26-5-2011

Visual Analytics Project

The visualization of NS data

Bram, Gerard, Willem

Visual Analytics Project

The visualization of NS data

1. Introduction

This project focuses on the development of a visual analytics system that visualizes large amounts of data from the Nederlandse Spoorwegen, the most prominent passenger railway operating company in the Netherlands. On average, the NS transports around 1,1 million passengers a day, spread over a total amount of 4500 trains (Treinreiziger.nl, 2006). Our goal is to visualize statistical information from a real-time updated NS dataset. The data-set typically contains information about NS train stations, fixed routes, train departures and delays.

The NS data is produced at unprecedented rate and our ability to obtain and store these data is increasing at a faster rate than our ability to analyze it. Because of this issue, visual analytics is needed to extract, interpret and analyze relevant information from the data. But what exactly is visual analytics?

1.1 Background

According to Keim et al. (2008) visual analytics is an iterative process that involves information gathering, data prepossessing, knowledge representation, interaction and decision making. In the process of visual analytics, the user typically interacts with the information to gain insight, to draw conclusions, and to ultimately make better decisions. Keim et al. (2008) explain that the ultimate goals of visual analytics is “to gain insight in the problem at hand which is described by vast amounts of data”. In our case, the problem is the vast amounts of NS data, which is uninterpretable in terms of how the data itself is related. To counter this problem, meaningful visualizations are needed for allowing the user to mentally visualize the data and its underlying concepts. Visualizations allow the user to:

- explore and extract different representations of the data
- extend the capacity of their cognitive system
- find and understand patterns
- make inferences and/or decisions
- monitor vast amounts of data in parallel

Transforming data to visualizations can rapidly be realized due to computational power of today's computers. As Albert Einstein (1879 - 1955) once remarked*: "Computers are incredibly fast, accurate, and stupid. Human beings are incredibly slow, inaccurate, and brilliant. Together they are powerful beyond imagination."* Visual Analytics aims to bring computational power and human intelligence together.

1.2 Scope

Keim et al. (2008) elaborate on the fact the visual analytics is more than only visualization. Visual Analytics can be seen as an integral, multi-disciplinary approach. The scope of Visual Analytics is explained in figure 1.

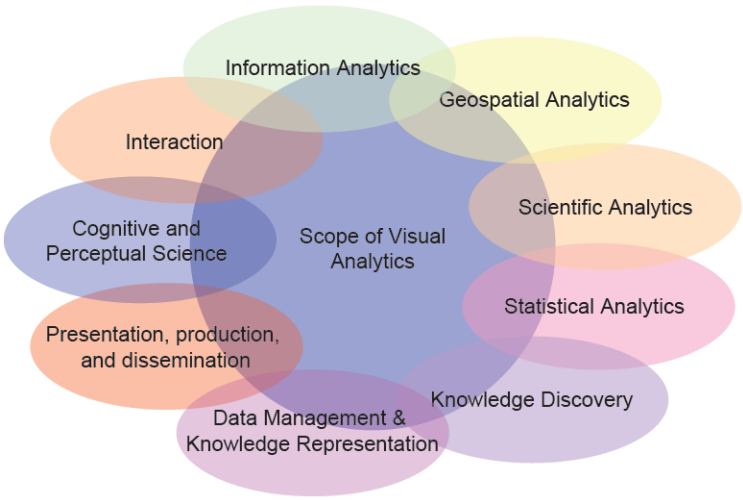

Figure 1: The scope of visual analytics (Keim et al., 2008)

Our aim is to explain the purpose of the NS data visualizations through the 'lens' of this scope. For instance, obtaining, preparing and analyzing the data integrates methodologies from the data management & knowledge representation field while visual components integrate results from the fields of geospatial analytics (e.g. Google Maps). Furthermore, Interaction is also listed in figure 1: we aim to provide proper human computer interaction which is explained in section 5. Visualization and organization of our NS dataset should support the process of statistical analysis, which is also listed in figure 1. Finally, the process of knowledge discovery (figure 1) manifests itself in the development of our system. This means that the project team acquires knowledge by dealing with limitations and constrains throughout the development process. However, knowledge discovery is not limited to the project team. We hope that the user acquires (new) knowledge by interacting with our system while retrieving relevant information.

1.3 Tasks

The user should be able to perform certain tasks with our system. In the start-up phase of this project, the tasks were globally defined by the team and are listed as following:

Main task

- to obtain statistical information from a real-time updated NS dataset containing delays, departures, routes and stations.

Sub-tasks

- to obtain delay information per route, per station, per unit of time.
- to obtain departure information per station.
- to obtain route information.
- to obtain station information.

We believe that the visualization of delay data comprehends the most promising statistical information because delays can be visualized per route, per station and per unit of time. Delay information is valuable because it represents how well the Nationale Spoorwegen is performing. Performance measures are interesting for consumer organizations like Rover (Vereniging Reizigers Openbaar Vervoer). Rover focuses on advocacy of public transit passengers, the overall quality of public transport and the improvement of mobility.

In short, 1) visualizing statistical information from a real-time updated NS dataset is our main goal 2) obtaining statistical information is the main task for the user and 3) delay information the most important statistical information for us to visualize.

2. Dataset

In this section of the report, we briefly describe the data set and its characteristics. Our first step was to retrieve that data in a proper way. The NS data was retrieved with the use of the NS API located at the ns.nl (NS.nl, 2011) and operates the following services: prices, current departures, disruptions and repairs, stations including Geo data and travel advice from station to station.

The list of train stations is updated once a day to make sure that new trains stations are added to the data set. The departure information of each station is updated every 10 minutes. Information on departing trains allows us to retrieve the actual routes of the trains. From these routes, we extracted all

direct connections between the stations. This resulted in 75.000 requests per day, storing 50.000 departures.

For each station its location (longitude, latitude and a country code) and some alternative names for that station are known. For example Schiphol and Amsterdam Airport are the same station and therefore have the same station code (shl). There are some international stations in the data set, mostly in Germany and Belgium, but because these are only destination stations, there is no departure information available and therefore it is not possible to extract the routes and connections automatically. These stations are therefore excluded from the visualizations.

From each station all departures are stored. This information includes the destination of the train, what kind of train it is (sprinter, intercity, etc), from which platform it departs, whether it is diverted and of course, one of the more interesting parts, how much it is delayed.

The amount of delay on a specific route is interpolated/extrapolated with known delays for the route. This is done linearly based on delay and departure information of stations on the route. Usually, there is no delay information for the last two stations on the route. Delay information is computed by selecting the last two stations on the route that do have delay information (extrapolation). It sometimes occurs that delay information from a station within the whole route misses. Delay information is computed by selecting its prior or following station (interpolation).

3. Data Analysis

The techniques used for our data analysis are mainly explained in section 2 of this report. The process of data analysis itself, however, has lead to certain conclusions which had direct impact on further development of our system. By browsing through our data, we discovered that:

- ~500 stations and ~50.000 departures a day should be visualized.
- Some stations occur twice in our data, however with different names indicated as ‘aliases’.
- Some stations are located in foreign countries. These stations need to be filtered because we only want to visualize NS stations located in The Netherlands.
- Some stations are irrelevant to the user (Spoorwegmuseum). These stations need to be filtered as well, because they are ‘non-operating’.
- The network (with stations as nodes and connections as edges) should be visualized based on ‘stoptrein’ and ‘sprinter’ routes. ‘Intercity’ routes typically ignore intermediate stations. ‘Stoptrein’ and ‘sprinter’ routes are the best approximation for connecting stations based on their longitude/latitude information.

4. Components

Our system comprehends various visualizations which are displayed within a dashboard setting in the users internet browser. What exactly is a dashboard representation? According to Stephen Few (2006), "A dashboard is a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance." Our system interface is displayed in figure 3. Note: there is some debate about the definition of 'dashboard'. A 'dashboard' can be seen as a visual overview that offers no interaction. However, our 'dashboard' does offer interaction, it's a control panel that's open to human adjustment of the system.

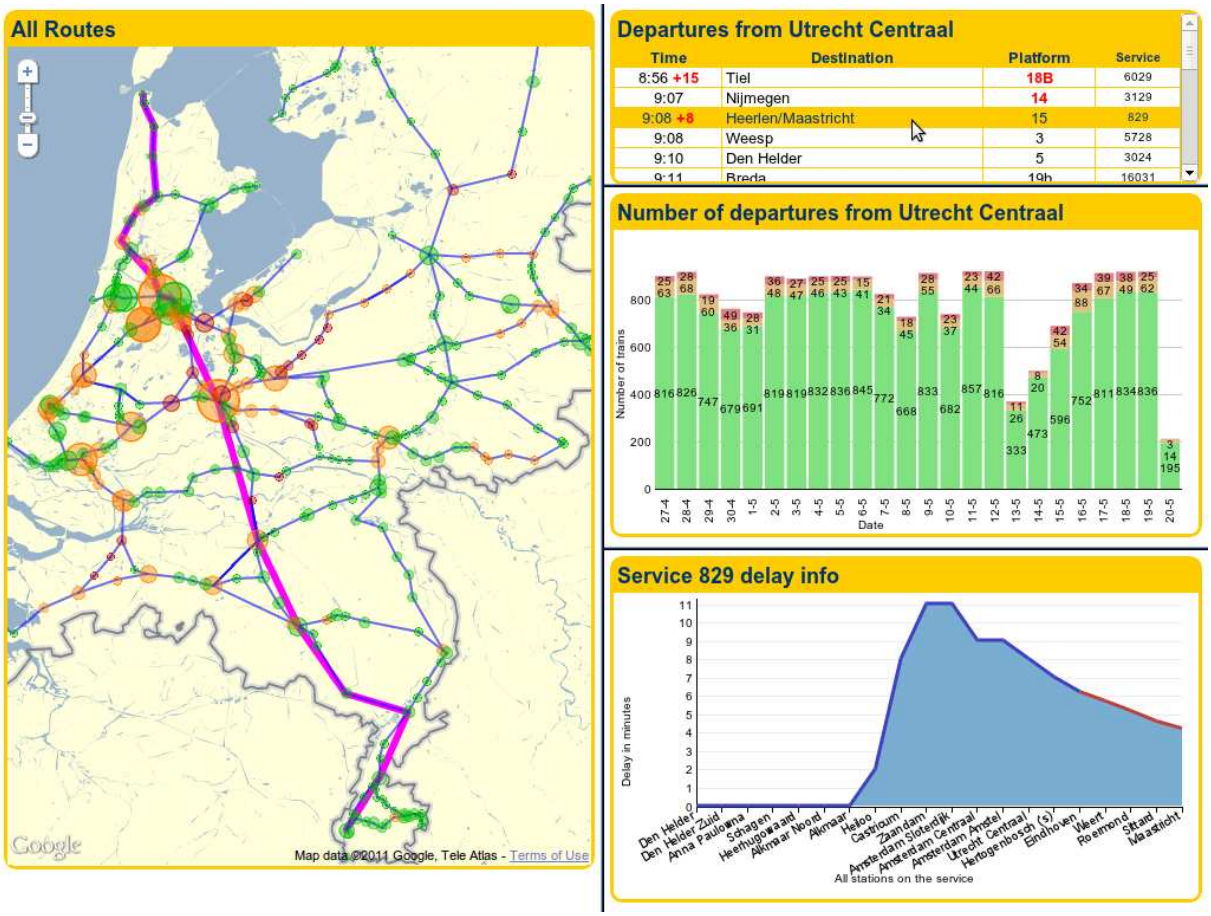

Figure 2: System interface

Component 1: geo visualization & network visualization

In *Information Dashboard Design*, Stephen Few (2006) elaborates on 13 design rules. His first, most important rule is to use one main screen. The main screen displays the most important information and is built out of various data layers. The first layer comprehends a geophysical layer: a Google Map with its focus on The Netherlands. The geophysical layer requires spatial and/or temporal components to display meaningful visualizations to the user. This is done by adding an extra layer: a network layer. The network visualization comprehends nodes (stations) and edges (routes) which are connected to one another; they represent a graph. The retrieval and processing of this graph is explained in section 2.1 (user-tasks).

Stations are rendered in Protovis as dots. Dot size expresses the average number of trains per day, per station. Dot size is relative and has a standard minimum value so that all stations are rendered properly. Dot color expresses the amount of delay per station. Green indicates 0 minutes of delay. Orange indicates >0 and <1 minute delay. Red indicates >= 1 minute delay. Routes are rendered in Protovis as blue lines with equal properties (all lines have the same thickness and color).

Component 2: numeric data visualization & timeline visualization

Numeric data is expressed in two different visualizations.

The **first** visualization comprehends a stacked bar chart, which displays the 1) average number of departures or 2) the number of departures from #station#. In both cases, the Y axis expresses the number of trains and the X axis the date (divided in days). Adding date information and visualizing it in an ordinal way creates another visualization: a timeline visualization. The timeline expresses whether certain events occurred.

What is an event in this case? One working definition of 'event' is that 'events' occur if user-defined conditions, which are expressed with respect to entities of a dataset, come true (Aigner et al., 2008). The main condition we want to visualize is 'delay'. Thus, the timeline expresses whether delays occurred and whether these delays can be interpreted as 'normal' or as 'exceptional'. We allow the user to gain insight in delay information over time. Furthermore, we allow the user to discover patterns and/or trends in the delay information. The user can communicate this information if something interesting has been found.

The width and the height of the bars are automatically adjusted to the window size of the users browser. Bar color expresses the amount of delay. Green indicates 0 minutes of delay. Orange indicates a 1~4 minute delay and red indicates a 5+ minute delay.

The **second** visualization comprehends a line chart, which displays the service delay info, or delay info for a certain route. The Y axis expresses the amount of delay in minutes, the X axis expresses all the stations for the specific route. The blue line corresponds to each stations delay information. It regularly occurs that there is no delay information for the last stations on a route. The estimated value for this delay is plotted with a red line, instead of a blue line. The computation of this red line is explained in section 2 (user task 4).

5. Interaction process

Interaction involves the dialog between the user and the system as the user explores the data set to uncover insights. There are many different ways to interact with a system. According to Yi et al. (2007), interaction can be categorized into 7 different classes. The interaction that we derived from our system can be mapped to these classes.

Select

The -select class- provides users with the ability to mark (a) data item(s) of interest to keep track of it. In our main screen, many data items are presented in one view; the map with a network on top of it. It is difficult for users to follow items of interest. The first select interaction occurs when the user selects a station on the map; departures from that station are displayed. The second select interaction occurs when a current departure is selected. The specific route is then displayed on the map. We made items of interest (stations, route) visually distinctive, so users can easily track them (the purple route line). The user can change his/her selection by selecting different departures or stations.

Explore

Yi et al. (2007) explain that the -explore class- enables the users to examine a different subset of data cases. This is needed because “users can view only a limited number of data items at a time because the scale of the data set, view and/or screen limitations as well as cognitive and perceptual limitations”. When the user loads our system, a map of The Netherlands is displayed. This design decision is based on the fact that we visualize all operating stations in The Netherlands. However, the initial map view displays overlapping stations. The user needs to make an ‘explore interaction’, so that only a part of the dataset is shown (in a more proper way).

Connect

One of the functionalities of the -connect class- is to highlight associations or relationships between data items that are already represented. The goals is to show relations when user selects an item and

identify the same item in a different view. When a station is selected, the number of departures from that station + departures from that station is displayed in another view. In other words, the selected item is identified in the main view, but also in the other views that display extra information about the selected item. The highlighting of relationships between data items is also done by the select class. When a departure is selected, the specific route is highlighted, a connection between several data items is displayed to the user.

Reconfigure

Reconfiguring allows the user to create different perspectives on the data by changing the spatial arrangements of representations. Hidden characteristics can be revealed by reconfiguring, one perspective is often not sufficient. For example, by selecting a station with the –select class-, data items (bars on the bar graph) are rearranged and differently aligned. Reconfiguring could wisely be used in a bar graph; to separate delays from non-delays.

Encode

The -encode class- allows the user to change the type of representation to reveal new insights. According to Yi et al. (2007), encoding can affect 'pre-attentive cognition'. This directly relates to how users understand relationships and distributions of the data items. In our visualization, the amount of trains that run through a station is encoded to a map, using different dot sizes for stations. Different dot colors are encoded to the map as well (expressing the amount of delay per station). The most important thing here is: the encodings do not alter the spatial arrangement of the map.

Abstract/elaborate

The -abstract/elaborate- class allows the user to adjust the level of abstraction of a data representation. An example of the class is geometric zooming. The Google Map allows the user to zoom in so that all stations are displayed without overlap. "The key point is that the representation is not fundamentally altered during zooming". (Yi et al., 2007)

Filter

The -filter class- enables the user to change the set of items being represented based on some specific conditions. For example, filtering occurs when the user selects a specific station to retrieve departures: only departures for the next couple of hours are displayed. Departures from the past are not relevant to the user, thus filtered.

6. Visual thinking design & Aesthetic

Visual thinking is a theory that assumes that people can think in images, or pictures. The concept of visual thinking often lacks in sound scientific justification, however, a model for visual thinking is proposed by Colin Ware (2004) in his book *Perception for Design*. One topic that Ware (2004) elaborates on is the human perception of patterns, which is related to the laws of pattern perception developed by Westheimer, Koffka and Kohler in 1912, also well-known as the gestalt principles. How are the gestalt laws incorporated in our system?

First of all, visual grouping by similarity is used. In our main view, stations are grouped as dots, routes are grouped as lines in one integral dimension. Associations and/or relations between stations and routes imply the gestalt law of connectedness. We also think the principle of continuity is represented; routes, as visual entities tend to be smooth and continuous. The gestalt principle of closure states that closed contours form regions. The 'closed contours' in our dashboard interface are the different 'regions', or panels displayed as yellow boxes.

But how does the user visually search through our system? As stated in section 1, delay information is the most important statistical information for us to visualize and present to the user. We want our user to search for delay information as quick as possible. Therefore, we use the laws of pre attentive display. Delay information must stand out on some simple dimension: color. Color is consistently used to display delays. We think that most users already have the idea that 'green is good' and 'red is bad', or that 'green shows that everything is ok' and 'red shows that there is something wrong'. Therefore, red, orange, green were used to indicate delays.

7. Discussion

Our goal was to visualize statistical information from a real-time updated NS dataset. Our user typically acquires statistical information about delays, departures, routes and stations. We considered delay information the most important information to visualize because it represents how well the Nationale Spoorwegen is performing. We visualized delay information per station, with actual delay information per service and/or route. Delay information is also visualized per station, per day and per route, per station.

The main insight we draw from the dashboard is that there is always some delay in the Netherlands. The map gives an immediate overview on where current delays occur. The insight is supported by the bar graph which shows a daily summary of the delays (nationwide or a specific station). This graph fosters us to conclude that the amount of no-delay, short-delay or long-delay is almost the same over each day.

Another insight we draw form the dashboard is that train-drivers are able to shorten a delay. This idea is visualized in the area graph. In the Netherlands, it is common knowledge that train-drivers are allowed to drive faster on certain tracks in order to shorten a delay. Both insights are visualized in figure 3; the amount of daily delays is most likely the same throughout weekdays. Delay of the selected route is shortened throughout the route.

We think our system is suitable enough to be presented to consumer organizations like Rover (as they are occupied with the performance of the Nationale Spoorwegen). As stated in section 4, we allow the user to discover patterns and/or trends in the delay information. This could however be improved by adding extra visualizations of how delays ‘behave’ over a longer period of time. Delay information could also be presented with respect to various regions of the Netherlands: where are the bottlenecks, where do delays occur the most, at which stations, over what periods of time? These questions can be answered by creating a more sophisticated system that gives more advanced insights in statistical delay information.

References

Aigner, W., Miksch, S., Mu̇ller, W., Schumann, H., and Tominski, C. (2008). Visual methods for analyzing Time-Oriented data. In *IEEE Trans. On Visualization and Computer Graphics*.

Few, S. (2006). *Information Dashboard Design: The Effective Visual Communication of Data*. O'Reilly, Sebastopol, 1 edition.

Keim, D., Mansmann, F., Schneidewind, J., Thomas, J., and Ziegler, H. (2008). Visual analytics: Scope and challenges. In *Simoff, S., Bȯhlen, M., and Mazeika, A., editors, Visual Data Mining, volume 4404 of Lecture*. Notes in *Computer Science,* chapter 6, pages 76–90. Springer Berlin / Heidelberg, Berlin, Heidelberg.

Ns.nl (2011) De NS Api.
Retrieved from treinreiziger.nl May 16th, 2011. http://www.ns.nl/reizigers/blind/_api-nieuw

Treinreiziger.nl (2006) *Eerste week nieuwe dienstregeling NS*. Published December 19th 2006.
Retrieved from treinreiziger.nl May 16th, 2011. http://bit.ly/kMER4Y

Yi, J. S., Kang, Y. A., Stasko, J., and Jacko, J. (2007). Toward a deeper understanding of the role of interaction in information visualization. In *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231.

Ware, C. (2004). *Information Visualization: Perception for Design (Interactive Technologies)*, Kaufmann 2nd ed.