Visualization of airport and flight data

Acharya, P.

Award date:
2009

Visualization of Airport and Flight Data

Pavithra Acharya
(0666506)
Master's Thesis

Thesis committee:
dr.ir. H.M.M. van de Wetering
dr. M.A. Westenberg
prof.dr. J.C.M Baeten

Eindhoven University of Technology
Department of Mathematics and Computer Science
Eindhoven, The Netherlands
August 2009

Abstract

A frequent flyer, an airport official or an airline company may have several questions related to the flights and airports and the delays caused in them. This information can be provided in form of complex tables. It is not so easy to interpret these tables. This project deals with the creation of an interactive tool called $AirVis$ which visualizes these results instead of displaying it in the form of tables. The tool uses a framework called Prefuse to create visualizations. The usage of this tool is discussed with the help of examples. In the end, the pros and cons of the tool are discussed.

Acknowledgements

First of all, I would like to thank prof.dr.ir. J.J. van Wijk, dr.ir. H.M.M. van de Wetering and dr. M.A. Westenberg for providing me with an opportunity to work with the Visualization group at the TU/e. A special thanks to my supervisors, dr.ir. H.M.M. van de Wetering and dr.M.A. Westenberg who have guided me throughout the project and help me come this far. I would like to thank prof.dr. J.C.M. Baeten who has been a constant source of motivation and who will also be evaluating the project. I am grateful to prof.dr. Manohara Pai.M.M and prof.dr. Radhika M.Pai who gave me this opportunity to be enrolled into the Dual Mater program. I further thank my supervisor from MIT Manipal, Mr.Balachandra who will also be evaluating my project. A sincere thanks to my friends and family for providing me moral support and constant encouragement.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

If you are traveling to some place and your flight gets delayed or canceled, have you wondered whether you could have anticipated it? This can be possible if more information could be provided. What's more is you can also find out the cause of delay or cancellation.

If an airport official is experiencing complications in the airport due to some inefficient flight operations, he might want to identify the sources causing it.

A carrier company which owns aircrafts might want to analyze the performance of its flights. It might need to compare its performance to other carrier companies.

Similarly, all these stake holders can experience a diverse set of problems based on airport and flight information.

This master thesis deals with the creation of a tool called *AirVis* which intends to answer such questions in the form of visual abstractions. Basically, the visual abstractions provide the answers to the questions by visually representing the flights, airports and aircrafts information.

To answer the questions put forward by the users there is a need for data. We have downloaded this data from several different sources which took a total space of about 12-13 GB. This data was given in the form of comma-separated files. To handle this, we put the data into a database called SQLite by transforming it into database tables for ease of access.

The type of the questions differ depending on the user using the tool. The users, who will be using the tool can be anyone who has a query about the functioning of flights and airports. It can be a customer who wants to travel to a certain city, a pilot who needs to know the details of an aircraft, a carrier company which may need to know the status of its aircrafts or an airport official who needs to know the details of the flights leaving and arriving in the airport.

1.1 Content Description

The rest of the document is organized as follows :

Section 2 contains the description of the type of data provided and the different questions which can be framed by the user. Section 3 is on Visualization, which discusses concepts of how the data is visualized, the different interactions possible, the types of queries which can be formed to answer a question framed by the user and the wares used for making this tool. Section 4 comprises of a set of results which map to the appropriate questions framed by the users. Section 5 includes the implementation details of the tool like the structure of the classes, the database and the visualization toolkit being used. Section 6 concludes with the conclusions and future work.

Chapter 2

Domain Analysis

Our domain of interest is the airport data. The project deals with the performance of airports and airlines for which delay is the main criterion in most cases. In our project, the data are confined to the airports and all the commercial flights within the USA.

As mentioned before, there are different kinds of users who can use this tool. The users will have a lot of questions in mind based on flight information and these questions can be answered using the data. Higher and precise flight information have been derived from the existing flight data. The given data is in the form of four data sets with each one concerned with a different area of interest. The data sets are elaborated and explained in the below section.

2.1 The Data

The data consists of information about all the flights, airports and aircrafts in USA from April 1987 to October 2008 which consists of about 120 million records. We have identified four distinguished data sets: flight on-time data, airport data, carriers data and plane data sets. The airline on-time data is incurred from The Research and Innovative Technology Administration(RITA), which is the Bureau of Transportation Statistics of the USA (1). The airports data, the carriers data and the plane data sets are collected from the Federal Aviation Administration of USA (FAA)(4). All the four data sets are related to each other. The roles of each data set are explained separately below.

2.1.1 Flight On-time Data Set

A flight is a journey of an aircraft from one airport to another in a scheduled route and a scheduled time. Flight On-time data set holds information about the performance of each flight within the USA. The locations and timing are given. In our project we concentrate more on the delay and causes. This data set is given on a yearly basis. This section deals with detailed explanations of the data set's most important attributes and its usages. These attributes are displayed in the tables 2.1, 2.2, 2.3 and 2.4. The complete set of attributes are given in detail in Appendix A.1. The time of arrivals, departures and total journey times are given in this dataset displayed in table 1. Recording of arrival time and flight time are done at the destination gate whereas recording of departure time are done from the Origin airport gate. Scheduled times of arrivals and departures are also given. The scheduled time attributes are named in similar fashion as their actual time attributes, but with a prefix *'CRS'*. For instance, *ArrTime* is an attribute in the data set holding the values of the actual arrival time of a flight whereas *CRSArrTime* holds the scheduled arrival time. The locations and aircraft details are also given in the data set. The aircraft information will be needed, if the user wants to know the age, type and the manufacturing details of an aircraft. These attributes are briefly explained in table 2.2. The delay, cancelations and diversion details are also given. These details give the cause of cancelation and the exact delay occurred, if there

Table 2.1: *The Flight On-time data table : Date and time of flight*

Name	Description
Year	The year of flight which ranges from 1987-2008.
Month	The month of flight, given in the form of 1-12.
DayOfMonth	The day of the month 1-31.
DayOfWeek	The day of week in form of 1(Monday)-7(Sunday).
DepTime	The time at which the flight departs. It is given in local time with the format hhmm. This is the time instance when the pilot releases the brakes after closing the aircraft doors.
CRSDepTime	The time at which the flight is scheduled to depart, given in local time, hhmm. CRS(Computer Reservation system) time is the departure time printed on the tickets.
ArrTime	The time at which the flight arrives at the destination. It is given in local time in the format hhmm. This is the instance at which the pilot sets the brakes in the arrival gate.
CRSArrTime	The time at which the flight is scheduled to arrive at the destination, given in local time, hhmm.This is time departure time printed on the tickets.
AirTime	The time computed from the moment at which the aircraft leaves the ground from the Origin until it touches the ground at the destination. It is computed in minutes.

Table 2.2: *The Flight On-time data table : Location and Aircraft details*

Name	Description
Origin	Origin is the airport code of the airport from where the flight leaves. This is a reference to the iata attribute of the carriers data set.
Dest	Destination is the airport code of airport to where the flight is scheduled to arrive. This is a reference to the iata attribute of the carriers data set.
UniqueCarrier	A Unique carrier code is a code assigned to the aircraft based on the carrier who owned it during the flight.
FlightNum	Flight number of the flight made of 4 characters. assigned to every flight based on the date of flight and the carrier. This is not unique to every flight. Different flight numbers can be assigned to the same aircraft as it can fly several times in the same day
TailNum	Plane tail number is the registration number of a flight. This number uniquely identifies the aircraft. This is similar to the registration number of an automobile. This is a reference to the tail number attribute of the plane data set in section 2.1.4.

was any. If a flight was canceled, the *ArrTime* and *DepTime* would not be applicable. If a flight was diverted, the *ArrTime* would not be applicable. The delay attributes are explained in detail in Table 2.3.

If a flight was delayed, the cause of delay can be any one of the five types given in Table 2.4. If all the types of delays have value 0, it means that the flight was not delayed. If there exists a non-

Table 2.3: *The Flight On-time data table : Delay and cancelation details*

Name	Description
ArrDelay	Arrival delay is the difference between the actual arrival time and the scheduled arrival time . This can also be written as: ArrDelay = ArrTime - CRSArrTime in minutes
DepDelay	Departure delay is the difference between the scheduled departure time and the actual departure time and can be written as: DepDelay = CRSDepTime - DepTime in minutes
canceled	Value which tells whether the flight is canceled or not. 1 if canceled, 0 if not.
CancellationCode	Cancelation code is a variable which gives the reason for cancelation of a flight. The cancelation code is A if the flight was canceled due to aircraft problems, i.e circumstances within the airline control, for example baggage loading, fueling, crew problems, maintenance problems or aircraft cleaning problems. If the flight was canceled due to weather problems, code is B. This can be caused by extreme meteorological conditions which prevents the operation of the flight such as tornado, blizzard or hurricane.
Diverted	A value indicating whether the aircraft was diverted to a destination other than the original scheduled destination due to reasons beyond the control of the pilot or the company. The value is 1 if the aircraft was diverted and 0 if not. When an aircraft is diverted, the arrival time of the flight will be set as 0.

Table 2.4: *The Flight On-time data table : Types of delay*

Name	Description
CarrierDelay	Reasons for flight delay can be categorized to 5 types. One of them is the carrier delay. The delay which occurred due to circumstances within the airline's control. For example: maintainence and crew problems, fueling, aircraft cleaning, baggage loading etc. This is given in minutes.
WeatherDelay	Delay which occurs due to severe meteorological conditions such as a blizzard, hurricane, heavy rain etc fall under this category.
NASDelay	Delays which fall under the category of National Aviation system such as heavy traffic volume, air-traffic control and airport operations.
SecurityDelay	The delay which occurred due to security conditions such as voiding the terminals or concourse, malfunctioned security equipments, long queues for screening and re-boarding of aircrafts due to security breach.
LADelay	Delays caused by the late arrival of the same aircraft from its previous flight causing the delay of this flight.

zero value under a specific type of delay,it means that the flight suffered from the corresponding cause of delay and specified in minutes.

Table 2.5: *The Airports data table*

Name	Description
iata	International Air Transport Association code or IATA code is a unique 3-letter airport code assigned by the IATA to many airports around the world and all the airports in the USA.
airport	Name of the airport.
city	The city in which the airport is located.
state	The state in which the airport is located. All the states in the USA are represented by their abbreviation.
country	The country in which the airport is located. This data set only consists of airports situated within the USA.
lat	The latitude of the geographical area in which the airport is located. The latitude and longitude information is mainly useful for drawing the airports on a map based on its geographical location
long	The longitude of the geographical area in which the airport is located.

An example is given to illustrate the usage of the flight on-time data set. The flight information is of an aircraft which was scheduled to fly for Delta Airlines Inc. from Denver to Atlanta and gets diverted. The aircraft was scheduled to leave on 16-1-2008 which was on a Tuesday. The flight departed on the exact scheduled time, i.e. 15:20 but did not arrive on the scheduled time, i.e. 20:13 to the scheduled destination. This is known because the arrival time was not applicable. Since flight did not arrive at the destination, the delays are not known. Therefore, the type of delay columns do not hold any values. The diverted column holds a value of 1, which indicates that the flight had arrived at some other location other than the scheduled destination.

2.1.2 Airports Data Set

The Airports data set consists of location information of all the airports within the USA. Between the years 1987 and 2008, a lot of airports were built, removed or moved to some other locations. But this data set holds information of all the airports irrespective of the fact that it was removed or moved. There were about 3300 airports in the USA since 1987. Only 283 airports continued to exist in 2008. The attributes in this data set are described in detail in Table 2.5.

The attribute values of an airport in *Seattle* are given as an example to demonstrate the usage of this data set. The '*Seattle-Tacoma airport*' is located in *Seattle* city of *Washington* state in USA. The latitude and longitude values are 47.44898194 and -122.3093131 respectively.

2.1.3 Carriers Data Set

The Carriers data set holds the carrier codes of all the carrier companies and their respective names. It is useful to keep this information because, if a user is given a carrier code, it is not easily known as to which carrier it refers to. For instance, it is not easy to interpret that the carrier code '*9E*' refers to the carrier '*Pinnacle Airlines Inc*'. Totally, there have been at least 1492 Carrier companies operating in the USA between 1987 to 2008. The description of the attributes of this data set is given in Table 2.6.

Table 2.6: *The Carriers data table*

Name	Description
Code	The code assigned to the carrier company
Description	The name of the carrier company which the code indicates.

Table 2.7: *The Plane data table*

Name	Description
TailNumber	This a Unique Identification number assigned to every aircraft. It is the registration number of the aircraft.
Type	This refers to the type of the owner who owns this aircraft. If it is owned by a Carrier company in the US, the type of the owner is 'Corporation', else if owned by a carrier company not from the USA, the type is 'Foreign Corporation'. It can be also owned by a private company i.e. 'Individual' or 'Government' if the aircraft is owned by the Government or military services.If the aircraft is co-owned by two carrier companies then the type is 'Co-owned' or if a non US carrier co-owns a fight with a another US carrier company either from USA or elsewhere, then the type is 'Foreign co-owned'.
Manufacturer	The name of the manufacturer of the aircraft with the most famous being the BOEING and AIRBUS industries.
model	The model number of the aircraft.
aircraft_type	The aircrafts can be of 5 types : Balloons are a type of hot-air balloons, rotorcrafts and gyroplanes are types of helicopters and fixed wing single-engines and fixed wing multi-engines are types of airplanes. The other types of aircrafts such as gliders and parachutes are not taken into consideration here.
year	Year of manufacture of the aircraft

2.1.4 The Plane Data Set

The plane data set carries the manufacturing and the type information of every aircraft based on its tail number. There were about 5030 aircrafts flying between 1987-2008. This type of information can be useful for the flight accident survey, pilots, maintenance services and the carrier company who owns the aircraft. The attributes in this data set are briefly described in Table 2.7.

An example for this type of data set value is given for an aircraft with a tail number '*N107UW*'. This *BOEING* aircraft, built in 1999 was registered for a certain carrier in the USA. Therefore the type of owner is a '*Corporation*' and the manufacturer is '*BOEING*'. The aircraft was an airplane with a '*Fixed-wing multi-engine*'.

2.2 User Queries

The data given in the four data sets can prove advantageous for different kinds of users. A variety of queries can be solved by extracting information from these data sets. For instance, a customer or a traveler who wants to fly will want to know the average delay caused by a carrier in a certain route, a pilot or a crew member will want to know how an aircraft has been functioning in the

recent past, an airport manager will want to know how his airport is faring, a carrier company will want to know the operations of its aircrafts or a user who is making a survey about airport delays.

2.2.1 Customer Queries

A traveler or a customer who wants to buy a ticket might have several questions about the flight details and delay occurring in the route he would be traveling. The traveler might also want to know the alternative routes to a certain destination he wants to travel to. Some other examples of user queries can be briefly summarized as follows:

- Show delays of all carriers in Chicago Int'l on a Sunday flying to Salt Lake city Int'l.
- Which is the closest alternative route to travel from Washington National airport to Seattle-Tacoma airport.
- Compare the average delay causing in all airports.
- Show the top 10 must busiest airports in USA.
- When is the best time of day/day of week/time of year to fly to minimize delays?
- Has the delay of flights increased or decreased after the 9/11 attack?
- Does the delay causing in one airport affect the delay in another airport?
- Show the security delays caused in New York airport after 9/11 attacks.

2.2.2 Carrier company Queries

A carrier company which owns a set of flights might have several questions in mind concerned with the functioning of their aircrafts and the delays caused by them due to maintenance. The company might want to introduce new flights to the system, or might want to replace old flights. For this, the flight delay, the routes and traffic information is needed which can be got from our tool. The company might also want to compare its operations to other carrier companies. Here are a few examples of this type of queries :

- Show the average delay statistics of American Airlines Inc in all airports.
- Compare American Airlines Inc. and North West Airlines performance between Washington and Denver.
- Show the ages of aircrafts in my carrier company which fly from Denver to Chicago Int'l.
- Show the delays caused by the aircrafts in American Airlines and also show the age of it so that the company can identify the old and delay causing aircrafts.

2.2.3 Airport Manager Statistics

A Supervisor or a Manager of an airport will have several questions in mind about the functioning of his airport. He might want to compare another equally busy airport to his airport. He might want to know the arrivals and departures information of flights if he wants to reduce the delay occurring in peak time. Some other questions that might cross his mind are:

- How is my airport faring compared to the nearby airports?
- Which airline is causing highest delays and is faring badly so that it can be taken into consideration for removal.
- Show the number of flights arriving at airports at peak time.
- Show how the flights in airports have increased/decreased in time.
- Which carriers in my airport fly old and polluting aircrafts ?

Chapter 3

The Visualization Tool

The previous chapter described the type of data used, the type of queries which can be asked by the users and answered in this tool. These queries are formed by the user by handling a set of attributes given in the user interface. This section discusses the concepts used in making the visualization, the details of building a query and the concepts used in making the user interface.

3.1 The Visual Concepts

Before realizing the user questions into appropriate visual answers, the concepts used to represent these answers have to be made known. A library called Prefuse is used to represent these answers visually. It is discussed later in section 3.4.1.

The basic concept used to represent the answers of the user queries, is by using a labeled multi-digraph. A multi-digraph is a directed multi graph G=(N,E,L) which consists of a set of nodes (N), a multi set of directed edges(E) and a set of labels (L) for the nodes and the edges. A multi set is a set whose members are not necessarily distinct. In this context, the members are the directed edges. A multi set of directed edges is a set of ordered pairs of nodes which can occur multiple times. Each member of the graph(nodes and edges) is associated with a label which contains information about the respective member.

In reference to the context of this tool, a multi-digraph is designed to represent the airport data. The nodes represent the airports and the edges represent the flights. Every node is associated with a label. This label contains information of the airport which is being represented by that node. All the labels represent similar kind of information like the name, location and other attributes which define the characteristics of the airport (see Appendix B for full list of attributes). Likewise, each edge is associated with a label which holds information about the flight which the edge represents. The kind of information the labels represent is dependent on the query formed by the user. The connectivity of this graph also depends on the query.

The attributes which define the characteristics of the airports and flights can be mapped to the visual properties of the nodes and edges in the visualization of the graph. These attributes are classified based on the type of data (9). There are three categories of attributes: nominal, ordinal and numerical attributes. Nominal(categorical) attributes are attributes which are grouped according to a characteristic which does not follow an ordered sequence. Ordinal attributes follow some kind of order. As in, it is possible to say which item comes before or after another. Numerical attributes are quantitative attributes (which follow an order based on numbers).

The airport data has several nominal, ordinal and numerical attributes. As said previously, these attributes will be mapped to the attributes of the nodes and the edges. The nodes and edges can be characterized by four attributes: color, shape, size and location. Any type of attribute can be mapped to color and shape of a node or an edge. However the location and size attributes can

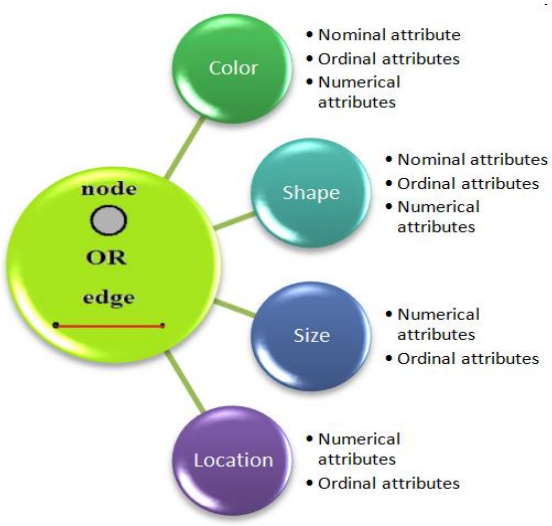


Figure 3.1: *Mapping of different types of attributes of airport data to the properties of nodes and edges*

only be ordinal or numerical. Figure 3.1 shows this mapping.

3.1.1 Nodes

The airports of USA are represented in the form of nodes in the visualization graph. Every airport is defined by attributes like airport code, airport name, latitude and longitude locations, average delay caused in each airport etc.

This section is categorized based on the four attributes of a node. Every section describes how mapping is done for different types of airport data.

Color Assignment
The color of a node can depend on any type of attribute. These attributes are chosen by the user in the user interface.

Color mapping for each type of attribute is explained below:

Color assignment for nominal attributes:
For nominal attributes, each unique data value is given a color in a linear scale from the user-specified color palette. The state and the city in which the airport is located is one of the examples of nominal data.

Nominal values are usually finite. Therefore, the number of colors used for coloring the nodes are also finite. The color legend representing the nodes also has a finite set of colors in the scale. This is equal to the number of colors used to color the node.

An example showing color assignment for nominal data would be mapping the state attribute of the airport to the color of the node. This means that every node belonging to the same state will have the same color. This is shown in figure 3.2. The figure is zoomed to central USA to get a clearer view (From here onwards, most pictures show only central USA in the images). Each

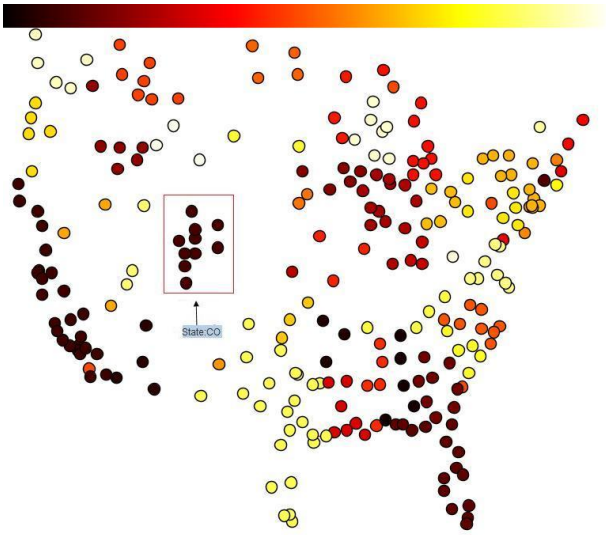


Figure 3.2: *Mapping of colors of node based on the state which is a nominal attribute. Hot color scale is used to color the nodes here*

unique data value is given a different color. The state attribute is currently mapped to the node color. A rectangle is drawn over a set of airports which come under the state CO(Colorado). All the airports in CO have the same color. Only the colors in the graph are shown in the color legend.

Color assignment for numerical attributes:

For ordinal and numerical attributes, color mapping is done by color ramping (5). The colors are assigned along a continuous scale. However, airport data does not have any interesting ordinal attributes. Therefore only numerical attributes are considered here. Some of examples of numerical attributes are the number of flights from the airports, the delay caused in each airport etc.

The color of a node is determined by the value of the mapped attribute. This is done by scaling the values and comparing it to the minimum and maximum values of the attribute. Suppose max and min are the maximum and minimum values of the attribute scale. Size is the total number of colors in the palette. Since there are 256 colors , *size=256*. Then the formula for color mapping a value into a color index in the color map and is given by the formula:

$$size \times \frac{value - min}{max - min}$$

An example showing the color assignment for numerical data is to show the average number of flights taking place in each airport. This is shown in the figure 3.3. It shows the color ramping for a grey scale palette. The airports with lesser number of flights operating are lighter in color than the airports with a large number of flights.

Shape assignment

The attributes of an airport are mapped on to a shape. This attribute is selected by the user. If no attribute is selected, then all the nodes in the graph will have the same shape.

Shape assignment for nominal attributes:

When an attribute is nominal, the shapes will be assigned to each unique data value in the order in which it is encountered. If the number of unique data values is more than the number of

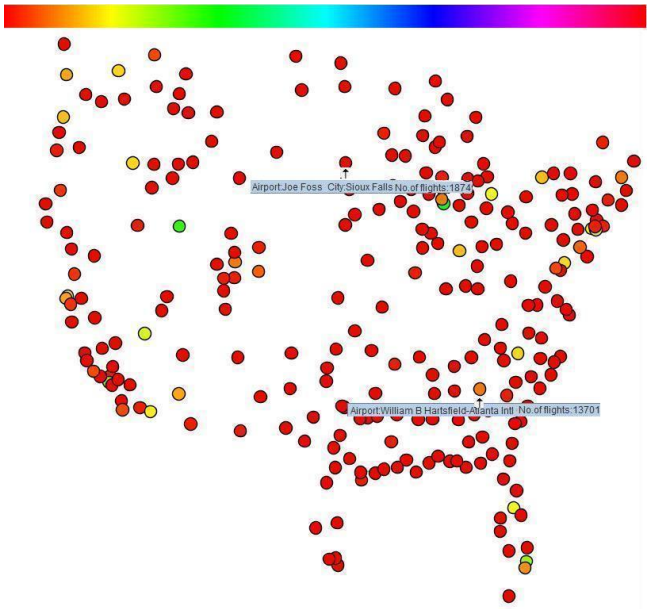


Figure 3.3: *Color assignment of nodes based on the average number of flights taking place in each airport. Rainbow color scale is used to color the nodes here.*

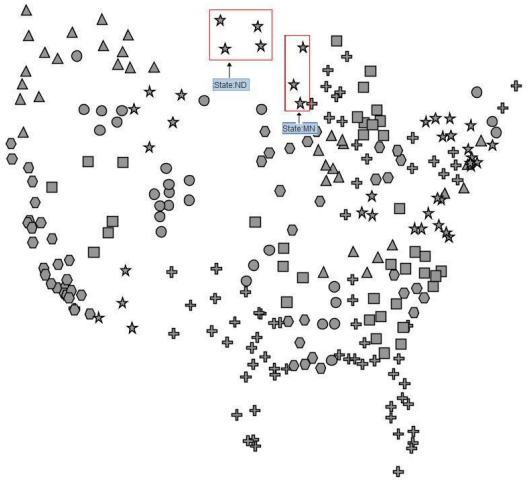


Figure 3.4: *Shape assignment of nodes based on the state in which the airport is located. Since there are 50 states in USA and only 6 types of shapes, different states will have same shape. A star is assigned to North Dakota and Minnesota airports*

shapes then redundant shapes will be assigned. Since the nominal attributes have a lot of unique values, each unique data value does not necessarily needs to have a unique shape. For example, if the state of the airport is assigned to the shape of the node (See figure 3.4). Since there are fifty states in USA and only six different shapes, some of the states will be represented by the same shape which makes it difficult to identify the airports in different states. In the figure, airports in North Dakota(ND) and Minnesota(MN) have the same shape assigned to it. Hence it is difficult to differentiate nominal attributes with more than 6 values by assigning it to the shape attribute. Since all the attributes of an airport have many more values than the number of shapes, it makes sense not to make shape assignments for nodes.

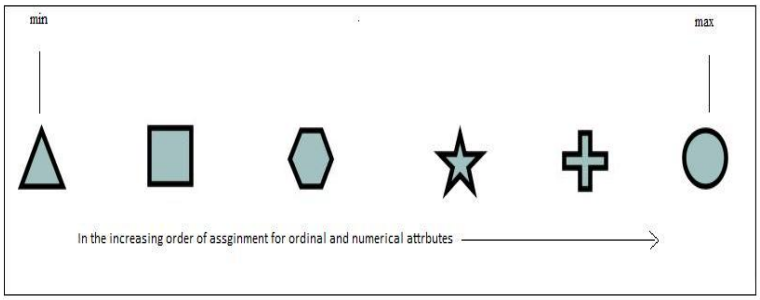


Figure 3.5: *Assignment of shapes to the nodes, in the increasing order, for numerical attributes. The shapes are ordered based on the number of edges in the shape.*

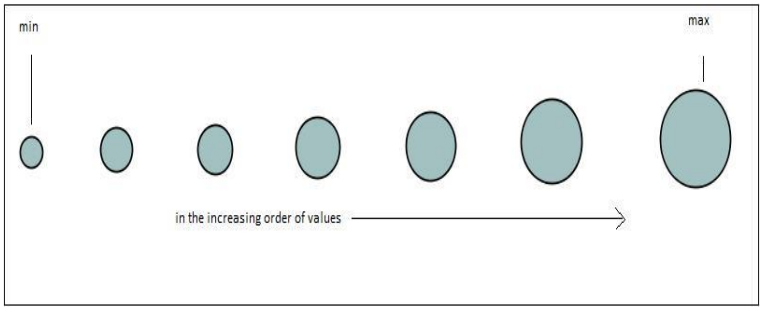


Figure 3.6: *Size representation of a node for seven different attribute values in the increasing order of values*

Shape assignment for numerical attributes:
By default, the attributes assigned for the shape of the node are nominal. Therefore, all the numerical attributes are transformed to interval based attributes where each interval of values is treated as a nominal value. Each unique interval is assigned to a specific shape in the order shown in figure 3.5. The interval with the least initial and final values are assigned to the first shape in the figure, the interval with the second least initial and final values are assigned to the second shape in figure and so on. The interval is built in a way such that the number of intervals are equal to the number of shapes.

An example for this type of shape assignment would be representation of node shapes based on the delay caused in each airport. The airport with delay less than 5 minutes are in the shape of a triangle, the airports with delay between 5 to 10 minutes are assigned to a shape of a cross and so on.

Size assignment
It is meaningful to assign ordered or numerical attributes to the size of the nodes. But the airports do not have any interesting data which can be ordered by size in a meaningful way. Therefore, only numerical attributes can be assigned to the size of the node.

The size of a node is determined by linear scaling the differences between the values of the attribute (2). If min and max are the minimum and maximum values of the attribute mapped to the size, then scaling of a $value$ is done by using the formula :

$$\frac{value - min}{max - min}$$

This value ranges between 0 and 1.

Figure 3.7: *Example of mapping the number of diverted flights in an airport to the size of the node*

The size assignment for a range of seven different values of an attribute is shown in figure 3.6. It is easily interpreted that the least value of the attribute is assigned to the smallest size, second least to the second smallest size and so on.

An example for size assignment of nodes can be the representation of number of diverted flights in each airport. The airports with the least size have one of the least number of diverted flights and the biggest airports in size have one of the largest number of diverted flights and so on. This is seen in figure 3.7.

Location assignment
The location of a node is dependent on numerical or ordered attributes. However, in our tool it is dependent on the X and Y co-ordinates. These values can only be numerical. The airport data provides the latitude and longitude values of the location of airport in the form of X and Y axis values. We have extracted this information from the airport data set. (refer section 2.1.2).

3.1.2 Edges

Flight route is a path followed by a set of flights whereas flight is a journey of a single aircraft. An example of a flight route could be all the flights flying from Miami to San Diego. Edges connecting any two nodes in a graph can either represent a flight or a flight route. The information of every flight occurred in the USA since 1987 to 2008 is given in the on-time data set section 2.1.1. This information is needed to represent the flights, as edges in the graph. The edges are not used to represent the actual path of a flight or a flight route, but to show the connection between the airports. From now on we refer, both flight and flight routes by one of the names.

The attributes of an edge are dependent on the query formed by the user. Initially, a flight route is characterized by three attributes, the origin airport, the destination airport and the average delay in the flight route. Some of the other attributes which might describe flight or flight routes

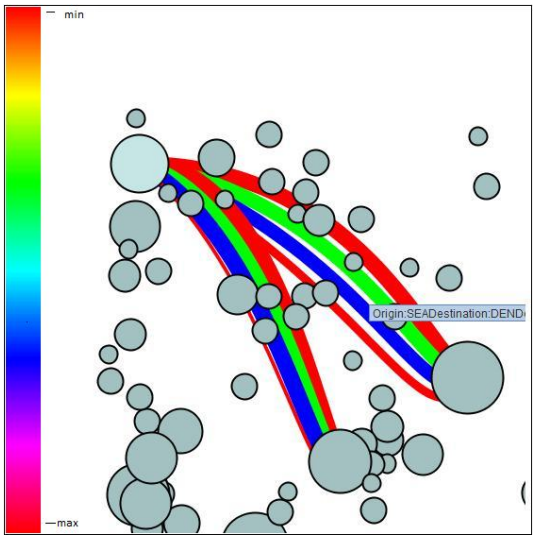


Figure 3.8: *Assignment of color to the first 4 months of flight and the delay to the thickness between Seattle and Salt Lake city and Seattle and Denver. Each edge represents the flight route taken every month.*

based on the query formed, are : the carrier of the flight, the day of the flight, the month of the flight, the tail number of the aircraft of the flight and a few others (refer Appendix B.2 for full list of attributes). Like airports, the flight routes also do not have any ordinal attributes. Therefore, ordinal attributes are not discussed in this section.

This section explains briefly how each type of attribute is mapped to the properties of the edges. It also includes a small section in the end which explains how multiple edges between two nodes are implemented.

Color Assignment

Any type of attribute can be mapped to the color of an edge. By default, the color of an edge represents the delay caused in the flight route. The default color palette used here is the rainbow scale. This can be later changed by the user in the user interface. (Refer general settings in section 3.3.2). The color legend used for coloring the edges is shown to the left of the visualization graph. Color assignment of edges and nodes are done in a similar fashion.

Color assignment for nominal attributes:

Some of the nominal attributes of a flight which can be mapped to the color of an edge by the user are: origin of flight, destination of flight, month of flight, day of flight, time of flight, carrier of the flight, manufacturer of the aircraft of flight, year of manufacture of the aircraft of flight etc.

Color assignment of nominal attributes can be explained with the help of the following example: If the user wants to compare the delay between Seattle and Denver, Seattle and Salt Lake city in the first four months of 2008, then the result will be as shown in the figure 3.8. The color of the edges are mapped according to the months of the year. Since the month attribute is a nominal value, each edge representing a month is assigned to a different color of the color scale.

Color assignment for numerical attributes:

There are a very few numerical attributes which can characterize a flight. Some of them are average delay, distance covered, number of flights in the flight route and so on.

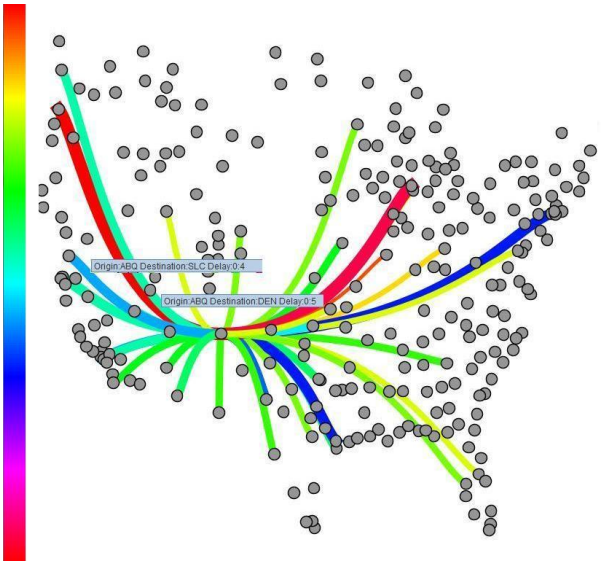


Figure 3.9: *Representation of average flight delay in the form of colors. Each edge represents the flight routes between two airports*

An example to describe the color assignment for numerical attributes would be the representation of delay in the form of color in figure 3.9. The colors of the edges which are closer to the end of color legend represent flights with higher delays and vice versa.

Shape assignment
The shape of an edge does not depend on any attribute.

Size Assignment
The size of an edge refers to the thickness. The method of size assignment to the attributes of edges and nodes is similar. Like nodes, it makes more sense to assign the thickness of an edge to numerical attributes only.

The same example in figure 3.9 is taken to shown thickness assignments. Here the delay is mapped to the thickness of the edges. Since the edge colored in deep red(represents month 4) is the thinnest in both the flight routes, it implies that the least delay caused in both the flight routes is on month 4 (April). This is implied from the fact that lesser delays have thinner edges and vice versa.

Location assignment
The location of an edge depends on the location of the source and the destination nodes. Therefore it does not depend on any attribute of a flight but indirectly depends on the x-axis and y-axis locations of the source and destination nodes.

Implementation of multiple edges
Depending on the requirement of a query, there can be one or more edges between two nodes. For example, if a user needs to know the delay caused by different airline companies between two cities, the result of this would require multiple edges where each edge represents a carrier company. Therefore curved edges with different heights are used (see figure 3.10(a)).

The Bezier curves used to draw the curved edges. A Bezier curve is a curve defined by a set of points (7). These points are called control points. We use a cubic Bezier curves for drawing the curved edges. Cubic Bezier curve need four control points points for defining a curve, two of which are the points defining the source node and the target node. Therefore we need to find the remaining two control points, cp_1 and cp_2.

When there is more than one edge between two nodes, an index is assigned to each one to differentiate between them. The first edge to be added between a pair of nodes is assigned index 1, the second to index 2 and so on. This number is also used to assign different heights to the edges in between two nodes. The height of a curve is determined by an imaginary line drawn between the node pair and the topmost point of a curve(See figure3.10(b)).

The algorithm for drawing these multiple curves is given below:

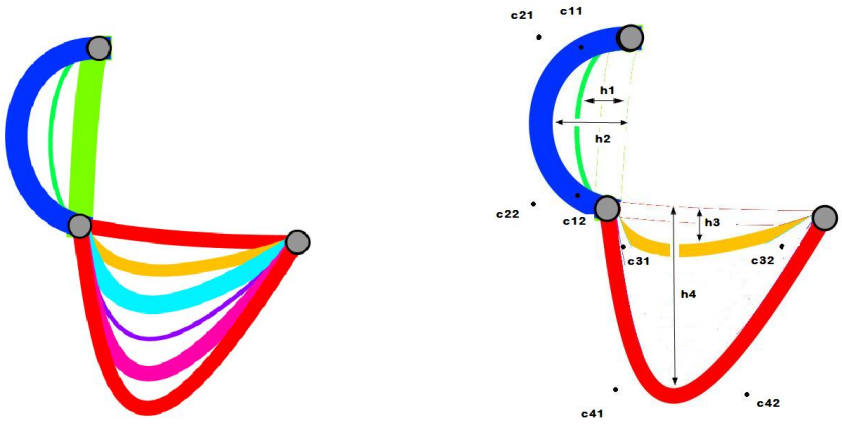
MultipleEdgeRenderer(double $x1$,double $y1$,double $x2$,double $y2$,int $index$)

► $(x1, y1)$ and $(x2, y2)$ are the start and end points
based on the number of edges between pairs of nodes.
$dx = x2 - x1, dy = y2 - y1$
$\vec{\alpha} = \begin{pmatrix} dx \\ dy \end{pmatrix}$
The above equation is a vector which represents the direction of the line$((x1, y1), (x2, y2))$
$\vec{\beta} = \frac{\binom{-dy}{dx}}{\sqrt{dx^2 + dy^2}}$
The above equation is a unit vector which represents the perpendicular direction to the line$((x1, y1), (x2, y2))$
$ratio$= a fraction of the length of the line$((x1, y1), (x2, y2))$.
$cp_1 = ((x_1, y_1) + (ratio * \vec{\alpha}) + (index * \vec{\beta}))$
$cp_2 = ((x_1, y_1) + ((1 - ratio) * \vec{\alpha}) + (index * \vec{\beta}))$
RenderBezierCurve$((x, y1), cp_1, cp_2, (x2, y2))$

In the algorithm, the control points, cp_1 and cp_2 are defined by a height $index$ from the line$((x1, y1), (x2, y2))$. $\vec{\alpha}$ and $\vec{\beta}$ are the directions at which $(x1, y1)$ should be incremented to get the control points cp_1 and cp_2 (see figure 3.11).

3.2 Queries

Basically, there are four steps in visualizing the graph. At first, the data required to form the initial graph are collected and rendered on the screen. Then the user interacts with the tool by filtering nodes. The user then makes selections in the tool according to the requirements. Edges can be filtered once the visual selection is done. As a result, a new image is rendered in the screen which is the required visualization graph. This process of getting the final image is also shown in figure 3.12. As seen in the figure, there are two steps where user interaction is needed: filtering and visual selection. These two steps form a part of a query. In this section, the explicit details of



(a) Example for multiple edges between nodes

(b) The green vertical edge has a height,*height1* and the points controlling the curvature are c11 and c22. the blue edge is of height,*height2* and the control points c21 and c22 and so on.

Figure 3.10: *Example for Bezier curves*

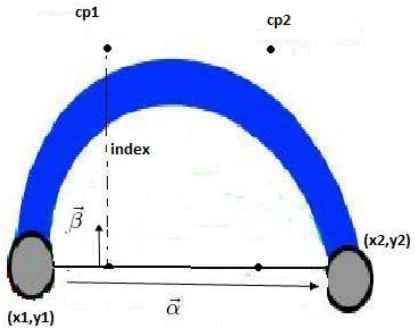


Figure 3.11: *Figure showing the significance of the variables used in the algorithm for multiple edge rendering*

filtering and visual selection are discussed.

3.2.1 Filtering

Filtering refers to removal of existing visual items (N,E and L) from the visualization. There are two types of items which can be filtered: the nodes and the edges. Edge filtering comes into the picture only after visual selection which is discussed in the next section. However, both node and edge filtering can be skipped and moved on to visual selection.

Node filtering is done by the user making a query which keeps the nodes required by the query and filters out the rest($N_{required} = N - N_{notneeded}$). Initially, all the existing airports in USA are rendered as nodes in the screen (N=all the nodes representing airports). Nodes can then be filtered using the node selection queries which is discussed in section 3.3.2. For instance, a query for filtering nodes would be choosing only the top 100 airports with causing least delay to display on the screen. The result of this is shown in the figure 3.13. Any kind of other selection done after node filtering would be performed only on the nodes remaining in the screen.

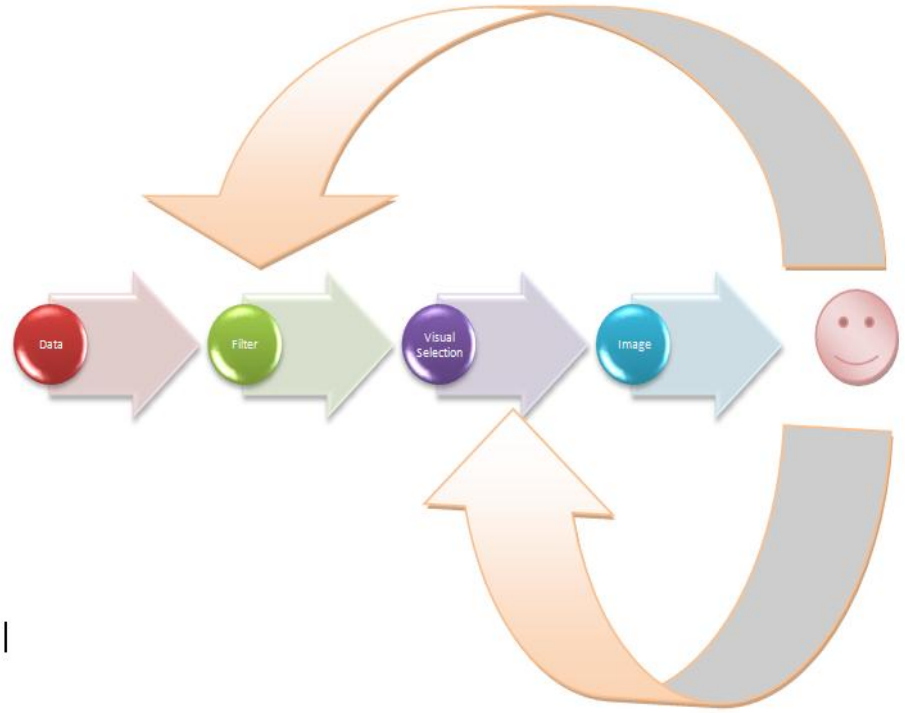


Figure 3.12: *Process of a query. Data is collected and filtered, the user can interact by visually selecting edges and performing filtering in the UI. The result is produced in the image*

There are no edges(E=ϕ) in the graph in the beginning, unlike the nodes, which are present on screen in the initial visualization. These edges will be created when the user performs a visual selection which triggers new edges to be added (refer section 3.2.2). Edge filtering can only be performed after the first set of edges appears on the screen. This filter action is performed by making a query which filters out the existing edges.

3.2.2 Visual Selection

Visual selection is a process of selecting or deselecting the attributes in the UI(User Interface). Visual selection can be done for both the nodes and edges. This can result in changes in the appearance of nodes and edges. A visual selection can result in three possibilities: change in appearance of nodes, change in appearance of edges, addition of new edges.

A change in the appearance of a node is a result of change in size, shape or color of a node. This change is a result of changing the attributes which are currently mapped to the node size, shape and color respectively.
A change in the appearance of an edge is a result of change in size and color. This change is also brought by selecting and unselecting the attributes of an edge and also changing the attributes which are currently mapped to the edge size, shape and color.

Besides the changes in appearance, edges can also be added by a visual selection. This occurs due to selection of attributes of edges which is discussed in edge selection settings in section

Figure 3.13: *Figure showing the result of a node filtering action which shows only the top 100 airports with least delay*

3.3.2.

3.3 User Interface Concepts

The user interface of *AirVis*, consists of a visualization display on the right and user panel on the left. A screen of this user panel is shown in the figure 3.14. This user panel is further divided into general settings and query settings. Each of these submodules are interpreted in the coming sections. The overview of the user panel is briefly summarized in the figure 3.15. Each module in the figure is explained briefly in the coming sections.

3.3.1 General Settings

Besides the settings made by the user to suit the query, there are a few general settings which do not concern the query. The general settings are mainly concerned with the selection of color scale used to color the nodes and edges.

Separate components exist for the color scale selection for nodes and edges both of which have the following four options of palettes: the gray scale, the rainbow color scale, cool colors (a blue to pink color scale) and hot colors (a white-yellow-red-black color scale). By default, the gray scale is used for color mapping the nodes and rainbow color scale is used for color mapping the edges.

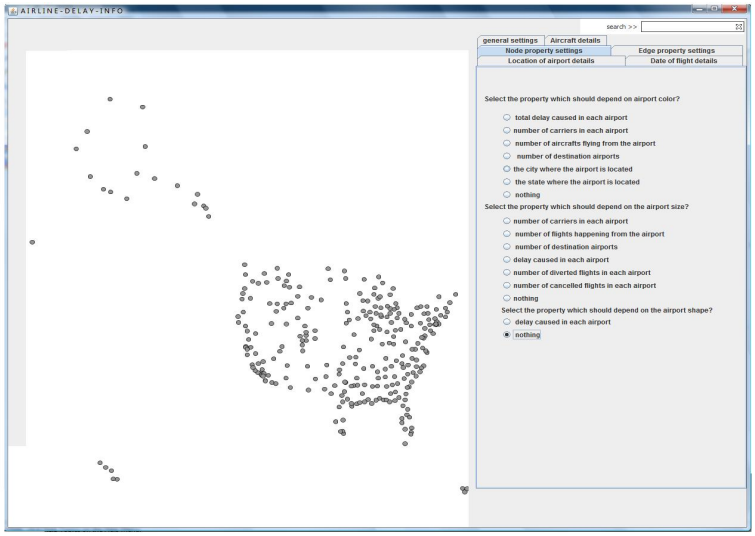


Figure 3.14: *Initial screenshot of the tool showing the visualization area on the left and user panel on the right.*

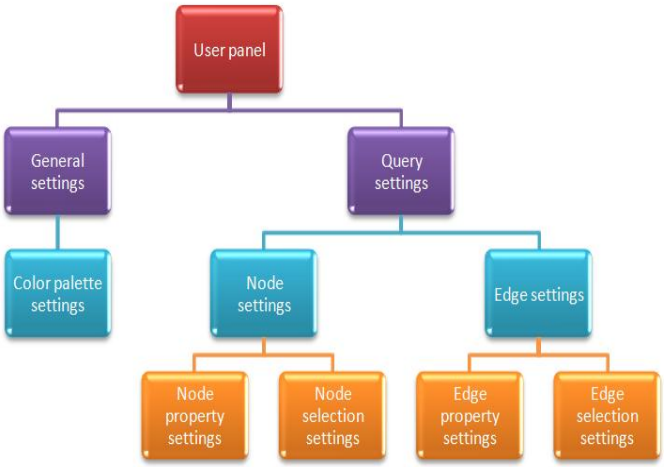


Figure 3.15: *Overview of the user panel. A user panel has a set of general settings and node settings. General settings constitue the color scale selection. Query settings can also be further divided into node and edge settings each of these components have a property settings component and a selection settings component.*

3.3.2 Query Settings

Query settings are settings made by the user to form a query. Query settings can be classified into node settings and edge settings.

Node settings

Settings which bring out changes in the appearance of the nodes of the graph in the visualization area are called node settings. Node queries can further be divided into node property queries and node selection queries.

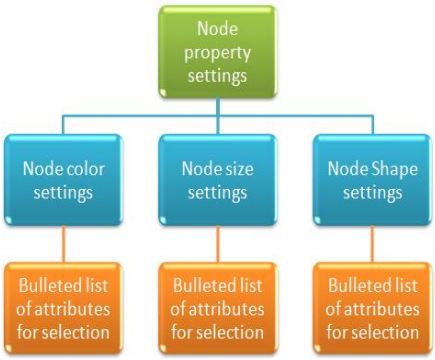


Figure 3.16: *Overview of how the node property is divided represented in the user panel. Each visual proeperty settings component of a node is in the form of a bulleted list.*

Node property settings:
The settings which bring out changes to the properties of the nodes like color, shape and size are node property settings(See figure 3.16). By default, no attribute is mapped to the color, shape and size of the node. Therefore all the nodes will be of the same color, shape and size. The user can set the attributes given in the user interface to these node properties, but N remains the same. Each property has a separate bulleted list of attributes from which the user can choose his preferences. The size of nodes can only be mapped to numerical attributes, the shape of the nodes can be mapped to nominal attributes and the color of the nodes can be mapped to nominal, ordinal and numerical attributes(Refer section 3.1). For example the size of the nodes can be mapped to the delay, the number of flights or the number of carriers in each airport which are all numerical attributes of the node.

Node selection settings:
Settings which deal with the filtering of nodes are node selection settings which changes N. There is a menu component for node selection. There are several options in this component which either hides or displays nodes based on particular criteria. For instance one of the options in this component can be, selection of the top ten busiest airports.

Edge settings
Edge settings show changes in the appearance of the edges. The edge settings are further divided into edge property settings and edge selection settings.

Edge property settings:
The edge property settings result in changes in the property of the edges like color, shape and thickness. Each property is a separate component where each component has a bulleted list of attributes of flights from which the user can select. Since the graph is a multi-digraph, there can be several edges between a pair of nodes based on the query.

Edge selection settings:
There are three ways by which an edge can be selected to be displayed : rubber-band selection, clicking and indirect method of selecting attributes of edges. Figure 3.17 shows all possible ways of edge selection.

Rubber band selection is a method of selecting multiple nodes from the visualization graph. This

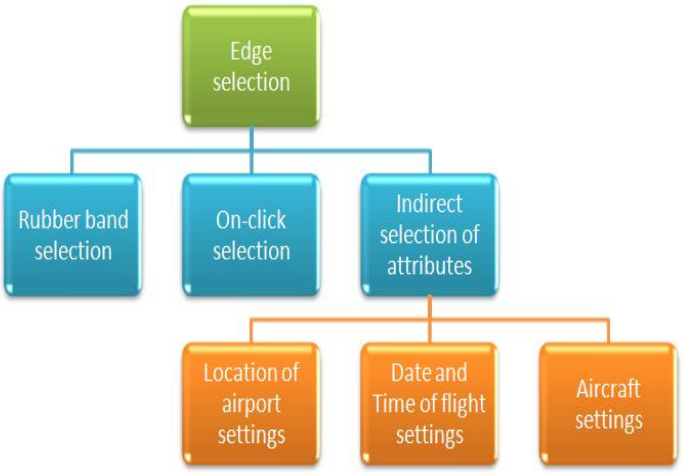


Figure 3.17: *Hierarchy of an edge selection. The edges can be selected in three ways (given in color green). The indirect method of selecting edges can be further classified into three types given in orange color.*

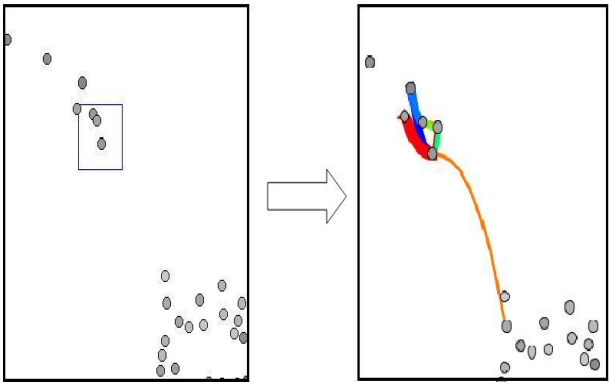


Figure 3.18: *Figure showing the selection of nodes to create edges by rubber band selection*

rubber band is formed by right clicking the mouse button and dragging it to form a rectangular rubber band. All the nodes which fall inside the rectangle are selected. The result of this selection would be displaying the edges showing the flights happening from all the airports which fall inside the rubber band. This is illustrated in figure 3.18. However, when a new selection is made, all the edges previously shown are removed from the Graph. Therefore, E is changed to a new set of edges E_{new} from E_{old}, here E_{old} was the set of edges created on last selection.

When a single node representing an airport is simply clicked, all the earlier edges representing the flights operating from that airport to all the other airports are drawn. All these edges will be removed when another selection is made.

Selection of edges is indirectly done by selecting values of attributes. Each attribute is a component with a table and set of bullets. The table consists of all the given possible values of the attribute. For instance, a carrier table contains a list of all the possible carriers which fly in the USA (as per the data given).The user can also arrange the carriers based on the increasing order of delay caused by each carrier, the number of aircrafts owned by the carrier, the number of des-

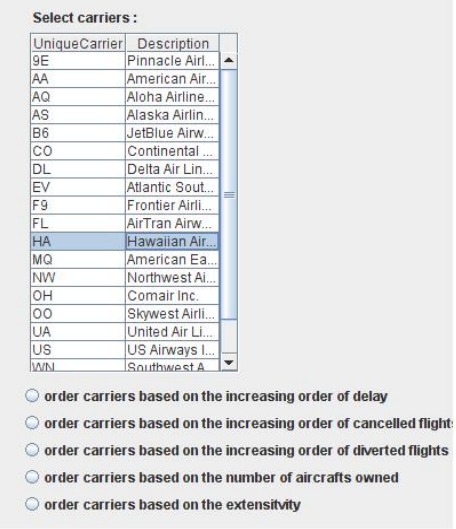


Figure 3.19: *Figure showing how the carrier attribute is displayed in the UI. The table consists of a set of all the carriers. The bulleted buttons shown below perform operations on the carrier Table.*

tinations to which the carrier flies, the number of diverted and canceled flights of the carrier and so on. All these options are given below the table in the form of bullets. This is shown in figure 3.19.

Like the carrier attribute, there are several other attributes which can influence creation or removal of edges. All these attributes can be grouped into airport attributes, aircraft attributes and date of flight attributes.

Attributes which are based on the location of the origin and destination of flight routes which have to be considered fall under airport attributes. Basically, selection of origin airports and destination airports is done here, based on the name of the airport, city or state in which the airport is located.
Attributes which are related to the date of flight come under date of flight settings. If the user wants to separately show the flights based on the day of flight, month of flight or year of flights, then the values in these attributes are selected.

Attributes which are related to the information of the aircraft come under the aircraft settings. Attributes like name of the carrier for which the aircraft flies, the tail number of the aircraft of flight, the aircraft type etc come under this category.

3.4 Implementation

AirVis is created with the help of rich set of libraries called Prefuse. SQlite is the database used for storing the data and accessing it whenever necessary. This section contains the specifics of the roles of Prefuse and SQLite in this tool.

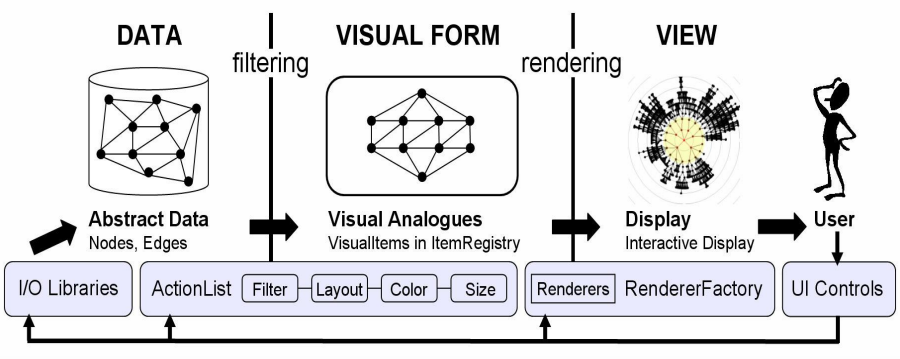


Figure 3.20: *The Prefuse visualization model. A list of actions transform the data into a visual form. The visual items are drawn in the visualization using renderers. The user can interact with these modules also setting the user interface controls which results in a change in the visualization*

3.4.1 Prefuse

Prefuse is an interactive visualization framework with a rich set of libraries (3). Prefuse is responsible for transforming the data into visual graph. The steps involved in this transformation are given in the figure 3.20.

At first, the data is collected and transformed into data tables. In our tool, the table contains the node tables and edge tables of the graph. These data tables are transformed to items which are suitable for visualization by performing filtering. The properties of the visual items can also be set by a set of actions. Finally these visual abstractions can be drawn in the display by using renderers. User interactions in the screen can also be done which would result in a change in the visualization. This can be caused by changing or updating the steps taking before (6).

3.4.2 SQLite

SQLite is an open source SQL database engine (8). SQLite is used in our tool to store the data.

Some features of SQLite makes it unique. The performance of data queries are comparatively faster than other popular databases, particularly the select function. It can also handle large databases. It is very simple and easy to use. It also provides a flexible set of features. Also, a database created in one machine can be used in a different machine with a different platform. Most of the other SQL databases reserve a fixed amount of space for each row depending on the declared data types. But SQLite in contrast, allocates the actual space needed to store the values in each row. This results in a relatively small database. This also improves the performance of queries.

Chapter 4

Results

A few interesting user queries were taken from the section 2.2 and executed in our tool. The results of these user queries are addressed here. The visuals are analyzed in detail. All the results are based on four months of 2008. This section is also divided based on the different categories of users.

Every query is organized in the following way. The first paragraph gives a brief introduction to the query. Next, an insight is given about transforming the query into visual selections and filtering which is followed by a snapshot of analysis of the result.

4.1 Customer Query Results

A customer or a frequent flyer usually asks queries concerned with the working of an aircraft or a flight route in which he is going to travel. He might also be interested to know how a certain carrier is functioning if he intends to purchase a miles card of that carrier. A few other questions can be also raised for general knowledge purposes. All these queries were listed out in section 2.2.1.

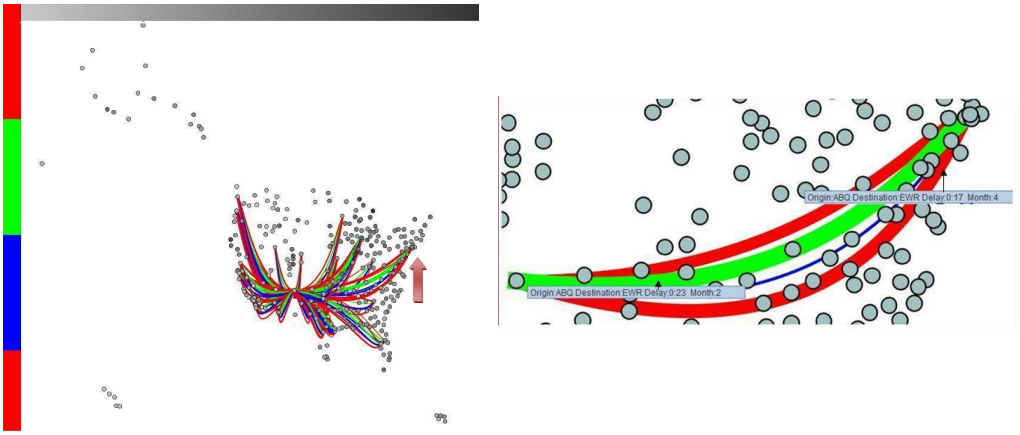
4.1.1 Customer Query 1

Which among the first four months of the years is the best to travel from Albuquerque(ABQ) to Newark(EWR) with minimal delay?

Result:

The result of this query is as shown in the figure 4.1(a). The color attribute is used to differentiate between months. This selection is done as follows: user interface→edge settings→edge color settings→month of flight. From the figure it is inferred that January, February, March and April will be are represented by deep-red, blue, green and red respectively. The delay caused in each month is represented by the thickness of the edge which is also chosen by the performing the following settings: user interface→edge thickness settings→delay. Accurate delay can be known by hovering the mouse over the desired edge for labels holding information.

If the user is intending to commute to a particular destination from Albuquerque, then he can choose the desired destination by selecting a location from the location of airport details component (Refer section 3.3.2). Here the user selects Newark(EWR) as the destination which is represented by the arrow in 4.1(a). It is also zoomed in for a clearer image in 4.1(b). We can see that the month which is assigned to the color blue has the thinnest edge. Blue is assigned to month 2(March). Therefore we can conclude that March is the best month to travel from Albuquerque



(a) Initial result

(b) Result of selecting Newark which is the airport pointed by arrow in (a)

Figure 4.1: *Figure showing the result of delay caused in the first four months of flight flying from Albuquerque.Each edge represents the flight route for each month. The color of the edges represent different months and the thickness indicates delay.*

to Newark.

4.1.2 Customer Query 2

Which is the best day of the week to fly from Seattle to New York with minimal delay?

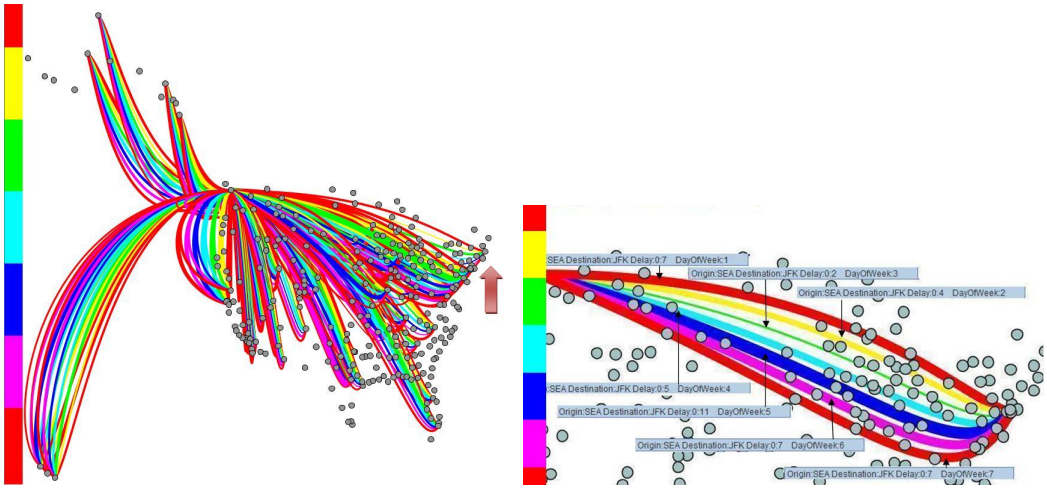
Result:

This query is similar to the previous query. Here we need to find the best day of week instead of month.

To formulate this query, the user has to choose an origin airport. In this scenario, the origin airport is Seattle Int'l (SEA). The user also has to select all the days of the week from the days of week component (Refer edge selection settings under section 3.3.2). Each day of week can be differentiated using colors. The delay is represented by the thickness of an edge.

The result of this query is shown in figure 4.2(a). Every flight route has 7 edges standing for 7 days of the week. Since there are a lot of flights operating from Seattle, the result will have a lot of edges.

Since the user wants to check the delay caused in the route from Seattle to New York city(JFK), New York is selected from destination component. This is exhibited in figure 4.2(b). We can see that the highest delay caused in this route is on day 5 (Friday) and the weekends and Mondays face high delays. The day with lowest flight delays is day 3 (Wednesday). Therefore, the best day of week to travel from Seattle to New York is on a Wednesday.



(a) Initial result

(b) Result of selecting New York which is the airport pointed by arrow in (a)

Figure 4.2: *The figures shows the delay caused on each day of week of flights flying from Seattle. Each edge represent the flights routes taken on each day. The color of the edges represent different days of week and the thickness indicates delay.*

4.2 Carrier Company questions

A carrier company might want to know how their flights have been operating in certain areas. They might also want to compare themselves with a rival carrier company to see where they stand. They might also want to look into the most potential airports so that they can set up their carriers in that airport.

4.2.1 Carrier Company Query 1

Show the delay statistics of Aloha airlines (AQ) all over USA.

Result:

This query is put forward by the Aloha Airlines.

This query can be formed by just choosing the appropriate carrier company from the list of carrier companies given in the carrier component (refer edge selection settings in 3.3.2). Since the main area of interest of this query is delay, we choose the delay to be represented in the form of both color as well as thickness of the edges.

Figure 4.3 shows the result of the query which is zoomed in by the user to show all the flight routes covered by Aloha airlines. Here we can see clearly see that all the flights flying to and from OGG (Kahului Airport, Hawaii) are causing a lot of delay. Out of this, the flight route from SMF (Sacramento airport) to OGG is causing the highest delay because the edge representing this route is the thickest among all. The color of this edge also represents maximum delay because the color is the existing maximum value in the color scale (In this example, the color is bright green). Therefore this flight route is operating with highest delay.

The company can further look into this issue by finding out the reasons for the delay. This op-

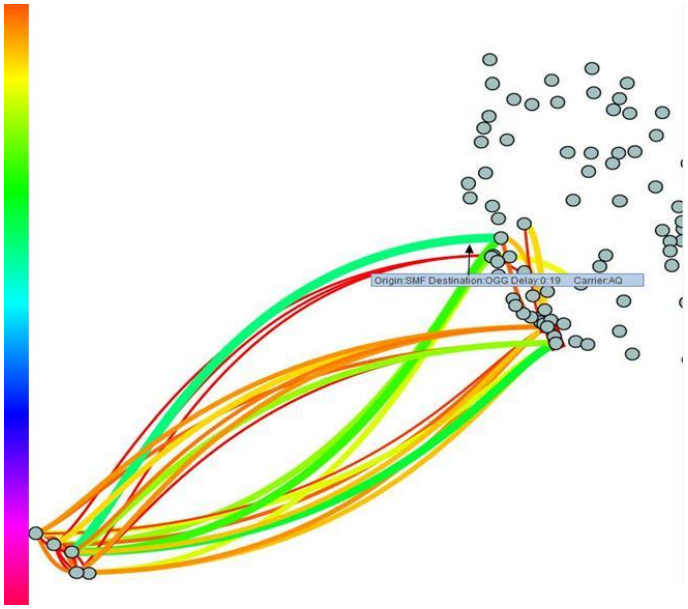


Figure 4.3: *Delay statistics of Aloha airlines. Each edge represents the flight route flown by Aloha Airlines. The thickness of the edge represents the delay caused in the flight route*

tion can be enabled by selecting different types of delay to be represented in the form of edge thickness. The five types of delays are mapped to edge thickness in figures 4.4(a), 4.4(b), 4.4(c), 4.4(d) and 4.4(e). From the figures, it can be incurred that the only delay caused in this route was the National aviation security(NAS) delay. Therefore, the delay in this route is not caused by the carrier company. However, a lot of carrier delay is occuring in the route from

4.2.2 Carrier Company Query 2

Show the delay statistics of a certain ”old” aircraft of Hawaiian Airlines (HA).

Result:

The delay statistics of old aircrafts can provide information for carrier company to decide whether the aircraft should be replaced or removed.

This query can be created by making the following selections: selecting the respective carrier company from the carrier component (Refer edge selection properties in section 3.3.2), selecting the tail numbers of the aircrafts which fly for HA and were manufactured in the 1980's. This can be performed by selecting the option of representing the tail numbers in the form of edge colors.

The result of this query is as shown in figure 4.5. We can see that the last edge colored in red is the thickest among most of the flight routes. This edge stands for the aircraft having the tail number ”598HA”.

Since the aircraft with tail number ”598HA” is giving the highest delay, only this aircraft is taken into consideration in figure 4.6. In this figure we can see the delays caused only by the 598HA aircraft. The average delay caused in between the routes from Honolulu to Phoenix and Honolulu to San Francisco are tremendously high (2 hour : 50 mins and 1hour : 5mins resp). The

carrier company can further look into this issue by checking the cause of delay as the done in the previous query.

4.2.3 Carrier Company Query 3

Compare the performance of the United Airways and Frontier Airlines flying from Denver airport to the all the airports in Washington state.

Result:

When a rival carrier company is performing well, a carrier company might want to compare its performance with respect to the rival so that the carrier company can know how big is the difference in performance. In this query, United Airways(UA) wants to compare its performance with the rival company, Frontier Airlines(F9).

This query can be created by the user by selecting appropriate origins and destinations from the location of airport details (Refer location of airport details in section 3.3.2). The color can be set denote the carrier companies and the thickness should be set to denote the delay.

The result of this query is as shown in figure 4.7. Here, we can see there are two airports in Washington to which there are flights from Denver namely the Seattle-Tacoma airport(SEA) and Spokane Int'l(GEG). The UA and F9 flights have been assigned to blue and pink colors respectively. It is clearly seen in the figure that the UA flights flying to the airports in Washington cause relatively higher delay compared to Frontier Airlines. Therefore the company will look into this issue by finding out the reasons for delay similar to finding causes for delay in example 4.2.1

4.2.4 Carrier Company Query 4

Show the ages of different aircrafts in my airline company (JetBlue Airways) which fly from the busiest airports of USA?

Result:

There are two queries embedded in this query. The first one is the query to find the busiest airports in USA. The second is to find the different ages of aircrafts which fly for the respective carrier company from the busiest airports only.

The first query is done by setting the size, shape or color of the nodes to denote the busyness of the airports. In this query, the attribute of the node chosen to represent the busyness of the airports is node size. This is done by setting the node size to show the number of flights flying into the airports in the nodes property settings component (Refer 3.3.2) because, the number of flights flying into the airports will determine the busyness of the airports.The result of the first query is as shown in the figure 4.8. All the busiest airports are the biggest airports in size.

If it is still not easy to determine which are the biggest airports, the user can go to any of the components representing the origin of the airports (like the origin airport component, state of origin airport component or city of airport origin component)(Refer Appendix B) and select the

button which orders the airports based on the number of flights flying in the airports. Then, the table holding the location attribute will be reordered and all the busiest airports will be present in the top most cells of the table of origins.

This query can also be performed by the filtering action (This is not done here). The user can select the option of displaying only the top ten busiest airports from the node selection settings. As a result only the top ten busiest airports will be displayed on the screen.

In the second query, the top four busiest airports are selected from the table which was ordered to represent the busiest airports. According to the table, the top four busiest airports are: Alaska, Chicago, Denver and Los Angeles. The carrier company for which the query is being done also has to be selected from the carrier component. All the years from the year of manufacture of aircraft table are selected. Finally the color of the edges is set to represent the different ages of the manufactured aircrafts.

The result of the query is as shown in the figure 4.9. Each edge represents the age of the carriers. It is seen from the figure that the flights flying from Denver have the largest diversity of ages of flights and all the aircrafts have almost the same average delay excepts for the aircraft flying from Chicago to New York City(JFK) which cause high delay although the aircraft was manufactured in the year 2000.

4.3 Airport manager queries

Airport managers or airport officials can formulate queries related to the performance of their airports. For instance, they might want to look into the factors which affect the delay occurring in the airport or compare their airport to other airports.

4.3.1 Airport Manager Query 1

Compare the delay caused in all the airports.

Result:

An airport official can compare delays caused in his airport with every other airport in the country just by using size or color to indicate it.

In this query, both size and color are selected to show the average delay. Since airports are represented in the form of nodes, nodes settings have to be done to formulate the query. In the node settings tab, there are three options by which delay can be represented: shape, size and color. But size and color assignments are more likely to give accurate results than shape. If the delay option is chosen in both size and shape representation of the nodes, then the result of the query is as shown in the figure 4.3.1. The red rectangle is drawn over the snapshot in 4.10(a) to indicate that the enclosing items are zoomed into (b)

Lesser delays are represented by the initial(lighter) colors and higher delays by darker colors of the gray scale which is given above the visualization.

In figure 4.10(b) we can see that among the airports within the figure, Jackson Hole airport (Jackson city) and Jack McNamara (Crescent city) are approximately of the same size. But the color used to shade Jackson Hole airport is darker than the color used to shade Jack McNamara . Furthermore, the color of Jackson Hole falls after the color of Jack McNamara in the color legend given in the figure.

4.3.2 Airport Manager Query 2

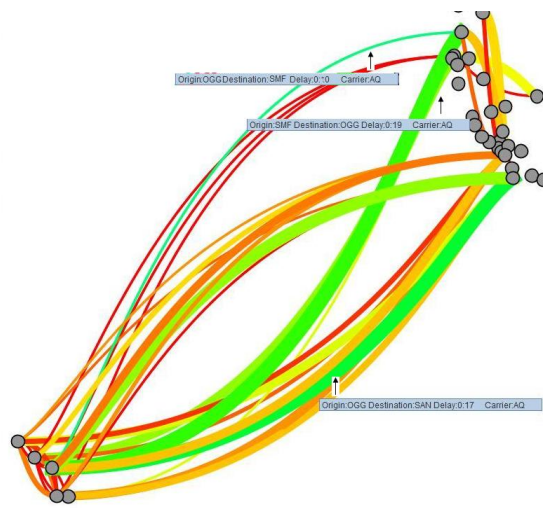
Which carriers in my airport(Austin) use old and polluting planes?

Result:

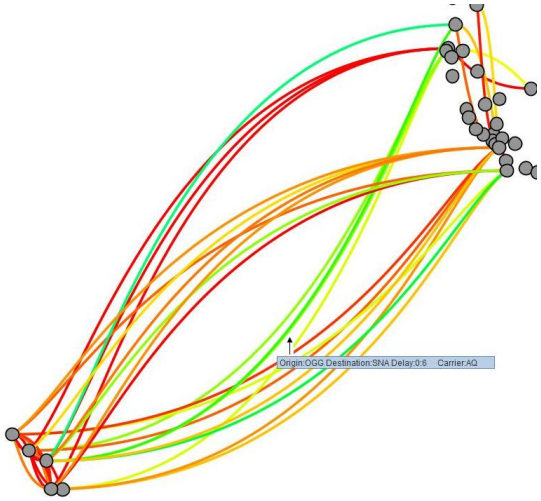
If the airport official wants to address an environmental issue, he might want to look into the carriers flying old and polluting aircrafts.

This query can be formed by selecting the required locations from the location of airport details (refer node selection settings in 3.3.2). All the carriers from the carriers component should also be selected. Appropriate years from the year of manufacture of aircraft component should be selected. In this query, manufacturing years ranging from 1959 to 1970 were selected. The color of the edges should be set to denote the carriers.

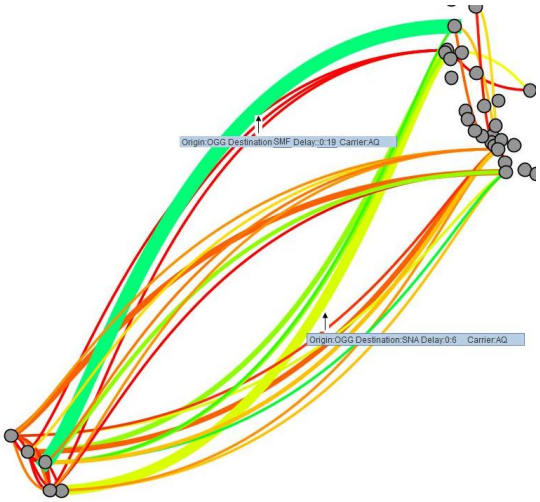
The result of this query is as shown in the figure 4.11. Based on the color of the edges, there are five carriers flying from Austin which have old and polluting aircrafts. In the figure, the blue edges represent the American Airlines (AA). From the figure, it is possible to say that AA has the highest number of old aircrafts out of which a few of them also produce a lot of delay. The rest of the carriers can be listed in the order of age and delay caused in the flights as follows: WN(Southwest airlines), NW(NorthWest Airlines), OH(Comair Inc) and CO(Continental Airlines).



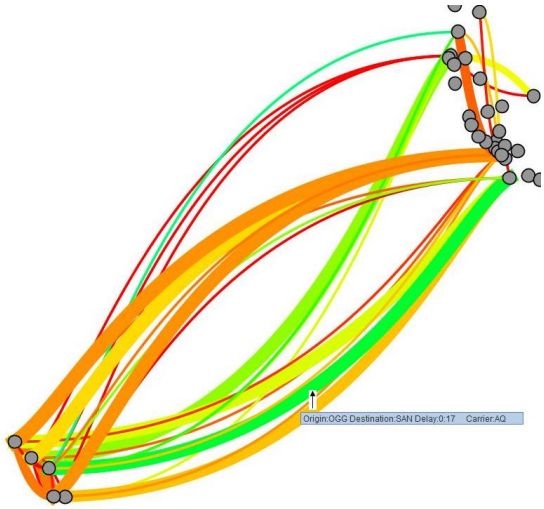
(a) Carrier delay



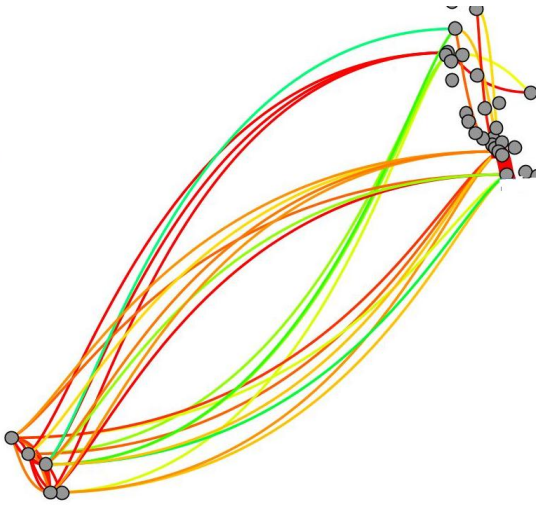
(b) Security delay



(c) National Aviation System delay



(d) Late Aircraft delay



(e) Weather delay

Figure 4.4: *The figures shows the different types of delays caused in the flight routes of Aloha Airlines*

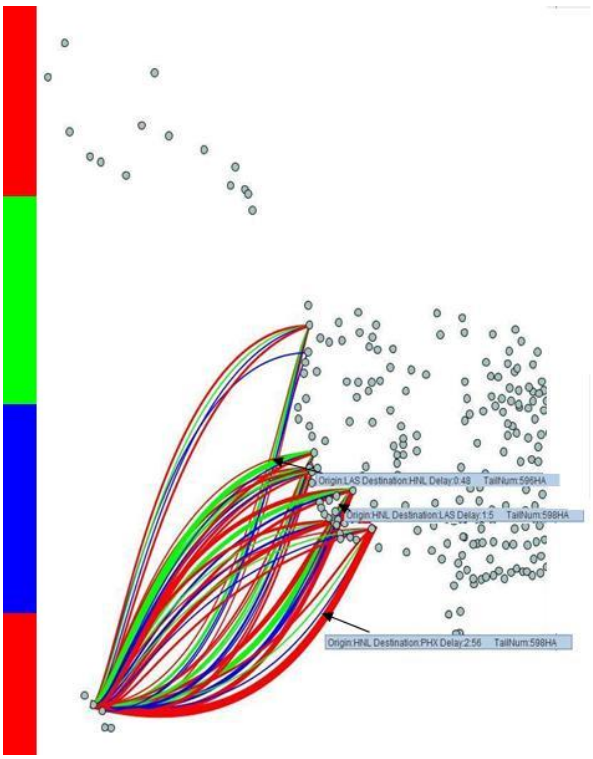


Figure 4.5: *Delay statistics of a few old aircrafts flying for Hawaiian airlines. The edge represents The thickness of edges represent the delay caused and the color of the edge represents the aircraft*

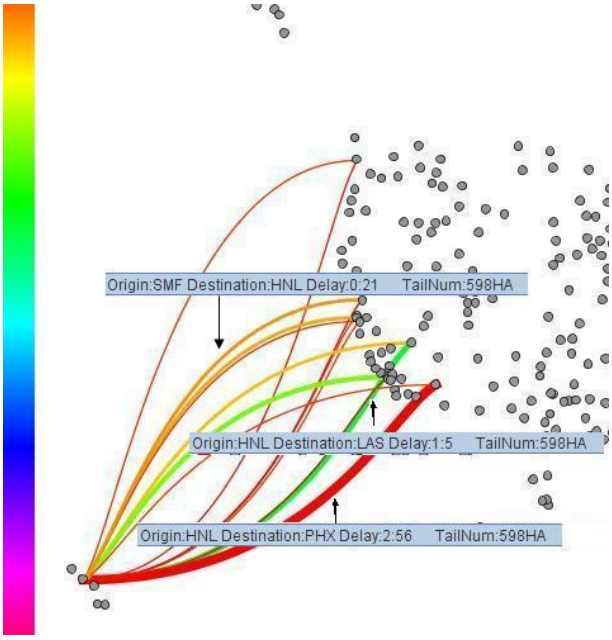


Figure 4.6: *Delay caused by the aircraft 598HA owned by Hawaiian Airlines*

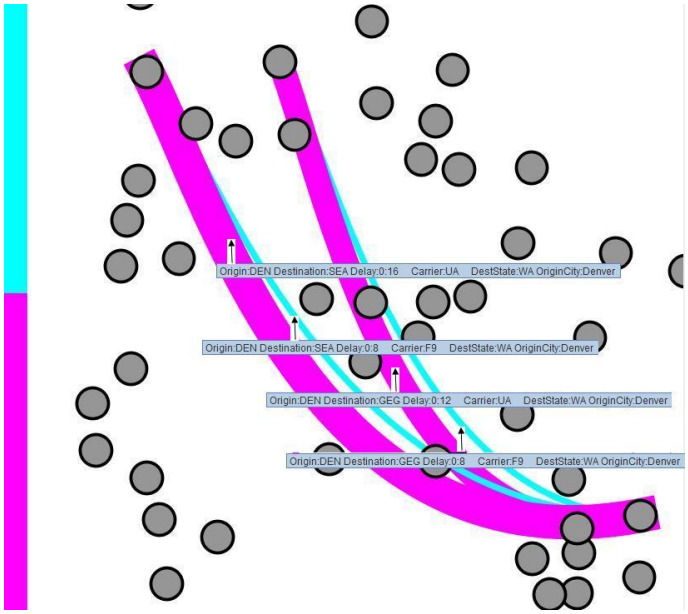


Figure 4.7: *Performance of United Airways and Frontier Airlines flying from Denver to all the airports in Washington state. The edges represent the flight routes of the respective carrier. The color of the edge differentiates the carriers and the thickness represents the delay caused in the flight route taken by the respective carrier.*

Figure 4.8: *Figure showing all the busyness of the airports by means of size*

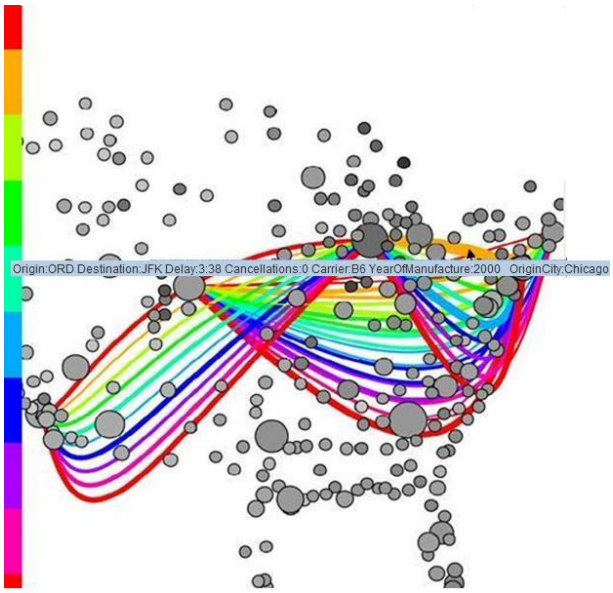


Figure 4.9: *Figure showing the different of the aircrafts flying for JetBlue airways in the top 4 airports of USA. Each edge represents the flight routes taken by different aircrafts of JetBlue airlines which are of the same age. The color of an edge represents the age and the thickness represents the delay.*

(a) Initial result (b) Result of zooming into the red portion of (a)

Figure 4.10: *Figure showing the delay caused in all the airports with size and color*

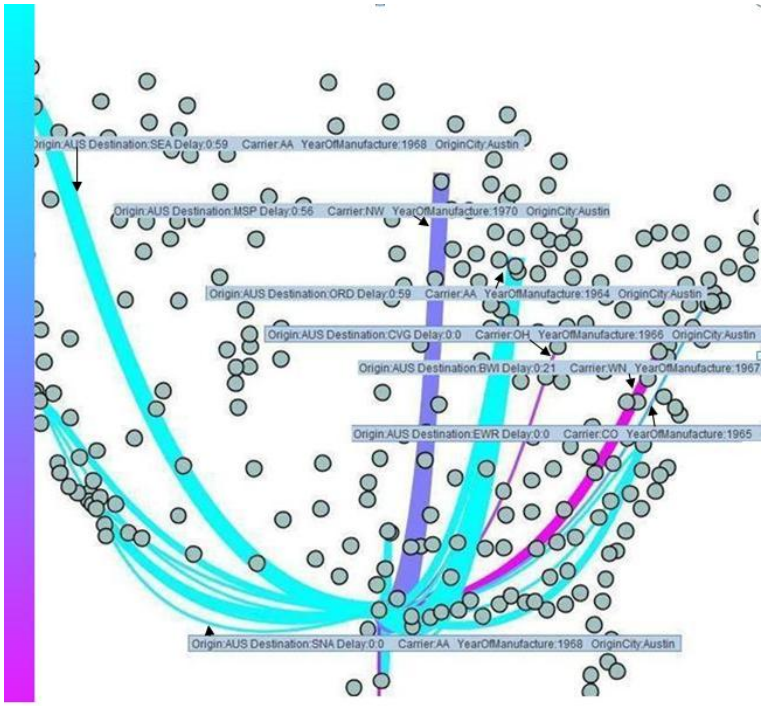


Figure 4.11: *Old and polluting aircrafts flying from Austin. Each edge represents the flight route of an old aircraft belonging to a carrier. The carrier is differentiated by the color of the edge and the thickness represents the delay.*

Chapter 5

Conclusion

The working of *AirVis* can be briefly summarized as follows. Data is accessed from the database and stored in the form of tables in Prefuse and then transformed it into visual abstractions. These visual abstractions constitute a Prefuse image. A user can interact with the tool by selection and filtering of items and their attributes. This will result a change in the data tables or the properties of the visual item. Since the Prefuse image continuously gets updated, this change is reflected in the image immediately.This is seen in the figure 5.1.

Airport data in real time is given in the form of Tables. It is not easy for the users to infer

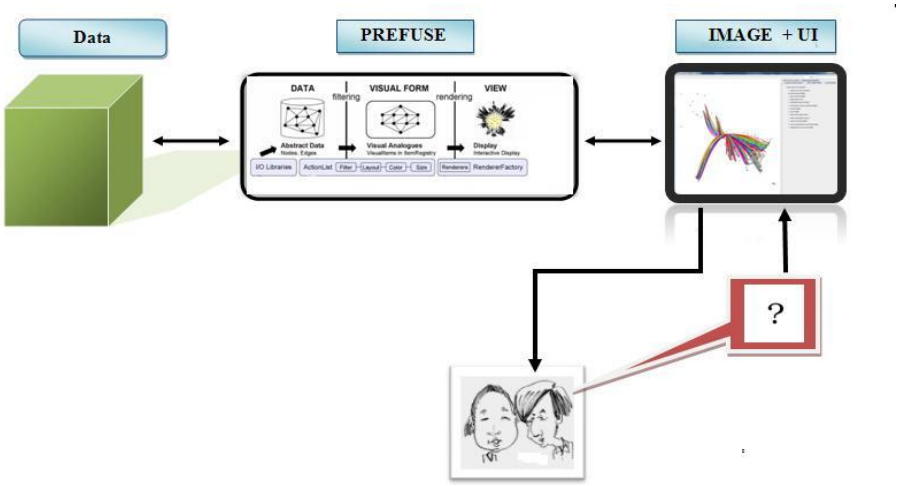


Figure 5.1: *Summary of the working of AirVis. At first, data is transformed into tables which is then mapped to visual items by Prefuse which results in a Prefuse image in the tool. The user can query this tool. Based on the selection in the UI, a query is formed which changes the tables and the attributes arising a change in the Prefuse image.*

results directly from these tables. The tool designed in this project alleviates this problem. Since all the airports are drawn based on their real locations, the figure looks like a real map. Therefore it is much easier for the users to understand the working of airports and the flights between these airports. On the other hand, this tool is also designed in a way such that it answers many of the user queries based on the given data. After getting to know the basic visual concepts used for representing this data, the user can understand the visual answers.

5.1 Future work

1. Replacing multiple edges by a single edge.
   When there are a lot of flights operating between airports, a lot of edges may have to be

added between the nodes which may result in edge crossing. However this can be reduced by substituting multiple edges between pairs of nodes by a single edge which represents the information of all the multiple edges. The edges can be divided into several sections, each representing a single edge.

2. Improving the performance of a database query in Prefuse.
   Since the data set used in this tool is quite large, executing a SQL query in Prefuse takes a lot of time. On each selection, a database query has to be performed. It is not the best way to perform the query before hand. Prefuse is in need of a new method to handle large data sets.

3. Visualizing data using small multiples.
   When a user wants to compare the flights involving two or more attributes, all the data is shown in a single visualization in the screen. Instead of this, many different visualizations can be rendered in the screen representing different values of the selected attribute in the form of small multiples (10).

4. Using node ordering to avoid overlapping of edges.
   In our tool, all the nodes are rendered in the visualization based on the latitude and longitude positions in a map. However, if the user thinks that this kind of layout is unimportant for his requirement, a mechanism for ordering the nodes in a way such that minimal edge crossings take place would help.

5. Incorporate edge bundling mechanism.
   Another way of reducing overlapping of edges is bundling edges. All the edges closely lying to each other and moving in the same direction can be grouped together to form a bundle. This mechanism can result in easier identification of edges.

Bibliography

[1] Airline on-time statistics and delay causes. http://www.transtats.bts.gov/OT_Delay/OT_DelayCause1.asp, 2003. U S Department of Transportation's (DOT) Bureau of Transportation Statistics (BTS).

[2] *The Prefuse Manual*, August 2007. http://prefuse.org/doc/manual.

[3] The prefuse visualization toolkit. http://prefuse.org/doc/api, 2007. Regents of University of California.

[4] Aircraft registration database. http://registry.faa.gov/aircraftinquiry/NNum_Inquiry.aspx, 2008. U S Department of Transportation Federal Aviation Administration.

[5] P. Bourke. Colour ramping for data visualization, July 1996.

[6] J. Heer, S. Card, and J. Landay. prefuse: A toolkit for interactive information visualization. page 421, 2005.

[7] J. Kennedy. A brief introduction to bezier curves. Mathematics Department, Santa Monica College, CA, USA.

[8] M. Owens. *A Definitive Guide to SQLite*. Apress Inc, 2006.

[9] C. Ware. *Information Visualization: Perception of Design*. San Francisco :Morgan Kaufmann, 2000.

[10] L. Wroblewski. Small multiples within a user interface. *Web Form Design*, 2005.

A-1 Appendix A

A-1.1 Full table of flight on-time data set

Name	Description
Year	The year of flight which ranges from 1987-2008.
Month	The month of flight, given in the form of 1-12.
DayOfMonth	The day of the month 1-31.
DayOfWeek	The day of week in form of 1(Monday)-7(Sunday).
DepTime	The time at which the flight departs. It is given in local time with the format hhmm. This is the time instance when the pilot releases the brakes after closing the aircraft doors.
CRSDepTime	The time at which the flight is scheduled to depart, given in local time, hhmm. CRS(Computer Reservation system) time is the departure time printed on the tickets.
ArrTime	The time at which the flight arrives at the destination. It is given in local time in the format hhmm. This is the instance at which the pilot sets the brakes in the arrival gate.
CRSArrTime	The time at which the flight is scheduled to arrive at the destination, given in local time, hhmm.This is time departure time printed on the tickets.
ActualElapsedTime	This is the time computed from the gate departure from the Origin to the gate arrival at the destination. This can be calculated as follows:ActualElapsedTime = DepTime - ArrTime in minutes
CRSElapsedTime	The time difference between the scheduled departure and scheduled arrival time can be given as follows: CRSElapsedTime = CRSDepTime - CRSArrTime in minutes
AirTime	The time computed from the moment at which the aircraft leaves the ground from the Origin until it touches the ground at the destination. It is computed in minutes.
Origin	Origin is the airport code of the airport from where the flight leaves. This is a reference to the iata attribute of the carriers data set.
Dest	Destination is the airport code of airport to where the flight is scheduled to arrive. This is a reference to the iata attribute of the carriers data set.
UniqueCarrier	A Unique carrier code is a code assigned to the aircraft based on the carrier who owned it during the flight.
FlightNum	Flight number of the flight. This is a 4-letter word assigned to every flight based on the date of flight and the carrier. This is not unique to every flight. Different flight numbers can be assigned to the same aircraft as it can fly several times in the same day
TailNum	Plane tail number is the registration number of a flight. This number uniquely identifies the aircraft. This is similar to the registration number of an automobile. This is a reference to the tail number attribute of the plane data set in section 2.1.4.

Name	Description
ArrDelay	Arrival delay is the difference between the actual arrival time and the scheduled arrival time . This can also be written as: ArrDelay = ArrTime - CRSArrTime in minutes
DepDelay	Departure delay is the difference between the scheduled departure time and the actual departure time and can be written as: DepDelay = CRSDepTime - DepTime in minutes
Distance	The distance between the origin and the destination, in miles.
TaxiIn	The time since the wheel down till arrival at the destination airport gate, in minutes.
TaxiOut	The time elapsed from the departure airport gate till the wheels are off the ground in the origin airport, in minutes.
canceled	Value which tells whether the flight is canceled or not. 1 if canceled, 0 if not.
CancellationCode	Cancelation code is a variable which gives the reason for cancelation of a flight. The cancelation code is A if the flight was canceled due to aircraft problems, i.e circumstances within the airline control, for example baggage loading, fueling, crew problems, maintenance problems or aircraft cleaning problems. If the flight was canceled due to weather problems, code is B. This can be caused by extreme meteorological conditions which prevents the operation of the flight such as tornado, blizzard or hurricane.
Diverted	A value indicating whether the aircraft was diverted to a destination other than the original scheduled destination due to reasons beyond the control of the pilot or the company. The value is 1 if the aircraft was diverted and 0 if not. When an aircraft is diverted, the arrival time of the flight will be set as 0.
CarrierDelay	Reasons for flight delay can be categorized to 5 types. One of them is the carrier delay. The delay which occurred due to circumstances within the airline's control. For example: maintainence and crew problems, fueling, aircraft cleaning, baggage loading etc. This is given in minutes.
WeatherDelay	Delay which occurs due to severe meteorological conditions such as a blizzard, hurricane, heavy rain etc fall under this category.
NASDelay	Delays which fall under the category of National Aviation system such as heavy traffic volume, air-traffic control and airport operations.
SecurityDelay	The delay which occurred due to security conditions such as voiding the terminals or concourse, malfunctioned security equipments, long queues for screening and re-boarding of aircrafts due to security breach.
LADelay	Delays caused by the late arrival of the same aircraft from it's previous flight causing the delay of this flight.

A-2 Appendix B

A-2.1 Table Containing the full set of attributes of airports:

Name	Description	Data Type
iata	airport code	Nominal
Airport	Name of the airport	Nominal
City	City in which the airport is located	Nominal
State	State in which the airport is located	Nominal
NoOfCarriers	The number of different types of carriers flying from the airport	numerical
NoOfFlights	Number of flights taking place from the airport	numerical
Delay	average delay caused in the airport	numerical
CarrierDelay	Average carrier delay caused in the airport	numerical
WeatherDelay	Average weather delay caused in the airport	numerical
SecurityDelay	Average security delay caused in the airport	numerical
NASDelay	Average NAS delay caused in the airport	numerical
LADelay	Average Late aircraft delay caused in the airport	numerical
Diverted	The number of diverted flights flown from the airport	numerical
Cancelled	The number of flights canceled in the airport	numerical

A-2.2 Table Containing the full set of attributes of Flights and Flight routes:

Name	Description	Data Type
Origin	origin airport code	Nominal
Dest	destination airport code	Nominal
Origin City	City in which the origin airport is located	Nominal
Dest City	City in which the destination airport is located	Nominal
Origin State	State in which the origin airport is located	Nominal
Dest State	State in which the destination airport is located	Nominal
Carrier	The carrier flying the flight/flight route	Nominal
Year	Year of flight/flight route	Nominal
Month	Month of flight/flight route	Nominal
DayOfWeek	day of week of flight/flight route	Nominal
Time of flight	time of flight/flight route	Nominal
Day	The day of flight/flight route	Nominal
Manufacturer	The manufacturer of the aircraft flying the flight/flight route	Nominal
TailNumber	The tail Number of the aircraft flying for the flight	Nominal
manufacture_year	The year of manufacture of the aircraft flying for the flight/flight route	Nominal
NoOfCarriers	The number of different types of carriers flying for the flight/flight route	Numerical
NoOfFlights	Number of flights in the flight route (only enabled if it is a flight route)	Numerical
Distance	the average distance covered by the the flight/flight route	Numerical
Delay	average delay caused in the flight/flight route	Numerical
CarrierDelay	Average carrier delay caused in the flight/flight route	Numerical
WeatherDelay	Average weather delay caused in the flight/flight route	Numerical
SecurityDelay	Average security delay caused in the flight/flight route	Numerical
NASDelay	Average NAS delay caused in the flight/flight route	Numerical
LADelay	Average Late aircraft delay caused in the flight/flight route	Numerical
Diverted	The number of diverted flights flown from the flight/flight route	Numerical
Cancelled	The number of flights canceled in the flight/flight route	Numerical
Cancellation code	The reason for cancellation, only gets invoked if single flight is canceled	Nominal