Theme River Email Visualization (T.R.E.V.)

Introduction

Our theme river visualization tool (TREV) is prototype software that visualizes thematic variations. We have chosen to represent thematic variations of the number of emails sent and received by a variety of contacts over time. The themes are represented as coloured currents or a series of connected nodes that represent correlated data. The design decision to implement a theme river style visualization over other similar visualization methods (such as a histogram) is a result of the advantage of being able to see the correlation between points because of the continuous line that defines “topic” boundaries.

Objective/implementation:

TREV is designed to help users identify patterns and trends of one’s email inbox and outbox. In our prototype we use time as a “serial dimension”. Users can interact with the visualization using a variety of methods: zooming, panning, mouseovers, filtering (all explained below). Each contact is represented by his or her own unique identifying color that is displayed in a legend below the graph display (see figure 1.0 below). The user can interact with the legend by panning, in order to see more user names.

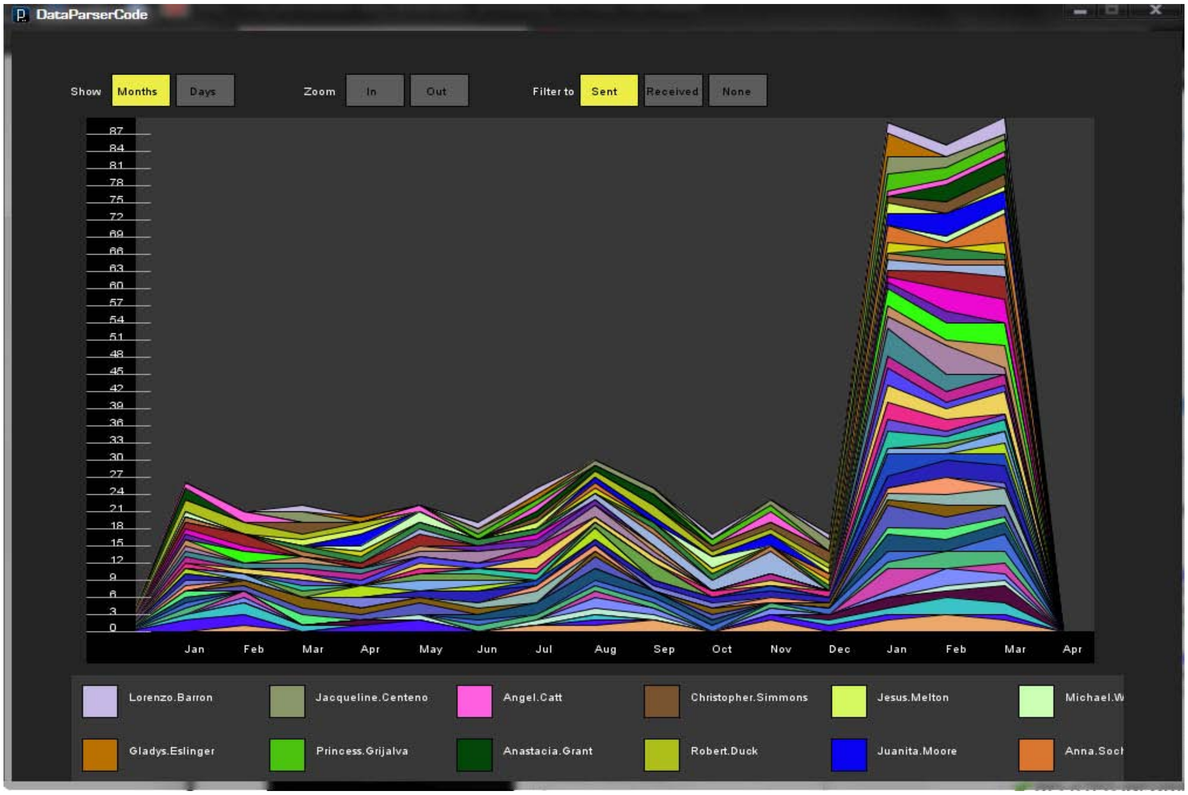

Figure 1.0: Email visualization by months, and sent emails only.

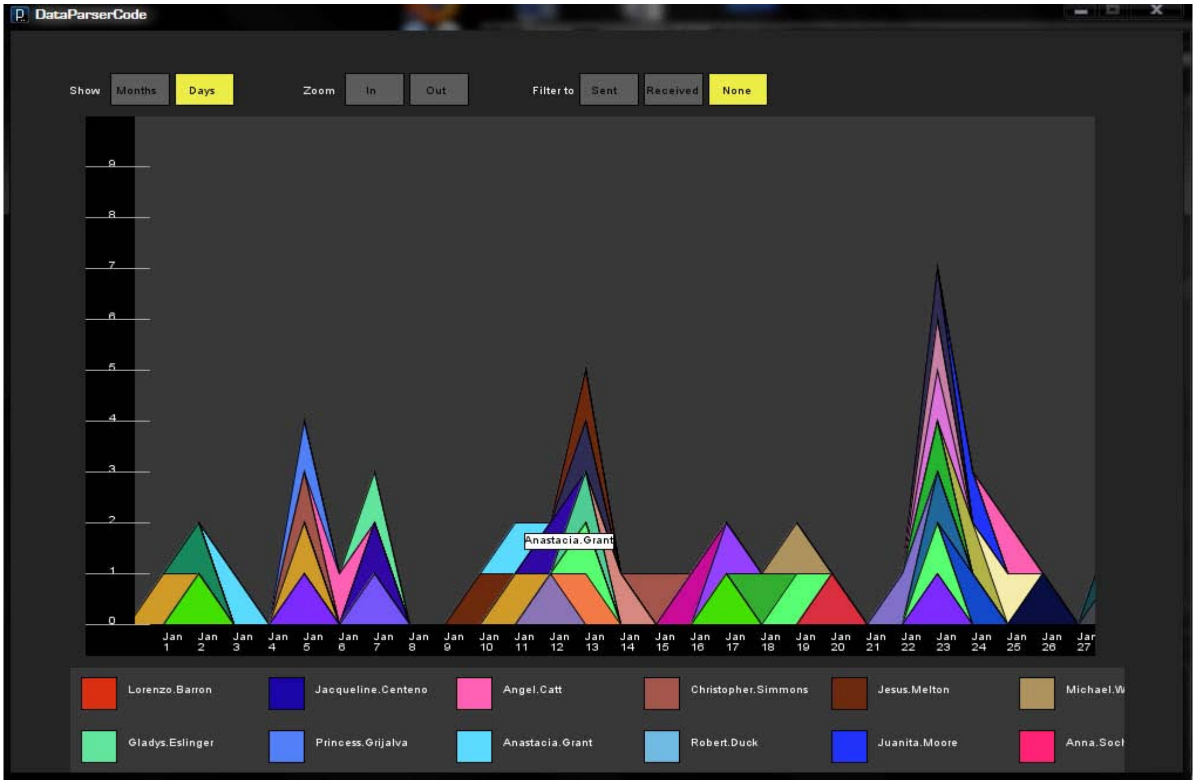

Figure 2.0: Visualization of days, featuring a display of both sent and received emails. Tool tip with contact name shown on mouse over.

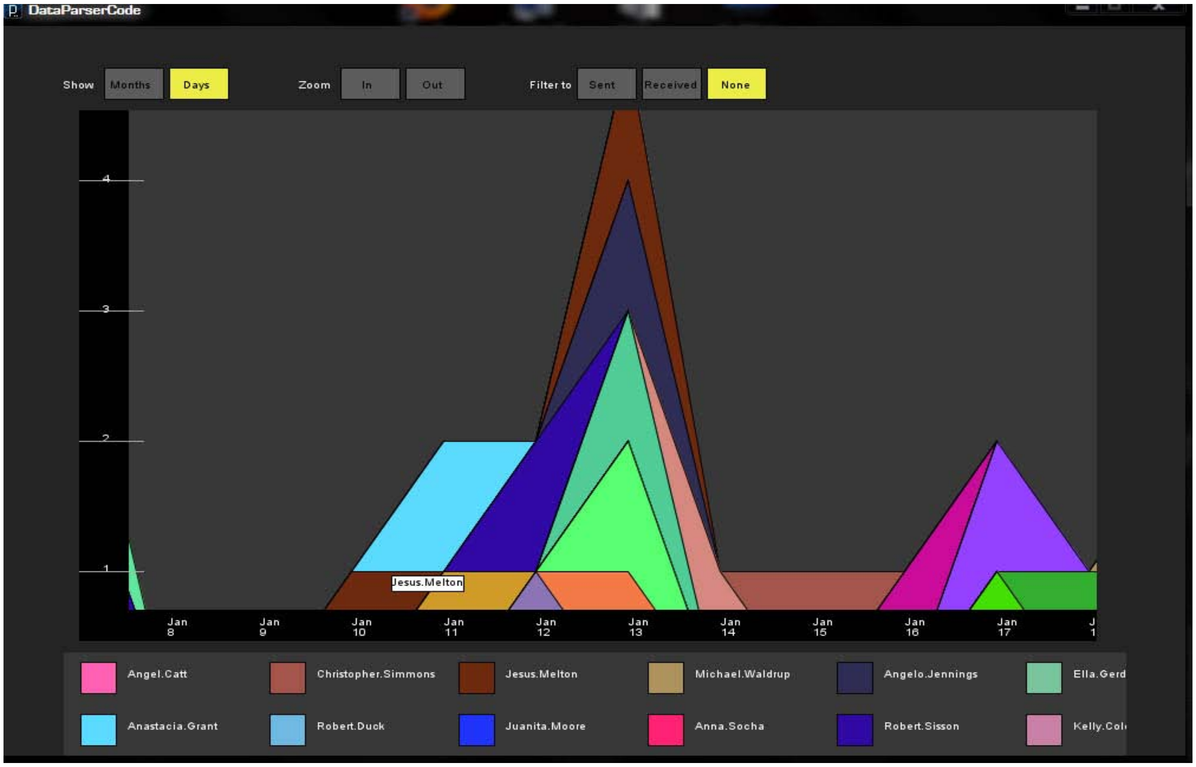

Figure 3.0: Zoomed in and panned view of days view. Shows a range from January 8th - January 18th. Number of emails range from 0-5. Also shows a panned contact legend.

Supported Interactions:

Zooming:

By using the control buttons at the top of the screen, the user will be able to zoom in and out by predetermined increments (to a certain limit). (See figure 3.0 above)

Panning:

Panning is handled by placing your mouse within the boundary of the graph (light grey box),clicking and dragging in any direction, and letting go to set the new graph position. Contrary to our initial design idea, omni directional panning is supported to allow users to pan upwards or downwards when zoomed in to a large increment.

User can also initiate panning with the contact legend in the same way as the email river graph. By clicking the mouse and dragging horrizontally in any direction the list of contact names will scroll accordingly. This was implemented due to the potential for having a large contact list and a limited space in which to display this list.

Changing units of time:

By using the custom made "combo box" users have a choice of what increments/units of time or time labels are displayed along the x axis. Our initial design featured hourly and weekly increments however for this iteration only months and days are supported(explained below).

Viewing metadata (mouse over):

Upon mousing over the area of a contact metadata in the form of the contact's name will be displayed in a "tooltip".

Implementation Challenges:

Time constraints:

Due to the limited nature of the time given (approx 3 weeks) we did not have a chance to implement additional features that may have aided our visualization software. Not included in our original report and design documentation was the added feature of contact filtering where a user would be able to filter the number (and which) contacts would be displayed.

Weekly/Hourly Display Difficulties:

As we all know weeks are essentially a subset of months that are comprised of 7 days. However when implementing our design, we did not account for the fact that each month has a different start date and end date which conflicts with the length of the weeks. This lack of consistency between months hindered our ability to implement a weekly sorting feature. Hourly display was not included because of the large amount of overlap in dealing with such a large amount of data.The code was left in to handle hourly sorting but due to the difficulties in displaying the data an option for the user to select hourly sorting was not included.

Dataset Difficulties:

While a dataset was made using Evan's inbox it was heavily weighted towards a few contacts and a few months which did not fully allow us to represent the true capabilities of our visualization. Therefore a randomized dataset was created with over 1000 emails total which while it is not actually "real" data it still shows the capacity of TREV in visualizing large numbers of emails (most inboxes most likely do not have 1000 emails saved).

Code breakdown:

TREV is an object oriented visualization tool that utilizes 3 classes to run:

Parser class(DataParserCode): This class is responsible for reading a specified datafile and parsing the data into a series of labels (i.e. "name", "identifier", "year", "month" etc) and data cases (i.e. bob.singer, sent, 2010 etc). This is a modified version of the original data parser given to us by Chris Shaw for the scatterplot assignment. A case switch was added to handle creating the individual data cases and to allow easy manipulation in the event that the labels of a particular dataset are not in a particular order.

Email class(Email): A simple class comprised of a constructor and return(get) functions to return the various metadata of each Email object (name, identifier, year,month etc).

Visualization/draw class(ThemeRiverPlot): The core class responsible for initializing arrays of names,months, days, hours, colors and mapping them to a visual dimension (translating and scaling). Class handles drawing of the user interface(buttons, displays, legend). Interactivity is done by calling processing's built in mouse event functions and by adjusting the scale and translation of the visualization according to the user's mouse interactions.