INVISIBLE CITIES

Understanding Urban Transportation Networks

An interactive exploration of a Dynamical System

CS 7450 - Data Visualization

October 18th, 2001

Martin C. Oriol
Yi Feng
Jose Sarmiento

Contents

Project Objective & Scope

"The city, however, does not tell its past, but contains it like the lines of a hand, written in the corners of the streets..."
Italo Calvino, Invisible Cities

Italo Calvino book *Invisible Cities* explores the idea that cities are much more than their physical description but rather it's identity is found on it's connections and how they are formed. Nevertheless, understanding these dynamic interactions is a challenging task. Cities are made of myriad of attributes, their rules are absurd, their perspectives deceitful, and everything conceals something else. Even with a more instrumented world were we can now observe and store a vast number of records extracting real intelligence and knowledge remains difficult.

Our project will focus on creating a data visualization application that will visualize the transportation network within a city as we believe the transportation system of a city will be able to help us understand its dynamism and its "hidden" stories. We will concentrate our efforts to create data visualizations that can demonstrate movement of physical entities through time which will illustrate the emerging patterns of the network and the interaction between the different components of the system. For the sake of simplicity, this document will principally focus on how our visualization application works with the city of Chicago but will use other cities as well.

We intend that with our dynamic visualization it will increase the understanding of the story of a city, how it emerges, how it evolves, where its deficiencies are located and will thus help in its comprehension, improvement and maintenance.

Project Description

The main visualization will consist on a topology map of the different stations (subway or bus stations) and the connections between them. However, the different shape, size and color of the different stations and connections will be encoding different attributes of the data, providing an effective way of visualizing several variables and their correlations at the same time. Then, mechanisms will be added in order to explore further details of the data. With all, the application will tell a story about the transportation network and will let the user explore that story.

Some example of the data attributes that will be shown are the following: (further details are given in the Data section)

- **frequencies** of trips between the different stations.
- **amount of passengers** on the different trips.
- **delays** that each station is suffering.
- **travel times** between the different stations of a line.
- **Fares** that consumers have to pay to travel

The application, additionally, will consist on another primary visualization that will show how far a user of the public transportation system can go within a specific amount of time (taking into account the delays) and money starting from a given station. This visualization can help in gaining insight about how the different lines and routes have evolved over time and, also, can help a user comprehend his interaction with public transportation.

The application is intended to be used both by users and managers of the public transportation system.

A user or the public transportation system could use the application to decide upon the best route to go from one point to another or upon the best time to travel. For instance, one could decide the best time to travel and the best route to use to get to work by exploring how the state of the network (delays or number of passengers) usually is in the hours before one has to get to work.

The application would be beneficial also to managers by providing a way of exploring where the network is performing well and where it has some deficiencies, and how this has evolved over time. It could be seen, for instance, that there is some section of the network in which the frequencies are low but the amount of passengers is very high, meaning that the frequencies should be increased.

Related work

Visualization creators have widely explored the area of transportation. However, most of the current visualizations lack interaction mechanisms or do not show enough data.

In *tracking taxi flow across the city*[1] (see fig. 1) and *NYC Subway ridership*[2] (see fig. 2), the frequency of taxi trips and ridership of subway trips is shown using a coloring scheme, and the evolution of this frequency can be explored. While offering an effective way of visualizing a transportation network, this applications shows a little amount of variables, lacking details about the performance of the network.

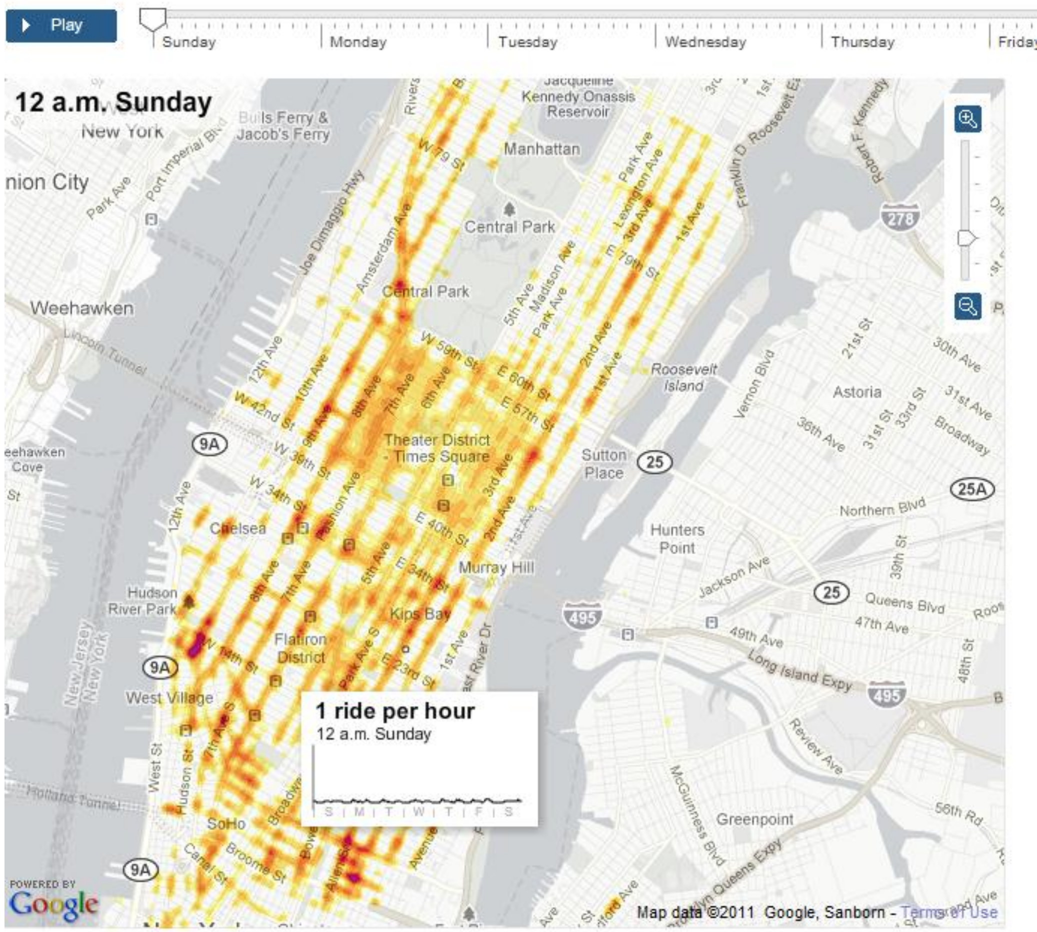


Fig. 1: Tracking Taxi Flow across the City main visualization screenshot

[1] http://www.nytimes.com/interactive/2010/04/02/nyregion/taxi-map.html

[2] http://www.diametunim.com/shashi/nyc_subways/

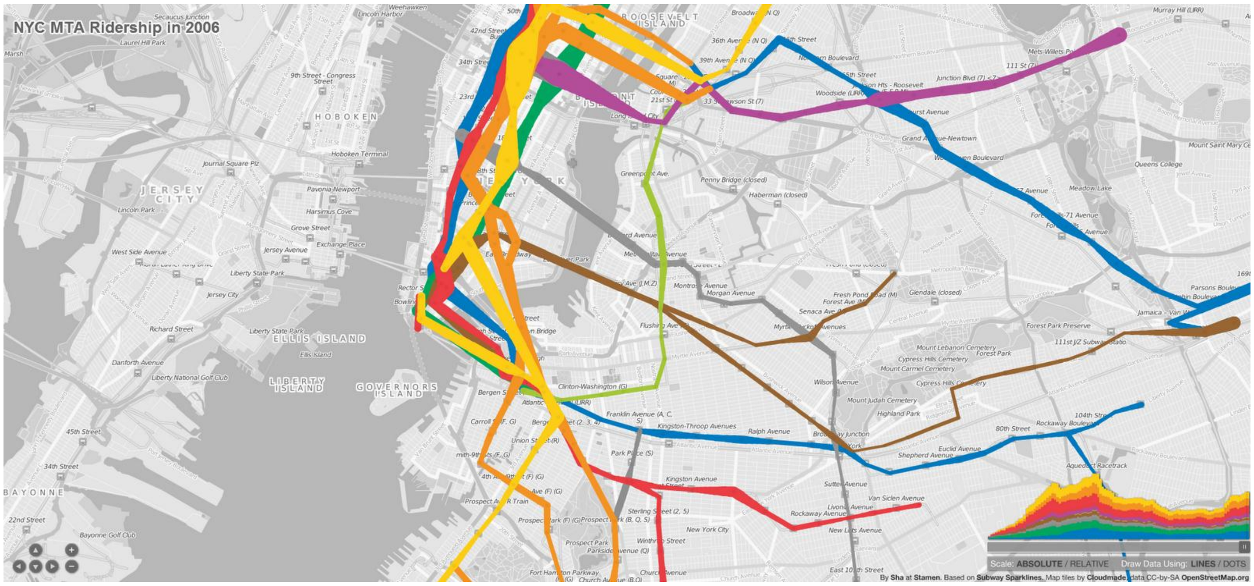


Fig. 2: NYC Ridership main visualization screenshot

Other applications show other kind of variables in a good way, but again, they lack the capability of showing more variables or the correlation between variables. *Non-geographic mapping*[3] (see fig. 3) show travel distances between different points in the world in a pleasant and funny way, *CTA train tracker predictor*[4] (see fig. 4) show data of the how currently trains in the city of Chicago are performing, and *Magnificent Chicago*[5] (see fig. 5) show where you can reach within an interval of time using public transport from the city of Chicago.

Graphserver[6] and *Animating Auckland's Public Transport*[7] (see fig. 6), static visualizations (with no interaction from the user), show the evolution of a transportation network in a way that lets the user understand the story of the data underlying that network. However, without interacting mechanisms the user cannot fully explore the data or search for answers to questions that he may find interesting.

[3] http://www.number27.org/assets/work/extras/maps/traveltime/index.html
[4] http://www.visualizing.org/full-screen/29821
[5] http://www.mapnificent.net/chicago
[6] http://graphserver.sourceforge.net/gallery.html
[7] http://infosthetics.com/archives/2011/08/animating_aucklands_public_transport.html

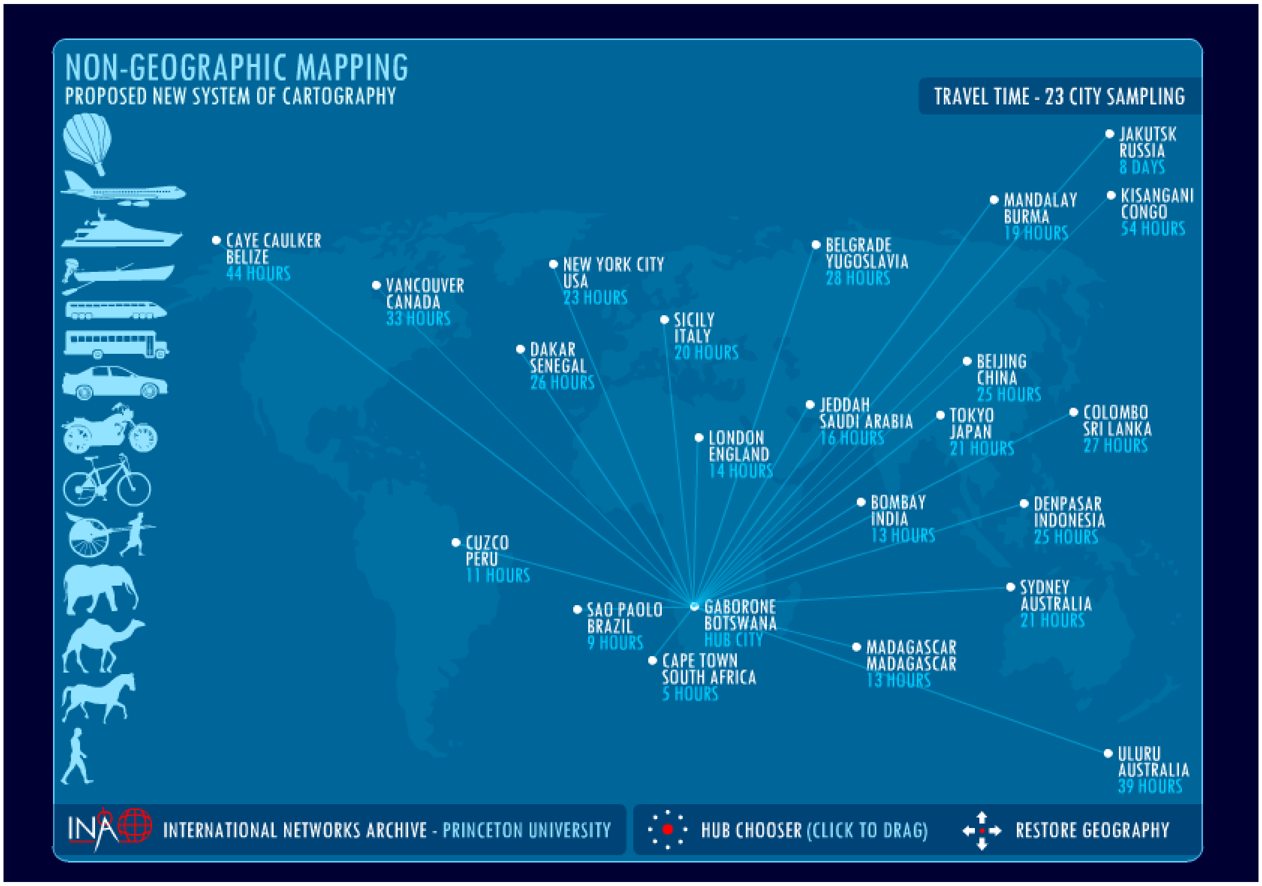


Fig. 3: Non-geographic mapping visualization screenshot showing the distance from a city to other cities.

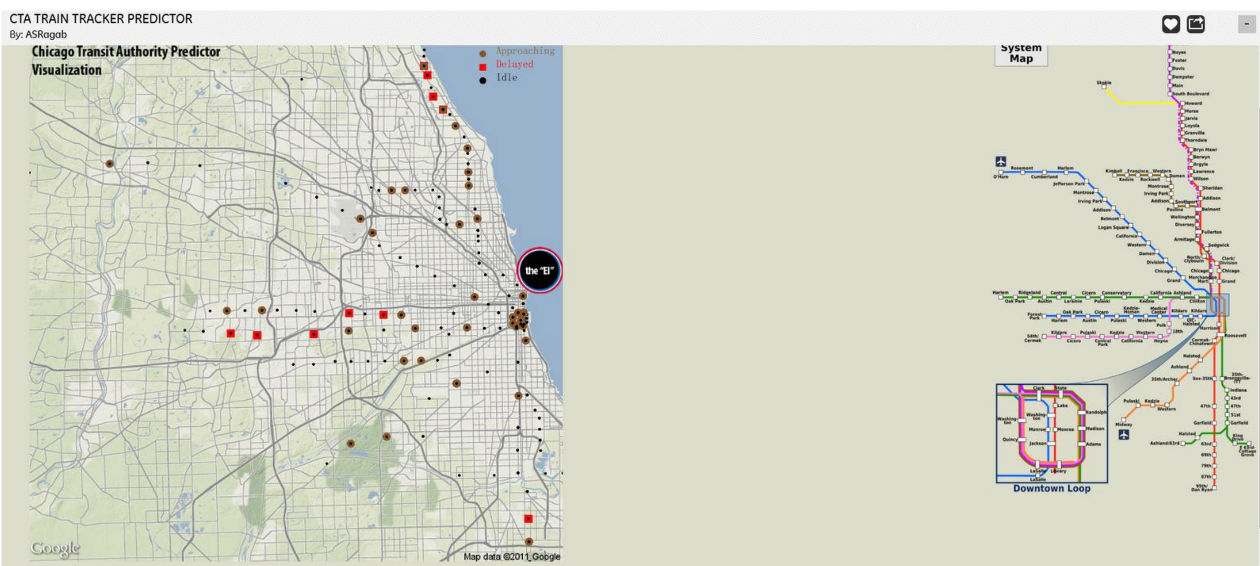


Fig. 4: CTA Train Tracker Predictor application screenshot

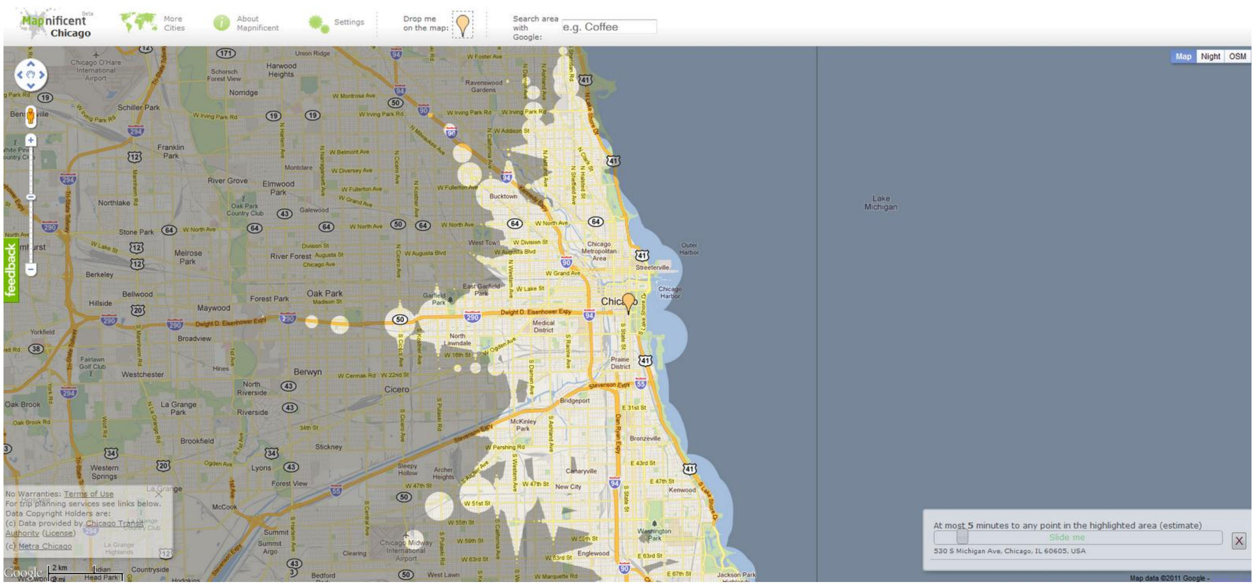


Fig. 5: Magnificent screenshot showing the points you can reach within a given time

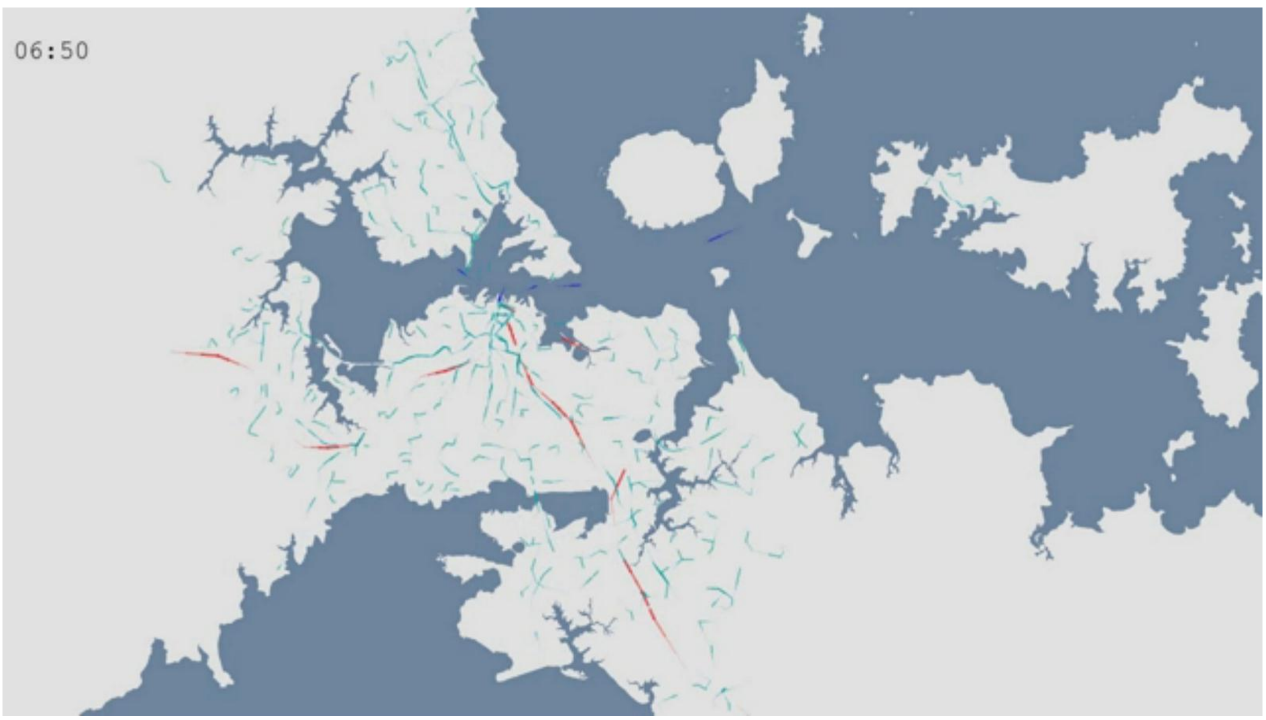


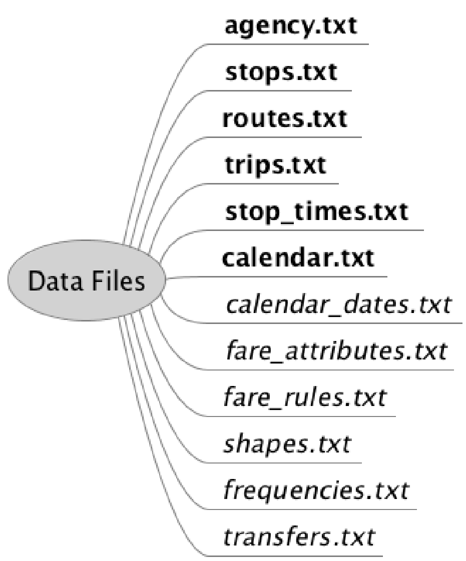
Fig. 6: Animating Auckland's Public Transport screenshot taken from the video

Data

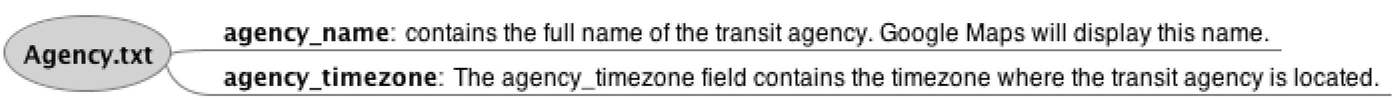
Our data is provided by *Chicago Transportation Authority* (CTA) and is compliant to the *General Transit Feed Specification* (GTFS). The GTFS defines a common format for public transportation schedules and associated geographic information.

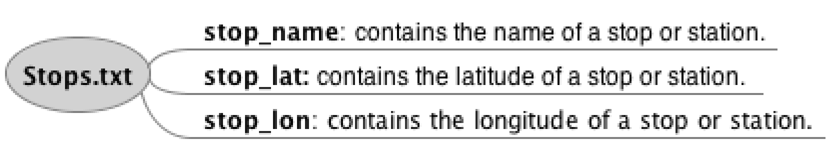
GTFS Data

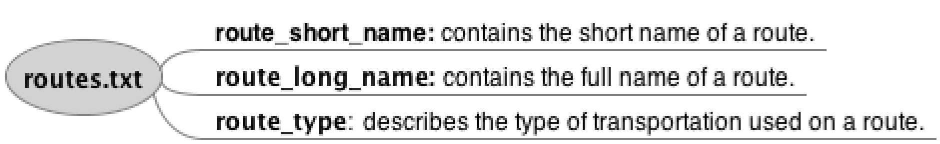
Here we show all files defining GTFS data. The bold ones are required, and italic ones are optional.



The following explains each field in each file:







- **trips.txt**
  - **service_id:** contains an ID that uniquely identifies a route.
  - **trip_headsign**: contains the text that appears on a sign that identifies the trip's destination to passengers.

- **stop_times.txt**
  - **arrival_time:** specifies the arrival time at a specific stop for a specific trip on a route.
  - **departure_time**: specifies the departure time from a specific stop for a specific trip on a route.
  - **stop_sequence**: identifies the order of the stops for a particular trip.

- **calendar.txt**
  - **monday**: contains a binary value that indicates whether the service is valid for all Mondays.
  - **tuesday**: contains a binary value that indicates whether the service is valid for all Tuesdays.
  - **wednesday**: contains a binary value that indicates whether the service is valid for all Wednesdays.
  - **thursday**: contains a binary value that indicates whether the service is valid for all Thursdays.
  - **friday**: contains a binary value that indicates whether the service is valid for all Fridays.
  - **saturday**: contains a binary value that indicates whether the service is valid for all Saturdays.
  - **sunday**: contains a binary value that indicates whether the service is valid for all Sundays.
  - **start_date**: contains the start date for the service.
  - **end_date**: contains the end date for the service.

- ***fare_attributes.txt***
  - **price**: contains the fare price, in the unit specified by **currency_type.**
  - **currency_type**: defines the currency used to pay the fare.
  - **payment_method**: indicates when the fare must be paid.
    - **0** - Fare is paid on board.
    - **1** - Fare must be paid before boarding.
  - **transfer**: specifies the number of transfers permitted on this fare.

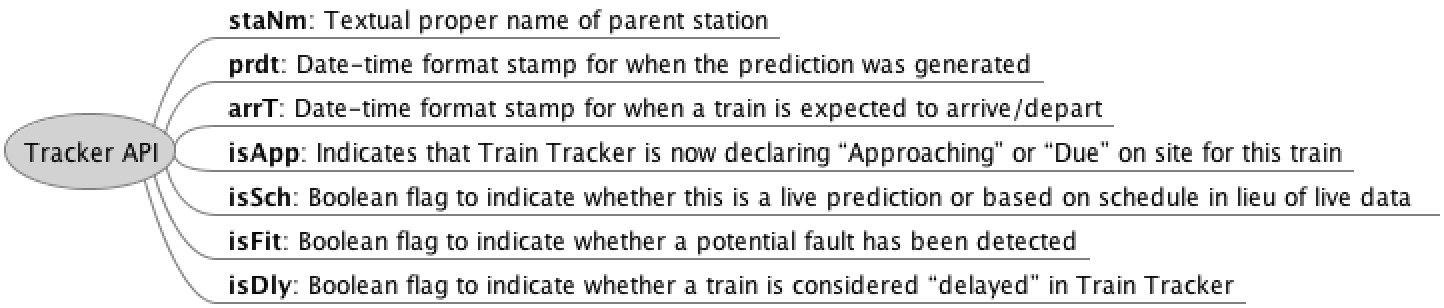
- **shapes.txt**
  - **shape_pt_lat**: associates a shape point's latitude.
  - **shape_pt_lon**: associates a shape point's longitude
  - **shape_pt_sequence**: associates the latitude and longitude of a shape point with its sequence order along the shape.

- ***frequencies.txt***
  - **start_time**: specifies the time at which service begins with the specified frequency.
  - **end_time**: indicates the time at which service changes to a different frequency (or ceases) at the first stop in the trip.
  - **headway_secs:** indicates the time between departures from the same stop (headway) for this trip type, during the time interval specified by .**start_time** and **end_time.**

- ***transfer.txt***
  - **from_stop_id**: contains a stop ID that identifies a stop or station where a connection between routes begins.
  - **to_stop_id**: contains a stop ID that identifies a stop or station where a connection between routes ends.
  - **transfer_type**: specifies the type of connection for the specified (from_stop_id, to_stop_id) pair.

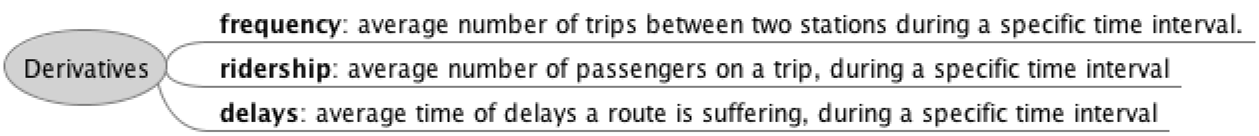
CTA Train Tracker API

This set of APIs provided by CTA produces a list of arrival predictions for all platforms at a given train station in a well-formed XML document. Prediction data is combined with other data and polished to help present information in the most meaningful way possible.



Derivative Data

Derivative data is calculated using the existing data from the files described above. We use it to provide other visualizations which can help users fetch more information.



Other Data

We are going to obtain some more data from the *City of Chicago Data portal*[8]. Specifically we will be obtaining ridership data from this site.

[8] http://data.cityofchicago.org/Technology/CTA-Views/gzmt-5k8a

Development Tools

Language

Processing[9]

Processing is an open source programming language and environment for people who want to create images, animations, and interactions.

Map

Unfolding[10]

Unifolding is a tile-based map library. Map tiles can have various geographic features, and come in all kind of styles.

OpenStreetMap[11]

OpenStreetMap is a free editable map of the whole world.

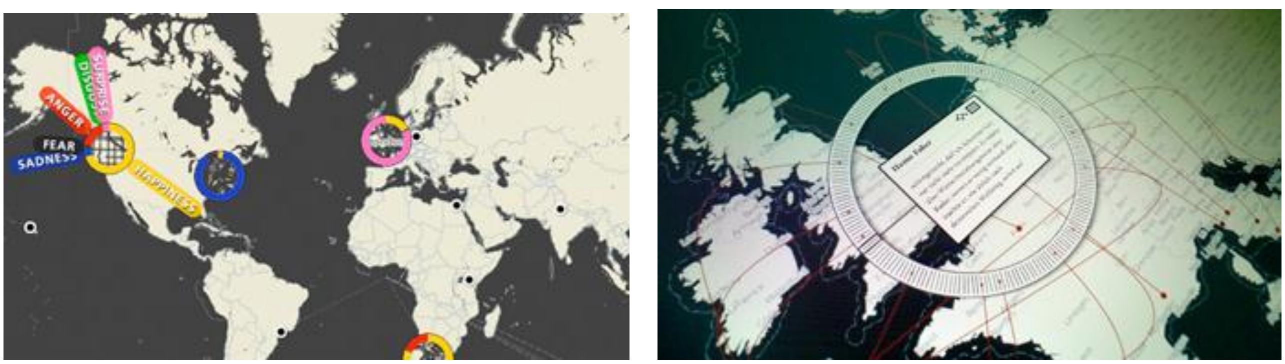


Fig. 7: Examples of Unfolding and OpenStreetMap

Graphics

Gephi[12]

Gephi is an interactive visualization and exploration platform for all kinds of networks and complex systems, dynamic and hierarchical graphs.

[9] http://processing.org/
[10] http://unfoldingmaps.org/
[11] http://www.openstreetmap.org/
[12] http://gephi.org/

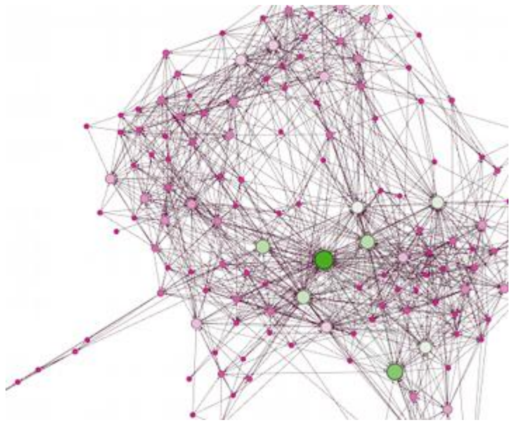


Fig. 8: Example of Gephi

JUNG[13]

JUNG is a software library that provides a common and extensible language for the modeling, analysis, and visualization of data that can be represented as a graph or network.

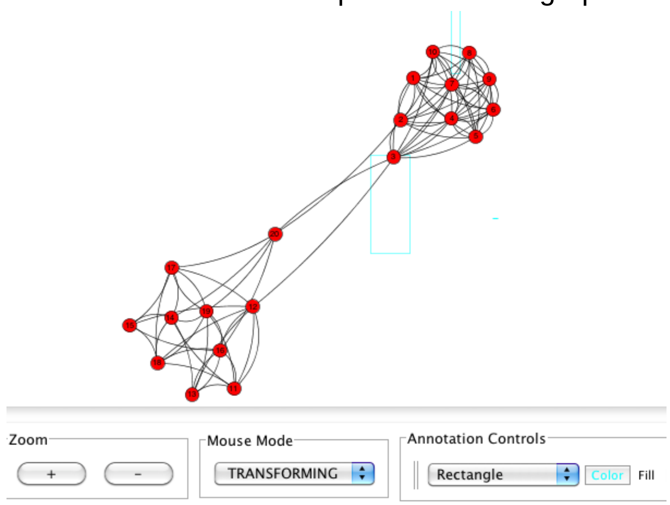


Fig. 9: Examples of JUNG

GLGraphics[14]

GLGraphics is a library intended to extend the capabilities of the OPENGL renderer in Processing.

Ani[15]

Ani is a lightweight and easy to use library for the programming environment Processing. Ani helps to create any kind of animations and transitions.

[13] http://jung.sourceforge.net/

[14] http://glgraphics.sourceforge.net/

[15] http://www.looksgood.de/log/2010/06/08/ani/

Sketches and Interaction

In this section we will show and describe sketches representing the screens and visualizations that will compose the application. Our intention is for the application to unravel the story of the city as its being explored.

As a whole we will keep a minimalistic design throughout the application in order to promote readability of data and increase understanding in an effort to use the principle of the data-ink ratio.

Our color pallet selection for the main components of the application are the following:

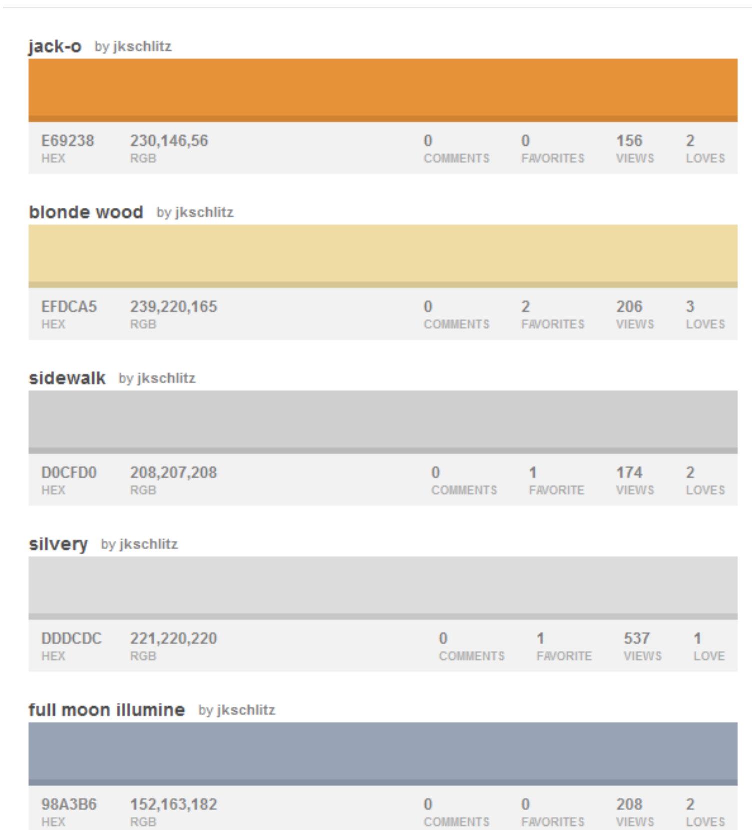


Fig. 10: Color pallet

The objective is for these color to encourage pre-attentive process for the user to easily understand boundaries of different visualizations and highlight important elements and tools for his disposal. The color selected are a good transition between contrasting color and analogous colors which will promote legibility according to Maureen Stone article about color selection. Color will be a main component discussed in our next visualizations and how we made great efforts to keep a modest use out of them because as Tufte said “avoiding catastrophe is the first principle when bringing color to information.”

Introduction Screen

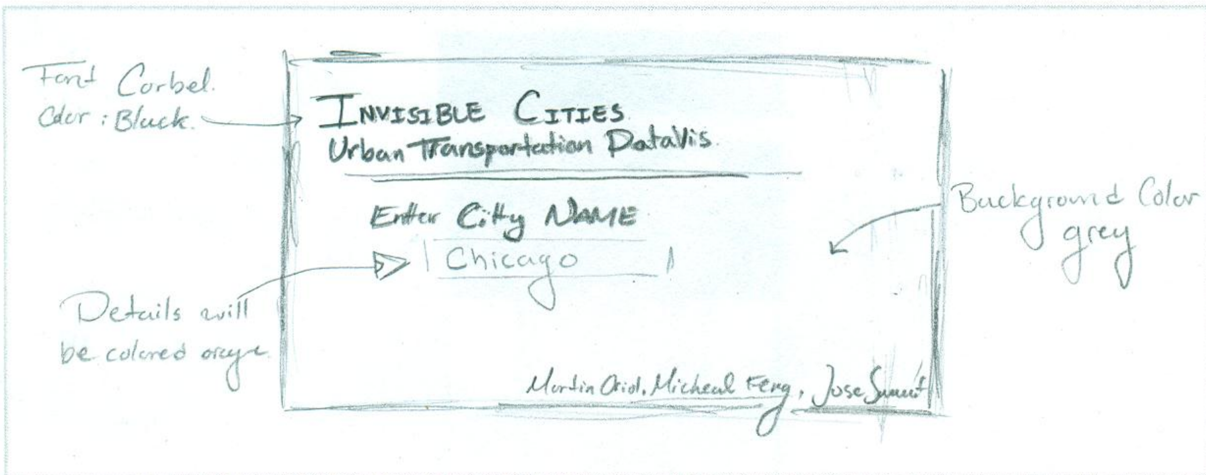


Fig. 11: Introduction Screen

This is just an introduction screen were the user will input in the text box the city it wants to explore. The design is minimalistic and intended to target the visualization to tell the story of one city.

Main Application Overview

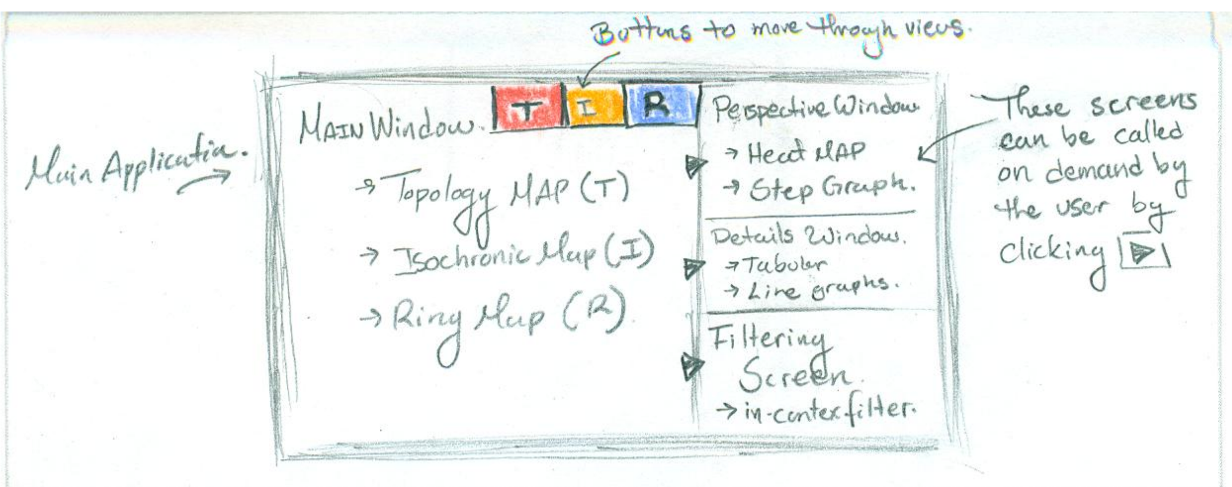


Fig. 12: Application Overivew

Above is a high-level view of our main screen to visualize the dynamism of transportation data. There are two core concepts of why our visualization is effective. One it's a "Ubiquitous Visualization", meaning that you can see "everything" - in Tufte's words "this shows the data" - every variable and every interaction among those variables can be seen either in the main window and further complemented in the support windows. The second concept is "In-

Context Manipulation", were variables and records can be filtered and manipulated in context and all visualization are updated simultaneously giving the user a strong intuition of how variables interact with one another and help him explore. Our system is moving away from the old system of tweaking numbers in a different out of context screen and later calling the plot command to a more natural interaction that helps data exploration.

In essence Invisible Cities aims create maps and diagrams which will not be a passive representation data, but rather a dynamic tool for interpretation, comprehension and possibly action. It will build a time-based narrative of the history of the public transportation movements of a city which will reveal some hidden patterns, or emergence relations which could be useful for the user to comprehend the the city is "really about".

Below we describe in detail the composition of each screen, how the user interacts through it, and what value does he gains.

The main application contains the following components:

- **Filtering Window**: This window will be shown in the bottom right corner and it will allow the user to select and filter the areas of the map he wants to visualize. The main window, then, will be updated according to the selections made.
- **Main Window:** This will contain the main visualizations, which will vary between three different visualizations according to the user's needs.
    - Topographical Map
    - Isochronic Map
    - Ring Chart
- **Details of Data Window**: This window will show the raw data in a classical tabular form. It will contain the exact values for the data cases that are being shown in the different visualizations.
- **Detailed Heat Maps & Step Graph Window:** These windows show several visualizations aimed at representing the evolution over time of the key variables.

Main Window

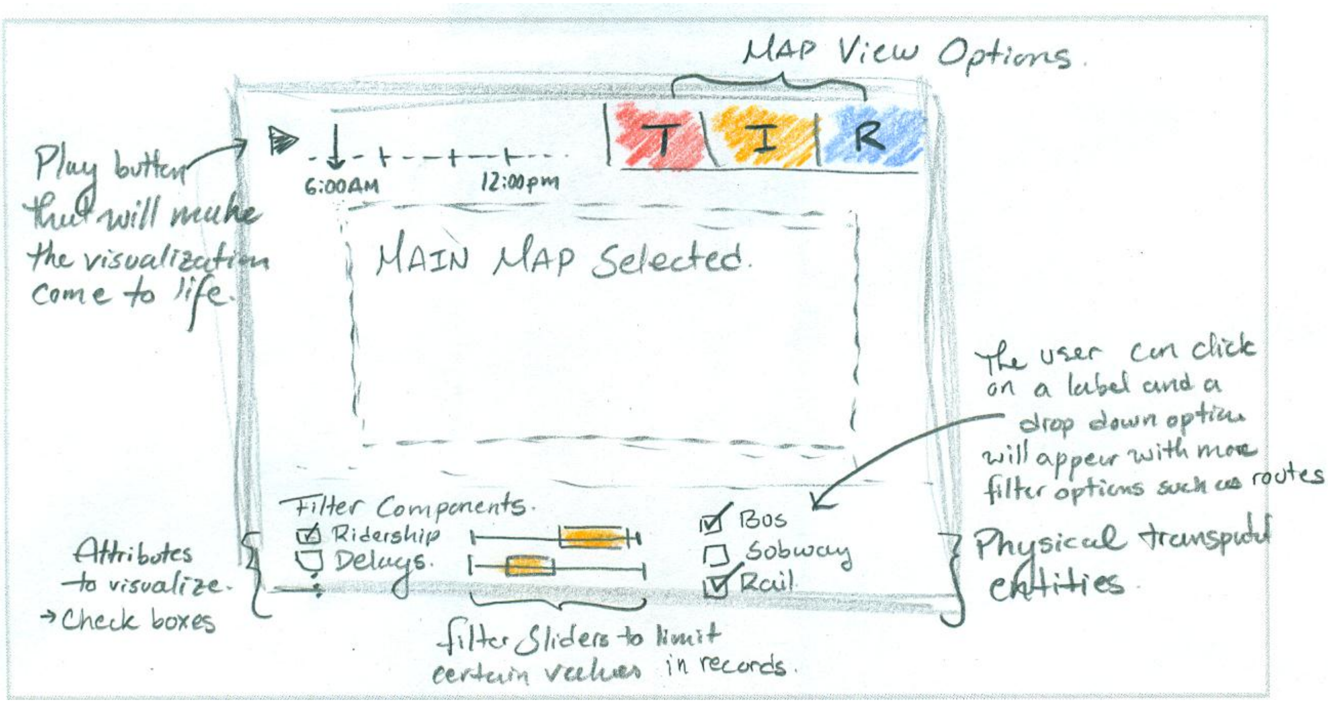


Fig. 13: Main Window

Description

Through this window the user will be able to explore the dynamics of a city transportation network and comprehend its nature. The main window will allow the user to explore the city in three different ways. It's intended that through this view the user will get a heuristic understanding of the cities movements and patterns and will later use the supportive screens to gain a more accurate and quantifiable understanding of the transportation dynamics.

1. **Topographical Map**: It will show the emergence of the city over time. The objective is to let the user explore how the city "wakes up" and goes "back to rest". It's intended that through this view the user will be able to intersect the transportation network and the geographical reality of the city.
2. **Isochronic Map**: This view will focus on a station and it will show a distorted map (with respect to the topographical map) in which stations will be arranged according to the travel time from the selected station. Stations will be connected according to the different paths that a traveler can take in order to get to each station from the user selected station.
3. **Ring**: It will show more information about each line in a ring. The objective is to let the user easily have access to detailed information of each line and the relationships between lines.

There will be the option to show the three types of visualizations simultaneously if the user desires. Nevertheless, in a effort to reduce clutter the visualization will be displayed at the lowest level of detail. Additionally, once the topographical map is opened the play button will automatically start unless otherwise stated by the user.

Topographical Map

Objective

The objective in this section is to help the user how variables in a transportation network such as bus frequencies, total ridership, and delays for example can be observed in a geographical representation of the city.

Fig. 14: Urban Murmors Map Design. Our intention is the create the same effect these designers have accomplished of showing dynamically how “rumors” and why certain patterns emerge through a city.

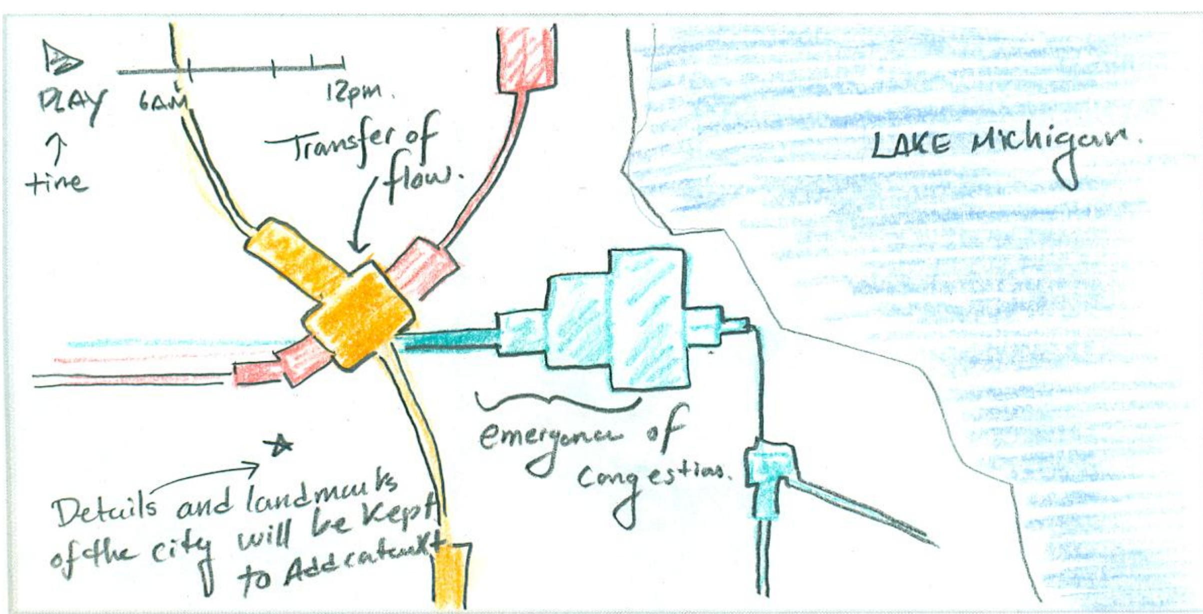


Fig. 15: Sketch of a segment of the topographical map for the city of Chicago.

The first view of the main window primarily contains a topology map showing the different subway and bus lines stops, with their variables encoded in different attributes of the shown lines and stops. The variables will be encoded in the attributes in the following way.

- Lines and stations **color hue** will encode the color of the **line** (Red Line, Blue Line, etc.). In case more than one line goes through a given station, the area of the station will be colored with multiple hues.
- Lines and stations **width** will encode the average **frequency** of trips during the selected time interval. Elements with higher frequency will be wider.
- Lines and stations **heat-based color contrast** will encode the average **ridership** (amount of passengers) on trips over each station or line during the selected time interval. Elements with higher ridership will have darker colors (more heat).
- Trailing dots will depict the **movement** of a transportation medium between stations and it rate of change will encode the average **delay** time on each station. The less delayed the path the will have faster the movement of the dots .

The color selection for the subway lines will be using the one of the pallets from color brewer. We placed focus on choosing canonical colors so the user could easily identify different stations and that contrast well with black so that encoded information could be easily read.

All of these elements will evolve and change over time providing the sensation to the user of a city that it growing and dynamically moving through the day. Since the size, color intensity, and patterns from by the lines will change. We are hoping that through these physical elements we can increase the user sensation of comprehending the transportation urban dynamics. In addition to show the transition through the day the colors in the visualization will turn darker during "night time" and become brighter during the "day".

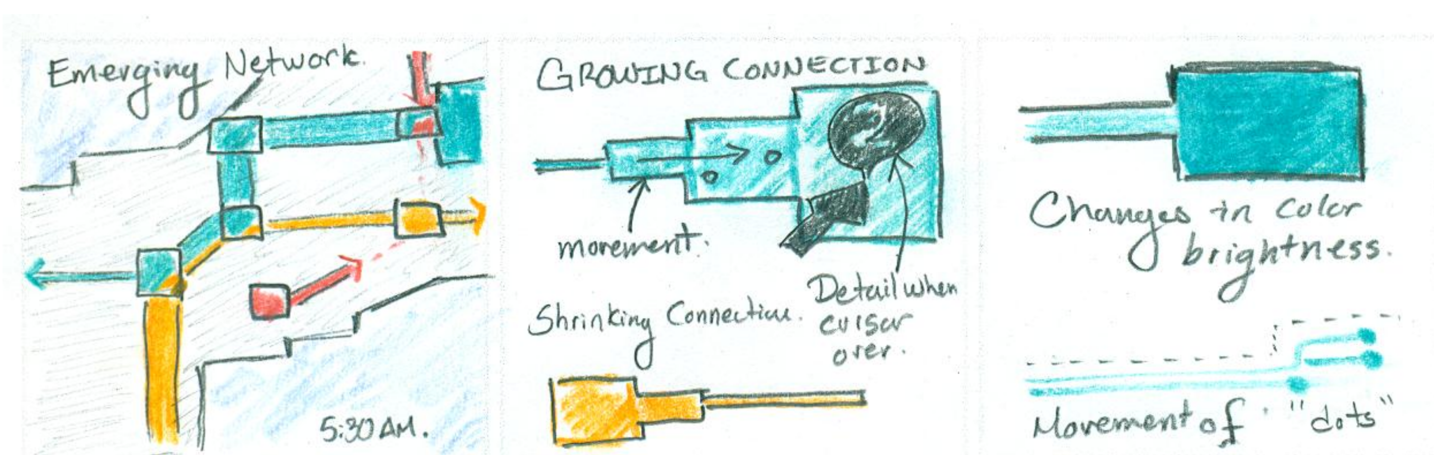


Fig. 16: Details on topographical map

In the end we are hoping that this technique will have an end result similar to the pictures shown in fig 17.

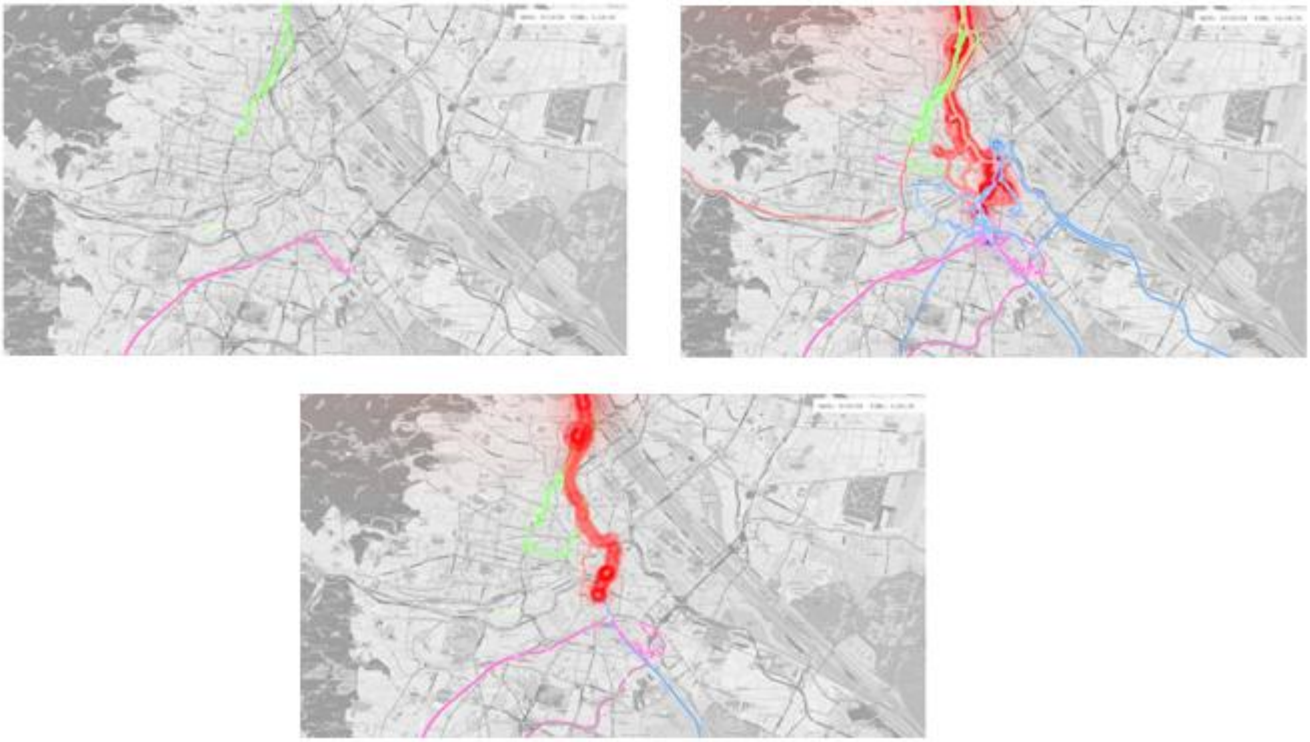


Fig. 17: Topological Map of the city being explored.Source: http://casualdata.com/senseofpatterns/

Interaction

The user will be able to interact with the different components in several ways:

- "double click" on a station: will change the visualization to the Isochronic map visualization.
- select a subset of the stations and/or routes: will highlight the shown data cases. This will provoke the highlighting of the same data cases on all the components that were showing these data cases.
- pass the mouse over a station or route: will show a tooltip with detailed data values about the selected element (name, exact average volume, exact average ridership, etc.). This is an example of the "pointing technique" by the ProtoVis team
- change values with *Filtering Sliders*: that will filter the data cases or the variables. This will provoke the update of all the components that were showing these data cases or variables and reduce the focus on the other excluded data cases. Here we are employing the "Context+Focus" interaction which "allows the viewer to inspect an interesting portion of the data in detail (the focus) without losing global context — the

global view is preserved at reduced detail, highlighting the focused region". All updates occur dynamically and simultaneously.

- <u>change values in the time slider</u>: will change the time interval, provoking the filter of the data cases to those involved in the new interval, such as the trips made anytime during the interval.

Additionally, the map will have facilities for zooming and panning it.

Isochronic map

Objective

Our understanding of a city transportation system isn't necessarily constricted to its physical structure but rather how other "system" factors affect it. The purpose of this window is to visualize the "mental" maps we personally create to understand a transportation network outside the static geography.

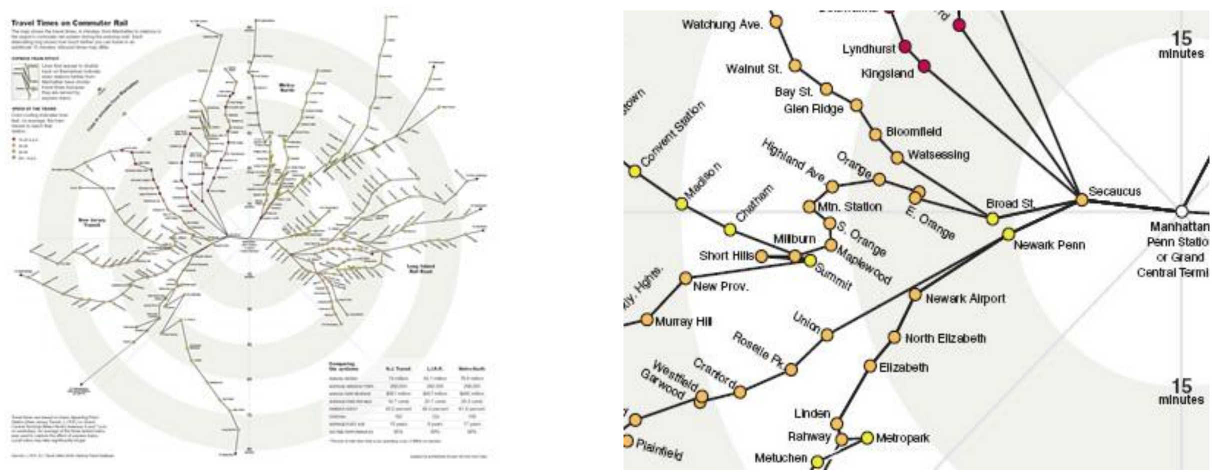


Fig. 18: Isochronic maps demonstrating road distortion to describe travel distances

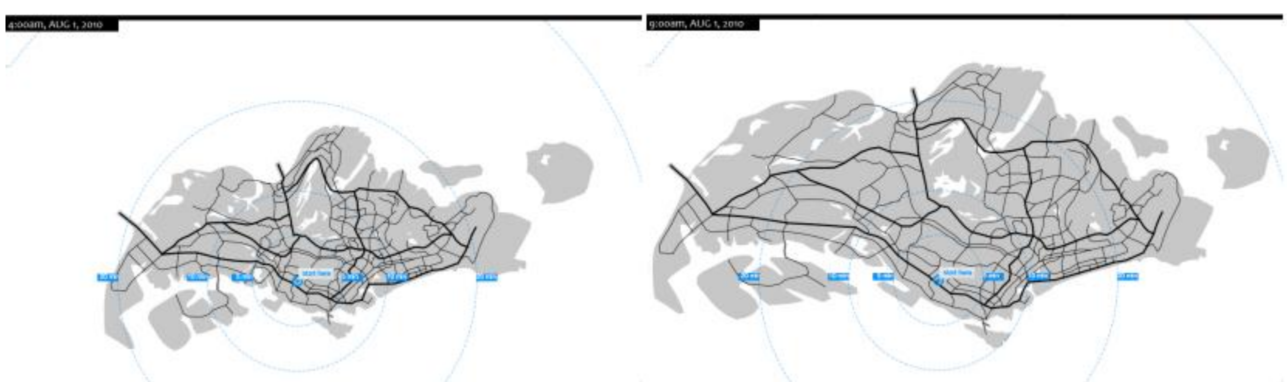


Fig. 19: Isochronic map showing geographical distortion to express traffic concentrations and clusters

Description

First either the user or the system will be select a “central” station as the point of focus for this view. The rest of the stations will be distributed in relation to this central point. Attributes that will be visualized in this visualization view will be travel time and fare rates. This view will also dynamically move through time in order for the user to observe the changes

- Lines and stations **color hue** will encode the color of the **line** (Red Line, Blue Line, etc.).
- **Circumferences** centered on the selected station will establish different **thresholds** of travel time.
- **Arrangement** of the stations will be according to the **travel time** from the selected station. Additionally stations will be connected according to the route that a user has to follow to get from the selected station to each station.
- Stations will be represented with trapezoids whose **width** will encode the **fare** required to travel from the selected station..

Interaction

The interaction in this screen will follow the same principles as the interaction mechanisms described in the Topological map window with the hope that these standardized methods to interact with the system will help the user in getting familiar with the application. There will be one additional interaction feature in this view.

- **Select and Observe:** The user can select a or a group of stations, lock them and later press play to see how these stations move through time.

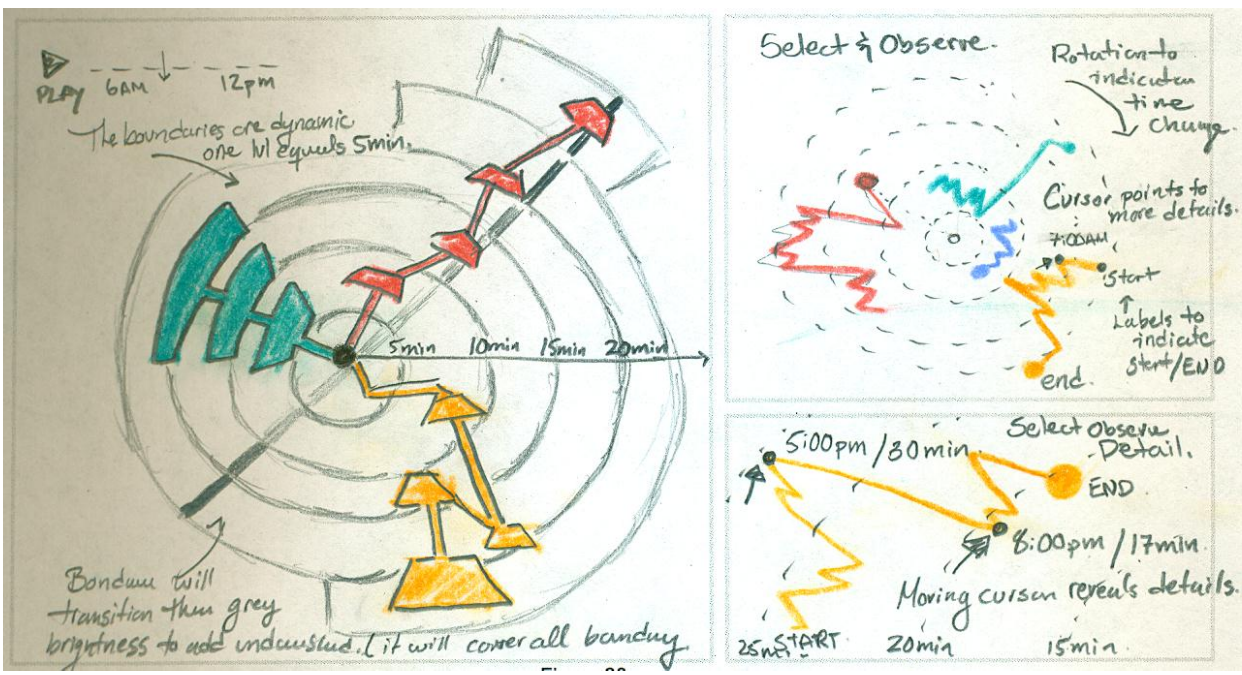


Fig. 20: Details on Isochronic Map

Ring Chart

Objective

Understanding that since we have geographic data its a reflexive impulse to map the data we do know that its important to show the data in another way in order to answer other — and sometimes more important — questions.

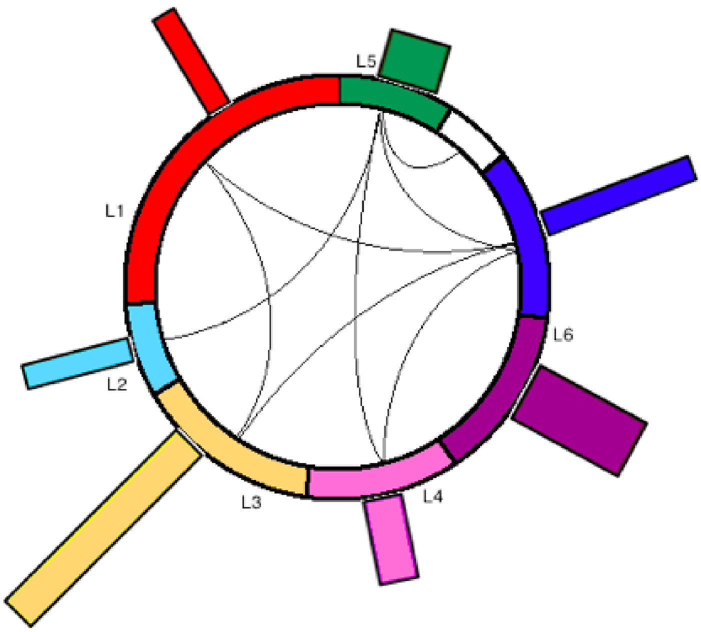


Fig. 21: Ring

Description:

This variant of a ring chart shows details of each line as well as the relationships between lines.

- Each **fan-shaped part** with different color indicates the **ridership** of that line during a day or a week. Higher values of ridership are encoded by longer colored areas in the perimeter. For example, L1 is the line with a higher ridership in fig 21.
- The bar on each section shows two different variables, the **width** of each bar illustrates the **frequency** of buses on that line, the **height** of each bar means the average **time interval** between two buses on that line.
- Inside the pie chart, if two lines have **intersections**, then they are connected by a **curve**.

This visualization will help users get an overview of the entire transportation system, also it will let users quickly have access to information of different lines. The ring intuitively shows the distribution of workload of each line as well as the direct relationships between each pair of lines. And users can get more information on a specific line by clicking the corresponding ring section (see interaction part below).

Interaction:

When users <u>click on a fan-shaped area</u>, for example the red L1, the chart spins (counter)clockwise until the L1 is on the right side of this ring, in the graph here, the circle spins clockwise about 100 degrees, puts the L1 section on three o'clock. Then several horizontal bars attached to this section show details on this line. Meanwhile all sections connected to L1 are highlighted.

Filtering Window

Fig. 22: Filtering Screen. Map Source: www.oberhaeuser.info

Description

The filtering screen is intended to be the principal tool for the user to explore the city and discover its secrets. The filter screen is an static physical map of the city being visualized. The above map is for the city of Würzburg, and was chosen for its simplicity to illustrate the user options. The user can dynamically manipulate context filter with the data in three different ways.

- User can <u>double-click on individual stations</u> one wants to target the visualization (Yellow Circle)
- User can <u>double-click and drag</u> to select multiple stations (Yellow line segment)
- User can <u>single-click on various places</u> on the map and form a polygon to select an area the user wants to target (Purple polygon).

Once the user has filtered the records, those that haven't been selected change to gray-scale or move down in hierarchical order depending on the visualization screen. This is called by the ProtoVis development team as the brush + link technique where contrast color allows the user to easily spot the data he wants to see in multiple views. The effectiveness of this technique can be seen in fig 14 and a fig 15 shows a sketch of our design.

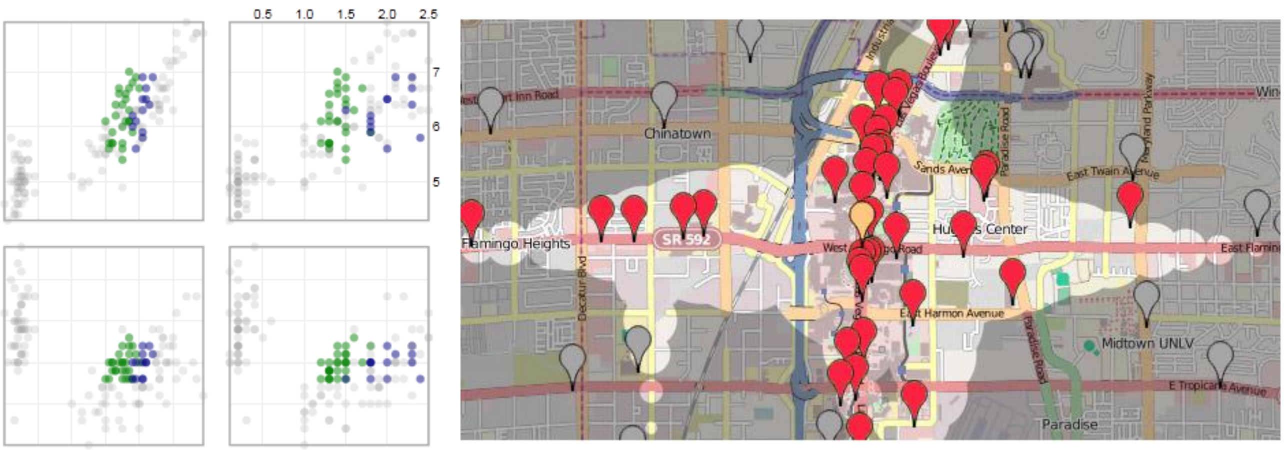


Fig. 23: Contrast Technique Source: http://www.mapnificent.net/ & http://mbostock.github.com/protovis

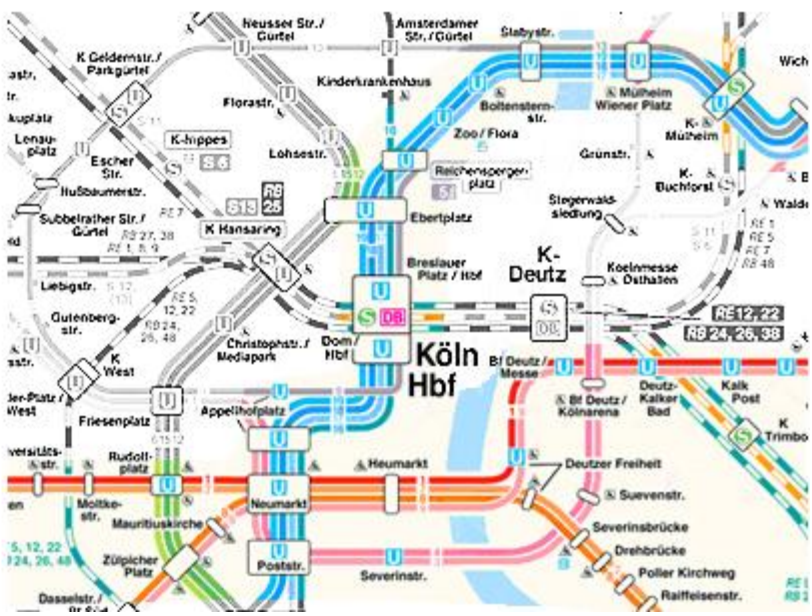


Fig. 24: Sketch of Technique used in our visualization. Unnecessary details will be removed later

Detailed Heat Maps & Step Graph Window

The goal for this screen is to bring supplementary detail regarding data shown in the different visualizations of the main windows. Specifically the window will provide visualizations detailing how the different key variables in our data (frequency, ridership, delays, etc.) have changed over time for the different stations and lines, helping the user to better understand the story of the data through a different visualization perspective.

Description

This component will be in the right part of the main windows and will contain the option to view the data in two ways.

Heat Map

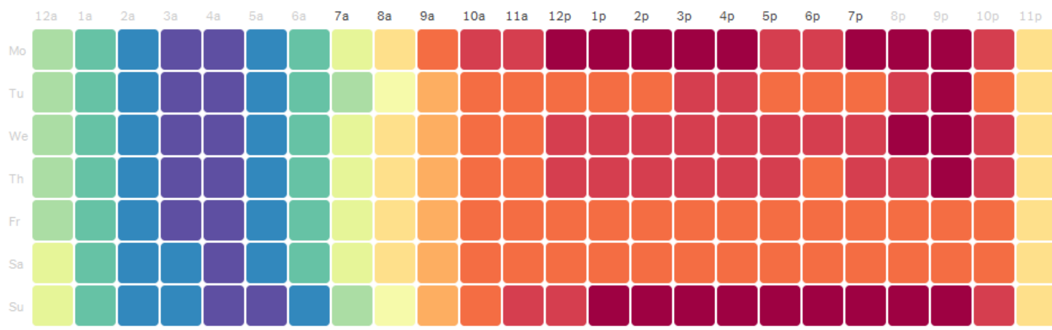


Fig. 25: Heat Map

Description:

One part it will contain visualizations showing the evolution of three key variables over time. It will consist on a set of three heat maps that will show how delay, frequency and ridership in the different stations has changed over time.

Each of the plots will have the following characteristics:

- **X axis** will contain **time** in intervals of one hour.
- **Y axis** will contain the stations. They will be rank according to their value the variable selected
- Cells **color hue** will encode the color of the **line** to which the station pertain. Note that if a station forms part of more than one line, it will be shown one time per each line because the values may be different for each of them.
- Cells **heat** will encode the average **value of the variable selected,** such as delays, for the different stations and time intervals.

Additionally, a "+" button will be is shown in the top right corner to of each of the plot areas to access further information about the variable shown on the plot (delay, volume or ridership).

The use of heat maps to show time series data has been previously studied in the literature, for instance Stephen Few talk about it in his book *Now you see it*. In the book, Few says that when the number of data cases or variables to show is very high heat maps have the ability to effectively show a large amount of data. This graphs will provide a clean way to see and compare how different variables for a large number of different stations have evolved over time.

Interaction

The component will support the following interaction from the user:

- <u>Pass the mouse over a cell:</u> will show a tooltip with detailed data values about the selected cell (station name, time interval, exact average value, etc.). In addition it will highlight
- <u>“Click” on the “+” button</u>: will open the *Further Details Window*.
- <u>Select a subset of the cells</u>: will highlight the data cases.

Additionally the user will be able to change the granularity to lines instead of stations. When selecting that, the Y axis of the heat maps will be updated to show the names of the lines instead of each individual station in each line.

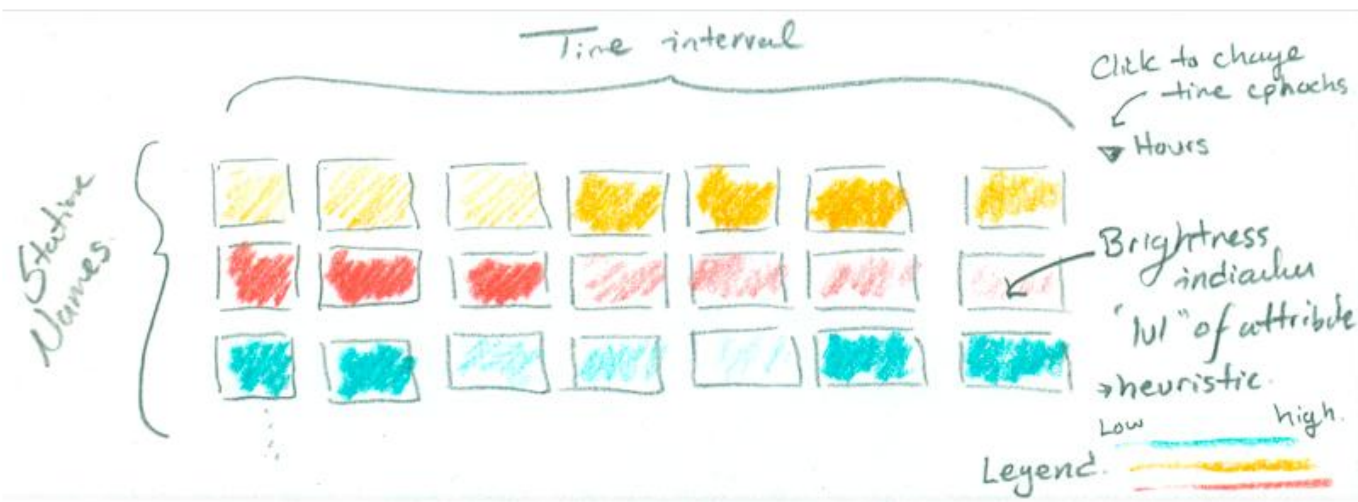


Fig. 26: Heat Map

Step Graph

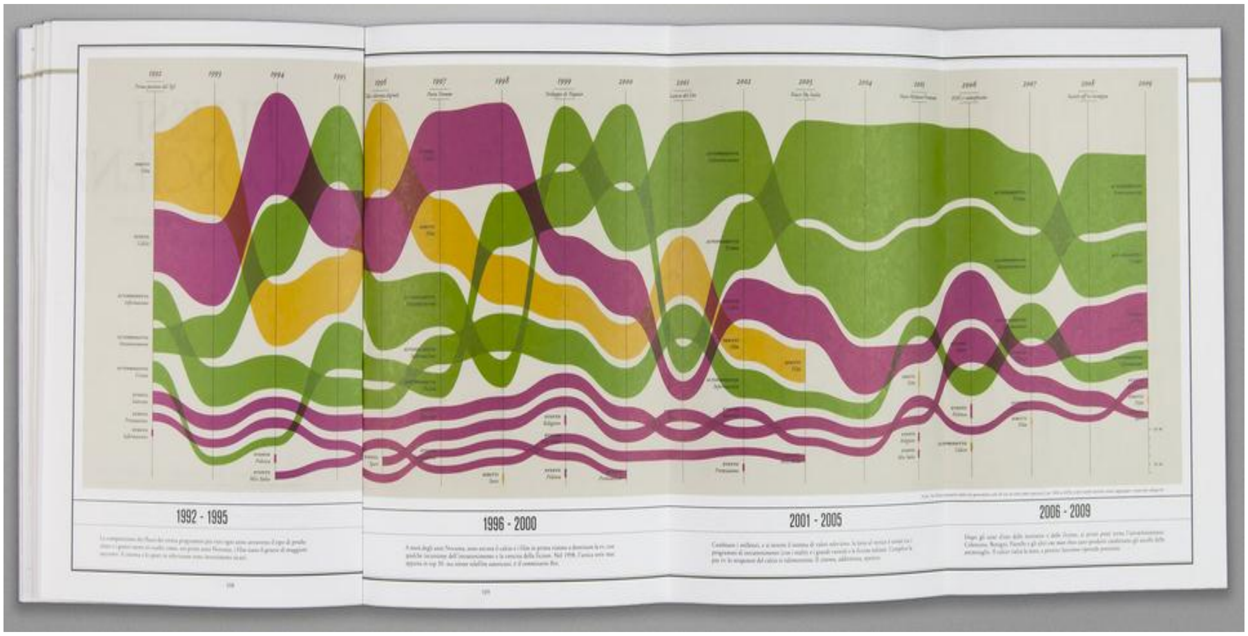


Fig. 27: Step Graph - X-axis represents time and the Y-Axis represents unique locations

This view will be principally be used for the user to observe how station "ranks" move over time and be able to encode other variables in other characteristics of the visualization.

The intention to show a step graph as an alternative visualization is that it provides the user a second perspective of how data is presented. In this view provides the user an understanding that events in a transportation network are actually discrete and that there actually exists an order to the chaos.

Description:

The variables in the data will be encoded in the following ways

- The **x-axis** will contain the time interval the data being observed "moves through"
    - The user will be able to select the granularity of the time intervals: Hours, Days, Months.
- The **y-axis** will either contain station names or route names
- Color **Hue** will identify the different routes or individual stations.
- **Width** will encoded the variable selected to visualize such as delay, bus frequency etc. The larger the width the will mean, for example in the ridership variable, the more riders in a given route.

Interaction:

Like in the previous screen the interactivity will be standardized through out the whole application such as:

- Clicking on “+” to see further details
- Focus + Context (Proto Viz)
- Pan + Zoom (Proto Viz)
- Brush + Link (Proto Viz)

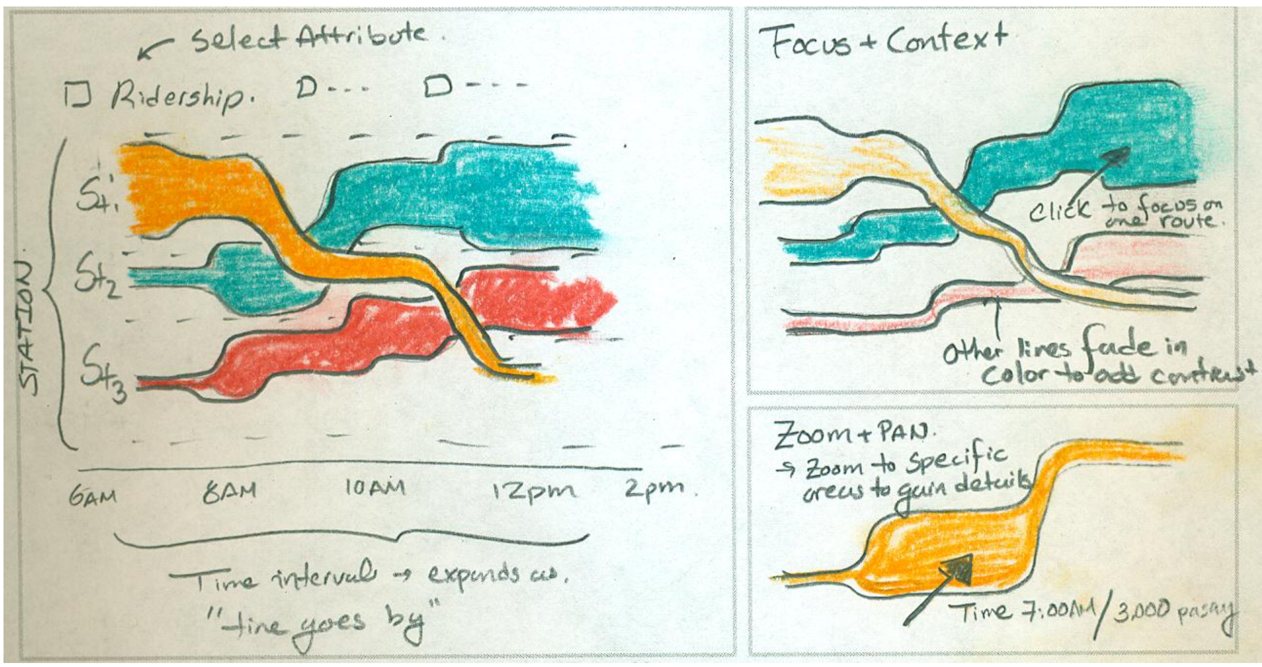


Fig. 28: Step Graph

Details of Data Window

City – CARACAS, VENEZUELA

Overview of Arrival & Departure times vs. Delays

Line	Departure Station	Departure Time	Arrival Station	Arrival Time	Delay (min)
Red South	Los Cortijos	7:15 am	Las Mercedes	7:18 am	-2
Red South	Las Mercedes	7:18	Altamira	7:22	3
Blue North	Petare	7:00	Los Ruices	7:15	4

Fig. 29 Tabular Data

Description

This component will contain the raw values for the different variables of the data cases that are being shown, such as the trips on the selected time interval. This will let the user explore the data in its maximum possible level of detail when looking for answers to specific questions.

The component will simply consist on a table showing the raw values for the data.

Interaction

The user will be able to select rows on the table to highlight the data cases. Additionally, a little button will be placed that will let the user hide this component in order to see a wider area of the map.

Further Details Window

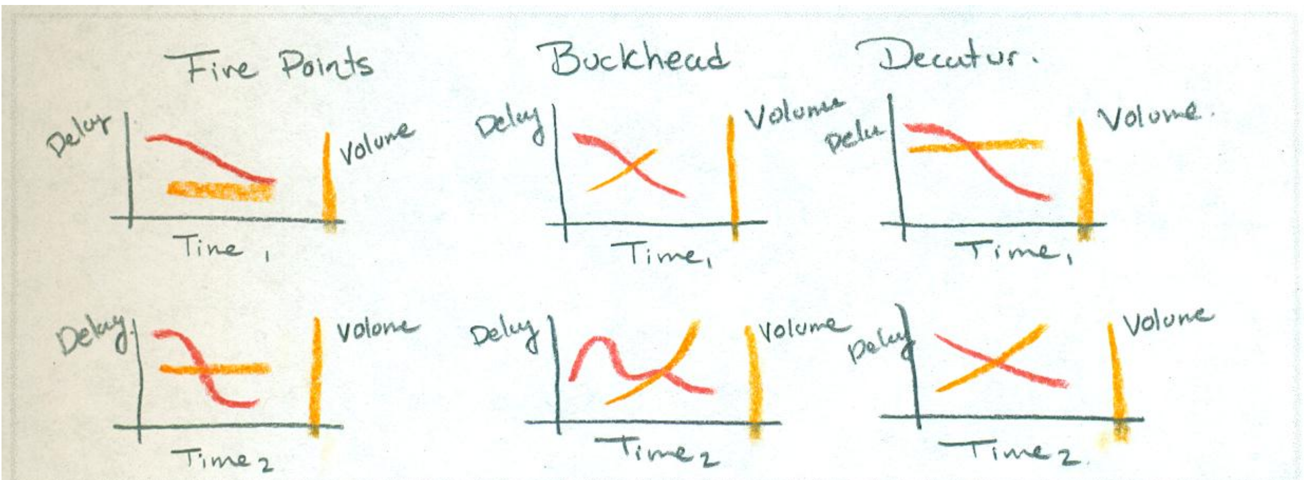


Fig. 30: Details Window

Description:

The graphs shown on the right components of the main windows do a good job of showing how the data has evolved through time, however, it still lacks the potential of showing patterns formed by the data or correlation between the different variables. Here our focus is to allow the user to access more detailed, accurate and quantifiable values.

This window focuses on a selected variable and contains a set of line graphs with double axis, aimed at comparing how the selected variable correlates to other variables and how this correlation has evolved through time.

The structure of this visualization has the following characteristics:

- There is a row of double axis line graphs, with one graph per station, where:
  - The **left Y axis** will encode amounts of **delay** time.
  - The **right Y axis** will encode **frequencies**.
  - The **X axis** will contain time in intervals of one hour.
  - In each graph, two lines will show the evolution of the average delay and frequency for that station.
- There is another row of double axis line graphs, with one graph per station, where:

- The **left Y axis** will encode amounts of **delay** time.
- The **right Y axis** will encode amounts of **ridership**.
- The **X axis** will contain time in intervals of one hour.
- In each graph, two lines will show the evolution of the average delay and ridership for that station.

In addition, stations will be organized in areas according to the line to which they pertain, in order to ease the data exploration.

Line graphs have been widely defined as the most effective way of showing time series data. However, they lack the ability to encode a large number of variables or to show a lot of data cases. In this visualizations, then, we are using the line graph technique to show several little pieces of data to provide the user with an effective way of discovering trends and correlations among variables. Having two axes in the graphs enhances the data exploration experience by providing a way to compare the evolution of two variables. Again this visualization contributes in telling the story of the data to the user.

Interaction

The user will interact with this window as follows:

- Pass the mouse over a line: will show a tool tip with detailed data values about the selected line (station name, exact average value, etc.).
- Select a subset of the lines: will highlight the data cases.

Additionally the user will be able to change the granularity to lines instead of stations. When selecting that, only one graph per line will be shown in each row instead of one graph per station.

Discussion

In order to achieve an effective visualization, there are two critical points: having a good representation and allowing the user to effectively interact with the representation.

Representational methods have been widely studied both in the industry and in the academia, allowing the creation of a wide range of effectively proven techniques. We have followed some of the best practices related to representational methods when designing the application in order to achieve a degree of effectiveness as high as possible. We use, for instance, line graphs to represent time series data, heat maps to encode a large number of data cases in little space and we encode a large number of variables in the main visualizations in order to let the user explore different facets of the data at the same time.

The interaction part, however, is a different story. Although there is more or less a consensus on the interaction methods that a good visualization should have, there is not an agreed general framework to represent that, which make more difficult to evaluate the effectiveness of the proposed design with the respect to the interaction with the user.

We find the framework proposed by Stephen Few in his book *Now you see it* an appropriate one to evaluate the design since it embraces a wide ranges of techniques. Therefore, we have used it as a "check list" for our design, commenting in each one of the proposed techniques with respect to the design:

- <u>Comparing</u>: The design allows a wide range of comparisons. In the main windows, different lines and stations can be compared while in the *Further Details Window* the evolution of different variables can be also compared.
- <u>Sorting</u>: This technique is not very relevant in our design given that there are no ranking-based visualization, therefore, the design does not contain any special mechanism for sorting.
- <u>Adding variables</u>: The designed filters allow for the (de)selection of variables to show, giving the user the opportunity to add or remove variables when he wants-
- <u>Filtering</u>: The *Filter Part* in the design can be used to filter the shown data cases or variables.
- <u>Highlighting</u>: The design contains interaction mechanisms to let the user select (or highlight) a subset of the data cases in a component with the automatic highlight of the same data in the other components.
- <u>Aggregating</u>: Both the *Detailed Heat Maps* and *Further Detail Window* allow the user to aggregate the data to the level of lines, being initially on the level of stations.
- <u>Re-expressing</u>: This technique is more common in general purpose tools so it makes no much sense in our design. However, there is a way to turn the heat maps shown in the main windows into line graphs, which can be seen as a form of re-expressing.

- Re-visualizing: This technique is intended for general purpose tools, so it does not apply to the designed system
- Zooming and panning: The maps visualizations shown in the main windows allow the user to zoom and pan through the map.
- Re-scaling: As re-expressing and re-visualizing this technique makes more sense in general purpose tools, so we have not considered it for the design.
- Accessing details on demand: All the designed visualizations allow the user to see detailed data in form of tooltips when pointing with the mouse to different areas of the visualizations.
- Annotating: We have not considered this form of interaction for this project, but it would be a good idea to consider it there is some time left after implementing the currently designed application.