Master's thesis
Master's Programme in Data Science

Comparison of Interactive Visualization Techniques for Origin-Destination Data Exploration

Viivi Nissilä

April 27, 2020

Supervisor(s): Professor Giulio Jacucci

Evaluator(s): Post. Doc. Mikko Kytö

Advisor(s): PhD Student Chen He

University of Helsinki
Faculty of Science

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty Faculty of Science	Koulutusohjelma — Utbildningsprogram — Degree programme Master's Programme in Data Science

Tekijä — Författare — Author

Viivi Nissilä

Työn nimi — Arbetets titel — Title

Comparison of Interactive Visualization Techniques for Origin-Destination Data Exploration

Työn laji — Arbetets art — Level Master's thesis	Aika — Datum — Month and year April 27, 2020	Sivumäärä — Sidantal — Number of pages 43

Tiivistelmä — Referat — Abstract

Origin-Destination (OD) data is a crucial part of price estimation in the aviation industry, and an OD flight is any number of flights a passenger takes in a single journey. OD data is a complex set of data that is both flow and multidimensional type of data.

In this work, the focus is to design interactive visualization techniques to support user exploration of OD data. The thesis work aims to find which of the two menu designs suit better for OD data visualization: breadth-first or depth-first menu design. The two menus follow Schneiderman's Task by Data Taxonomy, a broader version of the Information Seeking Mantra.

The first menu design is a parallel, breadth-first menu layout. The layout shows the variables in an open layout and is closer to the original data matrix. The second menu design is a hierarchical, depth-first layout. This layout is derived from the semantics of the data and is more compact in terms of screen space.

The two menu designs are compared in an online survey study conducted with the potential end users. The results of the online survey study are inconclusive, and therefore are complemented with an expert review. Both the survey study and expert review show that the Sankey graph is a good visualization type for this work, but the interaction of the two menu designs requires further improvements. Both of the menu designs received positive and negative feedback in the expert review. For future work, a solution that combines the positives of the two designs could be considered.

ACM Computing Classification System (CCS):
Human-Centered Computing → Visualization → Empirical Studies in Visualization
Human-centered computing → Interaction design → Interaction design process and methods → Interface design prototyping

Avainsanat — Nyckelord — Keywords

Interaction, Visualization, Sankey, OD Data, Civil Aviation, Flow Data, Multidimensional Data

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1. Introduction

In the aviation industry, following the passenger demand is crucial to price estimation in order to find the optimal prices for a given departure. Origin-Destination flights (OD) are any number of flights a passenger takes in a single journey. For example, a passenger wants to travel from A (Origin) to C (Destination) the actual flights booked are A-B-C and the OD is A-C. OD data gives the airline's passenger demand information to determine the future offered prices using revenue management.

In this thesis, the focus is to design interactive visualization techniques to support user exploration of OD data. The OD data used in the thesis has not been visualized before, and the user's interest is in comparing the changes in passenger flow. This thesis work aims to clarify the question, which interactive menu design is more suitable for OD data visualization: breadth-first or depth-first. The focus of the work is on addressing challenges in data complexity and how to ease the complexity through interaction.

The OD data consists of variables such as time, passenger flow amounts and multiple location variables. The visualization focuses on the two primary data types, flow and multidimensional data, and how to effectively showcase both of these types. Relevant research of flow and multidimensional data show multiple possible options for visualizing the two types of data. The visualization technique chosen for this work is the Sankey graph, and the Sankey graph is compared and verified against other visualizations types for flow or multidimensional data.

Interaction is one option to overcome the complexity of the data. The Information-Seeking Mantra guides how to design efficient and fluent interaction. A broader version of this mantra, Task by Data Type taxonomy, gives further insights into what interaction is suitable for the specific datatype. Both of these guidelines are used to determine the best possible interactive menu designs for the OD data.

Two interactive menu designs are implemented to determine which menu design is better suited for the OD data. The first menu design is an open, breadth-first menu design that resembles the original data matrix. The second design is a hierarchical, depth-first menu that is based on the variables of the data. The two designs are compared in an online survey study conducted at Finnair to determine which interactive design supports user exploration better. The survey study consists of task questions to familiarize with the application and qualitative SUS survey questions. The online survey study results are complemented with a separate expert review.

In this work, I will examine the relevant research regarding visualization and interaction. In chapter two I present flow and multidimensional visualization techniques. I will also introduce interaction techniques, and give applicable examples regarding the two data types. In chapter three I examine the target of the visualization, the OD data. First, I will introduce the origins of the demand variable through an example of a fare family forecasting model. Second, I will examine the basic information of OD data as well as the key factors from the data that affect the design.

In chapter four I will go through the design process. Using the design process, I create two distinctive interactive menu designs. I examine these designs more thoroughly in chapter five, along with the shared Sankey graph. I compare the menu designs in a two-step study consisting of an online survey and an expert review. I present these in chapters six and seven. In chapter six, I explain the structure of the online survey study and discuss shortly the results of the survey. Besides the survey study, I will discuss the two designs with two experts in a short interview known as expert review. This expert review complements the online user study. In the end, I will discuss the results of both of these studies and the points emphasized by the previous research in the last chapter.

2. Background

In this work, interactive visualization techniques are used to present a complex set of data, the Origin-Destination (*OD*) data. The OD data is flow and multidimensional data. Thus, this chapter reviews research on visualizing these two types of data, and interaction techniques to overcome the complexity of these two types. Example visualizations are also introduced to demonstrate the applicable visualization techniques for the respective data types.

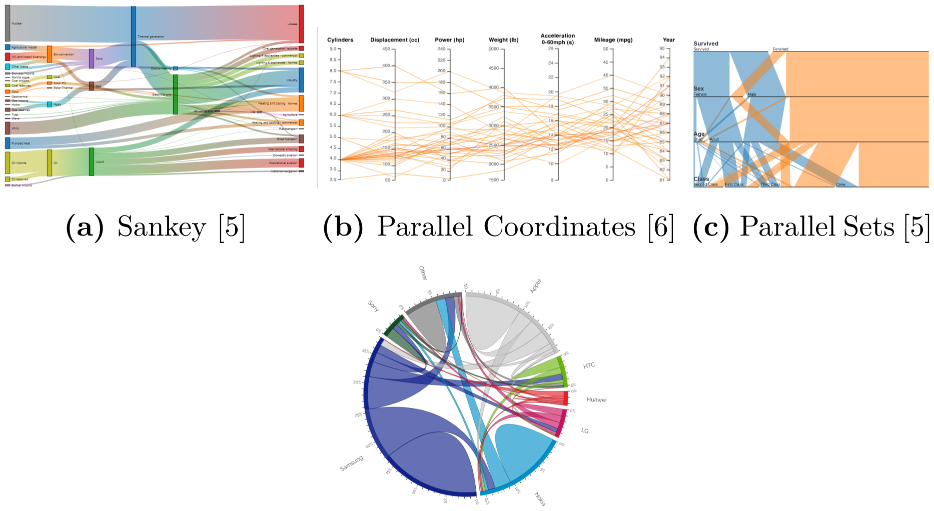
The flow data is a set of data that describes the transportation of quantities such as energy, material or passengers from a source to a target destination [1, 2]. The Sankey graph is commonly used to visualize flow data, and it shows the flow quantity as the width of the link [1, 2].

Multidimensional data is a set of data with more than three dimensions [3, 4]. The dataset can be challenging to visualize with only one diagram due to its complexity. Many visualizations support only two dimensions and visualizing all the data at once can create an overcrowded view for the user, also known as visual clutter.

2.1 Visualization Techniques by Data Type

The OD data is multidimensional data consisting of passenger flows, meaning the dataset consists of two separate data types: flow and multidimensional type. Depicting two separate types of data in one visualization requires design considerations to show both of the types respectively.

Firstly, the flow type of visualization needs to depict the direction and intensity (or amount) of flow between the source and target. Secondly, the visualization for multidimensional data requires enough data to be presented from all the possible axes. This can be done in separate visualizations or through a visualization designed to depict multiple axes.



(a) Sankey [5] **(b)** Parallel Coordinates [6] **(c)** Parallel Sets [5]

(d) Chord Diagram [7]

Figure 2.1: Example visualizations for flow data

2.1.1 Flow data

There are three variables that are important for visualizing flow type of data: source and target of the flow, and the intensity (or amount) of the flow [2]. Commonly flow is also considered directed, and the direction can also be presented visually. OD data is directed flow data: flights have directions and therefore the passenger flows have directions as well.

The flow data has multiple specific data visualization types. These types include Sankey, Flow Map [2] and Chord diagram, as well as the modified versions of these visualizations. Flow type of data can also be visualized with Parallel Sets and Parallel Coordinates, although they are originally developed to present sets and multidimensional data. Also, other types of visualizations can be considered, e.g. a matrix or a heatmap. These are originally developed to show quantities of network data or two-dimensional data and lack the visual representation of the flow that e.g. Sankey can provide [4].

Sankey graph is most commonly used to visualize flow data, such as energy or material in networks or processes [1]. It shows qualitative information about flows and flow transportations. The Sankey graph shows the relationships between flows and allows the user to compare them visually. Sankey graph is essentially an acyclic directed weighed network graph that additionally presents the intensity of the flow. The graph consists of nodes and the links that are the flows that connect these nodes. The intensity or amount of flow is presented as the width of the link. The nodes and flows can be aligned either top-to-bottom or left-to-right, and additional color coding can be added to nodes and flows.

Examples of Sankey graphs from the energy system implement the graph in two different ways [1, 2]. The first interactive Sankey graph follows the basic structure of the Sankey graph [1]. There are two implementations of this graph: a more detailed and interactive version for energy system experts and a simplified web version for the general public. The energy flows are arranged from top to bottom. Flow tracing is supported by an implementation of highlighting the selected flows. Nodes can be grouped, and the graph has an implementation of a panning tool. Similar to filters, This panning tool recalculates the flow and node connections based on the user input.

The second graph implements similar grouping operations as well as presents time-varying data [2]. The animation is used to present the changes over time by relocating the node positions. The Sankey graph also presents the distribution of energy flow emissions as three-dimensional bar charts at the end of each flow.

The two implementations of the Sankey graph for energy systems show that the visualization can convey complex information in a meaningful way [1, 2]. The visualizations received positive feedback from business domain professionals and proved to be useful for the general public as well. Especially tracing and comparing the flow transportation was considered to be useful. Although Sankey can at first seem more like a complex visualization, the research indicates that it is also successful in conveying information on a more general level [1, 2].

Besides receiving positive feedback from the end-users, the Sankey graph has other potential benefits compared to the other possible flow data visualizations listed above and in picture 2.1. The Sankey graph is not limited to one or two dimensions and shows the dimensions as vertical sets of nodes distinguished by the flows passing through the sets of nodes. This feature allows a more complex and versatile visualization than e.g the chord diagrams round shape can provide. Also, the single smaller flows with the same path from source to target are bundled together as one whole flow, which makes the interpretation of the graph easier for the user and allows the visual size comparison between the flows. The flow bundling and structure of the Sankey graph allows the user to trace the flow from beginning to end with one view.

Another alternative to Sankey is Parallel sets. The parallel sets visualization is very close to the Sankey graph because it contains the same type of components: visualization of nodes and flow between the nodes. The major difference is the flow structure, moreover what data the visualization presents. The parallel sets visualization is created to present categorical data. This presents in the graph as connections that span through each vertical set of nodes. Vertical sets of nodes present one categorical variable and nodes are the single categorical values in each variable. The connections represent the amount of specific value in the category. Compared to Parallel sets connections, Sankey graphs flows have no limitation as to which nodes the flow should reach. The parallel sets graph can be visualized with the Sankey algorithm. The only difference is the structure.

Although Sankey is a versatile graph option, there are scenarios where the Sankey graph is not a suitable choice. The Sankey graph presents an acyclic network and therefore is not suitable for networks with one or more cyclic connections. The basic Sankey layout assumes that data flows through nodes from left to right. Any exceptions to this flow, such as cyclic connections, are not acceptable in the basic Sankey creation algorithms [8]. The cyclic connections can be added with modifications to the traditional layout [9]. These types of modifications would benefit for instance product lifecycles, where a significant amount of products return to new use, i.e. create a cyclic connection back to the flows original node.

Implementing additional layers to Sankey graphs has proved to be cumbersome [1, 2]. At the time of implementing the first energy system Sankey graph (2005), complex animated interactions were found to be difficult to implement[1]. The additional 3D layer to nodes can also make visualizations less comparable to similar visualizations [2]. Adding a 3D layer to visualizations rarely adds any useful information, unless the depth perception is important [4], which is secondary for traditional Sankey graph implementations.

2.1.2 Multidimensional data

Multidimensional data is a set of data with n-dimensions (or variables) [3]. Typically most relational and statistical databases present multidimensional data. The dataset can be difficult to address with only one graph, and there is a limited set of visualization types available for multidimensional data. Parallel coordinates and modifications of scatter plots [3, 10] are visualization types that are capable of showing all variables at once.

A pairwise scatterplot shows a comparison of each pairwise combination of variables in a matrix[10]. The pairwise scatterplot is commonly used for statistical data and is mainly suitable for numerical data. The traditional pairwise scatterplot contains only two dimensions and lacks the capability of showing all the variables of multidimensional data. Adding an additional dimension allows pairwise scatterplot to show multidimensional data respectively. The benefit of using a modified pairwise scatterplot is the familiarity of the graph for the user. However, if the multidimensional data contains other variables than numerical, pairwise scatterplot requires additional interaction and other visualizations to complement it.

Parallel coordinates are efficient in displaying multidimensional data [3, 11]. The visualization shows the user all the possible variables as vertical axes and the individual rows as lines crossing these axes. The visualization can show both numerical and categorical data. Although being a versatile visualization, it has its limitations. With large datasets, perceiving the individual rows becomes difficult. A more meaningful way of looking at the graph would be bundling the individual rows together [12], which is already implemented in the Sankey graph.

Since there is only a limited amount of single visualization types available for multidimensional data, other implementations to overcome multidimensional datasets complexity are identified by previous research. The multidimensional data can be visualized with multiple separate visualizations that present different aspects of the data [4, 13]. The multiple visualizations that consist of the same type of visualization are known as *partitioned views*. The partitioned views can be aligned in two ways. The first option is list alignment, where views are grouped by a variable. For example, a grouped bar chart is list aligned. The second option is matrix alignment, and matrix aligned visualizations share the x- and y-axis across the partitioned views, much like in the pairwise scatterplots. The partitioned views also require that the data consists of the same variable types because they share the same axis.

When visualizing multidimensional data, one can also use a combination of visualization types, such as a heatmap and a bar chart. The separate different visualizations can be arranged either as *juxtapose* or *superimposed*. Juxtaposed visualizations are shown to the user side-by-side. The views can be coordinated together by e.g. using the same visual encoding, but this is not always necessary. Juxtaposed visualizations require plenty of screen space, but allow using more versatile visualizations.

Superimposed visualizations include techniques to layer visualizations together, such as adding multiple lines in a line chart or combining a smaller visualization into a larger visualization. The visualizations share the same coordinate system and the visualization layers appear to fluently mix together. The different layers can be distinguished by color encoding. Superimposed visualizations save screen space and enable comparison through proximity. Much like the partitioned views, superimposed layers share the same coordinate system and do not mix different data types together.

2.2 Interaction

Visualizing all the variables of multidimensional data at once is not relevant most of the time, because this can create visual clutter. Using interaction to limit the amount of data shown to a user can help to avoid overcrowded views [4]. Interaction can be added through any given variable or visualization of choice. These include actions such as arrangement, variable filtering, changing the viewpoint or aggregation. In other words, the user is given the ability to select variables or how to change the data (e.g aggregation).

The common way to implement interaction is to follow the Schneiderman's *Visual Information Seeking Mantra*, "Overview at first, zoom and filter, details on demand" [3]. The Information Seeking Mantra is also part of the *Task By Data Type Taxonomy* [3]. This taxonomy assumes that each data type has specific tasks where the user searches for information from the data. These tasks can then be generalized and applied as interaction to support visualizations.

Tasks for the multidimensional data include finding patterns, clusters, correlations among pairs of variables as well as gaps and outliers. To achieve these tasks, interaction is created beside the graph by allowing the user to create dynamic queries for the data. This way the user is shown only the data of interest, with a limited amount of visual clutter. For example, the 2D scatterplot can have a slider to change the data over time [3].

The general tasks that the Task by Data Type Taxonomy describe are [3]:

1. *Overview at first* – Gain an overview of the dataset at first
2. *Zoom* - Zoom into items of interest
3. *Filter* – Filter to view items of interest
4. *Details on Demand* – Select an item or group to gain more insights on them
5. *Relate* – View the relationship between items
6. *History* – Track the history of actions and allow to revert to and modify previous actions
7. *Extract* – Allow extraction of view or filtered dataset

The first four are the extracted tasks of the Information Seeking Mantra. The user is commonly first shown a *overview* of the data. For multidimensional data, this could be a summary of the variables. *Zooming* into items of interest allows the user to view items more closely. Traditionally, zooming refers to the geometric zooming of images or maps, but zooming can be done as semantic zooming as well [4]. Semantic zooming changes the viewpoint of the data by limiting and filtering the data and can be applied to data without geographical coordinates or images. Multidimensional data also requires *filtering* out uninteresting values with dynamic queries. Filtering can be applied to both variables and their values [4]. Once the data is zoomed and filtered into a smaller preferred set, the details of the items can be easily viewed (*Details on demand*).

When viewing the details of the chosen items, the user can compare similarities and differences between the items, such as the visualized relationship between the two items *Relate*. In case the previous zooming and filters need adjustments, the user is shown the *History* of previous actions as a list. The user can switch between variables or create a completely new dynamic query. When there is a need for distributing this information, the filtered dataset can be *extracted* and sent in the given format.

In an interactive visualization tool for multidimensional film data, the visualization is combined with filtering options to limit the data [4]. The multidimensional dataset is visualized as a scatterplot, where the color-coding presents the genre and axes are year and popularity. The filters include a time slider and selection for individual values. The exploration begins with an overview of the data. The user can choose and adjust the filters and the result of the filter choice is shown to the user immediately. Clicking a single point on the scatterplot reveals more detailed information of the movie, which are the remaining variables of the dataset.

The means of interaction can be applied to existing visualizations as well. The Sankey graphs for energy systems have implementations supporting the Visual Information Seeking Mantra. The visualizations provide first an overview of the energy distribution, the full Sankey graph. Zooming and details are implemented in two different ways. The first implementation supports zooming and panning with a small window to view the overall graph. The second implementation supports a change of view over time, where a time slider is used to adjust the data shown to the user. Details of the flows can be viewed from a tooltip.

2.2.1 Interactive Menu Layouts for Filtering

The task by data taxonomy for multidimensional data suggests filtering and dynamic queries to limit the data shown to the user. Implementing multiple filters for the data requires an interactive menu layout. The design considerations for interactive menu layouts consist of menu component and menu items allocation, structural cues and menu path flexibility [14].

Menu components are often presented in a hierarchical structure. The hierarchical structure is presented either as breadth or depth layout [14, 15]. The only conclusive finding between depth vs. breadth distinction is that increasing depth has been found to have more negative impacts. Response time and error rates increase when the depth level is increased. The breadth hierarchy is also associated with being more familiar to users. However, the depth hierarchy does have positive sides to it. Reducing the number of parallel menu components saves screen space and information overload, which is an important aspect when showing multiple filters for multidimensional data.

Inside the menu layout, the navigation of the menu can be eased by grouping the components or individual items together, known as chunking [14]. Positional chunking organizes components or items together based on spatial layout. Semantic chunking organizes components or items based on their meaning. In semantic chunking, the organization of the components or items is done in a way that has meaning to the user or is based on the user's logic of categorizing components or items. The chunking of menu components or items helps the user to recognize the menu structure and therefore reduces the time to navigate the menu.

The angle in which the menu components are navigated affects the time of navigation [16]. Vertical movements are faster than horizontal movements. The smaller the angle between the components, the harder it is to distinguish the items from one another. The alignment of components also affects the consumption of screen space. Components that are distinguished by angle can reduce the amount of screen space used, but are difficult to place near the screen edges. The traditional vertical components also hold more items and searching items inside them is easier.

The search for items inside the menu component is eased by providing sorting [16]. The random ordering is found to be slower than alphabetical or categorical. Alphabetical ordering is better when the user knows which item to search for, and therefore is less beneficial for novice users. Categorical or functional ordering can help to learn the menu components structure faster. The user can create memory groupings of items that reflect the visible groups, which makes the navigation faster. The effects of menu items sorting disappear when user's skills increase, and therefore the sorting only helps the novice users.

The navigation inside the menu component can be restricted to avoid possible errors. This is done by creating a tunnel of choosing, by for example steering a cursor inside a hierarchical dropdown menu [16]. If no item is chosen, the menu is closed to avoid erroneous clicking. Creating the tunnel decreases the number of errors made by the user, but it is slower and complex to use compared to non-restricted menu navigation.

The menu layout design options mentioned above are used differently by users with different skill levels [15, 16]. The user can roughly be categorized as *a novice* or *an expert*. A novice user is unfamiliar with the menu layout and commonly takes more time to scan the menu items than taking any action. In order to find items, the novice needs readable and meaningful labels for each item. While navigating, the novice often experiences disorientation. The disorientation causes the novice to return to already visited menu components and items and increases the cognitive load by having to remember component and item locations. In order to serve the purposes of the novice user, the menu layout should support guided exploration and learning.

Experts' navigation time consists mostly of motor movements or actions [15, 16]. Instead of visually searching items in general, experts often decide what item they want to select and navigate to this item instantly. Experts remember and rely on the content and locations of the menu items, which allows them to perform actions at a faster pace. The fast-paced actions are less precise movements, and therefore they can more easily miss a target. Experts also need to return to novice behavior, if the learned menu layout has changed or previously learned layout has been forgotten. It is more common that users switch between novice and expert behavior, rather than being either one of the two. To support both transition to expert and expert behavior, the menu layout should support effective navigation as well as content learning.

An example of an interactive menu layout is a parallel sets visualization tool developed to view multiple different datasets [17]. The parallel sets visualization is positioned in the center surrounded by two interactive menus. On the left side menu is the user-chosen filters, and on the right side, the user can change the data source for the visualization. The filter menu items are aligned vertically. The user drags and drops interesting variables for visualization. The categories' values (flows in Sankey graph) can be reordered since the categorical values do not have a fixed order or hierarchy. This allows the user to compare the variables more closely together if necessary. The user can highlight the categories' values that help to view all the related connections better.

The basic interaction on the parallel sets tool was shown to be easy to learn for the user. However, the more demanding dimension composition functionality and independent change of axes proved to be less intuitive. The tool was identified to require more expert behavior and knowledge from statistics, and therefore lacks the support for novice users.

The second identified issue is the difficulties in comparison when the categories have multiple different sizes or the amount of categories is huge. To solve this problem of visual clutter and inconvenience of comparison, an option was added for allowing the user to change the axis from vertical to horizontal and vice versa. In vertical mode, the nodes are aligned from top-to-bottom, whereas horizontal mode aligns nodes left-to-right. The change from vertical to horizontal resizes the categories and thus eases readability if the categories are hard to view in vertical mode. This, however, helps until a certain point and a more useful solution would be applying a more detailed semantic zooming, where the user can themself zoom into interesting categories.

3. Origin-Destination (OD) Data

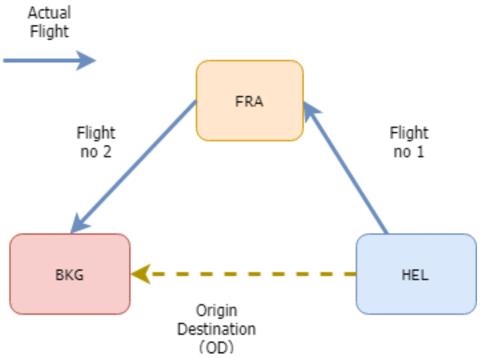


Figure 3.1: Example of what an Origin-Destination flight could be

Origin-Destination flights (OD) are any number of flights passengers take in a single journey. For example, a passenger wants to travel from Helsinki (Origin) to Bangkok (Destination) (fig 3.1). The optimal route for the passenger is with a connection through Frankfurt, and the physical flights are Helsinki-Frankfurt and Frankfurt-Bangkok. Thus, the OD flight for this passenger is Helsinki-Bangkok.

The OD flight data consists of the characteristics of the OD routes. The popularity of the OD route is estimated by the revenue management system in the form of forecasted demand. The OD data presents the current booked number of passengers and the demand for OD flights as well as the departure and arrival airports region information. In this work, the target is to present the demand and passenger amounts of the OD flights respectively.

The next section describes the origins of the demand and how the demand is estimated. Demand is part of the fare family forecasting, which aims at estimating the passenger's interest towards the fare family tickets and willingness to pay [18]. The functions describe the relation of demand to the customer choice behavior and further on, what is the role of the OD dataset in terms of demand.

3.1 Fare Family Forecasting

In order to understand the importance of the demand in the OD dataset, one must look behind the logic of forecasting demand. The demand forecast is part of the fare family forecasting in revenue management (RM) [18]. The target of fare family forecasting is to calculate the marginal revenue (or yield) to determine the lowest open fare for the fare families. Finnair uses a fare family forecasting model that applies the fare adjustment theory, a variation of the choice-based forecast model. The general fare family forecasting model created by Fiig et al. [18], is described here to gain insight from the origins of the demand data.

The fare family forecasting model is based on multiple previous airline revenue management models [18]. One of the important steps of this model is the segmentation of demand as price-oriented demand and product-oriented demand. The price demand is the estimate of passenger's willingness to pay. The product-demand (or *demand* in OD data) is estimated using traditional forecast methods that assume the demand as an independent variable. Although this is a faulty assumption and, demand is dependent on multiple factors, the transformation of demand into an independent variable helps in forecasting. This transformation is made when the demand is used to calculate marginal revenue, the final value of fare family forecasting, and therefore is not considered in the upcoming example model.

Assume we have two fare families, flex and economy [18]. The two fare families price points are: flex $f = f_1 \dots f_m$ and economy $p = p_1 \dots p_m$, where the flex family has fewer restrictions than economy. The lowest open fare in flex family is denoted as f and similarly, the lowest open fare in economy is denoted as p. The price points are discrete and in descending order with the restriction that $f > p$, and naturally $p >= p_m$ and $f >= f_m$, where p_m and f_m are the lowest price point available in each fare family. The two fare families can overlap, meaning that the lowest price points of f can be equal to the highest price points of p.

The demand for the flex fare is generally dependent on the passenger's willingness to pay (*price-oriented demand*) and their interest to either of the fare families relative to the price gap between the fare families (*product-oriented demand*) [18]. This assumption is the base for the fare family forecasting, and thus the model can be built upon customer choice behavior and the airline's fare family structure (f, p).

When the passenger is purchasing their ticket, they can make the choice of flex or economy, or no-go, as in passenger not purchasing the ticket at all [18]. This customer's behavior is modeled as a decision tree (fig 3.2), which begins with the question of purchasing the ticket, i.e. is the passenger willing to pay the cheapest open fare p or not. This is called the sell up. For the sake of clarity, assume the probability of sell up is exponential,

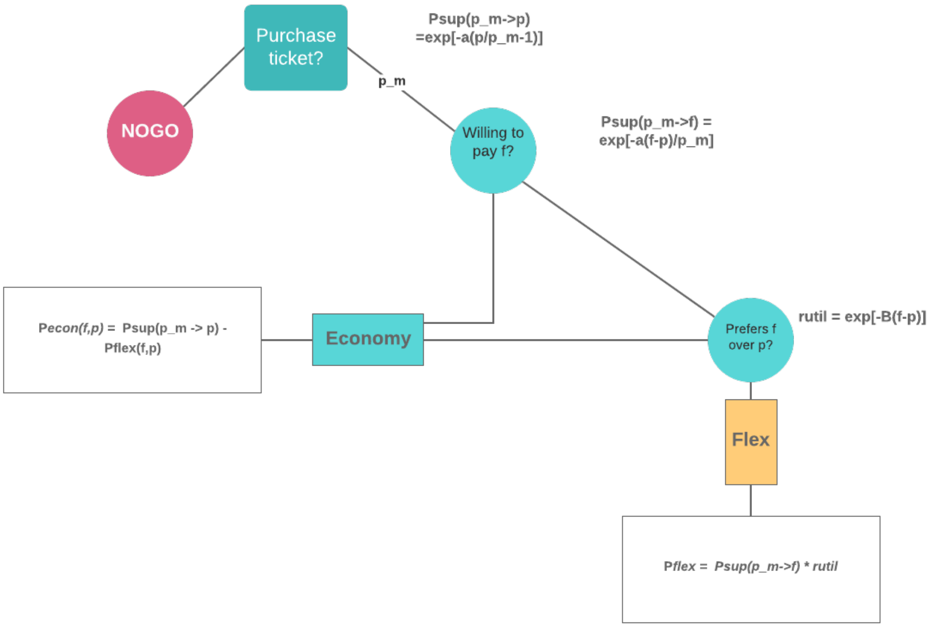

Purchase ticket?
Psup(p_m->p) =exp[-a(p/p_m-1)]
NOGO
p_m
Willing to pay f?
Psup(p_m->f) = exp[-a(f-p)/p_m]
Pecon(f,p) = Psup(p_m -> p) - Pflex(f,p)
Economy
Prefers f over p?
rutil = exp[-B(f-p)]
Flex
Pflex = Psup(p_m->f) * rutil

Figure 3.2: Decision tree of purchase behavior, original: [18]

and thus the exponential model is: $p_{sup}(p_m - > p) = \exp[a(p/p_m - 1)]$, where a is the price elasticity parameter. The sell up probability is normalized in this example, which in turn gives us a probability of no go as $1 - p_{sup}(p_m - > p)$.

The second decision is a question of the passenger's willingness to purchase sell up from p to f, the first two decisions from figure 3.2 [18]. This likelihood is composed of the probability of sell up from p to p_m and from p_m to f.

$$p_{sup}(p_m - > f) = \exp[-a(p/p_m - 1)] * \exp[-a(f - p/p_m)] = \exp[-a(f/p_m - 1)]$$

The third decision is the sell across, which indicates the probability of a customer purchasing ticket f [18]. This purchase decision is modeled with a random utility model ($rutil$). The random utility model depicts how much the customer values the difference between the two fare classes f and p, given that he can afford p. If the utility u is greater than the difference $f - p$, the customer will purchase f. For this demonstration, assume that utility is exponentially distributed. $U \sim \beta \exp(-\beta * u)$.

Based on the decision tree described above, the demand model is constructed [18]. For this demand model, it is assumed that there are two travel purposes: business (B) and leisure (L). These travel purposes comprise of different characteristics. Business purpose passengers have a high sell up and sell across probability, and tend to book tickets closer to departure, whereas leisure purpose passengers have lower sell up and sell across probability and book flights earlier.

The demand for both fare families is calculated for both travel purposes B and L.

The formula is constructed from the decision tree as:

$$dflex(f,p) = \Sigma_{psg=B,L} dflex_{psg}(f,p)$$

$$decon(f,p) = \Sigma_{psg=B,L} decon_{psg}(f,p)$$

The results from formulas above are the demands per each fare family. Therefore, the total demand is the sum of these quantities:

$$Q(f,p) = dflex(f,p) + decon(f,p)$$

The price-oriented revenue is calculated by multiplying the price policy to the demand:

$$TR(f,p) = f * dflex(f,p) + p * decon(f,p)$$

This is the first step towards calculating marginal revenue for the fare family forecasting. The modern choice-based models are comprised of a more complex set of travel purposes and fare families, and therefore require more sophisticated solutions for forecasting the marginal revenue.

3.2 Demand and the Variables of OD Data

As the family forecasting model shows, demand is an essential part of fare family forecasting. In general, demand is dependent on time and location, depicted by the rest of the variables of the OD dataset and the demand values are commonly constructed as a curve [18]. The calculated demand values produced by the revenue management system are dependent on the variables of the OD dataset, i.e. in reality, the rest of the variables are given as hyperparameters for the model. The demand is dependent on the geographical regions, time of departure (seasonality, weekends..) and point-of-sale (POS). All these variables are present in the OD dataset provided by Finnair.

The revenue management system outputs two kinds of demand measures: one calculated solely by the system and another one adjusted by analysts. Additionally, the system produces variables with different time constraints. The amount of demand changes based on the sales period: whether the whole sales period of OD flight is considered or the period from the current date to the OD flight departure date.

In addition to the previous time constraints, the demand can be set to be constrained by the aircraft's capacity. The constraint limits the booking to the maximum capacity of the aircraft, and therefore overbooking is not considered. The demand also changes when the number of booked passengers changes. The increased amount of bookings increases the amount of demand for the OD flight. Passenger bookings come in two forms: individual and group bookings. Bookings are not always fixed, and passengers can cancel their bookings. Therefore, it is also useful for the analysts to view the fixed booking amounts which are estimated based on the previous cancellation rates.

3.3 Analysis of Key Points for the Visualization

The OD data Finnair uses is a matrix that originates from the revenue management system. The total amount of variables used in this study is 31, and an additional five variables are created for the visualization. The variables can be roughly split into three categories: geographical regions, measures (demand, booked passengers), and other general variables that the measures depend on, namely departure date (weekday and month) and point-of-sale (POS). The individual rows are identified with an ID derived from the variables, and the total number of rows in this test dataset is 422'347.

The geographical region variables are categorical variables. These follow the standard IATA -coding used in the aviation industry [19] (*International Air Transportation Association*). The quantitative measures include two different types of variables: the demand variables and the booked passenger amounts. Other interesting variables presented in the visualization are the categorical variables that demand and booked passenger amounts are dependent on; point-of-sale (POS), weekday and month.

The test dataset provided by Finnair is one week's data from the future. Each row is one OD flight per cabin and country where the flights have been purchased. The interest is not in the single OD flights, but the total sum of measures, the demand or booked passengers per OD route. Aggregation is required to present the measure sums of OD routes.

The challenge of visualizing OD data lies in multidimensionality. As research suggests, presenting all the multidimensional variables at once is not meaningful and filtering the data to adjust the viewpoint is recommended in chapter 2. The use cases suggest that analysts prefer viewing single OD routes with regional filters, which is a requirement for filtering the data.

According to the Information Seeking Mantra, the visualization should first show an overview of the user. Generally, OD data has no single overview layer. The data is split into regions and an overview layer can be constructed per region and per measure. For this work, it was agreed in the beginning with the Finnair representative that the overview data would be the flows of the largest region variable, continent. Being the largest region, the continent layer has the fewest flows and therefore also the largest flow sums to depict the total quantity of measure.

There is a large variability in OD measures, and this creates a challenge for scaling the visualization. Including small and large figures into the same graph can cause distortion: the large figures take space from the smaller ones. There are techniques to overcome this difference, such as re-scaling the figures [20]. However, the interest is not in the actual measures value, but instead in the comparison of these values. The only requirement is that size comparison is visible, and the large variability itself is valuable information. Unless the variability creates other problems, such as overlapping figures, the visualization should not need adjustments for it.

The measure variables demand and booked passengers are independent. Small demand can occur although there are no booked seats and vice versa. This is because the demand is a forecast, as shown previously, and booked passenger amount is the actual number of booked seats at the time when the data has been extracted. Therefore, booked passenger amount measures are much more consistent than demand measures. This will be visible in the visualization when changing the measures.

Besides the nominal data, the categorical data has its challenges. The OD flights have in-region flights, which essentially create a circular connection. Traditional network data visualization, such as Sankey and Chord diagram, show non-circular data. In addition to a large number of variables, the amount of categorical values is also large. The dataset beholds over 200 IATA-airport codes [19], and similar to the variables, limiting the codes with filters can help to reduce the visual clutter of showing all the codes at once.

3.4 Use Cases

Before planning the implementation, a set of common use cases is defined. These use cases contain the most generic tasks performed when searching and comparing the demand or booked passenger amounts.

1. User wishes to view all OD flights from all continents
2. User wishes to view all flights from continent X to countries Y
3. User wishes to view all flights from countries Y to continent X
4. User wishes to see weekday distribution for OD route A to B

In general, the analyst wishes to see the demand or the current number of booked passengers for OD routes and visually compare them. Currently, the comparison is done in a matrix, which is the actual data produced by the revenue management system. There is no visual representation of the matrix that could show the total demand or booked passenger amounts for OD routes.

4. Pathway to The Suitable Implementation

The suitable implementation is evaluated based on the research on flow and multidimensional data and domain-specific key points. Before settling on the suitable visualization for OD data, other possible options for OD visualization are tested. The trial included four other visualization types besides Sankey: parallel coordinates, network, chord diagram and chord with additional heatmap. The visualizations are compared to Sankey in order to verify that Sankey is a suitable choice for this work. The test diagrams are static views of one day's data from the test OD dataset.

To complement the visualization, suitable interactive menu layouts are designed and tested. These designs focus on solving the problem of showing multidimensional data without overcrowded views. Filters are included in every design and the graph remains the same across all the designs.

4.1 Possible Graph Options

The previous research in chapter 2 already stated the positive features of Sankey graphs compared to other possible options. I implemented the test diagrams to have a more detailed view of the generated diagrams with this particular dataset and to confirm the positive features of the Sankey graph. While implementing the static test diagrams, it became clear that all the other diagram options lacked important features or are otherwise less suitable compared to the Sankey.

The parallel coordinates diagram has plenty of similarities compared to the Sankey. The main reason this visualization type is not considered in this work is that the level of detail it has is unnecessarily high for OD data. The parallel coordinates present every single OD route from the dataset as a drawn line and due to a large number of flights and routes in test data, each line has several overlapping lines. This results in difficulties of both finding the correct values of single routes as well as the total measure amounts. If the color scale is continent based, the crossing lines of the similar OD routes are indistinguishable. The advantages of single OD route lines are not relevant for this work, and therefore parallel coordinates would require edge bundling for this dataset, which is already implemented in Sankey.

The network graph is a standard visualization method for graph type of data, one or two-directional. It was also considered a valid option because OD data is essentially a network of flights. Compared to Sankey, network graph algorithms have multiple disadvantages. The network graph requires additional work to show the thickness of the edge bundling and direction of the connection if bi-directionality is taken into consideration. The lack of identification of multidimensional data also requires additional work: network visualization assumes that the data contains only nodes, source, and target, and connections. It is difficult to identify multiple layers in this structure, and implementing the data in a single layer can result in a structure that shows continents and countries as equal dimensions. Although the network graph has multiple options for positioning and bundling nodes, the Sankey has an existing implementation of both of these features.

The chord diagram visualizes flows like the Sankey graph, but instead in a round shape. The flows reach from one side of the sphere to another. The chord diagram manages to capture the thickness of the flow that is also implemented in the Sankey graph. The only negative about the chord diagram is its limitation on variables. Chord diagram can only fit a limited number of variables before the diagram becomes cluttered. The test I performed with a limited amount of data showed that the maximum is two variables, and for larger variables, such as the city, there is a need for detailed filters to narrow down and bundle the data into a reasonable amount of flows. This type of extensive filtering reduces the user's viewpoint significantly. These features make the chord diagram difficult to implement for large multidimensional datasets.

Previous research in chapter 2 shows that another option for visualizing multidimensional data is to split it into multiple different visualizations [4]. Therefore, the final trial is to combine a chord diagram and a heatmap as juxtaposed visualizations to increase the number of dimensions shown. Heatmap is a colored matrix, where cells change color based on their value. Heatmap is good for detailed numerical information, which is the reason it was chosen as the second visualization to complement chord diagrams' restricted dimensions. For OD data, heatmap did provide a good presentation for larger variables such as the city. The numerical values are visible next to the graph and the chord diagram shows the summary flows.

There are two reasons why this juxtaposed visualization is not considered as a good solution: 1) chord diagram still possesses issues with larger datasets, as discussed previously, and 2) heatmap expands into an unnecessarily large diagram for all the possible values. In other words, heatmap proved to be challenging specifically because of the OD datasets flows. Multiple combinations of regional OD variables posses measure values only for specific routes, i.e. not all routes are popular. The empty values cannot be eliminated if the region possesses even a one measure value, and this results in plenty of empty space in the heatmap. The size of the heatmap is large and requires scrolling and zooming so that the user can compare and view all the values. Therefore, this implementation would require restrictive filtering to avoid an overcrowded view.

Combined with the findings above, the Sankey is chosen to be the visualization option for this work. It combines the individual OD flights as flows of routes and can show multiple variables in the same visualization. At the time of writing this work, the research is not showing similar solutions for OD data that minimize the role of geographical location and emphasize the flow quantity comparison. However, solutions outside the field of aviation or transportation can still be applied, such as the interactive Sankey for energy distribution [1].

There is one feature of the data that requires adjustments: bi-directionality. Traditional Sankey algorithms do not visualize bi-directional connections and connections where departure and arrival node is the same [8]. This is resolved by transforming the data in two ways: 1) the departure and arrival nodes are split by adding a suffix to values 2) the sorting of links and nodes is turned off to create a parallel set. The reason for this is that distinguishing the same departure and arrival values avoids the cyclic loops in the data.

By doing so, the data is left with categorical connections, much like in a parallel set diagram. This does not, however, change the type of data, since the regional variables are categorical, to begin with. The reason I refer to this graph as the Sankey graph is that the aim is to visualize flows that have the source (*departure*) and target (*arrival*) nodes. Also, the visualization is implemented with the Sankey visualization algorithm. For the reasons mentioned above and for the sake of clarity, the implemented OD data diagram is called *the Sankey graph* in this work.

4.2 Implementation of Interaction for Dynamic Queries

The previous trial identified the Sankey graph as the visualization type for this work. However, the static Sankey requires that the layout has filters that generate dynamic queries to the multidimensional data respectfully. A static Sankey graph is only capable of presenting a smaller, fixed amount of simultaneous data sources (nodes), a static layer of data [2]. In order to create multiple layers with dynamic queries, interactive filtering menu designs are created by following the Information Seeking Mantra.

The interaction tasks required from the menu are extracted from the use cases and the task by data taxonomy. The interactions tasks are:

1. The user needs to be able to filter the regions to pick the interesting ones for comparison.

2. The user needs to view and see the details of the given OD flow.

3. The different flows need to be compared and viewed simultaneously.

4. The previous filters need to be visible and the user can modify and delete them.

Before designing the layouts, key components that are essential for the menu layout are identified. These key components are the ones used to filter the data, namely the variables of the dataset: regions, provider of flight (Finnair / all providers), POS, date values (Weekday, Month) and measure. These variables are included as filters that are designed as selection components on the filtering menu.

The choice of component type is based on the data type and description: regions, POS are larger categorical lists, the provider is a binary choice, and the measure and date values are both a limited categorical list. Regions and POS require a larger open selection, where individual values or multiple individual values can be chosen. The provider choice can be a smaller selection of clickable elements, like button groups or checkboxes. The measure values require a list as well. The user needs to identify them from the region selections because the measure is the comparable variable and is represented as the thickness of the flow. Weekday and month can also be represented as a list, but contain fewer items than others. Therefore all of their values can be added at the same time, e.g. with a button click.

In deciding the interaction and menu layout, multiple paper prototypes were made. The paper prototypes are simple drawn images that quickly present the structure for evaluation. These paper prototypes implement the key components as possible menu components.

After the paper prototyping phase, a few mock layouts were made with an online prototyping tool. These layouts represented the possible components and interactions of the future study and are based on the initial ideas drawn in the paper prototypes. The layouts are interactive user interface prototypes (mockups) that have one general use case implemented in them. In total, there are three different layouts, one hierarchical and two with open layout with different components to choose the suitable filters. The mock layouts show that the general layout could be usable and some work is to be put in designing them to be more fluent.

The mock layout prototyping phase identified two distinctive layout types: 1) a more simplistic, depth-first interface with a minimal amount of information showing, and 2) a breadth-first interface that shows all of the variables at once. The implemented menu layouts are structured according to this division.

5. OD Data Visualizer

Through the assessment of the design variations, two visualization tools are implemented to compare what type of interactive menu layout is more suitable for visualizing OD data. Each implementation is a web-based UI that consists of an interactive menu and the visualization. The menu design varies between the implementations. The first menu design consists of a hierarchical depth-first interactive menu structure. This structure guides the user through the application by showing gradually new menu items. The second menu design is a more open, breadth-first parallel layout that shows all possible choices and allows the user to navigate through the menu items freely.

The implementation is built as a full-stack web application. The frontend is built with React.js [21] and D3.js [5]. There are plenty of visualization libraries available, such as Plotly [22], Seaborn [23] and amCharts [24]. D3.js is a data-driven visualization library that supports modern browsers [5]. It supports the most common visualizations, as well as modifications and additional interactive features.

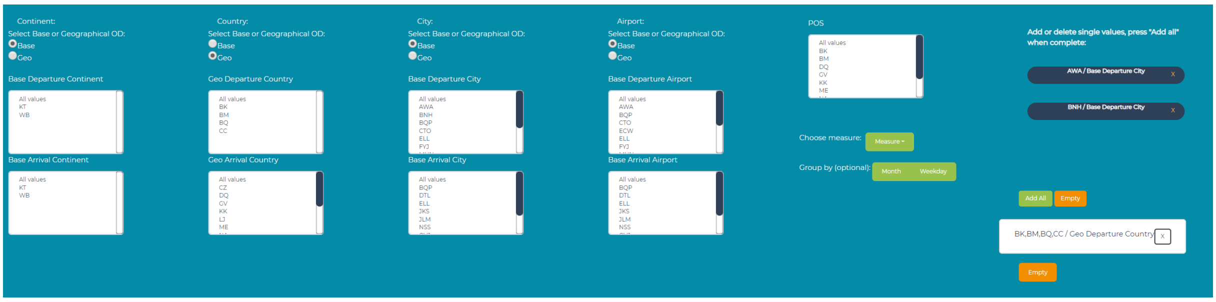
The backend is responsible for constructing the query as well as transforming the queried data. Backend receives the user input, reads the data from a database and then aggregates and transforms it into an appropriate form for visualization (*Source, Target, Value*). The backend is built with Python and utilizes the common data science libraries for data wrangling, *NumPy and Pandas* [25, 26].

5.1 User Interface

The user interface of both design variations consists of an interactive menu layout on top of the page and a dynamic visualization on the center of the page. The user is first shown an overview Sankey graph, the flow between all contents, and the chosen filters for this overview graph. The user then starts by adding more filters or generating completely different filters for a new graph. When the filters are chosen, the graph updates to match the selected variables and values.

The menu layouts filter selection is compared in this work. Since the focus is on filter selection, other sections of the layout are similar between the variations. Both menu designs have the same historical filter listing, and the Sankey graph is the same between the design layouts. In order to have a fair comparison, the style settings for the layouts are also identical.

5.1.1 Menu Design No.1 – Parallel Layout



Continent:
Select Base or Geographical OD:
● Base
○ Geo
Base Departure Continent
All values
KT
WB
Base Arrival Continent
All values
KT
WB

Country:
Select Base or Geographical OD:
○ Base
● Geo
Geo Departure Country
All values
BK
BM
BQ
CC
Geo Arrival Country
All values
CZ
DQ
GV
KK
LJ
ME

City:
Select Base or Geographical OD:
● Base
○ Geo
Base Departure City
All values
AWA
BNH
BQP
CTO
ELL
FYJ
Base Arrival City
All values
BQP
DTL
ELL
JKS
JLM
NSS

Airport:
Select Base or Geographical OD:
● Base
○ Geo
Base Departure Airport
All values
AWA
BQP
CTO
ECW
ELL
FYJ
Base Arrival Airport
All values
BQP
DTL
ELL
JKS
JLM
NSS

POS
All values
BK
BM
DQ
GV
KK
ME

Choose measure: Measure ▾

Group by (optional): Month Weekday

Add or delete single values, press "Add all" when complete:

AWA / Base Departure City X

BNH / Base Departure City X

Add All Empty

BK,BM,BQ,CC / Geo Departure Country X

Empty

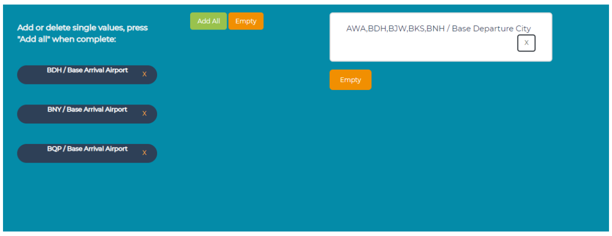
Figure 5.1: Parallel menu layout

The first graph has an open, breadth-first parallel layout. The parallel layout is based on the idea that it shows the structure of the data, a table or matrix, in a table-like format. Nothing is hidden, but instead, every variable is visible to the user. The selection boxes are ordered linearly from left-to-right in a grid, where the largest region is on the left. Each region selection is grouped vertically based on the region code: top selection is for departure values and bottom for arrival values. The provider can be changed from the top of the vertical group, which reloads the region values.

The ordering of values inside the selection components is alphabetical, except for the measures that are ordered based on type: demand first and passenger amounts second. For regional values and POS, there is no identified group of values that are the most used, and therefore alphabetical ordering is implemented. The measures are ordered by their type. The default measure is separated from the list on the top, followed by all demand variables and booking amount variables. The variable names indicate if it is the demand or booked passenger amount.

The task by data taxonomy specifies filter history, and how the user can view and modify the previously chosen filters. This is implemented in the design with two components: badges and filter listing. The user begins by choosing a single filter value from the list and continues to choose them until satisfied. The choice is not restricted, and the user can also add more values from multiple filters. The value choices are listed as badges that are shown on the right. These badges are not yet visualized. Instead, the badges allow the user to delete single values that they did not intend to choose.

The rationale behind the badge logic is that it reduces the redundant creation of dynamic queries in the middle of filter choosing. Secondly, it creates a functional buffer in case there is a lag between the frontend and backend calculation, in which case the user needs to wait for the results. The overall time to choose the filters was identified as a common issue in tools used by analysts. When the user is in the middle of choosing the filters, the dynamic query is updated. Instead of instantly rendering the visualization, the menu options are unavailable and the query appears to load for a significant amount of time with no sign of active loading, more commonly known as screen freezing. Changing the filters is more time consuming since the user has to wait between each filter change that the visualization is re-rendered. Therefore adding the badge functionality could assure the analysts that the program does not spend their time any more than what is necessary.



Add or delete single values, press "Add all" when complete:

AWA,BDH,BJW,BKS,BNH / Base Departure City

Add All
Empty
BDH / Base Arrival Airport X
BNY / Base Arrival Airport X
BQP / Base Arrival Airport X
X
Empty

Figure 5.2: List of filter choices. Badges are shown on the left and chosen filters are on the right

With a press of a button, the user confirms the badges and they are saved into the filter list. Once the user has chosen the minimum of two filters, a dynamic query is constructed and the data is filtered with the query. The resulting filtered raw data is then transformed into a suitable format for the Sankey graph: source, target and value of the flow. The filters can be deleted from the list. If after removal there are a minimum of two filters, the query is constructed with the remaining filters and the visualization is updated.

The menu no. 1 layout relies on the positive factors of the breadth-first layout based on the previous research in chapter 2: familiarity and ease-of-use. Previous research showed that the breadth-first layout is faster to navigate than depth-first [14, 15]. The content of the menu is easier to learn because it is visible during the navigation. The visibility of the filters and values can help the novice user to learn the structure easier.

The distances between each filter are short. The assumption is that the short distance encourages the user to choose multiple filters and to modify the filters afterward to build the desired dynamic query. There is a small gap between the components to ease the recognition of especially region filters types because these filters are presented in the same type of selection list components.

One clear issue with the layout is that it takes a significant amount of space. The larger layout forces the user to spend more time navigating back and forth and scrolling the page to look for the correct filters. This was also taken into consideration when designing the menu: implementing all variables of the dataset as separate components would not fit reasonably for the standard screen size of 14” without scrolling functionality. With all the variables, including the provider selection, separately shown as lists, the selection boxes total amount is 16. Therefore, a compromise was made to add radio button selection for the OD flight provider. This halved the amount of screen space required while keeping all the regional variables visible to the user.

The linear layout of the components is purely based on the structure of the dataset and is not optimized on what filters are used often. Based on the use cases and expectations towards the visualization, it is unclear which filters, other than measure, are the most often used. The amount of filters also poses a challenge on the component allocation. Most of the region filters are square lists of values and positioning them otherwise than in a grid is troublesome.

5.1.2 Menu Design No.2 – Hierarchical Layout

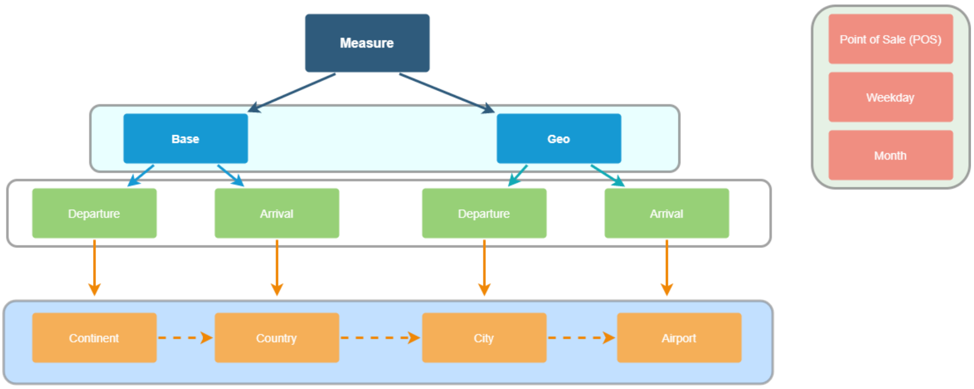


Figure 5.3: Hierarchical layout of filter options: Each box is considered as one choice. The box on the right consists of elements that are not dependent on the rest.

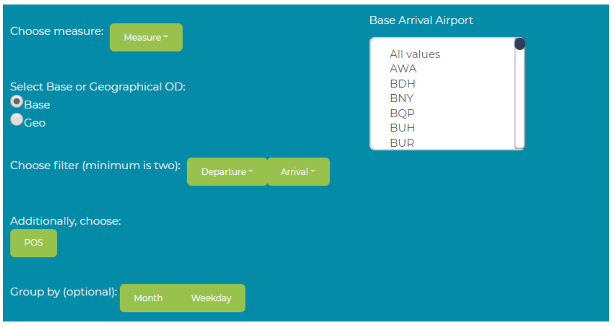
The variables of the OD dataset can be aligned in a semantic hierarchy. The visualization is for viewing and comparing the measures, and therefore the measures are considered as the parent of the hierarchy. The next choice is either the provider (lighter blue boxes) of the flight or the direction (green boxes). Either one can be aligned first, but the original variable naming has the provider option first, and therefore it is placed next in this hierarchy, followed by the direction. Both of these choices contain only two options.

After the direction choice, the region variables can be identified. The hierarchy of region variables is not strict since the user can choose any region despite the other regions. The hierarchy resides in the values themselves because continents are geographically larger and thus contain more OD routes than e.g. airports. Other variables POS, Month and

Weekday are considered separate from this hierarchy and can be chosen at any step to add additional value to the visualization.

The second graph has a depth-first menu layout that is based on the hierarchy described above. The initial idea is that the layout guides the user to choose filters according to the semantics of the data. Instead of giving all the options first and crowding the menu layout, the user chooses the variables per their type.

The hierarchical structure follows the Information Seeking Mantra [3]. The user is first given an overview of what type of information the filters contain and then the users can pick and choose what information they wish to see. Details, in this case, region codes, are presented when the descriptive values are chosen. This structure hides the original variable structure and focuses more on what type of information the variables present.



Select Base or Geographical OD:
● Base
○ Geo

Figure 5.4: Graph No.2 filter menu layout

The layout design is presented in pic 5.4. The hierarchical filter choice menu is presented on the left. The filters are aligned from top-to-bottom, and the selection list is updated on the right. Each departure and arrival dropdown menu components contain a list of regional variables. Filter listing and modification (*history*) is identical to one in menu design no. 1 described in subsection 5.1.1.

This type of menu layout is smaller and the user can view all the filter types at one glance. There is no need for the user to scroll the page to find the correct filters. The hierarchical structure creates a tunnel that guides the user to choose the correct variables. The top-to-bottom hierarchy ends close to the visualization. The user has a smaller distance to the graph after choosing all the variables, and the navigation time is reduced. Thus, the smaller layout is beneficial for smaller screen sizes as well.

The hierarchy and the positioning of components are based on the semantics of the data. Previous research has suggested that grouping components based on the user's logic can ease learning the menu layout. Also, the vertical movements between components are known to be faster than horizontal [16], and therefore the hierarchy is aligned top-to-bottom.

The depth-first layout has been proven to be slower to navigate in the previous research in chapter 2. This menu layout does require extra navigation steps for the user.

Since the levels of hierarchy increase navigation time, the hierarchy inside the components is reduced so that the visible depth of dropdown menus is only limited to two levels. This reduces the hierarchy inside the components and therefore the hierarchy is mostly on the first level as the semantic hierarchy. Thus the labels of the semantic hierarchy are constantly visible to the user while navigating the page.

5.1.3 Sankey Graph

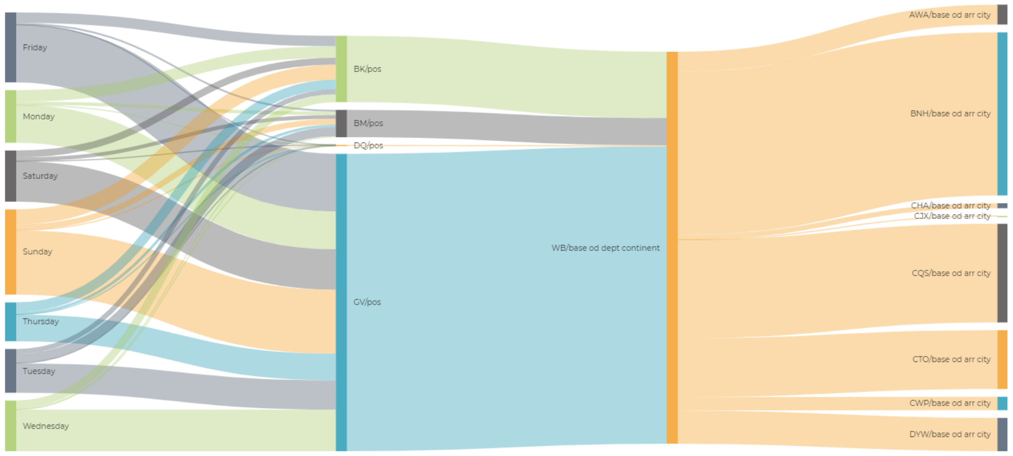


Figure 5.5: Sample Sankey graph

As described in the pathway to suitable implementation, the Sankey graph was chosen to be the visualization for this work. Figure 5.5 shows a graph generated with the application. From left-to-right, the column order is Weekday, POS, Departure Continent, and Arrival City. Nodes, other than in columns weekday and month, have the column name at the end. The flow amounts are shown in a tooltip. The tooltip indicates the exact amount of chosen measure, and source and target nodes names. When again, hovering over the node, the OD flow links from and to that node are highlighted.

The Sankey graph implements the task by data type taxonomy's task *relate*. Users can view and compare the flows to extract insights from the data. Besides adjusting the filters, the detailed view is implemented as a tooltip to avoid overlapping texts on top of nodes and flows. The user can hover the mouse over a single flow and view the exact measure and source and target node names from the appearing tooltip.

One issue that I found during implementation is that D3.js Sankey does not support dynamic graph sizes. The SVG elements frame size can be adjusted dynamically, but the links and nodes inside the element are constructed with the Sankey algorithm and are not allowed to be modified after creation on the current session. This creates a need for a change: the amount of nodes and links has to be fixed. By trial, the threshold for still clearly visible links was found to be 50 for the graph this size. Clearly visible means that after 50, the nodes and links become overlapping and equally small in thickness, which makes the graph harder to interpret.

There are two ways to overcome this issue, both of which I tested with a larger set of data. First is by sizing the graph so that all the flows would have a visible thickness. In order for the user to view the graph in this state, there needs to be a scrolling implementation. Scrolling decreases comparability between the flows since the user needs to keep in mind the shape and thickness of the flow while searching the one to compare it to. This type of comparison could be possible if the thickness of the flow differs significantly but when the change is smaller, remembering and comparing the size increases the user's cognitive load.

The second option is to limit the data by selection. If the user selects more than 50 region codes, the application gives an alert indicating that the user should change the filters. This furthermore limits the amount of data possible to show to the user, hence diminishing the number of possible views shown to the user. However, this option is more relevant. Resizing the graph on the 14" screen is not viable due to the need for extensive scrolling both horizontally and vertically. And since showing over 50 variables in one single view is not meaningful, implementing a limit is a better option for this work.

The Sankey graphs layout is the same for both interfaces. The layout options are adjusted to fit the Sankey into Parallel Sets layout with a few exceptions. The sorting of the nodes is turned off since it is not required for categorical connections in parallel sets. This creates more overlapping flows in the graph but stabilizes the structure so that the order of the node columns and flows is set by the input data [8].

There is a second reason why Sankey algorithms sorting is not used in this visualization: the semantics for the data require a specific node ordering. There are three conditions that the order must fulfill: 1) departures are on the left and arrivals on the right, 2) geographical regions are sorted from largest to smallest, starting from the edges of the graph, and 3) larger "all providers" variables are closer to edges, much like the geographical variable sorting. The sorting is in the backend implementation, so the Sankey graph receives data that is ready to be visualized.

There is no specific meaning linked to the coloring of the links and nodes. There are far more than 20 nodes and links, which is more than the number of colors in D3.js libraries' largest color schema can provide. The lack of color forces the graph algorithm to recycle the colors, and thus the colors are implemented only to distinguish the flows from one another. Coloring based on the node column is also an option, but this reduces the recognition of differences of the vertically adjacent flows.

6. Online User Study

In order to evaluate the interaction and usability for both of the menu layouts, a survey study is conducted. A survey study was chosen since it is a common practice to evaluate menu usability and it is a flexible study for the participants. Survey study does not require a specific lab setting and participants can fill it whenever they have the time for it. The survey process also allows participants to view the menu layouts and visualization on their laptops, which can provide useful information on how it is perceived in the work environment setting.

6.1 The Integrated SUS Survey

The two menu layouts are evaluated based on a qualitative online survey. The purpose of the study is to evaluate the two menu designs and how well the two designs manage to show the OD data for the end-users. The users participating in this study are analysts working for Finnair. The participants evaluate both of the layouts with the same set of questions. This allows the participants themselves to compare which of the two layouts they would use more likely.

The structure of the survey is the same for both versions of the menu design. The participants first start by introducing themselves to the menu design through task questions. The quantitative task questions guide participants into finding information from the menu and visualization, similar to what the actual use case would be. The contents or the answers to a question are not relevant for this study, and the correctness only indicates how hard it is to find individual information from the layouts. The task questions are different for both menu designs to avoid users' copying answers.

After familiarizing with the menu design, the participants are guided to answer a short set of qualitative questions. The qualitative evaluation questions are based on the system usability scale (SUS) [27]. SUS is a scalable questionnaire and can also be used for smaller sample sizes. The SUS questionnaire consists of 10 individual items rated between "Strongly Agree" and "Strongly Disagree". The answers are transformed into numerical values, where a single SUS score scale is 0-100 per each question. Although the scale is 0-100, this does not equal percentages. The SUS values can be normalized to get a percentile of the results. For this survey, eight SUS questions were chosen. These eight questions have an equal number of positive and negative claims. The qualitative questions are mixed to avoid the learning effect. There are two different measurements in this study: the accuracy of interaction and the SUS evaluation of the interaction.

6.1.1 The Pilot Study

Before the survey study, the survey answering process was tested and verified. The initial survey structure was verified with the thesis instructor, a user who is not familiar with the dataset, to ensure that the online survey process is fluent. This initial pilot study showed that there are misunderstandings of the filters are selected. The main issue was the filter modification being an unclear step, where the location of where the chosen filters reside is unclear. This is a potential challenge for the actual online survey.

After updating the application to the production environment, the application and survey answering process is tested with two users in a pilot study. The pilot study setting is the same as the online survey study. The particular two users were selected because they are familiar with the data, but have not seen this graph before. The verification process showed that answering the survey is fairly fast, with a mean time of 19 minutes. The task questions' answers have one false answer out of two, but the accuracy cannot be confirmed with only two users. This result is still inconclusive and is yet to be verified in the actual online study.

The application itself showed no major faults. The previously identified fault is in the tooltip, which is customized as opposed to using D3.js default tooltip. The hover functionality of the tooltip has an occasional lag and this could potentially show in the results.

6.1.2 The Online Survey Results

The pilot study showed that the online survey process has no significant faults and therefore the survey process is replicated as the online survey study at Finnair. The online survey received seven answers and the results from these answers are briefly discussed below.

The task questions show that finding information is relatively accurate from visualization. Menu design no. 1 (subsection 5.1.1) has only one fault answer in one of the questions. Menu design no. 2 (subsection 5.1.2) question one has 57% correct answers, while the second question has only one fault answer. This indicates that finding the information is not entirely difficult. But taking in count the amount of time the users spend on average on both menu designs, 26 minutes, it seems that navigating the page is still complicated.

The SUS scores average for parallel menu design no.1 (subsection 5.1.1) is 31.88 and for hierarchical menu design no. 2 (subsection 5.1.2) 29.38. For a total of eight questions, both scores are less than half (40 points). Below half-point is considered to be a poor result in SUS evaluation.

The negative about the SUS survey is that there is no clear way to assess the Task by Data Taxonomy's interaction and how well the users follow the interaction cues. Based on the SUS evaluation question scores, most of the participants think that the applications are unnecessarily cumbersome (SUS Score 30 for both), but the menu design no. 1 could be easily learned (SUS Score 40). The design no. 2 had more negative reviews in terms of ease of use (SUS Score 30) but is considered less complex (SUS Score 25). The results do not indicate which parts of the menu designs were considered cumbersome to use. Additional work is required to identify the exact challenges of the interaction.

The results are somewhat inconclusive. It was found that there are no participants for questions with a different order, so all of the participants answered the questions in an identical order with the parallel menu design no. 1 being first. The parallel layout received a higher score with two points, although the difference between the evaluations is little. Thus, no further conclusions can be made if either of the menu designs is better in terms of complexity and usability.

6.2 Summary

An online survey study was chosen as the study method for comparing the two menu designs. The survey starts with quantitative task questions to familiarize participants with the application. The usability is evaluated then with qualitative SUS survey questions. The pilot study showed that the survey answering is straightforward and minor inconsistencies in the app can be found.

The results of the survey are inconclusive and show that there are open questions yet to be answered. The menu designs received close to an equal SUS score and the scores are less than half each. It is unclear why both of the scores are equally low and no further evaluation of the menu designs can be made based on these scores.

The weight of the survey process is small, but the process itself could have been better. Giving users a chance to compare both menu designs is not ideal, instead of a fixed group of users who have only one graph to compare is more suitable for this situation. The survey form is also missing the open feedback option, which would have given users a chance to explain why they gave the particular score. This could help identify if the problem is in the implementation of the web application or the structure of the interactive menu.

7. Expert Review

The online survey study is limited in terms of feedback and more insights are required to answer the research questions. In order to gain further insights, a short interview of two experts is conducted. The first expert (E1) is an interaction design researcher from the University of Helsinki, the second (E2) is an analyst who uses OD data in his job at Finnair.

The focus of the expert review is in gathering feedback from general usability, interaction, and visualization, as well as future improvements concerning any aspects of the designs. The answers are categorized based on what specific aspect of the design they concern. The experts first familiarized themselves with both designs through a given task. After familiarization, feedback is gathered in a short interview which consists of open-ended feedback questions.

7.1 Filter Choice

Both experts (E1, E2) had issues finding how and when is the filter added to the selections. The filters had been left in the badge section, but the button to add them to the chosen filter list is never pressed. The parallel choosing of filters in design no. 1 was considered less intuitive (E1) because the user has to jump between different selections. The additional filter choice is also questioned (E1), and it also adds an extra step to the navigation.

Although the difficult interpretation, both experts (E1, E2) agreed that modifying the chosen filters is important. Second expert (E2) considered the modifying step between actual filters important from his own experience. A suggestion is made to add animation (E1) to show the user where the filter is moved to. Also, the possibility to interact straight with the graph (E1) is discussed.

7.2 Navigation

The experts have different views of which layout they prefer in terms of navigation. First expert (E1) indicates that navigating menu design no. 2 is more streamlined and fits the given task. The navigations hierarchy guides the user to choose the desired values. For the second expert (E2) the design no. 1 seems to be more intuitive throughout the interview. The layout shows all information at once, which reduces the time of navigating the application.

Also, it became clear that the lack of current visualization affects the results. OD data has no visualization currently, and therefore the only way to compare and view the data is through a matrix (or table) (E2). Matrix format data is high in the amount of information packed in one view but lacks the perspective of size comparison. This is also one explanation as to why design no. 2 had a lower review: the analysts are used to looking at data in a matrix format instead of a hierarchical format.

The structure of the task is also considered during the interview. The task can be more suitable for hierarchical navigation (E1). The listing of values follows the same order as the hierarchical menu. Finding the correct flow also proves to be difficult and unintuitive as the task only covers half of the final visualization (E2).

The provider choice is considered unintuitive as well. Base regions are significantly more often looked than Geo regions (E2) and therefore the analysts can naturally choose Base although the task indicates Geo. A second consideration, pointed by the previous research, is that these mixed-initiative menus with two types selections are slower to choose than single choice adaptive menus [28].

It was clear that showing all the OD variables at once would create too much information on one view. Although the menu design no. 2's usability can be learned (E2), a midway implementation from the current OD matrix to menu design no. 1 is more intuitive. The OD measures could be visible from the start (E2) and a comparison of two or more measures for the same regions could be applied. The filter menu could be hidden and exposed when changing filters (E2).

7.3 The Measures

The position and role of the measures are ambiguous in both of the menu designs. Changing the measure is considered less intuitive and left out (E1) when choosing the filters for the task. Comparing only one measure is limiting (E2). Comparing the same regions, but different measures would prove also useful.

The measures are intentionally moved at the top of the hierarchy on menu design no. 2, but in design no.1 the position of the measure is on the right where there is enough space for it. The positioning and highlighting this as the most important list could provide aid in finding it. Also, a suggestion is made that all the dropdown variables (E2) could be visible selections lists as well.

7.4 Summary

The expert evaluations provided valuable feedback on the two prototype menu designs. The general usability and real-life use cases are contradictory. The hierarchical navigation has a better flow, but the usage of the parallel layout can be grasped faster. For future reference, a design that implements both usability best practices and end-users preferences would be ideal.

8. Discussion of Results

The online survey results are inconclusive, but there are smaller findings that can be made from the results. These findings are the baseline for the discussion presented in this chapter. The expert review feedback on the usability of the menu designs is discussed here together with the online survey results findings.

I have categorized these findings according to the challenges described in the Introduction: complexity and interaction. Together with the discussion of the findings, suggestions are made for improvements that could be implemented in the future.

8.1 Complexity

From the online survey scores, it appears that complexity is persisting for both visualization tools. According to the results, interaction is not successful in transforming the data into a less complex format. The time it took to users to evaluate the menu designs, 26 minutes on average, indicates that it takes plenty of time to get to know into the layout and finding the appropriate information. However, there is little information if the user has kept the form open during other tasks than the survey, which could increase the answering time.

This type of application from the data is also new to the participants. Since they are experts of the data, but new to the menu design, they could be considered as being in the novice-to-expert transformation phase. Two distinctive findings could help to explain the poor results: 1) the visual representation does not match the participants learned expert behavior and 2) the novice-to-expert learning is not supported in the menu design, identified in chapter 2.

The cognitive model and visual perspective of the data could be different than what the menu designs and visualization shows. The participants may have imagined the data visually different than is shown here, and the threshold to understand this visual presentation is greater. The initial background information revealed that the analysts are accustomed to viewing the data as a matrix, which could help explain the poor survey results of the hierarchical menu design no. 2.

The expert review provides more insight into this though the process. The menu design no. 1 is preferred since it shows all the information at once, similar to a matrix. Since this type of menu design is closer to a matrix, it is more familiar for the analysts. This indicates that analysts can learn the menu design faster.

For future work, this type of visibility into the end-users perspective should be determined more carefully beforehand. This could mean including more end-users to design the application, and adding more visual cues that are more familiar and present during the navigation. Therefore this current state of the menu designs could be considered as the first test in the design process.

Despite the challenges of the menu design complexity, the Sankey graph, although being an unconventional visualization, is considered a valid technique to visualize the measures. The task in the expert review was considered to be inconsistent with the visualization since it only covered half of the Sankey graph. The lack of measure comparison was mentioned and this is something that could be implemented with additional juxtaposed visualizations.

Additional juxtaposed visualizations could give the user more insights into the data. However, the changes added should focus on the interaction and larger changes to the Sankey graph, such as adding an additional dimension to the Sankey, could over complicate the graph as shown in the previously mentioned example of an energy system graph [2]. Two distinctive Sankey graphs can be too large for comparison, so the size of the juxtaposed visualizations should be taken into account. It is also worth mentioning that showing the geographical location needs to be reconsidered.

8.2 Interaction

Based on the online survey results, it is left unclear whether part of the usability issues come from the small inconsistencies in the implementation. The pilot study feedback showed that the custom made tooltip had issues staying put and the website's custom made certificate had looked untrustworthy for some participants. These small details may have had an impact on the usability score. However, the issue of certificate was eliminated at the expert review and the tooltip did not receive any feedback, partially due to a limitation of one task.

The expert review results indicate that there are changes to be made to the filter selection and modification process, the badge and filter list functionalities. The choice of filter and its locations needs to be more visible. Modification of filters is important but could be integrated with the other filters. To reduce possibly slow rendering time, a button for updating visualization based on filters could be added. It should also be determined if partially merging the filtering functionality to the Sankey graph could make navigation faster and more fluent.

The navigation of both menus received contradictory feedback from the expert review, indicating that the general usability and real-life use cases do not meet. The hierarchical navigation in design no. 2 is streamlined and guides the user to choose the values, but the usage of the parallel design no. 1 can be grasped faster due to the open layout. The feedback suggests that the hierarchical layout is missing the visual cues that help the user navigate, but the usability of the parallel layout is decreased due to the size of the layout. Therefore a midway solution from the two layouts would be ideal, where the user is both guided to choose the filters and is shown the filter options openly.

The measures are not visible in the current designs and could be emphasized. Improved visual cues should be added and the current chosen measure needs to be visible outside the tooltips and filter lists. Also, the provider choice can be diminished in this scenario and instead provide a less visible option to choose geo variables.

The visualization application itself can be further expanded for future work. Animations, such as drag-and-drop filters, could ease the interaction with the menu and the visualization. The review suggests that a design that implements both usability best practices and end-users preferences would be ideal.

8.3 The Study

Overall, the results of the online survey study are poor, but there is a lesson to be learned here. Simplifying the graph layout and making sure everything non-study related functions as expected is one point. In practice, this could be making sure all components work, the lag is small and the application is easily accessible to users. Second is that to do this type of visualization tool, deeper design insights are needed. The expectations of users and the implementation of the design do not meet currently. In future work, the design should be upgraded to match their expectations.

The online survey scores have only a small impact, but together with the expert review, they provide some insights into the design differences. The interview gave more options to verify how the interaction is perceived by the participant and is the expectations of the Task by Data Taxonomy met. The interview revealed the challenges in the filter functionality, navigation and measure variables visibility that were unidentifiable from the online survey study.

9. Conclusions

This work compares two interactive menu designs for visualizing Origin-Destination data. Origin-Destination (OD) data is a crucial part of price estimation in the aviation industry, and an OD flight is any number of flights a passenger takes in a single journey. The OD data is a complex set of data that is both flow and multidimensional type of data.

The visualization type chosen for this work is the Sankey graph. In comparison with a static one days worth of data, Sankey graph proved to be the most suitable visualization in terms of showing both the flow data and multidimensional data. The comparison between other relevant visualization techniques showed that Sankey is capable of presenting multiple dimensions without excess visual clutter or the need for larger modification to the existing graph.

A static Sankey graph is only capable of showing a limited amount of data in one view. To overcome this limitation, interactive menu design is implemented. The menu consists of filter selection to create dynamic queries to filter the data. The interaction follows the Task by Data Taxonomy.

Different menu design options are tested and two designs are selected for this work: a parallel breadth-first layout and a hierarchical depth-first layout. The first menu design relies on the positives of the breadth-first layout and is closer to the underlying original OD dataset. The second layout follows the semantics of the data and is compact in terms of screen size.

The two designs are compared in an online survey study. The online survey study results are somewhat inconclusive. The survey study results prove to have little weight in terms of answering the research questions and the survey process could be improved as well. The online survey left unclear whether the menu designs are overcomplicated for the users or did the survey process had an effect on the results. In order to clarify and gain more insights, an expert review is conducted.

Both menus received positive and negative feedback from the experts. The filter listing and modification was difficult to find in both menu designs and the horizontal navigation received negative feedback in term of navigation time and consistency. Despite the difficulties, the filter navigation was considered an important part of the menu design.

The navigation of the menu designs received differing reviews. The matrix-like design of the menu design no. 1 was considered more familiar and the open layout received positive feedback. On the other hand, the flow of the menu design no. 2 was considered more streamlined and better suited for the given navigation task. Although the menu design no. 2 is more complex, to begin with, it could be learned with time.

The role of the measure variable was little in both of the designs and could be emphasized in future work. The measure is difficult to find and change of measure was not visibly indicated in the graph. Comparing two measures simultaneously is also mentioned during the interview.

In terms of visualizing complex data, the Sankey graph has proven to be a valid solution. Instead, the interaction of the menu designs is what requires further consideration. The results show that both of the graphs have potential, but there is a gap between the design and expectations of the analysts. For future work, a solution that combines the positives of the two designs could be considered. The visibility of both the filters and the chosen filters listing needs to be considered. This work can be considered as the first step in designing a valid interactive visualization tool for presenting OD data.

Bibliography

[1] P. Riehmann, M. Hanfler, and B. Froehlich. Interactive Sankey diagrams. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 233–240, Oct 2005.

[2] H. Alemasoom, F. F. Samavati, J. Brosz, and D. Layzell. Interactive Visualization of Energy System. In *2014 International Conference on Cyberworlds*, pages 229–236, Oct 2014.

[3] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343, Sept 1996.

[4] Tamara Munzner. *Visualization Analysis and Design.* A K Peters/CRC Press, New York, first edition edition, 2015.

[5] M. Bostok. D3.js. https://d3js.org/, 2019. [Online; accessed 20-June-2019].

[6] S. Ribecca. The Data Visualization Catalogue: Parallel Coordinates. https://datavizcatalogue.com/methods/parallel_coordinates.html, 2019. [Online; accessed 20-June-2019].

[7] N. Bremer. Visual Cinnamon: Using Data Storytelling with Chord. https://www.visualcinnamon.com/2014/12/using-data-storytelling-with-chord.html, 2019. [Online; accessed 20-June-2019].

[8] M. Bostok. D3 - Sankey. https://github.com/d3/d3-sankey, 2019. [Online; accessed 11-June-2019].

[9] R.C. Lupton and J.M. Allwood. Hybrid sankey diagrams: Visual analysis of multidimensional data for understanding resource use. *Resources, Conservation and Recycling*, 124:141 – 151, 2017.

[10] T. N. Dang and L. Wilkinson. Scagexplorer: Exploring scatterplots by their scagnostics. In *2014 IEEE Pacific Visualization Symposium*, pages 73–80, March 2014.

[11] Yao Zhonghua and Wu Lingda. 3d-parallel coordinates: Visualization for time varying multidimensional data. In *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 655–658, Aug 2016.

[12] G. Palmas, M. Bachynskyi, A. Oulasvirta, H. P. Seidel, and T. Weinkauf. An edge-bundling layout for interactive parallel coordinates. In *2014 IEEE Pacific Visualization Symposium*, pages 57–64, March 2014.

[13] J. Wang and K. Mueller. Visual causality analysis made practical. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 151–161, Oct 2017.

[14] A. Read, A. Tarrell, and A. Fruhling. Exploring user preference for the dashboard menu design. In *2009 42nd Hawaii International Conference on System Sciences*, pages 1–10, Jan 2009.

[15] David Ahlström, Andy Cockburn, Carl Gutwin, and Pourang Irani. Why it's quick to be square: Modelling new and existing hierarchical menu designs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1371–1380, New York, NY, USA, 2010. ACM.

[16] Krystian Samp. Designing graphical menus for novices and experts: Connecting design characteristics with design goals. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 3159–3168, New York, NY, USA, 2013. ACM.

[17] R. Kosara, F. Bendix, and H. Hauser. Parallel sets: interactive exploration and visual analysis of categorical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):558–568, July 2006.

[18] Thomas Fiig, Karl Isler, Craig Hopperstad, and Sara Olsen. Forecasting and optimization of fare families. *Journal of Revenue and Pricing Management*, 11, 05 2012.

[19] International Air Transportation Association. IATA Airline Coding Directory. https://www.iata.org/publications/store/pages/airline-coding-directory.aspx, 2019. [Online; accessed 07-December-2019].

[20] Nick Desbarats. Visualizing Wide-Variation Data. https://www.perceptualedge.com/articles/visual_business_intelligence/visualizing_wide-variation_data.pdf, 2019. [Online; accessed 03-July-2019].

[21] Facebook Inc. React JS. https://reactjs.org/, 2019. [Online; accessed 06-January-2020].

[22] Plotly. Plotly.js. https://plot.ly/graphing-libraries/, 2019. [Online; accessed 06-January-2020].

[23] Michael Waskom. seaborn. https://seaborn.pydata.org/, 2019. [Online; accessed 06-January-2020].

[24] amCharts. amCharts. https://www.amcharts.com/, 2019. [Online; accessed 06-January-2020].

[25] NumPy Developers. NumPy. https://numpy.org/, 2019. [Online; accessed 06-January-2020].

[26] Pandas project. Pandas. https://pandas.pydata.org/, 2019. [Online; accessed 06-January-2020].

[27] usability.gov. System Usability Scale (SUS). https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html, 2019. [Online; accessed 4-June-2019].

[28] K. Al-Omar and D. Rigas. A user performance evaluation of personalised menus. In *2009 Second International Conference on the Applications of Digital Information and Web Technologies*, pages 104–109, Aug 2009.