG and all contributing airlines for their support and time for discussions.

References

[AMST11] Aigner W., Miksch S., Schumann H., Tominski C.: *Visualization of Time-Oriented Data*. Springer, 2011. 1

[BBD08] Burch M., Beck F., Diehl S.: Timeline trees: visualizing sequences of transactions in information hierarchies. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (2008), pp. 75–82. 2

[BPS*07] Buono P., Plaisant C., Simeone A., Aris A., Shneiderman B., Shmueli G., Jank W.: Similarity-based forecasting with simultaneous previews: A river plot interface for time series forecasting. In *Proceedings of IEEE Information Visualization* (2007), pp. 191–196. 2

[Bre05] Brewer C.: *Designing Better Maps: A Guide for GIS Users*. ESRI Press, 2005. 6

[CSP*06] Card S., Sun B., Pendleton B., Heer J., Bodnar J.: Time tree: Exploring time changing hierarchies. In *Proceedings of IEEE VAST* (2006), pp. 3–10. 2

[CTA04] Cook A. J., Tanner G., Anderson S.: *Evaluating the true cost to airlines of one minute of airborne or ground delay: final report*. Tech. rep., Eurocontrol, 2004. 1

[GLG00] Goebbels G., Lalioti V., Goebbel M.: On collaboration in distributed virtual environments. *Journal of Three Dimensional Images 14*, 4 (2000), 42–47. 7

[GST05] Guo Y., Suhl L., Thiel M. P.: A decision support system for airline crew management: Crew scheduling and rescheduling. In *Supply Chain Management und Logistik*. Physica-Verlag HD, 2005, pp. 359–378. 4

[HBE96] Healey C. G., Booth K. S., Enns J. T.: High-speed visual estimation using preattentive processing. *ACM Trans. Comput.-Hum. Interact. 3* (June 1996), 107–135. 5

[Hen98] Henze C.: Feature detection in linked derived spaces. In *Proceedings of IEEE Visualization* (1998), pp. 87–94. 2

[Hös09] Hösel J.: *3D Information Visualization for Disruptions Monitoring in Airline Crew Tracking*. Master's thesis, Hochschule für Technik und Wirtschaft Berlin, 2009. 4

[IAT] IATA coding system. www.iata.org/codes. 28.03.2012. 2

[Jep] Jeppesen Systems AB: Disruption Management. www.jeppesen.com/industry-solutions/aviation/commercial/-carmen-integrated-operations-control.jsp. 09.03.2013. 3

[KH81] Kleiner B., Hartigan J. A.: Representing points in many dimensions by trees and castles. *Journal Am. Stat. Assoc. 76*, 374 (1981), 260–269. 5

[KL83] Kruskal J. B., Landwehr J. M.: Icicle plots: Better displays for hierarchical clustering. *The American Statistician 37*, 2 (1983), 162–168. 5

[KM01] Kosara R., Miksch S.: Metaphors of movement: a visualization and user interface for time-oriented, skeletal plans. *Artificial Intelligence in Medicine 22*, 2 (2001), 111–131. 2

[LTL*10] Luo X., Tian F., Liu W., Teng D., Dai G., Wang H.: Visualizing time-series data in processlines: design and evaluation of a process enterprise application. In *Proceedings of ACM Symp. Applied Computing* (2010), pp. 1165–1172. 2

[Luf] Lufthansa Systems AG: Operations Solutions. www.lhsystems.com/solutions-services/airline-solutions-services/operations-solutions.html. 09.03.2013. 2

[MELS95] Misue K., Eades P., Lai W., Sugiyama K.: Layout adjustment and the mental map. *Journal of Visual Languages & Computing 6*, 2 (1995), 183 – 210. 3, 6

[MHR10] Mazzocchi M., Hansstein F., Ragona M.: The 2010 volcanic ash cloud and its financial impact on the european airline industry. *CESifo Forum 11*, 2 (2010), 92–100. 1

[Mou10] Mouawad J.: As some flights resume, new confusion emerges. *The New York Times* (2010). 1

[MRC91] Mackinlay J. D., Robertson G. G., Card S. K.: The perspective wall: detail and context smoothly integrated. In *Proceedings of CHI* (1991), pp. 173–176. 7

[Now09] Nowak I.: A business perspective to robust airline planning. In *Proceedings of EURO* (2009). 1

[PBB12] Piringer H., Buchetics M., Benedik R.: Alvis: Situation awareness in the surveillance of road tunnels. In *Proceedings of IEEE VAST* (2012), pp. 153–162. 2

[PT94] Powsner S., Tufte E.: Graphical summary of patient status. *The Lancet 344*, 8919 (1994), 386 – 389. 6

[PW06] Plumlee M. D., Ware C.: Zooming versus multiple window interfaces: Cognitive costs of visual comparisons. *ACM Trans. Comput.-Hum. Interact. 13* (2006), 179–209. 5

[RC94] Rao R., Card S. K.: The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proceedings of CHI* (1994), pp. 318–322. 6, 7

[RTB96] Rogowitz B. E., Treinish L. A., Bryson S.: How not to lie with visualization. *Comput. Phys. 10* (1996), 268–273. 6

[SB94] Sarkar M., Brown M. H.: Graphical fisheye views. *Commun. ACM 37* (December 1994), 73–83. 7

[SHS11] Schulz H.-J., Hadlak S., Schumann H.: The design space of implicit hierarchy visualization: A survey. *IEEE Trans. Vis. Comput. Graphics 17* (2011), 393–411. 5

[SJ07] Sears A., Jacko J. A. (Eds.): *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*. Lawrence Erlbaum Assoc Inc, 2007. 3

[SKSH01] Smith M. J., Koubek R. J., Salvendy G., Harris D. (Eds.): *Usability Evaluation and Interface Design: Cognitive Engineering*. Lawrence Erlbaum Assoc Inc, 2001. 7

[SMM12] Sedlmair M., Meyer M., Munzner T.: Design study methodology: Reflections from the trenches and the stacks. *IEEE Trans. Vis. Comput. Graphics 18*, 12 (2012), 2431–2440. 2

[SZ00] Stasko J., Zhang E.: Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Proceedings of IEEE Information Visualization* (2000), pp. 57–65. 5

[TKAM06] Tory M., Kirkpatrick A. E., Atkins M. S., Moller T.: Visualization task performance with 2d, 3d, and combination displays. *IEEE Trans. Vis. Comput. Graphics 12* (2006), 2–13. 4

[Tuf90] Tufte E. R.: *Envisioning Information*. Graphics Press, 1990. 3, 5

[WGGP*11] Wongsuphasawat K., Guerra Gómez J. A., Plaisant C., Wang T. D., Taieb-Maimon M., Shneiderman B.: Lifeflow: visualizing an overview of event sequences. In *Proceedings of CHI* (2011), pp. 1747–1756. 2

[WPS*09] Wang T., Plaisant C., Shneiderman B., Spring N., Roseman D., Marchand G., Mukherjee V., Smith M.: Temporal summaries: Supporting temporal categorical searching, aggregation and comparison. *IEEE Trans. Vis. Comput. Graphics 15*, 6 (2009), 1049–1056. 2

[WRF*11] Waser J., Ribičić H., Fuchs R., Hirsch C., Schindler B., Blöschl G., Gröller M.: Nodes on ropes: A comprehensive data and control flow for steering ensemble simulations. *IEEE Trans. Vis. Comput. Graphics 17*, 12 (2011), 1872–1881. 2

[YWR02] Yang J., Ward M. O., Rundensteiner E. A.: Interring: An interactive tool for visually navigating and manipulating hierarchical structures. In *Proceedings of IEEE Information Visualization* (2002), pp. 77–84. 5