BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INFORMATION SYSTEMS
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

IMPACT OF SUBJECTIVE VISUAL PERCEPTION ON AUTOMATIC EVALUATION OF DASHBOARD DESIGN

VLIV SUBJEKTIVNÍHO VIZUÁLNÍHO VNÍMÁNÍ
NA AUTOMATICKÉ HODNOCENÍ VZHLEDU ROZHRANÍ DASHBOARD

DOCTORAL THESIS
DISERTAČNÍ PRÁCE

AUTOR **Ing. JIŘÍ HYNEK**
AUTOR PRÁCE

SUPERVISOR **prof. Ing. TOMÁŠ HRUŠKA, CSc.**
ŠKOLITEL

BRNO 2019

Abstract

Using metrics and quantitative design guidelines to analyze design aspects of user interfaces (UI) seems to be a promising way for the automatic evaluation of the visual quality of user interfaces. Since this approach is not able to replace user testing, it can provide additional information about possible design problems in early design phases and save time and expenses in the future. Analyses of used colors or UI layout are the examples of such evaluation. UI designers can use known pixel-based (e.g., *Colorfulness*) or object-based (e.g., *Balance* or *Symmetry*) metrics which measure chosen UI characteristics, based on the raster or structural representation of UI.

The problem of the metric-based approach is that it does not usually consider users' subjective perception (e.g., subjective perception of color and graphical elements located on a screen). Today's user interfaces (e.g., dashboards) are complex. They consist of several color layers, contain overlapping graphical elements, which might increase ambiguity of users' perception. It might be complicated to select graphical elements for the metric-based analysis of UI, so the selection reflects users' perception and principles of a visual grouping of the perceived shapes (as described by Gestalt psychology). Development of objective metrics and design guidelines usually requires a sufficiently large training set of user interface samples annotated by a sufficient number of users.

This thesis focuses on the automatic evaluation of dashboard design. It combines common knowledge about dashboards with the findings in the field of data visualization, visual perception and user interface evaluation, and explores the idea of the automatic evaluation of dashboard design using the metric-based approach. It analyzes chosen pixel-based and object-based metrics. It gathers the experience of users manually segmenting dashboard screens and uses the knowledge in order to analyze the ability of the object-based metrics to distinguish well-designed dashboards objectively. It establishes a framework for the design and improvement of metrics and proposes an improvement of selected metrics. It designs a new method for segmentation of dashboards into regions which are used as inputs for object-based metrics. Finally, it compares selected metrics with user reviews and asks questions suggesting future research tasks.

Keywords

dashboard, user interface, UX, usability evaluation, metrics, aesthetics, balance, visual perception, subjective perception, Gestalt laws, segmentation

Abstrakt

Analýza vlastností uživatelských rozhraní založená na použití metrik a kvantitativních pravidel grafického designu se zdá být slibným přístupem pro automatické hodnocení vizuální kvality uživatelských rozhraní. Přestože tento přístup nemůže plně nahradit uživatelské testování, může poskytnout dodatečné informace o možných problémech návrhu uživatelských rozhraní v počátečních fázích vývoje a ušetřit tím čas a výdaje v budoucnu. Příkladem je analýza použitých barev a rozvržení grafických elementů na obrazovce. Návrháři uživatelských rozhraní mohou měřit vlastnosti uživatelských rozhraní za použití známých metrik založených na analýze pixelů bitmapy (např. barevnost) nebo grafických elementů (např. vyvážení, symetrie).

Problémem použití metrik nicméně je, že tento přístup zpravidla nezohledňuje subjektivní vnímání uživatelů (např. subjektivní vnímání barev nebo grafických elementů rozmístěných na obrazovce). Dnešní uživatelská rozhraní (jako například rozhraní dashboard) jsou komplexní. Skládají se z několika barevných vrstev, obsahují překrývající se grafické elementy, což může zvyšovat nejednoznačnost vnímání tohoto rozhraní uživateli. Může být proto komplikované vybrat takové grafické elementy, které odpovídají elementům rozpoznaným uživateli v souvislosti s principy shlukování vnímaných tvarů (jak je popsáno Gestalt psychologií). Vývoj objektivních metrik a kvantitativních pravidel grafického designu obvykle vyžaduje dostatečně velkou trénovací množinu vzorků uživatelských rozhraní anotovaných dostatečným počtem uživatelů.

Tato práce se zaobírá automatickým ověřováním vzhledu uživatelských rozhraní dashboard. Práce kombinuje obecné znalosti týkající rozhraní dashboard s poznatky z oblasti vizualizace dat, vizuálního vnímání a ověřování kvality uživatelských rozhraní a následně zkoumá myšlenku automatického hodnocení vzhledu rozhraní dashboard s využitím metrik. Práce analyzuje vybrané metriky založené na analýze pixelů bitmapy a grafických elementů. Konkrétně zkoumá, jakým způsobem uživatelé rozpoznávají grafické elementy v rozhraních dashboard a výsledky aplikuje pro hodnocení schopnosti metrik (analyzujících grafické elementy rozhraní) objektivně rozpoznávat dobře navržené vzorky rozhraní dashboard. Dále představuje framework pro návrh a vylepšení metrik, který využívá pro vylepšení vybraných metrik. Následně navrhuje metodu pro segmentaci rozhraní dashboard do regionů, které mohou být použity jako vstupy pro tyto metriky. Závěrem práce porovnává vybrané metriky s hodnocením rozhraní uživateli a pokládá otázky vhodné pro budoucí výzkum.

Klíčová slova

dashboard, uživatelské rozhraní, UX, testování použitelnosti, metriky, estetičnost, vyvážení, vizuální vnímání, subjektivní vnímání, Gestalt principy, segmentace

Impact of Subjective Visual Perception on Automatic Evaluation of Dashboard Design

Declaration

Hereby I declare that this Ph.D. thesis was prepared as an original author's work under the supervision of prof. Ing. Tomáš Hruška CSc. All the relevant information sources, which were used during preparation of this thesis, are properly cited and included in the list of references.

....
Jiří Hynek
June 24, 2019

Acknowledgements

First of all, I would like to thank to my supervisor prof. Ing. Tomáš Hruška CSc. who kept supervising and consulting the topic of my thesis for all the years of the study. Then, I would like to thank to my colleagues at Brno University of Technology, especially at Department of Information Systems, Faculty of Information Technology. I thank to (in alphabetic order) Ing. Radek Burget Ph.D., doc. RNDr. Jitka Kreslíková, CSc., and RNDr. Libor Škarvada for advice and consultations. I thank to Ing. Olena Pastushenko for cooperation on the selected publications. I thank to all students who helped me improve the software used in the thesis, participated in the studies and experiments, or who somehow cooperated with me. Last, but not least, I would like to thank to my family, especially to my parents without whose support I wouldn't be able to finish the thesis. I also thank to friends and all people who supported me to finish the thesis.

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project 'IT4Innovations excellence in science – LQ1602'.

Reference

HYNEK, Jiří. *Impact of Subjective Visual Perception on Automatic Evaluation of Dashboard Design*. Brno, 2019. Doctoral thesis. Brno University of Technology, Faculty of Information Technology. Supervisor prof. Ing. Tomáš Hruška, CSc.

Contents

Chapter 1

Introduction

Information—“the facts provided or learned about something or someone” (Oxford Dictionary of English [Stevenson, 2010]); “the communication or reception of knowledge or intelligence” (Merriam-Webster’s collegiate dictionary [Merriam-Webster Inc., 2004]. Those are only examples of possible definitions. Information has been the aim of various philosophers during the history [Capurro and Hjørland, 2005]. It has various forms. It can be represented by a logical value deciding a single fact or a sequence of values with a complex meaning. In computer technology, we are talking about *data* and *semantics* of the data [Eckerson, 2006]. Biology and psychology focus on the perception of a signal (e.g., light, or sound) by human sensors and the transformation of the signal to information by the human brain [Gibson, 1950; Marr, 2010]. Then, we can consider information as an answer to a question or a stimulus for making decisions.

Since vision is the dominant human sense, it is important to pay high attention to data visualization, perception, and cognition. There are various possibilities of how to visualize the same data [Harris, 2000]. The goal of data visualization is to provide data in such form (graphical or textual) which helps users to understand the meaning of the data corresponding to the information we want to communicate (Figure 1.1) [Tufte, 2001]. Nowadays, people are surrounded by information technology providing them with increasing amount of data. It is often difficult for them to distinguish the data providing beneficial information (*knowledge*). People work with useless information, which leads to information overload [Yigitbasioglu and Velcu, 2012]. This problem can negatively affect their decisions, or even their health condition (e.g., increasing stress level [Levy, 2008]). Hence, there are tendencies to design visualization tools which would present only important information on a single screen and maximize its comprehensibility. An example of such visualization tool is dashboard.

Dashboard is a frequently used term connected with business intelligence and management information systems. It is a favorite tool used by many organizations to comprehensively present their data for operational, analytical, or strategic purposes. It presents key performance indicators which help to evaluate the progress and benefit of business activities [Eckerson, 2006]. Since dashboards support decision-making, they have become popular among a wide range of users for the management of personal activities. We can find numerous web applications providing dashboard templates to visualize data gathered from common services like social networks. The rising diversity of dashboards has led UI designers and researchers to think about the principles of high-quality dashboard design.

One of the first rules which brought some clarity to dashboard characteristics were provided by Stephen Few [2006]. He has worked with the idea of a single screen display

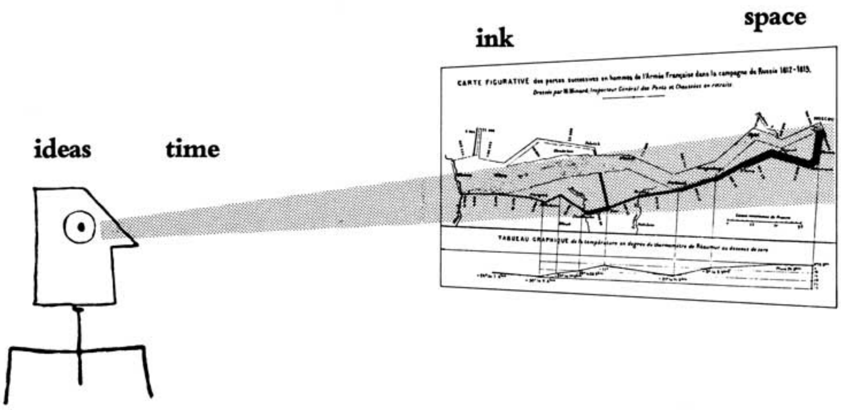


Figure 1.1: Motivational figure presented by Edward R. Tufte in [Tufte, 2001], which describes *graphical excellence* as “that which gives the viewer the greatest number of ideas in the shortest time with the least in the smallest space.” Stephen Few showed that we can work with this definition in the field of information technology. We can consider the ink as pixels and the space as a resolution of the screen [Few, 2006].

comprehensively presenting the most critical information to achieve goals. The requirement of the dashboard—“present information on a single screen”—is what distinguishes dashboards from other interfaces and, also, makes them difficult to design. UI designers need to focus on the design aspects such as strong simplification, elimination of unnecessary elements, highlighting significant relationships between data, or careful selection of graphical elements capable of comprehensively presenting a great deal of data using a small area. Few pointed out that most of the existing so-called dashboards break the requirement. He has provided a framework based on the knowledge of famous books regarding design and graphics (e.g., [Tufte, 2001; Ware, 2004]). This framework contains heuristics for the dashboard design, including examples of well-designed dashboards. Examples of such heuristics are:

- Eliminate the non-data pixels (decorations) to decrease the distraction of users (based on [Tufte, 2001]).
- Consider Gestalt laws to help a user recognize the coherent groups better (based on [Ware, 2004]).
- Select appropriate charts and colors for emphasizing the relationship between data and highlighting the critical information (Figure 1.2).

Even more than ten years after the release of Few’s publication, we can still observe that the majority of dashboards ignore Few’s heuristics or express them in their own way. We assume that the reason might be the complexity and vague definition of the framework and the lack of other sources which would provide formal and quantitative knowledge in the area of dashboard design. For instance, the selection of appropriate charts and colors usually depends on an actual context, and it cannot be completely generalized. A dashboard designer needs to be a person with experience in human-computer interaction and capable

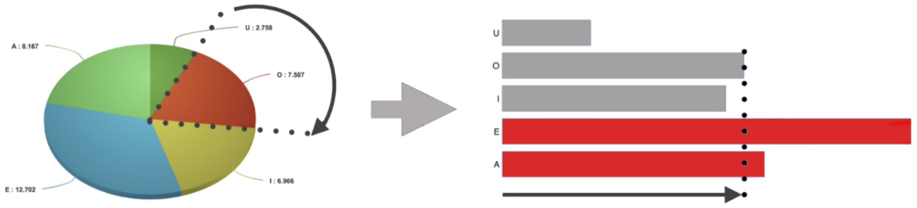


Figure 1.2: An example of a design heuristic. Few [2006] advises to use bar charts with subtle colors instead of pie charts using vivid colors to compare values. Viewers can note the differences between the values better. Vivid colors can be used to emphasize selected values (e.g., the values higher that a limit).

of applying the framework correctly. The presence of users is usually required to evaluate usability, which increases the time and expenses of the design phase.

A challenge in improving UI design and evaluation is that of finding quantitative guidelines which would detect some of the design problems and help to distinguish well-designed interfaces from poorly designed ones. Such guidelines could be applied automatically during the early design phase without the presence of users and specialists in UI design [Ivory and Hearst, 2001]. The simplicity of guidelines is, however, the major weakness of this approach. It is not usually easy to describe complex design attributes of a user interface since they usually depend on the subjective judgment of the viewer. Design guidelines are usually based on simple metrics (e.g., the average colorfulness based on saturation of screen pixels [Reinecke et al., 2013]).

One possible step in making the metric-based evaluation more reliable is to process a screen similarly as it is perceived by the human brain—not as a matrix of pixels but as a group of objects within a scene as described (for example) by Baker et al. [2009]. Then, we evaluate objects on a screen (e.g., controls and widgets) and their properties (e.g., size or position) as described by Charfi et al. [2014]. For this purpose, we use *object-based* metrics. We can measure advanced characteristics of a screen (e.g., the characteristics connected with layouts). For instance, Ngo et al. [2003] have published 13 advanced object-based metrics measuring aesthetic aspects of a screen—e.g., layout balance or symmetry. An example of practical application of Ngo's metrics is the QUESTIM tool designed by Zen and Vanderdonckt [2014]. Users can use it without specialized knowledge of UI design. They manually specify object boundaries according to their visual perception, and the tool calculates the values of Ngo's metrics using dimensions of the regions (Figure 1.3). The values can help them rate the overall quality of a user interface since it has been shown that aesthetics or even the first impression has an impact on usability and acceptability of the product [Tractinsky et al., 2000].

The main weakness of the applicability of object-based metrics is the ambiguous definition of the object. The QUESTIM tool depends on the user's subjective perception of objects. Two users will most likely specify object regions in a slightly different way, which may lead to ambiguous results (Figure 1.4). There were also attempts to extract the description of objects from the structural descriptions of web-pages [Purchase et al., 2011], or images of user interfaces [Reinecke et al., 2013]. The problem with these approaches is that they do not usually consider objects with the same complexity as people perceive them (e.g., the principles of objects grouping described by Gestalt laws [Koffka, 2013]).

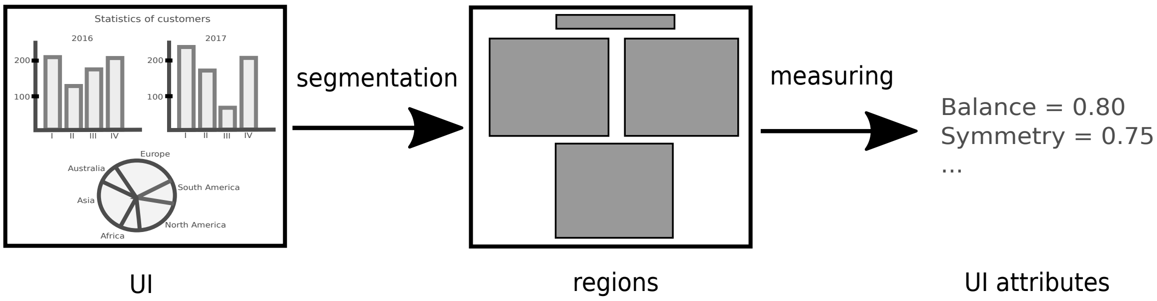


Figure 1.3: In the beginning, we have a screenshot of a user interface. We need to find a suitable segmentation method to specify regions representing visually dominant objects corresponding with the user perception. Then, we can use these regions as the inputs for object-based metrics measuring UI characteristics.

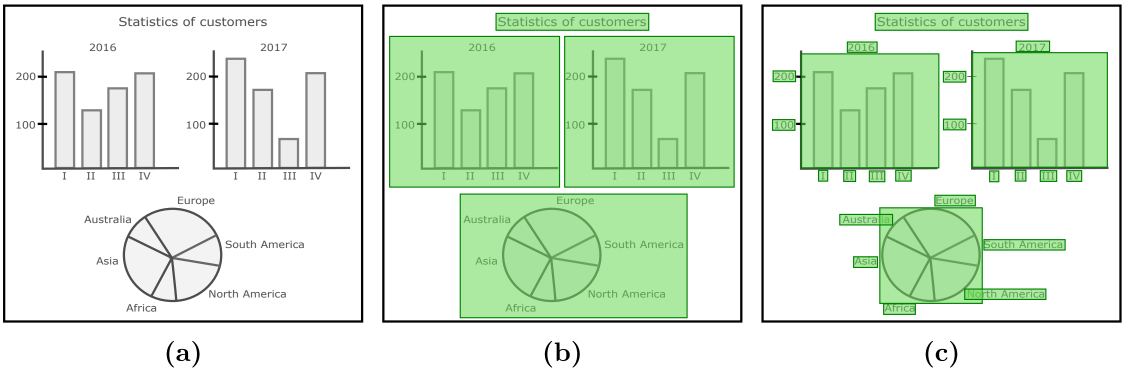


Figure 1.4: An example of the two different ways (b, c) of subjective perception of objects in a dashboard (a). The perceived objects are specified by the rectangular boundaries (regions), which are used as the inputs for object-based metrics (such as Balance or Symmetry).

1.1 Goal of the Research

The goal of this research is to explore the possibility to apply the metric-based evaluation for analysis of dashboard design quality. Specifically, this research focuses on the solution of the following issues:

- Analyze the common characteristics of dashboards. Focus on the perception of objects in dashboards by users, evaluate the subjective visual perception of the users and detect the presence of Gestalt laws.
- Explore existing metrics for analysis of UI attributes and consider their application for measuring quality and usability characteristics of dashboards.
- Focus on object-based metrics of aesthetics and analyze ambiguity of measured results caused by users' subjective perception of objects.
- Create a framework for evaluation of metrics' ability to objectively distinguish well-designed dashboard samples.

- Look for a new approach which would improve the metrics' ability to distinguish well-designed dashboard samples objectively.
- Design a method for segmentation of dashboards into regions which would correspond with the average perception of the users.
- Implement a tool which would provide functionality for loading, segmentation and objective measurement and analysis of chosen dashboard characteristics.

The research analyzes state of the art regarding data visualization, visual perception, and cognition (e.g., objects grouping and Gestalt psychology) and applies the knowledge in the field of metric-based analysis and evaluation of user interface quality. It provides own study of user perception which helps to understand the subjectivity of visual perception and object recognition and grouping in complex user interfaces like dashboards. It extends state of the art in the field of metric-based analysis, evaluation of UI and page segmentation and provides a tool for automatic analysis of dashboards and single screen UIs. The research works with static images of user interfaces, focusing strictly on the presentation aspect of UI. It does not consider the interaction of users with the analyzed user interface.

1.2 Document Structure

The thesis is structured into 12 chapters. Firstly, the chapters 2 - 4 discuss state of the art.

- **Chapter 2** (*Dashboard and Data Visualization*) introduces the dashboard visualization tool. It presents existing definitions of the dashboard term and discusses characteristics, classification, strengths, and weaknesses of dashboards. It focuses on design aspects of dashboards and introduces popular widgets used in dashboards. Finally, it presents Few's framework for dashboard design. It discusses frequently occurred design problems in dashboards and points out that dashboard design quality could be evaluated automatically during the design phase in order to decrease time and costs of the evaluation.
- **Chapter 3** (*Evaluation of User Interfaces*) provides state of the art regarding evaluation of user interfaces. It provides basic terminology and categorization of existing methods with examples. It compares methods according to different factors and analyzes their advantages and disadvantages. Then, it focuses on the possibility of automation of the evaluation process. It introduces the evaluation based on design heuristics and guidelines and presents existing quantitative guidelines based on metrics for measuring usability characteristics of user interfaces. It analyzes object-based metrics evaluating characteristics of UI objects (e.g., layout, aesthetics) and considers their application for evaluation of dashboard visual quality. Finally, it points out the problem of ambiguous recognition of objects within a user interface which represent inputs for object-based metrics.
- **Chapter 4** (*Recognition of Visual Components*) focuses on the recognition of visually emphasized objects within a scene. The first part of the chapter discusses the process of human visual perception of objects. It uses the knowledge of Gestalt psychology to describe known principles regarding the objects recognition and grouping. It emphasizes the importance of preattentive processing, visual attention, short-term memory

and subjective perception. The second part of the chapter focuses on automatic recognition of objects by a computer. It presents existing segmentation approaches which are usually used for segmentation of scanned documents and consider their application in segmentation of dashboard screenshots. Finally, it points out the complexity of visual perception and difficulty to simulate it by a computer.

Then, the chapters 5 - 10 describe the research work and presents its results.

- **Chapter 5** (*Decomposition of Problem*) detects the main problems of the research. It defines the process of the research and divides the research into tasks which provide solutions to the problems. Then, it establishes a model of internal representation of dashboard which can be processed by metrics of UI quality. It respects the categorization of metrics presented in Chapter 3, which recognizes pixel-based and object-based metrics, and provides the pixel-based and object-based internal representation. Then, it introduces software which works with the internal representation of dashboards. Finally, it presents test samples which are used to evaluate the results of the research tasks.

- **Chapter 6** (*Analysis of Pixel-based Metrics*) analyzes the metrics which work with the pixel-based representation of dashboards. It analyzes the three groups of metrics measuring: colorfulness, number and share of used colors, and distribution of colors (balance, symmetry). It considers the problem of image compression.

- **Chapter 7** (*Analysis of Object-based Metrics*) analyzes the metrics which work with the object-based representation of dashboards. In the beginning, it performs a study of visual perception of objects in dashboards in order to get a dataset describing ambiguity of user perception. It lets 251 users specify regions of the dashboards' objects according to their subjective perception. Then, it establishes a framework for processing the subjective descriptions of regions and measuring the ability of object-based metrics to distinguish a specific kind of UI samples (e.g., well-designed dashboards) objectively. Finally, it uses the framework to analyze 13 Ngo's metrics of aesthetics. It emphasizes the impact of the subjective perception of regions on the application of the metrics.

- **Chapter 8** (*Design and Improvement of Metrics*) proposes a framework for design and iterative improvement of metrics. It uses the framework to improve selected Ngo's object-based metrics (the metrics analyzing weights of regions based on the size and distribution of regions on a screen). It considers the combination of their object-based approach with the pixel-based approach analyzing colorfulness of regions. Firstly, it performs a small-scale study to evaluate the hypothesis considering the impact of color on weights of regions, which are analyzed by selected Ngo's object-based metrics. Then, it uses the knowledge gained from the results of the study and proposes a modification of selected metrics. It demonstrates the idea of the modification of the Balance metric. Finally, it analyzes the impact of the Balance improvement on the ability of the metric to distinguish well-designed dashboards objectively.

- **Chapter 9** (*Automatic Segmentation of Dashboards*) uses the knowledge of the study of visual perception of objects and design a new method for automatic segmentation of dashboards into regions. The method is divided into several phases, which preprocess the bitmap, detect UI primitives, construct the layout of the UI primitives, and search the dominant regions in the layout. It combines the top-down layout analysis to simulate the Gestalt law of enclosure with the bottom-up layout analysis to simulate the Gestalt law of proximity. Finally, the chapter evaluates the method with the regions specified by users and analyze the possibility to use the synthetic regions as the inputs for object-based metrics.

- **Chapter 10** (*Comparison of Metrics with User Reviews*) compares the values measured by the metrics with the perception of UI characteristics by users. It shows high ambiguity of user ratings, which might be caused either by the subjective perception of the users and the subjective quantification of their perception or by a different understanding of the principles of UI characteristics. In the end, the chapter presents several questions which might lead to further research tasks.

Finally, the chapters 11 and 12 evaluate and summarize the thesis.

- **Chapter 11** (*Discussion*) presents the summarized list of all results. It discusses their possible application, points out their limitations, and suggests further research tasks which can be done in the future to extend the results.
- **Chapter 12** (*Conclusions*) provides the overall summary of the thesis.

Some parts of the thesis describe the results which were made with the cooperation of students of Brno University of Technology. The students implemented software used for the purposes of this research as a part of their bachelor's or master's thesis supervised or consulted with the author of this thesis. Specifically:

- Subsection 5.3.2 describes Generator of Dashboard Samples developed as a part of the master's thesis of Olena Pastushenko [Pastushenko, 2017]. The generator was applied to generate dashboard samples used for the study described in Section 8.2. The study was performed with the cooperation of Olena Pastushenko as well and published by Pastushenko, Hynek and Hruška [2018, 2019].
- Subsection 5.3.1 describes the Dashboard Analyzer software (developed by the author of this thesis), which uses extensions created as parts of the bachelor's thesis of Adriana Jelenčíková [Jelenčíková, 2018] and the master's thesis of Santiago Mejía [Mejía, 2018]. Mejía's findings were used in the bottom-up analysis of the method for segmentation of dashboards, described in Subsection 9.2.7.
- Subsection 5.2.2 mentions the generators of charts developed as parts of the bachelor's theses of Filip Barič [Barič, 2017] and Natalya Loginova [Loginova, 2017]. The tools were not directly used in this research but they use the same (or similar) object-based model (designed by the author of this thesis) which describes the internal representation of dashboards used in this research.

Chapter 2

Dashboard and Data Visualization

The idea to present all information together on one scene (e.g., single document or screen) was applied in many cases in the history. Designers were aware of the disadvantage of textual representations. They tried to find optimal graphical representations which would squeeze all relevant data into a single scene so the viewer could realize important connection between data quickly. The following two examples presented in Tufte's book [Tufte, 2001] demonstrate major advantages and disadvantages of the approach.

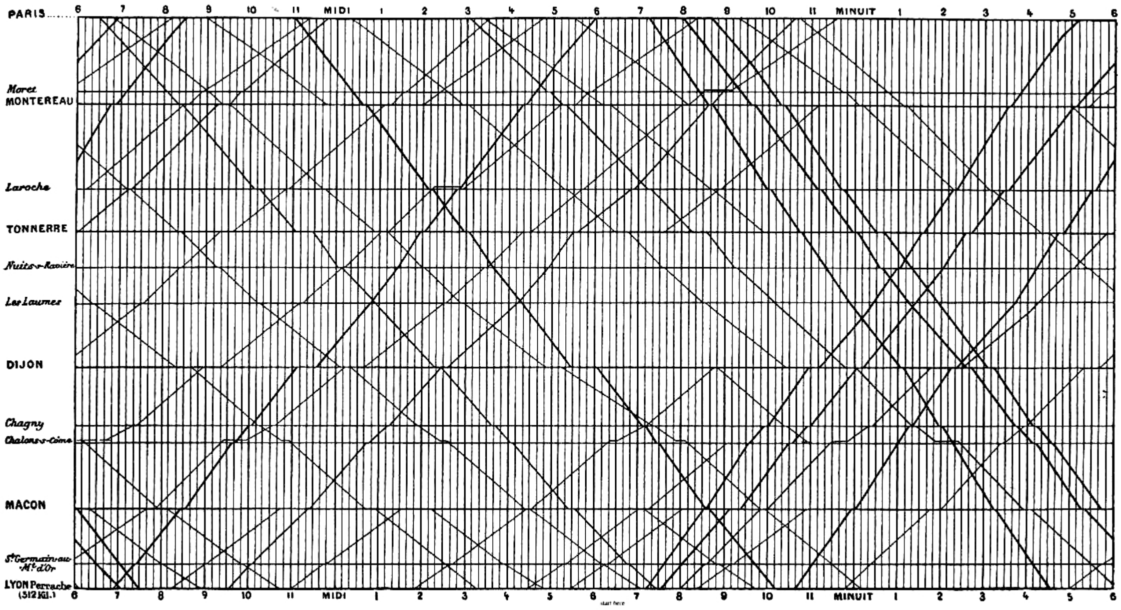


Figure 2.1: Marey's graphical train schedule. Used from [Tufte, 2001]. Original source: E. J. Marey, La Méthode Graphique [Marey, 1885].

The first example of a single-page visualization shown in Figure 2.1 is Étienne-Jules Marey's graphical train schedule. The graphics created in 1885 presents a schedule of all trains between Paris and Lyon. The vertical axis contains the train stations arranged according to their locations on the track, and the horizontal one represents the time of arrivals and departures. Then, the viewers can see the particular trains in the form of lines connecting the stations. They can compare the speed of the trains by examining of slopes of the lines. Also, they can find intersections of the lines representing locations and times of passings of two trains running in the opposite direction. Looking for the same information in a classical booklet train schedule would most likely take the viewers significantly higher time. The diagram is highly intuitive. Users can quickly understand the meaning of the data. Hence, this kind of visualization has been used for scheduling trains to this day.

The second example shown in Figure 2.2 is popular information graphics designed by Charles Joseph Minard in 1869. It presents the invasion of Napeoleon's army into Russia in 1812. We can monitor the size of the army due to the geographical location. In the beginning, we can see that the width of the colored zone represents 422 000 soldiers marching from the west towards Moscow. The size of the army is decreasing with the approaching to Moscow. Finally, we can see the remaining 100 000 soldiers returning back represented by the black zone. We can also notice that the chart at the bottom describes cold temperatures during the return. Minard's graphics, similar to the graphical train schedule, presents a great deal of data comprehensively on a single page. On the other hand, this example is less intuitive as the first one. The viewer needs to be familiarized with the principle of the graphics. This might be the reason the graphics contain the legend.

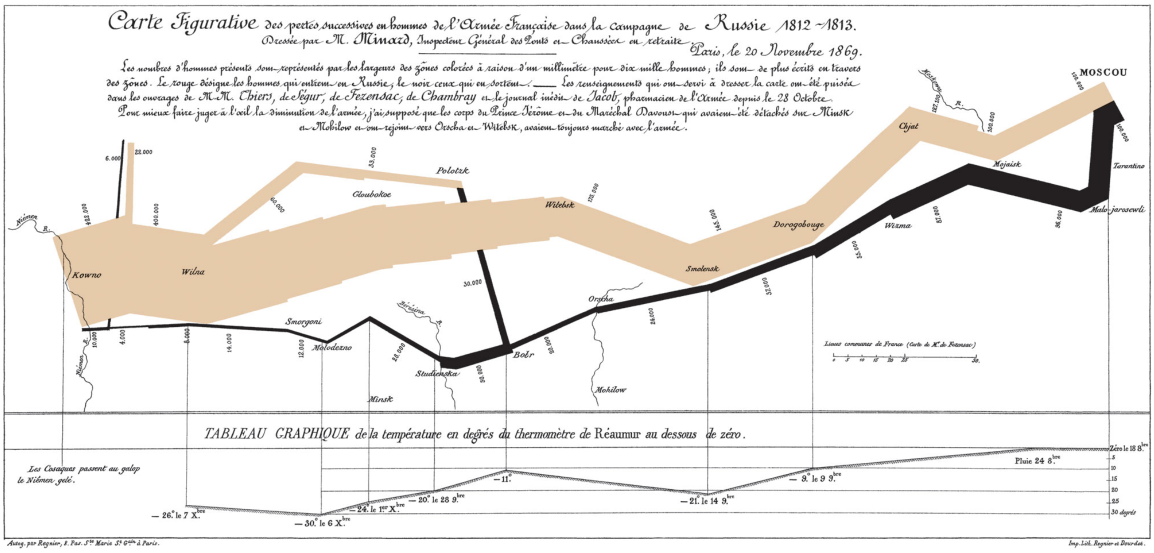


Figure 2.2: Minard's graphics of Napoleon's Russian campaign. Used from [Tufte, 2001]. Original source: E. J. Marey, La Méthode Graphique [Marey, 1885].

As shown in the two examples, data presented on a single scene might emphasize information which would be difficult to find if the viewer had to go through several scenes and connect the perceived information explicitly. On the other hand, it is not usually simple to fit all the important data in a limited space. UI designers need to find new kinds of graphics which are capable of projecting the data comprehensively. Their originality might contradict to quick comprehension of the visualized data by the viewers. Hence, UI designers should find a compromise between the amount of presented data and the way how the data is presented. A dashboard is an example of such compromise. It graphically visualizes important data related with the particular goals on a single screen (usually webpage) using well-known charts (e.g., bar charts or line charts).

2.1 Dashboard Definition

The original meaning of the word *dashboard* was used for "a screen on the front of a usually horse-drawn vehicle to intercept water, mud, or snow" (Merriam-Webster's collegiate dictionary [Merriam-Webster Inc., 2004]). Then, with the evolution of vehicles, dashboard has become an advanced and sophisticated panel containing instruments and controls crucial for driving today's motor vehicles. It provides actual information about the driving, which helps the driver to adjust the driving according to current conditions. Its main characteristic is intuitiveness. Drivers can use it without special attention, and they can focus on the driving, which is the primary task.

The philosophy of the dashboards used in vehicles is similar to the meaning of the dashboards used in information technology. It usually represents a single screen which provides important information about some state regarding specific tasks (the driving task is generalized). Users use it as a tool which helps them analyze the current situation and make appropriate decisions to fulfill specific goals. A well-designed dashboard should, similarly to dashboards used in a car, provide the critical information of the task and advice the user to perform appropriate actions without the need of special examination of the dashboard. For instance, dashboards are favorite tools of business intelligence. Companies use them to monitor and analyze critical information regarding the current state of their business (*key performance indicators*) [Eckerson, 2006].

There are several definitions of the *dashboard* term used in information technology. Oxford Dictionary of English [Stevenson, 2010] defines it as "a graphical summary of various pieces of important information, typically used to give an overview of a business". Malik [2005] defines a dashboard as "a rich computer interface with charts, reports, visual indicators, and alert mechanisms that are consolidated into a dynamic and relevant information platform." Wexler et al. [2017] define it simply as "a visual display of data used to monitor conditions and/or facilitate understanding." Looking for existing dashboards reveals us various dashboard examples, usually implemented in the form of a webpage. Dashboards became popular tools used to provide navigation and summarized overview of websites. Sometimes it might be difficult to distinguish between a dashboard and a regular webpage containing some analytical and statistical data.

Stephen Few [2006] examined existing dashboards and looked for common characteristics. He established a more strict definition of information dashboard:

Definition 2.1: A dashboard is a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance.

Most of the existing so-called dashboards break Few's definition. Existing dashboards often exceed the boundaries of one screen. They contain scrollbars, and users usually need to switch between several views. The information presented in dashboards is not usually related to specific goals. As Few pointed out, business intelligence vendors use dashboards more like a marketing tool which helps them to sell the product than a tool which should actually help users to monitor and analyze data effectively.

The following text considers the dashboard term as a visualization tool used in information technology with respect to Definition 2.1. This chapter provides a brief overview of dashboard characteristics, classification, graphical components, and design. It is based on Few's book [Few, 2006], which contains coherent knowledge regarding dashboards and dashboard design, important for this research. It corresponds the knowledge with additional books describing real-world dashboard examples [Wexler et al., 2017], dealing with the processing of data for dashboards [Jacobs and Rudis, 2014] and providing information about dashboards used in business intelligence [Eckerson, 2006; Malik, 2005; Rasmussen et al., 2009]. It also uses the knowledge regarding data visualization [Harris, 2000; Tufte, 2001; Ware, 2004]. Besides that, there are also books focusing on implementation of dashboards in commercial or free software and tools —e.g. Tableau [Stirrup, 2016], Microsoft Excel [Alexander and Walkenbach, 2010], or R Shiny [Beeley, 2018]. Readers can find additional information there.

2.2 Dashboard Characteristics

We can find various lists describing dashboard characteristics, depending on the used definition. For instance, [Malik, 2005] who focuses on enterprise dashboards, describes a dashboard in the list of attributes supporting effective organizational management. [Few, 2006] focuses more on general design characteristics of user interfaces based on Definition 2.1. Following list bases on Few's characteristics:

1. A dashboard is a visual display which prefers the graphical visualization of data over the textual presentation. The graphical presentation of data provides the overall view of the data in contrast to textual representation, which emphasizes singles values. Dashboard makes the viewer see values in a context, which helps the viewer understand the meaning of the data better. The graphical visualization can emphasize important relationships between data (as demonstrated in the graphical train schedule in Figure 2.1). Finally, the graphical presentation can hold more data than the textual representation, which is important dashboard's requirement. An example of the comparison of the textual and graphical representation is shown in Figure 2.3.

x	y
1	0.606
2	1.248
3	3.276
4	7.627
5	8.925

x	y
6	7.985
7	5.549
8	3.276
9	1.595
10	0.457

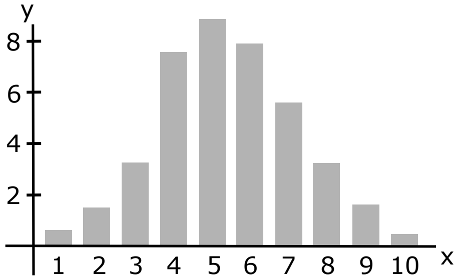
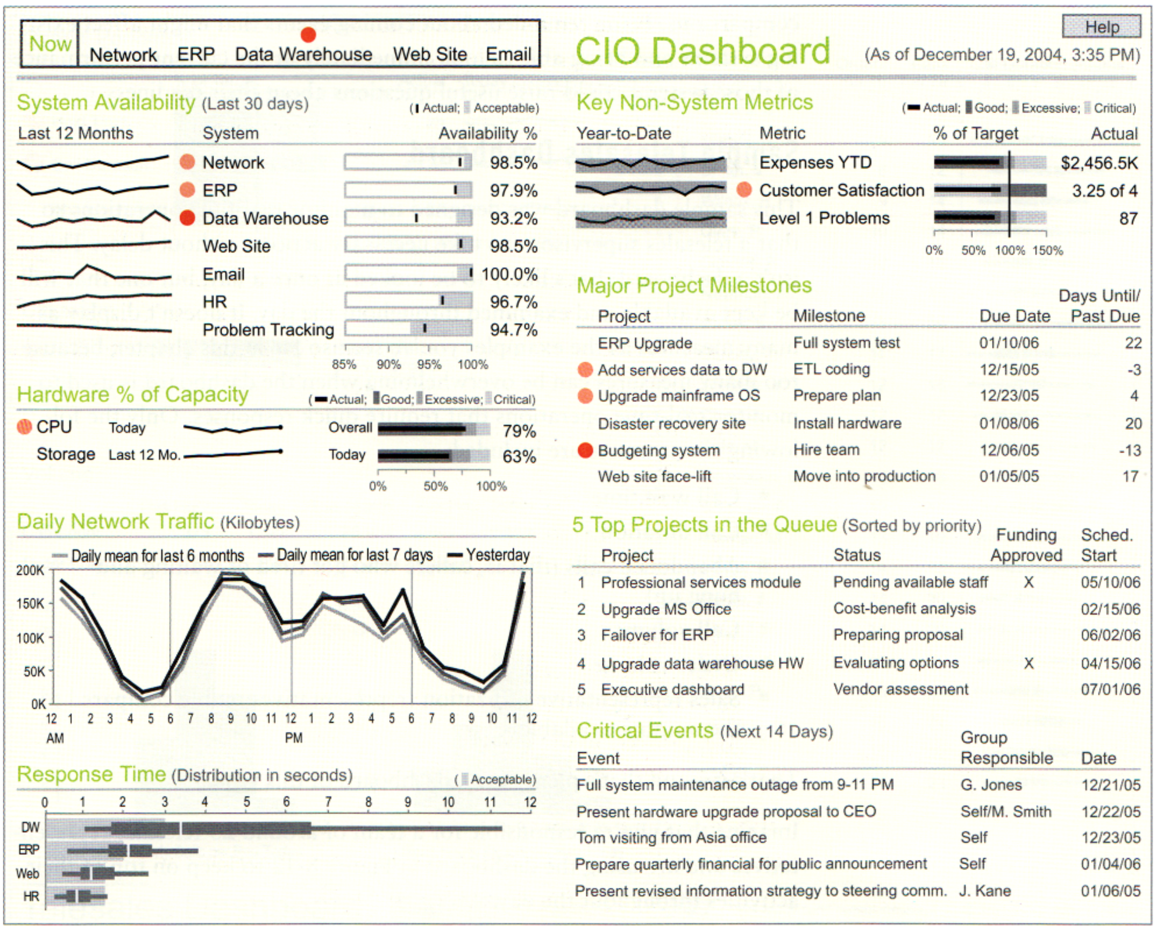


Figure 2.3: The same data are presented textually (left) and graphically (right). Relations between the values are more evident in the graphical presentation than the textual one.

2. A dashboard presents only the information which is important for achieving selected goals. UI designers should carefully analyze requirements of users and select the important data concerning these requirements. UI designers also need to consider that a viewer is able to focus on and process a limited amount of data at once. Hence, UI designers should find a suitable projection of the selected data. They should reflect the knowledge and experience of the users and provide the information in such form which is understandable for them.

3. A dashboard should fit a single screen. As shown in the initial motivation of this chapter, people can find important connections in data better if they can perceive all the data at once. Therefore, UI designers should avoid using widget controls which force the users to go through hidden data (e.g., scrollbars). On the other hand, the users should not be confused by a crammed screen. Graphical elements should be sufficiently large, and labels should be readable. Hence, UI designers should use a suitable layout to arrange widgets on a screen. They need to make a compromise between the amount of presented data and comprehensibility of a dashboard.

4. A dashboard should be intuitive so that the user could perceive the information at a glance. Similarly to dashboards used in a car, the user should be able to quickly find the data which is important for the current situation without special examination of the dashboard, understand its meaning and perform appropriate actions. Dashboards should emphasize the data that matters and deserve to be spotted at the current time.



Last 12 Months	System	Availability %
	● Network	98.5%
	● ERP	97.9%
	● Data Warehouse	93.2%
	Web Site	98.5%
	Email	100.0%
	HR	96.7%
	Problem Tracking	94.7%

● CPU	Today		Overall	79%
Storage	Last 12 Mo.		Today	63%

Year-to-Date	Metric	% of Target	Actual
	Expenses YTD		$2,456.5K
	● Customer Satisfaction		3.25 of 4
	Level 1 Problems		87

Major Project Milestones

Project	Milestone	Due Date	Days Until/ Past Due
ERP Upgrade	Full system test	01/10/06	22
● Add services data to DW	ETL coding	12/15/05	-3
● Upgrade mainframe OS	Prepare plan	12/23/05	4
Disaster recovery site	Install hardware	01/08/06	20
● Budgeting system	Hire team	12/06/05	-13
Web site face-lift	Move into production	01/05/05	17

5 Top Projects in the Queue (Sorted by priority)

	Project	Status	Funding Approved	Sched. Start
1	Professional services module	Pending available staff	X	05/10/06
2	Upgrade MS Office	Cost-benefit analysis		02/15/06
3	Failover for ERP	Preparing proposal		06/02/06
4	Upgrade data warehouse HW	Evaluating options	X	04/15/06
5	Executive dashboard	Vendor assessment		07/01/06

Critical Events (Next 14 Days)

Event	Group Responsible	Date
Full system maintenance outage from 9-11 PM	G. Jones	12/21/05
Present hardware upgrade proposal to CEO	Self/M. Smith	12/22/05
Tom visiting from Asia office	Self	12/23/05
Prepare quarterly financial for public announcement	Self	01/04/06
Present revised information strategy to steering comm.	J. Kane	01/06/05

Figure 2.4: Few's CIO dashboard sample [Few, 2006]. Few reduces non-data pixels and tries to provide important contextual information, well-arranged on one screen. Red alerts inform users about real-time (the top-left table) or long-term problems and lead the users to other screens displaying the reasons of the problems.

The four characteristics correspond to each other. For instance, the ability to arrange all data on one screen depends on the appropriate selection of the data and graphical presentation of the data. It is the same for the ability to make a dashboard intuitive. The problem is that it is usually not simple to meet all the design requirements. UI designers often need to make a compromise between them and correspond them with requirements of the users. Figure 2.4 presents an example of a dashboard which Few [2006] considers as well-designed concerning the visual design and meeting of user's requirements.

2.3 Classification of Dashboard

The ability to describe and classify a user interface and correctly define its purpose is crucial for quality design. Few [2006] presents several variables according to which we can classify dashboards. This section focuses on the three variables—*role*, *type of data* and *interactivity*—since they are the most important variables for this research. Some of the remaining variables (e.g., *update frequency*) are mentioned together with the three main variables.

2.3.1 Role of Dashboard

The categorization of dashboards based on a dashboard role is one of the most frequently used. It corresponds with a kind of business activity which the dashboard supports. Few [2006], Eckerson [2006] and Rasmussen et al. [2009] distinguish the three kinds of dashboards—*operational*, *analytical/tactical* and *strategic* dashboards:

- **Operational dashboards** provide monitoring of particular activities, usually to front-line workers. Their goal is to provide low-level data and notify the users about the situations which require some response. Data are usually updated in the real-time so the response can be performed quickly. Presentation of data is simple (text values, alert icons), so the users can quickly perceive and understand the information. We can find such operational dashboards for the monitoring of a stock market or traffic.

- **Analytical dashboards** (also called tactical) provide aggregated data and help to analyze the data. They are usually used for managers and analysts. On the contrary to operational dashboards, they work with a static snapshot of data. Hence, they use the advantage of graphical presentation to provide a better look at the context of the data. They allow users to interact with displayed media and perform operations with multidimensional data typical for OLAP analysis (e.g., *drill-down* [Codd et al., 1993; Wrembel and Koncilia, 2006]). Analysts usually use the dashboards to compare the values or look closer for the reasons of a current state.

- **Strategic dashboards** provide the most summarized view containing high-level data, usually to executives. They use the dashboards to analyze the progress of strategy fulfillment. Similarly to analytical dashboards, they contain static snapshots of long-term data. They, however, focus more on the performance and prediction of the future. Hence, the dashboards usually use the graphical presentation of data, but usually without the ability to provide advanced operations with the data. It is usually not required since the users usually focus more on the future than a current state.

2.3.2 Type of Data

Dashboard can present *quantitative* and *non-quantitative* data. Quantitative data is a measurable value (e.g., revenues, expenses or profits). It is common data kind in dashboards since we can graphically visualize it using various charts and users can compare the values. On the other hand, non-quantitative data represents information which cannot be expressed numerically. An example of such data is "the five most selling products." We can present this information quantitatively as well (e.g. bar chart containing the numbers of sold products). However, a viewer might find more useful to see the information directly than derive the information from values explicitly.

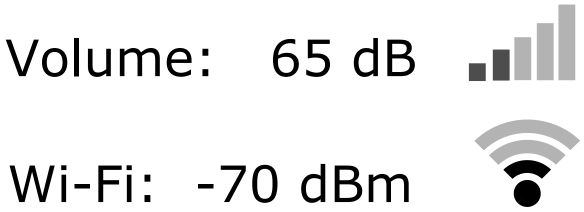


Figure 2.5: It is easier to understand the qualitative icons than the quantitative values.

Few [2006] recommends converting the quantitative data which needs to be perceived frequently into qualitative visual objects. Then, these objects describe a characteristic of the values (e.g., good or bad, high or low, rising or decreasing). They are usually presented by simple graphic icons, usually with some color combination (e.g., red ∼ bad, green ∼ good). It is easier for the viewer to distinguish a predefined palette of icons than perceive numerical values and connect the values with the meaning explicitly. Figure 2.5 presents a combination of quantitative and qualitative data.

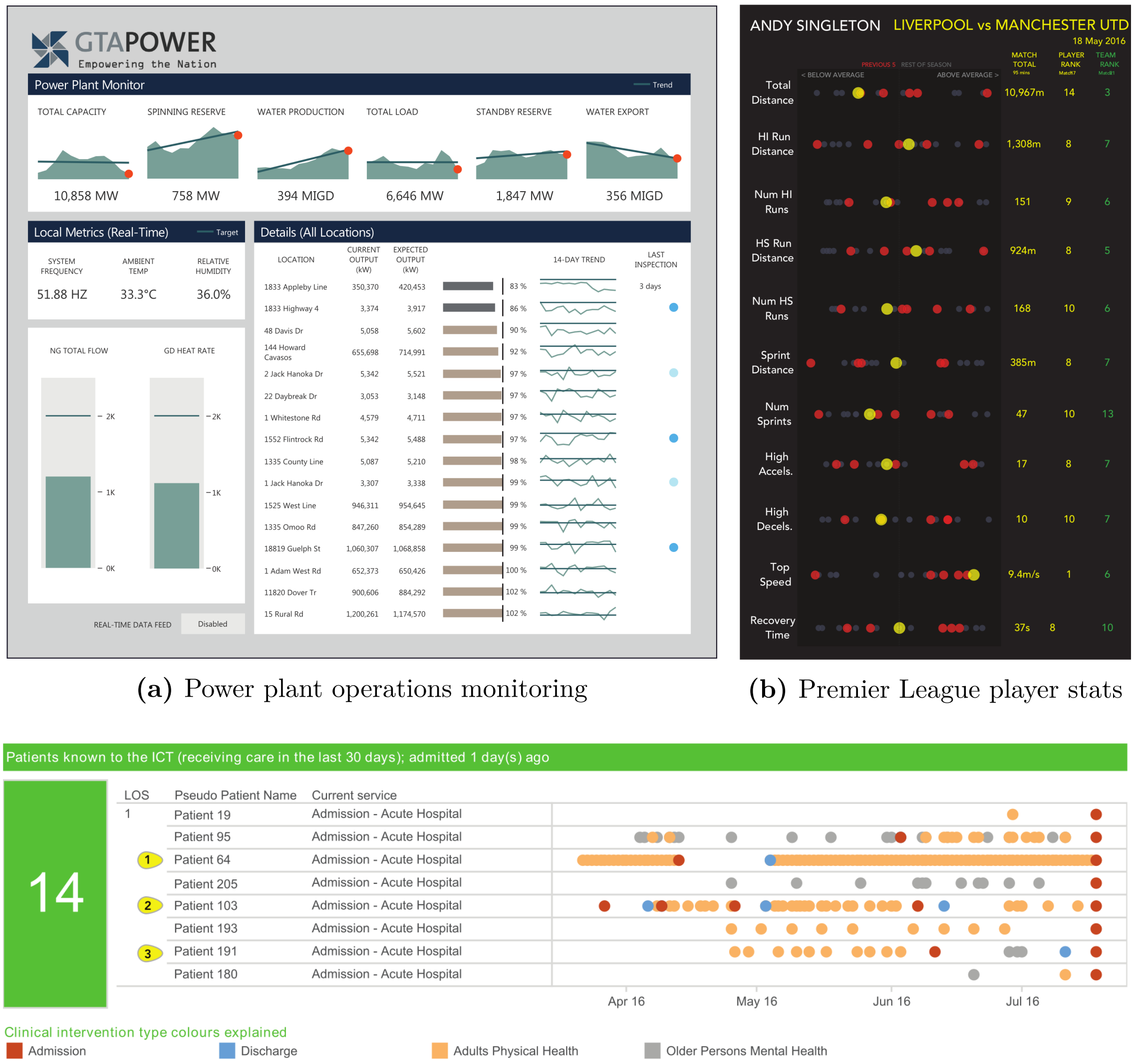
2.3.3 Interactivity of Dashboard

We can design dashboards as static displays, or we can allow users to interact with a user interface. As described in Subsection 2.3.1, analytical dashboards usually provides functionality for analysis of data. Operational dashboards allow users to react to an unexpected situation. On the other hand, strategic dashboards focus more on the presentation of data than interaction with the data.

Today's advanced chart libraries usually provide widgets which allow users basic interaction (e.g., the mouse hover action or zoom). Dashboards often contain many of such widgets. UI designers should consider it in the evaluation of the overall usability of dashboards. This research focuses on the presentation aspect of dashboards. It works with raster images representing screenshots of dashboards. Usability of interactive widgets is not considered.

2.4 Application of Dashboards

The purpose of a dashboard is a special variable which we can use to distinguish dashboards. In contrast to the variables explained in Section 2.3, there is not a finite list of dashboard categories. Every dashboard is unique. It has unique users with unique requirements which deserve special care of UI designers. Wexler et al. [2017] provide a list of very different scenarios of dashboard application. For instance, dashboards can be used to monitor power plant operations, analyze a patients' history of recent hospital admissions or watch the performance of Premier League players (Figure 2.6). On the contrary, Malik [2005] focuses only on enterprise dashboards used for improving processes and data analysis in organizations. He provides a comprehensive categorization of enterprise dashboards which is presented in Figure 2.7.



LOCATION	CURRENT OUTPUT (kW)	EXPECTED OUTPUT (kW)		LAST INSPECTION
1833 Appleby Line	350,370	420,453	83 %	3 days
1833 Highway 4	3,374	3,917	86 %	
48 Davis Dr	5,058	5,602	90 %	
144 Howard Cavasos	655,698	714,991	92 %	
2 Jack Hanoka Dr	5,342	5,521	97 %	
22 Daybreak Dr	3,053	3,148	97 %	
1 Whitestone Rd	4,579	4,711	97 %	
1552 Flintrock Rd	5,342	5,488	97 %	
1335 County Line	5,087	5,210	98 %	
1 Jack Hanoka Dr	3,307	3,338	99 %	
1525 West Line	946,311	954,645	99 %	
1335 Omoo Rd	847,260	854,289	99 %	
18819 Guelph St	1,060,307	1,068,858	99 %	
1 Adam West Rd	652,373	650,426	100 %	
11820 Dover Tr	900,606	884,292	102 %	
15 Rural Rd	1,200,261	1,174,570	102 %	

(a) Power plant operations monitoring

ANDY SINGLETON LIVERPOOL vs MANCHESTER UTD
18 May 2016

	MATCH TOTAL 95 mins	PLAYER RANK Match17	TEAM RANK Match21
Total Distance	10,967m	14	3
HI Run Distance	1,308m	8	7
Num HI Runs	151	9	6
HS Run Distance	924m	8	5
Num HS Runs	168	10	6
Sprint Distance	385m	8	7
Num Sprints	47	10	13
High Accels.	17	8	7
High Decels.	10	10	7
Top Speed	9.4m/s	1	6
Recovery Time	37s	8	10

(b) Premier League player stats

Patients known to the ICT (receiving care in the last 30 days); admitted 1 day(s) ago

LOS	Pseudo Patient Name	Current service
1	Patient 19	Admission - Acute Hospital
	Patient 95	Admission - Acute Hospital
1	Patient 64	Admission - Acute Hospital
	Patient 205	Admission - Acute Hospital
2	Patient 103	Admission - Acute Hospital
	Patient 193	Admission - Acute Hospital
3	Patient 191	Admission - Acute Hospital
	Patient 180	Admission - Acute Hospital

Clinical intervention type colours explained
🟥 Admission 🟦 Discharge 🟧 Adults Physical Health ⬜ Older Persons Mental Health

(c) Patient history analysis of recent hospital admissions (a part of dashboard)

Figure 2.6: Examples of different dashboard applications (source: [Wexler et al., 2017]).

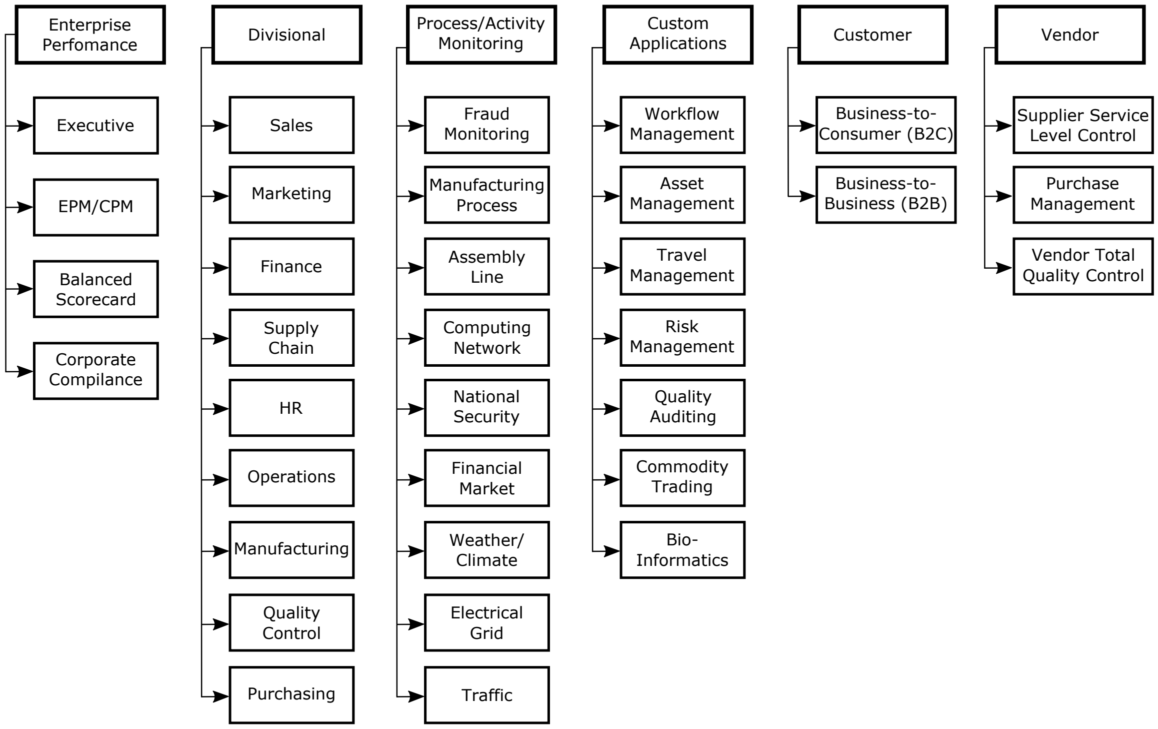


Figure 2.7: The categorization of enterprise dashboards according to Malik [2005].

2.4.1 Dashboards in Business Intelligence

Using dashboards for business purposes is one of the most common applications of dashboards. Companies need to work with data better to survive in today's business world. They need to store, process, monitor, analyze and use a great deal of data about the business environment, competitors, customers, and performance of a company. Continual increasing of the amount of data makes this tasks more and more difficult.

In the 1980s, organizations were starting to use Executive Information Systems as a special group of Decision Support Systems used to improve decision making of senior-level managers. These systems however quickly lost their popularity due to their narrow focus and insufficient support for data manipulation (extraction, transformation, and loading) [Eckerson, 2006; Rasmussen et al., 2009]. In 1989, Howard Dresner, a research analyst, popularized the *business intelligence* term. He described it as "a set of concepts and methods to improve business decision making by using fact-based support systems" [Rasmussen et al., 2009]. Business intelligence gained popularity in the 1990s. Its main advantage was that it helped all users, not only the senior-level managers, to access important information in order to improve their decision making. It increased coordination between departments [Eckerson, 2006].

Eckerson [2006] describes business intelligence as a set of "the processes, tools, and technologies required to turn data into information and information into knowledge and plans that drive effective business activity." He compares the business intelligence term to a "data refinery", which captures data from operational systems and process the data in steps until the users can use it to perform business actions. Finally, the actions of the users generate new data, which needs to be processed. Figure 2.8 presents the process of data transformation.

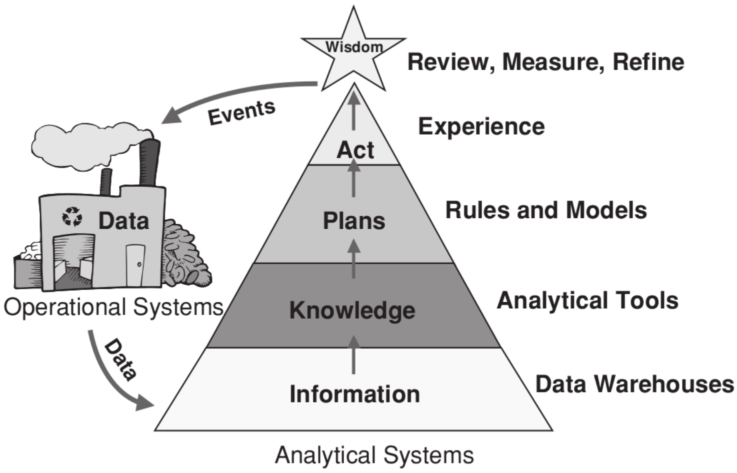


Figure 2.8: Data rafinery—Eckerson's description of business intelligence [Eckerson, 2006]. In the beginning, data warehouses capture and process raw data provided by operational systems and transform it into information. Then, users use analytical tools to find useful information—knowledge, which can be used to create rules. The rules are implemented in the form of plans, which consist of actions. Execution of actions generates business events, which are captured by operational systems. Every cycle of the process helps the organization to improve rules, plans and actions.

A simplified architecture of business intelligence consists of the three parts:

1. **Operational systems** contain operational data, which can be represented in various formats (e.g., files, or relational databases) and stored in various places. It is usually slow and difficult to perform complex queries in order to get knowledge important for a business strategy.

2. **Data warehouses** and **data marts** represent the middle layer (usually implemented as a relational database), which stores data in the format which is suitable for difficult queries, complex data analysis, and visualization. To get such data, we need to specify an appropriate data model (e.g., the multidimensional star scheme). Then, we need to extract, transform and load the operational data (ETL processes [Wrembel and Koncilia, 2006]). It is a difficult process. According to Eckerson [2006], it takes from 60 to 80 percent of the technical team's time.

3. The **presentational layer** is the third layer, which provides an interface between the users and the data stored in data warehouses or data marts. It provides the tools which allow the users to query the data without knowledge of query languages (e.g., SQL). It should not overload the users with a great deal of data, but it should help the users to find the important (usually aggregated) data. Then, in case of need, the users should be able to browse the data.

Performance dashboard is one of the popular business intelligence tools. It is "a multilayered application built on business intelligence and data integration infrastructure that enables organizations to measure, monitor, and manage business performance more effectively" [Eckerson, 2006]. It consists of the three layers: the summarized graphical view, multidimensional view, and detailed reporting view. It provides advanced visualization techniques to present data for strategic, analytical an operational purposes. The data are presented via measurements—*Key Performance Indicators* (KPI) evaluating the performance of the organization. They help to monitor whether an organization works efficiently and also effectively. While efficiency monitors results and compares them with costs (maximum result with minimum effort, resources and time), effectiveness focuses on the meaning of the results (the result matters and meets a strategy). Malik [2005] and Eckerson [2006] present detailed knowledge about specification and examples of KPIs.

As this section suggested, dashboards are not only visualization tools. They are complex systems composed of databases and tools for data processing and analysis. The design of high-quality dashboards does not depend only on the presentation layer but also on the selection of meaningful data, design of an appropriate data model, quick and accessible database or tools for complex data analysis. Trying to create a perfect presentation layer is meaningless without a high-quality back-end. This research focuses on the presentation layer, and usually, it uses the dashboard term in the meaning of the presentation layer. However, the readers should keep in mind the whole context of the problem.

2.5 Components of Dashboard Screen

Dashboard screens consist of display media presenting data. There are many kinds of display media. Readers can find comprehensive descriptions and references in specialized books [Harris, 2000] or surveys [Purchase, 2014]. Display media are more or less suitable for the dashboard requirements. Few [2006] recognizes the six kinds of display media: *text*, *organizers*, *graphs*, *icons*, *drawing objects* and *images*.

The simplest way how to present some data is **text**. Dashboards are typical for its preference to the graphical representation of data over the textual one. However, there are cases which require presenting data in the form of text. Firstly, text is used to present non-quantitative data which we cannot express numerically (Subsection 2.3.2). Secondly, UI designers use text to emphasize particular values. Text is usually presented in the form of single *labels* (usually as part of charts), or it is often organized in **organizers** (*lists*, *tables*, or *trees*).

A **graph** (also called *chart* or *plot*) is the major graphical presentation used in dashboards. The most popular graphs which are frequently connected with dashboards are *pie charts* and *gauge charts*. People can notice them in many dashboards samples. A pie chart is a circle divided into slices which present part-to-whole information. Nowadays, UI designers often use a modified variant—a *doughnut chart*. A gauge chart is circular graphics presenting a single value in a range. Despite their popularity, those two graphs were criticized by Few because of their circular shape (Figure 2.9). Pie charts are not effective media for the comparisons of values (Figure 1.2 in Introduction). Gauges usually occupy plenty of space comparing to the information volume they present.

Instead, Few recommends to use the following graphs in dashboards: popular *bar charts*, *line charts*, *scatter plots*, and less known but effective *bullet graphs*, *sparklines*, *box plots*, and *treemaps*. He also recommends combining selected graphs to save the space and emphasize relationships between the data—e.g., bar charts with line charts. Figures 2.10, 2.11 and 2.12 present examples of the recommended charts. Wexler et al. [2017] present illustrated glossary of charts suitable for dashboards.

Geographical maps are a special type of visual media. Few classifies them as organizers presenting spatial data, but sometimes, they are also classified as charts. They help users to connect data with geographical location. UI designers usually use them to compare regions or emphasize important geographical locations statistically.

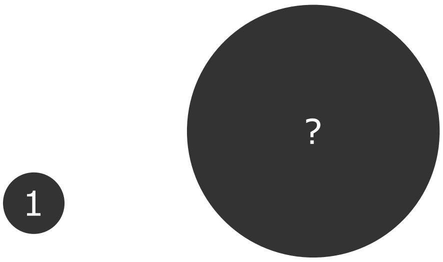


Figure 2.9: The circle on the right is 16 times bigger than the circle on the left. People, however, tend to underestimate differences in 2D areas. Redrawn from: [Few, 2006].

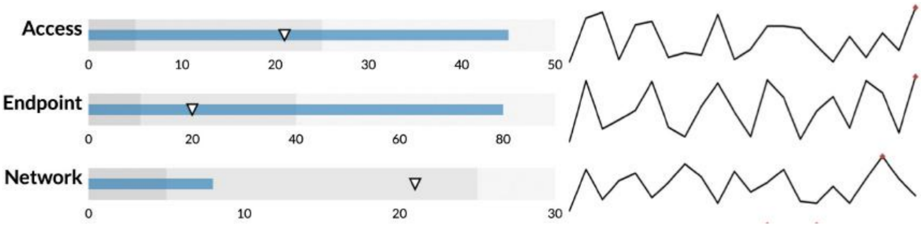


Figure 2.10: Combination of a bullet graph (left) invented by S. Few and sparklines (right) invented by E. R. Tufte. Bullet graphs present values in a range. Sparklines present evolution of values in time. Source: [Jacobs and Rudis, 2014].

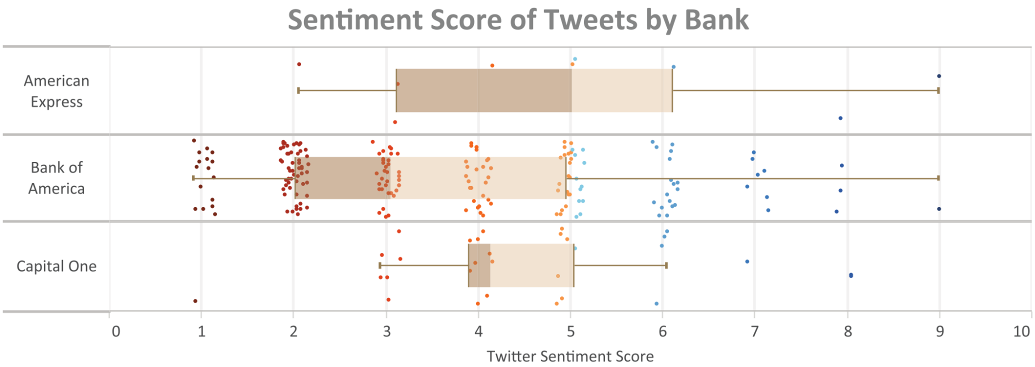


Figure 2.11: Combination of a box plot invented by J. W. Tukey and a dot plot. They present distribution of values. Source: [Wexler et al., 2017].

Icons are the next important media of dashboards. They are represented by symbols, sometimes, they have a specific color. As mentioned in Section 2.5, they present qualitative data. UI designers usually use them to emphasize important information which deserves attention of users. S. Few distinguishes alert icons, up/down icons and on/off icons. UI designers should use only well-known symbols.

The remaining media of Few's list are **drawing objects** and **images**. Drawing objects are less typical media. UI designers usually use them to graphically connect other media (e.g., lines or arrows). Images are usually used to present photographs (e.g., faces of people). UI designers should use images only if necessary and they should present them in a high resolution.

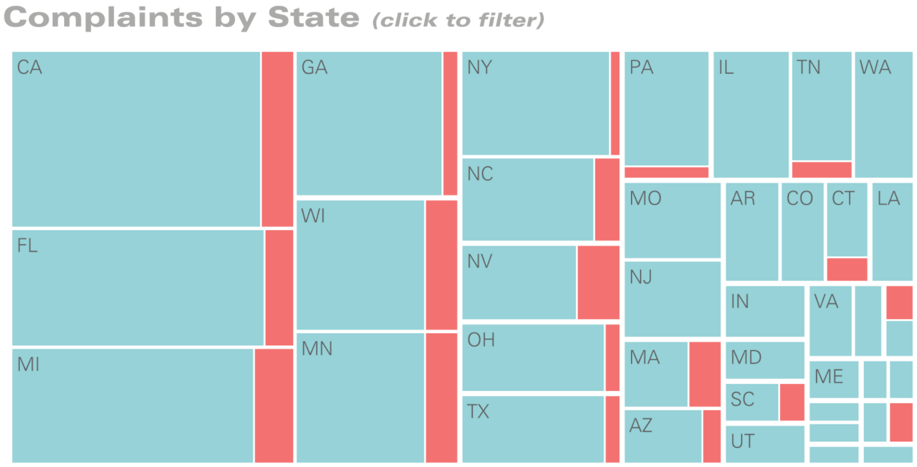


Figure 2.12: Treemap invented by B. Shneiderman for presentation of hierarchical data. The figure shows open and closed complaints by US state. Source: [Wexler et al., 2017].

2.6 Dashboard Design

The process of dashboard development should be, similarly to development of every other information system, based on some methodology for systems development life cycle. At least, it should consist of planning, analysis, design, implementation, testing, and maintenance [Kendall and Kendall, 2011]. This research focuses on the work of UI designers which consists of analysis and design of a user interface, including a continual evaluation of the UI usability. The result of their work is a dashboard prototype which looks the same as the expected result, but it lacks the real functionality. UI designers can use the prototype to perform initial tests of its usability. Then, they send the prototype to developers who implement a real dashboard ready for further tests and deployment. The readers interested in the implementation phase can find additional information in [Jacobs and Rudis, 2014; Stirrup, 2016; Alexander and Walkenbach, 2010; Beeley, 2018].

We can find a variety of design instructions and advice. For instance, Malik [2005] presents dashboard design steps in the form of the three questions: "What information? For whom? How to present?" On the other hand, Few [2006] focuses more on dashboard design problems and describes how to avoid the problems. Wexler et al. [2017] demonstrate it in the form of dashboard design scenarios. The following list uses this knowledge and presents important dashboard design steps which should not be missed during the design phase:

1. analysis of dashboard purpose
2. definition of a dataset
3. selection of display media
4. creation of a dashboard layout
5. simplification and evaluation

2.6.1 Analysis of Dashboard Purpose

In the beginning, UI designers and management should analyze the primary purpose of the dashboard.

- UI designers should determine end-users and understand their goals and needs. There are many techniques for this purpose (e.g., observing and interviewing users; creating personas, scenarios, story boards or use-case diagrams [Buley, 2013; Goodwin and Cooper, 2011; Nielsen, 1994b; Preece et al., 2015]).

- Management should perform a cost-benefit analysis and decide whether the dashboard is worthy of investment. They should keep in mind that the dashboard is neither a marketing tool nor a tool which can solve every problem with communication and data management [Eckerson, 2006].

2.6.2 Definition of Dataset

The definition of dataset depends on the information which we want to present to users. Analysts should select only the data which represents the information. These data are however usually difficult to understand at a glance. The next step is to derive performance measures (KPIs) from the raw data. Performance measures are often represented by aggregated values or qualitative values (e.g., customer satisfaction). The readers can find the instructions on how to create the performance measures, including examples, in [Malik, 2005; Eckerson, 2006; del-Rey-Chamorro et al., 2003].

2.6.3 Selection of Display Media

We can usually visualize the same data using different display media. According to Few [2006], the selection of an appropriate display medium should be based on the two factors:

1. “It must be the best means to display a particular type of information that is commonly found on dashboards.” The display medium should emphasize the information (meaning of the data) we want to convey to the users. It also means that the display medium should be compatible with the displayed kind of data (e.g., a chart should be able to present all dimensions of the data). Finally, the display medium should be comprehensible for users (it should reflect their experience).

2. “It must be able to serve its purpose even when sized to fit into a small space.” UI designers should consider the size of the space which the medium needs to occupy and compare it with the information volume it presents.

Then, UI designers should style the selected display media. Few [2006], Wexler et al. [2017] or Tufte [2001] provide recommendations—e.g.:

- Avoid rendering charts in 3D because it makes them more difficult to read.

- Use subtle instead of vivid colors in charts. Vivid colors should be used only for emphasizing of information (Figure 1.2).

- Design charts for color-blind people (Section 4.1).

- Minimize *chartjunk* (Subsection 2.6.5).

- Avoid distortion of charts. (Subsection 2.6.5)

2.6.4 Creation of Dashboard Layout

Created visual media need to be arranged on a screen. Locations of media on a screen can emphasize relationships between the data. They can also emphasize selected media. Few divides a screen into regions and describe "lucrativeness" of the screen regions (Figure 2.13).

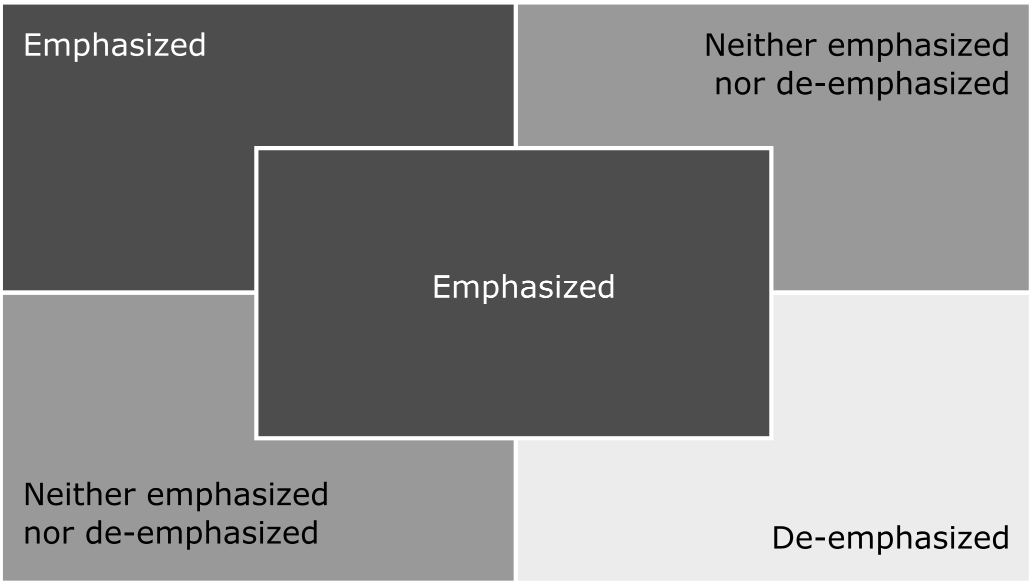

Emphasized
Neither emphasized nor de-emphasized
Emphasized
Neither emphasized nor de-emphasized
De-emphasized

Figure 2.13: Association of dashboard regions with different degrees of visual emphasis. Based on: Few [2006].

Since the space of a screen is limited, UI designers should focus on the clarity of the screen. They need to consider the possibility that the users can use a device with a small resolution of the screen (e.g., mobile phone). Web designers often use responsive web design [Marcotte, 2011]. UI libraries usually provide a set of layouts.

2.6.5 Evaluation and Simplification

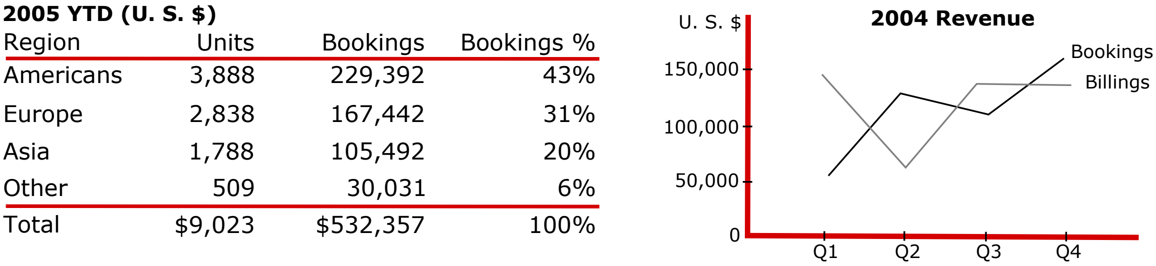
Evaluation of design quality is the essential step which UI designers should perform continuously during the design phase. There are many aspects which UI designers can evaluate. These aspects correspond with design recommendations (quality of selected graphs and layouts, presentation quality, simplicity or overall usability). For instance, Eckerson [2006] presents criteria for evaluation of performance dashboard design—e.g., layouts, the flexibility of graphs or the ability to personalize the dashboard. He describes them qualitatively. On the other hand, Tufte [2001] provides two quantitative metrics for evaluation of chart quality: *data-ink ratio* and *lie factor*.

$$\text{data-ink ratio} = \frac{\text{data-ink}}{\text{total ink}} \tag{2.1}$$

$$\text{lie factor} = \frac{\text{size of effect shown in graphics}}{\text{size of effect shown in data}} \tag{2.2}$$

The data-ink ratio measures the amount of *chartjunk*—a term created by E. Tufte describing elements of a chart which are used only as a decoration). The ratio compares the ink presenting data and the total ink which is needed to print graphics (Figure 2.14). Tufte [2001] recommends minimizing the non-data ink.

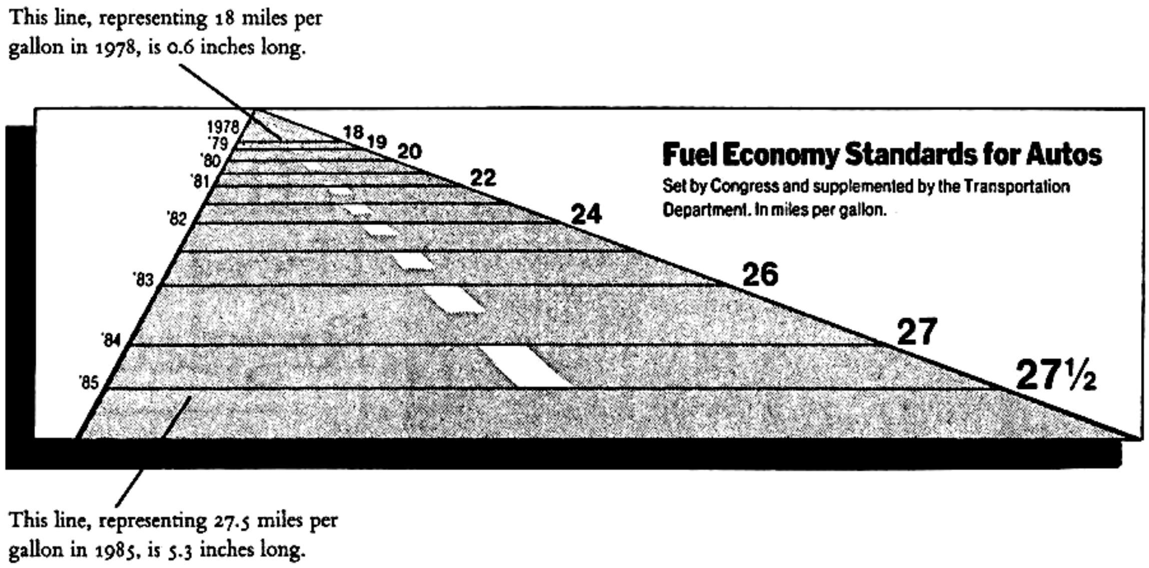
The lie factor measures distortion of graphics. The effect shown in graphics should be the same as it is shown in data (e.g., the two times bigger graphical element should represent the two times higher value). Figure 2.15 presents graphics with a high value of the lie factor.



2005 YTD (U. S. $)

Region	Units	Bookings	Bookings %
Americans	3,888	229,392	43%
Europe	2,838	167,442	31%
Asia	1,788	105,492	20%
Other	509	30,031	6%
Total	$9,023	$532,357	100%

Figure 2.14: Example of the charts containing low amount of non-data ink (it is represented by the red color). Viewers are not distracted by chartjunk and they can focus on presented data. Based on [Few, 2006].



This line, representing 18 miles per gallon in 1978, is 0.6 inches long.

This line, representing 27.5 miles per gallon in 1985, is 5.3 inches long.

Figure 2.15: Figure presented in New Your Times (9th August, 1978). The lie factor of the graphics is 14.8. Such a high value can cause confusion of viewers. Source: [Tufte, 2001].

Few [2006] works with the idea of data-ink ratio and applies it for evaluation of dashboard simplicity (the amount of ink is replaced by the number of pixels). He provides a framework for iterative improvement and simplification of dashboards. It consists of the following steps:

1. **"Reduce the Non-Data Pixels"**
   - (a) "Eliminate all unnecessary non-data pixels."
   - (b) "De-emphasize and regularize the non-data pixels that remain."
2. **"Enhance the Data Pixels"**
   - (a) "Eliminate all unnecessary data pixels."
   - (b) "Highlight the most important data pixels that remain."

Finally, UI designers should evaluate the result with the real users and analyze the real usability of the dashboard using usability metrics [Tullis and Albert, 2010].

2.7 Design Problems

The problem of dashboard design recommendations is that they are usually qualitative and it is difficult to evaluate them formally. Practical application of Tufte's quantitative metrics can be complicated as well (elaborate recognition of data and non-data pixels). Hence, UI designers need to perform testing with the real users.

Since user testing usually requires additional time and expenses, UI designers are often forced to do it insufficiently. They often test user interfaces by themselves, not with the real users. It can lead to later troubles. UI designers miss many usability problems since they can not see a user interface the way the users would do. The users are usually involved before deployment of the system. It is usually too late to make radical changes of the UI design in this phase. Few [2006] discusses the most common mistakes in dashboards which should be detected in the early design phase. The following list provides a brief overview:

- exceeding the boundaries of a single screen
- unimportant information and non-data pixels (e.g., decorations)
- important information is hidden
- inaccurate information, distorted display media
- unclear meaning of data, data are missing the context
- inappropriate display media, poorly designed display media
- inappropriate layout
- the viewer is distracted (vivid colors, decorations)

2.8 Summary

This chapter provided basic knowledge regarding dashboards—visual displays which present important information on a single screen to support decision making of users. They use the advantage of graphical presentation of data to convey the information to users comprehensively and emphasize important relationships between the data. We can use them for strategic, analytical and operational purposes. They are used in business intelligence. Performance dashboards are popular tools to evaluate the progress of strategy and monitor an organization's performance.

As Stephen Few [2006] explained in examples, well-designed dashboards can be very helpful. On the other hand, poorly designed dashboards can lead to serious usability problems. He showed that most of the existing dashboards contain some design problems. He presented an overview of frequently made design problems, which can serve as heuristics for usability evaluations. Their application is, however, limited by the knowledge of evaluators who need to be able to understand the heuristics and apply them in a particular situation correctly. Hence, it would be useful to detect some of the design problems automatically during the design phase and decrease the time and cost of user interface evaluation.

Chapter 3

Evaluation of User Interfaces

Evolution of information technologies has brought new possibilities for human-computer interaction. The decreasing costs of hardware components make computers available to more people. They use them in their daily routines. Computers are parts of mobile phones, home appliances or cars. Increasing internet penetration rate spreads their influence into distant and peripheral places. Information technologies provide various services—e.g., communication services, online shopping or e-government. For instance, in 2007, Estonia became the first country which used the Internet for general elections [Alvarez et al., 2009]. We can say that information technology became an essential part of people's life.

The dramatic spread of information technologies has brought new problems into the management of their usability. Designers of user interfaces need to consider usability requirements of a wider spectrum of users. There are specific groups of users like children or seniors who have different needs and preferences [Read and Markopoulos, 2013; Lee et al., 2011]. UI designers should correspond new approaches to human-computer interaction with the experience of users and their motivation to learn new things. Vendors should keep in mind that a user interface is the first part of the system which users see. The terms like aesthetics and first impression play a role in the acceptance of a system by users [Tractinsky et al., 2000]. A study has shown that users create a simple subjective opinion about a user interface in less than one second [Lindgaard et al., 2006]. Last but not least, users do not use information technologies only for accomplishments of some duties. They use them also for fun, satisfaction or enjoyment. UI designers should focus on the value of the product and improve the overall user experience [McCarthy and Wright, 2007].

The problem of user experience and user interface usability has been recognized for a long time. For instance, in 1987, Ben Schneiderman published the popular book [Shneiderman, 1987] providing the list of eight golden rules of interface design. In 1994, Jakob Nielsen [1994b] described the importance of user interface usability and its impact on the acceptability of the whole system. Since then, numerous studies and researches have been done in order to find a better way of design and evaluation of user interfaces. Even though researchers have proposed numerous approaches, we still need to consider unpredictability of human perception, thinking, and reactions, which cannot be completely generalized.

The field of human-computer interaction, user interface design and evaluation is elaborate. Readers can find many books and papers presenting knowledge about these fields. It is not the purpose of this chapter to provide an exhaustive list of all available publications and cover all aspects of the research area. This chapter provides a brief overview of the methods for user interface evaluation. It explains basic terms regarding the usability of user interfaces and provides existing classifications of evaluation methods. Then, it focuses on the evaluation approach based on heuristics and guidelines. It presents existing methods and considers their automation and application in the evaluation of dashboard interfaces. It analyzes Ngo's object-based metrics of aesthetics [Ngo et al., 2003] and shows the problem of the ambiguous definition of interface objects.

3.1 Terminology

The international standard ISO 9241-210 defines the usability term as "extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" [ISO, 2010]. We can consider usability in different contexts—e.g., software/product quality, system acceptability or user experience.

3.1.1 Software/Product Quality

First of all, we can recognize usability in the context of *software quality*. The international standard ISO/IEC 9126-1 describes quality models and metrics for evaluation of software quality. It defines software quality as "the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs" and specifies a set of the six characteristics: *functionality*, *reliability*, *usability*, *efficiency*, *maintainability*, *portability*. Usability is defined as "the capability of the software product to be understood, learned, used and attractive to the user when used under specified conditions." [ISO, 2001; Mendes and Mosley, 2006]. The standard specifies the five subcharacteristics of usability: *understandability*, *learnability*, *operability*, *attractiveness*, *usability compliance*.

In 2011, the ISO/IEC 9126-1 standard was replaced by the ISO/IEC 25010 standard. It presents the Product Quality Model which redefines the characteristics of software quality. The model consists of the 8 characteristics (instead of 6): *functional suitability*, *reliability*, *performance efficiency*, *usability*, *security*, *compatibility*, *maintainability*, and *portability* [ISO, 2011; Wieczorek et al., 2014]. The usability characteristic adapts the definition of the ISO 9241 standard, and it consists of the 6 subcharacteristics (instead of 5):

- *appropriateness recognizability*: "the degree to which users can recognize whether a product or system is appropriate for their needs."
- *learnability*: "the degree to which a product or system can be used by specified users to achieve specified goals of learning to use the product or system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use."
- *operability*: "the degree to which a product or system has attributes that make it easy to operate and control."
- *user error protection*: "the degree to which a system protects users against making errors."
- *user interface aesthetics*: "the degree to which a user interface enables pleasing and satisfying interaction for the user."
- *accessibility*: "the degree to which a product or system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use."

3.1.2 System Acceptability

Nielsen [1994b] describes the usability term in the context of *system acceptability*, which he defines as the ability "to satisfy all the needs and requirements of the users and other potential stakeholders, such as the users' clients and managers." He explains it as a combination of the social and practical acceptability and divides the practical acceptability into several categories—e.g., *cost*, *compatibility*, *reliability*, and *usefulness*. He focuses on the usefulness category, which represents the ability to use a system for the achievement of the desired goal. He bases on the work of Grudin [1992] and characterizes usefulness by the two attributes:

- *utility*: "the question of whether the functionality of the system in principle can do what is needed."
- *usability*: "the question of how well users can use the system functionality."

He divides usability into the five attributes: *learnability*, *efficiency*, *memorability*, *satisfaction*, and *errors*. Then, a usable system should be easy to learn, efficient to use, easy to remember, subjectively pleasing and it should contain few errors. Readers can notice that Nielsen's usability attributes describe similar requirements as the usability subcharacteristics of ISO/IEC 9126-1 and ISO/IEC 25010 (Subsection 3.1.1).

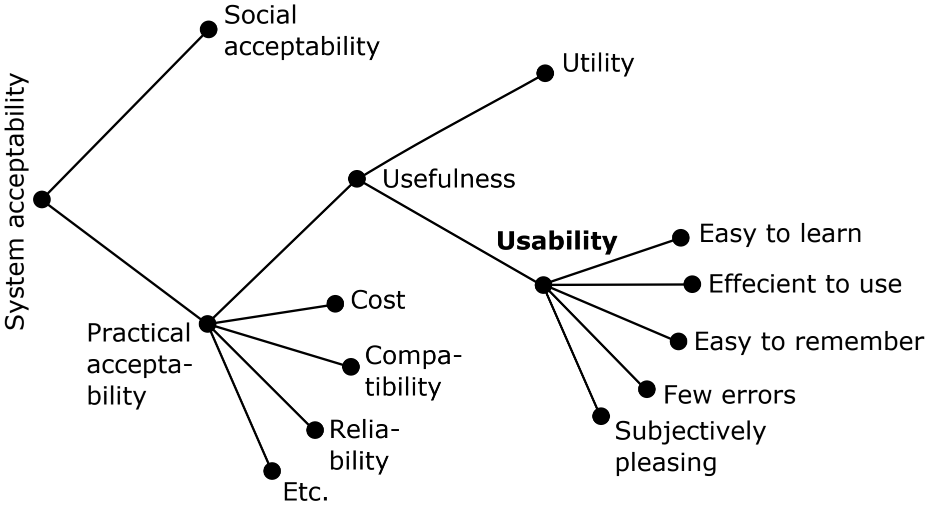


Figure 3.1: Usability in context of Nielsen's model of system acceptability. Redrawn from: [Nielsen, 1994b].

Figure 3.1 presents Nielsen's scheme of the model of system acceptability. Kim [2015] provides a further discussion about the importance of system acceptability (and proposes a discipline *Acceptability Engineering*).

3.1.3 User Experience

In recent years, UI designers have mentioned the usability term frequently in the context of the *user experience* (UX) term. Some of UI designers and researchers point out that usability is a narrow aspect of user interface. They argue that a product should be designed more than usable. It should reflect "non-utilitarian" [del-Rey-Chamorro et al., 2003] aspects of user interface like users' perception, emotions, feelings or satisfaction. UI designers should focus on the value of the product, and they should improve the users' overall experience of the product [Hassenzahl and Tractinsky, 2006; Law et al., 2009]. Usability is often considered as a part of user experience [Bevan, 2009].

We can find various meanings of user experience [Forlizzi and Battarbee, 2004; McCarthy and Wright, 2007; Tullis and Albert, 2010]. Law et al. [2009] performed a survey with 275 respondents from academia and industry in order to describe the scope and characteristics of UX. The results showed that "the respondents understand the notion of user experience very differently." We can find numerous further studies dealing with the meaning of user experience [Bargas-Avila and Hornbæk, 2011; Law et al., 2014; Lallemand et al., 2015]. The international standard ISO 9241-210 defines user experience as "person's perceptions and responses resulting from the use and/or anticipated use of a product, system or service" [ISO, 2010]. Morville [2005] created a framework to describe the seven UX facets defining product requirements: *useful*, *usable*, *desirable*, *findable*, *accessible*, *credible* and *valuable* (Figure 3.2). Roto and Rautava [2008] found the four elements: *utility*, *usability*, *social value*, and *enjoyment*. Bargas-Avila and Hornbæk [2011] analyzed existing publications describing UX from 2005 to 2009 and showed that the most common dimensions connected with UX are *emotions*, *enjoyment*, and *aesthetics*.

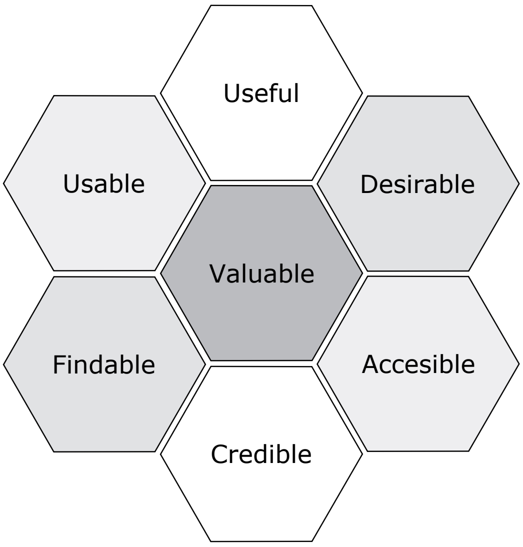


Figure 3.2: Morville's User Experience Honeycomb. Redrawn from: [Morville, 2005].

We can notice that many of the sources describing UX emphasize the importance of UI appearance and aesthetics. The aesthetics term is derived from the Greek *aisthanesthai*—to perceive. Merriam-Webster's dictionary [2004] explains it as "pleasurable to the senses" or "attractive." Its importance is discussed by many "non-UX" publications [Kristeller, 1951; Lavie and Tractinsky, 2004; Tractinsky, 2004]. This aspect of a UI is usually perceived very quickly before a viewer fully understands the content of the UI [Lindgaard et al., 2006]. It corresponds to various aspects—e.g., *simplicity* [Karvonen, 2000] / *complexity* [Michailidou et al., 2008]. Moshagen and Thielsch [2010] define the four facets of aesthetics: *simplicity*, *diversity*, *colorfulness*, and *craftsmanshift*. Aesthetics often plays an important role in the acceptance of a whole product. Moreover, it may improve interface usability [Kurosu and Kashimura, 1995; Tractinsky, 1997; Tractinsky et al., 2000]. We can notice that ISO 25010:2011 and Nielsen's model of product acceptability consider aesthetics (or subjective satisfaction) as part of usability (Subsections 3.1.1 and 3.1.2), on the contrary to some explanations of UX.

UX is sometimes criticized for being vague since it is connected with fuzzy and dynamic concepts [Hassenzahl and Tractinsky, 2006; Law et al., 2009]. Characteristics like user emotions or interface aesthetics are subjective. They can change during the time as the preferences and expectations of people are changing. Either way, user experience has become an actual trend in the field of UI design and evaluation. UI designers should not overlook it.

3.2 Classification of Methods

Since usability is connected with various aspects of user interface overall quality, there exist a variety of methods for analysis and evaluation of user interface usability. The ISO/TR 16982 standard provides information on human-centered usability methods applicable to design and evaluation [ISO, 2002]. However, there are many non-standardized methods which are based on an ad-hoc definition. Readers can find comprehensive lists of evaluation methods in [Ivory and Hearst, 2001; Fernandez et al., 2011]. They classified them according to various factors, which are more or less important for this research.

3.2.1 Common Classification

The most common classification of methods presented by Ivory and Hearst [2001] or Fernandez et al. [2011] consists of the five classes:

- **Testing methods** are a group of methods which are based on observation of user interaction with a user interface. Evaluators let users perform selected tasks and analyze the process of the completion of the tasks. They measure time, analyze users' behavior and detect usability problems. A favorite approach is to let users interact with a UI without any specification of tasks, so the evaluators can see how the users understand the user interface. Results are quantitative. Examples of methods are: *think aloud protocol* (user talks during the test) [Lewis and Rieman, 1993], *A/B testing* [Siroker and Koomen, 2013], *mouse tracking* [Freeman and Ambady, 2010], *eye tracking* [Duchowski, 2007] or *performance measurements* and *log file analysis* [Andrews, 1998]. Evaluators might capture video of testing and analyze it remotely (*remote testing*), or after the testing (*retrospective testing*), so they do not interfere the testing [Nielsen, 1994b].

- **Inquiry methods** represent a group of methods which are based on communication between an evaluator and a user. Evaluators let selected users work with a user interface, and then they acquire and analyze their feedback. Results of the evaluation are subjective and qualitative. They usually represent users' requirements and needs. The most common methods and tools are *user feedback*, *interviews*, *surveys*, and *questionnaires* [Buley, 2013; Goodwin and Cooper, 2011; Nielsen, 1994b; Preece et al., 2015].

- **Inspection methods** focus on evaluation of user interface by expert evaluators without the presence of users. The evaluators find usability problems using a predefined set of criteria or heuristics [Nielsen, 1994b,c; Hollingsed and Novick, 2007]. UI designers can use the methods in early phases of the development process. Some of the methods can be automatized. On the other hand, the methods cannot reflect all quality aspects (e.g., user experience, subjective perception). Examples of methods are *cognitive walkthrough* (evaluators walk through user tasks and detects usability problems) [Wharton et al., 1994], *heuristic evaluation*, or *guideline review* (evaluators analyze a UI and compare it against heuristics or more specific guidelines). See Section 3.3.

- **Analytical modeling methods** provide different kinds of models which evaluators use to generate usability predictions—e.g., usability problems, the execution, or learning time. An example of a frequently used model is the GOMS model modeling *goals*, *operators*, *methods* and *selection rules* to predict the execution and learning time [Card et al., 2018]. The model has many variations—e.g., KLM (*Keystroke-Level Model*) [Card et al., 1980; Kieras, 2001; Katsanos et al., 2013] or the original version CMN-GOMS.
- **Automation/Simulation methods** simulate user interaction using modeling languages or simulation algorithms, e.g., Petri nets or genetic algorithms [Ivory and Hearst, 2001].

3.2.2 Level of Automation

The next factor which is important for this research is the level of automation [Ivory and Hearst, 2001]. It considers the four types of methods:

- **None**: a method does not support any automation of evaluation.
- **Capture**: a method provides the ability to capture the process of evaluation (logs of interaction with UI).
- **Analysis**: a method automatically detects usability problems.
- **Critique**: a method automatically detects usability problems and offers solutions for the problems.

The methods based on subjective feedback of users are usually without any automation support. On the other hand, the methods based on inspection or modeling of usability usually offer some level of automation. Automation provides certain advantages like decrease of time, expenses and human resources. On the contrary, the methods with a higher level of automation are usually narrow-focused. They do not consider the context of evaluation and the subjective factor of users. They detect false-positive usability problems more frequently than the methods without automation.

3.2.3 Generalizability, Precision, Realism

Carpendale [2008] takes the selected evaluation methods and analyzes them concerning the three factors:

- **Generalizability**: the extent to which the results of evaluation can be generalized to other users or situations.
- **Precision** (reliability [Leung, 2015]): the extent to which the evaluator control all aspects of evaluation (results are reliable and replicable).

- **Realism**: the extent to which the context of evaluation is similar to the context of real usage.

Carpendale [2008] shows that improvement of one factor will decrease the level of the remaining two factors (Figure 3.3). For example, field study focuses on evaluation of user interface outside a laboratory in a real situation (real users and environment). Evaluators observe users without interference. Hence, the results of the evaluation are realistic but not reliable, replicable and usually not generalizable. In contrast to field study, laboratory experiment is based on performing arranged tasks. Evaluators provide users with instructions. Such results are usually reliable and replicable, but they do not reflect reality, and we cannot generalize them. Finally, a formal theory provides generalizability, but it usually lacks realism and precision.

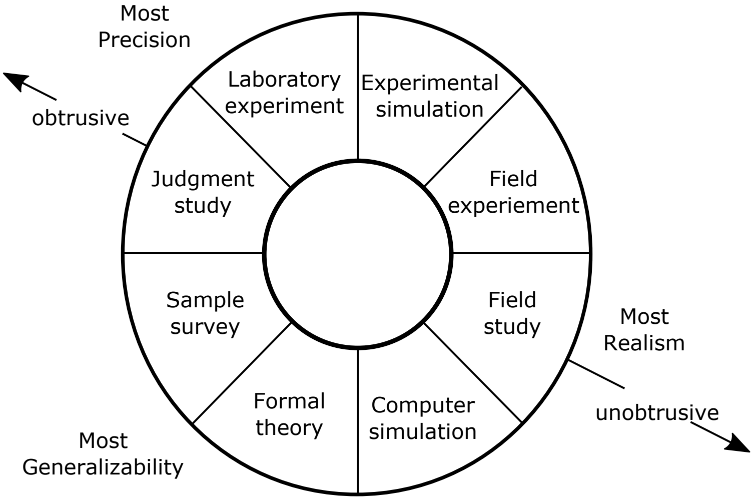


Figure 3.3: Comparison of selected methods according to the rates of generalizability, precision, realism and obtrusiveness. Redrawn from: [Carpendale, 2008].

We can see that the three factors correspond with *obtrusiveness* of users during an evaluation (e.g., interference between users and evaluators). The most precise methods are usually obtrusive. On the other hand, unobtrusive environment helps to generate realistic results. As pointed out by Preece et al. [2015], evaluators should combine several usability methods. Then, they can get generalizable, reliable and realistic results.

3.3 Evaluation Based on Heuristics and Guidelines

One of the goals of this research is to find metrics which could be used for automatic measuring of dashboard usability and overall quality. Such an approach is cheap and could be used during the design phase without the presence of users. The research follows the inspection methods, particularly the evaluation based on heuristics and guidelines.

Heuristic evaluation is a usability inspection method designed by Nielsen and Molich [1990]. The goal of the method is to analyze a user interface and correspond its characteristics with predefined usability principles called *heuristics*. This process leads to the detection of usability problems. We can find numerous heuristics for evaluation of user interfaces—e.g., Nielsen's usability heuristics [Nielsen, 1994a]. Few [2006], Jacobs and Rudis [2014] and Wexler et al. [2017] provide heuristics for evaluation of usability of dashboards. The problem of the proposed heuristics is that they are abstract. Specialists in UI design need to be present to understand and apply the heuristics correctly. It is also not recommended to base the results on one evaluator since evaluators do not need to reach an agreement in all usability problems [Jeffries et al., 1991].

Usability guidelines are more specific than usability heuristics [Jeffries et al., 1991]. Their application usually does not require a UI design expert. They are often based on quantitative metrics, which calculate values of selected user interface attributes. Engineers can transform such guidelines into runnable code and apply them for quick evaluation of a high number of user interfaces (e.g., web pages [Vanderdonckt et al., 2004]). On the other hand, they are more straightforward than heuristics. They do not evaluate interaction with a UI. They focus on basic UI characteristics (e.g., color, size or distribution of elements). It is usually not easy to design a guideline which would analyze advanced aspects of UI usability (e.g., aesthetics). It has been challenge for researchers and practitioners to design more and more advanced guidelines.

3.3.1 Pixel-based Evaluation

A user interface can be implemented in various programming languages, and it can use many technologies. It might be elaborate to create a tool which would be able to work with an internal representation of a user interface. Hence, it might be useful to take a static snapshot of the screen and evaluate the UI as a raster image. The following text presents selected guidelines based on pixel-based metrics which measure usage of individual color values, or distribution of those values in a raster image.

Colourfulness

As described in Subsection 2.6.3, one of Few's heuristics recommends UI designers to use subtle colors. We can evaluate this heuristic by measuring colorfulness of a UI snapshot. For instance, Yendrikhovskij et al. [1998] base colorfulness on the image saturation measured in the CIElab color space where the saturation is computed as the image chroma divided by the image lightness:

$$C_i = S_i + \sigma_i, \tag{3.1}$$

where S_i is the average saturation of an image i and σ_i its standard deviation. $C_i = 0$ represents achromatic image. Images with $C_i \approx 2$ can be considered highly colorful. The metric was used by Reinecke et al. [2013].

Number and Share of Used Colors

According to [Few, 2006], dashboards should contain a low number of color values. Common graphical libraries usually work with the RGB color space. The color values are usually stored as a 24-bit number (8 bit for every color channel), which makes more than $2^{24} = 16.77$ million distinct color values. There is a high probability, that human will not recognize all displayed color values (especially those with a low frequency of occurrence). Hence, it might be reasonable to reduce the number of used colors. We can use various algorithms like posterization or clustering of pixels into larger groups. The metric was used by Purchase et al. [2012].

Distribution of Colors

Inappropriate layout and distribution of graphical elements in user interface are frequent design problems [Few, 2006]. Since pixel-based metrics consider a UI as an image represented by a matrix of pixels, evaluators need to detect graphical elements in the image first. They can do it subjectively by themselves or by image processing methods (Section 4.2). It is, however, elaborate task, especially when they analyze complex UIs like dashboards. The second option is to analyze colors of pixels and their distribution in the image. For example, we can threshold the image and measure the distribution of the black and white pixels or convert the image into the grayscale color space and measure the distribution of color intensity.

Kim and Foley [1993] present a formula for measuring balance between the left and right side of a black-and-white image:

$$\text{left-right balance} = \frac{\text{total weight of less heavy side}}{\text{total weight of more heavy side}} \tag{3.2}$$

$$\text{total weight of side} = \sum 1\text{s} \cdot f(\text{distance away from center}) \tag{3.3}$$

where '1s' represents the black pixels (graphical elements) of a side in the black-and-white color space. Similarly, we could measure vertical balance or balance of an image represented in the grayscale color space (we could replace '1s' with value of normalized color intensity).

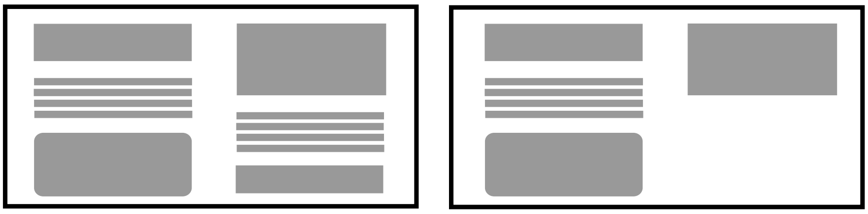


Figure 3.4: A simplified example of balanced (left) and unbalanced (right) screens.

Another option is to measure the symmetry between the sides of an image—e.g., left-right symmetry:

$$\text{left-right symmetry} = \frac{\text{hit}}{\text{hit} + \text{miss}} \tag{3.4}$$

$$\text{miss} = \sum_{(v_1,v_2)\in S} |v_1 - v_2|, \quad \text{hit} = \sum_{(v_1,v_2)\in S} 1 - |v_1 - v_2| \tag{3.5}$$

where S is a set of all pixel pairs (v_1, v_2) located in mirrored positions from the central axis. The pixels contain a logical value ('1s' or '0s') or normalized color intensity $[0, 1]$. Figures 3.4 and 3.5 present examples of balanced/unbalanced and symmetrical/asymmetrical screens.

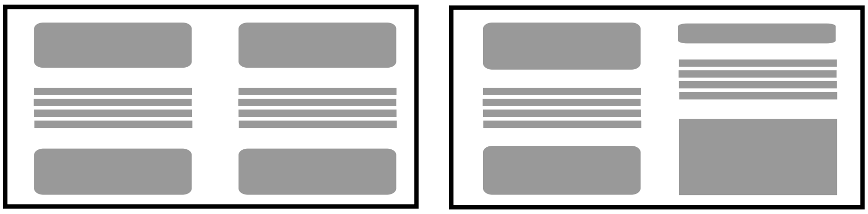


Figure 3.5: A simplified example of symmetrical (left) and asymmetrical (right) screens.

The pixel-based approach is a quick way how to measure simple characteristics of a user interface. These characteristics can be combined. For instance, Miniukovich and Angeli [2014] combine several pixel-based characteristics (e.g., colorfulness) for measuring UI complexity. On the other hand, the pixel-based approach does not reflect the perception of people who recognize objects within a scene as described by Baker et al. [2009] instead of a matrix of pixels.

3.3.2 Object-based Evaluation

The second approach focuses on the analysis of objects located in a user interface. Vanderdonckt and Gillo [1994] based on Foley and Van Dam [1982] recognize the two kinds of objects: *interaction* and *interactive* objects. Interaction objects (also *widgets* or *controls*) represent static (e.g., labels or separators) and dynamic (e.g., buttons, text fields) objects of a user interface. Interactive objects represent the remaining objects (e.g., drawings or pictures). Then, the rectangular boundaries of all objects (*regions*) form a layout of a user interface (Figure 3.6).

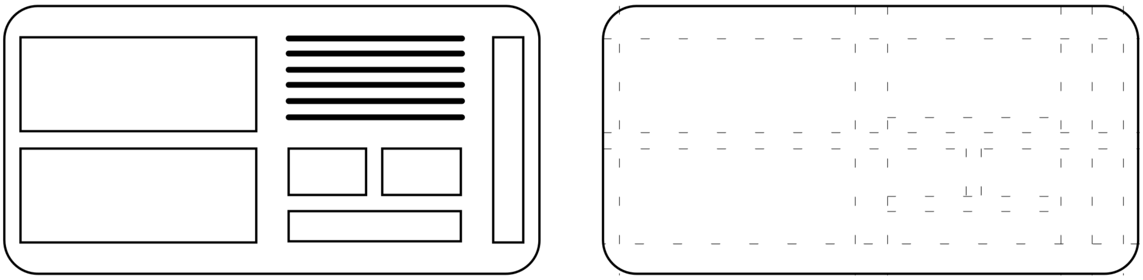


Figure 3.6: The left figure represents a simplified screen containing objects. The right figure represents the underlying layout grid. Source: Vanderdonckt and Gillo [1994].

Vanderdonckt and Gillo [1994] have published 30 advanced visual techniques for the analysis of screen layouts, divided into the five groups: *physical*, *composition*, *association* (and dissociation), *ordering* and *photographic* techniques. The techniques are described qualitatively by visual examples and descriptions. Some of them (like the physical ones) are easily convertible to an algorithm than others (like the photographic ones) which are more complex and focus on the subjective feeling of the viewer.

Quantitative measuring of object characteristics became significant with the evolution of graphical user interfaces. In the 1980s, UI designers used metrics to evaluate textual user interfaces [Smith and Mosier, 1986; Tullis, 1984]. In the 1990s, they applied metrics in the tools for the automatic design of graphical user interfaces [Ivory and Hearst, 2001]. Researchers tried to improve assistance which would help with the placement of interface objects. These tools focused on generating specified kinds of user interfaces (e.g., a dialog box) using a predefined strategy to create a suitable screen layout rather than the evaluating visual attributes of an arbitrary user interface.

Kim and Foley [1993] generate and analyze spatial properties of a dialog box layout using the tool called DON. They compare the size and shape of interface objects to help with creating suitably aligned layouts. Bodart et al. [1994] present two strategies for interface objects placement as a part of the TRIDENT project. They generalize the problem into the three subparts: *localization* (position), *dimensioning* (size) and *arrangement* (order) and present the set of simple mathematical relationships between interface objects in order to provide a better description of UI layout. Sears [1995] developed another tool called AIDE, which focuses on the layout efficiency (*layout appropriateness* [Sears, 1993]), alignment, balance and a possibility to specify custom constraints. Shneiderman et al. [1995] added more metrics for a spatial and textual layout like objects density, margins, aspect ratio or number of objects. They applied them in the tool called SHERLOCK for the evaluation of interface consistency [Mahajan and Shneiderman, 1997]. Parush et al. [1998] created a tool using a numerical model for measuring the size, alignment, grouping, and density of interface elements. They used the measures to compute screen complexity.

In the 2000s, the rapid evolution of the Internet made UI designers focus on the evaluation of webpage user interfaces. Ivory [2001] gathered knowledge about design guidelines and heuristics until 2001 and presented the list of 157 quantitative metrics for evaluation webpage elements (e.g., analysis of the amount of text on a page, color usage, and consistency). UI designers put higher emphasis on the soft design aspects like aesthetics and the first impression of users. Ngo, Teo and Byrne [2000a, 2000b, 2001a, 2001b, 2003] attempted to describe aesthetics formally. They presented the 13 quantitative object-based metrics of aesthetics:

- **Balance**: "difference between total weighting of components on each side of horizontal and vertical axis."
- **Equilibrium**: "difference between the center of mass of the displayed components and the physical center of the screen."
- **Symmetry**: "extent to which the screen is symmetrical in three directions: vertical, horizontal, and diagonal."
- **Sequence**: "measure of how information in display is ordered in a hierarchy of perceptual prominence corresponding to the intended reading sequence."
- **Cohesion**: "extent to which the screen components have the same aspect ratio."
- **Unity**: "extent to which visual components on a single screen all belong together."
- **Proportion**: "comparative relationship of the dimensions of components to certain proportional shapes."
- **Simplicity**: "extent to which component parts are minimized and relationships between the parts are simplified."
- **Density**: "extent to which the percentage of component areas on the entire screen is equal to the optimal level."
- **Regularity**: "extent to which the alignment points are consistently spaced."
- **Economy**: "extent to which the components are similar in size."
- **Homogeneity**: "measure of how evenly the components are distributed among the quadrants."
- **Rhythm**: "extent to which the components are systematically ordered."

The following text will refer these metrics as *Ngo's metrics*.

Ngo's metrics strongly correspond with the selected techniques published by Vanderdonckt and Gillo [1994]. Readers can also notice the similarity with the pixel-based metrics of image balance and symmetry presented in Subsection 3.3.1. On the contrary to the formulas of the pixel-based Balance (Eq. 3.2) and Symmetry (Eq. 3.4), Ngo's formulas do not consider color or shape of interface objects. They analyze a screen as a set of rectangles (regions) representing the boundaries of interface objects. The regions are described only by their dimensions (size and position). The result of every metric is a value of the $[0, 1]$ range, which represents the rate of an aesthetic factor.

The following example presents the formula of Ngo's Balance metric:

$$\mathrm{BM} = 1 - \frac{\mid \mathrm{BM}_{\mathrm{vertical}} \mid + \mid \mathrm{BM}_{\mathrm{horizontal}} \mid}{2} \in [0, 1] \tag{3.6}$$

$$\mathrm{BM}_{\mathrm{vertical}} = 1 - \frac{w_L + w_R}{\max(\mid w_L \mid, \mid w_R \mid)} \tag{3.7}$$

$$\mathrm{BM}_{\mathrm{horizontal}} = 1 - \frac{w_T + w_B}{\max(\mid w_T \mid, \mid w_B \mid)} \tag{3.8}$$

where w_j is a weighting of a side $j \in \{L, R, T, B\}$ (left, right, top, bottom) containing n_j regions:

$$w_j = \sum_{i}^{n_j} a_{ij} d_{ij} \tag{3.9}$$

The weight of a side depends on the area a_{ij} of a region and the distance d_{ij} of the region from the center of the screen. Readers can find all formulas in [Ngo et al., 2000a].

Besides Ngo's metrics, there were other attempts to formalize characteristics corresponding to aesthetics—e.g., [Harrington et al., 2004]. This thesis focuses on Ngo's metrics.

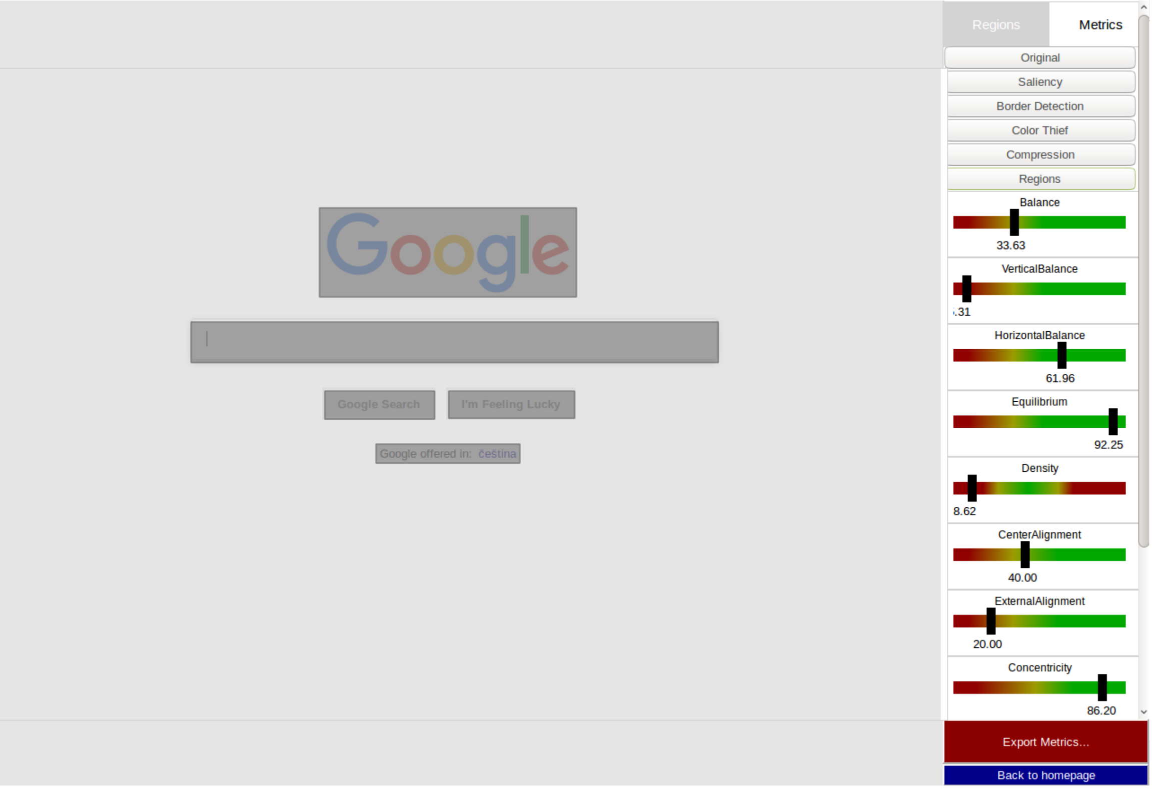
3.3.3 Ambiguity of Object-based Evaluation

In the 2000s and 2010s, numerous researchers evaluated the applicability of Ngo's metrics to the present time, especially for website interfaces. They usually based the evaluation of the metrics on the comparison of the measured results with the reviews of p users who rated n user interfaces. Their results depend on a selected group of users, analyzed user interfaces and approaches to the description of interface regions. I have detected four approaches of recognition of regions described in the following four paragraphs.

The first approach generates its own layouts containing exact descriptions of regions. The primary purpose is to simulate specific situations used for the comparison of user perception with the results given by a metric. Altaboli and Lin [2011] generate screens containing four black squares with different dimensions to test extreme values of the metrics Balance, Unity, and Sequence. Then, they demonstrate the correlation between these metrics and the user perception of aesthetics ($n = 8$, $p = 13$; users rated aesthetics on a 10-point scale). Salimun et al. [2010] generate layouts comprised of triangles. They confirm the effect of the metrics Cohesion, Economy, Regularity, Sequence, Symmetry, and Unity. However, they also point out that users prefer interfaces with a medium level of aesthetics ($n = 15$, $p = 72$; the users compared aesthetics between pairs of screens). Bauerly and Liu [2008] replace black squares with random images to make the displays look realistic. They show that a high number of interface objects decreases the aesthetic appeal ($n = 27$, $p = 16$).

The second approach is based on the analysis of the structural description of real interfaces. Purchase et al. [2011] use a browser extension to parse the document object model (DOM) of web pages to specify regions. They analyze most of Ngo's metrics (except Equilibrium, Symmetry, and Rhythm) and confirm the correlation between the metrics and user perception ($n = 15$, $p = 21$). However, their results show that the aesthetics do not match the interface usability, which contradicts the findings of Kurosu and Kashimura [1995] and Tractinsky [1997]. They pointed out that the approach of DOM processing does not consider the visual content of an identified component.

The third approach uses raster screenshots and tries to detect regions automatically, using image processing methods. It considers the visual aspect of screen compared to the previous approaches. Zheng et al. [2009] use the algorithm of iterative decomposition of a screen into quadrants of minimum entropy (*Quadtree decomposition*) based on low-level image statistics. They evaluate Balance, Symmetry, Equilibrium, and the number of quadrants and compare their influence on the judgment of the users ($n = 30$, $p = 22$). According to the results, the influence is not always the same (Balance has the highest influence, in contrast to Equilibrium). Reinecke et al. [2013] evaluate the same metrics as Zheng et al. They focus on the prediction of the visual complexity of an interface ($n = 450$, $p = 548$). They use Quadtree decomposition and *Space-based decomposition* (decomposition of a screen by separating the components along the vertical and horizontal spaces on the screen).



Balance 33.63
VerticalBalance .31
HorizontalBalance 61.96
Equilibrium 92.25
Density 8.62
CenterAlignment 40.00
ExternalAlignment 20.00
Concentricity 86.20

Figure 3.7: A screenshot of the QUESTIM tool designed by Zen and Vanderdonckt [2014]. The right part of the screen displays values of Ngo's metrics measured for the screenshot of Google homepage. The gray rectangles represents a manual description of regions.

The fourth approach depends on the manual selection of regions by users. Zain et al. [2008] describe an application for the manual dragging of interface objects combined with further image processing. They use the application to confirm the correspondence between the expected ranking of metrics and values calculated from regions gathered by the users dragging objects ($n = 12$, $p =$ unspecified) using Balance, Equilibrium, Symmetry, Sequence, and Rhythm. Mazumdar et al. [2015] base on this model, extend it with Cohesion and Unity and use it to evaluate the aesthetics of one type of interface—semantic web tools ($n = 11$, $p =$ unspecified). The measured values are similar for most of the analyzed interfaces. The recent research [Zen and Vanderdonckt, 2014] provide the QUESTIM tool, which enables loading of a website screenshot and lets users manually specify the regions representing the input for Ngo's metrics (Figure 3.7). According to their results evaluating all 13 metrics ($n = 4$, $p = 25$, 5-point Likert scale), only 5 of 13 metrics (Balance, Equilibrium, Density, Economy, and Proportion with the best results) correspond to the user reviews. However, they point out the small set of interface samples and the problem of the subjective selection of regions. They also suggest a possible improvement of the metric thresholds determining what is aesthetically efficient. Trausan-Matu and Dathan [2016] let users to reposition rectangular shapes in a window in the two cases: when the users were or were not told that the window should represent a configuration for a user interface. They observed that the users preferred regions to be organized in more sequential but less figurative way when they were supposed to represent objects of user interface. This finding supports the hypothesis, that values measured by Ngo's metrics does not necessarily need to be as high as possible.

As described in this subsection, we can specify objects at least according to three[1] different techniques. It makes objects ambiguous as well as the results of object-based formulas. Examination of the input variables of Ngo's formulas provides us closer information about the dependency of the formulas. We can characterize Ngo's metrics by the three kinds of dependency:

- Ω_{AD}: The metrics dependent on the accuracy of areas of regions and the distribution of regions on a screen: Balance, Equilibrium, Symmetry, Sequence, Density, Rhythm and Unity. The evaluator needs to specify the parts of the screen occupied by objects accurately.

- Ω_{AR}: The metrics based on the aspect ratios of regions: Cohesion and Proportion. The evaluator needs to specify the objects' ratios of width to height accurately.

- Ω_{G}: The metrics based on the level of screen granularity (number of regions, aligned points, or different sizes): Unity, Simplicity, Regularity, Economy, and Homogeneity. The evaluator needs to divide the parts of the screen occupied by objects accurately.

The three sets Ω_{AD}, Ω_{AR}, Ω_{G} will be used in the further analyses of Ngo's metrics considering the application of the metrics for dashboards evaluation.

[1] We can not consider the first approach which generates synthetic layouts as a technique for the description of interface regions.

3.4 Summary

Usability is an essential requirement of well-designed user interfaces. It affects product quality, acceptability and user experience. On the other hand, it is not the only requirement. Users expect a product to be valuable, provide them satisfaction, enjoyment and improve their overall experience. UI designers start focusing more on the design aspects like aesthetics, which have, according to some studies, [Kurosu and Kashimura, 1995; Tractinsky, 1997] an impact on usability.

There are various methods for evaluation of UI usability and overall quality. They are based on observation of users interacting with UI, communication with users, inspection of UI characteristics, simulation of user interaction or modeling usability predictions. They provide us with a different level of automation, generalizability, precision, and realism. Since observation of users interacting with a user interface provides realistic results, we cannot generalize the results, and the evaluation might be expensive. On the other hand, inspecting UI using design guidelines is cheaper, results are replicable, and we can apply it in the design phase without the presence of users. The design guidelines are however simple and focus on a narrow design aspect of UI.

Design guidelines are often based on quantitative metrics so they can be transformed into runnable code and used automatically. We can distinguish the metrics processing interface as a raster image represented by a matrix of pixels and the metrics analyzing objects of user interface (e.g., widgets). Pixel-based metrics can analyze UI characteristics connected with color—e.g., the number of used colors or colorfulness. For instance, we can use them to evaluate Few's heuristics which advice using a limited number of subtle colors. Objects-based metrics can be used for detection of an inappropriate layout, which is one of the most common mistakes presented by Few. Ngo et al. [2003] designed 13 metrics for measuring aesthetics and analysis of UI layouts.

Measuring object characteristics seems to be a promising approach for improvement of dashboard evaluation. However, we need to deal with ambiguity of object recognition. If we want to improve the realism of evaluation results, we should specify objects concerning the perception of users. Designing such a method requires well understanding of the principles of visual perception.

Chapter 4

Recognition of Visual Components

Vision is the dominant human sense. It allows us to process visible light and transform the signal into information. Since eyes are essential for the capturing of the signal, the brain plays a major role in the transformation of the signal into information. Understanding the principles of visual perception requires understanding how the brain works. Vision is still the aim of many researchers working in a variety of scientific disciplines—e.g., ophthalmology, neuroscience, psychology, or computer science. In contrast to computers, people do not process the visual signal as an image composed of pixels. They perceive a view as objects located within a scene [Baker et al., 2009]. They consider the objects with a different level of detail according to their actual attention. Moreover, every person is unique with a unique brain. Unfortunately, we are not able to predict entirely how a person would interpret the perceived view.

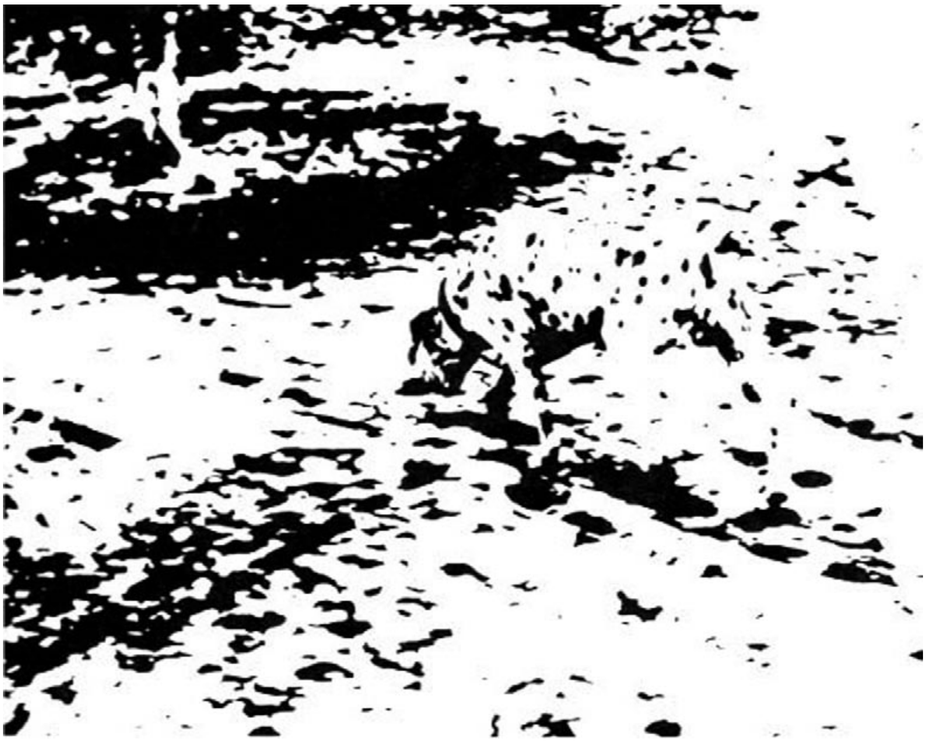


Figure 4.1: What do you see in the figure? Source: [Johnson, 2010; Marr, 2010].

Figure 4.1 demonstrates the unpredictability of visual perception. It consists of the black stains scattered in the white background. At the first glance, viewers would probably not find any meaning in the figure. After they are told it contains a Dalmatian dog sniffing the ground next to a tree, they should find the meaning and start to process the figure in this way. I repeated the experiment several times during the special lessons of the Information Systems course at Brno University of Technology, Faculty of Information with different students. There was usually a student (of the 50-200 students) who found the meaning without any clue and told the others.[1] On the other hand, some students were unable to see the meaning even after the revealing. Van Tonder and Ejima [2000] performed a survey which ended up with various results. Users found different objects in the figure (e.g., ground covered by snow, or various kinds of animals).

The experiment confirmed the fact that people perceive objects subjectively concerning their previous experience. Recognition of objects by a computer is a difficult task, and ambiguity of visual perception makes it even more complicated. This thesis focuses on the segmentation of dashboard screens into regions representing the visually dominant objects which can be used as inputs for object-based metrics. This chapter consists of two parts. The first part describes basic principles of visual perception which should be known for the segmentation of a screen. It focuses on the problem of objects recognition and grouping. The second part presents existing methods for page segmentation. It considers their applicability for the segmentation of dashboards.

4.1 Visual Perception of Objects

In the beginning, a viewer reacts to the *visible light* by visual receptors—the *rod* and *cone* cells—located in eyes' retina. The visible light is represented by the electromagnetic radiation of the wavelength range approximately from 400 to 700 nm [Ware, 2004]. Three types of cone cells detect the three frequencies of the light: lower, medium and higher frequencies (Figure 4.2). The second type of receptors—rod cell— is sensitive to brightness. They are located near the edges of the retina. They are used in peripheral vision and perception of low levels of the light [Johnson, 2010].

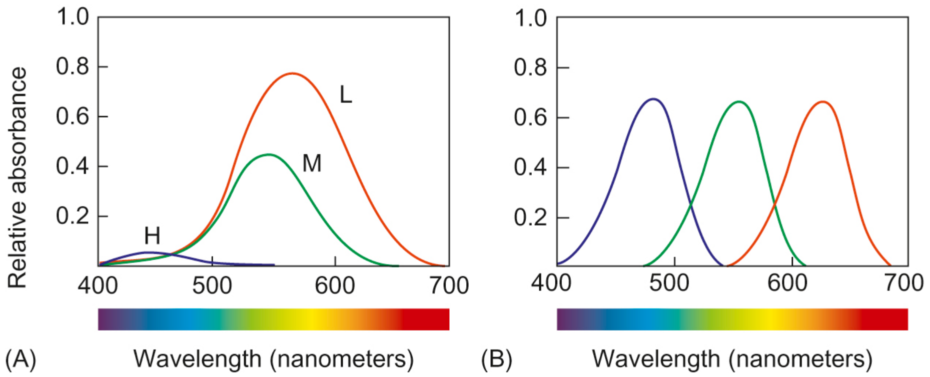


Figure 4.2: Comparison of the sensitivity of the three types of cones (left) with the sensitivity of the artificial red/green/blue receptors. Source: [Johnson, 2010].

[1] The reason might be previous experience with the figure.

The detected light is then transformed into an electrical signal which is transferred to the brain by the optic nerves [Gibson, 1950; Ware, 2004]. The brain perceives the visual signal and constructs an image of the perceived view. It combines the signal detected by the three kinds of cones and interprets color values (*color subtraction*). Then, the brain detects contrasting edges and recognizes basic shapes. Human vision is much more sensitive to the differences in color and brightness than absolute brightness level [Johnson, 2010]. Color of a shape is perceived relatively to surrounding colors (Figure 4.3). Johnson [2010] shows the three presentation factors affecting the ability to distinguish colors from each other—*paleness*, *color patch size* and *separation*. The paler (less saturated), smaller and more separated the two patches are, the more difficult it is to distinguish their colors.

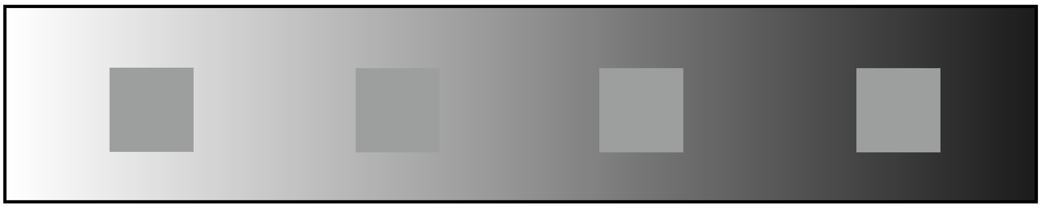


Figure 4.3: The color of the squares is the same. Perception of the color is however affected by the surrounding color. Redrawn from: [Few, 2006].

Perception of colors is subjective. Approximately 10% of population (mostly men) have the problem to distinguish certain colors [Few, 2006; Johnson, 2010]. Few [2006] recommends changing color intensity rather than color hue in a presentation to make sure that all viewers would be able to distinguish colors (Figure 4.4). For this purpose, UI designers can use alternative color models than RGB (red, green, blue). An example is the HSB (HSL) color model represented by hue, saturation, brightness/lightness. Color intensity refers to saturation and brightness/lightness. Another useful color space is the CIE L*a*b* color space (lightness, green-red, blue-yellow), which corresponds to human perception of colors [Ware, 2004].

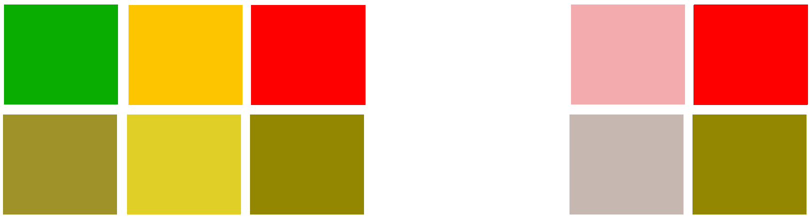


Figure 4.4: A hypothetical example of a color-blind vision (bottom squares). It is better to distinguish colors by varying color intensities (right) rather that color hues (left). Redrawn from: [Few, 2006].

Color is one of the attributes which play a role during the initial recognition of objects and construction of the image. It is done *preattentively* according to preattentive attributes. Preattentive processing is the perceptual task of object recognition which is performed very quickly without the user's attention (in less than 200 ms according to [Healey et al., 1996]). According to Healey et al. [1996], there are 17 preattentively perceived features which can be, according to Ware [2004], classified into the four categories: *color*, *form*, *spatial position*, and *motion* (Figure 4.5). The appropriate usage of the preattentive features can significantly decrease the time of dashboard sensemaking as shown by Few [2006]. Figure 4.6 presents an example of the preattentive processing and compares it with the attentive processing.

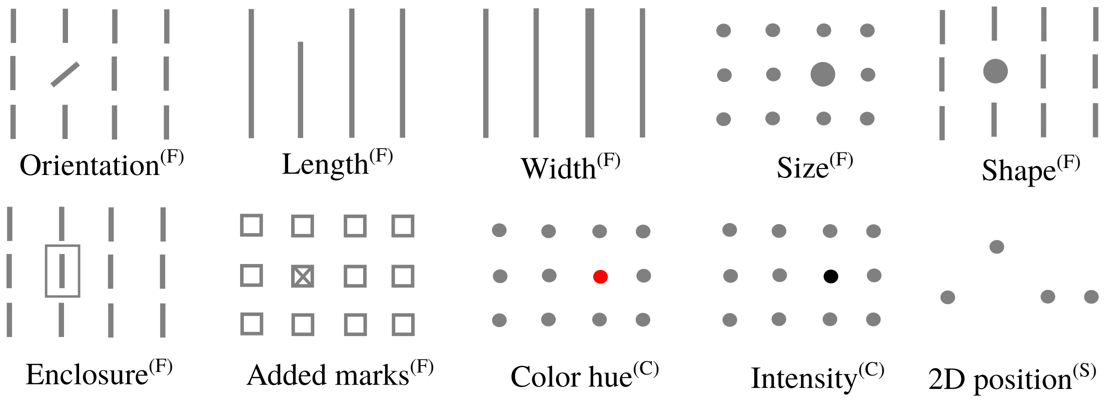


Figure 4.5: Examples of preattentive attributes inspired by [Few, 2006]. Legend: (C)—color, (F)—form, (S)—spatial position. Few also mentions the *flickering* attribute of the motion group. Readers can meet with further preattentive attributes in other literature (e. g., *curvature*, *blur* or *direction of motion*).

2460987245187249 246098724**5**187249
1895278629723678 189**5**278629723678
1672375323785137 167237**5**32378**5**137
2509012112523089 2**5**09012112**5**23089

Figure 4.6: The figure demonstrates the difference between the preattentive and attentive processing. It is easier to count the number of the digit '5' in the right side because we can distinguish their different color intensity preattentively. On the contrary, we need to process the digits in the left side attentively. Based on the example presented in [Few, 2006].

After the initial recognition of objects, the brain tries to comprehend the recognized objects, organize them and add meaning to them. Baker et al. [2009] call this process *sensemaking*. He explains it as "the ability to comprehend complex information, assimilate it, create order from it, and develop a mental model of the situation as a precursor to responding to the situation." Only a fraction of what a viewer focuses on is also the object of the viewer's attention [Few, 2006]. This fact corresponds with the limited capacity of the brain's short-term memory, which stores the objects of the actual focus of attention. Few presents the size of the short-term memory between 3 and 9 items, but we can find different interpretations—e.g., 3 - 5 items according to [Johnson, 2010].

According to Baker et al. [2009], a visual representation improves sensemaking in data exploration tasks when:

- It supports **the four basic visual perceptual approaches**:
  - *Association*: viewers are able to note similarities between objects and put them in the same group.
  - *Differentiation*: viewers are able to note differences between objects and put them in separate groups.
  - *Ordered perception*: viewers are able to compare objects with respect to noted differences of objects.
  - *Quantitative perception*: viewers are able to quantify the differences between the chosen characteristic of objects.
- It has **strong Gestalt properties**: viewers are able to find patterns supported by Gestalt properties.
- It is **consistent with viewers' knowledge**: viewers are able to associate the perceived view with previous experience.
- It supports **analogical reasoning**: viewers are able to associate the perceived view with a similar problem.

The detection of the differences and similarities between the perceived objects plays a role in object ordering and grouping. Since viewers can focus on a limited number of objects, they preattentively cluster simple graphical objects into larger visual groups and simplify the view. This fact was described by Gestalt psychology in the early 20th century [Koffka, 1922; Wertheimer, 1923; Köhler, 1925]. It presents laws describing the principles of object recognition and grouping—e.g.:

- **The law of simplicity** (Prägnanz or Good Gestalt): viewers interpret the view in the simplest form. This is the fundamental Gestalt law.
- **The law of figure/ground**: viewers separate the view into a figure (foreground) and ground (background). Foreground represents objects of the primary attention.
- **The law of proximity**: viewers group the objects which are located near to each other.
- **The law of similarity**: viewers group the objects which are similar (e.g., similar color, shape, or size).
- **The law of enclosure**: viewers group the objects which are enclosed by a border.
- **The law of closure**: viewers have a tendency to close and complete objects which are incomplete (e.g., objects containing hidden parts).
- **The law of continuity**: viewers group the objects which are aligned in a continuous direction.
- **The law connection**: viewers group the objects which are connected in some way (e.g. line).

- **The law of symmetry**: viewers tend to perceive objects symmetrical.

Figure 4.7 presents examples of Gestalt laws. Figure 4.8 points out possible ambiguity of object grouping. The list of laws is not complete and readers can find slightly different enumerations in different literature [Few, 2006; Johnson, 2010; Koffka, 2013; Ware, 2004].

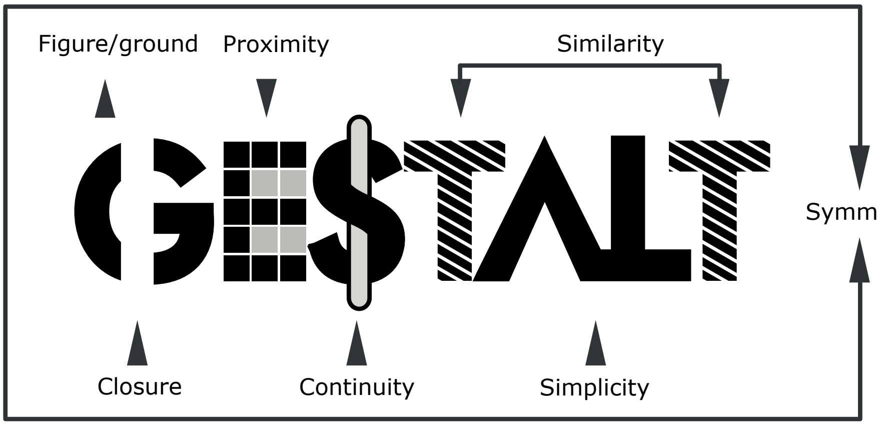


Figure 4.7: Examples of Gestalt laws. Author: Valessio S. Brito.[2]

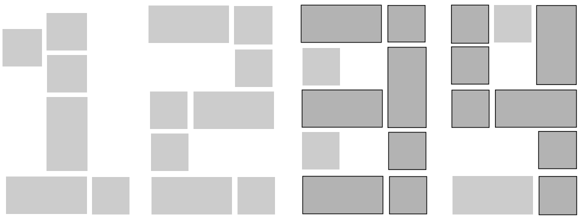


Figure 4.8: Ambiguity of objects grouping. A viewer can see the two possible sequences of numbers: "1234" or "1289" (explained by the Gestalt laws of proximity and similarity).

The problem of Gestalt laws is that they miss a mathematical model. Their quantitative description is still the aim of researchers [Jäkel et al., 2016]. This complicates conversion of the laws into computer algorithms which would automatically predict how a user perceives a displayed screen. Every viewer can process a different number of items at the same time. Orlov et al. [2016] performed an eye tracking study to analyze the effect of change of the number of objects in a dashboard on the perception of the dashboard. Also, we need to consider the subjective perception. Every viewer has a different experience, which also affects the visual perception [Johnson, 2010].

Visually emphasized objects together with background elements (larger scale, solid surfaces, and structures) make a scene of visual representation [Henderson and Hollingworth, 1999]. Every object within a scene can be described by its visual characteristics [Baker et al., 2009]. An appropriate choice and arrangement of objects within a scene are crucial for the interpretation of data by the viewer. They can emphasize various relations between the data, yet they can skew or hide other facts (examples in [Tufte, 2001]). Hence, an analysis of object characteristics within a scene can be useful during the design phase of a dashboard or user interface in general.

[2] Source: Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Gestalt.svg. The figure was translated to English.

4.2 Page Segmentation

Page segmentation is the important part of document processing and understanding. The goal of page segmentation is to divide a document page into coherent parts which can be classified and analyzed by further analyses. According to [Kise, 2014], page segmentation is “a task of extracting homogeneous components from page images.” Kise [2014] considers components as text blocks or zones, text-lines, graphics, tables, and pictures. Page segmentation is usually used for digitization of printed documents or analysis of web pages. The usual reason we want to segment a page is to analyze its content, appearance, and usability.

Since we can store a page in different kinds of media (electronic or printed media), there are different approaches to process the page. Printed documents need to be scanned, so we process them as raster images. On the contrary, web pages are represented by structural description. We need to use a browser to render their *Document Object Model* (DOM) and find the nodes representing coherent parts of the page.

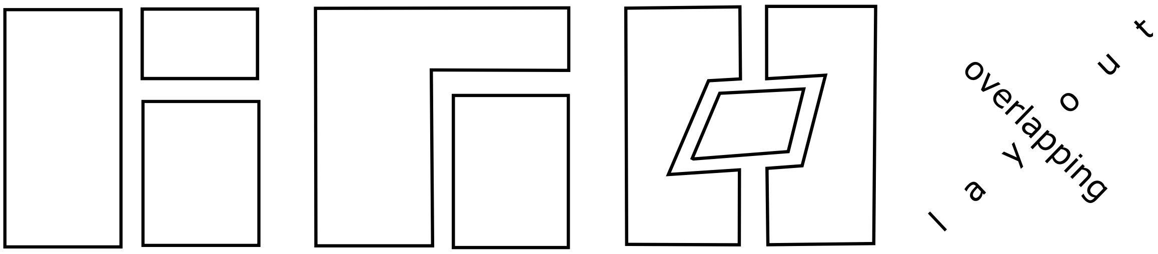
Segmentation of the pages represented by a structural description does not require to perform image processing methods (image preprocessing or OCR—*optical character recognition*). There is also no loss of quality caused by capturing of the raster image. On the other hand, a web page can contain dynamic content (JavaSript, AJAX), and some nodes can be invisible. It might be much more difficult to render the page since the result highly depends on the resolution of the screen and the browser interpreting the source code. Readers can find methods for the web page segmentation in [Burget, 2017; Feng et al., 2016].

Segmentation of the pages represented by a raster image focuses more on the way the page is presented to users than how the page is implemented. It analyzes and understands what is actually presented to users and therefore, it can predict better what is seen by the users. It, however, depends on the quality of a captured image. Applying image processing methods is usually more difficult than processing structural description. The image needs to be preprocessed and simplified.

Segmentation of raster images has been the aim of many researchers especially because of the rising need for computer processing and archiving of printed documents. Researches have developed many different methods for this purpose. Mao and Kanungo [2001]; Shafait et al. [2006] provide a methodology for performance comparison of segmentation methods. They compare the most famous methods. A comprehensive description of document image processing and recognition can be found in the book [Doermann et al., 2014]. Kise [2014] presents a thorough classification of segmentation methods according to the different attributes: *page layout*, *objects of analysis*, *primitives of analysis*, and *strategy of analysis*.

Page layout can contain *non-overlapping* and *overlapping* page elements. Kise [2014] distinguishes the four layout types (Figure 4.9):

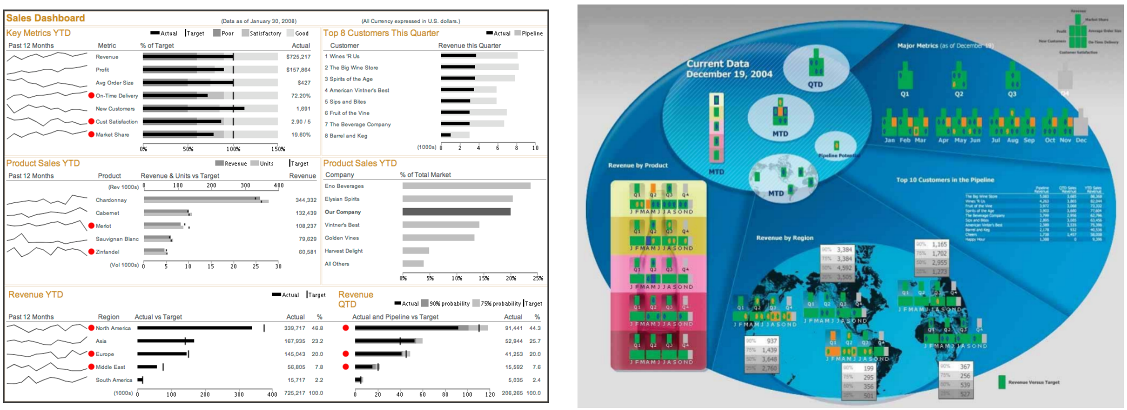
- *Rectangular layout*: the borders of page elements are represented by non-overlapping rectangles whose sides are parallel or perpendicular with the borders of the page.
- *Manhattan layout*: the borders of page elements are represented by non-overlapping shapes whose sides are parallel or perpendicular with the borders of the page.
- *Non-Manhattan layout*: the borders of page elements are represented by non-overlapping shapes.
- *Overlapping layout*: the borders of page elements are represented by overlapping shapes.



(a) rectangular (b) Manhattan (c) non-Manhattan (d) overlapping

Figure 4.9: Examples of the layout types according to [Kise, 2014].

Analysis of the overlapping layout is significantly more difficult. It uses the extraction of features and classification of page components based on unsupervised or supervised learning. Readers can find several methods for this purpose—e.g., [Jain and Zhong, 1996; Etemad et al., 1997]. There exist dashboards with overlapping elements (Figure 4.10b). The reason might be the need to fit the data into one screen or just exaggerated creativity of the designer. However, it is not common, and dashboards usually contain elements arranged in a simple non-overlapping rectangular or Manhattan layout.

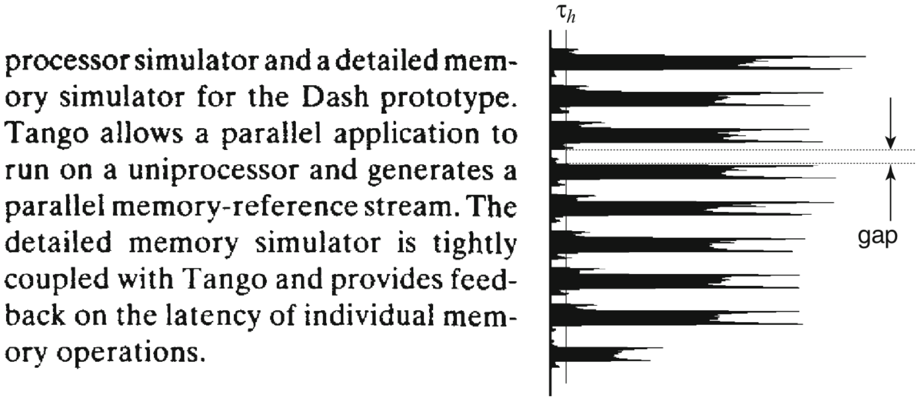


(a) (b)

Figure 4.10: An example of a dashboard with a simple layout using a reduced number of colors (left) and a highly colorful dashboard containing color gradients and overlapping widgets (right). Segmentation of the right dashboard would be more complicated compared to the left one. Source: [Few, 2006].

Objects of analysis specify whether we analyze background or foreground of a page. Printed documents usually consist of a black foreground (e.g., text) and white background, which can be separated by the methods based on image thresholding [Sezgin and Sankur, 2004; Russ, 2016]. On the other hand, dashboards often consist of hierarchically arranged frames, and the background is represented by multiple colors or color gradients (Figure 4.10b). We cannot use simple separation methods, e.g., thresholding. Minaee and Wang [2016] presented an example of advanced method for separation of foreground and background.

Primitives of analysis represent elements of the page *foreground* or *background* processed by a segmentation analysis. We can consider single pixels as primitives, but common segmentation methods usually work with larger groups—e.g., *connected components* (Figure 4.13) or *projection profiles* (Figure 4.11) [Kise, 2014]. This research works with the groups of same color pixels represented by their rectangular boundaries (regions). It uses heuristics to organize the regions in a tree structure representing a page layout.



processor simulator and a detailed memory simulator for the Dash prototype. Tango allows a parallel application to run on a uniprocessor and generates a parallel memory-reference stream. The detailed memory simulator is tightly coupled with Tango and provides feedback on the latency of individual memory operations.

Figure 4.11: An Example of the horizontal projection (right) of a document (right). The projection profile can be used to find vertical gaps between clusters of black pixels. Source: [Kise, 2014].

A page layout consists of a hierarchy of page primitives. There are the two **strategies of the layout processing**:

- The *top-down strategy* starts with a page and divides it into page elements representing leaves of the layout tree. The typical method using the top-down strategy is *Recursive XY-cut* [Nagy and Seth, 1984]. The method uses projection profiles of the page to detect gaps between the foreground pixels and splits the page into regions (Figure 4.12). Readers can find optimization of the method (e.g. [Ha et al., 1995]).

- The *bottom-up strategy* is reversed. It starts with simple primitives of the page (e.g., pixels or groups of pixels) and join them into larger coherent groups. Examples are *connected components*-based methods (e.g. [Simon et al., 1997]) described in Figure 4.13 or *smearing*-based (also *smoothing*-based) methods described in Figure 4.14.

Some methods combine both strategies or starts from the middle of a layout tree (*intermediate strategy*) [Kise, 2014].

There are also other factors which we need to consider—e.g., quality of a document. Since we work with user interfaces, we can assume that the samples can be captured in high quality if needed. For instance, we can convert the dashboards represented as web pages into raster images by using a headless browser (e.g., Phantom.js[3]), which can render a web page screenshot containing a specific resolution.

Finally, we also need to consider the similarity between the segmented samples. Whereas the printed documents are usually very similar, the appearance of dashboards varies in many visual aspects. There exist various dashboard templates using different layouts, widgets, colors, and styles. It complicates to design a universal segmentation algorithm. Figure 4.10 shows an example of the variability of dashboard samples.

[3] Phantom.js project's website: http://www.phantomjs.org

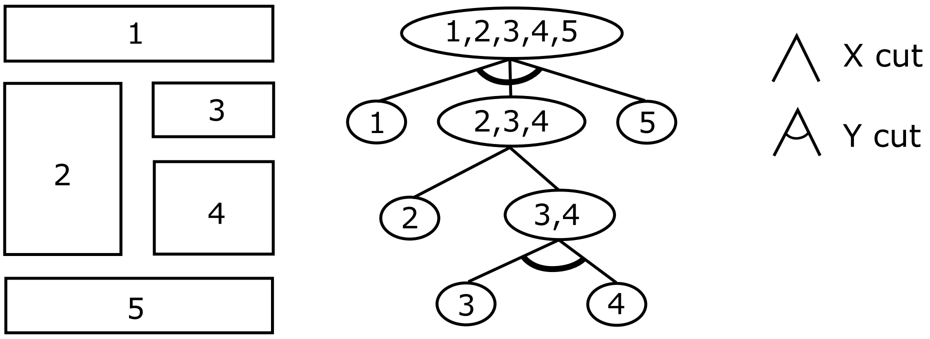


Figure 4.12: An example of the Recursive XY-Cut algorithm. It divides the page vertically and horizontally into regions using the top-down strategy. Redrawn from [Kise, 2014].

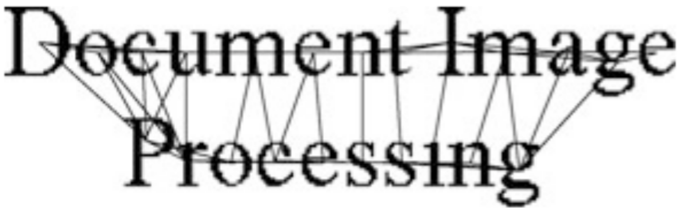


Figure 4.13: An example of connected components using k-nearest neighbor algorithm (the bottom-up strategy) [Kise, 2014].

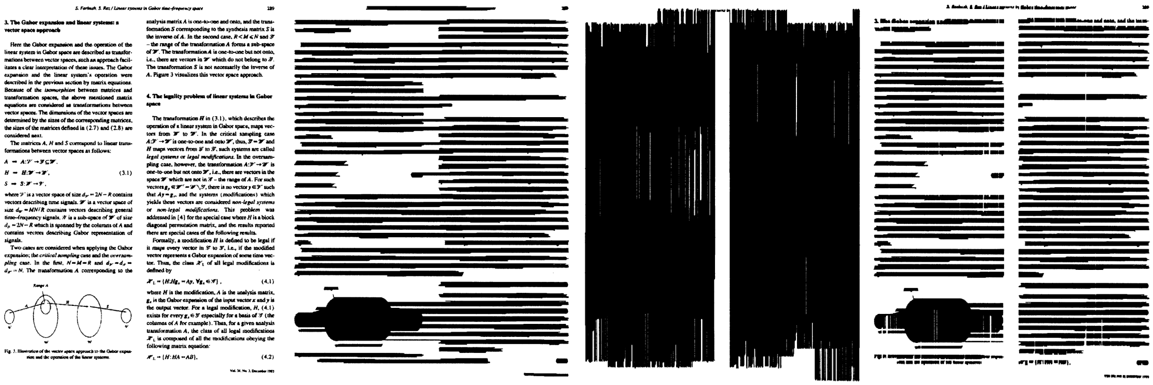

(a) (b) (c) (d)

Figure 4.14: An example of RLSA (*run length smearing algorithm*) described by [Wong et al., 1982]. It applies horizontal (a) and vertical (b) run-length smearing to connect the black pixels of the original page (a). Then, it performs the AND operation to split the smeared pages into regions (d). It combines the bottom-up and top-down strategies. Source: [Yin, 2001].

4.3 Summary

Visual perception is a complex process which is difficult to simulate. Evolution has made visual perception work with different kinds of scenes. Eyes are able to quickly adapt for a different level of lightness and focus on a specific point. The brain is able to simplify the view and imagine missing parts of the scene as explained by Gestalt laws. It is difficult to formally describe the principles of visual perception. The description can not be completely generalized. There are people having problems to distinguish certain colors. We also need to consider subjectivity of perception—e.g., a different size of the short-term memory, or different experience of a viewer. Two viewers might perceive a scene differently.

On the other hand, image segmentation methods are designed for a specific purpose (e.g., archiving of printed documents). Their ability to recognize objects within a scene can be more efficient, but limited. They are usually trained to work only with specific kinds of images. Examples are the page segmentation methods. They are a group of image processing methods whose goal is to segment a page into regions which could be processed by further analyses (e.g., OCR methods). They work well with printed documents. They separate the black foreground from the white background and look for regions representing text and figures using the top-down or bottom-up strategy. They could be used for segmentation of simple user interfaces.

Dashboards usually contain complex widgets and charts which makes them more difficult to segment. In contrast to printed documents, dashboards consist of a hierarchy of frames using different colours. Sometimes, widgets overlap each other. It is much more challenging to consider the principles of human perception (e.g., Gestalt laws describing the principles of object grouping). Application of page segmentation methods for preparation of inputs for object-based metrics measuring dashboard quality is questionable. The methods, however, represent a very good basis and inspiration for design of novel techniques usable for segmentation of advanced user interfaces.

Chapter 5

Decomposition of Problem

The previous chapters presented state of the art regarding the three issues:

1. **Dashboards, their characteristics, applications, components and the process of design:** It presented examples of frequently made design problems and showed that there is the need for an automatic approach of the evaluation of selected design problems during the design phase.
2. **Evaluation of UI quality:** It focused on the guideline reviews based on quantitative metrics suitable for automatic evaluation of UIs. It presented pixel-based and object-based metrics and considered the two main problems:
   (a) simplicity of pixel-based metrics, which are usually unable to measure advanced visual characteristics
   (b) ambiguity of object recognition, which is essential in preparing inputs for object-based metrics
3. **Recognition of objects:** It provided a brief overview of the recognition of objects by human, presented the process of visual perception and introduced basic principles describing object recognition and grouping (e.g., Gestalt laws). Then, it presented methods for segmentation of printed documents.

The research which is described in this thesis explored the possibility to combine the three issues and create a tool which would be able to:

1. **Load a dashboard:** take a screenshot of a dashboard displayed on a screen of a specific resolution.
2. **Convert the dashboard to an internal representation suitable for further analyses of design quality:**
   (a) Represent the dashboard as a bitmap in a suitable color model.
   (b) Represent the dashboard in a structural description describing the structure of the UI (use a segmentation algorithm to segment the dashboard into coherent regions representing the UI objects automatically—with respect to the human perception; provide tools for the manual description of the UI structure by a user).

3. **Use the internal representation of the dashboard to evaluate the design quality of the dashboard:**
   (a) Use the raster representation of the dashboard as the input for the pixel-based metrics which are suitable for measuring dashboard characteristics corresponding to its design quality.
   (b) Use the description of objects as the input for the object-based metrics which are suitable for measuring dashboard characteristics corresponding to its design quality.

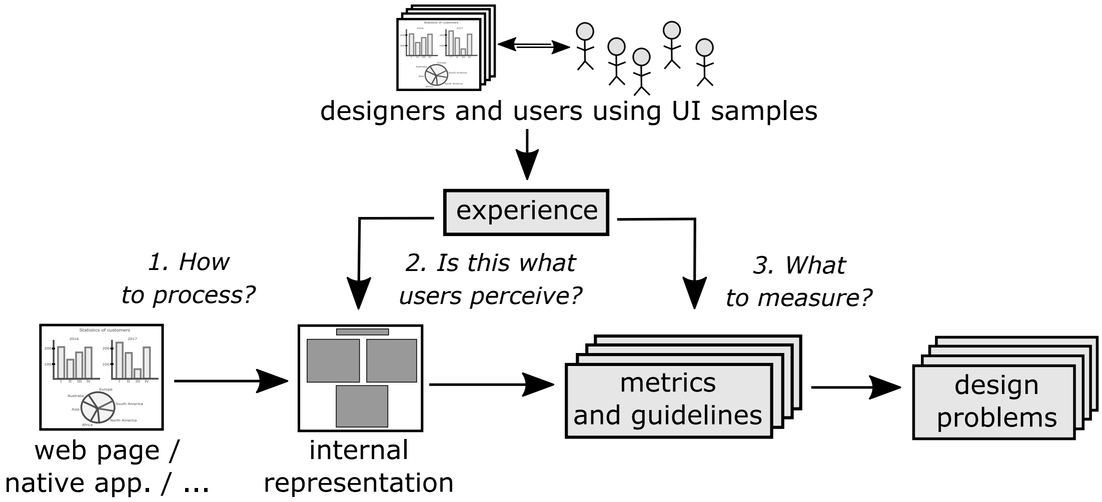


Figure 5.1: The process of UI evaluation and the main problems.

Figure 5.1 describes the process of the analysis of a dashboard. The process contains the following problematic parts:

1. The original dashboard can be represented in various formats and implemented in different technologies.
2. The result of the dashboard segmentation into regions representing UI objects should reflect the perception of the objects by users. The solution should deal with the subjective visual perception and principles of objects grouping (e.g., Gestalt laws).
3. The metrics should measure design characteristics which correspond to design quality and supports usability of dashboards. The solution should use such metrics which help distinguish well-designed samples from poorly designed ones and consider the subjective perception of users. It requires to find a sufficiently large test set of dashboard samples.

This chapter describes tasks which provide solutions to the problems. It provides a model which defines the internal representation of dashboard. Then, it introduces software which works with the internal representation of dashboards and helps to solve the problems. Finally, the chapter presents test samples which are used for evaluation of the proposed solutions.

5.1 Research Tasks

The research was split into the following tasks, which are represented as single chapters:

1. specification of a model, implementation of software, and preparation of test samples described in the following sections of this chapter
2. analysis of pixel-based metrics described in Chapter 6
3. analysis of object-based metrics described in Chapter 7
4. design and improvement of metrics described in Chapter 8
5. automatic segmentation of dashboards described in Chapter 9
6. comparison of metrics with user reviews described in Chapter 10

5.2 Model

As presented in Section 3.3, we can analyze dashboards from the two perspectives: the pixel-based and object-based perspective. For this purpose, a model of dashboard was created. It defines the structure of internal representation of dashboards suitable for the evaluation of the dashboards by pixel-based and object-based metrics. The results presented in this section were published in [Hynek and Hruška, 2015]. In contrast to this publication, the following text contains improved terminology and descriptions. It also provides additional information regarding the practical application of the model, which was researched later after the release of the publication.

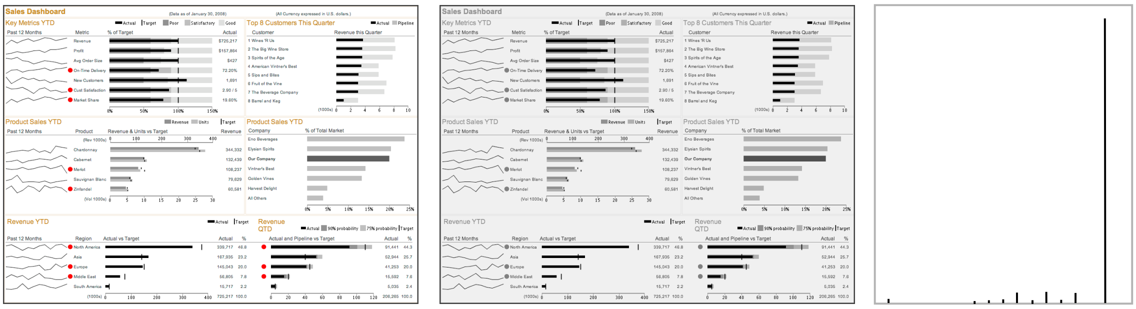
5.2.1 Pixel-based Representation

The first perspective represents a dashboard as a bitmap—a matrix of pixels which contain color values in a selected color space. The size of the matrix is defined by the pair: $(width, height)$, which indicates the image resolution. The RGB color space is the primary color space which is used to store dashboards. Bitmaps are then transformed into other models to reflect human perception better. The following list presents representations of dashboard bitmaps which are used in this research:

- **RGB bitmap:** all pixels of a bitmap are represented in the RGB color space, usually as 24-bit numbers (8 bits for every red/green/blue channel).
    - **Posterized n-bit RGB bitmap:** the bit width of all color channels is reduced from 24 bits to n bits using posterization ($\frac{1}{3}n$ is an integer representing the bit width of every channel including alpha).
    - **n-bit RGBa bitmap:** 24-bit RGB values are usually stored as 32-bit integers (8 additional bits are reserved for the alpha channel representing the degree of transparency. The alpha channel is, however, not used in this research. It is always set to the maximal non-transparent value. Hence, the following text prefers the “24-bit RGB” notation against “32-bit RGBa” notation.

- **CIELAB bitmap:** all pixels of a bitmap are represented in the CIE L*a*b* color space. The pixels contain floating point values representing *lightness* (L^*), the green-red (a^*) and blue-yellow (b^*) components. They also contain following derived values:
  - *chroma:* $C^* = \sqrt{(a^{*2} + b^{*2})}$
  - *hue:* $h^\circ = \arctan\left(\frac{b^*}{a^*}\right)$
  - *saturation:* $(s_{ab}) = \frac{C^*}{L^*}$
- **HSB bitmap:** all pixels of a bitmap are presented in the HSB color space. The pixels contain values representing *hue* (h), *saturation* (s) and *brightness* (b).
- **Grayscale bitmap:** all pixels of a bitmap are represented in the grayscale color space, usually as 8-bit number (0 represents the black color, 255 represents the white color). Since the grayscale color space does not contain many colors and the human vision is sensitive to the differences in brightness, it was useful to work with this representation in many cases (e.g., analysis of histograms). See Figure 5.2.
  - **Posterized n-bit Grayscale bitmap:** the bit width of all pixel values is reduced from 8 bits to n bits using posterization.
  - **Grayscale histogram:** a histogram of 256 values representing the number of occurrences of all 256 values of the grayscale color space used in the bitmap.
- **Black-and-white bitmap:** all pixel of a bitmap are represented in the black-and-white color space. The pixels are represented by a logical value indicating the presence of the black or white color. There are multiple possibilities of how to convert a bitmap into the black-and-white color space.
  - **Black-and-white bitmap(n):** a Grayscale bitmap is converted into the black-and-white color space using a threshold n.
  - **Adaptive Black-and-white bitmap:** a Grayscale bitmap is converted into the black-and-white color space using adaptive thresholding presented by Bradley and Roth [2007].

The pixel-based representations of dashboard are used as the inputs for pixel-based metrics (Chapter 6) and the method for segmentation of dashboards (Chapter 9).



(a) 24-RGB bitmap **(b)** Posterized 4-bit Grayscale bitmap **(c)** Histogram of (b)

Figure 5.2: A dashboard represented in various bitmaps. Source of the dashboard: [Few, 2006].

5.2.2 Object-based Representation

The second perspective considers a dashboard as a set of objects arranged on a screen which present data. The strategy of the object-based evaluation is to analyze the arrangement of objects on a screen. For this purpose, a simple theory was established. It defines the three levels of dashboard description: *model of dashboard's components*, *dashboard template*, and *dashboard sample*.

Model of dashboard's components M is a set of dashboard's components c (e.g., graphical elements, widgets) which can be contained by a dashboard. A dashboard's component c is defined as a quadruple (T_c, A_c, S_c, X_c), where:

- T_c: a type of the dashboard's component c representing the shape and implicit appearance of the component.
- A_c: actions of the dashboard's component c which can be performed by a user. The actions represent behavior of the component.
- S_c: a finite set of style attributes $s_c = (k_{s_c}, V_{s_c})$ of the dashboard's component c, where:
  - k_{s_c} is a unique key and identifier of the style attribute s_c,
  - V_{s_c} is a set of allowed values of the attribute s_c.
- X_c: a finite set of data dimensions $x_c = (k_{x_c}, V_{x_c})$ which can be displayed by the dashboard's component c, where:
  - k_{x_c} is a unique key and identifier of the data dimension x_c,
  - V_{x_c} is a set of allowed values of the data dimension x_c.

Dashboard template $d_M = (M, \chi)$ is a pair composed of a model of dashboard's components M and a set χ composed of pairs (c, S) representing a dashboard's component $c \in M$ arranged according to styles S which are represented by a set of pairs (k_{s_c}, v), where:

- k_{s_c} is an existing identifier of the style attribute $s_c \in S_c$ of the dashboard's component $c \in M$ (e.g., `background-color`).
- $v \in V_{s_c}$ is an allowed value assigned to the style attribute $s_c \in S_c$ of dashboard's component $c \in M$ (e.g., `blue` or `0000FF`).

Dashboard sample $d_M^{(i)} = (M, \psi)$ is a pair composed of a model of dashboard's components M and a set ψ composed of triples (c, S, X) representing a dashboard's component $c \in M$ arranged according to styles S and presenting a set X of data values val, where:

- $val = (i_{x_c}, v)$ represents multidimensional value composed of a set of pairs (k_{x_c}, v):
  - k_{x_c} is an existing identifier of the data dimension $x_c \in X_c$ of the dashboard's component $c \in M$.
  - $v \in V_{x_c}$ is an allowed value connected to the data dimension $x_c \in X_c$ of dashboard's component $c \in M$.

Then, the set: $\{\{(x,1),(y,2)\},\{(x,2),(y,4)\}\}$ represents two 2-dimensional values, which can be displayed in a 2-dimensional chart (e.g., scatter plot—Section 2.5).

The theory corresponds to the three elements of a dashboard: dashboard components (graphical elements, widgets), description of styles and data (Figure 5.3). This approach respects the MVC (*model-view-controller*) architecture. It is usually useful to separate the data (*model*) and description of style (*view*) from the logic handling the user actions and rendering dashboard components (*controller*).

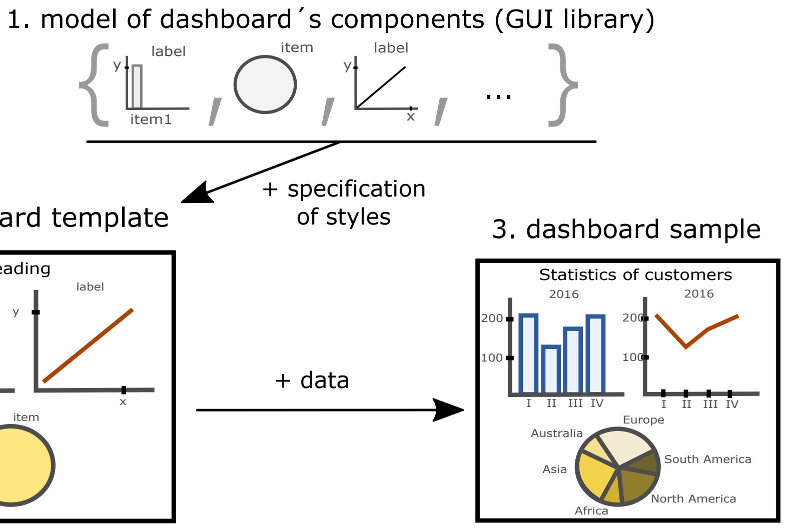


Figure 5.3: In the beginning, we have a GUI library which provides a set of reusable widgets (model of dashboard's components). Then, we use and arrange the widgets in a UI (dashboard template). Finally, we provide data (dashboard sample).

The theory is language-independent. This research implements the description of style and data in the XML language. The following listings demonstrate a simplified example of the description of one graphical element and dataset:

Listing 5.1: Description of dashboard's component with its style.

```
<graphicalElement>
  <type>SCATTER_PLOT</type> <!-- refers existing type of dashboard's -->
  <x>0</x> <!-- X coordination of the graphical element in pixels -->
  <y>0</y> <!-- Y coordination of the graphical element in pixels -->
  <width>200</width> <!-- width of the box in pixels -->
  <height>100</height> <!-- height of the box in pixels -->
  <style>
    <!-- ... -->
    <!-- XML elements describing style -->
    <!-- (according to the interface defined by a controller) -->
  </style>
</graphicalElement>
```

Listing 5.2: Description of dataset.

```
<dataset>
  <values>
    <value>
      <title>My dataset</title>
    </value>
    <value>
      <x>1</x>
      <y>2</y>
    </value>
    <value>
      <x>2</x>
      <y>4</y>
    </value>
    <!-- ... list of values -->
  </values>
</dataset>
```

Then, the controller is an algorithm which provides a model of dashboard's components (declaration of widgets, their behavior, and interface for setting style and data). It processes a description of style and data, maps the data into the dashboard's components and handles actions of users (Figure 5.4). This research uses the JavaScript and Java languages to implement instances of the controller.

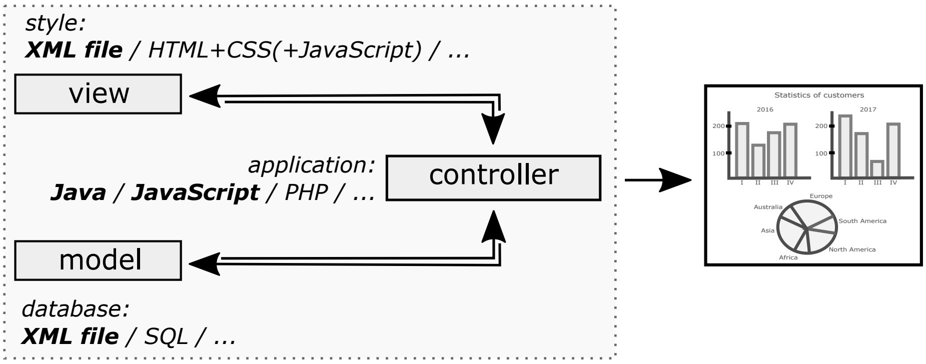


Figure 5.4: Comparison of the theory with the MVC architecture.

In some cases, it might be useful to allow the dashboard's components to contain nested components since it corresponds with the structure of web pages better. Then, a dashboard template would be specified recursively as $d_M = (M, \chi)$, where χ is a set of triples (c, S, χ') containing a set of nested arranged dashboard's components χ'. The χ' set can be empty. Similarly, a dashboard sample would be defined as: $d_M^{(i)} = (M, \psi)$, where ψ is a set of quadruples (c, S, D, ψ').

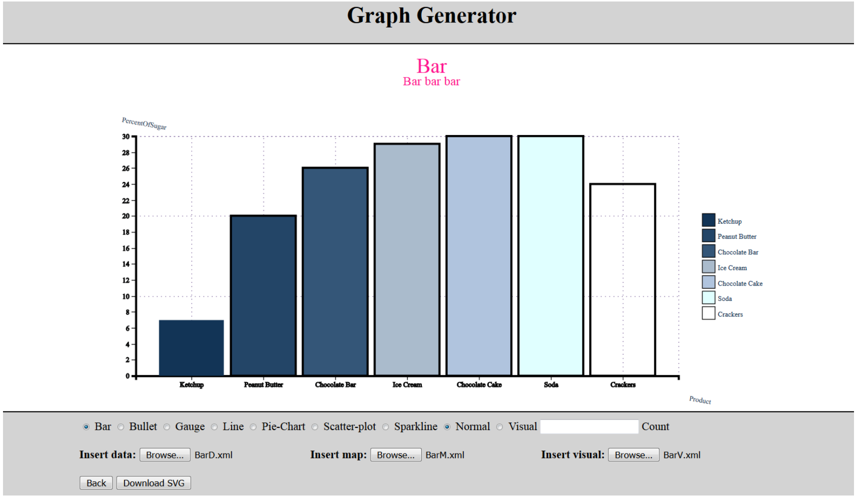
This research works with the internal representation of dashboard which contains one root dashboard's component represented by the `<dashboard>` XML root element. This component represents a dashboard screen. Then, the component contains a list of nested components representing the top-level graphical elements:

Listing 5.3: The simplified description of a dashboard and regions of graphical elements without a definition of style and dataset.

```
<dashboard>
  <x>0</x> <!-- horizontal offset -->
  <y>0</y> <!-- vertical offset -->
  <width>1280</width> <!-- width of screen -->
  <height>1024</height> <!-- height of screen -->
  <graphicalElement>
    <type>CHART</type>
    <x>10</x>
    <y>10</y>
    <width>200</width>
    <height>100</height>
  </graphicalElement>
  <!-- further graphical elements ... -->
</dashboard>
```

This research does not use the full capabilities of the theory. It works with a simplified model of dashboard's components. The internal representation of dashboard, however, represents a sufficient input for the object-based metrics evaluating the layout of dashboards. Graphical elements use the generic `CHART` type and the information about dimensions of objects (position and size). The internal representation does not work with any dataset, further styles, and nested dashboard's components.

Possible extensions of the style and data descriptions were presented in the bachelor's theses of Barič [2017] and Loginova [2017], consulted with the author of this thesis. They implemented tools for generation of widgets from the description of style and dataset (Fig. 5.5). Their controllers were implemented in the JavaScript language. They provide a set of charts frequently used in dashboards and interface for description of various styles and setting the data (usually 2D data). The model of dashboard's components is described in the DTD language. The controllers were not used in this research.



● Bar ○ Bullet ○ Gauge ○ Line ○ Pie-Chart ○ Scatter-plot ○ Sparkline ● Normal ○ Visual Count

Insert data: Browse... BarD.xml **Insert map:** Browse... BarM.xml **Insert visual:** Browse... BarV.xml

Figure 5.5: Barič's Graph Generator, which works with the object-based model. A user needs to upload XML files containing description of style, data, and data mapping.

5.3 Software

Several applications and tools were used during the research. They were developed in order to create, automatically generate, process and evaluate dashboards. Some parts of the software were developed with the cooperation of students of Brno University of Technology. They implemented the software as practical parts of their bachelor's or master's theses supervised or consulted with the author of this thesis. The following subsection contains detailed information about the software and authorship.

5.3.1 Dashboard Analyzer

Dashboard Analyzer is a Java application which provides tools for processing and analyzing of screenshots of dashboards. It was designed and developed primary by the author of this thesis (besides two extensions provided by the students of Brno University of Technology mentioned in the following text). The source code of the application is available online (see Appendix B.1). The application provides the functionality described in the following paragraphs.

Manual Description of Dashboard Object-based Representation

Users can load a dashboard bitmap and manually specify regions representing boundaries of the dashboard's components. The application provides a graphical editor containing drawing tools, including the undo and redo functions. Dashboards can be zoomed and presented in the fullscreen mode. The attachment tool helps the users to snap regions to the layout grid made by other regions. It helps the users to draw the regions quickly.

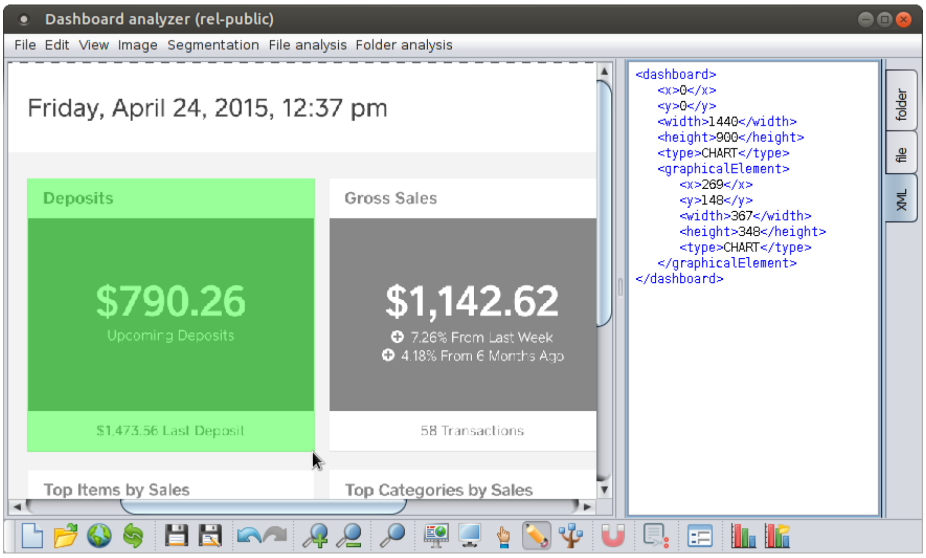
The graphical description of regions is serialized into the XML language, which respects the format of internal representation described in Subsection 5.2.2, Listing 5.3. The application contains a textual editor highlighting the XML language syntax. The content of the textual editor is updated according to the changes made in the graphical editor and vice versa. The serialized description of regions is saved in the XML file located in the same folder as the dashboard bitmap. Figure 5.6 presents an example of the manual description of regions. The functionality was used for the study of users' visual perception of objects described in Section 7.1.

Analysis of Dashboard Samples

A dashboard bitmap and description of regions are inputs for the metrics measuring visual characteristics of dashboards. The application implements several pixel-based and object-based metrics discussed in Chapters 6, 7, and 8. Users can analyze single files thoroughly (*file analysis*) or perform a large-scale analysis of many dashboard bitmaps containing multiple descriptions of regions (*folder analysis*). The application works with workspaces which are represented by folders containing:

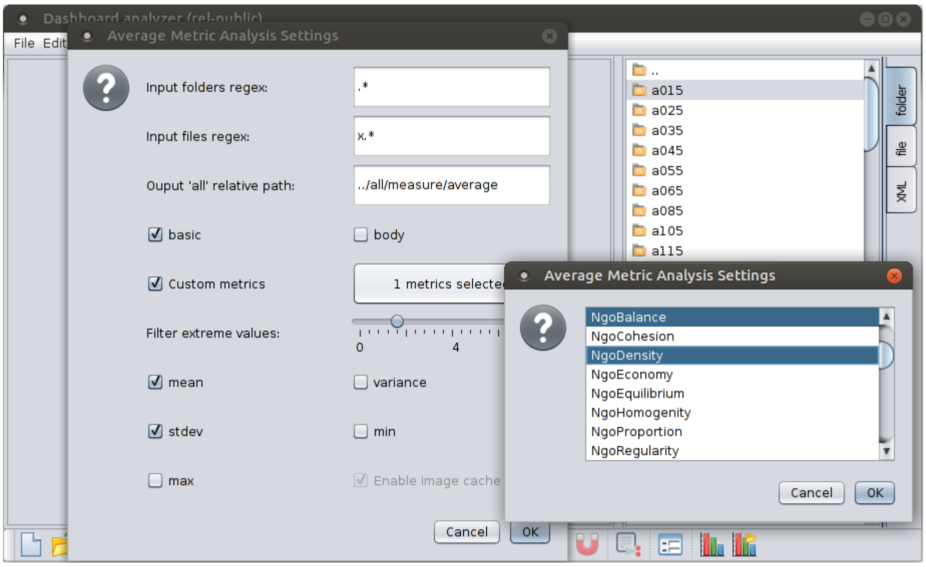
- Single dashboards represented by bitmap files (`.png`, `.jpg`, `.gif`, etc.) or the files containing description of regions (`.xml`).

- Subfolders containing multiple descriptions of regions of one dashboard: They are used in Chapter 7 for the study of visual perception of objects and the analysis of metrics objectivity. Users can use a folder browser to switch between the workspaces (Figure 5.7).



```
<dashboard>
   <x>0</x>
   <y>0</y>
   <width>1440</width>
   <height>900</height>
   <type>CHART</type>
   <graphicalElement>
      <x>269</x>
      <y>148</y>
      <width>367</width>
      <height>348</height>
      <type>CHART</type>
   </graphicalElement>
</dashboard>
```

Figure 5.6: An example of the description of regions using the Java application. The green area represents a selection of a visual region drawn by a user. The XML source code presented on the right is re-generated with every change of the regions in the canvas. This example contains a description of the dashboard and one region.


Dashboard analyzer (rel-public)
File Edit
Average Metric Analysis Settings
..
a015
a025
a035
a045
a055
a065
a085
a105
a115
folder
file
XML
Average Metric Analysis Settings
Cancel
OK
Cancel
OK

Input folders regex: .*

Input files regex: x.*

Ouput 'all' relative path: ../all/measure/average

☑ basic ☐ body

☑ Custom metrics 1 metrics selecte

Filter extreme values: 0 4

☑ mean ☐ variance

☑ stdev ☐ min

☐ max ☑ Enable image cache

NgoBalance
NgoCohesion
NgoDensity
NgoEconomy
NgoEquilibrium
NgoHomogenity
NgoProportion
NgoRegularity

Figure 5.7: The dialogs for selection of metrics and specification of pararameters of the analysis. A user can use regular expressions to choose subfolders and files which should be analyzed.

Dashboard Analyzer provides the API for implementation of new metrics. A programmer can declare parameters of metrics, which are registered by the UI of the application. Then, a user of Dashboard Analyzer can specify values of the parameters, which will be used during the evaluation of metrics (Figure 5.7). It can help the user to analyze the metrics and find optimal parameters suitable for evaluation of specific kinds of user interfaces.

Image Processing Tools

The application provides tools for simple operations with bitmaps. Examples of the operations are:

- reduction of colors: conversion to Grayscale bitmap, posterization, thresholding
- smoothing or sharpening: Median filter, Laplacian filter
- detection of edges: Sobel operator, Hough transform
- detection of areas having the same colors

The tools were used for the preparation of dashboard bitmaps used in the analysis of pixel-based metrics (Chapter 6) and the design of the method for segmentation of dashboards (Chapter 9).

Tools for Segmentation of Dashboards

The application provides the API for implementation of new segmentation algorithms. The API allows a programmer to bind a segmentation algorithm to the application's UI similarly as it is done by the API for the integration of new metrics. The programmer can debug the algorithm comfortably. This research focused on the design of a new method for segmentation of dashboard screens (Chapter 9). The method was implemented and integrated into the application.

Possibilities of dashboard segmentation were also investigated in Santiago Mejía's master's thesis supervised by the author of this thesis [Mejía, 2018]. Mejía provided a few tools, which were used by the bottom-up analysis of the method for dashboard segmentation (Subsection 9.2.7).

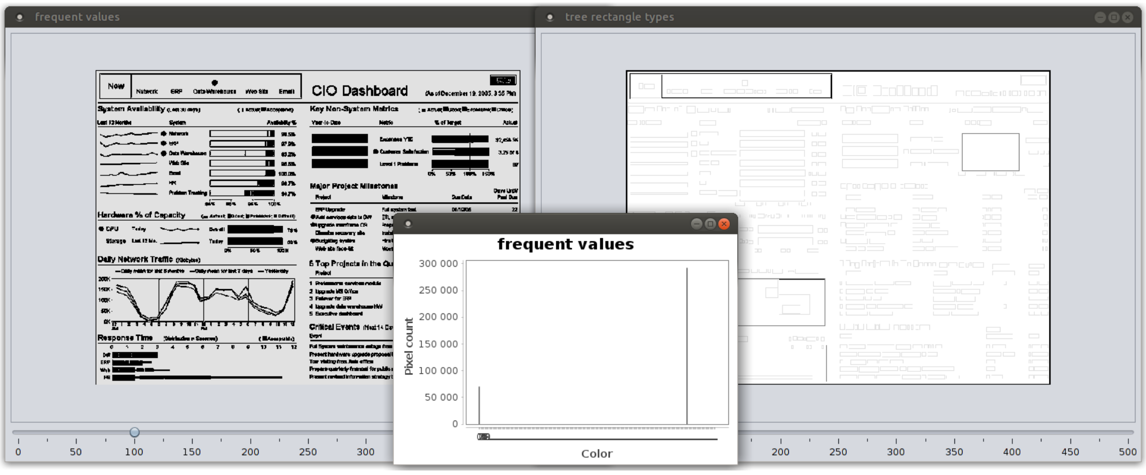
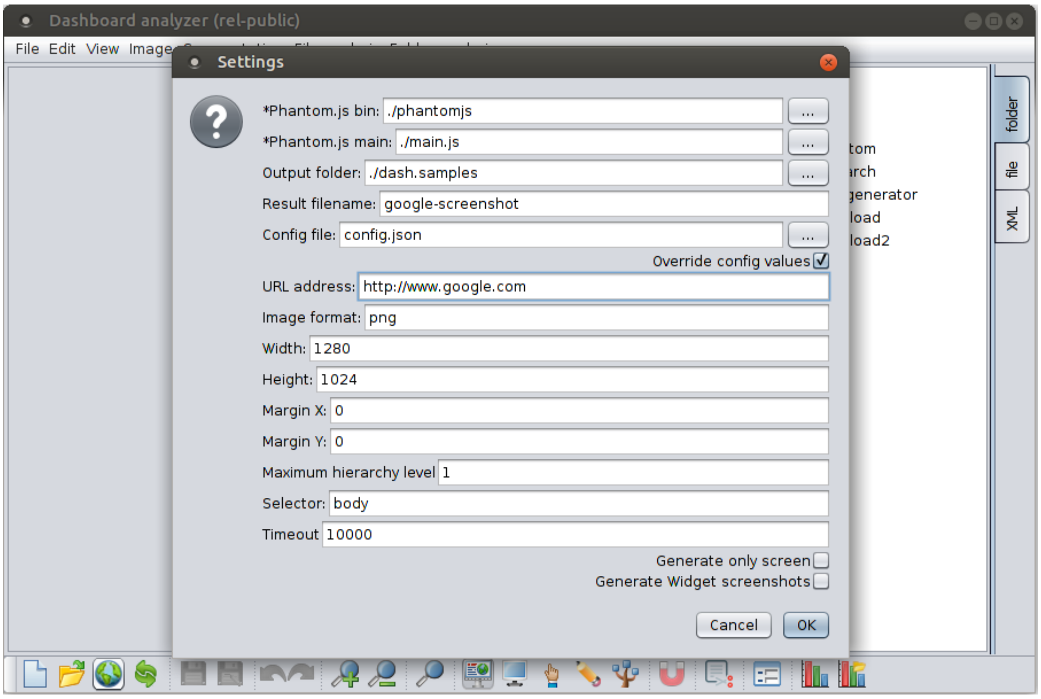


Figure 5.8: Dialogs for debugging of methods for segmentation of dashboards. They visualize single steps of the segmentation (e.g., processed bitmaps, regions, or histograms).

Webpage Download Tool

The application provides the support for downloading dashboards which are represented in the form of a webpage. The application contains a UI extension for a command-line tool designed by Adriana Jelenčíková as a part of her bachelor's thesis consulted with the author of this thesis [Jelenčíková, 2018]. The tool was implemented in the JavaScript language. It uses the PhantomJS[1] headless browser to render webpages defined by a URL. Then, it takes a screenshot and converts the webpage code into the XML internal representation described in Subsection 5.2.2. Users can specify parameters—e.g., the size of the screen, margins, or level of nesting in the DOM of the analyzed web page. (Figure 5.9).



*Phantom.js bin: ./phantomjs
*Phantom.js main: ./main.js
Output folder: ./dash.samples
Result filename: google-screenshot
Config file: config.json
Override config values ☑
URL address: http://www.google.com
Image format: png
Width: 1280
Height: 1024
Margin X: 0
Margin Y: 0
Maximum hierarchy level 1
Selector: body
Timeout 10000
Generate only screen □
Generate Widget screenshots □

Figure 5.9: Dialog for specification of a webpage URL and parameters of downloading.

5.3.2 Generator of Dashboard Samples

One of the problems of this research was to find suitable dashboard samples which could be used for analyses of specific situations (e.g., specific layouts, colors or dashboard components). It was needed to create realistic-looking dashboard samples including structural descriptions of dashboard's components quickly and effortlessly. The main requirements were:

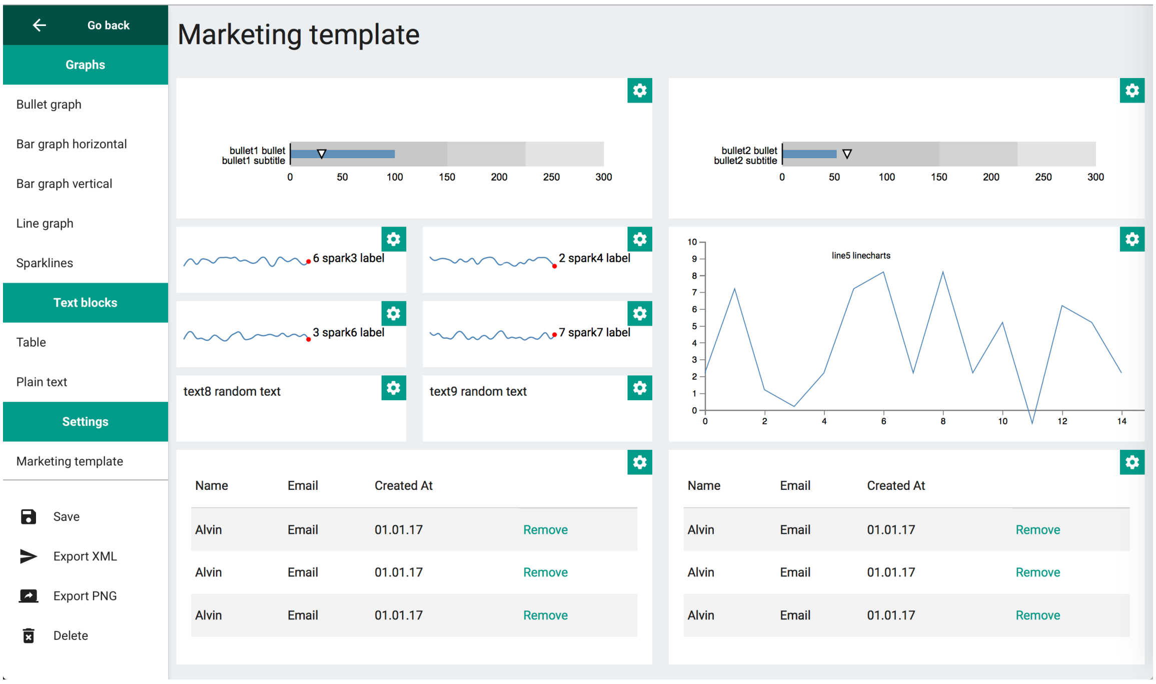
- the possibility to specify the structure and appearance of the UI samples effortlessly, quickly, in a declarative way and without the knowledge of implementation details of the tool. The XML format presented in Subsection 5.2.2 was preferred. This requirement is called ***ease of use*** in this thesis.

- the possibility to specify only a subset of the significant UI characteristics and let the rest of the characteristics to be set implicitly (visualization of charts without the need to specify a dataset). The requirement is called ***simplicity***.

[1] PhantomJS project's website: http://phantomjs.org/

- the possibility to change the values of specified attributes simply, so that we can create multiple instances of the same UI varying in a specific UI characteristic. The requirement is called ***flexibility***.
- the possibility to add new widgets, update the model or modify the tool according to actual purposes. The requirement is called ***extensibility***.

Existing commercial dashboard builders (e.g., Klipfolio, Datapine, Sesense, or theDash[2]) do not usually allow to modify the tool or to export the structural description of dashboard components (or in a limited way). The required tool for generation of dashboards was developed by Olena Pastushenko as a part of her master's thesis supervised by the author of this thesis [Pastushenko, 2017]. The generator was used for evaluation of the impact of color on object-based metrics (Section 8.2). The results were published by Pastushenko, Hynek and Hruška [2018, 2019].



text8 random text

text9 random text

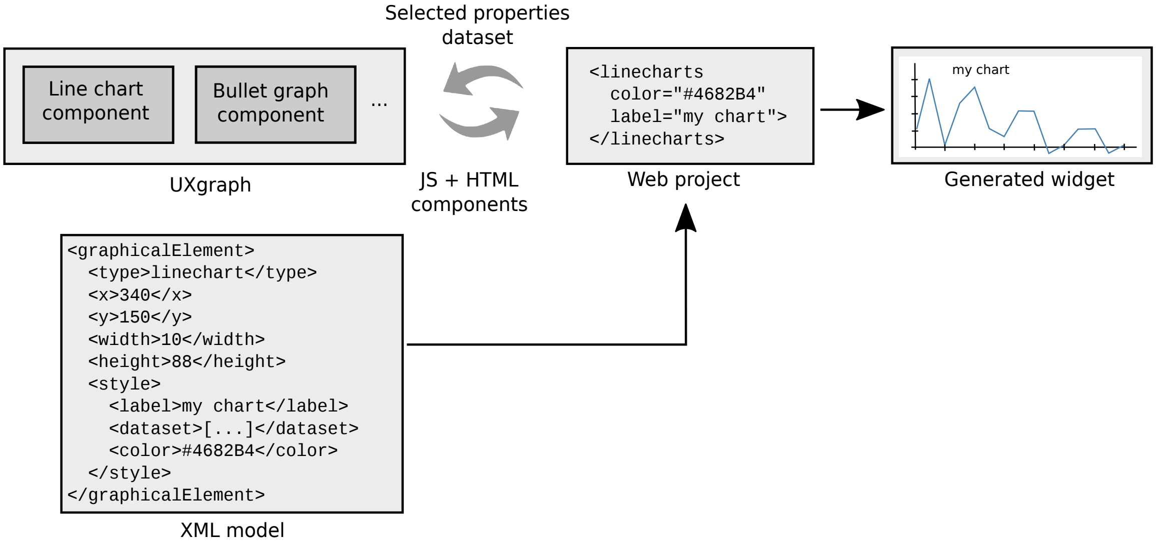
Name	Email	Created At	
Alvin	Email	01.01.17	Remove
Alvin	Email	01.01.17	Remove
Alvin	Email	01.01.17	Remove

Name	Email	Created At	
Alvin	Email	01.01.17	Remove
Alvin	Email	01.01.17	Remove
Alvin	Email	01.01.17	Remove

Figure 5.10: The UI of Generator of dashboard samples. The left sidebar displays the list of all available widgets, which can be added to the grid using drag-n-drop. Designers can export the dashboard in the XML and PNG format for further analysis.

Figure 5.10 presents the UI of the generator. It is a single-page application, which loads an HTML page consisting of a canvas and palette of dashboard components, which can be placed into the canvas by the users dynamically. The palette of components provides the charts recommended for dashboards—e.g., bullet graph, bar chart (horizontal and vertical), stacked bar chart (horizontal and vertical), line chart, and dynamic sparklines. The user can manually design one dashboard or generate a set of dashboards with predefined styles and data. Then, the user can export the internal representation of the dashboard (a bitmap and XML description of objects) including a simple description of styles and data (Figure 5.11).

[2] Projects' websites—Klipfolio: `https://www.klipfolio.com/`, Datapine: `https://www.datapine.com/`, Sisense: `https://www.sisense.com/`, The Dash: `https://www.thedash.com/`



```
<linecharts
  color="#4682B4"
  label="my chart">
</linecharts>
```

```
<graphicalElement>
  <type>linechart</type>
  <x>340</x>
  <y>150</y>
  <width>10</width>
  <height>88</height>
  <style>
    <label>my chart</label>
    <dataset>[...]</dataset>
    <color>#4682B4</color>
  </style>
</graphicalElement>
```

Figure 5.11: Custom HTML tags. A new HTML tag is defined for every widget type with the help of UXgraph library.

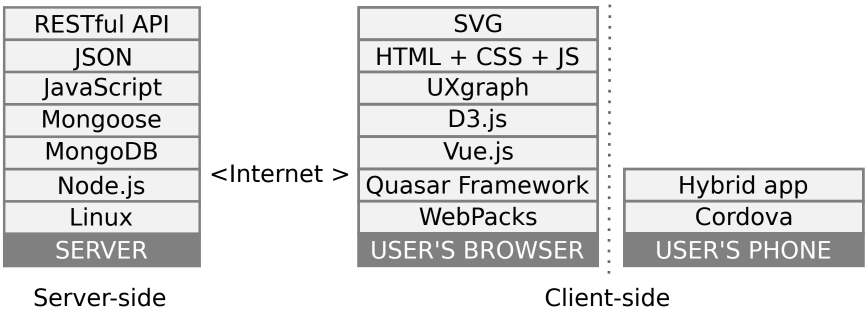


Figure 5.12: The architecture of the generator. It illustrates the technologies used for the back-end and front-end as independent applications. The front-end may be wrapped to a hybrid mobile application.

The architecture of the generator is shown in Figure 5.12. The back-end supports the RESTful API, which allows easier scalability of the application and independence of the server and client sides. It also allows extending the generator in order to support construction of other UI types (e.g., the UI for mobile devices). The back-end is based on the Node.js environment and the MongoDB database, which allows storing the model directly in the JSON format. The front-end is built with the Vue.js[3] framework using the UXgraph library, which was developed primarily for this generator. The UXgraph library combines the advantage of the Vue.js framework for building user interfaces with the D3.js[4] library for data visualization. The reason behind creating a unique library was to have a predefined set of reusable and easily scalable widgets. They use the same model but with the possibility of applying different styles. The source code of the UXgraph library is available online.[5] Another advantage of the developed application is that it may be extended to

[3] Vue.js project's website: https://vuejs.org/

[4] D3.js project's website: https://d3js.org/

[5] The UXgraph library project's repository: https://github.com/lirael/vuejs-d3-uxgraph-demo

a hybrid mobile application, using the Cordova wrapper. This advantage was not, however, used.

5.3.3 Interactive Survey Tool

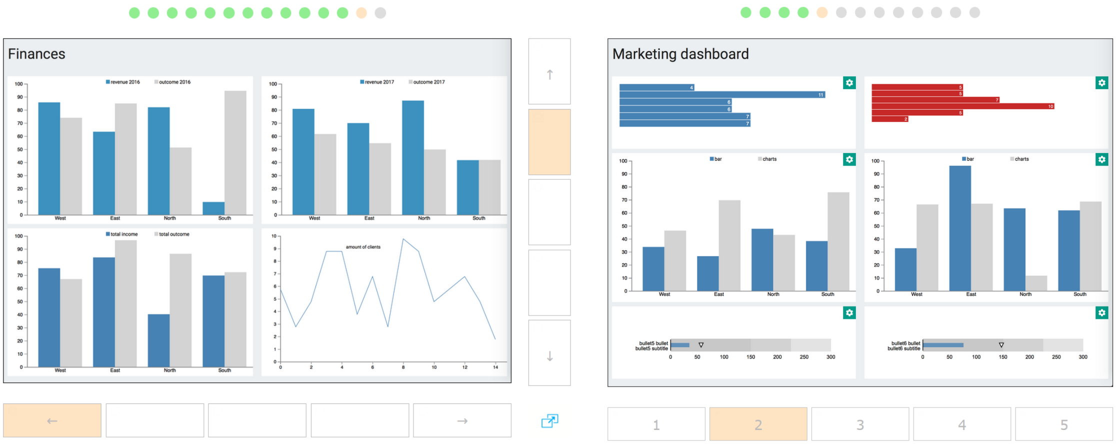
One of the tasks of the research was to let users rate characteristics of dashboards so that they could be compared with the results of metrics. For this purpose, a simple tool for creating interactive surveys was created by the author of this thesis. It was implemented in the HTML, CSS, and JavaScript languages. The functionality of the tool is following:

- It allows generating multi-page forms whose every page consists of one dashboard bitmap and buttons for answers of reviewers. The screen also contains buttons for fullscreen mode and monitoring the progress of answering. There are not any other graphical elements on the screen which could skew users' perception.

- One survey can contain multiple sets of dashboard bitmaps. Every set can be connected with different groups of users. It is useful for the surveys which let every user work with own set of dashboards.

- The tool can generate forms containing different questions and answers buttons for every set of dashboards. The specifications of answers are done via a configuration file in the TOML language. Examples of button sets are 5-point Likert scale or two sets of buttons evaluating the vertical or horizontal balance (Figure 5.13).

- The results of users' reviews are stored in the configuration files for every form and user so that further analyses can process the results automatically.



(a) vertical and horizontal balance

(b) overall symmetry

Figure 5.13: An example of forms with different kinds of answers. Users go through a set of samples and select values of UI characteristics according to their subjective perception.

The tool was used for analysis of the impact of color on object-based metrics (Section 8.2) and final comparisons of metrics with user reviews (Chapter 10). The source code of the application is available online (see Appendix B.2).

5.4 Dataset

Generator of Dashboard Samples described in Subsection 5.3.2 tries to generate realistic-looking samples. However, it can not provide the diversity of real dashboards. The generated samples were used to evaluate the hypotheses regarding the impact of dashboard visual characteristics on the usability of the dashboard (e.g., the importance of color during measuring aesthetics). The evaluation of metrics, however, required real dashboard samples in order to evaluate the real applicability of the metrics. Hence, 130 various dashboard bitmaps were gathered from the Internet. They were split into the two groups:

1. $D_{\text{(well)}}$: the group of 9 dashboards which were designed according to the design heuristics defined by [Few, 2006]. The dashboards were considered as "well-designed".

2. $D_{\text{(rand)}}$: the group of 121 randomly chosen dashboards which were collected from the Internet. No information about the usability of these dashboards was known. The dashboards were labeled as "random".

All dashboards ($D_{\text{(all)}} = D_{\text{(well)}} \cup D_{\text{(rand)}}$) were used to evaluate the applicability of chosen pixel-based (Chapter 6), object-based (Chapter 7), and combined (Chapter 8) metrics. The description of dashboards' regions were obtained from users (Section 7.1). They were used to train the method for segmentation of dashboards (Chapter 9).

The reason I chose the samples based on Few's knowledge to be well-designed was the lack of other samples based on similarly credible knowledge as Tufte [2001]; Ware [2004]. I did not perform user testing to distinguish well-designed dashboards. The evaluation of dashboard quality should not be based only on the first impression of users but also on an in-depth analysis of interface usability as it was provided by Few. Moreover, it was not the aim of this research to evaluate the correctness of Few's framework or to establish another one. The thesis uses the label "well-designed" in the following text. However, the readers of this thesis should consider the limitation of this label.

5.5 Summary

This chapter defined the three stages of evaluation of dashboard quality: (1.) loading a dashboard, (2.) conversion of the dashboard to the internal representation, and (3.) evaluation of the design quality using suitable metrics analyzing the internal representation of the dashboard. The process has to deal with various dashboard formats, subjectivity of user perception of objects, and suitability of the metrics for the evaluation. The internal representation of dashboards should unify various dashboard formats and help to find suitable objective metrics. For this purpose, it was needed to create a model of the internal representation and develop tools which would be able to convert dashboards to the internal representation and use it for evaluation of metrics.

Firstly, the model of the internal representation of dashboards was established. The model contains the two dashboard perspectives which respect the classification of metrics presented in Section 3.3—the pixel-based and object-based perspective. The pixel-based perspective represents dashboards as bitmaps stored in various color spaces. The object-based perspective describes the structure of a dashboard which consists of objects arranged on a screen. The theory for the description of a dashboard structure was established. It consists of the model of dashboard's components, dashboard template and dashboard sample. The model of dashboard's components specifies objects which can be contained by the dashboard, their implicit appearance, behavior, and description of visual characteristics and data which can be changed. A dashboard template specifies an arrangement and style of dashboard's components on a screen. The dashboard sample is one instance of a dashboard template presenting data. The theory is language-independent. This research works with the simplified object-based description represented in the XML language. It focuses on the size and dimension of objects, which are used for evaluation of layouts and aesthetics by Ngo's metrics. The behavior of objects is ignored.

Then, three applications dealing with the dashboard internal representation were implemented with the cooperation of students of bachelor's and master's study program at Brno University of Technology. The first application—*Dashboard Analyzer* provides tools for manipulation with dashboard bitmaps. It offers tools for description of dashboard components and the APIs for implementation and evaluation of metrics measuring dashboard characteristics and algorithms for segmentation of dashboards into dashboard components. The second application—*Generator of Dashboard Samples* provides the ability to manually create or automatically generate synthetic dashboard samples represented by a bitmap and structural description. Then, the dashboards can be used for evaluation of the impact of dashboard visual characteristics on usability and quality of the dashboard. The third application—*Interactive Survey Tool* helps to create interactive forms for surveying users about perceived characteristics of dashboards.

Finally, it was important to gather real dashboards which could be used for overall evaluations of metrics. 130 dashboard bitmaps were found on the Internet and divided into the two groups: the group of random dashboards and the group of dashboards which respect the design heuristics of Few [2006]. The dashboards were used for analyses of pixel-based (Chapter 6) and object-based (Chapter 7) metrics, evaluations of metrics improvements (Chapter 8), design of the algorithm for segmentation dashboards (Chapter 9), and comparison of the metrics with user reviews (Chapter 10).

Chapter 6

Analysis of Pixel-based Metrics

The goal of this research task was to analyze selected visual characteristics of user interfaces which are measurable by pixel-based metrics. Subsection 3.3.1 introduced the examples of visual characteristics of user interfaces which have an impact on UI quality. It provided the pixel-based metrics measuring those visual characteristics including heuristics recommending appropriate settings of the visual characteristics:

- **Colorfulness:** UI designers should use subtle colors instead of vivid colors. Vivid colors should be used only for emphasizing important data.
- **Number and Share of Used Colors:** UI designers should use a limited number of colors. Users can use online services recommending colors combinations.
- **Distributions of colors (balance, symmetry):** UI designers should use an appropriate layout. They should not distribute graphical elements on a screen arbitrarily.

This research task analyzed the possibility to use the pixel-based metrics for evaluation of dashboard design quality and recognition of well-designed dashboards. The hypothesis was that well-designed dashboards should more likely satisfy the design heuristics. It was expected that this hypothesis should be verified by using the pixel-based metrics. This chapter presents the description of the procedure and the results of the analysis of the pixel-based metrics. The results were published in [Hynek and Hruška, 2016].[1]

6.1 Procedure

The test set was composed of the 130 various dashboard bitmaps described in Section 5.4 divided into the group $D_{(\text{well})}$ of 9 well-designed and $D_{(\text{rand})}$ of 121 random dashboards. Besides that, a group labeled as $D'_{(\text{well})}$ was created. It contained all dashboards of $D_{(\text{well})}$ resized into the 50% of the original width and height. Every dashboard was stored as a bitmap in the 32-bit RGB color space. Further transformations into other color spaces were done for the purposes of particular metrics.

The pixel-based metrics were implemented using Dashboard Analyzer API presented in Subsection 5.3.1. I used the tools of Dashboard Analyzer to measure values of all dashboards of groups $D_{(\text{rand})}$, $D_{(\text{well})}$, and $D'_{(\text{well})}$. The measured values were used to calculate the arithmetic mean μ and standard deviation σ for all three groups.

[1] The publication used a slightly different set of dashboard samples. The values presented in this thesis might be a little bit different but the nature of the results is kept.

The following evaluations of the results were made:

- Analysis of $\mu_{D_{\text{(rand)}}}$ and $\sigma_{D_{\text{(rand)}}}$ to find the characteristic features of dashboards.
- Analysis of the standard deviations $\mu_{D_{\text{(well)}}}$ and $\sigma_{D_{\text{(well)}}}$ to find the characteristic features of well-designed dashboards.
- Comparisons of the results of $D_{\text{(rand)}}$ with the result of $D_{\text{(well)}}$ and $D'_{\text{(well)}}$ and analysis of the difference between the characteristics of well-designed and randomly chosen dashboards.
- Comparisons of the results of $D_{\text{(well)}}$ with the result of $D'_{\text{(well)}}$ and analysis of the influence of a bitmap's resolution on the precision of measuring.

All results described in this chapter are available in Appendix A.3.

6.2 Analysis of Metrics Measuring Colorfulness

Colorfulness of bitmap was measured according to Formula 3.1 of [Yendrikhovskij et al., 1998]. It was measured as the sum of the average saturation in the CIE L*a*b* color space and its standard deviation. For the experimental purposes, this formula was applied to the other color channels of the CIE L*a*b* and HSB color spaces:

- CIELAB bitmap: lightness, chroma, hue, saturation
- HSB bitmap: hue, saturation, brightness

Then, the generalized formula was the following:

$$C_i = c_i + \sigma_i, \tag{6.1}$$

where c_i represents the average value of a color channel of all pixels in a bitmap i and σ_i represents its standard deviation.

6.2.1 Results

Table 6.1 presents the results. The standard deviations $\sigma_{D_{\text{(rand)}}}$ show a high contrast between the values of dashboard colorfulness. This corresponds to the fact that we can find more or less colorful dashboards. The occurrence of less colorful dashboards is, however, higher among the dashboards of $D_{\text{(well)}}$. We can see that $\mu_{D_{\text{(well)}}}$ of the average colorfulness based on the CIELAB saturation is three times lower than the same kind of value measured for the dashboards of $D_{\text{(rand)}}$. For instance, the colorfulness of the Few's sales dashboard shown in Figure 4.10a is 0.162 while the colorfulness of the second sales dashboard in Figure 4.10b is 0.917. These results are supported by the results based on the HSB saturation.

As regards the remaining color channels, we can see that the $D_{\text{(well)}}$ group is characterized by the lower average values based on the HSB hue and CIELAB chroma and the higher average values based on the HSB and CIELAB brightness. On the other hand, there is not a big difference between the average values based on the CIELAB hue since the standard deviation $\sigma_{D_{\text{(well)}}}$ is too high.

Finally, the comparison of the $D_{\text{(well)}}$ and $D'_{\text{(well)}}$ groups shows that the resizing of the bitmaps has an impact on application of the results. However, the impact is not high.

Table 6.1: The results of the colorfulness analysis for the groups of dashboards: $D_{\text{(rand)}}$, $D_{\text{(well)}}$, and $D'_{\text{(well)}}$. The values of μ represents the average colorfulness of a dashboard group and σ its standard deviation. The CIELAB saturation represents the metric of colorfulness which was designed by Yendrikhovskij et al. [1998].

Metric	$\mu_{D_{\text{(rand)}}}$	$\sigma_{D_{\text{(rand)}}}$	$\mu_{D_{\text{(well)}}}$	$\sigma_{D_{\text{(well)}}}$	$\mu_{D'_{\text{(well)}}}$	$\sigma_{D'_{\text{(well)}}}$
HSB						
hue	0.438	0.160	0.191	0.062	0.201	0.043
saturation	0.384	0.209	0.125	0.039	0.114	0.035
brightness	1.000	0.187	1.089	0.027	1.081	0.029
CIELAB						
lightness	98.579	20.566	108.810	2.891	108.022	3.063
chroma	24.575	13.788	10.011	3.559	9.393	3.504
hue	317.044	28.557	256.484	94.569	257.477	95.253
saturation	0.690	0.581	0.265	0.209	0.209	0.159

6.2.2 Conclusions

Measuring dashboard colorfulness according to the formula of Yendrikhovskij et al. [1998] can be used as one of the approaches evaluating the design quality of dashboards. It has been shown that colorfulness of the dashboards designed according to Few's framework is usually low. These dashboards usually use subtle non-distracting colors, which decrease overall colorfulness of the dashboards. It has also been shown that it might be useful to analyze other color channels than the CIELAB saturation (e.g., the channels of the HSB color space). Analysis of bitmap colorfulness can warn a UI designer about overusing of vivid colors. The UI designer should, however, keep in mind the problem of different perception of colors by people (e.g., colorblindness).

6.3 Analysis of Metrics Measuring Color Share

The second group of pixel-based metrics focused on measuring the characteristics corresponding with the numbers of distinct color values used in a bitmap, and the numbers of pixels occupied by these colors. Particularly, the metrics analyzed the following bitmaps (with respect to the model in Subsection 5.2.1) and measured the following characteristics:

- **24-bit RGB bitmap** ($2^{24} \sim 16,77$ million possible color values):
  - the number of distinct color values
  - the share of the 1st most used color value in the bitmap
  - the share of the 1st+2nd most used color values in the bitmap
- **Posterized 12-bit RGB bitmap** ($2^{12} = 4096$ possible color values):
  - the same metrics as for 24-bit RGB bitmap

- **8-bit Grayscale bitmap** ($2^8 = 256$ possible color values):
  - the number of distinct color values
    - in total
    - whose share of bitmap area higher than: 0.1%, 0.5%, 1%, 2%, and 10%
  - the share of the 1st most used color value
  - the share of the 1st+2nd most used color values
- **Posterized 4-bit Grayscale bitmap** ($2^4 = 16$ possible color values):
  - the same metrics as for 24-bit 8-bit Grayscale bitmap
- **Adaptive Black-and-white bitmap** (2 color values):
  - the share of the black and white color values

The measurements should provide information about frequently used colors, which usually represent background layers.

6.3.1 Results

Table 6.2 presents the results. The number of colors used in the dashboards of $D_{(\text{well})}$ is usually lower than in the group of random dashboards. Few [2006] recommends avoiding an excessive number of colors. The dashboards of $D_{(\text{well})}$ do not usually contain color gradients, which significantly increase the number of colors. However, the number of colors is not a reliable metric. The difference between the average numbers of colors of $D_{(\text{well})}$ and $D'_{(\text{well})}$ is high. Bitmaps can be resized or compressed so that a viewer does not recognize the difference between the original and compressed bitmap. On the other hand, such a compression might cause significant reduction of colors, which has an impact on the usability of the metric. Conversion of samples to different color spaces was not helpful in this case.

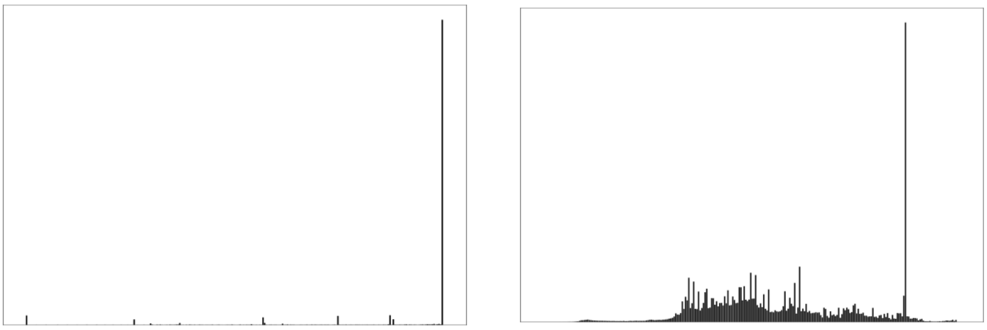
For this reason, the analysis was focused on the numbers of dominant colors which are represented by a sufficiently high number of pixels. According to the results, all dashboards of $D_{(\text{well})}$ contained only one color with a share higher than 10% in the posterized 4-bit grayscale color space, in contrast to the $D_{(\text{rand})}$ group. This finding brought up the idea to measure the share of the most used color values, which seems to be one of the metrics applicable for evaluation of dashboards. The dashboards of $D_{(\text{well})}$ usually contain one or two background layers represented by uniform (usually light or pale) color. On the other hand, dashboards of $D_{(\text{rand})}$ more often contain a background represented by a color gradient. There is also a higher occurrence of non-data pixels, which decrease the number of pixels representing the background. The difference between the $D_{(\text{well})}$ and $D_{(\text{rand})}$ groups is noticeable either in the RGB or the grayscale color space. Then, the posterizations of bitmaps made the dashboards of $D_{(\text{well})}$ even more recognizable by this characteristic since the standard deviations $\sigma_{D_{(\text{well})}}$ were decreased significantly. Compressions of bitmaps did not affect the results of the measuring radically.

Another way how to analyze the color share was to create color histograms. For this purpose, I used 8-bit Grayscale bitmaps, which consist of 256 distinct color values (color intensities). The histograms were created in Dashboard Analyzer. They visualized the differences between the color usage of the dashboards of $D_{(\text{well})}$ and $D_{(\text{rand})}$ groups comprehensively. Figure 6.1 shows that the dashboards with a low rate of colorfulness (Figure 6.1a)

Table 6.2: The results of the analysis of color share for the groups of dashboards: $D_{(\text{rand})}$, $D_{(\text{well})}$, and $D'_{(\text{well})}$. The values of μ represents the average values of a dashboard group and σ its standard deviation.

Metric	$\mu_{D_{(\text{rand})}}$	$\sigma_{D_{(\text{rand})}}$	$\mu_{D_{(\text{well})}}$	$\sigma_{D_{(\text{well})}}$	$\mu_{D'_{(\text{well})}}$	$\sigma_{D'_{(\text{well})}}$
24-bit RGB						
Number of color values	24 426	31 108	2 730	4 145	5 946	6 608
Share of the 1st color	37.57%	20.81%	65.55%	20.91%	61.59%	21.41%
Share of the 1st+2nd colors	47.99%	22.93%	75.44%	20.81%	69.75%	20.69%
12-bit RGB						
Number of color values	677	467	250	183	374	395
Share of the 1st color	54.50%	21.27%	81.62%	8.28%	79.46%	8.18%
Share of the 1st+2nd color	66.23%	19.14%	85.97%	5.26%	84.05%	5.05%
8-bit Grayscale						
Number of color values	230.30	35.12	185.67	69.06	192.33	69.10
. . . with share > 0.1%	82.24	55.17	41.67	26.92	53.77	31.26
. . . with share > 0.5%	22.45	14.46	8.67	3.81	10.67	3.84
. . . with share > 1%	11.58	6.91	6.11	2.57	6.22	2.82
. . . with share > 5%	2.83	1.55	1.56	0.52	1.78	0.83
. . . with share > 10%	1.48	0.98	1.33	0.50	1.22	0.44
Share of the 1st color	38.43%	20.52%	68.39%	16.39%	64.67%	16.39%
Share of the 1st+2nd color	50.18%	22.91%	79.61%	12.90%	73.80%	13.40%
4-bit Grayscale						
Number of color values	15.52	0.98	14.89	2.03	15.11	1.45
. . . with share > 0.1%	13.90	1.89	13.44	1.74	13.22	1.72
. . . with share > 0.5%	10.88	3.14	9.22	1.20	9.67	1.58
. . . with share > 1%	8.81	3.35	6.33	1.66	6.89	1.45
. . . with share > 5%	3.38	1.69	1.22	0.44	1.33	0.71
. . . with share > 10%	2.04	0.98	1.00	0.00	1.00	0.00
Share of the 1st color	57.47%	19.79%	84.81%	4.76%	82.89%	4.36%
Share of the 1st+2nd color	72.43%	18.15%	88.38%	3.55%	86.82%	3.76%
1-bit Black-and-white						
Share of black color	28.84%	15.80%	12.52%	3.37%	15.01%	3.44%
Share of white color	71.16%		87.48%		84.99%	

usually contain one dominant intensity (background) and a few other intensities with a low frequency of occurrence (the data pixels). On the contrary, the histograms of the colorful dashboards (Figure 6.1b) consist of many color intensities with a relatively high frequency of occurrence. Histograms can also be used for detection of background layers in the method for segmentation of dashboards (Chapter 9).



(a) Grayscale histogram of Fig. 4.10a **(b)** Grayscale histogram of Fig. 4.10b

Figure 6.1: Histograms of the two dashboard bitmaps presented in Figure 4.10 converted to the 8-bit grayscale color space. The horizontal axis represents the values of the 8-bit grayscale color space (0-255, from black to white). The vertical axis represents the number of pixels which represent a particular color value.

Finally, the thresholding of bitmaps confirmed the fact that the dashboards of $D_{(\text{well})}$ contain a higher share of background (white color) than the dashboards of $D_{(\text{rand})}$. The threshold was changed adaptively according to Bradley and Roth [2007]. It helped to consider local changes of the image contrast. However, not all dashboards can be divided binary into foreground and background. Some dashboards consist of more than two layers. It is not always easy to find an optimal threshold automatically. Hence, the metric measuring the share of the black and white color in Black-and-white bitmap does not seem to be much reliable for evaluation of dashboards.

6.3.2 Conclusions

Measuring the share of the most used colors in a dashboard bitmap can provide information about the dashboard background. Well-designed dashboards usually contain a small number of dominant colors (usually one or two), which occupy more than half of the bitmap. These colors represent background layers of the dashboards. Other dashboards usually contain a higher number of non-data pixels (e.g., decorations, color gradients), which decrease the share of uniform background and make the dashboards more distracting. Grayscale histograms of such dashboards usually miss color values which significantly exceed other values. Analysis of histograms can warn a UI designer about the possibility of inappropriate use of color gradients and a high number of non-data pixels.

On the contrary, measuring the number of all colors used in a bitmap does not seem to be a reliable metric. The results have shown that the group of well-designed dashboards usually contains a lower number of colors than the other dashboards, but the number highly depends on the quality (resolution) of the bitmap. For instance, a dashboard can contain a small element (e.g., line) containing a color gradient (a small shadow) hardly recognizable by a human. The color gradient can, however, strongly increase the number of used colors.

Another problem is that people might not be able to distinguish all color values. The RGB color space can describe more than 16 million different color values. On the contrary, a human would perceive many different color values as the same.

Finally, measuring the share of the black and white color in Black-and-white bitmap was analyzed. The metric was not considered as a reliable one since there might be problems with the thresholding of the dashboards which consist of more than two background layers. Thresholding such dashboards might be highly ambiguous.

6.4 Analysis of Metrics Measuring Color Distribution

The third group of pixel-based metrics focused on measuring the distribution of color in bitmaps—particularly on the metrics measuring:

- **Balance:** $\mathrm{BM}_{(\mathrm{pixel})} = 1 - \frac{|\mathrm{BM}_v| + |\mathrm{BM}_h|}{2} \in [0, 1]$, where BM_v and BM_h are values of the vertical and horizontal balance calculated according to Formula 3.2.
- **Symmetry:** $\mathrm{SM}_{(\mathrm{pixel})} = 1 - \frac{|\mathrm{SM}_v| + |\mathrm{SM}_h|}{2} \in [0, 1]$, where SM_v and SM_h are values of the vertical and horizontal symmetry calculated according to Formula 3.4.

The metrics analyzed the following bitmaps (with respect to the model in Subsection 5.2.1):

- 8-bit Grayscale bitmap
- Posterized 4-bit Grayscale bitmap
- Adaptive Black-and-white bitmap

6.4.1 Results

Table 6.3 presents the results. The dashboard bitmaps of all groups can be characterized as highly balanced and symmetrical in all three color spaces. The reason might be that people tend to see objects symmetrical as described by the Gestalt law of symmetry. There is a chance that designers more likely design symmetrical screens (which is the special case of a balanced screen). We can, however, see that the average values of balance and the symmetry are higher for the dashboards of $D_{(\mathrm{well})}$. This was expected since their graphical elements use the same or similar colors, which are uniformly distributed on a screen. The dashboards of $D_{(\mathrm{well})}$ does not contain any sidebars or menus (usually located on the left or top of a UI) which would break balance and symmetry of the screen.

The results of the $D'_{(\mathrm{well})}$ group are similar to those of the $D_{(\mathrm{well})}$ group. Hence, the pixel-based metrics of balance and symmetry are applicable for evaluation of dashboard bitmaps. Using Black-and-white bitmap is, however, not recommended because of the ambiguity of thresholding.

6.4.2 Conclusions

The pixel-based metrics of UI balance and symmetry are applicable for evaluation of dashboard quality. Results have shown that well-designed dashboards can be recognized by higher rates of balance and symmetry. Analysis of balance and symmetry might warn a designer about the possibility of unbalanced and asymmetrical distribution of colors (color intensities) on a screen.

Table 6.3: Results of the color distribution analysis for the groups of dashboards: $D_{(\text{rand})}$, $D_{(\text{well})}$, and $D'_{(\text{well})}$. The values of μ represents the average values of a dashboard group and σ its standard deviation.

Metric	$\mu_{D_{(\text{rand})}}$	$\sigma_{D_{(\text{rand})}}$	$\mu_{D_{(\text{well})}}$	$\sigma_{D_{(\text{well})}}$	$\mu_{D'_{(\text{well})}}$	$\sigma_{D'_{(\text{well})}}$
8-bit Gray						
balance	0.703	0.166	0.824	0.080	0.831	0.071
symmetry	0.844	0.063	0.917	0.020	0.916	0.020
4-bit Gray						
balance	0.761	0.139	0.906	0.036	0.907	0.036
symmetry	0.852	0.063	0.925	0.019	0.923	0.020
1-bit Black-and-white						
balance	0.726	0.146	0.807	0.081	0.840	0.080
symmetry	0.706	0.086	0.810	0.045	0.783	0.047

6.5 Limitations

The limitations of the results are similar for all three groups of metrics. Readers should take into consideration that the results are based on the limited number of dashboard samples. The well-designed samples were chosen among the dashboards based on the design heuristics of Few [2006]. Some metrics like colorfulness expect well-designed dashboards to have a light background and dark foreground. However, many dashboards have inverted colors since they might be used in a dark environment (e.g., at night). It should be appropriate to analyze color histograms and consider this fact (e.g., invert the colors before using the metrics).

The metrics evaluate simple visual characteristics which can be used for the detection of design problems. They often detect false positives and false negatives. Synthetic bitmaps respecting recommended rates of visual characteristics might be rated similarly as well-designed dashboards even though they do not represent dashboards. On the other hand, violation of some design recommendations does not necessarily mean that the dashboard is not usable. The metrics should not be used for direct detection of design mistakes. They should provide additional warnings about inappropriate usage of colors.

Some of the pixel-based metrics are not able to deal with a reduction of dashboard size. One dashboard represented by two differently scaled bitmaps might be rated differently. This goes especially for the metric measuring the number of used colors. Such metrics are less reliable. Dashboard bitmaps should be stored in the same resolution as they are presented on a screen.

Finally, there is the problem regarding ambiguous perception of color by users. The pixel-based metrics might not reflect an actual perception of color values by users. It might be more objective to analyze dashboard represented in the grayscale color space than the RGB color space.

6.6 Summary

The goal of this research task was to analyze selected visual characteristics of dashboards which are measurable by pixel-based metrics. It analyzed the possibility to use the pixel-based metrics for evaluation of dashboard design quality. For this purpose, the API and tools of Dashboard Analyzer were used. The results have indicated that the group of well-designed dashboards is less colorful than the randomly chosen ones. They usually contain one or two frequently used color values, which represent background layers. Background usually occupies more than half of a well-designed dashboard. It is usually represented by some light color with a low value of saturation and high value of brightness (e.g., white). The results have also shown a high rate of balance and symmetry for all dashboards. The well-designed dashboards were, however, more balanced and symmetrical on average than the randomly chosen dashboards.

The analysis of pixel-based metrics recommended to use the following metrics:

1. colorfulness based on color channels of the CIE L*a*b* color space (especially saturation as described by Yendrikhovskij et al. [1998])

2. colorfulness based on color channels measured in the HSB color space

3. the number of colors with the share higher than 10% in the posterized 4-bit grayscale color space

4. the share of the 1st most or 1st + 2nd most used color values measured in the posterized 4-bit grayscale or 12-bit RGB color space

5. pixel-based Balance and Symmetry measured in the posterized 4-bit grayscale color space

UI designers should keep in mind the problem regarding the ambiguous perception of color by users. Hence, it is recommended to use the posterized grayscale color space instead of the RGB color space. UI designers should also consider all the limitations of this research (e.g., the limited number of dashboard samples). The metrics should be used for additional analysis providing warnings about inappropriate use of colors.

Chapter 7

Analysis of Object-based Metrics

The goal of this research task was to analyze selected visual characteristics of user interfaces which are measurable by object-based metrics. It analyzed the possibility to apply Ngo's metrics described in Subsection 3.3.2 for evaluation of dashboard design quality and recognition of well-designed dashboards. In contrast to the analysis of pixel-based metrics, the analysis of object-based metrics had to deal with the ambiguity of metrics inputs.

The analysis followed the approach of Zen and Vanderdonckt [2014], who work with manually selected regions representing objects on a screen. It considered the subjective perception of users. Firstly, it let a group of users to manually draw regions on a screen. It used the descriptions of regions to measure the ambiguity of the users' perception (Section 7.1). Then, it established a framework for the processing of the descriptions of regions and measuring the ability of the object-based metrics to deal with the ambiguity of the descriptions of regions (Section 7.2). Finally, the framework was used to measure the impact of the ambiguous perception on the ability of Ngo's metrics to distinguish the group of well-designed dashboards objectively (Section 7.3).

The following sections provide a summarized overview of the results. More detailed results (including all user descriptions of regions and the statistics based on these descriptions) are available in Appendix A.3. The results were published in [Hynek and Hruška, 2018].

7.1 Study of Visual Perception of Objects

The first part of the research task was focused on the user perception of visually dominant objects. An experiment was performed to understand the principles of objects grouping and ambiguity of user perception.

7.1.1 Procedure

Gathering the Region Descriptions

Firstly, the 130 dashboard bitmaps described in Section 5.4 were divided into 13 groups of 20 samples (every sample was contained by two groups). Then, the groups were uniformly distributed among 361 users—third-year students (around 20 years old) of the Information Systems course (autumn 2016) at Brno University of Technology, Faculty of Information Technology. The students were asked to provide descriptions of regions representing their subjective perception of objects within a dashboard (*user description*). We dedicated one lecture to familiarize the students with the *dashboard* term and the fundamental principles of data visualization and visual perception. The task was performed in the form of an optional homework. 251 of 361 students decided to participate.

Participants described regions of 20 dashboards according to their subjective perception. They used a special release of Dashboard Analyzer (Subsection 5.3.1) which allowed them to draw the perceived regions and generate an XML file of the described regions which respects the format of the internal representation described in Subsection 5.2.2, Listing 5.3. The application did not allow the users to specify regions hierarchically (regions within regions) since the research was focused only on the top-level objects. A total of 251 users provided 5,020 user descriptions of regions in total (approximately 39 user descriptions for every dashboard).

Measuring Perception Ambiguity

First of all, I took the user descriptions of the regions of the same dashboard and combined them into one *average description* representing the probabilities $p_i \in [0,1]$ of region occurrences for every pixel i of the dashboard. Figure 7.1a shows a visualization of such an average description in the grayscale color space. Then, I used the average description to measure the entropy of the dashboard—a value representing the rate of user disagreement about the distribution of regions.

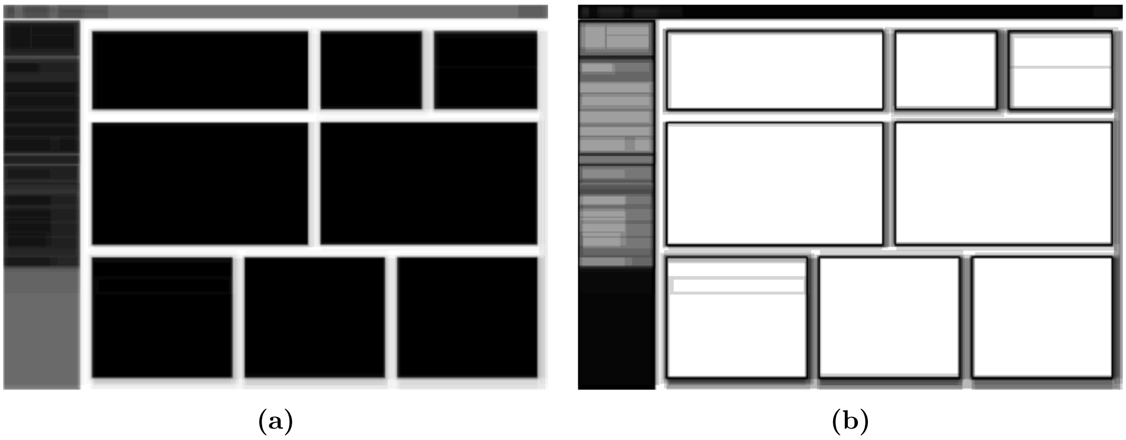


Figure 7.1: An example of an average description (a) and a visualization of pixel entropies (b) represented in the grayscale color space. The higher color intensity represents the higher probability (a) and higher entropy (b) of region occurrence. The pixels representing medium probabilities of region occurrence ($p_i \sim 0.5$) are represented by higher values of entropy. Such pixels usually create borders of visually dominant objects. They can also be found in management areas (toolbars, menus) on the borders of a screen.

The binary entropy of every pixel was calculated according to the following formula:

$$E_{p_i} = -(p_i \log_2 p_i + (1 - p_i) \log_2 (1 - p_i)) \tag{7.1}$$

where $p_i \in \{0,1\}$ represents the probability of region occurrence in the i-th pixel position ($i = (x,y)$) in the matrix and $E_{p_i} \in [0,1]$. An example of visualization of entropy values can be seen in Figure 7.1b.

Then, the entropy of a dashboard d was calculated as the average binary entropy of all the pixels in the dashboard:

$$E_d = \frac{\sum_{i=0}^{n} E_{p_i}}{n} \tag{7.2}$$

where n is the number of all the pixels in the dashboard $d \in D_{(\text{all})}$ and $E_d \in [0, 1]$. I measured the average entropy μ_E with its standard deviation σ_E for the set of all dashboards $D_{(\text{all})}$.

In addition to the entropy, I analyzed the number of regions in dashboard and the user disagreement about this value. For every description (set) of regions $R_d^{(u)}$ of a dashboard d provided by a user u, the number of regions $\nu_d^{(u)} = |R_d^{(u)}|$ was calculated. Then, for every dashboard $d \in D_{(\text{all})}$, the average number of regions μ_{ν_d} with its standard deviation σ_{ν_d} and the coefficient of variation $c_v(\mu_{\nu_d}, \sigma_{\nu_d})$ was calculated. Finally, I analyzed the average coefficient of variation c_{v_ν} for the set of all dashboards $D_{(\text{all})}$.

7.1.2 Results

The average entropy of all dashboards μ_E was 0.262 ($\sigma_E = 0.109$). It means that the value p_i of every pixel i was 0.955 on average (95.5% of the users agreed on the logical value of a pixel). Therefore, the average entropy can be considered as low. Visualization of entropy matrices then indicated that high entropy was detected on the borders of regions (the black borders of white rectangles in Figure 7.1b). Also, some users considered these areas as solid regions; other users split them into smaller logical regions (such as buttons and labels).

The average coefficient of variation c_{v_ν} was 0.78. It means that the standard deviations of the number of regions were relatively high compared to the average numbers of regions. It revealed the fact that the users usually agreed about the location of regions but disagreed about their quantity. They segmented the screen with different granularity, as it was suggested in Figure 1.4 in Introduction.

7.1.3 Limitations

The results are limited by the chosen sample of users who provided the descriptions of perceived regions. Although the number of users was relatively large compared to other evaluations described in Subsection 3.3.3, the users were mainly technical students. I expect that there may be slight, but interesting deviations between the perception of people of different specializations (e.g., persons having skills in the arts).

We also need to consider the fact, that the users might have specified the regions inaccurately. They might have performed the task quickly in order to accomplish the homework. The descriptions of regions might not represent the exact reflection of the users' perception.

7.1.4 Conclusions

In conclusion, the experiment confirmed the fact that people recognize visually emphasized objects similarly, which corresponds to Gestalt laws. Based on the results of the average entropy μ_E, it is expected that a designer should be able to create a description of visually dominant regions which will cover a similar area of the screen as the average description of regions made by a sufficient number of instructed users. On the other hand, the subjective factor of visual perception will always be present ($\mu_E > 0$). Another designer will most likely create a slightly different description. Two designers using their subjective descriptions to evaluate one user interface can end up with different results. Hence, they should use only sufficiently robust metrics which are able to consider certain differences caused by subjective perception. Specifically, they should not use object-based metrics which are highly dependent on the number of objects (due to the high value of c_{v_ν}).

7.2 Framework for Rating Object-Metrics

The second part of the research task was focused on the problem of quantifying the ambiguity of the values measured by object-based metrics. The section provides a framework for processing user descriptions of regions, which are used as inputs for object-based metrics, and quantifying the ambiguity of the values measured by object-based metrics. It defines characteristics of metrics which describe quantitatively the ability of a metric to distinguish two groups of UIs objectively (e.g., the group of well-designed and randomly chosen dashboards).

7.2.1 Processing User Descriptions of Regions

Figure 7.2 visually explains the four steps of processing of the user descriptions of regions by a researcher:

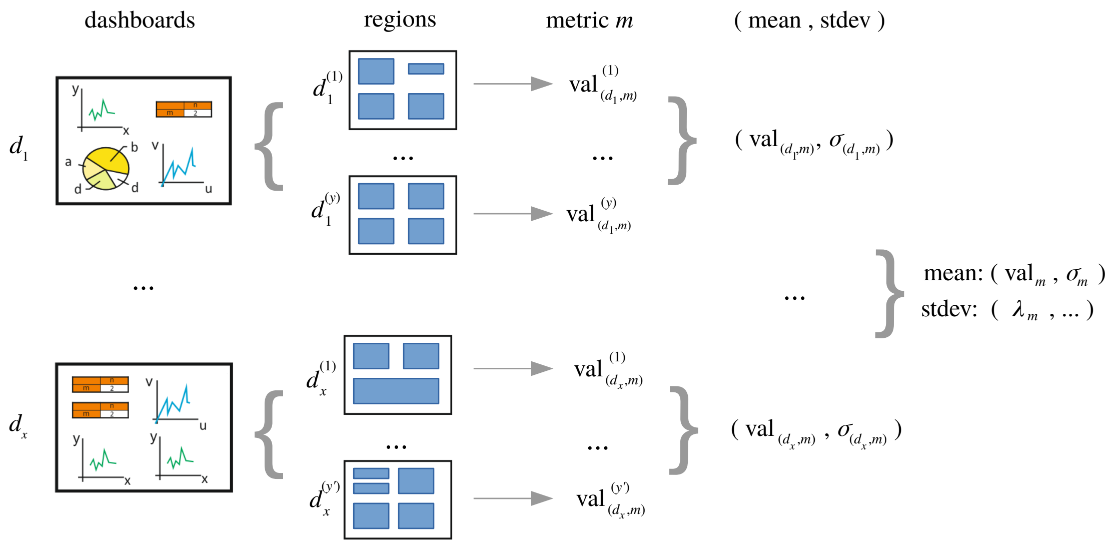

dashboards
regions
metric m
(mean , stdev)
d_1
$d_1^{(1)}$
$\text{val}^{(1)}_{(d_1,m)}$
...
$d_1^{(y)}$
$\text{val}^{(y)}_{(d_1,m)}$
($\text{val}_{(d_1,m)}$, $\sigma_{(d_1,m)}$)
...
...
mean: (val_m , σ_m)
stdev: (λ_m , ...)
d_x
$d_x^{(1)}$
$\text{val}^{(1)}_{(d_x,m)}$
...
$d_x^{(y')}$
$\text{val}^{(y')}_{(d_x,m)}$
($\text{val}_{(d_x,m)}$, $\sigma_{(d_x,m)}$)
x
y
u
v
a
b
d
m
n
2

Figure 7.2: The process of measuring the values of dashboard regions ($x = 130; y \sim 39$). Every user description is used as the input for a metric m. The measured values were used to create average values and standard deviations.

1. At the beginning, the researcher uses a metric m to measure values $\text{val}_{(d_i^{(u)},m)}$ for every description of regions $d_i^{(u)}$ of dashboard $d_i \in D$ provided by a users u. The values of a dashboard d_i are grouped in a set of values $V_{(d_i,m)}$. Since I worked with 130 dashboards, I created 130 sets $V_{(d_i,m)}$.
2. Then, it is appropriate to remove the values $\text{val}_{(d_i^{(u)},m)}$ with the highest distance from the average value of $V_{(d_i,m)}$. The reason is to filter the values calculated from the most extreme descriptions of regions. I decided to remove 10% of the values of every $V_{(d_i,m)}$.

3. Every filtered set of values $V'_{(d_i,m)}$ is used to calculate the average value $\mu_{V'_{(d_i,m)}}$ (simply $\text{val}_{(d_i,m)}$) with its standard deviation $\sigma_{(d_i,m)}$. Since I worked with 130 dashboards, I calculated 130 average values and standard deviations.

4. Finally, the average value $\text{val}_{(d_i,m)}$ is used to calculate one average value val_m with its standard deviation λ_m. Similarly, the standard deviations $\sigma_{(d_i,m)}$ are used to calculate one average standard deviation σ_m.

The procedure can be repeated for the subsets of user interfaces (e.g., well-designed and random dashboards).

7.2.2 Metrics Characteristics

After the processing of the user descriptions of regions, the researcher can analyze the aggregated variables σ_m and λ_m and rate the influence of subjective perception on the applicability of the metrics:

- σ_m: This measures the average impact of subjective perception on the precision of a metric m. If the value of σ_m rises, there is more likely to be a greater difference between the values measured by the metric m for two independent descriptions of regions of one dashboard. I named this characteristic *metric volatility* (the opposite of *metric stability*).

- λ_m: This measures the ability of a metric m to distinguish dashboards. If the value of λ_m rises, there is more likely to be a greater difference between the values measured by the metric m for the descriptions of regions of two different dashboards. I named this characteristic *metric scalability.*

Metric stability together with metric scalability represents the characteristic which I named *metric subjectivity*:

$$\text{subjectivity}_m = \frac{\sigma_m}{\lambda_m}. \tag{7.3}$$

It measures the average impact of subjective visual perception on the precision of a metric m relative to the range of the most frequently measured values. This means that a high value of metric volatility can be compensated by a high value of metric scalability.

To rate the ability of metrics to distinguish one group of user interfaces from another (e.g., well-designed from random dashboards), the variable γ_m was established:

$$\gamma_m = \text{overlap}(\text{val}_m^{(A)}, \lambda_m^{(A)}, \text{val}_m^{(B)}, \lambda_m^{(B)}) \in [0, 1] \tag{7.4}$$

where the overlap function measures the overlapping coefficient of two normal distributions (of the groups A and B) represented by a mean val_m and a standard deviation λ_m. If the value of the overlapping coefficient λ_m rises, it will be more difficult to distinguish these two groups by the metric m.

Finally, the overall rates of the metric m are:

$$\text{objectivity}_m = \text{subjectivity}_m^{-1} = \frac{\lambda_m}{\sigma_m} \tag{7.5}$$

$$\text{decisiveness}_m = \gamma_m^{-1} \tag{7.6}$$

The more objective (stable and scalable) the metric is, the less subjectively skewed results the metric provides. The more decisive the metric is, the greater the difference between the two groups the metric can find.

7.2.3 Metrics Classification

The purpose of the framework is not to observe particular metric values of objectivity and decisiveness since they depend on the group of users and the set of analyzed samples chosen for this research. The goal is to categorize and compare the metrics with each other. For this purpose, the following classification was established:

- **Class 0:** The metric m which can quantify a particular aspect of a user interface according to a specified formula.
- **Class 1:** The metric m of Class 0 with a *high value* of objectivity$_m$ which is able to consider the subjectivity of visual perception to a specified extent.
- **Class 2:** The metric m of Class 1 with a *high value* of decisiveness$_m$ which is able to distinguish two kinds of user interfaces to a specified extent.

The definitions of Class 1 and Class 2 do not intentionally contain specifications as to what the high values of objectivity and decisiveness are because they might be different for another experiment. For this research, I set the limit of both high values to be 2.0 (λ_m will be at least 2 times higher than σ_m; γ_m will be lower than 0.5). I chose rather weak limits.[1] However, these limits might be modified for future experiments.

7.3 Analysis of Ngo's Metrics of Aesthetics

The third part of the research task used the framework described in Section 7.2 to analyze the impact of the users' subjective perception on the ability of the 13 object-based metrics of aesthetics designed by Ngo et al. [2000a] to detect well-designed dashboards objectively.

7.3.1 Procedure

Firstly, I used the API of Dashboard Analyzer presented in Subsection 5.3.1 to implement all 13 Ngo's metrics of aesthetics. Then, I used the tools of Dashboard Analyzer to measure the values of all user descriptions of regions gathered in the study described in Section 7.1. The measured values were used to calculate the values of objectivity and decisiveness according to the framework described in Section 7.2. For measuring decisiveness, the set of dashboards was divided into the group of 9 well-designed ($D_{(\text{well})}$) and 121 random ($D_{(\text{rand})}$) dashboards.

Finally, the values of objectivity and decisiveness were used to compare Ngo's metrics and classify them into Class 0, 1, or 2 as described in Subsection 7.2.3. The final classification of the metrics was compared to the categorization of metric dependency described in Subsection 3.3.3 which distinguishes three sets of metrics:

- Ω_{AD} = {Balance, Equilibrium, Symmetry, Sequence, Density, Rhythm, and Unity}: the metrics which depend on the accuracy of regions' areas and the distribution of regions on a screen.
- Ω_{AR} = {Cohesion and Proportion}: the metrics which depend on the aspect ratios of regions.

[1] For instance, we need to consider that the group of randomly chosen dashboards might also contain well-designed dashboards, which might increase the value of γ_m.

- Ω_{G} = {Unity, Simplicity, Regularity, Economy, and Homogeneity}: the metrics which depend on the level of screen granularity.

7.3.2 Results

The first results, presented in Figure 7.3, describe the metrics objectivity. It was the first characteristic which was analyzed in order to distinguish the metrics of Class 0 and Class 1.

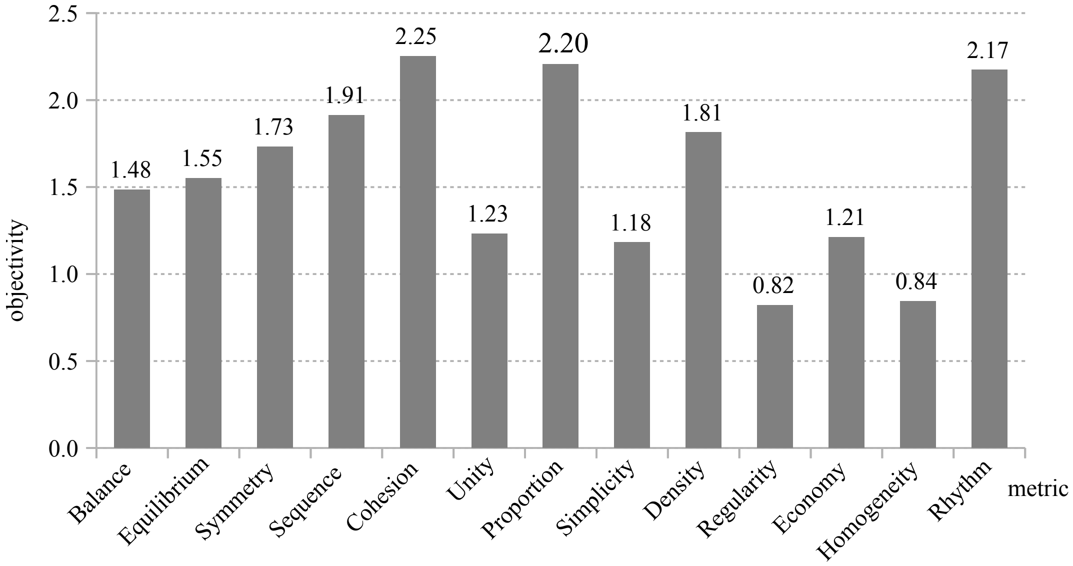


Figure 7.3: The values of metric objectivity measured for all Ngo's metrics.

The values of objectivity correlate with the categorization of metrics dependency. The metrics based on the analysis of screen granularity (Ω_{G}) have low values of objectivity, close to 1.0. The low rate of objectivity was expected because of the results of the study of visual perception of regions (the users segmented the screen with a different granularity). The results indicates that it might be complicated to use the metrics of Ω_{G} for a comparison of dashboard aesthetics. We can classify the metrics of Ω_{G} as members of Class 0.

On the other hand, the values of objectivity of the metrics based on the analysis of the aspect ratios of regions (Ω_{AR}) are higher than 2.0. It appeared that the subjective perception of the users had a low impact on the metrics of Ω_{AR}. Hence, we can consider the metrics of Ω_{AR} as members of Class 1.

The remaining six metrics based on the analysis of the area and distribution of regions on a screen (Ω_{AD}) appeared to be more objective than the metrics based on the analysis of screen granularity. The results correspond to the low average entropy measured in Experiment 1. However, except for Rhythm, the values of their objectivity are lower than 2.0, which makes them members of Class 0.

The next results, presented in Figure 7.4, describe the metrics decisiveness. Since only three metrics were categorized as members of Class 1—Cohesion, Proportion, and Rhythm—only these metrics could become members of Class 2. However, as shown in Figure 7.4, the values of decisiveness are low, except for one metric: Density. Thus, it would be complicated to use Ngo's metrics for the detection of the well-designed samples.

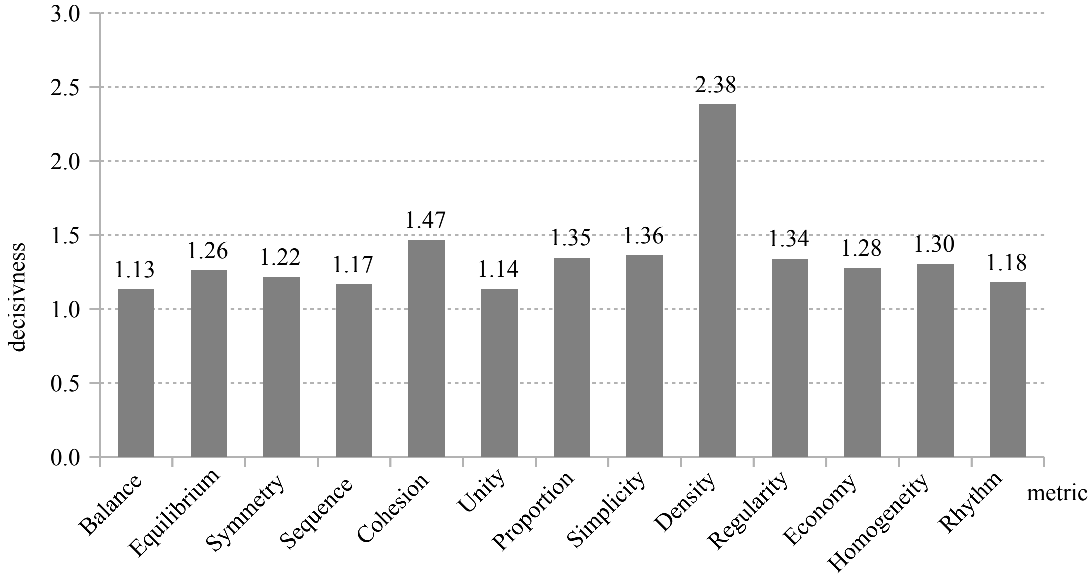


Figure 7.4: The values of metric decisiveness measured for all Ngo's metrics.

One possible reason for the low rates of decisiveness might be the insufficient number of well-designed samples. In addition, the group of randomly chosen dashboards may contain well-designed dashboards, which would make it harder to distinguish known well-designed samples. Finally, we need to consider the possibility that the characteristics of dashboards does not relate to the categorization of the selected samples into $D_{(\text{well})}$ and $D_{(\text{rand})}$.

7.3.3 Limitations

The results are limited by the descriptions of regions provided by the users, as described in Subsection 7.1.3. It should be evaluated whether a set of different descriptions of regions would lead to similar results.

The second limitation is caused by the chosen set of dashboard samples similarly as it was in the analysis of pixel-based metrics described in Section 6.5. The set of well-designed dashboards contains a small number of samples based on Few's design heuristics since not many examples are available. It should be evaluated whether the results of decisiveness would be similar for a different set of well-designed dashboard samples.

7.3.4 Conclusions

The experiment confirmed the problem regarding the low objectivity of several object-based metrics. UI designers should use the metrics based on the analysis of screen granularity with close attention. On the other hand, the metrics based on the analysis of the aspect ratios of regions (Cohesion and Proportion) or the distribution of regions on a screen (Rhythm) seem to be more immune to the subjective perception of the users. However, their application for the detection of well-designed dashboard samples is highly questionable.

7.4 Summary

The research task pointed out the fact that the ambiguous definition of UI objects can complicate the application of object-based metrics for evaluation of dashboard quality (and quality of a user interface in general). A different recognition method, even a different user, can define UI objects differently. The results have shown that users tend to perceive visually emphasized objects of a dashboard in a similar but not the same manner. Objects are usually composed of several simple graphical shapes clustered preattentively by the human brain, making logical parts of a screen (as described by Gestalt laws). The level of screen granularity was usually the main subject of disagreement between the users. It complicates the application of object-based metrics for evaluation of user interfaces—especially those which depend on the number of objects.

Then, the chapter described the framework which helps to quantify the impact of perception ambiguity on object-based metrics. The framework provides instructions on how to process subjective user descriptions of regions and use the data to measure the ability of a metric to distinguish a chosen kind of user interface from other kinds objectively. For this purpose, the two metric characteristics were presented: metric objectivity and decisiveness. It is expected that the proposed framework can be applied for evaluation of different metrics with a combination of various kinds of user interfaces.

Finally, the framework was used to evaluate the applicability of the 13 Ngo's metrics of aesthetics for measuring the design quality of dashboards. The results have shown that only the subset of Ngo's metrics (the metrics analyzing aspect ratios, areas and distribution of regions on a screen) can deal with the ambiguous description of regions. None of them were able to distinguish the well-designed samples from the group of randomly chosen dashboards objectively. Chapter 8 proposes a solution to the problem of the low values of the metrics' objectivity and decisiveness.

Chapter 8

Design and Improvement of Metrics

The goal of this research task was to find a solution for the problem presented in Chapter 7 regarding the inability of Ngo's metrics to distinguish the well-designed samples from the group of randomly chosen dashboards objectively. The chapter deals with the problem of the design of new metrics for evaluation of UI quality and improvement of existing ones.

Section 8.1 introduces a framework which defines the process of design and improvement of metrics. It describes generation of UI samples and gathering a user experience about the UI samples which is used to find important visual characteristics of UI and train metrics measuring those characteristics. Section 8.2 applies the framework in the analysis of the impact of color, type, and a dataset of UI objects on the perception of UI balance and symmetry. It describes a small-scale study gathering the user experience using dashboard samples generated by Generator of Dashboard Samples described in Subsection 5.3.2. Finally, Section 8.3 uses the user experience and proposes several approaches for measuring the color weight of UI objects. It tests the approaches in the improvement of selected Ngo's metrics.

The study gathering the user experience was performed in cooperation with Olena Pastushenko. The original goal was to test the applicability of Generator of Dashboard Samples, which was developed as part of Pastushenko's master's thesis supervised by the author of this thesis [Pastushenko, 2017]. The results of Section 8.1 and 8.2 were published by Pastushenko, Hynek and Hruška [2018, 2019]. The results of Section 8.3 were published in [Hynek and Hruška, 2018].

8.1 Framework for Design and Improvement of Metrics

A straightforward way to create a metric for automatic evaluation of the UI quality and usability would be training an artificial intelligence (e.g., based on deep neural networks) which would load a UI bitmap, consider pixel values as input features and directly decide whether the UI is usable or not (e.g., return a rate of UI quality). The problem of such an approach is that the quality and usability of UIs is a highly subjective characteristic which depends on the actual requirements of users. Training of such artificial intelligence would require a large training set of UI samples rated by a sufficiently large number of people. Such AI would not be probably applicable for evaluation of different kinds of UI. Finally, the evaluator would not receive any meaningful feedback describing reasons of the results of evaluation.

A different approach is to find a set of simpler metrics which would be replicable for evaluation of different kinds of user interfaces. Then, the goal of the researchers is to find UI characteristics which:

- have an impact on the quality and usability of user interfaces
- are measurable (can be derived from the description of UI)

This approach can provide the evaluator closer information about design problems of a UI. It is not, however, easy to find such a characteristic and design a quantitative metric measuring the characteristic. The researchers should have a basic conception of characteristics which they expect to be influential in the quality of a specific kind of user interface. Then, they can define a hypothesis about a UI characteristic, generate appropriate UI samples and get a user experience to prove or disprove the hypothesis. Finally, the user experience could be used for design and improvement of metrics. The problem is that the initial conception of researchers requires knowledge based on the user experience which is, however, the aim of the research. This makes the cyclic dependency shown in Figure 8.1.

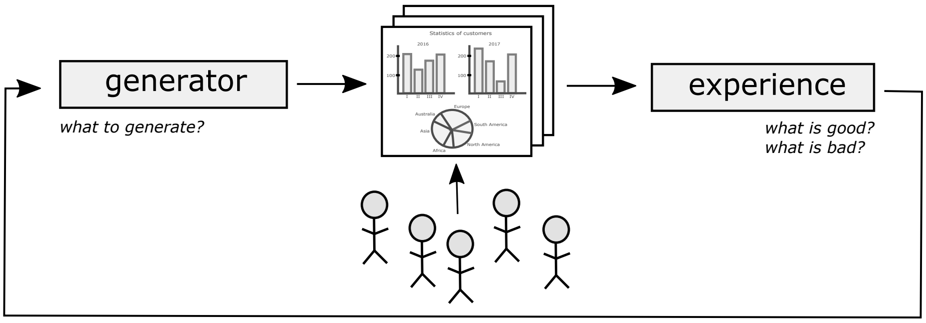


Figure 8.1: The cyclic dependency. We need to have a user experience to generate an appropriate set of UI samples which can be used to gain the user experience.

The framework for design and improvement of metrics proposes a solution to the problem of the cyclic dependency. It defines the process of iterative improvement of a model of UI internal representation and incremental generation of improved sets of UI samples which can be used for extension of the knowledge about user experience and improvement of metrics. Figure 8.2 describes the process, which consists of 2 parts:

1. **Improvement of UI model.**
   (a) In the beginning, the generator of UI samples works with a simple model of UI internal representation which allows specifying only dimensions and types of UI objects. An example of such a model has been described in Subsection 5.2.2, Listing 5.3. The model provides the fundamental variability of generated samples. The rest of the information is derived from the implicit appearance of UI.

   (b) Then, the researchers use the model to generate an initial set of UI samples which are used for making an initial conception of the UI characteristics which are important for the perception of UI quality.

(c) The model is extended iteratively according to the user experience which is obtained from the analysis of the user reviews of the generated samples. The more comprehensive the model is, the higher level of the variability of samples the generator provides. Researchers can use the constraints to filter unimportant types of samples when they want to analyze a particular design aspect.

2. **Improvement of metrics and design guidelines:** The lower part of Figure 8.2 describes the applicability of the user experience in the construction of the quantitative design guidelines which evaluate a UI in terms of the ratios of UI characteristics measured by metrics. Depending on a situation, the guidelines can represent a simple threshold or an advanced classification algorithm. Researchers should deal with the following problems:

   (a) A real UI can be represented in various formats (a raster image or a structured description—e.g., a web page). The researchers should find a way how to convert the original UI into internal representation compatible with the UI model which is recognized by the metrics. The model needs to be simplified, which might limit the metrics.

   (b) Researchers should use the user experience to improve the metrics, so the metrics reflect the user perception (of the characteristic or the overall quality of the UI).

   (c) Researchers should use the user experience to improve the design guidelines, so the guidelines reflect user perception and are able to use results of metrics for analysis of design problems, classification or rating of a user interface.

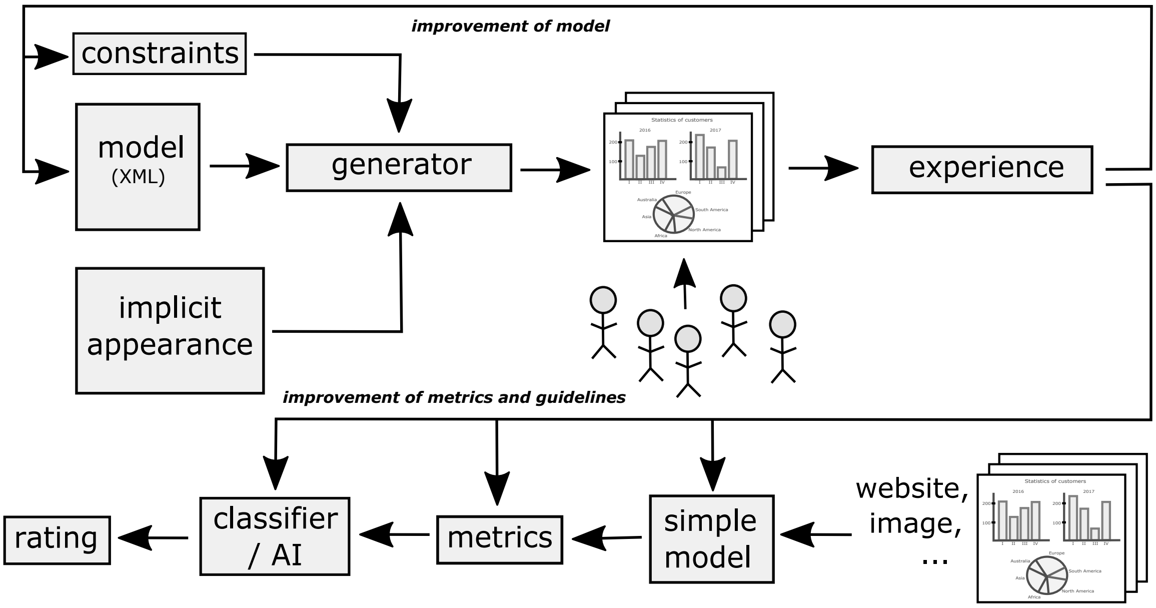


Figure 8.2: The scheme of the construction and evaluation of design guidelines. We can use the user experience for the improvement of the model (for the generation of better samples) and the improvement of the design guidelines.

8.2 Study of Color Impact on Object-based Metrics

One of the weaknesses of object-based metrics of aesthetics analyzed in Chapter 7 is that they consider only the dimensions of regions (position and size). Ngo et al. [2000a] suggested that object-based metrics of aesthetics should increase the scope of the measuring and also consider the color or shape of objects. For instance, they suggested that:

- the black color is visually heavier than the white color
- irregular shapes are visually heavier than regular shapes

Such aspects might have an impact, for instance, on the metrics analyzing the distribution of visual weight of a screen, like Balance or Symmetry. Figure 8.3 explains the hypothesis.

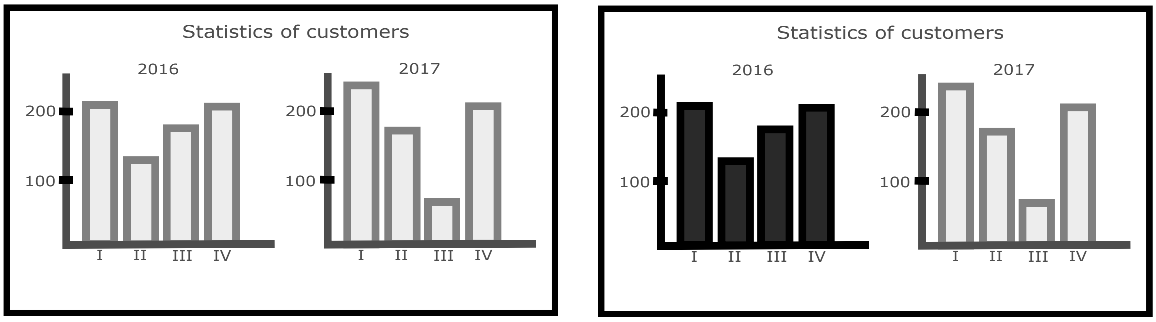


Figure 8.3: Impact of color on the metrics of aesthetics. Both screens would be rated by the same value of Balance and Symmetry according to formulas of [Ngo et al., 2000a] since the formulas consider only the dimensions of widgets. On the other hand, the hypothesis expects that a user would distinguish these two screens since they consist of a different color. The user would rate the left screen as more balanced and symmetrical than the right one because of unbalanced and asymmetrical color distribution in the right screen.

One possible way how to support the hypothesis is to perform a study based on user reviews. This research task used the framework to and let a small group of people rate the impact of color and shape of widgets on the perception of the balance of dashboards.

8.2.1 Procedure

The dashboard samples were generated by the Pastushenko's generator of dashboard samples described in Subsection 5.3.2. Firstly, we created four layouts: d_1, d_2 for the evaluation of the Balance metric, and d_3, d_4 for the Symmetry metric (Figure 8.4). The layouts can be considered as highly balanced and symmetrical with respect to the formulas of Balance and Symmetry presented by Ngo et al. [2000a]. Then, we generated a few realistic-looking dashboards for every layout. The dashboards varied in color (color intensity and hue), chart types (bar charts, line charts, and bullet graphs) and dataset, which changes the look of widgets as well (e.g., a higher value is represented by a larger bar, which increases the area occupied by the particular color value).

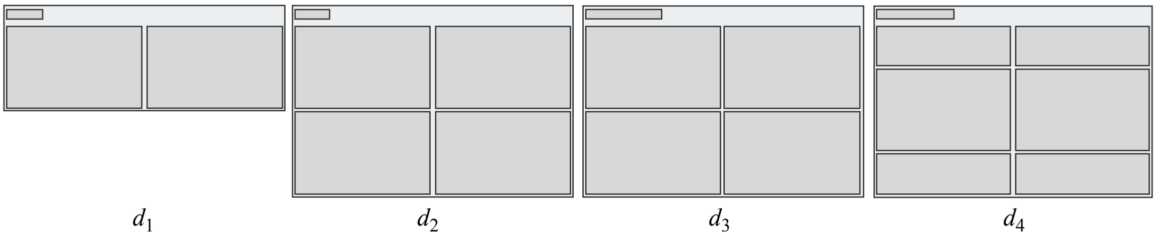


Figure 8.4: The four layouts used for the analysis of user perception and comparison of the users' reviews with results of metrics. They are highly (but not absolutely) balanced and symmetrical. Then, we replaced the layout regions with different charts, colors and dataset and asked the users to rate Balance and Symmetry of the screen. The layouts d_2 and d_3 differ in margins, and they are used for different metrics.

Then, we let users rate the UI samples. The evaluation of the Balance metric was performed with 12 users; the evaluation of the Symmetry metric with a different group of 13 users. Both evaluations were independent. The demographic and professional characteristics of participants varied in both cases. The age of participants was 22-50 years, experience with information technology varied. Some of them were students, some of them were employed, and the rest of them had only minimal experience with information technology.

The first group of users was asked to rate the horizontal and vertical balance (using the 5-point scale: $\{-2, \ldots, 2\}$; -2: left/bottom side is heavier; 0: balanced; 2: right/upper side is heavier). The second group was asked to rate the overall symmetry (using the 5-point scale: $\{0, \ldots, 5\}$; 0: very low; 5: very high). The participants of both groups used Interactive Survey Tool described in Subsection 5.3.3, which let them quickly select locations of perceived equilibriums or rate the dashboards. Finally, we compared the results of the metrics with the reviews of the users.

8.2.2 Results

All results including the generated dashboard samples are available in Appendix A.3.

Balance

Figure 8.5 presents the average values of the horizontal and vertical balance. The d_1 layout tested the change in the perception of the horizontal balance. The layout d_2 tested both axes—the horizontal and vertical one. Since the reference layouts d_1, d_2 were composed only of gray rectangles, users perceived them as highly balanced. Replacing the rectangles by real widgets did not change the level of perceived balance of $d_1^{(1)}$ much. However, even a low change of color intensity of the chart on one side caused a high deviation from the equilibrium ($d_1^{(2)}$). Increasing the color intensity on one side made the layout even more unbalanced. On the contrary, color hue had a low impact on the perceived balance ($d_1^{(5)}$).

Similar results were observed in the d_2 layout. In contrast to the d_1 layout, readers can notice the change in the vertical and horizontal value of balance. The d_2 layout was also used to analyze the impact of widget types. It used a bar chart and line chart. The bar chart was perceived as visually heavier than the line chart because the rectangles of bar charts usually occupy a larger area of the screen than the lines of line charts.

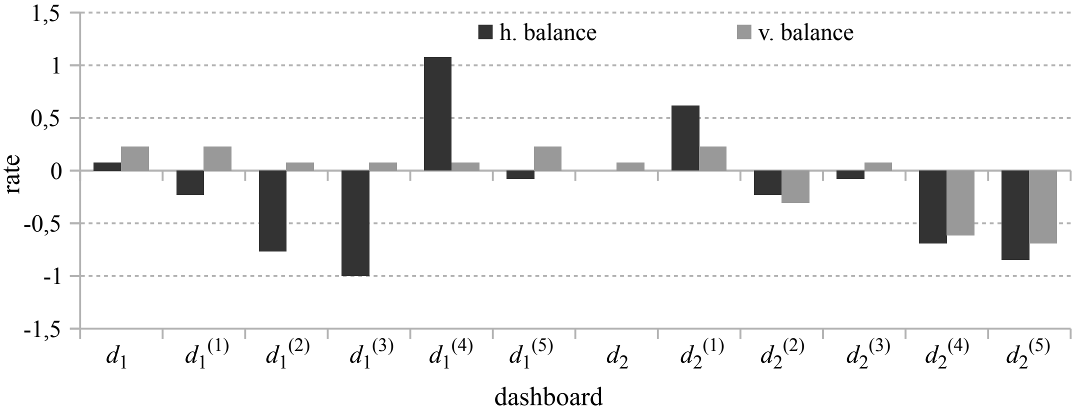


Figure 8.5: The average values of the horizontal and vertical balance perceived by the users for the two layouts d_1 and d_2. The layouts are highly balanced (close to zero) on the contrary to the other dashboards using the same layout but different colors and widget types.

Symmetry

In contrast to the evaluation of UI balance, the evaluation of UI symmetry did not analyze every axis separately (for instance, Ngo et al. [2000a] consider the vertical, horizontal and radial axes). The users directly rated the overall symmetry. Figure 8.6 presents the results. The reference layouts d_3 and d_4, which were composed of gray rectangles, were rated as highly symmetrical. Then, we replaced the rectangles with real widgets of the same type and color. The average values of symmetry of these screens were even slightly higher ($d_3^{(1)}$, $d_4^{(1)}$). Then, we modified widgets of one side o UI, particularly: dataset ($d_3^{(2)}$, $d_4^{(2)}$), color ($d_3^{(3)}$, $d_3^{(4)}$, $d_4^{(3)}$, $d_4^{(4)}$) and type ($d_3^{(5)}$). All modifications caused a decrease in the perceived value of symmetry. The results of the evaluation showed a similar tendency as the results of the evaluation of UI balance. Color, type of widgets and the displayed dataset have impact on the perceived weight of objects.

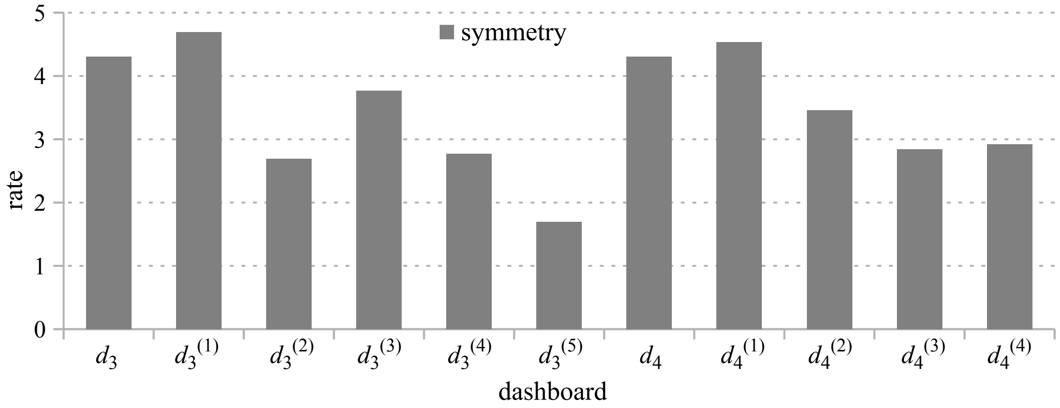


Figure 8.6: The average values of the overall symmetry perceived by the users for the two layouts d_3 and d_4. The reference layouts are highly symmetrical (close to 5) on the contrary to the other dashboards using the same layout but different colors, widget types, and dataset.

8.2.3 Limitations

The small numbers of samples and users represent the main limitations of the presented evaluation. On the other hand, the primary purpose of the evaluation was not to provide a large-scale study of object-based metrics. The primary purpose was to evaluate and demonstrate the applicability of the designed workflow and generator. We successfully gained the experience of the users.

We performed two studies with two groups of users analyzing different visual characteristics. Specialization and age of the users varied. We did not find any correlation between the results and characteristics of the users. However, it would be useful to perform a further study with more users analyzing the impact of users characteristics (e.g., age, or specialization) on their perception of a UI. For instance, perception of people having skills in the art might be different for other people.

8.2.4 Conclusions

The study analyzed the impact of color, type of widgets and displayed dataset on the perception of layout balance and symmetry. The results of the two independent user reviews confirmed the impact of these factors. The users tended to perceive the charts using highly intense colors as more weighty comparing the charts using less intense colors. Similarly, the charts containing large graphical elements (e.g., bar charts) were perceived as more weighty than the charts composed of thin lines (e.g., line charts). Finally, the displayed datasets affected graphical elements of charts (e.g., size of bars), which also affected the perception of the weight of charts.

The findings of the study are important for the application of the object-based metrics which evaluate the weight of UI objects. We can use the results for the improvement of Ngo's metrics—e.g., the Balance and Symmetry metrics. One possible suggestion is to consider the average color intensity of widgets in the formulas.

8.3 Improvement of Ngo's Object-based Metrics

The goal of the last part of the research task was to use the user experience gained in the study described in Section 8.2 and propose an improvement of Ngo's object-based metrics. It focused on the group of metrics based on the distribution of regions on a screen (Ω_{AD})—particularly on the Balance metric, which was rated as the least objective metric of this group. I tried to find a possible approach which would decrease the impact of the subjective perception of the users on the characteristics of metrics explained in Chapter 7 (metric objectivity and decisiveness). The approach combines the object-based metric with the objective pixel-based analysis of the color distribution on a screen.

8.3.1 Procedure

Ngo et al. [2000a] computes Balance as the difference between the total weighting of the components on each side of the horizontal and vertical axes as described in Formula 3.6. The weighing of a side is computed as:

$$w_j = \sum_{i}^{n_j} a_{ij} d_{ij} \qquad (8.1)$$

which means that it depends on the values a_{ij} representing the area of the region and d_{ij} representing the distance of the region from the center of the UI. It has been shown in Section 7.1 that these values might be ambiguous. The users usually agreed about the approximate area and distribution of regions on a screen, but they did not usually specify these regions with exactly the same precision.

The idea of the improvement is to include objective information about the color of subjectively specified regions in the Formula 8.1 in order to affect weightings of the regions objectively. Hence, I modified the formula of the Balance weighting:

$$w_j = \sum_{i}^{n_j} a_{ij} d_{ij} C_{ij} \tag{8.2}$$

where C_{ij} is the coefficient of color of a region i in a quadrant j representing the colorfulness of the region. Since two sides of a screen are always compared to each other, there is no problem in modifying the weightings of each side by adding C_{ij} to the formula and keeping the range of the formula: $[0, 1]$. I explored several approaches to measuring the coefficient of color using different color spaces:

- $C_r^{(1)} \in [0, 1]$: The average color intensity of a region r represented in the 8-bit grayscale color space converted from the RGB color space.

- $C_r^{(2)} \in [0, \inf]$: The average colorfulness of a region r inspired by [Yendrikhovskij et al., 1998; Reinecke et al., 2013]: $C_r = S_r + \sigma_r$ where S_r is a value of the average saturation of a region r in the CIE L*a*b* color space and σ_r is its standard deviation.

- $C_r^{(3)} \in [0, 1]$: The average value of all pixel values in a region r calculated as $1 - (b_i - b_i s_i)$ where $s_i \in \{0, 1\}$ is the saturation and $b_i \in \{0, 1\}$ is the brightness of the i-th pixel of the region in the HSB color space. The formula is based on the suggestion of [Ngo et al., 2000a] that users might assign visual importance to pixels with high saturation or low brightness. Figure 8.7 visualizes the dependency of the $C_r^{(3)}$ on the HSB saturation and brightness.

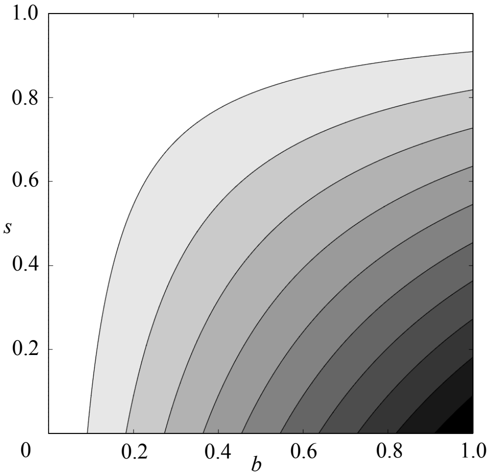
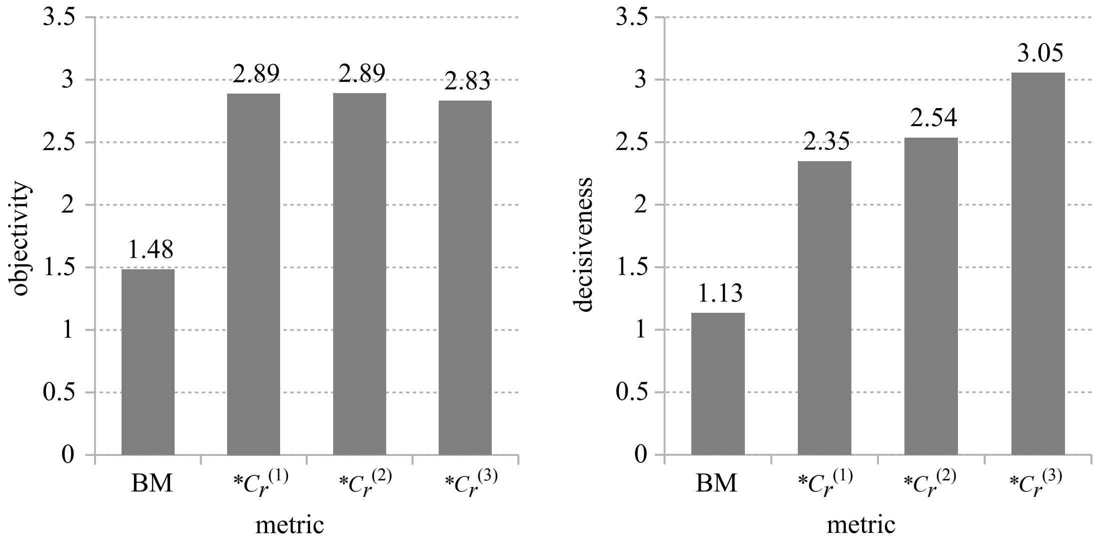


Figure 8.7: A contour plot visualizing the function of the coefficient of color $C_r^{(3)}$ based on the HSB saturation s and brightness b. The higher color intensity of the plot represents the lower value of $C_r^{(3)}$.

8.3.2 Results

Figure 8.8a and 8.8b present the results. We can see a significant improvement in all kinds of the coefficient of color. The values of objectivity and decisiveness are higher than 2.0, which makes Balance a member of Class 2. The best results were received for $C_r^{(3)}$ calculated according to the formula using the rate of saturation and brightness in the HSB color space, followed by the results for $C_r^{(2)}$ considering the colorfulness calculated in the CIE L*a*b* color space. From a practical point of view, the easiest method improving Balance is to use the coefficient $C_r^{(1)}$ based on the color intensity, since the color intensity can be simply calculated from the RGB color space. The color intensity might also correspond better with the perception of color blind people [Few, 2006]. In addition, the infinite range of $C_r^{(2)}$ might cause problems with the modification of some metrics.



(a) Balance objectivity

(b) Balance decisiveness

Figure 8.8: Change of the Balance objectivity and decisiveness for the metrics using the coefficients of color: $C_r^{(1)}$, $C_r^{(2)}$, and $C_r^{(3)}$.

Table 8.1 presents the average values and standard deviations of Balance measured for the well-designed and randomly chosen dashboard samples. The well-designed dashboards are more balanced than the randomly chosen ones for all types of Balance, including the modified ones. This confirms the results of the pixel-based Balance presented in Section 6.4. We can see the decrease in $\text{val}_{D_{(\text{rand})}}$ for the modified versions of Balance. This indicates that the modified versions of Balance are stricter than the original Balance. Since the original Balance rated some dashboards as balanced, the modified versions of Balance rated the dashboards as unbalanced because of their unbalanced distribution of color on a screen.

Table 8.1: The average values of UI balance (val) with their standard deviations (λ) for the groups of well-designed ($D_{\text{(well)}}$) and random dashboards ($D_{\text{(rand)}}$).

Metric	$\text{val}_{D_{\text{(well)}}}$	$\lambda_{D_{\text{(well)}}}$	$\text{val}_{D_{\text{(rand)}}}$	$\lambda_{D_{\text{(rand)}}}$
BM	0.873	0.107	0.843	0.107
$\text{BM}(C_r^{(1)})$	0.830	0.086	0.640	0.197
$\text{BM}(C_r^{(2)})$	0.819	0.072	0.643	0.182
$\text{BM}(C_r^{(3)})$	0.845	0.068	0.651	0.191

8.3.3 Limitations

The results are limited by the descriptions of regions similarly as the results of the analysis of the basic Ngo's metrics described in Subsection 7.3.3. Also, the improvement focuses only on the colorfulness of rectangular regions. It does not capture the image complexity (e.g., shapes of widgets), which affects user perception as well. It might be possible to perform edge detection and analyze the number of pixels representing the edges in regions.

8.3.4 Conclusions

The proposed improvement based on the inclusion of color in formulas increased the value of objectivity and decisiveness of the Balance metric. All three modified versions of the Balance metric are classified in Class 2, as the metric which is able to distinguish well-designed dashboards objectively. The coefficient of color based on the saturation and brightness measured in the HSB color space caused the highest increase of the metric ratings. It is expected that the other object-based metrics based on the distribution of regions on a screen (Ω_{AD}) could be improved using a similar approach. It would, however, require further evaluations.

8.4 Summary

This research task described the problem of design and improvement of metrics for measuring UI quality. The first part of the chapter introduced the framework describing the process of improvement of metrics and design guidelines. The main idea of the framework was based on the iterative extension of a UI model representing the internal representation of the UI. In the beginning, the model represents basic information about dimensions and types of UI components. The researchers extend the model by adding new attributes representing UI characteristics which are important for the evaluation of hypotheses about the impact of the characteristics on UI quality. Then, they generate UI samples varying in the UI characteristics described in the model and let users to rate the UI characteristics in order to gain a user experience. The user experience is used for further improvement of the UI model and design and improvement of metrics and design guidelines.

The second part of the chapter described the small-scale study which demonstrates usability of the framework. The study analyzed the impact of color, type of widgets and displayed dataset on the perception of the layout balance and symmetry. It used Generator of Dashboard Samples described in Subsection 5.3.2 to generate appropriate dashboard samples and Interactive Survey Tool described in Subsection 5.3.3 to get the user experience. Two independent user reviews confirmed the impact of these factors.

The third part of the chapter used the user experience for the improvement of the object-based metrics designed by Ngo et al. [2000a]. It proposed an improvement which combines object-based metrics with the pixel-based approach measuring the colorfulness of the interface regions. It demonstrated the approach on the improvement of the Balance metric. The improved metric was rated as objective and able to recognize the well-designed dashboard samples. It is expected that the proposed model can be generalized and applied for the evaluation of other metrics with a combination of other kinds of user interfaces.

The modified version of Balance using the coefficient of color can be used for the improvement of the tools designed for metric-based evaluation of user interfaces. Since existing tools apply different approaches to detect regions, it might be appropriate to use a metric which considers possible ambiguity of the inputs. For the full automation of the evaluation, it is necessary to design a segmentation algorithm for the automatic detection of regions based on the average user perception analyzed in Section 7.1. The problem of automatic segmentation of dashboards is the aim of Chapter 9.

Chapter 9

Automatic Segmentation of Dashboards

Segmentation of dashboards into regions is the essential requirement for using object-based metrics. Section 4.2 described possible approaches of segmentation:

- manual segmentation by a user
- derivation from a structural description of UI (source code)
- segmentation of a bitmap (a UI screenshot)

The study of visual perception (described in Section 7.1) analyzed perception of 251 users who manually specified regions of selected dashboards. It showed the ambiguity of the users' perception. Using the approach based on manual segmentation would require a sufficiently large number of users so the evaluators could create a model of the average users' perception. Such an approach is not usable for frequent evaluations of different UIs.

The structural description of a UI does not need to be always available. It requires a special parser for every kind of structural description. Hence, the approach based on segmentation of a bitmap seems to be the more usable approach than the approach based on derivation from a structural description of UI. Segmentation of dashboard bitmaps, however, needs to deal with the complexity of dashboards. Dashboards usually consist of several color layers, and it is complicated to segment them by well-known page segmentation methods which are usually used for segmentation of printed documents. Also, segmentation of bitmaps needs to consider subjective perception of users and principles of objects grouping (as Gestalt laws described in Section 4.1).

The goal of this research task was to find a method for automatic segmentation of dashboards into the regions which can be used as the inputs for object-based metrics. Section 9.1 analyzes the problem using the data of the study of user perception described in Section 7.1. Section 9.2 presents a novel method for automatic segmentation of dashboard screen images which considers the user experience. Then, Section 9.3 presents an evaluation of the method. It compares the descriptions of regions created by the method with the descriptions of regions specified by the users and analyzes the application of the descriptions for the Balance metric. Finally, the limitations and improvements are suggested.

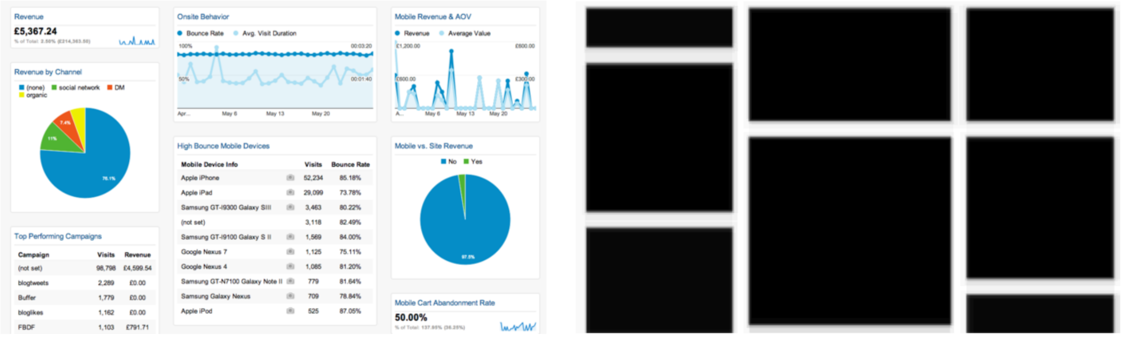
The results presented in this chapter were published in [Hynek and Hruška, 2019]. The method was implemented in the Java language and integrated into Dashboard Analyzer described in Subsection 5.3.1. Readers can find the reference to the project's repository in Appendix B.1.

9.1 Analysis

The method for segmentation of dashboards should reflect perception of users. The result of automatic segmentation should be as close to the average description as possible. Hence, the results of the study of user perception were used to analyze similarities of object recognition and grouping before designing the segmentation method.

The average descriptions indicated a high influence of the following Gestalt laws:

- **The Gestalt law of enclosure:** The users tended to group the screen elements which were explicitly grouped by a visually emphasizing frame. These frames are usually represented by borderlines or a different background, and they form a rectangular boundary of the widgets (Figure 9.1). Users group objects inside a boundary even if it is broken (the Gestalt law of closure).



Top Performing Campaigns

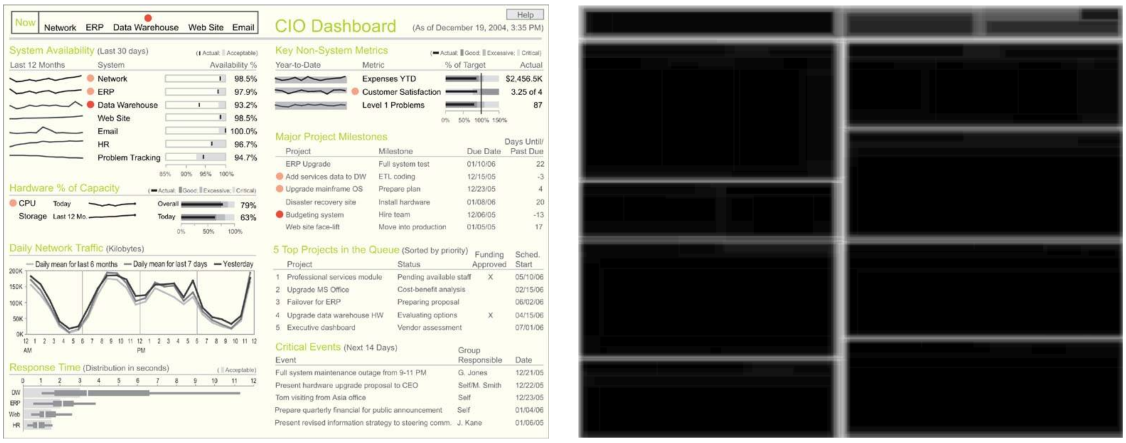
Campaign	Visits	Revenue
(not set)	98,798	£4,599.54
blogtweets	2,289	£0.00
Buffer	1,779	£0.00
bloglikes	1,162	£0.00
FBDF	1,103	£791.71

High Bounce Mobile Devices

Mobile Device Info	Visits	Bounce Rate
Apple iPhone	52,234	85.18%
Apple iPad	29,099	73.78%
Samsung GT-I9300 Galaxy SIII	3,463	80.22%
(not set)	3,118	82.49%
Samsung GT-I9100 Galaxy S II	1,569	84.00%
Google Nexus 7	1,125	75.11%
Google Nexus 4	1,085	81.20%
Samsung GT-N7100 Galaxy Note II	779	81.64%
Samsung Galaxy Nexus	709	78.84%
Apple iPod	525	87.05%

Figure 9.1: The average description of regions shows a high agreement about the regions.[1]

- **The Gestalt law of proximity:** Many dashboards avoid to use widget boundaries. It is not necessary to use boundaries since the viewers should group the widget parts which are close together. We can confirm this fact since our samples contain widgets without frames as well and the users grouped them (Figure 9.2).


Now
Network ERP Data Warehouse Web Site Email
CIO Dashboard (As of December 19, 2004, 3:35 PM)
Help
System Availability (Last 30 days)
(Actual; Acceptable)
Hardware % of Capacity
CPU Today Overall 79%
Storage Last 12 Mo. Today 63%
Daily Network Traffic (Kilobytes)
Daily mean for last 6 months
Daily mean for last 7 days
Yesterday
200K
150K
100K
50K
0K
AM
PM
Response Time (Distribution in seconds)
DW
ERP
Web
HR
Key Non-System Metrics
(Actual; Good; Excessive; Critical)

Last 12 Months	System	Availability %
	Network	98.5%
	ERP	97.9%
	Data Warehouse	93.2%
	Web Site	98.5%
	Email	100.0%
	HR	96.7%
	Problem Tracking	94.7%

Year-to-Date	Metric	% of Target	Actual
	Expenses YTD		$2,456.5K
	Customer Satisfaction		3.25 of 4
	Level 1 Problems		87

Major Project Milestones

Project	Milestone	Due Date	Days Until/ Past Due
ERP Upgrade	Full system test	01/10/06	22
Add services data to DW	ETL coding	12/15/05	-3
Upgrade mainframe OS	Prepare plan	12/23/05	4
Disaster recovery site	Install hardware	01/08/06	20
Budgeting system	Hire team	12/06/05	-13
Web site face-lift	Move into production	01/05/05	17

5 Top Projects in the Queue (Sorted by priority)

	Project	Status	Funding Approved	Sched. Start
1	Professional services module	Pending available staff	X	05/10/06
2	Upgrade MS Office	Cost-benefit analysis		02/15/06
3	Failover for ERP	Preparing proposal		06/02/06
4	Upgrade data warehouse HW	Evaluating options	X	04/15/06
5	Executive dashboard	Vendor assessment		07/01/06

Critical Events (Next 14 Days)

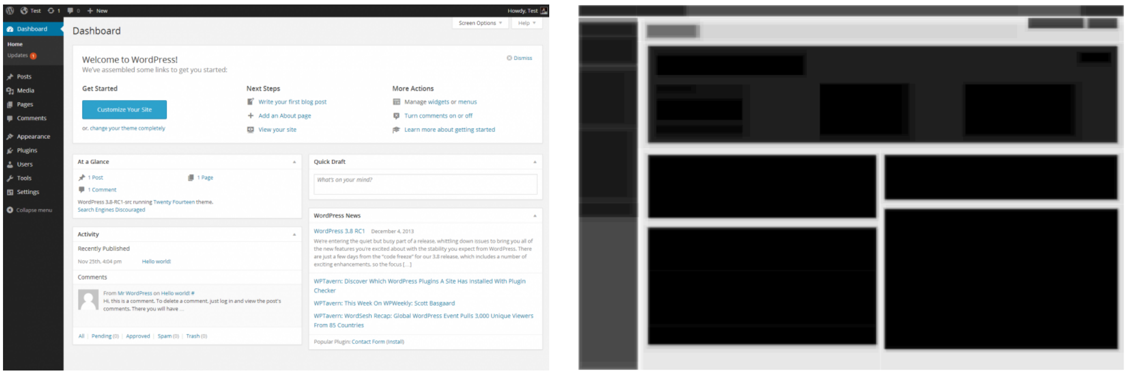
Event	Group Responsible	Date
Full system maintenance outage from 9-11 PM	G. Jones	12/21/05
Present hardware upgrade proposal to CEO	Self/M. Smith	12/22/05
Tom visiting from Asia office	Self	12/23/05
Prepare quarterly financial for public announcement	Self	01/04/06
Present revised information strategy to steering comm.	J. Kane	01/06/05

Figure 9.2: The average description of regions shows that users detected the regions without explicit boundaries. There is however a higher rate of ambiguity in the area between the regions. Source of the dashboard: [Few, 2006]

[1] Source of the dashboard: https://econsultancy.com/google-analytics-custom-dashboards/

The segmentation algorithm should consider similarities of the users' perception. On the other hand, the descriptions of regions exhibited the disagreement in granularity of screens. Some users considered large areas as solid regions. Other users split them into smaller coherent regions. This applies especially to management areas of dashboards (e.g., toolbars and headers against single buttons and labels), as shown in Figure 9.3. Ambiguity has also been monitored in the dashboards consisting of overlapping objects and objects with unclear borders (Figure 9.4)

Since it is not clear how these areas should be segmented, the segmentation algorithm does not need to be so strict in these cases. However, it should try to minimize the difference between the result of segmentation and average description. After the dashboard segmentation, evaluators should use sufficiently robust object-based metrics which are able to consider certain differences caused by subjective perception of users or imprecision of the segmentation (as suggested in Chapter 7).



Dashboard

Welcome to WordPress!
We've assembled some links to get you started:

or, change your theme completely

Write your first blog post
Add an About page
View your site

Manage widgets or menus
Turn comments on or off
Learn more about getting started

1 Post
1 Page
1 Comment
WordPress 3.8-RC1-src running Twenty Fourteen theme.
Search Engines Discouraged

Nov 25th, 4:04 pm Hello world!

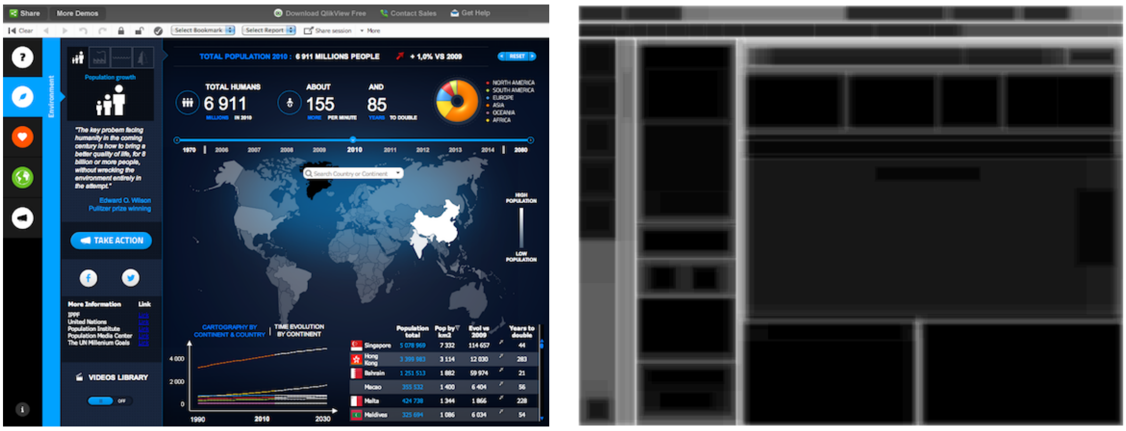
From Mr WordPress on Hello world! #
Hi, this is a comment. To delete a comment, just log in and view the post's comments. There you will have …

What's on your mind?

WordPress 3.8 RC1 December 4, 2013
We're entering the quiet but busy part of a release, whittling down issues to bring you all of the new features you're excited about with the stability you expect from WordPress. There are just a few days from the "code freeze" for our 3.8 release, which includes a number of exciting enhancements, so the focus […]

WPTavern: Discover Which WordPress Plugins A Site Has Installed With Plugin Checker
WPTavern: This Week On WPWeekly: Scott Basgaard
WPTavern: WordSesh Recap: Global WordPress Event Pulls 3,000 Unique Viewers From 85 Countries
Popular Plugin: Contact Form (Install)

Figure 9.3: The average description of regions shows ambiguity in the perception of dashboard's sidebar and header.[2]



TOTAL POPULATION 2010 : 6 911 MILLIONS PEOPLE + 1,0% VS 2009

"The key probem facing humanity in the coming century is how to bring a better quality of life, for 8 billion or more people, without wrecking the environment entirely in the attempt."
Edward O. Wilson
Pulitzer prize winning

More Information	Link
IPPF	Link
United Nations	Link
Population Institute	Link
Population Media Center	Link
The UN Millenium Goals	Link

	Population total	Pop by km2	Evol vs 2009		Years to double
Singapore	5 078 969	7 332	114 657	↗	44
Hong Kong	3 399 983	3 114	12 030	↗	283
Bahrain	1 251 513	1 882	59 974	↗	21
Macao	355 532	1 400	6 404	↗	56
Malta	424 738	1 344	1 866	↗	228
Maldives	325 694	1 086	6 034	↗	54

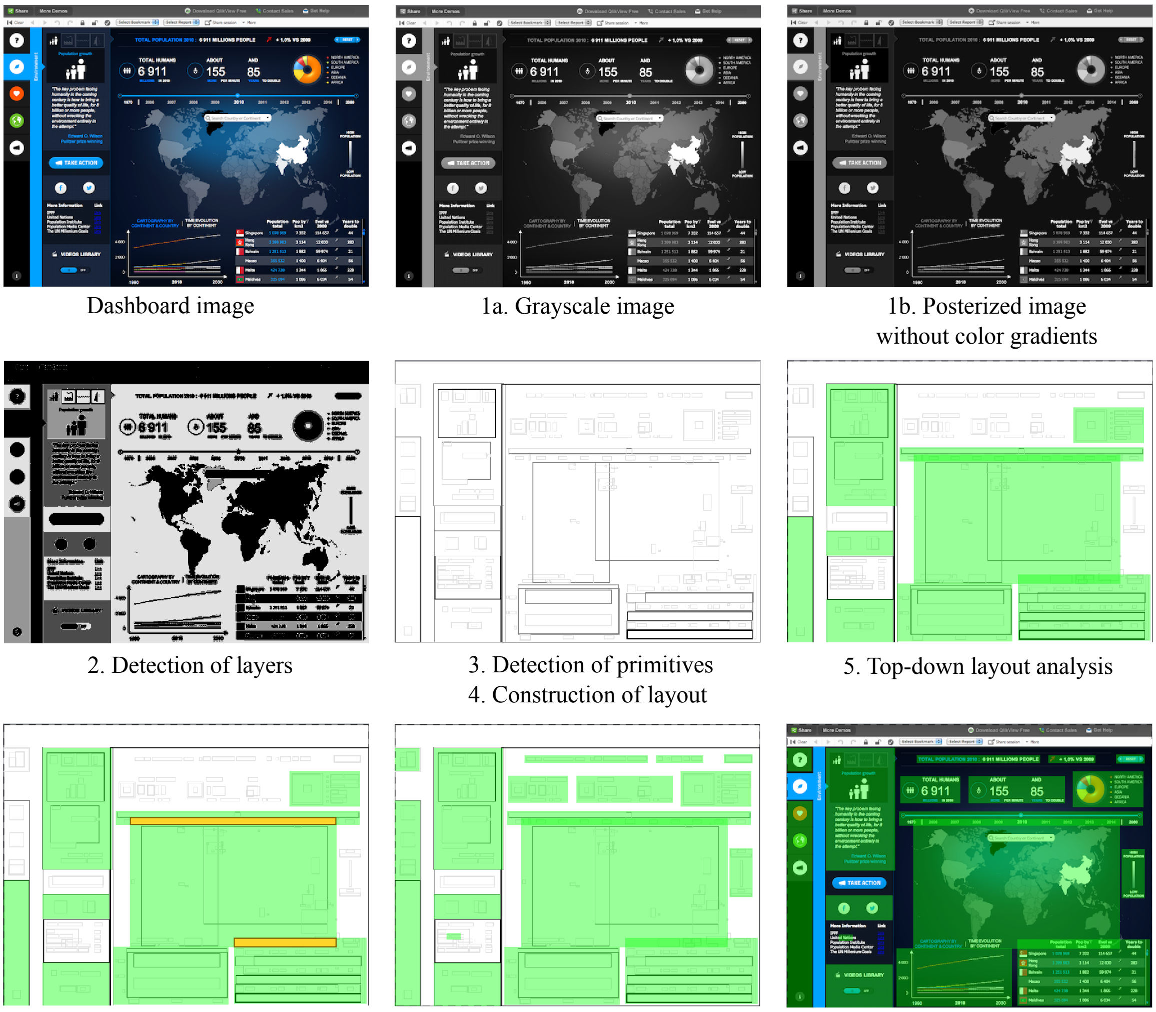
Figure 9.4: The average description of regions shows ambiguity in the perception of dashboard's body. The dashboard contains overlapping objects and color gradients, which might complicate automatic segmentation of the dashboard.[3]

[2] Source of the dashboard: Wordpress Admin's Dashboard—https://www.wordpress.com

[3] Author of the dashboard: Alexandre Perrot—https://us-b.demo.qlik.com/detail.aspx?appName=DoYouRealize_Keyrus.qvw

9.2 Method

The method for segmentation of dashboards consists of seven phases. Figure 9.5 demonstrates an example of the process of segmentation. The method initially focuses on reduction of image colors, which represents image layers. Then, it detects primitives which makes a screen layout. Finally, the method processes the screen layout using the combination of the top-down and bottom-up segmentation strategy and detects visually dominant regions. The following subsections briefly describe the phases. The readers can evaluate the phases using the source code which is available online (see Appendix A.3).



Dashboard image

1a. Grayscale image

1b. Posterized image without color gradients

2. Detection of layers

3. Detection of primitives
4. Construction of layout

5. Top-down layout analysis

6. Analysis of overlapping regions

7. Bottom-up layout analysis

Final selection of regions

Figure 9.5: An example of the segmentation of a highly colorful dashboard containing overlapping regions. Firstly, the method preprocesses the image, reduces the number of colors and detects the color layers. Then, it constructs the layout and finds the visually dominant regions (represented by the green rectangles). Readers can notice that the method ignores some widgets, especially in the management areas. The sixth phase does not affect the regions since the dashboard does not contain any highly overlapping regions.

9.2.1 Phase 1: Image Preprocessing

The image preprocessing is done in three steps:

1. In the beginning, the method converts a dashboard bitmap into the 8-bit grayscale color space representing color intensity to reduce the number of colors to 256. Color intensity is more suitable for a further analysis than color hue since it respects the problems with visual perception of colorblind people better.

2. Then, the method locates the areas represented by color gradients and replace the values of all pixels of the area by the average grayscale value of the area. It searches the areas by using a flood-fill-based algorithm. It adds a neighboring pixel into the flood-fill queue if the difference between the color values of neighboring pixels is lower or equal than a threshold t. The pixels of the color gradient are averaged (Figure 9.6). The optimal threshold t is determined heuristically by the analysis of Grayscale histogram of the bitmap (Figure 9.7). Figures 9.8 and 9.9 show examples of different thresholds.

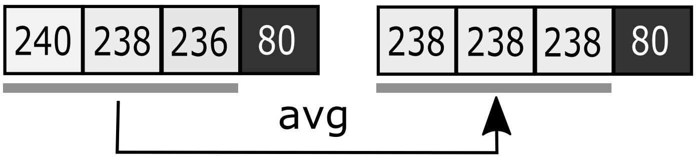


Figure 9.6: An example of a sequence of pixels. The first three pixels are grouped by the flood-filed-based algorithm using the threshold $t = 2$. The values of the pixels are averaged.

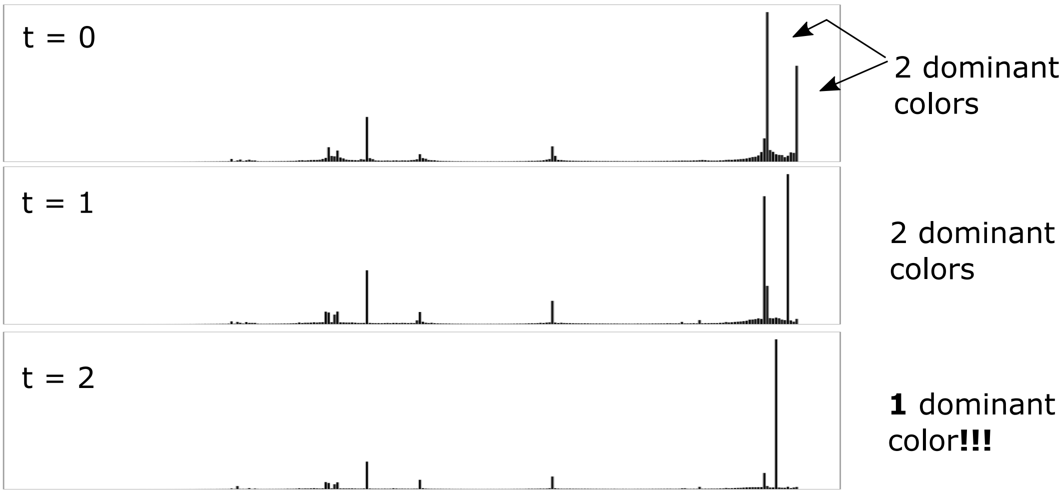


Figure 9.7: We increase the value of t iteratively and analyze changes of the histogram indicating a possible loss of relevant information. The first histogram contains two dominant colors represented by the two highest bars. The reduction of color gradients using the threshold $t = 1$ keeps the areas using the dominant colors distinguished. However, using $t = 2$ reduces the two colors into one (there is a possibility that visually different areas were joined). Hence, it is use the threshold: $t = 1$ in this case.

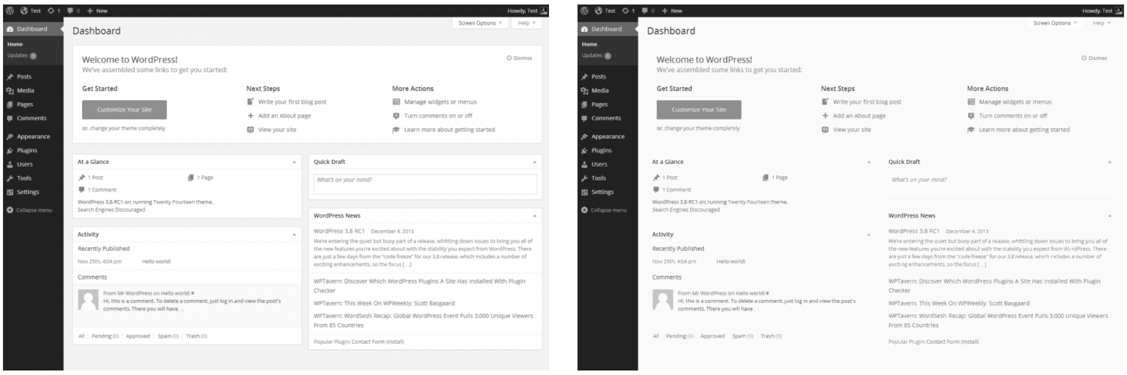


Figure 9.8: Using a high value of the threshold t might cause loss of the information about borders of UI widgets. Since the background color of the dashboard's body (light gray) and the background color of the dashboard's widgets (white) are similar, these colors are joined.

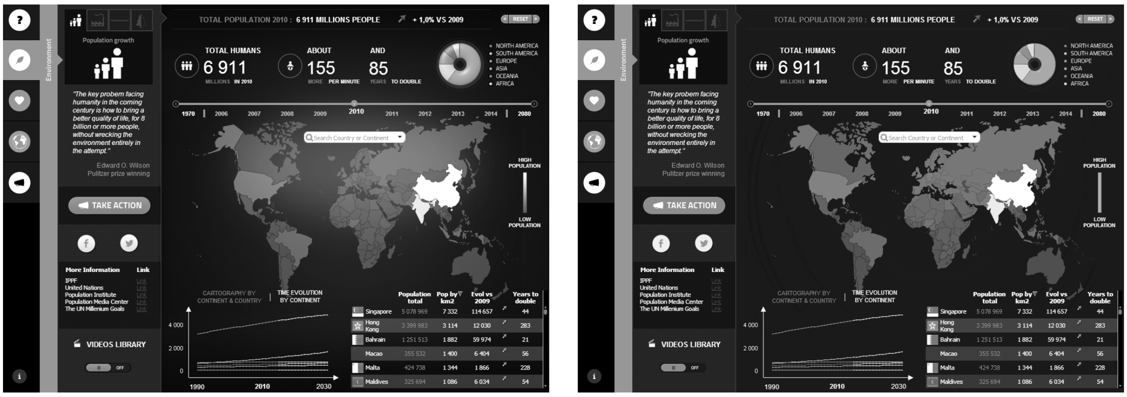


Figure 9.9: An optimal threshold t might emphasize different layers and widgets. The color gradient of the background of the dashboard's body is replaced by the average color so the body of dashboard represents one uniform area.

3. Finally, the method posterizes the image from the 8-bit to the [4 to 6]-bit color space. The optimal parameter of the posterization is searched by the analysis of color histograms similarly as the threshold t.

The result of the phase is a preprocessed bitmap coverted into the posterized grayscale color space with reduced color gradients.

9.2.2 Phase 2: Selection of Colour Layers

The second phase takes the preprocessed bitmap and selects the most frequently used colors of the bitmap. It performs the selection iteratively. It sorts all colors according to their frequency of occurrence and processes the colors from the most frequently used one. The colors are added to a list of the most frequently colors until the occurrence of the i-th color is higher than a heuristically chosen limit l_1 (0.1%, 5%, or 10% of the screen area) and the value representing the share of the most frequent colors is lower than a limit l_2 (50%, 60%, or 70% of the area, respectively to l_1).

The numbers of dominant colors can vary:

1. If the bitmap contains only one dominant color, we can easily separate background and foreground (the dominant color is replaced with the white color representing the background, the values of the remaining pixels are changed to the black color representing the foreground).

2. If the bitmap contains more than one dominant color (usually up to 10), the bitmap most likely contains more layers (e.g., widget frames). The method sorts the dominant colors according to their frequency (from the highest to the lowest) and appends a virtual color representing all remaining colors to the end of the sorted list. Then, it maps the list of the n colors to the range of n uniformly distributed grayscale values (from the white color to the black color). An example is shown in Figure 9.10.

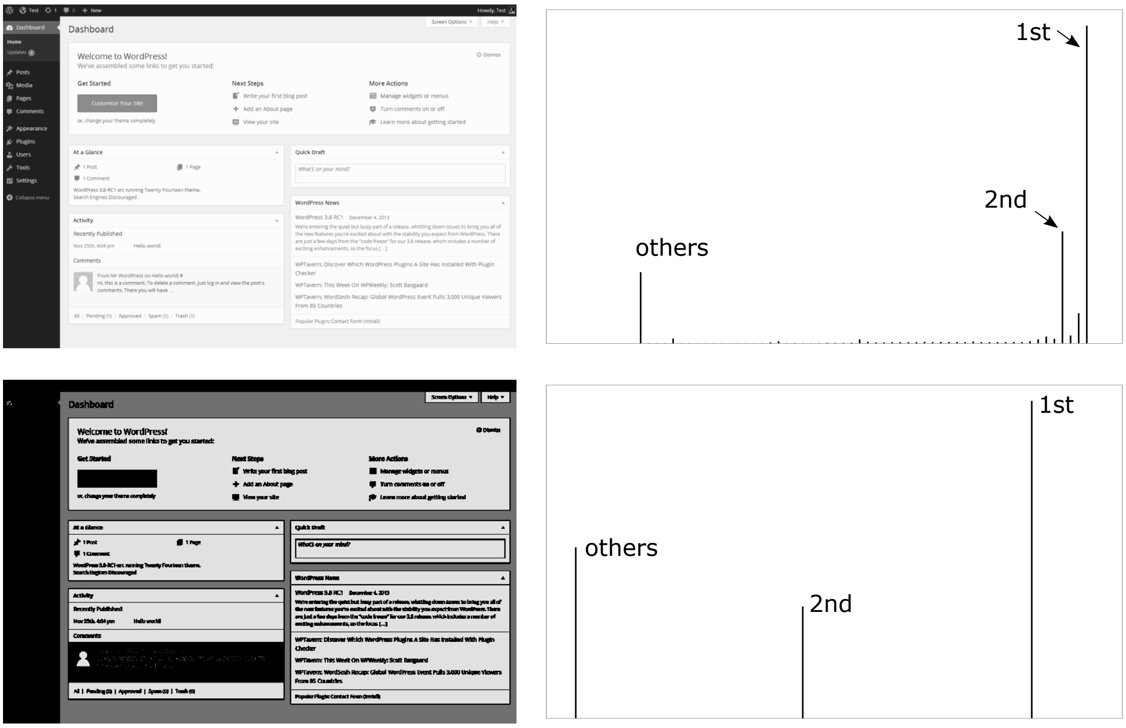
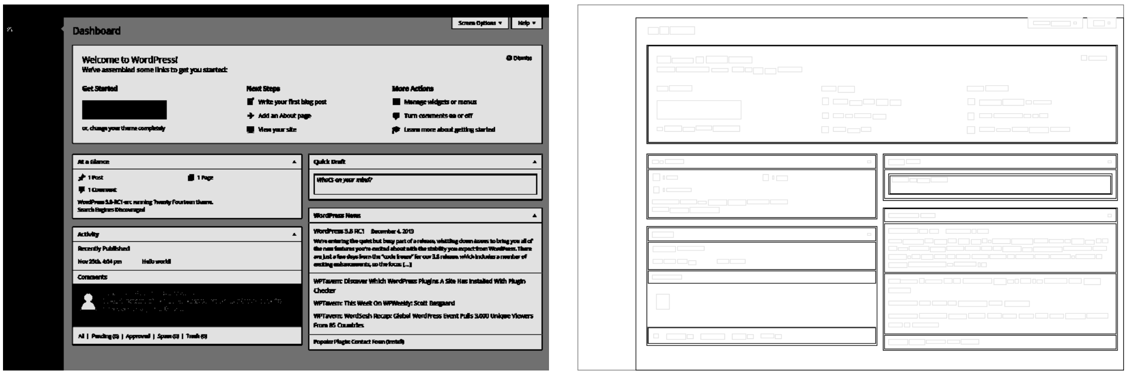


Figure 9.10: The input bitmap (upper) is represented by the 6-bit grayscale color space (64 colors). The algorithm detects 2 dominant colors, the third one is joined with other colors, which represent minority of the used colors. The final bitmap (bottom) contains only 3 colors, which represents the layers of the bitmap.

The result of the phase is a bitmap represented in *the number of dominant colors* + 1. The grayscale colors represent the layers of the bitmap which are suitable to detect page primitives and construct the page layout. Figure 9.10 presents an example of the selection of dominant colors.

9.2.3 Phase 3: Detection of Page Primitives

The third phase detects page primitives in the bitmap (Figure 9.11). Firstly, it uses a flood-fill-based algorithm to select the areas of pixels represented by the same color (layers). Then, it converts the areas into a set of regions representing rectangular boundaries of the areas. It keeps the information about the layers as attributes of the regions. Also, it measures the share of the number of pixels within boundaries of a region and keeps the value as another region's attribute. The attributes are stored for the heuristics which are applied in the selection of dominant regions (Section 9.2.5). Finally, the method filters tiny regions.


Dashboard
Screen Options ▾
Help ▾
Welcome to WordPress!
We've assembled some links to get you started:
Dismiss
Get Started
or, change your theme completely
Next Steps
Write your first blog post
Add an About page
View your site
More Actions
Manage widgets or menus
Turn comments on or off
Learn more about getting started
At a Glance
1 Post
1 Page
1 Comment
WordPress 3.8-RC1-src running Twenty Fourteen theme.
Search Engines Discouraged
Quick Draft
What's on your mind?
Activity
Recently Published
Nov 25th, 4:04 pm
Hello world!
Comments
All | Pending (0) | Approved | Spam (0) | Trash (0)
WordPress News
WordPress 3.8 RC1 December 4, 2013

We're entering the quiet but busy part of a release, whittling down issues to bring you all of the new features you're excited about with the stability you expect from WordPress. There are just a few days from the "code freeze" for our 3.8 release, which includes a number of exciting enhancements, so the focus […]

WPTavern: Discover Which WordPress Plugins A Site Has Installed With Plugin Checker
WPTavern: This Week On WPWeekly: Scott Basgaard
WPTavern: WordSesh Recap: Global WordPress Event Pulls 3,000 Unique Viewers From 85 Countries
Popular Plugin: Contact Form (Install)

Figure 9.11: Detection of page primitives. They are represented by the boundaries of the areas of neighboring pixels using the same color. The borders of detected regions are represented by a different color intensity to distinguish different types of regions (based on the size and the share of the number of pixels within its boundary).

9.2.4 Phase 4: Construction of Layout

The fourth phase converts the set of regions into a tree structure representing the page layout. In the beginning, it initializes the tree by creating a root node representing the area of the dashboard (the top-level region). Then, it goes through the set of regions and appends the regions into the tree according to the following rules:

1. If a region r_1 is located within r_2 represented by a node n_2, r_1 is compared with the children of n_2.
2. If a region r_1 surrounds a region r_2 represented by a node n_2, a node n_1 representing r_1 is created and attached to the same parent as n_2; n_2 is reattached to n_1.
3. If a region r_1 intersects a region r_2 or there is no region in actual scope, a node n_1 representing r_1 is created and attached to the parent node of actual scope.

Figure 9.12 visualizes the construction of a tree. The final tree contains hierarchically organized regions (from the top-level region representing a dashboard to the leaves representing small objects). It is important to note that one region can be represented by more than one node in the tree (an overlapping layout).

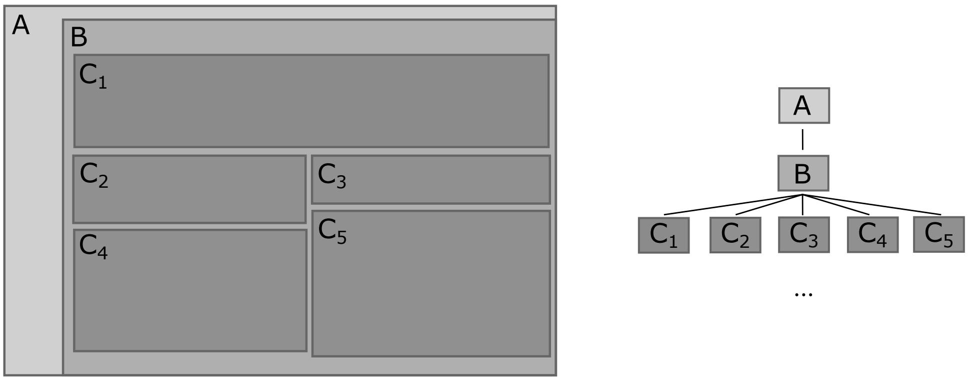


Figure 9.12: The algorithm builds a hierarchy of the regions, which represents the detected layout of a dashboard.

9.2.5 Phase 5: Top-down Layout Analysis

The next phase takes the tree of regions and searches the visually dominant regions which correspond with the user perception. The searching process starts with the top-level node. Firstly, it looks for a sidebar and header, which are frequently occurred regions in dashboards. Then, it continues with the largest region representing the body of the dashboard and analyzes its children. It sorts the children according to their size and analyzes their attributes gathered during the detection of primitives:

1. Small data regions (usually represented by the foreground layer) are filtered.
2. Very large regions which occupy the majority of the screen area are segmented (their child nodes are analyzed).
3. Remaining medium-size regions are considered as visually dominant regions.

Since the method focuses only on the large regions representing widget frames, the strategy works well with the dashboards which consist of the widgets surrounded by an explicit boundary. The body of such a dashboard contains a small number of large regions, which are detected by the top-down analysis. The users tend to recognize these regions similarly since there is a strong influence of the Gestalt law of enclosure.

On the other hand, if a dashboard contains the widgets without an explicit specification of their boundaries, the body of such a dashboard consists of a large number of small regions representing parts of widgets. Users tend to cluster these regions with correspondence to the Gestalt law of proximity. The top-down analysis ignores these regions because the regions are too small. Hence, the tree of regions is kept for Phase 7 (Section 9.2.7), which uses the reversed bottom-up strategy to cluster small regions located in the remaining areas of the dashboard. Figure 9.13a presents an example of the top-down analysis.

9.2.6 Phase 6: Analysis of Overlapping Regions

Since there are dashboards with non-rectangular layouts, it is possible that the result of the previous phase could contain overlapping regions. The sixth phase detects all intersections and compares the area of every intersection with the areas of the intersected regions.

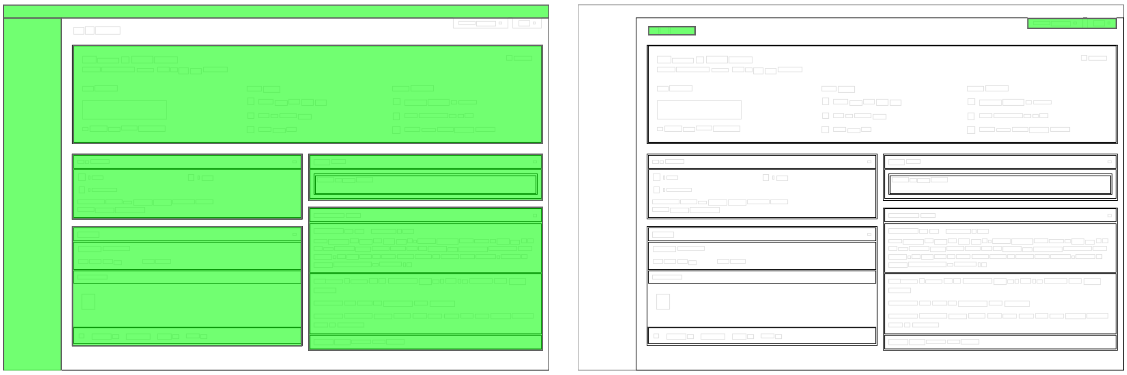
1. If the area of the intersection represents most of the area of one region (e.g., a region within another region or 2 highly overlapping regions, usually 33%), the method joins such regions into one region.

2. Else, the method ignores the intersection.

The result of the phase is a list of visually dominant regions with a reduced number of intersections.

9.2.7 Phase 7: Bottom-up Layout Analysis

The last phase focuses on the areas of a dashboard which does not contain any visually dominant region recognized in the previous phases. These areas might contain small regions which together create larger regions perceived by users with correspondence to the Gestalt law of proximity. The method takes the tree of regions representing the layout of a dashboard and analyzes it by using the bottom-up strategy. It measures the vertical and horizontal gaps between the small regions and joins the regions if the gaps are smaller than a heuristically chosen threshold. Other heuristics suitable for the bottom-up analysis were investigated in the master's thesis of Santiago Mejía supervised by the author of this thesis [Mejía, 2018].



(a) Top-down analysis **(b)** Bottom-up analysis

Figure 9.13: The top-down analysis processes the layout from the root node. It tries to detect sidebars and large regions. On the other hand, the bottom-up analysis processes the layout from the leaves of the tree. It analyzes only the areas of the bitmap which does not contain any regions detected by the top-down analysis. It clusters small primitives or tries to join them with regions.

Figure 9.13b presents an example of the bottom-up analysis. The regions detected in this phase together with the regions detected in the previous phases represent the result of the dashboard segmentation.

9.3 Evaluation

Evaluation of the method's applicability for segmentation of dashboards was based on the analysis of differences between the pixel values of:

1. the descriptions of regions specified by users (*user descriptions*)
2. the *average descriptions* representing the average perception of the users
3. the descriptions of regions created by the segmentation method (*generated descriptions*)

The evaluation used the set of 130 dashboard samples described in Section 5.4 and the descriptions of regions provided by 251 users described in Section 7.1. The differences were analyzed in the three ways:

1. **visual comparison** of the average descriptions with the generated descriptions
2. **quantitative comparison** of the user descriptions with the generated descriptions
3. **measuring Balance** using the user and generated descriptions and comparison of the results

The results and conclusions are described in the following subsections.

9.3.1 Visual Comparison with User Perception

The results of the segmentations were compared with the average descriptions visually in order to understand the main problems caused by the computer segmentation. It appears that the method successfully reduced the number of colors and detected layout primitives in most of the cases. The method works well with the dashboards which contain widgets surrounded by an explicit border (Figure 9.14). It reflects the Gestalt law of enclosure well.

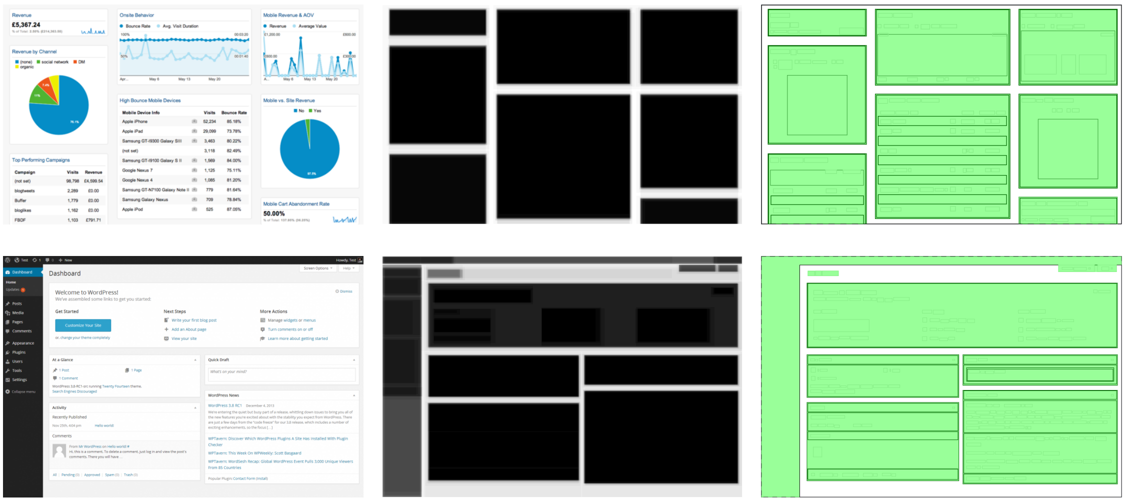


Figure 9.14: Examples of a well-segmented dashboards comparing to the perception of users. The boundaries of the widgets helped to detect the regions accurately. The sidebar and header of the second dashboard are considered as solid regions.

The following main problems were detected:

- Most of the deviations from the average perception were observed in complex dashboards which contained overlapping objects. These dashboards were usually represented by higher ambiguity of user perception of regions. Figure 9.16 presents an example of such a dashboard.

- The algorithm had occasional problems with the segmentation of management areas (e.g., toolbars and headers). It corresponds to the high disagreement of users about the description of visually dominant objects in these areas.

- Sometimes, the method incorrectly clustered small regions into a larger one, so the result insufficiently reflected the Gestalt law of proximity. Readers can notice this problem in Figure 9.15.

- The method usually inaccurately segmented the dashboard samples represented in a low resolution. This problem can be considered as a minor one since evaluators can usually provide a UI screenshot in sufficient quality.

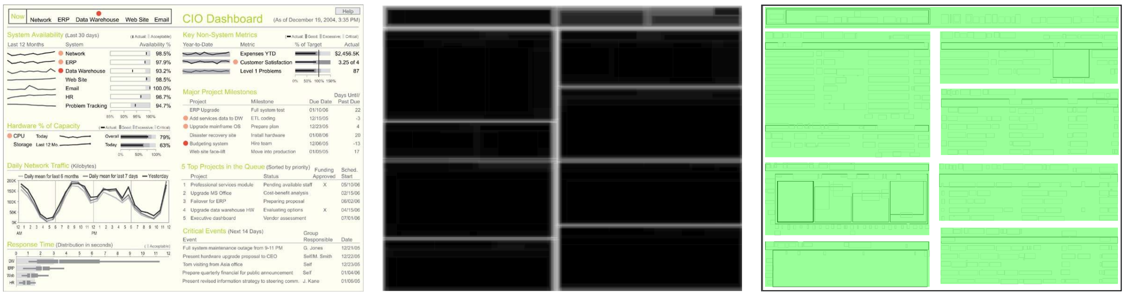


Figure 9.15: An example of a sufficiently segmented dashboard comparing to the perception of users. Missing boundaries of widgets complicated the segmentation, which relied on the bottom-up analysis of layout, which simulates the Gestalt law of proximity.

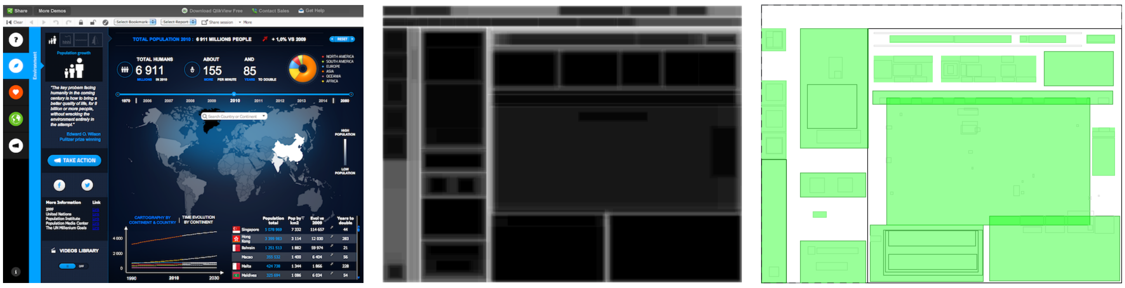


Figure 9.16: An example of the segmentation of a dashboard containing overlapping regions and color gradients, which complicate prepossessing of the image and detection of page primitives. Some of the regions were segmented insufficiently comparing to the perception of users. On the other hand, readers can notice that the average description of regions exhibits a high ambiguity of user perception as well.

9.3.2 Quantitative Comparison with User Perception

Firstly, the average descriptions were compared with the user descriptions. I calculated the difference $\delta_i^{(u)} \in [0,1]$ for every pixel i of a dashboard d:

$$\delta_i^{(u)} =\mid p_i - v_i^{(u)} \mid \tag{9.1}$$

where:

- p_i represents a probability of a region occurrence in the i-th pixel of the average description
- $v_i^{(u)} = \{0,1\}$ represents a logical value of a region occurrence in a user description provided by a user u

Then, I calculated the average difference $\delta_d^{(u)} \in [0,1]$ of all pixels in the dashboard d:

$$\delta_d^{(u)} = \frac{\sum_{i=0}^{n} \delta_i^{(u)}}{n} \tag{9.2}$$

Secondly, the average descriptions were compared with the descriptions generated by the segmentation algorithm. For every dashboard d of the 130 dashboards, a value $\delta_d^{(\mathrm{alg})}$ was calculated similarly as the values $\delta_d^{(u)}$.

Finally, every of the 130 $\delta_d^{(\mathrm{alg})}$ values was compared with the corresponding $\delta_i^{(u)}$ values in order to get the number of users for which $\delta_d^{(\mathrm{alg})} \leq \delta_d^{(u)}$. The results show that the descriptions generated by the segmentation algorithm are at least as close to the average descriptions ($\delta_d^{(\mathrm{alg})} \leq \delta_d^{(u)}$) as 33.90% of 5,020 descriptions provided by users. Figure 9.17 shows that 119 of 130 dashboards were segmented at least as close to the average description as they were segmented at least by one user. The closer to the average description the segmentation description is, the better it reflects perception of users.

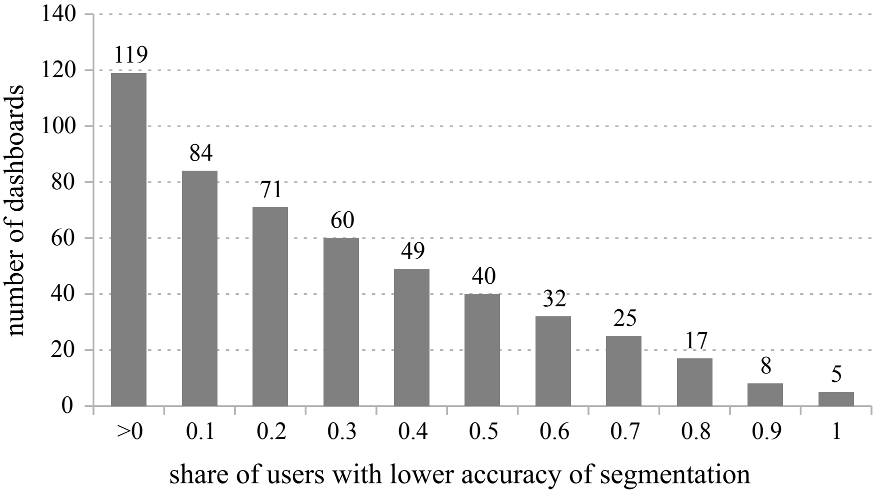


Figure 9.17: The numbers of dashboards where $\delta_d^{(\mathrm{alg})} \leq \delta_d^{(u)}$ for particular share of users. The vertical axis shows the number of dashboards. The horizontal axis represents the share of users for which $\delta_d^{(\mathrm{alg})} \leq \delta_d^{(u)}$.

9.3.3 Measuring Balance

The descriptions of regions were used to measure UI balance of the 130 dashboards according to the Balance (BM) formula designed by Ngo et al. [2000a] and its modification using the coefficients of colors described in Section 8.3. The following values were calculated for every dashboard d:

- the average value $\mathrm{BM}_d^{(\mathrm{users})}$ using the user descriptions
- $\mathrm{BM}_d^{(\mathrm{alg})}$ using the segmentation descriptions

Then, for every dashboard d, the value $\delta_{\mathrm{BM}}^{(d,\mathrm{users},\mathrm{alg})}$ representing the distance between $\mathrm{BM}_d^{(\mathrm{users})}$ and $\mathrm{BM}_d^{(\mathrm{alg})}$ was calculated.

Table 9.1: The average distance between the $\mathrm{BM}_d^{(\mathrm{users})}$ and $\mathrm{BM}_d^{(\mathrm{alg})}$ values measured by Ngo's Balance metric and its modified versions presented in Section 8.3 for the group of all dashboards $D_{(\mathrm{all})}$. The values of $\sigma_m^{(\mathrm{users},\mathrm{alg})}$ represents the standard deviations of the corresponding average values.

Metric m	$\delta_m^{(\mathrm{users},\mathrm{alg})}$	$\sigma_m^{(\mathrm{users},\mathrm{alg})}$
BM	0.100	0.086
$\mathrm{BM}(C_r^{(1)})$	0.090	0.082
$\mathrm{BM}(C_r^{(2)})$	0.089	0.090
$\mathrm{BM}(C_r^{(3)})$	0.090	0.087

Table 9.1 presents the average results for the basic Balance metric and the modified Balance metrics using the coefficients of color described in Section 8.3. The highest value was measured for the basic Balance metric: $\delta_{\mathrm{BM}}^{(\mathrm{users},\mathrm{alg})} = 0.100$. We can consider this value as low compared to the range of $\mathrm{BM} \in [0, 1]$. Then, we can notice the decrease of $\delta^{(\mathrm{users},\mathrm{alg})}$ for the modified versions of the Balance metric using the coefficients of colorfulness. Evaluators should, however, not neglect this deviation.

9.4 Limitations

The limitations of the method correspond to the limitations of the study of visual perception of regions described in Subsection 7.1.3. The method was trained with respect to the limited number of descriptions of regions provided by the limited number of users. The application of the method should be evaluated with other kinds of UI samples (not only dashboards). The group of users should consider a higher diversity of users (e.g., art-skilled users), which might provide us with a more objective point of view regarding visual perception. Finally, the results of the segmentation method should be evaluated with other metrics than Balance.

There are several possible improvements to the segmentation method which are suggested to do in the future:

- Improvement of the image preprocessing: the heuristics analyzing image histograms could be replaced with more advanced machine learning techniques using the histograms or dashboard samples as their training sets.
- Improvement of the heuristics used in the top-down and bottom-up analysis of dashboard layout in order to improve the correlation between the results of the segmentation and Gestalt laws (especially the law of proximity).
- Improvement of the processing of overlapping layouts and low-quality image samples.

9.5 Summary

This research task dealt with the problem of segmentation of user interfaces into regions, which can be used as inputs for object-based metrics of UI quality. It focused on the dashboards which usually contain complex widgets and charts which makes the dashboards difficult to segment. In contrast to printed documents, dashboards consist of a hierarchy of frames using different colors. The widgets often overlap each other. It was challenging to consider the principles of human visual perception (e.g., Gestalt laws).

The method uses the knowledge of the study of visual perception of objects by users. It consists of seven phases. In the beginning, it preprocesses a dashboard image (1), selects dominant colors to distinguish dashboard layers (2) and detects the layout primitives—regions (3). Then, it uses the regions to construct a dashboard layout (4). Finally, it processes the layout to find visually dominant regions. It processes the layout two times. The top-down strategy (5) selects the large widgets explicitly surrounded by frames (the Gestalt law of enclosure). The bottom-up strategy (7) clusters the remaining small regions into visually dominant widgets (the Gestalt law of proximity). The method also deals with overlapping regions (6).

The method was used to segment 130 dashboards. The results of the segmentation were compared with the average description of regions provided by the users. Most of the samples were segmented similarly to the average descriptions. There were samples which were more difficult to segment (e.g., Figure 9.16). However, the idea of the research task was to consider the subjectivity of user perception in the segmentation of user interfaces, which was successfully done.

The designed method was integrated into Dashboard Analyzer (described in Subsection 5.3.1). I assume that the method could be applied to other tools using object-based metrics (e.g., QUESTIM designed by Zen and Vanderdonckt [2014]). Users can use it for the initial detection of regions. Then, they can arrange possible inaccuracies in the selections of regions manually. There is also a possibility to train the parameters of segmentation according to further corrections of regions made by the users. In the future, it might be useful to improve the heuristics used for image preprocessing and analysis of dashboard layout and extend the method applicability to other kinds of user interfaces.

Chapter 10

Comparison of Metrics with User Reviews

The pixel-based and object-based metrics proposed in the previous chapters measure the specific characteristics of user interfaces which can be perceived by users. The users can rate UIs in terms of these characteristics directly. The direct rating of a UI characteristic by the users might be, however, complicated, for instance, by:

- **Subjective perception of the UI characteristic:** Since the visual perception of people is subjective (e.g., perception of colors, or past experience), perception of the UI characteristic might be ambiguous as well.
- **Subjective quantification of the perceived rate of the UI characteristic:** It might be difficult for the users to imagine the range of all possible forms of the UI characteristic. Ratings of the users represent approximate values of their subjective quantification of the perceived characteristic.
- **Subjective understanding of the UI characteristic:** Users can understand the meaning of the UI characteristic differently. Sometimes, it might be difficult to provide a verbal explanation of the meaning of a metric which measures the UI characteristic. Users might consider the influence of different aspects of UI appearance.

The goal of the last research task was to analyze user reviews of selected UI characteristics measured by the pixel-based and object-based metrics described in the previous chapters. It performed two different experiments with different groups of users. The users were let to rate the selected UI characteristics. Then, the experiments analyzed ambiguity of the user reviews and correlation of the user reviews with the values measured by the metrics.

The results of the experiments, including values gathered from the users and measured by the metrics, can be found in Appendix A. The results presented in this chapter have not been published since they should be presented in the context of the previous chapters. They provide additional information and statistics regarding the perception of users and ask additional questions for further research.

10.1 Experiment 1

The goal of the first experiment was to collect fast data representing the perception of a selected characteristic so the results could be used for the initial exploration of various aspects of user reviews. The experiment was performed with a small set of users. It focused only on one characteristic: the balance between visual weights of UI sides (vertical: top and bottom; horizontal: left and right). This characteristic was quantified by the pixel-based and object-based metrics which were analyzed in the previous chapters.

10.1.1 Procedure

The experiment was performed similarly to the study of visual perception described in Section 7.1. The users were selected among the students of the Advanced Information Systems course (spring 2017) in the master's degree program at Brno University of Technology, Faculty of Information Technology. We dedicated one lecture to familiarize the students with the *dashboard* term and the fundamental principles of data visualization and visual perception. We introduced the *balance* characteristic and possibilities of measuring this characteristic to the students. Then, we gave the students an optional homework which asked the students to rate the vertical, horizontal and overall balance of selected dashboard screenshots according to their subjective perception.

The test samples were composed of the following sets of samples:

1. $D_{(\text{all})}$: 130 dashboard bitmaps presented in Section 5.4.
2. $D_{(\text{BW})}$: 130 Black-and-white bitmaps visualizing descriptions of regions of the 130 dashboard samples of $D_{(\text{all})}$. The black-and-white descriptions of regions were generated from the average descriptions of regions gathered in the study described in Section 7.1. The conversion from the 8-bit color space to the black-and-white color space was done by using image thresholding. The value of threshold was calculated according to the formula:
$$t = 256\frac{(p_d + (1 - E_d))}{2}, \tag{10.1}$$
where $p_d = [0, 1]$ is the average pixel probability of regions occurrence in a dashboard d and $E_d = [0, 1]$ is the entropy of the dashboard d (the average entropy of all pixels of the dashboard d) calculated according to Formula 7.2. The constant 256 represents the number of colors in the 8-bit grayscale color space. Figure 10.1 presents an example of the thresholding.

The samples of $D_{(\text{all})}$ and $D_{(\text{BW})}$ were divided into three test groups. Every group contained one third (43 or 44) of the samples of $D_{(\text{all})}$ and corresponding samples of $D_{(\text{BW})}$ (86 or 88 samples in total). Every test group had two variants containing a different order of the samples. Every user rated ordered samples of one of the six test groups. The reason the users were asked to rate the black-and-white descriptions of regions was to analyze the difference between the ratings of the real dashboards and descriptions of regions.

The scale for rating balance was similar to the scale used in the study of the color impact on perception of balance in dashboards described in Section 8.2. In contrast to the study, the users also rated the overall balance, besides the vertical and horizontal balance. The users filled the ratings into a text file since Interactive Survey Tool (Subsection 5.3.3) had not been implemented yet. The text file containing ratings consisted of lines of the format shown in Listing 10.1.

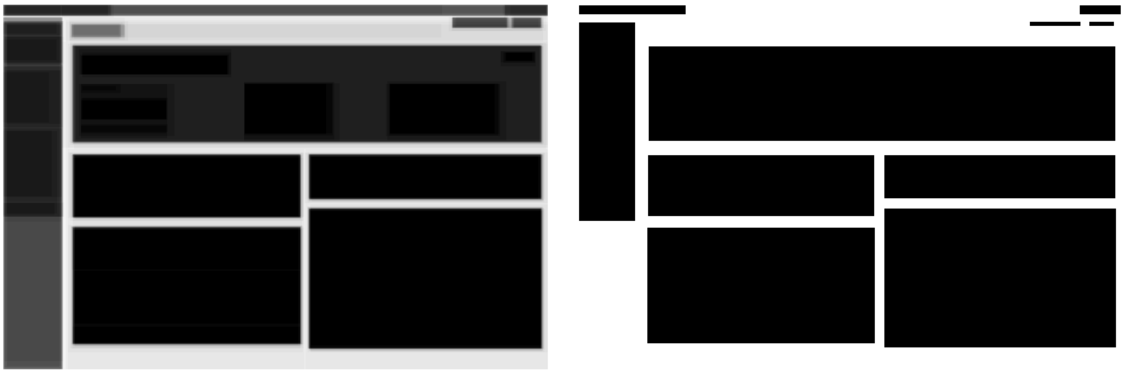


Figure 10.1: Thresholding of the average description of regions (left) according to Formula 10.1. The formula was designed experimentally.

Listing 10.1: The format of the file containing user's ratings of dashboards.

```
ID:v:h:o
```

where:

- ID is a name of dashboard (the users received a file with prefilled dashboard names).
- $v, h \in \{-2, -1, 0, 1, 2\}$ are the values of the vertical, or horizontal balance
- $o \in \{1, 2, 3, 4, 5\}$ is the value of the overall balance

The reason the users were let to rate the overall balance of samples was to compare the values of perceived overall balance val_o with the values of the overall balance val_{vh} derived from the values of perceived vertical and horizontal balance:

$$\mathrm{val}_{vh} = (2 - \frac{(\mathrm{val}_v + \mathrm{val}_h)}{2})\frac{4}{2} + 1 = 5 - (\mathrm{val}_v + \mathrm{val}_h) \tag{10.2}$$

The formula is based on Formula 3.6 measuring balance according to Ngo et al. [2000a], adjusted to the range $[-2, 2]$ of val_v and val_h and the range $[1, 5]$ of val_{vh}.

36 of 87 students decided to participate. Ratings provided by 35 users were valid or they contained correctable violations of the format shown in Listing 10.1. Ratings provided by one user were excluded since the text file violated the format.

10.1.2 Results

The results of the experiment are comprised of the following kinds of values:

- $\mathrm{val}_z^{(d,u)}$: the rating of one dashboard sample d made by a user u in a dimension $z \in \{o, v, h, vh\}$ representing the overall, vertical, or horizontal balance, or the balance calculated from vertical and horizontal balance.
- $\mathrm{val}_z^{(d)}$: the average rating of one dashboard sample d in a dimension $z \in \{o, v, h, vh\}$.
- $\sigma_z^{(d)}$: the standard deviation of the ratings of one dashboard sample d in a dimension $z \in \{o, v, h, vh\}$.

Then, the analysis of the results focused on the following aspects:

- ambiguity of user reviews represented by standard deviations
- the correlation between the ratings of $D_{(\text{all})}$ and $D_{(\text{BW})}$
- the correlation between the values of val_o and val_{vh}
- the correlation between the user reviews and the values calculated by metrics

Correlation between 2 lists of values L_1 and L_2 was quantified by Pearson correlation coefficient $r(L_1, L_2)$, which tests whether two continuous normally distributed variables indicates the linear correlation [Wherry, 2014].

Ambiguity of User Reviews

Table 10.1 presents the average standard deviations σ_o, σ_v, σ_h for the sets of samples: $D_{(\text{all})}$ and $D_{(\text{BW})}$. All ratings made by the users can be considered as ambiguous. Ambiguity of the ratings is similar for the set $D_{(\text{all})}$ of real dashboards and the set $D_{(\text{BW})}$ of descriptions of regions. An interesting finding is that the ambiguity of perception of the vertical balance is higher than the ambiguity of the horizontal balance. This might indicate that users are able to compare the weight of objects more objectively in the horizontal than the vertical direction.

Table 10.1: The average standard deviations in the ratings of the vertical σ_v, horizontal σ_h, and overall σ_o balance for the groups of real $D_{(\text{all})}$ and black-and-white $D_{(\text{BW})}$ dashboards. The values were normalized to the range $[0, 1]$.

	$D_{(\text{all})}$	$D_{(\text{BW})}$
σ_v	0.180	0.183
σ_h	0.168	0.163
σ_o	0.210	0.206

The rates of ambiguity might be a problem for further comparisons of the users' ratings of balance with the values provided by metrics. Ratings provided by different users might indicate a different level of correlation with the ratings provided by metrics.

Correlation between $D_{(\text{all})}$ and $D_{(\text{BW})}$

The second analysis of the results compared the balance ratings of the real dashboards $D_{(\text{all})}$ with the balance ratings of the corresponding Black-and-white bitmaps $D_{(\text{BW})}$ representing descriptions of regions. Table 10.2 presents the Pearson correlation coefficients r. The results indicate moderate correlations between the results of $D_{(\text{all})}$ and $D_{(\text{BW})}$. Most of the distances between the ratings were lower or equal than 1, which represents the minimal distance between the values of the rating scale. On the other hand, the differences indicate an impact of other UI features than dimensions of regions on the user perception and ratings of UI balance.

Table 10.2: The values of Pearson's correlation coefficients measuring the correlation between the real $D_{(\text{all})}$ and black-and-white $D_{(\text{BW})}$ dashboards for the values of the vertical (val_v) and horizontal (val_h) balance.

	val_o	val_v	val_h
$r(D_{(\text{all})}, D_{(\text{BW})})$	0.647	0.476	0.497

Readers should take into consideration that the results are limited by the average descriptions of regions, which are based on the perception of the limited sample of users. Also, the results are limited by the chosen thresholding approach.

Correlation between val_o and val_{vh}

The third analysis of the results compared the ratings of the overall balance val_o with the values of the balance val_{vh} calculated from the values of the vertical and horizontal balance according to Formula 10.2. Table 10.3 presents the Pearson correlation coefficients for the sets of samples: $D_{(\text{all})}$ and $D_{(\text{BW})}$.

Table 10.3: The values of Pearson's correlation coefficients measuring the correlation between the ratings of the overall balance val_o and balance calculated from the ratings of the vertical and horizontal val_{vh} for the groups of real $D_{(\text{all})}$ and black-and-white $D_{(\text{BW})}$ dashboards.

	$D_{(\text{all})}$	$D_{(\text{BW})}$
$r(\text{val}_o, \text{val}_{vh})$	0.567	0.585

The correlation coefficients are very similar for both $D_{(\text{all})}$ and $D_{(\text{BW})}$ sets of samples. They indicate a moderate correlation between the values in both cases. Most of the distances between the ratings were lower or equal than 1, which represents the minimal distance between the values of the rating scale. The results support the applicability of Formula 10.2 for calculation of the overall balance from the values of the vertical and horizontal balance.

Correlation between User Reviews and Metrics

The last analysis analyzed the average ratings $\text{val}_v^{(d)}$, $\text{val}_h^{(d)}$, $\text{val}_{vh}^{(d)}$ and $\text{val}_o^{(d)}$ for every dashboard of $d \in D_{(\text{all})}$. It compared them with the values calculated by the following metrics:

- BM representing Ngo's object-based Balance described in Formula 3.6
- BM_{C_1}, BM_{C_2}, BM_{C_3} representing the modified versions of Ngo's Balance described in Section 8.3
- $\text{BM}_{\text{8-bit}}$, $\text{BM}_{\text{4-bit}}$, BM_{BW} representing the pixel-based metrics measuring balance in the 8-bit grayscale, posterized 4-bit grayscale, and 1-bit black-and-white color space; described in Subsection 3.3.1

Table 10.4: The values of Pearson's correlation coefficients measuring the correlation between the ratings of the perceived vertical, horizontal and overall balance and the values of the balance measured by the selected metrics for the group of real dashboards $D_{(\text{all})}$.

	val_v	val_h	val_{vh}	val_o
$r(\text{val}, \text{BM})$	-0.013	-0.294	0.392	0.341
$r(\text{val}, \text{BM}_{C_1})$	0.035	-0.202	0.316	0.195
$r(\text{val}, \text{BM}_{C_2})$	-0.044	-0.104	0.195	0.133
$r(\text{val}, \text{BM}_{C_3})$	0.012	-0.176	0.253	0.172
$r(\text{val}, \text{BM}_{\text{8-bit}})$	-0.028	-0.184	0.354	0.197
$r(\text{val}, \text{BM}_{\text{4-bit}})$	0.020	-0.144	0.314	0.146
$r(\text{val}, \text{BM}_{\text{BW}})$	-0.025	-0.162	0.428	0.337

Table 10.4 presents the Pearson correlation coefficients. They are low for the ratings of the horizontal balance and close to zero for the values of the vertical balance. Then, the coefficients are higher for the values of the overall balance val_{vh} and val_o, but they still do not indicate any sign of solid correlation. These higher rates of correlation coefficients might be caused by the absolute values of the vertical and horizontal balance in the formula of the overall balance. The formula of the overall balance analyzes whether the UI is balanced or unbalanced in general, but ignores the selection of the sides with a higher optical weight. On the contrary, the rates of the vertical and horizontal balance represent the direct comparison of the weights of the UI sides.

10.1.3 Conclusions

The primary goal of the experiment was to analyze the correlation between the perceived values of UI balance with the corresponding values of UI balance measured by metrics. The results of the experiment have not shown any sign of correlation. Such results were not expected since they contradict to the idea of the metrics described by Ngo et al. [2000a]; Kim and Foley [1993]; Vanderdonckt and Gillo [1994]. On the other hand, ratings of users exhibited a considerable ambiguity. The experiment was performed with a small set of users. The users filled the rating into the text files, which might have distracted them and caused errors. All these aspects decrease the credibility of the results.

The results have provided some further information, though. They have shown the signs of correlation between ratings of $D_{(\text{all})}$ (representing real dashboards) and $D_{(\text{BW})}$ (representing descriptions of regions). This means that dimensions of regions play a role in perception of visual weights of the UI sides. On the other hand, we can see that there are other factors which might have an impact on the perceived optical weights.

The results have also shown a moderate correlation between the values of val_o (representing the overall users' ratings of balance) and val_{vh} (representing the ratings of balance derived from the users' ratings of the vertical and horizontal balance). This means that the value of the overall balance can be considered as the combination of the corresponding values of the vertical and horizontal balance. Since the correlation is not perfect, it might be useful to explore the importance of the vertical and horizontal balance, which might not be the same. This hypothesis is supported by the fact that the ambiguity of the user reviews of the vertical and horizontal balance is not the same.

The results describing the correlation between $D_{(\text{all})}$ and $D_{(\text{BW})}$, and between val_o and val_{vh} might be considered as more credible than the average values of balance since the results are based on the comparison of values provided by single users. The results of the experiment have, however, shown considerable ambiguity of the average user perception of balance. For this reason, it was decided to repeat the experiment.

10.2 Experiment 2

The second experiment was performed as a response to the results of the first experiment. The goal was to collect more accurate, objective and less ambiguous data than the first experiment in order to provide more accurate results regarding the correlation between the characteristics perceived by users and characteristics measured by metrics. The experiment was performed with a larger set of users ($n = 220$). The users used Interactive Survey Tool (described in Subsection 5.3.3) to rate the characteristics, which decreased the effort of the users to perform the experiment and prevented mistakes in reviews (e.g., a wrong format of the results as it happened in Experiment 1).

The experiment analyzed the balance between visual weights of UI sides similarly as the first experiment (vertical: top and bottom; horizontal: left and right). Besides that, it analyzed the colorfulness of UI. It focused only on ambiguity of the ratings and correlation of the ratings with the values measured by metrics.

10.2.1 Procedure

The users were selected among the students of the Information Systems course (autumn 2017) in the bachelor's degree program at Brno University of Technology, Faculty of Information Technology. The students were instructed similarly to the students of the first experiment. Then, the students were asked to do an optional homework—rate colorfulness, vertical and horizontal balance of selected bitmap samples according to their subjective perception.[1] 220 of 386 students decided to participate.

The test set were composed of the same samples as the test set of Experiment 1: $D_{(\text{all})}$ and $D_{(\text{BW})}$ containing 130 real dashboard samples and corresponding 130 description regions. Besides that, the $D_{(\text{gray})}$ set of samples containing 130 corresponding average descriptions of regions was included.

The samples were divided into 3 test groups. Every test group contained one third of the samples (43 or 44) from every set of samples $D_{(\text{all})}$, $D_{(\text{gray})}$, and $D_{(\text{BW})}$. Every user rated one test group:

- the vertical and horizontal balance for the samples of $D_{(\text{all})}$, $D_{(\text{gray})}$, and $D_{(\text{BW})}$ (129 or 132 samples in total); the overall balance was not rated in contrast to Experiment 1
- colorfulness for the samples of $D_{(\text{all})}$ (43 or 44 samples in total)

[1] Besides that, the users rated *density* of objects in UI and the *first impression* about UI. The thesis does not focus on these characteristics to keep the simplicity of description.

10.2.2 Results

The results of this experiment are composed of the same kinds of values as Experiment 1: $\text{val}_z^{(d,u)}$, $\text{val}_z^{(d)}$, and $\sigma_z^{(d)}$. There are, however following changes:

- The dimension $z \in \{v, h, vh, clr\}$ represents the vertical or horizontal balance, the balance calculated from the vertical and horizontal balance or colorfulness of bitmap. The overall balance was not rated.
- The dashboard sample d can be considered as a member of the groups: $D_{(\text{all})}$, $D_{(\text{gray})}$, or $D_{(\text{BW})}$.

The analysis of the results described in this section focuses on the following aspects:

- ambiguity of user reviews represented by standard deviations
- the correlation between the user reviews and the values calculated by metrics

Ambiguity of User Reviews

Table 10.5 presents the average standard deviations σ_v, σ_h, and σ_{clr} for the sets of samples $D_{(\text{all})}$, $D_{(\text{gray})}$, and $D_{(\text{BW})}$. In contrast to the expectations, the results are similar to the results of Experiment 1. They indicate considerable ambiguity of all ratings. Ambiguity of the ratings is very similar for all the sets of dashboards: $D_{(\text{all})}$, $D_{(\text{gray})}$, and $D_{(\text{BW})}$. The results also support the finding that the ambiguity of perception of the vertical balance is higher than the ambiguity of the horizontal balance.

The average standard deviation of colorfulness is even higher than the average standard deviation of the vertical balance. This result might correspond to different perception of color by people which was described in Chapter 4. It is also assumed that the users had difficulties with the estimation of colorfulness rate on the rating scale.

Table 10.5: The average standard deviations in the ratings of the vertical σ_v, horizontal σ_h, and overall σ_o balance for the groups of $D_{(\text{all})}$, $D_{(\text{gray})}$, and $D_{(\text{BW})}$. The values were normalized to the range $[0, 1]$.

	$D_{(\text{all})}$	$D_{(\text{gray})}$	$D_{(\text{BW})}$
σ_v	0.185	0.191	0.192
σ_h	0.184	0.170	0.176
σ_{clr}	0.205		

Correlation between User Reviews and Metrics

The second analysis calculated the average ratings $\text{val}_v^{(d)}$, $\text{val}_h^{(d)}$, $\text{val}_{vh}^{(d)}$ and $\text{val}_{clr}^{(d)}$ for every dashboard of $d \in D_{(\text{all})}$. The average ratings of $\text{val}_v^{(d)}$, $\text{val}_h^{(d)}$ and $\text{val}_{vh}^{(d)}$ were compared with the corresponding values calculated by the the same metrics as for Experiment 1 (BM, BM_{C_1}, BM_{C_2}, BM_{C_3}, $\text{BM}_{\text{8-bit}}$, $\text{BM}_{\text{4-bit}}$, BM_{BW}). The average ratings of colorfulness $\text{val}_{clr}^{(d)}$ were compared with the values measured by:

1. the C_{CIELAB} metric measuring colorfulness based on the image saturation of the CIE L*a*b* color space
2. the C_{HSB} metric measuring colorfulness based on image saturation of the HSB color space

Table 10.6 presents the values of Pearson correlation coefficients. The results do not show any significant increase in the correlation coefficients, but a similar tendency as the results of Experiment 1. The correlation coefficients are low for the ratings of the horizontal balance and even lower for the values of the vertical balance.

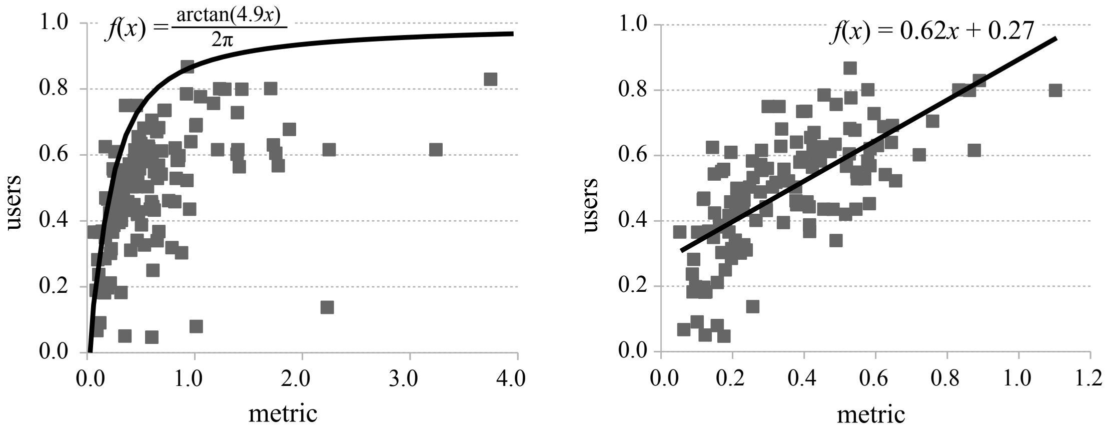
Table 10.6: The values of Pearson's correlation coefficients measuring correlation between the ratings of balance (or colorfulness) and the values of balance (or colorfulness) measured by the metrics for the group of real dashboards $D_{(\text{all})}$.

	val_v	val_h	val_{vh}	val_{clr}
$r(\text{val}, \text{BM})$	0.062	-0.255	0.272	
$r(\text{val}, \text{BM}_{C_1})$	0.119	-0.201	0.219	
$r(\text{val}, \text{BM}_{C_2})$	0.001	-0.085	0.162	
$r(\text{val}, \text{BM}_{C_3})$	0.084	-0.177	0.188	
$r(\text{val}, \text{BM}_{\text{8-bit}})$	0.118	-0.154	0.261	
$r(\text{val}, \text{BM}_{\text{4-bit}})$	0.121	-0.134	0.217	
$r(\text{val}, \text{BM}_{\text{BW}})$	0.095	-0.132	0.275	
$r(\text{val}, f(C_{\text{CIELAB}}))$				0.573
$r(\text{val}, C_{\text{HSB}})$				0.689

On the contrary, the correlation between the perceived and measured colorfulness is high, despite the high ambiguity of the users' ratings. Figure 10.2 shows the charts of the relations between the values. We can notice a linear correlation between the perceived colorfulness and the C_{HSB} colorfulness. The correlation between the perceived colorfulness and the C_{CIELAB} colorfulness is simulated by the function: $f = \frac{\arctan(kC_{\text{CIELAB}})}{2\pi}$ where $k = 4.9$ was found iteratively (step $= 0.1$) as the approximate value for the highest value of r.

10.2.3 Conclusions

The main goal of the second experiment was to improve the credibility of the results. It attempted to decrease the ambiguity of users' ratings and find a correlation between the ratings and values measured by metrics. Despite the higher set of users and availability of the tool for simple rating of UIs, the results of the experiment were similar to the results of the previous one. The users' ratings exhibited a similar level of ambiguity. The correlation between the average user ratings of UI balance and the measured values of UI balance has not been found. A possible reason for these results might be the insufficient explanation of the balance characteristic to the users who might have rated the weights of the UI sides according to different visual UI aspects than the Balance metrics rates them. Discussion with selected people indicated that some users might have understood the balance characteristic more like a the symmetry characteristic. The experiment should be repeated with a higher emphasis on the analysis of users' decisions. The UI aspects which have an impact



(a) C_{CIELAB}

(b) C_{HSB}

Figure 10.2: The relations between the average ratings of colorfulness and the values of colorfulness measured by the metrics.

on the balance ratings should be explored and possible improvements of existing metrics should be considered.

On the other hand, it has been shown the correlation between the average values of the perceived rates of colorfulness and the measured rates of colorfulness, even though the users disagreed about the rates of colorfulness. The principle of the colorfulness characteristic seems to be more straightforward than the principle of the balance characteristic and it corresponds with the metrics better.

10.3 Limitations

The results are limited by the selected group of users and their subjective perception of the analyzed characteristics. Besides that, we need to consider their subjective interpretation of the analyzed UI characteristics and limited ability to imagine the scale of possible values, which is crucial for the estimation of a characteristic's rating. As the results have shown, these limitations might have affected the users' ratings since they indicate a considerable level of ambiguity.

The results of the experiments open questions which suggest possible further research tasks:

- How do users understand balanced and unbalanced UIs and weights of the UI sides? Try to understand the decisions of users leading to their ratings.

- Are there any other UI visual aspects (e.g., shapes of objects) which might play a role in the perception of UI balance? Find and analyze the UI aspects.

- How do the rates of the vertical and horizontal balance affect the rates of the overall balance? Compare the influence of the vertical and horizontal balance on the overall balance.

The answers to the questions might serve as the user experience for the improvement of the metrics of UI balance, as it has been described by the framework for the design and improvement of metrics in Section 8.3.

10.4 Summary

This research task focused on user reviews of UI characteristics and their relation with the values measured by metrics. It analyzed and discussed the problem of the subjective perception of UI characteristics by users. The two independent experiments have shown that users rate UI balance and colorfulness of the selected 130 dashboards subjectively. This might be caused by the subjective perception of the characteristics, the subjective quantification of the perceived rate of the characteristics, or the subjective understanding of the UI characteristics.

The results have indicated the following signs of correlation:

- between the perceived colorfulness and the colorfulness measured by the two pixel-based metrics
- between the perceived overall balance and the overall balance calculated from the values of the perceived vertical and horizontal balance
- between the values of the UI balance perceived in the real dashboard samples and the UI balance perceived in the black-and-white bitmaps representing descriptions regions of the real dashboard samples (based on the average perception of regions by users)

On the other hand, the research task was unable to show the correlation between the perceived rates of balance (vertical and horizontal) and the values measured by 7 pixel-based and object-based metrics, including their combinations, trying to quantify UI balance. The users rated the visual weights of the UI sides differently. The reasons supporting their decisions should be analyzed in the future. The gained user experience could be used either for the further improvement of the metrics or the improvement of further experiments and explanation of the characteristics to the users before the experiments.

The correlation coefficients of the UI balance contradict the basic idea of the metrics proposed by Ngo et al. [2000a]; Kim and Foley [1993]; Vanderdonckt and Gillo [1994]. On the other hand, these results might not necessarily be a problem from the point of view of the automatic evaluation of UI quality. It is more important that the metrics correlate with the overall quality of UI rated by UI designers. This fact has been shown in Chapter 8.3.

Chapter 11

Discussion

The research explored the possibility of automatic metric-based evaluation of dashboard quality. The research was split into the 6 research tasks producing the following results:

1. **Design of the model for the internal representation of dashboards:** The research task established the model which corresponds with the two dashboard perspectives: the pixel-based and object-based perspective. Then, the software dealing with the dashboard's internal representation was implemented:

   - Dashboard Analyzer: a tool for manipulation with dashboard bitmaps, descriptions of dashboard internal representation, and for integration of metrics.
   - Generator of Dashboard Samples: a tool for generation of synthetic dashboard samples (developed as a part of the master's thesis of [Pastushenko, 2017]).
   - Interactive Survey Tool: a tool for surveying users about perceived characteristics of dashboards.

   Finally, 130 dashboard bitmaps were found on the Internet and divided into the groups of random dashboards and dashboards which respect the heuristics of Few [2006]. They represented the set of test samples for the further research tasks.

2. **Analysis of pixel-based metrics:** The research task provides the knowledge about the applicability of chosen pixel-based metrics for recognition of well-designed dashboards. The following metrics were considered as applicable:

   - colorfulness of UI based on the color channels of the CIE L*a*b* and HSB color space
   - the number of colors with the share higher than 10% in the posterized 4-bit grayscal color space
   - the share of the 1st most or 1st + 2nd most used color values measured in the posterized 4-bit grayscale or 12-bit RGB color space
   - the pixel-based Balance and Symmetry metrics measured in the posterized 4-bit grayscale color space

   The metrics were integrated into Dashboard Analyzer.

3. **Analysis of object-based metrics:** The research task provides:
   - The dataset of the subjective perception of regions representing visually dominant objects.
   - The framework for rating object-based metrics: It provides the instructions on how to process subjective user descriptions of regions and use the data to measure the metric's ability to distinguish a chosen kind of user interface from the others objectively. For this purpose, the two quantitative characteristics were established: metric objectivity and decisiveness.
   - The knowledge describing the applicability of chosen object-based metrics for objective recognition of well-designed dashboards: The research was focused on the 13 metrics of aesthetics designed by Ngo et al. [2000a]. It classified the metrics. Only Cohesion, Proportion, and Rhythm were classified in Class 1 (high rate of objectivity). None of the metrics were classified in Class 2 (high rate of objectivity and decisiveness).
4. **Improvement of object-based metrics:** The research task provides:
   - the framework describing the process of improvement of metrics and design guidelines
   - the knowledge describing the impact of color on the perception of UI balance based on a small-scale study
   - the idea of improvement of object-based metrics which combines object-based metrics with a pixel-based approach measuring the colorfulness of interface regions
   - 3 modified versions of Ngo's Balance metrics, which exhibit a high rate of objectivity and decisiveness (Class 2)
5. **Segmentation of dashboard screens:** The research tasks provides the method for segmentation of dashboard screens into the regions which can be used as inputs for object-based metrics. The method uses the knowledge of the study of visual perception of objects. The method was integrated into Dashboard Analyzer.
6. **Comparison of metrics with user reviews:** The research task provides:
   - the datasets of subjective perception of balance and colorfulness provided by 220 users.
   - the knowledge describing ambiguity of user perception and correlation of the user perception with the values of UI balance and colorfulness measured by Dashboard Analyzer. The results showed the correlation between the measured and perceived values colorfulness.

The following sections discuss the potential applications of the results, points out limitations of the results and suggests possible future tasks which might extend the results.

11.1 Application of Results

The results can be applied in the following areas:

- **Using Dashboard Analyzer to analyze other metrics measuring characteristics of dashboards and UIs in general:** Dashboard analyzer provides the API for implementation of own metrics, image processing methods, and segmentation algorithms. The software can be used for:
  - further research tasks exploring the possibility to use metrics for analysis and evaluation of quality and usability of user interfaces
  - design, implementation and debugging of novel methods for segmentation of user interfaces
  - gathering and processing internal representations of user interfaces representing the users' subjective perception of UIs
- **Using the framework for design and improvement of metrics to design novel metrics and improve existing ones:** Generator of Dashboard Samples can be used to generate specific UI samples which could be used to evaluate hypotheses about the impact of UI visual aspects on perception of UI characteristics similarly as it was done in the study described in Section 8.2. Interactive Survey Tool can be used to gather user reviews describing perception of UI characteristics by users.
- **Using the dataset of subjective perception of regions for further analyses and understanding of visual perception and Gestalt laws:** The descriptions of regions gathered in the study of visual perception of regions provides valuable information about the initial perception and interpretation of the structure of complex user interfaces by users. This dataset can be used for further analyses of visual perception. It can help understand the ambiguity of visual perception and object clustering corresponding with preattentive and attentive perception. It can be used for observation of Gestalt laws and help with the formalization of Gestalt laws, which is still the aim of the researchers [Jäkel et al., 2016]. Last but not least, the description of regions can be used for measuring of objectivity and decisiveness (the ability to distinguish two kinds of user interfaces objectively) of other metrics as described in Chapter 7.
- **Using the method for segmentation of dashboards to improve existing tools for analysis of user interfaces using object-based metrics:** Section 3.3.3 introduced the QUESTIM tool designed by Zen and Vanderdonckt [2014], which allows users to segment a UI screenshot manually by drawing rectangles around the objects of UI regions. Then, the regions are used as inputs for object-based metrics. Using the method for segmentation of dashboards might decrease the effort of users to draw the rectangles. For instance, the method might propose an initial arrangement of the UI rectangles and the users might adjust them according to their perception.
- **Last, but not least, using the knowledge about the metrics to create tools for design and evaluation of dashboards and user interfaces in general:** Dashboard Analyzer is a prototype of the tool which analyzes screenshots of user interfaces similarly as the QUESTIM tool. The knowledge presented in this thesis could be used to design other tools using metric-based evaluation of user interfaces—e.g.:

  - an extension for a web browser which would analyze dashboard web pages using the metric-based evaluation
  - an extension for existing dashboard builders which would provide information about a currently designed dashboard and warn the designers about potential problems regarding visual aspects of the dashboard (e.g., a high level of colorfulness, or a low level of balance)
  - a generator of dashboard layout templates which would meet the requirements of the well-designed dashboards

11.2 Limitations

This section summarizes the main limitations of the research:

- **The research ignores the interaction of users with dashboards:** The research focused only on the selected static visual aspects of dashboards. A quick interaction of a user with a dashboard plays an important role in the usability of the dashboard. It is crucial for the usability of the operational dashboards. The user needs to be able to react to actual problems displayed in a dashboard quickly. Hence, the metric-based evaluation of visual aspects of dashboards cannot replace user testing. However, this is not the idea of the metric-based evaluation. The idea is to provide additional information regarding visual characteristics dashboards. Such information can be available in the early design phases before the designers have a functional prototype of a dashboard. It can regulate the creativity of the designers and make the designers focus on the simplicity and clarity of a user interface. It can save time and expenses in later phases of dashboard development.

- **The results of the research are based on the perception of the limited number of users:** The research focused on the analysis of the subjective perception of users. It tried to maximize the number of participants providing descriptions of their subjective perception. The study of visual perception of regions was participated by 251 users. The user reviews of dashboard characteristics worked with two groups of 38 and 220 users. Only the study of the color impact on object-based metrics was participated by two small groups of users (12 and 13). The numbers of participants were high relatively to the other studies which were described in Section 3.3.3. On the other hand, the participants were similar (technical students, around 20-25 years old). The results should be verified with different types of users.

- **The results of the research are based on the limited number of dashboard samples:** The research worked with 130 dashboard samples. Analysis of a larger set of samples would require more time and resources. The priority of the samples selection was finding miscellaneous dashboard samples in order to maximize the credibility of the results. The further analyses using a different set of dashboard samples should be, however, done. Also, the group of well-designed dashboard samples should contain other samples than the dashboards created according to the design guidelines described by Few [2006].

- **The research was not focused on the construction of design guidelines:** It analyzed the ability of the chosen metrics to distinguish the group of well-designed dashboards from the randomly chosen samples objectively. It did not specify certain limits for the rates of UI characteristics in a well-designed dashboard. It is expected that these limits might vary in different situations.

11.3 Suggestions for Future Work

The results of the research opened some questions which could be beneficial to answer in the future:

1. Is the ambiguity of the user perception of visually dominant regions generally different in specific parts of a user interface? The study of visual perception of regions (Section 7.1) suggested that some logical parts of dashboards were usually more ambiguous than the rest of the dashboard—e.g., menus or toolbars.

2. Does a high value of the ambiguity of user perception have a negative influence on the usability and quality of the user interface? It could be useful to compare the usability of user interfaces varying in the share of the area containing ambiguous parts.

3. Is it possible to improve the objectivity and decisiveness of Ngo's remaining metrics without a radical change of their characteristics? Is it possible to use all coefficients of color C_{r_1}, C_{r_2} and C_{r_3} described in Section 8.3? Dashboard Analyzer contains a suggestion for improvement of the other Ngo's metrics which are based on the area and distribution of regions on a screen. Applicability of the modified metrics should be analyzed similarly as it was done for the Balance metric and its modifications.

4. What is the optimal level of objectivity and decisiveness? The limits were established for the purposes of this research, but these levels might be different for various types of user interfaces and users. Therefore, further experiments should be done.

5. Are there any other visual characteristics (e.g., image complexity, shapes of widgets) which can be used for the improvement of metrics similarly to color?

6. How do users understand balanced and unbalanced UIs and the weight of the UI sides? Try to understand the decisions of users leading to their ratings.

7. How do the rates of the vertical and horizontal balance affect the rate of the overall UI balance? Compare the influence of the vertical and horizontal balance on the overall balance.

There are various ways how to follow the results of this research. Further research tasks can be focused on:

- **Development and improvement of metrics:** This research focused on the chosen pixel-based and object-based metrics measuring colorfulness and layout aspects of user interfaces. It would be useful to design metrics which would provide a more in-depth analysis of the structure and content of user interfaces. Inspiration could be found in the thesis of Ivory [2001], which describes metrics analyzing the structure and content of webpages. It would, however, require:
    - **Extension of the model of dashboard's internal representation:** This research worked with the flat description of the top-level UI components representing regions of visually dominant objects. The description could be extended into a hierarchical description, which allows describing nested objects (representing simple parts of the top-level UI components—e.g., text or graphical elements). The description of components' styles should be extended as well. The displayed dataset should be also available.
    - **Improvement of the conversion of dashboard into the internal representation:** The method for conversion of dashboard into the internal representation should be able to capture the information regarding the extended model of internal representation. It would require an improvement of the method for segmentation of dashboards. The design of such a method might be complicated. Hence, the method might collaborate with the methods parsing document structure (e.g., parsing DOM of a web page).
- **Improving the knowledge about user perception:** Firstly, this research focused on the perception of visually dominant objects by users and analyzed ambiguity of their perception. The knowledge was used for the construction of the method for segmentation dashboards and preparation of inputs for object-based metrics. It focused on the Gestalt laws of enclosure and proximity. Secondly, the research analyzed user perception of selected characteristics of UI. It compared the user reviews with the values measured by metrics. Future research task might focus on:
    - **Closer understanding of the user perception of objects within a UI:**
        - Users often disagreed about the granularity of screens. Some users marked larger objects as solid regions, others split them into smaller ones. It might be beneficial if users specify a hierarchy of objects instead of the flat definition. Such definition might improve the knowledge about perception of UI layout, which might be used for improvement of the segmentation method.
        - Researchers might analyze the impact of the ambiguity of object perception on the quality and usability of a user interface. They can compare the ambiguity of various parts of dashboards and UIs in general. It might help to answer the questions: 1 and 2.
        - Also, the researchers might focus on the formalization of Gestalt laws, which is still the aim of researchers [Jäkel et al., 2016]. They might consider other Gestalt laws in the segmentation of a dashboard screen, not only the laws of enclosure and proximity.

  - **Closer understanding of the user perception of the characteristics relating to UI quality:** Users rated UI characteristics ambiguously. It might have been caused either by the subjective perception of the characteristics, the subjective quantification of the perceived rate of the characteristics, or the subjective understanding of the UI characteristics. Researchers might perform further studies gathering user reviews of UI characteristics. They should focus more on the reasons why the users decide to choose particular ratings. They should analyze how do their decisions correspond to the overall quality and usability of a UI. They should understand which visual aspects play a role in UI quality and usability of a UI. Such knowledge might help to improve the correlation of metrics with user perception and help to answer the questions: 5, 6, and 7. The knowledge might help to improve existing metrics and design guidelines.

- **Improving the credibility of the results:** As it has been described in Section 11.2, the results of this research are based on the subjective perception of the limited number of users. It might be useful to repeat the research tasks with different groups of users and dashboard samples.

  - **Increase the number and variability of users:** The research worked with the sets of similar users: students of IT around 20 years old with the technical specialization. It might be beneficial to explore the perception of different groups of users (e.g., children, seniors, or people having skills in arts).
  - **Increase the number and variability of UI samples:** The research focused only on dashboards. It might be beneficial to analyze the applicability of the metric-based evaluation to other kinds of user interfaces and compare the results with the results measured for dashboards.

- **Development of tools for metric-based evaluation of dashboards:** There are various online commercial tools which allow users to create their own dashboard. They usually do not contain any assistance which would review the design quality of the dashboard or recommends improvements of the interface. The knowledge of the metric-based evaluation of dashboards can be integrated into existing tools for dashboard design or used for implementation of a new one as it was suggested in Section 11.1.

Chapter 12

Conclusions

Today's people and organizations are overloaded by various data. Suitable presentation of data is crucial for quick understanding of the information conveyed by the data, and finding the beneficial knowledge which can be used for making the right decisions. Dashboards are favorite tools for displaying data. They try to fit all data representing the most important information on a single screen so the user can find the important links between the information better. They use the benefits of a graphical representation of data to maximize the amount of data displayed on one screen. Well-designed dashboards respecting design advice might improve the data processing by users significantly. On the other hand, an inappropriate arrangement and design mistakes might mislead the users.

Nowadays, it is not difficult to create a dashboard containing various kinds of charts by people who do not have the knowledge of programming languages and databases. There are many online dashboard interactive builders which allow users to design their own dashboard quickly using a palette of predefined widgets, and connect the dashboard with other information systems providing data (e.g., social networks). Using such tools is quick, effortless and cheaper than hiring specialists who would create a tailored solution. It, however, lacks expertise in usability and overall quality of the solution.

Guideline reviews offer a possibility of automatic evaluation of user interfaces. They are usually based on simple quantitative metrics measuring basic design aspects of user interfaces (e.g., an appropriate arrangement of widgets, or selection of colors). It can not replace user testing, which provides detailed feedback of users. However, it can provide basic analytics of user interface characteristics, warn about possible violations of design guidelines which might cause usability problems, and assist the users during the design of the dashboard.

The problem of automatic measuring of UI characteristics is that it requires a unified UI format. The first possibility is to take a screenshot of the user interface and analyze the bitmap composed of pixels. The examples of suitable pixel-based metrics for evaluation of dashboards are colorfulness, the number of dominant (the most used) colors, or the pixel-based Balance and Symmetry metrics. It has been shown that the dashboards designed according to design advice of Few [2006] exhibit lower colorfulness than randomly chosen dashboards. They have a uniform background which represents the most used color in the dashboard. Their colors are distributed in a more balanced and symmetrical manner.

The disadvantage of the pixel-based evaluation is that it does not consider a user interface in the way as people perceive it. It works with a matrix of pixels while people recognize objects of the UI (simple shapes) and cluster them into logical groups which have some meaning for them (e.g., widgets, controls, or charts). Object-based metrics measure characteristics of the UI connected with the UI objects. They require a description of the UI objects. This thesis proposed the language-independent model of internal representation of dashboards describing their objects, behavior, style and arrangement on the screen, and the data they display. Then, this research worked with the simplified model of internal representation represented in the XML language describing the size and dimension of UI objects. The model was sufficient for the evaluation of UI layout and aesthetics by Ngo's metrics [Ngo et al., 2000a].

The main problem of the object-based evaluation was, however, the vague definition of UI objects. Before the analysis of object-based metrics was done, the study of visual perception of objects in dashboards had been performed. 251 users had provided their subjective descriptions of regions representing the perception of object boundaries in UIs. Then, the descriptions of regions were used to analyze the ambiguity of user perception and its impact on the application of object-based metrics. For this purpose, the framework was established. It has defined the instructions on how to process the descriptions of regions and use them to calculate the values of the metric objectivity and decisiveness—the characteristics of metrics which quantify the ability of a metric to distinguish two groups of UIs objectively (e.g., the group well-designed and randomly chosen dashboards).

The analysis of Ngo's object-based metrics has shown that some of Ngo's metrics are not able to consider ambiguity in the visual perception of regions. This applies particularly to the metrics whose formula depends on the number of regions: Unity, Simplicity, Regularity, Economy, and Homogeneity. On the other hand, Cohesion and Proportion (the metrics which depend on the aspect ratios of regions) and Rhythm (one of the metrics based on the accuracy of regions' areas and the distribution of regions on a screen) have shown higher rates of objectivity.

As the response to the results of the analysis of Ngo's object-based metrics, the framework specifying the process of design and improvement of metrics was established. It was used to improve the objectivity and decisiveness of object-based metrics. The improvement combines object-based metrics with the pixel-based approach measuring colorfulness of the interface regions. Application of the improvement to the Balance metric has shown an increase in the rate of the metric objectivity and decisiveness.

Then, the method for automatic segmentation of dashboards into regions was designed and implemented. The method uses the knowledge of the study of visual perception of objects. Particularly, it considers the Gestalt laws of enclosure and proximity, which play a high role in the perception of regions, as the results of the study suggested. The method was used to segment the dashboard samples and the results were compared with the average descriptions of regions provided by the users. Most of the samples were segmented similarly to the average descriptions.

Finally, the metrics were compared with the reviews of dashboard characteristics by the two groups of 38 and 220 users. The reviews indicated high ambiguity of the values. The results showed signs of correlation between: (1) the perceived and measured colorfulness (2) the perceived overall balance and the overall balance based on the perceived vertical and horizontal balance; (3) the UI balance perceived in the real dashboards and the black-and-white bitmaps representing the descriptions of regions of the real dashboard samples (based on the average perception). On the other hand, the research task was unable to show the correlation between the perceived and measured UI balance (vertical and horizontal). The users rated the visual weights of the UI sides differently. The reasons supporting their decisions should be analyzed in the future.

All of the analyzed metrics including the method for segmentation of dashboards were implemented and integrated into Dashboard Analyzer—the Java application which can load a screenshot of a user interface (from a file or URL) and analyze the screenshot using the metrics. The application provides the APIs for implementation and debugging of own metrics and methods for segmentation of UIs. The source code of the application is available online (Appendix B.1). The application can be used in future research.

Future research might use the knowledge presented in this thesis to design new tools using the metric-based evaluation. For instance, it might be useful to implement an extension for a web browser which would analyze dashboard webpages. Also, the metrics could be integrated into existing dashboard builders. Besides that, a future research might focus on the improvement of existing metrics and searching for new ones. Dashboard Analyzer might be used for this purpose. The dataset of subjective perception of regions might be used for the improvement of the method for segmentation of dashboards as well as for further analyses and understanding of visual perception and Gestalt laws.

The goal of the research has been accomplished. The research has analyzed the common characteristics of dashboards and explored existing metrics for analysis of UI attributes and considered their application for measuring the quality and usability of dashboards. It has focused on the object-based metrics of aesthetics and analyzed the ambiguity of measured results caused by users' subjective perception of objects. It has created the framework for evaluation of the metrics' ability to distinguish well-designed dashboard samples objectively. It has found the new approach which improves the metrics' ability to distinguish well-designed dashboard samples objectively. It has designed the method for segmentation of dashboards into regions which correspond to the average perception of the users. It has implemented the tool which provide the functionality for loading, segmentation and objective measurement and analysis of chosen dashboard characteristics. Last but not least, the thesis has focused on the visual perception of objects in dashboards. It has evaluated the subjective visual perception of the users and detected presence of Gestalt laws. There exist many publications which focus on automatic evaluation using object-based metrics, but few of them tries to consider the subjective perception of objects by users. I see it as the major contribution of the research.

Bibliography

Alexander, M. and Walkenbach, J. (2010). *Excel Dashboards and Reports.* Wiley.

Altaboli, A. and Lin, Y. (2011). Investigating effects of screen layout elements on interface and screen design aesthetics. *Advances in Human-Computer Interaction*, 2011:1–10.

Alvarez, R. M., Hall, T. E., and Trechsel, A. H. (2009). Internet voting in comparative perspective: The case of Estonia. *PS: Political Science & Politics*, 42(3):497–505.

Andrews, J. H. (1998). Testing using log file analysis: tools, methods, and issues. In *Automated Software Engineering, 1998. Proceedings. 13th IEEE International Conference on*, pages 157–166. IEEE.

Baker, J., Jones, D., and Burkman, J. (2009). Using visual representations of data to enhance sensemaking in data exploration tasks. *Journal of the Association for Information Systems*, 10(7):533–559.

Bargas-Avila, J. A. and Hornbæk, K. (2011). Old wine in new bottles or novel challenges: a critical analysis of empirical studies of user experience. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2689–2698. ACM.

Barič, F. (2017). Attribute analysis of charts used for data presentation using information dashboards. Bachelor's thesis (Supervisor: Hruška T.), Brno University of Technology.

Bauerly, M. and Liu, Y. (2008). Effects of symmetry and number of compositional elements on interface and design aesthetics. *Intl. Journal of Human–Computer Interaction*, 24(3):275–287.

Beeley, C. (2018). *Hands-On Dashboard Development with Shiny: A practical guide to building effective web applications and dashboards.* Packt Publishing.

Bevan, N. (2009). What is the difference between the purpose of usability and user experience evaluation methods. In *Proceedings of the UXEM Workshop – INTERACT'09*, pages 1–4.

Bodart, F., Hennebert, A.-M., Leheureux, J.-M., and Vanderdonckt, J. (1994). Towards a dynamic strategy for computer-aided visual placement. In *Proceedings of the workshop on Advanced visual interfaces*, pages 78–87. ACM.

Bradley, D. and Roth, G. (2007). Adaptive thresholding using the integral image. *Journal of graphics tools*, 12(2):13–21.

Buley, L. (2013). *The User Experience Team of One: A Research and Design Survival Guide.* Rosenfeld Media.

Burget, R. (2017). Information extraction from the web by matching visual presentation patterns. In *Knowledge Graphs and Language Technology, LNCS*, volume 10579, pages 10–26.

Capurro, R. and Hjørland, B. (2005). The concept of information. *Annual review of information science and technology*, 37(1):343–411.

Card, S. K., Moran, T. P., and Newell, A. (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, 23(7):396–410.

Card, S. K., Moran, T. P., and Newell, A. (2018). *The psychology of human-computer interaction*. CRC Press.

Carpendale, S. (2008). Evaluating information visualizations. In *Information visualization, LNCS*, volume 4950, pages 19–45. Springer.

Charfi, S., Trabelsi, A., Ezzedine, H., and Kolski, C. (2014). Widgets dedicated to user interface evaluation. *International Journal of Human-Computer Interaction*, 30(5):408–421.

Codd, E. F., Codd, S. B., and Salley, C. T. (1993). *Providing OLAP (On-line Analytical Processing) to User-analysts: An IT Mandate*. Codd & Associates.

del-Rey-Chamorro, F. M., Roy, R., van Wegen, B., and Steele, A. (2003). A framework to create key performance indicators for knowledge management solutions. *Journal of Knowledge Management*, 7(2):46–62.

Doermann, D., Tombre, K., et al. (2014). *Handbook of Document Image Processing and Recognition*. Springer.

Duchowski, A. (2007). *Eye Tracking Methodology: Theory and Practice*. Springer.

Eckerson, W. W. (2006). *Performance Dashboards: Measuring, Monitoring, and Managing Your Business*. Wiley.

Etemad, K., Doermann, D., and Chellappa, R. (1997). Multiscale segmentation of unstructured document pages using soft decision integration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):92–96.

Feng, H., Zhang, W., Wu, H., and Wang, C.-J. (2016). Web page segmentation and its application for web information crawling. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 598–605. IEEE.

Fernandez, A., Insfran, E., and Abrahão, S. (2011). Usability evaluation methods for the web: A systematic mapping study. *Information and Software Technology*, 53(8):789–817.

Few, S. (2006). *Information Dashboard Design: The Effective Visual Communication of Data*. O'Reilly.

Foley, J. D. and Van Dam, A. (1982). *Fundamentals of Interactive Computer Graphics*. Addison-Wesley.

Forlizzi, J. and Battarbee, K. (2004). Understanding experience in interactive systems. In *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 261–268. ACM.

Freeman, J. B. and Ambady, N. (2010). Mousetracker: Software for studying real-time mental processing using a computer mouse-tracking method. *Behavior Research Methods*, 42(1):226–241.

Gibson, J. J. (1950). *The perception of the visual world.* Houghton Mifflin.

Goodwin, K. and Cooper, A. (2011). *Designing for the Digital Age: How to Create Human-Centered Products and Services.* Wiley.

Grudin, J. (1992). Utility and usability: research issues and development contexts. *Interacting with Computers*, 4(2):209–217.

Ha, J., Haralick, R. M., and Phillips, I. T. (1995). Recursive X-Y cut using bounding boxes of connected components. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, pages 952–955. IEEE.

Harrington, S. J., Naveda, J. F., Jones, R. P., Roetling, P., and Thakkar, N. (2004). Aesthetic measures for automated document layout. In *Proceedings of the 2004 ACM symposium on Document engineering (DocEng '04)*, pages 109–111. ACM Press.

Harris, R. L. (2000). *Information Graphics: A Comprehensive Illustrated Reference.* Oxford University Press.

Hassenzahl, M. and Tractinsky, N. (2006). User experience-a research agenda. *Behaviour & information technology*, 25(2):91–97.

Healey, C. G., Booth, K. S., and Enns, J. T. (1996). High-speed visual estimation using preattentive processing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 3(2):107–135.

Henderson, J. M. and Hollingworth, A. (1999). High-level scene perception. *Annual Review of Psychology*, 50(1):243–271.

Hollingsed, T. and Novick, D. G. (2007). Usability inspection methods after 15 years of research and practice. In *Proceedings of the 25th annual ACM international conference on Design of communication (SIGDOC '07)*, pages 249–255. ACM.

Hynek, J. and Hruška, T. (2015). Automatic evaluation of information dashboard usability. In *Proceedings of Second International Conference on Advances in Information Processing and Communication Technology - IPCT 2015*, pages 129–133. Institute of Research Engineers and Doctors.

Hynek, J. and Hruška, T. (2016). Pixel-based analysis of information dashboard attributes. In *ADBIS 2016: New Trends in Databases and Information Systems, CCIS*, pages 29–36. Springer.

Hynek, J. and Hruška, T. (2018). Application of object-based metrics for recognition of well-designed dashboards. *International Journal of Human–Computer Interaction*, pages 1–13, in press, doi: 10.1080/10447318.2018.1518004.

Hynek, J. and Hruška, T. (2019). Segmentation of dashboard screen images: Preparation of inputs for object-based metrics of ui quality. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 3: IVAPP*, pages 199–207. INSTICC, SciTePress.

International Organization for Standardization (ISO) (2001). *ISO/IEC 9126-1:2001 Software engineering – Product quality – Part 1: Quality model.* ISO.

International Organization for Standardization (ISO) (2002). *ISO/TR 16982:2002 Ergonomics of human-system interaction – Usability methods supporting human-centred design.* ISO.

International Organization for Standardization (ISO) (2010). *ISO 9241-210:2010 Ergonomics of human system interaction-Part 210: Human-centred design for interactive systems.* ISO.

International Organization for Standardization (ISO) (2011). *ISO/IEC 25010:2011 Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuaRE)-System and Software Quality Models.* ISO.

Ivory, M. Y. (2001). An empirical foundation for automated web interface evaluation. Dissertation thesis (Supervisor: Hearst, M.), University of California, Berkeley.

Ivory, M. Y. and Hearst, M. A. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys (CSUR)*, 33(4):470–516.

Jacobs, J. and Rudis, B. (2014). *Data-driven security: analysis, visualization and dashboards.* Wiley.

Jain, A. K. and Zhong, Y. (1996). Page segmentation using texture analysis. *Pattern recognition*, 29(5):743–770.

Jäkel, F., Singh, M., Wichmann, F. A., and Herzog, M. H. (2016). An overview of quantitative approaches in gestalt perception. *Vision Research*, 126:3–8.

Jeffries, R., Miller, J. R., Wharton, C., and Uyeda, K. (1991). User interface evaluation in the real world: a comparison of four techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '91)*, pages 119–124. ACM.

Jelenčíková, A. (2018). Analysis and processing of information dashboards. Bachelor's thesis (Supervisor: Hruška T.), Brno University of Technology.

Johnson, J. (2010). *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules.* Elsevier.

Karvonen, K. (2000). The beauty of simplicity. In *Proceedings on the 2000 conference on Universal Usability (CUU '00)*, pages 85–90. ACM.

Katsanos, C., Karousos, N., Tselios, N., Xenos, M., and Avouris, N. (2013). KLM Form Analyzer: Automated evaluation of web form filling tasks using human performance models. In *Human-Computer Interaction – INTERACT 2013, LNCS*, volume 8118, pages 530–537. Springer.

Kendall, K. E. and Kendall, J. E. (2011). *Systems analysis and design.* Prentice Hall.

Kieras, D. (2001). Using the keystroke-level model to estimate execution times. Technical report, University of Michigan.

Kim, H.-C. (2015). Acceptability engineering: the study of user acceptance of innovative technologies. *Journal of Applied Research and Technology*, 13(2):230–237.

Kim, W. C. and Foley, J. D. (1993). Providing high-level control and expert assistance in the user interface presentation design. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, pages 430–437. ACM.

Kise, K. (2014). Page segmentation techniques in document analysis. In *Handbook of Document Image Processing and Recognition*, chapter 5, pages 135–175. Springer.

Koffka, K. (1922). Perception: an introduction to the Gestalt-theorie. *Psychological Bulletin*, 19(10):531–585.

Koffka, K. (2013). *Principles of Gestalt Psychology*. The International Library of Psychology. Taylor & Francis.

Kristeller, P. O. (1951). The modern system of the arts: A study in the history of aesthetics part I. *Journal of the History of Ideas*, 12(4):496–527.

Kurosu, M. and Kashimura, K. (1995). Apparent usability vs. inherent usability: experimental analysis on the determinants of the apparent usability. In *Conference Companion on Human Factors in Computing Systems (CHI '95)*, pages 292–293. ACM.

Köhler, W. (1925). An aspect of Gestalt psychology. *The Pedagogical Seminary and Journal of Genetic Psychology*, 32(4):691–723.

Lallemand, C., Gronier, G., and Koenig, V. (2015). User experience: A concept without consensus? Exploring practitioners' perspectives through an international survey. *Computers in Human Behavior*, 43:35–48.

Lavie, T. and Tractinsky, N. (2004). Assessing dimensions of perceived visual aesthetics of web sites. *International Journal of Human-Computer Studies*, 60(3):269–298.

Law, E. L.-C., Roto, V., Hassenzahl, M., Vermeeren, A. P., and Kort, J. (2009). Understanding, scoping and defining user experience: a survey approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*, pages 719–728. ACM.

Law, E. L.-C., van Schaik, P., and Roto, V. (2014). Attitudes towards user experience (UX) measurement. *International Journal of Human-Computer Studies*, 72(6):526–541.

Lee, B., Chen, Y., and Hewitt, L. (2011). Age differences in constraints encountered by seniors in their use of computers and the internet. *Computers in Human Behavior*, 27(3):1231–1237.

Leung, L. (2015). Validity, reliability, and generalizability in qualitative research. *Journal of family medicine and primary care*, 4(3):324–327.

Levy, D. M. (2008). Information overload. In *The Handbook of Information and Computer Ethics*, chapter 20, pages 497–515. Wiley.

Lewis, C. and Rieman, J. (1993). *Task-centered User Interface Design: A Practical Introduction*. University of Colorado, Boulder, Department of Computer Science.

Lindgaard, G., Fernandes, G., Dudek, C., and Brown, J. (2006). Attention web designers: You have 50 milliseconds to make a good first impression! *Behaviour & Information Technology*, 25(2):115–126.

Loginova, N. (2017). Attribute analysis of data presentation using information dashboards. Bachelor's thesis (Supervisor: Hruška T.), Brno University of Technology.

Mahajan, R. and Shneiderman, B. (1997). Visual and textual consistency checking tools for graphical user interfaces. *IEEE Transactions on Software Engineering*, 23(11):722–735.

Malik, S. (2005). *Enterprise dashboards: design and best practices for IT*. Wiley.

Mao, S. and Kanungo, T. (2001). Empirical performance evaluation methodology and its application to page segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):242–256.

Marcotte, E. (2011). *Responsive Web Design*. A Book Apart.

Marey, E. (1885). *La méthode graphique dans les sciences expérimentales et principalement en physiologie et en médecine*. G. Masson.

Marr, D. (2010). *Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information*. MIT Press.

Mazumdar, S., Petrelli, D., Elbedweihy, K., Lanfranchi, V., and Ciravegna, F. (2015). Affective graphs: The visual appeal of linked data. *Semantic Web*, 6(3):277–312.

McCarthy, J. and Wright, P. (2007). *Technology as Experience*. MIT Press.

Mejía, S. (2018). Analysis of dashboard attributes based on automatic decomposition of screen. Master's thesis (Supervisor: Hynek J.), Brno University of Technology.

Mendes, E. and Mosley, N. (2006). *Web engineering*. Springer.

Merriam-Webster Inc. (2004). *Merriam-Webster's Collegiate Dictionary*. Merriam-Webster.

Michailidou, E., Harper, S., and Bechhofer, S. (2008). Visual complexity and aesthetic perception of web pages. In *Proceedings of the 26th annual ACM international conference on Design of communication (SIGDOC '08)*, pages 215–224. ACM.

Minaee, S. and Wang, Y. (2016). Screen content image segmentation using sparse decomposition and total variation minimization. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3882–3886. IEEE.

Miniukovich, A. and Angeli, A. D. (2014). Quantification of interface visual complexity. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI '14)*, pages 153–160. ACM.

Morville, P. (2005). Experience design unplugged. In *ACM SIGGRAPH 2005 Web program*, page 10. ACM.

Moshagen, M. and Thielsch, M. T. (2010). Facets of visual aesthetics. *International Journal of Human-Computer Studies*, 68(10):689–709.

Nagy, G. and Seth, S. (1984). Hierarchical representation of optically scanned documents. In *Proceedings of the 7th International Conference on Pattern Recognition*, pages 347–349. IEEE Computer Society Press.

Ngo, D., Teo, L., and Byrne, J. (2000a). Formalising guidelines for the design of screen layouts. *Displays*, 21(1):3–15.

Ngo, D. C. L. (2001). Measuring the aesthetic elements of screen designs. *Displays*, 22(3):73–78.

Ngo, D. C. L. and Byrne, J. G. (2001). Application of an aesthetic evaluation model to data entry screens. *Computers in Human Behavior*, 17(2):149–185.

Ngo, D. C. L., Samsudin, A., and Abdullah, R. (2000b). Aesthetic measures for assessing graphic screens. *Journal of Information Science and Engineering*, 16(1):97–116.

Ngo, D. C. L., Teo, L. S., and Byrne, J. G. (2003). Modelling interface aesthetics. *Information Sciences*, 152:25–46.

Nielsen, J. (1994a). Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing (CHI '94)*, pages 152–158. ACM.

Nielsen, J. (1994b). *Usability Engineering*. Academic Press.

Nielsen, J. (1994c). Usability inspection methods. In *Conference Companion on Human Factors in Computing Systems (CHI '94)*, pages 413–414. ACM.

Nielsen, J. and Molich, R. (1990). Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing (CHI '90)*, pages 249–256. ACM.

Orlov, P. A., Ermolova, T., Laptev, V., Mitrofanov, A., and Ivanov, V. (2016). The eye-tracking study of the line charts in dashboards design. In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 205–213. SCITEPRESS.

Parush, A., Nadir, R., and Shtub, A. (1998). Evaluating the layout of graphical user interface screens: Validation of a numerical computerized model. *International Journal of Human-Computer Interaction*, 10(4):343–360.

Pastushenko, O. (2017). Web application for information dashboard design. Master's thesis (Supervisor: Hynek J.), Brno University of Technology.

Pastushenko, O., Hynek, J., and Hruška, T. (2018). Generation of test samples for construction of dashboard design guidelines: Impact of color on layout balance. In *World Conference on Information Systems and Technologies*, pages 980–990. Springer.

Pastushenko, O., Hynek, J., and Hruška, T. (2019). Evaluation of user interface design metrics using generator of realistic-looking dashboard samples. *Expert Systems*, pages 1–19, in press, doi: 10.1111/exsy.12434.

Preece, J., Rogers, Y., and Sharp, H. (2015). *Interaction Design: Beyond Human-Computer Interaction.* Wiley.

Purchase, H. C. (2014). Twelve years of diagrams research. *Journal of Visual Languages & Computing*, 25(2):57–75.

Purchase, H. C., Freeman, E., and Hamer, J. (2012). An exploration of visual complexity. In *Diagrams 2012: Diagrammatic Representation and Inference, LNCS*, volume 7352, pages 200–213.

Purchase, H. C., Hamer, J., Jamieson, A., and Ryan, O. (2011). Investigating objective measures of web page aesthetics and usability. In *Proceedings of the Twelfth Australasian User Interface Conference - Volume 117 (AUIC '11)*, pages 19–28. Australian Computer Society.

Rasmussen, N. H., Bansal, M., and Chen, C. Y. (2009). *Business Dashboards: A Visual Catalog for Design and Deployment.* Wiley.

Read, J. C. and Markopoulos, P. (2013). Child–computer interaction. *International Journal of Child-Computer Interaction*, 1(1):2–6.

Reinecke, K., Yeh, T., Miratrix, L., Mardiko, R., Zhao, Y., Liu, J., and Gajos, K. Z. (2013). Predicting users' first impressions of website aesthetics with a quantification of perceived visual complexity and colorfulness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*, pages 2049–2058. ACM.

Roto, V. and Rautava, M. (2008). User experience elements and brand promise. In *International Engagability & Design Conference, in conjunction with NordiCHI'08 conference.*

Russ, J. C. (2016). *The Image Processing Handbook.* CRC Press.

Salimun, C., Purchase, H. C., Simmons, D. R., and Brewster, S. (2010). Preference ranking of screen layout principles. In *Proceedings of the 24th BCS Interaction Specialist Group Conference (BCS'10)*, pages 81–87. British Computer Society.

Sears, A. (1993). Layout appropriateness: A metric for evaluating user interface widget layout. *IEEE Transactions on Software Engineering*, 19(7):707–719.

Sears, A. (1995). AIDE: A step toward metric-based interface development tools. In *Proceedings of the 8th annual ACM symposium on User interface and software technology (UIST '95)*, pages 101–110. ACM.

Sezgin, M. and Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–166.

Shafait, F., Keysers, D., and Breuel, T. M. (2006). Performance comparison of six algorithms for page segmentation. In *Document Analysis Systems VII, LNCS*, volume 3872, pages 368–379.

Shneiderman, B. (1987). *Designing the user interface: strategies for effective human-computer interaction.* Addison-Wesley.

Shneiderman, B., Chimera, R., Jog, N., Stimart, R., and White, D. (1995). Evaluating spatial and textual style of displays. Technical report, University of Maryland.

Simon, A., Pret, J.-C., and Johnson, A. P. (1997). A fast algorithm for bottom-up document layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):273–277.

Siroker, D. and Koomen, P. (2013). *A/B Testing: The Most Powerful Way to Turn Clicks Into Customers*. Wiley.

Smith, S. L. and Mosier, J. N. (1986). Guidelines for designing user interface software. Technical report, Defense Technical Information Center.

Stevenson, A. (2010). *Oxford Dictionary of English*. Oxford Dictionary of English. Oxford University Press.

Stirrup, J. (2016). *Tableau Dashboard Cookbook*. Packt Publishing.

Tractinsky, N. (1997). Aesthetics and apparent usability: empirically assessing cultural and methodological issues. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '97)*. ACM Press.

Tractinsky, N. (2004). Toward the study of aesthetics in information technology. In *Proceedings of the 25th International Conference on Information Systems (ICIS 2004)*, pages 771–780. AIS Electronic Library.

Tractinsky, N., Katz, A. S., and Ikar, D. (2000). What is beautiful is usable. *Interacting with Computers*, 13(2):127–145.

Trausan-Matu, S. and Dathan, B. (2016). Perceived aesthetics of user-modifiable layouts: a comparison between an unspecified design and a gui. In *Proceedings of the 13th International Conference on Human-Computer Interaction (RoCHI'2016)*, pages 22–25. Matrix Rom.

Tufte, E. (2001). *The Visual Display of Quantitative Information*. Graphics Press.

Tullis, T. and Albert, W. (2010). *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Elsevier.

Tullis, T. S. (1984). Predicting the usability of alphanumeric displays. Dissertation thesis, Rice University.

van Tonder, G. J. and Ejima, Y. (2000). Bottom–up clues in target finding: Why a dalmatian may be mistaken for an elephant. *Perception*, 29(2):149–157.

Vanderdonckt, J., Beirekdar, A., and Noirhomme-Fraiture, M. (2004). Automated evaluation of web usability and accessibility by guideline review. In *Web Engineering, LNCS*, volume 3140, pages 17–30. Springer Berlin Heidelberg.

Vanderdonckt, J. and Gillo, X. (1994). Visual techniques for traditional and multimedia layouts. In *Proceedings of the workshop on Advanced visual interfaces (AVI '94)*, pages 95–104. ACM.

Ware, C. (2004). *Information Visualization: Perception for Design*. Elsevier.

Wertheimer, M. (1923). A brief introduction to gestalt, identifying key theories and principles. *Psychologische Forschung*, 4(1):301–350.

Wexler, S., Shaffer, J., and Cotgreave, A. (2017). *The Big Book of Dashboards: Visualizing Your Data Using Real-world Business Scenarios.* Wiley.

Wharton, C., Rieman, J., Lewis, C., and Polson, P. (1994). The cognitive walkthrough method: A practitioner's guide. In *Usability Inspection Methods*, chapter 5, pages 105–140. Wiley.

Wherry, R. (2014). *Contributions to Correlational Analysis.* Elsevier.

Wieczorek, M., Vos, D., and Bons, H. (2014). *Systems and Software Quality: The next step for industrialisation.* Springer.

Wong, K. Y., Casey, R. G., and Wahl, F. M. (1982). Document analysis system. *IBM Journal of Research and Development*, 26(6):647–656.

Wrembel, R. and Koncilia, C. (2006). *Data Warehouses and OLAP: Concepts, Architectures, and Solutions.* IRM Press.

Yendrikhovskij, S., Blommaert, F. J., and de Ridder, H. (1998). Optimizing color reproduction of natural images. In *The 6th Color and Imaging Conference: Color Science, Systems, and Applications*, pages 140–145. Society for Imaging Science and Technology.

Yigitbasioglu, O. M. and Velcu, O. (2012). A review of dashboards in performance management: Implications for design and research. *International Journal of Accounting Information Systems*, 13(1):41–59.

Yin, P.-Y. (2001). Skew detection and block classification of printed documents. *Image and Vision Computing*, 19(8):567–579.

Zain, J. M., Tey, M., and Goh, Y. (2008). Probing a self-developed aesthetics measurement application (SDA) in measuring aesthetics of mandarin learning web page interfaces. *International Journal of Computer Science and Network Security*, 8(1):31–40.

Zen, M. and Vanderdonckt, J. (2014). Towards an evaluation of graphical user interfaces aesthetics based on metrics. In *2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS)*, pages 1–12. IEEE.

Zheng, X. S., Chakraborty, I., Lin, J. J.-W., and Rauschenberger, R. (2009). Correlating low-level image statistics with users-rapid aesthetic and affective judgments of web pages. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*, pages 1–10. ACM.

Appendix A

Attachments

All attached files are available in the online repository:
https://github.com/jirka/dash.thesis

A.1 Dataset

The repository contains the datasets which were used in the research tasks:

- **dashboard samples** presented in Section 5.4
- **descriptions of regions** provided by users in Section 7.1
- **user reviews** of the selected UI characteristics presented Sections 10.1 and 10.2

Identifiers of the users were anonymized.

A.2 Workspace

The repository contains the workspace for the analyses which were performed in the particular research tasks using the dataset of Appendix A.1:

- **the scripts for preparation of the workspace used by Dashboard Analyzer** for:
    - analysis of the pixel-based metrics (Chapter 6)
    - analysis of the object-based metrics (Chapters 7)
    - analysis of the modified object-based metrics (Chapter 8)
    - debugging of the method for segmentation of dashboards (Chapter 9)
- **the scripts for analysis of the users reviews** presented in Chapter 10

A.3 Results

The repository contains the `.odt` files summarizing the results of the analyses (Appendix A.2) corresponding to the particular research tasks:

- **Analysis of Pixel-based Metrics**:
  - the values measured by the pixel-based metrics (Sections 6.2, 6.3, and 6.4)
- **Analysis of Object-based Metrics**:
  - the results of the study of visual perception of objects (Section 7.1)
  - the average values of UI characteristics measured by the Ngo's metrics, including the values of the metrics' objectivity and decisiveness (Section 7.3)
- **Design and Improvement of Metrics**:
  - the results of the study of the impact of color on object-based metrics (Section 8.2)
  - the average values measured by the modified versions of the Balance metrics, including the values of the metrics' objectivity and decisiveness (Section 8.3)
- **Automatic Segmentation of Dashboards**:
  - the results of the quantitative comparison of the user descriptions with the generated descriptions (Subsection 9.3.2)
  - the results of measuring Balance using the user and generated descriptions and comparison of the results (Subsection 9.3.3)
- **Comparison of Metrics with Reviews of Users**:
  - the results of Experiment 1 (Section 10.1)
  - the results of Experiment 2 (Section 10.2)

Appendix B

Software

B.1 Dashboard Analyzer

The source code, description, and license terms are available in the online repository:
https://github.com/jirka/dash

The source code contains implementation of:

- the pixel-based metrics described in Chapter 6
- the Ngo's object-based metrics described in Chapter 7
- the modified versions of the Balance metrics described in Chapter 8
- the method for the segmentation of dashboards described in Chapter 9

The software allows users to generate the results and values presented in this research using the dataset presented in Appendix A.

B.2 Interactive Survey Tool

The source code, description, and license terms are available in the online repository:
https://github.com/jirka/survey-tool