(Silvia Miksch)

(Daniel Keim)

PhD Thesis in Computer Science

Facets of Time

Making the Most of Time's Structure in Interactive Visualization

Conducted for the purpose of receiving the academic title
"Doktor der technischen Wissenschaften"

Supervisor
Silvia Miksch
Institute of Software Technology and Interactive Systems (E188)

Submitted at Vienna University of Technology
Faculty of Informatics

by
Tim Lammarsch
0726158
Ernst-Krenek-Straße 16a
3500 Krems
Austria

Krems, April 22, 2010

(Tim Lammarsch)

Übersicht

Zeitorientierte Daten sind in vielen praktischen Problemen aus Gebieten wie Gewerbe, Medizin oder Ökologie allgegenwärtig. Probleme, die mit zeitorientierten Daten in Zusammenhang stehen, bedürfen besonderer Aufmerksamkeit, da sich Zeitdaten von anderen Arten an Daten unterscheiden. Zeit hat eine inhärente Struktur, die man vor allem am Kalenderaspekt ersehen kann: Zeit ist aus Granularitäten wie Jahren oder Jahreszeiten zusammengesetzt. Da diese Granularitäten natürliche wie soziale Aspekte beeinflussen, die sich in den Daten wiederfinden, kann es die Menge gefunder Informationen deutlich vergrößern, wenn man die Struktur der Zeit in Datenanalysemethoden explizit ausnutzt.

Um Innovationen zu entwickeln, wie man am besten mit zeitorientierten Daten in Visual Analytics umgeht wurde das Projekt **DisCō** durchgeführt, unter Berücksichtigung einer Gruppe von BenutzerInnen, bei deren Tätigkeit es um die Analyse zeitorientierte Daten aus verschiedenen Sektoren geht. Bei diesen Anwendungsszenarien sind oft ungenau definierte Problemstellungen im Spiel, und die Anwendung der Struktur der Zeit ist eine der wenigen Methoden, die ihnen zur Verfügung stehen, um ihre Aufgaben zu erfüllen. Die neuartigen Visualisierungen, die wir präsentieren werden, wurden in Hinblick darauf evaluiert, ob sie die Bedürfnisse dieser BenutzerInnen erfüllen können.

Die Struktur der Zeit ist bereits ein zentraler Faktor in der Forschung des Bereichs Temporale Datenbanken. In diesem Kontext wurde auch der Konzept der Granularitäten erstmalig beschrieben. Mit Granularitäten können verschiedene Aspekte der Struktur der Zeit modelliert werden— ordinale vs. diskrete Skalen, lineare vs. zyklische Zeit, und vor allen Dingen Kalender. Wir bauen daher auf diesem bewährten Konzept auf, behandeln aber auch den Aspekt verschiedener Ansichten. Die Struktur der Zeit wird in den Visualisierungsprozess integriert, indem wir etablierte Modelle erweitern.

Das Ziel dieser Arbeit ist es, die Struktur der Zeit in interaktive Visualisierungen zu integrieren. Wir erwarten, dass ein großer Bereich potentieller BenutzerInnen mit vielfältigen Aufgaben davon profitieren wird. Viele der BenutzerInnen und Aufgabenstellungen, die wir bereits berücksichtigen konnten, stammen indess aus dem Projekt **DisCō**. Allerdings sind diese nur als Beispiele für den weit größeren Bereich von BenutzerInnen und Aufgabenstellungen zu betrachten, der vom Gebiet der Informationvisualisierung abgedeckt wird.

Übersicht

Um ein besseres Verständnis vom Stand der Technik zu gewinnen, aber auch um unsere neuartigen Ansätze besser beschreiben zu können, beinhaltet die Arbeit eine detaillierte Analyse davon, wie der Visualisierungsprozess vonstatten geht. Diese Analyse besteht aus einem prozessorientierten Modell sowie eine Taxonomy von Visualisierungen, die die Struktur der Zeit verwenden. Die Taxonomie basiert auf den verschiedenen Stufen des Prozessmodells. Im folgenden wird die Taxonomie auf verschiedene Visualisierungen des aktuellen Stands der Technik, die die Struktur der Zeit berücksichtigen, angewandt.

Basierend darauf, dass wir den Stand der Technik von konzeptueller wie benutzerInnenorientierter Seite analysiert haben, werden im Anschluss neue interaktive Visualisierungen präsentiert, die spezifisch entwickelt wurden, um die Struktur der Zeit auszunutzen. Das beinhaltet Methoden, die den Kalenderaspekt verwenden, wie auch Methoden, die darauf abzielen, verschiedene Ansichten zu zeigen. Viel Forschung in Hinblick auf das Zeigen verschiedener Ansichten konzentriert sich auf verschiedene Ansichten aus verschiedenen Quellen, wie z.B. verschiedenen Personen, wobei die einzelnen Angaben oder Werte sich auf den selben Punkt in der Zeit beziehen. Im Gegensatz dazu haben wir erkundet, wie es möglich ist, verschiedene Ansichten (die gewöhnlich von der selben Quelle stammen) zu zeigen, die sich über eine weitere Zeitlinie hinwer verändern (z.B. eine Person, die ihre Ansichten ändert). Weitere Visualisierungen in dieser Arbeit sind dazu da, Muster zu visualisieren, die von Methoden aus dem Bereich Data Mining gefunden wurden. Dabei verwenden wir die Struktur der Zeit, um BenutzerInnen ein besseres Verständnis zu ermöglichen.

Um die Bedienbarkeit, Brauchbarkeit, und Angemessenheit (engl. „usability, utility, and appropriateness“) unserer Ergebnisse zu bestätigen haben wir zwei Studien mit BenutzerInnen durchgeführt und außerdem eine Reihe von Einschätzungen durch Experten eingeholt. Weitere Schritte der Evaluierung sowie die Umsetzung unserer aktuellen Ergebnisse werden in unserem Kapitel über zukünftige Tätigkeit (engl. „future work“) aufgezeigt.

Abstract

Time-oriented data are ubiquitous in many real-world problems from areas like business, medicine, or ecology. Problems involving time-oriented data need special attention because time data are different from other kinds of data. Time has an inherent structure, most prominently seen in the calendar aspect of time being composed of smaller granularities, like years and seasons. As these granularities influence natural and social aspects found again in the data, explicitly harnessing the structures of time in data analysis methods can greatly improve the amount of information gained.

To develop innovations in how to deal with time-oriented data in Visual Analytics, the project **DisCō** has been pursued in the context of a certain set of users who handle application scenarios of time-oriented data analysis in different sectors. Those scenarios often involve ill-defined problems, and the structure of time is one of the few tools they have to solve those tasks. The novel visualizations are evaluated regarding the question whether they can fulfill these users' needs.

The structure of time is already a central focus of research in Temporal Databases. In that context, the concept of granularities was first laid out. Granularities can model several aspects of time's structure—ordinal vs. discrete scales, linear vs. cyclic time, and above all calendars. We build upon this approved concept, but also add support for different points of view. Introducing the structure of time into the modelling of the visualizaton process is done by expanding established process-oriented frameworks.

The goal in this thesis is to include the structure of time in interactive visualizations. We envision a broad range of any possible users and tasks that have a connection to this area. Many of the users and tasks we had access to stem from the research project **DisCō**, though. Still, those are only to be seen as examplary characteristics of the user and task space of Information Visualization in general.

To gain a better understanding of the state of the art, but also to have a better way of describing our novel approaches, we include a detailed analysis how the visual exploration process is performed. The analysis consistes of a process-oriented framework as well as a taxonomy for visualizations that make use of the structure of time. The taxonomy is based on the several levels of the framework. Then, the taxonomy is applied to various state-of-the-art visualizations that make use of the structure of time.

Based on the state of the art and its analysis from a theoretical as well as a user-centered point of view, new interactive visualization methods that are specifically designed to harness the structure of time are presented. These include methods to make use of the calendar aspect as well as showing different points of view. Much research regarding different points of view is focused on different sources, like different persons, who provide different values for the same point in time. Contrary to that, we research different views (usually from one source) that differ over their own timeline (for example, a person changing her mind). Further visualizations contained in this thesis have the task to disclose patterns from Data Mining methods to users, applying the structure of time for a better understanding.

To confirm the usability, utility, and appropriateness of our results, we have performed two user studies and acquired a number of expert assessments. Further evaluation steps as well as the application of the current results are presented in our chapter of future work.

Acknowledgments

I would like to thank my advisor Silvia Miksch for all her support over the course of my PhD study, the detailed and always motivating feedback, for her patience, and also for her explaining Austria to her puzzled protégés from abroad.

I also would like to thank my secondary referee Daniel Keim who, despite of his many responsibilities, spared his time to appraise this thesis and visit the viva voce.

Further thanks go to my fellow members of the **DisCō** project team as well as the colleagues from ike: Wolfgang Aigner, Ingrid Bauer, Alessio Bertone, Paolo Federico, Johannes Gärtner, Robert Klausner, Peter Klinka, Eva Mayr, Dieter Punzengruber, Alexander Rind, Hanna Risku, Alexander Schratt, Michael Smuc, Thomas Turic, and Sabine Wahl. All of them supported this work to the best of their particular skills. Special thanks go to Paolo Federico who took the time of providing an expert review for several of the novel visualizations and to Wolfgang Aigner who has mastered time to the point where he seemingly can generate more of it for reviewing his colleague's work.

As for typesetting, layout, and orthography reviewing, my thanks go to Joachim and Marion Lammarsch.

My position at ike and in the project **DisCō** was made possible by strong support from my family over my whole life, therefore I would like to thank Albert Kannegießer, Gertrud Kannegießer, Joachim Lammarsch, Renate Lammarsch, and Julia Weick. I often missed them in my time abroad.

Finally, many thanks go to my girlfriend Anita Grutzeck who believed in me managing to achieve a PhD when I was still looking for a suitable position in academic research. Anita has helped me with love, motivation, and luxury nutrition through the difficult time of writing this thesis.

Contents

—All our knowledge has its origins in our perceptions.

Leonardo da Vinci, 1452–1519

1 Introduction

1.1 Motivation

Recent technological advance has opened up numerous new data sources and enabled us to gather increasing amounts of raw data. However, the capabilities to process and understand the data have not been keeping pace. As a result, in many areas, like business, health care, and ecology, there is an increasing need of exploring and understanding data.

The need to transform these increasing amounts of collected and stored data into meaningful information has been the focus of the novel research area Visual Analytics [Thomas and Cook, 2005, Keim et al., 2008]. The analytical reasoning abilities of automated systems are combined with human reasoning and expert knowledge by the means of interactive visualization.

Time-oriented data is ubiquitous in many real-world problems from areas like business, health-care, public service, or environmental monitoring. Therefore, it is very important to develop novel methods for the analysis of time-oriented data. Problems involving time-oriented data need special attention because time data is different from other kinds of data [Aigner et al., 2007b, Kuchar et al., 2006]. Shneiderman [1996] identifies time-oriented data as one of the seven basic data types. A lot of research has already been done in visualization as well as automated analysis of time-oriented data—good overviews of this work are presented in [Aigner, 2006, Roddick and Spiliopoulou, 1999].

Typically, time-oriented data is multivariate and it stems from heterogeneous data sources. It also has another important and unique feature: Unlike other data dimensions which are usually considered "flat", time has an inherent structure. A particular aspect of this structure is the calendar aspect of time being composed of smaller granularities, like years and seasons. These granularities influence natural and social aspects found again in the data. For instance, the weekly time structure is tightly connected to business (market), religious (ceremonies), and other social activities (shopping behavior) [Zerubavel, 1989]. Therefore, explicitly harnessing the structures of time in data analysis methods can greatly improve the amount of information gained. In doing so, a large number of time-related aspects which can be classified in domain, primitives, determinacy, perspectives, and calendars [Goralwalla et al., 1998] need to be considered.

The analysis of data from such domains is an important task performed by business analysts as well as other companies and organizations. Time-oriented data is often non regularly gathered, multigranular, and multivariate. To find new Visual Analytics methods for analyzing such data, the project **DisCō**[1] has been performed by the Department of Information and Knowledge Engineering and the Department for Department for Knowledge and Communication Management at Danube University Krems as well as XIMES Gmbh, Vienna. A special focus of **DisCō** was dealing with the structure of time as a means to overcome the challenges of dealing with time-oriented data. Although the focus of this thesis are Interactive Information Visualization (InfoVis) methods that employ the structure of time, in order to present many of the **DisCō** results, we also have to digress into Temporal Data Mining. This is due to the fact that the Visual Analytics topcis of **DisCō** cannot fully be divided into the two strands—actually that was a main goal of **DisCō**.

1.2 Research Questions

1.2.1 Main Question

How can interactive visualization be used to explore time-oriented data according to the user's tasks while considering the structure of time?

> We have shown the importance of time-oriented data in Section 1.1. Without loss of generality, the focus of this thesis will be the use of interactive visualization to explore it. Moreover, using the structure of time will be the central aspect considering visualization.

[1] **DisCō** (Latin: "I learn")—Visual DIScovery and COmmunication of complex time patterns in Supported by the program "FIT-IT Visual Computing" of the Federal Ministry of Transport, Innovation and Technology, Austria (Project number: 813388). For more information see http://www.donau-uni.ac.at/disco.

1.2.2 Further Questions

Which operations are part of a visualization that considers the structure of time and what is the design space they span?

> For classifying related work regarding the main question and to develop innovative interactive visualizations, it is helpful to formulate a model. Such models are usually either (1) process-oriented frameworks based on a concept of operations or (2) taxonomies. Firstly, we provide such a process-oriented framework in Chapter 3, secondly, we use the framework as backbone of a multi-level taxonomy in Chapter 8.

Which visualizations exist that can be used if the structure of time is important?

> The state-of-the-art-analysis of this thesis abstains from a wider focus in favor of a more detailed description how the structure of time is contained in the visualizations based on our framework and taxonomy. An analysis of visualizations for time-oriented data in general has already been done in [Aigner, 2006].

What are the situations when it is feasible to use the structure of time for visualization even when it is less prominent in the data—and how can it be done?

> This question is composed of data that results from analytic methods that do not make use of the structure of time as well as visualizations based on no data at all beside the structure of time (e.g., calendars).

1.3 Approach

Most visualizations for time-oriented data fall into one of two categories. One kind treats time like any other attribute and therefore has limited design space. The other kind uses the the distinct features of time, for example its structure. We focus on visualizations of the latter kind. It is a complex task to properly define and describe a visualization that makes use of the structure of time. Some of those visualizations reduce complexity by focusing on certain aspects of the structure of time, for example using the concept of seasonal components. Our goal in this thesis is to reach a concept that ist easy to understand and flexible enough to model a large number of powerful visualizations that employ the structure of time. To reach this goal, the thesis will comprehend the following steps:

1. Aspects of time-oriented data are defined and analyzed in the context of the structure of time in Chapter 2. We base our work on the model for time-oriented data by Aigner et al. [2007a,b] as well as the concept of temporal granularities by Bettini et al. [2000].

2. A process-oriented framework that shows how such data can be visualized interactively is defined by us in Chapter 3. This framework is based on the data model from Chapter 2 as well as the general visualization process frameworks by Chi and Riedl [1998] and by Card et al. [1999]. Like those frameworks, we also apply the concept of visual mapping defined by Bertin [1983]. As the framework uses UML diagrams, we provide the UML conventions in Appendix A.

3. Before describing potential visualizations based on our data model and framework, we provide an analysis of potential users and their tasks in Chapter 5. This analysis has been performed with the help of real-world users as part of the project **DisCō** [Smuc et al., 2008, 2009].

4. The state-of-the-art-analysis is preceded by an overview of the sources we have analyzed regarding their intentions and approaches. This is done in Chapter 7.

5. Before listing the visualizations, we define a multi-level taxonomy based on our framework in Chapter 8. This taxonomy classifies interactive visualizations based their features regarding separate operations in the framework.

6. Chapter 9 lists all interactive visualizations employing the structure of time that we have analyzed. They are all classified based on our taxonomy.

7. Over the Chapters 12–18 we present novel interactive visualization methods we have developed over the course of **DisCō**. Those visualization make extensive use of the structure of time to aid several important user tasks we have attended during the project. They are also classified according to our taxonomy.

After each part, there is a summary, leading up to the main summary in Chapter 19 and the future work in Chapter 20.

1.4 Evaluation and Assessment

To confirm that our results are suitable to meet the users' needs and therefore answer the research questions, we have obtained assesments from

- the participants of the project **DisCō**;

- several users of our project partner from business consulting and time management;
- a further, external, visualization expert.

Moreover, we have already evaluated the visualizations from Chapter 12, which are a central part of our contribution, in two user studies lead by the Department for Knowledge and Communication Management of Danube University Krems, Austria. The planning for the next evaluation steps is detailed in Chapter 20.

The results of the assessments and evaluations are presented in the particular summary sections and Chapter 19. Our planned steps to account for these results are shown in Chapter 20.

1.5 Conventions

The work presented throughout this document was conducted over the course of the project **DisCō** in cooperation with the Department of Information and Knowledge Engineering (ike) at Danube University Krems, Austria, and other project partners. Even though most design and development ideas were carried out by myself, the research and its results were guided and revised based on valuable input from fellow project members and my advisors. Therefore, I decided to use the pronoun "we" rather than "I" throughout the thesis.

New technical terms being introduced are formatted in italic. Direct quotations are placed in quotation marks and followed by an indication of source in square brackets (for example, [Aigner, 2006]). If ideas or concepts of others are referred to, the respective reference is put at the appropriate position within a sentence, or right after the end of a sentence. If the names of the referenced authors are part of the sentence, for example the name of Aigner [2006], it is omitted from the square brackets.

References to the Bibliography chapter or other internal references are printed in green color. External references, like links to the world wide web, are printed in blue color. On a hypertext capable electronic display, all references are clickable and linked accordingly.

1.6 Dissemination

Some intermediate results presented in this thesis have already been published in several international publications. However, all of them have been revised, expanded, and placed in a more general context as part of their introduction into this work. In the following, we list the relevant publications that our research team has had:

Tim Lammarsch, Wolfgang Aigner, Alessio Bertone, Johannes Gärtner, Silvia Miksch, and Thomas Turic. A Comparison of Programming Platforms for Interactive Visualization in Web Browser Based Applications. In *Proceedings of 12th International Conference on Information Visualisation (IV08)*, pages 194–199. IEEE Computer Society Press, July 2008.

In this conference paper, we describe our preliminary research of the **DisCō** project regarding technical issues in web visualization. The results are not part of the PhD thesis, as they have no direct connection to its content. The prototype of our GROOVE visualization (see Chapter 12) used by real-world business consultants has partly been done based on the results from this paper.

Tim Lammarsch. A Compound Approach for Interactive Visualization of Time-Oriented Data. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology 2008 (VAST08)*, pages 177–178. IEEE Computer Society Press, October 2008.

In this PhD colloquium contribution, the idea of applying the structure of time as a means to improve interactive InfoVis of time-oriented data has been laid out.

Michael Smuc, Eva Mayr, Tim Lammarsch, Alessio Bertone, Wolfgang Aigner, Hanna Risku, and Silvia Miksch. Visualizations at First Sight: Do Insights Require Traininig? In *Proceedings of 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society (USAB 2008)*, pages 261–280. Springer, November 2008.

Michael Smuc, Eva Mayr, Tim Lammarsch, Wolfgang Aigner, Silvia Miksch, and Johannes Gärtner. To Score or Not to Score? Tripling Insights for Participatory Design. *IEEE Computer Graphics and Applications*, 29(3):29–38, May 2009.

These publications describe the initial user evaluations, the user studies based on the GROOVE visualization (see Chapter 12), and the application of the RIO visualization (see Chapter 5).

Tim Lammarsch, Wolfgang Aigner, Alessio Bertone, Johannes Gärtner, Eva Mayr, Silvia Miksch, and Michael Smuc. Hierarchical Temporal Patterns and Interactive Aggregated Views for Pixel-based Visualizations. In *Proceedings of 13th International Conference on Information Visualisation (IV09)*, pages 44–49. IEEE Computer Society Press, 2009.

This conference paper is focused on the development, variations, and application of the GROOVE visualization (see Chapter 12).

Wolfgang Aigner, Alessio Bertone, Tim Lammarsch, Silvia Miksch, and Alexander Rind. Interactively Exploring Time-Oriented Data. In *Workshop Notes of CHI 2009 workshop "Interacting with temporal data" at Conference on Human Factors in Computing Systems (CHI 2009)*. ACM Press, 2009.

Johannes Gärtner, Wolfgang Aigner, Alessio Bertone, Robert Klausner, Tim Lammarsch, Eva Mayr, Silvia Miksch, Hannah Risku, and Michael Smuc. Visual Analytics for Workforce Requirements. Accepted extended abstract at *19th International Symposium on Shiftwork and Working Time and Well-being in the 24-h Society*, Venezia, Italy, volume 29(3), pages 29–38. IEEE Computer Graphics & Application, 2009

This workshop notes and extended abstract are focused on applying the structure of time in practice—especially in work shift planning.

Alessio Bertone, Tim Lammarsch, Thomas Turic, Wolfgang Aigner, and Silvia Miksch. Does Jason Bourne need Visual Analytics to catch the Jackal? In *Proceedings of the European Symposium Visual Analytics Science and Technology (EuroVast2010), Bordeaux, France*, 2010, forthcoming.

This conference paper describes our Visual Analytics Framework for the MuTIny approach (see Chapter 15) and demonstrates its application to a fictional example.

The process-oriented framework from Chapter 3 and the taxonomy from Chapter 8 are to be published in an upcoming journal paper. The other topics of this thesis will be expanded in further projects.

Part I

Problem Analysis and Scope

—Time is not a reality [hupostasis], but a concept [noêma] or a measure [metron]. . .

Antiphon the Sophist, Truth

2 Time-oriented Data and the Structure of Time

In this chapter, we give a definition of time-oriented data. Moreover, we define data aspects and aspects of time-oriented data. As these definitions all depend on tasks at hand and methods applied to perform them, we consider different views on data, one of which we choose for this thesis.

2.1 Definitions

Several definitions for *time-oriented data* have been given in literature. Müller and Schumann [2003] define it as following:

> "Time dependent data[1] is characterized by data elements being a function of time. In general, data takes the following form
>
> $$d = f(t)$$
>
> For data defined at discrete time stamps t_i, this relation can be represented as:
>
> $$D = \{(t_1, d_1), (t_2, d_2), \ldots, (t_n, d_n)\}$$

[1] We consider *time dependent data* as a synomym for time-oriented data.

where

$$d_i = f(t_i)\text{''}$$

A similar definition is given by Weber et al. [2001b]:

> "Time series data is characterized by data elements being a function of time. In general, this data takes the following form:
>
> $$D = \{(t_1, y_1), (t_2, y_2), \ldots, (t_n, y_n)\}$$
>
> with
>
> $$y_i = f(t_i)\text{''}$$

The fact that the definitions are very similar to each other shows that there is some consensus how time-oriented data is reckoned. Furthermore, the second definition introduces the term *time series data* which is usually considered a synonym for time-oriented data. While we basically agree with this notion, we consider the term time series data somewhat misleading, as it implies nothing more than a consecutive order of data elements. We will show in Section 2.2.3 that there is much more to it.

A problem with these definitions is mentioned by Aigner [2006]: No concurrent data elements can exist. However, in real-world situations, this is often a case. For example, when considering starting planes at an airport, information about the flight plan may contain as many concurrent data elements as there are runways. To solve this problem, Aigner proposes a task-dependent approach and defines *time-oriented information* as

> "Information, where changes over time or temporal aspects play a central or are of interest."

This definition is broader than the definitions of time-oriented data by Müller and Schumann [2003] or Weber et al. [2001b].

To model this time-oriented information, Aigner [2006] introduces a *temporal data set* as a set of uniformly structured *temporal entities*:

$$\begin{aligned} \text{temporal data set } tds &:= \{te_0, te_1, \ldots, te_n\} \\ \text{temporal entity } te &:= \{a_0, a_1, \ldots, a_n\} \end{aligned}$$

According to Aigner, there is a restriction for these temporal entities, introducing the term *temporal attribute*:

$$\exists ta(ta \subseteq te \wedge ta = \{a | a \text{ is a temporal attribute}\} \wedge |ta| \geq 0)$$

Aigner defines a temporal attribute as "an attribute of a data set that represents (a part of) an aspect of time".

Consequently, Aigner focuses on a tree-oriented semantic structure of temporal entities that builds upon time-oriented data and thereby creates time-oriented information. This structure is very useful for human-centered understanding of time-oriented information. In this thesis, we will focus on an approach where users' tasks are relevant, but we do not try to model how they think about time-oriented data. Therefore, we will now present our own definitions regarding time-oriented data that are compatible to Aigner's definition of time-oriented data sets. Therefore, it is possible to apply Aigner's further definition to the work from this thesis, a fact that will be considered in Chapter 20, Future Work.

We start by defining *data*, *datasets*, *data elements*, *data values*, *data schemes*, and *data attributes*. We will add the time-oriented aspect later. Let data D be a set of several datasets s_i:

$$D := \{s_0, s_1, \dots, s_l\}$$

Let a dataset be a set of data elements $e_{i,j}$ that adhere to the data scheme c_{s_i} that is allocated to the dataset:

$$s_i := \{e_{i,0}, e_{i,1}, \dots, e_{i,m}\}\ \forall i \epsilon \{0, \dots, l\} \quad | e_{i,j} \text{ adheres to } c_{s_i}\ \forall i \epsilon \{0, \dots, l\}, j \epsilon \{0, \dots, m\}$$

Let a data scheme c_{s_i} be a set of data attributes $a_{s_i,\bar{k}}$:

$$c_{s_i} := \{a_{s_i,0}, a_{s_i,1}, \dots, a_{s_i,\bar{n}}\}\ \forall i \epsilon \{0, \dots, l\}$$

Let a data attribute be a specific description of data according to data type (see Section 2.2) and meaning. Let a data element be a set of data values $v_{i,j,k}$ that adhere to the data aspects of the corresponding scheme of the corresponding dataset. When a dataset adheres to a data scheme, it is imperative that $n = \bar{n}$:

$$\begin{aligned} e_{i,j} := \{v_{i,j,0}, v_{i,j,1}, \dots, v_{i,j,n}\}\ \forall i \epsilon \{0, \dots, l\}, j \epsilon \{0, \dots, m\} \\ | n = \bar{n} \wedge v_{i,j,k} \text{ adheres to } a_{s_i,\bar{k}} \\ \forall i \epsilon \{0, \dots, l\}, j \epsilon \{0, \dots, m\}, (k \epsilon 0, \dots, n, \bar{k} \epsilon 0, \dots, \bar{n} | n = \bar{n}), k = \bar{k} \end{aligned}$$

With these definitions, we can now define a *time-oriented dataset* as a dataset in whose data scheme at least one data attribute has a time-oriented semantic context (see Section 2.2.2). Let *time-oriented data* be data that contains at least one time-oriented dataset.

From a syntactic point of view, it is possible to view time-oriented aspects like any other kind of data aspects. For example, it is possible to consider a time-oriented aspect, like the day of year, the same as a space-oriented aspect, like the position of earth relative to the sun. By separating syntactic from semantic meaning, we can simplify our work without loosing the ability to account for the special characteristics of time-oriented data.

2.2 Data Aspects and Views on Data

Daassi et al. [2004] claim that time-oriented data can be divided into a structural dimension and a temporal dimension. Aigner [2006] seconds that point of view, but calls them data aspects and temporal aspects. As we have already pointed out in the last section, it is often difficult to make a clear distinction wheter some aspect is a data aspect or a temporal aspect. Furthermore, we consider the term "dimension" to be problematic. *Dimension* has been well defined in statistics research with a different meaning, so we argue against redefining the term for temporal reasoning.

As an alternative, we propose the concept of different *views* on the data. We define a view as a kind of semantical understanding of the data. By using the concept of views, we can define a time-oriented dataset as a dataset whose attributes can be considered in a time-oriented view. However, it is still possible to have a different, not time-oriented, view on the same dataset.

2.2.1 Data Type Taxonomy

As an example for some rather general views on data, we present a taxonomy based on the Task By Data Type Taxonomy by Shneiderman [1996]. We present an abbrevated form that we adapted from Aigner [2006] and restructured slightly.

1-dimensional This kind of structure refers to datasets that contain a single non-temporal variable. Examples for this kind of data include text documents, temperature measurements, sales numbers, etc.

map Map data is spatial data in two dimensions as for example geographic maps, floor plans, sitemaps, construction drawings, etc.

3-dimensional 3D structured data refers to volumetric (real-world) objects such as for example buildings, the human body, or molecules.

multi-dimensional Multivariate data structures contain multiple non-temporal, non-spatial variables as for example multi-parameter data from relational databases.

tree/hierarchical Accordingly structured datasets contain information about items and parent-child relationships whereas each node might have multiple parameters. Examples are directory data of computer storage systems or the classification of lifeforms.

network Datasets containing network data describe items (nodes) and relationships (edges) where one node might be connected to an arbitrary number of other nodes. This is in contrast to tree structures where one node might only have one parent node. Examples are hypertext documents or roadmap metadata.

temporal Datasets distincted by Shneiderman from 1-dimensional data by the fact that items have a start and finish time and that items may overlap. In the context of this thesis, we also consider datasets temporal, or, in our formulation, time-oriented, when there is any time-oriented aspect, so we are not limited to datasets distincted by start and finish times. Depending on the context, it is also possible to adapt a view that adds one of the other classes of this taxonomy to the data. For example, time-oriented data can be broken down in the form of a tree (see Subsection 9.1.9).

The concept of different views allows for viewing the same dataset as belonging to one or more of these classes. But it is also possible to use different views on a more detailed level.

2.2.2 Time-oriented Aspects

When a dataset qualifies for a time-oriented view, it is possible to model this view according to different aspects. A possible listing of these aspects has been done by Aigner et al. [2007a]:

Scale Time can be considered nominal, ordinal, discrete, or continuous. The possibiliy continuous can be excluded for almost all applications, not to mention the fact that quantum theory gives reason to persume that there is no continous time at all in real-world applications. Discrete time and ordinal time are closely related. When dealing with the possible existence of cycles only at the structure aspect and considering ordinal string values as interchangeable with discrete numbers, they can be considered the same. What remains is a distinction between nominal and ordinal scales. It is debatable whether a time scale like Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday is nominal or ordinal. Depending on the view, it is also possible to completely integrate the scale aspect into the calendar aspect.

Scope A data element can reflect a discrete point in time or an interval. In another work, Aigner [2006] combines this with the differentiation between anchored and unanchored time to the “primitives” aspect. The unanchored equivalent to an interval is a span. Depending on the view, scope can also partly be integrated into the calendar aspect.

Structure This aspect consists of linear time and cyclic time. In real-world applications cyclic models are very useful, but usually, only linear datasets are provided. It is an important task to generate cyclic approximations. Structure is the third aspect that can also be considered part of the calendar aspect in a different view.

Viewpoints The possibilities mentioned by Aigner et al. [2007a] are ordered time, branching time, and time with multiple perspectives. In the context of the project **DisCō**, a special case of these multiple perspective was very important: Different perspectives not from different sources but from different points in the temporal history of one source. More precisely, an important real world application is the evaluation of existing forecast systems, comparing their forecast values at different points in time to the actual value. As a side remark, data of this kind can be stored in databases using the bitemporal data model with a valid time and a transaction time. Our focus are ways to visualize such data, though.

Granularities This aspect is also called the *calendar aspect*. We use that name in this thesis, too. Aigner et al. [2007a] describe granularities as mappings from time values to larger or smaller conceptual units. Bettini et al. [2000] define a granularity as a mapping from integers to subsets of the time domain with certain restrictions. We will explain this definition in detail in the following subsection and formulate our definition of the calendar aspect based on it.

Together, these aspects clearly show that time is much more than just another data dimension. Depending on the task, one or more of the aspects have to be paid attention to when analyzing time-oriented data. We call the sum of these aspects and the effects they have on time-oriented data and time itself the *structure of time*.

2.2.3 The Calendar Aspect

Calendars are an important aspect of human social life. As calendars have always been based on natural phenomena, the calendar aspect can be found in nearly all datasets. The seven-day-cycle of the week can, for example, be found in social, biological, and geological contexts [Cornélissen et al., 2000]—perhaps all a result of the lunar cycle.

Bettini et al. [2000] define a calendar based on the concept of *granularities*. Thereby, they define a granularity as follows:

> "A granularity is a mapping G from the integers (the *index* set) to subsets of the time domain such that: (1) if $i < j$ and $G(i)$ and $G(j)$ are nonempty, then each element of $G(i)$ is less than all elements of $G(j)$, and (2) if $i < k < j$ and $G(i)$ and $G(j)$ are nonempty, then $G(k)$ is nonempty."

Bettini et al. [2000] also define granularity relationships, granularity conversions, *granularity systems*, and a number of algebraic operations on granularities, above all, granularity labelling. However, it would exceed the scope of this thesis to reproduce all of their work here. Therefore, we restrain ourselves to describe the calendar aspect and refer to [Bettini et al., 2000] for mathematical definitions.

Granularity Grouping and Perodical Granularity Grouping

A primary task of calendars is the division of time into units that can be handled by human users. It is near to impossible for us to imagine something to happen at 62,100 seconds. However, it seems perfectly clear what 17 hours and 15 minutes means—even if that might give humans a feeling of a greater degree of understanding than the one they actually have. In this example, we stay in the temporal context of one single day, but we already note a clear difference between the two possibilities. The units a granularity is composed of are called *granules* except for the finest granularity, where the units are called *chronons*. An example is shown in Figure 2.1.

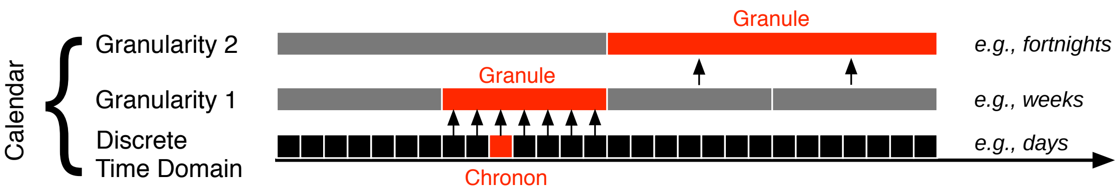


Figure 2.1: Example of a Discrete Time Domain with Multiple Granularities [Aigner et al., 2007a]. The smallest possible unit (chronon) is one *day*. Based on this, the granularity *weeks* contains granules that are defined as being a continuous set of seven days. Moreover, the granularity *fortnights* consists of granules that are a set of two consecutive weeks.

This part of the calendar aspect can be modelled by the operation of *grouping granularities*. 60 seconds group into one minute, 60 minutes into one hour, and so on. However, in the next minute, the seconds start again at zero. Therefore, the most common kind of granularity grouping in real-world applications is *periodical grouping*. Examples for grouping that is not periodical usually exist on a larger scale of our calendar. For example, the European history is often grouped into classical antiquity from 800 BC to 600 AD, the middle ages from 600 AD to 1500 AD, and modern ages from 1500 AD up to today.

When adopting a view regarding the structure of time, the scope aspect is enhanced. Except for the finest granularity, each granule can be seen as an instant on its own granularity, but at the same time it is an interval on the finer granularities. For spans, this difference does not exist, because one granule in the future or the past is still a span. The calendar aspect requires complicated mathematical operations to freely work with granularities. These operations have all be explained in detail [Bettini et al., 2000], we focus on methods for their application.

Another aspect that can be included into the calendar aspect by the means of granularity grouping is the structure aspect. When a granularity is grouped periodically into another granularity, a limited cycle in time regarding that granularity is modelled.

Granularity Labels

While it is natural for us to count granules with integers, like the second with the number zero up to the second with the number 59, the history example showed that sometimes they can be labelled with names. Named granules are also common for periodical grouping, for example the months of year or the days of week.

As a result, granularity labels enable the calendar aspect to encompass the scale aspect. Usually, the labels can be viewed under an ordinal scale. As we consider granularities as something that can be configured very freely, in some cases, like time-oriented states of a stochastic process, only a nominal scale is possible.

2.2.4 Different Views on Data Aspects

The same data can be viewed differently. Even if it is time-oriented, it is possible to adapt a view of the data being a simple accumulation of data elements. Therefore, it is possible to apply abstract Data Mining and visualization techniques. However, those techniques are not able to make use of the structure of time.

On the other hand, it is possible to have a view on time-oriented data that encompasses one or more aspects of time. Therefore, it is possible to model techniques that are specifically focused on those aspects to gain new insights into the data.

In this thesis, our goal is to reach best of both worlds. We define a framework that enables us to use the structure of time. This framework is defined in a flexible way and compatible to widely accepted frameworks. It is able to serve as a preprocessing for abstract techniques. Therefore, we can use the whole design space of InfoVis while still making use of the structure of time to the full extent. Speaking in views, we process the data in a view that respects the structure of time, but after that we provide an intermediate step of preprocessed data that already has the structure of time included and enable a view on that intermediata data that is neutral regarding data types. The following step is the visual representation, presented in the next chapter.

2.3 Summary

In this chapter, we have definied time-oriented data in a way that leaves room for other definitions by using different views on it. Our view is focused on the structure of time, combining the aspects of scale, scope, structure, viewpoints and granularities (the calendar aspect). Consequently, the contents of this chapter can be considered temporal reasoning. The following chapter will bridge the gap between the structure of time and InfoVis research.

—On ne confond plus langage et écriture, tournure de phrase et calligraphie. Mais on confond encore construction d'une image et qualité du trait. Combien de dessins admirablement exécutés et richement reproduits trahissent leur titre et ne communiquent qu'une information dérisoire et inutile ?

Jacques Bertin, Sémiologie Graphique

3 Visual Representation of Time-oriented Data

In this chapter, we will present the well-established existing concept of Visual Mapping and explain the established model for the Visualization process. Moreover, we will introduce a novel process-oriented framework. This framework has been developed by us in order to explain the special considerations needed for visualizing information according to the structure of time. The framework will serve as basis for a taxonomy of visualizations for time-oriented data in Part II of the thesis.

3.1 Visual Mapping

The concept of Visual Mapping goes back to Bertin [1967], later translated to English [Bertin, 1983]. Bertin introduces the concept of *visual variable*, defined as "components of the graphic sign-system". For Bertin, there are eight visual variables, formulated primarily with graphical marks in mind, for example on a map:

Two Dimensions of the Plane Those are the position on the horizontal axis and the position on the vertical axis (see Figure 3.1.a).

Size With size, the size of the mark is meant (see Figure 3.1.b). The scaling in our example is proportional, but in following chapters, we will also use size only in respect to one dimension, for example in bar plots.

Value In a graphical context, value has a similar meaning to brightness or lightness (see Figure 3.1.c). The exact meaning is dependent on the color model used (see below).

Texture Bertin makes a distinction between the pattern of the texture and the frequency of the repetition (see Figure 3.1.d).

Color What Bertin originally describes as color is today usually known as hue (see Figure 3.1.e).

Orientation Orientation is measured as the angle of a mark compared to other marks (see Figure 3.1.f).

Shape Bertin considers shape to have infinite variations regarding the marks (see Figure 3.1.g).

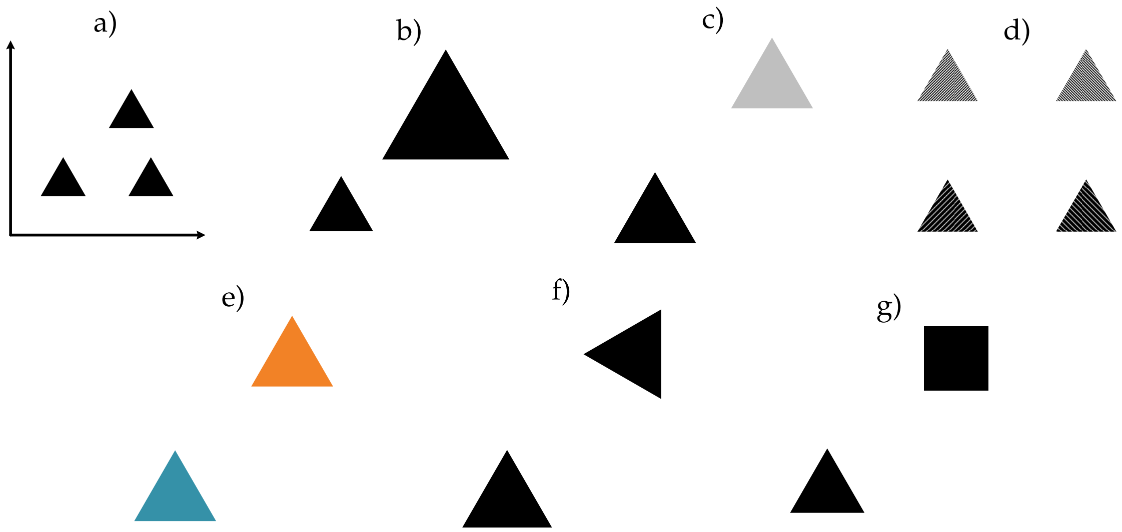


Figure 3.1: The Visual Variables Defined by Bertin [1967]. a) Position in horizontal and vertical directions—the other visualizations also show position because it would be difficult not to, but this one only shows position. b) Size c) Value d) Texture—left/right shows pattern, upper/lower frequency. e) Color f) Orientation g) Shape. Modern definitions of the visual variables have introduced some changes from these.

3.1.1 Expansion of the Concept of Visual Variables

Several expansions to the concept of visual variables have been made in later publications. For example, Mackinlay [1986] lists position, length, angle, slope, area, volume, density, color saturation, color hue, texture, connection, containment, and shape.

Newer definitions of visual variables, while all referring to [Bertin, 1983], often set themselves apart by two major distinctions:

1. They are not focused solely on marks, but also are applied in the context of line plots and other kinds of visualizations.

2. Bertin explicitely states that "We will only consider that which can be represented by readily available graphic means, on a flat sheet of white paper of standard size and under normal lighting." [Bertin, 1983]. Contrary to that, newer definitions are focused on computer displays that can display structures too complicated to generate manually. Moreover, computer displays can change the view over time, and even do this interactively.

Moreover, it is possible to make one or two steps up in the logical complexity of visual mapping. For example, Bostock and Heer [2009] provide primitives like axes that can be a mapping target for data. Thereby, it is easier to describe the visualizations compared to mapping the same data aspect to position, symbol, and so on.

Many commercial charting libraries, however, go a step further and provide fixed visualizations only. These visualizations contain some parameters that data aspects can be mapped to. However, such visualizations are often too constrictive to use for several state-of-the-art visualizations.

3.1.2 Visual Variables in the Context of this Thesis

Above all, to correctly interpret our use of visual variables, it is important to understand that we do not intend to provide instructions for the implementation of an actual visualization tool. We rather pursue the goal of providing classification methods for various existing and new visualizations.

Consequently and diverging from Bertin and several other authors, but similar to newer definitions of visual mapping like the one from Card et al. [1999], we consider visual variables a logical abstraction rather than a direct access to the actual visualization. After mapping data attributes to visual variables, it is still possible to perform visual transformations. For example, instead of just mapping two data attributes to two dimensions of space, it is possible two map three or even more data attributes to dimensions of space and perform a projection on the two-dimensional plane of a conventional computer display. We will delve into this topic in Section 3.2 and 3.3.

Furthermore, we include visual variable of a medium logical complexity, like axes, that are also not part of the original idea of visual variables. Visual variables are also conceivable with much higher complexity. For example, it is possible to map structured data (above all, time) to a visual variable complex enough to do so. On the contrary, we only consider visual variables in this thesis that are mapping target of much simpler values. The structure of time is simplified to such values by using the concept of granularities, but at earlier steps of our visualization process (see Section 3.3).

We will now list a number of visual variables that are important in this thesis. This list is not intended to be a full register of current visual variables. It is rather an excerpt of examples that will also be used prominently in Chapter 9 and 12–18.

Position in Space This is the same as Bertin's dimensions of the plane. The example visualizations in this thesis only use two dimensions. However, any number of dimensions is possible in theory. Projection can be used to map those extra dimensions to the two or three dimensions available on a computer display.

We expand the definition of this visual variable by allowing flexible scales. It is not mandatory to map the range of the data attribute to the whole space available for the visualization. Instead, only part of the visualization space can be used. Moreover, the offset of the used space can be visual variable for another data attribute. As a result, it is possible to generate sub-position in space. A renowned example for this is the recursive pattern arrangement by Keim et al. [1995], described in detail in Chapter 9.

Position is obviously a very powerful visual variable that can be used for almost any kind of data attribute. By using flexible scales and projection of higher dimensions, it is possible to use position for more than two data attributes, but doing this also reduces the ability of users to decode the visualization.

Hue As already mentioned, the visual variable called "color" by Bertin is usually referred to as "hue" today. However, there are several *color models* that define hue differently. Visualizations we have developed in the context of **DisCō** use the HCL color model [Ihaka, 2003] based on the CIELUV color space[1]

Hue is very suitable to distinguish several classes (up to about seven). However, it is dangerous to disclose an inherent order of those classes. Experts from computer graphics, InfoVis, and related areas tend to sort hues by the wavelength of light that produces them. Average users, on the other hand, could likely sort

[1] The CIELUV color space was defined by the International Commission on Illumination (Commission internationale de l'éclairage, CIE) in 1976 based on color-matching functions from the CIE 1931 standard. Details can be found at http://www.cie.co.at (accessed at April 9th, 2010).

them alphabetically or by personal preference. Hue should not be used for quantitative data or qualitative data with more than several classes, as the human perception systems tends to classify hues into a few bands [Rogowitz and Treinish, 1998], for example red, orange, yellow, green, cyan, blue, and violet. Furthermore, red-green color blind people (e.g., affected by deuteranomaly) cannot distinguish hues between red and green. While other forms of color blindness exist, this variant is by far the most common one. Therefore, it is more often considered in InfoVis.

Chroma Chroma is defined as the part of a color that describes its difference from gray. The HCL color model allows to visualize different hues at the same chroma, or different chromas for one hue. However, standard computer displays are not able to display correctly all variances that can be calculated. In other color models, similar concepts like saturation are used instead.

In some visualizations, chroma is only used to increase the available hues, adding gray as a different "color". There also is no exact counterpart for using chroma in Bertin's visual variables. More recent research has shown that chroma is quite suitable for quantitative values. More precisely, using a higher chroma has a visual effect of higher importance for humans [Silva et al., 2007].

Lightness Lightness is the part of a color humans perceive as light intensity. It is a subjective perception of brightness, sometimes (for example by Bertin) called value. Unfortunately, standard computer displays can only increase or decrease lightness in a rather limited range without decreasing chroma at the same time.

Lightness is also a suitable visual variable to show quantitative values. The difference to chroma is that it is understood semantically like a high/low variance rather than an important/unimportant variance [Silva et al., 2007].

Axes Mapping a data aspect to an axis is a higher level visual variable. It is the equivalent of mapping it to a position in space and also mapping it to a number of symbols that is rendered at the particular positions.

Legends A legend is a collection of all or the most important characteristic variants another visual variable can have. That visual variable is shown in parallel to a variable that can be encoded without prior knowledge, so for example, hues are accompanied by numbers to explain the meaning of the hues.

Comparing the visual variable, it is easier to read exact values when position is used, but as the number of data attributes to show often exceeds the position dimenions available, the visual variable that depend on color are a viable alternative at least for comparing values to each other and searching for patterns.

In the following, we will explain how Visual Mapping can be actually used to create visualizations from data.

3.2 Visualization Process

Visual mapping is an important part of the visualization process, but not the only one. Several authors have contributed to the development of a framework that is able to describe the visualization process and its sub-processes.

Most research papers that require such a framework rely on the Visualization Reference Model by Card et al. [1999] (see Figure 3.2) which is impressively simple and still captures the process completely. It contains data transformations, visual mappings and view transformations as performed actions. The states in between are raw data, data tables, visual structures, and views. Regarding the contents of the states, the visualization reference model does not contain the type details we want to build our framework on. Furthermore, there is a focus on the states, which makes it more difficult to model information related to state changes.

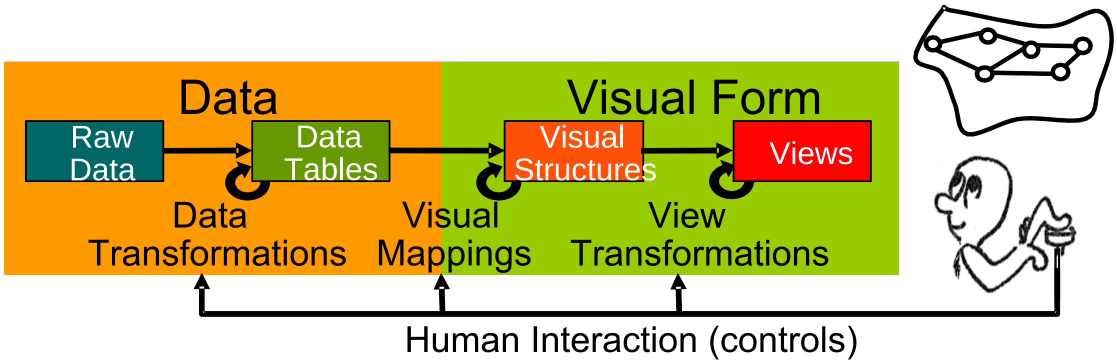


Figure 3.2: The Visualization Reference Model by Card et al. [1999]. The user has a task in mind and interacts with the model according to it.

Chi and Riedl [1998] developed a very detailed framework that incorporates state changes with the concept of *operators* that perform operations on the different stages of the visualization process (see Figure 3.3). They also define *operations* as being user interactions. This might be a bit puzzling, because very similar words stand for rather different things. The framework is thorough, but, it does not contain operators that take into account the structures of time. The reason is the fact that Chi's framework is based on a rather abstract view on the data. Still, it is possible to introduce new operators that are only available for certain kinds of data. We perform this in the following section.

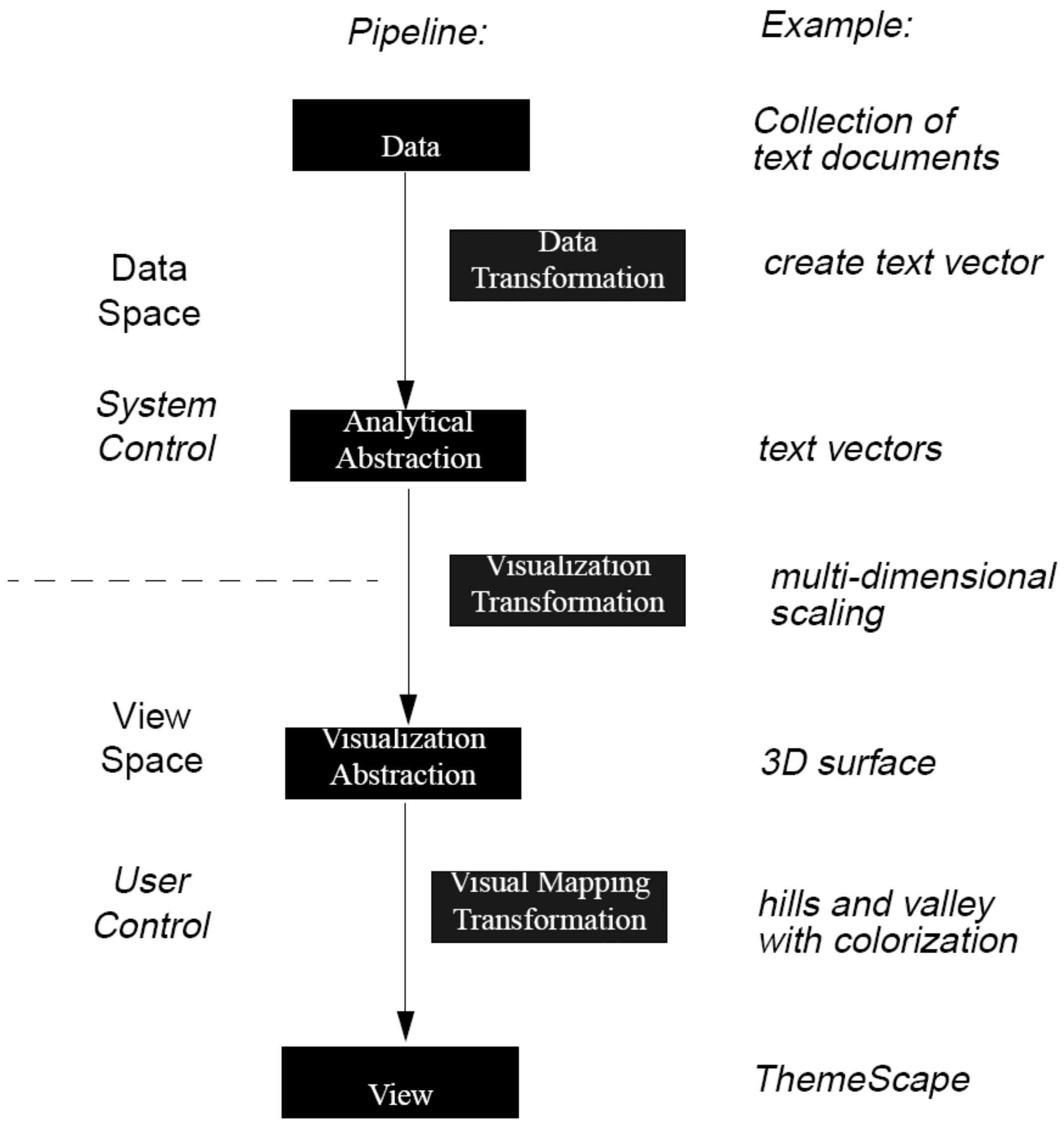


Figure 3.3: The Visualization Process Framework by Chi and Riedl [1998]. The states and transformations are similar to those of Card et al. [1999], but all are formed by combining *operations*.

3.3 Visualization Process of Time-oriented Data

We present a visualization process framework that is based on the one by Chi and Riedl [1998]. However, we consider the terms *operator* and *operation* too close to stand for something significantly different. Therefore, we do not call user interactions operations but stay with the original term. For us, *operation* is a synonym for *operator*. Operator is used when the process is in the focus of a description while operation is used when operands and result are described—no matter whether the user is involved in a change of the operands. In addition to all the existing operators defined by Chi and Riedl, we define operators that are specifically focused on dealing with the structure of time. The central point is the fact that each operator can be applied based on a different view on the data. Therefore, operators that do not need to deal with the structure of time can be defined much easier, but the framework as a whole is able to take every possible advantage of the structure of time.

Another important aspect in developing these operators is the ability of the resulting framework to describe the existing visualization that make use of the structure of time as good as possible. We present operators that model procedures unique to certain visualizations. Thereupon, these procedures can be transferred to other visualizations by help of the framework. Many "standard procedures" are shared by a large number of visualizations, though.

A novelty of the process framework of Chi and Riedl is that they make a clear distinction between data states and data transformations. Moreover, they introduce the concept of data stages. A data stage is a collection of data states. Data transformations can either result in a data state of the same stage or a data state of the following stage. We retain the concept of data stages and the stages defined by Chi and Riedl, but we show the distinction between stages and stage transformation in a visually more prominent way. Often, it is possible to switch the position of operators while still getting the same result. To present an application of the framework, a decision for one order has to be made without loss of generality. Chi and Riedl define operators specifically bound to a certain stage or stage transition. We consider some operators, like value filtering, to be applicable at different stages, like the data stage as well as the analytical abstraction stage. To keep it clear when an operator can be applied, it is necessary to define pre- and postconditions for operators. Each operator at a certain stage has the precondition that all operations of earlier stages have to be peformed. Regarding operators inside one stage, we consider them interchangeable unless an operation requires certain preconditions.

We now continue with the definition of operators related to the structures of time. An overview of the visualization process according to the framework as well as the most important operators for our task is depicted in Figure 3.4. However, to define operators in detail, we have decided to use Unified Modelling Language (UML) 2 activity diagrams. A short description of UML 2 ist provided in Appendix A.

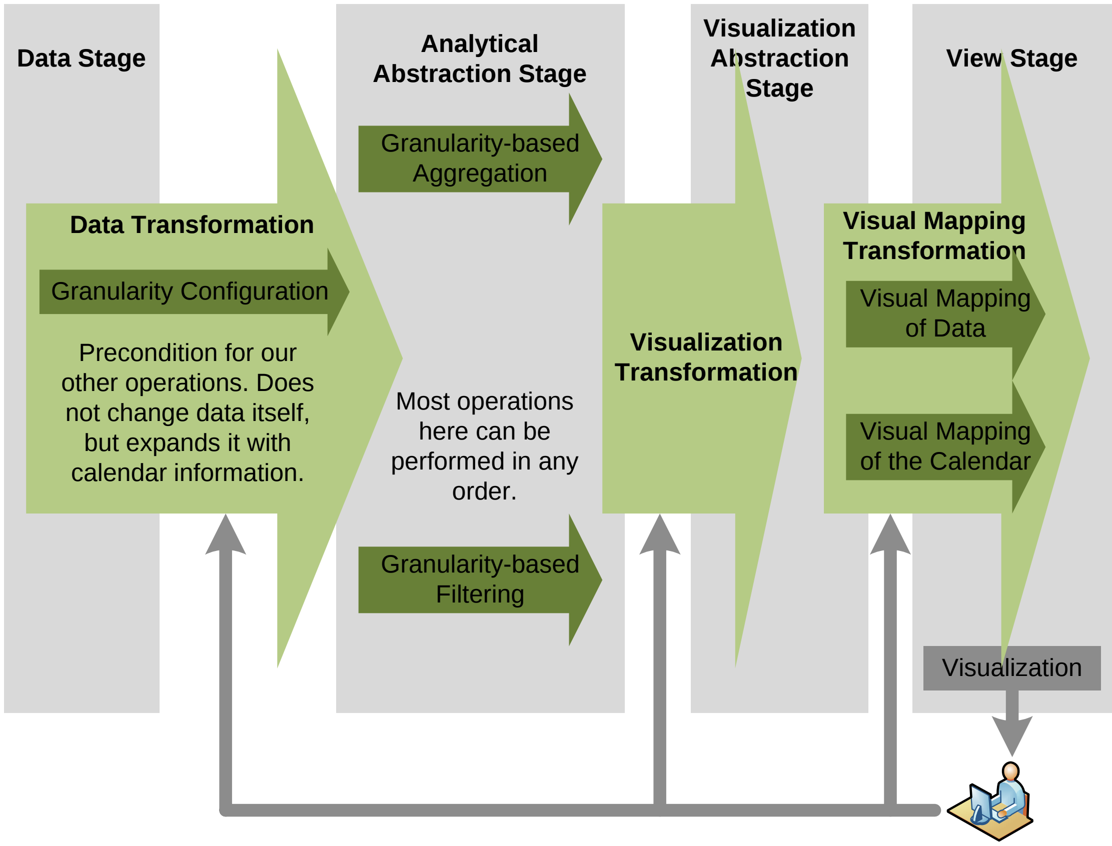


Figure 3.4: The Visualization Process. Operations are allocated to certain stages or transformations. This visualization is provided with more details in the poster attached to this dissertation.

3.3.1 Visualization Process according to the Structure of Time

Figure 3.5 shows the special case of the visualization process with the structure of time involved as a UML diagram. Before performing any other operations, granularity information for the dataset has to be defined. This part will be explained in detail in Subsection 3.3.2, followed by operations that are part of the Analytical Abstraction Stage, like Aggregation Operations or Filtering Operations, in Subsection 3.3.3. According to the process, those operations can be done in any order once the granularities needed for them are configured. In special cases, operations need others to be performed as precondition, but those cases cannot be modelled in a general process model. Before Visual Mapping transformations can be performed, a visualization abstraction must be chosen. The Visual Mapping in turn has to be performed before the View Stage is reached. Any other operations than Visual Mapping operations and operations at View Stage have to be performed before that. Visual Mapping in general has already been explained in Section 3.1. In the Subsection 3.3.6, we will explain it more detailed in the context of this framework. There, we will also account for the fact that several different combinations of aggregation and filtering operations can be used for one visualization. To limit the complexity of the UML diagram, we have only included the steps which are relevant in the context of applying the structure of time. User interaction can be made possible by the visualization at all of these stages, as shown by the blue part of Figure 3.5. If the user interactively modifies the parametrization for a stage, the control flow of an implementation has to go through all stages from that point again.

In many visualizations, visual mapping is done only once. In some visualizations though, more than on Visualization Abstraction is chosen in parallel, with different parameterizations regarding the Visual Mapping. Those result in two or more visualizations that are later combined on a visual level. An examples for this area is Multi-scale visualization by Shimabukuro et al. [2004]. In the next subsections, we describe the parts of the Visualization Process as shown in Figure 3.5.

3.3.2 Data Transformation Operations

Granularity Configuration

To use the calendar aspect based on granularities as described in Subsection 2.2.3, we need an operator that defines information about the granularities that are to be used by later operators. The granularity configuration operator is the precondition for all operators that make use of the structures of time.

Many real-world visualizations that support the structures of time are published by their creators with fixed calendar systems. Most of them can be enhanced by incorporating a granularity configuration operation, making them more flexible and powerful. Thereby, calendar systems can either use predefined or user-defined granularities.

A UML diagram showing the granularity configuration operator in detail is presented in Figure 3.6. In addition to the grouping of the granularities, a rule for granule label generation has to be defined for each of them. These labels can either be absolute or in the context of another granularity. In real-world scenarios, years are usually given as absolute values (e.g., 2009), while days are usually given in the context of months or weeks (e.g., 31st or Tuesday). For visualization systems, the labels have to be enumer-

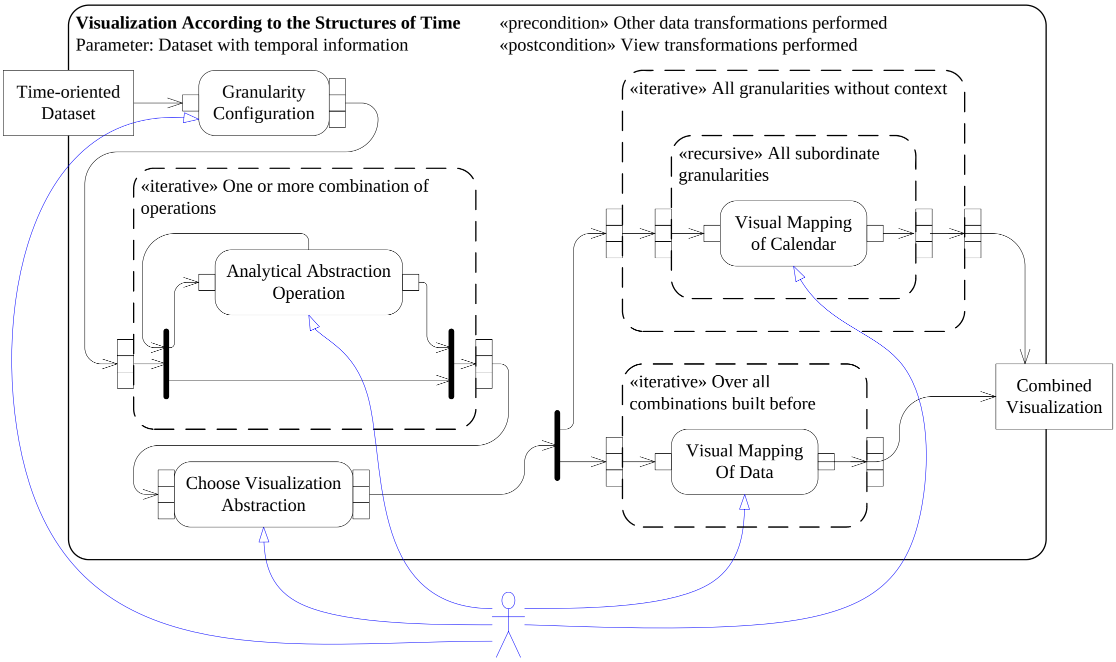


Figure 3.5: The Visualization Process as a UML 2 Diagram. The partial activities (or operators, from a point-of-view similar to the one of Chi et al.) are described in detail in Figures 3.6–3.12.

able and a numerical origin has to be defined. If they are cyclic, the origin defines the beginning of a cycle. For example, a week can start on Sunday or on Monday. A day can start at midnight or, for example, at 6am (start of a shift). If there is no superordinate granularity, the origin is a point like the first year of a calendar.

We consider data with granularity information an analytical abstraction (compare Subsection 3.3.3) although it is still compatible to the original data. Therefore, data stage operators can usually still be used on this kind of data after granularity configuration.

3.3.3 Analytical Abstraction Stage Operations

Operations at Analytical Abstraction Stage can have a wide range of applications. We exemplary present those kinds of operations regarding the stucture of time we need to describe the visualizations presented in Chapter 9 with our framework.

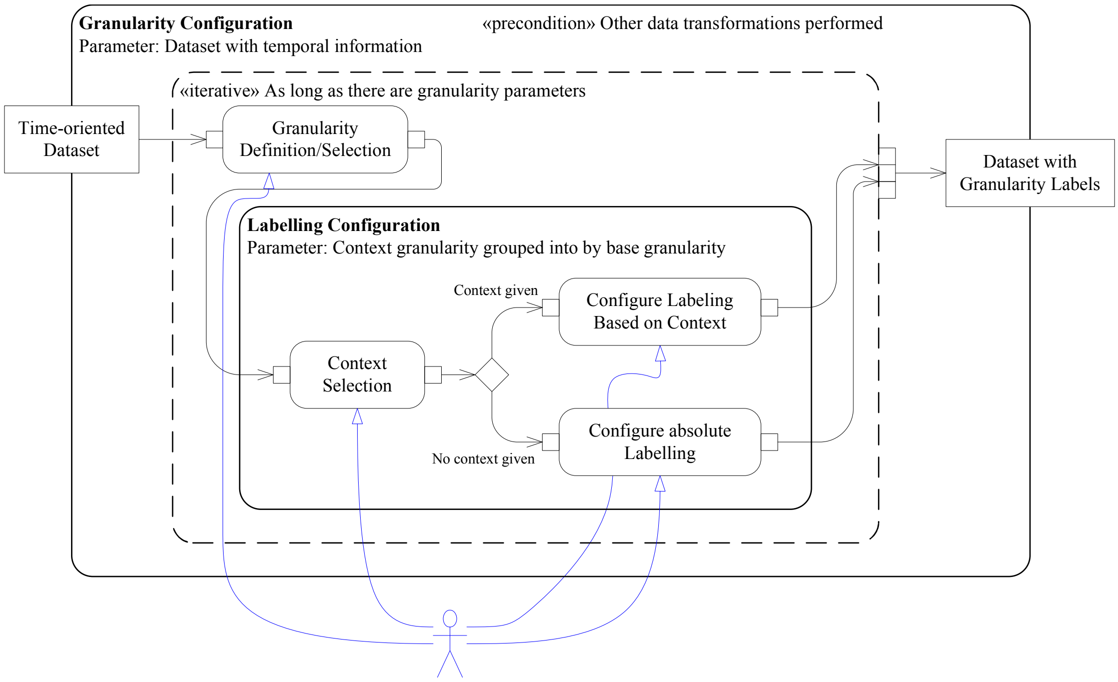


Figure 3.6: The Granularity Configuration Operator as a UML 2 Diagram.

Aggregation Operations

Aggregation is applied for several tasks. A particularly important one is the change of the discrete time domain. This task can be modelled using the the structures of time like this:

1. A target granularity is chosen.
2. All data elements that are part of the same granule in this granularity are combined and aggregated.
3. The result of each aggregation is a new data element. Possible aggregation operations are sum, mean value, and others.

If the target granularity is cyclic, granules with the same label in different cycles are treated as different granules. Therefore, the new dataset has to contain a sufficient number of cycles to contain all resulting data elements. This approach can be expanded for other possible applications. The basic method combines all granularites that are finer than the chosen granularity and ignores all granularities that are coarser. The same principle can be applied using different combinations. For example, let's assume a calendar system that consists of hours, days, and weeks. It is possible to aggregate all data that is part of the same day, explicitly combining data from different weeks but not from different hours. Therefore, intraday plots for the seven days of week that show average occurences for each type of day can be generated. Like aggregation for changing the scale, this variant is also common in real-world scenarios (see Chapter 9), where it is usually generated by building pivot tables first. Instead of using granularity labels as a basis for aggregation, it is also possible to use regular data values. Therefore, the building of classic pivot tables based on the data is only a subset of our definition of aggregation operations.

Figure 3.7 shows a UML 2 diagram for aggregation operations. First, a set of granularities is chosen. Then, the data elements, where all labels inside the chosen set of granularities are identical are grouped. Inside these groups, any kind of aggregation method can be performed. Examples are mean value, median value, sum, and virtually every statistical method. Finding a suitable method can be difficult for irregular granularities, like weeks and months. This problem also arises when aggregation operations do not take into account the structures of time. By considering them, a more organized view on the problem is possible. Single data elements can belong to more than one group, and the number of data elements in one group can be irregular in respect to the labels for individual granularities. The complexity of possible solutions is therefore rather reduced than increased by the structures of time.

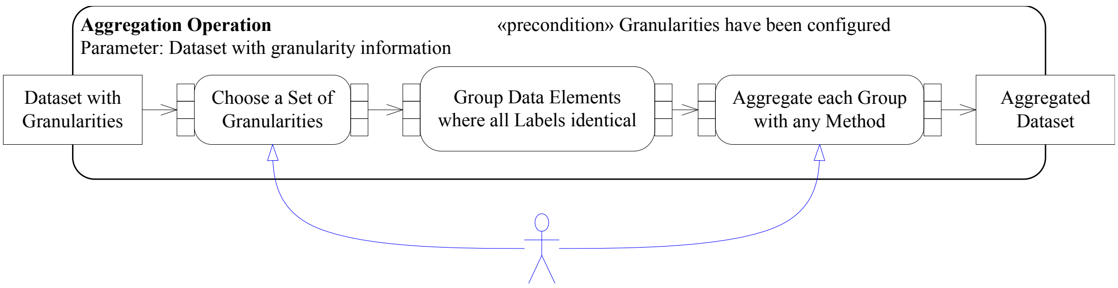


Figure 3.7: The Granularity-based Aggregation Operator as a UML 2 Diagram.

As for all operations that make use of the structure of time, granularities have to be configured to perform this kind of aggregation.

Rasterization Operations

Rasterization operations are a special kind of aggregation operations. The difference is that after performing a rasterization operation, only the dataset calculated by that operation may be used for further operations. Access to the original dataset is forbidden. Modelling rasterization operations while showing their difference to aggregation operations would require UML class diagrams instead of UML activity diagrams. In the diagrams we show here, rasterization operations look identical to aggregation operations. Therefore, we provide no separate diagram for rasterization operations.

Filtering Operations

Filtering in general can not only be used to limit visualized ranges, but also for other tasks like removal of erroneous data. By performing it at the analytical abstraction stage, it also affects other operations at this stages instead of only the final visualization.

Filtering Operations are performed as shown in Figure 3.8. First, a granularity has to be chosen. For this granularity, conditions for the labels have to be defined. These conditions can be exact values or value ranges (for example, Monday to Wednesday). All data elements that do not fit the conditions are removed from further processing. For performing this kind of filtering, granularities have to be configured. Filtering over more than one granularity can be modelled by performing several filtering operations on the same dataset.

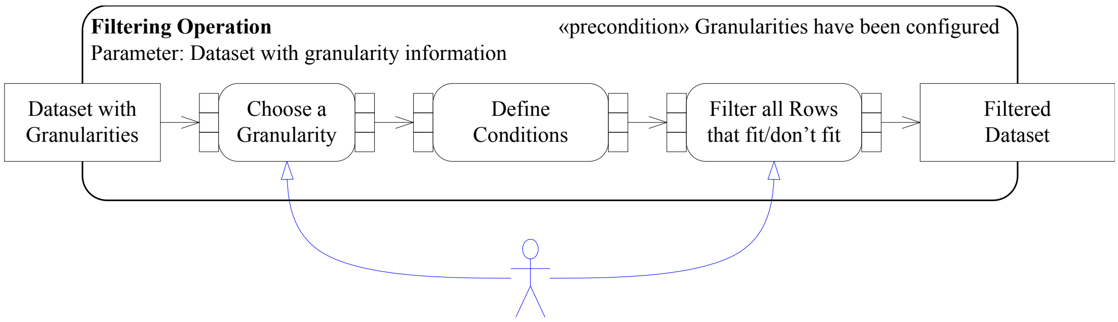


Figure 3.8: The Granularity-based Filtering Operator as a UML 2 Diagram.

For a logical build-up of visualization systems, it is helpful to account for important filter granularities in granularity definition already. If filtering for a range “Monday–Friday” is an important aspect of a visualization, using a granularity “business week” instead of “week”, hence ignoring the weekends, might be superior to filtering them out with an extra operation.

Calculation Operations

It is possible for visualizations to perform calculations on the data based on the structure of time. In Chapter 9, we will present a visualization that does not show absolute values. Instead, the visualizied values for each week are the differences between that week and the week before.

To get such values, calculation operations need to be performed at the Analytical Abstraction step. Calculation Operations are performed as shown in Figure 3.9. First, a granularity has to be chosen. For this granularity, specific granules have to be chosen. That step is usually performed in a relative way, like “each week and the week before”. Then, a calculation method, like difference, or quotient, is chosen. According to those parameters, the calculation is performed. The result is a dataset with an additional data attribute containing the result of the calculation.

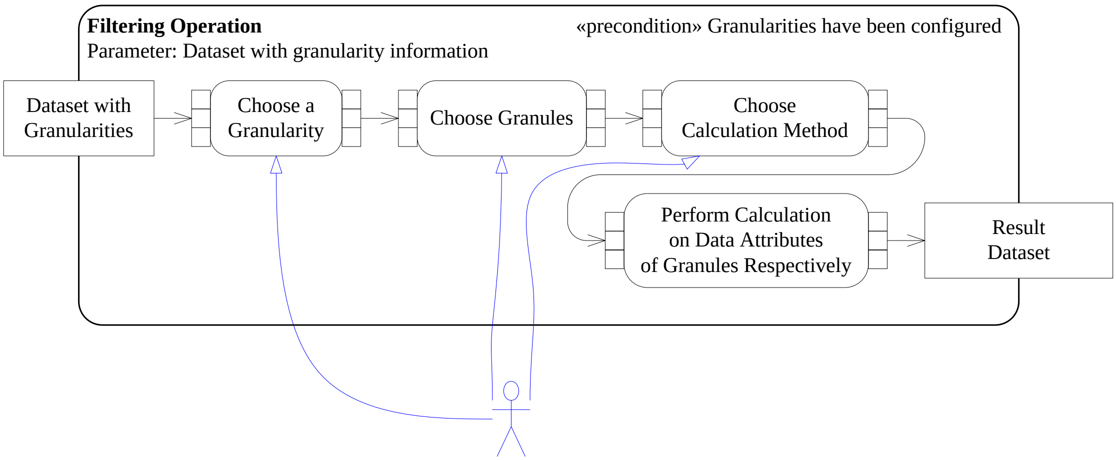


Figure 3.9: The Granularity-based Calculation Operator as a UML 2 Diagram.

3.3.4 Visualization Transformation Operations

In the framework by Chi and Riedl [1998], visualization transformations are mainly defined for aspects of non-time-oriented nature. For example, the multi-dimensional scaling of vectors or operations on the connectivity in networks. We do not need operations located at the visualization transformations to modify time-oriented aspects of our data.

3.3.5 Visualization Abstraction Operations

Operations at the visualization abstration stage make a preselection for the visualization abstraction. Data can have a direct meaning that can be mapped to a visualization, for example geo-spatial data. However, in InfoVis, much data is abstract, for example monetary values. In those cases it is necessary to decide for a general way of mapping the abstract data to a visualization. At visual abstraction stage, only a basic decision, like map, connectivity, or data values is made. The actual look of the visualization, like plot, or tree map, is done during visual mapping.

3.3.6 Visual Mapping Transformation Operations

To model the concept of visual variables as introduced by Bertin Bertin [1967], and explained in Section 3.1, in context of the structures of time, a distinction has to be made:

Visual Mapping of Data is done based on data elements. Granularity labels and data values can likewise be mapped. An activity diagram for this kind of visual mapping is depicted in Figure 3.10. The data that is mapped can be time-oriented or non-time-oriented aspects, in most visualizations both are mapped to different visual variables.

Visual Mapping of the Calendar is used to show information about the granularities themselves and their dependencies. This information is necessary for understanding the data. An activity diagram for this kind of visual mapping is depicted in Figure 3.12.

Operations for Visual Mapping of Data

For example, a line plot is used to show data values over days. The supporting points of the line plot are generated by mapping individual values of the data elements to visual variables. When taking into account the structures of time, the granularity labels can be used as values. In our example, the day is mapped on the horizontal axis and the data value is mapped on the vertical axis.

To perform this task, a visualization has to be chosen. In the context of Figure 3.5, this can be done several times, choosing the same visualization, a different visualization of the same type, or a completely different kind of visualization. The resulting visualizations have to be combined after that. Several view transformations are possible for this task, including multiple views and different overlay variants.

As one of the most simple examples we consider a classic line-plot of time-oriented data (see Figure 3.11). There are no granularities defined, so there is only the discrete time domain which we use like a granularity. There is no context defined, so the labels are based on absolute values. These label values are mapped to the position on the horizontal axis (a). The value according to each data-point is mapped to position on the vertical axis (b). By doing so, a number of support points is generated. These points are connected by lines (c), in their temporal order.

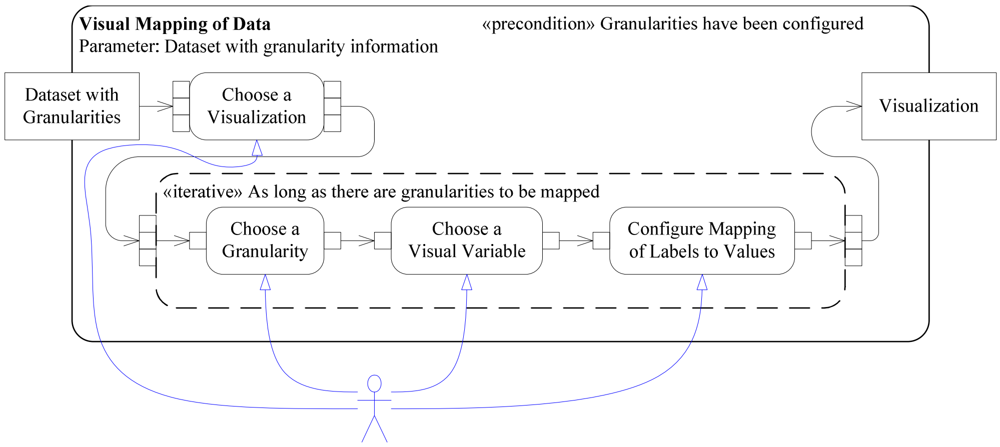


Figure 3.10: The Visual Mapping Operator for Data as a UML 2 Diagram. The difference to Figure 3.12 is explained in Subsection 3.3.6.

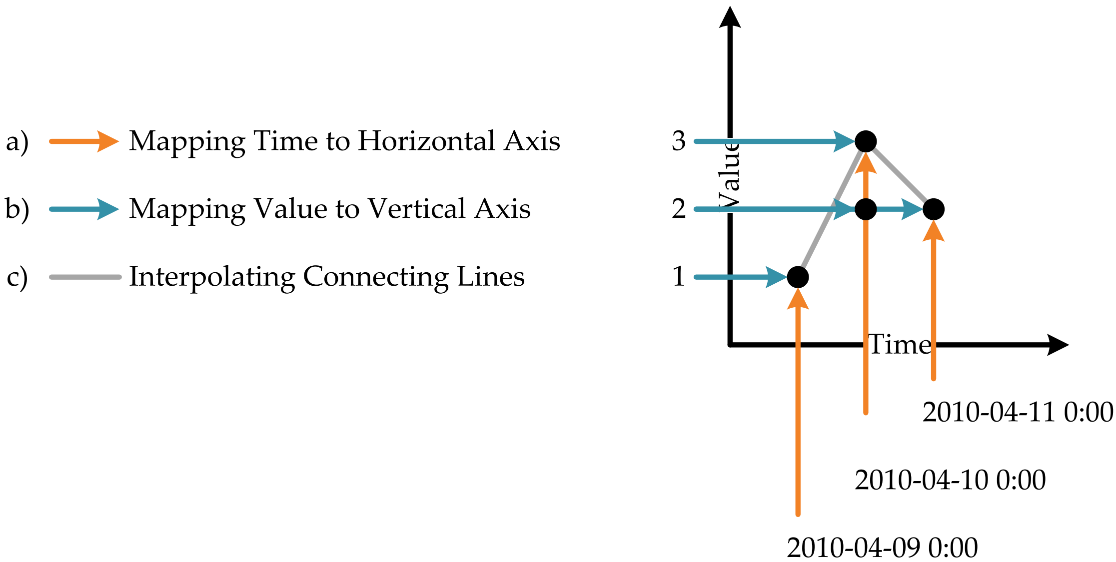


Figure 3.11: Examplary Visual Mapping in a Line Plot. The time label are mapped to the horizontal axis. The data values at those time labels are mapped to the vertical axis.

Operations for Visual Mapping of the Calendar

For the axis labels of our line plot example, information about the calendar system is advantageous. Therefore, labels can be given in a format like "MM/DD/YYYY", mapping the separate granularity labels on parts of the axis label string. For labels that are exactly at the position of data elements, this can still be done by visual mapping of data. For labels that are unrelated to the data, a different kind of operation is needed. We define visual mapping of the calendar by mapping possible combinations of granularity labels based on the calendar information to visual variables. In rare cases, visualizations can even be important for users without showing a specific dataset at all. The best example is a calendar sheet, which can be used for tasks like finding the day of week that corresponds to a certain day of month.

This kind of visual mapping is similar to the visual mapping of data, we can define the operator in a more straightforward way (see Figure 3.12). The level of multiple executions is not missing, though, but shifted to Figure 3.5.

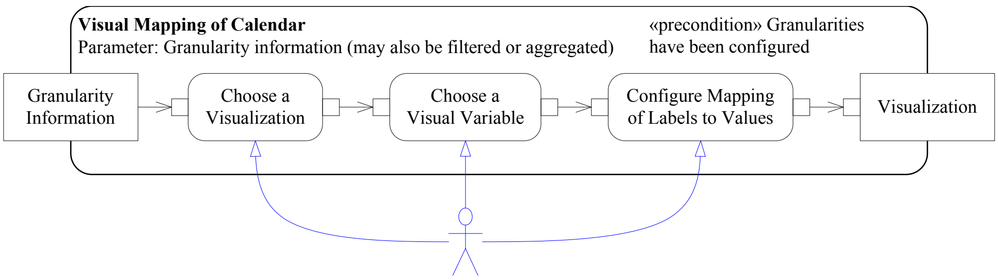


Figure 3.12: The Visual Mapping Operator for the Calendar as a UML 2 Diagram. The difference to Figure 3.10 is explained in Subsection 3.3.6.

The difference between granularity data visualized as part of the data visual mapping or the granularity visual mapping is the level of detail. During data visual mapping, only granularity labels that correspond with data points can be visualized. However, it is still necessary to allow for visualizing granularities in that way, because they often need to be in conjunction with the data values.

While our framework is focused on the structures of time, it is also capable of modelling typical visual mappings that are not related to time. As an example, we describe the modelling of value labels: The range of value data and sensible intermediate values (for example, "minimum value, maximum value, quartiles, median", or "minimum value, maximum value, half of mean, threetimes half of mean, mean" or "0, 10, 20, 30, 40") are calculated by using a combination of filtering and aggregation operations. Data visual mapping can be applied to map these values to the visual variable of label plotting.

3.4 Summary

In this chapter, we presented the concept of Visual Mapping and described two process-oriented frameworks that apply Visual Mapping and other steps in order to generate interactive visualizations. Based on those frameworks as well as our definitions for time-oriented data and the structure of time from Chapter 2, we introduced our own process-oriented framework that is more focused on our tasks at hand than the general frameworks, but by being compatible to them, it provides seamless expandibility and connections to other areas of InfoVis. Our framework will be applied to state-of-the-art as well as novel visualizations in Parts II and III of this thesis.

—The greatest challenge to any thinker is stating the problem in a way that will allow a solution.

Bertrand Russell

4 Special Challenges and Prospects when Dealing with the Structure of Time

The visualization process described in Chapter 3 is no automated solution. It does not only require user input, to work best, it has to be adapted to the data and tasks at hand by visualization experts. In this chapter, we will describe which challenges these experts have to overcome when dealing with the structure of time, but also, how they can use the structure of time to ease their task of providing the best interactive visualizations possible for users. These challenges and prospects result from visualization theory state of the art as well as our own insights from the project **DisCō**.

4.1 Choosing Granularities

Mackinlay [1986] describes terms to dispute how well a given visualization fulfills its requirements:

"Expressiveness criteria identify graphical languages that express the desired information. [. . .] A set of facts is expressible in a language if it contains a sentence that (1) encodes all the facts in the set, (2) encodes only the facts in the set."

"Effectiveness criteria identify which of these graphical languages [that are expressive], in a given situation, is the most effective at exploiting the capabilities of the output medium and the human visual system."

For a visualization that employs the structure of time, choosing the granularities, adequately is of great importance to maximize expressiveness and effectiveness.

In this context, several aspects of granularity choice have conflicting effects on the visualization:

Number of Elements at Finest Granularity The more data contained in the visualization, the more data can theoretically be absorbed by the user. Using methods of aggregation instead can easily hide important facts (an example from a real-world scenario is given in Section 12.1). On the other hand, a higher number of data points can lead to visual clutter [Rosenholtz et al., 2005]. The finest granularity can be finer if the visualizations provides a means of shifting the view, like scrolling.

Length of Cycle in Granules for cyclic granularities. In some visualizations, it can be a problem when cycle lengths differ too strongly. For example, when granularities are applied for recursive pattern arrangement (see Subsection 9.1.5), the fact that two dimensions are used limits the reasonable possibilities for spatial spread to proportions that are similar to the available display space.

Context from the Data vs. Context from the User Even though many time-oriented aspects in data stem indirectly from social conceptions, there are many applications (like, weather), where this is not true. Even in datasets from business applications, there might be differences. For example, our user analysis (see Chapter 5) has, as a side effect, provided the (somewhat surprising) knowledge that in business data the granularity month of year does not have much impact. At the same time, month of year is a rather important granularity for human conception of time and the ability to locate a day in a year. Therefore, there exists a tradeoff between granularities that support the detection of patterns in the data and granularities that guide users.

4.2 Irregularities and Gaps

Granularities are often irregular. For example, the most irregular granularity from customary social time is the month. Irregular granularities can create problems—for example, Recursive Pattern Arrangement (see Subsection 9.1.5) produces gaps in conjunction with irregular granularities. As already mentioned in Section 4.1, business data analysts despise this kind of granularities, like month, for that reason. Consequently, whenever irregular granularities provide an additional benefit, that benefit and the drawback of having the irregularities have to be weightet, while the visualization process has to be optimized to minimize the drawback.

It is also possible to intentionally use granularities that introduce some kind of irregularity. For example, the introduction of a business-week-granularity enables visualization designers to easily model the difference between workdays and holidays. Even though this is another irregularity the visualization has to cope with, the effort is usually worth it (see Chapter 12 for an example).

4.3 The Scale and Structure of Granularities

Granularities are defined being inherently connected to the calendar aspect. However, when dealing with them semantically in a visualization, the scale and structure aspect are also an important part of the decision space. Our user analysis process (see Chapter 5) revealed that, for example, analysts can spend much time contemplating whether days of week are nominal or ordinal.

4.3.1 Ordinal vs. Discrete Granularities

Dealing with the scale of time-oriented data is a process difficult to automate. The range of possible visual variables can roughly be divided into those that are suited better for nominal data and those that are suited better for ordinal data. When designing a visualization and specifying visual mappings, decisions have to be made.

While in theory it is possible to automate these decisions, doing so already falls into the topic of automated visualization layout. We propose sensible defaults for our visualizations and encourage implementations to provide as much configurability by users as possible, but do not deepen research on automated visualization as envisioned by Mackinlay [1986].

4.3.2 Start and End Granules

Start and end times of granularities can be considered a part of the scale aspect as well as the structure aspect. For discrete and cyclic granularities, it is necessary to define which granule is the first one of a new cycle. For example, day of week is a typical granularity that is defined differently in various social contexts. However, the first granularity does not even need to agree with common social defaults—a day can start at any time, e.g., at the time a shift change occurs. Moreover, it can end at any time, even leaving an intentional gap as described in Section 4.2.

Beside the obvious choice of using start and end granules as defined by social conventions (which is usually a sensible default), it is possible to use automated Data Mining methods to define them. However, defining them is another task where providing users a possibility to reconfigure the visualization can be very helpful.

4.4 Viewing Viewpoints: Logical or Temporal?

Work from temporal reasoning regarding multiple points of view in time-oriented data has been focusing on points of view by different sources. An example are different witnesses giving a time-oriented report about a crime. Further things that have been researched so far are decision trees and data elements with multiple timestamps, for example from transactional databases.

In the context of the data we had to work most recently, another variant was showed up prominently: For each point in time, defined by one timestamp, there are several data elements, distinguished by another timestamp. The ususal source for such data are forecasting algorithms. An example is shown in Table 4.1. In this example, an increase in demand results in a decreasing reserve. The forecast algorithm adapts and calculates new values. Between February 15th and February 21st, a new shipment arrives, resulting in new values again.

Table 4.1: Remaining Milk Packets at Different Points in Time Forecasted from Different Points in Time

Date of Occurence	**Number of Remaining Packets**	**Date of Forecast**
01.03.2010	8	01.02.2010
02.03.2010	6	01.02.2010
03.03.2010	4	01.02.2010
04.03.2010	2	01.02.2010
05.03.2010	0	01.02.2010
01.03.2010	6	08.02.2010
02.03.2010	3	08.02.2010
03.03.2010	1	08.02.2010
04.03.2010	0	08.02.2010
05.03.2010	0	08.02.2010
01.03.2010	1	15.02.2010
02.03.2010	0	15.02.2010
03.03.2010	0	15.02.2010
04.03.2010	0	15.02.2010
05.03.2010	0	15.02.2010
01.03.2010	34	21.02.2010
02.03.2010	31	21.02.2010
03.03.2010	28	21.02.2010
04.03.2010	25	21.02.2010
05.03.2010	22	21.02.2010

In the context of Temporal Databases, the same methods of using multiple timestamps can be applied. When visualizing the data, novel methods are needed to give users insights that reflect the actual context.

4.5 Summary

In this chapter, we have shown several important aspects that have come up as relevant when dealing with the structure of time. Those aspects are (1) choosing the right granularities, (2) dealing with the fact that they might be irregular, and (3) the fact that granule labels can be interpreted differently depending on the task but also on the visualization used. Task and visualization have to fit together for the best results. Furthermore, we introduced the kind of multiple views we had to deal with in our work, a variant less prominently researched in literature than other kinds.

—Time is money.

Benjamin Franklin, Advice to Young Tradesmen (1748)

5 User Analysis

The users targeted in the **DisCō** project handle application scenarios of data analysis in different industrial or service sectors (e.g., transportation, call centers, retail, health care), and the public sector. The project contained a participatory design process that started with initial interviews to assess their current task situation.

5.1 User Characteristics and Tasks

The participants described their work as business consultants, human resources planners, and controllers. Their tasks are to analyze, plan, and forecast personnel demand and to evaluate organizational interventions in this field. In addition, they analyze time-related personnel data for different group or locations in conjunction with metadata and various management ratios. The users described the application scenarios of data analysis in different industries, like transport and logistics, service industries, retail, health care, and the public sector.

Users reported that they are often confronted with ill-defined problems [Colman, 2001]. Although they are aware that they may have a problem or opportunity, they are frequently unable to nail it down. Hence, they usually start the data analysis process with an exploratory phase. Similar to the workflow in exploratory data analysis in scientific domains [Shneiderman, 2001, Pirolli and Card, 2005, Springmeyer et al., 1992, Trafton et al., 2000], the workflow in the field of human resources planning lacks linearity. It can rather be described as an iterative or circular process. During their data analysis process, users reported several cases where it was necessary to go back one or more steps and change, normalise, or filter data. On the data level, frequently reported cases are inconsistencies in data like missing values or implausible time shifts and missing consistency with other variables. These validation processes are supported by an extensive toolset and are reported as one of the most time consuming parts of the analysis. This toolset is mainly woven around the identification of temporal patterns that are influenced by different time granularities. An example might illustrate the nature of our users' tasks:

> A hospital wants to better align its personnel plans with customer requirements. To analyze data from this hospital our temporal analysts first have to understand and take into account two factors: (1) the internal, hospital-defined shifts, (2) the customers' demands. Then they have to find out how both are driven by structures of times—business days and seasons—and how special events such as holidays can influence the given data.

On the user level, new insights and deeper understanding of data and their relations, adapted frames of understanding [Pirolli and Card, 2005, Russell et al., 1993], and newly developed hypotheses (derived from previous results) often generate the necessity for recalculation. In the hospital example, our users have to (1) gain an overview of the data set, (2) identify relevant and define specific time granularities (e.g., one business day can last from 6am to 6pm, from midnight to midnight, etc.), and (3) find anomalies and relevant patterns, trends, and relations within this data set. Testing statistical hypotheses is not of high priority. Due to narrow time boundaries, results usually have to be gathered within minutes or hours, data analyses that span over weeks or months rarely occur. Visualizations are not seen as a "by-product" of the analysis (as stated by Springmeyer et al. [1992]), but as a tool supporting analysis and presentation of results. Users favour visualizations which can be presented to their management board or CEOs with little additional effort of editing or with automatic support for creating new visualizations to illustrate their insights and findings. Therefore, one of the requirements for the software application is intuitive visualization of highly complex multivariate, time-oriented data—so that it can be understood without extensive training lessons.

5.2 Methods of Measuring

To test the support of the human reasoning process, various novel methods have been developed over the last years (mostly at the Beliv conference, e.g., [Rester et al., 2006]). Classical benchmarking metrics, like efficiency and efficacy or time and error rate, turned out to be of limited use for the improvement of information visualizations as they are typically task based and used in a highly standardised experimental setting. Therefore, tasks have to be compact and predefined. Definitive, unambiguous, and distinctive answers are forced through the experimental setting and time constraints leave little room for deeper elaboration of the findings [North, 2006]. Due to the exploratory nature of knowledge discovery in Visual Analytics, new paradigms [Shneiderman and Plaisant, 2006, Plaisant et al., 2008] for testing "beyond time and errors" [Bertini et al., 2007, Ellis and Dix, 2006] were promoted to fill these explanatory gaps. One of these metrics is the qualitative and quantitative measurement of user reported insights [Saraiya et al., 2005] which was a primary method applied in our user studies.

The insights were counted by manual evaluation of analysis sessions with test users and the visualizations to be tested. A simple way of showing the results is providing a line plot of total insights over time. Another idea is comparing the total insights of different users or visualizations in a bar chart. However, much information that could be evaluated by our experienced team of usability experts could not be expressed in such visualizations. Therefore, we developed the *Relational Insight Organizer (RIO)* visualization [Smuc et al., 2008, 2009].

The basic layout of a RIO (see Figure 5.1) can be considered similar to a Gantt chart, but there are notable differences. The vertical space is divided in several insight classes. In our test environment, we chose the classes *Prior Knowledge* which covers insights users could not have from the visualization, but from their own domain knowledge, *Visualization Insights* which is relevant for users who are confronted with visualizations they do not know and understand at first, and *Data Insights* which are the ones that really count for users. Inside each class, vertical space can be used for parallel insights, but there is no mandatory vertical order. Each insight can be plotted as an instant, if it occured at a very short point in time, or as an interval, if the user developed it over a longer period. Arrows between insights can show cases when one insight leads to another.

The RIO visualization has been a central element in the evaluation [Smuc et al., 2008, 2009] of the GROOVE visualization [Lammarsch et al., 2009] (see Chapter 12).

5.3 Summary

In this chapter, we have provided an overview about the users we dealt with, especially during the project **DisCō**. We presented their occupation and, above all, the tasks they are dealing with. Furthermore, we gave a short insight about the methods we applied to perform our user studies. Still, this thesis can only give a glance at user analysis, because of its focus on interactive visualizations. The publications mentioned in this

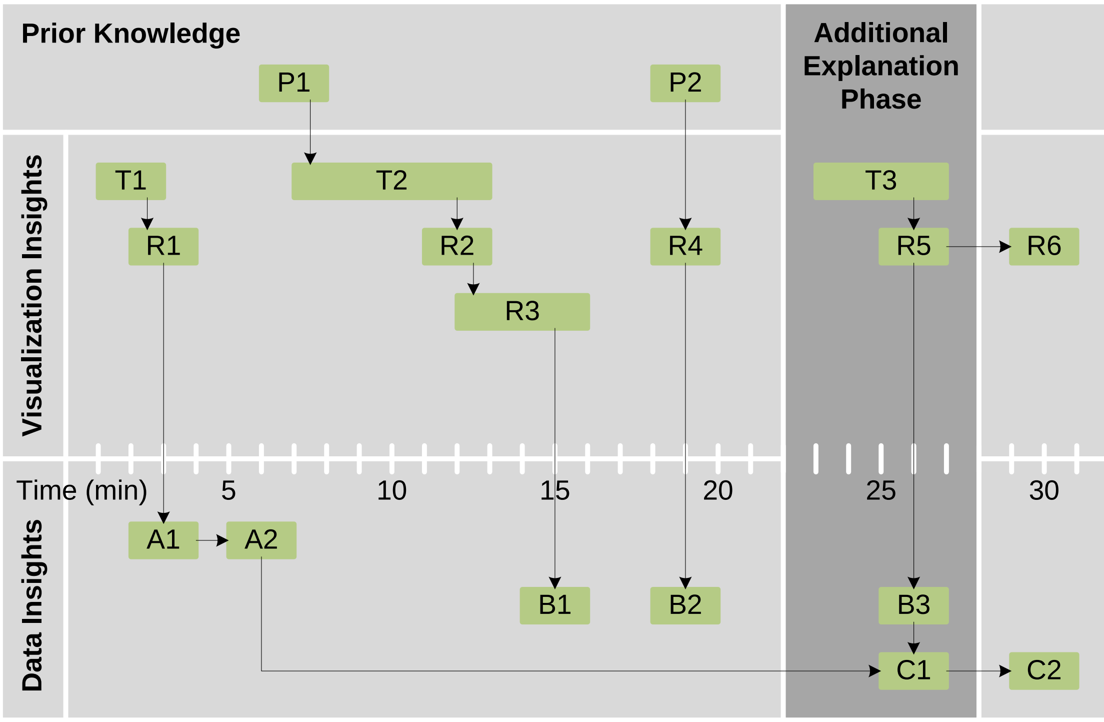


Figure 5.1: An Example of a RIO Visualization (adapted from [Smuc et al., 2008, 2009]). The Markers show when insights are gained. P: Insights based on a-priori knowledge. T: Insights about the visualization. R: Insights how to read the visualization. A, B, C: Insights about different aspects of the data.

chapter delve deeper into that area, as will upcoming publications from our project partners. In the context of this thesis, the reason to raise the topic is the fact that an understanding of the users and their tasks is necessary to understand the direction we proceeded in developing our novel visualizations.

—*It's not that I'm so smart, it's just that I stay with problems longer.*

Albert Einstein

6 Problem Analysis: Summary

In the first part, we defined time-oriented data, explained, how visual representation in general and with the focus of time-oriented data can be performed, and presented the users that are to be the primary beneficiaries of our work. The novelty we presented in this part is a process-oriented framework that enables us to describe the visualization process with a focus on the structure of time.

For the definition of time-oriented data in Chapter 2, we based our work on common existing definitions. Then we presented our own definition that is able to comprehend data as something that can be viewed under several aspects. Our focus, of course, are several time-oriented aspects that form the structure of time. The most important aspect for this thesis is the calendar aspect that has been modelled by Bettini et al. [2000] using the concept of granularities.

In Chapter 3 we used the granularity logic from Chapter 2 to define a novel process-oriented framework for the generation of visualizations. This framework is based on frameworks by Card et al. [1999] as well as by Chi and Riedl [1998] but has significant advances regarding the structure of time. The process of visual mapping is adapated based on work by Bertin [1983]. The resulting innovative framework is therefore able to describe visualizations that make use of the structure of time in a simple and efficient way. We can also easily transfer techniques from one visualization to new ones in a way not possible with frameworks that existed before.

Finally, the intended users that had been target of the **DisCō** project were described in Chapter 5. We presented their usual tasks as well as the needs that arise from them.

We also described, how the effectiveness of our new visualizations has been measured later.

The second part of this thesis will further the generation of a design-space for visualizations that employ the structure of time by providing a multi-level taxonomy based on the framework from Chapter 3. We will use this taxonomy to describe and compare several state-of-the-art visualizations that make use of the structure of time.

Part II

State of the Art

—You can't depend on your judgment when your imagination is out of focus.

Mark Twain

7 Scope and Focus

Over the last years, the state-oft-the-art in InfoVis has come far, but it also has strongly diversified into different solutions for different data domains. Chi and Riedl [1998] state that "the requirements in each domain are often dramatically different". Moreover, the most important domains, like time-oriented data, or geo-spatial data, can be classified further.

General views on visualization of time-oriented data have already been published, for example the one by Aigner et al. [2007b]. Comparisions of visualization tools to generate them have also been done, for example by Wohlfart et al. [2008]. So far, no state-of-the-art-analysis has focused solely on visualizations that make use of the structure of time. We will fill the gap in this part of the thesis.

Before the presentation of existing visualization techniques, we define a novel taxonomy of visualizations according to the structures of time in Chapter 8. This taxonomy is based on the process-oriented framework from Chapter 3 that serves as a "backbone" for the taxonomy levels. Therefore, we can use our taxonomy in a very broad range compared to several previous taxonomies.

In Chapter 9, the visualization techniques are described. The descriptions for each technique consists of a prosa part and a classification according to the taxonomy. Furthermore, we show the similiarities and differences between the visualization techniques with visualizations that are based on the taxonomy.

What we definitely do not intend with this state-of-the-art-analysis is presenting an overview of visualizations for time-oriented data (or even all domains covered by Info-

Vis) in general. Such overviews have already been done recently, and althogh it would be possible to do an update with newer techniques, we consider it more appropriate to present a more narrow focus of techniques, but with more details.

—An early step toward understanding any set of phenomena is to learn what kinds of things there are in the set—to develop a taxonomy.

Herbert Simon, The Sciences of the Artificial

8 Taxonomy

Simon's mantra from 1969 [Simon, 1969] (see above) is highly cited in papers about taxonomies for InfoVis and Visual Analytics. However a major reason for Simon to formulate that mantra was the fact that "this step has not yet been taken with respect to representations" [Simon, 1969]. Today, several well done and widely accepted taxonomies exist.

8.1 Existing Taxonomies for Time-oriented Data Models

While the Taxonomy we present in this Chapter is focused on Visualization, due to its multi-layered approach it also contains parts that are related to time-oriented data models. Therefore, we present taxonomies and collections of citations from that area.

Similar to the taxonomy in this thesis, Aigner et al. [2007b] organize visualizations for time-oriented data on the data and on the visualization level separately. Goralwalla et al. [1998] focus on the data models and define an object-oriented framework. They use their framework to describe several time-oriented data models.

Bertone [2009] lists numerous publications and gives detailed descriptions of the most important ones.

8.2 Existing Taxonomies for Information Visualization

Munzner [2008] demands synthesis at scales larger than a single paper for InfoVis. As a means, she presents a number of design levels used to classify papers. The levels are domain problem characterization, data/operation abstraction design, encoding/interaction technique design, and algorithm design. The visualization framework presented in Chapter 3 and the taxonomy presented here can be considered data/operation abstraction design. The examples from Chapters 9–12 can be considered encoding/interaction level.

A good overview of other taxonomies for InfoVis has been done by Chengzhi et al. [Chengzhi et al., 2003]. He classifies taxonomies according to their reliance on data type [Shneiderman, 1996, OLIVE, 1997, Last accessed 04/11/2009, Card et al., 1999], display mode [Keim and Kriegel, 1996], interaction style [Buja et al., 1996, Chuah and Roth], analytic task [Wehrend and Lewis, 1990, Zhou and Feiner, 1998], based model [Tory and Möller, 2004, Chi, 2000, Card and Mackinlay, 1997] or multiple factors [Keim, 2001, 2002, Tweedie, 1997, Pfitzner et al., 2003].

Most notable is the taxonomy by Chi [2000] because it is based on the process-oriented framework the author did before [Chi and Riedl, 1998] and therefore has some similarities to the taxonomy we will present in Section 8.3. Still, Chi does place some actitivies and thereby their distinction at a different level than we do. For example, many operations regarding scrolling and zooming, are placed at view stage, which is correct for purely geometric operations. Due to our focus on the structure of time, we are in favor of semantic operations at analytical abstraction stage.

Of the taxonomies mentioned here, none is focused on the structure of time, most of them do not even mention it. All taxonomies have in common that they describe one or more levels of the visualization process, but do not describe the process itself. The process is better described in a framework like the one we presented in Chapter 3.

Somtimes, those taxonomies are based on one single factor. Such an approach is easier to handle, but more difficult to use to differentiate all the intricacies of various visualizations. For taxonomies with several factors, it is difficult to define a logic that differentiates the factors themselves. We use the process-oriented framework from Chapter 3 like a backbone for several levels of taxonomies, each spanning one factor. Figure 8.1 shows how our framework spans taxonomy levels. We have our framework in the center as backbone. The blue levels to the left are general taxonomy levels, the orange levels to the right are the taxonmy levels regarding the structure of time we want to use.

We place the user at the beginning, because instead of emphasizing his/her involvement in the whole process, we like to state that a visualization system is usually used because a user has a task to fulfill.

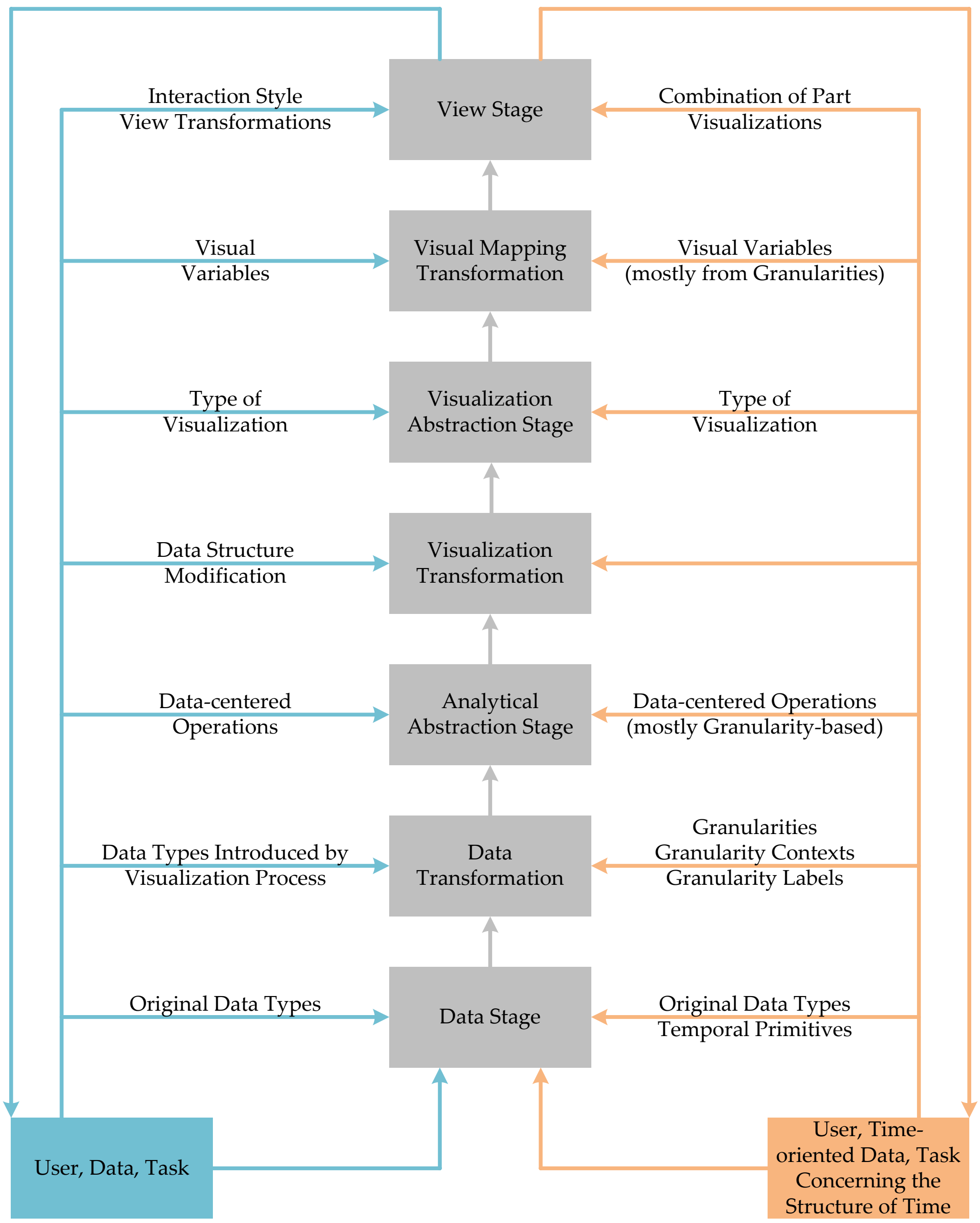


Figure 8.1: Set of Taxonomies that Describe Visualizations at Several Levels. The process-oriented framework from Chapter 3 serves as a backbone. The framework is shown in gray. The blue levels are what would be used for a general taxonomy. The orange levels are the detailed levell focused on the structure of time we use for our taxonomy.

8.3 Multi-level Taxonomy

Regarding the characteristics at the several levels, it is quite possible for a visualization to belong to more than one characteristic. An overview in form of a template is given in Table 8.1.

8.3.1 Tasks

Prior to the classification according to the visualization process, we present the tasks a visualization is most suited for. The choice of tasks has been done according to our practical experience with visualizations and users of visualizations over the course of the project **DisCō**. Therefore it is subjective to some degree. Moreover, any visualization enables users to perform any task at least to some degree. For example, users of the pixel-based GROOVE visualization (see Chapter 12) have been able to read values, but it was difficult for them.

8.3.2 Data Stage—Classification according to Data Type

As we are focusing on time-oriented data that is to be analyzed with the help of the structure of time, we only list data types that are of this context. If necessary, the classification according to data type can easily be expanded for use in another taxonomy.

Original Data Types

At the data stage we can only classify datasets that are the basis for visualizations based on the data types not related to time-oriented aspects because the data types of time-oriented aspects are not settled before granularity configuration.

Possible data types are defined in many publications, like the one from Aigner [2006]:

Nominal Data is based on a set of values that have no order and therefore does not fulfill the definitions of an algebraic field. An example are motion picture titles.

Ordinal Data is based on a set of values that do have an order bus still do not fulfill the definitions of an algebraic field. An example are motion picture ratings.

Discrete Data is based on a set of values that do fulfill the definitions of an algebraic field and are countable. An example are motion picture years of publication.

Continous Data is based on a set of values that do fulfill the definitions of an algebraic field and are not countable. This data type is of less importance for us, because it cannot be measured exactly in real-world data. Therefore, all data in

Table 8.1: A Template of the Taxonomy in Section 8.3. The individual attributes are explained over that section.

Tasks: Detect cycles; interpret cycles; detect trends; detect patterns; classify usual values; detect irregularities; detect regular structures more complex than cycles; compare granules; compare granule combinations; compare granules on statistical level; read values

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Nominal data/ordinal data/discrete data
	Primitives	Instant-based/interval-based/span-based
Data Transformation	Granularities	Any
	Granularity Contexts	For all granularities, either another granularity or total
	Granularity Labels	For example, numbers, letters, or words
Analytical Abstraction Stage	Rasterization/ Aggregation/Filtering/ Calculation of Difference	Based on granularities
	Other operations (for example, from Chi and Riedl [1998], Chi [2000])	Anything
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	One or more: spatial data/connectivity data/abstract value data
Visual Mapping Transformation	Data value distribution	Axes, legends
	For example, position, color, or symbols	Granularities or data attributes
View Stage	Combination	As needed by the decisions from visualization abstraction stage
	Other operations (for example, from Chi and Riedl [1998], Chi [2000])	For example, zooming or panning, omitted in our related work listing if not related to the structure of time

computer systems can be considered discrete, even if that might only be an approximation.

Sometimes *binary* data is considered a data type, but depending on the context that data can also be considered nominal or ordinal.

Primitives

Of the time-oriented aspects, the primitives aspect can already be considered at the data stage.

Instant-based Data contains a data values that are allocated to an instant, a single point in time. In discrete data, this is equivalent to a chronon.

Interval-based Data contains data values that are allocated to an anchored period of time. In discrete data, this is usually given as a start-chronon and an end-chronon, spanning all chronons in-between.

Span-based Data is occurs less often. Its data values are allocated to unanchored periods of time. In discrete data, those can be given as a number of chronons.

For many datasets, making a distinction between instant-based data and interval-based data is very hard to make, as any chronon can be divided further on a mathematical level—even one Planck-time which is the shortest timespan governed by the laws of physics. However, in our view of the structure of time, when a discrete scale is defined, we consider the chronons as being not dividable and define a single chronon as an instant, even though it has a duration.

8.3.3 Data Transformation

The data transformation step contains the important operation of granularity configuration. Therefore, on this level it is possible to classify by listing

Granularities that are chosen,

Granularity Contexts for all granularities—a granularity context determines the cycle of a granularity, for example a day can be used as a day of week or as a day of month,

Granularity Label Formatting for all granularities—a certain day of week can, for example, be given as Sunday, 7, 1, or 0.

8.3.4 Analytical Abstraction Stage

Rasterization

Rasterization can be performed by using the structure of time. A granularity is used as scale. Finer granularities become inaccessible by following operations. To change the scale, an aggregation operation is used. This operations can be very complex if the data points do not fit into the new scale, for example because one of the granularities is not based on the other one, but on a common origin granularity. Therefore, several classifications are possible:

Rasterization at Analytical Abstraction Stage with the possible characteristics of none (unrastered data) or any granularity.

Method of Rasterization is mathematically still an aggregation. The difference lies in the fact that when rasterized, finer levels become "hidden". Example methods are mean, median, or sum.

Aggregation

Aggregation is a more general application of rasterization. The same actions can be performed, but the result is not used as the only data basis for later operations. Instead, the data basis is enhanced by the result of an aggregation operations and follow-up operations can resort to several levels. Characteristics are:

Granularity of which all granules are targets of one aggregation bin each.

Method of Aggregation for example mean, median, or sum.

Filtering

Filtering of granularities can be used to model semantic user interactions. Few visualizations from state of the art do this explicitely, but most of them are suitable for introducing it as an additional possibility.

Granularity of which only a defined set of granules is taken for the following operations.

Granules in the form of granule label lists or ranges. The parameter determines which granules are to be used (or ignored) for further calculations.

Calculation of Difference

Calculation of difference can be performed to provide a meta-value that describes insides about the change of something. The possibility of calculating a difference between different data attributes is not included here because it does not have an inherent context related to the structure of time. Important, however, is regressive development of one data value in the context of time. The characteristics are:

Granularity that determines which two granules are used for the calculation.

Aggregation State also is a granularity. It determines which data exactly is used for the calculation.

Example: When the value on a Monday is substracted from the value on the Monday of the next week, the granularity used is week, but the aggregation state is day.

8.3.5 Visualization Transformation

We omit the visualization transformations from the taxonomy because it would make the design space more complex, but provide no additional information how a visualizations deals with the structure of time.

8.3.6 Visualization Abstraction Stage

As already mentioned in Chapter 3, our framework does not place the final decision about the look of the visualization inside Visualization Abstraction Stage. Therefore, the taxonomy can only make a partial characterization. This includes

Multiplicity with the values of a singular visualization or several parallel or integrated visualizations, and

General type of each visualization, with the possibilities of spatial data (for example, maps), connectivity data (for example, trees), and abstract value data (for example, line plots).

8.3.7 Visual Mapping Transformation

At the level of Visual Mapping Transformation, there are two different possible approaches for classification:

1. The data aspects are the characteristics and the visual variables used are the values.

2. The visual variables used are the characteristics and the data aspects are the values.

We are using the latter approach because the amount of possible visual variables, while being huge, is still smaller then the amount of possible data aspects. Furthermore, it is suitable to make a coarse characterization of visualizations by the visual variable they are using possible.

Data Value Distribution is shown in many visualizations. Data is usually mapped to an axis of space, either as a value axis or as a legend. This can be done with (modified) data aspects as well as with granularity labels.

Position in space Any content of the dataset can also be mapped to space in a line plot or in a scatter plot. This can be done separately for several elements, for example lines in a line plot. As already mentioned in Chapter 3, a position can be given absolute, or inside a range. The ranges itself are also a visual variable data can be mapped to. Good examples that show this are the recursive pattern arrangement [Keim et al., 1995] and the Cycle Plots [Cleveland, 1994], both explained in detail in Chapter 9.

Hue, chroma, lightness The three visual variables connected to color (or others if a different color model is used) can all be visual variable data is mapped to. Additional, it is possible to set fixed values for each.

For all characteristics that are visual variables, the value has to include whether it stems from visual mapping of data of from visual mapping of the calendar. When it is from visual mapping of data, the value has to determine wheter it is a native data aspect or a granularity label.

One of the most important possibilities in changing the visual mapping interactively is the modelling of semantic zooming using the structure of time.

8.3.8 View Stage

At View Stage, operations can combine or modify the partial visualizations from visual mapping transformations:

Combination Means that the result from several Visual Mapping Transformations is shown in the same space.

Many visulization perform user interactions like geometric zooming and panning or scrolling at view stage. We do not exclude these operations, but we do not consider them in our taxonomy because they are not related to the structure of time. Our operations like semantic zooming or paging are performed at analytical abstraction stage and visual mapping transformation respectively.

—The essence of knowledge is, having it, to apply it; not having it, to confess your ignorance.

Confucius

9 Visualization Methods that Make Use of the Structure of Time

This chapter contains a listing of several important visualizations that make use of the structure of time. They are classified according to the taxonomy from Chapter 8. Of course, the granularities used for building a visualization are usually interchangeable—even interactively. However, state-of-the-art visualizations are usually published with a fixed set of granularities. We present them with their original granularities and classify them accordingly. For some visualization, more than one way of applying our framework and taxonomy is possible. In those cases, we chose the one we consider most straightforward and easy to understand. Furthermore, we present visualizations that show the relationships between the visualizations we present.

9.1 Classification of Visualizations

9.1.1 Repeated Time Scale Graphs

Repeated Time Scale Graphs [Harris, 1999] are a quite basic variant of integrating the structure of time in visualization. In a line plot, a finer granule is used for the horizontal axis and several independend lines are drawn for each granule of a coarser granularity. Variants of this idea can be found as part of several more complex visualizations. An example using months and years is shown in Figure 9.1.

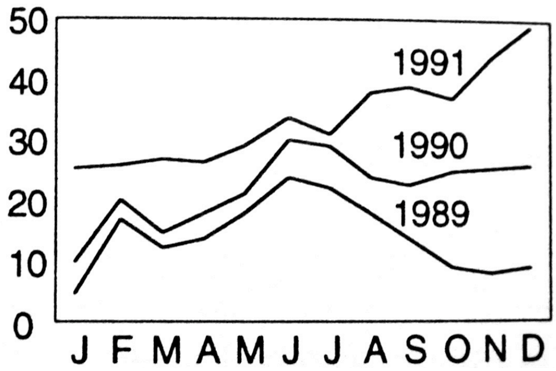


Figure 9.1: An Example Repeated Time Scale Graph from [Harris, 1999].

Table 9.1 shows the characteristics of the Repeated Time Scale Graph in Figure 9.1 according to our taxonomy from Chapter 8.

9.1.2 Seasonal Graphs

In order to see the cycle of months more easily but still show the trend over years, Seaonal Graphs [Harris, 1999] can be used. A line plot over average months is used to show the cycle. For each month, the trend over years is given as a small bar plot. The example shown in Figure 9.2 also depicts another aspect of using the structure of time in InfoVis: The beginning and end granule of the month granularity have been defined in a way that the two granules with the lowest aggregated data value are at the left and right end of the visualization.

Table 9.2 shows the characteristics of the Repeated Time Scale Graph in Figure 9.2 according to our taxonomy from Chapter 8.

9.1.3 Cycle Plots

Cycle Plots as introduced by Cleveland [1993] (see Figure 9.3) also use two different granularities. In his example of a CO_2 dataset, Cleveland uses months as chronon and years as additional granularity. Two combinations of operations are used to draw the two kinds of line traces:

1. The visual variables in this visualization can be enumerated as base position range on horizontal axis, detailed position inside range on horizontal axis, and position on vertical axis. For each datapoint, the month of year is mapped on the base position range on the horizontal axis and the year on the the detailed position inside this range. The data value is mapped on the position on the vertical

Table 9.1: The Repeated Time Scale Graph in Figure 9.1 Described according to the Taxonomy from Chapter 8

Tasks: Detect cycles; detect trends; compare granules; read values

Stage/Transformation	Characteristic	Value
Data Stage	Original Data Types	Discrete Data
	Primitives	Instant-based
Data Transformation	Granularities	Month, year
	Granularity Contexts	Month of year, year total
	Granularity Labels	First letter of English month names, discrete numbers for year
Analytical Abstraction Stage	Rasterization	Aggregate mean for each month/year combination
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	Abstract value data
Visual Mapping Transformation	Data value distribution: value axis, vertical	Data aspect: native
	Data value distribution: value axis, horizontal	Calendar: month of year
	Vertical position	Data aspect: rasterized to month/year combination
	Horizontal position	Data: month of year
	Line label vertical position	Data aspect: native
	Line label text	Data: year total
View Stage	–	–

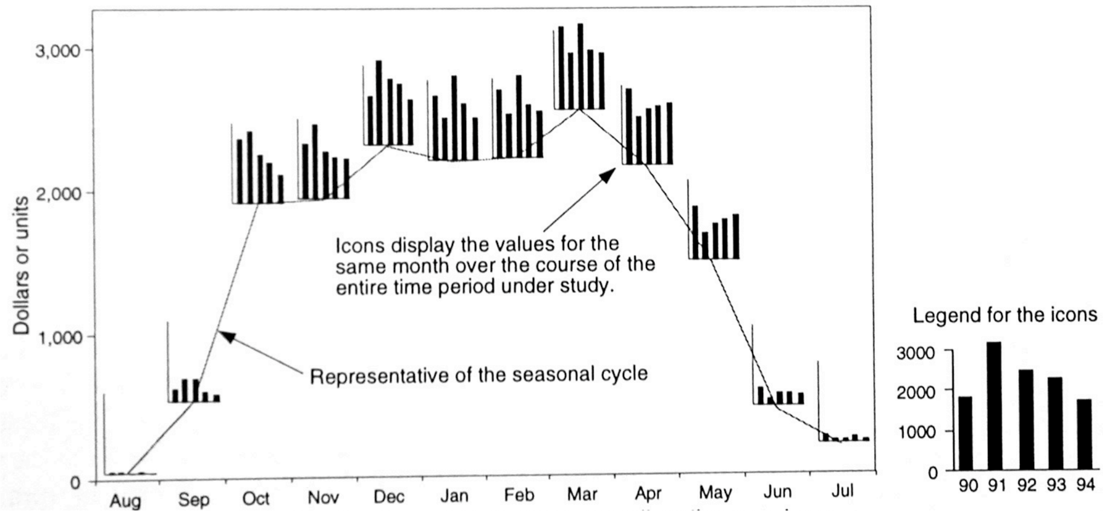


Figure 9.2: An Example Seasonal Graph from [Harris, 1999]. The line plot shows the cycle over average months. The small bar plots show the trend over years for each month separately.

axis. The resulting positions are connected by lineplots if the datapoints are part of the same month of year. This method of using the labels of a finer granularity for visual mapping on a coarser visual variable is similar to visualizations based on pivot tables.

2. The mean value over all years is calculated separately for each month of year. For each of those mean values, a horizontal line is drawn over the whole range that is used by the according month in combination 1. The position on the vertical axis is determined by the mean value.

In addition to these combinations, visual mapping of the calendar is used to draw labels on the horizontal axis for each month of year. The vertical axis is a value axis generated by visual mapping of data as described in subsection 3.1.

Table 9.3 shows the characteristics of the cycle plot in Figure 9.3 according to our taxonomy from Chapter 8.

9.1.4 Event Bands

Beard et al. [2008] present a visualization called Event Bands that are very similar to the timelines also used by Gantt Charts. Beard et al. also apply the concept of Small Multiples [Tufte, 1983]. Several timelines are combined and grouped according to nominal

Table 9.2: The Seasonal Graph in Figure 9.2 Described according to the Taxonomy from Chapter 8. We only present the description of the larger main visualization. The legend to the right consists of an example visualization, but that is done rather straightforward and later combined with the main visualization at View Stage.

Tasks: Detect cycles; detect trends; compare granules; read values

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data
	Primitives	Instant-based
Data Transformation	Granularities	Month, year
	Granularity Contexts	Month of year (August to July), year total
	Granularity Labels	First three letters of English month names, discrete numbers for year
Analytical Abstraction Stage	Rasterization	Aggregate mean for each month/year combination
	Aggregation	Aggregate mean for each month of year
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	2 times abstract value data
Visual Mapping Transformation	Data value distribution: value axis, vertical	Data aspect: native
	Data value distribution: value axis, horizontal	Calendar: month of year
	Vertical position (line plot)	Data aspect: aggregated to month of year
	Horizontal position (line plot)	Data: month of year
	Vertical base position (bar plot)	Data aspect: aggregated to month of year
	Horizontal base position (bar plot)	Data: month of year
	Vertical bar length (bar plot)	Data aspect: rasterized to month/year combination
	Horizontal detail position (bar plot)	Data: year total
View Stage	Combination	Line plot part, bar plot part

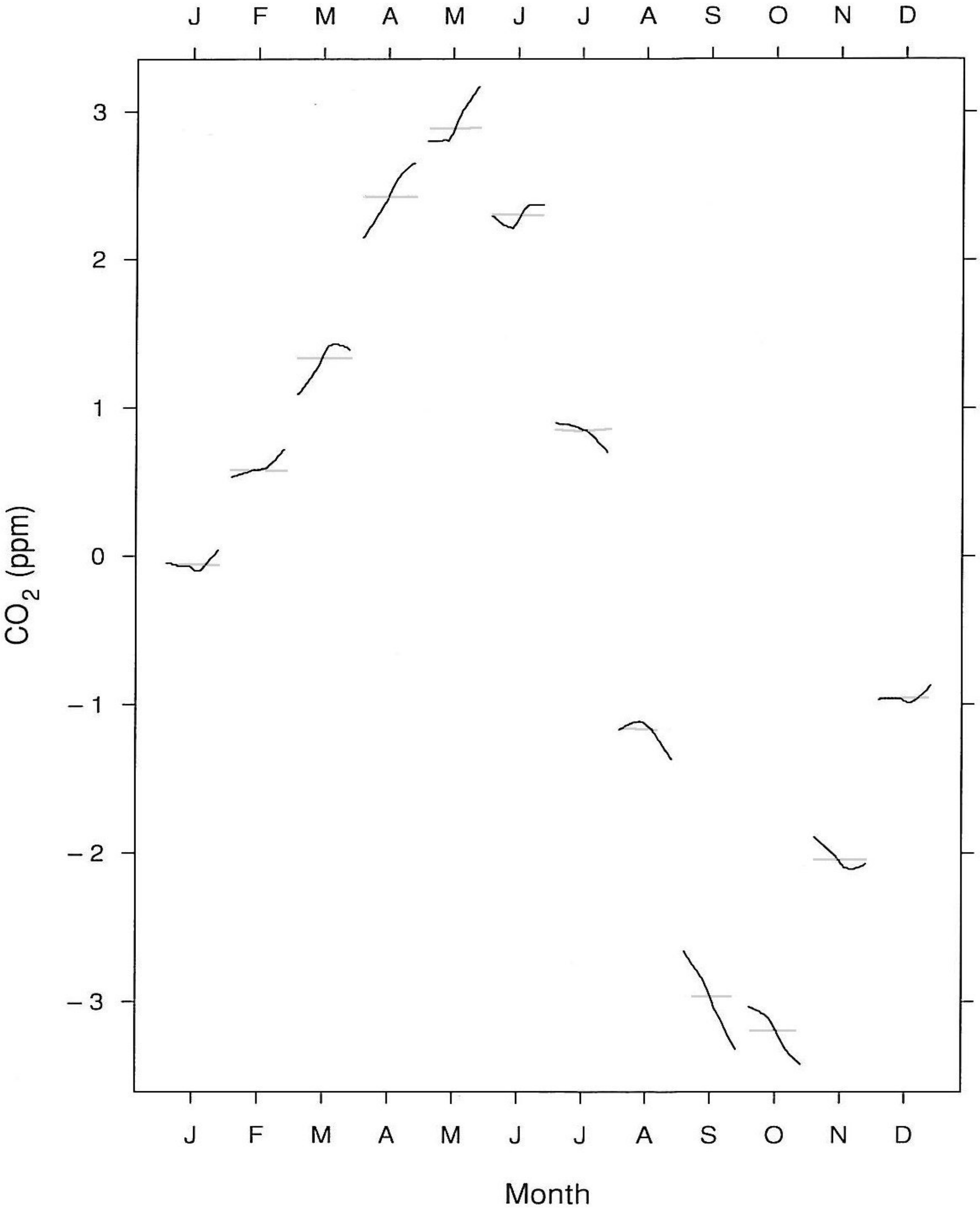


Figure 9.3: An Example Cycle Plot from [Cleveland, 1993]. A CO_2 dataset that is several years long is plotted separately for each month.

Table 9.3: The Cycle Plot in Figure 9.3 Described according to the Taxonomy from Chapter 8

Tasks: Interpret cycles; detect trends; compare granules; read values

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data (CO_2 Value)
	Primitives	Instant-based
Data Transformation	Granularities	Month, year
	Granularity Contexts	Month of year, year total
	Granularity Labels	First letter of English month names, unknown ordinal label for year
Analytical Abstraction Stage	Rasterization	Aggregate mean for each month total
	Aggregation	Aggregate mean for each month of year
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	2 times integrated abstract value data
Visual Mapping Transformation	Data value distribution: value axis, vertical	Data aspect: native
	Data value distribution: value axis, horizontal	Calendar: month of year
	Lightness (line 1)	0
	Absolute vertical position (line 1)	Data aspect: rasterized to month total
	Base horizontal position (line 1)	Data, granularity label: month of year
	Relative horizontal position (line 1)	Data, granularity label: year total
	Lightness (line 2)	0.5
	Absolute vertical position (line 2)	Data aspect: aggregated to month of year
	Absolute horizontal position (line 2)	Data, granularity label: month of year
View Stage	Combination	Value axes horizontal and vertical, line 1 and 2

or ordinal data attributes—including granularities. Figure 9.4 shows an example from their paper based on data from an ocean observation system.

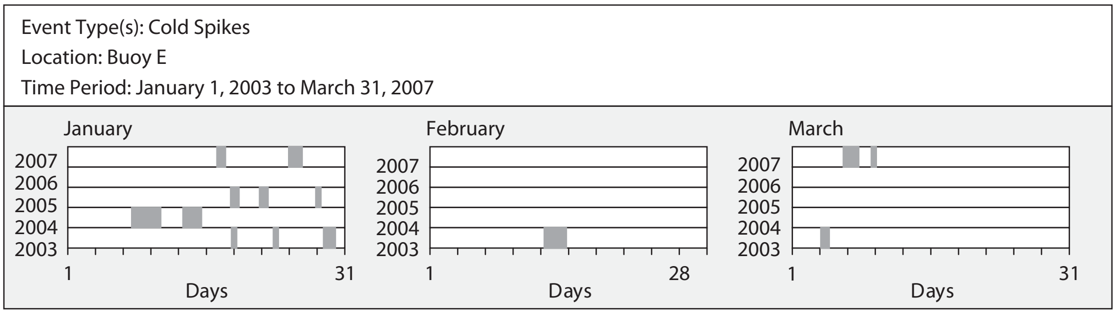


Figure 9.4: Event Bands by Beard et al. [2008]. Timelines are grouped according to several granularities.

Table 9.4 shows the characteristics of the cycle plot in Figure 9.4 according to our taxonomy from Chapter 8.

9.1.5 Recursive Pattern Arrangement

Pixel-based visualizations, as described by Keim et al. [1995], use position in two dimensions and color to encode data. For time-oriented data, the position within a two-dimensional grid can be determined by one time granularity for the position along the horizontal axis (e.g., day of week) and the value of another time granularity for the position on the vertical axis (e.g., week of month). In such visualizations, it is necessary to find a way to encode the value of single data points, as both axes are used for time granularities. One way is using different colors for different values. The result is sometimes referred to as "map", but we consider this term inaccurate as the position in space is used for abstract data.

Keim et al. [1995] enhance pixel-based visualizations by the inclusion of more data aspects to determine the position of pixels. They call this technique recursive pattern arrangement. Keim et al. propose several different kinds of recursive pattern arrangement (see Figure 9.5). The main advantages recursive pattern arrangement provides are a meaningful arrangement of pixels otherwise unsorted, and a lower distance between pixels that are related to data elements that are close in the dataset.

As an example for recursive pattern arrangement, we present Figure 9.6. This visualizations shows not one, but four recursive pattern arrangements, as the upmost level (separated by bars) distinguishes between several stocks and therefore is of nominal nature. There are two levels of recursion: Blocks of size three per three (presumably

Table 9.4: The Event Bands in Figure 9.4 Described according to the Taxonomy from Chapter 8

Tasks: Detect cycles; detect patterns; detect irregularities; detect regular structures more complex than cycles

Stage/Transformation	Characteristic	Value
Data Stage	Original Data Types	Discrete Data
	Primitives	Interval-based
Data Transformation	Granularities	Day, month, year
	Granularity Contexts	Day of month, month of year, year total
	Granularity Labels	Discrete numbers for day and year, English names for months
Analytical Abstraction Stage	–	–
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	Abstract value data
Visual Mapping Transformation	Base horizontal position	Data, granularity label: Month of year
	Vertical position	Data, granularity label: Year
	Relative horizontal position, start	Data (start), granularity label: Day of month
	Relative horizontal position, end	Data (end), granularity label: Day of month
	Data value distribution: value axis, horizontal outer	Calendar: month of year
	Data value distribution: value axis, vertical	Calendar: year
	Data value distribution: value axis, horizontal inner	Calendar: day of month
View Stage	–	–

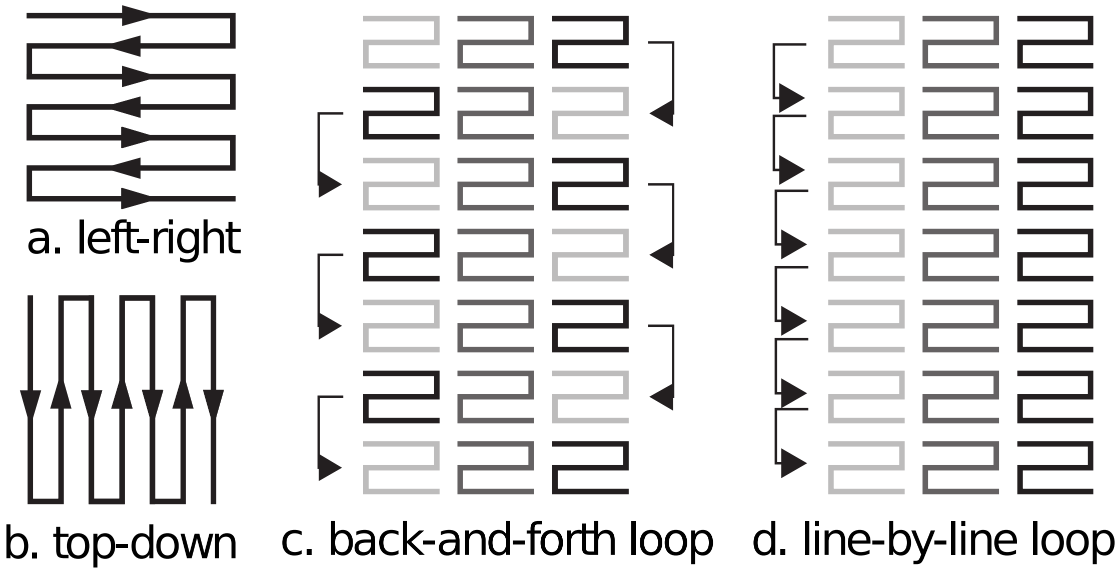


Figure 9.5: Different Variants of Recursive Pattern Arrangement [Keim et al., 1995]. The arrangement variants are a method to improve pixel-based visualizations.

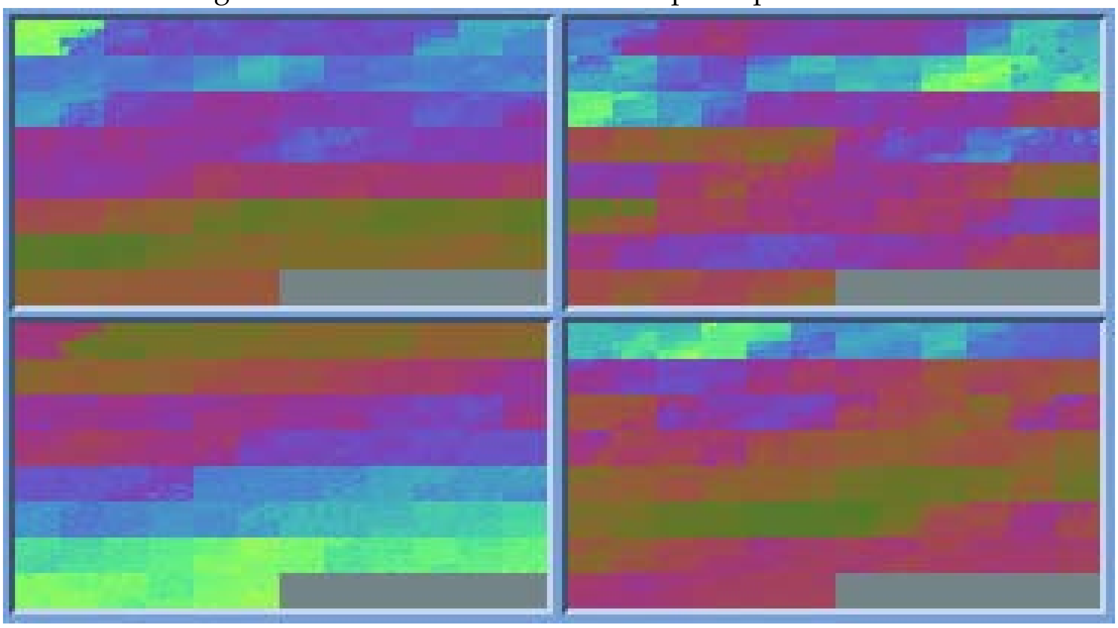


Figure 9.6: An Example for a Pixel-based Visualization Using Recursive Pattern Arrangement from [Keim et al., 1995]. For four different stocks, data is arranged according to year, month of year, business-week of month, and day of business-week.

Table 9.5: The Pixel-based Visualization Using Recursive Pattern Arrangement in Figure 9.6 Described according to the Taxonomy from Chapter 8

Tasks: Detect cycles; detect patterns; detect irregularities; detect regular structures more complex than cycles

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data
	Primitives	Instant-based
Data Transformation	Granularities	Hour, 3-hour-interval, business-day, business-week, month, year
	Granularity Contexts	Hour of 3-hour-interval, 3-hour-interval of business-day, business-day of business-week, business-week of month, month of year, year total
	Granularity Labels	Discrete numbers (mentioned in paper) for all granularities
Analytical Abstraction Stage	Rasterization	Aggregate mean for each hour/3-hour-interval/business-day/business-week/month/year combination
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	Abstract value data
Visual Mapping Transformation	Base vertical position	Data, granularity label: year total
	Base horizontal position	Data, granularity label: month of year
	Relative level 1 vertical position	Data, granularity label: business-week of month
	Relative level 1 horizontal position	Data, granularity label: business-day of business-week
	Relative level 2 vertical position	Data, granularity label: 3-hour-interval of business-day
	Relative level 2 horizontal position	Data, granularity label: hour of 3-hour-interval
	Hue	Data aspect, aggregated to hour/3-hour-interval/business-day/business-week/month/year combination
View Stage	–	–

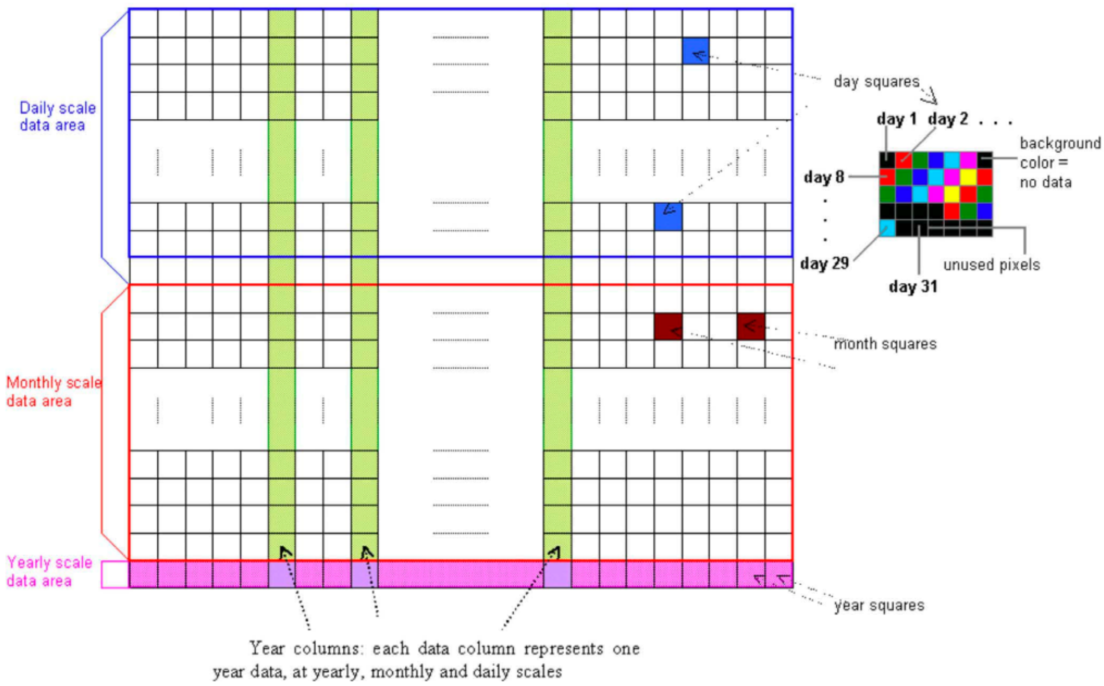


Figure 9.7: The Structure of Multi-scale Visualizations according to Shimabukuro et al. [2004]. Three complementing views show data based on a daily, a monthly and a yearly scale.

nine hours) represent one day. The five days of the business week and four weeks per month (adapted by an algorithm that prevents gaps and is not described in detail by Keim et al. [1995]) form the next recursion level. The out recursion level consists of twelve month per year and eight years. The characteristics of this visualization according to our taxonomy from Chapter 8 are shown in Table 9.5.

Beside the obvious novelties introduced by Keim et al. [1995], the paper is the first we know of that uses a neutral background color to fill gaps (the areas at the end of the datasets). In our pixel-based visualizations, we also apply that method (see Chapter 12), but if possible, we prefer the use of granularities that do not even produce gaps (see Chapter 14 for one example).

9.1.6 Multi-scale Visualizations

Multi-scale visualizations have been introduced by Shimabukuro et al. [2004] based on the recursive pattern arrangement by Keim et al. [1995] (see Subsection 9.1.5). Four

Table 9.6: The Multi-scale Visualization in Figure 9.7 Described according to the Taxonomy from Chapter 8

Tasks: Detect cycles; detect patterns; detect irregularities; detect regular structures more complex than cycles

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data
	Primitives	Instant-based
Data Transformation	Granularities	Day, week, month, year
	Granularity Contexts	Day of week, week of month, month of year, year total
	Granularity Labels	Unknown ordinal labels for all granularities
Analytical Abstraction Stage	Rasterization	Aggregate mean for each day/week/month/year combination
	Aggregation	Aggregate mean for each month/year combination
	Aggregation	Aggregate mean for each year total
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	3 times parallel abstract value data
Visual Mapping Transformation	Base vertical position (area 1)	Data, granularity label: month of year
	Base horizontal position (area 1)	Data, granularity label: year total
	Relative vertical position (area 1)	Data, granularity label: week of month
	Relative horizontal position (area 1)	Data, granularity label: day of week
	Hue (area 1)	Data aspect, rasterized to day/week/month/year combination
	Absolute vertical position (area 2)	Data, granularity label: month of year
	Absolute horizontal position (area 2)	Data, granularity label: year total
	Hue (area 2)	Data aspect, aggregated to month/year combination
	Absolute horizontal position (area 3)	Data, granularity label: year total
	Hue (area 3)	Data aspect, aggregated to year total
View Stage	Combination	3 areas

different granularities are used. Three pixel-based visualizations are build in parallel, then combined to the total visualization. The arrangement is shown in Figure 9.7.

1. The detail view is built by aggregating to days of month. The resulting data values are mapped on the pixel colors. The data is arranged in blocks, with each block representing one month. Inside the blocks, the week of month is mapped on the vertical position. The day of week is mapped on the horizontal position. This arrangement is similar to the one of a typical calendar sheet. The blocks are positioned over the whole view themselves by granularity labels. The year is mapped on the horizontal position. The month of year is mapped on the vertical position. This mapping results in recursive pattern arrangement.

2. The monthly average view is generated by aggregating to months of year. Each month is drawn as one uniform block, with the average data value mapped to the color again. The arrangement of the blocks is performed by mapping month of year and year, exactly like in the detail view.

3. The yearly average view is generated by aggregating to years. That third view is smaller than the others. It contains only one line with blocks that have the same size as in the monthly average view. Each year is drawn as one uniform block, with the average data value mapped to the color again. The arrangement of the blocks is performed my mapping the year to the horizontal position.

Table 9.6 shows the characteristics of the Multi-scale Visualization in Figure 9.7 according to our taxonomy from Chapter 8. We model the multiple scales by defining several separate visualizations, named areas in Table 9.6, that are later combined into one visualization.

9.1.7 Spiral Layouts

As already mentioned, the pixel-based visualization shown in the last subsections share the problem oft having the data broken into lines. One alternative has already been mentioned—using a back-and-forth-arrangement— but that arrangement is rather unintuitive for users.

Those problems can be solved by using a spiral layout. A spiral layout has an inherent reference to cyclic granularities. Each granule of a coarser granularity is mapped to one revolution, while each granule of a finer granularity is mapped to a small angle piece, the ideal case would be one pixel. Figure 9.8 shows an example called Spiral Graph by Weber et al. [2001a].

Originally, spiral layouts have been designed using the size of objects drawn into the spiral instead of colors. The oldest example from literature is from Gabaglio [1888]

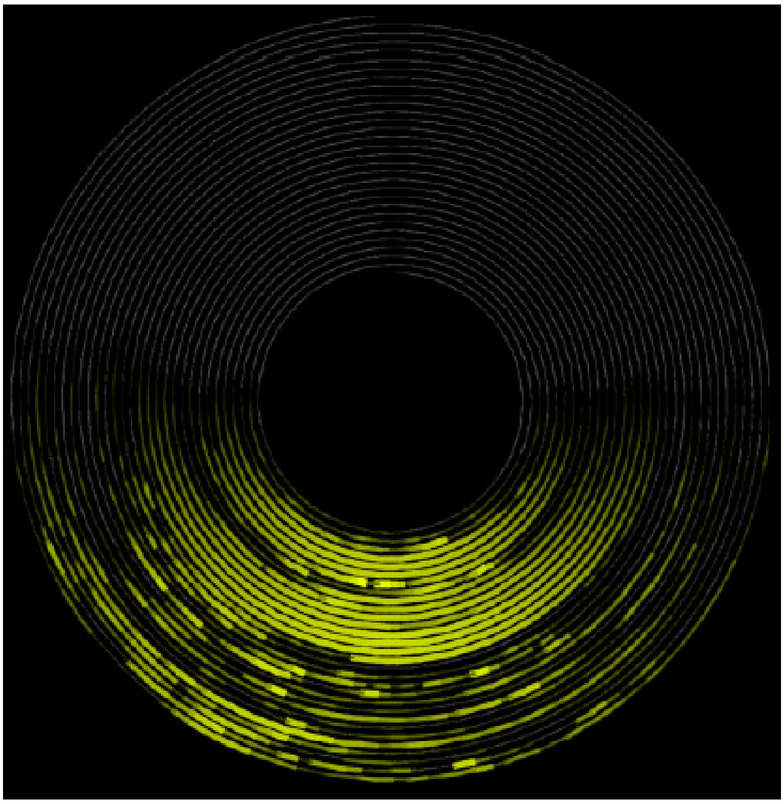


Figure 9.8: Spiral Graph [Weber et al., 2001a]. Data values are mapped to color, making it a pixel-based visualization. The main difference to other pixel-based visualization is the use of polar coordinates.

and depicts savings from the Italian post office. This historic visualization is shown in Figure 9.9.

An updated version of the visualization by Gabaglio is presented by Carlis and Konstan [1998]. The visualization, shown in Figure 9.10, uses a polar coordinate grid to better locate the data points and circles (called Blots) instead of boxes (which, of course, reduces the number of data values shown to one). Carlis and Konstan [1998] also present other variants for spiral layouts. Figure 9.11 shows a variant that contains intervals.

Table 9.7 shows the characteristics of the Blot Visualization Using Spiral Layout in Figure 9.10 according to our taxonomy from Chapter 8.

Table 9.8 shows the characteristics of the Interval Visualization Using Spiral Layout in Figure 9.11 according to our taxonomy from Chapter 8.

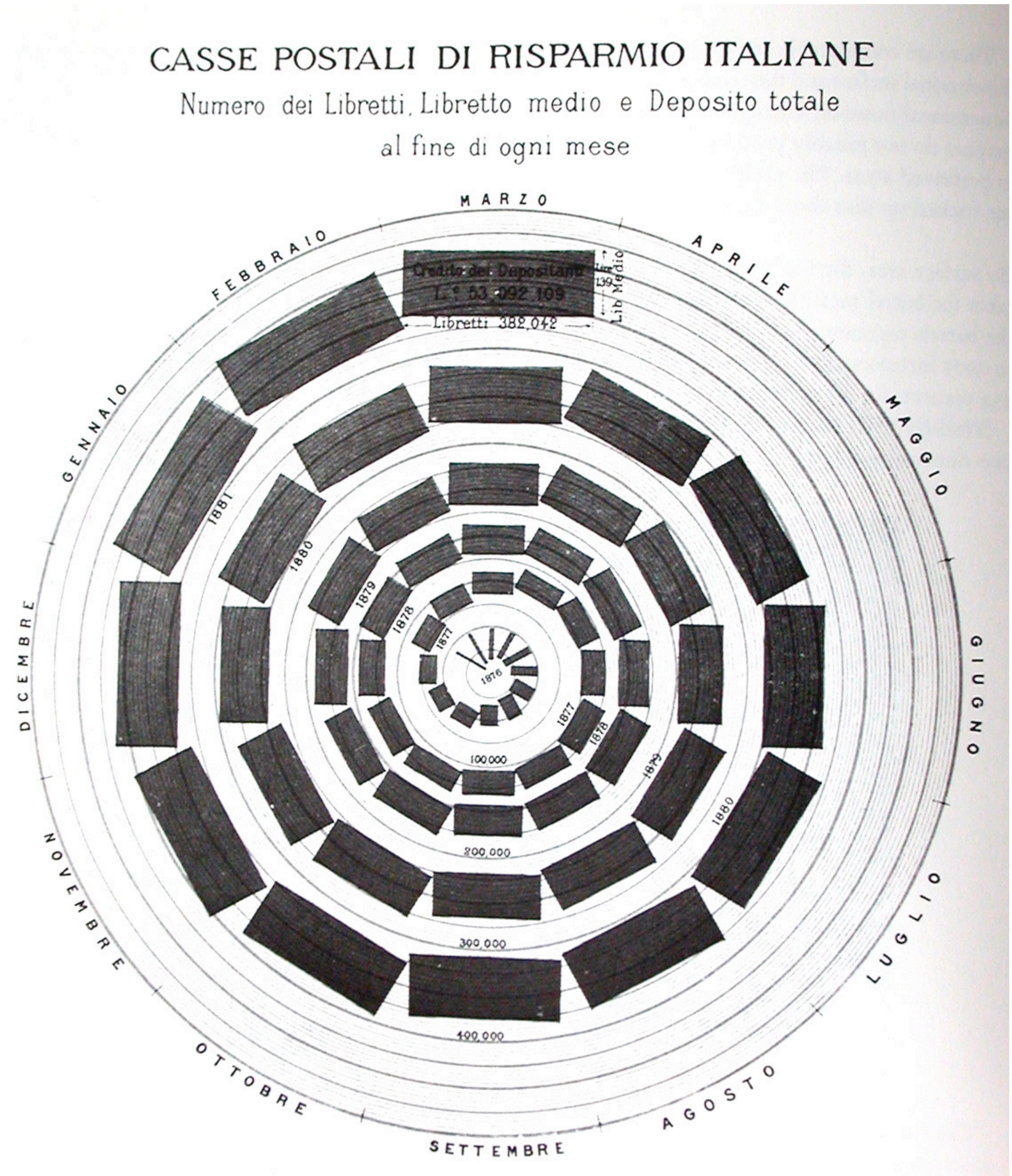


Figure 9.9: Historic Spiral Layout by Gabaglio [1888]. Post office savings data from several years is shown using granularities of month and year.

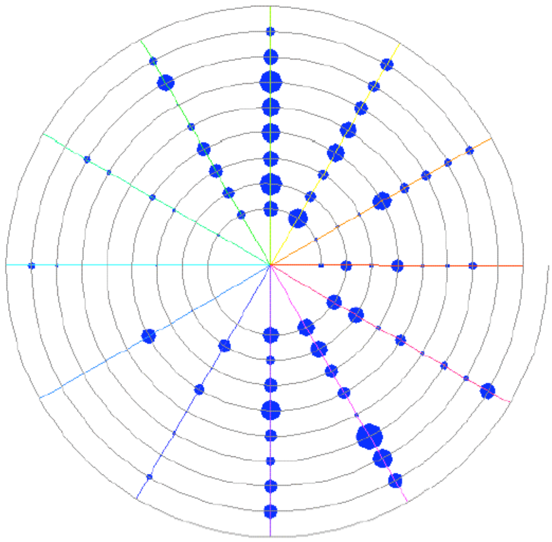


Figure 9.10: A Spiral Layout by Carlis and Konstan [1998] Using Blots to Show Data Values. The data is about food consumption of chimpanzees.

Table 9.7: The Blot Visualization Using Spiral Layout in Figure 9.10 Described according to the Taxonomy from Chapter 8

Tasks: Detect cycles; detect patterns; detect irregularities

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data
	Primitives	Instant-based
Data Transformation	Granularities	Month, year
	Granularity Contexts	Month of year, year total
	Granularity Labels	Unknown ordinal labels for all granularities
Analytical Abstraction Stage	Rasterization	Aggregate mean for each month/year combination
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	Abstract value data
Visual Mapping Transformation	Distance from center	Data, granularity label: month/year combination
	Angle	Data, granularity label: month of year
	Blot size	Data aspect, aggregated to month/year combination
View Stage	–	–

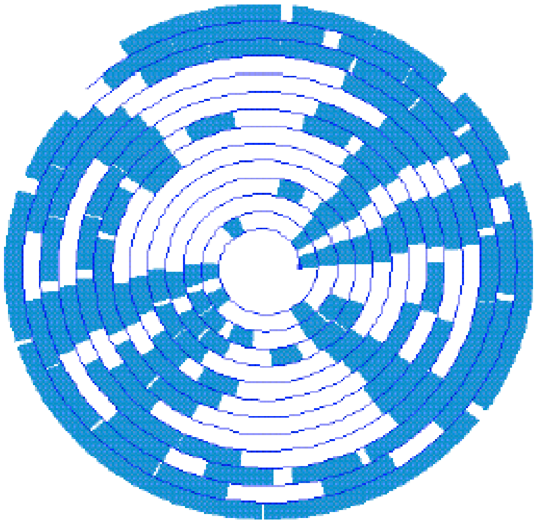


Figure 9.11: A Spiral Layout by Carlis and Konstan [1998] Showing Intervals. The data depicts times when instruments can be heard in a music recording.

9.1.8 Cyclic Layouts

Similar to spiral layouts, it is also possible to use cyclic layouts. They allow for a better allocation of the granules of the coarser granularity. On the downside, they have a cleft between the end of one cycle and the beginning of the next, similar to pixel-based visualizations in Cartesian coordinate systems. However, in cyclic layouts, this cleft is smaller.

One visualization method with a cyclic layout is the Concentric Circles Technique by Daassi et al. [2000] (see Figure 9.12). Besides obviously using the structure of time for choosing circle and angle, there is also the possibility to reconfigure the granules that are shown for the year granularity.

Table 9.9 shows the characteristics of the Concentric Circle Technique in Figure 9.12 according to our taxonomy from Chapter 8. Only one of the two spaces is described and the controls are omitted.

The examples show that while spiral layouts and circular layouts indeed have advantages, they also have a very big drawback. Granules with fixed and identical lengths are mapped to distances that are visually increasing by 2π. That fact impedes the comparison of values at different granules of a higher granularity level significantly. Furthermore, digital displays, usually having a Cartesian coordinate system, are sometimes only of limited use to draw data in polar coordinates. However, newer high-resolution

Table 9.8: The Interval Visualization Using Spiral Layout in Figure 9.11 Described according to the Taxonomy from Chapter 8

Tasks: Detect cycles; detect patterns; detect irregularities

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data
	Primitives	Interval-based
Data Transformation	Granularities	Millisecond, beat (of music)
	Granularity Contexts	Millisecond of beat, beat total
	Granularity Labels	Unknown ordinal labels for all granularities
Analytical Abstraction Stage	Rasterization	Generate data elements for all millisecond/beat combinations; set data to 1 after a note started and to 0 after it has ended
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	Abstract value data
Visual Mapping Transformation	Distance from center	Data, granularity label: millisecond/beat combination
	Angle	Data, granularity label: millisecond of beat
	Color	Data aspect, generated for millisecond/beat combinations
View Stage	–	–

displays reduce this problem more and more.

9.1.9 Treemaps based on Granularities

Especially when there is not much variance in the data, pixel-based visualizations might be confused with treemaps [Johnson and Shneiderman, 1991]. This is not true—we will elaborate the differences in this subsection. However, treemaps are also well-suited as a basic visualization that can enhanced by applying the structure of time. The concept has been proved by Wood et al. [2008] using squarified treemaps [Bruls et al., 2000] to show spatio-temporal data.

An example from the paper by Wood et al. is shown in Figure 9.13. The traffic volume is mapped to area, the speed is mapped to color. The distribution of areas is based on a tree structure (vehicle type $\hookrightarrow$ day of week $\hookrightarrow$ hour of day). Therefore, no direct mapping from granularity data to location in space is given. The granularity data determines the tree structure, the spatial layout is provided by the squarified treemap

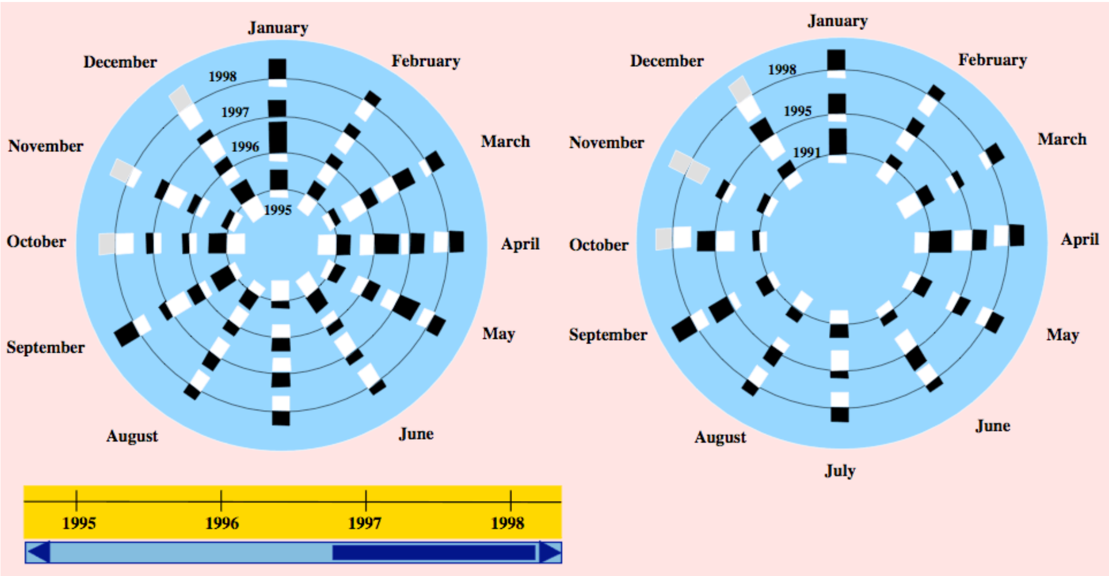


Figure 9.12: Concentric Circles Technique [Daassi et al., 2000]. Each circle represents a year, two independend data attributes can be plotted for each month as white and black bar. The left side is a "working area", where the years to be shown can be configured. The right side is a "storage space".

algorithm of Bruls et al. This fact is the main difference between a pixel-based visualization and a treemap.

To show the characteristics of the treemap in Figure 9.13 according to our taxonomy from Chapter 8, we have to resort to operators defined by Chi and Riedl [1998], Chi [2000]. Table 9.10 presents these characteristics.

9.1.10 Visualizations already Used by Analysts

These visualization do not stem from InfoVis literature, but they are frequently used in real-world application by analysts of time-oriented data we collaborated with over the course of the project **DisCō**. They are generated using the Visual Analytics application [TIS]—Time Intelligence Solutions by XIMES GmbH[1], Vienna.

[1]For more information, see http://www.ximes.com (accessed on April 9th, 2010).

Table 9.9: The Visualization Using Concentric Circle Technique in Figure 9.12 Described according to the Taxonomy from Chapter 8. Only one of the two spaces is described and the controls are omitted.

Tasks: Detect cycles; detect patterns; detect irregularities; read values

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data
	Primitives	Instant-based
Data Transformation	Granularities	Month, year
	Granularity Contexts	Month of year, year total
	Granularity Labels	Unknown ordinal labels for all granularities
Analytical Abstraction Stage	Rasterization	Aggregate mean for each month/year combination
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	Abstract value data
Visual Mapping Transformation	Distance from center	Data, granularity label: year total
	Angle	Data, granularity label: month of year
	Bar 1 length	Data aspect, aggregated to month/year combination
	Bar 2 length	Data aspect, aggregated to month/year combination
	Data value distribution: value axis, circles	Calendar: year total
	Data value distribution: value axis, rays	Calendar: month of year
View Stage	–	–

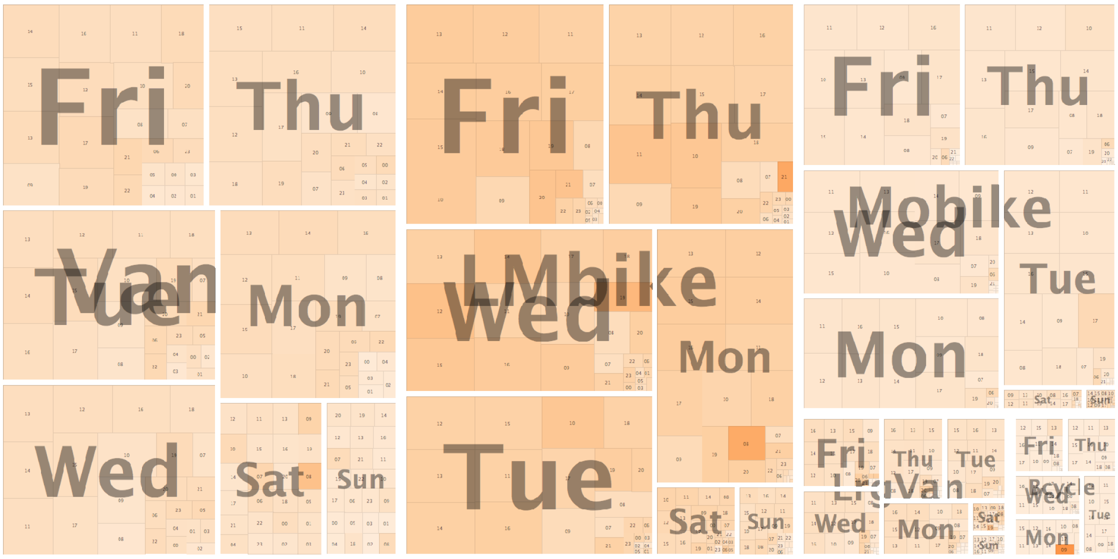


Figure 9.13: Spatio Temporal Treemap [Wood et al., 2008]. The traffic volume is mapped to area, the speed is mapped to color. Beside the nominal data attribute of vehicle type, the granularities day of week and hour of day are used.

Line Plots of Pivot Tables

A visualizations considered particularly useful by the analysts is based on a pivot table per day of week. Another important feature of this visualization is the fact that it is not based on points in time but on intervals that are touching each other. These intervals are plotted as horizontal lines that are connected by vertical lines where there is a boundary point of two intervals (see Figure 9.14). While the visualization was originally developed based on a pivot table and the formal definition was more difficult, it is very straightforward using our taxonomy.

The visualization can be adapted very easily using granularities. By replacing the granularity “day” with another one, for example “highschool holiday state”, new insights can be gained. In Germany, there are several blocks of highschool holidays distributed over the year: Christmas Holidays, Easter Holidays, Pentecost Holidays, and Summer Holidays. Together with “No Holidays”, these can be considered the granules of a granularity. This can be done without a problem, even though the order of granularities changes. A visualization based on this granularity is shown in Figure 9.15.

Table 9.11 shows the characteristics of the visualization in Figure 9.14 according to our taxonomy from Chapter 8. Table 9.12 shows the characteristics of the visualization in Figure 9.15 according to our taxonomy from Chapter 8.

Table 9.10: The Spatio-temporal Treemap in Figure 9.13 Described according to the Taxonomy from Chapter 8

Tasks: Detect patterns; compare granules; compare granule combinations

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data
	Primitives	Instant-based
Data Transformation	Granularities	Hour, day, week
	Granularity Contexts	Hour of day, day of week, week total
	Granularity Labels	Discrete number for hour of day, 3 letters of English name of day of week, unknown ordinal label for week total
Analytical Abstraction Stage	Rasterization	Aggregate mean for each hour/day/week combination
Analytical Abstraction Stage	Aggregation	Aggregate mean for each week total
Visualization Transformation	Hierarchy	Do breadth first traversal
Visualization Abstraction Stage	Type of visualization	Tree hierarchy: Nominal data attribute, granularity label: day of week, granularity label: hour of day
Visual Mapping Transformation	Position	According to squarified treemaps algorithm[Bruls et al., 2000]
	Area	Data attribute 1, aggregated to hour/day combination
	Color	Data attribute 2, aggregated to hour/day combination
View Stage	–	–

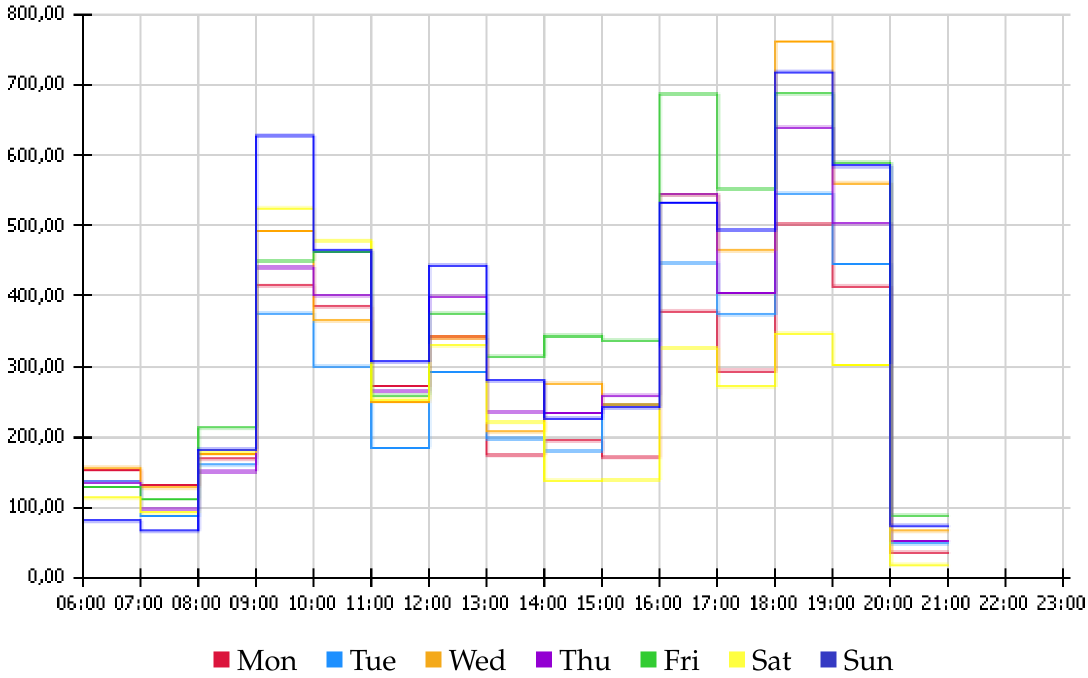


Figure 9.14: Line Plots based on Pivot Table over Days of Week. The visualization shows turnover data from a culinary Establishment.

Granularities in Stacked Bar Plots

Stacked bar plots are a reasonable choice to show aggregated data at more than one aggregation level. In the context of non-time-oriented data aspects the visualization method is well known even to lay people. However, it can also be used on data partitioned by two levels of granularities.

An example is shown in Figure 9.16. The data of half a year is aggregated to sums per week, but at the same time, the part belonging to holidays is colored differently from the part belonging to non-holidays (ignoring holidays caused by weekends, but including holidays that fall on weekends for this special case).

Figure 9.16 also is another nice example of the problem state-of-the-art visualizations have with irregularities, even when not using the problematic month granularity. Even though the dataset starts at January the 1st, 2006, there is a small bar labelled “2005, 52”. According to the norms DIN 1355 as well as ISO 8601, the date in question actually belongs to the last week of 2005, an irregularity caused by the fact that neither 365 nor 366 can be divided by 7 without remainder.

Table 9.11: The Visualization of Pivot Data over Days in Figure 9.14 Described according to the Taxonomy from Chapter 8

Tasks: Interpret cycles; detect irregularities; compare granules; read values

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data
	Primitives	Instant-based
Data Transformation	Granularities	Hour, Day, Week
	Granularity Contexts	Hour of day, day of week, week total
	Granularity Labels	Discrete numbers for hour of day, strings (ordinal or nominal possible) for day of week, unknown ordinal label for week total
Analytical Abstraction Stage	Rasterization	Aggregate mean for each hour/week combination (keeping day of week separate)
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	Abstract value data
Visual Mapping Transformation	Vertical position	Data, aggregated to hour/week combination
	Horizontal position	Data, granularity label: hour of day
	Hue	Data, granularity label: day of week
	Data value distribution: value axis, vertical	Data aspect: native
	Data value distribution: value axis, horizontal	Calendar: hour of day
	Data value distribution: legend, hue	Calendar: day of week
View Stage	–	–

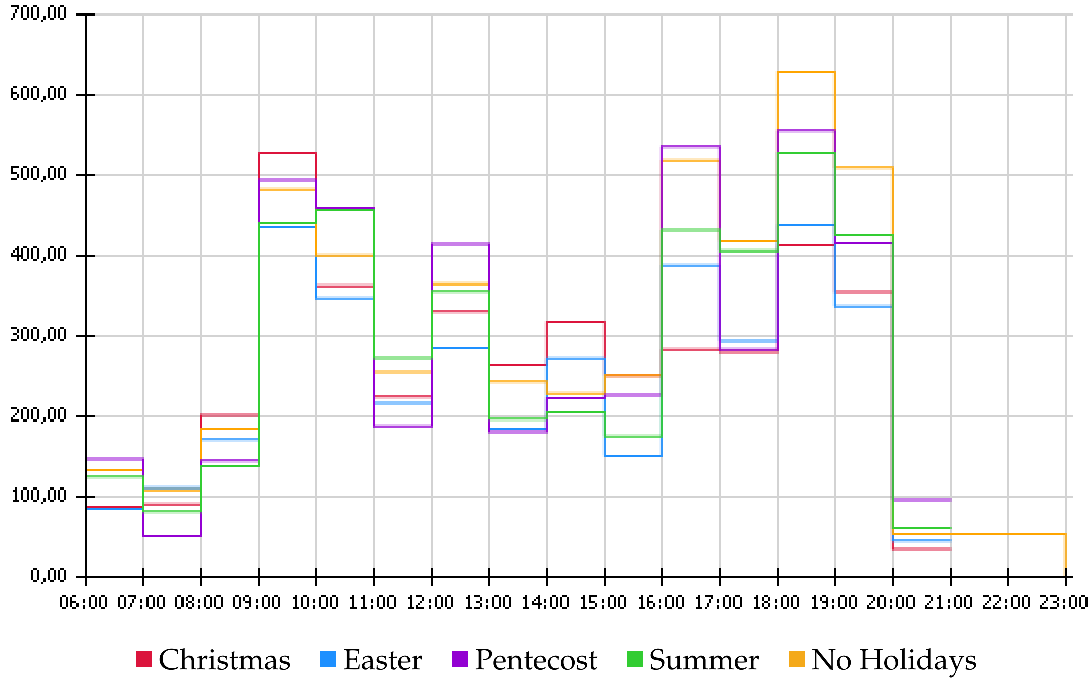


Figure 9.15: Line Plots based on Pivot Table over Highschool Holidays. The visualization shows turnover data from a culinary Establishment.

Table 9.13 shows the characteristics of the visualization in Figure 9.16 according to our taxonomy from Chapter 8.

Spread per Granule

Various variants of visualizations that show spread of data values are known in literature. In the context of time-oriented data, more information can be revealed by showing the spread of data values separately for different granules of a granularity.

An example is shown in Figure 9.17. The data over a longer period of time is shown in a way that reveals spreading. However, the data is divided in several bins per day of week. The idea has similarities to Cycle Plots (see Subsection 9.1.3), but the number of values per granule is significantly higher and the connection between points has been omitted.

Table 9.14 shows the characteristics of the visualization in Figure 9.17 according to our taxonomy from Chapter 8.

Table 9.12: The Visualization of Pivot Data over Holidays in Figure 9.15 Described according to the Taxonomy from Chapter 8

Tasks: Interpret cycles; detect irregularities; compare granules; read values

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data
	Primitives	Instant-based
Data Transformation	Granularities	Hour, week, holiday state
	Granularity Contexts	Hour of day, week of holiday state, holiday state total
	Granularity Labels	Discrete numbers for hour of day, unknown ordinal label for week of holiday state, strings (ordinal or nominal possible) for holiday state total
Analytical Abstraction Stage	Rasterization	Aggregate mean for each hour/week combination (keeping holiday state)
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	Abstract value data
Visual Mapping Transformation	Vertical position	Data, aggregated to hour/week combination
	Horizontal position	Data, granularity label: hour of day
	Hue	Data, granularity label: holiday state total
	Data value distribution: value axis, vertical	Data aspect: native
	Data value distribution: value axis, horizontal	Calendar: hour of day
	Data value distribution: legend, hue	Calendar: holiday state total
View Stage	–	–

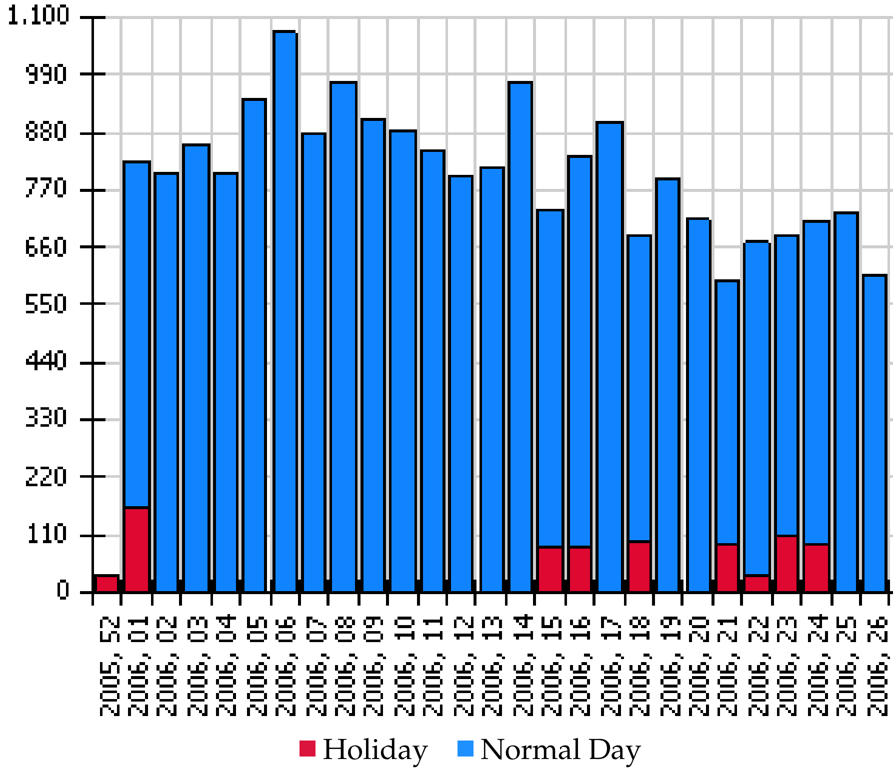


Figure 9.16: Stacked Bars Show Summed Data over Week of Year. The colors help to distinguish which amount resulted from a holiday, and which amount resulted from a normal day (weekends ignored). The visualization shows working hours of engine drivers.

Comparison with Past Values

Especially for visualizations that are continually renewed, it can be very helpful to provide the data in context of past values. This can also be modelled using granularities, but the size of granules has to be irregular. This is not unusual. For example, month of years are also irregular. However, in this example, three granules have a clear limited length, while the third one is disproportionately longer.

An example is shown in Figure 9.18. The last three days are shown including lines showing the development. Past values are added to show the distribution that had existed before.

Table 9.15 shows the characteristics of the visualization in Figure 9.18 according to our taxonomy from Chapter 8.

Table 9.13: The Visualization of Stacked Granularities in Figure 9.16 Described according to the Taxonomy from Chapter 8

Tasks: Detect irregularities; compare granules; read values

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data
	Primitives	Instant-based
Data Transformation	Granularities	Type, week
	Granularity Contexts	Type total, week of year, year total
	Granularity Labels	Nominal labels (holiday/normal) for type, discrete numbers for week of year, discrete numbers for year total
Analytical Abstraction Stage	Rasterization	Aggregate sum for each type/week combination
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	Abstract value data
Visual Mapping Transformation	Vertical position	Data, type (as 0 or 1) multiplied with data aggregated to type/week combination for holiday
	Vertical length	Data, aggregated to type/week combination
	Hue	Data, type
	Horizontal base position	Data, granularity label: year total
	Horizontal detail position	Data, granularity label: week of year
	Data value distribution: value axis, vertical	Data aspect: native
	Data value distribution: value axis, horizontal	Calendar: week/year combination
	Data value distribution: legend, hue	Calendar: type total
View Stage	–	–

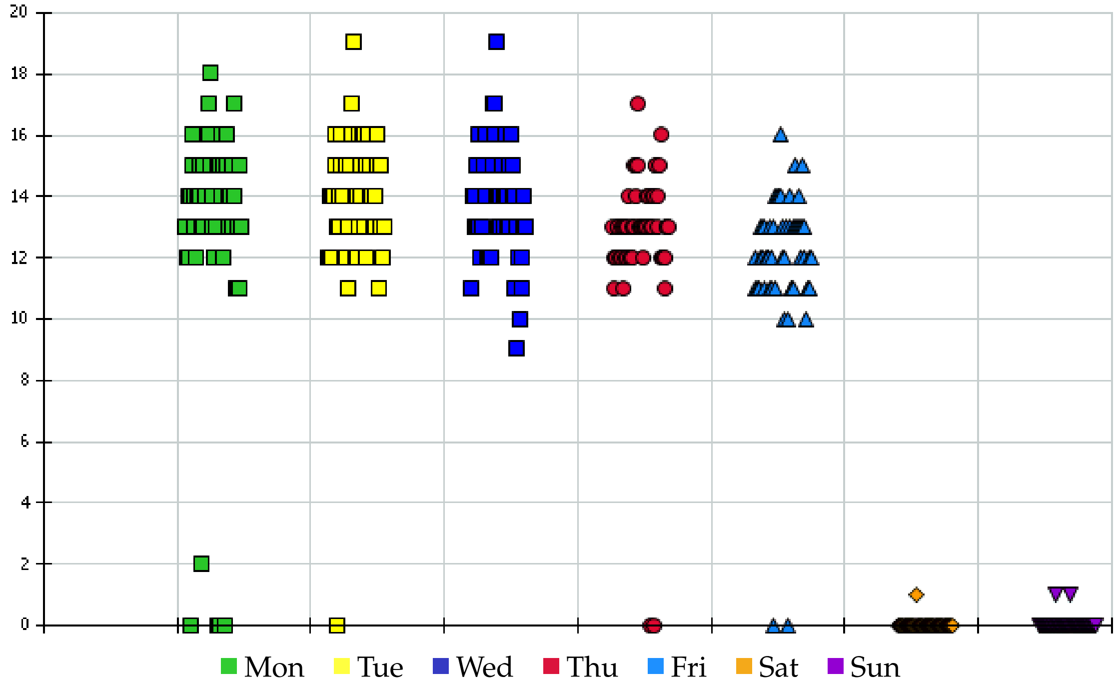


Figure 9.17: The Data Value Spreading is Shown for each Day of Week Separately. The visualization shows working hours of engine drivers.

Difference to the Week Before

Sometimes it is not necessary to show absolute values. In this example, development over time can be shown by only mapping the difference between two weeks to a visual variable.

Figure 9.19 contains an example. Three independent values are shown. For each day, the difference is shown, but not the difference to the day before, but to the same day of week in the week before. Therefore, unexpected trends become very prominent, but they can immediatly be broken down if they are associated with a certain day of week.

Table 9.16 shows the characteristics of the visualization in Figure 9.19 according to our taxonomy from Chapter 8.

9.2 Overview Visualizations

In order to get an overview of the visualization methods, we summarize their attributes according to the taxonomy. In the summary, we do only consider attributes suitable for

Table 9.14: The Visualization of Spread per Granule in Figure 9.17 Described according to the Taxonomy from Chapter 8

Tasks: Detect irregularities; compare granules; compare granules on statistical level; read values

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data
	Primitives	Instant-based
Data Transformation	Granularities	Type, week
	Granularity Contexts	Type total, week of year, year total
	Granularity Labels	Nominal labels (holiday/normal) for type, discrete numbers for week of year, discrete numbers for year total
Analytical Abstraction Stage	Rasterization	Aggregate sum for each type/week combination
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	Abstract value data
Visual Mapping Transformation	Vertical position	Data, type (as 0 or 1) multiplied with data aggregated to type/week combination for holiday
	Vertical length	Data, aggregated to type/week combination
	Hue	Data, type
	Horizontal base position	Data, granularity label: year total
	Horizontal detail position	Data, granularity label: week of year
	Data value distribution: value axis, vertical	Data aspect: native
	Data value distribution: value axis, horizontal	Calendar: week/year combination
	Data value distribution: legend, hue	Calendar: type total
View Stage	–	–

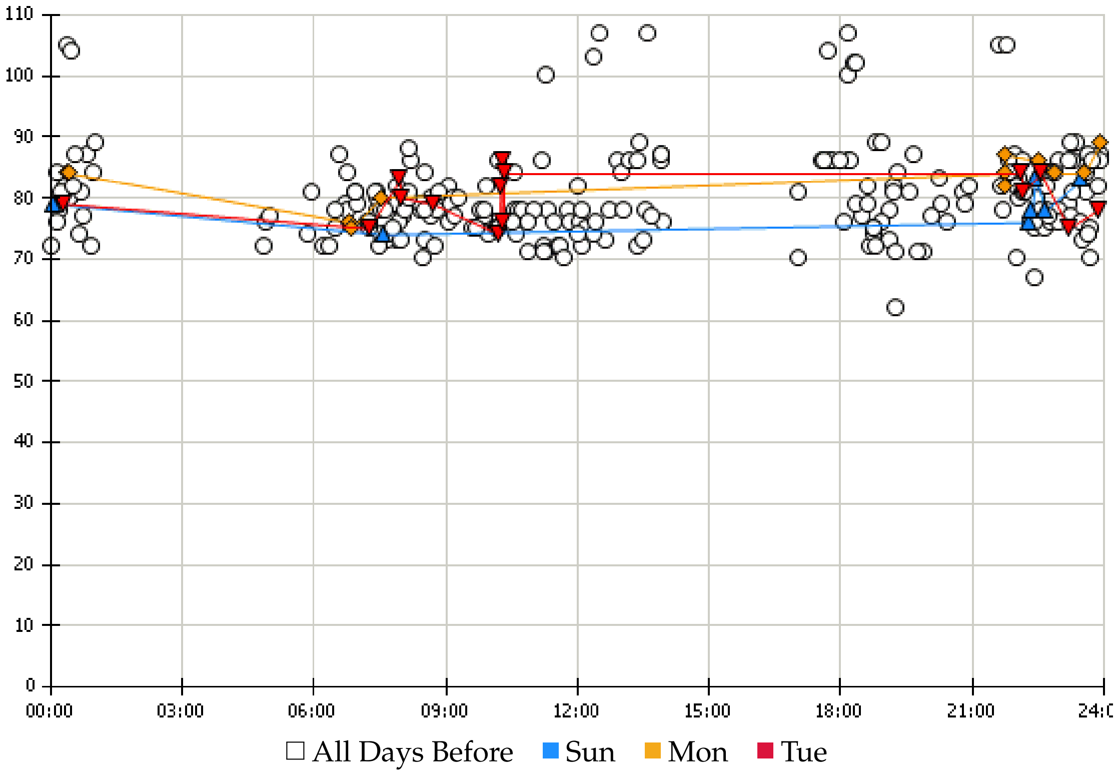


Figure 9.18: Comparison with Past Values. The development of values over the last three days, labelled as day of week, is compared to a number of values from days before those three days. The visualization shows working hours of engine drivers.

the comparison. For example, we do not consider which granularities are used, because all visualizations are also conceivable with different granularities.

In Table 9.17, we show an overview of the visualizations in the same order that was chosen manually in this chapter. The order represents a classification based on a human sense of similarities. The orange area shows which tasks we consider the visualizations suited for. A dark field represents that a visualization is suitable for a task. The purple to cyan areas represent summarized steps of the visualization process. A dark field means that an operation is performed in the creation of a visualization, or that a certain data type is needed to build it.

To give a better impression which visualizations are most similar according to the classification scheme we use in Table 9.17, the best order of visualization is that which has the least differences between two neighbored visualizations regarding dark and

Table 9.15: The Visualization of Comparison with Past Values in Figure 9.18 Described according to the Taxonomy from Chapter 8

Tasks: Detect irregularities; compare granules; compare granules on statistical level; read values

Stage/Trans-formation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data
	Primitives	Instant-based
Data Transformation	Granularities	Minute, hour, day, range
	Granularity Contexts	Minute of day, hour of day, day of range, range total
	Granularity Labels	Discrete numbers for minute of day, hour of day, unknown ordinal scale for day of range, ordinal labels for range
Analytical Abstraction Stage	Aggregation	Aggregate mean for each minute/range combination (excluding day)
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	Abstract value data
Visual Mapping Transformation	Vertical position	Data aggregated to minute/range combination
	Hue	Range
	Connectivity	Range
	Horizontal position	Data, granularity label: minute of day
	Data value distribution: value axis, vertical	Data aspect: native
	Data value distribution: value axis, horizontal	Calendar: hour of day
View Stage	–	–

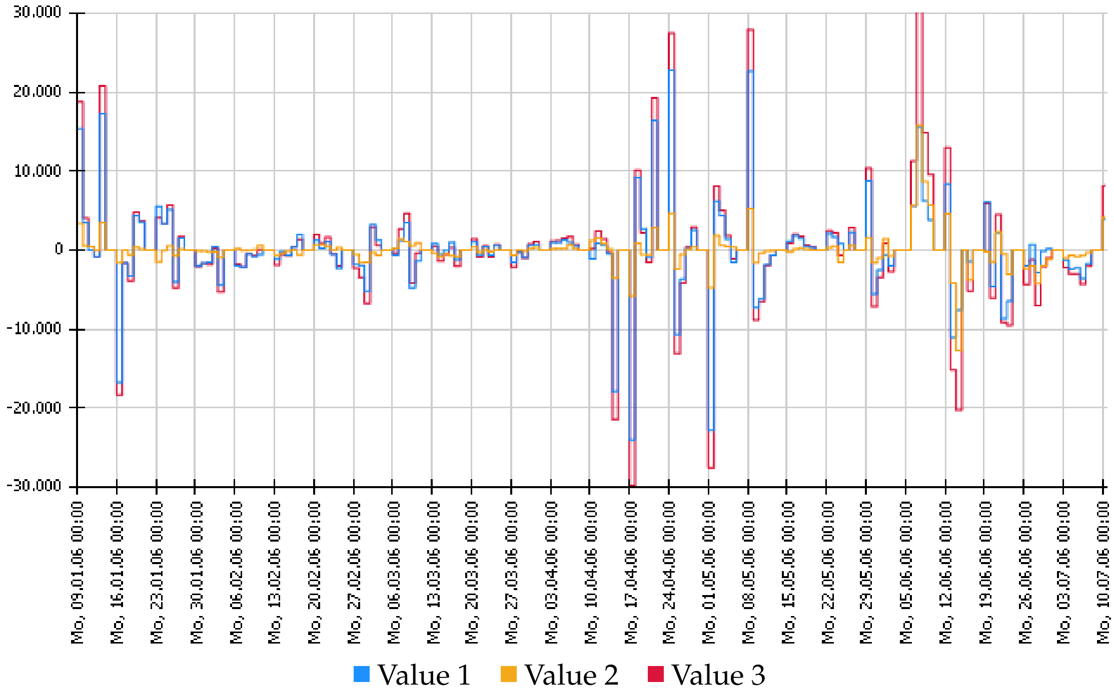


Figure 9.19: Comparison with Week Before. The difference with the same day of week in the prior week is calculated for each day. This visualization shows the method for three separate variables in one visualization. The visualization shows working hours of engine drivers classified according to task.

light areas. We have used an algorithm that calculates a new order of visualizations and minimizes the sum of the differences between two neighbored visualizations. The resulting order is shown in Table 9.18

The algorithm also considers the difference between the first and the last visualization in the table. However, this cannot be seen in a linear layout. Therefore, we have ordered the visualizations in a circle (see Figure 9.20). As the circle organization does not allow for a detail level as high as the tables, we aggregate the details to measures of similarity. For each area that has the same state of light or dark as the neighbored visualization, the connecting line between the visualization has its thickness increased by 1 pt[2]. This is done separately for tasks and visualization process aspects.

First we look at the orange connections that represent the tasks. The resulting visualization shows that most lines are rather thick, but some are quite thin. These thin lines can be used to make a discintion and to formulate several classes of visualizations:

[2]1 pt (point) is 0.3528 mm.

Table 9.16: The Visualization of Differences to Week Before in Figure 9.19 Described according to the Taxonomy from Chapter 8

Tasks: Detect trends; detect irregularities; compare granules on statistical level

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data
	Primitives	Instant-based
Data Transformation	Granularities	Day, week, month, year
	Granularity Contexts	Day of week, day of month, week total, month of year, year total
	Granularity Labels	Ordinal strings for day of week, discrete numbers for every other granularity
Analytical Abstraction Stage	Rasterization	3 times aggregate mean for each day/week combination
Analytical Abstraction Stage	Calculation of Difference	3 times calculate difference between weeks for days
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	3 times abstract value data
Visual Mapping Transformation	Vertical position	Data aggregated to day/week combination, differences
	Hue	Fixed
	Connectivity	Range
	Horizontal position	Data, granularity label: day/week combination
	Data value distribution: value axis, vertical	Data aspect: native
	Data value distribution: value axis, horizontal	Calendar: day of week, day of month, month of year, year total
View Stage	–	Draw the 3 separate visualizations in one space

Table 9.17: The Visualization Methods in Overview. The visualizations are ordered manually. The order is the same as in this chapter. Dark fields represent that an aspect is present, light fields represent that an aspect is not present in the according visualization.

Visualization	Task											Primitives		Abstraction Stages					Visual Mapping							
	Detect Cycles	Interpret Cycles	Detect Trends	Detect Patterns	Classify Usual Values	Detect Irregularities	Detect Complex Structures	Compare Granules	Compare Granule Combinations	Compare Granules Statistically	Read Values	Instant-based	Interval-based	Rasterization	Aggregation	Filtering	Calculation of Difference	Other	Time-oriented to Position	Time-oriented to Angle	Time-oriented to Color	DataVvalue to Position	Data Value to Color	Data Value to Size	Apply Relative Positions	More than one Visual Mapping
Repeated Time Scale Graph	■		■					■			■	■		■					■			■				
Seasonal Graph	■		■					■			■	■		■	■				■			■			■	■
Cycle Plot		■	■					■			■	■		■	■				■			■			■	■
Event Bands	■			■		■	■						■						■						■	
Pixel-based Recursive Pattern	■			■		■	■					■		■	■				■				■		■	
Multi-scale Visualization	■			■		■	■					■		■		■			■				■		■	■
Spiral Layout Blots	■			■		■						■		■					■	■				■		
Spiral Layout Intervals	■			■		■							■	■					■	■			■			
Concentric Circles	■			■		■					■	■		■					■	■				■		
Spatio-temporal Treemap				■				■	■		■	■		■	■			■	■				■	■		
Pivot Data over Days		■				■		■			■	■		■					■		■	■				
Pivot Data over Holidays		■				■		■			■	■		■					■		■	■				
Stacked Granularities						■		■			■	■		■					■		■	■		■	■	
Spread per Granule						■		■		■	■	■		■					■		■	■			■	
Comparison with Past Values						■		■		■	■	■			■				■		■	■				
Difference to Week before						■		■		■		■		■			■		■			■	■			

Table 9.18: The Visualization Methods in Overview. The visualizations are ordered by an algorithm that minimizes the differences between neighbored visualizations. Dark fields represent that an aspect is present, light fields represent that an aspect is not present in the according visualization.

Visualization	**Task**											**Primitives**		**Abstraction Stages**					**Visual Mapping**							
	Detect Cycles	Interpret Cycles	Detect Trends	Detect Patterns	Classify Usual Values	Detect Irregularities	Detect Complex Structures	Compare Granules	Compare Granule Combinations	Compare Granules Statistically	Read Values	Instant-based	Interval-based	Rasterization	Aggregation	Filtering	Calculation of Difference	Other	Time-oriented to Position	Time-oriented to Angle	Time-oriented to Color	DataVvalue to Position	Data Value to Color	Data Value to Size	Apply Relative Positions	More than one Visual Mapping
Cycle Plot		■	■					■			■	■		■	■				■			■			■	■
Seasonal Graph	■		■					■			■	■		■	■				■			■			■	■
Repeated Time Scale Graph	■		■					■			■	■		■					■			■				
Event Bands	■			■		■	■						■						■						■	
Pixel-based Recursive Pattern	■			■		■	■					■		■	■				■				■		■	
Multi-scale Visualization	■			■		■	■					■		■		■			■				■		■	■
Spiral Layout Intervals	■			■		■							■	■					■	■			■			
Spiral Layout Blots	■			■		■						■		■					■	■				■		
Concentric Circles	■			■		■					■	■		■					■	■				■		
Spatio-temporal Treemap				■				■	■		■	■		■	■			■	■				■	■		
Pivot Data over Days		■				■		■			■	■		■					■		■	■				
Pivot Data over Holidays		■				■		■			■	■		■					■		■	■				
Stacked Granularities						■		■			■	■		■					■		■	■		■	■	
Spread per Granule						■		■		■	■	■		■					■		■	■			■	
Comparison with Past Values						■		■		■	■	■			■				■		■	■				
Difference to Week before						■		■		■		■		■			■		■			■	■			

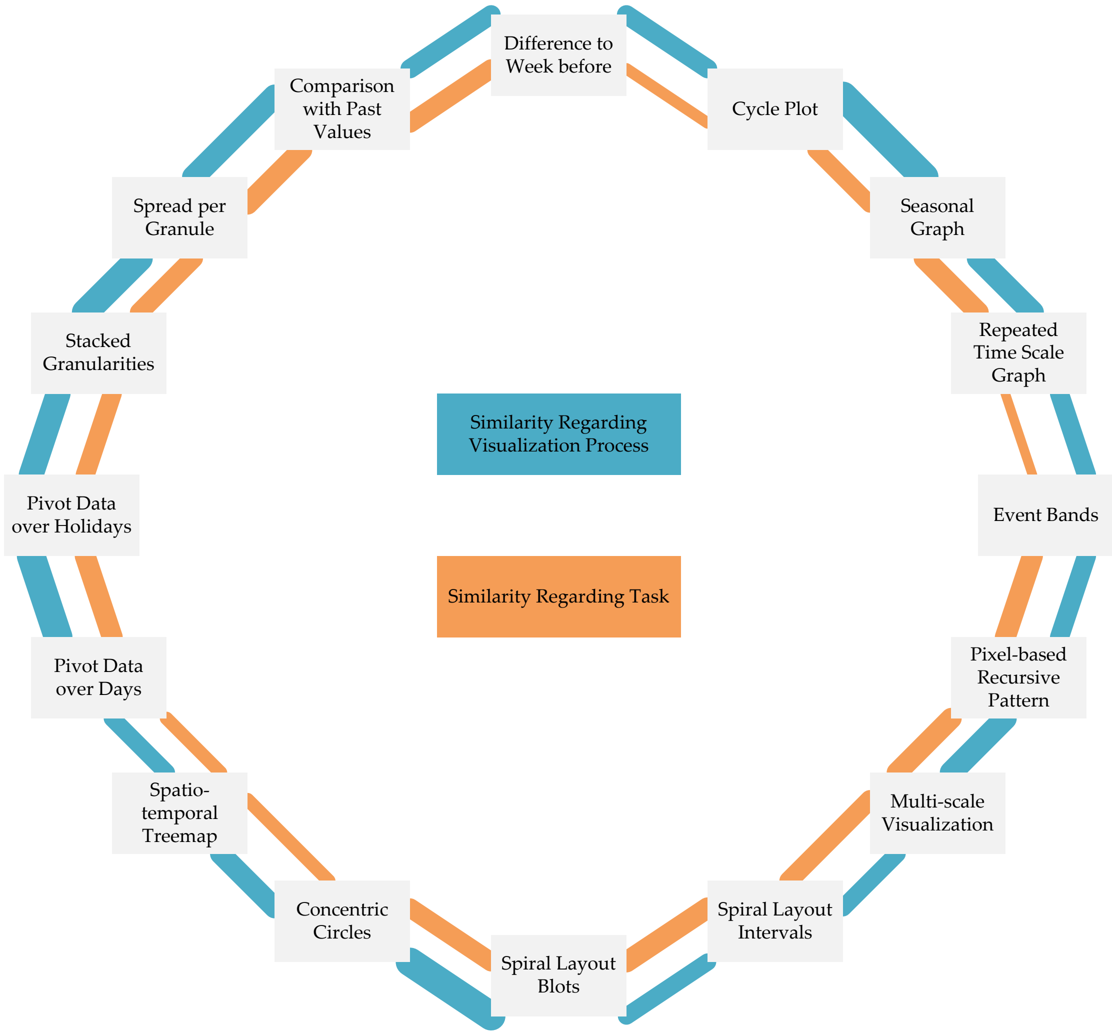


Figure 9.20: The Visualization Methods in Overview. The lines between the visualization methods represent similarity. Thicker lines connote more similarities.

- Cycle Plot/Seasonal Graph/Repeated Time Scale Graph
- Pixel-based visualizations and visualizations using spirals or circular layouts
- Spatio-temporal Treemap
- Visualizations Used by Business Analysts

Regarding the visualization process, it is more difficult to make a distinction. One possibility is to discern between visualizations that map the non-time-oriented-aspects to position and those that map them to color. Furthermore, Figure 9.20 shows connections that are somewhat blurred regarding the circular structures and spirals. We consider that an artifact of our choice of order, because the distinction between those two attributes cannot clearly be made. Still, we think that our visualizations can give a helpful overview on the design space regarding the structure of time.

9.3 Summary

We have presented numerous visualization techniques that employ the structure of time. We have also listed the tasks for each visualization that we consider them most adequate to solve. Furthermore, we have visually shown the similarities and differences between visualizations:

- Visual variables used in the visualizations;
- Analytical abstractions performed in the visualization processes;
- Tasks the visualizations are well-suited for.

Based on these similarities or differences, we have developed Figure 9.20 which shows how similar neighbored visualizations are regarding to their tasks and their design characteristics. Therefore, the visualization also shows at which visualizations similar design characterists are used for similar tasks, or whether there are differences between those two aspects.

—*In the present state of the art this is all that can be done.*

H.H. Suplee

10
State of the Art: Summary

In Part II, we have presented a novel taxonomy for visualizations that employ the structure of time. The taxonomy is not part of existing state of the art but a new development for this thesis. It makes use of the process-oriented framework from Chapter 3 in order to distinguish visualizations at several levels. Moreover, we have used the taxonomy to describe various state-of-the-art visualizations.

The descriptions reveals a final distinction a visualization can have. (1) There are visualizations that usually stem from InfoVis literature and have their focus on the novelty in presenting data and often also user interaction. While those visualizations make use of the structure of time, it is often not the focus, but rather another concept that can be used to make the visualization work. (2) Other visualizations, mainly from practical appliance, often have a big focus on using the structure of time to get more out of the data. They have shear limitless variation (we only showed a small number of variants) but are usually adapted from basic visualizations that are used for a long time now.

Overall, we hav shown in Part II that the structure of time, especially the calendar aspect, is already used successfully in powerful visualizations that help users solve tasks that would be much more difficult without using the structure of time.

However, there are a number of pitfalls in current visualizations, as well as opportunities not grasped yet:

1. Many visualizations presented here have no interaction methods or only rudimentary interaction methods that do not employ the structure of time. Especially

dynamic switching of granularities opens up the possiblity for new powerful semantic interactions.

2. The visualizations we presented lack the ability to disclose which values regarding the structure of time are “common”. For example, whether a tasks usually takes days or weeks to perform.

3. Several of the visualizations presented in Part II are classic visualizations which have been enhanced by integrating support for the structure of time. Similar to that, more classic visualizations can be enhanced, for example statistical visualization that could solve the task presented in Point 2 of this list.

4. The ways the structure of time has been used to enhance the classic visualizations often is independend from the visualization itself. By applying our process-oriented framework from Chapter 3, we are able to transfer those operations from one visualization to another, and make, for example, a combination of Cycle Plot and Multi-scale visualization.

5. Some visualizations could be useful for even more tasks than they currently are if they were easier to interpret for users. For example, the Multi-scale visualization could also be used to compare granules, even on different levels combined, and to interpret cycles rather than only detect them. However, our user studies showed that users still have difficulties grasping the recursive pattern arrangement, preventing them to pursue those more complex tasks.

6. As state before, the calendar aspect is by far dominant in visualizations making use of the structure of time. In real-world tasks we were also faced with the problem of dealing with multiple views from multiple points in time visually.

7. Visualizations are often described independently from their context. In the real world, InfoVis is increasingly often only part of a larger Visual Analytics concept. Even in focusing on interactive visualizations, it can help to widen the approach and also pay heed to automated analysis algorithms working together with the visualization.

In the upcoming Part III, we attend to these points. We combine visualizations, interactions, and other methods from InfoVis literature with visualizations, interactions, and other methods from real-world application. Methods regarding the structure of time can be transferred freely using our framework. Furthermore, we include innovative user interactions focused on the structure of time. We make visualizations more understandable for users, explore more aspects of the structure of time, and provide our work in a larger Visual Analytics context.

Part III

Novel Interactive Visualization Methods

—The most exciting phrase to hear in science, the one that heralds new discoveries, is not "Eureka!" but "That's funny..."

Isaac Asimov

11 Novel Interactive Visualization Methods: Introduction

The development of the visualization described in Part III of this thesis have been driven by two main reasons: (1) The pitfalls and opportunities described in Chapter 10 and (2) our users' needs described in Chapter 5.

As a reason to develop GROOVE (see Chapter 12), we had to design appropriate visualizations that cope well with large time-oriented datasets in an intuitive and easily interpretable way by exploiting the inherent structures of time. Multi-scale visualizations using the recursive pattern arrangement (see Chapter 9) are appropriate for that, but there is room for many improvements and novel developments.

Other visualization methods also can be improved by including the structure of time. Especially the range of methods to analyze statistical aspects of calendar-based structures in the data still has room for more innovative developments. Those are shown in Chapter 13.

For some datasets and tasks, domain knowledge and social calendar structures are not suitable. Structures in the data might have been caused by reasons human users yet have to understand. For those situations, we propose the definitions of granularities based on automated methods. We show their application for GROOVE in Chapter 14 and 16.

Automated methods have also been a central topic in the project **DisCō**. As a Visual Analytics project, **DisCō** required us to find interactive visualizations that not only show data but also allow users to intuitively interact with a complex Data Mining algorithm. Chapter 15 shows interactive visualizations we developed to solve those tasks. We also explain how we have been able to employ the structure of time in those visualizations.

While have been able to perform detailed user evaluations for GROOVE, due to time constraints only expert assessments could be obtained for the other visualizations. Beside our own assessments we got detailed feedback from an InfoVis expert who does not participate in the **DisCō** project. This expert (in the following: "the external InfoVis expert") has not seen our work prior to providing his assessment.

—I want to know God's thoughts; the rest are details.

Albert Einstein

12 Granularity Overview Overlay Visualizations

The first kind of novel visualizations we developed can be summarized as granularity overview overlay (GROOVE) visualizations. Our approach enhances pixel-based visualizations to overcome present limitations and provides additional features for the exploration of time-oriented data.

12.1 Design Requirements

As described in Chapter 5, our users have to (1) gain an overview of the data set, (2) identify relevant and define specific time granularities (e.g., one business day can last from 6am to 6pm, from midnight to midnight, etc.), and (3) find anomalies and relevant patterns, trends, and relations within this data set.

For the first step it is necessary to provide an overview at an appropriate aggregation level, as the following example illustrates:

> A business consultant analyzed a dataset from cash registers of a supermarket. He did not find any patterns using line plots of aggregated daily data. By visualizing the whole data set, he recognized unexpected anomalies that could be tracked down to temporally displaced accountings on some days.

Therefore, we concluded that aggregation should be kept at a minimum, as it can easily hide important facts. We also encountered that automated analysis is not a silver bullet, as it might miss information in the data that is unexpected and not covered in the search parameters.

For the second step, the Visual Analytics method has to build upon those time granularities that are relevant for the task at hand, and to provide smart control for them, as users work with time-oriented data of different granularities with inherent social and natural time structures. The Visual Analytics method should disclose patterns that are related to as many granularities as possible and it should be configurable freely concerning these granularities. User interaction with fast responses has a top priority regarding the granularity configuration, as an understanding of datasets should evolve quickly.

For the third step, users can identify anomalies in the data set to identify "special events", data errors, and outliers and react on them by transformations of the visualization. Additionally, the Visual Analytics method should enable users to formulate hypotheses on patterns, relations, and trends.

Taking into account the definitions by Mackinlay [Mackinlay, 1986], the visualization has to be expressive regarding the structures of time and the data. Owing to the complexity of the structures of time, in most cases it is impossible to express all aspects entirely as static visualization. We combine two methods to overcome this problem. The amount of complexity in one view is maximized by using Tufte's principle of micro-macro reading [Tufte, 1990]. To further increase expressiveness, we resort to user interaction.

12.2 GROOVE Concepts

GROOVE visualizations are pixel-based visualizations. They are related to the recursive pattern arrangement by Keim et al. [1995] (see Subsection 9.1.5) and the Multi-scale visualization by Shimabukuro et al. [2004] (see Subsection 9.1.6).

Our users reported that they need an arrangement for comparing not only values next to each other, but also corresponding values among different granules of all the granularities used. In other words, the advantage of a meaningful arrangement exceeds the advantage of close pixels being related to close data elements. Therefore, while all arrangements by Keim et al. have back-and-forth arrangement at some level, we recommend to also explore recursive pattern arrangement that strictly left-to-right when developing a new visualization, like Shimabukuro et al. [2004] do with their Multi-scale visualization. In the context of time-oriented data, such an arrangement is also easier to understand, as it is similar to the arrangement of socially established calendars.

Figure 12.1 shows how four different granularities are used to order pixels recursively. In this example, such a task would be comparing the first months of each period of the winter semester with each other.

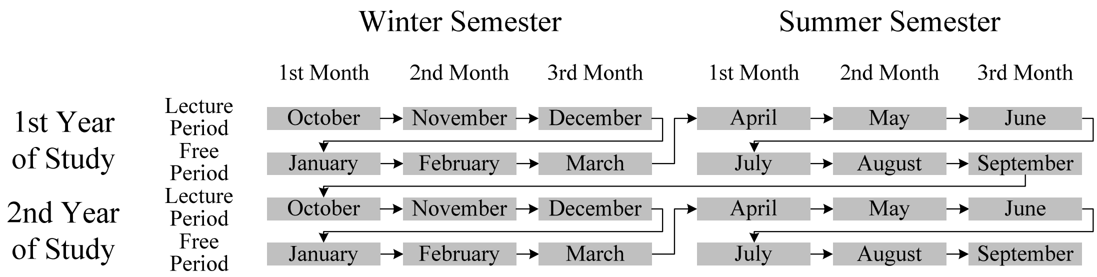


Figure 12.1: An Example how Recursive Patterns Are Used to Order Data Points. Each block represents one semester, ordered by month of period and period of semester. The blocks are ordered according to semester of study year and year of studies.

Even with recursive pattern arrangement, it is important for users to have a guideline to better discern the granularities. In compliance with the data-ink-ratio introduced by Tufte [1983], we wanted this guide to provide further information about the dataset itself. This need is fulfilled in the tripartite Multi-scale visualizations, which have been developed by Shimabukuro et al. [2004]. They use a parallel overview of average values as a guidance and information source as well. However, the overview is not optimal as a guidance, as it is spatially apart from the detail. The eye of the beholder constantly has to jump between the two parts, which is rather a long distance, comprising the danger of mistakes, and straining the working memory. It is also difficult to mentally integrate patterns found among coarser granularities with patterns found among finer granularities. Therefore, we use an overlay between the granularities of overview and detail. The overlay can be realized through one of three basic possibilities.

12.2.1 Color-based Overlay

A color can be deconstructed in different ways (see Subsection 3.1.2). One possibility is a composition of hue, chroma, and lightness in CIELUV color space. By mapping the overview value on one aspect of the color (e.g., the hue) and the detail value on another one (e.g., the lightness), it is possible not only to overlay both levels, but, at the same time, do halve the space needed to show the visualization. This visualization supports users in detecting patterns on an overview and detail level. The fact that the extraction of exact data values is very difficult does not impede this task [Smuc et al., 2008].

An example of this kind of overlay is shown in Figure 12.2. Daily turnover data from a shop has been plotted for one year, each block depicting one month. The hue varies from blue to red for lower or higher monthly averages. The lightness for a particular day is higher in case the turnover of that day is higher. For illustration purposes, we have kept the arrangement simple and the amount of visualized data low. Empty space emerges due to the irregularities imposed by having to combine the granularities week and month.

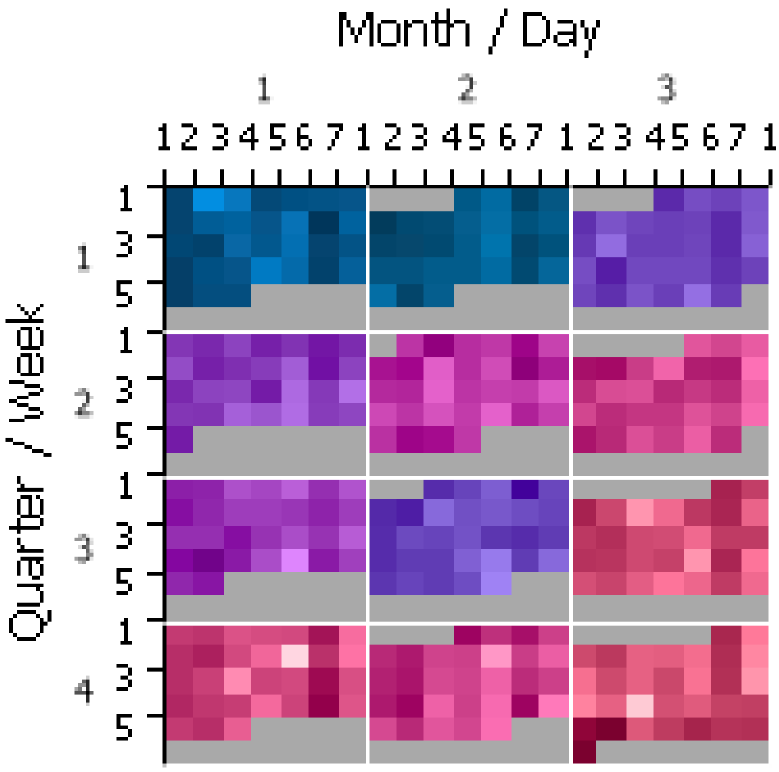


Figure 12.2: Color-based Overlay. Low average values are displayed as blue hues, high average values as red hues. Low detail values are displayed as dark pixels, high detail values as light pixels.

Table 12.1 shows the characteristics of the GROOVE visualization in Figure 12.2 according to our taxonomy from Chapter 8. Like for the Multi-scale visualization, we model the multiple scales by defining several separate visualizations, named areas in Tables 12.1–12.3, that are later combined into one visualization. Like in Chapter 9, the tables in this chapter only represent the very visualizations shown in the figures. As GROOVE visualizations have interactions methods that allow for a great deal of recon-

Table 12.1: The GROOVE Visualization in Figure 12.2 Described according to the Taxonomy from Chapter 8

Tasks: Detect cycles; interpret cycles; detect trends; detect patterns; detect irregularities; detect regular structures more complex than cycles; compare granules; compare granule combinations

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data
	Primitives	Instant-based
Data Transformation	Granularities	Day, week, month, quarter
	Granularity Contexts	Day of week, week of month, month of quarter, quarter of year
	Granularity Labels	Discrete numbers for all granularities
Analytical Abstraction Stage	Rasterization	Aggregate mean for each day/week/month/quarter combination
	Aggregation	Aggregate mean for each month/quarter combination
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	2 times overlaid abstract value data
Visual Mapping Transformation	Base vertical position (area 1)	Data, granularity label: quarter of year
	Base horizontal position (area 1)	Data, granularity label: month of quarter
	Relative vertical position (area 1)	Data, granularity label: week of month
	Relative horizontal position (area 1)	Data, granularity label: day of week
	Lightness (area 1)	Data aspect, rasterized to day/week/month/quarter combination
	Absolute vertical position (area 2)	Data, granularity label: quarter of year
	Absolute horizontal position (area 2)	Data, granularity label: month of quarter
	Hue (area 2)	Data aspect, aggregated to month/quarter combination
	Data value distribution: value axis, horizontal outer	Calendar: month of quarter
	Data value distribution: value axis, horizontal inner	Calendar: day of week
	Data value distribution: value axis, vertical outer	Calendar: week of month
	Data value distribution: value axis, vertical inner	Calendar: quarter of year
View Stage	Combination	Overlay areas by forming colors off hue and lightness

figuration, especially regarding the granularities used, they are only limited representatives, but including interactions in the tables would have made them too complex at the present state of our taxonomy.

12.2.2 Opacity Overlay

Several users prefer color palettes that require more than one aspect of colors to be used [Smuc et al., 2008] (e.g., a palette that reflects the glow of heated metal, ranging from black over dark red and orange to yellow and white). Also, some datasets are especially suited for diverging palettes. In such cases, the use of color aspects for the overlay complicates the visualization. Alternatively to color-based overlay, we have included the possibility of using transparency for the overlay. The level of transparency can be adjusted by users interactively: The overview is always drawn. In the same space, the details are drawn using the same color palette, but semi-transparent. Users can freely adjust the level of opacity, shifting the logical focus back and forth between overview and detail without having to shift their visual focus. An example visualization, using the same data and arrangement as Figure 12.2, is shown in Figure 12.3. As it is not possible to show the interactivity necessary for this GROOVE variant in print, we can only provide several examples. The opacity in Figure 12.3 amounts to 0 (only the average values are shown), 0.5 (the color for each pixel is determined to 50% by the average value and to 50% by the detail value), and 1 (only the detail values are shown). It is difficult to impart the strengths of this variant in a static view, as the visualization is highly dependent on user interaction.

Table 12.2 shows the characteristics of the GROOVE visualization in Figure 12.3 (under the assumption it would be interactive and not the statically rendered version shown in print) according to our taxonomy from Chapter 8.

12.2.3 Spatial Overlay

Instead of showing overview and detail in exactly the same space, it is also possible to show them spatially very close to each other. In this GROOVE variant, the detail view of each block is surrounded by a border in a color based on the overview value. Thereby, no restrictions are imposed on the palette. The visualization does not depend on a user interaction like the one from Subsection 12.2.2, but is very suitable for the ones presented in Subsection 12.2.4.

Figure 12.4 shows the dataset and arrangement from Figure 12.2 and Figure 12.3, this time using spatial overlay. Each month is surrounded by a border that visually encodes the average monthly value. In addition, we have also added weekly averages in this visualization, shown as bars above each week. Below each of these bars, the days of that week are shown in a row. We have added labels inside the visualization as an additional guideline for understanding.

This GROOVE variant requires considerably more space than the other variants, thus constraining the amount of data that can be visualized. On the other hand, the color coding is easier with spatial overlay. Therefore, we consider this variant as most comparable to the Multi-scale visualization by Shimabukuro et al. [2004]. The innovative arrangement, however, makes the integration of overview and detail much easier for users.

Table 12.3 shows the characteristics of the GROOVE visualization in Figure 12.4 according to our taxonomy from Chapter 8.

12.2.4 Using Granularities to Model Visual Operations Semantically

Based on the initial recursive pattern arrangement we used for GROOVE, users can gain new insights into different temporal patterns by interactively changing the visual mapping strategies of granularities, either using the same granularities, or different ones. Thereby, different well-known or novel visual responses can be triggered. These operations can all be performed interactively as reaction to user input.

Comparing Accordant Granules of one Granularity among Different Granules of another one

In the Figures 12.2–12.4, it is relatively easy to compare one day of week with the same day of week in the weeks before and after. When comparing over the borders of months, the days of week are easier to compare with the days of week of the equivalent month in another quarter. For example, the Mondays of the first quarter of month can be compared for all quarters. Another use case would be comparing the days of week for the very next month. This can be done by exchanging the granularities in mapping, for example day of week and week of month. The change in visual mapping results in the same effect a rotation of each month by 90 degrees combined with a vertical swapping would have (see Figure 12.5). By doing it using visual mapping, it becomes a semantic instead of a visual operation.

Masking or Collapsing

Comparing accordant granules can be more difficult when unimportant granules, like Saturday and Sunday, are "in the way" and even take up additional space. Granularities such as business week, leaving out weekends or even holidays, are a possible solution for this (see Section 4.2). The granules unimportant for users can be mapped to a neutral color or, when the arrangement allows it, even be left out completely by

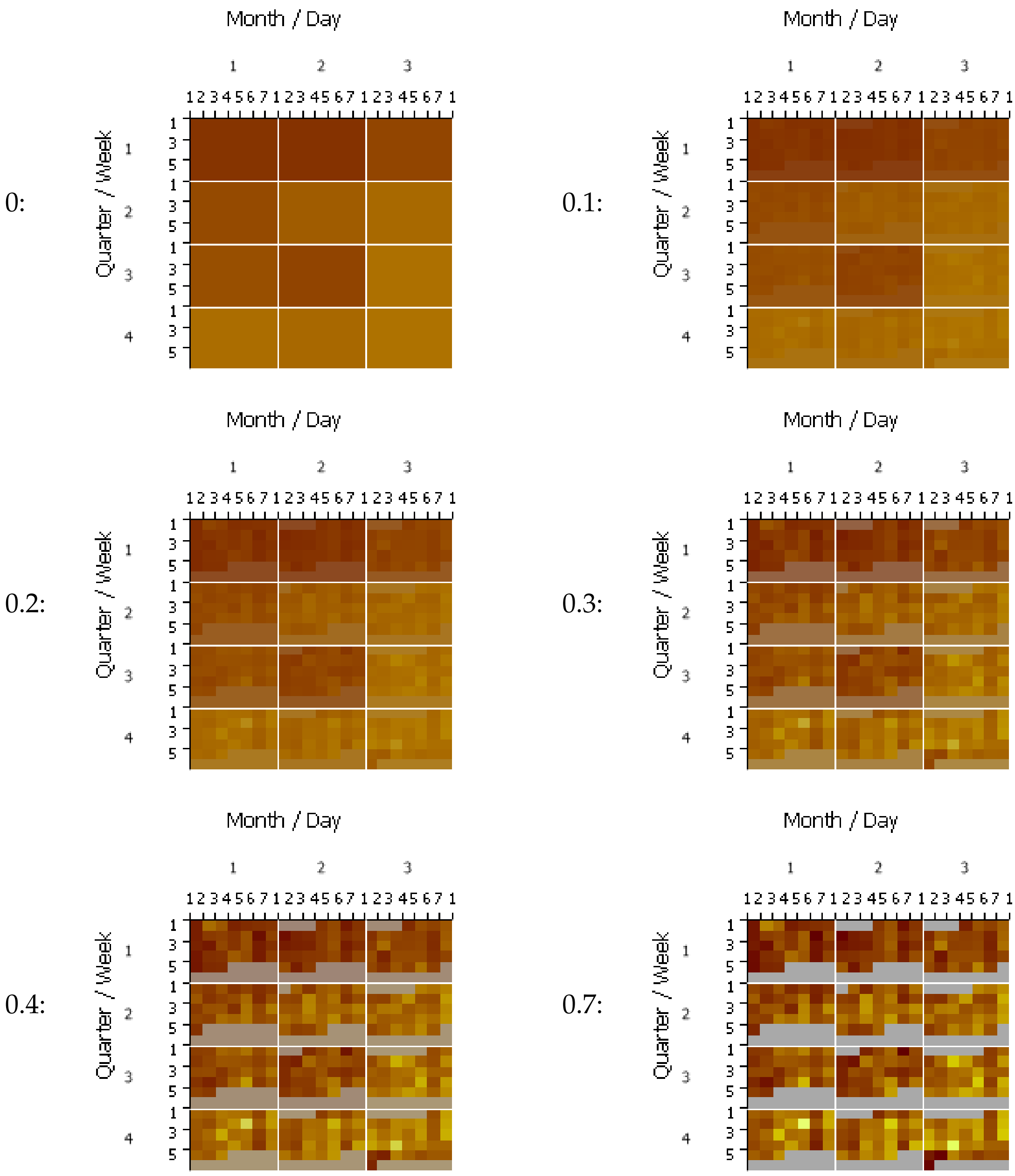


Figure 12.3: Opacity Overlay. Dark red represents low and light yellow represents high values. In this example, a monthly overview is combined with daily details. Normally, the opacity would be set by user interaction. We examplary show the opacities of 0, 0.1, 0.2, 0.4, 0.7, and 1.

Table 12.2: The GROOVE Visualization in Figure 12.3 Described according to the Taxonomy from Chapter 8

Tasks: Detect cycles; interpret cycles; detect trends; detect patterns; detect irregularities; detect regular structures more complex than cycles; compare granules; compare granule combinations

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data
	Primitives	Instant-based
Data Transformation	Granularities	Day, week, month, quarter
	Granularity Contexts	Day of week, week of month, month of quarter, quarter of year
	Granularity Labels	Discrete numbers for all granularities
Analytical Abstraction Stage	Rasterization	Aggregate mean for each day/week/month/quarter combination
	Aggregation	Aggregate mean for each month/quarter combination
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	2 times overlaid abstract value data
Visual Mapping Transformation	Base vertical position (area 1)	Data, granularity label: quarter of year
	Base horizontal position (area 1)	Data, granularity label: month of quarter
	Relative vertical position (area 1)	Data, granularity label: week of month
	Relative horizontal position (area 1)	Data, granularity label: day of week
	Lightness (area 1)	Data aspect, rasterized to day/week/month/quarter combination
	Absolute vertical position (area 2)	Data, granularity label: quarter of year
	Absolute horizontal position (area 2)	Data, granularity label: month of quarter
	Hue (area 2)	Data aspect, aggregated to month/quarter combination
	Data value distribution: value axis, horizontal outer	Calendar: month of quarter
	Data value distribution: value axis, horizontal inner	Calendar: day of week
	Data value distribution: value axis, vertical outer	Calendar: week of month
	Data value distribution: value axis, vertical inner	Calendar: quarter of year
View Stage	Combination	Render area 2 and render area 1 in the same space with opacity based on user interaction

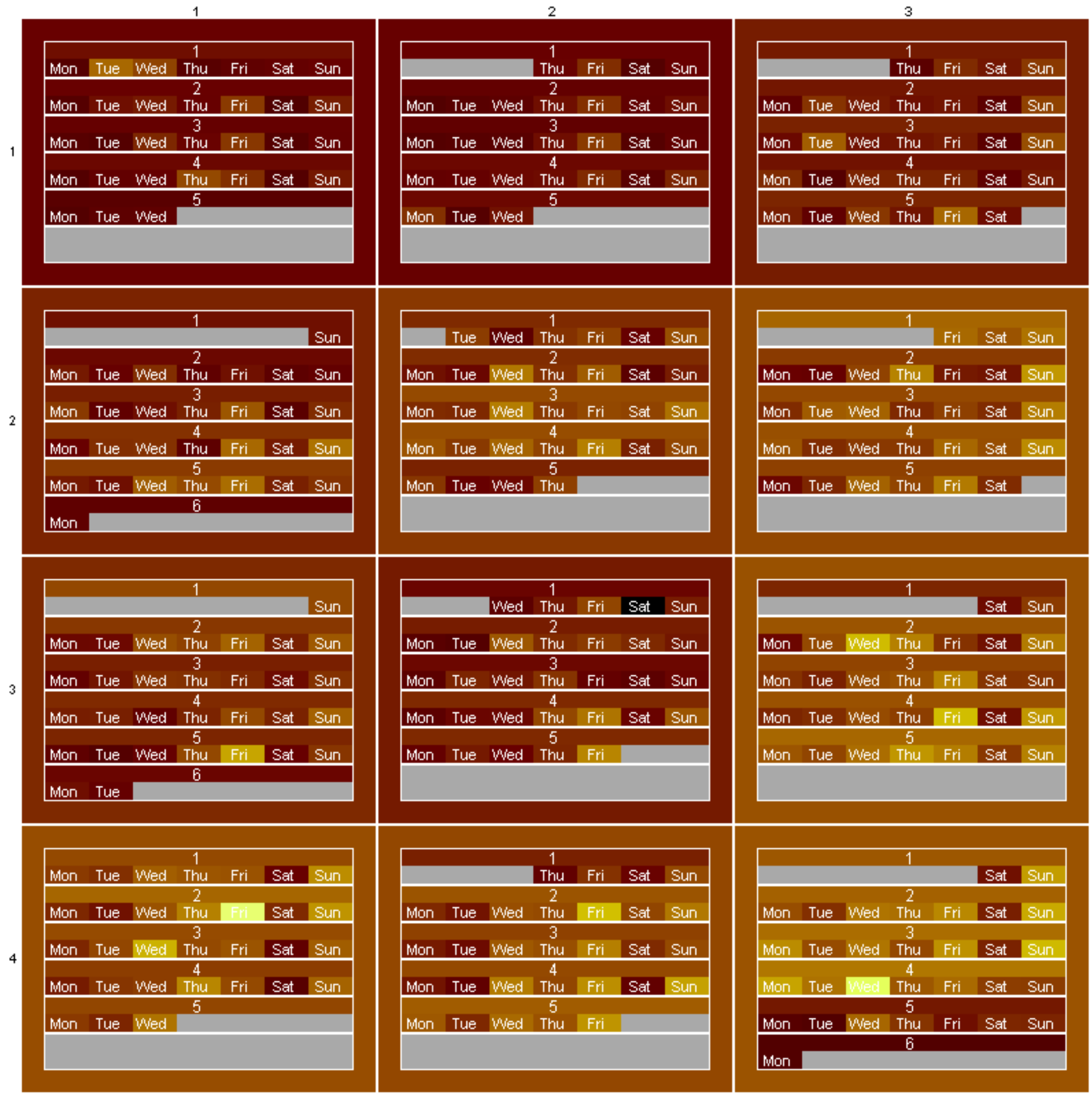


Figure 12.4: Spatial Overlay. Dark red represents low values, light yellow represents high values. Each month is surrounded by a border showing the average value. Each week's average is shown by a bar, with the daily detail below.

Table 12.3: The GROOVE Visualization in Figure 12.4 Described according to the Taxonomy from Chapter 8

Tasks: Detect cycles; interpret cycles; detect trends; detect patterns; detect irregularities; detect regular structures more complex than cycles; compare granules; compare granule combinations

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data
	Primitives	Instant-based
Data Transformation	Granularities	Day, week, month, quarter
	Granularity Contexts	Day of week, week of month, month of quarter, quarter of year
	Granularity Labels	Discrete numbers for all granularities
Analytical Abstraction Stage	Rasterization	Aggregate mean for each day/week/month/quarter combination
	Aggregation	Aggregate mean for each month/quarter combination
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	2 times overlaid abstract value data
Visual Mapping Transformation	Base vertical position (area 1)	Data, granularity label: quarter of year
	Base horizontal position (area 1)	Data, granularity label: month of quarter
	Relative vertical position (area 1)	Data, granularity label: week of month
	Relative horizontal position (area 1)	Data, granularity label: day of week
	Lightness (area 1)	Data aspect, rasterized to day/week/month/quarter combination
	Base vertical position (area 2)	Data, granularity label: quarter of year
	Relative vertical position (area 2)	Data, granularity label: week of month
	Absolute horizontal position (area 2)	Data, granularity label: month of quarter
	Hue (area 2)	Data aspect, aggregated to week/month/quarter combination
	Absolute vertical position (area 3)	Data, granularity label: quarter of year
	Absolute horizontal position (area 3)	Data, granularity label: month of quarter
	Hue (area 3)	Data aspect, aggregated to month/quarter combination
	Data value distribution: value axis, horizontal outer	Calendar: month of quarter
	Data value distribution: value axis, horizontal inner	Calendar: day of week
	Data value distribution: value axis, vertical outer	Calendar: week of month
	Data value distribution: value axis, vertical inner	Calendar: quarter of year
View Stage	Combination	Render area 3; render area 2 in the same space leaving a border for each part of area 3; render area 1 in the same space leaving one row empty between each row

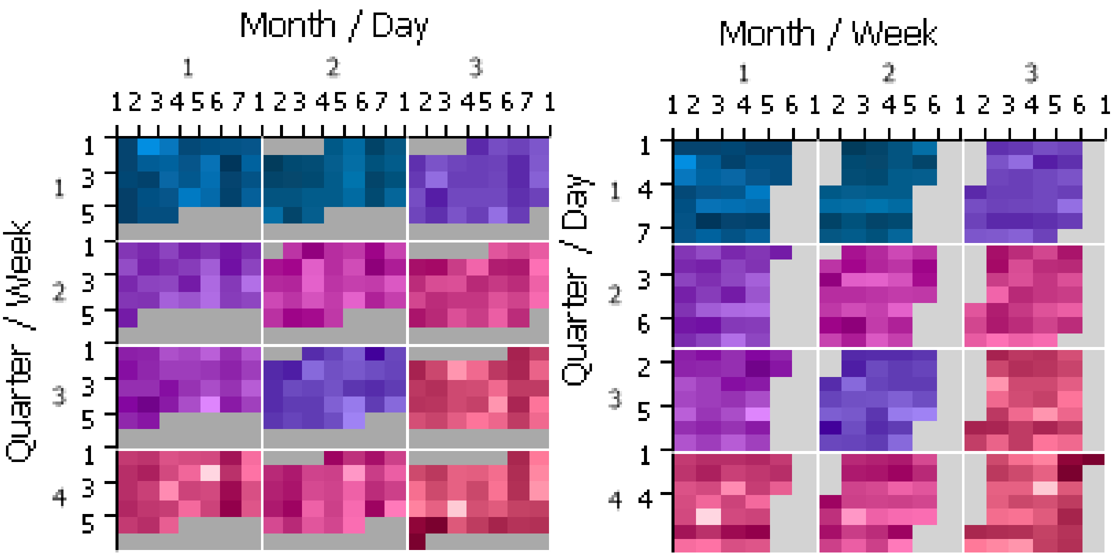


Figure 12.5: Comparing Days of Week with the Same from the Next Month or Next Quarter. Semantic changes result in visual changes, in this case rotation of 90 degrees and vertical swapping.

using a more complex spatial mapping. Thereby, the visual operation of masking or collapsing a certain area can be modelled as a semantic operation.

Scrolling, Panning, and Paging

With GROOVE, instead of using geometric zooming and panning, it is beneficial to use semantic zooming and paging. Paging is performed by defining a granularity coarser than the ones used by one view and showing only one granule at a time (based on user interaction).

Semantic Zoom

Semantic zooming can be performed by switching to coarser or finer combinations of granularities. As stated before, the granularities can be configured freely. However, users can also predefine combinations for quick switching.

Disclosing Cycles and Trends

As the Cycle Plot by Cleveland [1993] (see Subsection 9.1.3) shows, it can be beneficial to change the order of overview and detail. By ordering the blocks according to finer granularities (in our previous examples, that would be day and week) and the values inside the blocks according to coarser granularities (e.g., quarter and month), it is possible to gain new insights. The development over the year becomes visible for each day of month separately. Therefore, monthly cycles become separated from the trend over the year. This is an effect similar to the one of the Cycle Plot [Cleveland, 1993]. Figure 12.6.a shows data from police assignments. For one month, we show the number of assignments being given at different five-minute-intervals. Each block with one hue represents one day. In Figure 12.6.b, the order of granularities has changed. Each block now represents another 5-minute-interval of the day, but it shows the development that 5-minute-interval undergoes over the course of the whole month.

More Levels than Overview and Detail

In the spatial overlay it is possible to treat each pixel on the detail level as a block on its own and drill down to further levels of even finer granularities. It is also possible to combine more than one kind of overlay, for example, using color overlay for a rough overview and spatial overlay for a more detailed overview as well as a detail view. Regarding arrangement, such a three-level overlay is "fully recursive" as defined by Keim et al. [1995]. However, we do not consider recursion to be achieved "fully" at any level, because there is always the theoretical possibility of more levels, as long as there is space left.

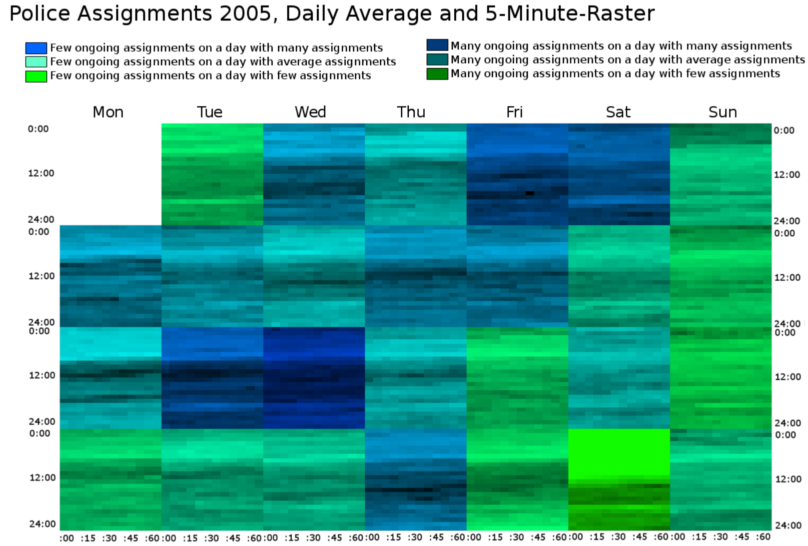
Introducing Completely New Granularities

The flexibility with respect to granularities is important, as the exploration of a dataset is often an iterative process. Assumptions made by users based on observations using one granularity configuration can be hardened or invalidated by switching to a different granularity configuration.

As an alternative to showing all possible data at once, it is possible to follow the Visual Information-Seeking Mantra by Shneiderman [1996]: overview first, zoom and filter, details on demand. It is possible to show only the overview level at first and interactively present the details for all blocks or only for some blocks once the user actively requests them. The details do not need to be presented immediately on a pixel level. It is also possible to switch from overview to showing only the rows of detail level and later switch to showing every pixel.

12.2.5 Mapping Visual Variables Independently

The default GROOVE layouts map the level of arrangement recursion to the same granularities as the level of aggregation. This is a sensible starting point for dealing with a)



b)

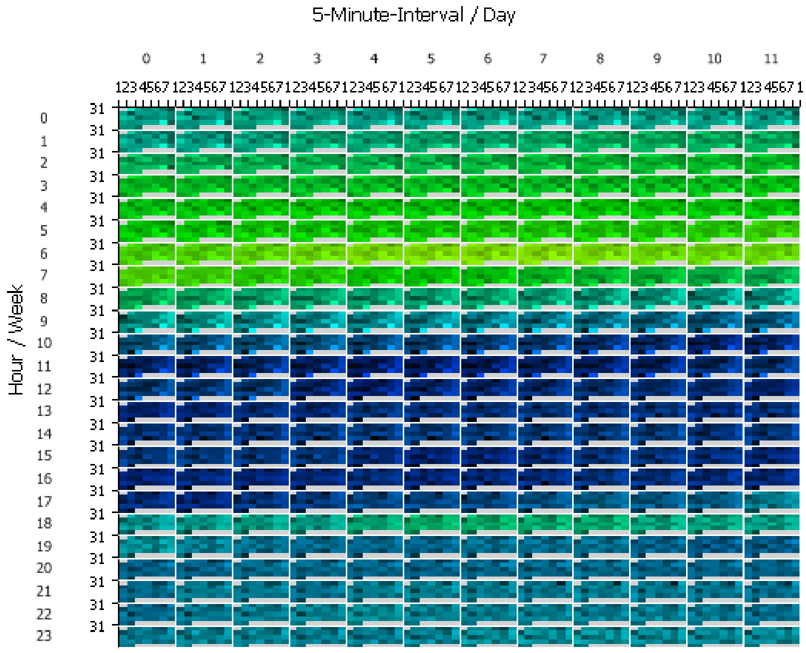


Figure 12.6: Combining Techniques from Cycle Plot with GROOVE. The arrangement below exchanges the order of granularities like a cycle plot does in order to reveal trends.

GROOVE visualizations, because it guides users regarding the recursive arrangement.

However, it is possible to use the separate detail layers for providing novel arrangement variants beside the ones defined by Keim et al. [1995] in their paper on the recursive pattern arrangement.

Above all, we have encountered that some users are in strong disfavour of recursive pattern arrangement altogether. While considering it possible that they might be wrong even about their personal ability to use the visualization, it still remains a good design principle not to patronize users. What our users usually accepted without exceptions was a pixel-based visualization using one level of left-to-right-layout. Such a layout can still contain several levels of overview and detail, for example by (1) calculating the average of each line for the overview level or (2) calculating the average for certain fractions of the line for the overview level.

12.2.6 Other Arrangements than Left-to-right

Initially, we had started research on back-and-forth arrangements like the one proposed by Keim et al. [1995]. However, the acceptance among users was very low, so we did not deepen that research.

Regarding top-down, right-left, or bottom-up arrangements we consider them as equivalent to our left-right arrangement. As Keim et al. already noticed, those are basically the same arrangements adapted for cultures differing from our (the research team's) left-right culture.

12.2.7 Multi-dimensional Datasets

We have also considered applying the ideas from GROOVE to the visualization of multi-dimensional datasets. The easiest way to do this is by calculating fractions of two dimensions, like turnover per customer, and visualizing the result. Sometimes, however, it might be necessary to separately visualize two or more separate values for each point in time. One possibility is to use the data headings like granules of a granularity. For example, a one-year dataset with accident counts for male and female drivers could be shown in GROOVE using two rows of blocks with four columns for the seasons. Each block could consist of weeks and days. Such an arrangement makes sense for separate values of similar statistical distribution, especially if the values are to be compared. The task is more difficult for variables that are not related in such a way, like market capitalization and return of a stock. In cases like these, pattern finding can be used on one value, generating a number of classes based on it which are used like granules. The other value is used to determine pixel color.

12.3 Evaluation

We evaluated GROOVE visualizations in two separate user studies. Beside providing incentives for further developments, the user studies provided new research possibilities in information design [Smuc et al., 2008, 2009]. We also pursued a long-time process of active use in real-world scenarios of customer data by our industry collaborator.

12.3.1 User Studies

In an initial user study, we tested the GROOVE variant using color-based overlay (but without interactions at the time of the study) with respect to their intuitive comprehensibility and potential improvements in semi-structured interviews with six potential users. They were asked to think-aloud while exploring the data and to suggest improvements and additional interactions that could optimize their workflow. We analyzed the think-aloud protocols with respect to users' generation of insights. The most promising outcome of these tests is that all users intuitively gained an understanding of GROOVE within short time. The metaphor of a calendar helped users to understand the visual alignment of the different time granularities. We coded and counted users' data insights considering the referred granularity. Users generated insights on an overview and on a detailed level. When we analyzed the time course of users' insights on both levels we found that most users are able to switch between the two levels of detail and overview and that they sometimes even integrate these levels. But there are also some indicators showing that the small number of detailed insights was due to the fact that the users saw the GROOVE variants they were being shown as more an overview than a detail visualization. This is mitigated by the fact that there is the possibility of interactive semantic zooming that we could not use at that time.

The next user study was focused on comparing the interaction of showing detail on demand we included in GROOVE to the parallel views used in the Multi-scale visualization. Therefore, the interactive prototype we used for the study, despite being able to handle spatial overlay, was configured to omit overlay and show overview and detail of each block only sequentially, based on user interaction. The results of the second user study are not fully evaluated, so only preliminary asumptions based on the results can be presented:

- User interaction helps in finding the correct pixel in a recursive pattern arrangement. When asked to find a pixel representing a certain date, users made twice the number of errors when using a parallel Multi-scale visualization than they did when using interaction.
- Users seemed irritated by the static visualization containing a larger number of

pixels. Therefore, they had more problems understand the color scale in a static Multi-scale visualization and less problems when they started with a reduced amount of complexity.

Based on these results, we assume that a combination of GROOVE with user interaction will definitely surpass other, similar visualizations.

12.3.2 GROOVE in Real-world Application

After the initial test implementation of GROOVE that was used to generate the first mock-ups, we developed two interactive working prototypes. One uses color overlay and is part of the Visual Analytics application [TIS]—Time Intelligence Solutions by XIMES GmbH[1], Vienna. The other one uses spatial overlay and has been implemented as a .NET Windows Forms application. It was used for the next user evaluation phase.

The business consultants of XIMES found GROOVE to be a valuable asset to gain much information from a dataset in a short amount of time. GROOVE also enabled them to correct misinterpretations they had made using conventional visualizations. They used GROOVE for several tasks in preprocessing (before trying to understand the data) as well as at later stages of data analysis:

1. Detection of structural changes (e.g., starting at a certain point of a time period, value levels are different for a certain time of day)
2. Finding missing values (which cannot easily be done automated as they are difficult to distinguish from time periods when there was no data to record)
3. Determining whether data has been collected at a wrong time or with wrong time stamps
4. Defining special events (e.g., refitting, stocktaking) when they have not been accounted for in data collection
5. Detecting cycles and trends

The flexibility of granularity definition has been adopted very well, especially regarding the definition of different variants of granularities (e.g., days from 6 am to 6 am). The business consultants also commended the ability to define more abstract granularities like “week type” with granules like “5-day-business-week”, “week with holiday on Friday”, “week with holiday on Monday” and so on, in order to analyze the effect of the holidays on work days.

[1] For more information, see http://www.ximes.com (accessed on April 9th, 2010).

—The nicest thing about standards is that there are so many of them to choose from.

Ken Olsen

13 Integrating the Structure of Time in Widely-used Visualizations

One of the great opportunities of InfoVis is the fact that straying from the default provides new chances—at least regarding the visualizations themselves, not the technical standards to draw them. On the other hand, staying too close to the established stable visualizations can severely reduce one's possiblities—today, many people limit themselves to what Excel can do. As a possibility to balance out this problem, we present ways to use the structure of time in some visualizations that are widely used, but do not consider the structure of time at their present state. As an example task for our visualizations in this chapter, we analyze a dataset that shows information about the duration patients have spent in a hospital.

13.1 Histogram of Granularities

The durations the patients have stayed are rather widely spread. To interpret the data in a social context, it was important to know how the durations are distributed among granularities. Therefore, we devised a histogram of granularities present in the dataset, shown in Figure 13.1. A patient residing in hospital would be counted towards the appropriate bin. For example, someone staying for five days would be counted under days, while someone staying for eight days would be counted under weeks, as he stayed longer than the length of one granule of the granularity week. This method can also be considered as "irregular binning", as the lenght of each bin of the histogram is different.

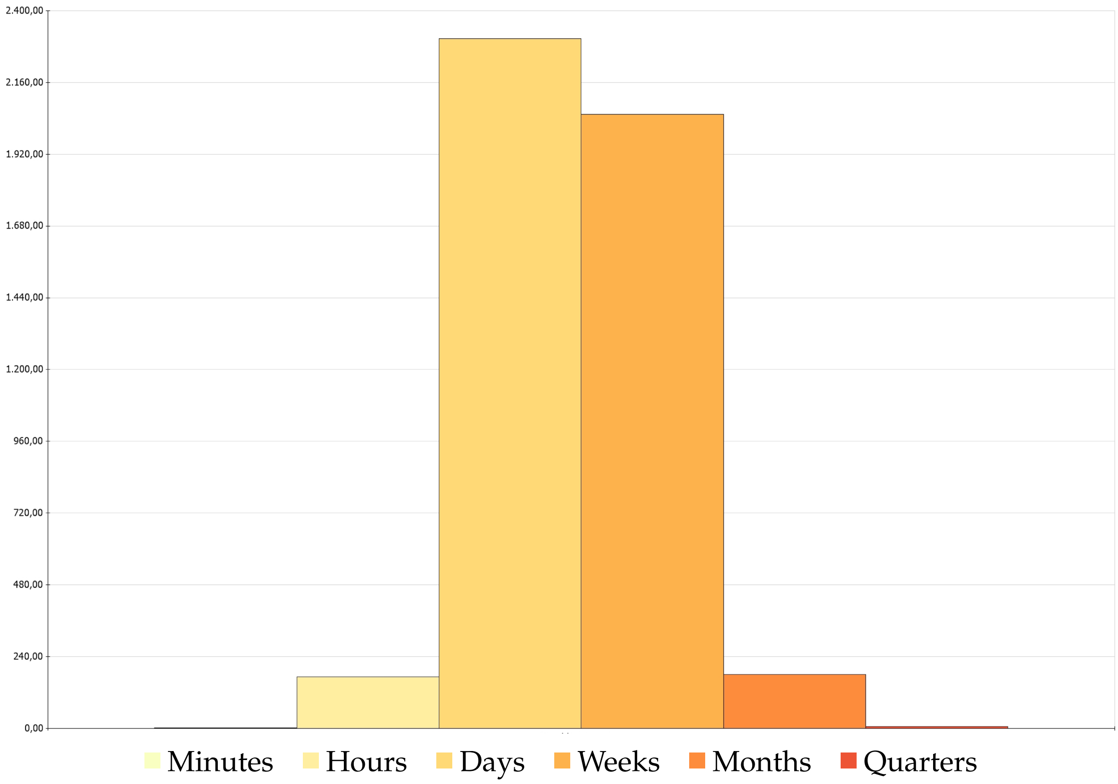


Figure 13.1: Histogram over Granularities. Patients are residing in some hospital a certain amount of time. Each residence is counted towards the coarsest granularity where one granule is shorter than the duration.

Table 13.1 shows the characteristics of the visualization in Figure 13.1 according to our taxonomy from Chapter 8.

13.2 Granularity Scale

The two scales that see most use in line plots and bar plots are the linear scale and the logarithmic scale. The biggest advantage of the linear scale is its understandibility. Values can directly be read and interpreted. However, for values with a larger spreading, like the ones from the hospital dataset in the last section, linear scales can result in smaller values to be too small to even be displayed. The usual alternative is the logarithmic scale. It enables experienced users to read visualizations with values that have

Table 13.1: The visualization in Figure 13.1 Described according to the Taxonomy from Chapter 8

Tasks: Build classes of normal valules; detect irregularities; compare granules; compare granules on statistical level; read values

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data
	Primitives	Interval-based
Data Transformation	Granularities	Minute, hour, day, week, month, quarter
	Granularity Contexts	Minute total, hour total, day total, week total, month total, quarter total
	Granularity Labels	Unknown discrete label for all granularities
Analytical Abstraction Stage	Aggregation	For all granularities, calculate end granule minus start granule; for the coarsest granularity with nonzero result; add one
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	Abstract value data
Visual Mapping Transformation	Horizontal position	Calendar: set of granularities
	Vertical position	Data, per granularitiy: aggregated sum
View Stage	–	–

a large spreading. However, it is difficult to understand correctly for less experienced users. Moreover, it has an inherent Lie Factor[1] when used for linear data. That lie factor might lead to wrong decisions in cases when the less experienced users are also the deciders.

For these reasons, we needed to introduce another scale that was better suited for data with large spreading than a linear scale, but also had a higher understandibility with less experienced users, especially the data analysts we are working with, their customers, and the superiors of the customers.

As a possible solution, we devised the granularity scale. A number of granularities is chosen. To optimize understandibility, a full cycle of granules of a granularity should

[1] The Lie Factor is a measurement introduced by Tufte [1983] and defined as

$$\frac{\text{size of effect shown in graphic}}{\text{size of effect in data}}.$$

It can be considered a necessary evil in some visualizations, but Tufte and Howard strongly critize visualizations where the lie factor is high without a necessity.

form a granule of the next coarser granularity. However, this is not mandatory. The scale starts with granules of the finest granularity. As soon as the total amount of granules matches the duration of one granule of the next coarser granularity, the granules of that granularity are enumerated and so on.

An example is shown in Figure 13.2. In that example we interpolated a month as 30 days and a quarter as 91 days. For eight patients from the dataset already described in the last section, the stay in hospital is shown in detail. The height of a bar shows the total stay based on different granularities. Therefore, it is possible to show patient 6 who only stays one day in the same visualization as patient 5 who stays for one quarter, one month, two weeks, and four days. The colors of the bars are the same as in Figure 13.1 and represent the coarsests granularity of each bar respectively.

Table 13.2 shows the characteristics of the visualization in Figure 13.2 according to our taxonomy from Chapter 8.

We enhanced the concept by using colors in a more complex way (see Figure 13.3). In addition to the base bars, the fraction of a duration that can be measured in finer granularities is shown in finer bars. Therefore, it is much easier to read exact values from the visualization. For example, Patient 5 stays for 1 quarter, 1 month, 2 weeks, and 4 days, easily read from the white grid by adding the partial bars.

Table 13.3 shows the characteristics of the visualization in Figure 13.3 according to our taxonomy from Chapter 8.

13.3 Assessment and Planned Evaluation

The visualizations presented here have been send to our industry collaborateur for assessment. The feedback has been very positive. A visualization similar to the histogram over granularities has been included in their set of templates for analysis of time-oriented data[2]. The integration of the granularity scale is in progress, but the development has been stalled due to technical difficulties. The granularity scale has also been assessed by the external InfoVis expert. He commended it for beeing more understandable than log scale. As the variant with multiple colors in one bar (see Figure 13.3) has been critized by several sources, we have modified it; it has not been assessed since then.

A detailed user study has not been pursued yet. The visualizations are to be included in the next development step regarding our planned Visual Analytics Framework (see Chapter 20) and will be evaluated as part of the user studies performed based on that framework.

[2] Visualizations from this template can be found at http://wiki.ximes.com/index.php?title=TIS:Vorlagen (accesssed at April 12th, 2010).

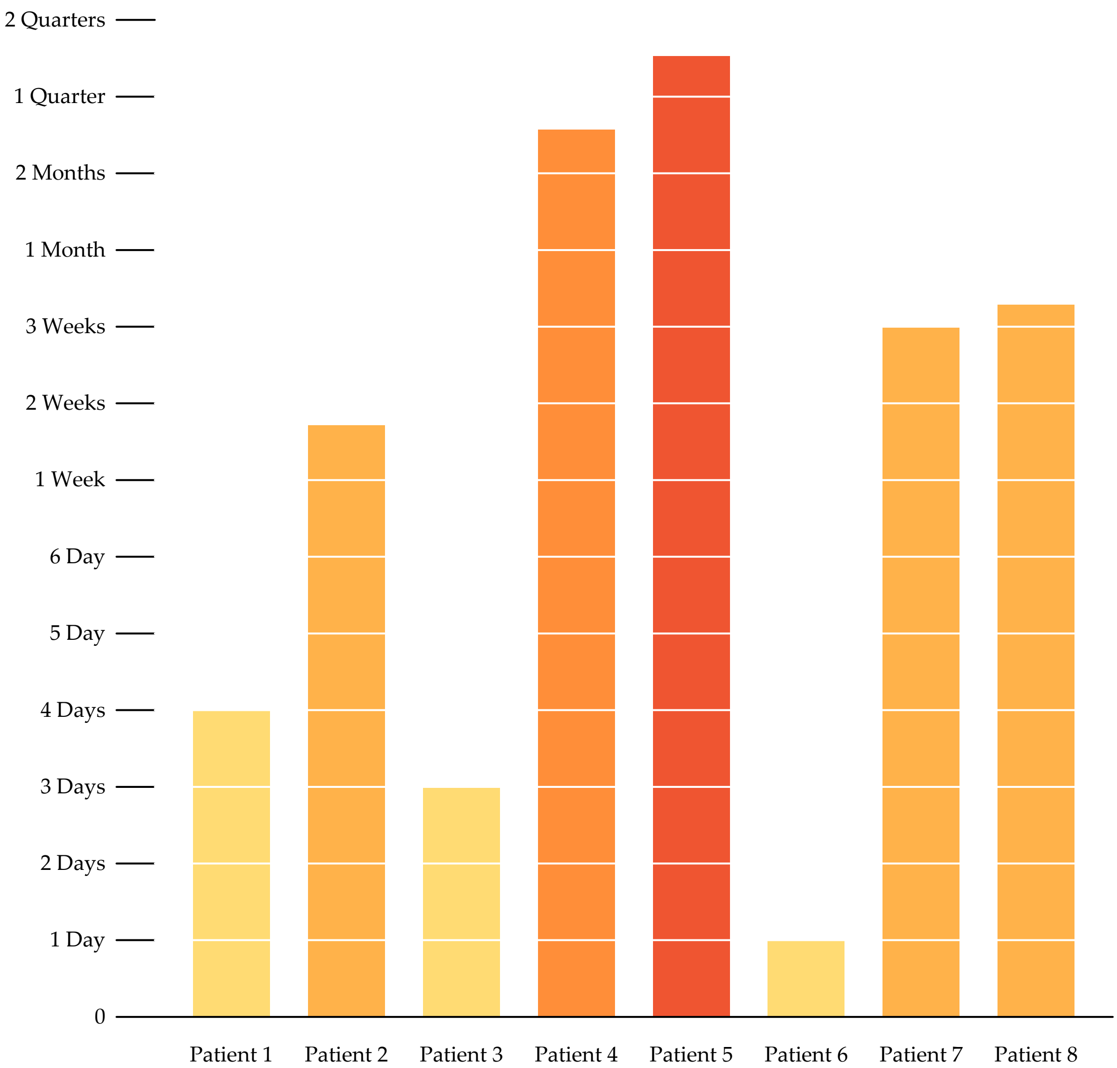


Figure 13.2: Granularity Scales for the Stay of Hospital Patients. The height of a bar represents the total duration of a stay. The scale changes according to the granularities, in upper areas, each step represents a longer period of time.

Table 13.2: The visualization in Figure 13.2 Described according to the Taxonomy from Chapter 8

Tasks: Classify usual values; detect irregularities; compare granules; compare granules on statistical level; read values

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data
	Primitives	Interval-based
Data Transformation	Granularities	Day, week, month, quarter
	Granularity Contexts	Day of week, week of month, month of quarter, quarter of year
	Granularity Labels	Unknown discrete label for all granularities
Analytical Abstraction Stage	Rasterization	Aggregate mean for each day/week/month/quarter combination
Visualization Transformation	Allocation	Allocate data elements to sets according to coarsest nonzero granularity
Visualization Abstraction Stage	Type of visualization	Up to 4 times abstract value data
Visual Mapping Transformation	Horizontal position	Data attribute
	Color	Fixed per part
	Vertical height part 1	Granularity label: day of week
	Vertical base height part 2	Granularity label: week of month
	Vertical relative height part 2	Granularity label: day of week
	Vertical base height part 3	Granularity label: month of quarter
	Vertical relative level 1 height part 3	Granularity label: week of month
	Vertical relative level 2 height part 3	Granularity label: day of week
	Vertical base height part 4	Granularity label: quarter of year
	Vertical relative level 1 height part 4	Granularity label: month of quarter
	Vertical relative level 2 height part 4	Granularity label: week of month
	Vertical relative level 3 height part 4	Granularity label: day of week
View Stage	Combination	Overlay part visualizations

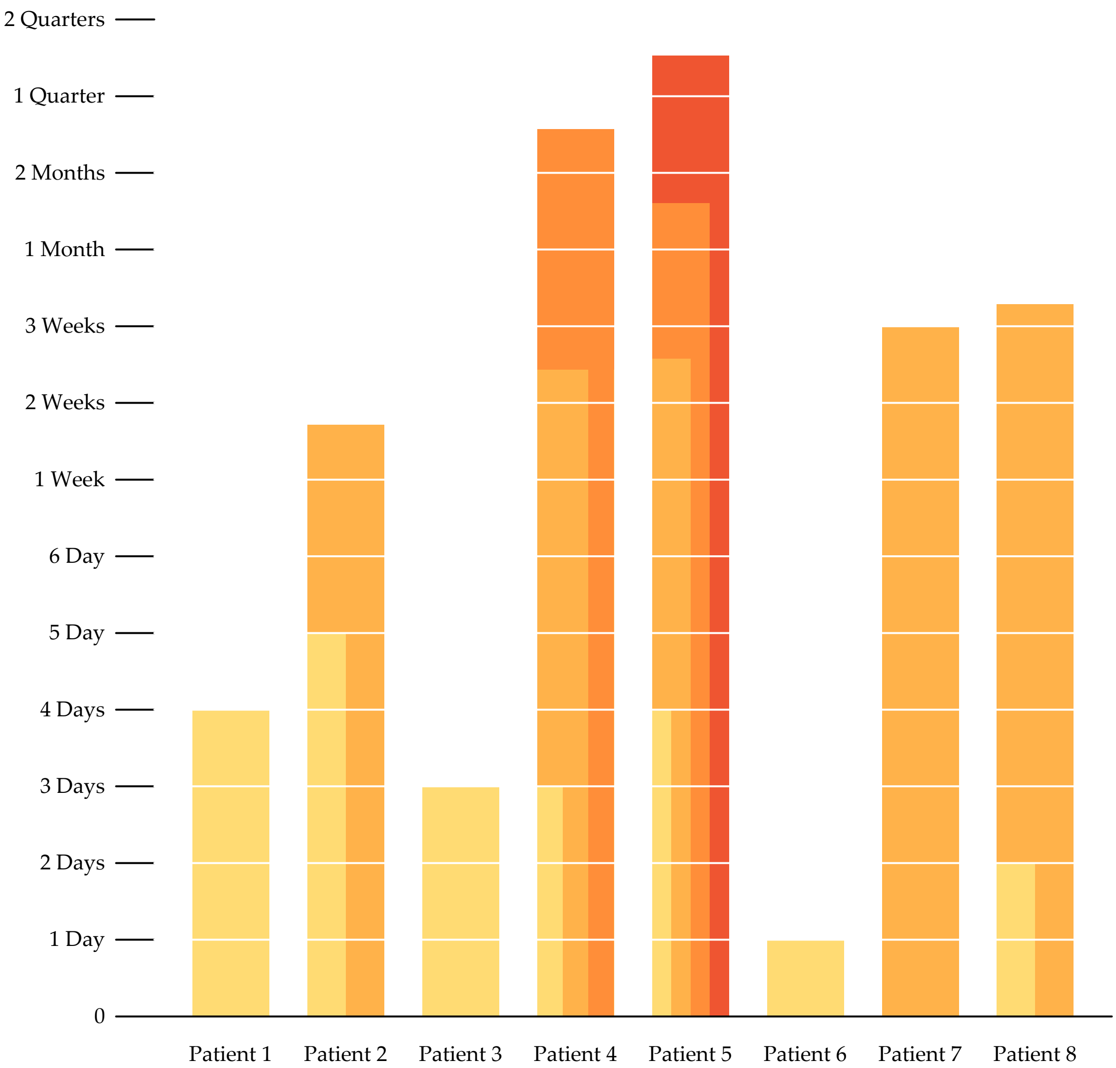


Figure 13.3: Granularity Scales for the Stay of Hospital Patients. The total height of a bar represents the total duration of a stay. The heights of the partial barts show the part of the durations that can be measured in finer granularities.

Table 13.3: The visualization in Figure 13.3 Described according to the Taxonomy from Chapter 8

Tasks: Classify usual values; detect irregularities; compare granules; compare granules on statistical level; read values

Stage/Transformation	**Characteristic**	**Value**
Data Stage	Original Data Types	Discrete Data
	Primitives	Interval-based
Data Transformation	Granularities	Day, week, month, quarter
	Granularity Contexts	Day of week, week of month, month of quarter, quarter of year
	Granularity Labels	Unknown discrete label for all granularities
Analytical Abstraction Stage	Rasterization	Aggregate mean for each day/week/month/quarter combination
Visualization Transformation	–	–
Visualization Abstraction Stage	Type of visualization	Up to 4 times abstract value data
Visual Mapping Transformation	Horizontal base position	Data attribute
	Color	Fixed per part
	Vertical height part 1	Granularity label: day of week
	Width part 1	Total width divided by number of granularities with coarseness equal or greater than day
	Horizontal relative position part 1	0
	Vertical base height part 2	Granularity label: week of month
	Vertical relative height part 2	Granularity label: day of week
	Width part 2	Total width divided by number of granularities with coarseness equal or greater than day
	Horizontal relative position part 2	Width part 2 multiplied with numer of granularities finer than week
	Vertical base height part 3	Granularity label: month of quarter
	Vertical relative level 1 height part 3	Granularity label: week of month
	Vertical relative level 2 height part 3	Granularity label: day of week
	Width part 3	Total width divided by number of granularities with coarseness equal or greater than day
	Horizontal relative position part 3	Width part 3 multiplied with numer of granularities finer than month
	Vertical base height part 4	Granularity label: quarter of year
	Vertical relative level 1 height part 4	Granularity label: month of quarter
	Vertical relative level 2 height part 4	Granularity label: week of month
	Vertical relative level 3 height part 4	Granularity label: day of week
	Width part 4	Total width divided by number of granularities with coarseness equal or greater than day
	Horizontal relative position part 4	Width part 4 multiplied with numer of granularities finer than quarter
View Stage	Combination	Overlay part visualizations

—*I never guess. It is a capital mistake to theorize before one has data. Insensibly one begins to twist facts to suit theories, instead of theories to suit facts.*

Sir Arthur Conan Doyle, The Sign of Four, A Scandal in Bohemia

14 Gaining Granularities from Data instead of Social Conventions

So far, we have worked only with granularities that are either based on well-known natural phenomena, or on social conventions. While these granularities are easy to understand, they might not be able to explain the true structure in the dataset caused by the structure of time.

As a dataset that is in accordance with this theory, we received one containing water-levels of the Traisen, a river in Lower Austria from the Office of the Lower Austrian Government. The data has been measured at the gaging station *** and is provided for the complete years 1997 to 2006. We rasterized the data to hours to make it manageable and because there is not much change at finer granularities.

14.1 GROOVE with Established Granularities

Figure 14.1 shows a GROOVE visualization of the water level dataset using color overlay. The granularities used are hour of day, day of month, month of year, and year total. As the water levels can have a very low, but sometimes very high variance, we have mapped the logarithm of the water level to the overview and detail color parts—hue for overview, lightness for detail. A general trend over years is visible, but it often has irregularities. In several months, there is a light and a dark part, hinting that the parts of the cycle that really belong together are beyond the borders of months. Moreover, the change often is not between lines, but in the midst of days. The gaps caused by the irregular month granularity are a disturbance.

As something positive in this visualization, users can easily integrate a-priori knowledge. In tests, people from Lower Austria have been able to detect floods they remember in the years 2000 and 2006. Still, we consider the customary granularities as not very suitable for this dataset.

14.2 Finding new Granularities

Consequently, we looked for granularities that are better suited to describe the structure in the data. By aligning the granularities to the regularities in the dataset, we expected irregularities to become detectable easily.

As a means of detecting regular structure in the dataset, we applied Discrete Fourier Transformation (DFT). As we wanted to have a manageble number of blocks, we looked for a frequency resulting in a rather low number of cycles. The strongest frequency in the dataset corresponds to a cycle of eleven elements (each lasting 7968 hours). This cycle fits our requirements very well, but as we could read from the DFT, it is shifted to the past of the beginning of the dataset by 2312 hours, a phase transition we had to consider for our following calculations.

To get the next finer granularity, we calculated an average cycle by using aggregation over all eleven cycles. After that, we calculated the DFT for our average cycle. The strongest frequency corresponds to a cycle of 6 granules with a length of 1328 hours each per superordinate granule, with a phase shifted 483 hours in the past.

For the next granularity, we calculated the DFT again, but had to ignore the five strongest frequencies there. We wanted our target visualization to use four granularities, and the stronger frequencies would have made it too unbalanced. The sixth strongest frequency corresponds to a cycle of 24 granules, with a length of $55\frac{1}{3}$ hours each and a phase shifted roughly 16 hours in the future. As we could not divide hours into smaller units without making the process much more complicated, we alternated between cycles with a length of 55 and 56 hours.

14.3 GROOVE with Granularities Tailored to the Dataset

The resulting GROOVE visualization is shown in Figure 14.2. As the granularities found using DFT are not sensible for human interpretation (except for one block having a length of roughly 55 days and each row having 332 days), we have instead added the start time of each block as scale.

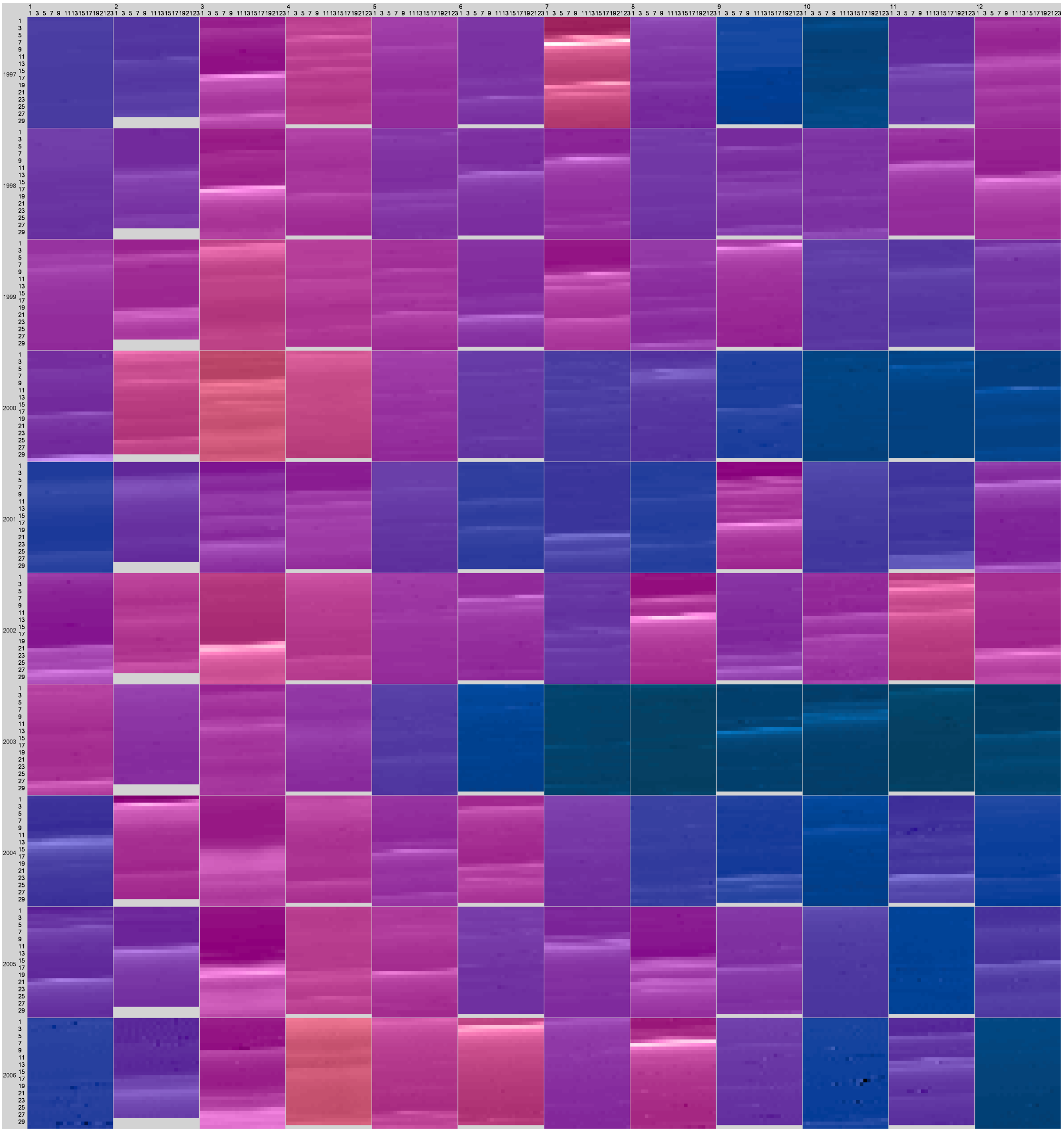


Figure 14.1: The Water Levels of the River Traisen in a GROOVE Visualization with Color Overlay. Established calendar granularities are used: Year, month, day, and hour.

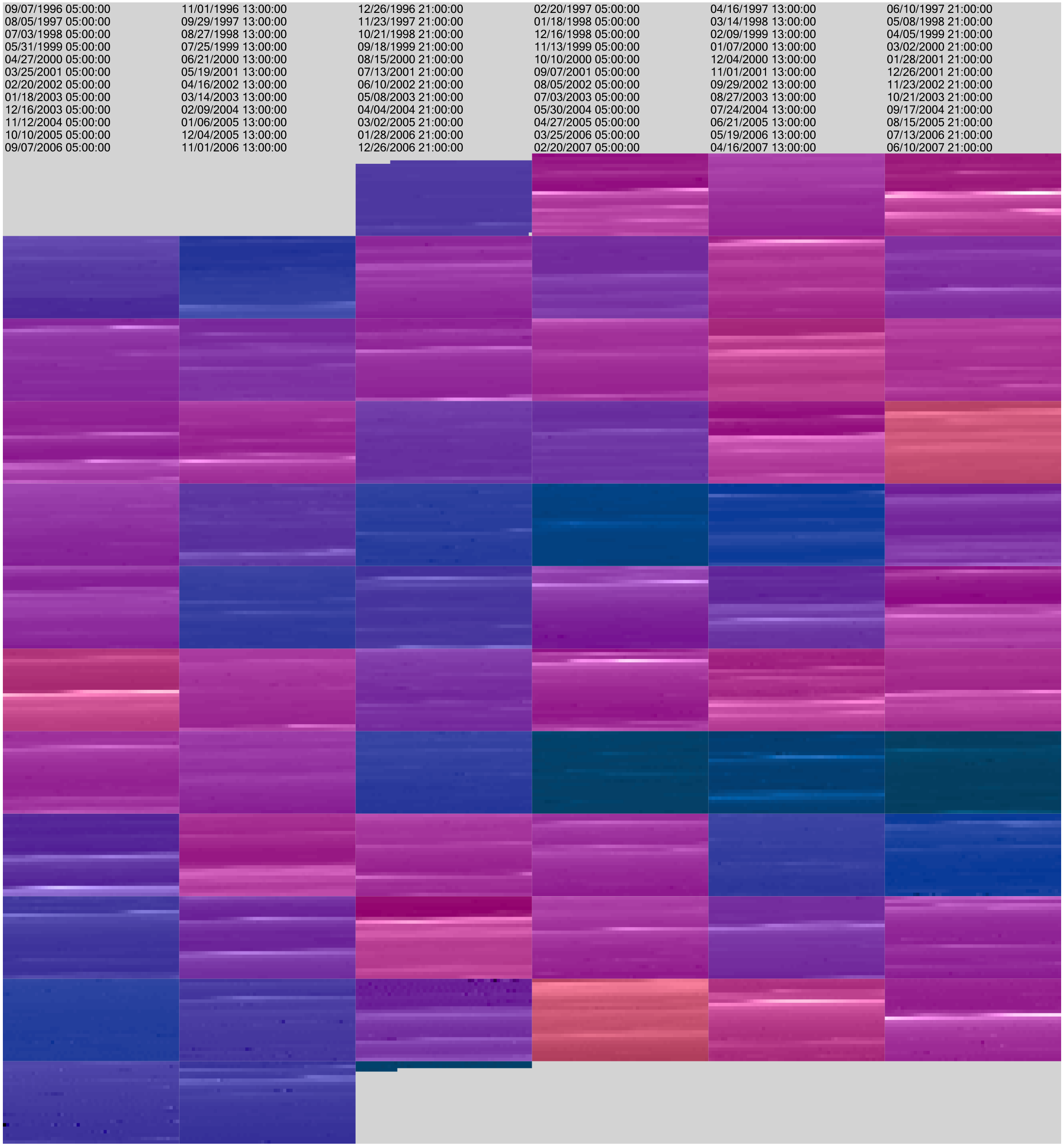


Figure 14.2: The Water Levels of the River Traisen in a GROOVE Visualization with Color Overlay. The granularities have been taken from the dataset by using Discrete Fourier Transformation.

In this visualization, the visual average of blocks, shown by hue, results in a much smaller number of classes than in Figure 14.1. There is a weak cycle over columns visible like the yearly cycle, but the difference is that outliers do not seem to be variances to the cycle, but instead are completely out of line. The might be a result of the fact that few blocks are composed of two different lightness area, most are consistent with some outliers. Consequently, the hues are less of random averages but true classifications. Outliers towards low values are scarce, but outliers towards high values are frequent—smaller and larger floods. In contrast, only two blocks show high water levels for a longer period of time. A catastrophic flood that started in late march 2006 (but still definitely during the month) is one of those. Interestingly, in this order, it starts exactly over the border between the blocks. The end of the dataset shows clear irregularities of very short duration. These irregularities are also present in Figure 14.1, but there they are less visible because that visualization has more clutter. The less cluttered Figure 14.2 is more suited to detect them. It remains unclear whether the irregularities are measurement errors, result of human interference into the water level, or something else.

Overall, Figure 14.2 gives new and important insights into the data. It supports the detection of irregular patterns by arranging the data according to regular patterns.

14.4 Evaluation and Assessment

The visualizations used in this Chapter are GROOVE visualizations, so many results from the User Studies regarding GROOVE (see Chapter 12) are applicable.

Regarding the use of granularities from automated methods, we have yet to perform user studies and refer to our future work (see Chapter 20).

The visualizations in this chapter have been compared and assessed by the external InfoVis expert. He agrees with our assessment that Figure 14.2 is less cluttered and that visual artifacts introduced by the less suitable granularities of the social calendar are removed. He also saw advantages in analyzing the important flood event from 2006. On the downside, he critizied the difficulty of understanding the duration and position in time of visual structures like the block separated by hue. He adviced to include axes even when the values on them are not granularities which can be understood by humans easily.

—We have long passed the Victorian era, when asterisks were followed after a certain interval by a baby.

W. Somerset Maugham

15 Visualizations Showing Patterns

Beside the visualization part, the project **DisCō** also sparked research in the area of Pattern Finding. The resulting framework enables users to interactively search for patterns in time-oriented datasets. It makes heavy use of interactive visualization as part of the user interface. In this chapter, we will shortly describe the pattern finding algorithm, and focus on its visual support.

15.1 Digression: Multi-time Interval Pattern Finding

As Bertone [2009] outlined, most of the analytical methods (i.e., sequence and interval mining methods) which try to find interesting patterns from time-oriented data, give as result a sequence of events, lacking any knowledge either about the intervals in between them or about after how much time a particular pattern will reoccur. Moreover, they usually do not involve the user in the analysis, but rather provide a sort of black box to be applied, neglecting the possibility to add any user knowledge.

An example which clarifies the problem is shown in Figure 15.1. A customer decides to buy a cellphone (event A). The next day, he buys a ringtone for that cellphone (event B). One month after that, he buys another ringtone (event C). For the following two months, he keeps buying a ringtone after one month each (events D, E). One month later, he buys a USB cable for the cellphone instead of a ringtone (event F). After that, he does not buy ringtones anymore. If we take into account only the sequence or the order of these events (ABCDEF), we cannot know after how much time the next item will be purchased.

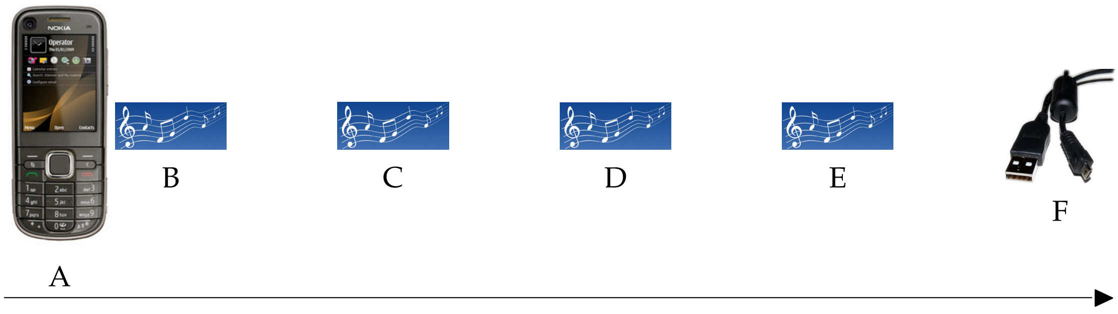


Figure 15.1: Buying a Cellphone, Ringtones, and an USB Cable as a Pattern of Events

Moreover, we cannot even know after how much time a similar sequence will occur. On the contrary, if also the time intervals are considered, we can not only profile the users according to their interests, habits and requirements, but we can also improve the selling strategies according to the timing of their shopping habits. As a matter of fact, the webshop can vary its offers and catalogues according to the users. For instance, it is possible to send e-mails or letters describing discounts on ringtones four months after the first purchase, possibly stalling the finalizing buy of the USB cable.

Therefore, we want our sequence ABCDEF to become something like A *1 day* B *1 month* C *1 month* D *1 month* E *1 month* F *1 month*. To reach this goal, Bertone [2009] proposed the use of so called multi-time interval patterns as well as a novel approach to preserve temporal information in between.

To solve these tasks, Bertone [2009] describes a a novel approach of using an extension of the i-Apriori algorithm named MuTIny. To perform the algorithm, users have to map each data element of a dataset to an event, calling them e_0, e_1, and so on. Multiple data elements can be instances of the same event depending on values of their data attributes. In our prototypical implementation of Bertone's algorithm, we have implemented the following variants for event definition:

- Ignoring a data attribute;
- Make each possible value of a data attribute a separate event;
- Use absolute ranges;
- Use quantiles.

Bertone calls this process of defining and finding events step 0.

In step 1, the most important events are chosen. Bertone proposes to choose the events with a relative frequency higher than a certain threshold, called the support. Furthermore, pre-defined intervals are necessary. For each event that has been chosen, all following events that are inside one of the intervals form a pattern. Steps 2 and beyond are repetitions of step 1, but the new patterns are based on patterns instead of events. The algorithm terminates simply because of the fact that the dataset ends at some point, but generally, it can be used to gain important insight at earlier steps already.

To gain these insights, we developed a Visual Analytics framework that uses interactive visualizations. These visualizations also have the ability for serve as an interface for users to actively influence the analysis process.

15.2 General Idea of the Visual Analytics Framework

In our Visual Analytics Framework that discloses MuTIny to users, we break up the linearity of the algorithm. Information and configuration options regarding any of the steps can be viewed and modified by users in parallel. Changes to steps at earlier stages are automatically handed over to views of later stages.

Thereby, we go from a linear view to a view of abstraction (see Figure 15.2). Users can access data at pattern, event, or value level. At each level, we have to distinguish between a class at that level or an object, an actual occurence.

To make such an approach possible, we have to start the algorithm with sensible defaults and partly configurations created automatically from the data. All these configuration options can be modified by users, resulting in a change of dependent values.

15.3 Support (Threshold) Configuration

The support can be manually adjusted in our framewok, separately for the calculation of each pattern length. As an alternative, users can have the support to be calculated automatically in a way that a configurable number of patterns can be used as a basis for the next pattern length.

Furthermore, it is possible for users to interactively choose any set of patterns that is used as a basis for the next pattern length, ignoring any support.

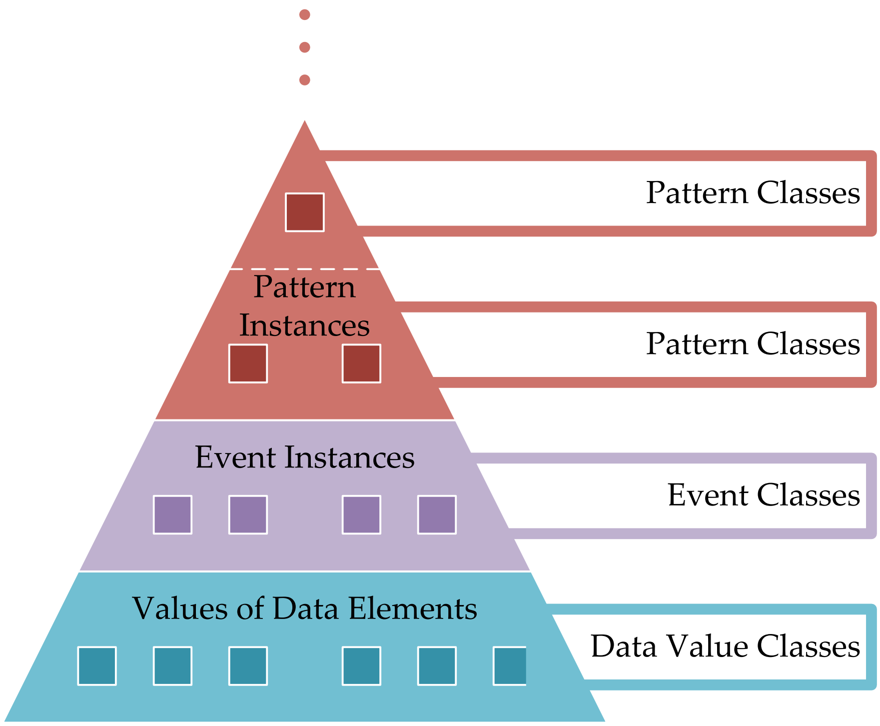


Figure 15.2: Different Abstraction Levels of our Visual Analytics Framework. The blue level is the data value level where we can see inside the events. The purple level is the one where the data values have been combined to events. At the red level, the events are combined to patterns. The pattern level can be seen as many sublevels with increasing pattern lengths. At each level, we can look at classes, for example event classes, that show which kind of events exist in the dataset. We can also look at instances (small rectangles) that exist at the various points in time.

15.4 Event Configuration

To help configure the events, we present users a simple statistical analysis of data attributes. For each data attribute, we show mean, median, and configurable number of quantiles (see Figure 15.3.a).

The attributes can be chosen for event definition (see Figure 15.3.b). For each chosen attribute, users can configure how they want it to be applied. When the chosen attribute is a time stamp, several granularity labels based on that timestamp are available to be chosen as exact values or ranges.

The last part of the event configuration shows how the chosen attributes and definitions translate into events. This is done using a visualization similar to a Parallel Coordinates View (see Figure 15.3.c). The main advancement over standard Parallel Coordinates is the ability to show value ranges using triangles instead of lines, similar to techniques presented by Bendix et al. [2005]. However, that limits the number of events that can be shown compared to the possibilities of standard Parallel Coordinates.

The event configuration interface also has a number of user interactions. Events can be relabelled here, changing the labels from $e_0, e_1, \ldots$ (which have been rather unpopular with users) to something that makes more sense to humans.

Events can be filtered by selecting vertical areas on the parallel axes. The filtering can be performed for this view only, but there's also the possibility to apply the choice of events for use in further steps of the algorithm.

15.5 Showing Pattern Classes

We have developed several visualization variants to show pattern classes. Figure 15.4 shows the 2C visualization introduced by Bertone [2007]. The events that start patterns are shown in the middle. The size of the circles representing the events shows their frequency. Following events are placed on concentric circles (hence 2C) around the start events. The intervals in between are visualized as lines between the events. For follow-up events, the total frequency of the pattern so far is represented by line thickness.

We have advanced development of the 2C visualization based on common design principles (see Figure 15.5). The circles have been replaced by parallel lines. Instead of mapping the frequency of events and patterns to width or size, it is mapped to vertical position. Therefore, Figure 15.5 can show twice the number of patterns in Figure 15.4 and is better readable.

a)	Passing Cars per Hour	Speed (mph)	Accidents per Hour
Mean	1848	63	3.5
Min	0	10	0
Lower Quartile	1000	50	1
Median	1500	60	3
Upper Quartile	2000	80	7
Max	2250	88	9

b)	Passing Cars per Hour	Speed (mph)	Accidents per Hour
	Range: 1500-	ignore	ignore
	Range: 1000-1500	ignore	ignore
	Range: 500-1000	ignore	ignore
	Range: 250-500	ignore	ignore
	Range: -250	ignore	ignore

c) Event Label	Passing Cars per Hour	Speed (mph)	Accidents per Hour

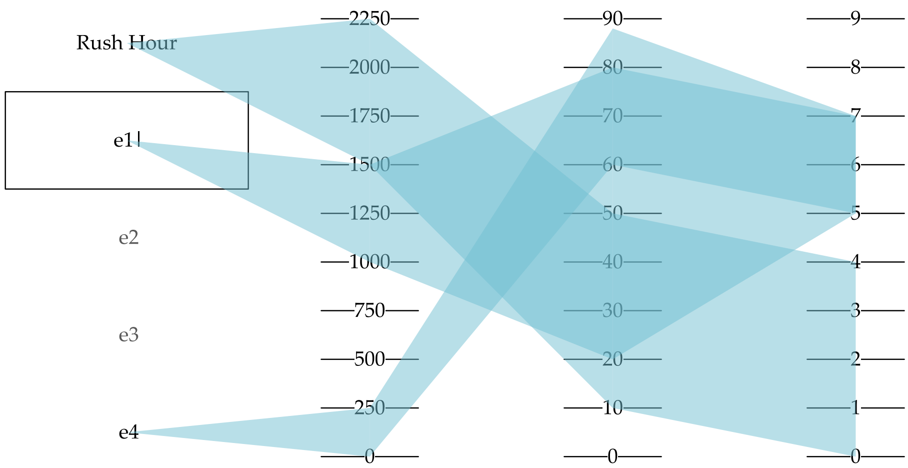


Figure 15.3: Event Configuration Abstraction. The dataset shows traffic control data. a) gives statistical background about the data attributes. b) is where the data values can be parameterized to events. Only cars per hour is configured (to range). The allocation to events is done by the algorithm. c) shows how this allocation results in events. The events can be relabelled and also deactivated (shown in gray). The ranges at speed and accidents are resulting from the event allocation—the ranges shown in c) are not given by users but show the actual values of the events.

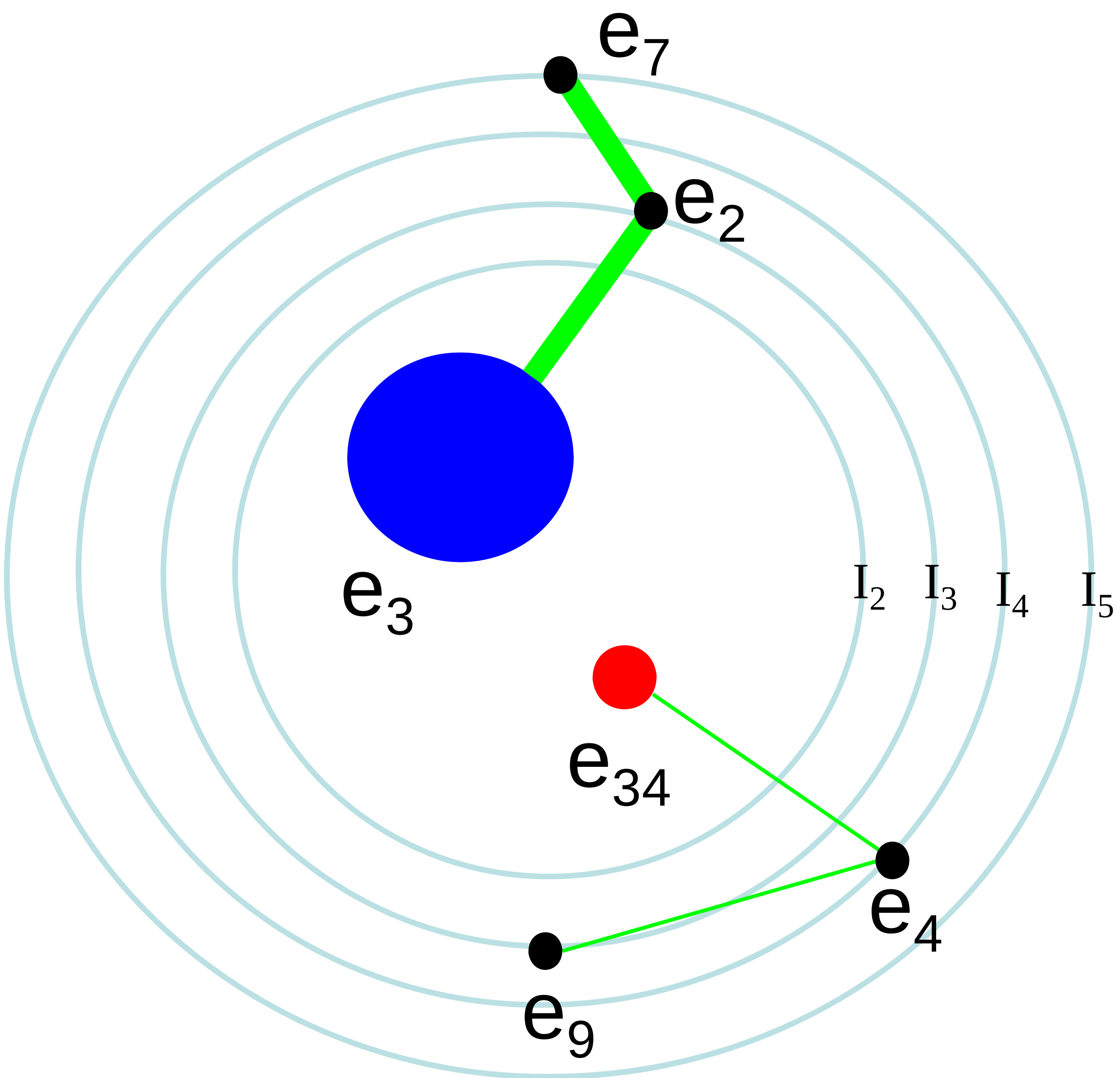


Figure 15.4: The 2C Visualization (from [Bertone, 2007]). Only patterns of length 2 are shown, such as $e_3I_3e_2I_5e_7$ and $e_34I_4e_4$ I_3e_9. The thickness of the segment connecting the events (e.g., e_34 to e_4, and e_4 to e_9) is proportional to the number of occurrences of the pattern (e.g., $e_34I_4e_4I_3e_9$).

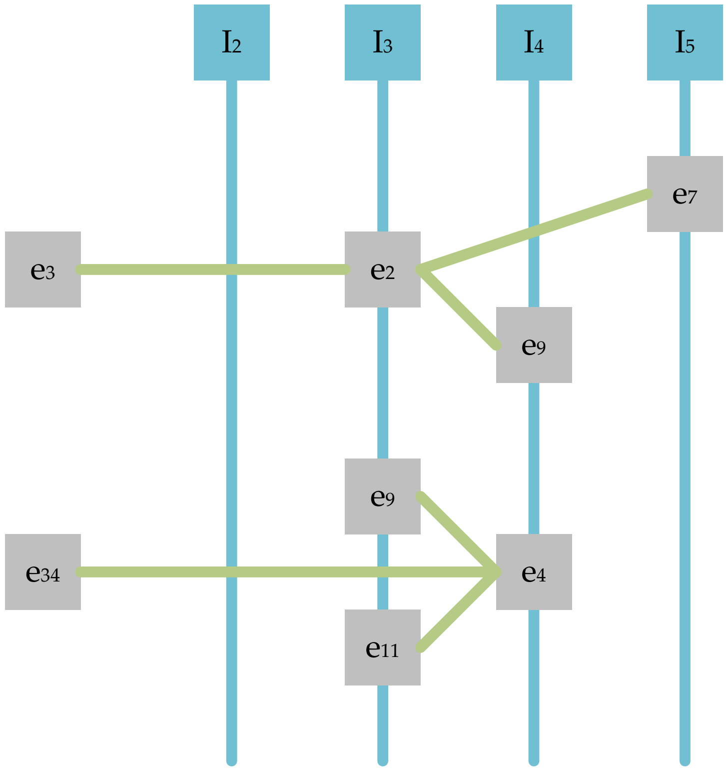


Figure 15.5: The Parallel Lines Visualization as Advanced Version of the 2C Visualization (see Figure 15.4). The patterns from Figure 15.4 are shown, as well as the patterns $e_3I_3e_2I_5e_9$ and $e_34I_4e_4$ I_3e_11. The vertical position of the starting events to the left shows how frequent they are. The frequency of a pattern or pattern start is encoded by the relative vertical position of the last event compared to the event before.

15.6 Showing Pattern Instances

When it is necessary to show each event and pattern instance at the exact point in time when they occur, it is necessary to place them on a time line in some way. Figure 15.6.a shows the basic idea. As such a visualization has a huge width and a small height, it is difficult to use. One possibility is a spiral layout. We have tried that way but not pursued it further as it was too difficult for early test users to compare time-oriented distances at different positions from the center.

As alternative, we have to cut the time line and stack several parts of it below each other (see Figure 15.6.b). To do this in the best way possible, it is necessary to decide where to cut. We can include the structure of time in our decision process by defining each row of the stack as one granule of a granularity. The decision which granularity to use can be based on user domain knowledge. Alternatively, we can define an artificial granularity (like in Chapter 14). To find this granularity, we use an algorithm that iterates over a number of spans that are possible based on the available space and data length. The algorithm chooses the granule length that results in the least number of patterns being cut between lines.

Figure 15.6.c shows that a number of rows in the stack (again, one granule of a granularity) can folded together in one row, saving space. Therefore, even more time can be shown inside one visualization. We call the approach shown in Figure 15.6 the Stack and Fold approach.

15.7 Showing Pattern Instance Repetitions

To show the repetition of pattern instances, Arc Diagrams [Wattenberg, 2002] can be helpful. On a timeline, all occurences of a certain pattern are connected to the next occurence by an arc. The height of the arc is dependend on its width, in order to give the arc the shape of a half circle. Therefore, the overlap is reduced.

In our variant (see Figure 15.7), we show different patterns in different colors. Therefore, the number of pattern classes that can be shown is limited by the number of discernable colors. However, the number patterns that can be shown is limited anyway because our tests showed that there would be too much overlap without preliminary filtering for the most important patterns. The blue arc in Figure 15.7 shows another characteristic of our approach: The width of arcs can vary between the two spans where it touches the timeline. The reason for this fact is located in the algorithm. An interval defined for the algorithm usually does not have a fixed span, but a range of spans, like “1 day to 2 days”. Therefore, instances of the same pattern can have different lengths.

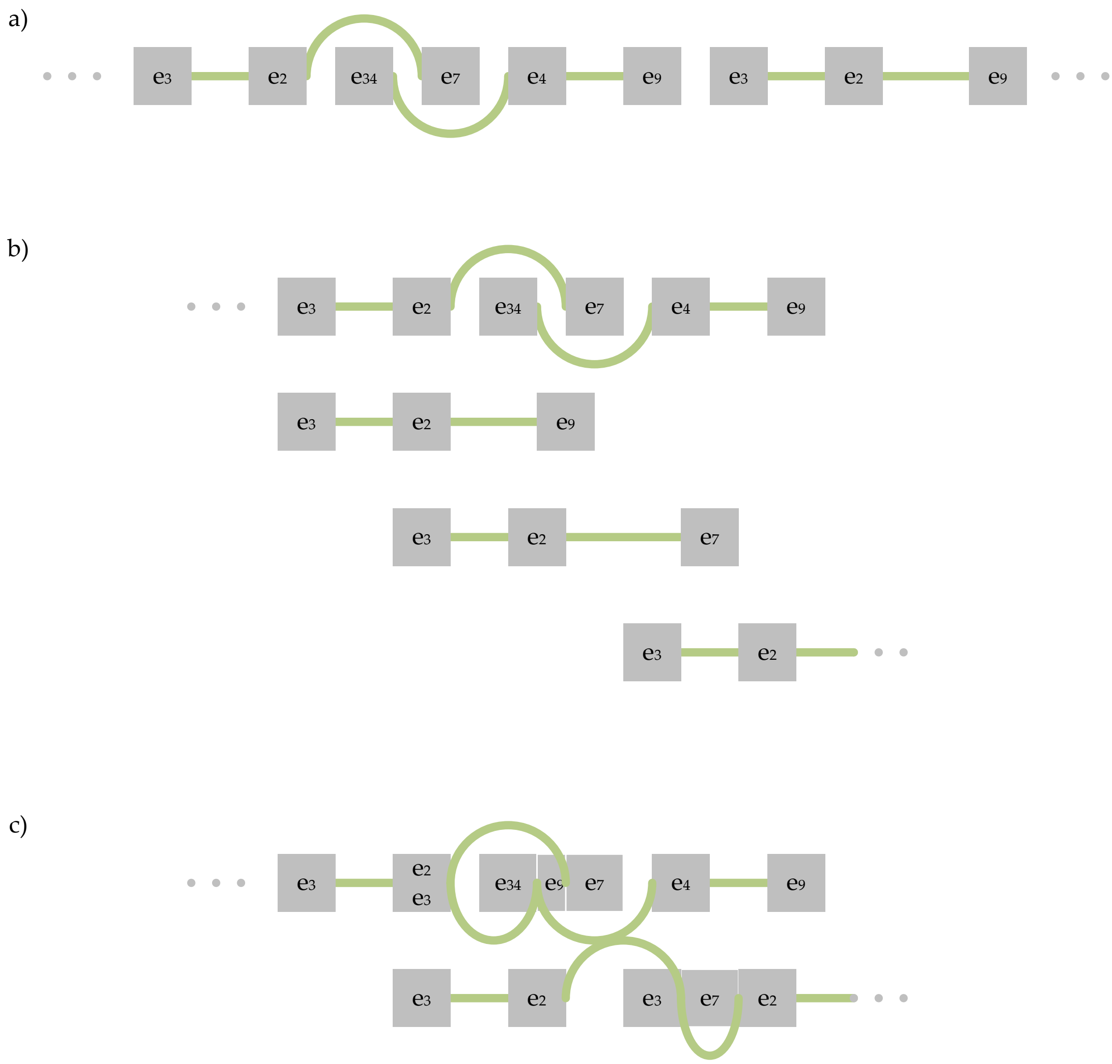


Figure 15.6: The Stack and Fold Approach. a) On one time line, the number of patterns that can be shown is very limited. b) By cutting the time line and stacking the parts, space can be used in a better way. c) Folding several lines into each other gives the possibility to show even more data.

Figure 15.7: Arc Diagrams Can Show the Repetition of Pattern Instances (Adapted from Wattenberg [2002]

15.8 Showing the Build-up of Pattern Classes

To show the build-up of pattern classes and also how frequent pattern instances of these classes as well as instances of partial classes occur, we consider a treemap as effective. Treemaps divide space based on data that is organized according to a tree structure. Each branch is given space either according to the number of nodes it contains, or according to the sum of a data value contained in each node.

Figure 15.8 shows an example based on our case of pattern data. The treemap shows which pattern classes exist and how many pattern instances exist for each pattern class. The size of each rectangle represents the number of instances. Moreover, it shows the total number of instances for different pattern classes starting with the same sequence of event and interval, as several smaller rectangles combine to rectangles at a larger level with partial pattern classes.

15.9 Showing Pattern Classes and up to Two Data Values

Showing events instead of patterns is a big abstraction step. An important question when looking at patterns is how the development of a pattern correspondents to developments in the values. To show at least two values that form a pattern, we provide a scatterplot view.

Figure 15.9.a shows two patterns of length 1 in a scatterplot. The black dots symbolize events, the blue arrows correspond to intervals. The position of each dot is dermined by two of the data values that correspond to the event. Therefore, it is not possible to encode the length of intervals, the arrows only show the connection between events.

Figure 15.9.b expands the definition of events from exact values to value ranges. Each

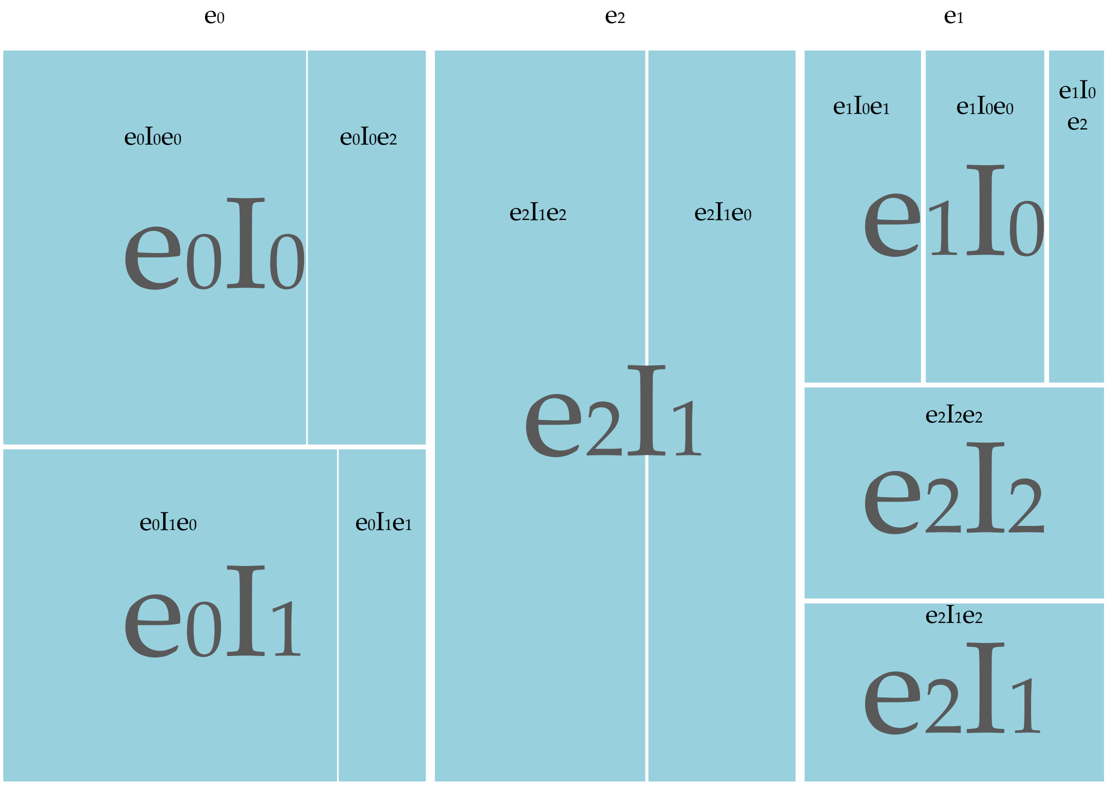


Figure 15.8: The Pattern Treemap. All patterns start with e_0, e_1, or e_2. As in every treemap, the areas are representing a number, in our case it is the number of instances for each event or pattern class. The patterns have length 1, and each pattern is distinguished first by the starting event, then by the interval, then by the final event.

gray box corresponds to one event. The dimensions of the box are given by the ranges of values that result in one data element being attributed to that event. The arrows, again, symbolize the intervals.

15.10 Interactions Usable with Most of the Visualizations

We have developed sub-visualizations and interactions that can be combined with the visualizations presented above. These interactions can affect not only the visualizations, but also the algorithm responsible for finding the patterns.

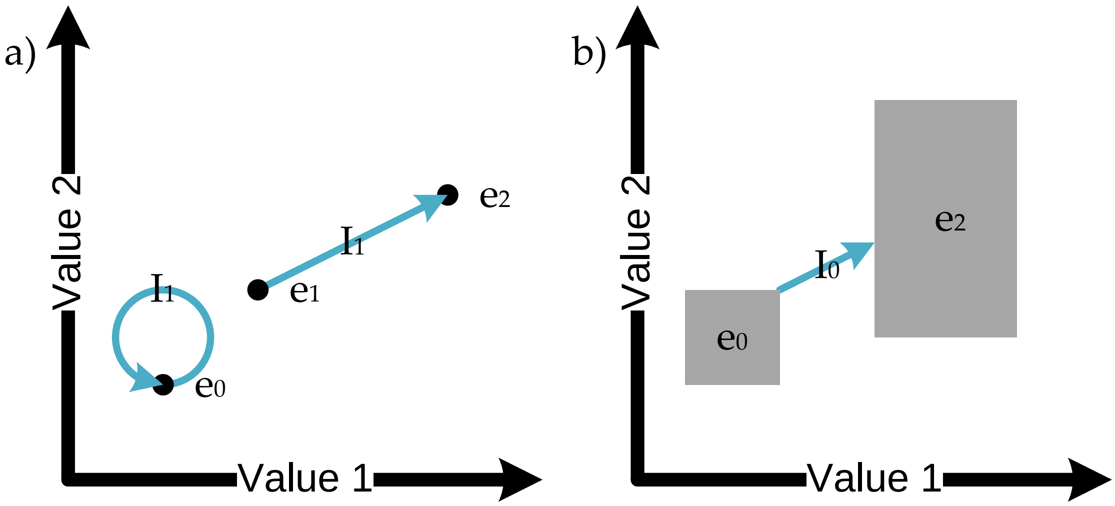


Figure 15.9: Showing Patterns with Values in Scatterplots. a) The patterns $e_0I_1e_0$ and $e_1I_1e_2$ are shown, with all events defined by fixed data values. b) The pattern $e_0I_0e_2$ is shown, both events are defined by a certain range in each data value.

15.10.1 Showing Pattern Distribution over Granules

To show the structure of time aspects of the results and interact with them, we have developed an interactive sub-visualization that is shown in combination with the other visualizations. It is developed with cyclic granularities in mind, therefore it is based on a circle. Figure 15.10.a shows an example based on day of week as granularity. Each granule is a segment; in the base visualization the segments are not filled.

When the visualization is connected to a dataset, for example a list of events, the segments corresponding to granules for which data elements exist are filled. Figure 15.10.b shows an example where events happen on weekdays, but not on weekends.

Users can interact with the sub-visualization and for example hide granules. The granule is not hidden in the sub-visualization itself. As Figure 15.10.c shows, it is slightly faded out. In the visualization that the sub-visualization is connected too, all data elements that are at hidden granules, are truely hidden.

Moreover, users can fully disable granules. Figure 15.10.d shows the combination of granules that do not have data elements, granules that are fully active, granules that are hidden, and granules that are fully disabled (Tuesday and Wednesday). Data Elements located at disabled granules are ignore when the algorithm calculates patterns of the next length. When users change the state of a granule, all depending datasets and visualizations are updated.

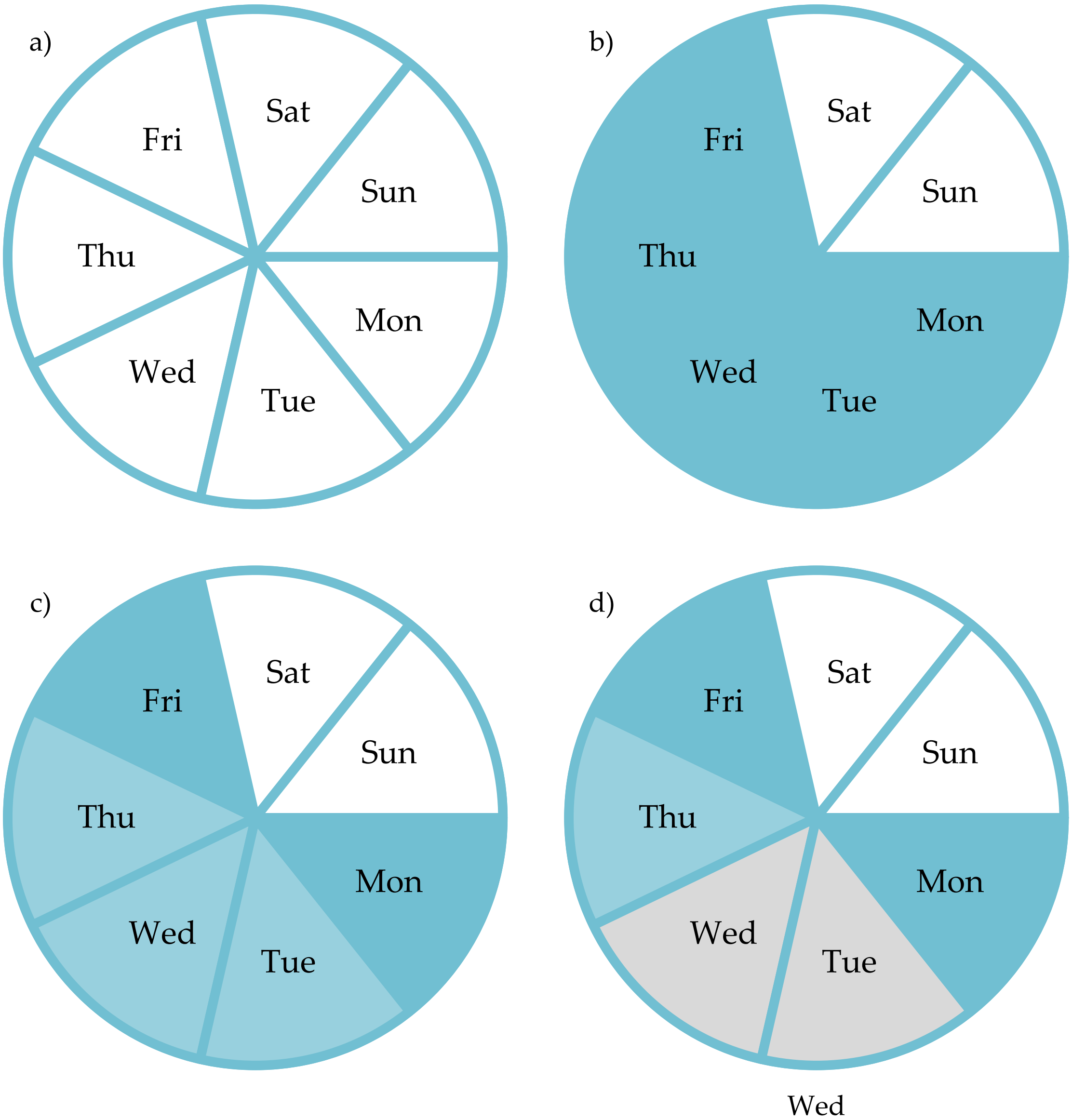


Figure 15.10: The Pattern Granule Visualization. It can also be used to choose granules. a) The basic visualization b) An example with data on weekdays but not on weekends c) User has hidden Tuesday to Thursday in the connected visualization. d) User has disabled Tuesday and Wednesday for the next algorithm step.

15.10.2 Selecting Patterns

In all visualizations that show patterns, it is possible to highlight the patterns that are to be used for the next pattern length according to the configured support. Users can also interactively add or remove the highlighting from patterns. As a result, the support is shifted to a value that results in the corresponding patterns being part of the next algorithm step—if possible. In case the patterns chosen by a user do not allow the support approach to be used, the support is replaced by a list of selected patterns that are to be used by the algorithm.

The reason to include this interaction is the fact that sometimes, only by specifically chosing pattern, users can get the effect they want. The primary reason for such situations are patterns that are very frequent but also very unimportant. One example we had in our tests was “average value on one day followed by average value on the next day”. It is important to filter out such patterns—at the current stage of development, such a decision can only be done by users, though.

15.11 Implementation

We have developed a Visual Analytics Framework in .NET based on the concepts described here. The framework contains an algorithmic implementation of the MuTIny approach, as well as interactive implementations of the 2C visualization, the Stack-and-fold visualization, and the Pattern Granule visualization. Figure 15.11 shows the Framework with the 2C visualization, a linked Pattern Granule visualization, and a list of all patterns. The Pattern Granule visualization currently only affects the 2C visualization, not the algorithm. Therefore, the algorithm currently has to be controlled using a textual interface. The other visualizations presented here are planned to be implemented in future versions, in order to bring the Visual Analytics framework from the textual interface to the interactive visual interface as described in this chapter.

15.12 Evaluation and Assessment

Evaluation of the MuTIny approach has been done [Bertone, 2009] but is not described here as we focus on the visualization part and have only described MuTIny in order to provide understanding of our motivation for developing the visualizations.

Evaluation of the visualizations in a user study could not be performed yet as most of them still need to be implemented in a prototype. Therefore, we refer to our Future Work (see Chapter 20) for information about planned user studies.

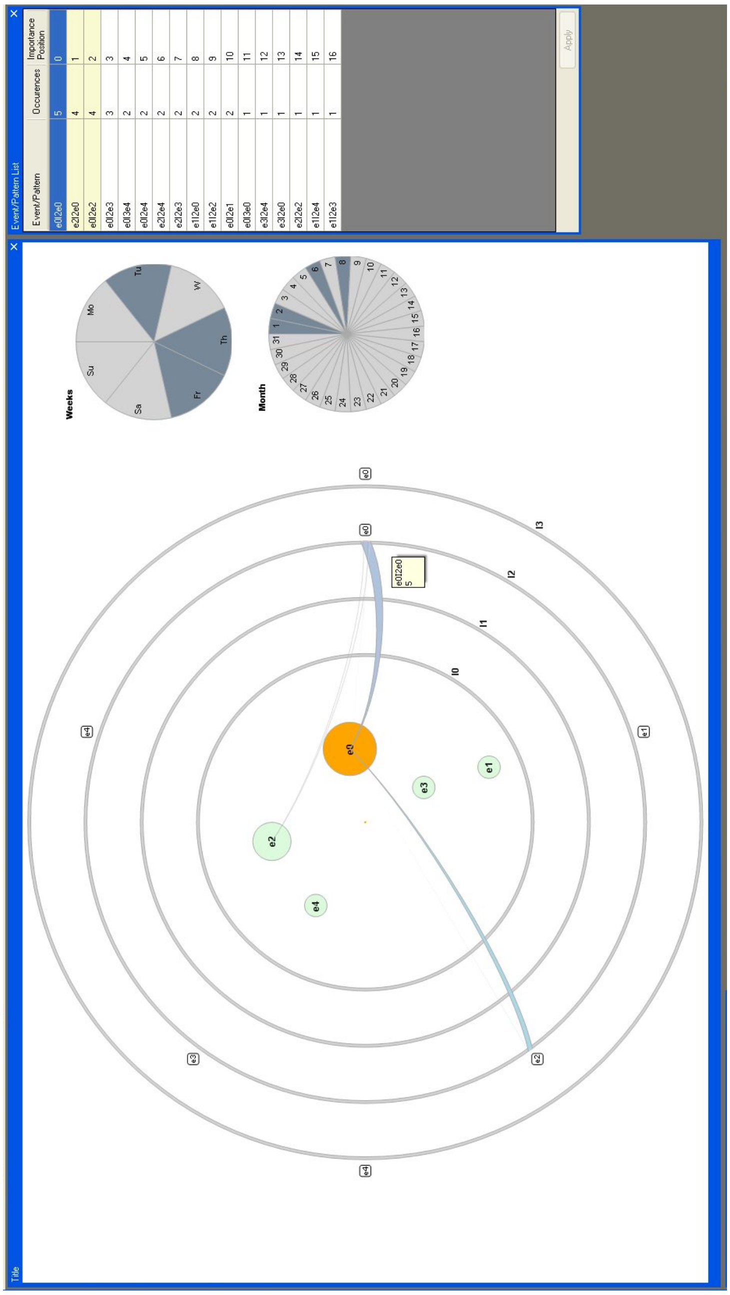


Figure 15.11: The MutTIny Framework. The main part shows the 2C visualization with a Pattern Granule visualization connected to it. To the right, there is a list of all patterns.

We got an assessment of the mock-ups we provided from the external InfoVis expert. He saw advantages for 2C as well as Parallel Lines visualization: 2C shows absolute frequencies of patterns, while Parallel Lines require less space and apply the more powerful visual variable of position.

The expert agreed to our assessment of the Stack-and-Fold visualization, but also interjected some criticism: He is missing a relation between spatial position and temporal consequentiality for events belonging to different patterns. furthermore, he considers the visual clutter introduced by the fold operation as rather high.

Regarding the Pattern Treemap, the external expert commended it for clearly showing the breakdown of patterns. However, he criticized the area to be less readable then position as visual variable. Furthermore, he claimed that only the frequency of the first event in a pattern is shown as an absolute value. This is not fully true, but overall, the assessment shows that the allocation from area to frequency seems to be too complicated to understand in this visualization.

For the Scatterplots showing patterns, the InfoVis expert suggested not to actually draw arrows for the intervals, but use a different visual variable, for example different colors for the order of events in patterns. Therefore longer and more patterns could be shown without the visualization becoming too cluttered.

—It was, of course, a lie what you read about my religious convictions, a lie which is being systematically repeated. I do not believe in a personal God and I have never denied this but have expressed it clearly. If something is in me which can be called religious then it is the unbounded admiration for the structure of the world so far as our science can reveal it.

Albert Einstein

16 Integrating Showing Data and Showing Patterns

16.1 Showing Data Based on Patterns

In Chapter 14, we have demonstrated that automated methods can be used to define artifical granularities for the GROOVE visualization that are specifically tailored to a data set. In Chapter 15, we have gained artifical granularities by cutting as few patterns as possible in the process. We now combine those two approaches, in order to get a GROOVE visualization that has the patterns found automatically in its organizational structure but does not actively show them.

Figure 16.1 shows, again, the data from police assignments. For one month, we show the number of assignments being given at different five-minute-intervals. The dataset contains 8498 data elements. We want to change the granularities used in a way that does cut as few MuTIny patterns of length 2 we have found in the dataset as possible. To retain some balance in the dataset, we are willing to accept granule lengths from five to fifteen data elements.

The patterns found by MuTIny are cut least when using granularities of lengths 5, 5*5, and 5*5*15. This might lead to the conclusion that more extreme distributions might be even better, but the visualization would become too unbalanced. Actually, the next length our algorithm based on MuTIny found was 12, but we could only apply 3 different granularities in a static visualization. For interactive paging over different

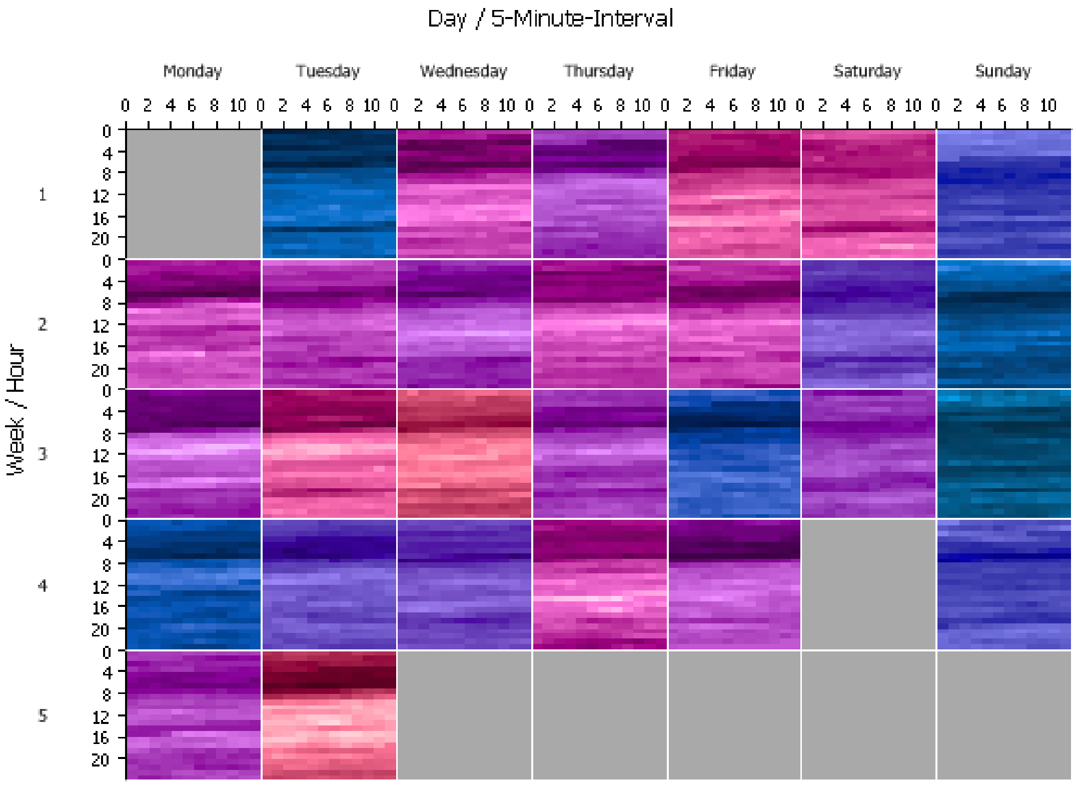


Figure 16.1: Data of Police Assignments Over One Month. Each pixel represents 5 minutes. The GROOVE visualization shows higher values in dark, lower values in light. Days with high average are blueish, days with low average reddish.

visualizations, showing 12 (instead of 23) rows per visualization would have been a good choice. A GROOVE visualization based on the lengths we found is shown in Figure 16.2. It can be seen that several kinds of block appear repeatedly in the GROOVE visualization when arranged according to our patterns. Other blocks appear to be outliers.

16.2 The Possibilities to Gain Granularities for a Visualization

After showing another possiblitiy to find granularities, we can now summarize the methods available to gain granularities for a visualization:

Default Calendar Granularities This is a sensible default that can be used to show a visualization from the start, without users having to make decisions.

Employing Users' Domain Knowledge The domain knowledge of users can be included through user interaction. Granularities given by users can be chosen from standard calendars, but also entered individually.

Using Automated Methods We have shown how to gain granularities from statistical methods in Chapter 14 and how to gain granularities using Data Mining in this chapter. Many other methods are conceivable.

16.3 Combining all Visual Variables

We can adapt a general view on a visualization that contains a large number of visual variables that can be partitioned in two kinds:

1. Position
    - Page
    - Vertical Position
    - Horizontal Position
    - Vertical Position (Level 2)
    - Horizontal Position (Level 2)
    - …
2. What to draw at position
    - Hue

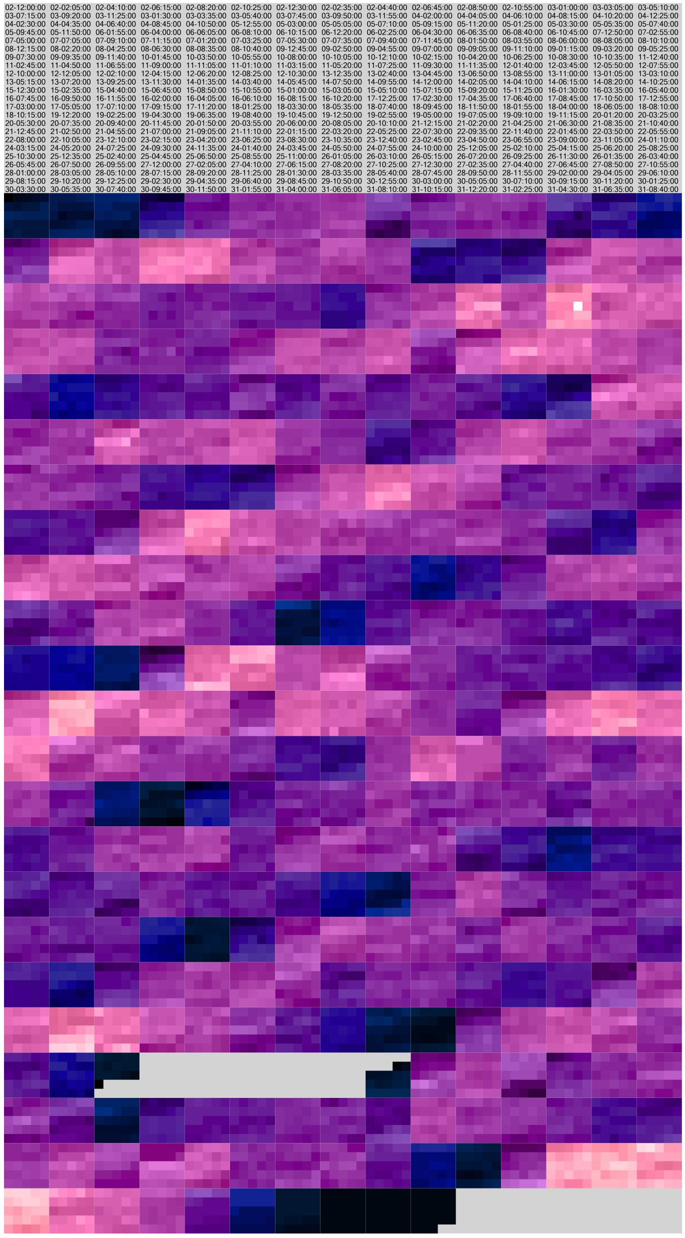


Figure 16.2: The Police Assignments from Figure 16.1 in a Different Arrangement. Each block now contains 125 minutes, each pixel still 5 minutes. The starting minutes of each block are shown above the visualization. The GROOVE visualization shows higher values in dark, lower values in light. Days with high average are blueish, days with low average reddish.

- Chroma
- Lightness
- Shape/Symbol (pixel in GROOVE, but others are possible)
- ...

Those visual variables can be mapped from granule labels or any other data values contained in the dataset. The granule labels can also stem from automated methods. Moreover, the data itself can stem from automated methods—at this point, we arrive at the Stack and Fold approach from Chapter 15 (see Figure 15).

Summarizing this, we have a visualization technique with nearly unlimited possiblities for Visual Mapping. By providing the possibility to make these Visual Mappings interactively switchable, the possibilities can be further enhanced.

16.4 Evaluation and Assessment

The visualizations used in this Chapter are GROOVE visualizations, so many results from the User Studies regarding GROOVE (see Chapter 12) are applicable.

Regarding the use of granularities from automated methods, we have yet to perform user studies and refer to our future work (see Chapter 20).

The assessment of the visualizations from this Chapter, done by the external InfoVis expert, is very similar to the one of the visualization from Chapter 14. However, he strongly critizes that the pattern itself are not shown. Therefore, we consider it necessary to develop a visualization variant that shows the patterns as well as the data in some parallel way.

—In science it often happens that scientists say, "You know that's a really good argument; my position is mistaken," and then they would actually change their minds and you never hear that old view from them again. They really do it. It doesn't happen as often as it should, because scientists are human and change is sometimes painful. But it happens every day. I cannot recall the last time someting like that happened in politics or religion.

Carl Sagan

17 Showing Different Points of View

When considering different points of view, people often equalize it with views from multiple sources. This aspect has been analyzed thoroughly in the past. We are dealing with another aspect of different points of view: Different points of view from the same source over a course of time. This aspect has seen less research, especially regarding visualization. As both aspects are relevant depending on data and task, we deepen resarch in the second direction.

17.1 Different Values over Time

Figure 17.1 shows how data from forecasted events can be classified according to two factors:

1. The point in time when an event happens (ignoring whether it happened or will happen).
2. The point in time the value of an event has been forecasted (ignoring when it actually happend or will happen).

Depending on the task at hand, it is possible to perform most of the InfoVis methods available for time-oriented data on both of these time axes. In the following sections of this Chapter, we will present an examplary use case and show some visualizations we have developed on this basis.

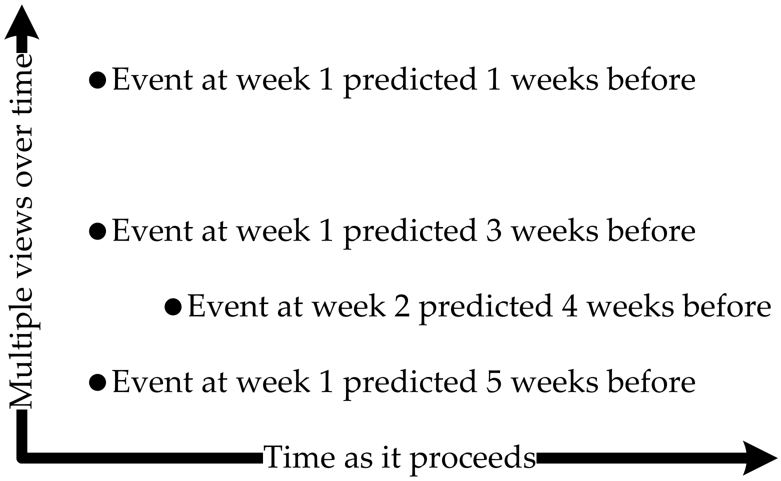


Figure 17.1: Classifying Events. It can be done according to their point in time or to the point in time they have been forecasted.

17.2 Example: Motor-Coach Capacity Utilization

For a cooperation partner, we had the task of evaluating the forecast calculated by a black-box-system for forecasting the utilization of coaches. The dataset we got contains data elements with two timestamps: One timestamp defines the departure date and time of a coach at a certain coach terminal, the other one defines the point in time a forecast had been made. Two further data attributes contain the actual number of passengers who boarded the coach before its departure, the other one contained the number of boarding passengers forecasted at the time given for that.

We have made some prior calculations to the dataset. For each coach that departs, we have calculated the error of the forecast for all days before the coach departs. Therefore, we have a dataset with two time dimensions:

1. The point in time when a coach departs.

2. The day before the departure a forecast have been made.

Only one data value remains: The deviation between forecast and actual number of boarding passengers. In Figure 17.2, we have mapped the second time dimension to the horizontal axis. For all coaches departing, no matter on which date, box plots are shown.

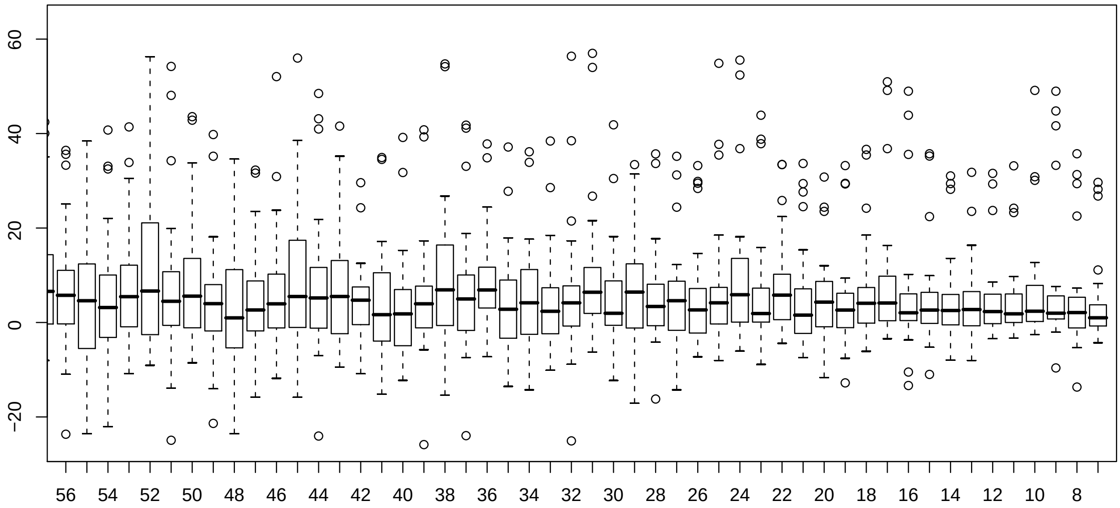


Figure 17.2: Box Plots over Forecast. The number of boarding passengers for all coaches departing on each day is compared to the forecast 7 to 56 days before. We show box plots of the absolute error value (ignoring coach size).

17.3 Granularities in the Point of View Time Axis

As we did for the time axis that determines when an event happens, we can apply the calendar aspect to the time axis that determines when a forecast happens. A time interval of two weeks is important for the company performing the coach runs because they have to determine the work shifts in two-week-intervals. However, the actual calculation of the forecast usually happens once a week. This forecast is calculated in full for all upcoming coach departures of the next two months. Of course, for each day, the actual number of days this forecast is before the departure is different. Therefore, a recalculation has to be done. For each departure date, the forecasts have to be found which fit most closely to the granules we need. We took each granule of the two-weeks-granularity and calculated the average forecast errors for each forecast that is closer to that granule than to any other granule.

Moreover, the forecast error for several days of week a coach departs is significantly different. Therefore, we made all our calculations separately depending on the day of week granularity (regarding the departure of coaches, not the date of forecasts). The result is shown in Figure 17.3. The visualization contains time-oriented data regarding both dimensions of time in our dataset. (1) The point in time a coach departs is contained in the multiple colored lines. (2) The forecast date (relative to the departure) is shown on the horizontal axis.

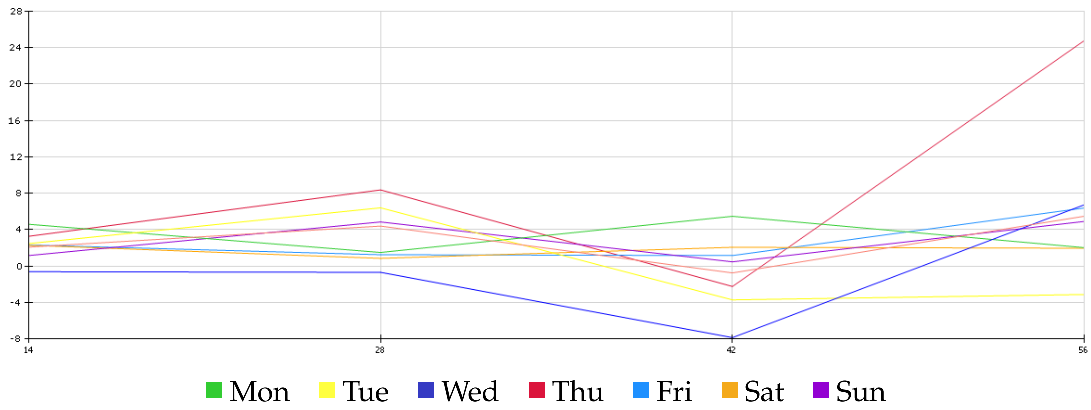


Figure 17.3: Deviance between Forecast and Actual Value. The values around certain number of days before (every two weeks) have been aggregated. The timestamp of events has been used to make a distinction between several days of week when an event could happen.

17.4 Two Dimensions of Visualizations for Two Dimensions of Time

In the last visualization we show based on our use case, we actually use both time axes pari passu. As we have two dimensions of time, we need to use two dimensions of space. As we have shown many times in this thesis, pixel-based visualization are suitable to encode the value when the two dimensions of space are already used for something else.

The resulting visualization (see Figure 17.4) is another variant of the GROOVE visualization from Chapter 12. The vertical axis shows the days the forecast has been done before the departure of a coach. The horizontal axis shows the day of departure (starting from the first departure contained in the dataset). The prominent angular lines are not corresponding to separate coach lines, but to separate forecasts. They are irregular, because forecasts are requested manually by analysts, resulting in the forecasts not being generated with regular intervals in between. We only show one coach line with one ride each day. However, as the forecast for each day is a different number of days in the past, there are only a few forecasts given per horizontal line. The forecast error for each forecast/departure combination is mapped to the lightness of the corresponding pixel. To provide an overview how good the forecast is at different weeks before the departure, the average forecast error for each week before the departure is mapped to the hue of all the pixels that belong to forecasts over that relative week.

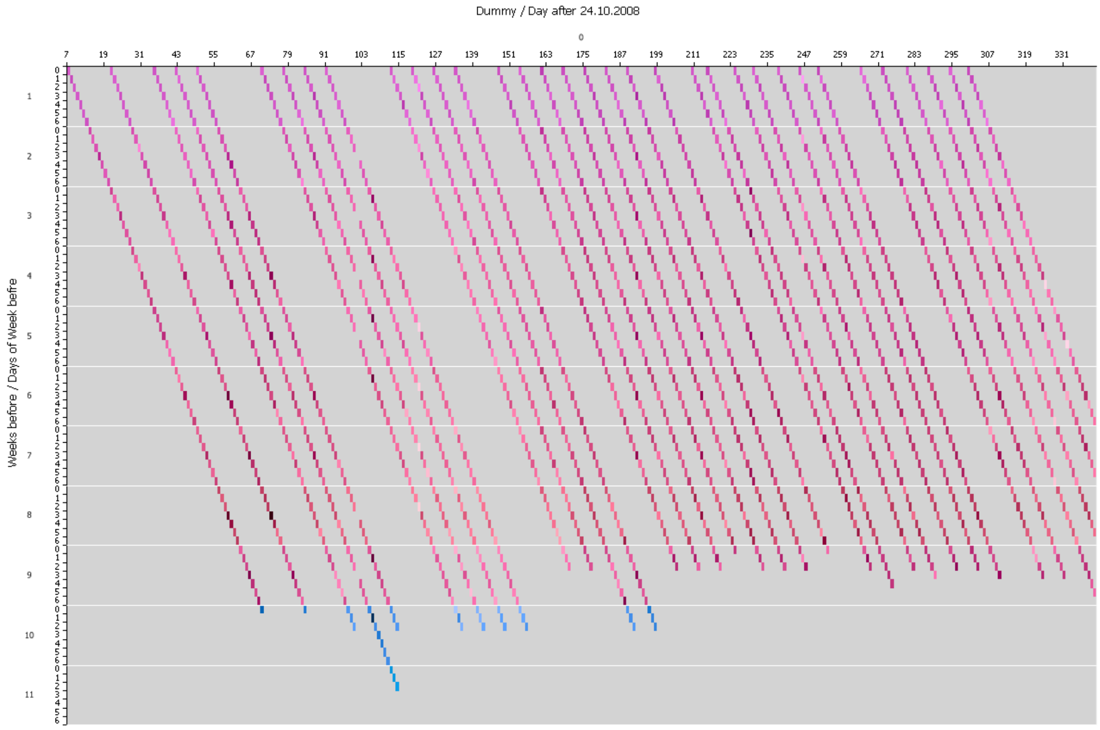


Figure 17.4: GROOVE-based Visualization for both Day of Departure and Day of Forecast. The angular lines do not correspond to separate departures but to separate forecasts.

The adavantage of our visualization is that much data can be shown without the need for aggregation. Many patterns can be spotted when the visualization is read in a correct way, but on the downside, it requires much training by users.

17.5 Assessment and Planned Evaluation

The visualizations have been developed in close cooperation with users who need to anaylze the forecasts for the coach capacities. We have explicitely targeted the needs they expressed. By applying the visualizations, we could gain some valuable insights and communicate them to the analysts.

As formal user study could not be performed yet. Chapter 20 details the questions we intend toe answer in an upcoming user study based on the visualizations presented here.

—Why, on the theory of Creation, should there be so much variety and so little real novelty?

Charles Darwin

18 Novel Interactive Visualization Methods: Summary

The third part of this thesis is the one that contains most (but not all) of our innovations.

Chapter 12 introduced the GROOVE visualizations, a central part of our contributions. Currently, there are three variants of GROOVE visualizations with different ways of performing the overlay. We also introduced granularity arrangement and interaction options that make GROOVE very flexible.

In Chapter 13, we showed that the structure of time can also be included in visualizations that are well-known even to lay users, for example bar diagrams.

In many use cases, granularities from social time and domain knowledge of users are most suitable to gain insight. For other datasets, it is necessary to gain new granularities not known to users. We demonstrated in Chapter 14 and 16 that visualizations like GROOVE are not inherently based on traditional granularities. Granularities can also be defined based on automated methods. The granularities based on those methods also have the advantage of preventing the gaps in granularities that have developed in social contexts. However, the automated methods do not replace the visualizations. Rather, they are used to provide additional information to users.

Another important aspect of our research project was the development of the MuTIny approach for performing the i-Apriori algorithm. In Chapter 15, we presented the interactive visualizations developed for this approach. The interactions in this chapter do not only influence the view, but also the parameterization of the algorithm itself.

Finally, in Chapter 17, we showed another aspect of the structure of time; multiple points of view. However, the approach to that aspect we introduced in this thesis is not showing different views by different sources, but different views over time.

The GROOVE visualizations have been evaluated in two user studies that showed advantages of our approach regarding the understandability of the visualizations. For the other visualizations, we have made preliminary assessments, but user studies have to be performed yet. We list detailed plans which questions to analyze in these studies in Chapter 20.

We could use the state of the art in InfoVis regarding the structure of time as a very good basis for our research. It was also possible, to improve several classical visualizations for our research. However, all of the visualizations developed by us introduce important new features regarding the structure of time as well as user interactions that do not only allow them to be used more efficiently, it also allows them to be used in ways not envisioned before.

Part IV

Conclusion and Future Work

—Finally, in conclusion, let me say just this.

Peter Sellers

19 Summary and Conclusion

We have presented novel scientific work in the Parts I, II, and III. The main part of our contribution is content of Part III, but in order to provide a logical organization that is easy to understand, we had also placed some of it in the Parts I and II.

19.1 Modelling the Visualization Process

In Chapter 2, we introduced the topic of time-oriented data by giving basic definitions needed to establish a common ground of understanding. We also presented what we consider the structure of time and showed the concept of granularities [Bettini et al., 2000] which can be used to model the calendar aspect. Our contributions depend on this basis.

Following the basic definitions, we described the concepts of visual variables [Bertin, 1967] in Chapter 3 and presented process-oriented frameworks that model the process of generating interactive visualizations. Based on the concepts and models we had mentioned at that point, we introduced a novel process-oriented framework which shares several characteristics with existing frameworks but includes operations specifically tailored to deal with the structure of time. With our framework, we can model in detail how visualizations deal with the structure of time and transfer the concepts existing visualizations use to new visualizations as well as visualizations that exist, but do not employ the structure of time before the inclusion of operations to deal with it.

When using the structure of time in visualization, several specific aspects have to be considered. These aspects can be difficulties for visualization developers, but some allow to seize new chances. We have researched those aspects that we learned from literature but we also came across novel aspects during user-based research in the project **DisCō**. Both kinds of aspects were described in Chapter 4.

In Chapter 5, Part I of the thesis was concluded by a description of users who usually have to deal with tasks for which the structure of time is relevant. We also described some of those tasks in detail. Finally, we also presented a way of measuring how users can gain insights from visualizations—the RIO visualization by Smuc et al. [2008].

The process-oriented framework we introduced in Part I served as backbone for the taxonomy introduced in the next part of the thesis.

19.2 Describing Visualizations

Part II of the thesis is where we consider state-of-the-art visualizations. Before doing this, we used our process-oriented framework to develop a new taxonomy that describes the design space of visualizations that make use of the structure of time. This taxonomy is not "flat", it rather spans the design space at several levels, according to the process-oriented framework, but also according to users' tasks. Therefore, we can make a large number of distinctions between visualizations, but also look specifically at some kind of distinctions, gaining order in our look at the design space.

In our state-of-the-art part we then described those visualizations that already existed and make use of the structure of time. By modelling them according to our taxonomy, we provided the building blocks to transfer the possibilities of state-of-the-art visualizations to new visualization methods. We also presented overview visualizations that show which visualizations are most similar to each other, and how similar they are. Furthermore, we gave an overview about the actual characteristics of all visualizations.

19.3 Novel Visualizations

Topic of Part III of the thesis were novel interactive visualization methods that make use of the structure of time.

First, we presented the GROOVE visualizations which since their development have become one of our staple visualizations used as a basis to test new ideas and gain a first insight in new datasets. The user evaluation of GROOVE showed that the calendar metaphor can help users to understand the arrangement and consequently to understand the data, no matter whether there actually is relevance of the structure of time in the data itself or whether there isn't. The same holds true for our interaction methods.

Still, the structural aspects that have influenced the data are also of great importance, at least according to assessments by our cooperation partner, our own assessments, and the assessments of the external InfoVis expert, who all agree on that matter. Therefore, a balance has to be found and future developements have to be made, based on further user studies. Several methods were presented by us how the structure of time inherent to a dataset can be found by automated methods, as an alternative to and expansion of users' domain knowledge.

We also showed other new visualizations as well as user interactions that make use of the structure of time. Particularly, we presented the MuTIny approach by Bertone [2009], adapting the i-Apriori algorithm for use with the structure of time, especially regarding the visualization. Even when the structure of time is not used in the MuTIny configuration, it is very helpful in the visualizations disclosing patterns to users. We also presented several interactive visualizations that help users to understand the results from MuTIny.

For the classification of our new visualizations, we applied the taxonomy we also used for the state of the art. Therefore, we can show easily which aspects were transferred from existing visualizations, and which are innovative features. We also demonstrate the potential of our approach in performing this transfer.

19.4 Assessment and Evaluation

User studies regarding our innovative visualization methods have mostly been focused on the GROOVE visualizations. However, those studies are not only evaluations of GROOVE; they have also provided experience we could use to make a general assessment about using the structure of time in visualization. The results clearly indicate that using the structure of time provides important advantages in regards to understandibilty of visualizations by users. User interaction possiblities like "details on demand" [Shneiderman, 1996] further help users in orienting themselves. The amount of insights users could gain from GROOVE during our user studies [Smuc et al., 2008, 2009] was impressive.

The other visualizations currently require formal user studies in order to be evaluated. We present our plans for the conduction of these user studies in Chapter 20. However, we have assessments from several experienced users we have cooperated with during the project **DisCō**. Furthermore, we gained a comprehensive assessment for most of the visualizations from the external InfoVis expert, who was predominantly positive on our results, but also provided some important points of criticism and suggestions for improvement.

19.5 Answers to Research Questions

The full answers to the research questions are distributed over the whole thesis. We summarize them in a way that answers the questions directly.

How can interactive visualization be used to explore time-oriented data according to the user's tasks while considering the structure of time?

> We have analyzed state-of-the-art visualizations using our process-oriented framework and our taxonomy. Based on our users tasks and on what we learned from these visualizations, we have generated our own innovative visualizations. These visualizations also employ user interaction and are examples how interactive visualization can be used to explore time-oriented data according to the user's tasks while considering the structure of time.

Which operations are part of a visualization that considers the structure of time and what is the design space they span?

> The process-oriented framework contains the operations that are part of a visualization that considers the structure of time. The taxonomy lays out their design space as far as it has been explored till now by the state of the art and our novel visualizations. By applying the taxonomy to classify the state of the art of existing visualizations that make use of the structure of time, we show that the operations indeed are the ones we have been looking for.

Which visualizations exist that can be used if the structure of time is important?

> The state-of-the-art-analysis of this thesis describes the visualizations that have been used if the structure of time is important so far. Furthermore, we have expanded the amount of visualizations for such situations considerably by adding our own innovative visualizations.

What are the situations when it is feasible to use the structure of time for visualization even when it is less prominent in the data—and how can it be done?

> The most important usage for the structure of time beside paying heed to the artifacts it has caused in the data is guiding users. For example, our user studies showed that the calendar metaphor is a major advantage in guiding users independently of structures in the data. The most extreme variant is the calendar sheet itself, being void of additional data, only showing the structure of time itself, still providing important information to users.

Additionally, we have shown in Chapter 15 that patterns found by automated methods, like the MuTIny approach, can be used to generate structural information about the dataset. Independently of the connection between that structure and the data itself, it can be used to make the visualization better understandable, for example by reducing the number of patterns cut in their visualization.

19.6 Main Contributions

Process-oriented Framework for dealing with time-oriented data according to the structure of time. Our framework enables the description of visualizations that make use of the structure of time. It serves as a backbone for our taxonomy. It enables visualization designers to transfer methods from one visualization to another. By making the framework compatible to other established frameworks, we hope that InfoVis experts from other domains will participate in making InfoVis a huge design space of operations, capable of powerful visualization tailored specifically for users' needs, and reducing the importance of predefined "visualization classes", like horizontal bar plot, vertical bar plot, horizontal stacked bar plot, and so on. The applicability of the framework has been assessed by using it as a "backbone" for the taxonomy.

Taxonomy of visualization that make use of the structure of time. The taxonomy spans the design space, describes existing visualizations, and serves as a set of tools for novel visualizations. The taxonomy has been assessed by collecting the visualizations from current state of the art which employ the structure of time and classifying them according to the taxonomy.

Interactive Visualization Methods that make use of the structure of time. We have presented the GROOVE visualizations and shown their application in conjunction with granularities generated by automated methods. Well-known statistical visualization methods have been expanded by applying the structure of time. Furthermore, we have presented novel interactive visualization methods that are related to the MuTIny approach, showing patterns from the i-Apriori algorithm empoying the structure of time as part of a Visual Analytics framework. We have also presented our approach at showing different points of view. The interactive visualizations, above all the GROOVE visualization, have been developed and tested in cooperation with users who frequently deal with challenges revolving around time-oriented data and the structure of time. The GROOVE visualizations have been evaluated in two user studies, for the other visualizations we currently only have assessments by visualization experts or experienced users, but we show concrete plans for future user studies in Chapter 20.

—The best way to predict the future is to invent it.

Alan Kay

20 Future Work

Our future work can be divided in four strands.

1. Naturally, we need to implement the interactive visualizations we have already planned.
2. Afterwards, these visualizations need to be tested in user studies. User input from these studies as well as the studies we already have performed.
3. InfoVis ist becoming more and more of an integrated part in the new science of Visual Analytics. Therefore, we need to deepen the integration of interactive visualizations and automated methods that we already have in two ways:
   a) We want to integrate the interactive visualizations that are currently implemented separately into an Visual Analytics framework.
   b) We want to proceed from our Visual Analytics framework that is currently focused on the MuTIny approach to a Visual Analytics framework that is expandable and provides several automated methods as well as interactive visualizations, all being integrated with each other in an information discovery process.
4. We like to deepen the research in visualization theory started in the Chapters 2, 3, and 8. We want to design a theoretical framework for the Visual Analytics process of time-oriented data according to the structure of time and include the parts for InfoVis we already have developed.

Many of these tasks have been shifted to future work because we intended to publish the thesis with the current results simultaneously with the closure of the research project **DisCō**. We see great prospect in the fact that the current future work from the thesis provides a perfect starting point for a new project.

20.1 Finishing Prototypical Implementation of Current Ideas

Several visualizations from Chapter 15 currently only exist as mock-ups. Furthermore, not all possibilities of GROOVE are implemented in our current prototypes. The automated methods used for gaining new granularities need to be used manually instead of being integrated in one application. All those visualizations need to be implemented including the user interactions. These tasks are intended to be performed in an integrated Visual Analytics Framework we have planned (see Section 20.3).

20.1.1 GROOVE Featuring Automated Analysis Methods

The main development direction regarding GROOVE goes towards the integrated visualization providing many visual variables, as outlined in Chapter 16. Furthermore, we intend for this integrated visualization to contain all user interactions, above all the various methods of applying granularity switching for semantic paging and zooming, as well as detail on demand.

The automated analysis methods to generate granularities need to be included, giving users a wide selection of granularities from social time, granularities found by automated methods, and granularities they can define themselves.

Finally, the new versions of our prototypical GROOVE implementation have to include the feedback we gained from the User Studies regarding GROOVE (see Subsection 20.2.1).

20.1.2 Interactive Visualizations Showing Patterns

Our goal is to include all of the visualizations we have presented in Chapter 15 in the framework we have implemented prototypically. Furthermore, the new as well as the existing visualizations are to be connected to the algorithm as described in that chapter. After finishing the implementations, user studies have to show how further development regarding these visualizations has to be conducted.

20.2 Responding to Input from Users and the InfoVis Community

Over the course of our developments, we already used the possibility of introducing input from our user studies, as well as colleagues from within and without the **DisCō** project. However, much feedback we already have collected still has to be incorporated prototypically.

20.2.1 GROOVE User Studies

Most of the feedback we got revolves around the fact of developing a legend that shows all variants of colors in GROOVE, but that users could still understand, especially regarding color overlay. We also have ideas for user interactions performed on the legend and influencing the coloring of the GROOVE visualization itself.

Moreover, most user input is focused on combining all the possiblities we already have in on implementation. For example, user input on the variant for the business consultants, which has a less immediate user interaction handling regarding the switching of granularities, spawned user input that demanded exactly that. On the other hand, the implementation used by the business consultants provides tooltips to interactively show exact values. These tooltips were demanded in the tests with the interactive prototype which, on the contrary, provides powerful and fast methods for granularity switching. Consequently, we only need to combine the possibilities we already have implemented and tested in our prototypes into one prototype in order to respond to these user inputs.

20.2.2 Gaining Granularities from Data

While the idea itself has been assessed complaisantly, we need further innovations to show the relation between the calendar structures found in the data and social calendar structures known to human users. A first step would be the reintroduction of axes as proposed by the external InfoVis expert.

Depending on the algorithm used to gain the patterns, it will also be necessary to show results from that algorithm used in granularity generation. An important example for this case are the patterns found by the MuTIny approach. In case patterns do not overlap (which unfortunately, is rarely the case at the current development state of MuTIny), an interesting variant might be using hue to highlight the patterns and lightness to encode actual values.

20.2.3 Visualizations Showing Patterns

We have analyzed preliminary results from showing our MuTIny approach to users. These results hint that people who are not experts in Data Mining have problems in understanding patterns based on abstract values like $e_0 I_1 e_1$. For those users, but also for Data Mining experts, we need alternatives that give patterns designations that can be understood better by humans. The visualizations presented in Chapter 15 already show possiblities for doing this. We have to analyze whether these ideas will solve the problem. In implementing the visualizations mentioned there, we will also include the verbose assessment results from the external InfoVis expert.

20.2.4 Further User Evaluation

Follwing the implementation of our visualizations presented in Chapters 13–16, we intend to pursue further user studies:

- The visualizations from Chapter 13 will be evaluated as part of the evaluation of the Visual Analytics framework planned (see Section 20.3). Main questions:
  - Can the visualizations provide information for users that other visualizations can't?
  - Is the lie factor low enough so they can be applied safely?
- The visualizations from Chapter 15 will be implemented either in the existing or in the planned Visual Analytics framework (which will be reimplemented based on experiences from the existing framwork). Afterwards, there will be a user study with these questions:
  - Are the visualizations adequate to disclose information about the patterns to users?
  - Are the visualizations adequate to serve as a control mechanism for the algorithm?
- As the Chapters 14 and 16 are primarily not about the visulization technique (GROOVE) used there but about gaining granularities through automated methods, the user study for these chapters has to focus on that question. Therefore, we will choose one or two visualization techniques and compare different kinds of granularities, in order to answer the following questions:
  - Are granularities from automated sources suitable for users?
  - Depending on task, which methods for automated granularity generation is best?

  - To what extend are granularities important because they correspond to structures in the data, and to what extend are granularities important to guide users in understanding visualizations?

- Regarding the visualizations from Chapter 17, the following questions will have to be answered in an upcoming user study, most likely initially performed based on the visualizations we have presented in that chapter:
  - Can users interpret the visualizations correctly?
  - Which insights can be gained from the visualizations?
  - Are the insights gained from the visualizations sufficient for that tasks or are further visualizations required?

20.3 From the Visualization Process to the Visual Analytics Process

The focus of this thesis has been using the structure of time in InfoVis. Still, being part of a Visual Analytics project, several connections to automated methods had to be drawn. The structure of time, with many prior work stemming from Data Mining, also has to be used in automated methods. The most promising approach is using it integrated over a whole Visual Analytics process.

In Chapter 16 we have summarized methods to gain granularities necessary for applying the structure of time in a Visual Analytics process. Those methods themselves can also be considered as an inital part of this process. Further steps include analyzing the data based on these granularities.

One model for the Visual Analytics process has been proposed by Keim et al. [2008]. We have expanded it and present our version in Figure 20.1.

We want to expand our application of the structure of time from the InfoVis process to the whole Visual Analytics process. Partly, this step has already been performed. For visualizations, it has been done in this thesis, for automated analysis, several authors, like Bertone [2007], have based their work on the granularity algebra by Bettini et al. [2000]. Including more than one Visual Analytics step in one single scientific contribution has also been done, even in the context of the structure of time, for example by van Wijk and van Selow [1999]. The step of building scientific models from hypotheses (bold in Figure 20.1) is one we consider especially worth of future research. Our user studies often resulted in users formulating promising hypotheses, but without proper transformation in a way that is suitable for automated methods, it cannot be used for further computation but only written down as—preliminary—insights.

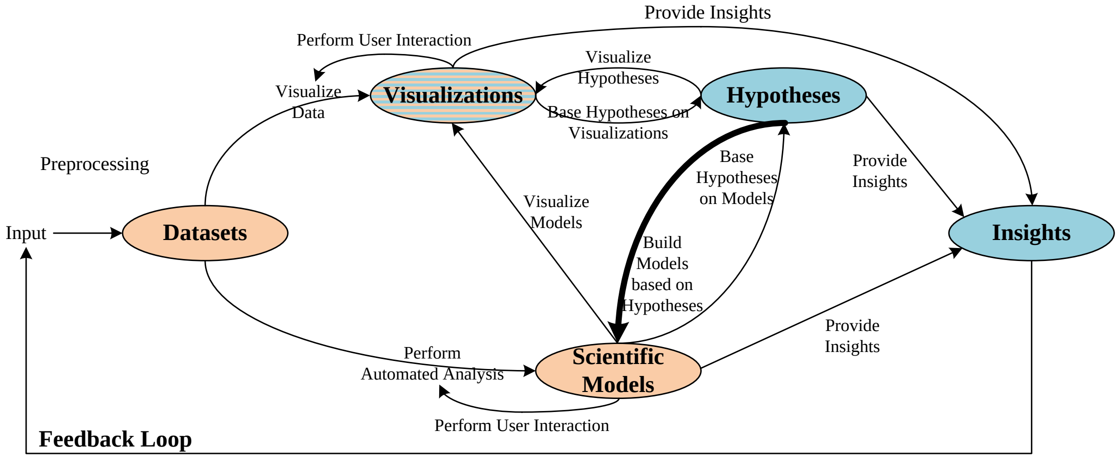


Figure 20.1: Visual Analytics Process Adapted from Keim et al. [2008]. Orange steps are usually performed by automated systems, blue steps by human users. Visualizations are a connection between those two parts of the process.

Therefore, our goal is to provide methods for all steps in the Visual Analytics process network, regarding time-oriented data, and focused on using the structure of time. These methods are to be integrated in a Visual Analytics framework that can be used for user studies and further research.

20.4 Expanding Visualization Theory

Our process-oriented framework from Chapter 3 and our taxonomy from Chapter 8 have been very helpful for our work of classifying the state of the art as well as developing new interactive visualizations. Despite of that, we have encountered many aspects or tasks that require an expansion of the models, or an even greater model.

20.4.1 More Complex Visual Variables

As we already mentioned, visual variables can be defined at various complexity levels. Bertin [1967] only used the most basic variants. Later research resulted in visual variables of increasing complexity [Mackinlay, 1986], up to a level where even an axis can be considered a visual variable [Bostock and Heer, 2009], an approach we also used in this thesis. It is possible to consider even more complex visual variables. For example, in tools like Excel, a visualization is chosen and data is mapped to it. However, we consider that approach too limited. This opinion is shared by other scientists, for example the already mentioned Bostock and Herr, who use deliberately a complexity level for their Protovis visualization library [Bostock and Heer, 2009] which is above the usual compexity of visual variables, but below the fixed templates of software like Excel.

Another option for more complex visual variables is increasing the complexity of the data side instead of the complexity of the visual side. In our framework, we can map granule labels to axes, but dealing with the calendar aspect is done separately. The axis as defined by us "knows nothing of the structure of time". That approach is suitable for our goal of providing a framework that can break up the theory behind a visualization as simple as possible, but also as complete as possible.

Defining visual variables like axes that have their own access to the structure of time and can be target of a data attribute more complex than a granularity label would make the deconstruction of visualizations shorter and more elegant, resulting in smaller tables with less repetition than the ones we have.

Our goal is to combine the completeness of the breakdown we used in this thesis with the elegance of those more complex visual variables. As possible solution, we envision an object-oriented approach, but at our current state of research, there are no detailed ideas how such an approach could be formulated.

20.4.2 Modelling Interaction

At the current state of our process-oriented framework, interaction is modelled in a way that users change the parameterization of operations and therefore the visualization changes. This model is simple and correct, but it also is not very detailed. We intend to find a way of using formal user interactions with formal changes in the parameterization of the operators, so that by looking at the model of a visualization, it is clear what can be done with user interactions at the current state of development. Clearly, it is complicated to find a working and correct model to do this, so we do not expect results in the near future.

20.4.3 The Visual Analytics Process

As a basis for the Visual Analytics framework implementation we have planned, it is necessary to have a theoretical model of the process. Therefore, a primary goal of our work is to build a theoretical Visual Analytics framework that works similar to the InfoVis framework we have presented in Chapter 3. If possible, we want to integrate the InfoVis framework as one part of the larger Visual Analytics framework.

20.4.4 Other Areas of Visual Analytics

Our focus in Visual Analytics is time-oriented data. In that context, we have a further focus on analyzing data according to the structure of time. Other research groups are focusing on Visual Analytics regarding different kinds of data, for example geo-spatial data. By cooperating with other research groups, we hope that one day it will be possible to provide a theoretical model that allows full understanding of a working Visual Analytics process. Such a model would be immensely complex, and finding it might well be one of the Grand Challenges of Visual Analytics. The Grand Challenges for Information Visualization [Grinstein et al., 2008] are being discussed since 2008, so it might be time for Visual Analytics to gain its own Grand Challenges.

Appendix

A

UML 2 Conventions

In this chapter we focus on UML 2 activity diagrams (an example is shown in Figure A.1). Data states are represented as rectangles without rounded corners in UML 2. However, they are usually omitted. Activities, in our case, data transformations, are represanted as rectangles with rounded corners. It is assumed that between two data transformations there always is a data state, even if it is omitted. This notation is adequate for a process-oriented framework.

Often, UML 2 activity diagrams contain an initial and a final node. As the diagrams from this dissertation only show small excerpts of a larger process, we do not use those nodes. Instead, we show the entry end exit points as data states that represent datasets or visualizations at different stages.

UML 2 activity diagrams support encapsulation of data transformations that belong together. Each diagram represents a superordinate activity that is surrounded by a big rounded rectangle. Smaller compound activities can be encapsulated by a dashed rounded rectangle that is called an expansion region in UML terms. Partial activities inside expansion regions can be performed multiple times. In the upper left corner of these rectangles, name and parameters are given. Below, information about iteration or recursion is given. In the upper right corner, conditions are defined: *Preconditions* have to be fulfilled before the activity can be performed. *Postconditions* have to be guaranteed to be fulfilled after the activity is finished.

While small rounded rectangles can be seen as atomic activities, at least in the context of the respective UML diagram's detail level, the dashed rounded rectangles can be seen as composite activities. Both can be connected with arrows showing the control flow. The small squares between rectangles and arrows are input and output pins. We use them to model the fact that by defining granularities, time-oriented data becomes multivariate regarding time. Many partial activities can be performed in any order or even multiple times, depending on implementation and user interaction. This is modelled by using fork and join nodes, as well as dataflow backwards to these nodes.

Transcending the UML standard, we have parts shown in blue that depict the points where user interaction is possible.

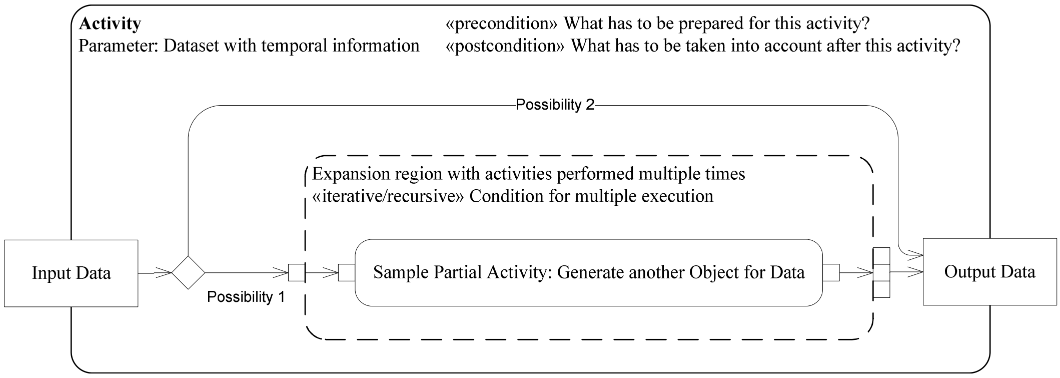


Figure A.1: An Example UML 2 Diagram. All elements are described in this appendix.

Bibliography

Wolfgang Aigner. *Visualization of Time and Time-Oriented Information: Challenges and Conceptual Design*. Theses, Vienna University of Technology, Institute of Software Technology and Interactive Systems, February 2006. PhD Thesis; Supervisors: Silvia Miksch (Vienna University of Technology), Heidrun Schumann (University of Rostock).

Wolfgang Aigner, Alessio Bertone, Silvia Miksch, Heidrun Schumann, and Christian Tominski. Towards a Conceptual Framework for Visual Analytics of Time and Time-Oriented Data. In S. G. Henderson, B. Biller, M. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, editors, *Proceedings of the 2007 Winter Simulation Conference*, pages 721–729, December 2007a. URL http://ike.donau-uni.ac.at/publications/PDF/2007/wsc07/final/aigner_2007_wintersim_framework.pdf. invited Paper.

Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, and Christian Tominski. Visualizing Time-Oriented Data - A Systematic View. *Computers & Graphics*, 31(3):401–409, June 2007b. URL http://www.donau-uni.ac.at/imperia/md/content/department/ike/ike_publications/2007/refereedjournalarticles/aigner_2007_cg_visualizing-time-oriented-data.pdf.

Wolfgang Aigner, Alessio Bertone, Tim Lammarsch, Silvia Miksch, and Alexander Rind. Interactively Exploring Time-Oriented Data. In *Workshop Notes of CHI 2009 workshop "Interacting with temporal data" at Conference on Human Factors in Computing Systems (CHI 2009)*. ACM Press, 2009. URL http://ike.donau-uni.ac.at/publications/PDF/2009/CHI09/final/aigner_chi09_interacting-time.pdf.

Kate Beard, Heather Deese, and Neal R. Pettigrew. A Framework for Visualization and Exploration of Events. *Information Visualization*, 7(2):133–151, 2008.

Fabian Bendix, Robert Kosara, and Helwig Hauser. Parallel Sets: Visual Analysis of Categorical Data. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005*, pages 133–140, 2005.

Jacques Bertin. Semiologie Graphique: Les Diagrammes, les Reseaux, les Cartes. 1967.

Jacques Bertin. *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, 1983.

Enrico Bertini, Catherine Plaisant, and Giuseppe Santucci. BELIV'06: Beyond Time and Errors; Novel Evaluation Methods for Information Visualization. *Interactions*, 14(3): 60, 2007.

Alessio Bertone. *A Matter of Time: Multi-time Interval Pattern Discovery to Preserve the Temporal Information in between*. Theses, December 2009. Supervisors: Silvia Miksch (Danube University Krems), Margit Pohl (Vienna University of Technology).

Alessio Bertone. A Matter of Time: how to visually analyze multivariate (and multidimensional) data, irregularly sampled and having multiple granularities. In *Doctoral Consortium in AIME 07 Conference*, pages 9–16, 2007. URL http://ike.donau-uni.ac.at/publications/PDF/2007/aime_2007/Alex_Matter_of_Time/Final%20Version/Aime07-%20Final%20Version.pdf.

Claudio Bettini, Sushil G. Jajodia, and Sean X. Wang. *Time Granularities in Databases, Data Mining and Temporal Reasoning*. Springer-Verlag New York, Secaucus, NJ, USA, 2000.

Michael Bostock and Jeffrey Heer. Protovis: A Graphical Toolkit for Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1121–1128, 2009.

Mark Bruls, Kees Huizing, and Jarke J. van Wijk. Squarified Treemaps. In *Proceedings of the joint Eurographics and IEEE TCVG Symposium on Visualization*, pages 33–42. Citeseer, 2000.

Andreas Buja, Dianne Cook, and Deborah F. Swayne. Interactive High-dimensional Data Visualization. *Journal of Computational and Graphical Statistics*, pages 78–99, 1996.

Stuart K. Card and Jock D. Mackinlay. The Structure of the Information Visualization Design Space. In *IEEE Symposium on Information Visualization, 1997. Proceedings.*, pages 92–99, 1997.

Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, San Francisco, 1999.

John V. Carlis and Joseph A. Konstan. Interactive Visualization of Serial Periodic Data. In *Proceedings of the 11th annual ACM symposium on User interface software and technology*, page 38. ACM, 1998.

Qin Chengzhi, Zhou Chenghu, and Pei Tao. Taxonomy of Visualization Techniques and Systems—Concerns between Users and Developers are Different. In *Asia GIS Conference*, pages 31–45, 2003.

Ed H. Chi. A Taxonomy of Visualization Techniques Using the Data State Reference Model. In *IEEE Symposium on Information Visualization, 2000. InfoVis 2000*, pages 69–75, 2000.

Ed H. Chi and John T. Riedl. An Operator Interaction Framework for Visualization Systems. In *IEEE Symposium on Information Visualization, 1998. Proceedings*, pages 63–70, 1998.

Mei C. Chuah and Steven F. Roth. On the Semantics of Interactive Visualization. In *Proceedings IEEE Symposium on Information Visualization*, volume 96, pages 29–36.

William S. Cleveland. *Visualizing Data*. Hobart Press, Summit, NJ, USA, 1993.

William S. Cleveland. *The Elements of Graphing Data, Revised Edition*. Hobart Press, Summit, NJ, USA, 1994.

Andrew M. Colman. *A Dictionary of Psychology*. Oxford University Press New York, 2001.

G. Cornélissen, M. Engebretson, D. Johnson, K. Otsuka, N. Burioka, J. Posch, and F. Halberg. The Week, Inherited in Neonatal Human Twins, Found also in Geomagnetic Pulsations in Isolated Antarctica. *Biomedicine & Pharmacotherapy*, 55:32–50, 2000.

Chaouki Daassi, Marlon Dumas, Marie-Christine Fauvet, Laurence Nigay, and Pierre-Claude Scholl. Visual Exploration of Temporal Object Databases. In *Proceedings of 16 iemes Journees Bases de Donnees Avancees (BDA, French Conference on Databases)*, pages 159–178, Oct 2000.

Chaouki Daassi, Laurence Nigay, and Marie-Christine Fauvet. Visualization Process of Temporal Data. *Lecture notes in computer science*, pages 914–924, 2004.

Geoffrey Ellis and Alan Dix. An Explorative Analysis of User Evaluation Studies in Information Visualisation. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, page 7. ACM, 2006.

Antonio Gabaglio. *Theoria Generale della Statistica*. Milan, 2nd edition edition, 1888.

Iqbal A. Goralwalla, M. Tamer Özsu, and Duane Szafron. *An Object-Oriented Framework for Temporal Data Models*, pages 1–35. Springer, 1998.

Georges G. Grinstein, Daniel A. Keim, and Tamara Munzner. Grand Challenges for Information Visualization. In *IEEE Information Visualization Conference*, 2008.

Johannes Gärtner, Wolfgang Aigner, Alessio Bertone, Robert Klausner, Tim Lammarsch, Eva Mayr, Silvia Miksch, Hannah Risku, and Michael Smuc. Visual Analytics for Workforce Requirements. In *Accepted extended abstract at 19th International Symposium on Shiftwork and Working Time and Well-being in the 24-h Society, Venezia, Italy*, volume 29(3), pages 29–38. IEEE Computer Graphics & Application, 2009.

Robert L. Harris. *Information Graphics: A Comprehensive Illustrated Reference*. Oxford University Press, 1999.

Ross Ihaka. Colour for Presentation Graphics. In *Proceedings of DSC*, page 2, 2003.

Brian Johnson and Ben Shneiderman. Tree-maps: A Space-filling Approach to the Visualization of Hierarchical Information Structures. In *Proceedings of the 2nd conference on Visualization'91*, pages 284–291. IEEE Computer Society Press Los Alamitos, CA, USA, 1991.

Daniel A. Keim. Visual Exploration of Large Databases. *Communications of the ACM*, 44 (8):38–44, 2001.

Daniel A. Keim. Information Visualization and Visual Data Mining. *IEEE transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

Daniel A. Keim and Hans-Peter Kriegel. Visualization Techniques for Mining Large Databases: A Comparison. *IEEE Transactions on Knowledge and Data Engineering*, 8(6): 923–938, 1996.

Daniel A. Keim, Hans-Peter Kriegel, and Mihael Ankerst. Recursive Pattern: A Technique for Visualizing very Large Amounts of Data. In *Proceedings of IEEE Visualization (Vis95)*, pages 279–286, 1995.

Daniel A. Keim, Florian Mansmann, Jörn Schneidewind, Jim Thomas, and Hartmut Ziegler. Visual Analytics: Scope and Challenges. *Lecture Notes In Computer Science*, 4404:76–90, 2008.

Olga A. Kuchar, Thomas J. Hoeft, Susan Havre, and Kenneth A. Perrine. Isn't it about Time? *IEEE Computer Graphics and Applications*, 26(3):80–83, 2006.

Tim Lammarsch. A Compound Approach for Interactive Visualization of Time-Oriented Data. In *Proceedings of IEEE Symposium on Visual Analytics Science and*

Technology 2008 (VAST08), pages 177–178. IEEE Computer Society Press, October 2008. URL http://ike.donau-uni.ac.at/publications/PDF/2008/visweek_2008/phd_colloquium/submission%20and%20final/DoctoralColloquium.pdf.

Tim Lammarsch, Wolfgang Aigner, Alessio Bertone, Johannes Gärtner, Silvia Miksch, and Thomas Turic. A Comparison of Programming Platforms for Interactive Visualization in Web Browser Based Applications. In *Proceedings of 12th International Conference on Information Visualisation (IV08)*, pages 194–199. IEEE Computer Society Press, July 2008. URL http://ike.donau-uni.ac.at/publications/PDF/2008/lammarscht-PlatformsForVisualization.pdf.

Tim Lammarsch, Wolfgang Aigner, Alessio Bertone, Johannes Gärtner, Eva Mayr, Silvia Miksch, and Michael Smuc. Hierarchical Temporal Patterns and Interactive Aggregated Views for Pixel-based Visualizations. In *Proceedings of 13th International Conference on Information Visualisation (IV09)*, pages 44–49. IEEE Computer Society Press, 2009. URL http://ike.donau-uni.ac.at/publications/PDF/2009/lammarsch2009_htpaiavfpv.pdf.

Jock D. Mackinlay. Automating the Design of Graphical Presentations of Relational Information. *ACM Transactions on Graphics*, 5(2), 1986.

Wolfgang Müller and Heidrun Schumann. Visualization Methods for Time-dependent Data—an Overview. In *Simulation Conference, 2003. Proceedings of the 2003 Winter*, volume 1, 2003.

Tamara Munzner. Outward and Inward Grand Challenges, 2008. Part of VisWeek08 Panel: Grand Challenges for Information Visualization.

Chris North. Toward Measuring Visualization Insight. *IEEE Computer Graphics and Applications*, pages 6–9, 2006.

OLIVE. On-line Library of Information Visualization Environments, 1997, Last accessed 04/11/2009. URL https://otal.umd.edu/olive/.

Darius Pfitzner, Vaughan Hobbs, and David M.W. Powers. A Unified Taxonomic Framework for Information Visualization. In Tim Pattison and Bruce Thomas, editors, *Australian Symposium on Information Visualisation, (invis.au'03)*, volume 24 of *CRPIT*, pages 57–66, Adelaide, Australia, 2003. ACS.

Peter Pirolli and Stuart K. Card. The Sensemaking Process and Leverage Points for Analyst Technology as Identified through Cognitive Task Analysis. In *Proceedings of International Conference on Intelligence Analysis*, volume 2005, pages 2–4, 2005.

Catherine Plaisant, Jean-Daniel Fekete, and Georges G. Grinstein. Promoting Insight-based Evaluation of Visualizations: From Contest to Benchmark Repository. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):120–134, 2008.

Markus Rester, Margit Pohl, Klaius Hinum, Silvia Miksch, Christian Popow, Susanne Ohmann, and Staven Banovic. Methods for the Evaluation of an Interactive Info Vis Tool Supporting Exploratory Reasoning Processes. In *BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV 2006) held in conjunction with Advanced Visual Interfaces (AVI-2006), International Working Conference*, 2006. URL http://ike.donau-uni.ac.at/publications/PDF/2006/beliv_2006/final/rester_2006_beliv_evaluation-concept.pdf.

John F. Roddick and Myra Spiliopoulou. A Bibliography of Temporal, Spatial and Spatio-temporal Data Mining Research. *ACM SIGKDD Explorations Newsletter*, 1(1):34–38, 1999.

Bernice E. Rogowitz and Lloyd A. Treinish. Data Visualization: The End of the Rainbow. *IEEE spectrum*, 35(12):52–59, 1998.

Ruth Rosenholtz, Yuanzhen Li, Jonathan Mansfield, and Zhenlan Jin. Feature Congestion: A Measure of Display Clutter. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 761–770. ACM New York, NY, USA, 2005.

Daniel M. Russell, Mark J. Stefik, Peter Pirolli, and Stuart K. Card. The Cost Structure of Sensemaking. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 269–276. ACM New York, NY, USA, 1993.

Purvi Saraiya, Chris North, and Karen Duca. An Insight-based Methodology for Evaluating Bioinformatics Visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):443–456, 2005.

Milton H. Shimabukuro, Edilson F. Flores, Maria C.F. de Oliveira, and Haim Levkowitz. Coordinated Views to Assist Exploration of Spatio-Temporal Data: A Case Study. In *Proceedings of the 2nd International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV04)*, pages 107–117. IEEE, July 13 2004.

Ben Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *IEEE Symposium on Visual Languages, 1996. Proceedings.*, pages 336–343, 1996.

Ben Shneiderman. Inventing Discovery Tools: Combining Information Visualization with Data Mining. *Lecture Notes in Computer Science*, pages 17–28, 2001.

Ben Shneiderman and Catherine Plaisant. Strategies for Evaluating Information Visualization Tools: Multi-dimensional In-depth Long-term Case Studies. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, page 7. ACM, 2006.

Samuel Silva, Joaquim Madeira, and Beatriz S. Santos. There is More to Color Scales than Meets the Eye: A Review on the Use of Color in Visualization. In *Information Visualization, 2007. IV'07. 11th International Conference*, pages 943–950, 2007.

Herbert A. Simon. *The Sciences of the Artificial*. MIT press, 1969.

Michael Smuc, Eva Mayr, Tim Lammarsch, Alessio Bertone, Wolfgang Aigner, Hanna Risku, and Silvia Miksch. Visualizations at First Sight: Do Insights Require Traininig? In *Proceedings of 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society (USAB 2008)*, pages 261–280. Springer, November 2008. URL http://ike.donau-uni.ac.at/publications/PDF/2008/paper_USAB08_2008-09-15_final.pdf.

Michael Smuc, Eva Mayr, Tim Lammarsch, Wolfgang Aigner, Silvia Miksch, and Johannes Gärtner. To Score or Not to Score? Tripling Insights for Participatory Design. *IEEE Computer Graphics and Applications*, 29(3):29–38, May 2009. URL http://ike.donau-uni.ac.at/publications/PDF/2009/smuc_2009_ToScoreOrNotToScore.pdf.

Rebecca R. Springmeyer, Meera M. Blattner, and Nelson L. Max. A Characterization of the Scientific Data Analysis Process. In *Proceedings of the 3rd conference on Visualization'92*, pages 235–242. IEEE Computer Society Press Los Alamitos, CA, USA, 1992.

James J. Thomas and Kristin A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE, 2005. URL http://nvac.pnl.gov/agenda.stm.

Melanie Tory and Torsten Möller. A Model-based Visualization Taxonomy. In *Proceedings of the 2004 IEEE Symposium on Information Visualization*, 2004.

J. Gregory Trafton, Susan S. Kirschenbaum, Ted L. Tsui, Robert T. Miyamoto, James A. Ballas, and Paula D. Raymond. Turning Pictures into Numbers: Extracting and Generating Information from Complex Visualizations. *International Journal of Human-Computers Studies*, 53(5):827–850, 2000.

Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press Cheshire, CT, 1983.

Edward R. Tufte. Envisioning Information. Cheshire, CT, 1990.

Lisa Tweedie. Characterizing Interactive Externalizations. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 375–382. ACM New York, NY, USA, 1997.

Jarke J. van Wijk and Edward R. van Selow. Cluster and Calendar Based Visualization of Time Series Data. In *Proceedings of the IEEE Symposium on Information Visualization 1999 (InfoVis99)*, pages 4–9, 1999.

Martin Wattenberg. Arc Diagrams: Visualizing Structure in Strings. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis02)*, pages 110–116. IEEE, October 28–29 2002.

Marc Weber, Marc Alexa, and Wolfgang Müller. Visualizing Time-Series on Spirals. In *Proceedings of the IEEE Symposium on Information Visualization 2001 (InfoVis01)*, pages 7–14, October 2001a.

Marc Weber, Marc Alexa, and Wolfgang Müller. Visualizing Time-series on Spirals. In *proceedings of the IEEE Symposium on Information Visualization*, page 7, 2001b.

Stephen Wehrend and Clayton Lewis. A Problem-oriented Classification of Visualization Techniques. In *Visualization, 1990. Visualization'90., Proceedings of the First IEEE Conference on*, pages 139–143, 1990.

Eva Wohlfart, Wolfgang Aigner, Alessio Bertone, and Silvia Miksch. Comparing Information Visualization Tools Focusing on the Temporal Dimensions. In *Proceedings of 12th International Conference on Information Visualisation (IV08)*, pages 69–74, July 2008. URL http://ike.donau-uni.ac.at/publications/PDF/2008/Wohlfart%20-%20%20Comparison_short.pdf.

Jo Wood, Aidan Slingsby, and Jason Dykes. Using Treemaps for Variable Selection in Spatio-temporal Visualisation. *Information Visualization, 7*, 3(4):210–224, 2008. URL http://geoanalytics.net/GeoVis08/a26.pdf.

Eviatar Zerubavel. *The Seven Day Circle: The History and Meaning of the Week*. University Of Chicago Press, 1989.

Michelle X. Zhou and Steven K. Feiner. 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. New York, NY, USA, 1998.

Curriculum Vitae

Current Address	Ernst-Krenek-Straße 16a 3500 Krems, Austria
Secondary (Long-term) Adress	Schillerstraße 8 68549 Ilvesheim, Germany
Nationality	German
Homepage	http://www.lammarsch.de

Education

06/2007–PRESENT	PhD studies at Vienna University of Technology (TU Vienna), Austria
10/1999–10/2005	Diploma (Bachelor+Master) studies at the University of Mannheim, Germany Master thesis on "Modelling of Cutting Operations in the Eye Surgery Simulator EYESI"

Professional employment

05/2007–PRESENT	Research Associate, Department of Information and Knowledge Engineering, Danube University Krems, Austria
10/2005–04/2007	Software developer for mediDOK GmbH, Dossenheim, Germany
2002–2003	Student research assistant at the University of Mannheim, Germany
1999–2001	Webdesigner as well as software developer at MBIT GmbH, Mannheim, Germany
1994–1995	Freelance occupation as a typesetter for Thomson Publishing, Bonn, Germany

Research Area

Visual Analytics, Information Visualization, Computer Graphics, Pattern Recognition, and Human-Computer Interaction, (HCI)

Further skills and competences

Honors and Awards	2001: Team award for best Cludeo AI at the University of Mannheim
Scientific community	Reviewer for various conferences, symposia, workshop and journals
Teaching skills	Tutor for Computer Science, Programming Methodology, and Information Visualization

Krems, April 22, 2010