Parallel Hierarchies: Interactive Visualization of Multidimensional Hierarchical Aggregates

Dissertation

zur Erlangung des akademischen Grades Doktoringenieur (Dr.-Ing.)

vorgelegt an der
Technischen Universität Dresden
Fakultät Informatik

eingereicht von
M.Sc. Zana Vosough
geboren am 1.09.1985 in Sanandaj, Iran

Betreuender Hochschullehrer:
Prof. Dr.-Ing. habil. Rainer Groh (Technische Universität Dresden)

Gutachter:
Prof. Dr.-Ing. habil. Heidrun Schumann (Universität Rostock)

Fachreferent:
Prof. Dr. rer. pol. Susanne Strahringer (Technische Universität Dresden)

Tag der Einreichung: 16.08.2019
Tag der Verteidigung: 11.12.2019

To all people from my homeland who never had the chance
to follow their dreams ...

Acknowledgements

There are a lot of people who contributed directly or indirectly to this thesis, and here I would like to show my gratitude to everyone who helped me on this journey. First of all, I am very grateful to my advisor Prof. Rainer Groh, for his invaluable support, guidance, and his confidence in my ideas. I would also like to thank Prof. Heidrun Schumann for taking the time and agreeing to review my thesis as well as for her excellent advices and suggestions. Special thanks also go to Prof. Susanne Strahringer for her support and help shaping my ideas.

I am especially grateful to Hans-Jörg Schulz, with whom I spent much time discussing Parallel Hierarchies. Thanks for all the constructive and worthwhile discussions we had, and for acting as my research coach from the very beginning.

Within the Chair of Media Design Dresden, I have had the chance to work among friends. My greatest thanks go to Dietrich Kammer and Mandy Keck with whom I learned a lot. Thanks for your efforts in proofreading the entire thesis. I will miss all those wine evenings we had together. I would also like to thank Marius Hogröfer for helping with the implementation, and to Esther Lapczyna for advising me on design aspects of the project.

The work on this thesis was officially supported by SAP SE. At SAP Dresden the community of colleagues made life as a PhD student a great experience. I am very thankful to all of them for supporting me on each step of this research. Especially, Jochen Rode – my manager – and Stefan Hesse – my advisor – who deserve my deepest gratitude and admiration. Thanks for believing in my ideas, giving me the chance to work with several customers, and pushing this project forward within SAP. Jochen knows how to lead a team and attain a goal and working in his team was a life lesson for me. Thank you Jochen.

Last but not least, I want to thank my family and friends for their love and encouragement. My dad who was the reason I decided to pursue a PhD degree in the first place, and my mom for always reminding me to enjoy life to the fullest. My sisters for believing in me and encouraging me to go my own way. Finally, my deepest gratitude goes to Loïc Royer for not only helping me with the biological use cases and proofreading the thesis, but also for his support and love despite the 9,280 kilometers that separate us. *Merci!* This thesis could not have been written without you.

Abstract

Exploring multi-dimensional hierarchical data is a long-standing problem present in a wide range of fields such as bioinformatics, software systems, social sciences and business intelligence. While each hierarchical dimension within these data structures can be explored in isolation, critical information lies in the relationships between dimensions. Existing approaches can either simultaneously visualize multiple non-hierarchical dimensions, or only one or two hierarchical dimensions. Yet, the challenge of visualizing multi-dimensional hierarchical data remains open.

To address this problem, we developed a novel data visualization approach – Parallel Hierarchies – that we demonstrate on a real-life SAP SE product called *SAP Product Lifecycle Costing*. The starting point of the research is a thorough customer-driven requirement engineering phase including an iterative design process. To avoid restricting ourselves to a domain-specific solution, we abstract the data and tasks gathered from users, and demonstrate the approach generality by applying Parallel Hierarchies to datasets from bioinformatics and social sciences. Moreover, we report on a qualitative user study conducted in an industrial scenario with 15 experts from 9 different companies. As a result of this co-innovation experience, several SAP customers requested a product feature out of our solution. Moreover, Parallel Hierarchies integration as a standard diagram type into *SAP Analytics Cloud* platform is in progress.

This thesis further introduces different uncertainty representation methods applicable to Parallel Hierarchies and in general to flow diagrams. We also present a visual comparison taxonomy for time-series of hierarchically structured data with one or multiple dimensions. Moreover, we propose several visual solutions for comparing hierarchies employing flow diagrams. Finally, after presenting two application examples of Parallel Hierarchies on industrial datasets, we detail two validation methods to examine the effectiveness of the visualization solution. Particularly, we introduce a novel design validation table to assess the perceptual aspects of eight different visualization solutions including Parallel Hierarchies.

Zusammenfassung

Multidimensionale, hierarchische Daten müssen seit langem in vielen verschiedenen Bereichen untersucht werden, wie z.B. Bioinformatik, Softwaresysteme, Sozialwissenschaften und Business Intelligence. Während jede hierarchische Dimension innerhalb dieser Daten isoliert betrachtet werden kann, liegen die entscheidenden Informationen in den gegenseitigen Beziehungen. Bestehende Ansätze stellen entweder gleichzeitig mehrere Dimensionen ohne Hierarchie oder höchstens zwei hierarchische Dimensionen dar. Folglich ist die Visualisierung multidimensionaler, hierarchischer Daten noch immer eine erhebliche Herausforderung.

Die Lösung für diese Herausforderung bildet ein neuer Datenvisualisierungsansatz – Parallel Hierarchies, der am Beispiel eines echten SAP SE-Produkts (SAP Product Lifecycle Costing) entwickelt wird. Ausgangspunkt der Forschung ist eine umfassende, kundenorientierte Anforderungsanalyse mit einem integriertem, iterativen Designprozess. Durch die Abstraktion der erhobenen Daten und Nutzeraufgaben wird eine allgemeine Lösung erarbeitet. Die Anwendung des Visualisierungsansatzes der Parallel Hierarchies auf bioinformatische und sozialwissenschaftliche Datensätze bestätigt die Generalisierbarkeit. 15 Experten aus neun verschiedenen Unternehmen nahmen an einer qualitativen Nutzerstudie teil und wünschten sich im Ergebnis, ein Tool basierend auf dem vorgestellten Lösungsansatz einzusetzen. Infolgedessen wurde bereits die Integration der Parallel Hierarchies als Standard-Diagrammtyp in die SAP Analytics Cloud-Plattform initiiert.

Darüber hinaus werden in dieser Dissertation verschiedene Darstellungsmethoden für unsichere Datensätze gezeigt, die auf Parallel Hierarchies und auf flussbasierte Diagramme im Allgemeinen anwendbar sind. Weiterhin wird eine visuelle Vergleichstaxonomie für zeitabhängige, hierarchisch strukturierte Daten mit beliebig vielen Dimensionen vorgeschlagen. Auf dieser Grundlage werden mehrere Visualisierungslösungen für den Vergleich von Hierarchien mit Hilfe von flussbasierten Diagrammen entwickelt. Nach der Vorstellung von zwei Anwendungsbeispielen der Parallel Hierarchies basierend auf Industrien Datensätzen, werden zwei Validierungsmethoden beschrieben, um die Effektivität der Visualisierungslösung zu überprüfen. Dafür wird eine neuartige Validierungsdesigntabelle präsentiert, welche die Wahrnehmungsaspekte von acht verschiedenen Visualisierungslösungen einschließlich Parallel Hierarchies untersucht.

Publications

This thesis is partially based on the following publications:

Visualization Approaches for Understanding Uncertainty in Flow Diagrams
Zana Vosough, Dietrich Kammer, Mandy Keck, and Rainer Groh
Journal of Computer Languages, Elsevier 2019.

Parallel Hierarchies: A Visualization for Cross-Tabulating Hierarchical Categories
Zana Vosough, Marius Hogräfer, Loic A. Royer, Rainer Groh, Hans-Jörg Schulz
Journal of Computers & Graphics, Elsevier 2018.

Using Parallel Sets for Visualizing Results of Machine Learning Based Plausibility Checks in Product Costing
Zana Vosough, Volodymyr Vasyutynskyy
In Visual Interfaces for Big Data Environments in Industrial Applications, Workshop at AVI, ACM 2018.

Visualizing the Results of Product Costing Plausibility Checks with Parallel Hierarchies
Zana Vosough
VIS IEEE 2018: Poster.

Mirroring Sankey Diagrams for Visual Comparison Tasks
Zana Vosough, Dietrich Kammer, Mandy Keck, and Rainer Groh
In 9th International Conference on Information Visualization Theory and Applications (IVAPP) 2018.

Visualizing Uncertainty in Flow Diagrams: A Case Study in Product Costing
Zana Vosough, Dietrich Kammer, Mandy Keck, and Rainer Groh
In Proceedings of the 10th International Symposium on Visual Information Communication and Interaction (VINCI), ACM 2017.

On Establishing Visualization Requirements: A Case Study in Product Costing
Zana Vosough, Rainer Groh, and Hans-Jörg Schulz
In EurographicsConference on Visualization (EuroVis) 2017.

Having Fun with Customers: Lessons Learned From an Agile Development of a Business Software
Zana Vosough, Matthias Walther, Jochen Rode, Stefan Hesse, and Rainer Groh
In Stakeholder Involvement in Agile Development, Workshop at NordiCHI, ACM 2016.

Contents

Chapter 1

Introduction

With the rapid increase in the size and complexity of data, there is a growing need for advanced information visualization techniques outfitted with sophisticated interaction. In many organizations, information is collected and used to support analysis and decision-making. The visual representation of data is a broad application field with a long tradition. In the last few decades, the field of data visualization has produced many novel techniques to handle large amounts of data, reduce the likelihood of errors during data analysis, support comprehension, and foster insights. However, very little is known about applying advanced visualization techniques to business intelligence. Tables and simple graphs are often used to report quantitative business information [Few, 2004]. Table-based applications are typically used for business intelligence applications, while graphs are mainly used for data analytics purposes. However, important insights often lie in the data's hidden relationships, and representing these relationships with tables is cumbersome and often impossible. When spreadsheets emerged in the early 1960s, they were mostly used for small applications. Surprisingly, nowadays, spreadsheets are still widely used by large organizations to explore and analyze most of their data. In contrast, there is ample literature proving that data analysis and decision making supported by spreadsheets is error prone [Caulkins et al., 2007]. Despite the obvious necessity for better visualization principles and tools, there is only limited approaches available to help facilitate decision making. Increasingly, organizations now use the dashboard metaphor, showing data with simple pie or bar charts. While an improvement over spreadsheets, dashboards still cannot fully address complex relationships and interdependencies.

In summary, the field of business intelligence offers new visualization challenges that are opportunities to develop novel fundamental visualization principles and techniques. A frequent question is: “How can we reveal structures and patterns that cannot be seen with standard spreadsheets or dashboards while simultaneously preserving details?”

1.1 Motivation and Problem Statement

Large numbers have become part of our daily news cycle: billions of dollars lost, millions of cars affected, thousands of workers protesting, hundreds of flights canceled, etc. Yet, the details behind these numbers are rarely revealed. For example, how are these large numbers divided across financial markets, car manufacturers, income groups, or airlines. And even after splitting these numbers up, the results still remain large aggregates that can be further broken down into companies and products, models and engine types, job sectors and occupations, flight operators and destinations, and so forth. Layer by layer, we can drill down into these numbers along various categories. However, the most interesting facts behind these numbers become apparent when connecting these decompositions – e.g., breaking down the number of affected cars by manufacturer and by country, to answer questions like: *In which country drive most of the affected Fords?* and *Which car manufacturer is most affected in France?* If we could interactively decompose such aggregates along various interlinked hierarchical categories, we would be able to gain much more nuanced insights than by just looking at the aggregate as a whole. In particular, this decomposition task plays an essential role in business data analysis. As fundamental as this scenario seems, there is surprisingly little visualization support for hierarchical decompositions that span multiple categories. Multi-dimensional data techniques have been researched extensively. Categorical data appear in data tables both in scientific and business domains. However, existing visualizations are either specialized to break down aggregates along multiple, but flat categories (e.g., Mosaic Matrices, Parallel Sets) or are specialized to break down aggregates along a singular, but hierarchical property (e.g., Treemaps, Icicle Plots). In an effort to close the gap for multiple hierarchical categories, this thesis proposes a visualization technique that combines the best of both worlds.

Moreover, several application domains often handle datasets that contain uncertain values. According to Winston Churchill, “true genius resides in the capacity for evaluation of uncertain, hazardous, and conflicting information”. In some fields such as business intelligence, one key to success is accurate decision making, which itself depends on data quality and uncertainty. Users must assess not only the information presented to them, but also the confidence in that information. This motivates the need for novel ways to visualize uncertainty in data. Importantly, this uncertainty information must be well integrated with the rest of the data to prevent discounting of that uncertainty. Unfortunately, despite the obvious necessity of understanding uncertainty in data, explicit representations of uncertainty are often missing in visualization solutions. This is partially due to the complexity of both uncertainty as a concept, and visual clutter occurring when incorporating uncertainty.

As Pettersson states, the main goal in information design is *clarity of communication* [Pettersson, 2010]. In order to fulfill this goal, information visualizations must be well designed, be correctly interpreted, and understood by end users. The field of information visualization draws on ideas from several disciplines: computer science, psychology, semiotics, graphic design, cartography, and art. However, visualization research often focuses on concepts from computer graphics and human-computer interaction. Often, visualization research neglects the aesthetic and cognitive aspects of visual design. More fruitful outcomes can be reached when considering jointly the limitations and opportunities afforded by both human perception and cognition, as well as the limitations and opportunities of machine computation and display. In particular, when the solution is designed for big organizations – for example in the context of business intelligence applications – validating the effectiveness of a visual design is crucial. The next section introduces three open challenges tackled in this thesis.

1.2 Research Goals

This thesis tackles the following three main challenges relevant to the visualization and validation of multi-dimensional hierarchical data:

Challenge 1: Visualizing hierarchical multi-dimensional aggregates.
Motivation Breaking down large aggregates across multiple categories is a typical data analysis task required when investigating different distributions and their relationships. Common visualization techniques are Mosaic Plots or Parallel Sets. What is rarely considered is that most categorizations are actually hierarchical in nature and are better explored as such. Yet, exploration of categorical aggregates by their multiple hierarchical properties remains an open problem. Another fundamental challenge is developing proper interaction techniques to keep the balance between: On the one hand, the necessity for abstraction, and, on the other hand, the preservation of details.

Question How to visualize large hierarchical multi-dimensional aggregates? In particular, how to convey the relationships and patterns within and between multiple dimensions of hierarchical data without loss of information?

Challenge 2: Introducing an approach to visualize both data and its uncertainty in flow diagrams.
Motivation While uncertainty can be visualized separately from the rest of the data, in some domains such as business intelligence it needs to be addressed jointly with the data. Although this might complicate visual interpretation, it also reduces the risk for uncertainty information to be ignored. In the past, numerous visualization techniques have been proposed to show uncertainty for single values. However, designing a comprehensive view that combines a representation of data structure and uncertainty without overwhelming the user with visual clutter is still a challenging task. Moreover, many available solutions for representing uncertainty are domain-specific since the visualization process is unique from task to task.

Question How to visualize both hierarchical multi-dimensional data and its uncertainty in an integrated manner? How to extend flow diagrams in order to directly incorporate uncertainty information?

Challenge 3: Validating the perceptual effectiveness of visualization techniques.
Motivation The main purpose of information visualization is to communicate information clearly and effectively through graphical representations. In order to provide insights and intuitive ways to perceive complex data, both the aesthetic and functional aspects of data visualizations need to be considered. Colin Ware defines the fourth and last stage of the data visualization process – after collecting, transforming, and displaying – as perception by the human cognitive system [Ware, 2013, p. 5]. Often, the first three stages of visualization are covered in visualization projects, but the final stage – how humans perceive the image – is neglected. In an effort to fill this gap, a systematic method to validate visualization solutions based on well-known perceptual principles is needed. Therefore, it is important to understand how current solutions follow visual perception principles to effectively design future visualizations.

Question How to assess the value of current perceptual principles for the purpose of evaluating a visualization? What are the implications for further research on visual design evaluation?

1.3 Outline and Contributions

The organization of this thesis and the dependencies between chapters are outlined in Figure 1.1. After the introduction, chapter 2 reviews the necessary foundations in data visualization, and then in chapter 3, we review literature relevant to this work. The next five chapters provide answers to the three open problems described above. First, chapter 4 describes the process of identifying visualization requirements. The results of requirements elicitation from co-innovation customers helps us to characterize the research's domain situation and find the three solutions shown in the next three chapters. Chapter 5 introduces a novel approach for visual decomposition of categorical aggregates. In chapter 6, we extend this approach to represent data uncertainty in a holistic manner. Chapter 7 reports contributions in visual comparison tasks. Chapter 8 describes two application examples of our solution to industrial datasets. Chapter 9 presents our visualization validation method and applies it to our novel approach in general, as well as to several domain-specific applications. Finally, a short summary of the discussion and worthwhile directions for future work are given in Chapter 10.

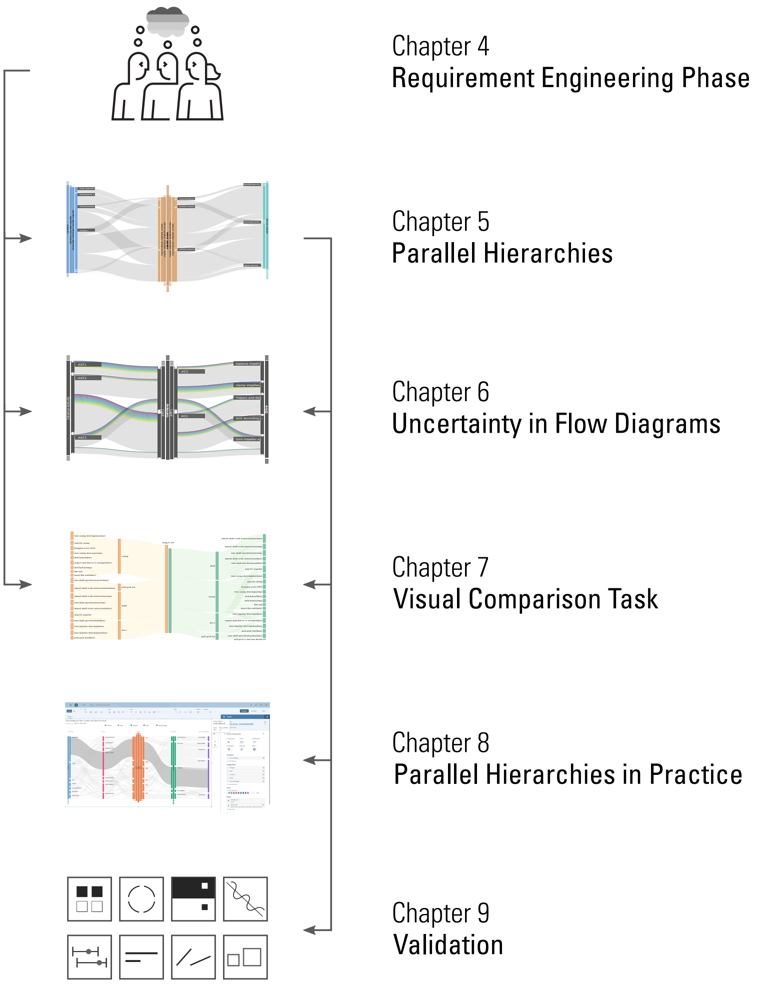


Figure 1.1: Dependencies between thesis chapters.

The results of this research project have been published in several journals and conferences [Vosough et al., 2017a, Vosough et al., 2017b, Vosough et al., 2018a, Vosough et al., 2018b, Vosough et al., 2019], and the source code is accessible to the community on GitHub: https://parallelhierarchies.github.io/.

Chapter 2

Foundations of Visualization

This chapter summarizes and reviews the necessary foundations of this thesis from related areas to information visualization. In a first part, we give a detailed description of the terms associated with information visualization. We give an overview of the various data models, task categories, visual interactions, and visualization techniques that are involved in interactive information visualizations. In a second part, we review the human visual perception and it's importance in the design and evaluation of data visualization tools. Finally, we introduce the term *flow diagrams* as the main visualization technique used in the context of this work and provide application examples.

2.1 Information Visualization

To have a common understanding of the terms used in the context of this work, we first describe terms and concepts associated with information visualization. Then we will review the three main components of each visualization system that are designed to answer three main questions [Aigner et al., 2011, p. 4]: The question of "what has to be presented?" refers to the specification of the data that user sees and will be explained in section 2.1.2. The question of "why does it have to be presented?" refers to the specification of the task that user intends to achieve with the visualization tool and will be described in section 2.1.3. The question of "how to visualize the data considering the intended task?" refers to all selected visual design and interactions to construct the visualization tool and it will be discussed in detail in chapter 2.1.4 and 2.1.5.

2.1.1 Terms and Definition

There are several definitions for *visualization*. One of the most commonly used definition by the community was introduced by Card et al. in 1997. They defined the notion of visualization as: "The use of computer-supported, interactive, visual representations of data to amplify cognition" [Card et al., 1999, p. 6]. The purpose of visualization is to get insights to make discovery, help with decision-making process and explanation. Importantly, Card et al. distinguished scientific visualization from information visualization. The term *scientific visualization* is used in case of visualizing physical data such as human body, the earth or molecules, and *information visualization* is defined as: "The use of computer-supported, interactive, visual representations of *abstract* data to amplify cognition" [Card et al., 1999, p. 7]. One important term in this definition is "abstract data", which is related to the fact that no explicit physical or

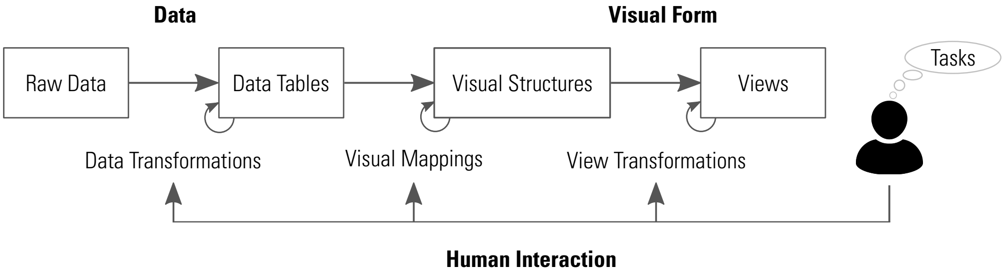


Figure 2.1: Reference Model for Visualization [Card et al., 1999, p. 17].

spatial mappings can be assigned to the data such as business information or financial data. The main goal of visualization is to transform abstract data into an appropriate visual representation. To get an appropriate visual representation, an adequate mapping of data into visual form is needed. Card et al. introduced a process for this mapping that is depicted in Figure 2.1, known as reference model for visualization. It describes a series of transformations from raw data to several views. The human is involved in the interaction process, influencing each transformation step. The series of data transformations begins with raw data, that is usually available in an unstructured format and can be transformed into sets of relations that are structured and called data tables. The data transformation phase involves the loss or gain of information since errors or missing values are removed and statistical calculations might add additional information. The core of this reference model are the visual mappings that map the data tables to visual structures. In contrast to scientific visualization, in information visualization the abstract data space without spatial reference should be mapped to a meaningful visual structure that supports the interpretation of data. Bertin's work from 1983 is one of the fundamental efforts to map data tables to visual structures [Bertin, 1983] (see section 2.2). Data tables can often be mapped in several ways into visual structures, but an effective mapping must preserve the important information and should be readily perceived. Two factors are important when evaluating the effectiveness of a mapping: task completion time (interpretation speed), and task completion error (number of human mistakes) [Card et al., 1999, p. 23]. Finally, the view transformations create views by modifying the graphical parameters like position, scaling, and clipping. In addition, human interaction completes the loop between human and visual forms by changing visualization parameters that can lead to new visual structures.

The analysis of large amounts of data is an important information visualization task needed to recognize patterns, trends, and correlations. Another well-established research field in information visualization is the interdisciplinary field of visual analytics. One of the earliest definition of *visual analytics* was proposed by Cook and Thomas as: “Visual analytics is the science of analytical reasoning facilitated by interactive visual interfaces” [Cook and Thomas, 2005, p. 4]. Yet, visual analytics is more about combining automated analysis techniques with interactive information visualizations in order to aid analysis and decision-making processes in dealing with very large and complex amounts of data. Later, Keim et al. proposed a more precise definition: “Visual analytics combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning, and decision making on the basis of very large and complex datasets” [Ellis and Mansmann, 2010, p. 10]. This definition emphasizes

the interdisciplinary nature of visual analytics that combines analysis techniques and interactive information visualizations with a focus on large amounts of data.
In this thesis, we rely on the definition of information visualization as defined by Card et al. [Card et al., 1999]. The focus of this work is on how information is visualized, used, and perceived interactively. We target large complex datasets and in chapter 8 automated data processing from visual analytics area is used to manage large amounts of data that need to be processed by decision makers. However, we do not cover the whole visual analytics process in this work.

2.1.2 What: Data Structures

In order to transform data into an efficient visual presentation, a comprehensive understanding of the underlying data is essential. Moreover, visualization techniques are often described based on the data types they represent and different data types are often tied to specific tasks. There are several classifications used to categorize data types. In this section we cover three main methods to classify data:

A) The first categorization is based on *data types*. Card and Mackinlay established one of the first data classifications and introduced three data types [Card et al., 1999, p. 20]:

- Nominal: an unordered set without any quantitative semantics such as gender or language.
- Ordinal: data values that can be ordered by relationship between them such as shirt sizes or letter grades.
- Quantitative: a numeric range such as height or width.

Ward et al. classified ordinal data attributes further to binary, discrete, or continuous data. In their classification, both discrete and continuous data types might have numerical values [Ward et al., 2010, p. 46]. Later Munzner proposed another version of Ward et al. data attribute classification. She differentiates between *categorical* and *ordered* data [Munzner, 2014, p. 32], and then the ordered data can be further subdivided into *ordinal* and *quantitative* data. In addition, she introduces three possible directions between ordered data: *sequential*, *diverging* or *cycling*.

B) Another important aspect of the data that needs to be considered is the availability of dataset types. Data can be *static*, which means the entire dataset is available at once, or the dataset can be *dynamic*, where the dataset is completed over the time [Munzner, 2014, p. 31].

C) Finally, it's also possible to classify data by *dimensions*. This is one of the most common ways to identify tasks and problems the user is trying to solve. Shneiderman has divided data types into the following groups [Shneiderman, 1996]:

- 1-dimensional data: linear data types consisting of text documents, program source code, and lists, which are ordered in a sequential manner.
- 2-dimensional data: spatial data such as newspaper layouts or geographical data that is projected onto maps or floor-plans.

- 3-dimensional data: real world objects with complex three-dimensional relationships such as the human body, molecules, and buildings.
- Temporal data: time-dependent data that differs from 1-dimensional data in having a start and end time and elements that may overlap, such as medical records or project management data.
- Multi-dimensional data: data in relational and statistical databases with *n* attributes that can be assigned to n-dimensional space.
- Tree data: hierarchical structured data in which all items are arranged in a parent-child relationship and all items and links can have multiple attributes, such as directories in a digital file system.
- Network data: items are linked to an arbitrary number of other items such as co-authorships relationship between scientists.

The goal of this classification is not to cover all data types, but rather to reflect an abstraction of reality. Many prototypes might use combinations of these groups. Facilitating discoveries and discussion are the main goals of Shneiderman's taxonomy [Shneiderman, 1996]. In 2003, Keim et al. introduced another taxonomy built on top of Shneiderman's classification, which differs mainly in the dimensions of the data variables. They considered the number of variables as dimensionality of the dataset. In their classification, datasets may be one-dimensional, two-dimensional, multi-dimensional, or may have more complex data types such as text & hypertext, hierarchies & graphs, or algorithm & software [Keim, 2005]. For instance they place temporal data either in the one-dimensional or in the multi-dimensional category depending on the number of variables assigned to each point in time. Also, they do not differentiate between three and multi-dimensional data, as visualizing the third dimension on two-dimensional screen is straightforward. In this thesis, we consider datasets with three or more dimensions as multi-dimensional data, and the focus of this work is on *categorical*, *static*, *multi-dimensional*, and *tree* data.

2.1.3 Why: Visualization Tasks

A visualization tool that is efficient for one task might be unfit for another. Therefore, understanding user tasks is one of the fundamental steps before designing or for evaluating an information visualization. One of the main reasons why a visual design might be considered ineffective is because it does not match the intended task. It is best practice to break down complex tasks and consider one goal of the user at a time in order to better define user tasks. Then, the output of one task can be the input of the next one. In this section, we review different ways of defining and classifying visualization tasks. The visualization tasks are classified and defined at different levels of granularity and sometimes different terms are used to describe similar tasks.

The definition of visualization tasks is often confused by the research community with concrete interactions in the visualization. Visual interactions help to solve various visualization tasks, but do not define them. In the following, we will introduce some visual task taxonomies that are defined based on visual interactions to determine which interaction techniques or combination of techniques will best serve a given set of user tasks. For example, the visualization task classifications by Shneiderman and Buja

Task	Description
Overview	Gain an overview of the entire collection of data.
Zoom	Zoom in on items of interest.
Filter	Filter out uninteresting items.
Details-on-demand	Select an item or group and get details when needed.
Relate	View relationships among items.
History	Keep a history of actions to support undo, replay, and refinement.
Extract	Allow extraction of sub-collections and of the query parameters.

Table 2.1: Task Taxonomy by Shneiderman [Shneiderman, 1996].

et al. from 1996 are based on different interaction types [Shneiderman, 1996, Buja et al., 1996]. Shneiderman's *Task by Data Type Taxonomy* is one of the fundamental classifications of visualization tasks. He assumes that users are dealing with collections of items, and items have multiple attributes. Accordingly, all items from the seven data types described before in section 2.1.2 have attributes. Finding all items with attribute values matching a set of values, is considered in this classification as a basic search task. This taxonomy is built on top of Shneiderman's Visual Information Seeking Mantra by adding the tasks *relate*, *history*, and *extract*. Table 2.1 shows the seven tasks from the task by data type taxonomy of Shneiderman [Shneiderman, 1996]. The seven tasks in Shneiderman classification are low-level tasks that can be abstracted to the high-level tasks of *exploration* and *search*.

Buja et al. visual task classification is based on a taxonomy of interaction with visualizations. They considered three classes of interactions: *focusing*, *linking*, and *arranging views*. Based on the three classes, they identified three fundamental tasks for data exploration: *finding gestalt*, *posing queries*, and *making comparisons*. Finding linear or nonlinear patterns of interest, like discontinuities, clusters, or discreteness are considered as the task *finding gestalt*. *Posing queries* is the natural task after gestalt features are found and characterized, in order to comprehend parts of the data. Two types of comparisons are considered for the *making comparisons* task: the comparison of variables or projections and the comparison of subsets of data. Further, Buja et al. introduced a relationship between the proposed tasks and the interaction called manipulation view. *Finding gestalt* is assigned as focusing individual views, *posing queries* aims to link multiple views, and *making comparisons* is related to arranging many views [Buja et al., 1996].

In addition, Buja et al. listed a set of low-level interaction techniques that match high-level task groups [Buja et al., 1996]. Later Chuah and Roth proposed a set of basic visualization interaction primitives, introducing another set of low-level visual interactions that could be abstracted into three high-level tasks: *Data operations*, *Set operations*, and *Graphical operations* [Chuah and Roth, 1996]. Also several user-centered task classifications were introduced that do not only consider the interaction and manipulation of visualizations but focus more on the user's intended tasks [Zhou and Feiner, 1998, Ward et al., 2010, Yi et al., 2007, Pike et al., 2009, Fluit et al., 2006]. Another important and commonly used classification for low-level analysis tasks is the taxonomy of Amar et al. which consists of: *retrieve value*, *filter*, *compute derived value*, *find extremum*, *sort*, *determine range*, *characterize distribution*, *find anomalies*,

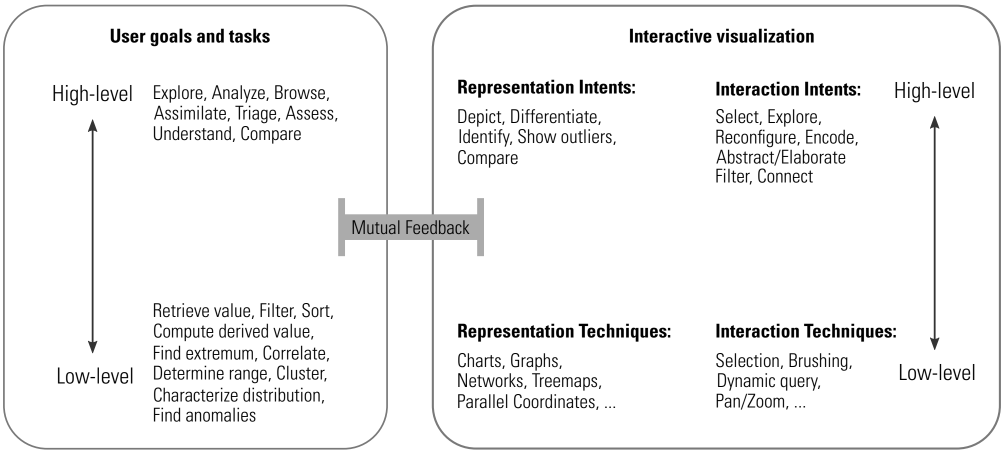

User goals and tasks
High-level
Explore, Analyze, Browse, Assimilate, Triage, Assess, Understand, Compare
Low-level
Retrieve value, Filter, Sort, Compute derived value, Find extremum, Correlate, Determine range, Cluster, Characterize distribution, Find anomalies
Mutual Feedback
Interactive visualization
Representation Intents:
Depict, Differentiate, Identify, Show outliers, Compare
Interaction Intents:
Select, Explore, Reconfigure, Encode, Abstract/Elaborate Filter, Connect
High-level
Representation Techniques:
Charts, Graphs, Networks, Treemaps, Parallel Coordinates, ...
Interaction Techniques:
Selection, Brushing, Dynamic query, Pan/Zoom, ...
Low-level

Figure 2.2: High and low level task and interactive visualization classification by [Pike et al., 2009].

cluster, and *correlate*. However, other taxonomies like Pike at al.'s comprehensive classification covers the relationship between users tasks and interaction based on user's intentions in a broader way. Their classification attempts to differentiate between high-level and low-level interaction techniques serving high-level and low-level user tasks and goals. It focuses on a mutual feedback between user goals and the result of analyzing and changing a representation through interaction [Pike et al., 2009]. Figure 2.2 illustrates their classification of high- and low-level user tasks and goals and a mutual feedback between user goals and tasks and interactive visualizations. The classification of Pike et al. considers the interaction value and user's goal and tasks from two perspectives of information visualization and visual analytics. It gives a good overview of two level task and interaction levels but the differentiation of high- and low-level tasks is not very clear.

In contrast, Munzner introduces another user-centered task classification which considers three levels of actions defining user goals [Munzner, 2014, p. 42]:
Analyze: this high-level task addresses how the visualization system can be used to analyze data. This means, the choices are whether users want to consume existing data or produce new material.
Search: the user searches for elements of interest, which is a mid-level task. Search is classified further based on whether the target and location are known or not.
Query: as the low-level user goal queries a target or set of targets in one of these three ways: identify one target, compare some targets, or summarize all of the targets.

Another simple but relevant classification of visualization tasks that is suggested in different works under different names [Schumann and Müller, 2013, Keim et al., 2006, Pike et al., 2009], classifies user tasks into three groups:
Presentation: the purpose is to communicate the results of an analysis. The information to be presented and the goals are clear and defined upfront. The choice of visualization techniques depends on the designer or developer and the user is not actively involved and therefore there is very little interaction involved.
Exploration: there is no clear hypothesis about the data in the beginning. An overview

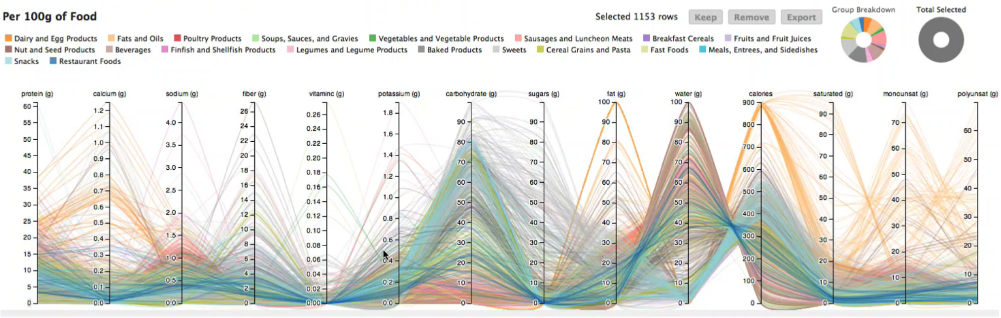


Figure 2.3: Parallel Coordinates showing the USDA Nutrient database with several nutrient dimensions [Kirk, 2016, p. 185].

of the data is shown and users browse and analyze the data to find trends, relationships, or outliers in the dataset. The main purpose of visual exploration is to gain insight, the goal is unclear and the process is not directed. The user is highly involved in this process and is the driving force.

Analysis: there are hypotheses about the data and the user tries to examine these hypotheses visually. The main goal is to confirm or reject those hypotheses. Therefore, the user carries out a directed search and a lot of interactions are involved.

2.1.4 How: Visualization Techniques

During the last decades, many visualization techniques have been developed with different approaches for different domains. The existing visualization techniques can be classified based on various criteria such as data, tasks, interactions, or the stages of data processing. There are two main reasons why we need a taxonomy for visualization techniques. First, to help users to choose proper visualization techniques for their questions by making their way of thinking and application goal clear. Second, to discover the limitations of current visualization techniques, that might trigger the creativity of researchers to develop new techniques [Buja et al., 1996]. Surveys of current visualization techniques can be found in a number of books [Card et al., 1999, Schumann and Müller, 2013, Ware, 2013]. One of the most common ways of classifying information visualization is based on data. Card and Mackinlay introduced a taxonomy based on *data type values* (ordered, nominal, and quantitative) described in section 2.1.2. They classified visualization techniques first in 1997 into scientific visualization, GIS, multi-dimensional plots, multi-dimensional tables, information landscapes and spaces, node and link, trees, and text transformation [Card and Mackinlay, 1997]. However, in 1999 they revised their classification slightly after redefining information visualization as a separate field from scientific visualization [Card et al., 1999].

Shneiderman and Keim introduced classifications depend on type of data taxonomy as well. Shneiderman's classification was based on his taxonomy for data types described in section 2.1.2, and its correlation with visualization tasks described in section 2.1.3. Later Keim et al. enhanced the classification of visualization techniques by defining the correlation between data types and interaction techniques [Keim, 2005] (see Figure 2.9). The groundwork classification used in this work is the taxonomy by

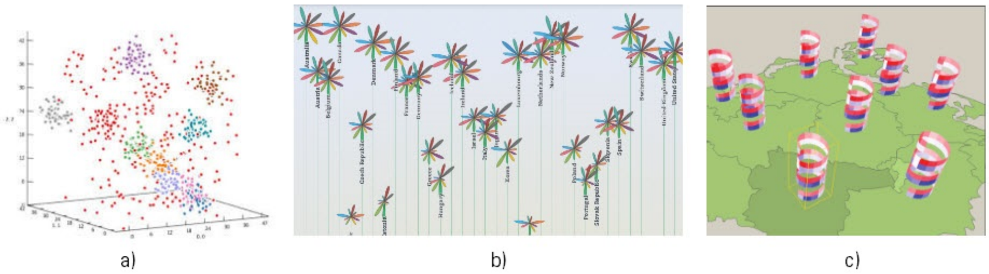


Figure 2.4: Examples of geometric and icon-based techniques: a) A 3D-Scatterplot using color as further dimension [Sanftmann and Weiskopf, 2012], b) OECD Better Life Index: flower glyphs are used to visualize different countries [Kirk, 2016, p.89], c) Helix glyphs on maps for analyzing cyclic temporal patterns for two diseases [Tominski et al., 2005].

Keim et al. They grouped the visualization techniques for multi-dimensional data into the following 5 categories [Keim, 2005]:

Geometric Techniques: in geometric techniques, the data attributes are projected into a position in geometric space. This projection is used to find patterns and trends in the datasets (in particular in multi-dimensional datasets). The geometric transformation can be performed in various ways. *Scatterplots* are well-known visualization techniques to encode two and at most three dimensions in the Cartesian coordinate system [Andrews, 1972]. However, by using icon-based techniques such as items in scatterplots, or using visual variables like shape, color, and size, the number of displayed attributes can be increased up to seven [Mazza, 2009]. Yet, the number of attributes to be displayed is limited and combining many visual variables is affecting perceptual effectiveness (see Figure 2.4 a). The widely used visualization technique of *Parallel Coordinates* can overcome this problem by arranging any number of attributes on equidistant axes which are parallel to one of the screen axes (see Figure 2.3) [Inselberg, 1985]. Parallel Coordinates axes are arranged parallel to each other, however there are other geometric techniques such as star plots [Chambers, 2017, p. 23] or TimeWheel [Tominski et al., 2004], in which similarly one axis represents one dimension but the axes are arranged differently – for instance in a radial layout. Moreover, different methods or interaction techniques have been developed to reduce clutter in Parallel Coordinates such as sampling [Ellis and Dix, 2006], clustering [Fua et al., 1999], or splatting approaches [Zhou et al., 2009].

Icon-based Techniques: in iconic-based techniques, each date entry is represented as an independent visual object, also known as a glyph. The data attributes are projected into visual variables of a glyph such as size, shape, color, and orientation [Ward, 2008]. The major strength of glyphs is that multivariate data involving more than two attribute dimensions can often be shown in the context of a spatial relationship (see Figure 2.4.b). However, the main drawback is the encoding capability due to the size, limited capacity of individual visual channels, and cognitive challenge on learning and memorization [Borgo et al., 2013]. Icon-based techniques are very common solutions to be combined with other visualization techniques (see Figure 2.4.c) [Tominski et al., 2005]. They are also widely used to convey uncertainty along with other aspects of the data [Pang et al., 1997].

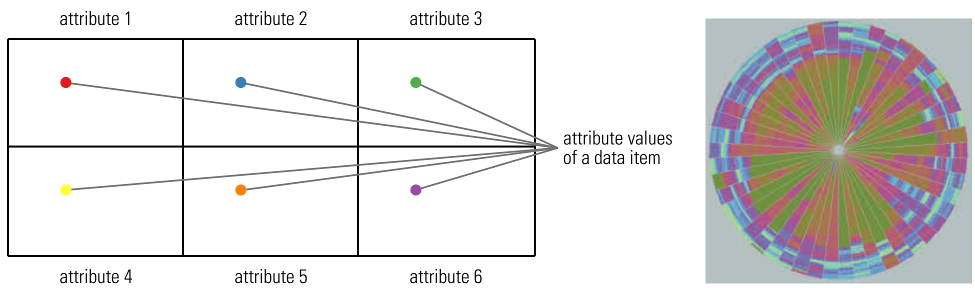


Figure 2.5: Pixel-based techniques: left) each attribute value is represented by one colored pixel, right) 50 attributes in circle segment windows and linear pixel arrangement [Keim, 2000].

Pixel-oriented Techniques: in pixel-oriented techniques, each attribute of the dataset is represented by one colored pixel [Keim, 2000]. The screen is divided into a set of windows and each window represents one dimension of the dataset (see Figure 2.5.a). This visualization technique is appropriate for large datasets since the huge number of pixels on a screen can accommodate a large number of data points. However, it has also limitations. The two main challenges are (1) the number of visual variables and (2) the suitable arrangement of pixels. Keim introduced the concept of using sub-windows in various ways such as circle segment techniques or rectangular techniques. Figure 2.5 shows one example of radial layout where each attribute is shown by one circle segment [Keim, 2000]. Heat maps are another simple and commonly used form of pixel-based techniques and are especially useful to identify outliers by looking for strongly deviating colors [Eisen et al., 1998].

Hierarchical Techniques: these techniques are used to visualize the hierarchical structure within an attribute or between multiple attributes. Visualizing this type of data described by Chen as one of the “most mature and active branches in information visualization” [Chen, 2006]. Meirelles classifies these techniques into stacked and nested visualizations [Meirelles, 2013]. In stacked visualizations, the elements are arranged in relationship to each other, often connected by lines. A well-known example of this kind is node-link visualizations, in which the items are represented as points, and the relations between the entities as connecting lines (see Figure 2.6: a, b, d). For nested visualizations, the elements are composed of containers, grouped or associated according to their hierarchy (see Figure 2.6: c, e, f, g, h). One well-known nested visualization example is Treemap, which is described as an optimally space-efficient technique as it allows for a weighted partitioning of the area. Figure 2.6 shows several basic kinds of hierarchical techniques collected by McGuffin and Robert [McGuffin and Robert, 2010]. They mention space-efficiency as the major challenge in designing and comparing 2D tree visualizations, and their analysis prove that the concentric squares representations are more space-efficient than classical layered or concentric circle representations.

Graph-based Techniques: these techniques help to clearly and quickly visualize large graphs. Graphs are a mathematical concept that can be defined in the following way: A graph $G = (V, E)$ is a set of nodes V and a set of edges $E \subseteq V \times V$ [Tutte, 1998]. The visualization techniques to represent graphs can be divided into four

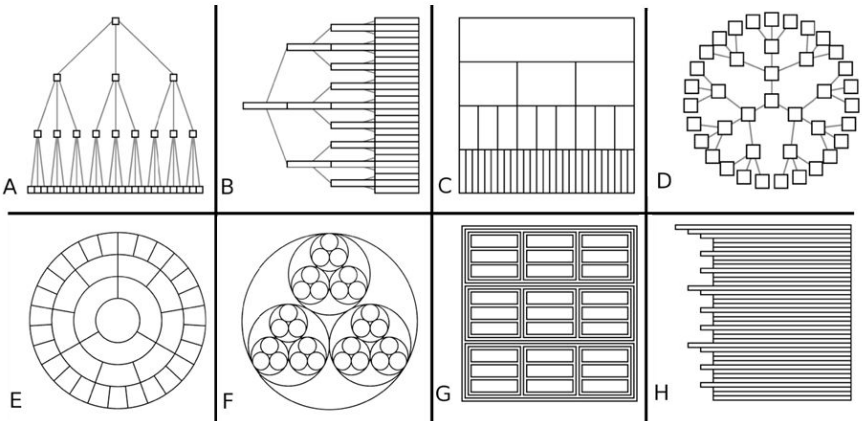


Figure 2.6: Different visual encoding techniques representing the same tree dataset: a) classical (layered) node-link, b) a variation on (a) in which the shape of nodes better accommodates long labels, c) icicle, d) radial, e) concentric circles, f) nested circles, similar to g) Treemap, and h) indented outline, sometimes called a tree list [McGuffin and Robert, 2010].

groups: *matrix representations*, *explicit node-link representations*, *implicit representations*, as well as their *hybrids* [Von Landesberger et al., 2011].

Matrix representations: represent a direct visualization of the adjacency matrix of a graph. A matrix representation is a grid of nodes along with the cells representing the edge weights. Those weights are depicted by coloring the cells according to a given color scale (see Figure 2.7.a). The main challenge in this type of visualizations is how to find a good ordering of the nodes together with rows and columns. This technique focuses more on the graph edges as they assign the most drawing space to represent information about them.

Explicit representations: or node-link representations are one of the most common graph-based solutions which use points to depict nodes and lines to represent the connection between the nodes (see Figure 2.7.b). The main problem is caused by occlusion of nodes and links. There are many layout structures defined to optimize this problem. This representation gives equal importance to nodes and links in the graph structure. However, there are many techniques developed to improve the appearance and readability of node-link diagrams apart from their layout structures [Dwyer et al., 2005, Holten and Van Wijk, 2009].

Implicit representations: in this kind of representation instead of employing links for showing node relationships, the relationship is implicitly encoded, like Treemap [Johnson and Shneiderman, 1991], Icicle Plot [Kruskal and Landwehr, 1983], InterRing [Yang et al., 2002], or Power Graphs [Royer et al., 2008]. These techniques are also known as space filling techniques, as the absence of links allows to make use of available drawing space (see Figure 2.7.c). Another advantage of this representation is their emphasize on nodes attributes as object size. The major challenge in this type of visualization is finding a layout for nodes that represents the graph and positional relationships in the best way while still achieving space efficiency. Also, the positional relationships can create visual clutter by using layouts like overlap or inclusion.

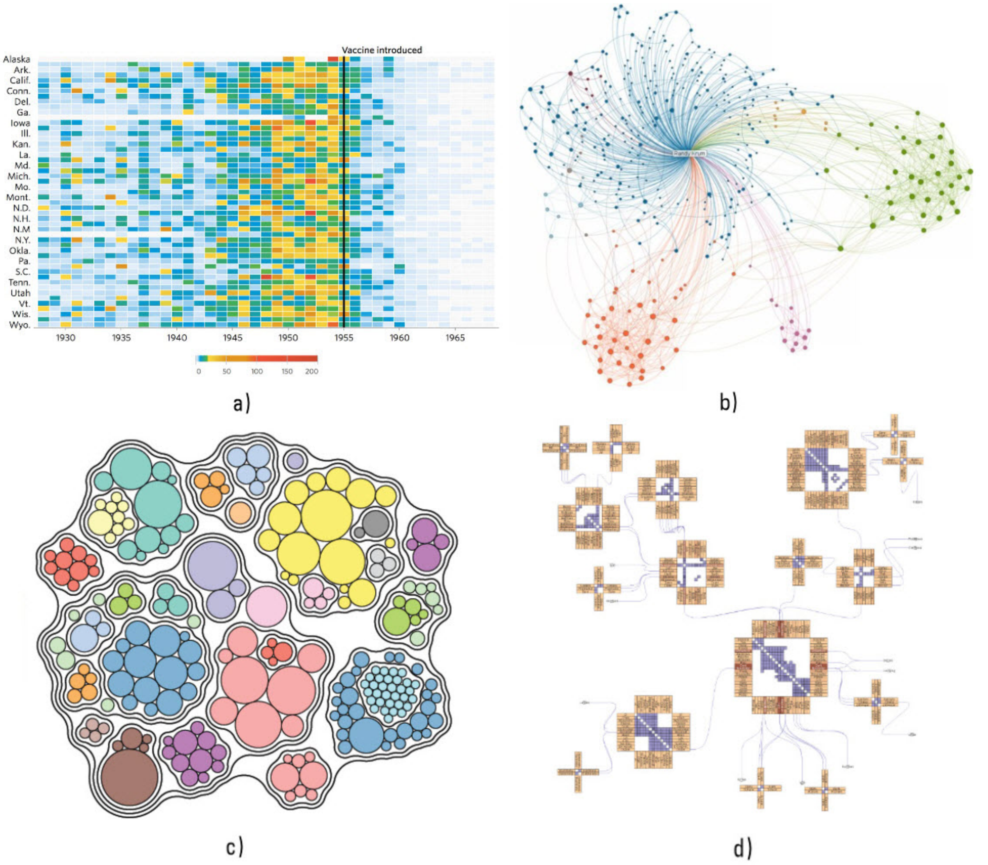


Figure 2.7: Graph-based visualization techniques: a) matrix representation of Polio vaccines impact over time [Kirk, 2016, p. 273], b) LinkedIN Maps visualizing LinkedIN network [Krum, 2013], c) Bubble Treemap visualizing the package structure of the Flare software [Görtler et al., 2018], d) Node Trix visualizing the information visualization field [Henry et al., 2007].

Hybrid representations: are a new representations that consist in the combination of the techniques explained above. This new representation type can help to compensate the disadvantages of each type. Interaction techniques are widely used to facilitate the use of hybrid approaches and help the users to explore regions of interest. Figure 2.7.d shows an example of hybrid approach using the combination of matrix and node-link representation [Henry et al., 2007]. This combination provides a much clearer view of the overall structure of the graph with less clutter.
Based on the classification introduced above, most visualization techniques discussed in the context of this thesis are *Hierarchical* and *Geometric* techniques.

2.1.5 How: Interaction Techniques

Interaction techniques are an essential part of information visualization, especially because of their power in solving the given visualization tasks through interactive user interfaces. They help to overcome the problems raised by large data size in visualization applications. In most data visualizations, users need to interact with more information than what can be displayed on a single screen. Interaction techniques can help uncover relationships in the data, which would not be easily visible in static visualizations [Munzner, 2014, p.9]. Dix et al. describe interaction in the broader context of human computer interaction, as “the communication between user and the

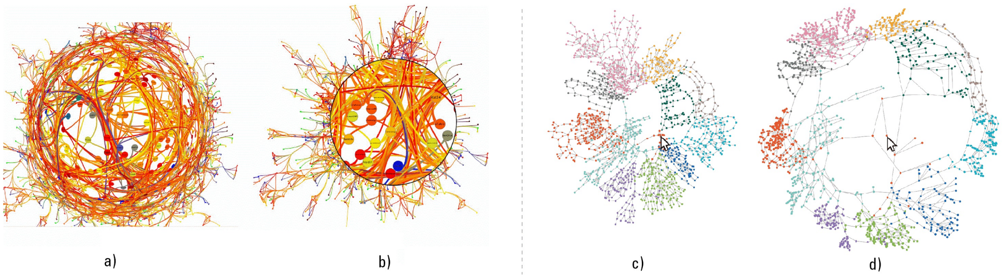


Figure 2.8: Two different focus+context approaches for graph exploration: a) Fisheye lens for local inspection, b) Magnifying lens for zooming on a local region [Antoine et al., 2010], c) Magnifying lens, and d) Structure-aware Fisheye maintains the shapes of clusters and minimizes their distortions [Wang et al., 2019].

system" [Dix, 2009, p. 124]. There are several classifications, concepts and techniques introduced during the past decades for visual interactions. In this section, we only review approaches relevant to this thesis.

One of the first interaction-based approaches developed to explore large amounts of data on screens is the *Visual Information Seeking Mantra* proposed by Shneiderman [Shneiderman, 1996]: *"Overview first, zoom and filter, then details-on-demand"*. The starting point should be a global overview of the entire information space. Zoom provides the possibility to focus on the relevant information areas, and irrelevant information can be hidden through filter operations. Finally, if needed, additional information can be provided to the user, for example by inspecting data items. The Information Seeking Mantra is not a classification of visual interactions, but rather a starting point to design user interfaces by considering the *task by data type* taxonomy of information visualization. The mantra further associates seven data types to seven high-level tasks (see section 2.1.3).

Later, the Information Seeking Mantra was extended in several different works, for example Cockburn et al. defined "overview+context" as a solution to represent focused information entities and contextual information by spatial separation [Cockburn et al., 2009]. They distinguish between four approaches to move between focused and contextual views: (1) Overview+detail: which uses a spatial separation between focused and contextual views, (2) Zooming: which uses a temporal separation, (3) Focus+context: which minimizes the seam between views by displaying the focus area within the context, and (4) Cue-based techniques which selectively highlight or suppress items within the information space. Figure 2.8 shows two examples of applying focus+context methods (Fisheye and Magnifying lens) to graphs. Keim also refined the Information Seeking Mantra by classifying the interaction techniques into *distortion techniques* (simple or complex) and *data visualization techniques*: 1. Data-to-visualization mapping, 2. Projection, 3. Zoom, 4. Filtering (selection, querying), 5. Details on demand, and 6. Linking & brushing [Keim, 1997]. Figure 2.9 shows Keim's classifications of interaction, visualization techniques (see section 2.1.4), and data type (see section 2.1.2). Keim's taxonomy is based on the effects of the interaction methods on the display.

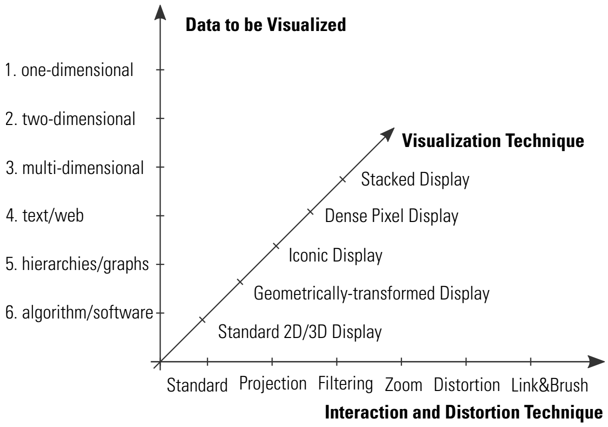


Figure 2.9: Interaction classification in visualization systems according to [Keim, 2005].

In contrast to Keim's classification, Hearst proposed context-oriented interaction techniques for supporting dynamic, interactive use of information visualization within abstract data space: brushing and linking, panning and zooming, focus+context, magic lenses, and animation to retain context [Hearst, 1999]. For example, zooming is combined with panning, where panning provides the overview of contextual information, zooming looses that. Heer & Shneiderman suggested a taxonomy of interactive dynamics that contains 12 task types grouped into three categories: (1) data and view specification, (2) view manipulation, and (3) analysis process and provenance [Heer and Shneiderman, 2012]. Their taxonomy (shown in Table 2.2) distinguishes between data, view, and process centric tasks. Similarly, Brehmer and Munzner distinguish 3 classes of interaction techniques: encoding data, manipulating existing elements in a visualization, and introducing new elements into a visualization [Brehmer and Munzner, 2013]:

Encode: describes how the data is initially encoded as a visual representation. This approach is similar to the data-to-visualization mapping in Keim's taxonomy [Keim, 1997], and in Heer et al. classification the equivalent is Visualize, under the Data & View Specification group [Heer and Shneiderman, 2012].

Manipulate: includes methods that affect existing visual elements such as, *select:* change the granularity of visualization element (see Figure 2.10 all four examples),

Data & View Specification	**Visualize** data by choosing visual encodings **Filter** out data to focus on relevant items **Sort** items to expose patterns **Derive** values or models from source data
View Manipulation	**Select** items to highlight, filter, or manipulate them **Navigate** to examine high-level patterns and low-level detail **Coordinate** views for linked, multi-dimensional exploration **Organize** multiple windows and work-spaces
Process & Provenance	**Record** analysis histories for revisitation, review and sharing **Annotate** patterns to document findings **Share** views and annotations to enable collaboration **Guide** users through analysis tasks or stories

Table 2.2: Task Taxonomy by Heer & Shneiderman [Heer and Shneiderman, 2012].

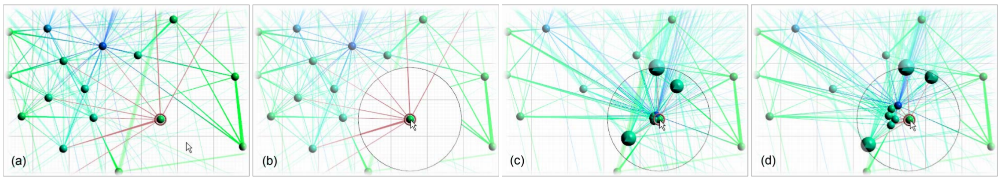


Figure 2.10: Interaction graph lenses: a) view with a focused node, b) the local edge lens removes edge clutter, c) the layout lens gathers nodes that are adjacent to the focus node but might be scattered in the layout, d) The composite lens combines (b), (c), and a fisheye lens. According to [Tominski et al., 2009].

navigate: altering user's viewpoint, *arrange:* organizing visualization elements spatially (see Figure 2.10 b, c, d), *change:* altering visual encoding (2.10 b) by adopting the transparency of edges in the graph), *filter:* the exclusion and inclusion of visual elements, *aggregate:* changing the granularity of visualization elements.

Introduce: consists of methods that add new visual elements such as *annotate:* adding graphical or textual annotations associated with one or more visualization elements, *import:* including new data elements, *derive*: computing new data elements given existing data elements, *record:* save or capture visualization elements as persistent artefacts.

Several of the interaction techniques described in this section are used in this thesis. Particularly, the interaction methods under the *manipulate* category play a crucial role in the design of our solution.

2.2 Visual Perception

Understanding visual perception is fundamental in the context of information visualization. Colin Ware relates visual perception to information design and claims: "When data is presented in certain ways, the patterns can be readily perceived. If we can understand how perception works, our knowledge can be translated into rules for displaying information. Following perception-based rules, we can present our data in such a way that the important and informative patterns stand out." [Ware, 2013, p.xxi]. When creating custom or novel visualization tools, designers must consider a variety of concerns such as perceptual effectiveness and aesthetic choices. Therefore, we cover some basics in this field to better understand visual perception and design guidelines relevant to information visualization. After reviewing the visual attributes, we will give an account of preattentive and attentive attributes, and then we briefly describe the Gestalt principles.

2.2.1 Visual Variables

Visual variables are information carriers. Different types of relationships are shown by different forms of data visualizations using variations of visual components that can be used to communicate them effectively. They were first applied in cartography, but were later adopted by information visualization in general. First, we describe the visual variables based on the pioneering works of Bertin [Bertin, 1983] and Mackinlay [Mackinlay, 1986]. Bertin introduced visual variables and their mutual relationships in 1974. Based on his definition, graphical representations use three elementary types of geometrical elements or marks: *points*, *lines*, *areas*. Marks can be changed by

	Position	Size	Value	Texture	Color	Orientation	Shape
Selective	✓	✓	✓	✓	✓	✓	∼
Associative	✓	✓	✓	✓	✓	✓	∼
Quantitative	✓	∼	∼	×	×	×	×
Order	✓	✓	✓	×	×	×	×
Length	✓	✓	∼	✓	✓	✓	✓

Table 2.3: Characteristics of visual variables based on Bertin's visual presentation.

	Position	Size	Value	Texture	Color	Orientation	Shape
Nominal	✓	✓	✓	✓	✓	✓	✓
Ordinal	✓	✓	✓	∼	×	×	×
Quantitative	✓	✓	∼	×	×	×	×

Table 2.4: Characteristics of visual variables based on data types.

seven elements of visual information: *position* (place), *shape*, *size*, *value* (light, size), *texture* (within the shape), *hue*, and *orientation/direction* [Bertin, 1983]. Bertin classified visual variables further by the way they are perceived: *selective*: a change in this variable is enough to select it from a group, *associative*: a change in this variable is enough to perceive them as a group, *quantitative*: a numerical reading is obtainable from changes in this variable, *order*: changes in this variable are perceived as ordered, and *length*: over many changes in this variable, distinctions could be perceptible. Table 2.3 shows the efficiency of Bertin's visual variables by their perception abilities. Except *shape*, all visual variables are *selective* and *associative*. In addition, *position*, then *size*, and after that *value* appear to be the strongest visual variables based on this classification. These rules are extensively useful in the design and evaluation process of information visualization tools. Furthermore, the efficiency of visual variables for presenting different data types based on Card and Mackinlay's data type classification (see section 2.1.2) is summarized in Table 2.4. All seven visual variables can be used to represent *nominal* data. *Position*, *size*, and then *value* are the strongest visual variables that could be used to represent all different data types.

While Bertin subdivided the visual variables into *retinal variables* and *layout*, Card et al. proposed a different classification that refines the model of Bertin and consists of *spatial substrates*, *marks*, and *graphical properties* [Mackinlay, 1986]. Spatial substrate defines the dimension of the representation. Each dimension can be considered as an axis that is either linear or radial. Furthermore, the axis types are nominal, ordinal, quantitative, or unstructured. *Marks* are extension of Bertin's definition by adding volumes (3D), to point (0D), line (1D), and surface (2D) [Card et al., 1999, p. 28]. Moreover, based on Card and Bertin's work described above, Mackinlay proposes a ranking of visual variables for perception, based on quantitative, ordinal, and nominal data types [Mackinlay, 1986]. Both Bertin and Mackinlay address the effectiveness of different visual variables for perceptual tasks. Subsequently, based on the work of Bertin, Card, Mackinlay, and Ware, Munzner assesses the visual variables in terms of their effectiveness (see Figure 2.11). She distinguishes between categorical and ordered data attributes and accordingly defines two channels named *magnitude* and *identity* channels. Moreover, a channel effectiveness is defined based on accuracy, discriminability, separability, popout, and grouping features of visual channels.

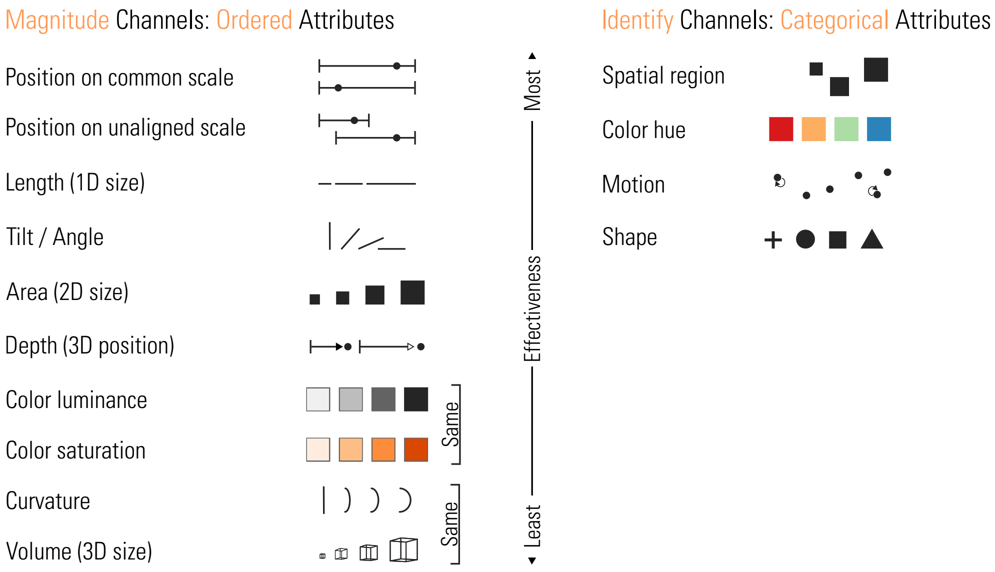


Figure 2.11: Ranking of visual variables based on data values by [Munzner, 2014, p.94].

2.2.2 Attributes of Preattentive and Attentive Processing

Understanding the preattentive and attentive attributes guides us to design information visualization tools that emphasize the most important aspects of the data, and make other visual elements less dominant. This section summarizes fundamental vision perception theories that are relevant for this research. However, physiological aspects of vision perception are not considered here.

Preattentive features are a limited set of visual attributes that can be simultaneously detected by humans at an early stage of visual perception which lasts less than 250 milliseconds. That means that certain information can be processed in parallel at an extremely high speed and this allows identifying an object preattentively [Ward et al., 2010]. Based on Ward et al.'s categorization, the four primitive variables are luminance and brightness, color, shape, and texture [Ward et al., 2010]. The detection of those variables is also termed by *pop-out effect*. Figure 2.12 shows Colin Ware's classification for preattentive attributes, organizing the visual variables in 4 categories: *form* (orientation, line length, line width, size, shape, curvature, added marks, and enclosure), *color* (hue, intensity), *spatial position* (2-D position), and *motion* (flicker, direction) [Ware, 2013]. Some of these attributes work better to express quantitative values, and others for categorical values. Bertin has suggested that each visual variable can be used to encode quantitative or qualitative values. The visual attributes that can be perceived quantitatively are those that can be used for categorical data, like color intensity or size. Another fundamental work in this field, which provides important insights into human visual perception is Treisman's work on preattentive processing [Treisman and Gelade, 1980]. She tried to answer two main questions in her research: first, determining those visual properties that could be detected preattentively, and second, formulating hypothesis about how the human visual system performs preattentive processing.

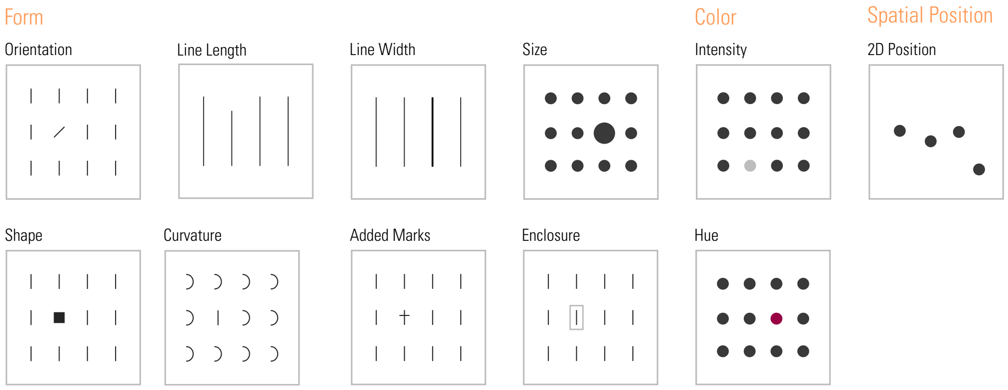


Figure 2.12: List of attributes of preattentive processing based on [Few, 2004, p. 98].

Besides the preattentive processing stage, Ware also proposed two more stages for perceptual processing: *Pattern Perception* and *Sequential Goal-Directed Processing*. These two stages that we consider as attentive processing are also called postattentive vision [Ward et al., 2010] or directed attention [Wolfe and Gray, 2007]. In stage one (preattentive processing), information is processed in parallel to extract basic and low level visual features. At the second stage, rapid active processes divide the visual field into regions of different color, texture, and motion patterns. At the last stage of Ware's perceptual model, the information is reduced by the demands of active attention and only few objects are in focus of the attention. Those objects are constructed from available patterns to solve a given visual query task [Ware, 2013, p.21].

Moreover, much work has been done to measure the human ability to distinguish visual perceptions. As Colin Ware discusses in his book, there are limitations in what we can distinguish preatentively when the variety of distracters increases [Ware, 2013, p.154]. In 1956, George Miller claimed that our brain has limited capacity for receiving, processing, and remembering information. The result of his research suggests that we cannot perceive more than *seven* levels of data at once. This is referred to as the *magical number seven* in cognitive science. He also suggests some solutions to increase our memory capacity by using a grouping strategy or designing visualizations so that they rely on relative judgments rather than absolute ones [Miller, 1956]. Besides that, Stephen Few stresses the importance of considering the *context* of visual attributes with regards to their perception. Some visual attributes like color or size are often perceived by their differences not their exact values. Figure 2.13 illustrates two examples: the one on the left shows how using the same gray color in relation to two different blue colors can impact our perception, and the one on the right illustrates the influence of diagonal lines at the end of two horizontal lines with the exact same length. They appear to be different to us because we perceive them in relation to their

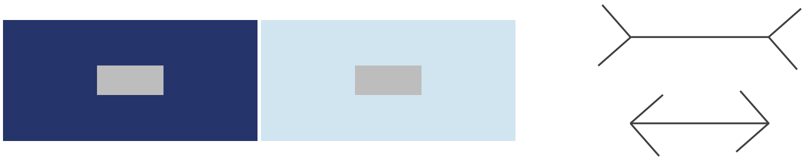


Figure 2.13: The perception of color and line length in different contexts.

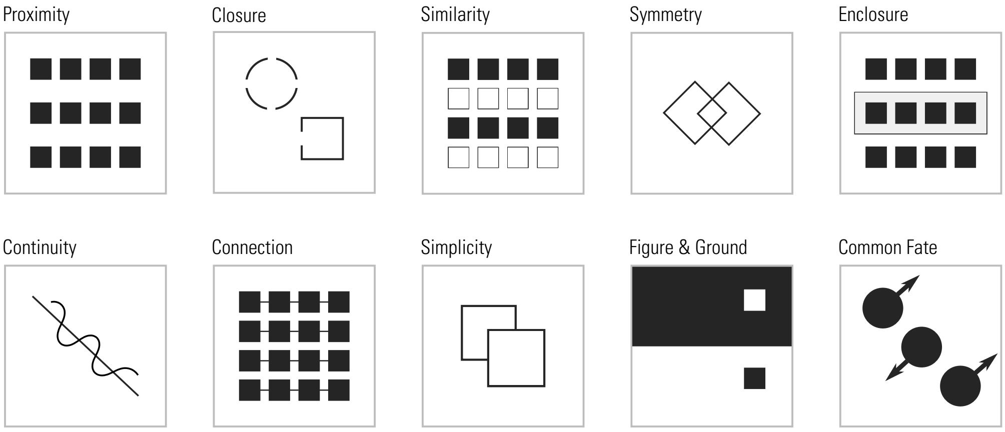


Figure 2.14: Gestalt laws of perception.

surroundings. These considerations will be taken into account for the design of our visualization solutions described in section 5.

2.2.3 Gestalt Principles

The Gestalt School of Psychology strives to understand how we perceive patterns, and organize what we see. Researchers in this school compiled a list of Gestalt principles of how we perceive and group objects in particular ways [Koffka, 2013]. These observations became a set of design principles that can be beneficial in designing information visualization tools [Ware, 2013, p.189]. Therefore, in the following we will explain these Gestalt laws briefly (see Figure 2.14):

- Law of **Proximity:** one perceives items that are closer to each other as a group. This principle is widely used in user interface design.
- Law of **Similarity:** items with similar attributes such as shape, color, size, texture, etc. tend to be perceived as a group.
- Law of **Figure and Ground:** images are perceived as an object (figure) in the foreground, and a surface (ground) that lies behind the figure.
- Law of **Closure:** things are grouped together if they seem to complete some entity. Cognitive processes in the brain often ignore contradictory information and have a perceptual tendency to fill in gaps in contours.
- Law of **Continuity:** unfinished objects are perceived as complete and closed when there is a way to interpret them as such.
- Law of **Simplicity:** our mind perceives figures as simple elements instead of complicated shapes.
- Law of **Connection:** objects that are connected are perceived as a group. Connectedness is a powerful grouping principle that is stronger than proximity or even similarity attributes like color and shape.
- Law of **Symmetry:** similarities are perceived more readily when objects are arranged symmetrically. This law provides a powerful organization principle.
- Law of **Common Fate:** objects that move together or point in the same direction tend to be perceived as a unified group.
- Law of **Enclosure:** objects can be perceived as a group by enclosing them. It can be typically done by adding a border around the items.

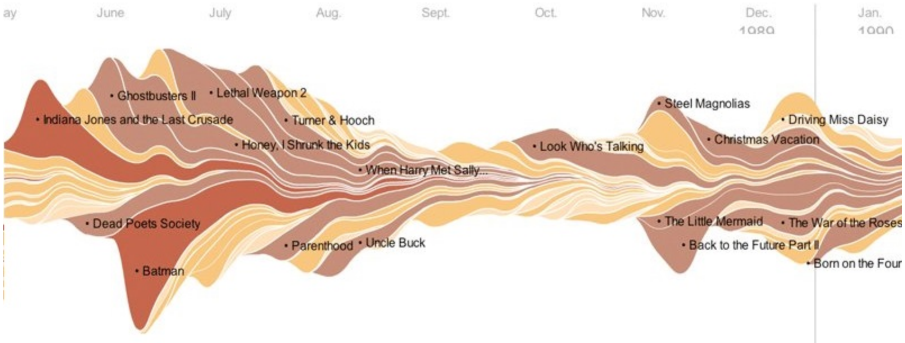


Figure 2.15: ThemeRiver: using a continuous flow diagram to represent temporal structures. It shows how movies have fared at the box office between 1986-2008 [Meirelles, 2013, p.109].

There is a body of research investigating the role of design in information visualization [Moere and Purchase, 2011]. More and more sophisticated visualizations are being developed, but there is still very little consensus on design principles and best practices. For example, the result of preattentive processing is used for the glyphs or symbols' design [Borgo et al., 2013], and Purchase examines the impact of aesthetics in the design process of graphs [Purchase, 1997]. In this work, we give a detailed account of the design process followed during the creation of our visualization solution.

2.3 Flow Diagrams

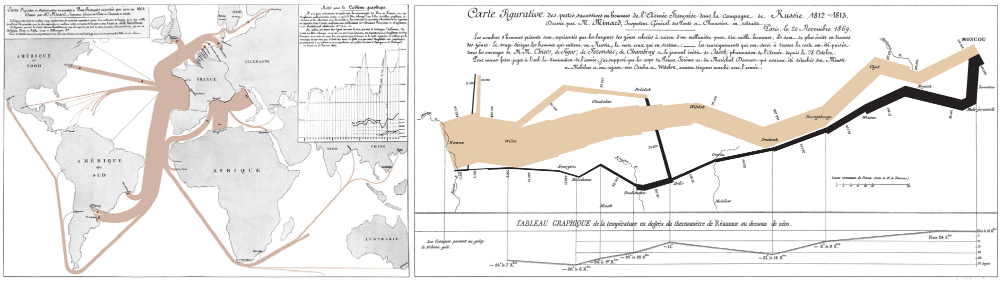
The term *flow diagram* used in this thesis refers to a kind of diagram that represents weighted flows or sets of relationships in a dataset. It shows interrelated information between data items or physical routes when different objects are connected by *area* marks (see section 2.2.1). In flow diagrams, the thickness of each individual flow shape corresponds to the values in each category. It does not refer to flow charts that represent an algorithm, workflow, or process. The aim of using flow diagrams is not to express statistical quantities, but to convey relationships and patterns in the datasets that cannot be seen in numbers with bare eyes or requires significant cognitive effort. In the following we propose a taxonomy to classify flow diagrams.

2.3.1 Classifications of Flow Diagrams

We classify flow diagrams into two main groups. The first category consists of diagrams designed to visualize continuous phenomena with concrete flows. The second group refers to diagrams representing relationships in network structures or graphs using abstract flows, where the flow thickness represents quantities.

Concrete Flows

Concrete flow diagrams show curvilinear phenomena that involve continuous movement and connection between points. The thickness of flows represent magnitude at each point which means it uses variations in width to show variations in strength. Flows demonstrate concrete phenomena such as time and space that are continuous. However, one of the disadvantages of this type of diagrams is that too many node occlusions and link crossings can occur when used for large datasets. We classify concrete flow diagrams further into the following three categories:



a)

b)

Figure 2.16: Using flow diagrams to represent: a) spatial structures: Minard's 1864 flow map of wine exports from France, b) spatio-temporal structures: Minard's 1869 map of Napoleon's 1812 campaign into Russia and the losses he suffered 1812 - 1813 [Tufte, 1983].

a) Temporal: One of the simple and effective diagrams to represent changes in data over time of different categories with smooth and continuous curves are flow diagrams. In this case, the width of the horizontal flow represents the change of a variable over time. Figure 2.15 shows an example of a temporal flow diagram called *ThemeRiver*, which is a visualization of theme change over time [Havre et al., 2000]. It is a stacked area graph, that has values displayed around a varying central baseline. In this example, each shape shows the commercial success of one film and the area and color of the shape correspond to the film's total earnings.

b) Spatial: Geographical data is of crucial importance. Much change in our physical environment and in human society depends on the movement of factors such as humans, money, viruses, or material. One way of depicting geographical movement is by visualizing them using flow maps [Guo, 2009]. Cartographers have long used continuous flow diagrams to show the movement of objects from one location to another [Phan et al., 2005]. One of the very first examples of these diagrams also called flow maps was presented by Minard in 1864 [Tufte, 1983, p.25] (see Figure 2.16 a). The first instances of flow maps were hand drawn and static. However, many of the recent designs are dynamic and benefit from more sophisticated methods such as edge bundling to reduce visual clutter [Phan et al., 2005].

c) Spatio-temporal: Another effective way to use flow diagrams is to add spatial dimensions to temporal displays so that the data is moving over space as well as over time. One of the first well-known examples is Minard's famous graph showing the decreasing size of the french army with two ribbons. The brown flow represents the size of the army decreasing over time as it marches to Moscow and the black flow represents the army on the way back from right to left (Figure 2.16 b). Tufte considers this graph as one of the best statistical graphics ever drawn [Tufte, 1983, p.41]. One more example of visualizing space-time dynamics using flows but in circular space is *Rank Clock* by Batty [Batty, 2006].

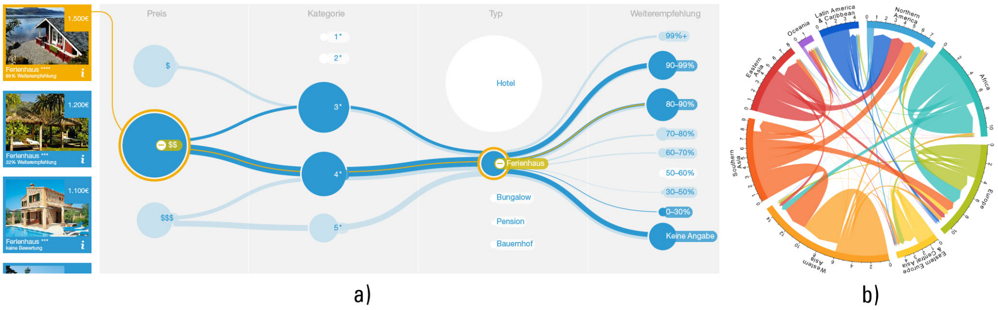


Figure 2.17: Examples of abstract flow diagrams a) Parallel Sets designed for travel search task with multiple filter possibilities on one or different axes [Keck et al., 2014], b) Chord diagrams representing estimated 5 year migrant flows between 2010 - 2015 [Abel, 2018].

Abstract Flows

Another common application for flow diagrams is to represent the relationships in a weighted graph. In this case, we are dealing with a collection of nodes and links with a particular structure. The diagram is not representing concrete flows, rather discrete values showing the connection between items in a graph. A node is an item in the network structure and a link represents the relationships between the nodes. However, links in flow diagrams do not only show the path and connectivity between nodes, but similar to concrete flow diagrams, their width represents magnitude. This is the main difference of abstract flow diagrams with node link diagrams (see section 2.1.4) that uses lines not areas to illustrate the connections. In abstract flows, links are not shown as continuous flows but the nodes in the data structure are also represented. That means the whole network structure along with some extra information on the weight of different items in the structure is shown. Abstract flow diagrams can be arranged in linear or circular shapes. We classify them further into the following groups:

a) Force Directed: Force directed flow diagrams preserve the positions of items in the data structure (similar to spatio and temporal flow diagrams). Therefore, the items cannot be rearranged by the user since there is an inherent logic behind the order and position of the nodes. Sankey diagrams are one example of force directed flow diagrams. In Sankey diagrams, nodes can be organized vertically and links horizontally or the other way around, but their order shows how the flows of elements such as energy or materials are distributed [Riehmann et al., 2005]. Alluvial diagrams are another example of flow diagrams, in which the order of items matter because they tend to show the changes in composition over time.

b) Not-Force Directed: In this type of abstract flow diagrams, items can be reordered on demand, since the relationship between nodes and how the values are distributed are of importance not their order. They are usually arranged in the following ways:
Linear: Parallel Sets are an example of linear not-forced directed flow diagrams. The order of dimensions along with the order of items per each dimension can be rearranged by end users or different algorithms [Bendix et al., 2005]. Figure 2.17.a shows one example of an interface designed for travel search based on the principle of Parallel Sets. The order of visualized products can be rearranged as the goal is to gain insights into the overall structure of the dataset [Keck et al., 2014].

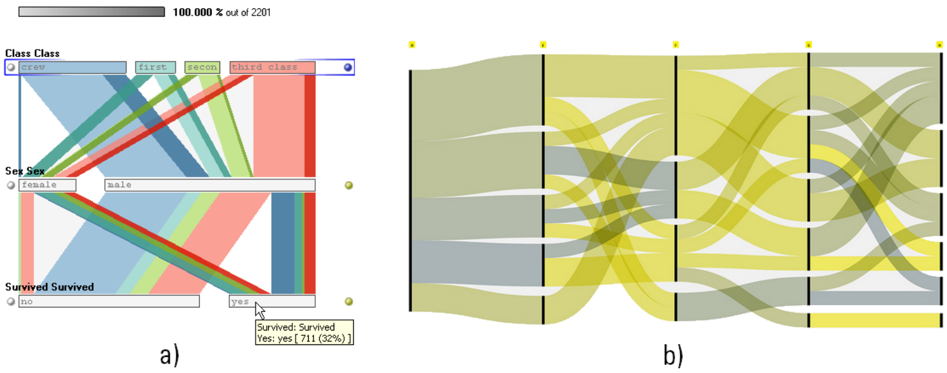


Figure 2.18: Examples of Parallel Sets with different flow shapes: a) represents the level of dependence between the class, the sex, and the survived passengers on the Titanic [Bendix et al., 2005], b) represents mapping the Republic of the Letters using Bézier curve [Meirelles, 2013, p. 70].

Circular: Chord diagrams are another type of abstract flow diagrams, in which the nodes are formed around a radial display. The relationships between nodes are displayed through the flows between and within categories. Figure 2.17.b shows the connections of estimated immigration between and within different world regions between 2010 and 2015 using Chord diagrams [Abel, 2018]. The primary arrangement logic in not-forced directed abstract flow diagrams is by sorting items, either by generating logical meaning through the neighbouring categories or axes, or by applying crossing minimization methods.

2.3.2 Main Visual Features

Since the focus of this work is on abstract flow diagrams, in this section we will review their main visual features along with their variations and alternatives. The two main visual features of abstract flow diagrams are *nodes* and weighted *links* that connect the nodes. One of the most common ways of plotting nodes is via rectangular shapes. The classical Sankey diagrams and Parallel Sets both use a stacked bar chart, displaying nodes proportionally sized to a quantitative measure [Riehmann et al., 2005, Bendix et al., 2005]. However, some more recent variations use circles or other novel shapes to indicate the value of node items by their area [Keck et al., 2014]. In Figure 2.18, two Parallel Sets visualizations are shown that apply different flow shapes. The connecting links of abstract flows diagrams are traditionally displayed using straight shapes (see Figure 2.18.a). However, curved bands are now used more often to connect items in the data structure. Using Bézier curves to plot the curved bands makes the curves look smooth and more pleasant to the eyes (see Figure 2.18.b). The Sigmoid function is commonly used to draw the curves. Moreover, it reduces the visual clutter and helps with the perceived readability of the diagrams. In most cases, the thickness of curves represents the quantity, which is equal all the way from the origin node to the destination node.

In Section 5, 6, and 7, we contribute novel ways of manipulating the two main visual features of flow diagrams in order to convey different aspects of the data.

2.4 Summary

In this chapter, we reviewed the relevant foundations of this thesis. In a first part, we gave a detailed description of the information visualization field structured by three questions: *what* is visualized, *why* it is visualized, and *how* is it visualized. In this context, we reviewed the various disciplines, techniques, and approaches that are introduced into the field of information visualization. We discussed various classifications of data types, tasks, visualizations, and interactions techniques in information visualization.

Subsequently, we reported principles of visual design that need to be considered while designing new information visualization tools. Finally, we introduced flow diagrams that are the main focus of this work, and several related visualization examples were shown. The main objective of this chapter was to give a common basis for the terms used in the context of interactive information visualization.

In the following chapter, we review the related work relevant to our research questions in order to situate the contributions of this thesis in the context of the state of the art.

Chapter 3

Related Work

The topics covered in this thesis touch upon different areas of information visualization. The main focus of this work is on aggregated data that is multi-dimensional, hierarchical, uncertain, and time-dependent. To survey the state of the art on this type of data, we will divide the problem into three parts. First, we give a detailed description of the available data visualization for multi-dimensional hierarchical aggregates. Second, we review the background for visualizing uncertainty in data. Finally, we review existing methods for time-dependent data with a focus on visual comparison task.

3.1 Cross-tabulating Hierarchical Categories

In this section, we review three main research fields related to the hierarchical cross-tabulation problem: *Tree*, *Set*, and *Categorical* data visualizations (see Figure 3.2). In section 2.1.4, we described different techniques for *tree* visualizations, namely under hierarchical and graph-based techniques. Also, some examples of *set* visualization were covered, specifically under geometric, icon, and pixel-based techniques. One typical visualization solution for visualizing item sets is *Parallel Coordinates* [Inselberg, 2009, Heinrich and Weiskopf, 2013, Johansson and Forsell, 2016]. It uses interconnected parallel axes to represent multiple dimensions, which have been proposed for a variety of data types (see Figure 2.3). Examples are *Parallel Tag Clouds* for textual data [Collins et al., 2009], *Temporal Density Parallel Coordinates* for time-varying data [Johansson et al., 2007], and *Parallel Node-Link Bands* for multi-modal social networks [Ghani et al., 2013] (see Figure 3.1). However, visualization techniques for categorical data have not yet been covered explicitly.

Visualization techniques tailored to *categorical* datasets with additional properties have been presented in various contexts. These techniques span from time-oriented categorical data e.g., to study patient data over time [Monroe et al., 2013, Wongsuphasawat et al., 2011], to geospatial categorical data e.g., to study election results [Stoffel et al., 2012, Schulz et al., 2013]. There are two principal approaches to visualize categorical data: (1) converting categorical data to quantitative data and solve the problem in a different data space. (2) using visual representations specifically designed for categorical data, called explicit representations. Both variants have their benefits and each of them is suitable for specific visual analysis tasks. Fernstad & Johansson provide a guide to understand which visualization approach is most useful

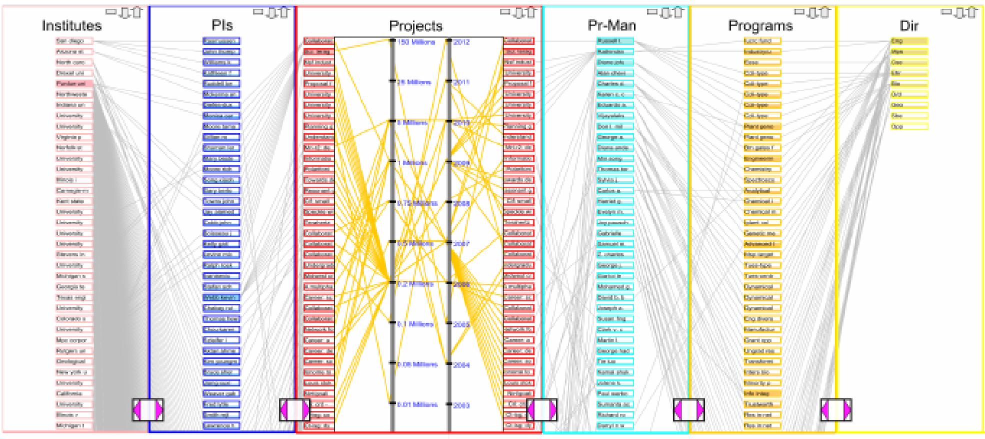


Figure 3.1: Parallel node-link bands visualizing multi-modal NSF funding data consisting of Institutions, PIs, Projects, program managers, NSF programs, and NSF directorates [Ghani et al., 2013].

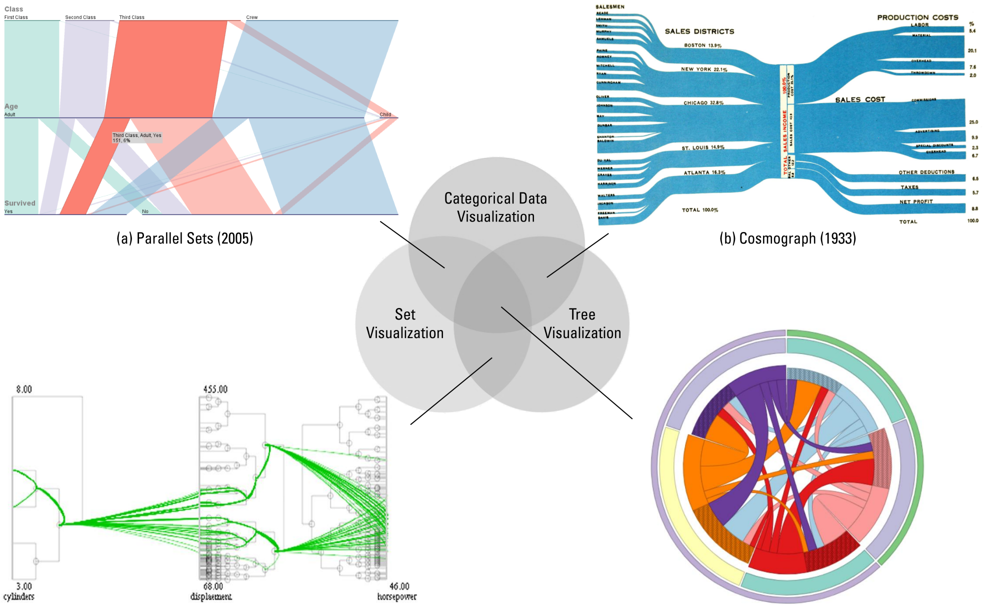
in the context of two basic data analysis tasks. Their results show that the quantification approach works better for tasks related to the similarity of data items, answering questions like, "Which two categories are most similar?". Eexplicit representation was found to work better for tasks related to the frequency of categories, addressing questions like, "Which category is the most common?" [Fernstad and Johansson, 2011].

Considering the data we are researching (see chapter 4.4.1) and after reviewing the related work, the necessity of finding a new data visualization with cross-sectional character becomes apparent: sets of data items are usually shown using set visualization [Freiler et al., 2008, Alsallakh et al., 2016], hierarchical structures are commonly displayed using tree visualization techniques [Rusu, 2013, Schulz, 2011], and quantitative aggregates and their distribution over various categories are mostly dealt with categorical data visualization [Blasius and Greenacre, 1998, Friendly, 2000]. Consequently, novel visualization techniques suitable for hierarchical categorical aggregates must incorporate multiple aspects from these approaches, as shown in Figure 3.2. In the following, we discuss representative visualization examples for different data combinations.

3.1.1 Visualizing Categorical Aggregates of Item Sets

Visualizations of this type show the pairwise frequency counts between a number of properties of an item set. What they lack is the ability to represent hierarchical structure. Alsallakh et al. review state-of-the-art techniques for set visualization [Alsallakh et al., 2016]. They classify these techniques into six main categories. In this work we are dealing with *aggregation-based techniques* from their taxonomy.

Parallel Sets [Bendix et al., 2005, Kosara et al., 2006] represent each property as an axis and connect the categories of neighboring axes with ribbons which width is proportional to the number of items that exhibit both categorical traits. In the example in Figure 3.2.a, the item set constitutes the people aboard the cruise ship *Titanic* and the axes denote their categorical properties, such as booked class, age group, and



(a) Parallel Sets (2005)

(b) Cosmograph (1933)

(c) Hierarchical Virtual Nodes (2016)

(d) Hierarchical Chord Diagram (2017)

Figure 3.2: An overview of the visualization techniques related to this work: a) Parallel Sets shows the Titanic dataset [Bendix et al., 2005], b) Cosmograph illustration on IBM business data, c) Hierarchical Virtual Nodes shows a dataset of cars [Huang et al., 2016], d) GiViP shows the communication among workers [Arleo et al., 2017].

whether they survived. This allows cross-tabulating the various categories to answer questions like “How did first class passengers fare compared to second class passengers?” However, this representation does not convey any hierarchical structure of the categories along the axes – for instance, only *adults* and *children* are distinguished in the age property, but no further drill down into the large adult category is possible. As a result, we cannot, for example, specifically investigate the fate of elderly passengers among the adults.

3.1.2 Hierarchical Visualization of Categorical Aggregates

Visualizations of this type show two or more hierarchies defined over various categories together with a numerical aggregate. However, an underlying item set is missing. As most data is available in the form of individual data items, visualizations of this type are rare.

One example is a visualization technique and device called *Cosmograph* [Brinton, 1939] that was marketed by IBM in the 1930s. It was designed to manually generate flow diagrams without the help of an “accomplished draughtsman” [Strickland, 2012]. Cosmograph shows hierarchically organized categories on each side – for instance, the salesmen grouped into sales districts on the left and the various costs aggregated into a simple hierarchy on the right in Figure 3.2.b. For each category, the numerical aggregate by which it contributes to the total income or cost is shown as a percentage. This allows identifying the salesmen and districts with high sales volumes, as well as the cost positions contributing most to the expenses. Yet, as it lacks the underlying set of individual sales transactions, we cannot actually cross-tabulate the categories. Only with the sales transactions – what was sold and its detailed production costs,

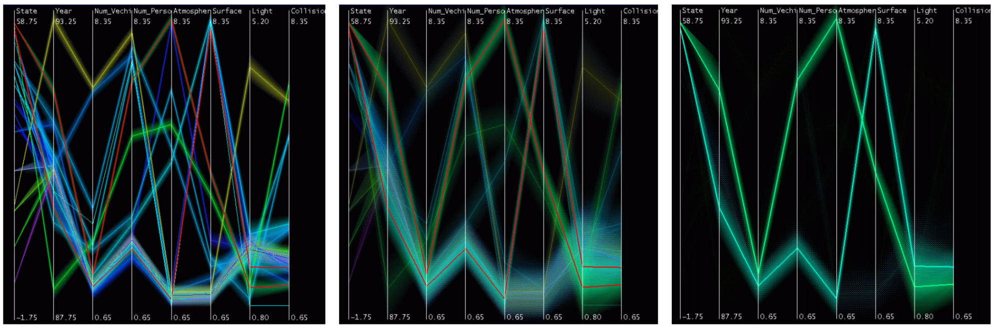


Figure 3.3: Hierarchical Parallel Coordinates show the Traffic Safety dataset without dynamic masking (left), partial fading (middle), and complete fading (right) [Fua et al., 1999].

how it was sold and its detailed sales costs, and who sold it where – we could be interested in relating the different categories to each other to find out, for example, “Which sales district incurs the highest sales costs?” This information cannot be read from the visualization because the flows are bundled in the center of the visualization, making it impossible to cross-tabulate categories from both sides.

3.1.3 Visualizing Item Sets and Their Hierarchical Properties

These visualizations show item sets distributed across hierarchical data properties. Yet, they lack a numerical aggregate that would quantify this distribution.

An example of this kind of visualization is *Hierarchical Virtual Nodes (HVN)* [Huang et al., 2016], which adds hierarchical displays to each axis of a Parallel Coordinates plot [Inselberg, 2009]. This way, each item from the dataset is displayed as a sequence of curves routing through the tree structure of each hierarchical axis. The example in Figure 3.2.c shows a dataset of cars and the drawn curves – one for each car – give a rough impression of which attribute values certain cars cluster around. A similar technique for textual data are *Parallel hierarchical Coordinates (PhC)* [Candan et al., 2012]. In some sense, HVN and PhC can be understood as generalizations of *Hierarchical Parallel Coordinates* (HPC) [Fua et al., 1999]: where HPC clusters the item set as a whole (as shown in Figure 3.3), so that navigating the cluster hierarchy steers the overall number of polylines across all axes, HVN and PhC do so on a per-axis basis. Yet, HVN does not convey the number of data items grouped in the hierarchically structured categories. In the example, this would mean, not only showing the cars as individual curves, but also showing how many there are and how they are distributed over other data properties.

3.1.4 Hierarchical Visualization of Categorical Set Aggregates

Visualizations of this type actually show all of the mentioned data aspects: the distribution of numerical aggregates of a set of data items over hierarchically organized categories. This visualizations are the closest to what we are looking for.

When only a few hierarchical data properties need to be visualized, the literature mentions *Hierarchical Chord diagrams* [Argyriou et al., 2014, Arleo et al., 2017] – an example of which is depicted in Figure 3.2.d. These diagrams basically extend a regular Chord diagram by showing an “inverted” *Sunburst* visualization [Stasko and Zhang, 2000] of the different hierarchies on the outside and connecting their categories with

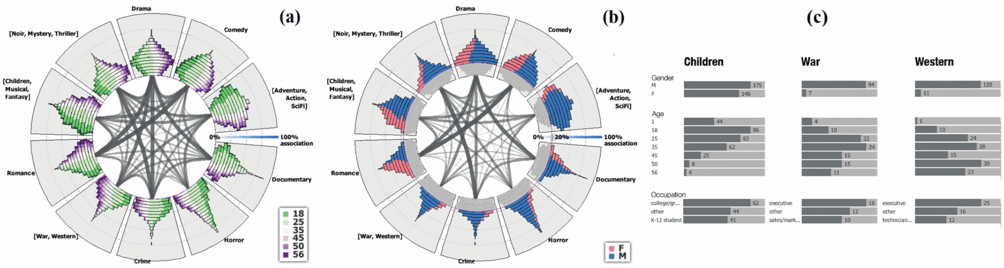


Figure 3.4: Contingency Wheel representing associations between users and movie genres (a) colored by age, (b) colored by gender, (c) details about selected genres [Alsallakh et al., 2012].

ribbons on the inside. This is similar to the *Contingency Wheel* [Alsallakh et al., 2012], with the wheel being a tree visualization, and Holten's radial tree visualization, with the bundled edges replaced by ribbons [Holten, 2006a]. Figure 3.4 shows an example of Contingency Wheel, designed to discover and analyze associations in contingency tables to find patterns in large categorical data. Note that the inversion of the Sunburst scheme into an "outside-in tree visualization" [Keahey et al., 2018] turns its inherent benefit into a drawback. Sunbursts grow outwards so that with every level more space is available on the circumference to show the increasing details of the hierarchy. However, drawing Sunbursts outside-in negates this effect so that there is actually less space available with every hierarchy level shown.

Hierarchical Chord diagrams are used as visual representations in some application domains, such as computer networking and life sciences. To the best of our knowledge, Hierarchical Chord diagrams were never formally introduced or evaluated as a visualization technique.

3.2 Uncertainty Visualization

This section reviews the related work on visualization techniques that apply to datasets containing uncertain values. The term *uncertainty* – also referred to as *data quality* – indicates the "degree to which the lack of knowledge about the amount of error is responsible for hesitancy in accepting results and observations" [Hunter and Goodchild, 1993]. In the visualization field, uncertainty is often considered as an additional dimension as it has to be visualized in addition to the primary values. Some research fields offer uncertainty visualizations for specific application scenarios and others suggest solutions to convey uncertainty to the general public. For example, Wittenbrink et al. suggest using glyphs to visualize uncertainty in the temporal or spatial domains [Wittenbrink et al., 1995]. Jackson proposes a shading technique for illustrating uncertainty [Jackson, 2008]. Moreover, Ehlschlaeger et al. show how animation could be used to depict uncertainty [Ehlschlaeger et al., 1997]. In the following two sections, we first discuss different taxonomies on visualizing uncertainty applied to different domains. Then, the current solutions applicable to flow diagrams – that are the main visualization focus of this research – will be shown.

3.2.1 Uncertainty Taxonomies

Various *taxonomies* to define uncertainty have been proposed. Taylor and Kuyatt categorize sources of uncertainty in four types: statistical, error, range, and scientific judgment [Taylor and Kuyatt, 1994]. All of these sources lead to the uncertainty of a value that needs to be visualized. In this thesis, we focus on uncertainty occurring in sampled data, in contrast to uncertainty caused by models or visualization processes.

Pang et al. classify uncertainty visualization methods, by considering discrete or continuous solutions for scalar, multivariate, vector, and tensor data types [Pang et al., 1997]. They introduce and apply seven new uncertainty visualization methods: *adding glyphs*, *adding geometry*, *modifying attributes*, *modifying geometry*, *animation*, *sonification*, and *psycho-visual* approaches. Also, Buttenfield and Ganter were among the first who created a framework for categorizing types of uncertainty, kinds of data, and methods of representation [Buttenfield and Ganter, 1990]. Their approach matches five categories of data quality with three data types: *discrete* (point and line features), *categorical* (area features assigned to categories or attributes assigned to classes), and *continuous* (surfaces and volumes), and then suggests the most appropriate visual variables (see section 2.2.1) to depict each category. Almost all of this early research was instigated and further developed by the geographic information system (GIS) community [Thomson et al., 2005, Gershon, 1998, MacEachren et al., 2005]. Subsequently, researchers from other areas such as the scientific visualization and information visualization investigated this topic [Gershon, 1998, Johnson and Sanderson, 2003]. Tak et al. propose three categories for techniques that can be used to visualize uncertainty in combination with the data [Tak et al., 2014].

Techniques from the **first** category vary *visual variables* such as color, size, blur, and transparency. One of the most common solutions to depict uncertainty is to use color [Aerts et al., 2003, MacEachren, 1992, Xie et al., 2006]. Another previous solution is blur – widely used to visualize fuzziness and ambiguity in data [MacEachren, 1992, Bisantz et al., 2002]. Correll et al. suggested a new approach to represent uncertainty along with the data on static charts called Value-Suppressing Uncertainty Palettes (VSUPs). The result of their evaluation showed that superimposed charts perform significantly better than juxtaposed. Moreover, charts with discrete bins performed much better than charts with continuous color maps as they cause more perceptual errors [Correll et al., 2018]. Moreover, Lodha et al. developed a method for ribbon flows using different numerical integration algorithms and different time steps to visualize uncertainty in fluid flow [Lodha et al., 1996].

The **second** category of techniques adds uncertainty information in the form of *glyphs, geometric features* such as contour lines and isosurfaces, labels, or icons. These techniques are similar to Wittenbrink et al.'s approach that uses glyphs for uncertainty in vector fields [Wittenbrink et al., 1996]. Figure 3.5.b shows various types of uncertainty glyphs designed so that meteorologists could visualize ensemble uncertainty. In this example the overall size indicates the maximum deviation of members from the mean value at a given grid location, and the small core represents few and large outliers. The results of their evaluation shows that this glyph technique is useful in assessing uncertainty – especially in finding outliers [Sanyal et al., 2010].

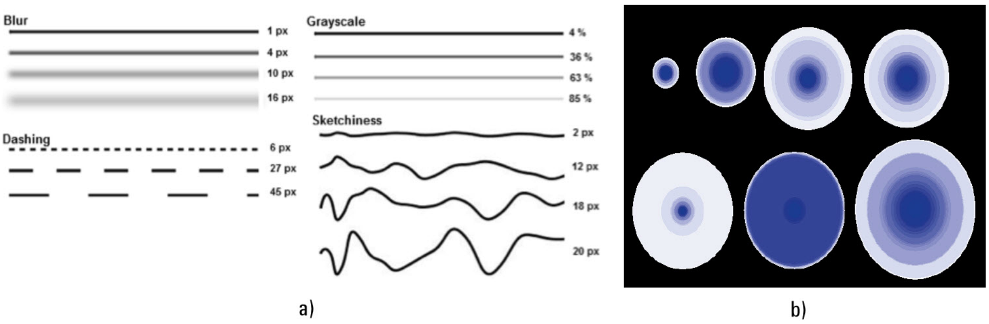


Figure 3.5: a) Four visual variables that are commonly used in the literature to depict uncertainty on line graphical primitives [Boukhelifa et al., 2012], b) Various types of graduated uncertainty glyphs generated from the 18 ensemble [Sanyal et al., 2010].

Finally, the **third** category from Tak et al. encompasses *animation* techniques to represent data uncertainty [Ehlschlaeger et al., 1997]. Although Tufte considers visual vibration as a cumbersome solution to users [Tufte, 1983, p. 111], Brown presents a visual vibration technique for presenting uncertain data [Brown, 2004].
In this work, we focus on the first category, showing uncertainty by varying the visual elements to give an overview of the uncertainty in the whole dataset. Techniques from the second category are more suitable when exact values need to be conveyed to the user. From the second category, we use the addition of geometric features to display uncertainty. Because of the drawbacks of animations, we do not consider techniques from the third category.

3.2.2 Uncertainty in Flow Diagrams

While research exists on visualizing uncertainty in tree structures [Lee et al., 2007,Griethe et al., 2006, Pang et al., 1997, Thomson et al., 2005, Streit et al., 2008] or node diagrams, the **graph** structure itself is altered. Hence, uncertainty is conveyed by the existence of nodes, edges, and edge attributes [Holten, 2006b, Eichelberger, 2003, Collins et al., 2007]. For instance, Hesse uses multiple levels of blurriness to express correlations between different nodes of one graph [Hesse, 2015]. Instead of applying blurriness to a single edge, a surrounding area is applied to the nodes, but this mainly indicates the strength of a relationship between two nodes in the data structure. Yet, little has been done to apply uncertainty to node attributes. There is no definitive technique for visualizing uncertainty using nodes and edges in graphs or network data [Cesario et al., 2011]. Hence, there is great potential to develop novel techniques that simultaneously visualize graph structural information and uncertainty values. Flow diagrams are also used to represent graph or network data (see section 2.3), and are the main focus of this research. Therefore, we investigate possible ways to convey uncertainty with flow diagrams, in a way that integrates the resulting visualization of data and uncertainty. This decision has been made based on user tasks and the importance of presenting an accurate depiction of the data to the user.

Flow diagrams have two main visual features as discussed in chapter 2.3.2: *nodes* and *flows*. In the following, we review publications that apply uncertainty to the main two visual features of flow diagrams. As the height of nodes and the thickness of flows

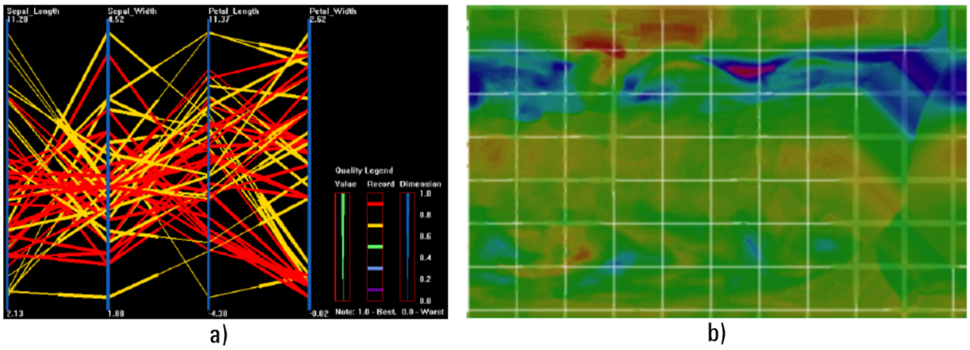


Figure 3.6: Different techniques to depict uncertainty by modifying lines: a) varying the color and width of Parallel Coordinates lines [Xie et al., 2006], b) applying width variation techniques to the annotation lines of the grid that overlays the data [Cedilnik and Rheingans, 2000].

represent the same value in flow diagrams related to this work, we decided to examine both features to convey the uncertainty of individual items.

Considering the links of graphs, Xie et al. address the visualization of structural information and uncertainty by varying the color and width of Parallel Coordinates lines [Xie et al., 2006]. Figure 3.6.a shows one example in which record quality is mapped to color, while value quality and dimension quality are both mapped to width of Parallel Coordinates polylines. Cedilinck and Rheingans propose a method to encode uncertainty information in the annotation lines of the grid which overlays the data [Cedilnik and Rheingans, 2000]. They use lines representing the graph edges to indicate uncertainty, and they use four techniques to distort annotation lines: *width variation, noise, sharpness*, and *amplitude modulated* distortion. Figure 3.6.b shows one example of width variation. Another inspiring work is shown in Figure 3.5.a Boukhelifa et al. studied four visual variables to depict uncertainty using line marks: *blur, dashing, grayscale*, and *sketchiness*. Their evaluation shows that blur is more intuitive. Yet, people prefer dashing over blur, grayscale, and sketchiness [Boukhelifa et al., 2012]. However, to the best of our knowledge, there is no work that applies uncertainty to the area (2D marks) instead of lines in graph structures.

Furthermore, recent work shows how to visualize uncertainty for rectangular bars that could be applicable to nodes of of flow diagrams. For example, Gschwandtnei et al. compare different representations of uncertainty for temporal data in the form of time intervals. They recommend using ambiguation (see Figure 3.7) that uses a lighter color value to represent uncertain regions for judging duration and temporal bounds technique. We picked this technique as one of the design approaches for nodes of flow diagrams. One of the most common encodings for sample means with associated error are bars in bar charts. Correll & Gleicher propose a set of alternatives designs to more effectively communicate uncertainty in bar charts (see Figure 3.8) [Correll and Gleicher, 2014]. They show how a simple redesign of error bars can improve user performance for a wide range of tasks. The design of error bars considers the semiotics of the visual display of uncertain data. Similarly, in this work, we show in chapter 6 how to redesign the nodes of flow diagrams to convey uncertainty.

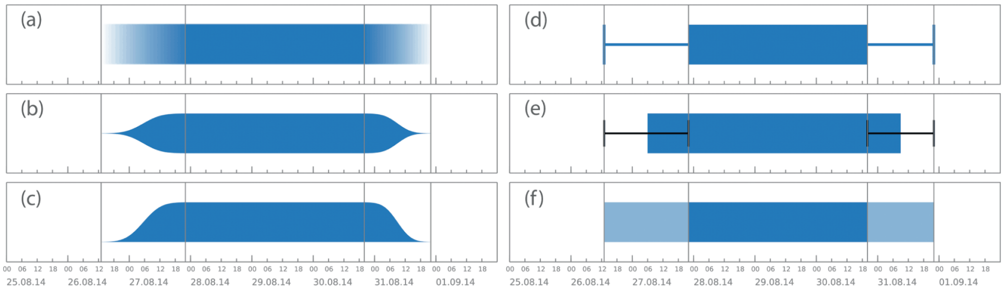


Figure 3.7: Six different visual encodings of start/end uncertainty of temporal intervals: a) gradient plot, b) violin plot, c) accumulated probability plot, d) error bars, e) centered error bars, and f) ambiguation [Gschwandtnei et al., 2016].

3.3 Time-Series Data Visualization

Time-series data are everywhere. Different phenomena change over time and analyzing those changes is important for different application domains such as business, medicine, science, demographics, and planning. Interactive visualization tools are useful to represent those changes in different domains for easy data exploration. Even long before computers appeared, different data visualizations were used to illustrate time-series data such as Minard's historical map from 1861 shown in section 2.3 (see Figure 2.17).

However, visualizing temporal datasets is still challenging, especially when several other dimensions are involved and because of the hierarchical structure of time with years, months, days, hours, minutes, and so forth [Aigner et al., 2011]. Therefore, most data visualization approaches are specifically designed for a particular analysis problem. Several taxonomies, conceptual models, and design spaces exist for temporal visualizations [Aigner et al., 2007,Andrienko et al., 2011,Andrienko and Andrienko, 2013]. Steiner introduced four categories of time-related datasets with respect to the valid and transaction time domains [Steiner, 1998]. Valid or internal time refers to time that is or will be recorded in real world. In contrast, transaction or external time indicates the recording time of a real world data in a database, as there is often a delay in the processing of information. Four temporal datasets can be introduced depending on the value changes of valid and transaction times (see Figure 3.9). Non-temporal databases capture the real world time at a certain state, while they record the changes of the database itself. However, static database capture the history of value changes with respect to the real-world. Understanding the differences between

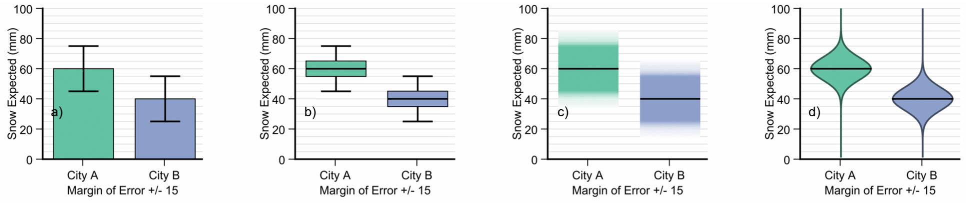


Figure 3.8: Four encodings for mean and uncertainty with bars charts: a) Bar chart, b) Modified box plot, c) Gradient plot, and d) Violin plot [Correll and Gleicher, 2014].

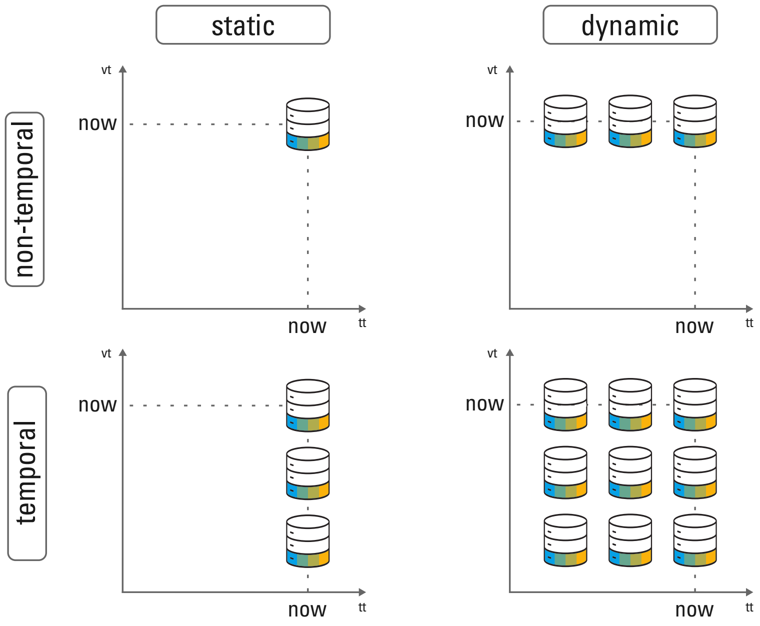


Figure 3.9: Temporal database taxonomy with respect to valid time (vt) and transaction time (tt), adapted from [Steiner, 1998].

those four temporal datasets is an important step towards designing an adequate visualization solution. In this work, we will focus on *static temporal* data which can be understood as the history of data with respect to the real world time changes.

Aigner et al. developed a systematic view on visualization methods for time-oriented data. They considered three main criteria of *time & data*, *user tasks*, and *visual representations* [Aigner et al., 2007]. The next section will discuss those criteria in details.

3.3.1 Time & Data

Time: The first criterion addresses the characteristics of the time axis. Aigner et al. present several design aspects for modeling time [Aigner et al., 2011] that need to be considered:

- *Scale:* three general domains are defined: the *ordinal* domain deals with only relative order relations (e.g., before, after), the *discrete* domain maps a set of integers to time values for quantitative modeling, and the *continuous* domain maps time points to a set of real numbers so that between each pair of time points another point can be defined.
- *Scope:* the second aspect divides the time elements into point-based and interval-based domains. In the point-based domain, the time points refer to single discrete time points (May 2019), in contrast to the interval-based domain that covers intervals or a period of time (May 1st 2019 - May 31st 2019).
- *Arrangement:* the third design aspect considers two temporary arrangements: either linear, where time proceeds from past to future, or cyclic that refers to a set of recurring time values such as seasons. Figure 3.10 shows two examples of Circos visualization showing both types of arrangements (left: linear, right: cyclic) [Krzywinski et al., 2009].

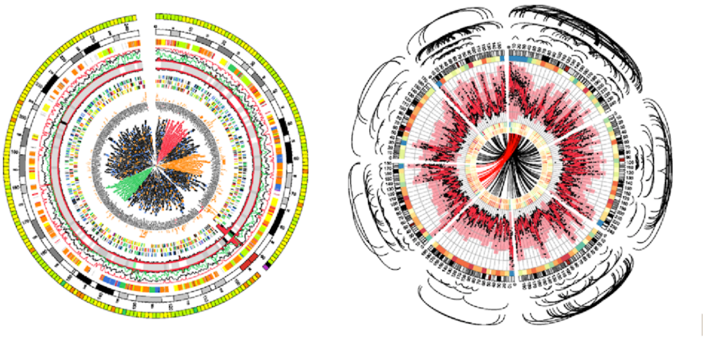


Figure 3.10: Circos tool represents human chromosome data using point plots, line plots, histograms, heat maps, tiles, connectors (right), and text. Derived from [Aigner et al., 2011].

- *Viewpoint:* the fourth aspect considers three views of time: *ordered*, *branching*, or *multiple perspectives*. The ordered time domain considers linear time. The branching time domain is relevant for planning or prediction, where only one branching path is considered. In contrast to branching time, multiple perspectives cover simultaneous views of time.

After reviewing the design aspects explained above, this work focuses on flow diagrams on *discrete, point-based, linear*, and *ordered* time.

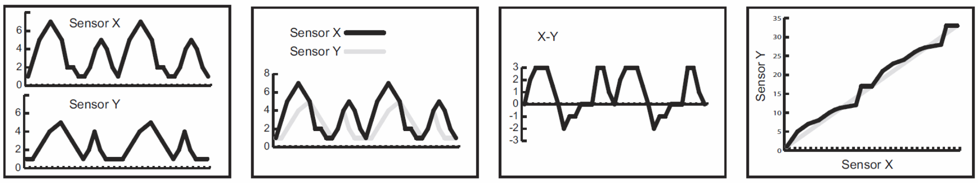
Data: To design adequate visualization approaches for time-oriented data, we first need to understand how data is bound to the time axis. The terms related to data are described in chapter 2.1.2.

- *Scale:* we only distinguish between quantitative (discrete or continuous) and qualitative (nominal or ordinal) data that are bound to time.
- *Frame of reference:* it is also critical to distinguish between spatial and abstract data. For abstract data, no spatial mapping is given, therefore, the spatial layout needs to be established first. In contrast, for spatial data, the given spatial information can be used to find an intuitive mapping of data to screen.
- *Number of variables:* refers to the number of time-dependent variables. They can be divided into univariate and multivariate data.
- *Kind of data:* refers to events or states that need to be visualized. States are the phases of continuity between events and events are state changes.

The focus of this thesis is on *quantitative, abstract, states*, and *multivariate* data.

3.3.2 User Tasks

As described in section 2.1.3, different tasks require different visualization solutions. Often when we talk about visualizing time-oriented data, the users' task is to analyze the data's temporal evolution. However, in our case, the task we are considering is rather different (see section 4.3.2). In the taxonomy of Andrienko & Andrienko it is considered as *comparison synoptic* task. Synoptic tasks consider sets of values in their entirety rather than individual elements [Andrienko and Andrienko, 2006, p. 119], and are further divided into descriptive and connectional tasks. Descriptive tasks are subdivided into lookup, comparison, and relation seeking tasks. Moreover, depending



a) Juxtaposition b) Superposition c) Explicit Encoding: Difference d) Explicit Encoding: Time Warp

Figure 3.11: A simple example of comparative visualizations: comparing two time series with three basic categories, which can also be combined [Gleicher et al., 2011].

on the type of comparison – corresponding to sets of references or sets of characteristics – direct and inverse comparison tasks are defined [Andrienko and Andrienko, 2006, p. 112].

Different taxonomies are suggested for comparison solutions, for instance, Graham and Kennedy investigate suitable task areas for different tree visualizations, varying from single trees to pair trees and multiple trees [Graham and Kennedy, 2010]. Pang et al. report on the importance of comparative visualization for fluid dynamics data and some possible solutions [Verma and Pang, 2004]. Figure 3.11 shows a general taxonomy proposed by Gleicher et al. based on a design strategy for visual comparison that categorizes all designs of comparative visualization into three basic categories: *juxtaposition*, *superposition*, and *explicit encoding*, which can also be combined [Gleicher et al., 2011]. However, each approach is suitable for specific tasks depending on data and problem domain.

3.3.3 Visual Representation

After defining the data input and specifying the problem domain, we need to determine how the data can be presented visually. We will consider only visualizations that apply to *hierarchical multi-dimensional* data, which is the focus of this thesis.
Some of the tree visualization solutions from section 2.1.4 can be also used for comparing hierarchical structures. The task of comparing multiple node-link diagrams has been often solved in two ways. The *first* approach uses these common tree representations and extends them in a way that can be used not only for exploration tasks but also for comparing two or more tree structures. However, the *second* approach focuses on designing completely new visualization solutions for specific scenarios

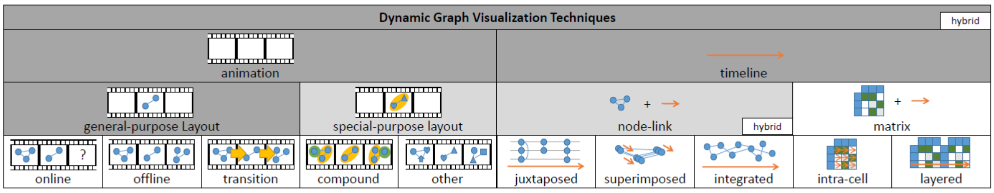


Figure 3.12: Illustrated hierarchical taxonomy of dynamic graph visualization techniques. The number of published techniques per taxonomic category is encoded in the brightness of the background [Beck et al., 2014].

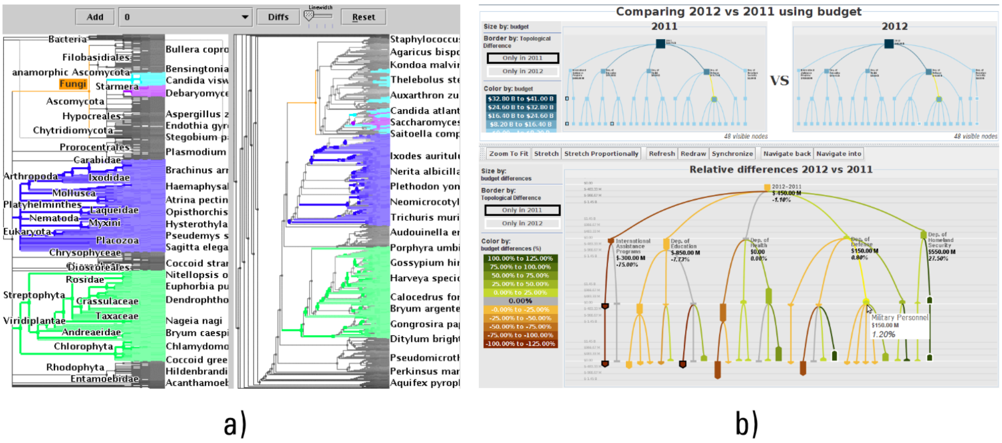


Figure 3.13: a) TreeJuxtaposer: a system for the structural comparison of large trees [Munzner et al., 2003], b) TreeVersity: Top: two original trees are compared (juxtaposition), Bottom: DiffTree shows the amount of change for each node (explicit encoding) [Guerra-Gómez et al., 2013].

like ActiviTree [Vrotsou et al., 2009] and Multiple Trees through DAG Representations [Graham and Kennedy, 2007]. Figure 3.12 shows Beck et al.'s taxonomy for dynamic graph visualization techniques [Beck et al., 2014]. They distinguish between *animation* and *timeline*, and within the timeline category, *juxtaposed, superimposed* and *integrated* approaches can be considered for node-link structures. However, In the integrated approaches, the graphs are interlinked and cannot be separated. Figure 3.13 shows two visualization approaches using juxtaposition: (a) TreeJuxtaposer and explicit encoding: (b) TreeVersity approach for comparing large trees. Yet, a key challenge in juxtaposition design is aiding the viewer in seeing the relationships between separate objects. TreeVersity overcomes this issue by using dual comparison techniques (side-by-side and explicit differences). It is an interactive solution that supports the detection of both node value changes and topological differences. However, it covers only the comparison of two hierarchical structures. Other solutions such as MultiTrees are capable of visualizing multiple hierarchies by merging them into a single graph which reuses the hierarchical substructures [Furnas and Zacks, 1994]. Yet, perceiving the structure of each individual hierarchy – which is of crucial importance in visualizing time-series data – is not straightforward with MultiTrees.

Another practical approach to compare two or multiple hierarchical structures is *linking of matches* applicable to both linear [Holten, 2006b] and circular [Meyer et al., 2009] layouts. Links are drawn between common tree leaf nodes and node sorting and edge bundling is often required for clutter reduction purpose. Holten et al. propose to bundle edges to show the relationships between trees by extending Icicle Plots [Holten, 2006b]. Both solutions extend an implicit representation to compare different versions of hierarchically organized software. It visualizes explicitly inter-hierarchy relations by placing two node-link representations facing each other with inter-hierarchy relations drawn between nodes. Also, Lex et al. introduce *Caleydo Matchmaker* to compare multiple, separately clustered groups of dimensions. Their primer contribution is a Focus+Context technique employing details-on-demand and drill-down capabilities [Lex

et al., 2010]. Another solution that extends the implicit presentations of trees are Contrast Treemaps [Tu and Shen, 2007] and Generalized Treemaps [Vliegen et al., 2006]. They visualize changes of hierarchical data by extending Treemaps.
In this work, we propose new approaches to extend our selected explicit visual representations (flow diagrams) for comparison of two or several hierarchical structures. In addition, we use both juxtaposition and superposition approaches for comparing the structures, depending on the size of the tree and dimensionality.

3.4 Summary

This chapter covered different data characteristics that are relevant to the problem domain considered in this thesis. Figure 3.14 shows those data characteristics and assigns an icon to each of them, which will be used in the following chapters. It also represents the gap we are planning to cover in the course of this thesis.

First, we described the hierarchical cross-tabulation problem. As we saw, despite many promising general visualization approaches, the state of the art does not provide a technique for interactive visualization of categorical hierarchical aggregates (see Figure 3.14). Because of different data scales and the requirement to combine the visualization with other underlying data characteristics (uncertainty and time-series), the few current available concepts such as Hierarchical Chord diagrams are not suitable. Therefore, we aim to bridge this gap and find a novel alternative to Hierarchical Chord diagrams for future applications. Our solution for categorical hierarchical aggregates will be described comprehensively in chapter 5.

Second, section 3.2 reviewed current techniques for representing data uncertainty. We summarized potential solutions for conveying uncertainty using visual features of flow diagrams. The rational for using these visual features is that they are already present in the visualization and hence, the distraction from the actual data visualization is minimal. Literature shows that when dealing with *categorical* datasets, techniques that *modify attributes* and *modify geometry* are the best approaches to convey uncertainty. Moreover, building upon the theoretical framework of Bertin on visual semiotics [Bertin, 1983], varying only a single visual variable at a time (e.g., color hue, fuzziness) is preferable. Also, visual variables that can be processed preattentively are put forward for consideration. As Figure 3.14 shows, there is no solution currently available to represent uncertainty on flow diagrams. Our solutions for conveying uncertainty will be presented in chapter 6.

Finally, in section 3.3 we looked at visualizing time-oriented data by systematically reviewing the three main questions that need to be answered: what is visualized? (data & time), why is it visualized? (user tasks), and how is it visualized? (visual representations). We put our focus on the comparison tasks and reviewed the current solutions to compare two or multiple tree structures over time. In chapter 7, we give an account of our solutions for the three visual comparison tasks in the context of time-series hierarchical data.

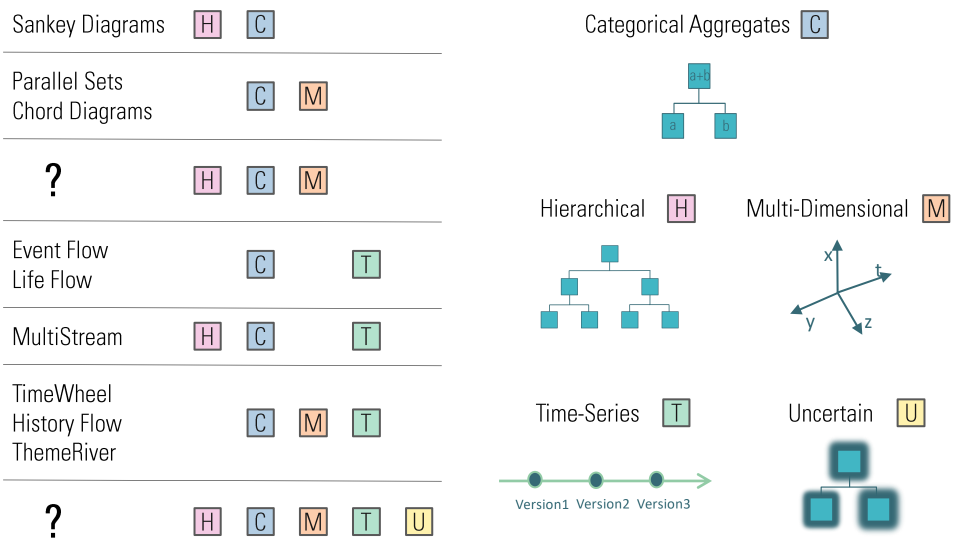


Figure 3.14: A summary of some ribbon-based visualization (flow diagrams) examples for categorical, hierarchical, multi-dimensional, uncertain, and time-series data along with the gaps we are planning to fill in.

In the following chapter, we introduce the research domain and the requirement engineering phase to define the users tasks and data characteristics. Consequently, we describe our solutions for each of the data aspects shown in Figure 3.14 – hierarchical categorical aggregates, uncertainty, times-series – discussed in this chapter.

Chapter 4

Requirement Engineering Phase

The process of identifying visualization requirements is an important part of every visualization researcher's and practitioner's activity. Munzner proposes splitting the visualization design process into the four cascading levels shown in Figure 4.1. At the first level, a specific domain situation needs to be defined in order to specify the users' target, their domain, their questions, and their data. This chapter covers the *first and second level of Munzner's nested model* to identify the domain situation and then to derive the domain-independent vocabulary [Munzner, 2014, p.68]. Parts of the research presented in this chapter have previously been published in [Vosough et al., 2016, Vosough et al., 2017a].

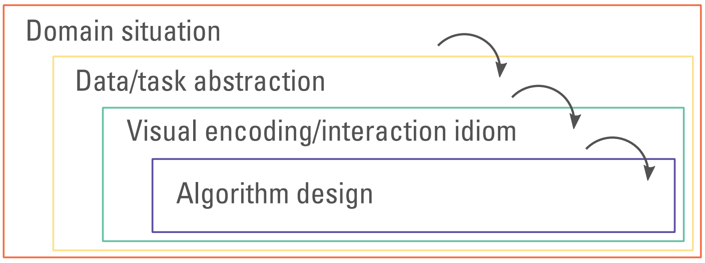


Figure 4.1: The four nested levels of visualization design by [Munzner, 2014, p.68].

4.1 Introduction

Requirements engineering is the process of identifying, specifying, and verifying requirements for a system under construction [Pohl, 2010]. Literature dedicated to requirements engineering in visualization usually assumes a *user-centered visualization design* process [Tory and Möller, 2004, Kulyk et al., 2006, Koh et al., 2011, Roberts et al., 2016]. Its result is produced through an iterative negotiation between the needs of the end users and the technical and representational possibilities available to the visualization designer. For this process to succeed, the literature stresses the importance of having direct access to end users who are willing to invest their time in the design process, as well as having access to the actual or at least realistic data [Sedlmair et al., 2012, Slingsby and Dykes, 2012]. Running this research in an industrial environment made it attainable for us to work jointly with end users of the target software in all different phases of the project. Involving customers from an early stage brings value to the project in different ways. First, customers know the market needs intimately and can share their knowledge and experiences. Second, the close collaboration with customers facilitates the collection of requirements and thus maximizes the fit of the solution to their needs through development cycles of design and validation. Less costly rework, higher customer satisfaction, and a readiness to respond to change are just some of the resulting benefits. However, often in industrial collaborations, visualization design gets “messy, iterative, and complex” [McKenna et al., 2014], as the visualization researcher has to deal with managerial hierarchies and issues of confidentiality in addition to the engineering challenges [Sedlmair et al., 2011, Crisan et al., 2016].

In this chapter, we report on this process, the information we gathered during it, and all complexities we encountered during our development of a visualization solution for the SAP Product Lifecycle Costing application. In the following, we will first introduce the environment this research was conducted in. Then we introduce the case study and retrace the steps we took to establish visualization requirements from the initial idea to the first prototypes. In addition, we report on the problems encountered during these steps and how we resolved, circumvented, or mitigated them. Finally, we abstract our results to a generic form in order to support domains other than product costing.

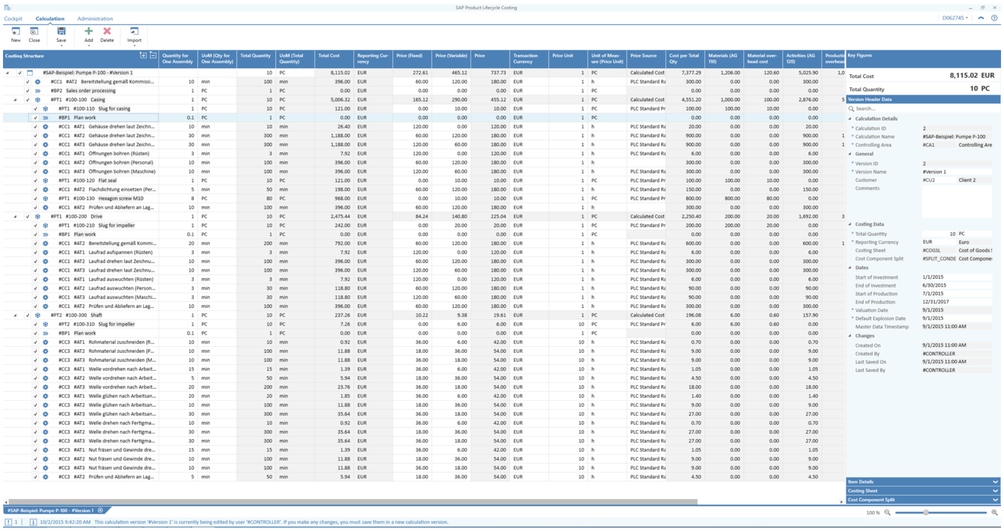
4.2 Environment

This research project started at SAP SE with the purpose of designing and developing a novel data visualization for a new standard business software called SAP Product Lifecycle Costing. In the following, we describe the product, our interaction strategy with co-innovation customers, as well as the development processes for this product.

4.2.1 The Product

SAP Product Lifecycle Costing is a solution to calculate costs for new products or quotations, to quickly identify cost drivers and to easily simulate and compare alternatives. SAP Product Lifecycle Costing subsumes all methods for estimating and optimizing the costs a product incurs over its lifetime – from its initial design, to manufacturing, using, maintaining, and retiring it. The idea behind this procedure is to reduce costs as early as possible, as the bulk of a product’s total cost is determined by early design decisions, such as which parts and materials to use [Asiedu and Gu, 1998]. SAP Product Lifecycle Costing was developed in close collaboration with customers and partners over a period of five years, and development of new releases is ongoing.
Figure 4.2 shows a screen-shot of the application’s user interface. Similar to many other business intelligence applications, the user interface is spreadsheet-based.

The project team is comprised of two sub-teams. The go-to-market team is responsible for establishing and maintaining customer relationships, for marketing and for supporting sales. The development team led by the chief product owner develops the software. Requirements analysis and functional design is shared between the two teams and is conducted in close cooperation with customers. The development team is organized into four *scrum* teams of about 10 developers each [Schwaber, 1997].



Costing Structure	Quantity for One Assembly	UoM (Qty for One Assembly)	Total Quantity	UoM (Total Quantity)	Total Cost	Reporting Currency	Price (Fixed)	Price (Variable)	Price	Transaction Currency	Price Unit	Unit of Measure (Price Unit)	Price Source	Cost per Total Qty	Materials (AG 110)	Material overhead cost	Activities (AG 120)	Productio overhead
#SAP-Beispiel: Pumpe P-100 - #Version 1			10	PC	8,115.02	EUR	272.61	465.12	737.73	EUR	1	PC	Calculated Cost	7,377.29	1,206.00	120.60	5,025.90	1,0
#CC1 #AT2 Bereitstellung gemäß Kommissi...	10	min	100	min	396.00	EUR	60.00	120.00	180.00	EUR	1	h	PLC Standard Ra	300.00	0.00	0.00	300.00	
#BP2 Sales order processing	1	PC	1	PC	0.00	EUR	0.00	0.00	0.00	EUR	1	PC		0.00	0.00	0.00	0.00	
#PT1 #100-100 Casing	1	PC	10	PC	5,006.32	EUR	165.12	290.00	455.12	EUR	1	PC	Calculated Cost	4,551.20	1,000.00	100.00	2,876.00	5
#PT1 #100-110 Slug for casing	1	PC	10	PC	121.00	EUR	0.00	10.00	10.00	EUR	1	PC	PLC Standard Pr	100.00	100.00	10.00	0.00	
#BP1 Plan work	0.1	PC	1	PC	0.00	EUR	0.00	0.00	0.00	EUR	1	PC		0.00	0.00	0.00	0.00	
#CC1 #AT1 Gehäuse drehen laut Zeichn...	10	min	10	min	26.40	EUR	120.00	0.00	120.00	EUR	1	h	PLC Standard Ra	20.00	0.00	0.00	20.00	
#CC1 #AT2 Gehäuse drehen laut Zeichn...	30	min	300	min	1,188.00	EUR	60.00	120.00	180.00	EUR	1	h	PLC Standard Ra	900.00	0.00	0.00	900.00	1
#CC1 #AT3 Gehäuse drehen laut Zeichn...	30	min	300	min	1,188.00	EUR	120.00	60.00	180.00	EUR	1	h	PLC Standard Ra	900.00	0.00	0.00	900.00	1
#CC1 #AT1 Öffnungen bohren (Rüsten)	3	min	3	min	7.92	EUR	120.00	0.00	120.00	EUR	1	h	PLC Standard Ra	6.00	0.00	0.00	6.00	
#CC1 #AT2 Öffnungen bohren (Personal)	10	min	100	min	396.00	EUR	60.00	120.00	180.00	EUR	1	h		300.00	0.00	0.00	300.00	
#CC1 #AT3 Öffnungen bohren (Maschine)	10	min	100	min	396.00	EUR	120.00	60.00	180.00	EUR	1	h	PLC Standard Ra	300.00	0.00	0.00	300.00	
#PT1 #100-120 Flat seal	1	PC	10	PC	121.00	EUR	0.00	10.00	10.00	EUR	1	PC	PLC Standard Pr	100.00	100.00	10.00	0.00	
#CC1 #AT2 Flachdichtung einsetzen (Per...	5	min	50	min	198.00	EUR	60.00	120.00	180.00	EUR	1	h	PLC Standard Ra	150.00	0.00	0.00	150.00	
#PT1 #100-130 Hexagon screw M10	8	PC	80	PC	968.00	EUR	0.00	10.00	10.00	EUR	1	PC	PLC Standard Pr	800.00	800.00	80.00	0.00	
#CC1 #AT2 Prüfen und Abliefern an Lag...	10	min	100	min	396.00	EUR	60.00	120.00	180.00	EUR	1	h		300.00	0.00	0.00	300.00	
#PT1 #100-200 Drive	1	PC	10	PC	2,475.44	EUR	84.24	140.80	225.04	EUR	1	PC	Calculated Cost	2,250.40	200.00	20.00	1,692.00	3
#PT1 #100-210 Slug for impeller	1	PC	10	PC	242.00	EUR	0.00	20.00	20.00	EUR	1	PC	PLC Standard Pr	200.00	200.00	20.00	0.00	
#BP1 Plan work	0.1	PC	1	PC	0.00	EUR	0.00	0.00	0.00	EUR	1	PC		0.00	0.00	0.00	0.00	
#CC1 #AT2 Bereitstellung gemäß Kommi...	20	min	200	min	792.00	EUR	60.00	120.00	180.00	EUR	1	h	PLC Standard Ra	600.00	0.00	0.00	600.00	1
#CC1 #AT1 Laufrad aufspannen (Rüsten)	3	min	3	min	7.92	EUR	120.00	0.00	120.00	EUR	1	h	PLC Standard Ra	6.00	0.00	0.00	6.00	
#CC1 #AT2 Laufrad drehen laut Zeichnu...	10	min	100	min	396.00	EUR	60.00	120.00	180.00	EUR	1	h	PLC Standard Ra	300.00	0.00	0.00	300.00	
#CC1 #AT3 Laufrad drehen laut Zeichnu...	10	min	100	min	396.00	EUR	120.00	60.00	180.00	EUR	1	h	PLC Standard Ra	300.00	0.00	0.00	300.00	
#CC1 #AT1 Laufrad auswuchten (Rüsten)	3	min	3	min	7.92	EUR	120.00	0.00	120.00	EUR	1	h	PLC Standard Ra	6.00	0.00	0.00	6.00	
#CC1 #AT2 Laufrad auswuchten (Person...	3	min	30	min	118.80	EUR	60.00	120.00	180.00	EUR	1	h	PLC Standard Ra	90.00	0.00	0.00	90.00	
#CC1 #AT3 Laufrad auswuchten (Maschi...	3	min	30	min	118.80	EUR	120.00	60.00	180.00	EUR	1	h	PLC Standard Ra	90.00	0.00	0.00	90.00	
#CC1 #AT2 Prüfen und Abliefern an Lag...	10	min	100	min	396.00	EUR	60.00	120.00	180.00	EUR	1	h	PLC Standard Ra	300.00	0.00	0.00	300.00	
#PT2 #100-300 Shaft	1	PC	10	PC	237.26	EUR	10.22	9.38	19.61	EUR	1	PC	Calculated Cost	196.08	6.00	0.60	157.90	
#PT2 #100-310 Slug for impeller	1	PC	10	PC	7.26	EUR	0.00	6.00	6.00	EUR	10	PC	PLC Standard Pr	6.00	6.00	0.60	0.00	
#BP1 Plan work	0.1	PC	1	PC	0.00	EUR	0.00	0.00	0.00	EUR	1	PC		0.00	0.00	0.00	0.00	
#CC3 #AT1 Rohmaterial zuschneiden (R...	10	min	10	min	0.92	EUR	36.00	6.00	42.00	EUR	10	h	PLC Standard Ra	0.70	0.00	0.00	0.70	
#CC3 #AT2 Rohmaterial zuschneiden (P...	10	min	100	min	11.88	EUR	18.00	36.00	54.00	EUR	10	h	PLC Standard Ra	9.00	0.00	0.00	9.00	
#CC3 #AT3 Rohmaterial zuschneiden (M...	10	min	100	min	11.88	EUR	36.00	18.00	54.00	EUR	10	h	PLC Standard Ra	9.00	0.00	0.00	9.00	
#CC3 #AT1 Welle vordrehen nach Arbeit...	15	min	15	min	1.39	EUR	36.00	6.00	42.00	EUR	10	h	PLC Standard Ra	1.05	0.00	0.00	1.05	
#CC3 #AT2 Welle vordrehen nach Arbeit...	5	min	50	min	5.94	EUR	18.00	36.00	54.00	EUR	10	h	PLC Standard Ra	4.50	0.00	0.00	4.50	
#CC3 #AT3 Welle vordrehen nach Arbeit...	20	min	200	min	23.76	EUR	36.00	18.00	54.00	EUR	10	h	PLC Standard Ra	18.00	0.00	0.00	18.00	
#CC3 #AT1 Welle glühen nach Arbeitsan...	20	min	20	min	1.85	EUR	36.00	6.00	42.00	EUR	10	h	PLC Standard Ra	1.40	0.00	0.00	1.40	
#CC3 #AT2 Welle glühen nach Arbeitsan...	10	min	100	min	11.88	EUR	18.00	36.00	54.00	EUR	10	h	PLC Standard Ra	9.00	0.00	0.00	9.00	
#CC3 #AT3 Welle glühen nach Arbeitsan...	30	min	300	min	35.64	EUR	36.00	18.00	54.00	EUR	10	h	PLC Standard Ra	27.00	0.00	0.00	27.00	
#CC3 #AT1 Welle drehen nach Fertigma...	10	min	10	min	0.92	EUR	36.00	6.00	42.00	EUR	10	h	PLC Standard Ra	0.70	0.00	0.00	0.70	
#CC3 #AT2 Welle drehen nach Fertigma...	30	min	300	min	35.64	EUR	18.00	36.00	54.00	EUR	10	h	PLC Standard Ra	27.00	0.00	0.00	27.00	
#CC3 #AT3 Welle drehen nach Fertigma...	30	min	300	min	35.64	EUR	36.00	18.00	54.00	EUR	10	h	PLC Standard Ra	27.00	0.00	0.00	27.00	
#CC3 #AT1 Nut fräsen und Gewinde dre...	15	min	15	min	1.39	EUR	36.00	6.00	42.00	EUR	10	h	PLC Standard Ra	1.05	0.00	0.00	1.05	
#CC3 #AT2 Nut fräsen und Gewinde dre...	10	min	100	min	11.88	EUR	18.00	36.00	54.00	EUR	10	h	PLC Standard Ra	9.00	0.00	0.00	9.00	
#CC3 #AT3 Nut fräsen und Gewinde dre...	10	min	100	min	11.88	EUR	36.00	18.00	54.00	EUR	10	h	PLC Standard Ra	9.00	0.00	0.00	9.00	
#CC3 #AT2 Prüfen und Abliefern an Lag...	5	min	50	min	5.94	EUR	18.00	36.00	54.00	EUR	10	h	PLC Standard Ra	4.50	0.00	0.00	4.50	

Key Figures

Total Cost 8,115.02 EUR

Total Quantity 10 PC

Version Header Data

Search...

Calculation Details
- * Calculation ID: 2
- * Calculation Name: #SAP-Beispiel: Pumpe P-100
- * Controlling Area: #CA1 Controlling Are

General
- * Version ID: 2
- * Version Name: #Version 1
- Customer: #CU2 Client 2
- Comments:

Costing Data
- * Total Quantity: 10 PC
- * Reporting Currency: EUR Euro
- Costing Sheet: #COGSL Cost of Goods !
- Cost Component Split: #SPLIT_CONDE Cost Componei

Dates
- Start of Investment: 1/1/2015
- End of Investment: 6/30/2015
- Start of Production: 7/1/2015
- End of Production: 12/31/2017
- * Valuation Date: 9/1/2015
- * Default Explosion Date: 9/1/2015
- Master Data Timestamp: 9/1/2015 11:00 AM

Changes
- Created On: 9/1/2015 11:00 AM
- Created By: #CONTROLLER
- Last Saved On: 9/1/2015 11:00 AM
- Last Saved By: #CONTROLLER

Item Details

Costing Sheet

Cost Component Split

#SAP-Beispiel: Pumpe P-100 - #Version 1

1 | 10/2/2015 9:42:20 AM This calculation version '#Version 1' is currently being edited by user '#CONTROLLER'. If you make any changes, you must save them in a new calculation version.

100 %

Figure 4.2: A screen-shot of the UI of SAP Product Lifecycle Costing application.

Each of the four scrum teams has its own product owner and scrum master, taking full responsibility for developing agreed-upon functionalities (user stories and backlog items). In addition, several colleagues have cross-team responsibilities, such as developing the business architecture, technical architecture, writing documentation, testing functional correctness, usability, and performance. Two of the scrum teams are based in Germany and two in Romania.

4.2.2 The Customers and Development Methodology

More than 30 co-innovation customers have been working with the team and helped established requirements. The initial trigger for developing a standard software for early product costing was a request from an SAP customer. After this, two groups of co-innovation customers were formed. One group of German customers comprising companies from the automotive industry as well as industrial machinery and a second group of US-American companies from different industries. Since SAP Product Lifecycle Costing has become generally available in September 2015, new software releases are produced approximately every quarter. In the same quarterly rhythm, regular co-innovation workshops are conducted with the German customer group. Every workshop lasts one and a half days. On the first day typically an overview of the newly available functionalities are given and then a usability test is run. Usually, six to seven stations are set up where three end-users each (mostly highly experienced product controllers or IT experts) from different companies have to perform a set of predefined tasks. The team observes these sessions and takes notes of noteworthy incidents [Morgan, 1996], positive or negative. In the end, every group summarizes their impressions in the plenum (usually around 50 participants including customers, partners and representatives from the go-to-market team, and the development team). On the second day of the workshop, requirements are discussed jointly rough design sketches are produced. To this end, the *Design Thinking* methodology is exploited. Similar workshops are conducted with the US-American co-innovation group, but only once or twice per year in a joint location, and as a shorter online meeting otherwise.

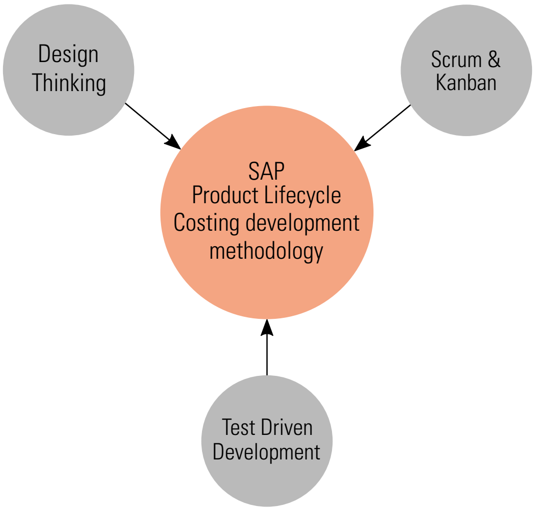


Figure 4.3: The SAP Product Lifecycle Costing development methodology is a mix of agile methodologies.

This co-development process gave us a distinctive opportunity to design, develop, and evaluate the visualization solution early on with the co-innovation customers.

Methodology

The development methodology of the product was a mix of established best-practices such as *Design Thinking*, *Scrum*, and *Test-Driven Development* (see Figure 4.3) as well as several custom adjustments and optimizations.

Design Thinking (DT) is a collaborative problem solving approach involving people with different backgrounds [Plattner et al., 2010]. DT advocates the fast generation of ideas and prototypes with the goal of designing a solution that is feasible ("it can be done"), viable ("it can be sold"), and desirable ("users will love it"). In our co-innovation workshops, we used a DT-like approach for eliciting requirements from end-users and for designing rough solution sketches. Every DT session starts with a design challenge. Often, customers explain how their processes are run today and mention related challenges. Then, after some silent "brain-writing" (typically 5-7 min of writing ideas on sticky notes) every participant presents his ideas to the group (typically no more than 8-10 people). Others may ask, comment, and extend the ideas but cannot "shoot them down". Finally, all input is grouped and prioritized and a joint prototype is developed on a white board. Also, paper, pens, and any other materials may be used for developing mockups. We explicitly ask every participant to refrain from using their computers and phones during the sessions to avoid distractions.

The advantage of discussing requirements, priorities, and designs with a group instead of individually is that it is easier to reach agreements. When developing standard software (instead of customer-specific software), the development team must balance the requirements of all customers. The problem is that they are almost never the same. Compromises are inevitable but are easier to attain if the discussion is between customers rather than between one customer and the development team. After the workshop, low-fidelity and high-fidelity UI mockups are designed for the new functionality based on ideas and prototypes developed during the workshop. These designs are usually presented at the next customer workshop and customers give final feedback before implementation starts. Typically, this requires only minor adjusments to the design. However, we rarely had a session where the designs were accepted without change requests. The implementation begins after this feedback. Before a software release is shipped to our customers, we have three weeks of integration testing and final bug fixing. After the go to market team and SAP-internal consultants have tested all features, our partners and some of our co-innovation customers do the same during the so-called *Acceptance Testing*. The same methodology is used for the visualization project of this thesis, as this iterative feedback process ensures that functionality addresses the requirements.

4.2.3 Lessons Learned

During the last years of co-development with customers, the team learned many lessons. Most importantly, it was learned that there is no *one-size-fits-all* process. While the development team grew from only a handful of developers to four fully staffed teams-of-ten, the processes had to be constantly adjusted, mixing established *best practices* like scrum with their own optimizations. The team has done both, under-managed and over-managed the development. When the project started, the agile development methodology was followed without long-term planning. This did not scale beyond the number of people that fit into a small room. Later, the number of meetings was increased to the point that the perceived time remaining for development was less than that used for communication. Therefore, our meeting schedule was changed once again, to a more sustainable level. Communication and synchronization, once being the source of most pain, got much easier after these adoptions. Continuous learning is key for the team, although it takes time. When junior colleagues joined the team, much effort was spent for knowledge-transfer through self-paced online courses, class-room trainings, mentoring, and learning by doing (e.g. fixing simple bugs). In hindsight, this time was well invested, yet underestimated in the beginning. Usually it takes many months and even years before a junior software developer reaches the level of productivity of a senior developer. Effort estimation proved to be another challenge, which took much time to get under control. Especially in the beginning of the development, effort estimates often were off the mark by a factor of 2 or 3. We learned that a good effort estimate requires a detailed design. Also, we now routinely multiply our estimates for not development days with a factor that covers design, tests, and unexpected. All estimations are done jointly by the product owners (senior developers), which further improves their accuracy.

Our focus on co-innovation with customers proved to be highly successful. All of our developers met and talked to our customers personally. This has proved to be highly beneficial not only for understanding but also for feeling customer needs. Consequently, the team routinely receives highest marks when asking our customers about their level of trust and satisfaction.

4.3 Visualization Requirements for Product Costing

This research started with a requirement engineering phase in an industrial setting to introduce visualization to the field of product costing. As we explained above, the process of *co-innovation* allows for discussing experimental features and getting early feedback in a user and task-based design approach [Tory and Möller, 2004]. The visualization requirements were established in four steps: current visualization practice, visualization tasks, characteristics of the visualized data, and further detailed requirements by means of discussing first prototypes. The following subsections will briefly outline our procedure and results for each of the four steps.

4.3.1 Current Visualization Practice

Procedure: Our first step was to understand our customers' current solutions and needs with respect to visualization support. Note that this step is often neglected, because most projects start directly with task and data requirements. We decided to invest extra time to make sure that our research is not based on false assumptions of a need for visualization where there is actually none [Munzner, 2009]. To that end, we conducted a survey with 21 co-innovation customers. They were asked broadly, which product costing solutions they currently employ, which data visualizations they use, as well as for which parts of the costing analysis process they use data visualization. The survey used for this interview can be found in the appendix C. The online survey was sent to the customers via e-mail and we noticed only limited contribution of our customers in online surveys despite our regular e-mail reminders. Therefore, the survey results were complemented with personal interviews conducted with customers at co-innovation workshops.

Results: The collected answers confirmed our subjective experiences that most customers currently use spreadsheet solutions and the common visualization techniques offered by them – predominantly pie charts and bar charts – for costing analyses. Yet, the charts are utilized almost exclusively at the end of the analysis process for reporting and presentation purposes – the only exception being visual comparisons that are occasionally performed during the analysis. While spreadsheets and their associated visualizations match the needs of tabular data analysis, cost analyses involve more intricate tasks. For example, choosing a certain material for a product may make it cheaper to buy, but more delicate to handle, and thus produces more defective goods during manufacturing. Tracing such dependencies and observing the repercussions of different design choices can hardly be captured in a tabular form. Hence, we saw potential for a novel visualization tailored to product costing. As we show in chapter 5.4, this novel visualization is applicable and generalizes to other application domains.

4.3.2 Visualization Tasks

Procedure: To substantiate our hypothesis that parts of the cost analysis are ill-suited to be pursued with a tabular data display alone, we conducted hour-long group discussions with a total of 30 participants from 16 companies during co-innovation workshops in Germany and in the US. The discussions focused on the end users' tasks that are perceived as complex, that take a long time to perform, or that are currently simply impossible to carry out, as these tasks could potentially benefit from visualization support. Note that while the scientific literature abounds with abstract visualization tasks and taxonomies, there is little research on how to conduct a visualization task analysis. We used the general recommendations given in [Kulyk et al., 2006] as orientation.

Results: From these discussions, 15 tasks were gathered, and the following four tasks are what all participants agreed to:

Task 1: Identify the main cost drivers by comparing multiple cost calculations with each other, so as to gauge the impact of adding or removing individual items or assemblies on the overall costs.

Task 2: See how close the calculation is to a defined cost target and which assemblies are above or below targets – including prognostics, i.e., which assemblies get less or more costly over time.

Task 3: Determine incomplete or inconsistent cost calculations. For example to find missing prices and see where prices have been estimated instead of being derived from reliable master data.

Task 4: Assess the reliability of the overall cost calculated from price sources with different confidence levels. Determine best, realistic, and worst cases based on projections for future costs.

4.3.3 Data Structure and Size

Procedure: To better understand the datasets on which our customers operate, a survey was conducted with 12 customers. Again, while there exists a number of different papers on metadata, dataspace notations, and data descriptors for visualization, literature on the best practices for collecting this information from users is limited. To shape our survey, we took inspiration from two studies conducted previously on enterprise data analysis [Kandel et al., 2012, Kandogan et al., 2014].

Results: A first observation is that the costing data structure is *hierarchical* and *additive*, as the overall product cost is basically the sum of the parts' costs, and recursively for sub-parts, and finally raw materials and labor. Furthermore, the product cost consists of several calculations, with each calculation existing in several versions. These Calculation Versions (CV) are used to take different scenarios into account, such as optimistic versus pessimistic price dynamics, so that the cost development of a product can be projected into the future and factored into the analysis. As it can be seen

	Min	Typical	Max
No. calculations	10	100	300K
No. CV per calculation	1	5	120
Size of a CV	20	1K	900K
Tree depth	3	5	20
No. materials	40K	400K	3Mio

Table 4.1: Product costing data sizes as surveyed from 12 different SAP customers.

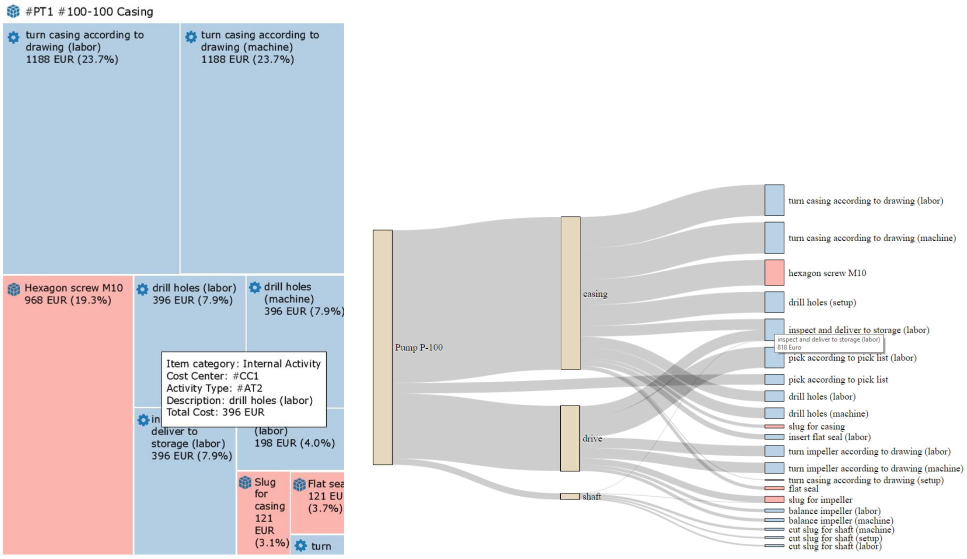
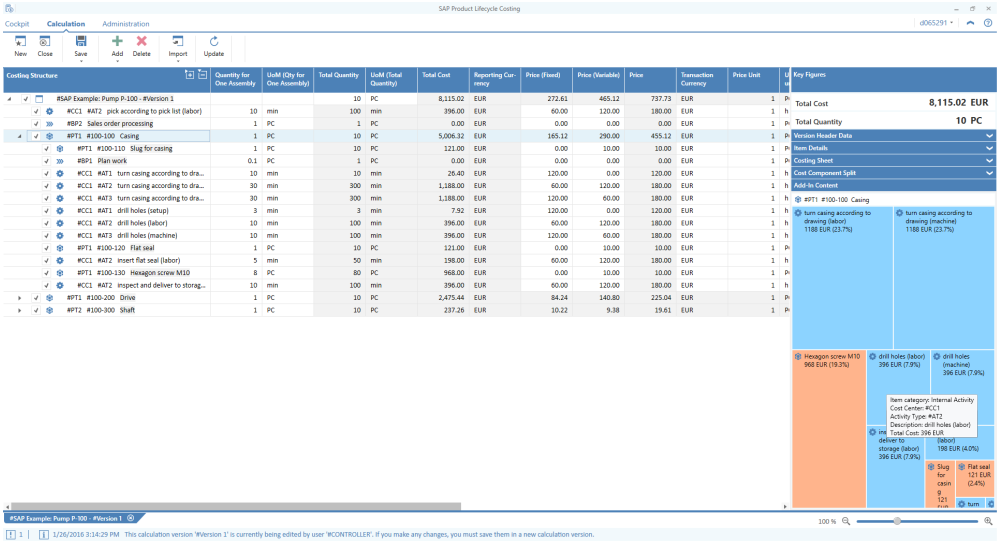


Figure 4.4: Early visualization prototypes by: left) Treemap, right) Sankey diagram. Both figures show a small dataset of an industrial pump. The blue components represent activities, whereas the pink components indicate materials.

from Table 4.1, the variety of data sizes among the customers is substantial. For example, the size of a calculation version can vary from 20 to 900K, which makes it very challenging to find a generic solution for all customers. The reason is that, some companies build new products based on previous product versions and thus they can analyze their product costs by slightly changing calculation version for each new product. Whereas other companies build completely different products and thus have to add entirely new costing structures with new calculations for each product.

4.3.4 Early Visualization Prototypes

Procedure: To deepen the discussion with the customers, we needed to produce something tangible that allowed to break free from bar charts and pie charts. Building early prototypes [Slingsby and Dykes, 2012, Koh et al., 2011] and mock-ups [Roberts et al., 2016] are common strategies in visualization design, where end users from application domains often decide on visualizations in an “I-know-it-when-I-see-it” manner. Hence, we developed two prototypes shown in Figure 4.4 for substantiating the discussion of task, data, and representation: a Squarified Treemap [Bruls et al., 2000] and a Sankey diagram [Riehmann et al., 2005]. The Treemap is an obvious choice for attribute-centric tasks, such as looking at cost drivers (Task 1) or target costs (Task 2), while still keeping the costs hierarchically organized through the layout. As Treemaps are known to scale quite well [Fekete and Plaisant, 2002], they are a good fit for the wide range of tree sizes specified by the customers. The Sankey diagram is more geared towards structure-centric tasks, such as tracing inconsistencies to their origin (Task 3) or analyzing the propagation of reliability scores through the hierarchical cost structure (Task 4), as it provides a clear view of the costing structure. Both visualization prototypes were outfitted with basic interactions, such as folding/unfolding hierarchical branches and tooltips for detailed information. We discussed both prototypes in an informal setting at two customer workshops.


SAP Product Lifecycle Costing
Cockpit
Calculation
Administration
New
Close
Save
Add
Delete
Import
Update
d065291
Version Header Data
Item Details
Costing Sheet
Cost Component Split
Add-In Content
#PT1 #100-100 Casing
turn casing according to drawing (labor) 1188 EUR (23.7%)
turn casing according to drawing (machine) 1188 EUR (23.7%)
Hexagon screw M10 968 EUR (19.3%)
drill holes (labor) 396 EUR (7.9%)
drill holes (machine) 396 EUR (7.9%)
ins deliver to storage (labor) 396 EUR (7.9%)
(labor) 198 EUR (4.0%)
Slug for casin g 121
Flat seal 121 EUR (2.4%)
turn
100 %
#SAP Example: Pump P-100 - #Version 1

Costing Structure	Quantity for One Assembly	UoM (Qty for One Assembly)	Total Quantity	UoM (Total Quantity)	Total Cost	Reporting Currency	Price (Fixed)	Price (Variable)	Price	Transaction Currency	Price Unit	U
#SAP Example: Pump P-100 - #Version 1			10	PC	8,115.02	EUR	272.61	465.12	737.73	EUR	1	P
#CC1 #AT2 pick according to pick list (labor)	10	min	100	min	396.00	EUR	60.00	120.00	180.00	EUR	1	h
#BP2 Sales order processing	1	PC	1	PC	0.00	EUR	0.00	0.00	0.00	EUR	1	P
#PT1 #100-100 Casing	1	PC	10	PC	5,006.32	EUR	165.12	290.00	455.12	EUR	1	P
#PT1 #100-110 Slug for casing	1	PC	10	PC	121.00	EUR	0.00	10.00	10.00	EUR	1	P
#BP1 Plan work	0.1	PC	1	PC	0.00	EUR	0.00	0.00	0.00	EUR	1	P
#CC1 #AT1 turn casing according to dra...	10	min	10	min	26.40	EUR	120.00	0.00	120.00	EUR	1	h
#CC1 #AT2 turn casing according to dra...	30	min	300	min	1,188.00	EUR	60.00	120.00	180.00	EUR	1	h
#CC1 #AT3 turn casing according to dra...	30	min	300	min	1,188.00	EUR	120.00	60.00	180.00	EUR	1	h
#CC1 #AT1 drill holes (setup)	3	min	3	min	7.92	EUR	120.00	0.00	120.00	EUR	1	h
#CC1 #AT2 drill holes (labor)	10	min	100	min	396.00	EUR	60.00	120.00	180.00	EUR	1	h
#CC1 #AT3 drill holes (machine)	10	min	100	min	396.00	EUR	120.00	60.00	180.00	EUR	1	h
#PT1 #100-120 Flat seal	1	PC	10	PC	121.00	EUR	0.00	10.00	10.00	EUR	1	P
#CC1 #AT2 insert flat seal (labor)	5	min	50	min	198.00	EUR	60.00	120.00	180.00	EUR	1	h
#PT1 #100-130 Hexagon screw M10	8	PC	80	PC	968.00	EUR	0.00	10.00	10.00	EUR	1	P
#CC1 #AT2 inspect and deliver to storag...	10	min	100	min	396.00	EUR	60.00	120.00	180.00	EUR	1	h
#PT1 #100-200 Drive	1	PC	10	PC	2,475.44	EUR	84.24	140.80	225.04	EUR	1	P
#PT2 #100-300 Shaft	1	PC	10	PC	237.26	EUR	10.22	9.38	19.61	EUR	1	P

Key Figures

Total Cost	8,115.02 EUR
Total Quantity	10 PC

Item category: Internal Activity
Cost Center: #CC1
Activity Type: #AT2
Description: drill holes (labor)
Total Cost: 396 EUR

1 | 1/26/2016 3:14:29 PM This calculation version '#Version 1' is currently being edited by user '#CONTROLLER'. If you make any changes, you must save them in a new calculation version.

Figure 4.5: Interactive Treemap visualization is integrated into the side panel of SAP Product Lifecycle Costing application for evaluation purpose.

Results: The most interesting aspect that surfaced during these discussions was a preference of the customers towards the Sankey diagram. One reason for this tendency was that while the costing structure is hierarchical in nature, it still occurs that items of the same type – often screws or bolts – are used in the assembly of different components of the product. Since the Treemap is strictly hierarchical, these parts are shown individually for each component in multiple places, which makes it hard to judge their overall impact on the product cost. Moreover, Treemap requires little space and they have been integrated into the side panel of the application for testing purposes (Figure 4.5). Users could interactively work with their data in the spreadsheet UI and analyze the Treemap representation from the side panel. This prototype was never used actively by the customers of the project. However, the Sankey diagram allows for a more flexible representation and Figure 4.4 shows that the item "inspect and deliver to storage" is connected to different extent to all three of the higher level components "casing", "drive", and "shaft".

4.3.5 Challenges and Lessons Learned

In this section, we complement the report of our customer-driven process by reporting challenges that we encountered during the requirement engineering phase. In particular, as the literature on establishing visualization requirements is sparse, we deem it important to report on those challenges and how we solved them:

- Many co-innovation customers turned out not to be end users, but "gatekeepers" [Sedlmair et al., 2012] who at best fit the roles of "human-as-viewer" [Winters et al., 2016] or "consumers of analysis" [Kandel et al., 2012].
- Many of the customers were hesitant to share information for confidentiality reasons.
- Those who were willing to share were hard to reach and often unresponsive, as among their management duties, getting new visualization options was not high on their priority list.

- And from those few who were excited to help, we got mostly a wild mix of overly specific feature requests that changed with every new project they worked on.

As the co-innovation customers were the only ones we had access to, it was also not an option to find better collaboration partners, as suggested in some literature. So, we had to find creative ways to nevertheless get the necessary requirements for designing our visualization within these "constraints" [Crisan et al., 2016].

Getting Information from End Users

Having only limited access to actual day-to-day users makes a user-centered visualization design impossible. The literature lists alternatives ranging from *activity-centered design* to *goal-directed design* [Williams, 2009]. Yet again, to get information about activities or goals, we would need end users to work with us. In cases where we could not ask the end users themselves, we tried to ask the people who work with them to get "second-hand requirements". At SAP, this is the solution management team who interacts with the companies that require technical support. They know the problems of the users first hand and could give us information ranging from the companies' software landscape to frequently asked questions. In addition, we sent out surveys to our co-innovation customers that were intentionally phrased in more technical jargon and asking for completion in a rather short time-frame. Putting up these additional hurdles ensured that only end users who know their costing software and analysis practices could easily fill out these surveys within the allotted time. Thus, we could be confident to only receive responses either from knowledgeable managers, or from costing analysts to whom the questionnaire was passed on. While such measures may increase the quality of responses, they unfortunately also decrease their quantity and thus have to be employed with care.

Overcoming the Hurdles of Information Sharing

The hesitance to share information is a wide-spread problem in requirements engineering for which there exists no simple solution [Goguen, 1993, Goguen and Linde, 1993]. The field of Social Engineering has developed methods to build trust and convince people to disclose information [Hadnagy, 2010]. Yet, many of these methods are deceptive, bordering on the unethical, and stand in stark contrast to the best practices of open and honest customer communication. To lower these hurdles, we followed a combination of the *employee-pull* and *researcher-push* strategies [Sedlmair et al., 2011]. For the employee-pull part, we utilized the *toolsmith method* [van Wijk, 2006] making sure that our discussion focused on the needs of the customers and not on our needs for their data and feedback. As of course the needs of the customers can only be sensibly discussed in the context of their specific data and tasks, we got the needed information almost as a byproduct of the customer interviews. Once we had collected enough information to start producing early visualization prototypes, we switched to the researcher-push part, following the *design-first* method [Paul et al., 2015]. This means that without having yet detailed knowledge of data or tasks, we already realized two prototypes showing synthetic data. These allowed the customers to move from discussing their confidential data structures and business questions to discussing our prototypes.

Collecting Information from Unresponsive Partners

We made the observation that, compared to other business-related surveys and cooperation requests, the co-innovation customers did not respond as eagerly to our surveys. For example, from the 21 customers surveyed in the beginning of the process, only 5 initially returned the filled-out questionnaires. This is problematic, as so few returned questionnaires not only raise doubts about their representativeness, but also introduce a high potential for non-response bias in the results [Deming, 1990, p.66]. As many co-innovation customers are in middle managerial positions, it is no wonder that answering a survey on business-related questions comes more easily to them than one about the implications of product costing on visualization. In practice, we used two modes of collecting the answers we needed: the e-mail survey and personal interviews to follow up. The survey via e-mail has the advantage of being asynchronous, so that the customers can complete it at their leisure. It also offers the possibility to speak with the actual analyst or even to delegate filling out the survey directly to them. The downsides are its non-committal nature, as well as the possibility that questions are misunderstood or not answered with enough detail. Personal interviews at customer workshops have just the opposite characteristics, which makes them perfect to complement the e-mail surveys. Interviews were particularly well received for issues that would have required lengthy written explanations, but that could simply be demonstrated by pointing at the screen during an interview.

Yielding Focused and Relevant Information

It is no secret that customers have different goals than visualization designers: Where customers prioritize the needs of their specific current project, the visualization designers try to realize a visualization that supports costing analysis in general. Note that this gap between the special-purpose tool the customer requires and the general-purpose tool the researcher desires is different from the customer/researcher gaps observed by Wijk [van Wijk, 2006].To find common ground and avoid accidental overfitting, we planned multiple quarterly discussions and interviews over the course of the project. Looking at which problems and questions reappear over the course of these multiple meetings helped us separate principal requirements from the specific “problems of the day”. These principal requirements served as a baseline describing typical costing analysis, which usually means to find out what makes the product expensive (task T1) on datasets of typical size (see Table 4.1). None of these commonplace requirements were particularly “exciting” – neither to the customers, nor to us. But it is this common denominator that a visualization should first and foremost be able to handle, while special cases can be added onto this “base visualization” in subsequent design iterations.

Lessons Learned

Apart from the concrete approaches outlined above, we made the following observations that helped us obtain requirements from our customers, and that may serve as general recommendations:

- **Interview participants in their native language.** We observed that participants are more talkative when using their mother tongue. That was the reason for setting up two independent workshops in Germany and the US, so that each group of customers had the opportunity to partake in the group discussions and interviews in their native language.

- **Ask about problems, not solutions.** It is easier to get people to open up by asking them about their gripes and grievances with the current system than having them suggest features they might want to use in the future. This is in line with existing findings that middle-level managers are more likely to perceive problems than to come up with potentially useful features [Sutcliffe, 2002, p.89].
- **Provide a neutral context for discussions.** By presenting our early and imperfect visualization prototypes to the participants, we not only gave them something to criticize (see previous point on problems vs. solutions), but also a neutral context in which to frame their statements. This catalyzed their conversation with us, but also their discussion among each other, as they did no longer have to worry about disclosing internal company details.

There are many more challenges in requirements engineering that still need research and further reports from visualization practitioners on how to resolve them – for example, how to manage conflicting requirements? While some of our observations and lessons learned seem to be tacit knowledge in the visualization community, we deem it important to make them explicit. In this way, these guidelines can be collected and properly debated in meta studies and surveys to someday yield the still missing compilation of best practices for working with visualization end users.

4.4 Data and Task Abstraction

Many data visualization techniques have been created in the past decades for specific tasks and data types. However, many of them are rarely reused in practice for other datasets or tasks. Munzner suggests a task and data abstraction phase in order to re-frame the user's task and data from a domain-specific form into an abstract form. In this section, we abstract the product costing problem and the data types gathered from the customers of the project (see sections 4.3.2 and 4.3.3), into a generic representation with a domain independent vocabulary [Munzner, 2014, p.78]. This phase is the second step of the nested model introduced by Munzner shown previously in Figure 4.1. We give an account first, of the data abstraction and second, of the task abstraction.

4.4.1 Data Abstraction

Our first step in abstracting our domain-specific data (see section 4.3.3) to a generic framework was to derive the data types used in the product costing domain. After reviewing the conducted surveys, we observed that all customers' data have five common characteristics shown in Figure 4.6. However, the typical users' tasks are arising from the first three characteristics of the data: hierarchical, categorical aggregates, multi-dimensional and the two other characteristics are covered as extensions. Therefore, we started with formally defining and characterizing a complex compound hierarchical data structure applicable to many other domains such as biology or demographics.

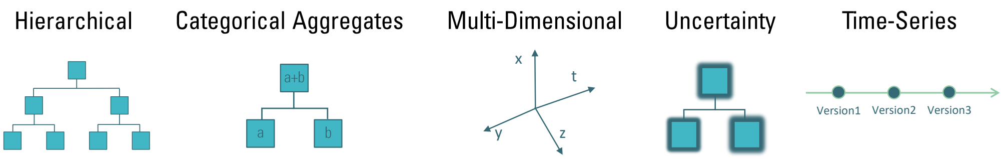


Figure 4.6: The main five characteristics of the data researched in this thesis.

An example. Let's consider a product costing example. We consider the different parts of a product – in our case a car – and define for each part its cost as well as additional properties per part such as where in the world is the part made. Figure 4.7 (left) shows that the car is made of a body, an engine, and several wheels, and the body itself is made of doors, windows and seats. The cost of car is equal to the sum of the body, engine and wheels' costs, and similarly the cost of body is equal to the sum of doors, windows and seats' costs.

Multiple Hierarchies. Figure 4.8 shows the relationship between the two graphs shown in Figure 4.7. The parts constituting a car (doors, windows, seats, wheels, engine) can originate from or be assembled at different world locations – these locations form a hierarchy with different hierarchy levels for cities, regions, and countries. Another example of hierarchy for the data described before woukd be *materials* such as metal, wood, textile, and then different kinds of metal such as steel or brass, as well as different types of textile. Part properties can also have a single level in their hierarchy, for example, the *durability* attribute could take three values: imperishable, long-lasting, short-lived. All hierarchies must have a root element. In some cases, the root element of the attribute hierarchy is obvious, for instance for locations the root is the whole world, similarly for materials the root is the all encompassing *material* attribute value. However, when the root is not obvious – like in the case of the durability attribute – we can simply introduce an abstract root element.

All in all, the data model behind the hierarchical decomposition problem has three important characteristics: data items with categorical properties, hierarchies defined over these categorical properties, and aggregate numbers for each category.

Data Items. The basis of our data is formed by a set X of items $x_1, \ldots, x_n$. Each data item x_i has various properties $a_1, \ldots, a_m$. The properties of interest to us are the categorical ones, such as location, material, and durability in the costing dataset. If necessary, numerical properties can be transformed into categorical ones to also include them in the analysis [Johansson et al., 2008].

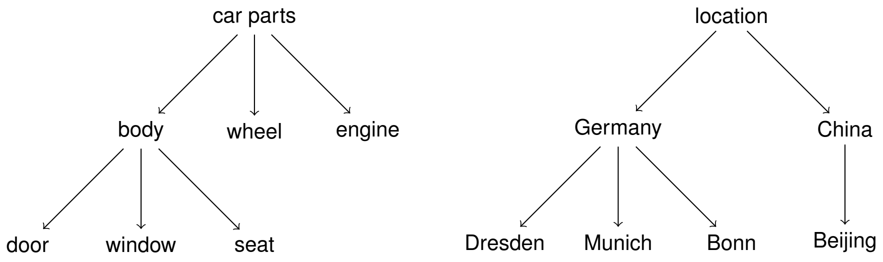


Figure 4.7: Sample costing dataset with two hierarchies.

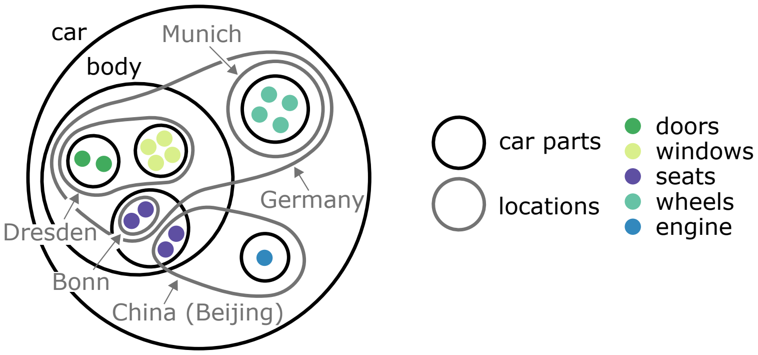


Figure 4.8: Sample car example with two collection sets, breaking down the data based on car parts (black) and location (gray).

Hierarchies. In many cases, the data properties imply a hierarchical structure, which means that although being a flat categorization, we can hierarchically group the categories. Formally, a hierarchy over property a_i forms a set collection $\mathcal{A}_i$ fulfilling the following three constraints:

1. $\forall U \in \mathcal{A}_i : U \subseteq X$, i.e., all sets in $\mathcal{A}_i$ are subsets of X;
2. $X = \bigcup_{U \in \mathcal{A}_i} U$, i.e., all elements in X are covered by $\mathcal{A}_i$;
3. $\forall U, V \in \mathcal{A}_i : U \cap V = \emptyset \ \ or \ \ U \cap V = U \ \ or \ \ U \cap V = V$, i.e., the set collection $\mathcal{A}_i$ forms an *inclusion hierarchy* [Ahl and Allen, 1996, Pumain, 2006].

In fact, it is hard to find an aspect of our life that has not yet been organized by some kind of hierarchical structure. We have the ICD-10 classification of diseases in medicine, managerial hierarchies in companies, the ACM computing classification system to index scientific papers, etc. But, it is also possible to define or derive hierarchical structures from a flat categorization. Defining such a hierarchy involves some background knowledge about the data properties. For example, by grouping people's ages into age groups such as babies, children, teenagers, young adults, etc. Yet, it can also be automatically derived without any background knowledge on the data property, for example, by using a hierarchical clustering algorithm for categorical data like ROCK [Guha et al., 2000]. Each hierarchical property forms a set collection with the sets being structured as an *inclusion hierarchy* [Ahl and Allen, 1996, Pumain, 2006]. This means individual items are grouped into smaller sets, smaller sets are subsequently grouped into larger sets, until a single unifying set is formed at the very top of the hierarchy – for example, places of birth (e.g., municipalities) being grouped first by county, then by state, country, and continent, until everything is unified under the singular set "world". As we have multiple hierarchical categorizations given, these can also be understood as special cases of *multitrees* [Furnas and Zacks, 1994] or *polyarchies* [Robertson et al., 2002] defined over the item set X.

Aggregates. Finally, we have aggregates defined over our categories and their hierarchical groupings into set collections. An aggregate can be understood as a function $c : U \in X \mapsto \mathbb{R}^+$ that maps any data subset onto a positive real value. The most common aggregate is the simply the number of elements in a set: $c(U) = |U|$. For the costing data, this would translate to the cost of parts in a category. In addition, it is

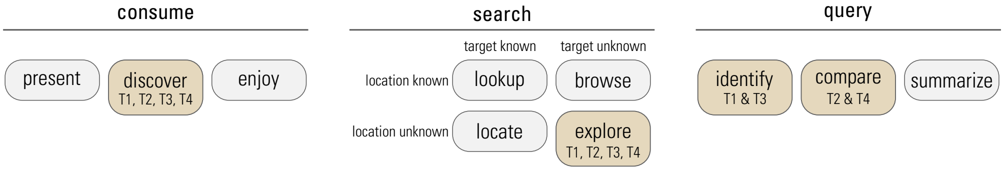


Figure 4.9: Our multi-level typology of task abstraction: high-level (discover from consume), mid-level (explore from search), low-level (identify & compare from query), based on [Brehmer and Munzner, 2013].

also possible to involve another, numerical data property w as a positive weight being associated with each individual data item. Instead of adding up the mere number of items, one can also add up their weights to generate the set aggregates:

$$c_w(U) = \sum_{\forall u \in U} w(u)$$

For the costing data, we could use each part's uncertainty as a possible weight, so as not to compare the costs in different categories, but actually their combined uncertainty. Weighing by other factors such as profit might make more sense to marketers and sales people, trying to target groups of items according to their benefits for the company.

4.4.2 Task Abstraction

Based on the tasks gathered from the customers of the project (see section 4.3.2), the tasks users aim to perform on costing data has two main aspects: (1) to find meaningful aggregation levels for the different hierarchies defined over the categorical properties of the dataset, and (2) to investigate the interrelation between the different categories of the dataset.

For finding suitable aggregation levels, the dataset is either grouped and clustered in a bottom-up manner, or an overall aggregate is decomposed into a number of subsets. The bottom-up and top-down approaches form a dual perspective on the task, as they approach the same objective from opposite directions. Each of these perspectives comes with its own emphasis on particular aspects of the visual encoding (e.g., showing the individual items vs. showing part-whole relations) and interaction design (e.g., interactive grouping/roll-up vs. interactive partitioning/drill-down) [Elmqvist and Fekete, 2010]. We take on the top-down, decomposition perspective of having a large aggregate that needs to be unpacked, instead of a wealth of individual items that need to be accumulated. In terms of the costing example, this means that we are looking at the total cost, instead of a large number of items costs.

In order to discover the relationships between the different categories, the data is *cross-tabulated*, as it is known in statistics. This means given two collections $\mathcal{A}_i$ and $\mathcal{A}_j$, and two sets $U \in \mathcal{A}_i$ and $V \in \mathcal{A}_j$, we are interested in the intersection $U \cap V$ and

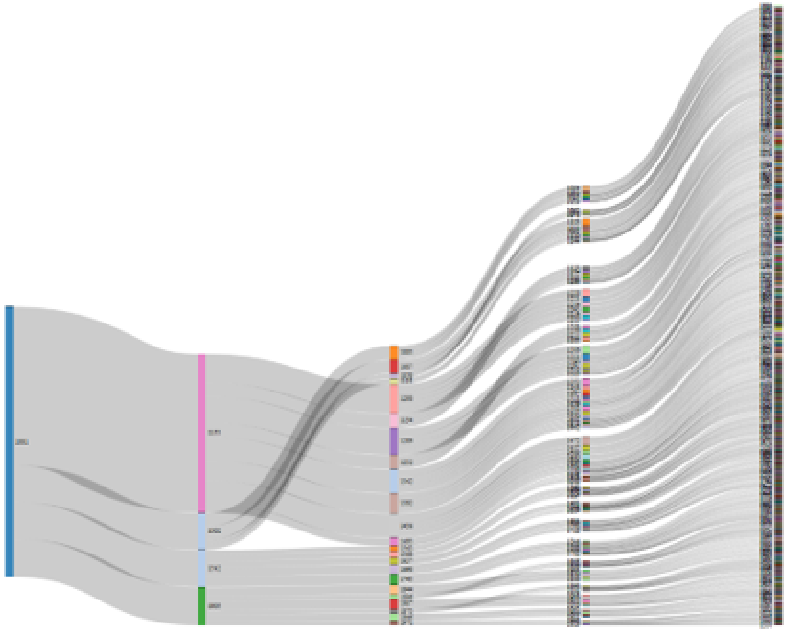


Figure 4.10: Visualizing a typical product costing dataset with Sankey diagram.

its corresponding aggregate value $c(U \cap V)$. For the costing data, this means when U consists of all items produced in Germany and V consists of all items made of steel, the intersection $U \cap V$ consists of all items fulfilling both of these properties. Doing this for all possible pairwise property combinations yields a *bivariate joint frequency distribution*, which is usually displayed using *contingency tables* [Meyer et al., 2008]. The current visualization techniques are not geared towards handling both of these two aspects – drill-down along multiple hierarchies and cross-tabulation – at the same time, let alone to go back and forth between them. This is the gap that our solution is designed to fill.

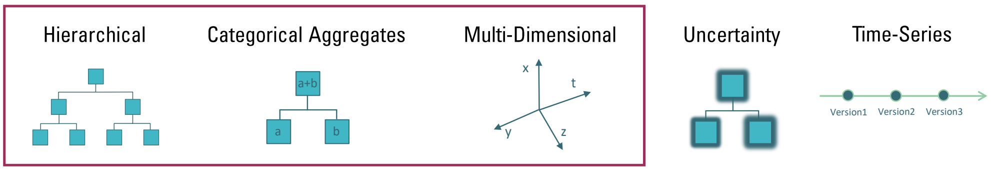
In addition, in order to map customer tasks to more general task categories, available task taxonomies for visualization design and evaluation were reviewed (see section 4.3.2). Brehmer and Munzner suggest a multi-level typology of abstract visualization tasks, which is detailed in section 2.1.2 [Brehmer and Munzner, 2013]. We reviewed the characteristics of the four common tasks gathered from the customers and Figure 4.9 summarizes our task abstraction. All four tasks fit to the *discover* goal from the consume category on the highest level of the taxonomy. A mid-level target of all tasks is *explore*, as the identity and location of the search items are unknown. Finally, after the targets in the search category are established, the low-level goal for Task 1 and Task 3 is *identify* and for Task 2 and Task 4 *compare*.

4.5 Summary and Outlook

In this chapter, we presented our requirement engineering methodology. First, we described the environment this research was established in. Second, we reported on our experiences and lessons learned during the requirement engineering process. We reviewed the results of requirements elicitation from co-innovation customers, along with problems faced during this phase and solutions found. Customer's current solutions, data characteristics, and user's goal were investigated to characterize the domain situation of this research. Moreover, we realized two early prototypes to get quick feedback from users: a Treemap and a Sankey diagram. Both prototypes were evaluated at two workshops and based on users inclination towards Sankey-based diagrams, we applied Sankey diagrams to a typical data size shown in Figure 4.10. From these preliminary activities, it became clear that the current implementations of Sankey diagrams are not capable of handling all characteristics of our researched data and a novel solution is needed.

Finally, to avoid having a domain-specific solution, we abstracted the data and tasks gathered from users into generic data and tasks, which apply to other domains such as bioinformatics, software systems, and social science.

This chapter covered the first two nested levels of visualization design shown in Figure 4.1. In the following chapter, we describe the process of designing and implementing a visual representation as well as interaction techniques for the abstract data and tasks defined in section 4.4.1 and 4.4.2.



Chapter 5

Parallel Hierarchies

In this chapter, we describe Parallel Hierarchies, a visual-interactive solution to the problem of cross-tabulating numerical aggregates over hierarchical categories. Parallel Hierarchies combine ideas and approaches from various fields. This section unpacks and describes this problem by breaking it down into the properties of the input data (described in section 4.3.3) and the necessary affordances of the visual output (described in section 4.3.2). This chapter covers the third and fourth nested model levels of visualization design that attempt to specify the *visual encoding and interaction idioms* along with all of the design decisions in creating an *algorithm* for them (see Figure 4.1). Parts of the research presented in this chapter have previously been published in [Vosough et al., 2018a].

5.1 Introduction

Many aspects of our daily lives are hierarchically organized: our professions are organized in the Standard Occupational Classification (SOC) hierarchy [Bureau of Labour Statistics, 2018], the books we loan from the library are organized by the Dewey Decimal Classification (DDC) [Chan and Mitchell, 2003], and the illnesses we get are catalogued in the International Statistical Classification of Diseases and Related Health Problems (ICD-10) [World Health Organization, 2016]. One of the most interesting aspects of these hierarchical categorizations is when they get applied to the same set of individuals or items – as this lets us systematically explore dependencies or cross-correlations between them. For example, people in certain occupations may be more likely to get certain health problems, and people with particular health problems may be more likely to read books on specific self-help topics, and vice versa. In particular, before knowing these dependencies, joint interactive exploration of different hierarchies can reveal unexpectedly high or low numbers between categories from different hierarchies.

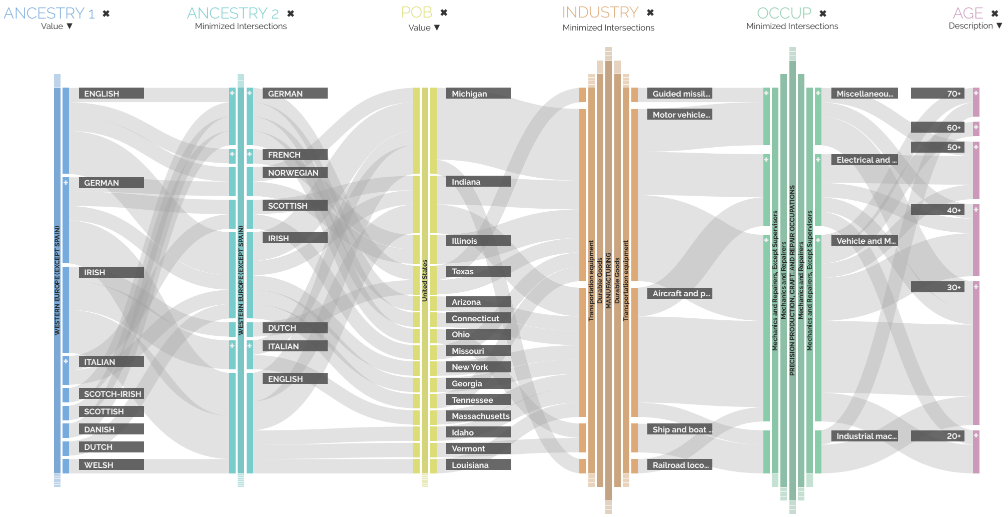


Figure 5.1: Parallel Hierarchies show some data properties of the US Census 1990 dataset.

Data visualization can enable interactive exploration of numerical distributions across multiple hierarchies. For relating hierarchical data to each other, the most common visualization approach is to draw different hierarchies side by side and to connect them with visual links. In lack of a name for this type of visualization, it has been alluded to “as what Parallel Coordinates would resemble if the axes were hierarchical in nature” [Graham and Kennedy, 2010, p.10]. Often, this approach is focused on structural comparisons between similar hierarchies, such as showing the overlap between them, or determining which nodes may have been added, removed, or changed with each version of a hierarchy [Holten and Van Wijk, 2008, Telea and Auber, 2008]. Yet, for quantitative comparisons between entirely different hierarchies, this type of visualization has never been formally introduced, its design implications have never been discussed, and the resulting representations have never been evaluated.

This chapter addresses these desiderata by introducing our solution, called *Parallel Hierarchies*, illustrated in Figure 5.1. The current view shows the distribution of people with Western European roots who were born in the US and work as mechanics on any kind of transportation equipment. One can see from this figure, that this was a young profession in the 1990's with approx. half of these people being in their 20's and 30's. Also, the state of Michigan stands out, which is no wonder as it is not only famous for its automotive industry, but also for its people of European descent. Our visualization technique *Parallel Hierarchies* is specifically tailored to take hierarchical categorizations into account. With Parallel Hierarchies, it is possible to individually adjust the desired level of detail for each categorical data property through drill-down and roll-up operations. This enables the analyst to selectively change levels of detail as the data analysis progresses and new questions arise. We illustrate the utility of Parallel Hierarchies with a demographic use case based on the 1990 US Census data, of which 1% and 5% samples are publicly available[1].
In combination, the described data and task abstractions in section 4 specify the type of problems for which Parallel Hierarchies provide a visual-interactive solution.

[1] https://www2.census.gov/census_1990/

5.2 The Parallel Hierarchies Technique

Parallel Hierarchies technique is designed specifically to (1) navigate multiple hierarchies defined over categorical data properties to find suitable aggregation levels, (2) cross-tabulate pairs of categorical data properties at their respective aggregation level, and (3) switch effortlessly between the two. Together with common guidelines for designing categorical displays [Guchev et al., 2012, Fernstad and Johansson, 2011, Zeileis et al., 2009], these specifications informed our visualization design.

Parallel Hierarchies feature an arrangement of *vertical axes* representing the different hierarchical properties and allows for drill-down and roll-up interaction. These axes are connected via curved *horizontal ribbons* that represent the pairwise (weighted) frequency counts for cross-tabulating neighboring axes. The following sections describe the combined visual and interaction design of the base visualization in three steps: for an individual axis, for pairs of axes linked by ribbons, and for a series of multiple such connected axes. In a fourth step, it details additional customization possibilities for further fine-tuning of Parallel Hierarchies – mainly by means of re-ordering axes, categories, and ribbons.

5.2.1 The Individual Axis: Showing Hierarchical Categories

Each individual axis in Parallel Hierarchies encode one hierarchical set collection. To serve as an axis, the hierarchy display must use as little screen space as possible and needs to be lean and uncluttered even for large hierarchies. To achieve this, we utilize simplified *Icicle Plots* [Kruskal and Landwehr, 1983] where "simplified" means that the hierarchy is not shown in full breadth and depth, but only the currently focused branch is displayed. We thus employ a top-down approach, which is exemplified in Figure 5.2 where we perform a 3-step drill-down into a hierarchical property.

On the left of Figure 5.2, the topmost level of the data property is shown. The different categories are stacked into a *Spine Plot* [Cox, 2016] to convey the univariate distribution of items among them – i.e., the height of the vertical bar representing a category is proportional to the aggregate value of that category. A mouse click on one of the categories drills-down and leads to the second view in Figure 5.2. Note that the updated view focuses solely on the clicked category, which we call the *active category* and displays its subcategories, which we likewise term *active subcategories*. All siblings of the active category – i.e., those categories that we did not click on in the last step – are now reduced to a stylized representation at the top and/or bottom to provide a contextual indication of their number and their positioning according to the current ordering scheme. Another click on one of the active subcategories drills-down further by making the clicked subcategory the new active category and displaying its subcategories. The path to the initially selected category is always visible as the clicked categories get stacked from left to right. For a roll-up, the user can simply click on one of these ancestral categories to make it the active category again. This interplay of the interactive exploration of a tree's topology with a dynamic adaptation of the tree's display is reminiscent of the *SpaceTree* technique [Plaisant et al., 2002].

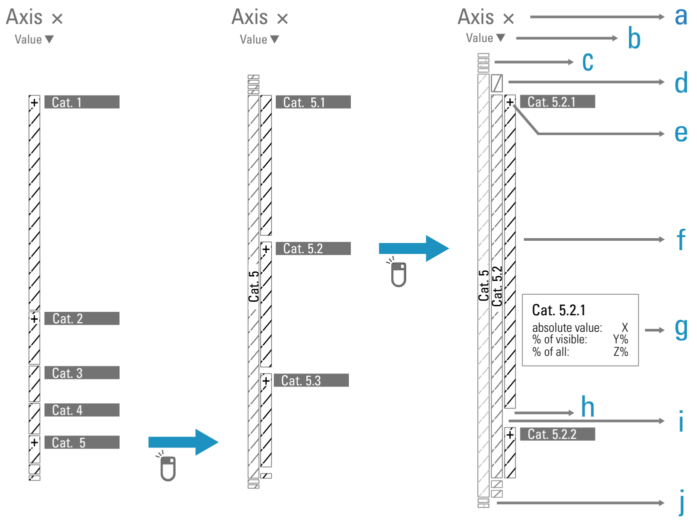


Figure 5.2: An individual axis functioning as a simplified Icicle Plot. Clicking on a category drills-down into the hierarchy. This condenses all siblings of the clicked category to make space for unfolding the next level of subcategories.

Albeit we utilize a more condensed and simplified adaptation mechanism that focuses solely on the path from the root to the active subcategories – i.e., the “ancestor path”, as it is called in SpaceTrees.

This minimalistic approach for exploring trees under space constraints has been described in the literature, with a preliminary evaluation suggesting that it outperforms other tree views on small displays for tasks involving known targets [Band and White, 2006]. In a sense, this form of drill-down/roll-up forms a combination of pivoting certain values on an axis [Nielsen and Grønbæk, 2015], grouping related values on an axis [Palmas et al., 2014, Richer et al., 2018], and filtering values on an axis [Siirtola and Räihä, 2006]. The following list briefly describes the individual aspects of this hierarchy representation as they are labeled from (a) through (j) in Figure 5.2:

a) Name of the hierarchical data property, which can also be interpreted as the root of the hierarchy. Note that we do not otherwise show the root node in an axis as it has no added value to show the “distribution” of a singular item.

b) Current ordering scheme of the categories, and interaction handle to change the scheme. Details on different orderings are given in Section 5.2.4.

c) Unselected siblings positioned before the selected category according to the currently chosen ordering scheme. The representation is stylized, meaning that no absolute values are encoded in their height; they merely indicate their number.

d) Ancestors of the active category (i). The ancestors represent the path of clicked categories that led to the current view. Ancestors also serve as interaction handles to trigger roll-up operations back to their level.

e) Small **+** marker indicating that this particular active subcategory splits even further into more detailed subcategories. If such a more detailed categorization is needed, an active subcategory with such a marker can be clicked to trigger a drill-down operation.

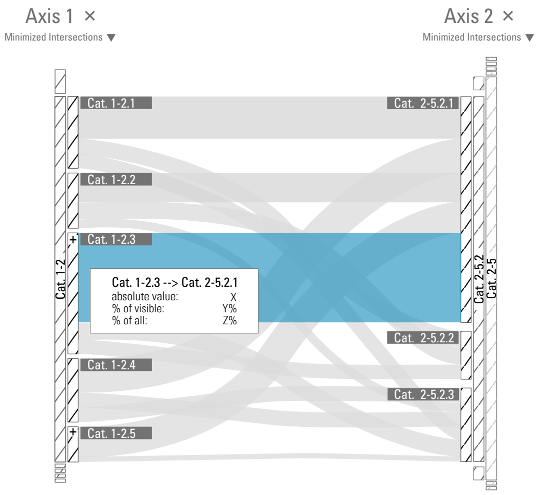


Figure 5.3: Connecting the active subcategories of two axes with ribbons to show their pairwise frequency counts. Hovering over a ribbon with the mouse highlights it and displays detail information.

f) Active subcategories split the aggregate value of the current active category (i) and encode these splits proportionally in their respective heights. In addition, they present possible options for further drill-down operations.

g) On demand detail information showing up when mousing over a category – ancestor (d), active category (i), or active subcategory (f). The shown details include the full name of the category, and its absolute aggregate value, as well as the relative aggregate value w.r.t. the data subset currently visible and w.r.t. the total dataset.

h) Gaps in between active subcategories (f) delineate the subcategories from each other. The size of the gaps varies (see second axis in Figure 5.2) with the number of active subcategories, as the available whitespace is distributed equally between subcategories.

i) Active category indicating the current focus of the axis, as every drill-down operation also implies a filtering of sibling nodes of the active category and of its ancestors (c,j).

j) Unselected siblings positioned after the selected category according to the current ordering scheme.

5.2.2 Two Interlinked Axes: Showing Pairwise Frequencies

When combining two of these axes, we can display two different hierarchical set collections, one on each axis. To indicate their pairwise intersections for all active subcategories of both axes, we connect both axes with curved ribbons whose width is proportional to the aggregate value of the intersection they represent. Together, these ribbons provide for a full cross-tabulation of both sets of active subcategories.

Figure 5.3 gives an example of two connected axes and cross-tabulating their active subcategories. The two axes stack the hierarchy levels towards each other, so as to make connecting them easier. Curved, light gray ribbons are drawn from one active subcategory on one axis to another active subcategory of the respective other axis if these share at least one data item – i.e., their corresponding aggregate value is larger than 0. The width of the individual ribbons corresponds to the magnitude of the associated frequency count, which means for our example, the more people fall into both subcategories, the wider their connecting ribbon. In sum, the width of the ribbons stack up to the overall aggregate of both connected active categories. The ribbons generally follow any interactive operation (drill-down, roll-up, reorder, etc.) performed on the axes. Mousing over an active subcategory highlights all incident ribbons. Mousing over a ribbon highlights both incident active subcategories. The latter also yields more detailed information on demand and shows for example the absolute and relative aggregate value for a selected intersection.

It has to be noted that the space between two axes is constrained by the available screen space, which can be problematic for drilling into particularly deep hierarchies. Resulting layout problems can be that both hierarchies do not leave enough space for routing the ribbons in between them, or the Icicle Plots could even meet in the middle. To avoid these problems, we define a maximum number of levels to be stacked onto an axis. If this number is exceeded, ancestor categories further away than this maximum number of levels from the active category become increasingly thinner to save horizontal space.

5.2.3 Multiple Linked Axes: Propagating Frequencies

Concatenating multiple instances of the bivariate display introduced in the previous section effectively extends it into a full-fledged Parallel Hierarchies visualization. From a visual perspective, this extension is simple: to be able to connect ribbons to intermediate axes from both sides, we simply mirror the Icicle Plots for all axes except the leftmost and rightmost ones. This is illustrated in Figure 5.4. Conceptually, this extension brings up a number of new aspects that need to be considered for the axes, as well as for the ribbons. Showing only (weighted) pairwise frequency counts between neighboring axes limits the analysis to 2-way cross-tabulations between shown categorical properties. To counter this effect, Parallel Hierarchies allow for flexible arrangement of the axes – i.e., which properties are shown as axes and in which order – as it is done for Parallel Coordinates. This configuration is manually adjusted by adding, moving, and removing axes until the view is appropriately configured to answer a given analysis question.

Yet, however purposefully configured, the resulting view is still restricted to only cross-tabulating neighboring pairs of axes. To mitigate this restriction to some extent, the hovering/highlighting mechanisms for active subcategories and ribbons is further refined to propagate the highlighted items across axes and to show their spread across the whole view. For example, on the right side of Figure 5.4 the ribbon connecting “Cat.2-5.2.1” with “Cat.3-5.2” is moused over. This highlights the ribbon itself and shows how many data items fall into the intersection of these two properties. In addition, this subset of data items is further propagated to the left, showing how these data items are distributed across the different active subcategories of “Cat.1-2”. While all involved ribbons between “Axis 1” and “Axis 2” are highlighted, the distribution of the moused over subset among these ribbons is indicated with a darker highlight at the bottom of each ribbon. In this case, we see that most of these data items fall into category “Cat.1-2.3”.

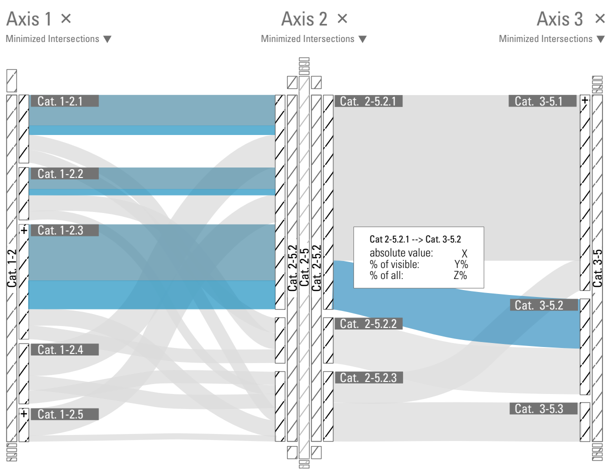


Figure 5.4: Concatenating multiple axes and interlinking them with curved ribbons to represent multiple pairwise frequency counts. When highlighting a ribbon in this setup, the highlighted data subset is visually propagated to indicate its distribution across axes.

It has to be noted that all axes are linked into a unified visualization. This means that any interactive change (roll-up/drill-down) to one of the axes affects the item set shown as a whole and not just those on the changed axis. For example, if one was to drill-down further, for example into the subcategory “Cat.3-5.1” in Figure 5.4, this would not only filter items on “Axis 3” and its incident ribbons, but also for the entire visualization. I.e., the height of the active subcategories on all other axes and the width of their incident ribbons would be adjusted to then reflect the particular distribution among those data items only present in “Cat.3-5.1”. As a result of this interlinkage, a few drill-down operations on some of the axes can significantly reduce the shown data, to the point that few items remain visible to be able to draw meaningful conclusions. To give the user an overview of how much of the dataset has already been filtered out and is no longer visible, a small column chart at the top of the visualization gives an indication of this information. See Figure 5.5 for the column chart corresponding to the data shown in Figure 5.1. The column chart gives the user not only an idea of what portion of the data is missing overall (gray column), but also to which degree the drill-down operations on the different axes are responsible for it, as the columns follow the color coding and axis ordering of the main visualization. If desired, the user can also change the column chart to indicate the percentage of the data that remains visible.

5.2.4 Fine-tuning Parallel Hierarchies through Reordering

The previous sections describe the base visualization of Parallel Hierarchies including interactive means of its adaptation, such as drill-down and roll-up of hierarchy levels, or adding and removing axes. This section expands on possible adaptations of the base technique by discussing the finer details of its layout and exploring their degrees of freedom. These degrees of freedom mainly stem from the fact that neither the different data properties (axes), their different categories (active subcategories), nor the pairwise intersections between them (ribbons) have an inherent order that dictates their position on the screen.

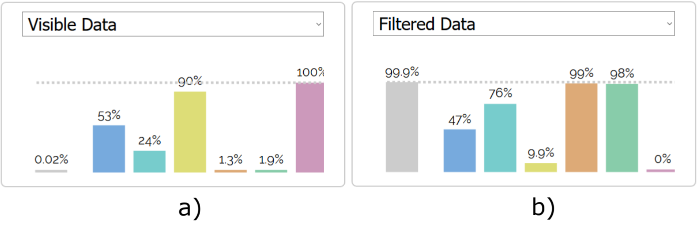


Figure 5.5: Column chart indication how much of the overall dataset is a) currently visible or b) filtered out. The column in gray gives an overall indication, whereas the colored columns indicate to which degree the drill-down on the corresponding axes is responsible for this.

Reordering Strategies for the Axes

When deciding for an axes arrangement, a key question is what purpose is served. We distinguish between two cases: providing an overview of the data with all data properties being equally important, and looking at details of how a specific data property of interest relates to the other ones.

In the first case, where no particular data attribute is of more interest than any other, we are free to arrange the axes in whichever way produces the clearest and least cluttered overview. The common approach used for Parallel Coordinates aims to identify correlations, convergences, or other patterns between axes, so that placing these axes next to each other yields a less cluttered output that clearly exhibits the detected patterns [Heinrich and Weiskopf, 2013, Sec.6.2]. This approach does not work for Parallel Hierarchies, as we are cross-tabulating all active subcategories between two axes. This means, in many cases we have a complete many-to-many connectivity between the axes, where the connections only differ in their respective pairwise frequency count – i.e., all possible ribbons are present and differ only in their respective widths. To reduce some clutter, we suggest to reduce the number of ribbons by applying one of the following two heuristics:

- *Alternate axes with many active subcategories and axes with few active subcategories.* As potentially all subcategories of an axis are connected to all neighboring subcategories, it makes sense not to place two axes with many subcategories next to each other, as that would produce a high number of ribbons and thus massive clutter between them.
- *Place axes with a one-to-one relation next to each other.* If two data attributes exhibit such a relation, it is even possible to relate them without any ribbon crossings at all. An example would be the sales districts and salesmen from Figure 3.2 (b), where each salesman is associated with exactly one district.

In the second case, where we want to explore the interrelations of one particular data property to all other properties, we need to prioritize this aim over clutter reduction. This can be done in two ways:

- *Positioning the axis of interest in the middle.* This simple idea stems from the observation that assessing propagated item sets across multiple axes becomes increasingly difficult for more axes. This applies to every axis, the item set gets further spread into increasingly thinner partial ribbons that are further and further away from the originally highlighted subcategory or ribbon. By placing the axis of interest in the middle of the view, this effect is lessened to some extend.

- *Adding the axis of interest multiple times and interleaving it with the others.* Parallel Hierarchies also permits to add the same axis multiple times, so that it can be placed in alternating sequence with the other axes. This approach makes it easier to cross-tabulate the axis of interest with all others, as it does not rely on propagation. Yet, it also requires more horizontal screen space to accommodate the duplicates.

Note that these four strategies can of course be combined with each other. For example, we can position an axis of interest in the middle and still arrange all other axes in an alternating fashion with respect to their number of subcategories around the central axis.

Reordering Strategies for the Categories
Much has been published on how to establish a sensible sorting of items with no inherent order along an axis [Ma and Hellerstein, 1999,Beygelzimer et al., 2001,Rosario et al., 2004]. The consensus is that there is no universal order that would satisfy all possible visualization needs and support all possible visualization tasks. Instead, it is paramount to be able to switch between different sorting strategies depending on the analysis task. This loosely aligns with the task taxonomy by Andrienko and Andrienko [Andrienko and Andrienko, 2006]:

- Direct look-up tasks (*given: category, sought: corresponding value/count*) benefit from an alphabetical order that allows users to quickly find categories by their name.
- Indirect look-up tasks (*given: aggregate value/count, sought: category*) benefit from an ordering according to the aggregate values or counts of each category.
- Comparison tasks (*given: categories, sought: their relation*) benefit from an ordering of categories that minimizes clutter and crossings when connecting to neighboring axes to visually identify categories with similar and/or different connection patterns.
- Relation-seeking tasks (*given: a relation, sought: categories conforming to that relation*) benefit from any ordering of categories that take the given relation into account – e.g., ordering based on a correspondence analysis when looking for similar categories.

Ordering based on results from a correspondence analysis has been treated in depth by Johansson and Johansson [Johansson and Johansson, 2009], and we refer the interested reader to their work for details. Reordering categorical items on parallel axes to reduce crossings between their connecting lines or ribbons can be formalized as an instance of the *k-Layer Straightline Crossing Minimization* problem, which is known to be NP-hard even for $k = 2$ [Eades and Whitesides, 1994]. For computing optimized orderings, we employ Step II of the well-known *Sugiyama layout heuristic* [Sugiyama et al., 1981] in combination with the *barycenter* method. This has already proven to be an efficient heuristic for crossing minimization for bipartite graphs [Jünger and Mutzel, 1997], and it has also been successfully used to reduce ribbon crossings in

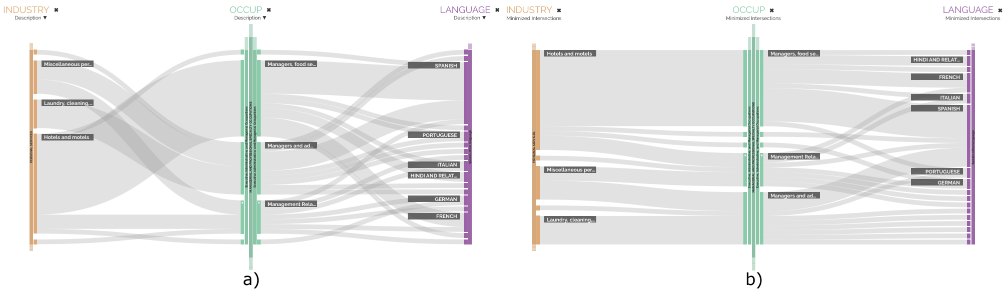


Figure 5.6: a) before and b) after applying the crossing minimization to the ordering of categories along the axes. In this example, the crossing minimization reduces the number of ribbon crossings from 134 down to 23. In general, it can be observed that the crossing minimization is most effective for drilled-down views where no longer every category is connected to every other category, as the data gets more sparse.

storyline visualizations [Liu et al., 2013, Gad et al., 2015]. As an additional optimization, a greedy switching heuristic can optionally be used in addition to the barycenter method [Mäkinen, 1990]. While this additional heuristic is able to generate slightly better results in most cases, it also requires a much longer runtime. This makes it ill-suited for use in exploratory analyses, but it is a good option for "pretty printing" a final result for presentation purposes.

Other quality metrics for ribbon-based visualizations aside their crossing number have been proposed in the literature. For example, Perin et al. considered large link heights – i.e., large vertical spans of ribbons – to be disadvantageous to reading a visualization [Perin et al., 2016]. The reason is that ribbons of large height are more susceptible to the line width illusion if they are straight [Hofmann and Vendettuoli, 2013] or to the sine illusion if they are curved [VanderPlas and Hofmann, 2015]. Both of these illusions alter the appearance of a ribbon's width depending on its slope. This is echoed by observations on the legibility of *Stacked Graphs*, where bands of the same nominal thickness – yet with very different slopes – appear to be of different width [Byron and Wattenberg, 2008]. These considerations are also a concern for Parallel Hierarchies. But since ribbons with a large vertical span are also prone to cross a large number of other ribbons, the applied crossing minimization implicitly also reduces the link heights.

Another often-found consideration in relation to visual clutter is the angular resolution of those lines that nevertheless do cross each other [Ellis and Dix, 2006]. Taking this into account, one might want to reorder so as to also maximize angular resolution, which is beneficial for tracing crossing lines, as it avoids confusing the lines due to narrow crossing angles. Yet, this issue is not as prevalent when ribbons cross each other, as in most cases the ribbons have different widths and can thus easily be identified when tracing them across parts where they overlap. In case of doubt, a simple mouse over highlights any ribbon showing exactly where it originates and where it leads. As Parallel Hierarchies visualization tends to also incorporate axes with an inherent order (e.g., age groups) or a user specified order (e.g., sort by name or value), we keep the categories on those axes fixed and use them as starting points for computing the order of the remaining axes. The effect of reordering the categories along the axes is illustrated in Figure 5.6, which shows how this approach unclutters the view. From our experience, the Sugiyama layout heuristic using the barycentric algorithm reduces crossings on average by 20% to 25%. The additional reduction by postprocessing the barycentric order with the greedy heuristic lies around 1%-2%, but can be up to 7% in rare cases.

Reordering Strategies for the Ribbons

The third possibility to fine-tune the appearance of Parallel Hierarchies is to adjust the vertical order of multiple ribbons connecting from/to the same active subcategory of an axis. This order is not predetermined by the data, yet most axes-based visualizations featuring ribbons use a "source-based" ordering that sorts ribbons along an axis according to their order along the neighboring axis to which they connect. This makes sense as it eases tracing ribbons from one axis to another by giving us a rough idea of where a ribbon should land on the far side: if it originates from a subcategory at the bottom, it will also connect to the bottom of any subcategory on the other axis, and vice versa. As for the rendering order of the ribbons – i.e., which ribbons to draw first and which last – this is only an issue for opaque ribbons that would overplot already drawn ones. For opaque ribbons, one could for example choose to draw wide ribbons first and thin ribbons last. This strategy would ensure that a few thin "outlier ribbons" are not covered up by the wider ones. Or the ribbons could be drawn in the opposite order, if one wants to make sure that a few wide "main trend ribbons" are not drowned out by a criss-cross of hundreds of thin ones. In our implementation of Parallel Hierarchies, this point is of lesser interest, as we use semi-transparent ribbons with alpha blending. Yet in the same vain as the different ordering strategies, we could of course assign different alpha values to emphasize certain ribbons more than others, making sure they are well visible.

5.3 Design Choices

In addition to the base visual and interaction design discussed up to this point, there are additional design considerations to Parallel Hierarchies visualization. This section addresses finer details of the decisions made during the design process. These decisions are mainly based on empirical studies or suggestions in the literature. However, the regular feedback from the customers of the project was considered as well.

2D versus 1D & 3D

As described in section 2.1.4, multi-dimensional data can be displayed in many forms. The common approaches to convey the information and structure of high-dimensional datasets are tabular display of the data, for example by using spreadsheets or visual encoding of the data using graphs or explicit presentations. While the 1D display of data in spreadsheets can be considered as the most precise for reading individual values, it is not the best solution to discover trends or outliers in the data. Further, there are some difficulties to efficiently analyze large graphical representations (2D), like the limitation of the available screen space. Hence, one of our fundamental questions was when to use graphical representations or tables for communicating quantitative information. Stephen Few gives some guidelines on this which are summarized in

Use Tables When:	Use Graphs When:
The document will be used to look up individual values.	The message is contained in the shape of the values.
The document will be used to compare individual values.	The document will be used to reveal relationships among multiple values.
Precise values are required.	
The quantitative information to be communicated involves more than one unit of measure.	

Table 5.1: Summary of when to use graphs or tables according to [Few, 2004, p.46].

Table 5.1 [Few, 2004, p.46]. Another well known problem with using graphs is that when the dataset is large, the visualization tends to use a lot of space and becomes cluttered [Rosenholtz et al., 2005]. As overlapping nodes can make it impossible to understand the data, we have developed a technique for reducing such visual clutter (see paragraph 5.2.4). Also, the tasks gathered from the customers of the project (see section 4.3.2) can benefit from 2D representations, as they require understanding of the structure of the data and the relationships between data items.

Moreover, using a third spatial dimension to convey information was also considered during the design process. Using 3D visualizations to encode extra information used to be considered efficient in abstract datasets. However, there are many issues in using 3D user interfaces summarized in several recent literature reviews [St. John et al., 2001, Cockburn and McKenzie, 2004, Ware, 2013]. The results of all surveys lead to the conclusion that 3D should be used depending on the task and application. For example, it has been shown that 3D visualizations are often used to represent shape of complex objects such as molecules or in general when shape-understanding tasks are involved [St. John et al., 2001]. In other cases, the costs of adding a third dimension need to be carefully analyzed. Due to two main reasons, we did not consider using a third dimension in this work. First and most importantly, occlusion as an important depth cue causes several problems. For example important objects might get hidden and it requires interaction techniques which consequently cost time. Also, understanding the 3D structures of unfamiliar shapes is challenging and might cause considerable cognitive load. Second, perspective distortion causes objects to appear smaller in distance and objects cannot be simply compared as typical perceptual tasks [Munzner, 2014, p.120].

Spatial Axes Orientation

The arrangment of items in space is one of the first and most important decisions to make in the design phase of a visualization. Based on Munzner's classification, there are three ways to orient spatial axes: *rectilinear*, *parallel*, or *radial* [Munzner, 2014, p.144]. In *rectilinear* layouts, items are distributed in the Cartesian coordinate system. The most commonly used visualization with rectilinear layout is scatterplots (see section 2.1.4), which are suited for representing two data attributes. More attributes can be coded with combinations of non-spatial features, but very complex patterns cannot be preattentively processed and the number of features that can be combined effectively is limited. In a *radial* layout, items are distributed around a circle,

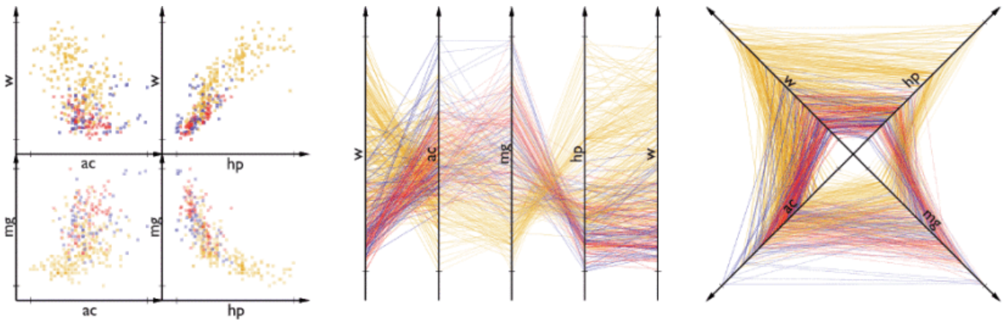


Figure 5.7: Examples of different spatial orientations visualizing multivariate cars dataset with: left) scatterplots, middle) Parallel Coordinates plot, and right) Radar chart, all taken from [Claessen and Van Wijk, 2011].

using polar coordinates to represent several linear spatial attributes. Circular diagrams are not the best approach for large networks and large values. There is very little room for displaying all ribbons, and oversimplifying can compromise the essence of the content. Also, from a perceptual point of view, rectilinear layouts can be more accurately perceived than radial layouts, and radial layouts are often used for representing periodic data [Draper et al., 2009].

The third and most practical approach to effectively represent several data attributes using spatial visual features is *parallel* layouts. In a parallel layout, axes that represent different attributes are organized parallel to each other. Parallel Coordinates is one of the most commonly applied approaches using parallel layouts, which is used mainly for finding correlations between attributes. Figure 5.7 shows an example visualizing a multivariate dataset with all three methods described above.

We decided to use parallel layout in our design, both because of the advantage of being able to visualize many attributes at once, and the possibility to accurately perceive spatial positions in comparison to radial layout. The default version is designed so that axes are placed horizontally and data items vertically (Figure 5.1). This decision is based on datasets from SAP customers. In their datasets, the number of attributes to be compared or analyzed was typically more than the number of items on each axis. However, we also provide the possibility to rotate the visualization vertically.

Single View versus Multiple Views

Dealing with different datasets or one dataset with several stratifications is challenging and can be handled by multiple views or a single view supported by interactive navigation. Navigating within a single view where the display changes constantly requires remembering past views, potentially an issue considering human memory limitations. In contrast, using one view that shows an overview and several others that display details imposes a significant cognitive load on the viewer. Another primary issue with multiple views is finding connection between views. Using lines to connect them is a commonly used solution to overcome this problem although it can also cause occlusion [Ware, 2013, p.344]. Our visualization solution follows a one-view-fits-all approach. The rational for using a single view along with navigation techniques to show or hide details on demand is the importance of keeping the overview and maintaining a sense of orientation. Also, our visualization solution focuses more on representing the relationship between elements versus within elements. Our focus is not the individual items and their attributes, but rather the connections that bind individual items together. Customer requirements include the ability to explore the magnitude and the relationships of individual items. Consequently, following Shneiderman's mantra: "overview first, zoom and filter, then details on demand", Parallel Hierarchies provide an overview by aggregation, the ability to zoom into arbitrary parts while preserving the context, and interactively seeing embedded detail views for individual items via tooltips. In our design, both detailed focus and overview context information are embedded in a single view. Moreover, Parallel Hierarchies were further developed to be capable of showing uncertainty and to be capable of comparing different versions over time. In that case, for handling more data characteristics, top-down exploration is not the only feasible approach and we are following Keim's visual analytics mantra: "Analyse First, Show the Important, Zoom, Filter and Analyse Further, Details on Demand" [Keim et al., 2006]. This is further discussed in section 9.3.

Interaction Techniques

Parallel Hierarchies, are particularly useful when offered with interactive features, enabling the user to manipulate the display to facilitate visual exploration. Considering the interaction taxonomy by Heer and Shneiderman described in section 2.1.5, the following interactions are supported in our solution:

Data & view specification: To specify the data and view of interest, the possibility to choose the visualization and datasets are provided (*Visualize*). To filter out unrelated information and show desired data dimensions to focus on, the data attributes can be selected by a drop-down menu (*Filter*). To reveal patterns or anomalies, sorting as another fundamental operation is provided by a drop-down menu on each axis (*Sort*). The categories can be ordered according to value or description, which disables the minimized intersection option. However, we have not designed any interaction method to support *Derive* functionality as a common method for visual analytics solutions.

View manipulation: Once the view has been created, Parallel Hierarchies views can be manipulated to facilitate pattern exploration and get more insight of the current view. Highlighting as a common and effective solution to make some information stand out is supported by mouse hover (*Select*). The integrated version into SAP Analytics Cloud applies the "blur" method to highlight important information by blurring non relevant information [Kosara et al., 2002]. Another novel interaction technique, which is highly beneficial, is the possibility to focus not only at one category at a time (currently via mouse over), but at two or more categories on the same axis. Showing how their corresponding ribbons propagate across the various axes, where they intersect, and how big these intersections gives Parallel Hierarchies an entirely new capabilities for comparative analyses. One example that illustrates the utility of multi-focus selection is given in Figure 5.8. This idea can be leveraged to better distinguish items from multiple foci as they propagate. For example, Figure 5.8 shows how people with Eastern European ancestors distribute across other data properties, as compared to people with Western European ancestors.

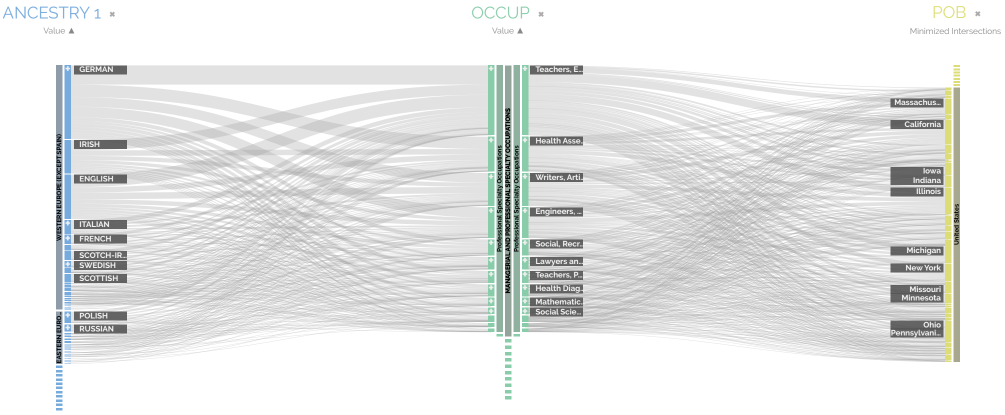


Figure 5.8: Example of the multi-focus selection: along the left axes, two categories have been selected in parallel – Western Europe and Eastern Europe – to allow for the investigation of all people with European ancestors.

Moreover, this example shows, how selecting multiple categories on the same axis can be helpful when working with categorizations that do not align with a particular analysis question.

One of the most important focus-plus-context interaction method designed for Parallel Hierarchies is the drill-down and roll-up operations used for hierarchical decomposition (*Navigate*). Each axis becomes an interaction handle to traverse the hierarchical data category using drill-down and roll-up interaction to tune the granularity of each individual axis to any desired level of detail. We believe that the corresponding navigation cost – which refers to the cognitive effort and required time to set new visual objects – should be less than typical zooming methods. Also, fisheye distortion is another basic navigation strategy used for the displaying large structures in a limited space. In our solution, when the number of items per dimension exceed a threshold, the size of items gets small and their labels are not shown anymore. Yet, when the items are very small (also siblings of the selected ancestors) hovering or selecting them is not simply achievable. To that end, a distortion technique is needed to demagnify contextual regions. Fisheye distortion can be enabled and disabled to overcome this problem by manipulating the display space.

In addition to fisheye, we designed a novel navigation technique called *Accordion* to address the readability of ancestors' names in deep hierarchies. After applying the focus-plus-context method to drill down to a particular level of interest, users must be able to see where they start. The name of currently selected ancestors are shown only for the first 5 levels of the hierarchy, after that the rectangles' widths get smaller for space efficiency purposes and the labels are not shown. With the help of mouse over or fisheye techniques labels are displayed one by one. Our evaluation showed that using those techniques are not always practical due to human memory limitation. To give users the possibility to analyze where they are or where they came from in deep hierarchies we built a novel interaction technique. Users can extend or reduce the width size of rectangles on demand and consequently see their labels by using

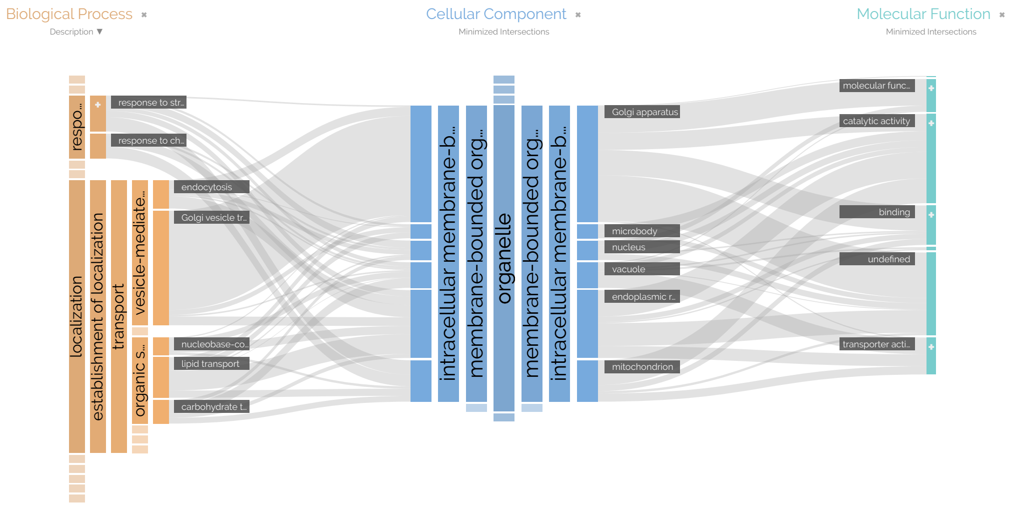


Figure 5.9: Applying accordion interaction method to the middle and left axes of Parallel Hierarchies for better readability of the ancestors' labels.

the mouse roller on one axis at a time. Figure 5.9 shows one example of applying accordion interaction on the left (less extended) and middle (more extended) axes. The categories on the right axis are showing the typical size of the rectangular shapes. From this interaction category, *Coordinate* and *Organize* interaction types were not covered in the original implementation as we followed the single view approach. However, in the final integrated version, multiple views are supported and thus both task types are addressed.

Process and provenance: In visual analytics tools, the process of iterative data exploration plays a crucial role. From this task category, we have supported *Share* task type to export views or data subsets for sharing and revisitation.

Shape of the Visual Features

It is possible to map different data attributes to a wide range of visual variables like: position, color, texture and so on. Each mapping makes some information more distinct and the other information less distinct. Therefore, finding suitable visual variables to use for different data attributes is of crucial importance. As discussed before, we looked for a representation that works better for tasks related to category frequency, and explicit visualizations were picked. Thus, a ribbon-based technique was chosen, which scales better in this regard. For our ribbon-based technique, the parallel approach was preferred, where the ribbons are not split based on a focus dimension. The reason for this decision is the interactive nature of the proposed visualization techniques, where we expect analysts to frequently change their focus dimension. Also, we are interested in analyzing more than three dimensions at a time, at which point the splitting produces too much clutter. The main objective was to embed the quality information in graphical attributes of existing visualizations. Hence, the selection of visual variables was one of the key factors in determining whether the visualization can enable users to interpret the quality information and draw reliable conclusions quickly.

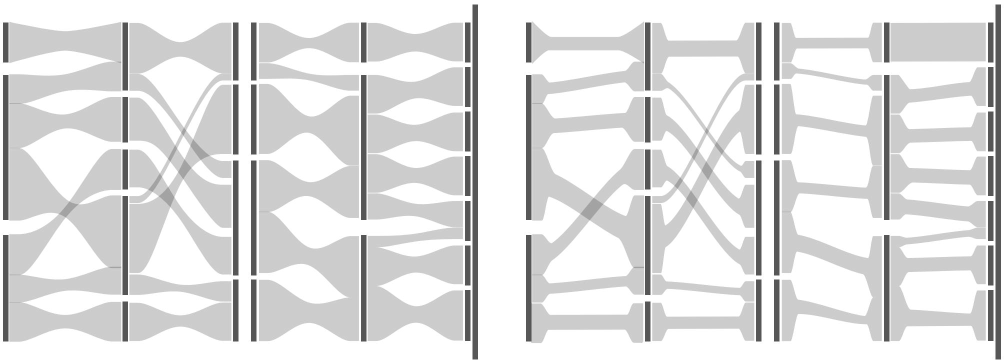


Figure 5.10: Customizing Parallel Hierarchies ribbon design.

The two main visual features of flow diagrams are *nodes* and *links* that connect the nodes (see section 2.3). Based on Munzner's ranking for the effectiveness of different visual variables shown in Figure 2.11, *spatial position* is one of the most effective channels. In the selected visualization type, this channel is reserved for locating items and data attributes on spatial positions. The second most effective channel to convey "ordered attributes" is *length* (1D size), and then after that *angle*, and *area* (2D size). We used *area* for showing the quantitative value on both nodes and ribbons. For the node shapes (the width is fixed and only the height is changing) rectangular area is used, as it facilitates the comparison of different shapes based on graphical perception [Cleveland and McGill, 1984]. Two more advantages of using rectangular shapes in our solution are: first, ancestor items can be efficiently put on right and left side of the nodes, resulting in a symmetric shape that facilitates similarity perception. Second, after drilling down to the level of interest, something can be put out of the scope on top/bottom of the selected items for future navigation tasks. By keeping those items on the screen as smaller rectangles an easily recognizable symmetric *diamond shape* is produced on both sides. Later, for conveying more information such as uncertainty, other novel symmetric shapes are proposed and used.

Regarding the links of the visualization solution, we decided to use curved instead of straight ribbons – specifically cubic Bézier curves. Indeed, smooth paths and contours are known to be easier to follow than straight ones [Ware, 2013, p.191]. This is particularly important, as the ribbons potentially connect all active subcategories of one axis with all active subcategories of a neighboring axis, leading to a cluttered display that can be challenging for tracing individual ribbons. However, to minimize ribbons' intersections and avoid occlusion, different ribbon shapes were designed and tested. Figure 5.10 shows two examples of our customized ribbon design for Parallel Hierarchies. Instead of using the same height on the entire ribbon, we assigned equal height to the beginning and end part and decreased the height of the middle part. Yet, the connectivity and symmetry principles are preserved. After, evaluating our designs with three experienced visualization designers, we decided to stay with the original Bézier curved ribbons as they were unanimously considered easier to follow.

Color Encoding
Mapping color as a non-spatial visual attribute is a very challenging task in the visual design process. Color mapping presents the problem of scalability due to human's visual system and common problems such as red-green color blindness. As the number of colors increases, it gets more difficult to distinguish them. A basic perceptual principle is that less than 12 colors are distinguishable when showing categorical data [Ware, 2013, p.125]. The datasets we researched contain more than 12 categories per dimension. Therefore, assigning different colors to the categories of each dimension cannot be an effective solution. In the first prototype's design, to better distinguish different axes from each other, our particular realization applies a color coding to the axes that assigns different colors to different axes, as shown in Figure 5.1. The colors of the axes follow Paul Tol's categorical color scheme Palette II [Tol, 2012], which is specifically designed to be compatible with both light and dark backgrounds. This makes it easy to plug Parallel Hierarchies into different applications regardless of whether they use a light or a dark user interface theme. The first and open source version of the project covers both dark and light background, and the number of colors assigned to axes was limited to twelve. Indeed, during our co-innovation workshops, SAP customers rarely used more than 6 dimensions for their data exploration.

However, in the final version integrated into SAP Analytics Cloud, users have the freedom to make different choices about encoding with categories, axes, or both together, depending on the size and characteristics of their datasets. In addition, on each axis the path to the initially selected category that is stacked is colormapped by a color gradient (changing saturation). We assign darker shades to the "older" ancestral categories that have been unfolded and explored previously, and lighter shades to the "younger" active subcategories that were just recently added to the stack of categories and that are currently explored. The generated colors depend on the number of drilled-in levels, a shade between the gray color assigned to the ribbons and the premier color assigned to the corresponding axis.

5.4 Applying Parallel Hierarchies

The following describes two use cases out of our industrial scenario in which we applied and evaluated Parallel Hierarchies to show the extendability of the solution. One use case looking at US demographic data, and the other dealing with yeast genomic data. These two examples provide a first impression of Parallel Hierarchies in action in two very different fields.

5.4.1 US Census Data

The US Census dataset has already been briefly introduced in Section 5.1. In this section, we look at a subset of 100k items. The dataset features 68 attributes, some of them numerical, others categorical, and out of the categorical ones a few with a given hierarchy. In this example, we focus mainly, but not exclusively on hierarchical attributes. The view in Figure 5.11 shows three of them: `POB` (indicating the country in which the person was born), `INDUSTRY` (the type of industry in which a person works),

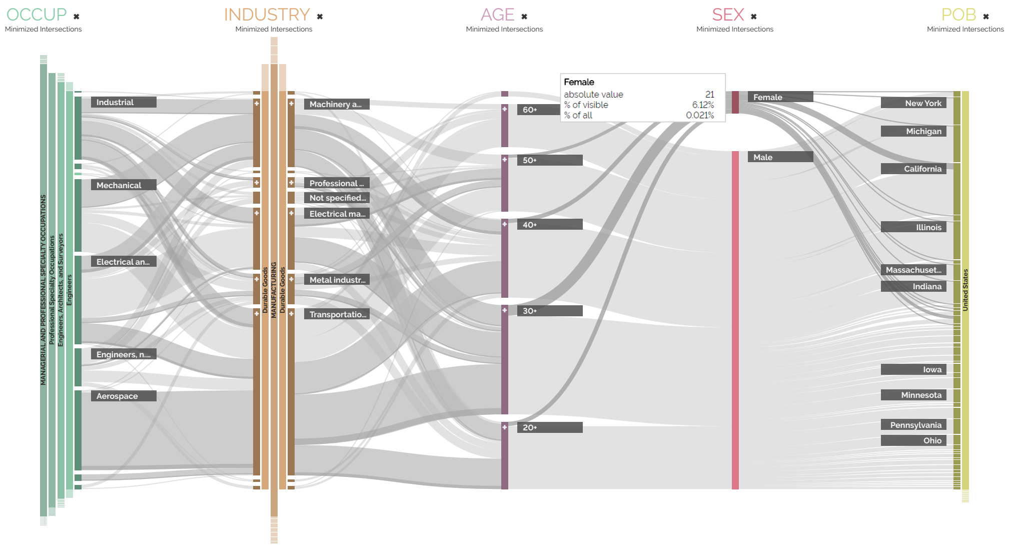


Figure 5.11: Parallel Hierarchies visualization of a 100k sample of the US Census 1990 dataset. The highlight emphasizes the few women among the US-born engineers who work in the production of durable goods.

and OCCUP (a person's primary occupation). Furthermore, we added a hierarchization of the numerical category AGE into age groups and the flat categorical attribute SEX. The aggregate values represented by the height of the categories and by the width of the ribbons are the number of people.

The purpose is to investigate the relation between different work environments as signified by the attributes OCCUP and INDUSTRY, and the people's corresponding AGE and SEX. Hence in Figure 5.11, we rearranged the axes so that OCCUP and INDUSTRY are in the center of the view. Moreover, we avoided placing POB and OCCUP next to each other as they both have many subcategories which would produce a high number of ribbons. In addition, we adjusted the ordering strategies for some axes. For example, the POB axis is ordered descending according to the aggregate value, so that it supports the indirect look-up of states from where many or few people stem. In contrast, the AGE axis benefits from an ordering by description, so that the different age groups are ordered ascending from youngest to oldest, thus supporting a direct look-up strategy. In this example, we drilled-down to the people born in the US, who work as engineers in any industry manufacturing durable goods. We immediately see the small number of women working in this area. Upon hovering over this category, the women in this view are highlighted and we learn from the tooltip that only 6% of all employees in this area are women – 21 in total. From the spread of the highlighted ribbons, we also see that these women are mainly in their 30's and about a third of them stem from California. When changing the aggregate value from the number of people to their income, which is not depicted in the figure, one can further find that while there are roughly 13 times more men than women working in this field, these men make 20 times the money that the women make.

The visualization was showcased to demographers from the Max Planck Institute for demographic research. They work with such data on a daily basis, and they immediately noticed the generality and capability of the Parallel Hierarchies approach in their

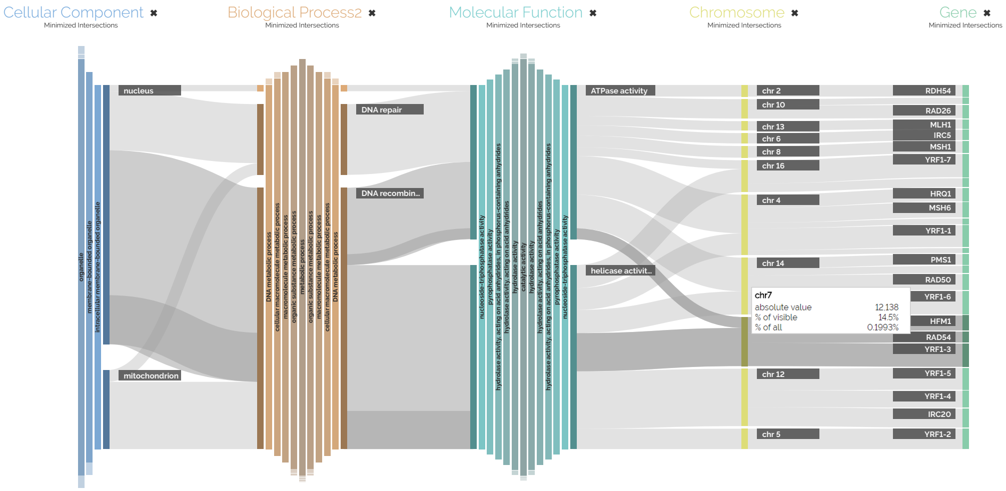


Figure 5.12: Parallel Hierarchies visualization of the yeast (*S. cerevisiae*) genome and its Gene Ontology annotations. The view was drilled-down to display only genes that relate to the DNA metabolic process in intracellular membrane-bound organelles whose function involves nucleoside-triphosphatase. The current highlight emphasizes the subset of genes that are encoded on chromosome 7 – which pinpoints exactly three out of the more than 3,000 genes.

domain. In particular for questions involving cross correlations, they were eager to use the visualization – e.g., to investigate how the degree of education, place of birth, and level of income influence life expectancy and fertility rates as predicted by different statistical models using different parameter settings.

5.4.2 Yeast Gene Ontology Annotations

Understanding how cells function requires an understanding of the molecular parts of the cell, its genes and the proteins they encode. While ribbon-based visualizations for biomedical use cases have been proposed in the past (see e.g., *StratomeX* [Lex et al., 2012] or *CooccurViewer* [Sarikaya et al., 2016]), they are still far from commonplace in the toolbox of biomedical researchers. In this use case, we consider a dataset of 3,813 genes of the *S. cerevisiae* (yeast) genome, which we obtained together with their chromosomal locations from the Saccharomyces Genome Database (SGD, `https://www.yeastgenome.org`). For the categorical properties, we used their annotations from the Gene Ontology (GO, `http://www.geneontology.org/page/downloads`). These annotations form three hierarchies: *cellular component*, *biological process*, and *molecular function*. As shown in Figure 5.12 (right side), Parallel Hierarchies lets us visualize the yeast genes and their chromosomal localization. In addition, we can display their Gene Ontology annotations as hierarchical axes in Figure 5.12 (left side).

In this use case, we seek to identify genes involved in a specific biological process, having a given molecular function, and located in a particular cellular compartment. Specifically, we want to find the genes that are localized in the intracellular membrane-bound organelle (cellular component), that bind nucleoside- triphosphatase activity (molecular function), and that are involved in DNA metabolic process (biological process). This task is challenging because it requires simultaneously navigating several hierarchies and performing a joint selection. Identifying such genes would typically require writing a database query or custom script. There are two obstacles to this: First, the actual terms and concepts to be queried against have to be known in the first place, which is not obvious when dealing with an ontology consisting of tens of thousands of concepts. Second, assuming that the parameters of the query are known, it remains necessary to write and perform the query itself – something that requires a certain degree of expertise. In the words of one of our biologist interviewee: "*Asking a biologist to write a complex database query is equivalent to asking a computer scientist to run a gel electrophoresis, possible in theory but unlikely in practice.*" Online tools such as genome browsers or Gene Ontology browsers exist to aid biologists, but these cannot visualize the interplay between different concepts, let alone allow to specify the sought genes across different hierarchies.

A first advantage of Parallel Hierarchies is that drilling-down into a specific hierarchy reduces the drill-down choices available for the other hierarchies – a key feature to facilitate interactive exploration of the data. For example, drilling down to a specific cellular compartment can decrease the number of corresponding genes by two orders of magnitude but also reduces the number of relevant molecular functions and biological processes. Indeed, when we asked biologists to test our visualization, they mentioned that "*It is great that selecting a specific term deep in a hierarchy can teach us what other terms are relevant in other hierarchies.*"

Another advantage of the joint visualization is that the relationships between hierarchies becomes explicit. For example, Figure 5.12 shows the preponderance of genes encoding helicases in the Yeast cell nucleus. Moreover, by rolling up the *molecular function* dimension we can visualize the proportion of nucleus genes that function by means of nucleoside-triphosphatase versus all genes localized in the DNA metabolic process. The highlighting feature of Parallel Hierarchies allows us furthermore to highlight specific chromosomes (here chromosome 7) and see which nucleoside-triphosphatase activity, intracellular membrane-bounded organelle genes are located on that chromosome. This facilitates interesting and complex observations such as that all genes involved in nucleoside-triphosphatase activity found in the nucleus except for HFM1 are encoded on chromosome 7, and are directly involved in DNA recombination. Biologists to which we showed this interactive querying mentioned that "*exploring such datasets often requires that we ask an expert such as a bioinformatician to do the analysis for us, write our own scripts, or go through some tedious manual search. With this solution I just have to look and click.*"

The ribbons' height – which is proportional to the number of genes – helps to convey the relative number of genes involved in different biological processes, molecular functions, localized on different cellular compartments, and encoded on different chromosomes. Besides the interactive visual representation of such a dataset, this use case illustrates how Parallel Hierarchies can help to find a needle in a haystack by providing a faceted search interface through the individual drill-down/roll-up of the different hierarchical axes.

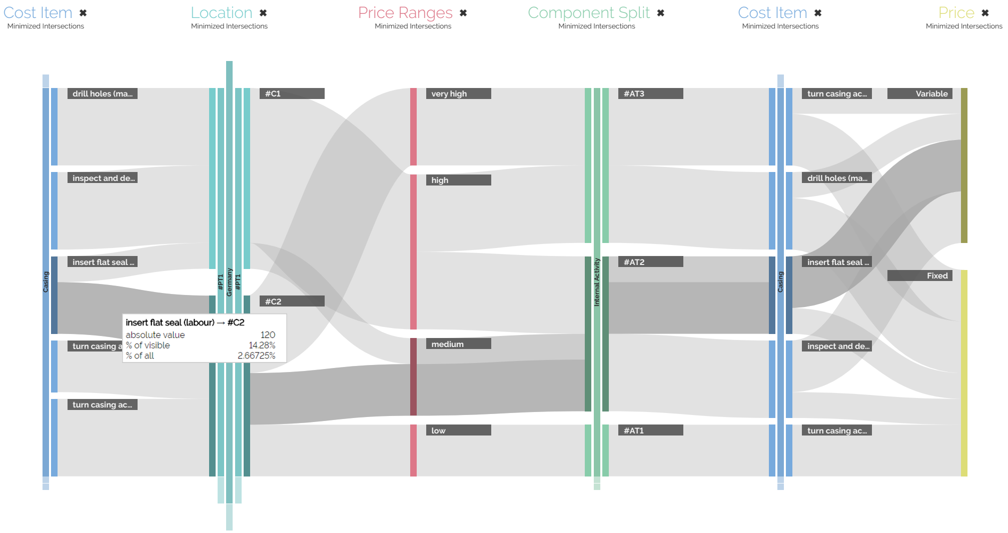


Figure 5.13: Parallel Hierarchies visualization of the costing structure of an industrial pump with 92 items. In this view, we show the particular costing structure of the casing. To ease observing its distribution across the other data properties, the axis “Cost Item” has been added twice. The current highlight emphasizes the labor of inserting a flat seal, which is in the medium price range and entirely variable in its price, as wages are often a fluctuating factor.

5.5 Evaluation

Our particular realization of Parallel Hierarchies, as it was described in the previous chapter, was initially developed as an interactive visual analysis technique in the domain of product costing. Product costing involves analyses where one wants to break down the overall costs of a product along various aspects, such as cost types (e.g., labor, materials, patent fees, and taxes) and product components (e.g., frame, tires, electronics, engine, and seats) to find cost drivers and thus potential savings when designing a new product. The requirements for a visualization, which is able to support such interactive analyses, were established over the course of multiple formative user studies described in section 4.3. As none of the used diagrams by the customers of the project is a good fit for the complex nature of the costing data, we exposed the participants in a second study to Treemaps [Johnson and Shneiderman, 1991] and Sankey diagrams [Riehmann et al., 2005] to show the hierarchical break-down of costs along the categories. These visualizations were met with great enthusiasm by the costing experts and the Parallel Hierarchies technique was the logical combination of the Treemap hierarchy display (albeit now shown as Icicle Plots) and of the interconnection among categories with ribbons in the Sankey diagrams.

Since these initial studies were conducted, more than 30 customers and partners who partake in that program have given input on the design of Parallel Hierarchies, which took about 2 years from start to finish. It is in this setting that we conducted an *empirical qualitative user study* [Tory and Möller, 2004] of the resultant Parallel Hierarchies technique, on which this section reports.

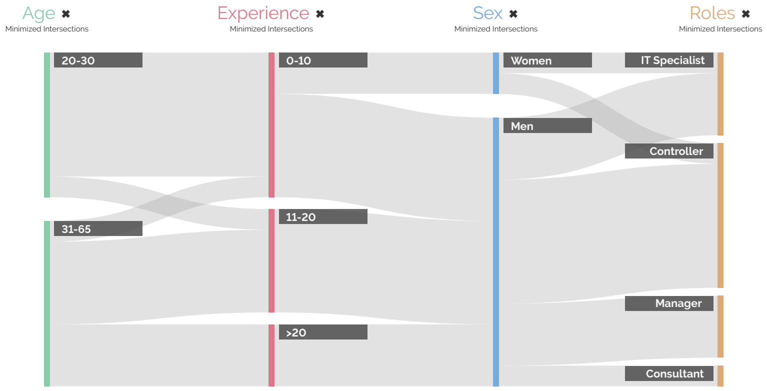


Figure 5.14: Structural break-down of the group of study participants.

5.5.1 Setup of the Evaluation

The Data: The general properties of the dataset used in the evaluation follow the costing scenario: The set of data items contains the individual product parts, including intangible parts such as software licenses and measures for quality control. The hierarchical categories are defined over the categorical attributes of these product parts – e.g., material type, place of production, and the part/whole relationship that via multiple stages forms the overall product from these parts. While *material type* does not strike one as being of hierarchical nature, individual materials can in fact be hierarchically grouped – for example, into raw materials, packaging materials, services, etc. The numerical aggregate was the cost. For our evaluation we used a small realistic dataset for an industrial pump with 92 parts and 6 attributes associated with each part. This dataset was based on a real-world dataset from one of our customers, but was slightly modified by us to obscure its source and to inject a known ground truth for the participants to find. The dataset is shown in Figure 5.13.

The Participants: We conducted our evaluation with 15 product costing experts from 9 companies in individual 1-hour sessions over the course of two days. The participants were recruited during a customer workshop and all of them had no prior visualization experience beyond standard charts as they are available in most spreadsheet and business intelligence software. Their application backgrounds are mainly the automotive and machine building industries, where they work in various roles from IT specialists to managers. This group of participants had the following structure, which is also shown in Figure 5.14:

> *Age: min*=21, *avg*=37.7, *max*=61
> *Years of Experience: min*=1, *avg*=12.2, *max*=35
> *Gender:* 2 female, 13 male
> *Roles:* 6 controllers, 4 IT specialists, 3 managers, 2 consultants

The Tasks: Six tasks were chosen to cover *topology-based* and *attribute-based* tasks, which we “borrowed” from the field of graph visualization [Lee et al., 2006]. While being a completely different area of visualization, graph visualization tasks work well in our case, as Parallel Hierarchies include topology (the hierarchies and the structure defined by the ribbons) and numerical attributes (the aggregate values).

Thus choosing tasks, which demand to traverse the topology and to identify and compare numerical attributes in any combination, seems to provide a good sample of possible tasks performed with Parallel Hierarchies in real-world applications. After formulating the tasks, all of the tasks were reviewed by project members who are experts in the field of product costing in order to make sure that they have enough granularity and reflect the users' daily base tasks. The exact questionnaires of the evaluation can be found in appendix A. Concretely, the six tasks were:

T1 Which country does the main part of the *Drive* come from?

T2 What is the price range of most *Shaft* sub-items? And from which country are most of the items in that price range?

T3 Which item among those with a *manual* price source has the most sub-items?

T4 Which component split has only *variable* cost portions?

T5 What percentage of the cost for the *Casing* comes from *Overheads* component split?

T6 What percentage of the total cost stems from the *fixed* cost portion?

The domain language masks in particular the topology-based nature of some tasks. Yet, for example, it is evident that T2 requires a traversal of ribbons or that T3 requires a drill-down into the hierarchy to identify the most sub-items, which cannot be gleaned from the height of the bars as these encode their cost and not their quantity.

5.5.2 Procedure of the Evaluation

The goal of our study was to check for comprehension and interaction hurdles with the visual representation, as well as to observe how users without prior visualization knowledge actually use Parallel Hierarchies and what they think of it. To that end, our study followed a defined procedure that consisted of five steps:

1. Background questions to establish participants' levels of experience and corporate role
2. Explanation and exemplification of the Parallel Hierarchies visualization and its interactive features
3. Practical warm-up for the participants to familiarize themselves with the technique
4. Performing the six product costing tasks outlined above
5. Wrap-up questionnaire to gather overall user experience indicators and open feedback

Data was gathered through *semi-structured interviews* [Lindlof and Taylor, 2011] for step 1, through *think-aloud* protocol [Boren and Ramey, 2000] for the practical steps 2 to 4, and through the standardized *User Experience Questionnaire* (UEQ) [Laugwitz et al., 2008] for step 5. Questions arising during the practical steps were noted down and answered afterwards, as helping participants does not introduce bias to their overall experience.

5.5.3 Results from the Evaluation

The evaluation yielded three types of results: our observations while the participants were solving the tasks, numerical results from the UEQ questionnaire, and the users' free-form feedback.

Observations

Our first observation was an apparent relation between age and learning curve, by which we mean "how fast the user will learn the set of skills required to perform tasks with a given visualization" [Lallé et al., 2016]. After dividing the participants into two age groups, we observed that the first group of 7 participants (from 20 to 30 years old) performed the tasks with more ease and confidence than the second group of 8 participants (from 31 to 65 years old). Where younger participants had no major problems using Parallel Hierarchies after our short 5-minute explanation, older participants still needed a lot of guidance in using it. This observation is supported by the fact that participants from the younger age group asked on average for help during one task, whereas participants from the older age group required our help on average during 2.5 tasks. Surprisingly, this observation was indeed aligned with the age of the participants and not with the years of experience – i.e., it did not really matter for how long a participant had already been working with the current tools of the trade and there was apparently nothing they needed to "unlearn" first to be able to learn the new visualization.

The participants' questions with regard to Parallel Hierarchies were on one hand geared towards understanding the visual mapping – particularly the meaning of the ribbons. Most participants first tried to solve the tasks by looking at and comparing the heights of the active subcategories, which makes sense given their familiarity with bar charts. Only when that did not work, they invested the extra efforts of switching their mental map to parse and trace the unfamiliar ribbons. One of the participants mentioned that "*First I do my best to find the answers by the bars, then tool-tip information. If none of them works, then I will try to understand the flows.*"

Questions with respect to the interactive adaptation of Parallel Hierarchies came usually up when the participants were expected to adjust the hierarchy levels using drill-down and roll-up operations, but did not realize this possibility or what could be gained from it. This was mainly the case when working on task T3, where most participants got stuck when trying to solve it with a singular view – i.e., they were trying to find one perfect view that answered the question. Yet this was not possible in this case, as to solve this task they were required to drill-down and roll-up on three different subcategories of the "Cost Item" axis, and then to compare the values they found for each subcategory. This indicates that the participants perceived drill-down and roll-up mainly for adjusting the visualization until the sought information comes into view, which can then be analyzed. It was not part of their repertoire of strategies to use them by continuously going back and forth between hierarchy levels. To form a larger picture of an insight that cannot be pinpointed on a single level of detail. In particular, this latter observation is highly promising for Parallel Hierarchies, as it opens up a way to gain such insights in this particular domain.

Last but not least, we made the observation that participants were not eager to change the order of axes. They started the first task by placing the "Cost Item" axis on the left side and except 2 participants (again younger ones), all of them kept it always on the left side. Even after playing with the reordering functionality of axes, before

Dimension	Avg. Value	Std. Error	Alpha
Attractiveness	2.095	0.770	0.86
Perspicuity	1.893	0.944	0.87
Dependability	1.946	1.253	0.75
Efficiency	2.054	0.701	0.74
Stimulation	2.000	0.679	0.80
Novelty	2.482	0.616	0.79

Table 5.2: Overall UEQ test results from our study.

writing down their final answer they would place the “Cost Item” axis on the very left side of the screen. This refers to a well-known spatial memory concept from cognitive psychology discussed in detail by Ware [Ware, 2000]. We observed how fixating axes help participants to explore data quicker and gives them more confidence about their answers.

UEQ Results:

The UEQ test uses 26 adjective pairs which are assigned to six user experience factors: attractiveness, perspicuity, efficiency, dependability, stimulation, and novelty. Each adjective pair (e.g., from *attractive* to *unattractive*, from *predictable* to *unpredictable*, or from *easy to learn* to *difficult to learn*) uses a seven point Likert scale where the polarity is determined randomly for each pair. Table 5.2 shows the averaged results for the six user experience factors in a range between -3 (negative) to +3 (positive). Overall, all factors received a value around +2 with only novelty being ranked slightly higher. This relative outlier is probably due to having conducted this study with non-experts in the field of visualization, who may not be as current on the visualization state-of-the-art.

These results by themselves give a general indication that Parallel Hierarchies is deemed useful and appealing to the participants. In comparison to the benchmark dataset, which currently contains UEQ test data from 246 scientific and industrial user studies of software products with overall 9,905 participants [Schrepp et al., 2017], Parallel Hierarchies rank for most factors among the top 10% of the studies. Only for perspicuity, Parallel Hierarchies is in the second tier of 75% - 90% and thus among the top 25% of the products included in the benchmark dataset. While the UEQ test results are difficult to interpret just by themselves, they will allow comparing future enhancements and alterations of the Parallel Hierarchies technique to this baseline.

Feedback

From the free-form feedback given by the participants, two main themes emerged: 6 out of the 15 participants suggested to show a table on the side of the view with further detail information and 7 out of the 15 participants wanted to have the possibility to export snapshots for reporting purposes.

The first suggestion of adding a tabular display was somewhat counter intuitive at first, as the participants had already struggled with all the visual details that go well beyond their customary bar chart. Yet, it also confirms what we established in the beginning, namely that costing experts are familiar with spreadsheets and tables. It seems that at least until this user group has gained a good understanding of what Parallel Hierarchies can and cannot do for them, they feel uneasy with just the visualization and no means to look at the numbers behind it. Thus, adding such a table as a way to ease the transition from their accustomed software tools to Parallel Hierarchies seems like a good path to establish Parallel Hierarchies in this domain.

The second suggestion – taking snapshots – further underlines this aspect: being asked about this rather common feature request, the participants revealed that they did not see themselves using Parallel Hierarchies in their day to day costing analyses. For their daily analyses, they have their standard tool chain with which they are familiar and which is deeply embedded in their companies' IT infrastructure and general workflows. Instead, they wanted to use Parallel Hierarchies mainly for communication and presentation purposes – in particular with "the higher-ups" who did not understand or care for complicated spreadsheets. They believed that Parallel Hierarchies is a perfect way to break down their analyses for the decision makers. One of the participants who works as a controller said "*I need exactly something like this visualization to communicate my discoveries in the dataset with my managers for future cost optimization decisions*". The most popular idea brought forth by 1/3 of the participants was to have a tablet version of the visualization, which would preserve the interactivity of the technique. This way, they could even investigate different costing alternatives together with the decision makers using a touch-based interface.

In addition, we asked the participants for the top 3 use-cases that they could imagine for their daily work. Overall, we gathered 22 use-cases. In particular, participants with management and consulting roles were eager to use the visualization on a daily basis. Three companies partaking in the co-development program and thus being introduced to Parallel Hierarchies requested it as an add-on for the SAP product costing suite.

5.5.4 Validity of the Evaluation

When designing Parallel Hierarchies, we had one fundamental use case in mind: breaking down a numerical aggregate along multiple hierarchical categories. We then evaluated Parallel Hierarchies with a rather homogeneous group of people from the product costing domain and with tasks that work towards the goal of breaking down a large cost aggregate. Their homogeneity is underlined by high Cronbach's alpha-coefficients for the UEQ results given in Table 5.2, which lie between 0.74 and 0.87. These indicate *acceptable* to *good* scale consistency among participants [DeVellis, 2012, p.109], meaning that our participants mostly agree on the interpretation of the 26 UEQ adjective pairs. The evaluation results show that for this scenario, Parallel Hierarchies work well notwithstanding certain adoption and learning hurdles.

However, another evaluation was designed to validate the solution with the customers of product costing and get overall feedback from them. The main purpose was to see if the customers have real use-cases for the visualization, and whether our prototype fits the requirements gathered in the requirement phase of the project.

Procedure The designed prototype was shown during one of the co-innovation customer workshops to 30 customers from 12 different companies. The solution was shown and explained with a realistic use-case from product costing to the participants. Then the customers were asked in one-hour group discussion meeting to answer the following questions:
1) How can this visualization help you? What are realistic use cases?
2) What dimensions would be useful to analyze in this visualization?
3) How would you imagine integrating this visualization in your costing tool?

Results The entire group including customers, partners, and representatives from the go to market and the development team summarized the use-cases and their impressions on the new data visualization. In a one hour meeting, 22 use-cases were gathered for the visualization from different industries with diverse and very specific requirements and data characteristics. After two follow-up meetings, 14 of them were documented as a unique and applicable use-cases to the visualization solution. In the wrap-up session at the end of the workshop the visualization was considered as the highlight of the two-days workshop by more than 80% of participants. The feedback we received showed the applicability of our solution to the product costing and how it fits their daily work requirements. After this final evaluation round, the co-innovation customers requested integration of the visualization into a general platform (not costing specific).

Yet when discussing Parallel Hierarchies with the biologists in the context of the Gene Ontology use case from Section 5.4.2, we found that one could also use the visualization quite differently – namely as an interface for *dynamic queries* [Shneiderman, 1994], *faceted search* [Tunkelang, 2009], or *cross-filtering* [Weaver, 2010]. While being the same fundamental visualization technique, this usage context is quite different as its aim is not to break-down the item distribution across categories, but to find data subsets with very specific properties within the dataset. Because of that difference, we are reluctant to transfer our evaluation results to this very different usage scenario that comes with different requirements and tasks. To establish the suitability of Parallel Hierarchies in this scenario will require conducting a separate evaluation.

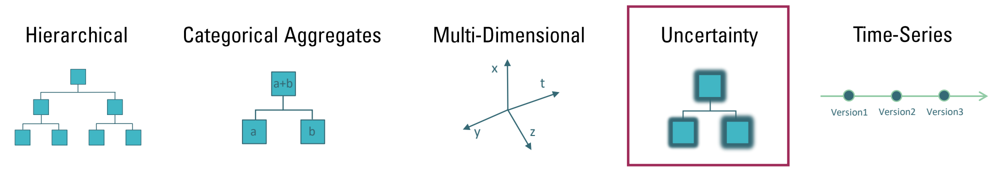
5.6 Summary and Outlook

In an effort to close the gap for multiple hierarchical categories, this chapter introduced Parallel Hierarchies — a visualization technique that combines the best of both worlds: Parallel Hierarchies combine the arbitrary number of axes from Parallel Sets with interactive Icicle Plots at each axis. Each axis becomes an interaction handle to traverse the hierarchical data category using drill-down and roll-up operations to tune the granularity of each individual axis to any desired level of detail. We first defined the visualization problem that Parallel Hierarchies are designed to resolve based on the requirements discussed in section 4.3.

After introducing the visual and interaction design of Parallel Hierarchies, we detailed the design choices made during the project. The utility of the newly introduced visualization technique is showcased with demographic and biological use cases. Then, we reported on our qualitative user study in an industrial scenario with 15 costing experts. Finally, the chapter concludes with discussing the validity of the evaluation.

Our work demonstrates the value of co-innovation design and qualitative evaluation in real world scenarios, especially for evaluating different designs and features of a novel visualization technique. This process helped not only in finding usability issues, but also in quickly assessing the clarity of the conceptual design. It gave us the chance to gather 14 use cases from 9 companies. During this experience, we spotted major issues early on and received valuable guidance to iteratively refined the solution.
As for disseminating Parallel Hierarchies, we worked with the team behind SAP Analytics Cloud – SAP's commercial BI framework – to integrate Parallel Hierarchies as a standard diagram type. Furthermore, as a result of our user study reported in Section 5.5, several customers of the project requested Parallel Hierarchies as a product feature in SAP's *Life-cycle Costing* application. The first integration as an add-on for customers from the automotive industry has been finalized and others will follow.

Moreover, our aim is to cope with more challenges such as visualizing uncertainty and time-dependency in costing data. Both play important roles for particular use cases of product costing and other business intelligence scenarios. In the next two chapters, we describe the visual approaches we pursued.



Chapter 6

Visualizing Uncertainty in Flow Diagrams

In this chapter, we describe our solution for visualizing both data (described in section 4.4.1) and uncertainty. In the more general context, visualization designed for business intelligence applications must visualize hierarchical data structures associated with uncertainty values. We focused on flow diagrams, which fulfill the basic requirements of product costing as explained in section 4.

In the following, after describing the motivation of this research, we introduce a case study conducted to establish uncertainty visualization requirements. Then, we describe three design techniques to convey uncertainty on the ribbons, and five approaches for the nodes of flow diagrams. Moreover, we report on two user studies conducted to evaluate the proposed techniques. A new approach was designed using the insights obtained from the first user study but also from additional feedback gathered from experts in both product costing and data visualization. The application of the new design was tested on Sankey diagrams, Parallel Sets, and Parallel Hierarchies. Finally, we will summarize the results of our user study and discuss possible future work. Parts of the research presented in this chapter have previously been published in [Vosough et al., 2019, Vosough et al., 2017b].

6.1 Introduction

Unfortunately, real-world data contain inaccurate information and unreliable data values. In many business intelligence applications, data uncertainty needs to be addressed. Visualizing the impact of this uncertainty is considered a critical research task [Johnson et al., 2006]. Researchers have introduced numerous methods to visualize different kinds of data structures including uncertainty for single values. However, a holistic view that combines both data structures and uncertainty information, in an integrated manner, without producing visual clutter, is still a challenging task [Pang et al., 1997]. Decision makers need able to evaluate data uncertainty. Indeed, high risk decisions based on uncertainty information may have a large impact on market success and need to be well-founded. Previous research shows how the quality and performance of decisions are improved by making uncertainty information available to users [Nadav-Greenberg and Joslyn, 2009].

Product costing is one application where accuracy in data is of great importance. Regardless of how powerful and well-designed a product is, it cannot be successful without the ability to provide predictable time and cost schedules [Marshall and Meckling, 1962]. These tools are needed by the responsible personnel such as controllers, engineers, and purchasers, in order to reduce the whole life cost of a product. One of the key success factors of product costing directly depends on the quality of the decision-making process, which involves not only assessing the available information, but also the confidence in that information. Risk assessment and risk reduction is of high importance during the costing process.

User studies concerned with the effectiveness of uncertainty visualization exist, but most available solutions are domain-specific since the visualization process is unique from task to task, e.g. [Blenkinsop et al., 2000, MacEachren et al., 1998]. The structure of costing data is hierarchical and there are many available techniques to visualize large graphs [Von Landesberger et al., 2011]. However, there are no readily available solutions that incorporate uncertainty in existing graph visualization systems. Recent research on Parallel Coordinates to depict uncertainty [Xie et al., 2006, Cedilnik and Rheingans, 2000] motivates us to find general solutions that can be applied to different types of flow diagrams. Flow diagrams have the potential to visualize uncertainty information without losing the focus on the data structure. In addition, after reviewing current solutions, we noticed that although a variety of different methods are recommended for visualizing uncertainty, the evaluation of these new techniques for different tasks and data types is not yet well covered. Therefore, we believe that this is an open topic that requires further research.

6.2 Uncertainty in Product Costing

A product's cost depends on a large number of characteristics, features, or competitive prices – i.e. dimensions. Being able to quickly assess and reduce costs for a product or its parts is critical. There are different methods and tools to calculate costs but the vital role of uncertainty in data is often ignored, especially with regards to visualization solutions. This section first describes the impact of uncertainty in the whole process of lifecycle costing. Second, we discuss an interview that was conducted to discover the main sources of uncertainty in product costing.

6.2.1 Background

As described in section 4.3.2, one of the customers' primary requirements was to have an efficient solution to assess the reliability of the overall cost calculated from price sources with different uncertainty causes (T4). Building upon this work, in this chapter we try to find an efficient visualization technique (adaptable with other solutions) that shows data and uncertainty together in a holistic view.

At the beginning of product development, there is not enough stable information for calculating a costing goal. Over time, more and more data enhance the available information. However, several other factors might increase uncertainty, such as the inflation in material prices or the impact of currency exchange rates on the final cost of a product. This makes cost estimation across the lifecycle of a product extremely challenging. Therefore, applying powerful visualization techniques to the lifecycle costing process can have a crucial impact on enhancing the user's understanding of the underlying data. In this process, actual prices are unknown for a defined period of time until all uncertainties become clear at Start of Production (SOP).

Figure 6.1 shows a simple lifecycle total cost graph of an exemplified phone production process. The graph displays three cost curves against the project time: a worst cost, a realistic cost, and a best cost. This is a simple and customers' common solution to present data in combination with uncertainty. Usually, these three sets of costs are required for the analysis of data and for the awareness of the price fluctuations. Since this diagram cannot cover the special requirements of product lifecycle costing with regards to multiple dimensions and the involvement of all cost items, a novel approach is required for representing errors and uncertainties. Users might need to evaluate the impact of uncertainty for different parts of a product's final cost. Each node of the graph has an explicit value along with the established amount of uncertainty during the product's lifecycle. At this point, it is apparent that the product's costs are not evenly spread through the graph and time. But with such diagrams, end users cannot identify *individual items*, it is only possible to check the final product cost. Clearly, the visual presentation has to be improved to provide more details.
In this part of the research, we are working on finding a visualization for quickly understanding the uncertainty at particular points in time. The problem description shows the issue of defining a suitable graphical presentation for data uncertainty. Such a presentation should on the one hand be based on data characteristics and temporal evolution within a complex graph-based structure, and on the other hand support the user's reasoning process and goals.

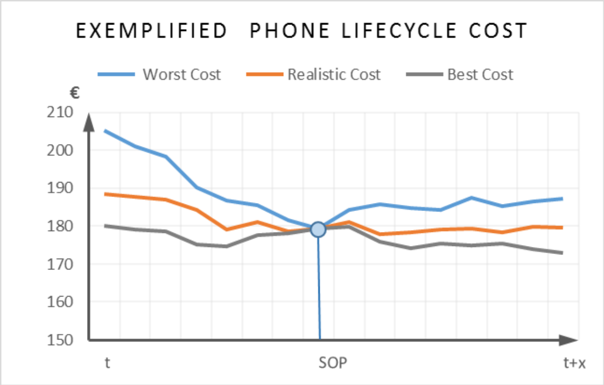


Figure 6.1: Example of a phone's total lifecycle cost graph.

6.2.2 Main Causes of Bad Quality in Costing Data

Several different sources of uncertainty exist in a product lifecycle. In order to detect the most important issues and to find a suitable visualization, a solid understanding of the sources of uncertainty is necessary. First, the main uncertainty factors through the development of a product in SAP Prodcut Lifecycle Costing was determined by interviewing 10 domain experts. The questionnaire can be found in appendix B.

Procedure

Ten people were selected for the interview, mainly from the solution management team who are in close collaboration with customers and responsible for establishing and maintaining customer relationships, for producing marketing collateral, and for supporting sales. In addition, we interviewed the chief product owner who is the development team lead and the business architect of the team (see section 4.2). The interviewees were asked about: (1) The questions they expect to be answered by this visualization with regards to the quality. (2) The attributes that are the main causes of uncertainty in product costing data. (3) The common ways of defining the level of uncertainty for each of the attributes, and how to calculate the overall uncertainty. (4) The importance of visualizing the uncertainty in relationship with items cost. (5) The necessity of showing the exact amount of uncertainty for each cost item, and the importance of comparing different items best, realistic, and worst cost values.

Results

All main causes of uncertainty in product costing data based on our participants' responses were summarized in 12 dimensions (question 2): (1) Quantity of items, (2) Price sources, (3) The price setting date, (4) Not well defined prices, (5) The structures are out of date, (6) When the master data are temporary, (7) Missing values for a specific item, (8) Funny looking numbers (like 666666), (9) Suppliers unknown or changes, (10) The bigger the unit of measure, the more precise should the number be, (11) Missing calculation sheet, (12) Level of customization. Skeels et al. categorize these factors adopted in different kinds of uncertainty sources, ranging from measurement precision to completeness issues and inference (see [Skeels et al., 2010]). Each source is calculated differently (question 3) and this complexity prevents a complete representation of the data and is rarely needed by end users (question 5). To reduce the complexity of the uncertainty visualization, aggregation of uncertainty values is used, called confidence level. As research shows, scalar values are typically used to present the uncertainty [Cedilnik and Rheingans, 2000]. However, based on the interview results, only 5 discrete values are often shown to end users (Table 6.1) that

	Confidence Level	Range
1	Very Low	± 100 %
2	Low	± 50 %
3	Medium	± 20 %
4	High	± 5 %
5	Very High	± 0 %

Table 6.1: Examples for cost confident level ranges.

suffice for an intelligible visualization of uncertainty, and further details must be shown only on demand. The way of calculating these values is different for each product costing application and our visualization solution needs to adapt accordingly. The results of this interview are the foundation for our user study described in the next section.

6.3 Visualization Concepts

Based on the concept presented in section 3.2, we concluded that a holistic view containing both data and uncertainty was needed for the domain of product costing. The challenge is to include all information without overwhelming users.
We decided to embed the uncertainty information in existing visual attributes. Therefore, the selection of visual variables is one of the key factors in determining whether the visualization can enable users to interpret the uncertainty information and draw fast and reliable conclusions. The first prototype was based on Sankey diagrams as a simple flow diagram representing one hierarchical dimension of the data. In our solution, both rectangles and flows of the flow diagram represent the cost of components in larger component groups. Hence, on the very right side, singular components are displayed and their cost is subsumed in component groups to the left (see Figure 6.3).

The research question was how to include the uncertainty (also called confidence level in section 6.2.2) that our users are accustomed to in their current spreadsheet solutions. To this end, we considered different possible solutions for flow diagrams that address the requirements and tasks of users in product costing. Two visual features of Sankey diagrams can be used to convey the uncertainty information without adding more visual elements to the visualization: the nodes or the links. As we described in section 3.2.2, in previous work uncertainty values were depicted using stacked bar charts [Correll and Gleicher, 2014, Gschwandtnei et al., 2016], which could be applicable to the node rectangles. Moreover, Tak et al. put forth seven techniques to show uncertainty using lines [Tak et al., 2014] that inspired us towards the visual solution design of the flow diagrams ribbons.

Figure 6.2 shows the list of possible visual variables to use for conveying uncertainty on flow diagrams (see section 2.2.1). Considering our goal to help users quickly identify data with very low or high uncertainty, we decided to use visual variables that can be preattentively processed (see section 2.2.2). From the list of visual variables shown in Figure 2.11, we selected five attributes plus texture. All five variables are from *form* and *color* category shown in Figure 2.12, taking into account that spatial position of visual features cannot be changed in flow diagrams. All six attributes are examined on both flow and node attributes and *length* was not considered, as the length of visual features (height of nodes and thickness of flows) are representing quantitative values and cannot be changed. Similarly, *orientation* of both flows and nodes has to be preserved, since it is representing the connection between items and cannot be manipulated. Moreover, *texture* is taken out of the list of possible solutions as it is not a preattentive variable. All in all, *area, color value*, and *color hue* are preattentive visual attributes that will be used in the context of this work to depict uncertainty on

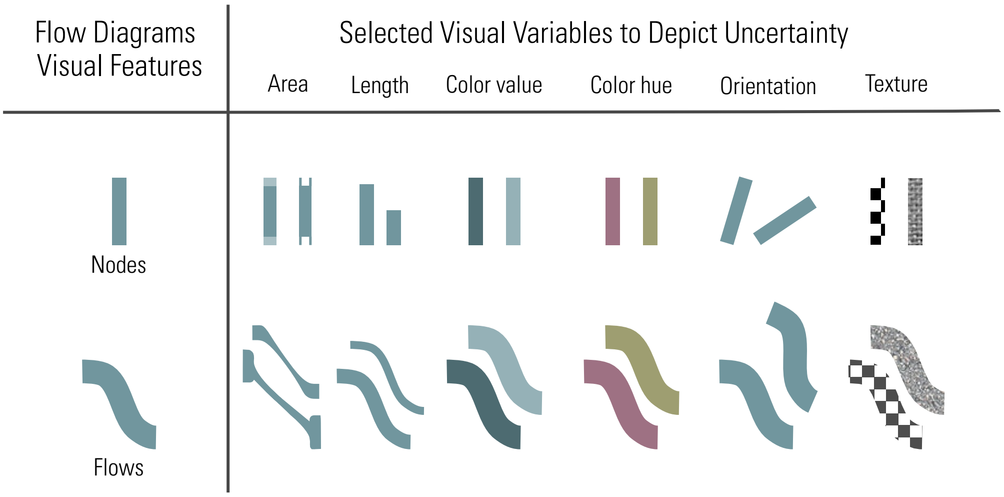


Figure 6.2: Varying different visual variables of flow diagrams visual features.

nodes and ribbons of flow diagrams. However, *area* is defined by changing the shape of visual features symmetric and in a way that original length of items are preserved. In the following, we first discuss the techniques that we used to visualize uncertainty on ribbons, then we present the user study we conducted, and then, we discuss the study results in detail.

6.4 Uncertainty Visualization using Ribbons

In the first stage, we intended to keep the information conveyed by the nodes regarding the average values clear and use the ribbons that are a more prominent feature in the visualization to convey uncertainty. In this section, we will represent the approaches we used for ribbons, and how the user study we conducted led us to the final solution.

6.4.1 Selected Visualization Techniques

After analyzing the techniques shown in Figure 6.2, we identified three suitable to be adapted for the flows depicting costing uncertainty: Gradient, Color-code, and Margin. These methods can be grouped according to the categories of [Boukhelifa et al., 2012] and [Pang et al., 1997]:

Color-code is a color-based technique, which represents uncertainty by modifying the attributes, in this case by applying different colors to the flows (5 shades of blue). Flows with a lighter shade of blue have more uncertainty and those with darker colors have less uncertainty[1] (see Figure 6.3.a).

Gradient is a focus-based technique, which shows the amount of uncertainty by modifying the geometry. We change the shape of flow ribbons by applying blurriness around the flow of an items. The gradient is calculated based on the worst possible cost and visualized as a blurry border. Therefore, flows get more blurry as the uncertainty of the item increases (see Figure 6.3.b).

[1] Colors are derived from www.ColorBrewer.org, retrieved on 10.07.2019

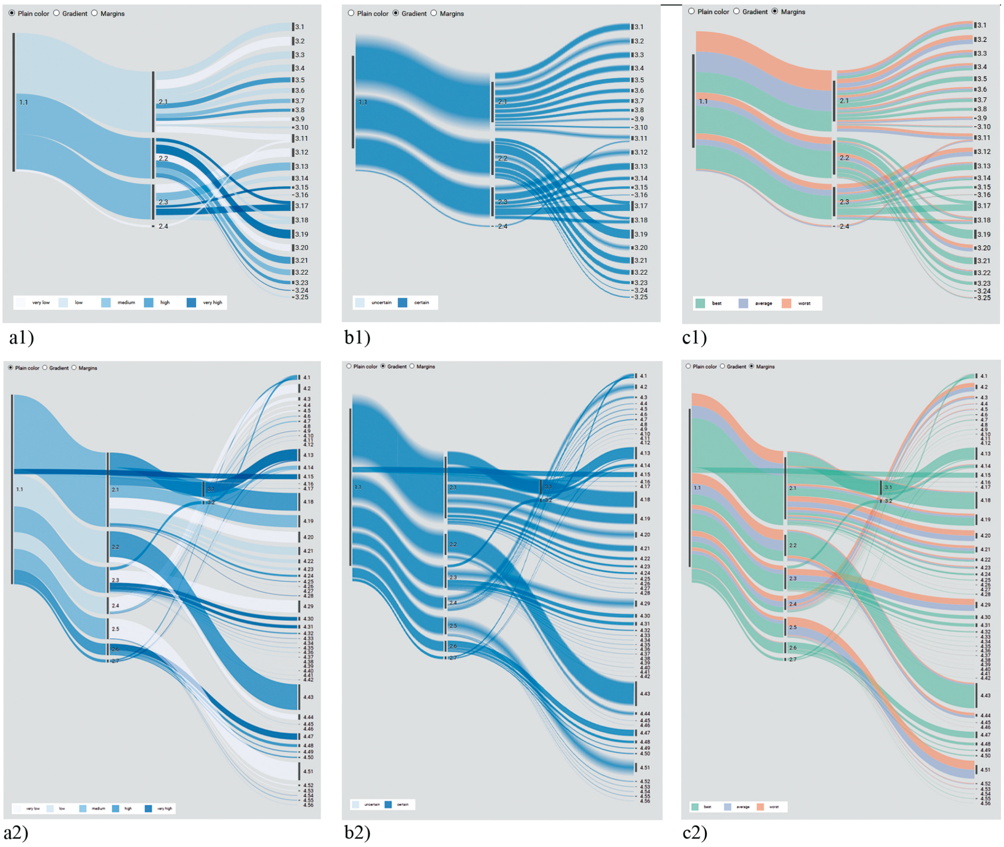

● Plain color ○ Gradient ○ Margins
○ Plain color ● Gradient ○ Margins
○ Plain color ○ Gradient ● Margins
very low
low
medium
high
very high
uncertain
certain
best
average
worst
1.1
2.1
2.2
2.3
2.4
a1)
b1)
c1)
a2)
b2)
c2)

Figure 6.3: Test setting with three interfaces and different datasets: Color-code (a1 & a2), Gradient (b1 & b2), Margin (c1 & c2).

Margin is a geometry-based technique, which adds geometry (margins) around the flow surface. This solution is similar to Hunter's work in GIS research [Hunter, 1999] and customers are used to this kind of visualization for the total cost of a product (see Figure 6.1). This method represents uncertainty by adding two margins to the flows in order to represent the best and worst case values in addition to the average value. Green encodes the best case, the colors green and blue the average case and red the worst case (see Figure 6.3.c). Similar to Gradient, the thickness of the flows including the margins indicate the worst possible cost, while the rectangular nodes keep showing the average cost.

There is a distinct difference between Color-code and the other two solutions. The flows in Color-code only show the absolute uncertainty levels and users are not further supported in judging the impact on the average cost. Gradient and Margin visually assist the user by showing more information about this impact. However, Color-code is less visually cluttered and can be more suitable for easy tasks such as identifying different levels of uncertainty.

All solutions are designed to give a quick impression of data uncertainty and then find important items to investigate further. Although a tooltip was integrated in all solutions, our study does not depend on the tooltip information and was not made available to the participants.

6.4.2 Study Design and Procedure

In the user study, we compared the three uncertainty visualization techniques that are described in section 6.4. These three interfaces were implemented in JavaScript using jQuery[2] and the visualization library D3[3].

Hypotheses
The following task-related hypotheses were tested in our user study:

H1 *Color-code is more accurate for low complex tasks such as finding a specific level of uncertainty*. In these tasks, the most and least certain items have to be identified, hence, dark and bright colors should support best in solving this task.

H2 *Gradient method is more accurate for finding the impact of uncertainty and cost of a part on the total cost*. The blurry parts are larger as the size of the flow increases. This results in more blurriness when costs have a higher impact on the parent item, which intuitively conveys uncertainty.

H3 *Margin method is more accurate when comparing the potential impact of different parts on the total cost*. This is important when users need to discover if the impact of uncertainty can yield a cheaper or more expensive cost in comparison to the other items. Because the best and worst cases are shown in the flows, the margin heights can be easily compared with each other.

Data
Two realistic datasets were prepared with different complexity levels. The dataset with low complexity described a simplified version of a realistic industrial pump dataset containing 30 items. The second dataset with more complexity was from one of the customers of the SAP product costing project and contained 67 items. Both datasets were anonymized, using numbers to label the items. This also facilitated the answering of questions in the questionnaire since items could be identified via a number.
We limited the complexity of the datasets so that they could be effectively plotted using Sankey diagrams. Larger real-life datasets in product costing contain even more items, but we propose to use aggregation techniques as is done in Parallel Hierarchies to make them usable for our approach.

Participants
The study included 32 participants (8 females) in the age range of 21 – 65 (mean value M = 32.16, standard error SE = 1.39). They came mainly from two different backgrounds: The first group were people from SAP with costing background and the second group were chosen from the university with background in data visualization. According to their personal assessment (scale from 1 = less experience to 5 = very good experience) most of them did not have much experience with product costing (scale 1 – 2: 75%, M = 1.97, SE = 0.20). Most of the participants had experience with data visualization (scale 3 – 5: 68.75%, M = 3.19, SE = 0.24), but less experience with flow diagrams or Sankey diagrams (scale 3 – 5: 37.5%, M = 2.25, SE = 0.24).

[2] https://jquery.com/, retrieved on 10.07.2019
[3] https://d3js.org/, retrieved on 10.07.2019

Methodology

We used an online form for the user study, which was sent by email to the participants. In this form, we explained the basics about product costing, the underlying dataset, and the three different visualization methods (*Color-code*, *Gradient*, and *Margin*). After this introduction, the participants had to solve 30 tasks for each interface. We randomized the order in which the visualization methods were used to solve the tasks. The task order was not randomized, but we used a randomization method to change the item costs for each question. Hence, participants were presented the same data structure, but with different costs and different uncertainty values.

Based on the results in our interviews mentioned in section 6.2.2, we divided these 30 tasks into three task blocks to test our hypotheses (5 tasks with low complexity data and 5 tasks with higher complexity data):

Search_Certainty Finding product parts that are least/most uncertain, e.g. "What is the most uncertain sub-item of item 2.1?"

Search_Impact The impact of uncertainty and cost of a part on the total cost, e.g. "Which item's uncertainty from level 3 has most impact on item 2.1?"

Comp_Impact Comparison of the potential impact of different parts on the total cost, e.g. "Can item 2.1 be cheaper than item 2.2?"

Before the participants started to solve the tasks with each interface solution, an example question with the description of the correct answer was provided to train users on how to solve the task types with each visualization technique.

Each task was presented on a single page of the online form, containing the question and a picture of the visualization. For *Search_Certainty* and *Search_Impact*, users had to enter the correct answer, i.e. the label of the item, for *Comp_Impact* users answered yes or no. For each task we counted errors to measure the error rate. Every correct answer was given a score of 0 and every wrong answer the score of 1.

After the experiment, participants were asked to complete a questionnaire for each interface. We used the standardized User Experience Questionnaire (UEQ) to measure the User Experience (see section 5.5) [Laugwitz et al., 2008]. After filling out the UEQ, the participants were asked to give their feedback on the following questions:

- "What are the benefits and drawbacks of each methods from your point of view?"
- "Which solution do you prefer to work with?"
- "Do you have any other comments or ideas?"

6.4.3 Results

Accuracy was subjected to 3 (*interface: Color-code, Gradient, Margin*) x 3 (*task: Search_Certainty, Search_Impact, Comp_Impact*) repeated measures ANOVAs. Furthermore, we used a 3 (*interface: Color-code, Gradient, Margin*) x 6 (*factor: Attractiveness, Perspicuity, Efficiency, Dependability, Stimulation, and Novelty*) repeated measures ANOVA to compare the subjective rating from the questionnaires. We did not distinguish between high and low complexity datasets in the evaluation, since we did not see any significant difference between the results. That means for each Task 10 questions were considered instead of 5.

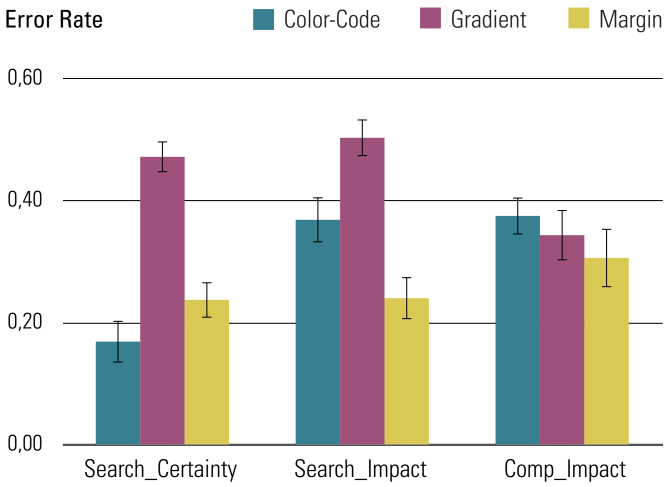


Figure 6.4: Error Rate for *Color_Code*, *Gradient*, and *Margin*.

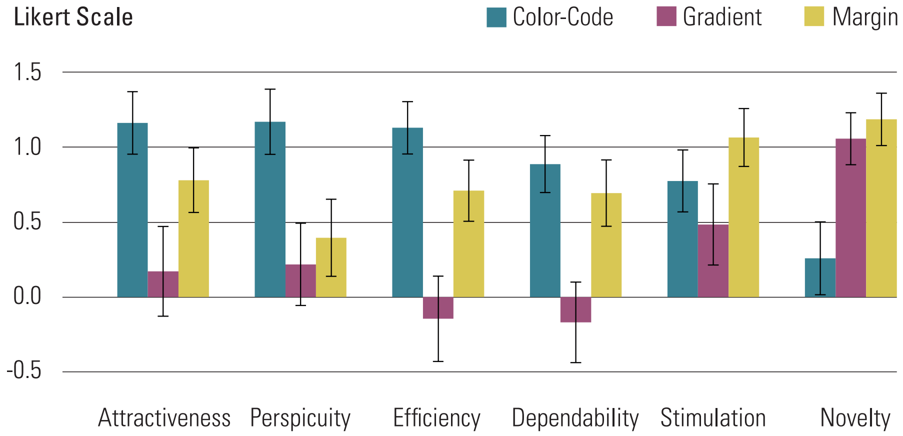


Figure 6.5: UEQ for *Color_Code*, *Gradient*, and *Margin* with a Likert scale range from -3 to 3.

Error Rate

In terms of error rate, there was a significant main effect for *interface*, $F(2,63) = 26.092$, $p < 0.001$. In pairwise comparison, *Gradient* had more errors than *Color-code* and *Margin*, both $p < 0.001$, but there was no significant effect between *Color-code* and *Margin*, $p = 0.336$ ($M_{Color\text{-}code} = 0.304$, $SE_{Color\text{-}code} = 0.019$, $M_{Gradient} = 0.440$, $SE_{Gradient} = 0.020$, $M_{Margin} = 0.261$, $SE_{Margin} = 0.023$). Furthermore, there was an interaction between *interface* and *task*, $F(4,124) = 9.665$, $p < 0.001$. In *Search_Certainty*, *Gradient* had more errors than *Color-code* and *Margin*, both $p < 0.001$, but the difference between *Color-code* and *Margin* was not significant, $p = 0.281$. In *Search_Impact* all differences between the interfaces were significant: *Gradient* had more errors than *Color-code* ($p = 0.002$) and *Margin* ($p < 0.001$), and *Color-code* had more errors than *Margin* ($p = 0.025$). In *Comp_Impact*, no differences between the interfaces were detected, all $p > 0.475$ (see Figure 6.4).

User Experiences Questionnaire

The evaluation of the questionnaire shows a significant main effect for *interface*, $F(2,60) = 3.643$, $p = 0.032$. In pairwise comparison, *Gradient* had lower values than *Color-*

code, p = 0.048, but there was no significant difference between the other pairs ($M_{Color\text{-}code}$ = 0.897, $SE_{Color\text{-}code}$ = 0.166, $M_{Gradient}$ = 0.269, $SE_{Gradient}$ = 0.238. M_{Margin} = 0.805, SE_{Margin} = 0.191). Furthermore, there was an interaction between *interface* and *factor*, $F(10{,}300) = 13.406$, $p < 0.001$ (see Figure 6.5). In terms of *Attractiveness* and *Perspicuity*, there were significant differences between *Color-code* and *Gradient*, both p = 0.014, and *Color-code* was rated better than *Gradient*. With regard to *Efficiency*, significant differences between two pairs were found: *Gradient* had lower values than *Color-code*, $p < 0.001$, and *Margin*, p = 0.050. Regarding *Dependability*, *Color-code* had significant better values than *Gradient*, p = 0.004, but there were no significant differences between the other pairs. The factor *Stimulation* showed no significant differences between all pairs. In terms of *Novelty*, significant differences between *Color-code* and the other two interfaces could be found: *Color-code* had lower values than *Gradient* (p = 0.013) and *Margin* (p = 0.002).

6.4.4 Discussion

Regarding our hypotheses, the study showed that H1 holds true. The Color-coding method works best in the task for searching certainty values. Users made significantly less errors in comparison to *Gradient* and performed equally good in comparison to *Margin*. In contrast, H2 was not fulfilled. Although we hypothesized that *Gradient* would perform better with the task of identifying impacts, the study data showed the opposite. In fact, with both *Margin* and *Color-code*, users made significantly less errors. *Margin* proved to be the best method with regards to the error rate for this task. Finally, the study results did not reject H3, however, no significant differences in the three methods could be found. The high error rates in *Gradient* can be explained: there were three questions that all participants answered wrong. The problem was that a very cheap item with high uncertainty was the correct answer, which is only displayed by a very thin flow. Hence, participants did not recognize the correct answer and selected a more expensive item with less uncertainty (see fourth flow in Figure 6.6).

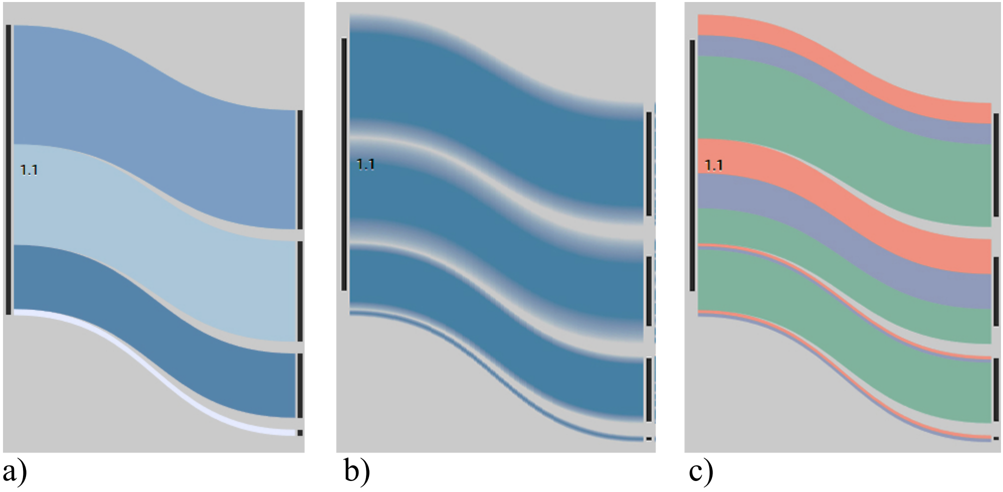


Figure 6.6: Comparison of the same data values with: a) Color-code, b) Gradient, c) Margin.

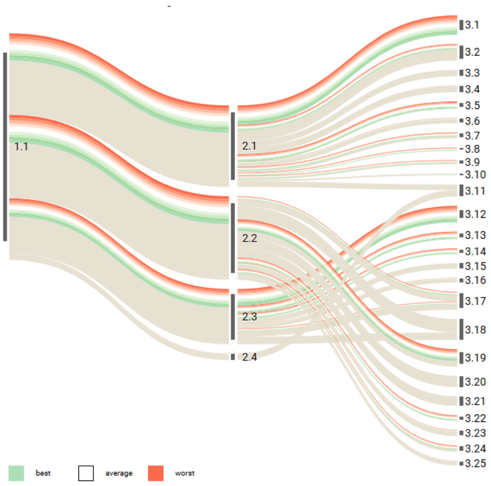


Figure 6.7: The new approach for Sankey diagrams ribbons based on the user study results.

The UEQ is in accordance with the other findings, showing that *Gradient* received the worst ratings. Only regarding novelty, *Color-code* was rated worse than *Gradient* and *Margin*. Overall, there was no significant difference between *Color-code* and *Margin*. With regards to all factors in the UEQ, *Color-code* was rated better than *Margin* except for stimulation and novelty, which influence the hedonic quality criteria. Users were finally asked to choose their favourite solution in the questionnaire. *Margin* received 13 votes, *Color-code* 11 votes and *Gradient* 7 votes. This direct feedback supports the results of the UEQ.
Subjective feedback regarding all three methods contained comments regarding *Gradient* becoming too blurry and not working well with very thin flows. One user also stated that "the blurred look makes it very unpleasant to look at". While some users openly stated that they found Gradient the worst solution, it was also mentioned that it is the best way to solve the *Search_Impact* task – which would support H2. For *Color-code*, users mentioned that less information about uncertainty is conveyed. For instance, a user stated that: "in color-code you just see the uncertainty but not the impact which is a very valuable information for cost reduction". However, many users stated that it was very easy to understand and fast for the *Search_Certainty* task. Although users found it hard to evaluate the actual values of best, average, and worst costs, it is not as cluttered as the other solutions and following connections between items is easier.

In contrast to *Color-code*, users mentioned that *Margin* was more difficult to learn in the beginning, but once it was mastered, it offered the most information in an intelligible way. *Margin* was especially recommended for the *Comp_Impact* task, which supports H3. On the one hand, the different colors were appreciated, although the choice of colors was disputed. On the other hand, *Margin* was considered as being very cluttered with three different colors belonging to the same item: "margin method offers the most detail, but can become a little bit cluttered, [which is] not so good for quick assessment, but better for details, [hence a] combination of color and margin would be interesting".

6.5 Revised Visualization Approach using Ribbons

The feedback obtained in our user study suggests the design of a hybrid approach that leverages the benefits of the three techniques. This novel hybrid approach is based on the margin method but uses discrete gradients for visualizing the best and worst cases. We believe this new approach will address the criticism that the choice of colors in *Margin* received (see 6.4.4). In addition, it will exploit the intuitiveness of *Gradient* to capture the uncertainty without distracting users by the blurriness in this solution. In the following, we describe two use cases in which the hybrid solution is applied on Sankey diagrams and Parallel Sets.

6.5.1 Application to Sankey Diagram

First, we applied our new solution for visualizing uncertainty in flow diagrams to Sankey diagrams. Figure 6.7 shows one example applying the hybrid approach on the pump dataset that was briefly introduced before. In this small dataset, we consider ten discrete gradient values showing the uncertainty on ribbon flows, ranging from green (best case) over white (average case) to red (worst case). In addition, the height of black rectangular bars represents the *average* value of each item.

The new solution was shown to 5 participants of the user-study individually in an informal session. All participants found this new approach easier and faster to understand and interpret in comparison to the three methods used in section 6.2.2. Another key aspect to consider are details-on-demand techniques [Shneiderman, 1996], integrated in our solution without changing the representational context. For instance, we included a tooltip that shows the exact values of cost and uncertainty along with an “ambiguated bar chart” [Olston and Mackinlay, 2002].

6.5.2 Application to Parallel Sets

Second, as a second application of our technique, we applied the hybrid approach to another type of flow diagram called Parallel Sets described in section 3. Sankey diagrams are a perfect solutions to show one dimensional hierarchical data. However, Parallel sets are common solutions for understanding the relationship between different non hierarchical dimensions of the data along with their quantities (via the ribbons). Here, twenty discrete gradient values are generated to depict the uncertainty. Figure 6.8 shows one example of applying our technique to the ribbons of Parallel Sets visualizing selected data from the same industrial pump and its different dimensions such as country of origin, price range, and price type.

As described in section 6.2, each costing item in the data structure has one uncertainty value, but the uncertainty value for the other attributes of the data is not available. To calculate the impact of cost items uncertainty on different dimensions, the *weighted average* uncertainty is calculated for each category. This solution can be extremely useful for data analysts, as it reveals some hidden relationships in the data that could not be easily captured otherwise. For instance in Figure 6.8, it is evident that the items produced in China have the most uncertainty, which are mainly in the high or very high price ranges. Further, after China, Romania and then Germany have

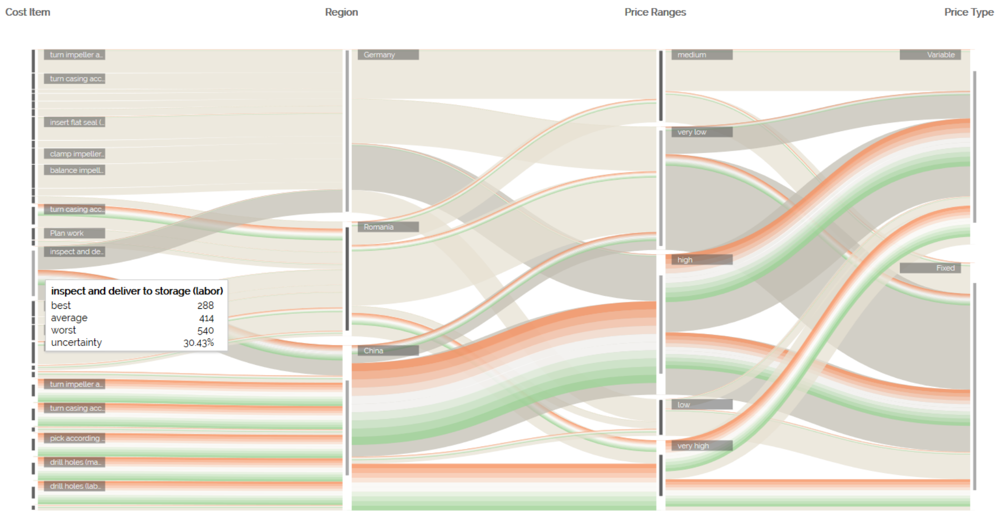


Figure 6.8: Applying the new approach based on the results of the user study to Parallel Sets.

less unreliable products cost. In this example, each item has two types of price. The fixed or variable cost, or sometimes both together. Figure 6.8 depicts the fact that the amount of uncertainty in the fixed prices are much more than variable prices. Similarly to the Sankey diagrams solution, the visualization is augmented with suitable interactions. Hovering with the mouse over one item highlights all connected items and ribbons, in combination with tooltip information to show the best, average, and worst case price.

6.5.3 Application to Parallel Hierarchies

Finally, building upon the Parallel Hierarchies technique introduced in section 5, we show how to convey uncertainty in costing data. Our first choice of color maps was mainly based on colors used by SAP customers to depict uncertainty. In this phase, to make the solution more generic and applicable to other datasets visualised with Parallel Hierarchies, we reconsidered our choice of colors. The new colors are based on a recently developed Value-Suppressing Uncertainty Palettes (VSUP) containing bivariate palettes that are proven to be better perceptually distinguished. The VSUPs univariate quantitative color maps are sufficiently far apart because the color resolution is reduced in the particular areas where uncertainty is high [Correll et al., 2018].

Figure 6.9 shows one example of applying our technique to the ribbons of Parallel Hierarchies with the new color palette. It illustrates selected data from the same industrial pump and its different hierarchical and categorical dimensions. The final result using the new color palette was shown to costing domain experts and they mentioned that what can be seen with our solution was discovered by them after years of working with costing data. They believe our solution can mainly facilitate the tasks of data analysts and help them discover trends in the data and make *uncertainty-aware decisions* regarding their future tasks. In the next section, we go beyond the modification of the ribbons in flow diagrams and investigate the visualization of uncertainty using the node rectangles.

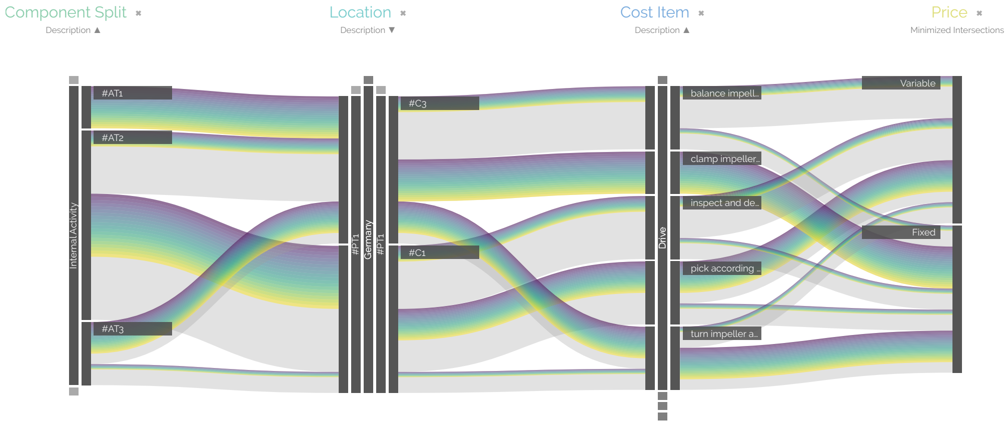


Figure 6.9: Applying our new approach to depict uncertainty using the value-suppressing uncertainty palettes to Parallel Hierarchies ribbons.

6.6 Uncertainty Visualization using Nodes

Up to this point, we focused mainly on visualizing the uncertainty using the ribbon flows of flow diagrams. Another dominant visual feature of flow diagrams are the rectangular nodes (see section3.2). After applying the uncertainty on ribbon flows of bigger datasets or on Parallel Sets and Parallel Hierarchies, we noticed the necessity of presenting the overall uncertainty of each data item directly on the nodes, and not only via the ribbons. This helps the analysts to depict the data uncertainty as a whole. Therefore, we designed five visualization approaches to represent the best, average, and worst case values in very simple and intuitive ways on the rectangular nodes. In the following, we first discuss the visual design of each solution and then the evaluation with eight experts. The goal was to select one option for our product costing application.

6.6.1 Visual Design of Nodes

Figure 6.10 shows the five different alternatives to easily convey the impact of uncertainty values on data items. In all five solutions, the fully filled rectangle in the middle shows the best price and the height of the filled box plus the added shape on one side represents the average price. The overall height including both sides shows the worst price. There are different design approaches for visualizing uncertainty on rectangular bars. For example, Gschwandtnei et al. considered six different visual encodings of uncertainty for temporal intervals: gradient plot, violin plot, accumulated probability plot, error bars, centered error bars, and ambiguation [Gschwandtnei et al., 2016]. For applying uncertainty to rectangles, we excluded gradient based solutions as the bars are very small and changing the transparency of items does not help to distinguish differences. Moreover, the rectangles with curved beginnings or endings like violin plots do not fit to the overall design of our flow diagrams.

All of our design solutions are symmetric. As in our scenario the difference between the best and average case is always equal to the difference between worst and average uncertainty values. The *Diamond* concept in Figure 6.10.a uses two convex rectangles on top and bottom of the bars to indicate the area in which the uncertainty

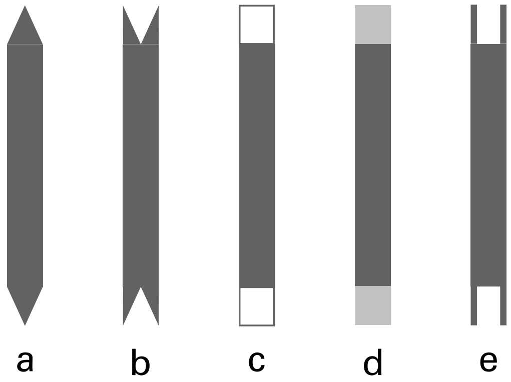


Figure 6.10: Five designed approaches to visualize uncertainty on flow diagrams nodes: a) Diamond, b) Butterfly, c) Border, Blocks, d) Filled Blocks, and e) Fork.

value varies. We designed the other methods slightly differently to overcome the main problem of the first solution. The ribbon flows were disconnected from the bar because the sum of the height of ribbons connects to the bars are equal to the worst case value not the best case. The *Butterfly* method shown in Figure 6.10.b solves the problem of discontinuity by using concave triangles on both sides of the rectangular bars. They also perform better when the number of items on one axis increases. As the number of items gets bigger, the gaps between items get smaller but because of the concavity design of this solution, the items can be still distinguished easily.
The *Border Blocks* solution in Figure 6.10.c draws two white rectangles around the average value with dark outline. The advantage of this solution is the consistency of the shape of the bars. Another alternative for this solution could be using a lighter gray color without drawing the outline around the uncertainty area called *Filled Blocks* (see Figure 6.10.d). Other than the previous solution, the distinction between individual items, especially small ones, is difficult because the rectangles are stacked on top of each other. The last designed approach combines the benefits of solutions b and c (*Fork*, see Figure 6.10.e). First, the shape of the rectangle is kept consistent for every item regardless of its overall size. Second, the small gap at the top and bottom of the rectangle helps to distinguish the individual items from each other.

All methods were applied to the Parallel Sets and Parallel Hierarchies solutions and evaluated with 8 visualization designers. Figure 6.11 shows one of the examples used for the evaluation representing the same dataset from Figure 6.8, but this time by using Fork method to depict uncertainty on rectangular nodes as well.

6.6.2 Expert Evaluation

Eight design experts were asked to provide their professional opinions about the designed visualization approaches for nodes of flow diagrams. The prototypes were shown to them individually and they were asked to evaluate the design and tell us how easy it is to work with different solutions, distinguish the uncertainty on different items, and compare the uncertainty values. Each participant spent about half an hour to understand the topic and compare different methods. They were asked to give a score between 0 to 2 for each prototype. A score of zero meant that this design was rejected for our use case. A score of one was given if the design was found suitable

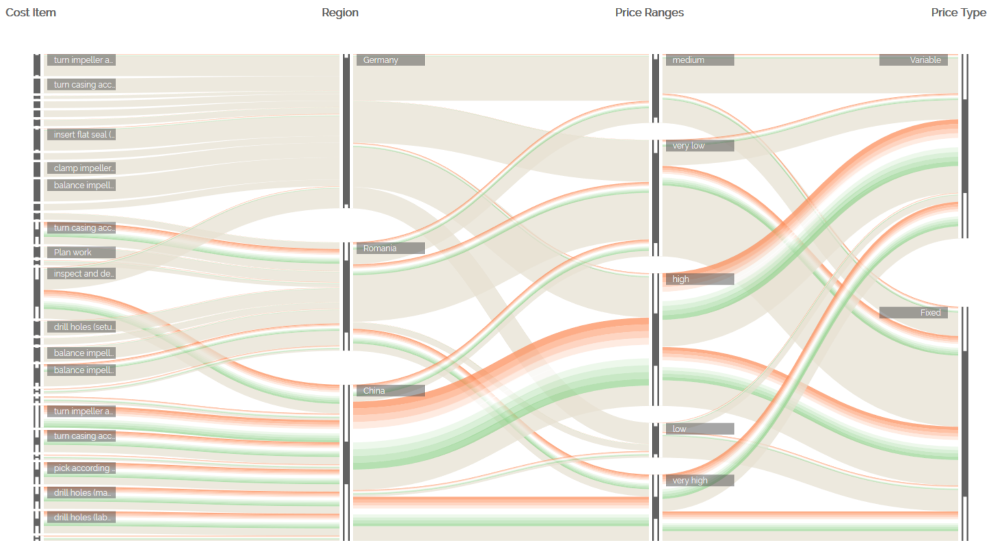


Figure 6.11: Applying the Fork method to depict uncertainty on Parallel Sets nodes.

and easy to understand and a score of two signaled the preferred version. However, a score of two could be assigned only to one solution. Table 6.2 shows the final result of our evaluation for each of the proposed solutions.

The *Diamond* solution got the worst score since designers agreed that it is not intuitive and has perception issues. They found it hard to see the beginning and the end of the uncertain area as bigger as the items and their uncertainty get. Moreover, they mentioned that it is difficult to compare the impact of uncertainty on different items with this approach. Another drawback of the *Diamond* and *Butterfly* solutions is having different shapes depending on the amount of uncertainty and the item value. From our experts' point of view, the visual appearance changes a lot and looks very inconsistent. Although they all agreed the *Butterfly* has some advantages over the *Diamond* solution. The *Butterfly* solution has a more precise shape and the shapes are more consistent for comparison – although they found it overall cluttered. It helps to distinguish the individual items on completely filled axes because of the empty white space on both sides. In comparison, the *Border Blocks* approach was appreciated because of the consistent shape, however, the experts mentioned that it is difficult to distinguish the small items and to follow the corresponding flows to these items. With this approach, the participants could easily perceive the extreme differences, but not the weak ones.

The *Filled Blocks* method was preferred over the *Border Blocks* by some participants. Almost half of the participants agreed that using another color code (light gray) makes it very color-full and cluttered. On the other hand, the second group preferred it. In

	a) Diamond	b) Butterfly	c) Border Blocks	d) Filled Blocks	e) Fork
Votes	0	1	6	9	9

Table 6.2: The result of evaluating five nodes visual designs with eight design experts.

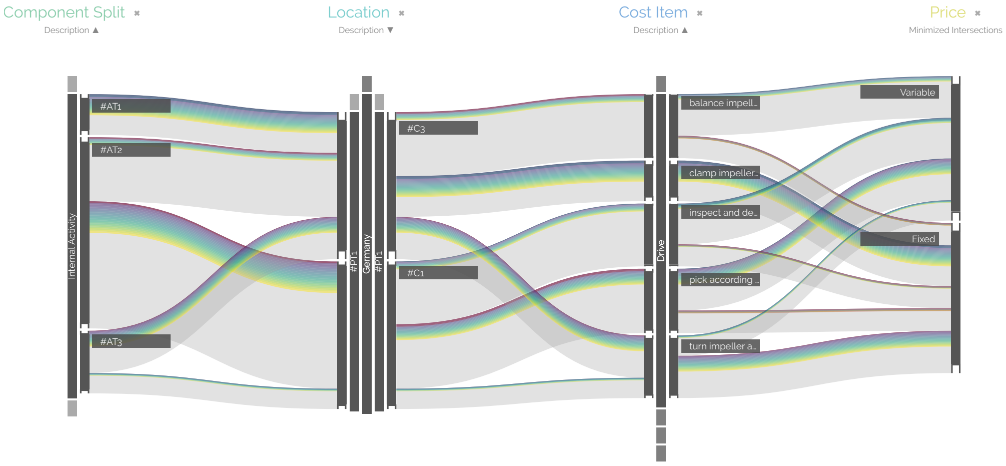


Figure 6.12: Applying one of the selected approaches (Fork) to visualize uncertainty on Parallel Hierarchies nodes.

other solutions the white space is not recognizable because the visualization background is also kept white. Finally, many participants agreed that the *Fork* solution that is a combination of *Butterfly* and *Border Blocks* facilitates the comparison task and is very consistent with regards to the item shapes and makes it easy to follow the corresponding flows. Another reason that this solution got a high score is because it is very easy to separate the neighbouring items, similarly to Butterfly approach. Figure 6.12 shows another realistic example of using the Fork method to convey uncertainty on the nodes of Parallel Hierarchies. We can perceive how the total impact of uncertainty on each category of this visualization can be quickly depicted.

In summary, based on the result of our expert evaluation, we decided to investigate the Fork and Filled Blocks method further for our application. We believe that each visual design has its own advantages and drawbacks depending on the user tasks. To this end, another user study needed to be designed and conducted.

6.7 Summary and Outlook

In this chapter, after describing the motivation, we summarized a case study conducted with ten domain experts to establish uncertainty visualization requirements. Then, we presented three uncertainty representation methods applicable to the ribbons of flow diagrams. The suggested three methods can encode the five confidence levels currently used in product costing applications. The first solution is used to convey uncertainty with the *Color-code* method, which is suitable for small ranges of discrete values. In contrast, our two other solutions (*Gradient* and *Margin*) can also represent continuous values. Our user study showed that the *Color-code* and *Margin* methods perform better and are generally more appreciated by users than the *Gradient* method. Although color coding performed best in identifying items with least and most uncertainty, for more complex tasks, *Margin* was the most accurate solution. Another drawback of *Color-code* is that it only works for a limited number of uncertainty categories. Five categories can be visualized using different colors, but distinguishing more colors leads to a high cognitive load for users.

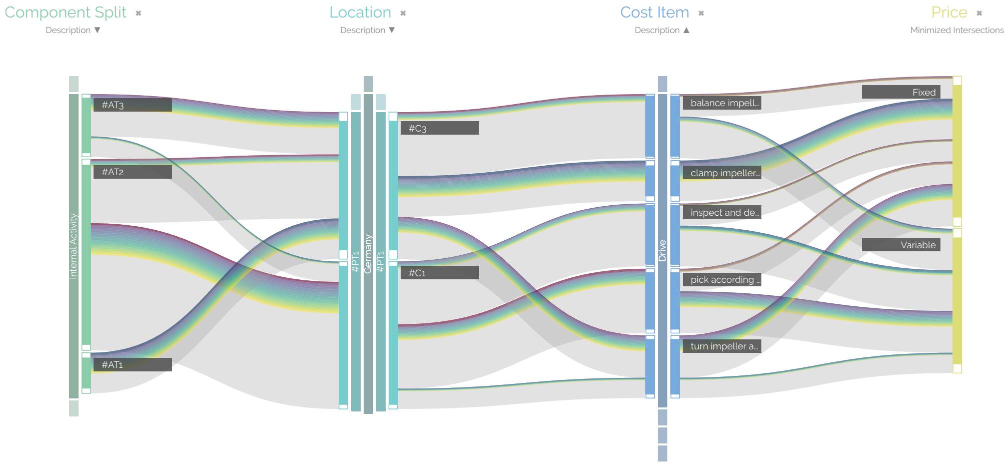
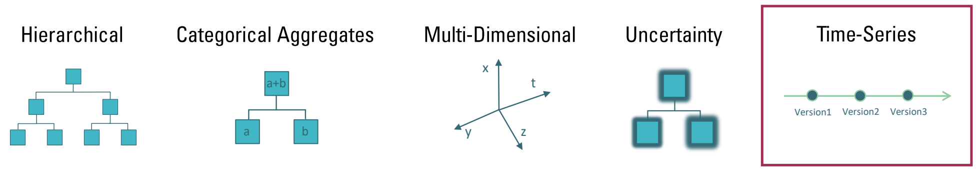


Figure 6.13: Applying one of the selected approaches (Border Blocks) to visualize uncertainty on Parallel Hierarchies nodes and ribbons.

The result of this user study motivated the design of a new approach that benefits from all three proposed methods. This approach is based on the margin method but uses discrete gradients for visualizing the best and worst cases. Furthermore, we applied our revised solution on Sankey diagrams, Parallel Sets, and Parallel Hierarchies. However, the color maps were later changed to the Value-Suppressing Uncertainty Palettes for dealing with this complexity of value comparison in unreliable regions of the data. This is our solution of choice not only for product costing application, but also for other similar datasets. However, the idea can be further expanded by modifying other geometries like item nodes.

Therefore, we suggested 5 different solutions to modify the rectangular nodes of the flow diagrams to represent uncertainty information. In this phase of the research, we conducted an expert evaluation in order to choose the most suitable visual representations to be used in our application. The preferred solutions for the nodes was adding forks or filled blocks. In the future, the preferred designs must be evaluated in a controlled user study by setting up different user tasks.

However, when datasets are larger and more complex, other approaches beyond what was discussed in this section might become necessary. Figure 6.13 shows one example using the final design of Parallel Hierarchies along with uncertainty shown on both ribbons and nodes using Border Blocks. For the simplicity's purpose, we used dark gray rectangular nodes to evaluate our visual design. However, when we assign different colors to the axes, it might impact the way we depict uncertainty values. This can be also further evaluated in our future user studies. In this work, we used two visual features of flow diagrams to convey the uncertainty information but in the future more visual elements can be added to the visualization, such as glyph-based solutions for more complicated datasets.



Chapter 7

Visual Comparison Task

Time is an abstract concept. Yet, the terminology that our users employed for time was based on their concrete experience of creating different versions of product costing calculations (section 4). As discussed before, the focus of this research is on flow diagrams as a powerful visualization technique to understand the structure of costing datasets. In many costing application scenarios, there is a need to quickly understand all facets of the data and compare different versions to make decisions (see Task1 & Task2 from 4.3.2). Therefore, in this chapter we examine how our currently developed visualization solution can be extended to cover the fifth aspect of product costing data – namely time – and be used to solve visual comparison tasks. We present different strategies to visualize changes of hierarchical data using flow diagrams.

In the following, we introduce a taxonomy of four groups covering different comparison tasks on time-series data. Then, we put forth three techniques to solve three of those task groups with flow diagrams. Parts of the research presented in this chapter have previously been published in [Vosough et al., 2018b].

7.1 Introduction

Many real-world datasets carry the temporal dimension which defines when the data has been collected or generated. Our perception of those datasets is often incomplete without considering the temporal context. There is a large body of research on different ways to visualize time-series data. To characterize the dimension of time in the product costing domain, we first reviewed the tasks that users seek to accomplish using visualization methods discussed in section 4.3.2. The main tasks that customers need to perform is the comparison of two or more hierarchical datasets generated over time. Comparison of two individual hierarchies or comparison of multivariate or dynamic graphs [Andrews et al., 2009] are critical tasks in many domains such as biology, software systems, medicine, or social science [Munzner et al., 2003, Holten and Van Wijk, 2008, Vrotsou et al., 2009, Procter et al., 2010]. Because of the emphasis of our research on costing data visualization, the focus throughout this chapter is on the

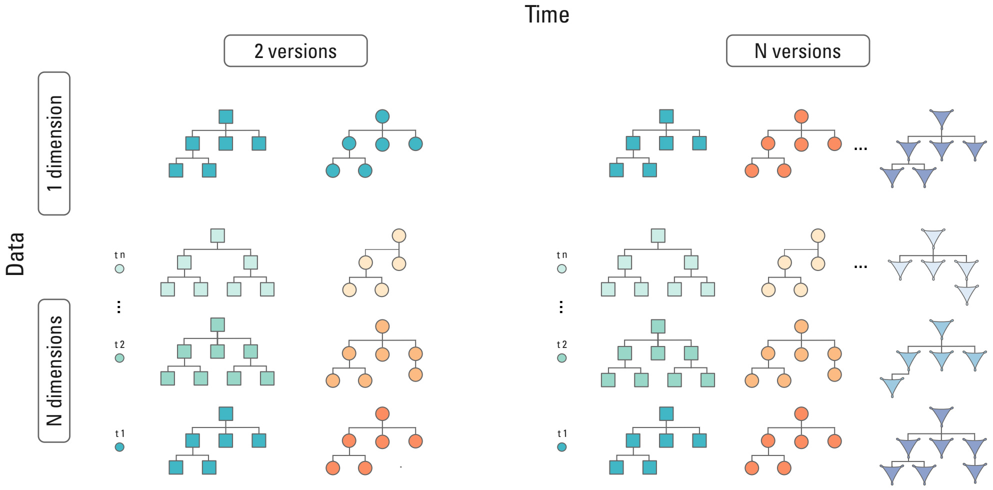


Figure 7.1: A comparison task taxonomy for hierarchical time-series data based on the number of trees compared with number of time variates.

comparison of different versions of hierarchically organized datasets. However, the visual tree comparison techniques presented in this chapter could be applied to other kinds of hierarchical data as well. Moreover, the visualization solutions are meant to provide a global overview of the differences and similarities between two or several hierarchies in a single picture. While information visualization tools are used widely for understanding single hierarchies or for comparison of two structures, the comparison of multiple hierarchies is still a challenging task. However, most existing approaches only provides specific strategies that can be applied to individual problems. Especially, for varying datasets with different sizes and complexities, existing solutions cannot be reused and new general tools for comparison tasks are needed. This highlights the demand for visualizations that provide an interactive graphical access to aspects of data (lifecycle costing) that can hardly be captured in a tabular display, such as costing dependencies, the hierarchical compounding of costs, or version comparison.

Using visualization to assist hierarchical data comparisons has been widely studied before. Different data visualization techniques (shown in Figure 2.6) can be adopted to convey data dimensions for comparison purposes. For example *contrast Treemaps* is designed to effectively compare two Treemap snapshots and highlight the changes in one view [Tu and Shen, 2007]. Similarly, we are aiming to develop visualization solutions that compare two or multiple tree structures over time using flow diagrams. After considering the design aspects explained in section 3.3, and users' tasks in product costing domain, we established a taxonomy summarizing different comparison tasks on hierarchical time-series data. An important factor to categorize multiple tree visualizations is to distinguish whether multiplicity is based on the number of trees, or the number of time variants, or both. Figure 7.1 shows a summary of our categorization and the four basic cases it produces. The four cases are defined depending on the number of hierarchical data dimensions (two dimensional or multi-dimensional data) associated with time primitives, and the number of time primitives to be compared (2 versions or multiple versions). This taxonomy shapes the outline of the rest of this

Structure 1			Structure 2	
Level	Item Name	Cost	Country	Company Code
1	Pump P-100	20.200	-	-
2	Casing	8000	-	-
3	TCD (setup)	826	USA	#CC2
3	TCD (machine)	1888	USA	#CC1
3	Slug for casing	921	Germany	#CC11
2	Pick-pick list	1496	-	-
3	Turn shaft-specification	621	USA	#CC2

Table 7.1: Industrial pump with different dimensions of components.

section, focusing on different visualization design solutions for flow diagrams. In order to illustrate our approach, we use product costing as application domain. Building upon our work described before, this research extends flow diagrams to solve more complex tasks. On the one hand, an effective comparison between different versions needs to be visually presented to the user. On the other hand, different dimensions of the product components need to be considered.

7.2 Comparing Two One-dimensional Time Steps

In this section, we describe our solutions for comparing a tree using two different structures or a tree from two points in time. We will describe two task types from the product costing problem domain, which the visualization concept can be applied to, and then a set of guidelines for future research directions. The solution is designed to address Task1 and Task2 from the list of customer visualization tasks (see section 4.3.2). We propose a mirroring method with the appropriate interaction techniques based on Sankey diagrams.

7.2.1 Problem Statement

As described before, one of the challenges that the customers of product costing project are dealing with, is to find the main cost drivers by comparing multiple cost calculations with each other. Users need to gauge the impact of adding or removing individual items or assemblies on the overall costs. This challenge can be addressed by the following two individual tasks.

Structure Comparison (T1)

The costing data structure is hierarchical and multi-dimensional, since the overall product cost is the sum of the sub-part costs, raw materials, and associated activities. In addition, the total cost can be broken down based on different dimensions such as cost component split, material types, countries of origin, maturity levels, cost centers, plans, or weight. Table 7.1 shows a small part of a selected dataset from an industrial pump and its different dimensions such as component split and country of origin. Complete datasets also provide company codes for each country and cannot be handled appropriately by the current Sankey diagrams. Showing two hierarchical trees with currently available Sankey diagrams (means the cost impacts of each company code or country along with the whole data structure) is not possible but would be extremely helpful for customers in order to leverage the information contained in their data structures.

Version Comparison (T2)
One product calculation consists of several versions. These calculation versions (CV) are used to take different scenarios into account, so that the cost development of a product can be projected into the future and factored into the analysis. Cost of an item changes over time for different reasons, such as impact of learning curves, currency fluctuation, governance laws, commodity price changes, or inflation rates. Table 7.2 shows the same example from Table 7.1 with the first structure, but at two different points in time. It represents how the cost of *casing* and *pick-pick list* in the second level change based on the cost of their sub-items. Moreover, new items might be added or removed from the cost structure.

		Version 1	**Version 2**
Level	Item Name	Cost	
1	Pump P-100	20.200	19.400
2	Casing	8000	7.300
3	TCD (setup)	826	826
3	TCD (machine)	1888	1188
3	Slug for casing	921	921
2	Pick-pick list	1496	1596
3	Mill groove	-	190

Table 7.2: Two versions and associated costs of an industrial pump.

The next section describes our visualization concept for supporting more effective perception and understanding of end-users these two data characteristics.

7.2.2 Visualization Design

Based on the customer feedback reported in section 4.3.4, our work focuses on using flow diagrams to visualize costing data. Our solution for this task type is based on Sankey diagrams that emphasize quantities in a dataset [Riehmann et al., 2005]. The thickness of the links (flows) between the items (rectangular nodes) shows their quantity, which corresponds to the cost of a component in our solution. Hence, on the very right side, singular components are displayed and their cost is subsumed in component groups to the left. Sankey diagrams can be created with multiple levels of connections and facilitate finding items that dominantly contribute to the total product cost. They are particularly suitable for understanding how a data structure is composed and for understanding relationships between elements [Schmidt, 2008]. In order to solve the two main tasks outlined earlier, we considered different comparison solutions that leverage the strength of Sankey diagrams to show many-to-many mappings between two domains or multiple paths through a set of stages. Our solution is designed to keep the parent nodes or the leaf nodes of two costing structures in the middle and then visualize the complete structures on both sides. In the following, we

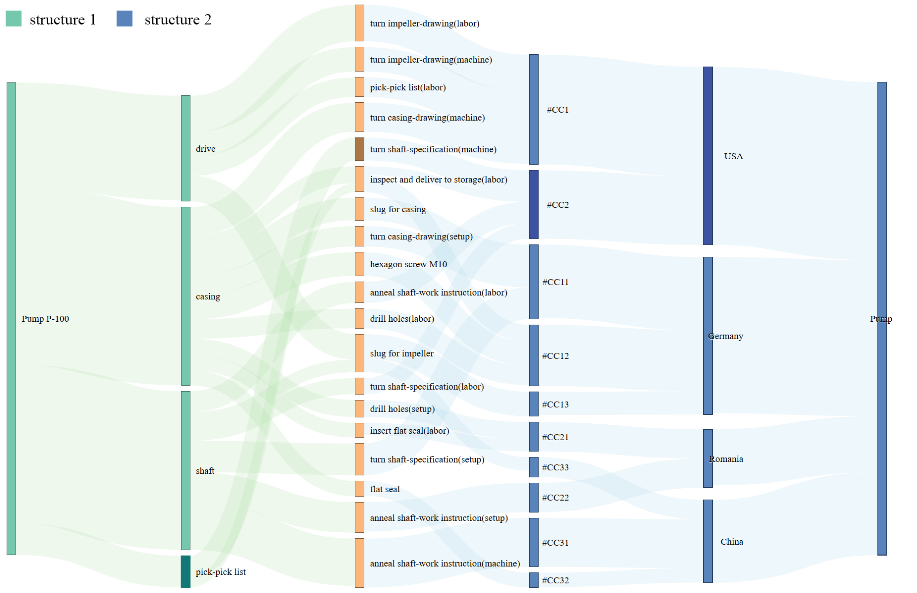


Figure 7.2: Visualizing a costing data structure with two facades with Sankey diagrams. The data is structured based on cost component split (green) and based on location (blue). Parents node of the selected component are shown by highlighted bars.

explain each solution in more details. All visualization prototypes were implemented in JavaScript using jQuery[1] and the visualization library D3[2] and the color of choice for dimensions are from the ColorBrewer website [Harrower and Brewer, 2003].

Structure Comparison Task (T1)

The first task addresses the problem that each cost calculation can be built upon different criteria, which results in different hierarchical data structures. In the typically used Sankey diagrams, two hierarchies cannot be visualized simultaneously, and the main challenge was to find an efficient way to see the relationships between two hierarchical structures.

Finding an appropriate visualization for a single hierarchical graph is not an easy task. However, when comparing two graphs this task becomes even more difficult. One problem is that users need to perceive the relationship both within one graph and between two graphs. Considering the Gleicher's taxonomy of visual design for comparison, three approaches are common: juxtaposition, superposition and explicit encoding. Also, the three designs can be combined to create hybrid solutions that benefit from features of two solutions [Gleicher et al., 2011]. **Juxtaposition** is a simple solution that puts different objects next to each other. It is a simple approach but not always efficient as it requires more space and relies on the user's memory to build the connection between objects. **Superposition** works by overlaying objects on top of each other, and **explicit encoding** computes and directly shows relationships between objects.

[1] https://jquery.com/, retrieved on 10.07.2019
[2] https://d3js.org/, retrieved on 10.07.2019

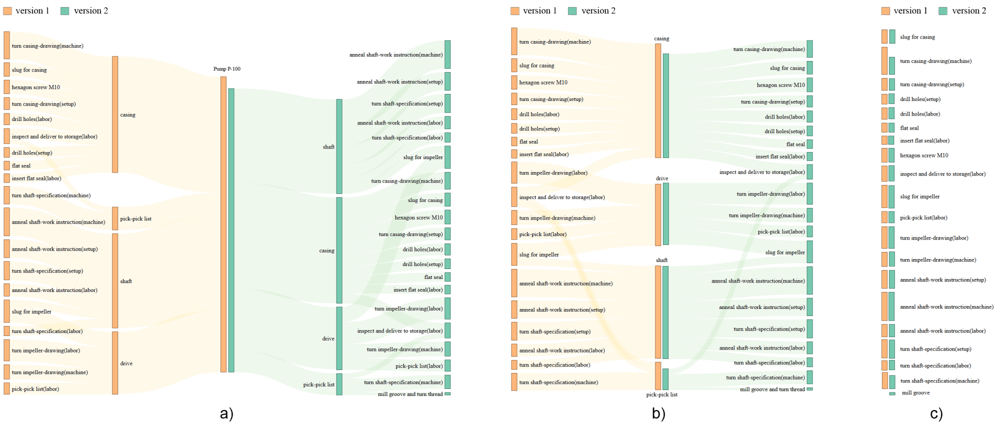


Figure 7.3: Visualizing two cost calculation versions with Sankey diagrams: ”version 1” is shown in orange, and ”version 2” in green. It shows a comparison between the items on, a) first level, b) second level, and c) third level, close to each other along with their sub-items.

Superposition is not a practical solution for Sankey diagrams, since the data structures are totally different for each scenario and overlaying prevents users from seeing relationship both within and between two graphs. A solution to avoid visual cluttering is to show only differences between two superimposed diagrams. However, showing the hierarchy is important for costing data and with this approach information is lost. Moreover, the leaf nodes on the costing graphs stay consistent and only the items in the middle levels and subsequently the parent node change.

We designed a Sankey diagram that keeps the leaf nodes in the middle and placed two structures on both sides (mirroring around leaf items). With this solution, both the tree structures and the relationships between different items are preserved. However, we save space by visualizing the leaf nodes that are the dominant items in the costing data structure only once. In addition, the problem of information overload for users is mitigated, since the items have connections and recognizing relationships between them becomes easier. This is further facilitated by adding supportive interaction methods. Figure 7.2 shows one example of an industrial pump that has two dimensions, cost component split and location. The leaf nodes are represented in the middle with a different color (orange) in order to better distinguish between component split structures on the left (green) and location on the right (blue).

Interaction techniques are a pivotal tool to enhance visual comparison. One common solution in the domain of information visualization is visual filtering using highlighting, described by Becker et al. as brushing with a special color to paint an object [Becker and Cleveland, 1987]. In order to assist the comparison task, we use common brushing interaction to establish connections between related components. By hovering an item with the mouse, the corresponding parents are highlighted and it is easily observable to which components this item belongs and where it originates. Furthermore, the relationships between the middle levels can be inspected by hovering with the mouse. For instance, when hovering over the casing item (second level left), the companies and countries that this component mainly originates from are highlighted. These interaction techniques play an important role in enhancing the visual understanding of hidden relationships in costing data.

Version Comparison Task (T2)

As described in section 7.2.1, the second task is about analyzing the changes in structure within the costing data over time. This comparison task has different data characteristics. In this task, two changes can occur, the data structure stays consistent and the item costs change slightly or few items can be added or removed in the data structures. In section 3.3 we reviewed the hierarchical taxonomy proposed by Beck et al. (see Figure 3.12). They distinguish between animation and timeline for dynamic graph visualization techniques [Beck et al., 2014]. Within the timeline category, juxtaposed, superimposed and integrated approaches can be considered for node-link structures. In the integrated approaches, the graphs are interlinked and cannot be separated, which is not a suitable solution for our problem. Another possible solution is switching between two versions of the same diagram. A fast prototype was implemented to switch in a certain interval between two images of Sankey diagrams. The arrangement of the items proved to be a problem in this approach. Sankey diagrams use different techniques to arrange the items. For instance, in our solution items are arranged based on their size (cost), but this can change in different versions as the item values change. Although research shows that timeline approaches provide better analysis instead of animations [Tversky et al., 2002].

Our proposed solution was designed by placing two data structures (versions) next to each other, since juxtaposition can be a suitable solution for comparing only two versions. In contrast to the structure comparison task, the parent item is placed in the middle and the sub-items of both structures are arranged around them (mirroring around parent items). The rationale behind this design is that users usually want to identify the differences between the total costs at the first glance. By putting the parent nodes (bars) close to each other, this comparison becomes easier (see Figure 7.3).
As the values of other items cannot be easily compared, especially when the differences are small, applying proper data abstraction is necessary. There are different solutions available to visualize comparison of complex data that decrease the complexity by abstracting data [Amenta and Klingner, 2002]. In this work, for a more accurate comparison, the levels of interest can be clicked and thus placed next to each other. The resulting image is similar to a simple bar chart containing two bars, which is easier and faster to interpret. By double clicking on the levels in the middle, the graph is unfolded and switches to the initial view. Although this strategy follows the juxtaposition strategy, due to the interaction with the different views, it still conveys the feeling that the images are overlaid [Roberts, 2004].

Figure 7.3 shows an example of two versions of a cost calculation for an industrial pump from different points in time. The first version represents the items on the left side in orange and the second version on the right side in green. In the first picture (a) all items are presented and it can be seen immediately that the total cost decreases over time. By hovering each item with the mouse, additional information such as the exact price and relative price are shown in a tooltip. By clicking on any item on the second level, the graph is folded and items on the second level are moved next to each other (b). Subsequently, by clicking on any item from the third level, those items

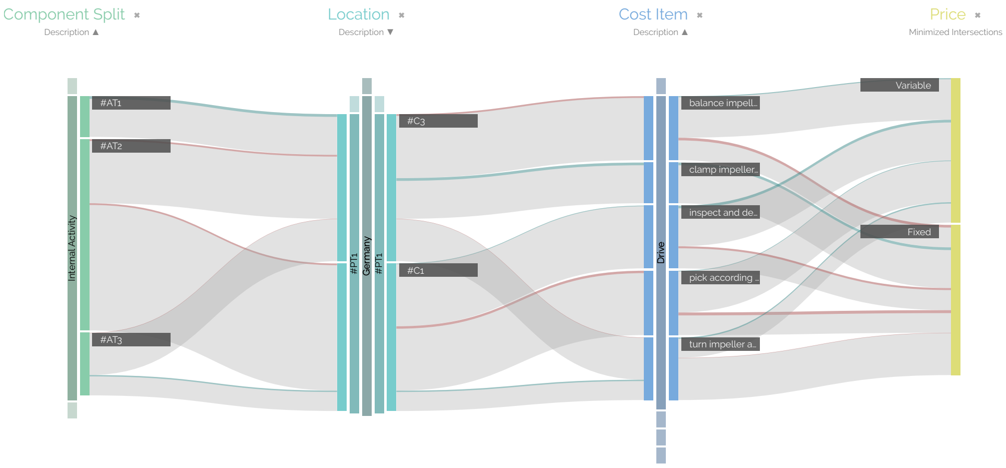


Figure 7.4: Comparing two n-dimensional time steps with Parallel Hierarchies.

are moved next to each other (c). By double clicking anywhere on the screen, the visualization switches back to the first view and shows all items again (a).

Both solutions described above are designed and tested for rather small datasets. However, they can be extended for bigger structures with more items on each level by applying zooming or details-on-demand techniques similar to the designed solutions for Parallel Hierarchies (see section 5). The proposed solutions were evaluated during informal sessions with the customers of the project. Beside feedback on scalability issue, participants mainly asked for a solution to compare not only two but several dimensions of the data. The gathered feedback led us to design the solution described in the following.

7.3 Comparing Two N-dimensional Time Steps

The solutions we proposed for comparing two one-dimensional data structures, were designed and evaluated before introducing Parallel Hierarchies into the product costing field. Our first prototype was based on Sankey diagrams as a perfect solution to show one dimensional hierarchical data. Parallel Hierarchies were introduced later to extend the first prototype and overcome the problem of representing multiple hierarchical aggregates. Consequently, a new solution for the visual comparison task described in section 7.2.1 was required to cover multiple dimensions of the data instead of only one. To that end, we considered an extended version of Parallel Hierarchies applying superimposing techniques for comparing two calculated versions. The structures of two version are stacked on top of each other with different color codes. Figure 7.4 shows an example of comparing two version of the same industrial pump dataset used before. One can immediately see where the values (costs in product costing example) have increased (indicated by red) or decreased (indicated by green). Not only the changes in the item values can be simply tracked with this solution, but also missing or new data items in the structure can be shown in the similar way. For that reason, we assign red to the newly added data items as they are rising the cost, and green to the missing ones in the data structure as they cause cost reduction. This solution was shown and evaluated with the customers of SAP product costing project.

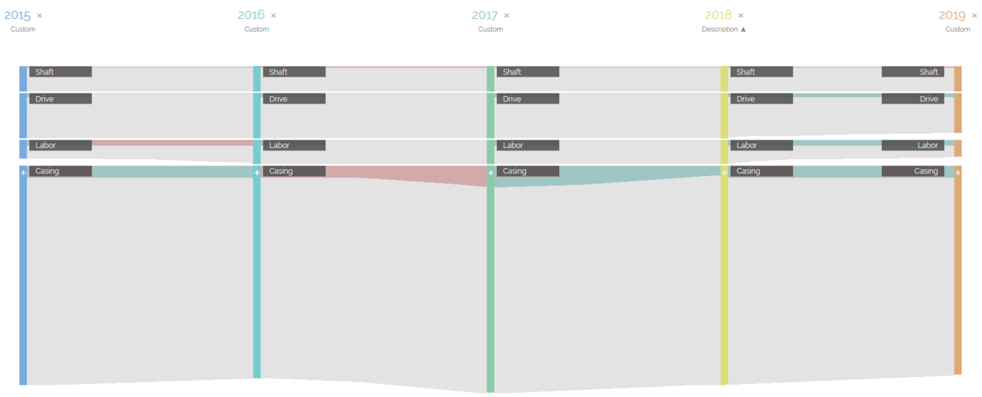


Figure 7.5: Comparing N one-dimensional time steps with Parallel Hierarchies.

Almost everybody agreed on the effectiveness of this approach for getting a high-level overview of the changes between two time steps. However, the customers also mentioned that one of the main limitations of this approach is that it cannot be used for the comparison of several time steps or observing overall cost changes. Nevertheless, based on the insights obtained from users, we decided to integrate this approach into the final solution. The amount of changes over time in costing data structures is typically modest. However, when the data structure changes massively over time, the superposition approach is not suitable, since the visualization becomes extremely cluttered and unintelligible. We used an approach similar to the *Filled Blocks* method designed for uncertainty (see Figure 6.10), to represent the value changes on the nodes of Parallel Hierarchies. This approach can be very practical when used along with the ribbons or alone when the ribbons are used to depict uncertainty values.
In the following, we will describe how the same approach can be used for a different task type.

7.4 Comparing Several One-dimensional Time Steps

The third and last taxonomy from our comparison classification is the comparison of multiple trees over time. Novel solutions to address this challenge have been developed, such as the visualization method developed by Holton et al. for comparing hierarchically organized data which is the most similar approach to our solution [Holten and Van Wijk, 2008].

One common request from SAP customers was to solve this comparison task using Parallel Hierarchies. The idea was to use vertical axes to represent the hierarchical structures at different time steps. In product costing scenarios, the user compares the hierarchical structure of the bill of materials over time. The users' objective is to determine which items' costs have changed over time, then explore their sub-items, and find the main drivers for those changes. To this end, the similar approach to what was shown in section 7.3 was examined for this taxonomy. However, the proposed solution in section 7.3 is used to compare two hierarchies (superimposed approach) and the solution discussed in this section is designed to aid comparison of more than two hierarchies by juxtaposition. Although this approach is able to compare multiple time steps, only one dimension of the data (Cost Item from Figure 7.4) can be shown at a time. Figure 7.5 shows an example of comparing the first/leaf level of the pump dataset over five years. One can immediately see how the “Casing” item’s cost has changed over the past five years. On-demand drill down let’s one find items with most impact on cost variation. Labor has also faced extreme cost changes from 2015 to 2016 and 2018 to 2019. Such general overviews can be captured with this solution. However, there might be other simpler ways to represent these data and flow diagrams are not necessarily the best approach to visualize our data structures. This approach was developed based on user requests so as to provide them with the possibility to quickly switch from the exploration task to the comparison task. The particular argument for this request was to avoid using multiple views with different visualization types that requires time to understand. This solution has not yet been formally evaluated with end users, but it was considered as a simple and understandable solution based on informal feedback gathered from our users.

Our proposed technique worked well in the costing domain, in which the structures of the hierarchies are very similar but not necessarily identical at different points of time. However, the solutions need to be applied and evaluated with alternative datasets, where structure of the hierarchies can differ considerably. Moreover, as an alternative to the horizontal layout, a vertical layout could be used instead. In particular, when there is a sizable number of nodes to be shown on each axis. From the perspective of visual perception, it might be easier for users to detect changes between a pair of axes that are positioned in this way.

7.5 Summary and Outlook

In this section, we presented a taxonomy to classify visual comparison tasks in the context of product costing shown in Figure 7.6. Our classification is defined on time-series of hierarchically structured data with one or several dimensions. We proposed visual solutions applicable to flow diagrams for three out of four categories of the taxonomy. For the first category, we proposed two Sankey-based visualizations along with an interaction concept to facilitate two specific comparison tasks. Both visualizations are based on a juxtaposition approach and connect two Sankey diagrams by putting either the parent or leaf nodes in the center of the visualization. The solutions are designed to compare two complex graphs, but this work can be extended to multiple graphs, since both scenarios typically contain more than two dimensions. One approach would be connecting many Sankey diagrams on the horizontal axis and pan left and right to see the different versions. Another improvement idea to enhance the visual perception is to add more visual components to make the comparison tasks easier. Another interactive feature, which would be highly beneficial, is the possibility to visualize different quantities at the same time for better comparison and easier decision making.

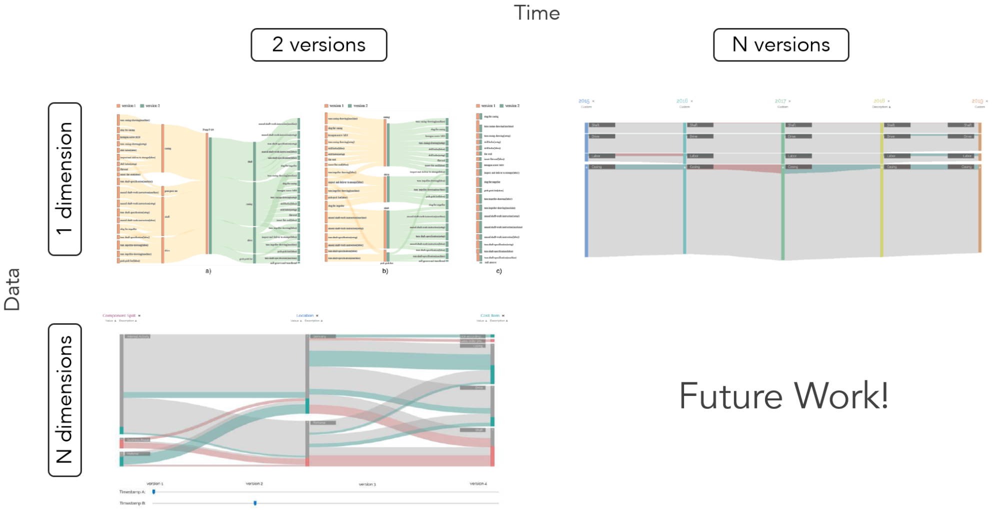


Figure 7.6: Summary of the developed visual solutions for the comparison task taxonomy shown in Figure 7.1.

For the second and third categories we used our recent visualization method – *Parallel Hierarchies* to compare either several dimensions or one dimension over time. We used color codes to indicate the changes in the hierarchical data structures. When item costs increase, this can be indicated by a red color and green when it decreases. This visual feature can be applied both on the ribbons or bars of Parallel Hierarchies. We are planning to evaluate the last described visualization approach with our customers during one of the upcoming customer workshops. Although the concepts and design decisions have been made based on customer interviews and workshops, the presented techniques have not yet been evaluated in a formal setting. Since the presented solutions are still work-in-progress, different variations could be compared for their effectiveness in a user study.

There are some limitations with regard to scalability when the number of nodes becomes exceedingly high. In this case, the zooming and details-on-demand techniques already developed and discussed in chapter 5 could be leveraged. However, additional interaction techniques or views could be added to provide a user with a convenient way to obtain more information.

Finally, the most complicated category in our taxonomy is to compare multiple hierarchical dimensions over multiple time steps. This is currently not covered and remains a topic for future work. Finding a solution to cover this case requires more sophisticated interaction techniques or multiple views.

Chapter 8

Parallel Hierarchies in Practice

In section 5.4, after introducing Parallel Hierarchies into the product costing domain, we showcased the extensibility of the visualization solution with two use cases, looking at US demographic and yeast genomic data. In this section, we present applications of Parallel Hierarchies on industrial datasets from different SAP customers. We represent two more projects that were triggered by SAP customers. The first project aims at visualizing the results of a machine learning based plausibility check process. The second project covers two scenarios from customers of a general visual analytics framework for business data. All use-cases and projects discussed in this section have been defined in close collaboration with SAP customers. Parts of the research presented in this chapter have previously been published in [Vosough, 2018, Vosough and Vasyutynskyy, 2018].

8.1 Application to Plausibility Check Task

A prerequisite for any subsequent analysis of business data is its quality. One obvious solution to this problem is manual maintenance and validation. However, this approach is unrealistic because it is too costly and too time consuming given the scale of the data. It is therefore of prime importance to develop automated means for quick assessment and validation. The recent development and stunning successes of machine learning offer sophisticated data processing algorithms, which can potentially help validate complex business datasets at scale [Bose and Mahapatra, 2001]. However, as we will show, these analysis results are often themselves large, complex, multi-dimensional and thus require novel means of analysis and interpretation. It follows that novel interactive visualization tools are needed to understand the output of machine learning algorithms applied to big data validation.

One enterprise application, where data validity and reliability is extremely important is product costing. Product costing applications can help a wide range of stakeholders (controllers, engineers, purchasers, etc.) in reducing the whole product's life cost. Importantly, product costing application users must assess not only the information presented to them, but also the confidence they have in that information. For example, there is often inaccurate or missing information in costing calculations because users make mistakes while entering data or lack information. Product costing data validation – termed *plausibility checking* – is a critical issue since mistakes can have a dramatic impact on cost estimates and thus on business decisions. Plausibility checking is typically performed manually on projects containing hundreds of product cost estimates with up to millions of cost items - a tedious if not at times impossible task incurring high costs. Solutions leverage Machine Learning (ML) and data mining algorithms, which are in general a very effective approach to validate quality and automate plausibility checking for costing data [Witten et al., 2016]. However, due to the scale of the data, plausibility checking will often return a large number of potential errors, which makes their analysis and exploration challenging. What is needed is a visualization tool that would assist interactive exploration by providing (i) an overview over the different types of problems, (ii) a localization of problem areas, (iii) a dive-in view to explore details.

8.1.1 Plausibility Check Process

This section first describes the topic and research context and then explains the product costing's *plausibility check* problem.

Context
During the early phases of product costing, item prices are unknown or undefined for some time. As more data becomes available, the cost estimates are refined into new versions that eventually converge to a stable state. In this stable state, most parts of the product's cost structure can be delivered with a precise price fit to the desired costing goal. During this process, many decisions are made by experts such as controllers and engineers towards an estimation of the cost structure. This complex task and multiplication of individual contributors leads to data entry mistakes or wrong estimations. This makes the estimation of costs across the lifecycle of products extremely challenging to achieve. Errors are often uncovered too late and have a dramatic impact on the quality of cost estimates. In the following, we will describe the plausibility check process that can help to detect common errors.

Plausibility Checks in Product Costing
Plausibility checks are intended to help users find potential errors by assessing the validity and plausibility of manual entries. The goal is to automate the detection of such errors to the largest extend possible. On the one hand, some trivial errors can be detected by simple hard-coded rules. For example, if a price for a material has not been set at all, a simple rule can detect cases where null values are present. On the other hand, there are plenty of less evident errors, the detection of which would require deep knowledge about the typical values and structures of the products. To detect such errors, we introduce the notion of *plausibility checks* that aim to detect potential errors such as anomalies, e.g. substantial deviations from typical values and structures.

Each type of plausibility checks is realized as a separate function producing its own validation messages. Further, the plausibility checks functions have the same input and output interfaces. The input interface includes the following parameters:

- List of calculations to be checked.
- List of calculations on which the models should be trained.
- List of settings for check functions, like thresholds or model settings.

The output of check functions is a list of validation messages, the structure of which is explained in the next section. Thus, such checks can be flexibly combined to fit to the company specifics, independent of the used methods and underlying models.
while very different models and methods can be used for plausibility checks, in general they consist of the following phases:

- Training models: In this phase, we analyze historical data for similar products and automatically extract the typical values and (sub-) structures for different aspects of the costing structures, as well as their typical variations.
- Detecting anomalies: During this phase, potential errors are detected as anomalies and their impact is calculated.

The product calculations have hierarchical structures, which consist of items and modules of a product with properties such as prices, quantity, process duration, maturity, etc. Accordingly, the plausibility checks use different models that evaluate different aspects of those structures. Depending on the models used, the plausibility checks can be classified into the following groups:

- Scalar value checks: These check whether scalar values like prices or duration of manufacturing processes significantly deviate from typical values.
- Structure checks: These check the current structures deviate from the typical pattern structures of similar products. For example, one detects if some item is missing in the sub-module of the product, whereas it is always present in similar submodules from the training dataset.
- Cost share checks: Due to different product designs, the shares of different types of costs like ratio between material costs and processing costs may get unacceptable. This will indicate the general structural problems within the product.

Plausibility Check Results
Model training and anomaly detection in plausibility checks can be realized by different data analysis and machine learning methods, from basic statistical approaches, to classical machine learning methods all the way to deep learning. When using statistical approaches for scalar value checks, the statistical indicators like average, median and standard deviation are at first calculated from the training data. Anomalies are then detected using the variance test, which indicates value deviation by more than 1.5 times the standard deviation. Statistical approaches work well in case of small to medium training datasets. They allow for a quick training and assess detecting plenty of anomalies which would be otherwise very hard to identify manually. If more complex dependencies are available in the data, more complex approaches are needed. For example, to model the dependency between subparts processing time and final product parameters we have used Support Vector Machines – which work well in case of non-linear dependencies. For classification and prediction of the substructures, recurrent neural networks have been used instead.

We have trained a set of plausibility check models on the datasets from 4 customers representing different industries such as original equipment manufacturers (OEMs), machine tool producers, and automotive suppliers. Notably, the datasets have great structural variety, having from a few up to several hundred calculations or from 10 up to 50000 items per calculation. The results of training and plausibility checks have been validated together with customers. An example of the resulting list of plausibility check messages is shown in Table 8.1.

	Item	Msg Typ	Message Text	Cost Impc
1	Slug for casing	10	Price (variable portion) of 20.0 €differs from usual one of 10.0 (variance of 0.6)	10.0 €
2	Slug for casing	11	Price (fixed portion) of 1.0 €differs from usual one of 0.0 (variance of 0.0)	1.0 €
3	Pick according to pick list	15	Duration of 30.0 Min differs from usual one of 16.0 (variance of 4.82)	528.0 €
4	Inspect and deliver to storage	16	Duration of 10.0 Min under item '100-300 Shaft' differs from usual one of 5.00 (variance of 0.0)	5.40 €
5	No Item	22	The item 'Pick according to pick list' is present 3 times, usual is 2.0 times (variance of 0.0)	765.0 €
6	No Item	26	The item 'Clamp impeller (setup)' is missing under assembly '100-200 Drive' (normal probability of 1.00)	7.20 €
7	Flat seal	51	Maturity: Item was last modified 170.68 days before last calculation version update and may not be up-to-date	110.0 €
8	Calculation Version	90	Cost component 'Materials (AG 110)' has share of 13.63% which differs from usual one of 14.86% (variance of 0.73%)	127.0 €
9	Calculation Version	90	Cost component 'Activities (AG 120)' has share of 63.09% which differs from usual one of 61.93% (variance of 0.67%)	119.6 €
10	Calculation Version	91	Calculation version has total cost of 9649.67 €which differs from usual one of 8154.61 €(variance of 128.41 €)	149.1 €

Table 8.1: Plausibility check result with different types of messages.

The resulting messages contain the following fields:

- **Item**: indicates which item of the costing structure the message refers to. Some messages refer to a specific item, while others refer to the whole calculation version.
- **Message Type** (Msg Typ): each plausibility check method can produce at least one message type. The unique message types allow for a quick overview and filtering of messages.
- **Message Text**: contains the detailed description of identified issues, including problematic field, and current values. Further, it contains typical values and variations for this kind of item, which allows to follow why the plausibility check message was triggered. Then, the user can see how far the current value is from the expected one and thus justify it, so that the system can learn from his feedback.
- **Cost Impact** (Cost Impc): presents which sum of the total cost may be potentially affected by the issue. This allows the user to prioritize the issues and address the most critical ones first.

All these details help the user to follow the causes of the identified issues and to make the decisions on how to correct them. For example, in Table 8.1 the item 'Pick according to pick list' has the most critical cost impact with sums of 528.0 EUR and 765.0 EUR, and thus should be considered first. Further, line 3 gives a hint that the processing duration of 30 minutes is unusually long and may be corrected towards 16 minutes. In line 5, this processing step is present 3 times which may be 1 too many times. The impact on the cost shares and total costs indicated in lines 8, 9 and 10 are subsequent deviations caused by anomalies on the item levels, giving a hint that the share of materials in the whole calculation is too small due to wrong entries on processing steps. Above all, the plausibility checks only give hints on the deviations from the typical values and the messages do not necessarily indicate errors. In some cases, such deviations are intended for new products and the user can accept them. Furthermore, some kinds of checks may require adaptation to company specifics. Depending on the calculation size, quality of the data, and the used plausibility checks, the number of validation messages can vary from 0 to several thousands. Presenting a plain flat list of validation messages would overwhelm the user. First, because of the sheer number of messages, and second, because of the large number of dimensions (20) associated with each message. To help the user to make sense of this deluge of data, a novel interactive visualization tool is required.

8.1.2 Visual Exploration of Machine Learning Results

In chapter 2.1.4, several visualization solutions to analyzing large multi-dimensional datasets were shown. Based on Keim's classification for multi-dimensional visualization techniques, geometric techniques emerge as a common solution for representing a large number of dimensions but with few data items per dimension. Flow diagrams such as Parallel Sets or Parallel Hierarchies are obvious choices in that category for visualizing multi-dimensional categorical data. Considering the task taxonomy in Alsallakh et al. [Alsallakh et al., 2014], we selected Parallel Sets and Parallel Hierarchies to represents the results of our ML-based product costing plausibility checks since they both can support exploration of the relationship between multiple dimensions.

Applying Parallel Sets

In the following, we look at two realistic datasets from the manufacturing machine and automotive industry. Figure 8.1 represents the validation messages found in one of the SAP customers datasets. We applied our ML-based product costing plausibility check algorithms to this dataset and found 1911 potential errors.

Beside the validation errors, the plausibility result contains 11 individual data properties. In Figure 8.1, four of the categorical dimensions are shown and the number of validation messages is represented by the thickness of the ribbons. Each vertical axis represents an individual data property. The first dimension is the *Calculation_Version_Id* shown by the first axis (blue). Typical costing projects have several calculation versions created over the course of the project's lifetime. The second dimension is the *Validation Message* (Validation_MSG) shown in pink. The validation message represents different message types returned by our plausibility check algorithm. The third dimension is the *Cost_Module_Description* shown in green. This

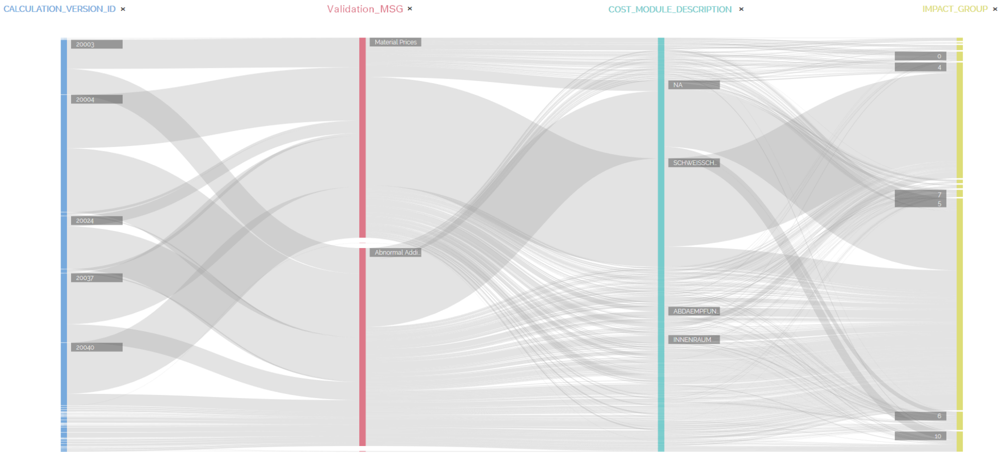


Figure 8.1: Visualizing the result of our machine learning method on product costing structure with 92 items with Parallel Sets.

dimension indicates which product modules these validation messages belong to. Furthermore, different error messages have different impacts on the product's total cost. Those impacts are categorized in 10 different categories and show on the *Impact_Group* dimension (yellow).

To follow the validation results, a possibility of deep dive into the values is provided. The users can get the exact and absolute values of an item using mouse-over tooltips. In case of material price checks, the users can see the problematic actual value, as well as the average value and the standard deviation for training data in a candle diagram [Morris, 2006]. This allows to understand how far is detected value from typical ones. Further, this suggests possible remedy, e.g. to set the new value within the interval of typical values. In this way, the important task of cost optimization [Walter et al., 2018] can be implemented.

The second example shown in Figure 8.2 shows the 261 error messages found in another customer's dataset. The data has 13 dimensions, and among those we show three in Figure 8.2. The thickness of ribbons represents the cost impact in this example. Different validation messages (pink) are shown along with different calculations (blue) and corresponding calculation versions (green). In this example, we see immediately that the cost impact of the selected calculation – shown in the picture – is mainly caused by two validation messages. The first message is on "Material Prices", indicating that this item's price differs from usual one. The second message refers to the "Abnormal Addition Item" validation message, which happens when an item is not expected to present in such calculation.

The first drop down on top of the screen (left) is used to add new dimensions, the second one (middle) to change the data quantity and the third one (right) to change the datasets. Axes can be manually rearranged by dragging or removed by the small cross symbol placed on the right side of their names that appears in red color after hovering the mouse over a dimension's area. The colors of the axes follow Paul Tol's categorical color scheme Palette II [Tol, 2012]. The figures give an overview of which

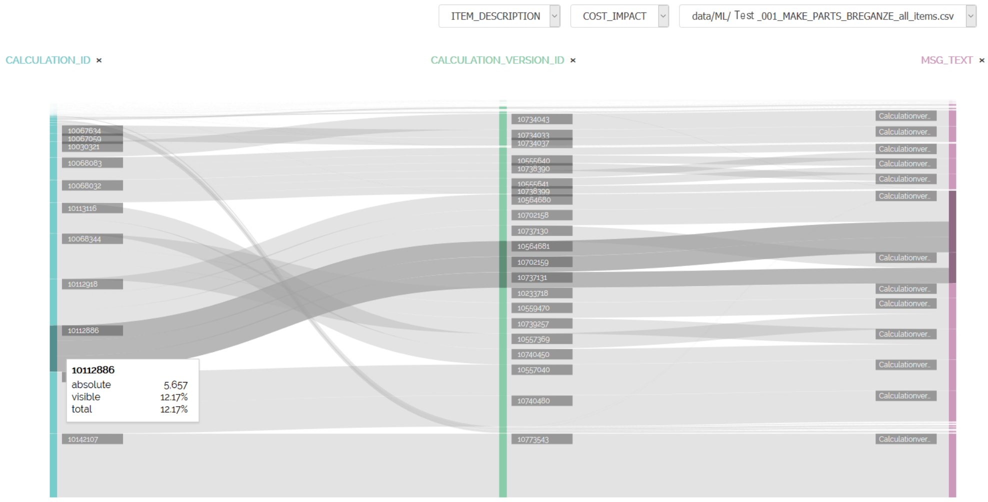


Figure 8.2: Parallel Sets visualization of the result of our machine learning method on product costing structure .

validation messages have been found for each module. They help to quickly detect the problematic areas and assign blame to the responsible contributor. Also, when selecting an item in one dimension, all connected items and relevant ribbons are highlighted.

Applying Parallel Hierarchies

The list of validation messages not only contains multiple dimensions, but also hierarchical categories per dimension. For instance, each calculation project consists of several calculations, and each calculation has several calculation versions. Moreover, the work centers can be organized by different continents, countries, cities and companies. Therefore, a visualization solution capable of handling hierarchical decompositions can be more appropriate. To interactively decompose such aggregates along various interlinked hierarchical categories, we opted to represent the validation errors with *Parallel Hierarchies*. This gives us the ability to gain much more nuanced insights compared to Parallel Sets that look at the aggregate as a whole.

Figure 8.3 shows one example of looking at six hierarchical attributes for validation errors: Work Centers (indicating the location in which the item is produced), Staff (organization and user that has worked on that item's entry), Material (the type of material the item has), Modules (functional module of the item), Validation (the project, calculation, and calculation version that item belongs to), and Message (different message types returned by the plausibility check algorithm). The dataset is from one of our customers and 189 error messages are found by our plausibility check method. Moreover, the dataset shown is in anonymized form. A drop down dialog is designed to quickly switch between different aggregate values. In this example, the height of the categories and the width of the ribbons represents the cost impact, which shows the sum of the total cost that might be potentially affected by all error messages in each category.

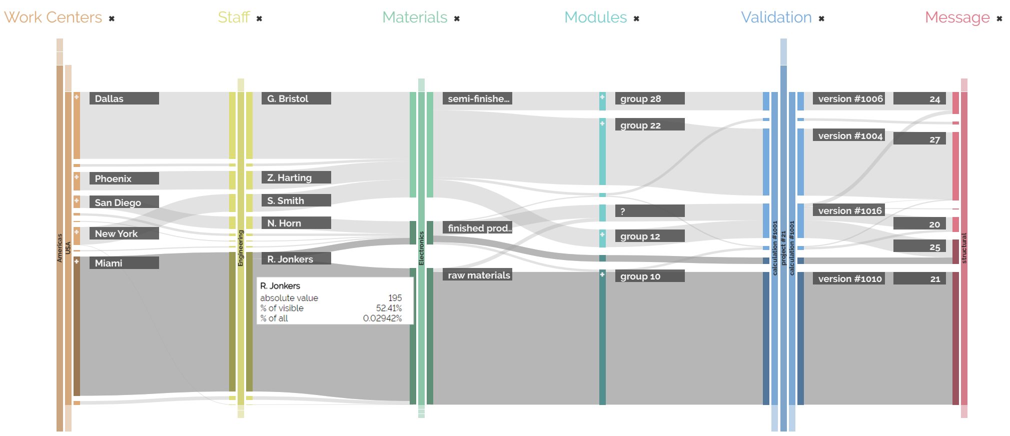


Figure 8.3: Parallel Hierarchies visualization of the list of validation messages found by the plausibility check machine learning method. The highlight emphasizes the big impact of errors produced by user “R.Jonkers” in the current filter.

As illustrated in Figure 8.3, we drill-down to the structural errors produced in different states of America, that are produced by engineers, from electronic materials, and belong to the project 21, calculation 1001. We immediately see the big impact of error messages produced by user “R.Jonkers” in the current filter. Upon hovering over this item, the errors caused by this user are highlighted and we learn from the tool-tip the absolute and relative impact of the error messages produced by this user. From the spread of the highlighted ribbons, we also see that these errors are mainly produced in Miami, in raw materials, from module 10 and calculation version 1010, which mainly belong to message type 21. When changing the quantity value from the cost impact to the number of error messages (not depicted in the figure) one can further see that the number of messages produced by this user is equal or smaller than other users given the same filtering. However, while the number of errors is comparable, their impact on the total cost is very different. All these details help the user to prioritize issues and address the most critical ones at first. As a result, for example, ‘R.Jonkers” needs better training or be replaced by someone more capable, as he is apparently working at a crucial point in the organization where small mistakes can have a big impact.

The interactivity of our solution is essential to facilitate exploration of the data. Users can navigate the hierarchies by simply drilling-down and rolling-up to the desired levels of details for effective data exploration. Moreover, the column chart shown in Figure 8.4 indicates how much of the overall dataset is currently visible in Figure 8.3. The gray column shows the overall indication, whereas the colored columns show how much of the corresponding axes are visible.

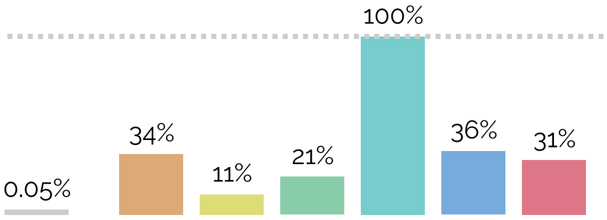
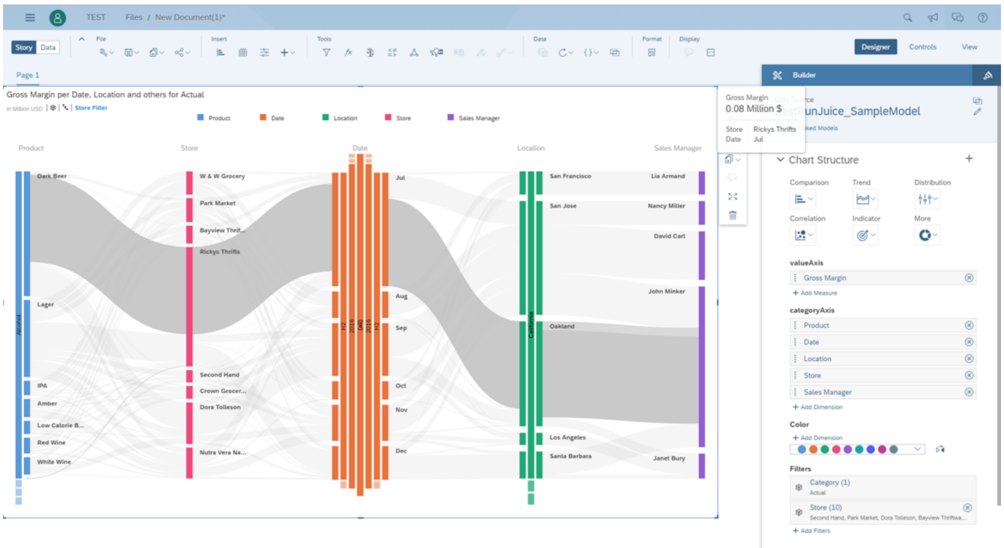


Figure 8.4: Column chart indication how much of the overall dataset shown in Figure 8.3 is currently visible.



Gross Margin per Date, Location and others for Actual

Gross Margin
0.08 Million $
Store Rickys Thrifts
Date Jul

BestRunJuice_SampleModel

valueAxis
Gross Margin

categoryAxis
Product
Date
Location
Store
Sales Manager

Filters
Category (1)
Actual
Store (10)
Second Hand, Park Market, Dora Tolleson, Bayview Thriftwa...

Figure 8.5: Integrated version of Parallel Hierarchies into SAP Analytics Cloud.

8.2 Integration into SAP Analytics Cloud

In light of the project's success and because of SAP customers' frequent requests, SAP was interested in the integration of Parallel Hierarchies solution into other SAP products such as SAP Digital Board Room and SAP Analytics Cloud. In this section, we describe briefly two applications of the project along with the use cases we gathered from users.

8.2.1 SAP Analytics Cloud

SAP Analytics Cloud combines business intelligence, planning and real-time predictive analytics into one cloud solution. One can simply generate focused reports and collaborative tools for online discussion. The product is used for story-boarding, planning and data discovery, and targets all analysts, from the lowest level to senior management. The solution provides access to all possible data so that the user can visualize, plan and predict using real-time data. All these features make this product a great platform for our visualization solution.

Figure 8.5 shows the *Best Run Juice Company* example. The thickness of ribbons represents the "Fixed Price". In this example the goal is to analyze, report, and even make financial planning decisions on which are the best juices to sell in the US. Best Run Juice sells beverages at retail locations across different states in the US. With our visualization they can simply track their success and find areas for improvement. One can simply see if the company has exceeded its target for revenue and gross margin. By drilling down to see the next level of detail we can see which cities in which states are under or over performing. We can also see which stores contribute to those observations, and even the contribution of each manager to these successes or failures. Furthermore, the relationships between hierarchies becomes explicit and we see immediately how successful California has performed in this example and analyze the

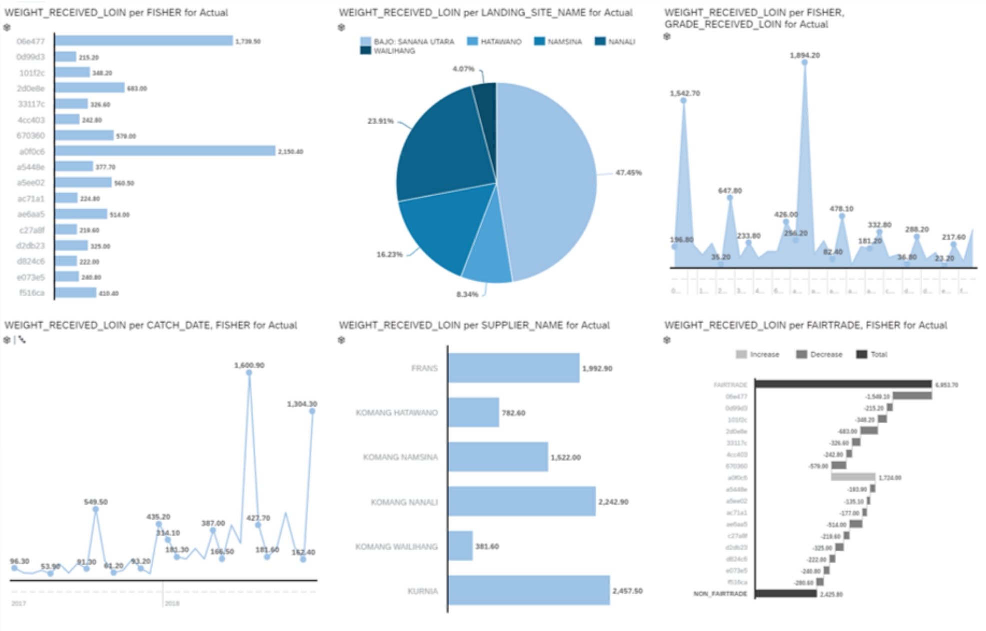


Figure 8.6: Standard way of showing overall supplier use-case with SAP Analytics Cloud.

performance of different sales managers during time (year, season, month). Switching between different quantities is simply achievable along with many other interaction functionalities such as filtering the top N items, getting further details by tool-tips, or excluding a specific item. Moreover, the colors assigned to dimensions, labels, and all text font can also be customized.

Indeed, when we asked product owners to test our visualization with this dataset, they mentioned that "what can be represented with Parallel Hierarchies is normally visible in other charts". Yet, they also said that "It is much easier to track the progress and identify the strengths and weaknesses and gain insight into the key business drivers with this new visualization". After evaluating the visualization within SAP, we started creating more business stories for the customers of SAP eAnalytics Cloud with their real data. In the following, we describe one project example that uses Parallel Hierarchies for analytics tasks.

8.2.2 Ocean to Table Project

Building transparency throughout the supply chain is a decisive factor for success in the food industry. Often many stakeholders are involved and they need to trust data entered by others. A project was started at SAP with the purpose of validating existing blockchain services. It aims to fulfill all requirements for a tracing use case involving different stakeholders. The project is called *Ocean to Table*. The goal was to validate a pilot with one customer, who is selling tuna to restaurants and retailers. The results of the pilot should help the team consider the right problems. Furthermore, it should help the blockchain team to adapt blockchain services in case of identified gaps.

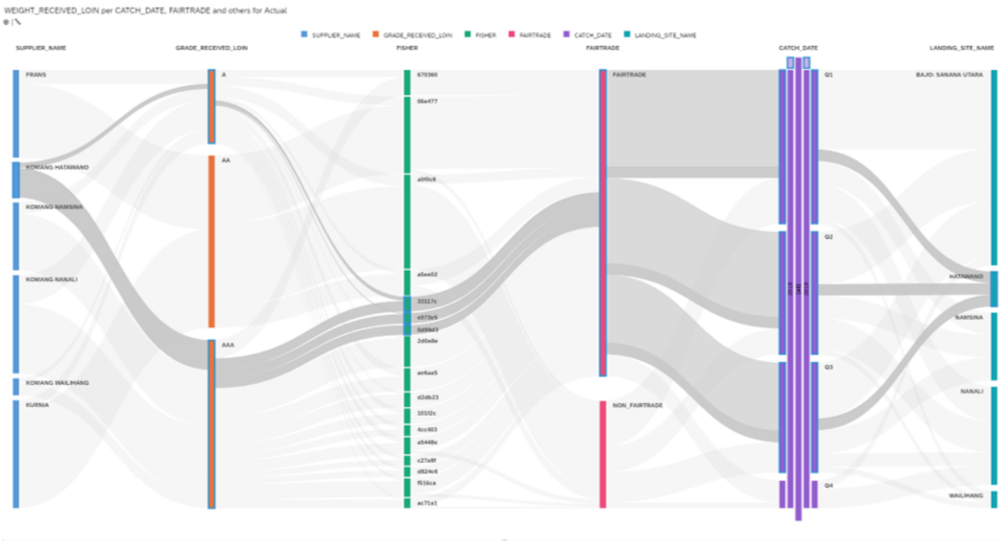


Figure 8.7: Using Parallel Hierarchies to show overall supplier use-case with SAP Analytics Cloud.

The first step toward this goal was to understand the current challenges within the food industry, such as the problem of handling many loosely coupled stakeholders which have low or even no IT budget, or finding a solution to handle data in a way that each stakeholder can only see the data needed for their business. We listed the architectural questions caused by those challenges such as: who should own a node and who will share a node?, what data should be stored in the blockchain and what data should be stored centrally?, or how to support different permissions for different stakeholders. Finally, we gathered the questions related to requirements, which should be answered before developing the pilot: who are the relevant stakeholders and what information do they need? could these requirements be addressed by a standard solution? If not, what types of extensibility would be needed in a standard solution to address these requirements? The proposed solution addresses two scenarios: one pilot scenario should enable a consumer to trace one specific fish until its origin and get information on whether fishing was done in a sustainable way. The second pilot scenario should help further analytics of fishing over time.

SAP Analytics Cloud was used in this project to visualize the final data for both consumers and stack-holders. One example of current visualizations to analyze the data is shown in Figure 8.6. Mainly bar charts, line charts, and pie charts were considered. After two rounds of group discussions with one of the main customers of the project "Bumble Bee Foods" and SAP product owners, the following use-cases were listed:

- *Best Fisher Use-Case*: Who is the best performing fisher by landing site and supplier?
- *Fish Quality Use-Case*: What is the quality of the fish (grade/ fairtrade) got by each fisher?
- *Catch Size Distribution Use-Case*: How catch size varies over time with respect to different suppliers?
- *Overall Supplier Use-Case*: How to get an overview of different suppliers over all data categories?

Figure 8.7 shows one example of the overall supplier use-case using real data from Bumble Bee Foods. One can simply get an overview of the performance of “Komang Hatawano” supplier highlighted in this example, by comparing the outcome of the involved landing sites and fishers, the fish quality shown by “Fairtrade” and “Grade_Recived_Lion”, and how this performance varies over time. Currently, visualizing the same relationship between six dimensions of the data with SAP Analytics Cloud is simply not achievable using one single chart type and it might requires several different charts as it is shown in Figure 8.6. Moreover, end user’s effort is required to analyze the relationship between those different charts to obtain the same information. After evaluation of Parallel Hierarchies using customers dataset and gathered use-cases, customers immediately asked to get access to the visualization solution for their future analysis. They found the visualization to be one of the most useful chart type they have seen for their analytics tasks. Due to several requests by SAP customers from different industries, the visualization will be integrated into SAP Analytics Cloud as a standard chart type.

8.3 Summary and Outlook

In this section, we presented two application examples of Parallel Hierarchies that illustrate the effectiveness of the visualization on real datasets. The first example showed how a machine learning based product costing validation tool can be augmented with Parallel Hierarchies. We are planning to evaluate the utility of described approach with more customers during one of the upcoming customer’s workshop.

The second application example described two scenarios conducted to validate the effectiveness of Parallel Hierarchies for SAP Analytics Cloud customers – using *Best Run Juice Company* and *Ocean to Table* datasets. The derived results and feedback and the high demand from several customers lead to the integration of Parallel Hierarchies as a standard diagram type into SAP Analytics Cloud.

Chapter 9

Validation

Validating the effectiveness of a visual design is an important phase of every visualization project. The design space is vast and one visualization that is suitable for one task might be ineffective for other tasks. Also, finding the right questions to ask when verifying that a visual solution meets design goals is crucial and very challenging. Outside of traditional usability metrics, measuring the success of an information visualization is often subjective and complex [Munzner, 2014, p. 67]. In this chapter, we will give an account of the validation process for Parallel Hierarchies.

9.1 Introduction

One leading approach to guide designers towards interactive visualization techniques that solve their intended tasks is the design triangle. It considers three main factors: (1) the characteristics of the data, (2) the users, (3) the users' tasks. Understanding these aspects determines, which interactive visualization and automated analysis techniques are suitable [Miksch and Aigner, 2014]. Moreover, the design triangle covers some of the quality criteria for visualization such as: expressiveness, effectiveness, and appropriateness [Schumann and Müller, 2013]. To review the validity of Parallel Hierarchies based on these quality criteria, we have identified the following requirements of the design triangle for the field of product costing:

Data: Product costing data: multi-dimensional, hierarchical aggregates with additional attributes such as uncertainty and time.

Users: controllers, IT specialists, consultants, and business management users.

Tasks: Exploring cost distribution along different dimensions.

Table 9.1 describes the same criteria for the design of Parallel Hierarchies, based on Munzner's "What, Why, How" analysis framework. It summarizes the tasks, data, and visual encoding/interaction in terms of design choices. Based on these factors, different approaches to validate the following criteria are discussed in this chapter:

Expressiveness: indicates the importance of visualizing exactly the desired information, nothing less or more [Mackinlay, 1986]. We posit that our visualization is potentially expressive, as it has the potential – depending on the interaction of the user – to represent all desired information. All five aspects of the data defined in section 4 can be shown with Parallel Hierarchies. To validate this criteria further, chapter

System	**Parallel Hierarchies**
What: Data	Multi-dimensional table: multiple categorical attributes, one quantitative value attribute.
What: Derived	one quantitative attribute.
Why: Tasks	Find correlation between attributes, and trends, outliers, and extremes within items.
How: Encode	Parallel layout: horizontal spatial position represents different attributes, vertical spatial position shows hierarchical structures, area mark with length channel expresses the quantitative value, colored by dimensions.
How: Manipulate	Select and navigate.
How: Reduce	Item aggregate and item filtering.
Scale:	Items: one million. Attributes: dozens.

Table 9.1: Summary of Parallel Hierarchies based om Munzner's "What, Why, How" analysis framework [Munzner, 2014].

8 summarized the current applications of Parallel Hierarchies.

Effectiveness: refers to the importance of considering the capabilities of the output medium and the human visual system [Mackinlay, 1986]. Effectiveness criteria can be judged with respect to a number of different factors such as the ability to interpret accuracy and completeness of a design with which users achieve certain goals. Since perceptual tasks might be in conflict, effectiveness criteria can be based on the comparison of the perceptual tasks required by alternative graphical languages. For instance, Bertin's graphical technique explained in section 2.2.1 can be used as a graphical language to encode information in presentation graphics. This motivates us to gather some of the fundamental works in this area to validate the effectiveness of different information visualization techniques including Parallel Hierarchies (see section 9.3).

Appropriateness: refers to a cost-value ratio in order to assess the benefit of the visualization process with respect to achieving a given task [Van Wijk, 2006]. In addition to usability questions, perceptual and comprehensibility questions such as those considered in perceptual psychology are important in assessing the appropriateness of a representational encoding and the readability of visual designs. In order to examine the effectiveness and appropriateness of our visualization solution, we describe a novel approach in the following.

9.2 Nested Model Validation Approach

In the course of this research, we followed a top-down approach consisting of the four nested levels of visual design introduced in chapter 4. This methodology is an helpful approach in problem-driven processes, particularly to avoid skipping important steps. In addition, it can be used to validate the visualization design. Munzner suggests different threats to check the validity at each level of this model. She proposes to validate each level of the nested model separately [Munzner, 2014, p. 67]. In the following, we discuss how the effectiveness of our visualization tool can be validated based on this set of threats summarized in Figure 9.1:

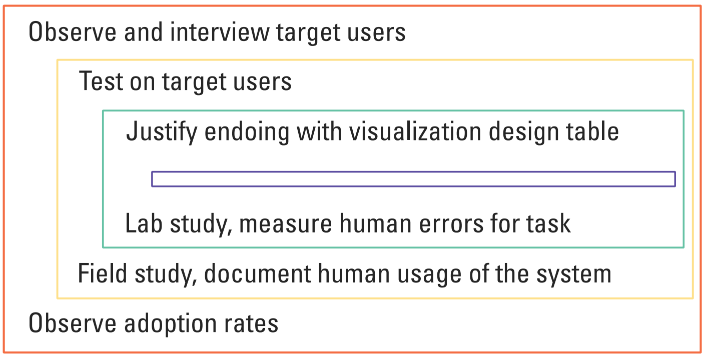

Observe and interview target users
Test on target users
Justify endoing with visualization design table
Lab study, measure human errors for task
Field study, document human usage of the system
Observe adoption rates

Figure 9.1: Parallel Hierarchies four nested validation levels based on [Munzner, 2014, p. 76].

Domain validation: This level of validation is about making sure we have not picked the wrong problem or misunderstood customers' needs. The domain situation phase of this project was addressed in section 4.3. We discussed the process of researching product costing application domain with regards to information visualization. We identified existing end users tools and requirements. To fulfill the validation step, we can refer to several interviews and user studies with SAP customers during co-innovation workshops. From the very beginning of the project, we observed and interviewed target users and to make sure the problem is not mischaracterized, we defined the tasks and requirements in several iterations. The potential users were involved in different phases of this research to help us in understanding requirements. Lastly, we checked the adoption rate of the tool by our intended users.

Abstraction validation: After understanding customers' requirements, we gave an account of the data and task abstraction process in section 4.4 to avoid a domain-specific solution. To validate this threat, we ran several usability tests with the customers of the project during our co-innovation workshops (see section 4.2). Apart from all initial field study and interviews, the visualization design has been shown and evaluated with end users doing their daily work. Three customers of the project, right after testing the tool, requested a product based on our visualization solution. We prepared further examples for those customers interested in using the visualization with their own data to let them make discoveries. Finally, the adopted version of the tool was also evaluated and tested by bio-informaticians from the Chan-Zuckerberg Biohub[1] and demographers from the Max-Planck Institute.

Idiom validation: At the third level of the nested model, we established a solution to represent and manipulate abstract data guided by the abstract tasks (see section 5). One way to validate the chosen idioms is to verify the visualization design based on established perceptual and cognitive principles. To that end, a design table is proposed to validate and compare Parallel Hierarchies with seven other relevant data visualizations in section 9.3. However, except for Hierarchical Chord, which covers a smaller number of data items, other visualizations could not be considered as alternatives to our solution. Although, there is no alternative solution to visualize the same scale of multi-dimensional hierarchical aggregates, each selected visualization covers one or two characteristics of the researched data.

[1] www.czbiohub.org

Also, during our formal qualitative evaluation described in section 5.5, we used the User Experience Questionnaire (UEQ) to check if the solution fulfils the general expectations [Laugwitz et al., 2008]. All measured factors received a value around 2, which is in the *good* category based on the benchmark classification. For the uncertainty solution we quantitatively evaluated the participant performance such as error rate again along the lines of UEQ. To evaluate the new design of flow diagrams' nodes, we used the design expert evaluation method. Based on their feedback obtained during informal validation sessions with users, we optimized important quality metrics of the solution. For example, the crossing optimization algorithm was developed as a solution to minimize the number of edge crossings. Also, limiting discrete gradient method to ten for visualizing the uncertainty on flow diagrams ribbons was another metric defined with the design experts. The currently applied interaction techniques have also been defined and adopted based on customers' requirements.

Algorithm validation: The last level's concern is about computational issues. In designing each visualization solution, there are different choices involved in creating a quick and effective solution. Our solution has been improved during the implementation and evaluation phase in terms of memory performance. The main change was made in regard to the way items are stored in JavaScript Object Notation (JSON) format. The implementation has been constantly improved in order to represent more items on the screen. However, Munzner suggests analyzing algorithm complexity based on the number of pixels in the display. For instance, measuring the wall-clock time and memory performance of the implementation is an interesting future avenue [Munzner, 2014, p. 80].

9.3 Perceptual Validation of Visualization Techniques

Up to this point, we focused on how our expert users have executed their complex data exploration and analysis tasks as efficiently and effectively as possible. In this section, we introduce a design validation table that incorporates three previously published design principles. The principles will be discussed in the context of eight visualization solutions relevant to this work, representing different data characteristics but similar tasks. Although there are several lists of usability evaluation techniques to validate information visualization, there are very few specifically tailored to visual design. Zuk et al. propose a hierarchical way of grouping heuristics for information visualizations [Zuk et al., 2006]. In their taxonomy shown in Figure 9.2, the perceptual aspect of information visualization can be evaluated separately by considering criteria such as color, gestalt, aesthetics, and preattentive. However, there is no common way to quantify or benchmark the perceptual part of the evaluation tree suggested by Zuk et al. Some previous works attempted to introduce mathematical metrics to define aesthetic quality of interfaces [Ngo et al., 2003], but there is currently a lack of methods to help visualization designers or a lack of theoretical frameworks to analyze the current design rationales [Moere and Purchase, 2011].

In presenting a model of potential principles for visualization design, we suggest a design validation table inspired by some fundamental design criteria that can be extended and applied to information visualization. First, we picked the well-known design statements from Gestalt laws and turned them into visualization statements. Our approach is informed by a minimal set of principles by refining a larger potential set of heuristics into a small set that is intended to be general and easily understandable. The proposed validation table permits the comparison of different information visualizations, and the subjective interpretation of the design validation. It can be used both during the design and evaluation phases of development. Moreover, it can also be applied to paper-based designs before the first working prototype is created. We aim to assist the visualization designer in understanding the important cognitive principles that can improve their design choices. Therefore, the proposed design table should be understood as a set of design guidelines or criteria for evaluation of information visualizations rather than as a proposal of design methods.

A number of authors have offered design principles in different areas of information visualization design, such as Tufte's general design [Tufte, 1983], or information design [Pettersson, 2002], or even a recent automated design tool by Moritz et al. [Moritz et al., 2019]. Some of these design principles are rather broad and general, while others are quite specific. Our goal is not to use these design principles and develop design rules telling designers how to design an adequate visualization solution for a specific data and task sets. The goal is rather to help information designer to analyze and understand the impact of specific design principles, and find more practical design solutions.

9.3.1 Design Validation Table

The design validation table incorporates three sets of previously published design principles to assess the visual decisions of specific data visualizations. The selected approaches are considered as visual perception and design guidelines that are general and act as a validation of design choices. Other principles related to aesthetics, choice of colors, or readability of text can also arise in this evaluation. However, we are focusing on more general aspects that can be simply generalized. For example, aesthetics is not covered here as it is a very subjective concept. Users have different opinions of what they find beautiful. The information designer has to consider the readability of text and choice of colors with concrete examples. There are numerous ways to implement different data visualizations, but our focus is on general visual form rather than interaction or analysis. For instance, a Treemap is designed in

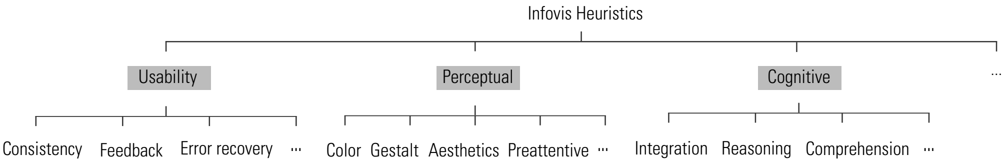


Figure 9.2: Evaluation Tree derived from [Zuk et al., 2006].

many different ways: outfitted with diverse interaction techniques or designed in entirely different ways. However, to explain the idea better and assist easier examination of the general layout, one implementation example of each distinct visualization types is shown in Figure 9.3: Treemap [2], Icicle Plot [3], Sankey diagram [4], Sunburst [5], Parallel Sets [Meirelles, 2013, p. 70], Chord diagram [6], Parallel Hierarchies [Vosough et al., 2018a], and Hierarchical Chord diagram [Meirelles, 2013, p. 64].

[2] ncva.itn.liu.se/education-geovisual-analytics/treemap
[3] www.cs.middlebury.edu/ candrews/showcase/infovis_techniques_s16/icicle_plots
[4] ec.europa.eu/eurostat/web/products-eurostat-news/-/WDN-20190329-1
[5] https://food52.com/blog/15618-what-it-means-to-reinvent-the-coffee-flavor-wheel
[6] www.data-to-viz.com/graph/edge_bundling.html

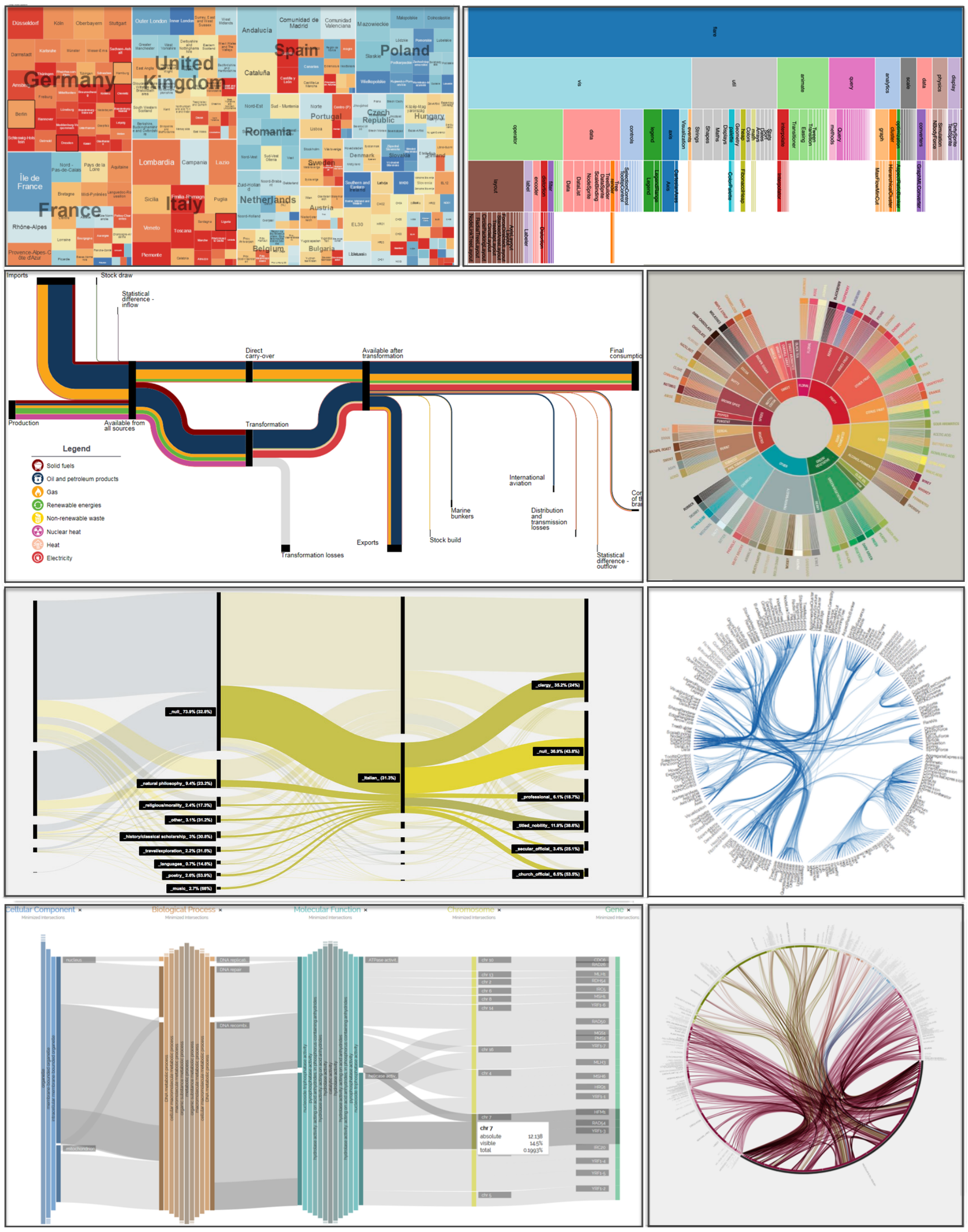
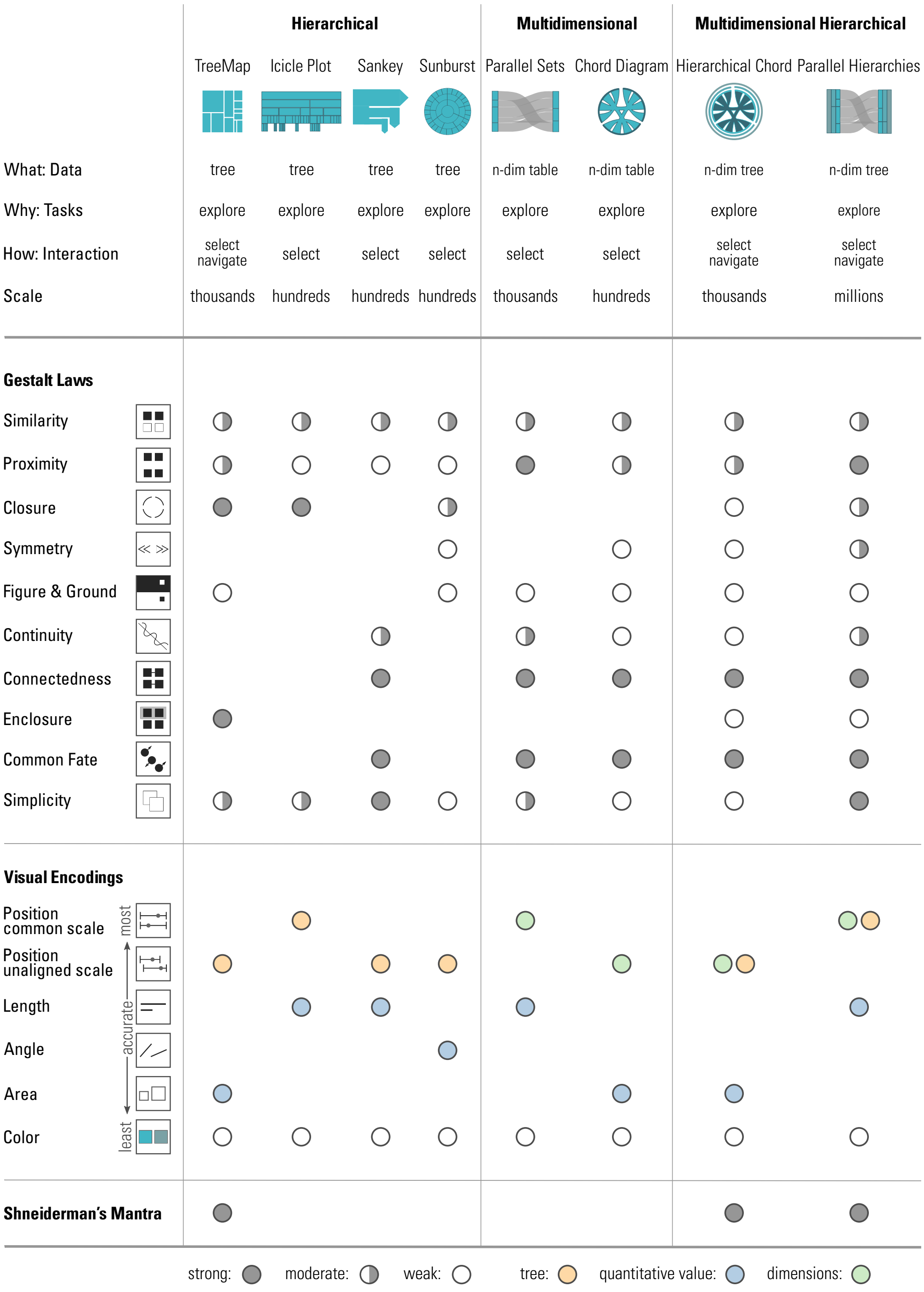


Figure 9.3: Examples of the eight visualization types used to be examined by our design validation table: Treemap, Icicle Plot Sankey diagram, Sunburst, Parallel Sets, Chord diagram, Parallel Hierarchies, and Hierarchical Chord diagram.

The suggested visualization design validation table includes, in addition to the introductory analysis of the data, tasks, and common interactions based on Munzner's classifications [Munzner, 2014], the following design principles:

Gestalt Laws

One of the fundamental works to explain cognitive processes is the Gestalt principles of perception (see section 2.2.3). Gestalt is German for *form* and refers to the interplay



	Hierarchical				Multidimensional		Multidimensional Hierarchical	
	TreeMap	Icicle Plot	Sankey	Sunburst	Parallel Sets	Chord Diagram	Hierarchical Chord	Parallel Hierarchies
What: Data	tree	tree	tree	tree	n-dim table	n-dim table	n-dim tree	n-dim tree
Why: Tasks	explore	explore	explore	explore	explore	explore	explore	explore
How: Interaction	select navigate	select	select	select	select	select	select navigate	select navigate
Scale	thousands	hundreds	hundreds	hundreds	thousands	hundreds	thousands	millions
Gestalt Laws								
Similarity	◐	◐	◐	◐	◐	◐	◐	◐
Proximity	◐	○	○	○	●	◐	◐	●
Closure	●	●		◐			○	◐
Symmetry				○		○	○	◐
Figure & Ground	○			○	○	○	○	○
Continuity			◐		◐	○	○	◐
Connectedness			●		●	●	●	●
Enclosure	●						○	○
Common Fate			●		●	●	●	●
Simplicity	◐	◐	●	○	◐	○	○	●
Visual Encodings								
Position common scale		🟠			🟢			🟢🟠
Position unaligned scale	🟠		🟠	🟠		🟢	🟢🟠	
Length		🔵	🔵		🔵			🔵
Angle				🔵				
Area	🔵					🔵	🔵	
Color	○	○	○	○	○	○	○	○
Shneiderman's Mantra	●						●	●

strong: ● moderate: ◐ weak: ○ tree: 🟠 quantitative value: 🔵 dimensions: 🟢

Figure 9.4: Validating the design of eight data visualizations relevant to this work with respect to known perceptual principles.

between the parts and the whole. We apply it on the visualizations' overall form and try to analyze how users typically gain meaningful perceptions from different visualization designs. The goal is to utilize the Gestalt principles in order to help visual designers conceive solutions that are easier in identifying organisationally structured elements. Ideally, these elements should be naturally and simply understood to be able to display even very complex datasets. Gestalt principles help understand how we perceive a picture or graph. Our intention in the context of information visualization is to exploit this characteristic to interpret user attention towards the important parts of the visualization. In order to adapt Gestalt principles to the field of visualization, we first provide unambiguous definitions for the laws. Then, the definitions will be further explained by the visualization examples shown in Figure 9.4. The original Gestalt statements are of a very general nature and we are aware that the laws can be defined in other ways. We first examine whether the laws are relevant to each visualization type and then to which degree. To avoid having a too simplistic judgment – by assigning only true or false – and to facilitate understanding of each law's impact on the whole picture, we assign a circular symbol to each column of the table. The symbol indicates the *grouping impact* via an empty (weak), half filled (moderate), or entirely filled (strong) circle. The following Gestalt principles of organisation have been considered to validate the perceived visualization form or structure in the overall layout:

1. The law of **similarity** is a very common and general law that can be found in different aspects of any visualization solution such as similarities or differences in size, shape, color, etc. Similarity occurs in visualizations when we perceive objects as group or pattern because they have the same appearance. To understand the different levels of importance of those different graphic traits (size, color, etc.), the level of accuracy of visual encodings is included in this table. For example, in all examples listed in Figure 9.4, similar form, color, and size is used to facilitate the grouping of items. Color code, as a less accurate factor, is used for indicating different levels of hierarchy in hierarchical visualizations such as *Icicle Plot* or *Sunburst* where it can be extremely helpful to distinguish and group items. For example, the law of similarity help perceive the items on the same angle as a group in *Sunburst*. The impact of similarity is ranked as medium in all eight examples. Each visualization type addresses different visual channels that do not contribute to our assessment. Rather, it is based on comparing the impact of all Gestalt laws used for grouping.

2. The law of **proximity** is a widely used principle in designing user interfaces and it is very important for visualization design as well. Based on this principle, items that are placed near to each other form a group. In visualization design, this law is used to position items with similar characteristics close to each other. This principle help group items in all eight examples shown in Figure 9.4. It provides a powerful organization principle and facilitates the detection and search for data items. This law can be one of the strongest principles among the listed laws in the table, when used effectively. For example, in *Parallel Sets* and *Parallel Hierarchies* this organization principle is a very strong tool for visualizing multiple dimensions. Although, it is used with the same objective in *Chord* and *Hierarchical Chord* diagrams, it does not appear to have the same strength, due to space limitation of circular layouts.

3. The law of **closure** shows how items in a complex arrangement tend to be grouped into a single recognizable structure despite the absence of some of its parts. In information visualization design, this law is considered to be relevant when it assists the grouping of data items by filling gaps. Among the eight visualization examples, *Treemap*, *Icicle Plot*, and *Sunburst*, are decent examples of how our brains have a perceptual tendency to fill-in gaps in the contour of rectangles (Treemap and Icicle Plot) and circles (Sunburst). Similarly, for *Parallel Hierarchies*' axes this law helps to group ancestor items and see a simple diamond shape, or in *Hierarchical Chord* diagram it helps perceiving arcs as enclosing all their sub items. In all previous examples, lines are used to divide a rectangle or circle and shape hierarchical structures, rendering the entire picture complete and recognisable.

4. The law of **symmetry** states that the symmetrical areas tend to be grouped as figures against the asymmetrical background. The mind prefers symmetry over imbalance, but this does not mean that the visualization should reflect a mirror image (symmetrical balance). For instance, visualizations examples with circular layouts can be seen and remembered because their layout without the content is symmetric. Moreover, the principle of symmetry is explicitly used in the design of *Parallel Hierarchies*' axes to represent the same items on both sides of the middle axes. This supports the law of continuity and connectedness on the ribbons of the visualization, which leads to a better grouping and classification of connected items. The law of symmetry helps in determining figure-ground perception as well.

5. The law of **figure & ground** states that our visual system tends to distinguish items as figure from the background. Surroundedness, size, symmetry, parallelism, and extremal edges are five important factors that help perceive figure and ground [Palmer and Ghose, 2008]. Giving a general assessment of this design law is hard because it depends on concrete instances of each visualization type. However, based on the mentioned factors, visualization types that fit into a simple shape (rectangular or circular) such as *Treemap*, *Sunburst*, *Parallel Sets* or *Parallel Hierarchies*, and *Chord* or *Hierarchical Chord* diagrams can be recognized immediately as the figure. However, Icicle Plot and Sankey diagram's accidental form do not follow this case and they need to be examined individually.

6. The law of **continuity** states that our visual perception tends to follow continuous, straight or curved lines even when they are disconnected. This is an important principle in information visualization design, especially in the design of maps and networks. In all represented ribbon-based visualization types (flow diagrams), this law assists perception of the occluded parts and seeing them as a continuous lines or curves. In *Chord* and *Hierarchical Chord* diagrams this occurs by resolving the ambiguity caused by ribbon crossings which can cause disconnections in the objects. In *Sankey*, *Parallel Sets*, and *Parallel Hierarchies* continuation occurs when the ribbons are seen as connected despite the gaps. The impact of this law on grouping connected data items across multiple dimensions is stronger in parallel layouts in comparison to circular layouts, due to the smooth continuous flow between items without abrupt changes in direction.

7. The law of **connectedness** is similar to the law of continuity, and is of crucial importance in the design of networks such as flow diagrams. This law is used to group elements by connecting them using lines or other shapes. All five flow diagrams shown in Figure 9.4 (*Sankey, Parallel Sets, Chord, Hierarchical Chord*, and *Parallel Hierarchies*) employ this connectedness principle to group items that are connected.
This helps users to explore patterns and relationships between connected items. This grouping effect is powerful even when it contradicts other principles such as proximity and similarity.

8. The law of **enclosure** states that items that are enclosed within a closed region by a border or boundary are perceived as a group. This law is followed in many but not all *Treemap* implementations. Also, in both *Parallel Hierarchies* and Hierarchical Chord diagrams this principle can be identified as a grouping tool for hierarchical characteristics. This law can be also effective in interaction design for different visualizations.

9. The law of **common fate** states that we perceive elements as moving together or pointing in the same direction as belonging to a group. Similarly, in information visualization design, if certain elements have all the same direction, they are seen as one group. The selected examples in this table do not involve movement, but all ribbon-based diagrams follow this principle when a group of ribbons are pointing to the same direction. Parallel ribbons will be grouped into a group, whereas nonparallel ones are perceived separately. The impact of this law in grouping *flow diagram* ribbons is so strong that in many design examples they are merged and shown as one thicker flow instead of showing individual ribbons.

10. The law of **simplicity** is a very general concept that is the fundamental principle of Gestalt. It stipulates that elements are perceived as the simplest forms possible. It is basically the main purpose in information visualization design. Any visualization that supports the other Gestalt laws also aims to support the law of simplicity. This means that the resulting structure is seen as simple as possible. For instance, in the *Sunburst* structure, instead of several segments, one might first perceive the concentric circles. Following this principle assists end users perceive complicated structures easily and quickly. In our method, the simplicity of different visualization types is calculated based on the *average* strength of different laws applied to that particular visualization.

Visual Encodings
Often, design decisions employed by common visualization solutions do not follow graphical perception concepts from elementary works such as Bertin or Card et al.'s principles [Bertin, 1983, Card et al., 1999]. A visual encoding (also known as visual channel, or variable) is a set of primitive visual representations that control the appearance of different values of a variable. Chen and Floridi proposed a taxonomy to classify visual encodings in four categories, called *geometric*, *optical*, *topological*, and *semantic* channels [Chen and Floridi, 2013]. From these categories, the geometric channels are the most relevant. The visual encodings included in our design validation table are derived from Munzner's classification shown before in Figure 2.11. They

have been suggested as a metric to measure the effectiveness of information visualization implementations [Munzner, 2014, p. 102].

Six visual encodings are relevant to this work: position common scale, position unaligned scale, length, angle, area, and color, have been listed (see Figure 9.4). In order to examine their accuracy on the eight selected visualization types. Moreover, to determine which data attributes have been encoded by each visual channel, we assign different colors to the three main attributes. The color channel is assigned only white circles, as it depends on the implementation examples and therefore it does not play a crucial role in our evaluation process. Spatial position is one of the most accurate channels and has been used in several 2D information visualizations to convey information by positioning the data items in the plane. Normally, the important question is which attribute should be encoded with this channel. In the selected examples, hierarchical and multi-dimensional hierarchical visualization types use the position channel to indicate the hierarchical structure of the data. *Icicle Plot* and *Parallel Hierarchies* use aligned spatial positions, which are easier to perceive in comparison to *Treemap*, *Sankey* diagram, *Sunburst*, and *Hierarchical Chord* that use unaligned positions to indicate the hierarchies. In addition, the position channel is used to categorize dimensions of data in all four selected multi-dimensional visualization examples. Both circular visualizations use unaligned position to arrange dimensions in space, but the parallel visualizations use aligned positioning, which can be more accurately perceived.

The channels that are used to encode quantitative values into the spatial encodings are length, angle, and area. After position, length is the most effective variable for encoding quantitative attributes. In many of the shown examples such as *Icicle Plot, Sankey, Parallel Sets*, and *Parallel Hierarchies*, length variation is used to encode the primary quantitative attribute (this is not area since the width of rectangular nodes are fixed). Although angle encoding as used in *Sunburst* performs worse than length, it is still more accurate than area. Moreover, Heer and Bostock verified in their experiment, how the rectangular area (used in *Treemap*) is more accurate than circular area (used in *Chord* or *Hierarchical Chord* diagram) [Heer and Bostock, 2010]. Finally, color is used in all selected examples, but mainly to represent categorical attributes such as tree levels or to distinguish dimensions.

Shneiderman's Mantra

Perception of modern interactive information visualizations is particularly difficult. One main issue is size of the data and the way data items are represented within a limited space. One of the groundwork for designing information visualizations is the Shneiderman's mantra of "Overview first, zoom and filter, details-on-deman" (see section 2.1.3). Amar and Stasko claim that this mantra summarizes the design philosophy of many modern information visualization solutions [Amar and Stasko, 2004]. The mantra has been used widely as design justification in many novel information visualization systems, and as a prescriptive principle for many information visualization designers [Craft and Cairns, 2005]. Therefore, this mantra is considered as another criteria included in the design validation table.

Moreover, we believe that another important aspect needs to be added to this mantra. After “overview first”, a frame of reference should be defined before “zoom and filter” are applied. A frame of reference creates the same conditions for all objects and also for the observers of the objects. In this way the complexity of the overall situation is reduced. In the context of information visualization, defining an inertial frame would help observers perceive a constant and stable picture [Ware, 2013]. For two-dimensional images or diagrams, a rectangular frame is the simplest frame of reference. If the frame changes or is not chosen well, the visual perception may be more complex than necessary. Therefore, we would suggest refining the mantra as “Overview first, *define frame of reference*, zoom and filter, details-on-demand”.

The frame of reference is fixed in Parallel Hierarchies even after drilling down and up in the hierarchies. This helps perceive the data because users can simply return back to the overview. Groh introduces the terms *immersion* and *emersion* in the context of visualization design [Groh, 2017, p. 110]. The term *immersion* describes the state where users can manipulate the data and their perspective. Users are plunged into the dataset often in virtual reality or 3D visualizations. Therefore, orientation is needed. The frame of reference is not fixed, and it can be changed in the flow of interaction. This can be practical when data has a third dimension and changing the perspective helps gaining more insights. In contrast, *emersion* is defined for 2D graphs, maps, or movies. Users have an overview of the data, and they can define a fixed frame of reference for the entire data analysis. This state is practical for data analysis or comparison tasks. There is no depth dimension in the image and comparison of visual items is more accurate. The distance between users and the image does not affect the perception of the contents unlike with images that contain perspective. Moreover, only when a frame of reference is defined, standard design tools such as grids, rasters, or rulers can be used. Therefore, defining a frame of reference in emersive systems is of great importance in designing information visualizations. It also helps with the mnemonic system to represent information in a way that allows for efficient retention. In Parallel Hierarchies’ design we integrate detail views into the overview frame of reference. Thereby, the relation between overview and detail is more obvious and helps the mnemonic system.

In the design validation table, our focus is on showing different levels of granularity by offering overview first, define frame of reference, and then details on demand. The “Zoom & Filter” phase that aims to reduce the complexity of the visualization by removing extra data items from view can be covered by all visualization types represented in Figure 9.4. The mantra can be considered as a guiding principle for information visualization design. However, in the context of visual analytics, the guide has been extended to “Analyse first, show the important, zoom/filter, analyse further, details on demand [Keim et al., 2006], which falls outside the scope of this thesis.

The mantra is an important methodological contribution used in the design of a number of visualization techniques, such as *Treemap*, *Hierarchical Chord*, and *Parallel Hierarchies* from the list shown in the table. The three mentioned visualization types, implement the overview first and details-on-demand technique to overcome the problems presented by large hierarchical datasets. In all examples, the first level of the tree is presented first, and then users can drill down to the levels of interest on demand. As it can be seen in the table, those three visualization types can represent many thousands or millions of data items.

9.3.2 Discussion

While most recent work on the evaluation of information visualization systems typically centers on correspondence of representation between data and task, there remains uncertainty about the ability of current solutions to adequately follow design principles. The suggested design table is intended to reverse engineer the graphical perception of different data visualization types and to facilitate understanding of how different perceptual concepts affect the decoding of visualized data. Although the table can contribute new insights for visualization design, our goal is not to list design guidelines. As Welie et al. have noted, guidelines are often difficult to select, interpret, and apply, they may be too simplistic or abstract, they may conflict with each other, and their validity may not be proven [Van Welie et al., 2001]. Therefore, we propose this table as a methodology for analyzing different visualization designs, which can be also used by visual designers to understand perceptual principles and make better decisions depending on their data and task types.

Three types of visualizations are examined based on the relevant data characteristics to this work: “hierarchical”, “multi-dimensional”, and “multi-dimensional hierarchical”. In each category one circular and one ribbon-based visualization type is included. The user goals for all selected visualizations are intentionally limited to the same tasks, which are based on Munzner’s task classification (shown in Figure 4.9): the high level action of “discover”, the middle level of “explore”, and the low level of “identify” and “compare” [Munzner, 2014]. Accordingly, only “select” and “navigate” interaction techniques are considered for the defined tasks. “Navigate” is applied on visualization types that implement the “overview first” and “details-on-demand” technique of Shneiderman’s mantra. Consequently, those visualization types applying “navigate” interaction cover larger scale of items in their own category. For example, *Treemap* from the hierarchical category is able to represent thousands of items, although the three others cannot be used to represent beyond hundreds of items without developing extra interaction techniques.

A quick look at the table reveals how by using more complex data, more Gestalt principles are required to facilitate the perception of the charts. One can immediately see that more laws have been applied to the charts on the right side of the table such as *Parallel Hierarchies* and *Hierarchical Chord*. Although the laws are used differently in different visualization types, they all aim to support perceptual inference and enhance detection and recognition. *The law of simplicity is defined in this table as a principle that summarizes all Gestalt laws*. It reveals that *Sankey* diagram and *Parallel Hierarchies* are the simplest visualizations, overall, as well as within their own group. Under the multi-dimensional group, *Parallel Sets* appear to be simpler than *Chord* diagram.

The visual encodings were analyzed as another factor that has direct influence on graphical perception. For different set of tasks, different channels might be appropriate. Since the intended tasks have been limited in the current table, the effectiveness of the selected visualizations can be compared based on the level of accuracy of the applied visual encodings on each visualization type. For example, under the hierarchical category, *Icicle Plot* can be ranked as the most effective visualization type from the perceptual points of view. Moreover, *Parallel Sets* are considered more effective than *Chord* diagram as both visual encodings used to represent multi-dimensionality and quantitative value are ranked as more accurate. Similarly, all three visual encodings used to convey information in *Parallel Hierarchies* are more accurate than *Hierarchical Chord* diagram.

Although the included examples in the table are very limited and do not permit general conclusions on the impact of Gestalt law on visualization design, there can be nevertheless some directions derived for further evaluations. For example, when we have several laws in one figure they might be in conflict, but one can immediately see the strong grouping impact of connectedness and common fate in the design of *flow diagrams*. The grouping impact of continuity and proximity is stronger on Parallel flow diagrams rather than circular ones, which makes them simpler to perceive. In all three categories, the visualizations with circular layouts are in general ranked worse than the others and one can see how hard it is to identify clusters in them because of the poor usage of the law of proximity. That is because there is imitated space for arrangement in circular layouts. When the law of proximity (e.g. in *Parallel Sets* and *Parallel Hierarchies*) and enclosure (e.g. in *Treemap*) are properly used, they are the most powerful organization principles. Moreover, using vertical and horizontal spatial channels in the proper way, and for critical data features, can improve visual perception. For example, in *Parallel Hierarchies*, the horizontal spatial position is used to show axes, and the vertical spatial position is used to express value along each aligned axis or vice versa. In summary, *Parallel Hierarchies can be ranked as a sufficiently good solution that takes advantage of all Gestalt laws and the choice of visual encodings appears to be better than in other similar visualizations, along with taking advantage of Shneiderman's mantra to represent more number of items.*

All in all, this table shows the importance of understanding human perception in data visualization. Human perception processes large amounts of information rapidly, and the best visualizations are designed to make the best use of it. Depending on the visualization requirements, Gestalt Laws can be used effectively to partition the visual space, group objects, enhance patterns, and highlight relationships between objects such as is-a, part-of, belongs-to and so on. This makes them a powerful perceptual tool for visualization design. However, visualization designers cannot rely on the perceptual system alone. Due to the complex nature of the data and tasks that visualizations support, it is often not possible to meet the conditions required for ideal perceptual processing. Therefore, although perception is important, it cannot drive visualization design on its own. For example, circular visualizations like *Chord* and *Hierarchical Chord* diagrams follow a more “natural” style, but perceiving the information they convey is more difficult and they are not “visually efficient”. Our work has involved only examples related to Parallel Hierarchies, but it can be simply extended in the future to help other visualization designers to produce effective designs for their intended data and task and avoid common pitfalls. Moreover, Heer et al. mention that graphical perception can be affected by other design parameters and data characteristics, such as contrast effects, plotting density, and changes to chart size, scale, or aspect ratio [Heer and Bostock, 2010]. Also, the relative judgments in perception or well-known visual illusions can be examined and included in the design table [Meirelles, 2013], or preattentive features can be considered with specific examples. Preattentive features are typically used to facilitate target detection, region tracking, counting, and estimation [Meirelles, 2013, p. 22], which are out of the scope of this work. Color was considered as an extra factor to optimize and improve the visualization. Visualization designers often apply colors at the end, in contrast to artists who mainly start with colors.

9.4 Summary and Outlook

Validating a visualization design is considered a difficult task because there are many possible questions that one can ask and numerous aspects of the visualization design that need to be considered [Munzner, 2014, p. 67]. In chapter 5, after introducing Parallel Hierarchies to the field of product costing, we presented its utility with demographic and biological use cases. Then, we presented the evaluation process in an industrial scenario, and later in chapter 6 and 7 the extensions built to cover other aspects of the data were illustrated. Yet, a validation method to argue why we believe the visualization solution is effective from different perspectives was required. In this section, we justified different validation methods as follows. *First*, Munzner's four levels for validation was discussed to examine different threats of validity. Three validation levels, *domain situation*, *data/task abstraction*, and *visual encoding/interaction* idioms were covered in this work as shown in Figure 9.1. However, algorithm validation is not discussed comprehensively and needs to be further researched.

Second, we suggested a visual design table to examine the perceptual aspects of our visualization solution in comparison to seven other relevant visualizations. Overall, the design validation table showed why Parallel Hierarchies can be ranked as a good solution from perceptual aspects. It takes advantage of all Gestalt laws and the choice of visual encodings appears to be better than other similar visualizations, along with taking advantage of Shneiderman's mantra to represent more number of items. There are certainly other perceptual laws, or principles, that can be added to our design table. Some might be more useful for designing information visualizations, others for validating them. But in general, we can conclude that Gestalt laws do give good advice, such as how to group elements that belong together, how to point the attention to important elements, or how to create an impression of simplicity. The main goal was to develop something that can assist designers effectively building visualizations that support perceptual inference. However, more structured evaluation by designers is required in the future. We shortly assessed the value of the current principles in evaluating a visualization and suggest implications for further research of the process of visual design evaluation.

Chapter 10

Conclusion and Outlook

The digital revolution in science and industry has brought a steady accumulation of data. Its sheer size and complexity have become a significant challenge to its assessment and analysis. The field of Business Intelligence exemplifies how the ability to analyze, validate and visualize large and complex multi-dimensional datasets can have a significant societal impact. In the past, critical business decisions were made based on spreadsheets and basic visualizations – such as pie charts or bar charts. Now, with increasingly large and complex datasets, it is simply impossible to access all aspects of the data. Novel information visualization tools are needed to help users see the hidden patterns.

In this thesis, we first reviewed the foundations of data visualization (chapter 2), and discussed important relevant work (chapter 3). Then, we introduced the four nested levels of design based on Munzner's work [Munzner, 2014] (chapter 4). Figure 10.1 shows the summary of our contribution in the context of the nested model design. Both the *domain situation* and *data/task abstraction* levels were covered (chapter 4). Moreover, several *visual encoding and interaction* idioms were discussed (chapter 5, 6, and 7). The last level, which corresponds to all *algorithmic* design choices, is partially covered in chapter 5. Finally, chapter 8 and 9 provide a validation of our approach.

The starting point for this work was a concrete visualization problem derived from the product costing domain. Together with more than 30 co-innovation customers, we followed a user-centric design approach that iteratively refined the solution from an initial idea to the first prototypes. We adopted and mixed best practices and added our own optimizations. While there are different alternatives to our process, our results have

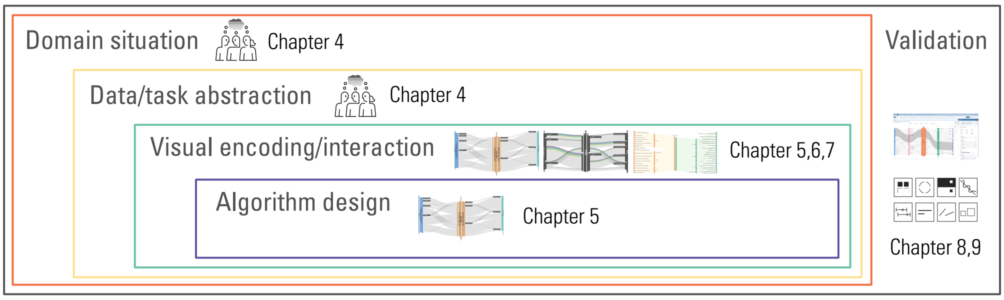


Figure 10.1: Thesis chapters coverage based on the four nested levels of design and validation.

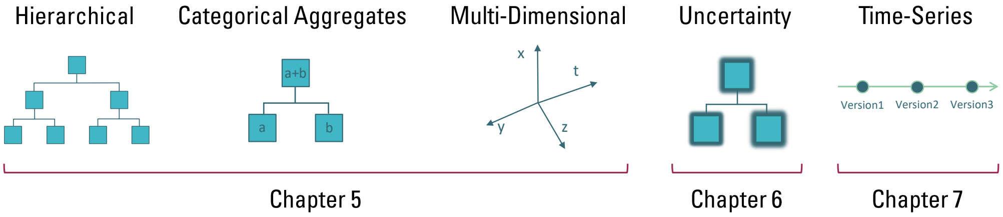


Figure 10.2: Thesis chapters coverage based on data characteristics.

been positively received by the industrial partner (SAP), as well as by our users and collaborators. One notable achievement was to succeed in bridging the gap between visualization science and a concrete industrial problem. After solving this specific visualization problem, we gathered the tasks and the characteristics of the visualized data, and abstracted these into a domain-independent vocabulary to construct a general solution applicable to other domains such as Bioinformatics and Demographics. Figure 10.2 summarizes the data characteristics addressed in the different chapters of this thesis.

In the following, we revisit the open problems stated in section 1 and summarize the key contributions of this thesis. We also discuss the limits, potential improvements, as well as future directions and ideas.

10.1 Summary of Findings

Revisiting challenge 1: Visualizing hierarchical multi-dimensional aggregates. *How to visualize large hierarchical multi-dimensional aggregates? In particular, how to convey the relationships and patterns within and between multiple dimensions of hierarchical data without loss of information?*

In chapter 5, we introduced Parallel Hierarchies as a novel way to display, explore, and most importantly decompose categorical aggregates. Its combination of tree and set visualization elements within the same display space allows for the simultaneous interaction between hierarchical and categorical aspects of the data. This interaction can be utilized for a variety of analysis goals: such as to drill-down into large datasets to find data items with particular characteristics, to identify data items that contribute most or least to a given aggregate, or to trace a subset of data items from a particular category across multiple properties to see how they are distributed.

To demonstrate the generality of Parallel Hierarchies, we applied our solution to two use cases from different domains. One use case considers demographic data, and the other deals with biological data. In addition to a regular validation process during the co-innovation workshops with SAP customers, we conducted a final qualitative user study evaluating Parallel Hierarchies in an industrial scenario. As an additional benefit from this user study, several of the study participants got interested in using Parallel Hierarchies. Since then, these users have requested the integration of Parallel Hierarchies as a product feature in the SAP *Product Life-cycle Costing* suite. Consequently, we worked with different customers datasets and gathered a considerable number of new use-cases for Parallel Hierarchies in the product costing domain (cannot be reproduced here for confidentiality reasons). Moreover, in light of the project's success with SAP customers, Parallel Hierarchies visualization has been integrated as a new chart type into SAP Analytics Cloud – SAP's commercial BI framework. Abstracting the users' tasks from domain-specific form into abstract form allowed us to reuse the solution across many different real-world usage contexts.

Availability: The Parallel Hierarchies visualization is widely accessible to the community on GitHub: https://parallelhierarchies.github.io/. Beside the possibility to import your own dataset, the *pump costing dataset*, the *1990 US Census dataset*, and the *Yeast Gene Ontology dataset* are all included in the GitHub project.

Revisiting challenge 2: Introducing an approach to visualize both data and its uncertainty in flow diagrams.
How to visualize both hierarchical multi-dimensional data and its uncertainty in an integrated manner? How to extend flow diagrams in order to directly incorporate uncertainty information?

In chapter 6, we described different approaches designed to convey uncertainty in the data by modifying the two main visual features of flow diagrams. After conducting interviews with ten product costing experts, we defined all main sources of uncertainty in product costing data. To reduce the complexity of the uncertainty visualization, uncertainty values were discretely aggregated into confidence levels.

First, we presented three uncertainty representation methods applicable to ribbons of flow diagrams. The suggested three methods can encode the five confidence level values currently used in product costing applications. The first solution is used to convey uncertainty with the "Color-code" method, which is suitable for small ranges of discrete values. In contrast, our two other solutions "Gradient" and "Margin" can represent continuous values. We conducted a user study with 32 participants involving the solution of different product costing tasks using the three different visualizations. The result of the user study motivated the design of a new approach that benefits from all three proposed methods. This approach is based on the margin method but uses discrete gradients for visualizing the best and worst cases. Also, the revised version's color codes are based on a recent Value-Suppressing Uncertainty Palette that are proven to be better perceptually distinguished. Furthermore, we applied our revised solution on Sankey diagrams, Parallel Sets, and Parallel Hierarchies. Moreover, we suggested 5 different solutions to modify the rectangular nodes of the flow diagrams to represent uncertainty information. In this phase of the research, we conducted an expert evaluation in order to choose the most suitable visual representations to be used in our application. The preferred solutions for the nodes was adding forks or filled blocks. All solutions are designed to give first a quick overview of uncertainty in the data as a preliminary step before engaging in further investigation.

Revisiting challenge 3: Validating the perceptual effectiveness of visualization techniques.
How to assess perceptual principles used for evaluating visualizations? What are the implications for further research on visualization visual design?

In chapter 9, in an effort to validate the effectiveness of our visualization design we established a design validation table that examined the perceptual aspects of different visualizations. The suggested table extends the notion of Gestalt principles to the field of information visualization. To that end, we turned the well-known design statements of Gestalt laws into visualization statements. The validation table incorporates two other previously published design principles that can be generalized. The visual channels derived from Munzner's classification examines the effectiveness of these channels to encode different data attributes in each visualization type. The last aspect included in the table is Shneiderman's mantra that makes stipulations on how modern interactive information visualizations should be designed with regards to dataset size.

The suggested validation table can be used both in design and evaluation phases of a visualization project, but it also can be applied to paper-based designs – before the first working prototype is created. We examined eight visualization solutions relevant to this thesis – hierarchical, multi-dimensional, and multi-dimensional hierarchical, and discussed their characteristics as they pertain to perceptual and cognitive processes. For example, our validation method helps identify the most effective Gestalt laws for the design of flow diagrams.

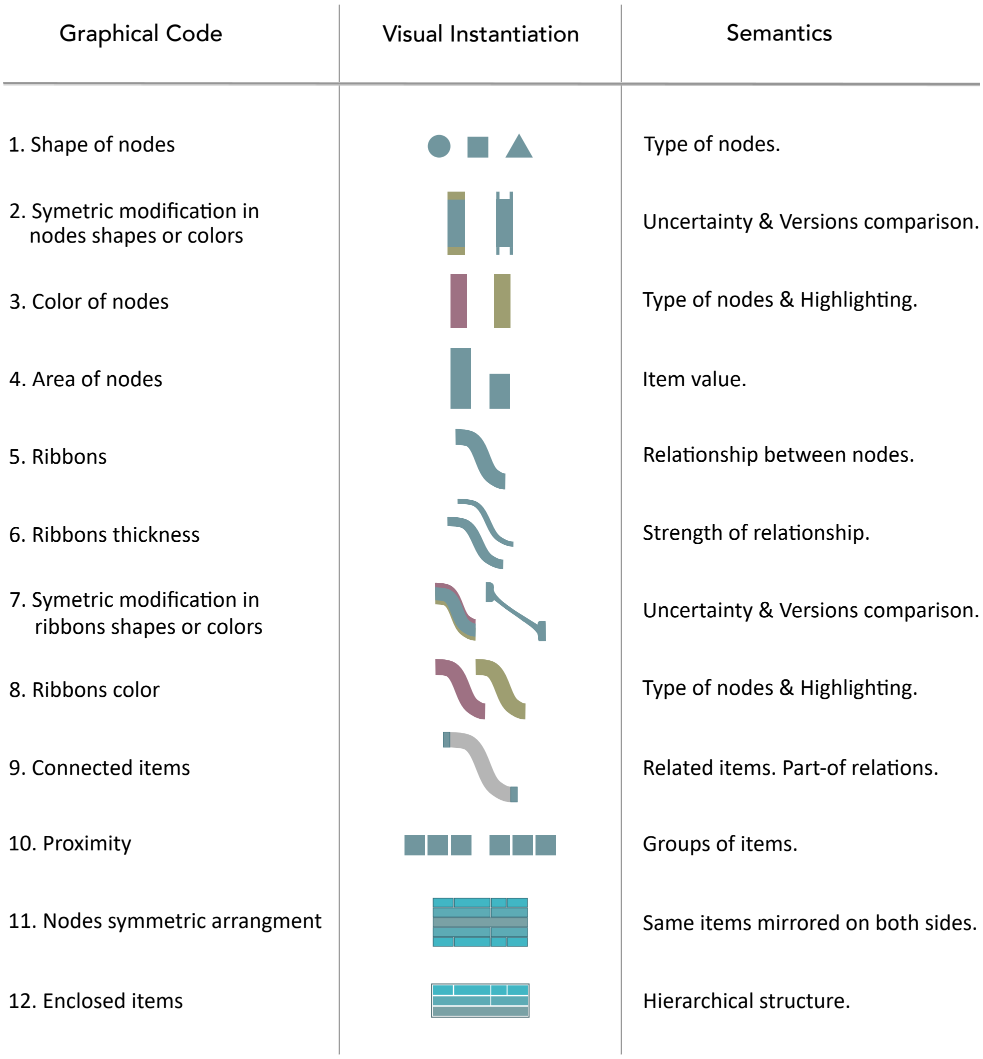
Accordingly, we summarize the visual grammar of flow diagrams' elements in Figure 10.3. All design decisions for different solutions discussed in the course of this research follow the grammars shown here.

10.2 Discussion

In the following, we review the limitations of our approach and possible improvements for future work.

Parallel Hierarchies Scalability Considerations
As with any visualization, after surpassing a certain data size, Parallel Hierarchies become ill-suited. While our visual and algorithmic design aims to push this point as far out as possible, at some point, the number of axes and categories reaches a point at which they cannot be faithfully represented anymore.

As for its *visual scalability*, according to our experience, Parallel Hierarchies' sweet spot lies at 3-5 axes showing hierarchies that are 3-5 levels deep with a branching factor of 2-10 subcategories per category. Displaying 6 or 7 axes is already hard to read and interpret, assuming that no duplicate axes are present. However, in practice we decided not to restrict the number of axes, as in some cases users may want to generate wide mural-like Parallel Hierarchies visualizations for illustration purposes.



Graphical Code	Visual Instantiation	Semantics
1. Shape of nodes	● ■ ▲	Type of nodes.
2. Symetric modification in nodes shapes or colors		Uncertainty & Versions comparison.
3. Color of nodes		Type of nodes & Highlighting.
4. Area of nodes		Item value.
5. Ribbons		Relationship between nodes.
6. Ribbons thickness		Strength of relationship.
7. Symetric modification in ribbons shapes or colors		Uncertainty & Versions comparison.
8. Ribbons color		Type of nodes & Highlighting.
9. Connected items		Related items. Part-of relations.
10. Proximity	■■■ ■■■	Groups of items.
11. Nodes symmetric arrangment		Same items mirrored on both sides.
12. Enclosed items		Hierarchical structure.

Figure 10.3: The visual grammar of flow diagram elements. Four of the graphical codes (2, 7, 11, and 12) have been introduced to the field in this work.

Deep hierarchies of 10 or more levels are tedious to explore as they require many drill-down interactions. Similarly, wide hierarchies with a branching factor of more than 20 will make the visualization indecipherable. The ribbons get thinner and this makes them harder to trace, to select, as well as too small to be labeled. To enable interactive exploration of even such thin bundles of ribbons, we have added fisheye distortion and accordion interaction that enlarge small categories and thin ribbons underneath the mouse cursor for simpler selection and temporary label placement.

Lastly, there is the issue of ribbon clutter between axes that remains even after applying crossing minimization and axes reordering. To ease ribbon tracing across such clutter, we are currently investigating a suggestion made by one of the demographers with whom we worked (see Section 5). This expert proposed to completely remove all unrelated ribbons from the view when hovering over a subcategory or ribbon with the mouse. It remains to be seen whether the contextual information provided by the other ribbons is actually necessary, or whether these can indeed just be temporarily cleared while focusing on a category or ribbon of interest. However, common filtering

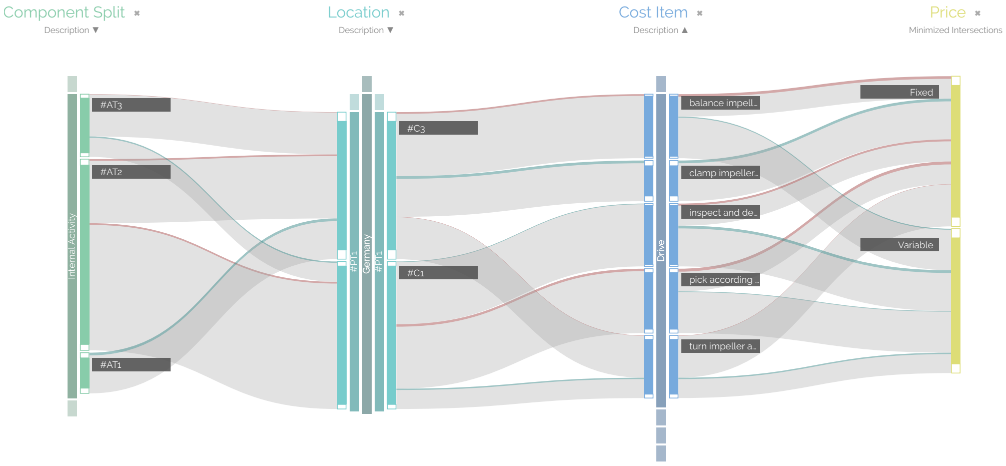


Figure 10.4: Visualizing several dimensions of the pump dataset along with the uncertainty using nodes and 2 versions via ribbons with Parallel Hierarchies.

techniques can easily be added onto the base visualization, such as showing only ribbons above a certain threshold to see the larger trends, or below a certain threshold to see outliers. Features like these would help to scale the visualization to more and thus smaller categories with even thinner ribbons connecting them, by removing them or putting them in focus, respectively. These features are planned for the *SAP Analytics Cloud* version.

As for its *algorithmic scalability*, which includes runtime and memory issues, our browser-based implementation can handle datasets with up to 500k items while maintaining the responsiveness necessary for exploratory analysis. The main technical limits are (1) that all computations are currently done entirely on the client side – from parsing, checking, and processing the data to computing the splines for the ribbons, and (2) that we use SVG-based rendering through the D3 visualization library [Bostock et al., 2011], which inflates the DOM tree to the point where it becomes too complex even for simple jQuery operations.

Parallel Hierarchies Extensions' Limitations

The proposed uncertainty visualization solutions presented in chapter 6, and the approaches proposed for comparing different versions over time presented in section 7 were all designed exclusively for flow diagrams. Explicitly, both tasks were meant to be carried out by extending this specific type of visualization. The proposed solutions for conveying *uncertainty* on both ribbons and nodes of flow diagrams are designed to give a quick impression of the uncertainty in the data. We limited the complexity of the datasets so that they could be effectively plotted using flow diagrams. However, real-life datasets might contain more uncertainty factors which need to be represented individually, and aggregation technique can be deemed as a too simplistic. Also, when datasets get larger and more complicated, our uncertainty approach might not be adequate, because of both algorithmic scalability and limits in visual perception.

Moreover, in the case of the *version comparison* task, apart from the Parallel Hierarchies extension that we presented, we also presented two Sankey diagram-based visualizations – both following the juxtaposition approach – that focused on methods to compare two complex graphs. Further extending flow diagrams beyond two graphs containing multiple dimensions does not appear to be straightforward – it would most likely produce too much visual clutter. Solving this case would warrant a deeper future investigation.

In addition, Figure 10.4 shows one example in which the nodes of Parallel Hierarchies are used to depict uncertainty values, and the ribbons to compare two calculation versions over time for the dataset described in section 5.5. Although is feasible to represent all five characteristics of the data with Parallel Hierarchies as determined during the requirement engineering phase – shown in Figure 10.2 – it is not recommended. The current browser-based implementation (available on GitHub) allows for uncertainty and version comparison to be presented together, but the integrated version into SAP Analytics cloud does not. The Parallel Hierarchies solution was developed to possibly be adapted for version comparison task *or* to depict the uncertainty in the data. Yet, we did not yet find any realistic use-case that require both tasks simultaneously.

10.3 Outlook

Technical Improvements. In the future, a number of technical improvements are desirable. One could attempt to scale-up our solution to larger datasets by following current trends in web-based Parallel Coordinate displays that use Apache Spark backends in combination with hardware accelerated WebGL rendering [Heinrich and Broeksema, 2015,Sansen et al., 2017]. Another approach would be to utilize a hierarchical data format that follows the hierarchies defined over the categorical values and allows us to selectively load only pre-aggregated portions for the current view – thus speeding up data transfer. Consequently, the algorithm validation threat of the nested model can be covered by analyzing the computational complexity of our solution such as by considering the maximum number of pixels displayed in the screen.

In addition, better data abstraction and reduction approaches are required to scale our solution with respect to dataset size. Several techniques may be used to abstract multi-dimensional data such as dimension reduction, random sampling, aggregation, and segmentation [De Oliveira and Levkowitz, 2003]. Our first steps towards extending Parallel Hierarchies will focus on scalability. Support for larger datasets will be done using clustering methods and unsupervised machine learning algorithms. In practice, our initial observations suggest that clustering approaches will need to be tailored for different users and datasets. Some datasets might be homogeneous and suitable for abstraction, while others might be heterogeneous and challenging to abstract. For a given dataset, proper aggregation methods can be identified and applied when the number of items exceeds a threshold per dimension. It remains to be seen whether this is acceptable for users. We are not aware of any existing comprehensive way to solve this problem for a wide range of heterogeneous datasets.

Further Evaluation. Another direction for future work is to conduct more evaluations on analysis tasks carried by end users with the version of Parallel Hierarchies integrated into the SAP Analytic Clouds solution. Interesting questions are: How often do users work with the visualization solution? Which type of discoveries they have made? And how much quicker our solution performs compared to other available chart types or spreadsheets? Also, interesting would be to establish the suitability of Parallel Hierarchies in biological research scenarios. Two collaborations on this topic are ongoing. First, the health check project at SAP aims at visualizing and analyzing medical datasets with Parallel Hierarchies. Second, researchers at CZ BioHub are using our visualization solution to enhance their bioinformatics analysis particularly on single cell transcriptome datasets. Finally, evaluating eye movements of the end users while navigating Parallel Hierarchies would help understand the cognitive impact of design elements of the visualization [Toker et al., 2013].

Extending the Design Table. The current design validation table, described in section 9 incorporates three sets of previously published design principles. We suggested this table as a methodology for performing structural analysis of visualization design elements. This table can be further extended by incorporating other relevant perceptual principles. We are planing to examine more visualization types from all 5 groups of visualization techniques for multi-dimensional data by Keim et al. [Keim, 2005]. Such analysis can help derive design principles for visualizing large quantities of information – while retaining interpretability and assist in perceiving patterns.

Flow Diagrams Toolkit. Another planned future work is designing a tool for modular flow diagram creation that allows design of customized flow diagrams. One could simply replace different flow diagram visual features with different visual marks or change their arrangement in space. It would allow users to either, directly pick the flow diagram type, or the tool would automatically suggest the proper visualization based on the data. The visual variables could be customized without any custom programming, through direct manipulation techniques such as drag-and-drop. A project to develop the first prototype for such an extensible, model-driven, and interactive toolkit has been started [Hogräfer, 2018].

Applying Novel Interaction Techniques for Large Displays. Our plans for Parallel Hierarchies' future will also pick up where the integrated version into SAP Analytics Cloud leaves off. One planned future application is the integration into *SAP Digital Boardroom*. This will require introducing new interaction techniques both for large displays and dynamic multi-device setups. Interaction types with visualizations presented on large displays depend on the input capabilities of the display. One popular feedback from our evaluation participants (see section 5.5.3) was to have a tablet version of the visualization. An obvious challenge there is limited display space. This issue can be minimized when mobile devices such as tablets are used in combination with large displays. Therefore, techniques such as GraSp [Kister et al., 2017] or Vistribute [Horak et al., 2019] can be adopted to interact with our SAP *Digital Boardroom* solution.

Adding an Additional On-demand Dimension to Parallel Hierarchies. Another open problem discussed in section 7.5 is to extend Parallel Hierarchies for comparison of multiple hierarchical dimensions over multiple time steps. One possible solution would be to add an additional on-demand dimension to the solution for visual comparison task. Hube et al. introduced a technique to add a third on-demand dimension to Parallel Sets [Hube et al., 2017]. They develop their first prototype for the visualization of search results with faceted search techniques using tablets. We are planning to examine this approach with Parallel Hierarchies in the future. This would allow us to create a hybrid system space of immersion and emersion (see section 9.3.1) for on-demand comparison task [Groh, 2017, p. 109]. Combining depth and surface could allow users to switch between different versions by moving or rotating the visualization. In any case, developing sophisticated interaction techniques is key for the success of such hybrid systems.

Appendix A

Questionnaires of the Evaluation

Introduction

Thank you for participating in my usability test for a new product costing data visualization!
This questionnaire is designed to evaluate the functionality and usability of a new data visualization built for the SAP Product Lifecycle Costing application. The results of this questionnaire will be used for the purpose of my dissertation, which I write on behalf of the Product Lifecycle Costing project. Additionally, implemented concepts and results based on your feedback might be included in upcoming release of PLC. Please keep in mind that you will be working on a prototype. All data is anonymous and cannot be tracked back to you at any time.
Please don't hesitate to ask for help if you need it, and provide me you r honest feedback. I ask you kindly to follow the "think aloud" technique. That means verbalize your thoughts while interacting with the system. Please speak out loud whatever comes to your mind while interacting with the system: what you see, like, dislike, wonder about or expect.
Thank you in advance for your support!

BACKGROUND QUESTIONS

General Information
Company name:
Age:
Gender:
Years of work experience:

What role(s) do you have in your company (multiple answers possible)?
Controller
Product Controller
Manager
Project Manager
Sales Specialist
Marketing Specialist

Engineer
IT Specialist
Purchaser
Analyst
Cost Accountant
Other:

Please choose the industrial sector of your company:
Industrial Machinery & Components
High-tech
Automotive
Other:

BACKGROUND SCENARIO

First, I will explain how the visualization works with some examples, followed by 5 minutes for you to play with the prototype and get familiar with the system. Next, you will be asked to solve six tasks with the visualization prototype. All tasks are designed to check the functionality of our new prototype. Then you will get two wrap-up questions on visualization use-cases. Lastly, you will be asked to fill in a User Experience Questionnaire (UEQ).

Tasks

Task 1

From which country (Location) does the main part of item “Drive” come from?
Solution:

Task 2

2.1. What is the Price Range of most “Shaft” sub-items?
Solution:

2.2. And from which country (Location) are most of these items (in the price range referred above) come from?
Solution:

Task 3

Which item has the most sub-items of “manual” Price Source?
a) Casing b) Shaft c) Drive
Solution:

Task 4

Which Component Split has only “variable” Cost Portion?
a) Activities b) Materials c) Overheads d) Business Process
Solution:

Task 5

What percentage of “Casing” item comes from “Overheads” (Component Split)?
Solution:

Task 6

What percentage of the total cost belongs to “Fix” Cost Portion?
Solution:

Wrap-up Questions

1. What would be the top 3 use-cases that you could imagine using this visualization for in your daily work?

2. Do you see any use-cases where displaying more than 2 or 3 dimensions would be useful?

3. Do you have any other comments, questions, or concerns?

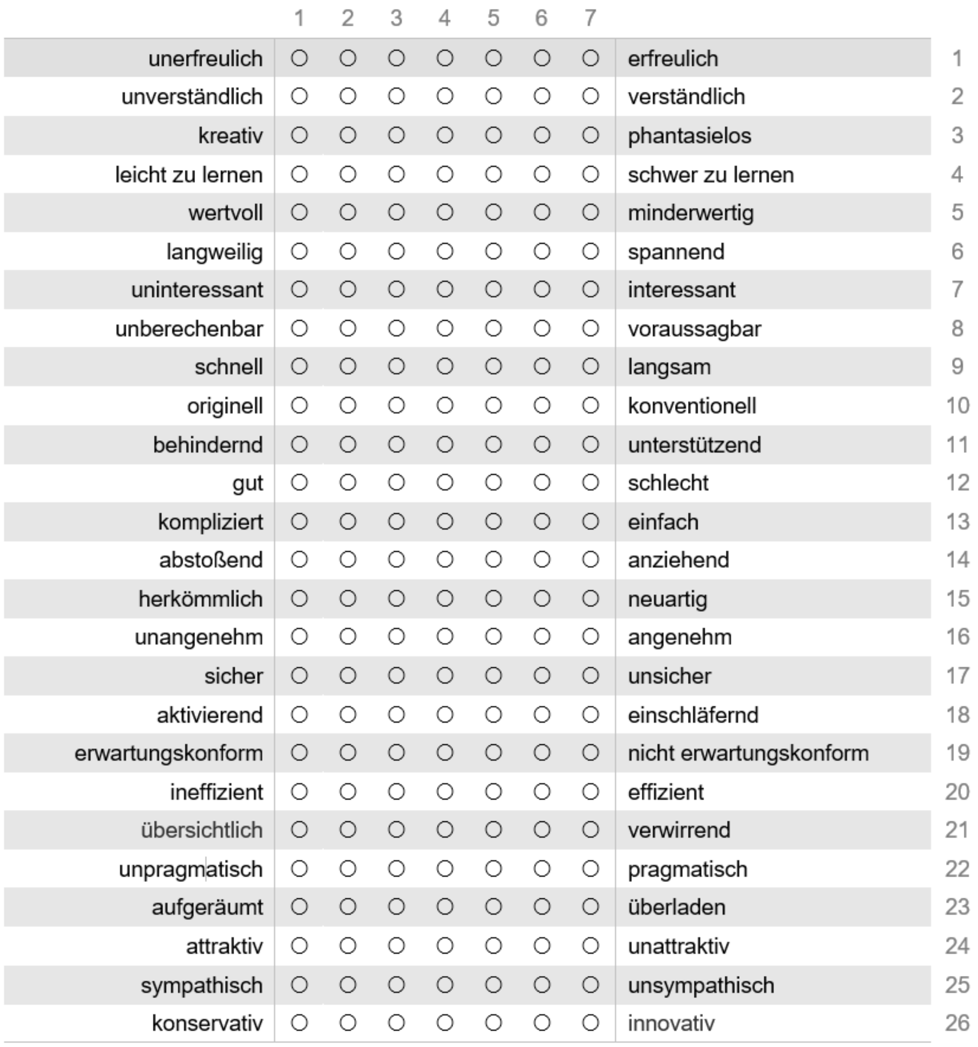
Post-Test Survey

PLEASE MAKE YOUR EVALUATION NOW.
For the assessment of the product, please fill out the following questionnaire. The questionnaire consists of pairs of contrasting attributes that may apply to the product. The circles between the attributes represent gradations between the opposites. You can express your agreement with the attributes by ticking the circle that most closely reflects your impression.

Example: attractive ⊙ ⊗ ⊙ ⊙ ⊙ ⊙ ⊙ unattractive
This response means that you rate the application as more attractive than unattractive.

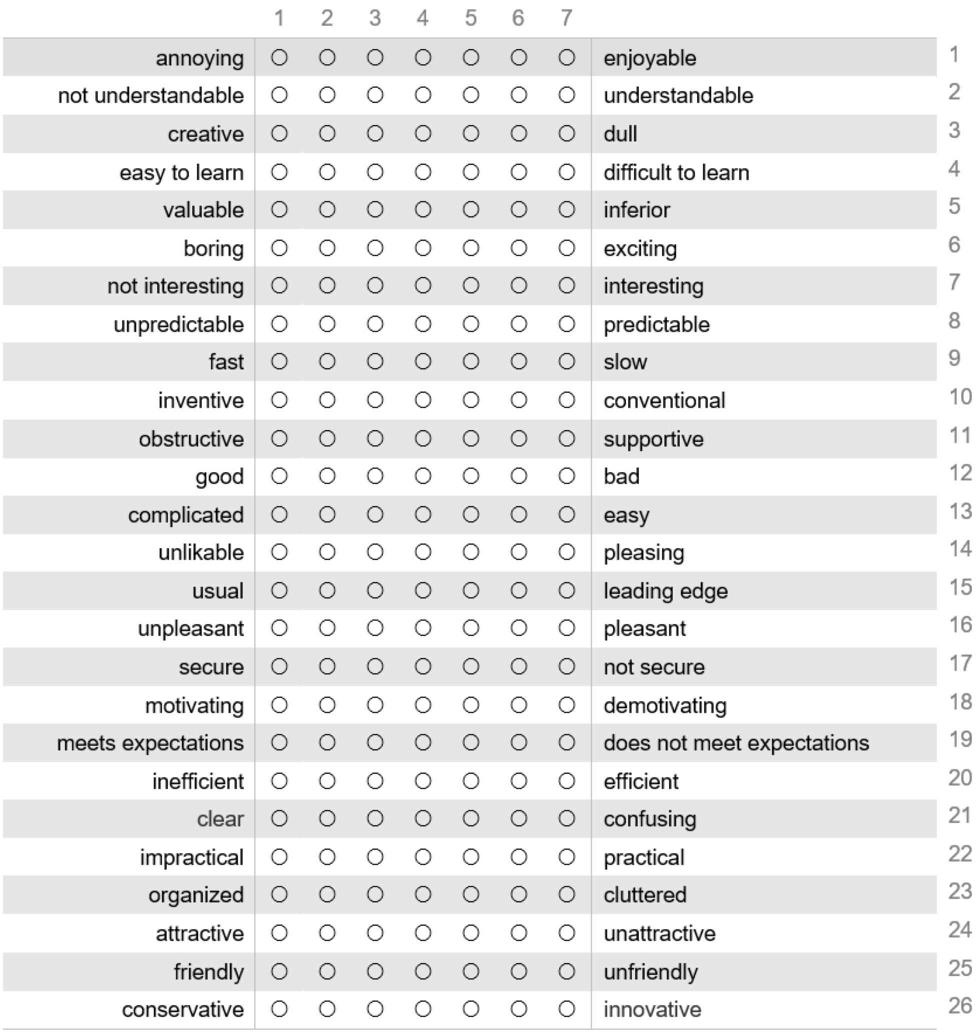
Please decide spontaneously. Don't think too long about your decision to make sure that you convey your original impression.
Sometimes you may not be completely sure about your agreement with a particular attribute or you may find that the attribute does not apply completely to the particular product. Nevertheless, please tick a circle in every line.
It is your personal opinion that counts. Please remember: there is no wrong or right answer!
Please choose between the German and English version and fill out ONLY one of the two questionnaires!

Bitte geben Sie nun Ihre Einschätzung des Produkts ab. Kreuzen Sie bitte nur einen Kreis pro Zeile an.



	1	2	3	4	5	6	7		
unerfreulich	○	○	○	○	○	○	○	erfreulich	1
unverständlich	○	○	○	○	○	○	○	verständlich	2
kreativ	○	○	○	○	○	○	○	phantasielos	3
leicht zu lernen	○	○	○	○	○	○	○	schwer zu lernen	4
wertvoll	○	○	○	○	○	○	○	minderwertig	5
langweilig	○	○	○	○	○	○	○	spannend	6
uninteressant	○	○	○	○	○	○	○	interessant	7
unberechenbar	○	○	○	○	○	○	○	voraussagbar	8
schnell	○	○	○	○	○	○	○	langsam	9
originell	○	○	○	○	○	○	○	konventionell	10
behindernd	○	○	○	○	○	○	○	unterstützend	11
gut	○	○	○	○	○	○	○	schlecht	12
kompliziert	○	○	○	○	○	○	○	einfach	13
abstoßend	○	○	○	○	○	○	○	anziehend	14
herkömmlich	○	○	○	○	○	○	○	neuartig	15
unangenehm	○	○	○	○	○	○	○	angenehm	16
sicher	○	○	○	○	○	○	○	unsicher	17
aktivierend	○	○	○	○	○	○	○	einschläfernd	18
erwartungskonform	○	○	○	○	○	○	○	nicht erwartungskonform	19
ineffizient	○	○	○	○	○	○	○	effizient	20
übersichtlich	○	○	○	○	○	○	○	verwirrend	21
unpragmatisch	○	○	○	○	○	○	○	pragmatisch	22
aufgeräumt	○	○	○	○	○	○	○	überladen	23
attraktiv	○	○	○	○	○	○	○	unattraktiv	24
sympathisch	○	○	○	○	○	○	○	unsympathisch	25
konservativ	○	○	○	○	○	○	○	innovativ	26

Please assess the product now by ticking one circle per line.



	1	2	3	4	5	6	7		
annoying	○	○	○	○	○	○	○	enjoyable	1
not understandable	○	○	○	○	○	○	○	understandable	2
creative	○	○	○	○	○	○	○	dull	3
easy to learn	○	○	○	○	○	○	○	difficult to learn	4
valuable	○	○	○	○	○	○	○	inferior	5
boring	○	○	○	○	○	○	○	exciting	6
not interesting	○	○	○	○	○	○	○	interesting	7
unpredictable	○	○	○	○	○	○	○	predictable	8
fast	○	○	○	○	○	○	○	slow	9
inventive	○	○	○	○	○	○	○	conventional	10
obstructive	○	○	○	○	○	○	○	supportive	11
good	○	○	○	○	○	○	○	bad	12
complicated	○	○	○	○	○	○	○	easy	13
unlikable	○	○	○	○	○	○	○	pleasing	14
usual	○	○	○	○	○	○	○	leading edge	15
unpleasant	○	○	○	○	○	○	○	pleasant	16
secure	○	○	○	○	○	○	○	not secure	17
motivating	○	○	○	○	○	○	○	demotivating	18
meets expectations	○	○	○	○	○	○	○	does not meet expectations	19
inefficient	○	○	○	○	○	○	○	efficient	20
clear	○	○	○	○	○	○	○	confusing	21
impractical	○	○	○	○	○	○	○	practical	22
organized	○	○	○	○	○	○	○	cluttered	23
attractive	○	○	○	○	○	○	○	unattractive	24
friendly	○	○	○	○	○	○	○	unfriendly	25
conservative	○	○	○	○	○	○	○	innovative	26

Thank you for participating!

I appreciate your contribution to my research regarding Product Lifecycle Costing data visualization.

If you have any comments or questions, please feel free to contact me.

Zana Vosough, Ph.D. Candidate

SAP SE, Chemnitzer Str. 48, 01187 Dresden, Germany

Telephone: +49 62 277-45650

Email: zana.vosough@sap.com

Appendix B

Survey of the Quality of Product Costing Data

Phone/ face-to-face Interview:
Date:

Introduction

One of the key success factors of product costing directly depends on the quality of the decision making process, which relates to the quality of data and data certainty. Customers must assess not only the information presented to them, but also the confidence they have in that information. Therefore, they should be aware of the quality of the data for better decision-makings. A good visualization that creates awareness of these underlying characteristics of data, can provide a more complete and realistic visual communication in such a way that the relevant characteristics can be easily grasped by users. The main aim of this research is to find a novel and effective way to visualize the quality of the final cost and the main cause items across the lifecycle of a product. A visualization that help users to investigate the causes of uncertainty in a product and let them to access the impact of this uncertainty on the final price.
The purpose of this interview is to understand the characteristics of data and business requirements better in the SAP-Product Lifecycle Costing.

General Information:
Name:
Gender:
Years of work experience related to product costing:
The primary role in the company:

Questions Regarding Visualizing the Quality of Product Cost Calculation:

Motivation:
What do you think are the main motivations of the customers to see the quality of data?

Which questions do you expect to be answered by this visualization?

Defining rules
What attributes do you think are the main causes of unreliable data in product costing process? (e.g. confidence level)

How could you imagine to calculate the overall data quality?

Data Characteristics
Does the way of presenting data hierarchy matter in your opinion? The importance of the relationship among nodes and showing the tree topology.

How do you think considering the impact of time in the visualization help?(like visualizing cost changes over different versions)

Task Definition
What do the users want to do after finding the items with low quality? (What are the main tasks that should be solved with the visualization?)

What sort of interactive features could you imagine would help the users?

Visualization parameters

How do you expect to see the impact of one item's uncertainty on its parent items? (Top-down or bottom-up approach)

Do you have a target for confidence and if yes how do you manage it? On what level do you set this target confidence?

If and how does the change of uncertainty over time matter to the users?

Do you think the entire tree should be visualized, or only the uncertain item? What about reducing the level of details? (like abstracting the level of details and combining the items with low amount of uncertainties)

How do you consider line items when analyzing the cost? And how important is this perspective? (e.g. 100 screws that appear in different places of one calculation)

Visualization Details

Apart from cost and confidence, would you like to visualize other attributes? If yes which, why and how? (Personalized visualization)

Do you think it is important to show the causes of uncertainty in an aggregated visualization for all items? And if so why and how?

What is your opinion about a 3D visualization?

Good Visualization Criteria

Do you think any constraints could be considered for the visualization?

How do you define the criteria to evaluate a good visualization for this task based on the customers' requirements?

Others

What ideas to you have for visualization of costs? How would your ideal cost visualization look like?

Do you have any other comments or ideas?

Appendix C

Questionnaire of Current Practice

Thank you for participating in my survey regarding data visualization in product costing!

The results of this requirements analysis will be used for the purpose of my dissertation, which I write in the context of the SAP Product Lifecycle Costing project. Additionally, new features based on your feedback might be included in upcoming releases of SAP-PLC.

Thank you in advance for your support!

Introduction

What do I mean with “data visualization”?

A diagram or graphic that shows patterns, trends and correlations in the data for purposes of reporting, analysis, etc., and to facilitate the process of understanding information easily and quickly. The above figure shows some commonly used visualizations.

General Information:

Company name:
Age:
Gender:
Years of work experience related to product costing:
Please provide your e-mail if I may contact you to discuss your feedback:

What is your primary role in your company?
Controller ❒
Manager ❒
Project Manager ❒
Marketing/ Sales Specialist ❒
Engineer ❒

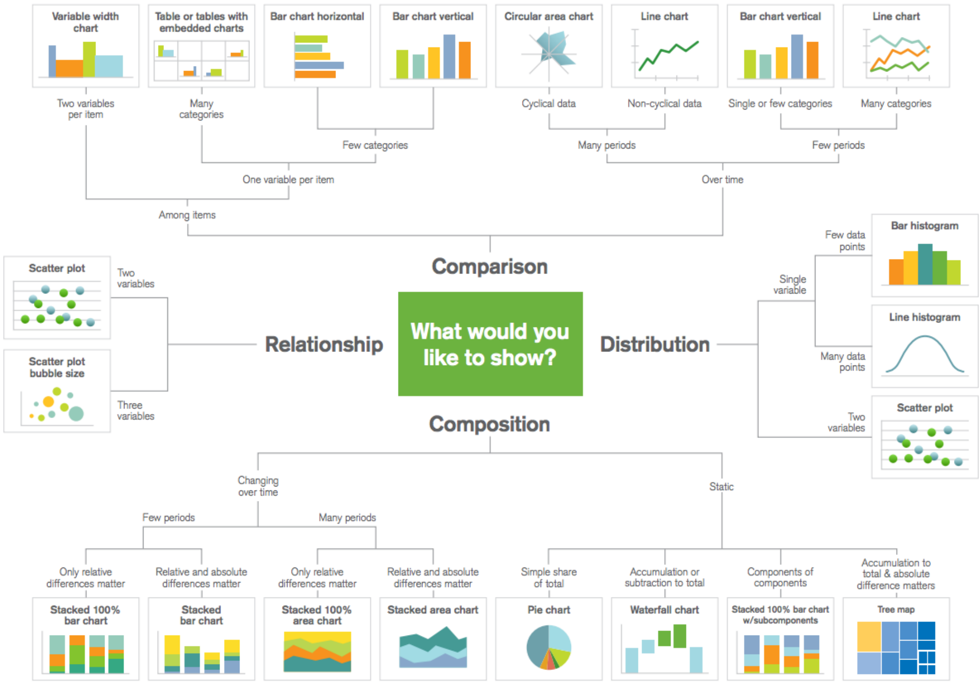


Figure C.1: Derived from:
http://bigdata.black/analytics-predictions/visual-analytics/how-to-choose-the-right-chart/

IT Specialist ❒
Purchaser ❒
Other:

Please choose the industrial sector of your company:
Industrial Machinery & Components ❒
High-tech ❒
Automotive ❒
Other:

Questions Regarding Your Current Product Costing Solution:

Which tool(s) do you currently use to support your product costing process?
Microsoft Excel ❒
Product costing solution developed in-house ❒
3rd party products, please specify (optional):

Data Visualization in Product Costing

For each visualization, please briefly describe the use case, the purpose, the type of visualization used and how it helps you. Screenshots or mockups by email to Zana.Vosough@sap.com are appreciated and are handled confidentially.

In which product costing-related processes could data visualizations help you to finish tasks or solve problems more efficiently (i.e. faster or with fewer errors)? (For each idea, please describe the use case, the purpose, and if applicable, your ideas for a

Where and how do you currently use data visualizations to support your product costing process?	Where do you see shortcomings in these data visualizations? How could they be improved?	How often do you use this visualization? (Always=1, Often=2, Occasionally=3, Rarely=4, Never=5)

visualization. Please also describe how the visualization would be helpful.)

Please rate the following statements by selecting the number that best describes your level of agreement:

Data visualizations are important to increase the efficiency within the product costing process:
strongly agree ❒ agree, neutral ❒ disagree ❒ strongly disagree ❒

In the product costing process, appropriate data visualizations can help to uncover cost savings and help to reduce product costs:
strongly agree ❒ agree, neutral ❒ disagree ❒ strongly disagree ❒

How do you rate the importance of being able to interact with a data visualization (by a visualization that allows you to interact with data to discover more details and explore the dataset via manipulation of the data visualization, for instance by selecting desired data elements, filtering, highlighting or modifying options to change data perspectives)?
very important ❒ important ❒ moderately important ❒ slightly important ❒ not important ❒

What kind of interaction features would be useful to you? Please give examples.

Do you use any UI controls (buttons, radio boxes, sliders, checkboxes, zoom buttons or etc.) for interacting with your data visualization? If yes, for which purpose (filtering, zooming, selecting parts of the data, highlighting, sorting, clustering, etc.) and how often do you use them? (Always=1, Often=2, Occasionally=3, Rarely=4, Never=5).

Do you have any other comments or ideas?

Thank you for participating!

I appreciate your contribution to my research !

If you have any comments or questions, please feel free to contact me.

Zana Vosough, Ph.D. Candidate
Research & Innovation I SAP Innovation Center Network
SAP SE, Chemnitzer Str. 48, 01187 Dresden, Germany
Telephone: +49 62 277-45650
Email: zana.vosough@sap.com

List of Figures

List of Tables

References

[Abel, 2018] Abel, G. J. (2018). Estimates of global bilateral migration flows by gender between 1960 and 20151. *International Migration Review*, 52(3):809–852.

[Aerts et al., 2003] Aerts, J. C., Clarke, K. C., and Keuper, A. D. (2003). Testing popular visualization techniques for representing model uncertainty. *Cartography and Geographic Information Science*, 30(3):249–261.

[Ahl and Allen, 1996] Ahl, V. and Allen, T. (1996). *Hierarchy Theory: A Vision, Vocabulary, and Epistomology*. Columbia University Press.

[Aigner et al., 2007] Aigner, W., Miksch, S., Müller, W., Schumann, H., and Tominski, C. (2007). Visualizing time-oriented data—a systematic view. *Computers & Graphics*, 31(3):401–409.

[Aigner et al., 2011] Aigner, W., Miksch, S., Schumann, H., and Tominski, C. (2011). *Visualization of time-oriented data*. Springer Science & Business Media.

[Alsallakh et al., 2012] Alsallakh, B., Aigner, W., Miksch, S., and Gröller, E. (2012). Reinventing the Contingency Wheel: Scalable visual analytics of large categorical data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2849–2858.

[Alsallakh et al., 2014] Alsallakh, B., Micallef, L., Aigner, W., Hauser, H., Miksch, S., and Rodgers, P. (2014). Visualizing sets and set-typed data: State-of-the-art and future challenges. In *Eurographics conference on Visualization (EuroVis)–State of The Art Reports*, pages 1–21.

[Alsallakh et al., 2016] Alsallakh, B., Micallef, L., Aigner, W., Hauser, H., Miksch, S., and Rodgers, P. (2016). The state-of-the-art of set visualization. *Computer Graphics Forum*, 35(1):234–260.

[Amar and Stasko, 2004] Amar, R. and Stasko, J. (2004). Best paper: A knowledge task-based framework for design and evaluation of information visualizations. In *IEEE Symposium on Information Visualization*, pages 143–150. IEEE.

[Amenta and Klingner, 2002] Amenta, N. and Klingner, J. (2002). Case study: Visualizing sets of evolutionary trees. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 71–74. IEEE.

[Andrews, 1972] Andrews, D. F. (1972). Plots of high-dimensional data. *Biometrics*, pages 125–136.

[Andrews et al., 2009] Andrews, K., Wohlfahrt, M., and Wurzinger, G. (2009). Visual graph comparison. In *Information Visualisation, 2009 13th International Conference*, pages 62–67. IEEE.

[Andrienko et al., 2011] Andrienko, G., Andrienko, N., Bak, P., Keim, D., Kisilevich, S., and Wrobel, S. (2011). A conceptual framework and taxonomy of techniques for analyzing movement. *Journal of Visual Languages & Computing*, 22(3):213–232.

[Andrienko and Andrienko, 2006] Andrienko, N. and Andrienko, G. (2006). *Exploratory analysis of spatial and temporal data: a systematic approach*. Springer Science & Business Media.

[Andrienko and Andrienko, 2013] Andrienko, N. and Andrienko, G. (2013). Visual analytics of movement: An overview of methods, tools and procedures. *Information Visualization*, 12(1):3–24.

[Antoine et al., 2010] Antoine, L., David, A., and Melançon, G. (2010). Living flows: enhanced exploration of edge-bundled graphs based on gpu-intensive edge rendering. In *Information Visualisation (IV), 2010 14th International Conference*, pages 523–530. IEEE.

[Argyriou et al., 2014] Argyriou, E. N., Symvonis, A., and Vassiliou, V. (2014). A fraud detection visualization system utilizing radial drawings and heat-maps. In *Proc. of the International Conference on Information Visualization Theory and Applications (IVAPP'14)*, pages 153–160. IEEE.

[Arleo et al., 2017] Arleo, A., Didimo, W., Liotta, G., and Montecchiani, F. (2017). GiViP: A visual profiler for distributed graph processing systems. In *Proc. of the International Symposium on Graph Drawing and Network Visualization (GD'17)*. Springer. to appear.

[Asiedu and Gu, 1998] Asiedu, Y. and Gu, P. (1998). Product life cycle cost analysis: State of the art review. *International Journal of Production Research*, 36(4):883–908.

[Band and White, 2006] Band, Z. and White, R. W. (2006). PygmyBrowse: A small screen tree browser. In *Extended abstracts of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'06)*, pages 514–519. ACM.

[Batty, 2006] Batty, M. (2006). Rank clocks. *Nature*, 444(7119):592.

[Beck et al., 2014] Beck, F., Burch, M., Diehl, S., and Weiskopf, D. (2014). The state of the art in visualizing dynamic graphs. *EuroVis STAR*, 2.

[Becker and Cleveland, 1987] Becker, R. A. and Cleveland, W. S. (1987). Brushing scatterplots. *Technometrics*, 29(2):127–142.

[Bendix et al., 2005] Bendix, F., Kosara, R., and Hauser, H. (2005). Parallel Sets: Visual analysis of categorical data. In *Proc. of the IEEE Symposium on Information Visualization (InfoVis'05)*, pages 133–140. IEEE.

[Bertin, 1983] Bertin, J. (1983). Semiology of graphics: diagrams, networks, maps.

[Beygelzimer et al., 2001] Beygelzimer, A., Perng, C.-S., and Ma, S. (2001). Fast ordering of large categorical datasets for better visualization. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*, pages 239–244. ACM.

[Bisantz et al., 2002] Bisantz, A. M., Kesevadas, T., Scott, P., Lee, D., Basapur, S., Bhide, P., Bhide, P., and Bhide, P. (2002). Holistic battlespace visualization: advanced concepts in information visualization and cognitive studies. *U. Buffalo*.

[Blasius and Greenacre, 1998] Blasius, J. and Greenacre, M., editors (1998). *Visualization of Categorical Data*. Academic Press.

[Blenkinsop et al., 2000] Blenkinsop, S., Fisher, P., Bastin, L., and Wood, J. (2000). Evaluating the perception of uncertainty in alternative visualization strategies. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 37(1):1–14.

[Boren and Ramey, 2000] Boren, M. T. and Ramey, J. (2000). Thinking aloud: reconciling theory and practice. *IEEE Transactions on Professional Communication*, 43(3):261–278.

[Borgo et al., 2013] Borgo, R., Kehrer, J., Chung, D. H., Maguire, E., Laramee, R. S., Hauser, H., Ward, M., and Chen, M. (2013). Glyph-based visualization: Foundations, design guidelines, techniques and applications. In *Eurographics (STARs)*, pages 39–63.

[Bose and Mahapatra, 2001] Bose, I. and Mahapatra, R. K. (2001). Business data mining—a machine learning perspective. *Information & management*, 39(3):211–225.

[Bostock et al., 2011] Bostock, M., Ogievetsky, V., and Heer, J. (2011). D3: Data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309.

[Boukhelifa et al., 2012] Boukhelifa, N., Bezerianos, A., Isenberg, T., and Fekete, J.-D. (2012). Evaluating sketchiness as a visual variable for the depiction of qualitative uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2769–2778.

[Brehmer and Munzner, 2013] Brehmer, M. and Munzner, T. (2013). A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385.

[Brinton, 1939] Brinton, W. C. (1939). *Graphic Presentation*. Brinton Associates.

[Brown, 2004] Brown, R. (2004). Animated visual vibrations as an uncertainty visualisation technique. In *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 84–89. ACM.

[Bruls et al., 2000] Bruls, M., Huizing, K., and van Wijk, J. J. (2000). Squarified treemaps. In de Leeuw, W. and van Liere, R., editors, *Proceedings of the Joint Eurographics IEEE TCVG Symposium on Visualization*, pages 33–42. Eurographics Association.

[Buja et al., 1996] Buja, A., Cook, D., and Swayne, D. F. (1996). Interactive high-dimensional data visualization. *Journal of computational and graphical statistics*, 5(1):78–99.

[Bureau of Labour Statistics, 2018] Bureau of Labour Statistics, (BLS). (2018). *Standard Occupational Classification Manual*. US Office of Employment and Unemployment Statistics.

[Buttenfield and Ganter, 1990] Buttenfield, B. P. and Ganter, J. H. (1990). Visualization and gis: What should we see? what might we miss. In *Proceedings of the 4th International Symposium on Spatial Data Handling*, volume 1, pages 307–316.

[Byron and Wattenberg, 2008] Byron, L. and Wattenberg, M. (2008). Stacked graphs – geometry & aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1245–1252.

[Candan et al., 2012] Candan, K. S., Di Caro, L., and Sapino, M. L. (2012). PhC: Multiresolution visualization and exploration of text corpora with parallel hierarchical coordinates. *ACM Transactions on Intelligent Systems and Technology*, 3(2):22:1–22:36.

[Card et al., 1999] Card, S., Mackinlay, J., and Shneiderman, B. (1999). Readings in information visualization: using vision to think. 1999. *San Francisco: Morgan Kauffman*.

[Card and Mackinlay, 1997] Card, S. K. and Mackinlay, J. (1997). The structure of the information visualization design space. In *Information Visualization, 1997. Proceedings., IEEE Symposium on*, pages 92–99. IEEE.

[Caulkins et al., 2007] Caulkins, J. P., Morrison, E. L., and Weidemann, T. (2007). Spreadsheet errors and decision making: evidence from field interviews. *Journal of Organizational and End User Computing (JOEUC)*, 19(3):1–23.

[Cedilnik and Rheingans, 2000] Cedilnik, A. and Rheingans, P. (2000). Procedural annotation of uncertain information. In *Visualization 2000. Proceedings*, pages 77–84. IEEE.

[Cesario et al., 2011] Cesario, N., Pang, A., and Singh, L. (2011). Visualizing node attribute uncertainty in graphs. In *IS&T/SPIE Electronic Imaging*, pages 78680H–78680H. International Society for Optics and Photonics.

[Chambers, 2017] Chambers, J. M. (2017). *Graphical Methods for Data Analysis: 0*. Chapman and Hall/CRC.

[Chan and Mitchell, 2003] Chan, L. M. and Mitchell, J. S. (2003). *Dewey Decimal Classification: Principles and Application*. OCLC, 3rd edition.

[Chen, 2006] Chen, C. (2006). *Information visualization: Beyond the horizon*. Springer Science & Business Media.

[Chen and Floridi, 2013] Chen, M. and Floridi, L. (2013). An analysis of information visualisation. *Synthese*, 190(16):3421–3438.

[Chuah and Roth, 1996] Chuah, M. C. and Roth, S. F. (1996). On the semantics of interactive visualizations. In *Information Visualization'96, Proceedings IEEE Symposium on*, pages 29–36. IEEE.

[Claessen and Van Wijk, 2011] Claessen, J. H. and Van Wijk, J. J. (2011). Flexible linked axes for multivariate data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2310–2316.

[Cleveland and McGill, 1984] Cleveland, W. S. and McGill, R. (1984). Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American statistical association*, 79(387):531–554.

[Cockburn et al., 2009] Cockburn, A., Karlson, A., and Bederson, B. B. (2009). A review of overview+ detail, zooming, and focus+ context interfaces. *ACM Computing Surveys (CSUR)*, 41(1):2.

[Cockburn and McKenzie, 2004] Cockburn, A. and McKenzie, B. (2004). Evaluating spatial memory in two and three dimensions. *International Journal of Human-Computer Studies*, 61(3):359–373.

[Collins et al., 2007] Collins, C., Carpendale, M. S. T., and Penn, G. (2007). Visualization of uncertainty in lattices to support decision-making. In *EuroVis*, pages 51–58.

[Collins et al., 2009] Collins, C., Viégas, F. B., and Wattenberg, M. (2009). Parallel tag clouds to explore and analyze faceted text corpora. In *Proc. of the IEEE Symposium on Visual Analytics Science and Technology (VAST'09)*, pages 91–98. IEEE.

[Cook and Thomas, 2005] Cook, K. A. and Thomas, J. J. (2005). Illuminating the path: The research and development agenda for visual analytics. Technical report, Pacific Northwest National Lab.(PNNL), Richland, WA (United States).

[Correll and Gleicher, 2014] Correll, M. and Gleicher, M. (2014). Error bars considered harmful: Exploring alternate encodings for mean and error. *IEEE transactions on visualization and computer graphics*, 20(12):2142–2151.

[Correll et al., 2018] Correll, M., Moritz, D., and Heer, J. (2018). Value-suppressing uncertainty palettes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 642. ACM.

[Cox, 2016] Cox, N. J. (2016). Speaking Stata: Spineplots and their kin. *The Stata Journal*, 8(1):105–121.

[Craft and Cairns, 2005] Craft, B. and Cairns, P. (2005). Beyond guidelines: what can we learn from the visual information seeking mantra? In *Ninth International Conference on Information Visualisation (IV'05)*, pages 110–118. IEEE.

[Crisan et al., 2016] Crisan, A., Gardy, J. L., and Munzner, T. (2016). On regulatory and organizational constraints in visualization design and evaluation. In Sedlmair, M., Isenberg, P., Isenberg, T., Mahyar, N., and Lam, H., editors, *Proceedings of the BELIV Workshop: Beyond Time and Errors – Novel Evaluation Methods for Visualization*, pages 1–9. ACM.

[De Oliveira and Levkowitz, 2003] De Oliveira, M. F. and Levkowitz, H. (2003). From visual data exploration to visual data mining: a survey. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):378–394.

[Deming, 1990] Deming, W. E. (1990). *Sample Design in Business Research*. John Wiley & Sons.

[DeVellis, 2012] DeVellis, R. F. (2012). *Scale development: Theory and applications*. Sage, 3rd edition.

[Dix, 2009] Dix, A. (2009). Human-computer interaction. In *Encyclopedia of database systems*, pages 1327–1331. Springer.

[Draper et al., 2009] Draper, G. M., Livnat, Y., and Riesenfeld, R. F. (2009). A survey of radial methods for information visualization. *IEEE transactions on visualization and computer graphics*, 15(5):759–776.

[Dwyer et al., 2005] Dwyer, T., Marriott, K., and Stuckey, P. J. (2005). Fast node overlap removal. In *International Symposium on Graph Drawing*, pages 153–164. Springer.

[Eades and Whitesides, 1994] Eades, P. and Whitesides, S. (1994). Drawing graph in two layers. *Theoretical Computer Science*, 131(2):361–374.

[Ehlschlaeger et al., 1997] Ehlschlaeger, C. R., Shortridge, A. M., and Goodchild, M. F. (1997). Visualizing spatial data uncertainty using animation. *Computers & Geosciences*, 23(4):387–395.

[Eichelberger, 2003] Eichelberger, H. (2003). Nice class diagrams admit good design? In *Proceedings of the 2003 ACM symposium on Software visualization*, pages 159–ff. ACM.

[Eisen et al., 1998] Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868.

[Ellis and Dix, 2006] Ellis, G. and Dix, A. (2006). Enabling automatic clutter reduction in parallel coordinate plots. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):717–724.

[Ellis and Mansmann, 2010] Ellis, G. and Mansmann, F. (2010). Mastering the information age solving problems with visual analytics. In *Eurographics*, volume 2, page 5.

[Elmqvist and Fekete, 2010] Elmqvist, N. and Fekete, J.-D. (2010). Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):439–454.

[Fekete and Plaisant, 2002] Fekete, J.-D. and Plaisant, C. (2002). Interactive information visualization of a million items. In Wong, P. C. and Andrews, K., editors, *Proceedings of the IEEE Symposium on Information Visualization*, pages 117–124. IEEE.

[Fernstad and Johansson, 2011] Fernstad, S. J. and Johansson, J. (2011). A task based performance evaluation of visualization approaches for categorical data analysis. In *Proc. of the International Conference on Information Visualisation (IV'11)*, pages 80–89. IEEE.

[Few, 2004] Few, S. (2004). *Show me the numbers*. Analytics Pres.

[Fluit et al., 2006] Fluit, C., Sabou, M., and Van Harmelen, F. (2006). Ontology-based information visualization: toward semantic web applications. In *Visualizing the semantic web*, pages 45–58. Springer.

[Freiler et al., 2008] Freiler, W., Matković, K., and Hauser, H. (2008). Interactive visual analysis of set-typed data. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1340–1347.

[Friendly, 2000] Friendly, M. (2000). *Visualizing Categorical Data*. SAS Institute.

[Fua et al., 1999] Fua, Y.-H., Ward, M. O., and Rundensteiner, E. A. (1999). Hierarchical parallel coordinates for exploration of large datasets. In *Proc. of the IEEE Conference on Visualization (Vis'99)*, pages 43–50. IEEE.

[Furnas and Zacks, 1994] Furnas, G. W. and Zacks, J. (1994). Multitrees: Enriching and reusing hierarchical structure. In *Proc. of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'94)*, pages 330–336. ACM.

[Gad et al., 2015] Gad, S., Javed, W., Ghani, S., Elmqvist, N., Ewing, T., Hampton, K. N., and Ramakrishnan, N. (2015). ThemeDelta: Dynamic segmentations over temporal topic models. *IEEE Transactions on Visualization and Computer Graphics*, 21(5):672–685.

[Gershon, 1998] Gershon, N. (1998). Visualization of an imperfect world. *IEEE Computer Graphics and Applications*, 18(4):43–45.

[Ghani et al., 2013] Ghani, S., Kwon, B. C., Lee, S., Yi, J. S., and Elmqvist, N. (2013). Visual analytics for multimodal social network analysis: A design study with social scientists. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2032–2041.

[Gleicher et al., 2011] Gleicher, M., Albers, D., Walker, R., Jusufi, I., Hansen, C. D., and Roberts, J. C. (2011). Visual comparison for information visualization. *Information Visualization*, 10(4):289–309.

[Goguen, 1993] Goguen, J. A. (1993). Social issues in requirements engineering. In *Proceedings of the IEEE International Symposium on Requirements Engineering*, pages 194–195. IEEE.

[Goguen and Linde, 1993] Goguen, J. A. and Linde, C. (1993). Techniques for requirements elicitation. In *Proceedings of the IEEE International Symposium on Requirements Engineering*, pages 152–164. IEEE.

[Görtler et al., 2018] Görtler, J., Schulz, C., Weiskopf, D., and Deussen, O. (2018). Bubble treemaps for uncertainty visualization. *IEEE transactions on visualization and computer graphics*, 24(1):719–728.

[Graham and Kennedy, 2007] Graham, M. and Kennedy, J. (2007). Exploring multiple trees through dag representations. *IEEE transactions on visualization and computer graphics*, 13(6):1294–1301.

[Graham and Kennedy, 2010] Graham, M. and Kennedy, J. (2010). A survey of multiple tree visualisation. *Information Visualization*, 9(4):235–252.

[Griethe et al., 2006] Griethe, H., Schumann, H., et al. (2006). The visualization of uncertain data: Methods and problems. In *SimVis*, pages 143–156.

[Groh, 2017] Groh, R. (2017). *An Iconography of Interaction*. TUDpress, 1st edition.

[Gschwandtnei et al., 2016] Gschwandtnei, T., Bögl, M., Federico, P., and Miksch, S. (2016). Visual encodings of temporal uncertainty: A comparative user study. *IEEE transactions on visualization and computer graphics*, 22(1):539–548.

[Guchev et al., 2012] Guchev, V., Mecella, M., and Santucci, G. (2012). Design guidelines for correlated quantitative data visualizations. In *Proc. of the International Working Conference on Advanced Visual Interfaces (AVI'12)*, pages 761–764. ACM.

[Guerra-Gómez et al., 2013] Guerra-Gómez, J. A., Buck-Coleman, A., Pack, M. L., Plaisant, C., and Shneiderman, B. (2013). Treeversity: interactive visualizations for comparing hierarchical data sets. *Transportation Research Record*, 2392(1):48–58.

[Guha et al., 2000] Guha, S., Rastogi, R., and Shim, K. (2000). Rock: A robust clustering algorithm for categorical attributes. *Information Systems*, 25(5):345–366.

[Guo, 2009] Guo, D. (2009). Flow mapping and multivariate visualization of large spatial interaction data. *IEEE Transactions on Visualization and Computer Graphics*, 15(6).

[Hadnagy, 2010] Hadnagy, C. (2010). *Social Engineering: The Art of Human Hacking*. John Wiley & Sons.

[Harrower and Brewer, 2003] Harrower, M. A. and Brewer, C. A. (2003). ColorBrewer.org: An online tool for selecting color schemes for maps. *The Cartographic Journal*, 40(1):27–37.

[Havre et al., 2000] Havre, S., Hetzler, B., and Nowell, L. (2000). Themeriver: Visualizing theme changes over time. In *Information visualization, 2000. InfoVis 2000. IEEE symposium on*, pages 115–123. IEEE.

[Hearst, 1999] Hearst, M. A. (1999). User interfaces and visualization. *Modern information retrieval*, pages 257–323.

[Heer and Bostock, 2010] Heer, J. and Bostock, M. (2010). Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 203–212. ACM.

[Heer and Shneiderman, 2012] Heer, J. and Shneiderman, B. (2012). Interactive dynamics for visual analysis. *Queue*, 10(2):30.

[Heinrich and Broeksema, 2015] Heinrich, J. and Broeksema, B. (2015). Big data visual analytics with parallel coordinates. In *Proc. of Big Data Visual Analytics (BDVA'15)*, pages 121–122. IEEE.

[Heinrich and Weiskopf, 2013] Heinrich, J. and Weiskopf, D. (2013). State of the art of parallel coordinates. In *Eurographics 2013 - State of the Art Reports*, pages 95–116. Eurographics.

[Henry et al., 2007] Henry, N., Fekete, J.-D., and McGuffin, M. J. (2007). Nodetrix: a hybrid visualization of social networks. *IEEE transactions on visualization and computer graphics*, 13(6):1302–1309.

[Hesse, 2015] Hesse, S. (2015). *Struktur und Gestaltung von Informationsvisualisierungen zur Entscheidungsunterstützung*. phdthesis, Technischen Universität Dresden.

[Hofmann and Vendettuoli, 2013] Hofmann, H. and Vendettuoli, M. (2013). Common angle plots as perception-true visualizations of categorical associations. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2297–2305.

[Hogräfer, 2018] Hogräfer, M. (2018). Towards a toolkit for extensible, model-driven and interactive information visualizations. Master's thesis, Technische Universität Dresden.

[Holten, 2006a] Holten, D. (2006a). Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748.

[Holten, 2006b] Holten, D. (2006b). Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on visualization and computer graphics*, 12(5):741–748.

[Holten and Van Wijk, 2008] Holten, D. and Van Wijk, J. J. (2008). Visual comparison of hierarchically organized data. In *Computer Graphics Forum*, volume 27, pages 759–766.

[Holten and Van Wijk, 2009] Holten, D. and Van Wijk, J. J. (2009). Force-directed edge bundling for graph visualization. In *Computer graphics forum*, volume 28, pages 983–990. Wiley Online Library.

[Horak et al., 2019] Horak, T., Mathisen, A., Klokmose, C. N., Dachselt, R., and Elmqvist, N. (2019). Vistribute: Distributing interactive visualizations in dynamic multi-device setups. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, page 616. ACM.

[Huang et al., 2016] Huang, M. L., Huang, T.-H., and Zhang, X. (2016). A novel virtual node approach for interactive visual analytics of big datasets in parallel coordinates. *Future Generation Computer Systems*, 55:510–523.

[Hube et al., 2017] Hube, N., Müller, M., and Groh, R. (2017). Additional on-demand dimension for data visualization. In *Proceedings of the Eurographics/IEEE VGTC Conference on Visualization: Short Papers*, EuroVis '17, pages 163–167, Goslar Germany, Germany. Eurographics Association.

[Hunter, 1999] Hunter, G. J. (1999). New tools for handling spatial data quality: moving from academic concepts to practical reality. *URISA Journal*, 11(2):25–34.

[Hunter and Goodchild, 1993] Hunter, G. J. and Goodchild, M. (1993). Managing uncertainty in spatial databases: Putting theory into practice. In *Papers from the Annual Conference-Urban and Regional Information Systems Association*, pages 15–15. URISA URBAN AND REGIONAL INFORMATION SYSTEMS.

[Inselberg, 1985] Inselberg, A. (1985). The plane with parallel coordinates. *The visual computer*, 1(2):69–91.

[Inselberg, 2009] Inselberg, A. (2009). *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer.

[Jackson, 2008] Jackson, C. H. (2008). Displaying uncertainty with shading. *The American Statistician*, 62(4):340–347.

[Johansson and Forsell, 2016] Johansson, J. and Forsell, C. (2016). Evaluation of parallel coordinates: Overview, categorization and guidelines for future research. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):579–588.

[Johansson et al., 2007] Johansson, J., Ljung, P., and Cooper, M. (2007). Depth cues and density in temporal parallel coordinates. In *Proc. of the Eurographics/ IEEE-VGTC Symposium on Visualization (EuroVis'07)*, pages 35–42. Eurographics Association.

[Johansson et al., 2008] Johansson, S., Jern, M., and Johansson, J. (2008). Interactive quantification of categorical variables in mixed data sets. In *Proc. of the International Conference Information Visualisation, (IV'08*, pages 3–10. IEEE.

[Johansson and Johansson, 2009] Johansson, S. and Johansson, J. (2009). Visual analysis of mixed data sets using interactive quantification. *SIGKDD Explorations Newsletter*, 11(2):29–38.

[Johnson and Shneiderman, 1991] Johnson, B. and Shneiderman, B. (1991). Tree-Maps: A space-filling approach to the visualization of hierarchical information structures. In Nielson, G. M. and Rosenblum, L., editors, *Proceedings of the IEEE Conference on Visualization*, pages 284–291. IEEE.

[Johnson et al., 2006] Johnson, C., Moorhead, R., Munzner, T., Pfister, H., Rheingans, P., and Yoo, T. S. (2006). Nih/nsf visualization research challenges report. In *Los Alamitos, Ca: IEEE Computing Society*. Citeseer.

[Johnson and Sanderson, 2003] Johnson, C. R. and Sanderson, A. R. (2003). A next step: Visualizing errors and uncertainty. *IEEE Computer Graphics and Applications*, 23(5):6–10.

[Jünger and Mutzel, 1997] Jünger, M. and Mutzel, P. (1997). 2-layer straightline crossing minimization: Performance of exact and heuristic algorithms. *Journal of Graph Algorithms and Applications*, 1(1):1–25.

[Kandel et al., 2012] Kandel, S., Paepcke, A., Hellerstein, J. M., and Heer, J. (2012). Enterprise data analysis and visualization: An interview study. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2917–2926.

[Kandogan et al., 2014] Kandogan, E., Balakrishnan, A., Haber, E. M., and Pierce, J. S. (2014). From data to insight: Work practices of analysts in the enterprise. *IEEE Computer Graphics and Applications*, 34(5):42–50.

[Keahey et al., 2018] Keahey, T. A., Rope, D. J., and Wills, G. J. (2018). Generating an outside-in hierarchical tree visualization, Patent Application US 20180046690 A1, filed August 12, 2016, published February 15, 2018.

[Keck et al., 2014] Keck, M., Herrmann, M., Henkens, D., Taranko, S., Nguyen, V., Funke, F., Schattenberg, S., Both, A., and Groh, R. (2014). Visual innovations for product search interfaces. In *44. Jahrestagung der Gesellschaft für Informatik, Informatik 2014, Big Data - Komplexität meistern*, pages 679–688, Stuttgart, Germany. GI-Jahrestagung 2014. 1st International Workshop on Future Search Engines at INFORMATIK 2014.

[Keim, 1997] Keim, D. A. (1997). Visual techniques for exploring databases. In *Knowledge Discovery in Databases (KDD'97)*.

[Keim, 2000] Keim, D. A. (2000). Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Trans. on Visualization and Computer Graphics*, 6(1):59–78.

[Keim, 2005] Keim, D. A. (2005). Information visualization: Scope, techniques and opportunities for geovisualization. In *Exploring geovisualization*, pages 21–52. Elsevier.

[Keim et al., 2006] Keim, D. A., Mansmann, F., Schneidewind, J., and Ziegler, H. (2006). Challenges in visual data analysis. In *Tenth International Conference on Information Visualisation (IV'06)*, pages 9–16. IEEE.

[Kirk, 2016] Kirk, A. (2016). *Data visualisation: a handbook for data driven design*. Sage.

[Kister et al., 2017] Kister, U., Klamka, K., Tominski, C., and Dachselt, R. (2017). Grasp: Combining spatially-aware mobile devices and a display wall for graph visualization and interaction. In *Computer Graphics Forum*, volume 36, pages 503–514. Wiley Online Library.

[Koffka, 2013] Koffka, K. (2013). *Principles of Gestalt psychology*. Routledge.

[Koh et al., 2011] Koh, L. C., Slingsby, A., Dykes, J., and Kam, T. S. (2011). Developing and applying a user-centered model for the design and implementation of information visualization tools. In Banissi, E., Bertschi, S., Burkhard, R., Cvek, U., Eppler, M., Forsell, C., Grinstein, G., Johansson, J., Kenderdine, S., Marchese, F. T., Maple, C., Trutschl, M., Sarfraz, M., Stuart, L., Ursyn, A., and Wyeld, T. G., editors, *Proceedings of the International Conference on Information Visualisation (IV)*, pages 90–95. IEEE.

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

[Kosara et al., 2002] Kosara, R., Miksch, S., and Hauser, H. (2002). Focus+ context taken literally. *IEEE Computer Graphics and Applications*, 22(1):22–29.

[Krum, 2013] Krum, R. (2013). *Cool infographics: Effective communication with data visualization and design*. John Wiley & Sons.

[Kruskal and Landwehr, 1983] Kruskal, J. B. and Landwehr, J. M. (1983). Icicle plots: Better displays for hierarchical clustering. *The American Statistician*, 37(2):162–168.

[Krzywinski et al., 2009] Krzywinski, M., Schein, J., Birol, I., Connors, J., Gascoyne, R., Horsman, D., Jones, S. J., and Marra, M. A. (2009). Circos: an information aesthetic for comparative genomics. *Genome research*, 19(9):1639–1645.

[Kulyk et al., 2006] Kulyk, O., Kosara, R., Urquiza, J., and Wassink, I. (2006). Human-centered aspects. In Kerren, A., Ebert, A., and Meyer, J., editors, *Human-Centered Visualization Environments*, pages 13–75. Springer.

[Lallé et al., 2016] Lallé, S., Conati, C., and Carenini, G. (2016). Prediction of individual learning curves across information visualizations. *User Modeling and User-Adapted Interaction*, 26(4):307–345.

[Laugwitz et al., 2008] Laugwitz, B., Held, T., and Schrepp, M. (2008). *Construction and Evaluation of a User Experience Questionnaire*, pages 63–76. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Lee et al., 2006] Lee, B., Plaisant, C., Parr, C. S., Fekete, J.-D., and Henry, N. (2006). Task taxonomy for graph visualization. In *Proc. of AVI Workshop "BEyond Time and Errors: Novel Evaluation Methods for Information Visualization" (BELIV'06)*, pages 1–5. ACM Press.

[Lee et al., 2007] Lee, B., Robertson, G. G., Czerwinski, M., and Parr, C. S. (2007). Candidtree: visualizing structural uncertainty in similar hierarchies. *Information Visualization*, 6(3):233–246.

[Lex et al., 2010] Lex, A., Streit, M., Partl, C., Kashofer, K., and Schmalstieg, D. (2010). Comparative analysis of multidimensional, quantitative data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1027–1035.

[Lex et al., 2012] Lex, A., Streit, M., Schulz, H.-J., Partl, C., Schmalstieg, D., Park, P. J., and Gehlenborg, N. (2012). StratomeX: Visual analysis of large-scale heterogeneous genomics data for cancer subtype characterization. *Computer Graphics Forum*, 31(3):1175–1184.

[Lindlof and Taylor, 2011] Lindlof, T. R. and Taylor, B. C. (2011). *Qualitative Communication Research Methods*. SAGE Publications, 3rd edition.

[Liu et al., 2013] Liu, S., Wu, Y., Wei, E., Liu, M., and Liu, Y. (2013). StoryFlow: Tracking the evolution of stories. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2436–2445.

[Lodha et al., 1996] Lodha, S. K., Pang, A., Sheehan, R. E., and Wittenbrink, C. M. (1996). Uflow: Visualizing uncertainty in fluid flow. In *Proceedings of the 7th conference on Visualization'96*, pages 249–ff. IEEE Computer Society Press.

[Ma and Hellerstein, 1999] Ma, S. and Hellerstein, J. F. (1999). Ordering categorical data to improve visualization. In *Late Breaking Hot Topics of the IEEE Information Visualization Symposium (InfoVis'99)*, pages 15–18. IEEE.

[MacEachren, 1992] MacEachren, A. M. (1992). Visualizing uncertain information. *Cartographic Perspectives*, (13):10–19.

[MacEachren et al., 1998] MacEachren, A. M., Brewer, C. A., and Pickle, L. W. (1998). Visualizing georeferenced data: representing reliability of health statistics. *Environment and planning A*, 30(9):1547–1561.

[MacEachren et al., 2005] MacEachren, A. M., Robinson, A., Hopper, S., Gardner, S., Murray, R., Gahegan, M., and Hetzler, E. (2005). Visualizing geospatial information uncertainty: What we know and what we need to know. *Cartography and Geographic Information Science*, 32(3):139–160.

[Mackinlay, 1986] Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. *Acm Transactions On Graphics (Tog)*, 5(2):110–141.

[Mäkinen, 1990] Mäkinen, E. (1990). Experiments on drawing 2-level hierarchical graphs. *International Journal of Computer Mathematics*, 37(3-4):129–135.

[Marshall and Meckling, 1962] Marshall, A. W. and Meckling, W. H. (1962). Predictability of the costs, time, and success of development. In *The rate and direction of inventive activity: Economic and social factors*, pages 461–476. Princeton University Press.

[Mazza, 2009] Mazza, R. (2009). *Introduction to information visualization*. Springer Science & Business Media.

[McGuffin and Robert, 2010] McGuffin, M. J. and Robert, J.-M. (2010). Quantifying the space-efficiency of 2d graphical representations of trees. *Information Visualization*, 9(2):115–140.

[McKenna et al., 2014] McKenna, S., Mazur, D., Agutter, J., and Meyer, M. (2014). Design activity framework for visualization design. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2191–2200.

[Meirelles, 2013] Meirelles, I. (2013). *Design for information: an introduction to the histories, theories, and best practices behind effective information visualizations*. Rockport publishers.

[Meyer et al., 2008] Meyer, D., Zeileis, A., and Hornik, K. (2008). Visualizing contingency tables. In Chen, C., Härdle, W., and Unwin, A., editors, *Handbook of Data Visualization*, pages 617–642. Springer.

[Meyer et al., 2009] Meyer, M., Munzner, T., and Pfister, H. (2009). Mizbee: a multiscale synteny browser. *IEEE transactions on visualization and computer graphics*, 15(6):897–904.

[Miksch and Aigner, 2014] Miksch, S. and Aigner, W. (2014). A matter of time: Applying a data–users–tasks design triangle to visual analytics of time-oriented data. *Computers & Graphics*, 38:286–290.

[Miller, 1956] Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81.

[Moere and Purchase, 2011] Moere, A. V. and Purchase, H. (2011). On the role of design in information visualization. *Information Visualization*, 10(4):356–371.

[Monroe et al., 2013] Monroe, M., Lan, R., Lee, H., Plaisant, C., and Shneiderman, B. (2013). Temporal event sequence simplification. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2227–2236.

[Morgan, 1996] Morgan, D. L. (1996). *Focus groups as qualitative research*, volume 16. Sage publications.

[Moritz et al., 2019] Moritz, D., Wang, C., Nelson, G. L., Lin, H., Smith, A. M., Howe, B., and Heer, J. (2019). Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. *IEEE transactions on visualization and computer graphics*, 25(1):438–448.

[Morris, 2006] Morris, G. L. (2006). *Candlestick Charting Explained: Timeless Techniques for Trading Stocks and Futures: Timeless Techniques for Trading stocks and Sutures*. McGraw Hill Professional.

[Munzner, 2009] Munzner, T. (2009). A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928.

[Munzner, 2014] Munzner, T. (2014). *Visualization analysis and design*. AK Peters/CRC Press.

[Munzner et al., 2003] Munzner, T., Guimbretière, F., Tasiran, S., Zhang, L., and Zhou, Y. (2003). Treejuxtaposer: scalable tree comparison using focus+ context with guaranteed visibility. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 453–462. ACM.

[Nadav-Greenberg and Joslyn, 2009] Nadav-Greenberg, L. and Joslyn, S. L. (2009). Uncertainty forecasts improve decision making among nonexperts. *Journal of Cognitive Engineering and Decision Making*, 3(3):209–227.

[Ngo et al., 2003] Ngo, D. C. L., Teo, L. S., and Byrne, J. G. (2003). Modelling interface aesthetics. *Information Sciences*, 152:25–46.

[Nielsen and Grønbæk, 2015] Nielsen, M. and Grønbæk, K. (2015). PivotViz: Interactive visual analysis of multidimensional library transaction data. In *Proc. of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'15)*, pages 139–142. ACM.

[Olston and Mackinlay, 2002] Olston, C. and Mackinlay, J. D. (2002). Visualizing data with bounded uncertainty. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 37–40. IEEE.

[Palmas et al., 2014] Palmas, G., Bachynskyi, M., Oulasvirta, A., Seidel, H. P., and Weinkauf, T. (2014). An edge-bundling layout for interactive parallel coordinates. In *Proc. of the IEEE Pacific Visualization Symposium (PacificVis'14)*, pages 57–64. IEEE.

[Palmer and Ghose, 2008] Palmer, S. E. and Ghose, T. (2008). Extremal edge: A powerful cue to depth perception and figure-ground organization. *Psychological science*, 19(1):77–83.

[Pang et al., 1997] Pang, A. T., Wittenbrink, C. M., and Lodha, S. K. (1997). Approaches to uncertainty visualization. *The Visual Computer*, 13(8):370–390.

[Paul et al., 2015] Paul, C. L., Rohrer, R., and Nebesh, B. (2015). A “design first” approach to visualization innovation. *IEEE Computer Graphics and Applications*, 35(1):12–18.

[Perin et al., 2016] Perin, C., Boy, J., and Vernier, F. (2016). Using gap charts to visualize the temporal evolution of ranks and scores. *IEEE Computer Graphics and Applications*, 36(5):38–49.

[Pettersson, 2002] Pettersson, R. (2002). *Information design: An introduction*, volume 3. John Benjamins Publishing.

[Pettersson, 2010] Pettersson, R. (2010). Information design–principles and guidelines. *Journal of Visual Literacy*, 29(2):167–182.

[Phan et al., 2005] Phan, D., Xiao, L., Yeh, R., Hanrahan, P., and Winograd, T. (2005). Flow map layout.

[Pike et al., 2009] Pike, W. A., Stasko, J., Chang, R., and O'connell, T. A. (2009). The science of interaction. *Information Visualization*, 8(4):263–274.

[Plaisant et al., 2002] Plaisant, C., Grosjean, J., and Bederson, B. B. (2002). SpaceTree: Supporting exploration in large node link tree, design evolution and empirical evaluation. In *Proc. of the IEEE Symposium on Information Visualization (InfoVis'02)*, pages 57–64. IEEE.

[Plattner et al., 2010] Plattner, H., Meinel, C., and Leifer, L. (2010). *Design thinking: understand–improve–apply*. Springer Science & Business Media.

[Pohl, 2010] Pohl, K. (2010). *Requirements engineering: fundamentals, principles, and techniques*. Springer Publishing Company, Incorporated.

[Procter et al., 2010] Procter, J. B., Thompson, J., Letunic, I., Creevey, C., Jossinet, F., and Barton, G. J. (2010). Visualization of multiple alignments, phylogenies and gene family evolution. *Nature methods*, 7:S16–S25.

[Pumain, 2006] Pumain, D., editor (2006). *Hierarchy in Natural and Social Sciences*. Springer.

[Purchase, 1997] Purchase, H. (1997). Which aesthetic has the greatest effect on human understanding? In *International Symposium on Graph Drawing*, pages 248–261. Springer.

[Richer et al., 2018] Richer, G., Sansen, J., Lalanne, F., Auber, D., and Bourqui, R. (2018). Enabling hierarchical exploration for large-scale multidimensional data with abstract parallel coordinates. In *Proc. of the Workshops of the EDBT/ICDT 2018 Joint Conference*, pages 76–83. CEUR-WS.

[Riehmann et al., 2005] Riehmann, P., Hanfler, M., and Froehlich, B. (2005). Interactive Sankey diagrams. In Stasko, J. and Ward, M. O., editors, *Proceedings of the IEEE Symposium on Information Visualization*, pages 233–240. IEEE.

[Roberts, 2004] Roberts, J. C. (2004). Exploratory visualization with multiple linked views.

[Roberts et al., 2016] Roberts, J. C., Headleand, C., and Ritsos, P. D. (2016). Sketching designs using the five design-sheet methodology. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):419–428.

[Robertson et al., 2002] Robertson, G., Cameron, K., Czerwinski, M., and Robbins, D. (2002). Polyarchy visualization: Visualizing multiple intersecting hierarchies. In *Proc. of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'02)*, pages 423–430. ACM.

[Rosario et al., 2004] Rosario, G. E., Rundensteiner, E. A., Brown, D. C., Ward, M. O., and Huang, S. (2004). Mapping nominal values to numbers for effective visualization. *Information Visualization*, 3(2):80–95.

[Rosenholtz et al., 2005] Rosenholtz, R., Li, Y., Mansfield, J., and Jin, Z. (2005). Feature congestion: a measure of display clutter. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 761–770. ACM.

[Royer et al., 2008] Royer, L., Reimann, M., Andreopoulos, B., and Schroeder, M. (2008). Unraveling protein networks with power graph analysis. *PLoS computational biology*, 4(7):e1000108.

[Rusu, 2013] Rusu, A. (2013). Tree drawing algorithms. In Tamassia, R., editor, *Handbook of Graph Drawing and Visualization*, pages 155–192. CRC Press.

[Sanftmann and Weiskopf, 2012] Sanftmann, H. and Weiskopf, D. (2012). 3d scatterplot navigation. *IEEE Transactions on Visualization and Computer Graphics*, 18(11):1969–1978.

[Sansen et al., 2017] Sansen, J., Richer, G., Jourde, T., Lalanne, F., Auber, D., and Bourqui, R. (2017). Visual exploration of large multidimensional data using parallel coordinates on big data infrastructure. *Informatics*, 4(3):21:1–21:22.

[Sanyal et al., 2010] Sanyal, J., Zhang, S., Dyer, J., Mercer, A., Amburn, P., and Moorhead, R. (2010). Noodles: A tool for visualization of numerical weather model ensemble uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1421–1430.

[Sarikaya et al., 2016] Sarikaya, A., Correli, M., Dinis, J. M., O'Connor, D. H., and Gleicher, M. (2016). Visualizing co-occurrence of events in populations of viral genome sequences. *Computer Graphics Forum*, 35(3):151–160.

[Schmidt, 2008] Schmidt, M. (2008). The sankey diagram in energy and material flow management. *Journal of industrial ecology*, 12(1):82–94.

[Schrepp et al., 2017] Schrepp, M., Hinderks, A., and Thomaschewski, J. (2017). Construction of a benchmark for the user experience questionnaire (UEQ). *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(4):40–44.

[Schulz, 2011] Schulz, H.-J. (2011). Treevis.net: A Tree Visualization Reference. *IEEE Computer Graphics and Applications*, 31(6):11–15.

[Schulz et al., 2013] Schulz, H.-J., Hadlak, S., and Schumann, H. (2013). A visualization approach for cross-level exploration of spatiotemporal data. In *Proc. of the International Conference on Knowledge Management and Knowledge Technologies (i-Know'13)*, pages 2:1–2:8. ACM.

[Schumann and Müller, 2013] Schumann, H. and Müller, W. (2013). *Visualisierung: Grundlagen und allgemeine Methoden*. Springer-Verlag.

[Schwaber, 1997] Schwaber, K. (1997). Scrum development process. In *Business object design and implementation*, pages 117–134. Springer.

[Sedlmair et al., 2011] Sedlmair, M., Isenberg, P., Baur, D., and Butz, A. (2011). Information visualization evaluation in large companies: Challenges, experiences and recommendations. *Information Visualization*, 10(3):248–266.

[Sedlmair et al., 2012] Sedlmair, M., Meyer, M., and Munzner, T. (2012). Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440.

[Shneiderman, 1994] Shneiderman, B. (1994). Dynamic queries for visual information seeking. *IEEE Software*, 11(6):70–77.

[Shneiderman, 1996] Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. of the IEEE Symposium on Visual Languages (VL'96)*, pages 336–343. IEEE.

[Siirtola and Räihä, 2006] Siirtola, H. and Räihä, K.-J. (2006). Interacting with parallel coordinates. *Interacting with Computers*, 18(6):1278–1309.

[Skeels et al., 2010] Skeels, M., Lee, B., Smith, G., and Robertson, G. G. (2010). Revealing uncertainty for information visualization. *Information Visualization*, 9(1):70–81.

[Slingsby and Dykes, 2012] Slingsby, A. and Dykes, J. (2012). Experiences in involving analysts in visualisation design. In Bertini, E., Perer, A., Lam, H., Isenberg, P., and Isenberg, T., editors, *Proceedings of the BELIV Workshop: Beyond Time and Errors – Novel Evaluation Methods for Visualization*. ACM.

[St. John et al., 2001] St. John, M., Cowen, M. B., Smallman, H. S., and Oonk, H. M. (2001). The use of 2d and 3d displays for shape-understanding versus relative-position tasks. *Human Factors*, 43(1):79–98.

[Stasko and Zhang, 2000] Stasko, J. and Zhang, E. (2000). Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Proc. of the IEEE Symposium on Information Visualization (InfoVis'00)*, pages 57–65. IEEE.

[Steiner, 1998] Steiner, A. (1998). *A Generalisation Approach to Temporal Data Models and their Implementations*. PhD thesis, Swiss Federal Institute of Technology.

[Stoffel et al., 2012] Stoffel, F., Janetzko, H., and Mansmann, F. (2012). Proportions in categorical and geographic data: Visualizing the results of political elections. In *Proc. of the International Working Conference on Advanced Visual Interfaces (AVI'12)*, pages 457–464. ACM.

[Streit et al., 2008] Streit, A., Pham, B., and Brown, R. (2008). A spreadsheet approach to facilitate visualization of uncertainty in information. *IEEE transactions on visualization and computer graphics*, 14(1):61–72.

[Strickland, 2012] Strickland, J. (2012). Cosmograph? What's a Cosmograph? *Computer History Museum Volunteer Information Exchange*, 2(16):2–3.

[Sugiyama et al., 1981] Sugiyama, K., Tagawa, S., and Toda, M. (1981). Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125.

[Sutcliffe, 2002] Sutcliffe, A. (2002). *User-Centred Requirements Engineering*. Springer.

[Tak et al., 2014] Tak, S., Toet, A., and van Erp, J. (2014). The perception of visual uncertaintyrepresentation by non-experts. *IEEE transactions on visualization and computer graphics*, 20(6):935–943.

[Taylor and Kuyatt, 1994] Taylor, B. N. and Kuyatt, C. E. (1994). *Guidelines for evaluating and expressing the uncertainty of NIST measurement results*. US Department of Commerce, Technology Administration, National Institute of Standards and Technology Gaithersburg, MD.

[Telea and Auber, 2008] Telea, A. and Auber, D. (2008). Code flows: Visualizing structural evolution of source code. *Computer Graphics Forum*, 27(3):831–838.

[Thomson et al., 2005] Thomson, J., Hetzler, E., MacEachren, A., Gahegan, M., and Pavel, M. (2005). A typology for visualizing uncertainty. In *Electronic Imaging 2005*, pages 146–157. International Society for Optics and Photonics.

[Toker et al., 2013] Toker, D., Conati, C., Steichen, B., and Carenini, G. (2013). Individual user characteristics and information visualization: connecting the dots through eye tracking. In *proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 295–304. ACM.

[Tol, 2012] Tol, P. (2012). Colour schemes. Technical Report SRON/EPS/TN/09-002 v.2.2, SRON Netherlands Institute for Space Research.

[Tominski et al., 2004] Tominski, C., Abello, J., and Schumann, H. (2004). Axes-based visualizations with radial layouts. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 1242–1247. ACM.

[Tominski et al., 2009] Tominski, C., Abello, J., and Schumann, H. (2009). Cgv—an interactive graph visualization system. *Computers & Graphics*, 33(6):660–678.

[Tominski et al., 2005] Tominski, C., Schulze-Wollgast, P., and Schumann, H. (2005). 3d information visualization for time dependent data on maps. In *Information Visualisation, 2005. Proceedings. Ninth International Conference on*, pages 175–181. IEEE.

[Tory and Möller, 2004] Tory, M. and Möller, T. (2004). Human factors in visualization research. *IEEE Transactions on Visualization and Computer Graphics*, 10(1):72–84.

[Treisman and Gelade, 1980] Treisman, A. M. and Gelade, G. (1980). A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136.

[Tu and Shen, 2007] Tu, Y. and Shen, H.-W. (2007). Visualizing changes of hierarchical data using treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1286–1293.

[Tufte, 1983] Tufte, E. R. (1983). *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT.

[Tunkelang, 2009] Tunkelang, D. (2009). *Faceted Search*. Morgan and Claypool Publishers.

[Tutte, 1998] Tutte, W. (1998). *As I Have Known It*. Oxford University Press.

[Tversky et al., 2002] Tversky, B., Morrison, J. B., and Betrancourt, M. (2002). Animation: can it facilitate? *International journal of human-computer studies*, 57(4):247–262.

[Van Welie et al., 2001] Van Welie, M., Van Der Veer, G. C., and Eliëns, A. (2001). Patterns as tools for user interface design. In *Tools for Working with Guidelines*, pages 313–324. Springer.

[van Wijk, 2006] van Wijk, J. J. (2006). Bridging the gaps. *IEEE Computer Graphics and Applications*, 26(6):6–9.

[Van Wijk, 2006] Van Wijk, J. J. (2006). Views on visualization. *IEEE transactions on visualization and computer graphics*, 12(4):421–432.

[VanderPlas and Hofmann, 2015] VanderPlas, S. and Hofmann, H. (2015). Signs of the sine illusion – why we need to care. *Journal of Computational and Graphical Statistics*, 24(4):1170–1190.

[Verma and Pang, 2004] Verma, V. and Pang, A. (2004). Comparative flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 10(6):609–624.

[Vliegen et al., 2006] Vliegen, R., Van Wijk, J. J., and van der Linden, E.-J. (2006). Visualizing business data with generalized treemaps. *IEEE Transactions on visualization and computer graphics*, 12(5):789–796.

[Von Landesberger et al., 2011] Von Landesberger, T., Kuijper, A., Schreck, T., Kohlhammer, J., van Wijk, J. J., Fekete, J.-D., and Fellner, D. W. (2011). Visual analysis of large graphs: state-of-the-art and future research challenges. In *Computer graphics forum*, volume 30, pages 1719–1749. Wiley Online Library.

[Vosough, 2018] Vosough, Z. (2018). Visualizing the results of product costing plausibility checks with parallel hierarchies.

[Vosough et al., 2017a] Vosough, Z., Groh, R., and Schulz, H.-J. (2017a). On establishing visualization requirements: A case study in product costing. In *Short Paper Proc. of the Eurographics Conference on Visualization (EuroVis'17)*, pages 97–101. Eurographics Association.

[Vosough et al., 2018a] Vosough, Z., Hogröfer, M., Royer, L. A., Groh, R., and Schulz, H.-J. (2018a). Parallel hierarchies: A visualization for cross-tabulating hierarchical categories. *Computers & Graphics*.

[Vosough et al., 2017b] Vosough, Z., Kammer, D., Keck, M., and Groh, R. (2017b). Visualizing uncertainty in flow diagrams: A case study in product costing. In *Proceedings of the 10th International Symposium on Visual Information Communication and Interaction*, VINCI '17, pages 1–8, New York, NY, USA. ACM.

[Vosough et al., 2018b] Vosough, Z., Kammer, D., Keck, M., and Groh, R. (2018b). Mirroring sankey diagrams for visual comparison tasks. In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - Volume 3: IVAPP, Funchal, Madeira, Portugal, January 27-29, 2018.*, pages 349–355.

[Vosough et al., 2019] Vosough, Z., Kammer, D., Keck, M., and Groh, R. (2019). Visualization approaches for understanding uncertainty in flow diagrams. *Journal of Computer Languages*, 52:44–54.

[Vosough and Vasyutynskyy, 2018] Vosough, Z. and Vasyutynskyy, V. (2018). Using parallel sets for visualizing results of machine learning based plausibility checks in product costing. In *VisBIA: Visual Interfaces for Big Data Environments in Industrial Applications - Workshop at ACM AVI 2018*, AVI. ACM.

[Vosough et al., 2016] Vosough, Z., Walther, M., Rode, J., Hesse, S., and Groh, R. (2016). Having fun with customers: Lessons learned from an agile development of a business software. In *Stakeholder Involvement in Agile Development - Workshop at ACM NordiCHI 2016*, NordiChi. ACM.

[Vrotsou et al., 2009] Vrotsou, K., Johansson, J., and Cooper, M. (2009). Activitree: Interactive visual exploration of sequences in event-based data using graph similarity. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):945–952.

[Walter et al., 2018] Walter, M., Leyh, C., and Strahringer, S. (2018). Toward early product cost optimization: Requirements for an integrated measure management approach. In *Proceedings of Multiconference Wirtschaftsinformatik 2018 (MKWI2018). Band V: Data driven X — Turning Data into Value.*, pages 2057–2068. Leuphana University Lueneburg.

[Wang et al., 2019] Wang, Y., Wang, Y., Zhang, H., Sun, Y., Fu, C.-W., Sedlmair, M., Chen, B., and Deussen, O. (2019). Structure-aware fisheye views for efficient large graph exploration. *IEEE transactions on visualization and computer graphics*, 25(1):566–575.

[Ward et al., 2010] Ward, M., Grinstein, G., and Keim, D. (2010). *Interactive Data Visualization: Foundations, Techniques, and Applications*. A. K. Peters, Ltd., Natick, MA, USA.

[Ward, 2008] Ward, M. O. (2008). Multivariate data glyphs: Principles and practice. In *Handbook of data visualization*, pages 179–198. Springer.

[Ware, 2000] Ware, C. (2000). Designing with a $21/2$d attitude. *Information Design Journal*, 10(3):258–265.

[Ware, 2013] Ware, C. (2013). *Information Visualization – Perception for Design*. Morgan Kaufmann, 3rd edition.

[Weaver, 2010] Weaver, C. (2010). Cross-filtered views for multidimensional visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):192–204.

[Williams, 2009] Williams, A. (2009). User-centered design, activity-centered design, and goal-directed design: A review of three methods for designing web applications. In Mehlenbacher, B., Protopsaltis, A., Williams, A., and Slattery, S., editors, *Proceedings of the ACM International Conference on Design of Communication (SIGDOC)*, pages 1–8. ACM.

[Winters et al., 2016] Winters, K. M., Lach, D., and Cushing, J. B. (2016). A conceptual model for characterizing the problem domain. *Information Visualization*, 15(6):301–311.

[Witten et al., 2016] Witten, I. H., Frank, E., and Hall, M. A. (2016). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, Amsterdam, 4 edition.

[Wittenbrink et al., 1995] Wittenbrink, C. M., Pang, A. T., and Lodha, S. K. (1995). *Verity visualization: Visual mappings*. Computer Research Laboratory [University of California, Santa Cruz].

[Wittenbrink et al., 1996] Wittenbrink, C. M., Pang, A. T., and Lodha, S. K. (1996). Glyphs for visualizing uncertainty in vector fields. *IEEE transactions on Visualization and Computer Graphics*, 2(3):266–279.

[Wolfe and Gray, 2007] Wolfe, J. M. and Gray, W. (2007). Guided search 4.0. *Integrated models of cognitive systems*, pages 99–119.

[Wongsuphasawat et al., 2011] Wongsuphasawat, K., Guerra Gómez, J. A., Plaisant, C., Wang, T. D., Taieb-Maimon, M., and Shneiderman, B. (2011). Lifeflow: visualizing an overview of event sequences. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1747–1756. ACM.

[World Health Organization, 2016] World Health Organization, (WHO). (2016). *International Statistical Classification of Diseases and Related Health Problems, 10th Revision*. WHO Press, 5th edition.

[Xie et al., 2006] Xie, Z., Huang, S., Ward, M. O., and Rundensteiner, E. A. (2006). Exploratory visualization of multivariate data with variable quality. In *Visual Analytics Science And Technology, 2006 IEEE Symposium On*, pages 183–190. IEEE.

[Yang et al., 2002] Yang, J., Ward, M. O., and Rundensteiner, E. A. (2002). Interring: An interactive tool for visually navigating and manipulating hierarchical structures. In *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.*, pages 77–84. IEEE.

[Yi et al., 2007] Yi, J. S., ah Kang, Y., Stasko, J. T., Jacko, J. A., et al. (2007). Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization & Computer Graphics*, (6).

[Zeileis et al., 2009] Zeileis, A., Hornik, K., and Murrell, P. (2009). Escaping RGBland: Selecting colors for statistical graphics. *Computational Statistics & Data Analysis*, 53(9):3259–3270.

[Zhou et al., 2009] Zhou, H., Cui, W., Qu, H., Wu, Y., Yuan, X., and Zhuo, W. (2009). Splatting the lines in parallel coordinates. In *Computer Graphics Forum*, volume 28, pages 759–766. Wiley Online Library.

[Zhou and Feiner, 1998] Zhou, M. X. and Feiner, S. K. (1998). Visual task characterization for automated visual discourse synthesis. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 392–399. ACM Press/Addison-Wesley Publishing Co.

[Zuk et al., 2006] Zuk, T., Schlesier, L., Neumann, P., Hancock, M. S., and Carpendale, S. (2006). Heuristics for information visualization evaluation. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, pages 1–6. ACM.