CHRISTIAN KAISER

A User Interaction Concept for an Informational Cockpit in Virtual Development

Master's Thesis

Graz University of Technology

Institute for Knowledge Technologies
Head: Univ.-Prof. Dipl-Inf. Dr. Stefanie Lindstaedt

Assessor: Dipl-Ing. Dr.techn. Vedran Sabol

Advisors: Dipl-Ing. Selver Softic and
Dr.techn. MSc Eduardo Enrique Veas

Graz, September 2014

On the journey of a master's thesis, lots of people and actions influence what happens and try to put the work into the direction of success and deserve the thankfulness I want to express. Among those people, some deserve heartfelt thanks, as they helped me when I got stuck, pushed me to keep focused or gave me the background needed which only a family can give to write a thesis. I want to name...

Marion Schimpl, my darling, you had to bear with my moods and always supported me in my actions. I love you so much.

Maria & Ernst Kaiser, mum and dad, you gave me the chance to study, trusted and never questioned me, and you provided the basis for my life.

René Kaiser, my big brother, you and only you are the reason why I studied; you have influenced my life more than you can imagine.

Andrea Denger and Johannes Fritz, your wise input, motivation dialogues, feedback and belief in me helped me a lot; you do a really good job no matter what you do, and I am thankful for the chance to work with you.

Manfred Rosenberger, Thomas Traunmüller, David Purkarthofer and Jürgen Zernig, all of you put tremendous effort into implementing the proof of concept as a team. I really enjoyed working with you; the spirit was amazing.

Selver Softic, with a nice mixture of criticism, praise and input you did not only supervise my thesis, but also me as a person in my development.

Alex Stocker, thank you so much for helping me in difficult times.

Bernd Fachbach and Markus Zoier, thank you for offering me the chance to do my thesis within the FIND project, which is based on your expertise and therefore the basis of my work.

Vedran Sabol and Eduardo Enrique Veas, as supervisors from the university, you gave me scientific input when needed, brought in your expertise and showed me a view on the topic from outside the project team.

The department Information & Process Management and to the whole Virtual Vehicle Research Center, thank you for providing a perfect environment. Thanks to all the 20 test users of the Thinking Aloud test.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, ________________________ ____________________________

Date Signature

Eidesstattliche Erklärung[1]

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am ________________________ ____________________________

Datum Unterschrift

[1]Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008; Genehmigung des Senates am 1.12.2008

Acknowledgement

The author would like to acknowledge the financial support of the "COMET - Competence Centers for Excellent Technologies Programme" of the Austrian Federal Ministry for Transport, Innovation and Technology (bmvit), the Austrian Federal Ministry of Science, Research and Economy (bmwfw), the Austrian Research Promotion Agency (FFG), the Province of Styria and the Styrian Business Promotion Agency (SFG).

Furthermore the author would like to express thanks to supporting industrial and scientific project partners, namely "Audi", "Bosch", "Porsche", "Volkswagen" and to the Graz University of Technology.

Abstract

This master's thesis investigates interfaces for visual search in an automotive enterprise environment. The work in this thesis contributes to research in human computer interaction and usability of the information retrieval process in engineering. Literature research on underlying topics, for example investigated interfaces, is provided as well as a user interaction concept for an informational cockpit. The cockpit consists of a visual user interface to search for information in an enterprise environment. It takes keywords as input and provides search results both as a list and in a relation graph called *chord diagram*.

Online search engines are able to find relevant results for a query due to a proper amount of data and the special web architecture using links. Based on that, a relevance calculation is optimised by several features to provide the most relevant results at the top of the result list. Conversely, in a more restricted world of enterprise search, the result relevance calculation is an ongoing research topic; the best results might not be found among the top listed ones. As users are experienced with online search engines, they might not expect that. A solution is to narrow down the number of results using a faceted search, keeping results sharing chosen facets in the result list only. Beside that, the possibility of bookmarking results and search queries was addressed.

The concept of a new faceted search interface was realised in a proof of concept implementation, and two different versions were evaluated by test users. Findings about varying preferences and implementation improvement opportunities are mentioned in the evaluation. The faceted search interface versions were embedded in a proof of concept implementation of the information cockpit user interface concept provided; therefore, findings for the overall information cockpit are provided, as well. Overall, it could

be concluded that the demand for interactive visualisations is available; however, it seems that usability is a key success factor and the reason why people in engineering struggle to change.

Contents

List of Figures

1. INTRODUCTION

Facing a massive growth of data both online and offline, a user has to deal with an information overload. To assist the user in finding the information currently needed for a task, online search engines and enterprise search engines aid in the exploration of, for example, heterogeneous documents, websites, intranet platforms or databases. Improving the productivity of knowledge workers is probably the biggest recent challenge in management [Drucker, 1999]. Turkay et al. [Turkay et al., 2014] state, that *"data analysts have to deal with increasingly growing volumes and a diversity of highly complex, multi-dimensional and often weakly-structured and noisy data sets and increasing amounts of unstructured information."* While web search is considered as a solved problem due to optimised algorithms for ranking results to meet user needs, many different approaches of ranking results by enterprise search engines exist; however, ranking results appropriately in enterprise search is an ongoing research topic. One possible solution is to provide assisting features, for example a faceted search interface, based on metadata of results, assists to narrow down the result set. [Kaiser et al., 2013]

> *"While a simple query box and a ranked list of search results may be all that is needed for targeted searches, these interfaces do not adequately meet the needs of searchers in many other situations (e.g. when their goal is not well defined or ambiguous, or when there is a desire to explore the breadth of information available for a specific topic). For such search tasks, providing visual and interactive support for the fundamental activities of crafting a query and exploring the search results has the potential to greatly improve the ability of searchers to find the information they are seeking."* [Hoeber et al., 2009]

Virtual vehicle developers, the main targeted users of the information cockpit developed in this thesis, have the following tasks: development, simulation and optimisation of vehicle parts or simulation and optimisation of a full vehicle, for example for the purpose of emission reduction. These tasks are representative for technical engineers in the automotive industry working in the area of simulation. A virtual simulation takes a virtual model of a physical object based on measurements as input for calculations. To set up a virtual model of a physical object for optimisation calculations, information about the real physical object is needed to make the model more accurate. Different departments may have useful information chunks available. If users are not familiar with a particular information system or file system and its structuring concept, then they face a serious challenge when searching and gathering information by using the structuring concept. In a future workplace scenario, a user should not have to deal with such problems - the information system should take over the task of presenting the information in a unified, understandable way. This idea is, to a certain degree, comparable to an online search: instead of scanning or searching different web pages to find information, search engines are used as a single point of access. In an enterprise environment and for simulation developers as the targeted group, the describing name *information cockpit* fits, as the cockpit can be used for different purposes: [Kaiser et al., 2013]

- As a basis for collaboration on results found.
- To get in touch with authors of documents of the topic searched for.
- As an information source for the development of simulation models.
- When providing the information about relations between results and facets, patterns can be detected and give insights.
- To provide an overview of activities on a topic in connected data sources.

In this master's thesis, a concept for an information cockpit was designed and a proof of concept built. The concept design was presented at the *International Conference on Web Information Systems and Technologies* (WebIST) [Krempels and Stocker, 2013], Aachen 2013.

The information cockpit developed represents a unified visual point of access providing search and visualisation of information stored in heterogeneous data sources. Due to meta-information, it is possible to drill down a result set via a faceted search. Available meta-data is in use and follows the Dublin

Core specification. Examples attributes of the Dublin Core specification are *creator*, *format* or *date*. [Dublin Core Metadata Initiative (DCMI), 2014]

The focus of this work is on the concept and design of the graphical user interface for a target user group within the automotive industry. The concept and design of a graphical user interface has been identified as a key success factor for faceted search user interfaces [Hearst, 2006]. The concept is based on visualisations offered by available visualisation libraries, such as the Data Driven Documents (D3) JavaScript library [Bostock, 2013]. A prototypical implementation has been evaluated.

1.1. MOTIVATION

The time for retrieving all information needed for a calculation in simulation data management takes on average 51% of the worker's time [Krastel, 2001]. Sometimes information can only be accessed from within a department's borders or if certain access rights are granted. An information cockpit with a single point of access will reduce the time for authentification and search. As searching with a textual cockpit will accelerate the information retrieval process, visualisations which highlight certain aspects of the data may provide the missing overview on big amounts of data. Examples for single purposes exist; however, a recommended solution for enterprise search is missing. Free visualisation libraries provide a basis for implementations in an enterprise environment. The important aspect is the difference between an online search engine like Google and an enterprise information cockpit. Google is able to provide an excellent prediction as to which website is searched for, due to its relevance calculation. However, the relevance calculation for information in an enterprise environment, for example file systems and structured information in databases, is challenging and not sufficient yet. This leads to the problem that a query may find the information searched for, but due to an incorrect relevance prediction, the information of interest is not ranked among the top results. Users tend to look at the top results only, as the top results produce high perceived usefulness if the information searched for is included. If the information is not found within the first results, then the users' perceived usefulness is lower. A model of Davis

[Davis, 1986] explains that there is a direct influence of perceived usefulness to system usage.

As large companies produce gigabytes of data per day, the relevant information may even not be found within the first 1000 results. This is the reason why a faceted search is used to provide a drill-down functionality to assist the user in refining his query. The more precise a query is, the fewer possible results are available, increasing the likelihood that the information searched for will be found among the first results. The output quality of a system is not the only characteristic influencing the perceived usefulness; amongst others, a subjective norm or things, images and experience influence it as well [Venkatesh and Davis, 2000]. Enterprise tools are often packed with features and designed conservatively in such a way that users of previous versions do not face any problems when changing. Sophisticated visualisations may exceed subjective norms for users of such enterprise tools, which may be positive or negative for their perceived usefulness.

Literature research showed that evaluations of enterprise search interfaces rarely exist. Therefore, the evaluation in Chapter 6 is pioneer work and can give insights into the usage of visualisations in enterprise search in an automotive industry environment as well as into preferences in the comparison of text-based versions to visualisation-based versions.

1.2. ENVIRONMENT

The use case for the concept and the implementation of this master's thesis is a vehicle simulation engineer, a knowledge worker who builds models of vehicle parts to calculate predictions of, for example, the rigidity of vehicle parts in special situations. The following sketches the scenario of this use case:

A single crash test for a passenger vehicle costs about $100,000.00 or more and requires days of preparation work and knowledge to interpret the results correctly. A single crash test dummy costs about $125,000.00. A vehicle prototype used in a test costs about $250,000.00. [Hoffman, 2014]

At least in the European Union, "by law, all new car models must pass certain safety tests" [EURO NCAP, 2014] before entering the market. Safety is a key element when deciding about purchasing a car. [EURO NCAP, 2014]

The later a need for a change of a product occurs, the more expensive it is. A product almost ready for sale therefore entails the highest change costs. Modifications of a car due to a crash test failure are more expensive if the test is done late in the car development process, as related parts are influenced and have to be redeveloped as well. For example, if the B-pillar does not meet the safety requirements, then it will be redesigned and its form and the weight balance will change. As a result, the space for interior will be affected. An example of a B-pillar is shown in Figure 1.1.

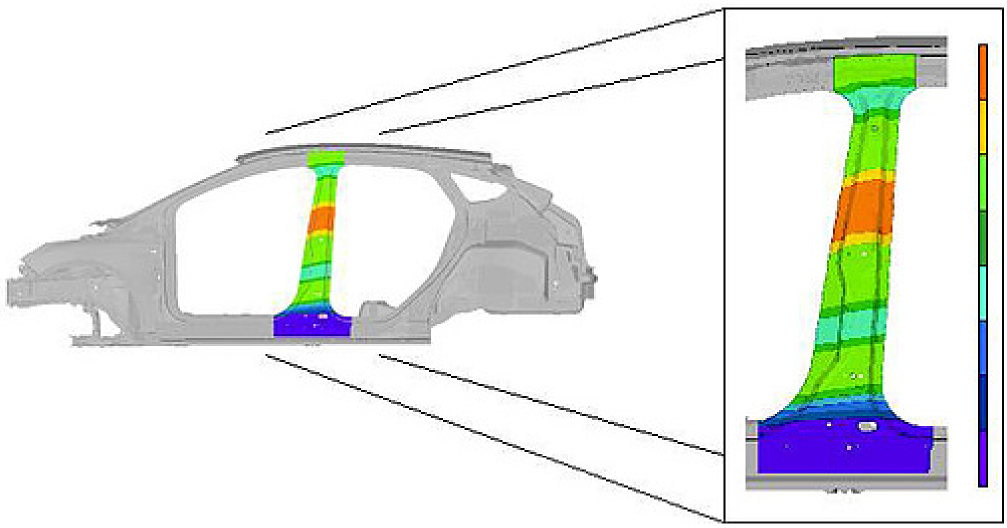


Figure 1.1.: B-pillar design of a Ford Focus. The colour shows the thickness (blue: thin; orange: thick). source: [Weissler, 2014].

An alternative was developed to reduce development costs. Crash tests have been analysed to build up virtual models, which try to reproduce the behaviour of a crash test. If parameters change, then the behaviour can be predicted virtually without the need for a real-life test. It has to be considered that the significance of simulation results amongst others depends on the virtual model quality. Real-life tests at the end of the development process called NCAP (New Car Assessment Program) crash tests, which are needed to be granted admission to the European car market, show the correctness of the models. As mentioned, if anything is not correct at the end of the development, a change is more expensive than earlier in the development process. Therefore, vehicle simulation engineers try to be as accurate as possible when developing models. [Eigner, 2009]

The engineer has a great demand for information about the vehicle to simulate; hence, access to various heterogeneous data sources is needed. An excerpt of possible information chunk sources is listed below:

- A bill of materials, available as a document (Microsoft Word, pure text file, PDF, etc.) or as a database entry.
- Developer decision protocols, as documents or Wiki entries.
- Meeting minutes, as documents.
- E-mails, as pure text or documents.
- CAD drawings, as files of special engineering tools or images.
- Calculations, as files of special engineering tools, documents or database entries.
- Other simulation models.

Additionally, the incomplete list of information sources may be multiplied by the number of departments involved in the development of the vehicle and its sub-parts, for example for the interior, covers and openings, power train and the engine, due to the fact that departments do not share all information for security reasons. Information artefacts can be available in several versions, where some of them are marked as *in progress* and others are *released*.

Information artefacts or chunks of natural language, for example text documents, e-mails or websites are defined to be unstructured due to a possibly missing link to a related topic. As e-mail provides an author, recipient(s), maybe a title and content, clustering into topics might be difficult. The extraction of relevant information out of artefacts, so-called facets, is done with a mix of statistical models, a system of rules and reference books. Information artefacts using a specific schema, for example database entries, are structured. Facets of structured data are available as machine readable; some normalisation may be needed. Information artefacts using a schema for some parts are semi-structured, for example Microsoft Excel files. Facets of semi-structured data are extracted using the methods of both unstructured and structured data.

1.3. RESEARCH QUESTION

Using characters and digits representing facets or results in an enterprise search interface is widely used. Style-sheets are in use for graphical design. Humans have certain visual perception capabilities to process so-called pre-attentive attributes in parallel. This cannot be used for text or numbers, as the brain processes the perception of text and numbers serially. Aiming at an improvement in time, for example when choosing facets to narrow down a set of results, human visual perception capabilities could add such an improvement. Visualisations using graphical designs may be used in the realisation. Therefore, research question number one asks how to achieve this:

> How can visualisations add an advantage within the context of enterprise search interfaces?

For the evaluation of a concept, a proof of concept implementation provides a basis for discussions. However, many possible ways of implementation exist. Consequently, research question number two arises:

> How could a proof of concept be implemented?

1.4. THESIS IN A PICTURE

In this section, the chapters of the thesis are introduced briefly.

To be able to answer the research questions, discussions with colleagues about the daily work of simulation engineers were held as well as literature research about visual information processing of humans. Insights about the power of human perception, attentive and pre-attentive attributes or, for example, usability, were the basis for the designed mockups. As the aim is to use the knowledge of human perception to develop an enterprise search interface, knowledge about how data is extracted from data sources and how drill-down mechanisms can be used was gathered. For the design, basics about dashboards and search interfaces and common mistakes of implementations were taken as input. The findings are described in Chapter 2.

A mockup of the concept was drawn and discussed. The concept is described in detail in Chapter 3. It is the basis of the proof of concept implementation.

The proof of concept was implemented focusing on aspects of the faceted search interface and result view. The two implemented versions are described in Chapter 4. As the focused faceted search interface and result view are components of a search interface, the technologies chosen, the development method and the other components of the proof of concept are described, as well.

Chapter 5 guides the reader through a case study, to show how the information cockpit is used and how it assists the user in tasks.

Chapter 6 presents the evaluation of the implemented concept. 20 test users were questioned to fulfil search tasks using different versions of the faceted search. The test users were recorded in audio and video. Quantitative as well as qualitative evaluation results were collected and are presented. The results are the basis for a discussion of the implementation.

In Chapter 7 a summary of the thesis is provided, followed by an outlook of possible future work.

2. VISUAL SEARCH INTERFACES

The following chapter deals with visual search interfaces and related work. Before visualisation history and usability can be explained, human perception of humans "dominating sense: vision" [Few, 2013] and information processing have to be introduced.

Ware emphasises the importance of visual perception as follows:

> *"Why should we be interested in visualization? Because the human visual system is a pattern seeker of enormous power and subtlety. The eye and the visual cortex of the brain form a massively parallel processor that provides the highest bandwidth channel into human cognitive centers. At higher levels of processing, perception and cognition are closely interrelated, which is the reason why the words 'understanding' and 'seeing' are synonymous. However, the visual system has its own rules. We can easily see patterns presented in certain ways, but if they are presented in other ways they become invisible." ... "The more general point is that when data is presented in certain ways the patterns can be readily perceived. If we can understand how perception works, our knowledge can be translated into guidelines for displaying information."* [Ware, 2013]

This chapter underpins why knowledge about human perception is the basis for designers [Bradley, 2011]. Section 2.1.1 explains the visualisation process and Sections 2.1.2 and 2.1.3 focus on human perception. Section 2.1.4 shows visualisation examples starting from historical cave paintings up to interactive visualisations. Furthermore, Section 2.1.5 introduces the usability and evaluation of visualisation. Related work and knowledge about common concepts of information retrieval follow. Facets are introduced in Section

2.2.1; an example of facet extraction from documents and the search index of a search engine are described. The facet hierarchy is described, which forms the basis of the faceted search. The faceted search is described in Section 2.2.2. Section 2.2.3 introduces information cockpits, search interfaces, dashboards and visual faceted search interfaces.

2.1. VISUAL INFORMATION PROCESSING

2.1.1. THE VISUALISATION PROCESS

This section introduces the visualisation process, the basis for visual perception. Ware describes the process of data visualisation using four stages: [Ware, 2013]

- Data storage.
- Data preprocessing, for example manipulating the data.
- Graphic engine to compute visual representation. Manipulable by user input.
- Human perception.

The stages are shown in Figure 2.1. Stored data is transformed and computed with the graphic engine before the user can interpret it. Data exploration, view manipulation and data gathering are feedback loops. Data gathering involves different data sources. Data exploration involves, for example, reduction or "looking for *interesting* things" [Ward et al., 2010] like anomalies, clusters or trends. View manipulation involves user interactions, for example the highlighting of a subset of data. [Ware, 2013, Ward et al., 2010]

2.1.2. HUMAN PERCEPTION

The power of visualisations depends on their perception, which is why this section explains human visual perception. [Turkay et al., 2014]

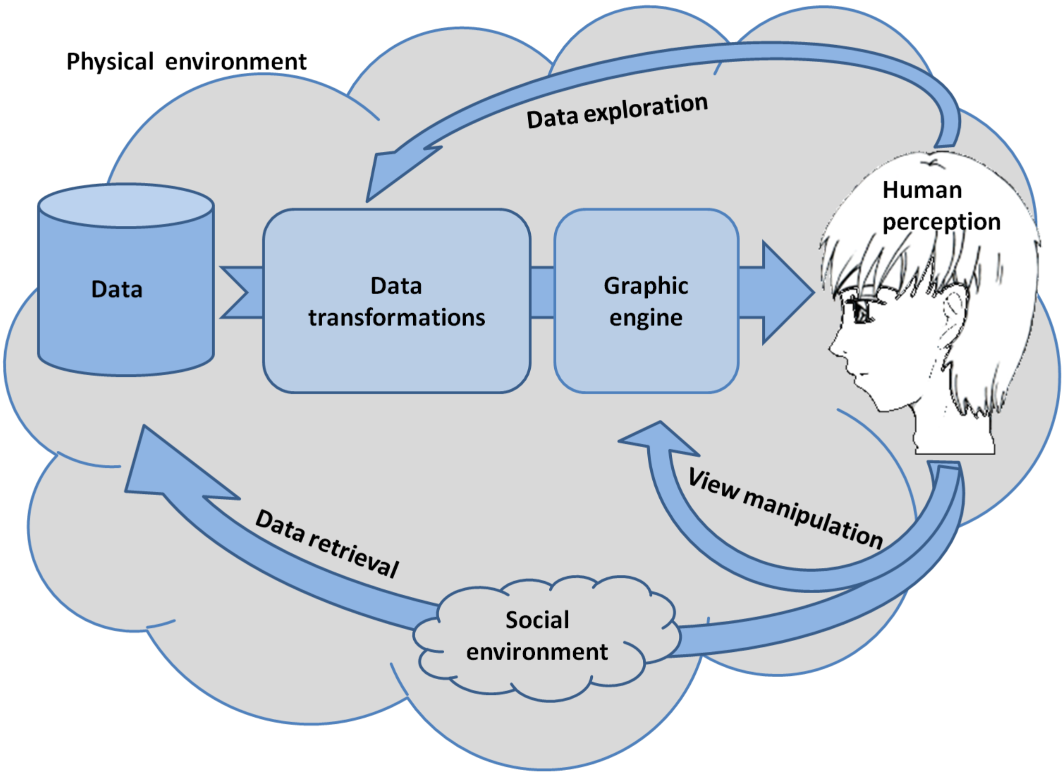
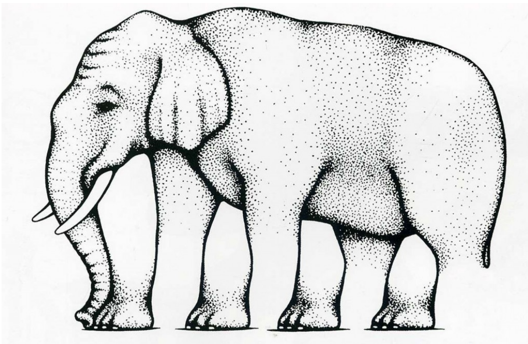


Figure 2.1.: The visualisation process. Compare [Ware, 2013].

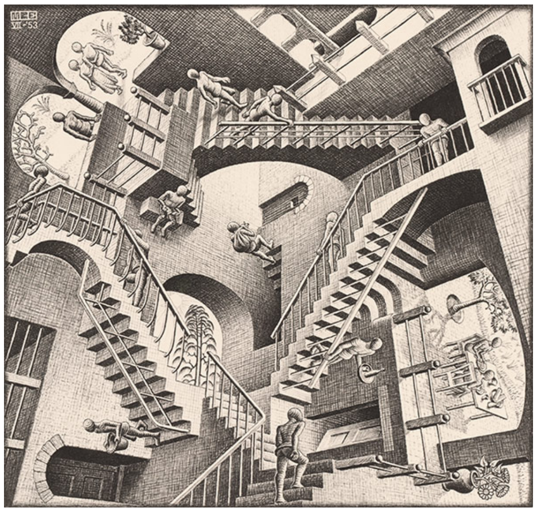
> *"Many definitions of perception exist, most define perception as the process of recognizing, organizing and interpreting sensory information."* [Ward et al., 2010]

Trying to interpret the environment, the brain makes assumptions. Assumptions can lead to misinterpretations, e.g. illusions. [Ward et al., 2010] Examples of illusions are shown in Figure 2.2. Due to the illusion in Shepard's *L'egs-istential Quandary*, which is known as the *impossible elephant*, the elephant seems to have no legs or even more than four legs, as shown in 2.2a. Another example is Escher's drawing called *Relativity*, exhibited in Figure 2.2b, where the illusion seems to outsmart relativity.

The assumptions and interpretations are a result of the process of the perceptual system. Two main approaches - one deals with measuring perception,



(a)The *impossible elephant*. [Shepard, 1990]



(b)*Relativity*. [Escher, 2013]

Figure 2.2.: Optical illusion examples.

the other one with models - exist in perception research. The approaches can be related to each other. Measurements, e.g. asking a user which of two lines is longer, can be used to develop models for future predictions. Understanding human perception is important in visualisation development, in order to be able to make use of human visual perception capabilities when needed. As the Kano model seen in Figure 3.2 explains, as time goes by, the state-of-the-art usability level, expected by users, increases. Exploiting human visual perception capabilities may be state-of-the-art soon.

An example of visual perception capabilities is the pre-attentive attribute colour hue. Colour is perceived using photoreceptors. Using specific colours in fact means using specific electromagnetic frequency lengths in the visible light spectrum between ultraviolet, about 380nm, and infrared, about 700nm. This spectrum is dependent on the individual user. "The retina of the human eye contains the photoreceptors responsible for the visual perception of our external world" [Ward et al., 2010]. The photosensitive cells are called *rods*, sensitive to intensity, and *cones*, sensitive to colour perception. [Ward et al., 2010, Huch, 2011] Most of the approximately 6 million cones are located around the *fovea centralis* dent, which is the place where humans see sharpest. When humans fix their gaze on an object, the human optical apparatus manages to bundle the light exactly in the fovea centralis. [Huch, 2011, Ward et al., 2010]

Image 2.3 shows a drawing of a cross-section of an eye on the left side and marks some important parts, e.g. the iris, the lens and the fovea (short for fovea centralis mentioned in the last paragraph). On the right side of the image, details of the fovea, the photoreceptor cells - cones and rods, are enlarged and presented in a simplified way.

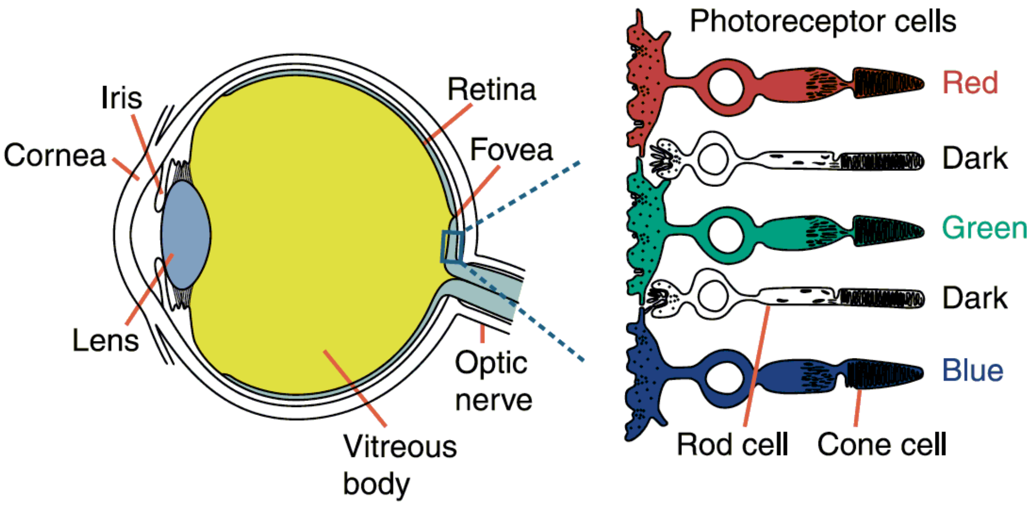


Figure 2.3.: The human eye and the photoreceptor cells *cone* and *rod*. [CUDO, 2013]

The human fovea focuses one small area after the other with jerky movements of the eye, very quickly, to detect the visual environment. However, not everything is sensed; only a portion. A resulting piece of advice is, for example, that a message has to be clear and easily understandable. [Bradley, 2011, Few, 2013]

Bradley outlines the following detections of the environment scanning without conscious awareness as they can be addressed in design to attract attention: [Bradley, 2011]

- Motion;
- Edges of shapes;
- Colour;

- Contour;
- Contrast.

Wolfe and Horowitz published a list of attributes which have the "ability to guide deployment of attention" [Wolfe and Horowitz, 2004]. Visual search is described by Wolfe and Horowitz as follows:

> *In a typical visual search task, an observer looks for a target item among distracting items.* [Wolfe and Horowitz, 2004]

In Wolfe and Horowitz's opinion article, guidance is separated from the main pathway to object recognition; thus, the term pre-attentive is avoided. "Attributes that might guide the deployment of attention" are listed instead. Undoubted attributes are: [Wolfe and Horowitz, 2004]

- Motion;
- Orientation;
- Colour;
- Size.

Treisman [Treisman and Gelade, 1980] differentiates between features which are "registered early, automatically and in parallel across the visual field" and objects which are "identified separately and only at a later stage, which requires focused attention." It is assumed that the visual scene is scanned in a number of separable dimensions: [Treisman and Gelade, 1980]

- Spatial frequency;
- Orientation;
- Colour;
- Brightness;
- Direction of movement.

Mackinlay [Mackinlay, 1986] extends findings of Cleveland and McGill [Cleveland and McGill, 1984] about accuracy of attributes in quantitative perceptual tasks to findings in ordinal and nominal perceptual tasks. Size, called *area* by Mackinlay, is placed moderate in the middle of *more accurate* and *less accurate*, slightly on the less accurate side for quantitative tasks and definitely on the less accurate side for ordinal and nominal perceptual tasks. The difficulty of judging relative sizes of different areas accurately is mentioned if the areas are not different enough in size. As for the proposed usage to represent how much results share a facet in the proof of concept implementation, the exact amount or the absolute difference between elements is not of high priority in the concept. Mckinlay comparably mentions: "small misjudgements about the size of an area only lead to small misperceptions about the corresponding quantitative value that is encoded" [Mackinlay, 1986].

Beside that, it should be mentioned that the influence of emotion to perception and cognition, in particular attention, memory, and reasoning, is huge and underestimated. [Holzinger et al., 2013]

2.1.3. ATTENTIVE AND PRE-ATTENTIVE ATTRIBUTES

Not all features or attributes of visualisation are processed equally. This chapter explains why and shows related work.

> *"Human vision rapidly and automatically categorizes visual images into regions and properties based on simple computations that can be made in parallel across an image."* [Healey and Enns, 2012]

There are differences in interpretation time between reading text and numbers or visual attributes. A self-experiment is provided in Figure 2.4. Both subfigures show a list of letters and the task is to detect meaningful words or a sentence. Measuring the time, it is generally agreed that a test person is faster in Figure 2.4b. The reason is related to the information processing of humans. The detection of words in Figure 2.4a is complex to process pre-attentively and therefore involves serial attentive processing, whereas in Figure 2.4b, due to the visual attribute colour hue, it is done very fast, pre-attentively and in parallel. [Few, 2004]

Andrews summarises the findings of [Few, 2004] as follows: [Andrews, 2013b]

- *Text and numbers have to be processed attentively, which requires cognitive effort and proceeds in serial (slow).*
- *Certain visual attributes can be processed pre-attentively, which happens without conscious effort and in parallel (fast).*

AJDSFHSDJUHFSIUGFHSRNG
SJVSJTHISJFNSFJBNSBJSF
DJFHDFSJUDFUISFDJSDFJJ
KJSFVUIVNSFUVWTZUIOOPD
TWUDHCHRISTIANSGZXHBXH
XKCJHIUSAJNIUWAHCYBNMM
QWETGHIJAPOALORSMASTER
HEIALTHESISREISTPOLDFA
EIFJUIEJFXYBNMAUIILASK

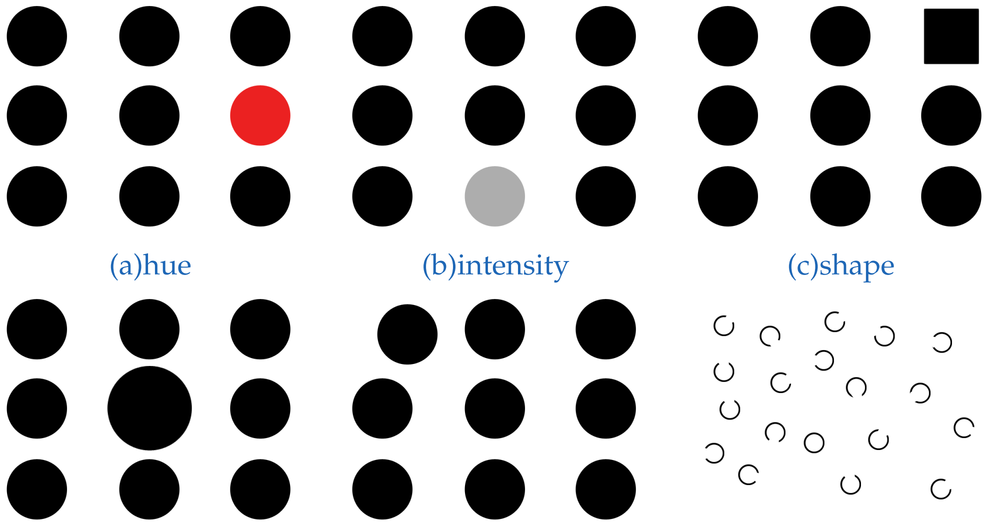
(a)words can hardly be detected.

AJDSFHSDJUHFSIUGFHSRNG
SJVSJTHISJFNSFJBNSBJSF
DJFHDFSJUDFUISFDJSDFJJ
KJSFVUIVNSFUVWTZUIOOPD
TWUDHCHRISTIANSGZXHBXH
XKCJHIUSAJNIUWAHCYBNMM
QWETGHIJAPOALORSMASTER
HEIALTHESISREISTPOLDFA
EIFJUIEJFXYBNMAUIILASK

(b)coloured words can be detected.

Figure 2.4.: Two examples of a list of Arabic characters, compared [Bradley, 2011, Few, 2013, Ware, 2013].

As there are several visual attributes, they are different in human perception. Figure 2.5 shows some examples. Andrews mentions the attributes colour hue, an example of which is shown in Figure 2.5a, colour intensity, an example of which is shown in Figure 2.5b, shape, an example of which is shown in Figure 2.5c and size, an example of which is shown in Figure 2.5d as "particularly useful for encoding categories" [Andrews, 2013b]. This is due to the fact that pre-attentive effects are unequally strong. Usage of several dimensions where the user can use any dimension to distinguish the same groups, for example in colour, size and orientation, is called redundant coding. The issue is that the search is not necessarily improved in redundant coding. Conjunctions of features are difficult to find, for example searching for red squares within a bunch of elements which are either blue or red and at the same time either square or circular. However, there are some feature pairs which do allow for conjunction without losing the pre-attentiveness. Andrews [Andrews, 2013b] states: "Conjunctions of motion, depth, colour, and orientation can remain pre-attentive." [Ware, 2013]



(a)hue (b)intensity (c)shape

(d)size (e)spatial position (f)closure

Figure 2.5.: Chosen visual attribute examples, compared [Few, 2004, Few, 2013].

2.1.4. BACKGROUND OF VISUALISATION HISTORY

This section provides an excerpt from visualisation history, starting with cave paintings and finishing with interactive visualisations. More detailed historical examples can be found in Michael Friendly's gallery [Friendly, 2014] or in [Ward et al., 2010].

Early examples discovered in research are cave paintings [Peter, 2014]. One of the oldest European cave paintings can be seen in Figure 2.6, palaeolithic cave paintings about 15,000 – 20,000 years old from the Tito Bustillo cave in northern-Spain, showing horses. [Schabereiter-Gurtner et al., 2002]

Cave paintings were invented 8,000-13,000 years earlier than writings (6,000 B.C.) [Haarmann, 2002]. However, the purpose of cave paintings is still unclear.

Later in history, combinations between text and visualisation techniques were used, for instance the first *time visualisation* in 1765 by Joseph Priestley, using lines as shown in Figure 2.7 [Davis et al., 2010]. Priestley, a history

Figure 2.6.: Cave paintings from Tito Bustillo (Spain), source: [De Balbin Behrmann, 2014].

teacher, invented the visualisations for his own purpose. However, they reached a large audience when published [Twyman, 1990].

Tree visualisations, for example, were used to visualise the system of human knowledge. They represent hierarchical structures where an explicit path between a root node and every leaf node exists. Two examples are shown in Figure 2.8.

In Figure 2.8a the ordering schema of Diderot and d'Alembert is shown as a taxonomy. In Figure 2.8b the data is shown as iconography, a "pictorial material relating to or illustrating a subject" [Merriam-Webster, 2014]. The figure has the structure of a tree, including a trunk, branches and leaves. Each branch has a meaning; a related text is placed in an ellipse on the branch.

Tree visualisations start with a root node and may have child nodes. Figure 2.8a has the root node *ENTENDEMENT*, the French term for comprehension, and the child nodes *MEMOIRE*, *RAISON* and *IMAGINATION* for memory, reason (sanity) and imagination. Each child node has one or more child nodes. Figure 2.8b has its root node at the very bottom- it is the trunk of the tree.

Tree visualisations are well-studied and used. Currently, 275 different visualisation and layout techniques are referenced in the treevis.net project

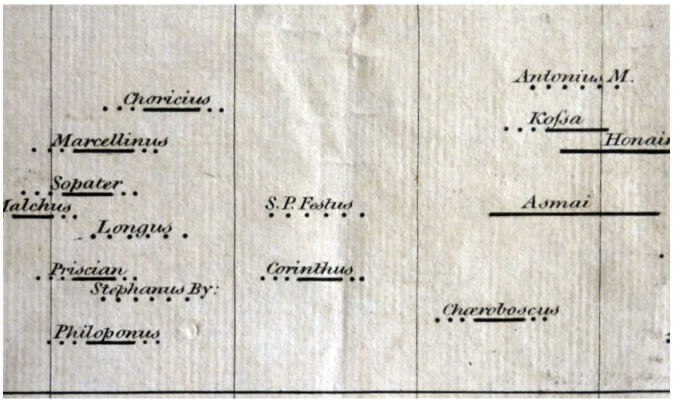
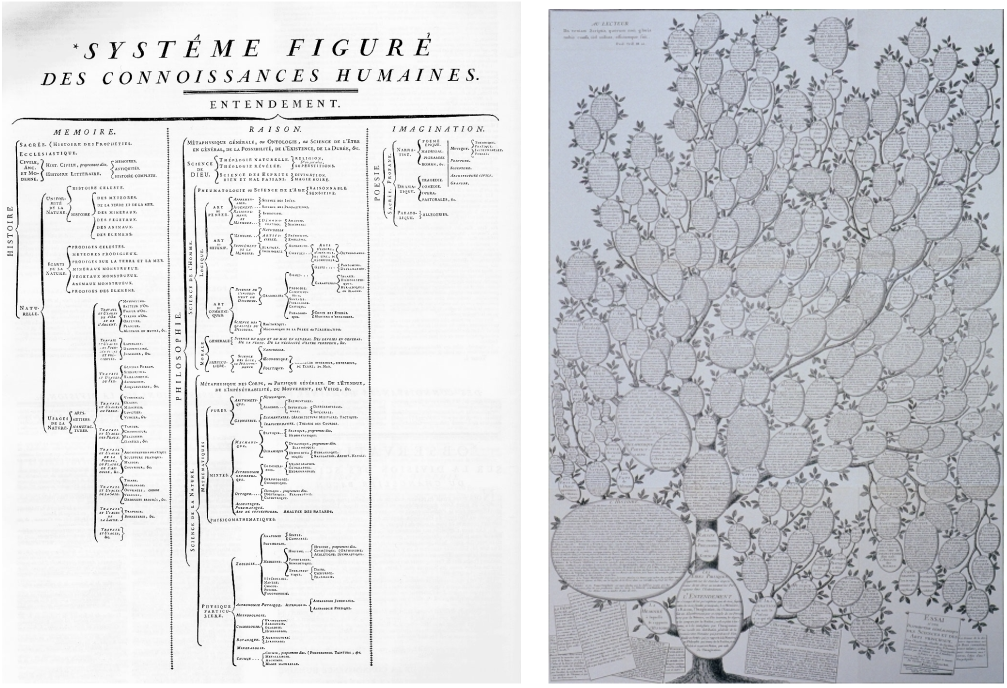


Figure 2.7.: "A Chart of Biography". A Time visualisation in 1765. Source: [Priestley, 1765].

[Schulz, 2014, Schulz, 2011]. Popular usages are file systems, library classification systems and family trees [Andrews, 2013b]. However, once they reach a certain amount of data and complexity, simple tree visualisations, without additional visual effects, turn hard to read. Take statistical data from government spending in Great Britain, as shown in Figure 2.9, where the sum is the root node and the children are the departments. The visualisation was published by the *The Guardian*, a newspaper in Great Britain. Additional information like the percent share of difference to the spending of the year before and the absolute amount of money spent lead to a hardly readable tree. The additional effects, such as the colour intensity (children have a lighter colour) and the bubble size in relation to the percent share are a solution example for managing complexity. In general, newspapers have tended to use information graphics more often in recent years. One reason why a tree visualisation with bubbles representing the amount of money and highlighting effects is used could be user attention, as colour and other visual features attract the attention of the human eye [Treisman and Gelade, 1980].

The Open Data Movement is a case, where big amounts of data are linked to each other, for example in the Linking Open Data project. Figure 2.10 shows the Linking Open Data cloud diagram from September 2011. The figure shows how big amounts of data can be represented, by keeping the overview present. The links are not necessarily hierarchical due to possible



(a)Taxonomy.
Source: [Diderot, 1751].

(b)Iconography.
Source: [Mouchon, 1780].

Figure 2.8.: Tree visualisation examples: The system of human knowledge.

cross-links between elements, either from the same or from different sources included. The Open Data Movement aims “at making data freely available to everyone” [W3C, 2014] by connecting “related data that wasn’t previously linked” [Heath, 2014]. [W3C, 2014, Heath, 2014]

Using layout techniques, additional information on the focus of an element can be provided. As a result, an overview is presented, enriched with details on demand. Figure 2.11 shows two different screenshots of a sunburst visualisation. Figure 2.11a is a screenshot of the sunburst visualisation when no mouse is placed over the interactive visualisation. Except for some information about fragmentation, no further information can be perceived. Mouse-over on any element provides information about the element itself; in this situation the percentage value, it highlights the hierarchy it belongs to and additionally, when the option *legend* is chosen, provides the hierarchy

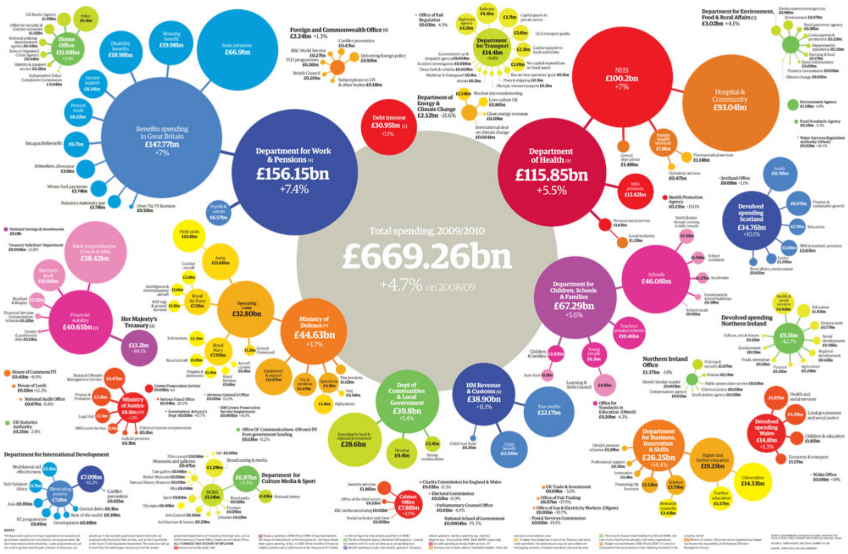


Figure 2.9.: *Government spending by department, 2009-10.* Source: [Guardian, 2014].

path to the element as a sequence of labelled arrows shown on the very top.

Interacting with visualisations is called epistemic actions- actions to uncover new information [Kirsh and Maglio, 1994]. Adequate visualisations allow drilling down and finding additional data on something in focus. [Ware, 2013] Ben Shneiderman provides a guide for information-seeking behaviour also applicable for interfaces, the so-called *visual information seeking mantra*: [Shneiderman, 1998]

> *"Overview first, zoom and filter, then details on demand."*

Keim extends the mantra to the following: [Keim et al., 2006]

> *"Analyse First - Show the Important - Zoom, Filter and Analyse Further - Details on Demand."*

What the original mantra by Shneiderman means in detail is summarised by Craft and Cairns as follows: [Craft, 2005]

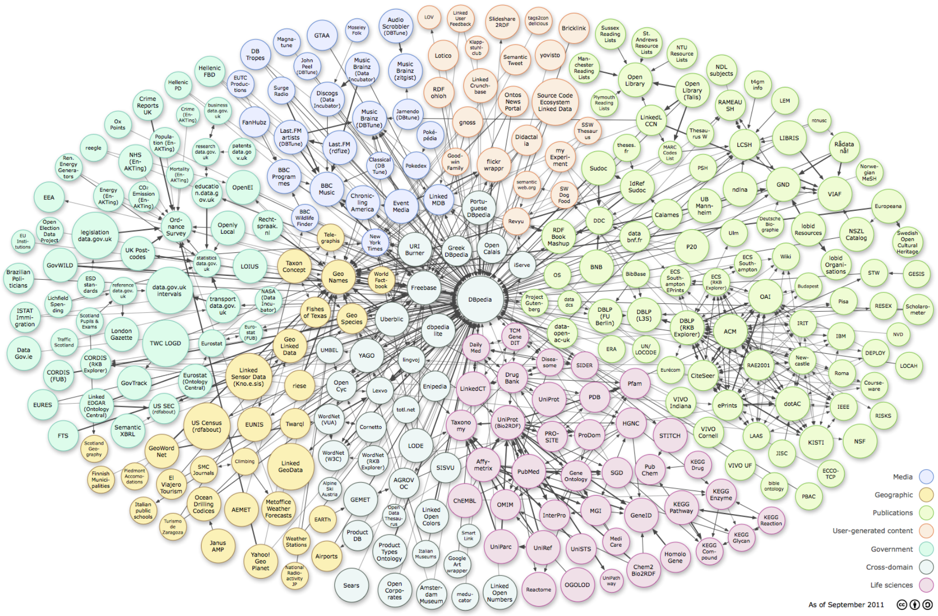
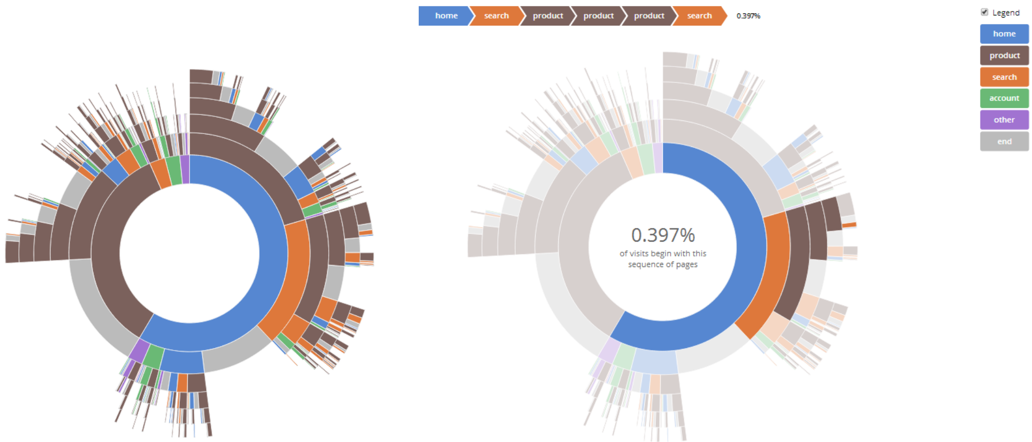


Figure 2.10.: Linking Open Data cloud diagram. Source: [Jentzsch, 2014].

- OVERVIEW FIRST: An overview can provide a general context for the understanding of the data, the big picture. Patterns can sometimes just be detected from an overview perspective. Significant parts can be identified for further inspection.
- ZOOM AND FILTER: Both features involve reducing the complexity of the representation by removing information. Zooming in enlarges data elements of interest and simultaneously removes objects which are out of focus. "Zooming can be regarded as filtering by navigation and change of representational vantage point." [Craft, 2005]
- DETAILS ON DEMAND: The number of data elements in the overview can vary from dozens to millions. Details on demand means adding additional information on a point-by-point basis to the view currently being used, like a tooltip on mouse-over or an action on selection.

When choosing a visualisation technique, it is important to take the user and possible categories of usage into account. Users can be divided into the following groups in the case of visualisations: [Ward et al., 2010]



(a)The overview. (b)Focus on an element.

Figure 2.11.: Sequences sunburst diagram. Source: [Rodden, 2014].

- RESEARCH: Determine the content and recognise patterns.
- CONFIRMATION: Criteria or pattern recognised in related data, analyse data to confirm hypothesis that criteria or pattern is present.
- PRESENTATION: Present data in a special form to public, visualisation helps to highlight facts.

2.1.5. USABILITY

This section points out definitions of usability and evaluation techniques and methods to ensure usability.

Usability is the interaction between the user and the machine [Düweke, 2011]. The ISO standard 9241-11 defines usability as: [ISO 9241-11, 1998]

> *"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use."*

ISO 9241-210 standard defines it as: [ISO 9241-210, 2009]

> *"A person's perceptions and responses that result from the use or anticipated use of a product, system or service."*

In the following section, four common evaluation techniques for information visualisation are described with the aim of ensuring usability: [Ward et al., 2010]

- USABILITY TESTS: Focusing "*the Five E's*: effective, efficient, engaging, error tolerant, and easy to learn" [Stone, 2005], test users are pleased to perform tasks while observing them. Difficulties, features used and the satisfaction are noted by an observer. The evaluation indicates if a goal or requirement is satisfied.
- EXPERT INTERVIEWS: Either visualisation experts or domain experts evaluate the visualisation. In a heuristic evaluation as this one, the visualisation is assessed against a checklist of features. A small number of participants may be satisfiable.
- FIELD TESTS: "Field tests are performed in the natural environment of the typical user and may last for weeks or months" [Ward et al., 2010, Plaisant, 2004]. The initial curiosity phase and the acquaintance can be filtered. Results may change significantly if clarifications happen over time.
- CASE STUDIES AND USE CASES: Visualisation researchers show real examples of the system usage to evaluate it. Examples should be sufficiently realistic.

Andrews [Andrews, 2008] mentions nine common evaluation methods for information visualisation, shown in Table 2.1. The purposes stated in the table are described like this: [Andrews, 2008]

- *"EXPLORATORY: Exploratory evaluation provides evidence of how an interface is used and what it is used for.*
- *PREDICTIVE: Predictive evaluation produces an estimate of user performance based on an interface design.*
- *FORMATIVE: Formative evaluation provides design feedback, often in the form of a list of problems and recommended solutions.*
- *SUMMATIVE: Summative evaluation provides an overall assessment of a single interface or a comparison of multiple interfaces, often in the form of numerical data which is statistically analysed."*

Method	*Type*	*Purpose*	*Description*
Observational Study	Testing	Exploratory	A longer term study following a small sample of users as they use an interface for their own tasks. Observations and anecdotal evidence are collected and assessed.
Action Analysis	Inspection	Predictive	An evaluator produces an estimate of the time an expert user will take to complete a given task, by breaking the task down into ever-smaller steps and then summing up the atomic action times.
Heuristic Evaluation	Inspection	Formative	A small team of evaluators inspects an interface using a small checklist of general principles and produces an aggregate list of potential problems.
Guideline Checking	Inspection	Formative	An evaluator checks an interface against a detailed list of specific guidelines and produces a list of deviations from the guidelines.
Cognitive Walk through	Inspection	Formative	A small team walks through a typical task in the mind set of a novice user and produces a success or failure story at each step along the correct path.
Thinking Aloud	Testing	Formative	Representative test users are asked to think out loud while performing a set of typical tasks. The insight gained into why problems arise is used to produce a list of recommendations.
Guideline Scoring	Inspection	Summative	An evaluator scores an interface against a detailed list of specific guidelines and produces a total score representing the degree to which an interface follows the guidelines.
Questionnaires	Testing	Summative	After using one or more interfaces for some typical tasks, test users are asked to rate the interface(s) on a series of scales.
Formal Experiment	Testing	Summative	A larger sample of users performs a set of tasks on one or more interfaces. Objective measurement data is collected and analysed statistically.

Table 2.1.: Table of evaluation methods. Source: [Andrews, 2008]

2.2. FACETED SEARCH INTERFACES

2.2.1. FACET EXTRACTION PROCESS

Related work and knowledge about concepts of information retrieval is the basis for a faceted search. Facets are introduced in this section. Facet hierarchies used for faceted search are described here, as well.

For the ease of understanding, the following fictive, but somehow close to reality, scenario is used: An accurate virtual model of a product is the task; therefore, information about the product is needed, in this case material specifications. While browsing through the network share, amongst others, text documents are found. For the identification of what a text document is about, the document could be read by the user, but in this scenario another 2000 text documents, each of them several pages long, are located in the same directory. To shorten the time of the identification process, assisting systems can provide a summary or a list of keywords of each document. "A well-established principle of human memory is that it is often easier to recognize a word or name than it is to think up that word" [Hearst, 2009b]. Representative words and metadata can be used as so-called facets. As described in 2.2.2, facets can be organised hierarchically into categories. "In search interfaces, categories are typically used either for selecting a subset of documents out from the rest, thus narrowing the results, or for grouping documents, .." [Hearst, 2009b]. In the following, the so-called facet extraction process is described.

A document consists, for instance, of text. The first task is to identify sentences. Indicators for sentences could be line breaks, full stops or capital letters. In the second step, each sentence is cut into word pieces. In the third step, the identified words are transformed into their main form, e.g. the word *gone* turns into *go*. When all sentences are broken down into arrays of words, one approach is to count the appearances of the words to give the user a list of the most used words. To normalise the results, so-called *stop words* like *and*, *or*, etc. are filtered with the help of black lists. [Manning et al., 2008]

The fictive example in Figure 2.12 shows sentence *a* from fictive document A.

The engine needs a 3 seconds heating phase and a 1 minute cool down.

This is transformed to:

engine need 3 second heat phase 1 minute cool down

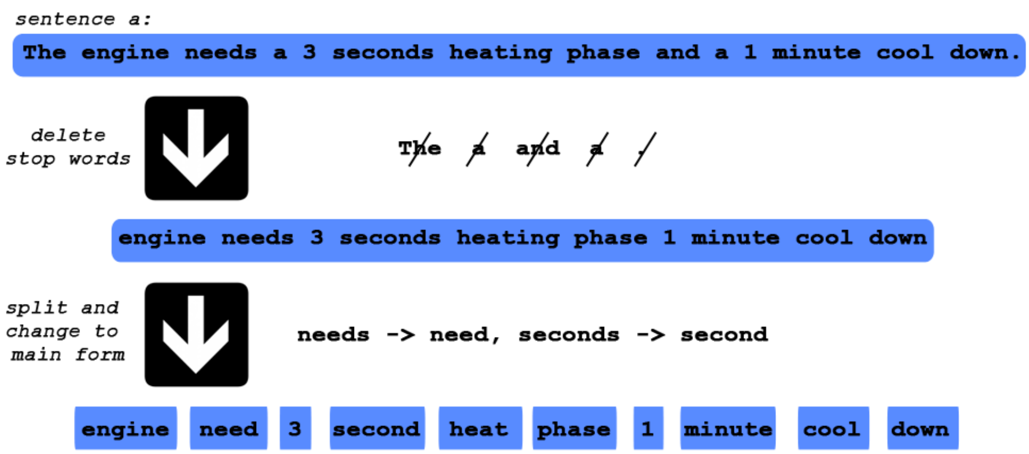


Figure 2.12.: Facet extraction.

In fictive document B the sentence (*b*) is:

The seat-heating needs 36 seconds before reaching 36 degrees.

This is transformed to:

seat heat need 36 second reach 36 degree

The index is an array of all words found in the searched information sources of the transformed sentences. If documents A and B were the sources for the index and they contained the two above-mentioned sentences *a* and *b*, then the index would look as shown in Table 2.2.

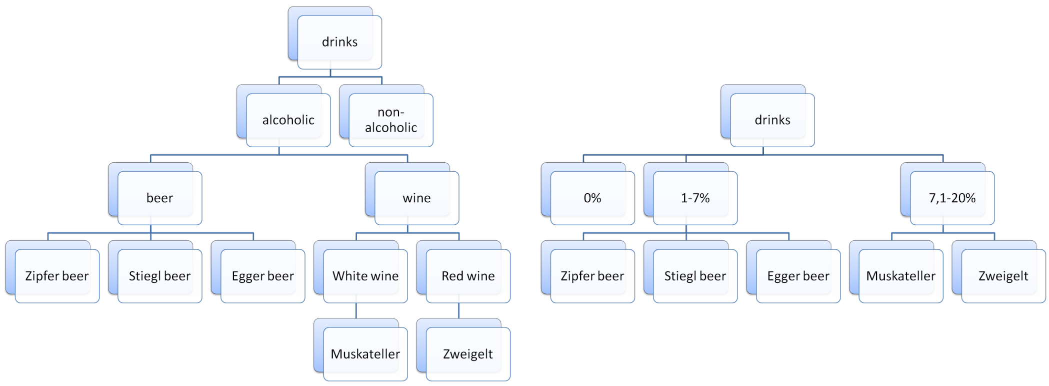
To make use of the extracted facets, the next step is a facet hierarchy. A facet hierarchy is developed for a certain environment, often by a domain

Word	**List of [document][position]**
engine	[A][1]
need	[A][2],[B][3]
3	[A][3]
second	[A][4],[B][5]
heat	[A][5],[B][2]
phase	[A][6]
1	[A][7]
minute	[A][8]
cool	[A][9]
down	[A][10]
seat	[B][1]
36	[B][4],[B][7]
reach	[B][6]
degree	[B][8]

Table 2.2.: Table of an index example

expert [Wei et al., 2013]. An advantage of a hierarchy shown to users is that it provides users with guidance toward potentially interesting subsets [Tunkelang, 2009]. Seeing different interesting subsets can lead to serendipity, which means accidental discoveries [Roberts, 1989]. As seen in the facet extraction example in Table 2.2, a word can be found in different documents, for example the word *second*. The index provides a list of occurrences of each word. The problem is that if two sentences use the same word, the word does not necessarily have to have the same meaning in both sentences. For that reason, a domain expert develops a facet hierarchy which describes the environment and defines in which context a word belongs to a category or topic in a hierarchical structure. The process is done iteratively: A facet hierarchy and its rules are developed and tested on the data or on a test-set and redefined if needed. [Prieto-Díaz, 1987, Prieto-Díaz, 1990]

Another very important aspect, as the chapter evaluation (6) shows, is the structure of the hierarchy itself. A hierarchy about drinks can be structured deep or wide, as exhibited in Figure 2.13. Figure 2.13a uses several categories



(a)Deep hierarchy example.

(b)Wide hierarchy example.

Figure 2.13.: Drinks in a hierarchy, two examples showing hierarchy possibilities.

to describe the elements, for example the element *Zweigelt* belongs to the categories red wine, wine, alcoholic and drinks. Figure 2.13b uses fewer stages of categories. The advantage is that users might not have to click that often if they go through the stages step by step starting at the top. However, a disadvantage is that the description of the elements is not that clear. A user has to know what percent of alcohol a beer typically has in order to find it directly. The actual usage and the resulting user satisfaction have to be evaluated to find an optimised hierarchy for a certain use case in a certain domain.

As the facet extraction process is applied to text documents, it can be applied to office documents, databases, wiki systems or other information tools with a connector, as well. The connector uses tool interfaces or specifications about the data storage to retrieve data and metadata for facet extraction, for instance the current owner of a file or its date of creation as it can be obtained with *Windows Explorer*. Connectors for standard tools are available, where this is not the case for many authoring tools in industrial usage.

Another aspect for improvement is the information that is possibly extractable from industrial tools such as computer aided design (CAD) software. As presented in Figure 2.14, an export of rattleCAD bicycle building software [Rosenberger, 2013], such tools are used for visual design. The

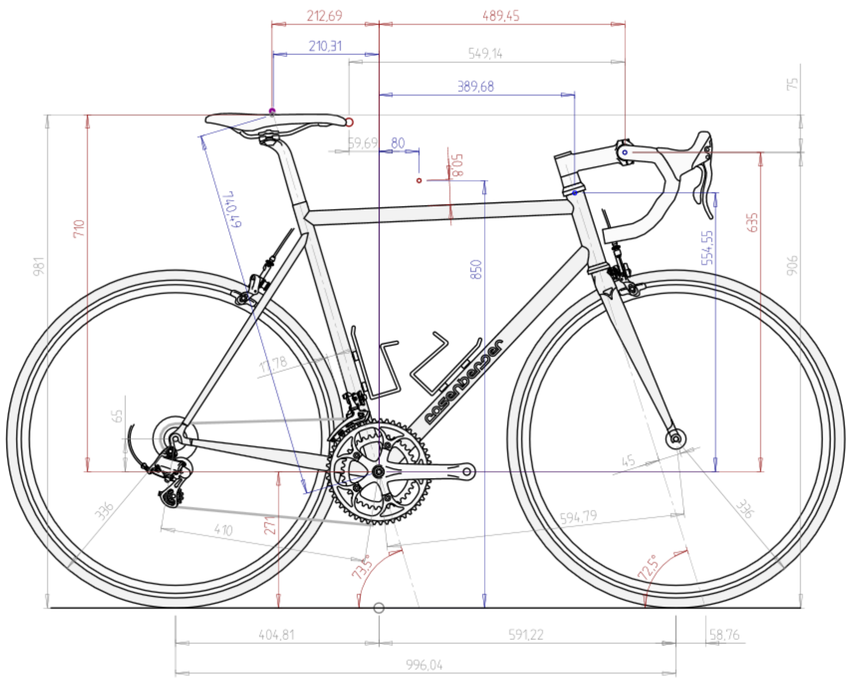


Figure 2.14.: Bicycle construction. Author: rattleCAD, source: [Rosenberger, 2013].

information of a designed object consists of coordinates and information of normalised and not normalised products used within the object, as well as meta-information added by the developer. The object itself is a complex data type, and information extraction is done via software interfaces of the authoring tool provided. The key success factor is the interface provided, as it decides how much information can be potentially retrieved. Information retrieval from files in the file system is possible, but many file formats carry binary data, non-natively interpretable by humans. It is comparable with compiled software, where it is hard to extract the source code from.

2.2.2. FACETED SEARCH

The faceted search is described in this section. It is a common technique to search within large data bases.

As remarked on in Section 2.2.1, a facet hierarchy represents topics and elements ordered in a hierarchy which are related to facets found in a specific context. The more data sources are indexed, the more topics of the facet hierarchy would have one or more elements, for example documents, database entries, wiki pages, etc. or generally spoken structured and unstructured data, which belong to the topic or are somehow related to it.

Searching for a topic shows related structured and unstructured information objects in a result list. There are topics where no facet, respecting the facets context, belongs to and are therefore empty. If the result list is filtered for an empty topic only, then the result list becomes empty as well. This is why empty topics can be removed from a facet hierarchy. For topics related to one or more results, the number of relations can give insight into the data structure and a general overview. If there are one hundred results, no filters are applied, and a facet hierarchy topic represents file extensions *.pdf* and *.doc*, then the ratio between the topic members in regard to the number of results sharing this facet can offer insight. The number of results is shown next to each facet as can be seen in Figure 2.15. The figure shows

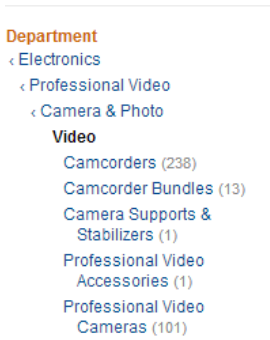


Figure 2.15.: A faceted search using a facet hierarchy. Source: [Amazon.com, 2014].

a faceted search using a facet hierarchy. The search was started without providing a keyword and the category *Electronics* was chosen. Next the category *Professional Video* was chosen followed by the category *Camera & Photo* and then the category *Video*. As seen in the figure, there are 238 *camcorders* available.

Tunkelang describes the faceted search as follows, while referencing the work of Neal [Neal, 1997] about parametric search: [Tunkelang, 2009]

> *"A parametric search interface is essentially a Boolean search interface for a faceted content collection: it allows users to formulate queries by visually specifying a set of constraints on the facet values. A query is typically an AND of ORs: values selected within a single facet are combined using a logical OR, whereas constraints associated with different facets are combined using a logical AND. The system responds to a query with the set of objects in the collection that satisfy it."*

The difference between a filter and a faceted search is the possibility of filter combinations. For example, a document which is listed in the result list can have zero, one or more facets present in the facet hierarchy. A sample hierarchy shown in Figure 2.16 represents a tree structure. In the category *product*, there are the 2 sub-categories, *module* and *project*. The ticked checkbox of the element *power train* in the category *module* means the result list is filtered to only show results sharing the facet *power train*, and the number is 368 results. Ticking the checkbox of *interior* adds all results with the facet *interior*; another 363 results are added with the logical OR conjunction. All added elements are different to the 368 elements with the facet *power train*, because the members of a topic are disjunct in the very lowest level. Adding project *C007* of topic *project* leads to a logical AND conjunction and therefore filters the result list to results which have either the facets *C007* and *power train*, or the facets *C007* and *interior*. 13 results are found. [Tunkelang, 2009]

To sum up, a faceted search restricts the results to topics of interest. The idea is that a user has a certain topic in mind when searching, firstly searches for keywords using the search field and afterwards restricts the results more and more due to chosen facets of the facet hierarchy.

The reason why this is used in enterprise searches is that ranking algorithms used in web searches like Google PageRank cannot be used for enterprise

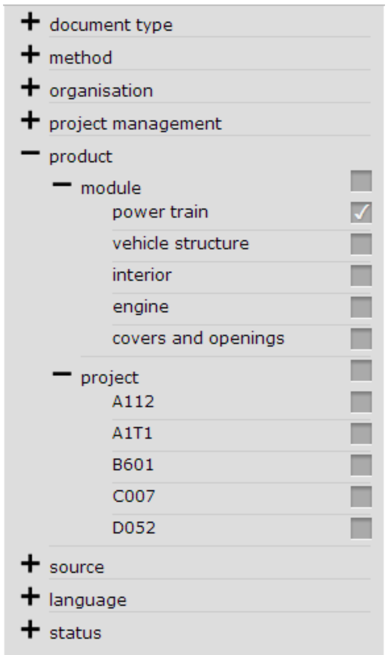


Figure 2.16.: Facet hierarchy, facet *power train* is chosen.

search due to the missing hyperlink structure despite some internal Web pages. Some ranking algorithms for enterprise search did not succeed and therefore, it is easier to let the user restrict the number of results to a small number, then scan every result and decide if it is relevant or not. [Branscombe, 2014, Hawking, 2004]

A problem of the faceted search is that at the very deepest level, a result has to share at least one of the facets of the group to not be restricted if any member is chosen, so if all *project* members are chosen, for example, then 1492 results are listed. Only 1491 of them, one less, can be found using the *module* members. 1477 results have a facet *status*. In every topic group a member *undefined* or *not identified* is missing to not lose some results with every selection.

2.2.3. DASHBOARDS AND SEARCH INTERFACES

This section introduces dashboards, information cockpits and visual faceted search interfaces, instances or a special form of graphical communication. Information is presented on a display to monitor what is going on, or information is presented on a display to create views on it, search for trends or get an overview.

DASHBOARDS

The concept of dashboards and information cockpits is similar: collect data, compress it and show parts of it or views of it in a visualisation. According to Hearst [Hearst, 2009a, Hearst, 2008, Hearst, 2006] the success of a user interface, for example with the purpose of searching, depends on its design. An illustration of a dashboard is shown in Figure 2.17; a time frame can be specified in the top right corner, according to the time frame information about spending, volume and carriers is displayed. [Few, 2013]

> *"Dashboards are among the latest new information technology (IT) waves to hit the shore in the last 10 years. ... Like all methods of communicating data graphically, they can obscure, fail to communicate, and even misrepresent information if used without awareness of those principles and practices."* [Few, 2013]

However, what turns out to be true for information cockpits as well as for dashboards, is that the "success as a medium of communication is a product of design" [Few, 2013], for example if knowledge about human perception is used. Years ago, poor design was common and users rarely saw good design examples; as a result, most people were "honestly unaware that things could be better - a lot better" [Cooper, 1999]. Technologies for data retrieval, data transformation and finally data representation, such as representing data in an applicable way for reporting, have been developed, where it is hard to derive the real value from information. In order to assist users in exploring and analysing information, an engaging interaction, using the strength of human intelligence, is needed, based on knowledge about human perception. [Few, 2013]

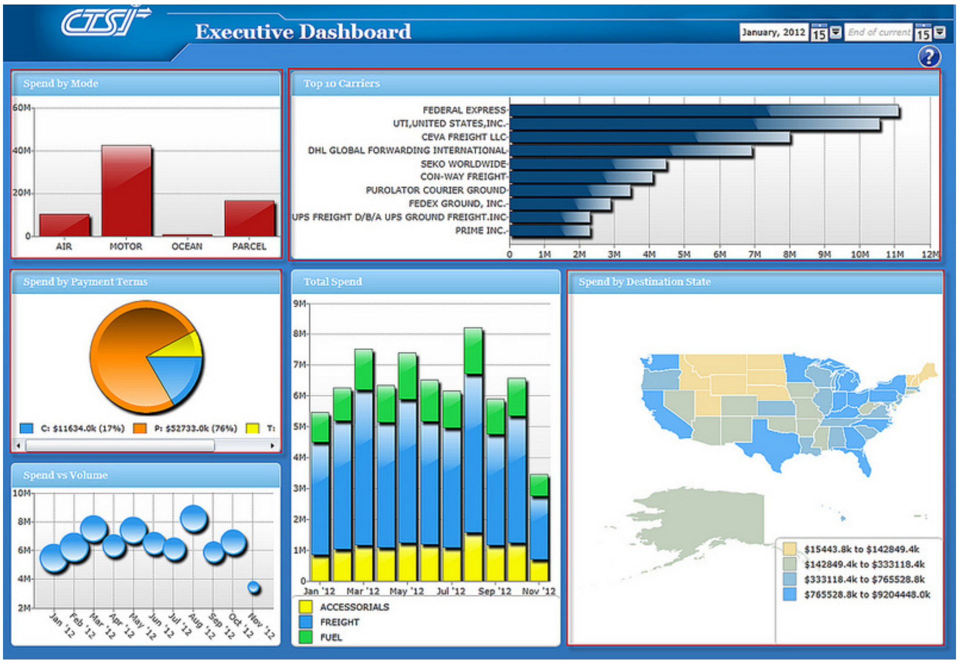


Figure 2.17.: An executive dashboard of the company CTSI-Global. Source: [CTSI-Global, 2014].

Dashboards were historically called *Executive Information Systems (EISs)* and their purpose was *"to display a handful of key financial measures through a simple interface that 'even an executive could understand'"* [Few, 2013]. Early problems were the missing data warehouses and business intelligence, which led to incomplete, unreliable data. As a result, dashboards were rarely present on the market. As time went by, key performance indicators and business intelligence technologies emerged. In parallel, *computer based displays for manufacturing control operators* entered the market, developed by software engineers instead of human computer interface designers. Common mistakes in dashboard design are known. Few mentions 13 different common mistakes, where some are applicable for information cockpit design as well: [Few, 2013]

1. EXCEEDING THE BOUNDARIES OF A SINGLE SCREEN: Information should be displayed without exceeding a single screen as it

turns out that relations seen within eye span can be perceived easier. Amongst others, a problem is that humans can keep a limited number of information chunks at a time in working memory, comparable to RAM in personal computers. Simultaneous perception enables comparisons that lead to insights.

2. SUPPLYING INADEQUATE CONTEXT FOR THE DATA: Fragmented information, for example different data sets, should be made accessible by navigation. Fragmented information a user would benefit from seeing it together should not be separated. For example the current air temperature of a city is 35°C, which could be a typical value for the city or an atypical value, depending on values of air temperature of the last years. The context provided assists the user in decision making. The best amount of context depends on the purpose and needs of the viewers.
3. DISPLAYING EXCESSIVE DETAIL OR PRECISION: The opposite extreme of the last common mistake is providing too much detail. Dashboards should display fairly high level information to provide an overview, whereas information cockpits need to provide an overview as well as precise data. For example: Should the value *1,209,867.53* be displayed exactly like that or would displaying it as *1.2M* be sufficient enough? Another example: When a document is searched for, providing the date-time value of the last change is of interest, the format could include the year, month, date, hour, minute and second. It somehow depends on the number of documents found if the values hour, minute and second are needed; however, if the values can be skipped, then a user might be faster in detecting the date difference between documents.
4. EXPRESSING MEASURES INDIRECTLY: The message that most directly supports the user needs should be used. For example, it has to be decided if absolute values or values relative to a basis are shown. The value has no meaning if the user does not know what is measured and its unit.
5. CHOOSING INAPPROPRIATE DISPLAY MEDIA: A graph is not always a better representation than a table of numbers. In unlabelled pie charts, if some parts are of about comparable size but not equal, then detecting the order is difficult, especially if the pie is shown 3-dimensionally. Bar charts with proper axis labelling should be taken

into account as they use human perceptions' ability of comparing a single dimension: *length*.

6. INTRODUCING MEANINGLESS VARIETY: If one type of chart is used heavily in a dashboard or an information cockpit, than users might get bored by monotony. The most appropriate display media still has to be used as a first choice, because users will not be bored if they get exactly the data needed. Variety in display media needs different perception strategies for interpretation and therefore energy in the changeover.
7. USING POORLY DESIGNED DISPLAY MEDIA: Things such as excessive use of bright colours, missing labelling or the use of a legend that forces a user to bounce between the graph and the legend, a missing ordering of objects, colours which are similar, overlapping objects, not placing objects to be compared next to each other, occlusion in 3-dimensional graphs, etc. are common mistakes.
8. ENCODING QUANTITATIVE DATA INACCURATELY: If a bar chart's y-axis does not start at zero while representing quantitative values, it might be misleading in value comparison due to wrong proportions. If bar a has a value of 10 and bar b has a value of 8, then b is 80 percent of a. However, if the chart's y-axis starts at 7, then it looks like b is 30 percent of a.
9. ARRANGING INFORMATION POORLY: If lots of information has to be shown at the same time, then placement has to be appropriate. Placement should be based on importance and groups should be framed rather than being placed in a fragmented way. Objects requiring interaction should be highlighted. The usage of different fonts and font sizes is critical, as it can mislead attention.
10. HIGHLIGHTING IMPORTANT INFORMATION INEFFECTIVELY OR NOT AT ALL: Looking at an information cockpit or a dashboard, a user should immediately be attracted to the most important part. If everything is prominently designed, then nothing stands out.
11. CLUTTERING THE DISPLAY WITH VISUAL EFFECTS: Useless embellishment, for example letting a dashboard or an information cockpit look like an electronic control panel or a distracting background image is not helpful when searching for information. However, chart junk makes a visualisation memorable. Unique visualisation types have significantly higher memorability scores than a bar chart. [Borkin et al., 2013]

12. MISUSING OR OVERUSING COLOUR: Colour should not be chosen by chance. Colours are perceived differently and therefore provide a different level of attraction. Colour blindness has to be taken into account.
13. DESIGNING AN UNATTRACTIVE VISUAL DISPLAY: An unattractive design may distract the user from retrieving information.

SEARCH INTERFACES

Whereas dashboards provide measured or calculated data to inform a user, a search interface has a different purpose. A search interface provides information which might be relevant for a given keyword. If a lot of possibly relevant information chunks are found, then a relevance calculation can assist in doing a proper sorting. Relevance calculation is traditionally based on several criteria, the resulting value of which is also called *Retrieval Status Value* (RSV). [Eckstein, 2011]

> *"Relevance is a multidimensional cognitive concept whose meaning is largely dependent on users' perception of information and their own information need situations."* [Schamber et al., 1990]

In 2006 Gary Marchionini presented a new conception for human information interaction and information retrieval. The following requirements were mentioned: [Marchionini, 2006]

> *"From the perspective of an active human with information needs, information skills, powerful digital library resources:*
>
> - *Systems should aim to get people closer to the information they need, especially to the meaning; that is, systems can no longer only deliver the relevant documents, but must also provide facilities for making meaning with those documents.*
> - *Systems should increase user responsibility as well as control; that is, information systems require human intellectual effort, and good effort is rewarded.*

- *Systems should have flexible architectures so they may evolve and adapt to increasingly more demanding and knowledgeable installed bases of users over time.*
- *Systems should aim to be part of information ecology of personal and shared memories and tools rather than discrete standalone services.*
- *Systems should support the entire information life cycle (from creation to preservation) rather than only the dissemination or use phase.*
- *Systems should support tuning by end users and especially by information professionals who add value to information resources.*
- *Systems should be engaging and fun to use."*

What is described by Marchionini as an approach for human information interaction and information retrieval requires interface designs going beyond a simple graphical search interface. This is what is meant by the term *information cockpit* in Section 3.2 about the developed user interaction concept for the information cockpit.

VISUAL FACETED SEARCH INTERFACES

Kienreich [Kienreich et al., 2008], Smith [Smith et al., 2006a] as well as the book of Fill [Fill, 2009] provide insights about visual support for search. Smith states that the point where a folder hierarchy for documents is sufficient to organise, keep track and search is already long past. However, many of the existing systems providing search and browse capabilities are still textual. Given ongoing explosion of gathered content enabled by automatic data *"collection mechanisms, a text-based interface may become increasingly overwhelming to users"* [Smith et al., 2006a], especially novices. Kienreich introduces visual representations for filtering search results along topical, geographic and temporal dimensions. Tvarozek [Tvarozek and Bielikova, 2008] shows the usage of novel navigation and visualisation approaches for search visualisation such as visual depiction of facets and restrictions, visual navigation and graph-like exploration of search results. Polowinski [Polowinski, 2009] states that faceted search is particularly suitable for structured but heterogeneous data with explicit semantics. [Kaiser et al., 2013]

The first example called FacetLens is an "interactive visualisation system in supporting exploration and sense-making within faceted datasets" [Lee et al., 2009]. This is illustrated in Figure 2.18; the main area shows

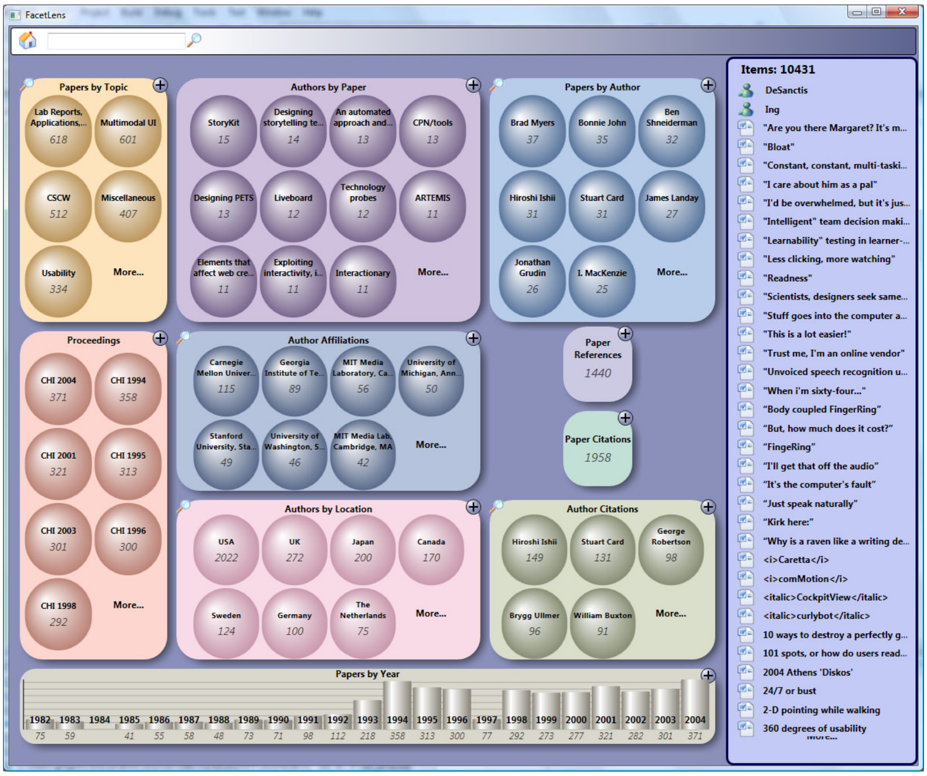


Figure 2.18.: The FacetLens visual search interface. Source: [Smith et al., 2014].

currently available facets and attribute values. Attribute values are grouped by facets and placed within a uniquely coloured box. The items area on the right shows the result set. Clicking on attribute values adds the value as filter and therefore narrows down the result set. If more than one filter is applied, then the filters narrow down the result set using a logical AND junction of all filters. Filters which do not share any results anymore after applying filters are removed. The system developed is based on the FacetMap interface, a scalable search and browse visualisation, query driven, interactive and generalisable [Smith et al., 2006b]. The FacetLens interface was evaluated running a formative usability study to identify issues. One issue was confusion when choosing a name, for example *Bob*, of available authors, because facet values for *Author Citations*, *Paper Citations* and *References* exist. Author Citations represent authors cited by Bob's papers. Paper Citations represent papers cited by Bob's papers. References represent papers that reference a paper of Bob's. Another issue also targets the author selection, as it is not clear to the user that after choosing an author, the other names there represent co-authors of the chosen author. There were tooltip problems, as well. Nevertheless, the average user satisfaction was high: 6.1 on a scale of 1 to 7 and the system was rated to be easy to learn (avg.: 5.7) and to discover trends (avg.: 6.1). [Lee et al., 2009]

Figure 2.19 shows a faceted attribute selector where available facets are

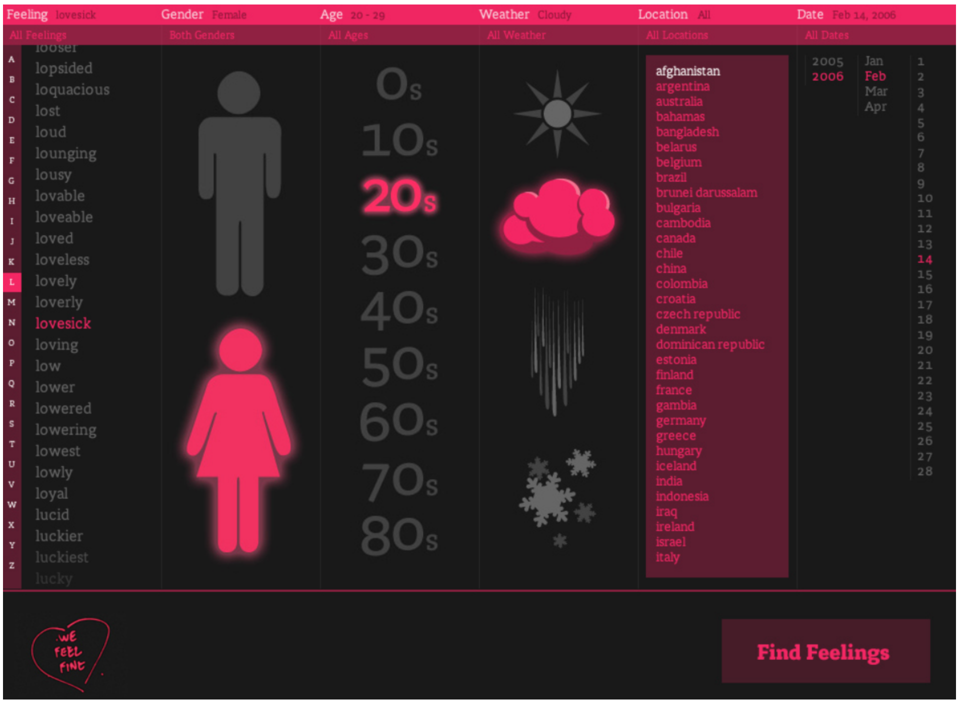


Figure 2.19.: A faceted attribute selector of wefeelfine.org. Source: [Harris and Kamvar, 2014].

shown in columns and the attribute values are either represented as symbols or as text. Hearst [Hearst, 2008] calls it fascinating, but criticises the amount of screen space needed.

Websites like the photo sharing site Flickr [Yahoo!, 2014] or APA labs [Kienreich et al., 2008], a web-based platform for retrieval and analysis of news articles, use tag clouds for faceted search. In case of Flickr, users may add tags to photos to describe them. The words used most for a topic are calculated and shown. The more often a word appears, the more important it is, and in a tag cloud *"usually the tag importance is depicted by font size"* [Kienreich et al., 2008]. A minimum number of appearances and a list of words which should not be included assist in creating a meaningful tag cloud. An example is shown in Figure 2.20a, a tag cloud about tag clouds. According to the image, the words *presentation*, *used*, *words*, *clouds* and *frequently* are representative for tag clouds. Social tagging of collections enables flat classification schemes. However, a negative side is that tagging classification schemes ".. are hard to present to users. Therefore, they are usually only used to aid keyword search rather than to help users interactively browse through documents, .." [Wilson et al., 2010].

The WebTOC system [Nation et al., 1997] represents "the structure and number of documents" [Hearst, 2009b] of subhierarchies by an "embedded coloured histogram" [Wilson et al., 2010] as displayed in Figure 2.20b.

The example in Figure 2.21a uses a tag cloud and statistical graphs to inform the user which and how many elements are available in the filterable areas. Filtering is done using sliders.

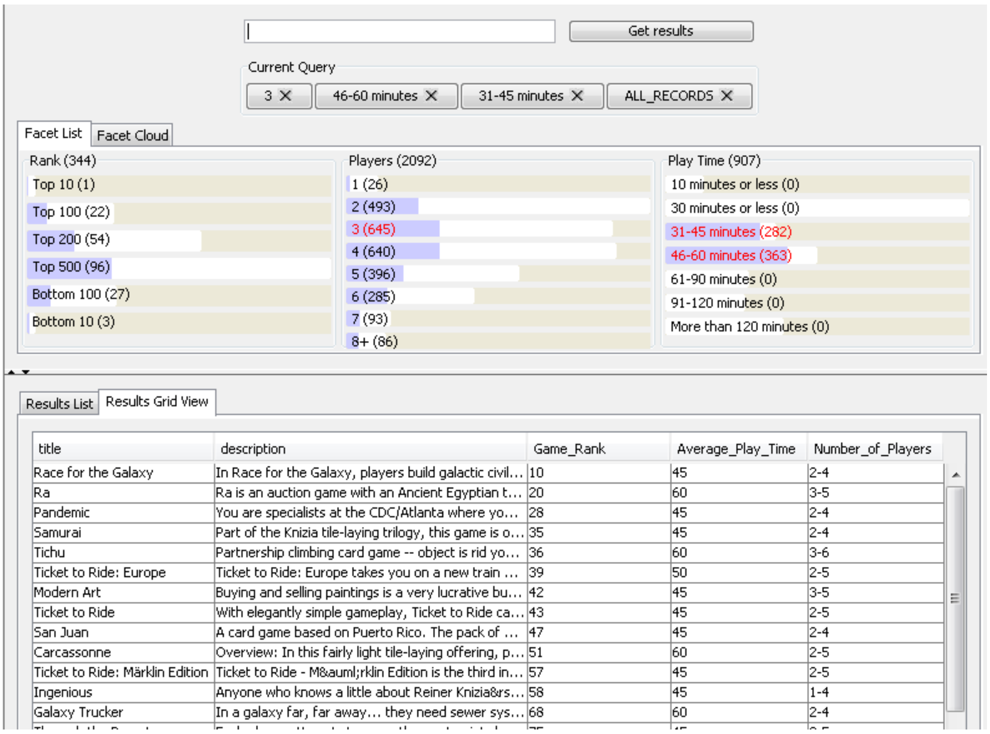
The last example in this section is the *Relation Browser* version seven, developed from 2006 to 2009, exhibited in Figure 2.21b. A notable feature is the query preview strategy. "Graphical representations behind each item in each facet show how many documents can be found by selecting it. When a user hovers over any item in any facet, the size of the bar in the graphical representations is reduced to indicate how many documents will remain under each annotation .." [Wilson et al., 2010].

(a)A tag cloud about tag clouds. Source: [Bilinsky, 2014].

(b)A faceted attribute selector called WebTOC. Source: [Marchionini et al., 2014].

Figure 2.20.: A tag cloud and WebTOC, a visual search interface example.

(a)A faceted attribute selector of viewshare.org. Source: [The Library of Congress, 2014].



(b)The Relation Browser in version seven. Source: [Marchionini and Capra, 2014].

Figure 2.21.: Visual search interface examples viewshare.org and RelationBrowser.

3. CONCEPT DESIGN OF THE INFORMATION COCKPIT

Chapter 3 gives insight into the design phase of the developed user interaction concept of the information cockpit. Subsection 3.1 introduces design methods. Subsection 3.2 describes the information cockpit concept of this thesis in detail. Finally Subsection 3.3 discusses and concludes the chapter.

3.1. DESIGN METHODOLOGY

User interface designers face the task of understanding other disciplines and therefore cooperating with application domain experts. The job often lacks a terminology, a common language, to exchange ideas and guidelines [Borchers, 2001]. Design methods try to assist the designer.

In the following, interface design methods are described. Dix pointed out that *"probably 90% of the value of any interface design technique is that it forces the designer to remember that someone (and in particular someone else) will use the system under construction"*[Dix et al., 1998].

Nielsen defined eleven activities for a usability engineering life cycle model in 1993, which are described in the following list: [Nielsen, 1993]

1. *Know the user.* Only the targeted end user can provide insights to needs. Often the designer is not allowed to talk to the targeted end user directly, as was the case in this thesis.

2. *Competitive analysis.* Analyse existing products according to established usability guidelines. Existing products can be used for empirical tests. As prototyping is a commonly used in iterative development, an existing product is often the best prototype available for a new product. Existing user interfaces and visualisation libraries were consulted for this concept.
3. *Setting usability goals.* Usability is a multi-dimensional attribute; however, attributes may conflict.
4. *Parallel design.* In parallel design, at least two designers start to design the interface in parallel and independently. This was not the case in this thesis.
5. *Participatory design.* Users may not be able to explain what they want or do not know what is possible. Participatory design aims to include the user in the design process in a proper way, for example by providing prototypes or paper mockups in the early stage. In this thesis paper mockups were a starting point and agile development methods assisted an iterative implementation process.
6. *Coordinated design of the total interface.* "Consistency is one of the most important usability characteristics. Consistency should apply across the different media which form the total user interface." [Nielsen, 1993]
7. *Apply guidelines and heuristic analysis.* Principles of user interface design should be applied; therefore, guidelines should be provided. This was not the case in this thesis, as the team was rather small.
8. *Prototyping.* Evaluating prototypes early in the design and implementation process helps to find issues sooner. It is stated that this process is faster and cheaper. This thesis provides the first prototype and has been evaluated. Findings will influence the next prototype.
9. *Empirical testing.* Practical experience can test how targeted users use, perform and like the system.
10. *Iterative design.* Findings of the evaluation will be targeted in the design iteration.
11. *Collect feedback from field use.* Short-term user evaluation can give insights for the development. Collecting data from long-term field use shows how users actually use the system. This will not be done with the implementation of this thesis, but perhaps with a future version.

As a fact, human computer interaction (HCI) research, hereby the willingness to use inspirations in the product development process is meant, struggles to integrate design. Academic institutions attempted to do so; however, despite excitement until 2007, no agreed model existed which teamed up designers and HCI researchers to contribute. An added design perspective would assist HCI research. [Zimmerman et al., 2007]

> *".. many HCI researchers commonly view design as providing surface structure or decoration."* [Zimmerman et al., 2007]

HCI designers claim that in history *"they were often brought in at the end of the process and asked to make the interface 'pretty'."* [Zimmerman et al., 2007]

Designers themselves mention three advantages when asked about what they can add in interviews with Zimmermann et al.: [Zimmerman et al., 2007]

- Designers take a holistic process to solve under-constraint problems. An under-constraint problem could be if a developer invested a lot of time finishing a component. If then, for design reasons, the component does not fit then the developer has a personal constraint problem.
- Designers can integrate ideas from art, design, science and engineering into an aesthetic looking interface.
- Designers have empathy for users.

As a result, a user-centred design process evolved. The design process, comparable to the software development process, reacts to increased usability requirements and changes the process to be user-centred. The ISO standard 9241-210 mentions the following requirements for human-centred design: [ISO 9241-210, 2009]

> *"Project planning shall allocate time and resources for the human-centred activities. This shall include time for iteration and the incorporation of user feedback, and for evaluating whether the design solution satisfies the user requirements. Relevant user and stakeholder groups shall be identified and their relationship with the proposed development described in terms of key goals and constraints. There are four linked human-centred design activities that shall take place during the design of any interactive system:*
>
> *a Understand and specify the context of use;*
> *b Specify the user requirements;*

c *Produce design solutions;*
d *Evaluate."*

The following figure (3.1) visualises the workflow of the process. As soon as the need is identified, the iteration starts. At first, the usage context has to be understood and described. Secondly, requirements have to be specified. Thirdly, a solution has to be designed. In the fourth step, the solution has to be evaluated against the defined requirements. If the requirements are satisfied, then a solution is found. If not, then the next iteration starts, depending on the problem in step one, two or three.

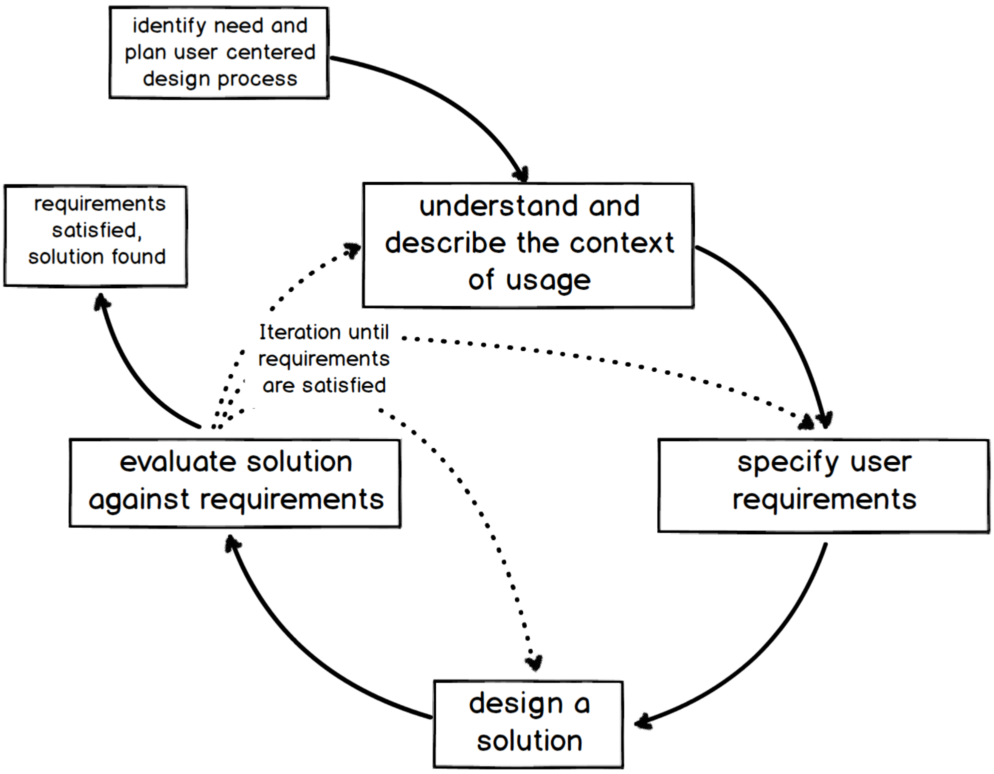


Figure 3.1.: The user-centred design process. Compare [ISO 9241-210, 2009].

Requirements may be unequally important for the satisfaction of user and stakeholder. The model of Kano as shown in Figure 3.2 divides the satisfaction of product criteria into three categories: [Pohl and Rupp, 2010]

- Basis factors are system criteria taken for granted and not explicitly mentioned. If all basis requirements are fulfilled, then the demander is not unhappy. For example, a basis factor of a search interface is a keyword search.
- Power factors are system criteria explicitly demanded. Fulfilled power factor requirements lead to satisfaction, e.g. if a bookmark section is demanded and provided.
- Enthusiasm factors are system criteria not demanded but appreciated during usage, e.g. providing a screenshot of a document as an overview in a search interface result list.

Over time, enthusiasm factors become power factors and power factors become basis factors due to routine.

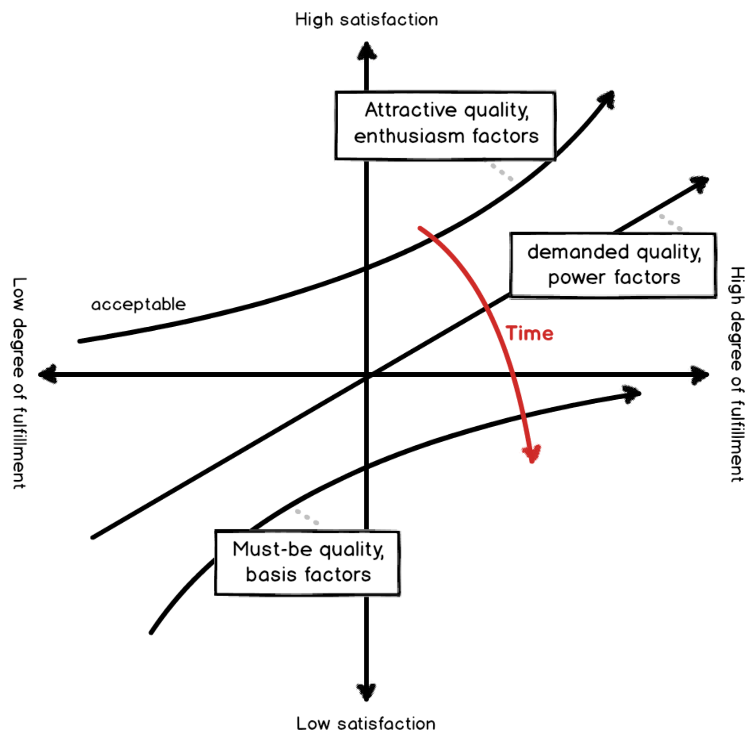


Figure 3.2.: The Kano model. Compare [Pohl and Rupp, 2010].

3.2. USER INTERACTION CONCEPT FOR THE INFORMATION COCKPIT

One main aspect of this thesis is the development of the user interaction concept in an enterprise search environment using visualisations as described in the following.

What is important is that people prefer to be allowed to navigate and filter along multiple metadata fields, which provides them with a sense of control over the data.

3.2.1. REQUIREMENTS

In the past, expert interviews with simulation managers showed a demand for enterprise-search improvement in the IT systems of AUDI, BMW, DaimlerChrysler, Deutsche Bahn, EADS, Ford, Iveco Magirus, Keiper, Kieckert, Porsche and VW. For example the time for retrieving all information needed for a calculation in simulation data management takes on average 51% of the worker's time. [Krastel, 2001]

As this thesis is part of an industry project, the goal is, amongst others, to develop a strategy and a software prototype which is the basis for further measurements in cross-linking, search and analysis of information artefacts from heterogeneous, structured and unstructured information sources within a company. [Kaiser et al., 2013]

The following list summarises improvable characteristics of enterprise search systems relevant for this thesis:

- SYSTEM SOURCE: An enterprise search engine should be able to indicate various heterogeneous data sources, for instance enterprise file systems (semi or unstructured data) together with enterprise databases (structured data). The concept of this prototype focuses on enterprise file systems. Future prototypes will focus on structured data sources as well, e.g. wikis.

- RECOMMENDER SYSTEMS: Provides information automatically based on context, user profile, etc. This will be a component of a future prototype.
- VISUALISATIONS: Visualisations which envision results can highlight attributes to provide an overview. With user interaction, different views can be created. Using visualisations can make life or work easier because it may be fun. This is addressed by the faceted search interface and a visualisation of result relations.
- EXPLORING AND NAVIGATION FUNCTIONS: Imagine that a new employee might not know what to search for and therefore would like to explore what is available. When useful information is found, a user might want to know more about the author and therefore navigate to information about the author or furthermore to colleagues of the author. If all directly connected things of results are shown, then navigating through the system can provide helpful insights. Exploring is provided by the faceted search interface. Without using any keyword, the set of results can be drilled down to see what results are available. Navigation through the data is not a focus of this prototype.
- ACCESS RESTRICTIONS: Frequently, a user searches for something but cannot find it owing to access right restrictions. One solution might be to inform the user that there is something, for example by adding contact details of the original author. This will be a component of future prototypes.
- WEB-BASED USER INTERFACE The targeted engineers have restrictions as to what software is allowed to be installed, which could lead to the problem that a proof of concept prototype is possibly not allowed to run in the user's environment. However, state of the art functionality and design is expected. This was solved by providing an executable software package where all necessary components were included, such as a portable version of the browser.

3.2.2. USER ANALYSIS

Due to restrictions, designers may not have the possibility to talk to end users directly, which is what happened to the author. For the purpose of this thesis, the author was granted access to interview protocols. In a second interview round, details were clarified together with the interviewers. The targeted user is an engineer in the vehicle manufacturing industry with the task of doing calculations for simulations. The user might be of any age between 18 and 65. As mentioned, there is a need for assistance in the information retrieval process, as this takes up more than 50% of the total working time. [Kaiser et al., 2013]

3.2.3. THE CONCEPT MOCKUP AND THE COMPONENT DESCRIPTIONS

The concept provided in the following subsection was presented to the public in a concept paper by the author [Kaiser et al., 2013].

At this point, it is assumed that the data is prepared and searchable through the representational state transfer (short: *REST*) application programming interface (short: *API*); the API separates the web-based interface at the client from the search engine on the server.

The next and most important step is the design (and later implementation) of the graphical user interface (short: *GUI*). There are two design guidelines that were followed:

- Ensure usability, a key success factor in regard to user acceptance.
- Provide different alternatives for the faceted search and the result view to assist different user types.
- The proposed usage of the information cockpit should be self-descriptive, clear and intuitive.

The mockup of the information cockpit design is shown in Figure 3.3 and described in detail in the following.

In the rest of this section, a motivation and discussion of design decisions is provided.

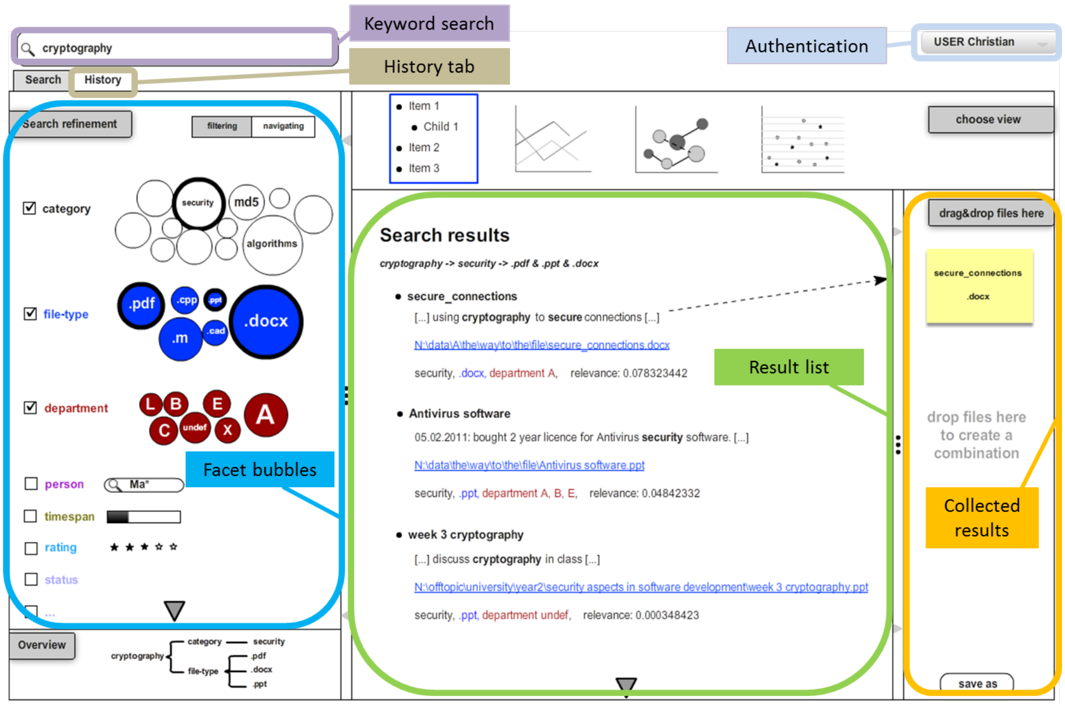


Figure 3.3.: A mockup of the GUI.

KEYWORD SEARCH

As the main goal of the GUI is to provide a search functionality through a graphical interface, a text field at the very top is provided. The user can type one or more keywords into the text field and submit the search query to the search engine pressing the *search* button. A visualisation acting as a faceted search interface, which is discussed in the next subsection, can be used to filter the result set according to the user's criteria.

FACETED SEARCH INTERFACE

A search query may retrieve a high number of potential results. A possibility to narrow down and optimise the results is provided by a faceted search interface (on the left in 3.3), which is a well-known, proven exploration and discovery technique [Hearst, 2008]. The intent is to use the faceted search interface in addition to a keyword search. It is possible to start with either exploration via the faceted search interface or by providing a keyword. When the user clicks on a facet, a new search query is generated and sent to the web service of the search engine, which filters the result set with regard to the chosen facets. Using any number of different facets for filtering is possible, but it might lead to zero results due to the junction of filters as explained in Section 2.2.2. The facet categories, like *file type* and *department*, rely on the data indexed previously by the search engine. The facet hierarchy is defined by a domain expert and the search engine configuration ensures that the facets are extractable from the results, e.g. the author of a document.

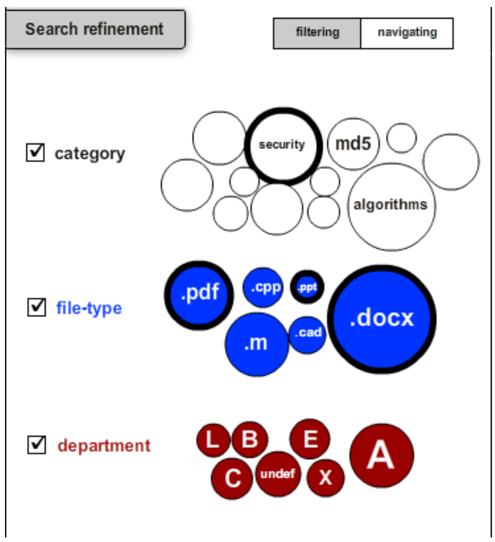


Figure 3.4.: A mockup of the facet bubbles.

The bubbles represent the identified facet possibilities of each category as can be seen in Figure 3.4. The bigger a bubble is, the more found results share this facet. A minimum bubble size is defined to increase readability of the descriptive text which is inserted into the bubble. Bubbles share a colour if they are members of the same category. Alternative representations are used for non-textual data, for example a star representation for *rating* or a timeline (or a progress bar) for temporal information.

RESULT VISUALISATION

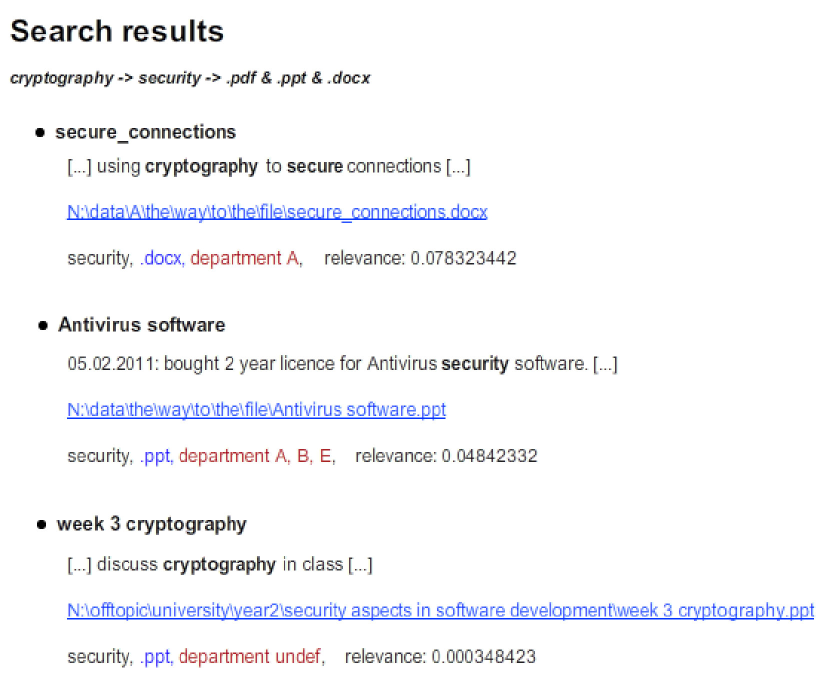


Figure 3.5.: A mockup of the result list.

Whenever the search query changes, either because of typing in a new keyword or selecting facets, a new query is sent to the API. The response is displayed in the result box in the centre. By default, results are shown in a list as shown in Figure 3.5, as this is the most commonly used and widely known format of presenting them. Above the result list, a palette with available possibilities for visualising the results is provided. By default, the list view is chosen, with a blue rectangle around the list button showing that this view is in use at the moment.

COLLECTING AND SAVING RESULTS

A user might not be interested in only one single search result, but in a current status of a group of files, e.g. when collecting all necessary data for

Figure 3.6.: A mockup of the collecting and saving results area.

a milestone report. For this purpose, a so-called *collection box* (implemented as bookmark section, as they are not saved permanently in the portable version) is integrated on the right side of the GUI as visualised in Figure 3.6. The box content is persistent and does not change with new search queries, allowing the user to collect results over time. Adding hits from the search result list to the collection works via drag and drop. The user can click on the *save as* button and provide a destination directory where the collected results are then saved.

HISTORY VIEW

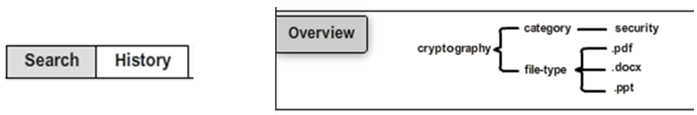


Figure 3.7.: A mockup of the history tab to open the history view as well as the overview.

While a user has the possibility to save the current status of results as discussed in the last section, an overview of all previously used search queries might be needed as well, for example to execute the same search again by clicking on the saved query instead of retyping the keywords and reselecting the facets. The view of the GUI can be changed by clicking on the *History* tab below the text field as can be seen on the left in Figure 3.7, where the used search queries are represented as trees. The root element is the keyword used in the query or *no keyword* if no keyword was used. The child elements are all the facets where an element was chosen. For example, if only the file extensions *.m* and *.pdf* have been chosen, the child element of the keyword would be *file-extension* and again the child elements would be *.pdf* and *.m*. If the user clicks on a search tree listed in the History tab, the query will be opened in the search tab. The *Overview* box on the bottom left shows the current search tree (which is also saved in the *History* tab) as can be seen on the right in Figure 3.7.

AUTHENTICATION

The approach described in this thesis works with unstructured data, e.g. files of a network drive, and has to respect access permissions. In the use case, the information cockpit assists the user in finding a solution to a problem. If files are used to save knowledge to a hard drive, authentication and authorisation systems provide access right handling. Predefined users and groups either have permission to read/write/execute or not. A simple way to do that is to force users to login with an account to authenticate. An open question is if a user without access rights to a file should be able to see at least some information, for example the author of the file where the solution can be found. A hint regarding whom to ask would improve the problem solution finding process immensely.

In the user interface the restricted result could be presented as the others, but it should not be possible to open, for instance, the document. If so, the remaining problem is that a user might be able to retrieve all critical information out of the metadata, e.g. the snippet, and can therefore get insights he should not get. As a conclusion, which metadata attributes to show has to be elaborated on. As access right handling is not in scope of this prototype, it is not elaborated on in this thesis.

3.3. DISCUSSION

The concept bears the usage of state of the art visualisations for a faceted search and for the result view in mind. Visualisations can use advantages of human perception and may therefore lead to an increase in performance as pointed out in Sections 2.1.3 and 2.2.3. This answers the research question regarding on how visualisations can add an advantage within the context of enterprise search interfaces. Result lists are known from popular online search engines; hence, users might expect result lists as a basis factor, explained by the Kano model in Figure 3.2. The faceted search interface using a visualisation which shows every facet as bubble with a size regarding the number of results sharing this facet might not be the best solution for every engineer, as people are different in their ability to use visual attributes; for example colour blind people could have problems in this case [Velez et al., 2005]. For this reason, a second alternative was designed, using text in a hierarchy. Due to the faceted search interfaces used to drill down result sets and the result views, together with the possibility of storing and exporting results, a user can get assistance for the time-consuming information retrieval process.

4. PROOF OF CONCEPT IMPLEMENTATION

This chapter presents the proof of concept implementation of the concept provided in Chapter 3. Starting with a listing of requirements, the system architecture and the process method are described, followed by a detailed description of the information cockpit prototype components.

4.1. REQUIREMENTS

Based on the discussion in the previous chapter the requirements for the technical solution are derived, these are listed in Tables 4.1, 4.2 and 4.3, representing technical, general and user-interface requirements:

Technical requirements:

ID	Title	Description / Comment
1	Access right handling and personalisation	Should be possible or at least not made impossible by the solution chosen.
2	Backend interface	The information cockpit will later be connected to a real search engine, which is currently not available. An API has to be developed, to make a future connection possible.
3	Result shareability	Queries and results should be shareable.
4	Expandability of the implementation	The information cockpit should be expandable, for example by adding further visualisations in the result view.
5	Search history	A history view of the latest searches should be provided.
6	Result reproducibility	Queries and results should be bookmarkable and reproducible.

Table 4.1.: Table of technical requirements.

General requirements:

ID	Title	Description / Comment
6	Web solution	Web solutions have the advantage that the end user does not have to install software beside the software needed for other websites, e.g. Java, JavaScript or a browser.
7	Dynamic structure	Structures may differ for the users, for example the facet hierarchy. Therefore the GUI should build up those parts dynamically.

Table 4.2.: Table of general requirements.

User interface requirements:

ID	Title	Description / Comment
8	Search interface	Provide a classical search interface with listed results sorted by relevance, date, etc.
9	Explorational drill down	Provide a faceted search interface for exploration and filtering.
10	Visualisations	State of the art visualisations are often available in a library. It should be possible to bind several different libraries and use them at the same time independently. Artwork should be included, for example icons.
11	Navigation	If possible, provide means for navigation through the data.
12	Information source type identification	A user should be able to identify if the a result's information source is structured or unstructured.
13	Rating	There should be a possibility to rate results or sources individually to influence the relevance.

Table 4.3.: Table of user interface requirements.

4.2. ARCHITECTURE

Based on the requirements, research in state-of-the-art technologies and visualisations has been done. A Python web framework called *django* [Django Software Foundation, 2014c] has been selected as a standard solution for the personalised graphical user interface. Modern web frameworks use a model-view-controller (short: *MVC*) software design pattern to keep the data storage and the design of the GUI separated and independent. Django employs a variation of the MVC concept, the so-called model-view-template (short: *MVT*) pattern as shown in Figure 4.1, which divides the parts into the following: [Django Software Foundation, 2014b, Croft, 2007]

- THE MODEL LAYER: An abstraction layer for structuring and manipulating the data of a web application. It describes the data structure.

- THE VIEW LAYER: A concept of *views* to encapsulate the logic responsible for processing a user request and for returning the response. It determines what data is to be displayed, retrieves it from the database, and passes it off to the template. It is possible to have a particular view for each *type* of page. An example showing a *blog* could have several views - one for the landing page displaying a list of all posts and one for a monthly archive.
- THE TEMPLATE LAYER: The template layer provides a designer-friendly syntax for rendering the information to be presented to the user. Typically the template is an HTML formatted page. The template receives a context from the view layer. Its job is to insert the context into an HTML formatted page by following defined rules.

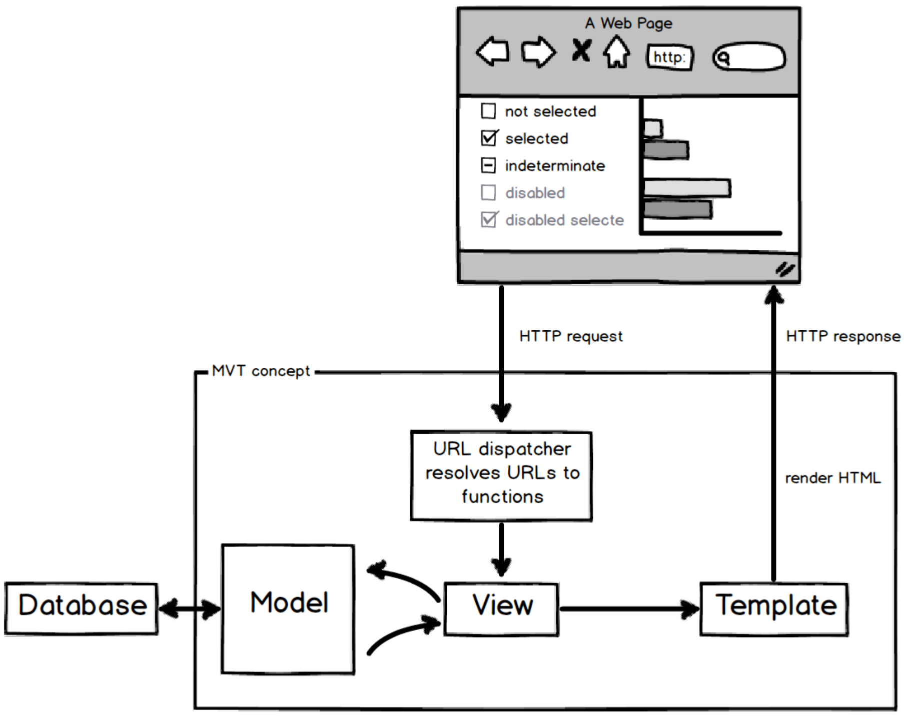


Figure 4.1.: The Model-View-Template concept. Compare [Cloudera, 2014].

The MVT concept is especially designed for the reuse of code. Libraries for authentication and user handling exist and are ready to use. Django has a large community as can be seen in the graph in Figure 4.2 showing the number of contributors that commit to the master development branch, excluding commits which aim to merge different versions. A large community provides support for development, problem solving, bug fixing and assistance in case of help.

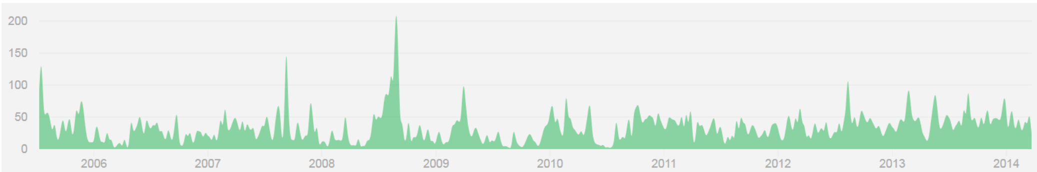


Figure 4.2.: A graph about the number of contributors that commit to the django master development branch, excluding merge commits. Image screenshot from [Django Software Foundation, 2014a].

Django supports all typically used web page development techniques. For data manipulation in the developed GUI, JavaScript is in use on the client side, due to the fact that most visualisation libraries considered are based on JavaScript. For the layout and the structure of the GUI, Cascading Style Sheets (CSS) together with Hypertext Markup Language (HTML), standardised core technologies in web design are in use [The World Wide Web Consortium (W3C), 2014]. Another JavaScript library in consideration is jQuery, as it provides easy to use Document Object Model (DOM) navigation and manipulation functions [jQuery Foundation, 2014]. Drag and drop functionality is realised with HTML5 and CSS3, the latest versions of HTML and CSS, standards by W3C. These chosen technologies fulfil the requirement concerning expandability (requirement ID = 4, see Table 4.1) as well as the requirement about a web solution (requirement ID = 6, see Table 4.2).

The JavaScript library in use is called Data Driven Documents (D3) and provides functionality for web visualisations, for everything from drawing objects, selecting objects, making transitions (e.g. size), scaling, positioning elements, to creating Scalable Vector Graphics. Taking values for position and size as input, it can produce graphical elements like circles and rectangles. The D3 website provides examples how to use the library or how it was used to produce visualisations of data including code examples.

Existing cases can assist in development. In the chord diagram, which is used for result visualisation, parts could be reused from existing examples, whereas on the oval bubble faceted search visualisation Vallandingham's gates example [Vallandingham, 2012d] was used as the starting point for development. [Bostock, 2013]

4.3. AGILE DEVELOPMENT METHODS

When the implementation of the developed information cockpit concept described in Subsection 3.2 was started, an unforeseeable tight deadline for the first prototype emerged. A development team was formed to achieve the goal. The author of this thesis initially led the team and therefore planned and managed the development. Later, the managing role was outsourced as the author implemented the visualisations. Introduced by the author, Scrumban, a mixture of two agile development methods, was utilised to manage the process.

The upcoming section introduces common agile development methods such as Scrum and Kanban. The author of the thesis had an initial team leadership for three developers to start the implementation. Setting up an implementation concept, the plan and the environment were tasks. A mixture of the two agile methods called Scrumban was in use to assist in the proof of concept implementation. The implementation benefited from using the method as follows:

- Daily stand-up meetings of the team members helped to collaborate.
- Flexibility in task processing to achieve a goal.
- The Scrumban board provided an overview and was used for task planning.

Agile development methods have become common in the past seven years, as a questionnaire in Switzerland revealed in 2012. It states that 57% of 140 companies asked use agile methods in development. 51% of them use *Scrum* and 5% use *Kanban*. When IT professionals were asked, 68% out of 194 answered that they use agile methods. The study states that non-agile working companies and IT professionals are less satisfied with their own progress than the companies and IT professionals using agile methods. [Kropp and Meier, 2012]

In Subsections 4.3.1 - 4.3.4, the topics agile development and the methods Scrum, Kanban and Scrumban are described briefly.

4.3.1. AGILE DEVELOPMENT

Agile development methods gained popularity in use by developers in the years 1999 to 2003. One cause for the revolution was the "crushing weight of corporate bureaucracy" ... "and the dehumanizing effects of detailed plan-driven development" [Boehm, 2002]. Agile development methods are more agile, as the name reveals, as the methods "address the challenge of an unpredictable world" [Dybå and Dingsøyr, 2009] " by relying on the tacit knowledge embodied in the team" [Boehm, 2002]. Agile method benefits appear amongst others in customer collaboration, problem solution processes, mutual learning and work definition for developers, whereas good interpersonal skills is an important character trait for team members and in practice a focus on human and social factors is necessary [Dybå and Dingsøyr, 2009]. The following *Manifesto for Agile Software Development* describes what is emphasised.

> *"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:*
>
> - *Individuals and interactions over processes and tools*
> - *Working software over comprehensive documentation*
> - *Customer collaboration over contract negotiation*
> - *Responding to change over following a plan*
>
> *That is, while there is value in the items on the right, we value the items on the left more."* [Beck et al., 2014]

4.3.2. SCRUM

A small (around 5 people), self-organised team with different skills gets a task to solve. The task is divided into concrete deliverable packets. The Scrum process can be seen in Figure 4.3. The list of deliverable packets is sorted by their priority and the team estimates the time for implementation of every packet. An iteration time frame is defined before the project start and should not be changed. After each iteration, a deliverable product is provided. Respecting the constraints of iteration time, available man-power and packets with rough implementation times, packets from the list are chosen to be solved within in the next iteration during the planning meeting. A Scrum board is used to monitor which packages are in which

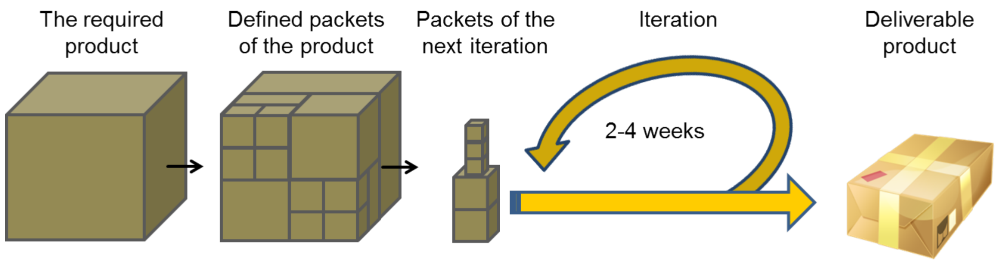


Figure 4.3.: The Scrum process. Compare [Mountain Goat Software, 2014].

state. The board is emptied after each iteration. Scrum limits the work in progress along the defined time frame. Priorities may change over time and are discussed with the stakeholder after each iteration. The foreseen review after each iteration may help to improve the implementation process. [Kniberg and Skarin, 2010]

4.3.3. KANBAN

Kanban visualises the work. The work is divided into small packages and every package is written onto a gummed label. The gummed label is placed on a Kanban board, basically a big enough place divided into columns representing process states the packages are running through from left to right, as illustrated in Figure 4.4. Example columns are *queued*, *started*, *ready for review or test* and *done*. The gummed label of a package should pass all states over time. The number of packages progressed simultaneously in one column is limited. Therefore, Kanban limits the work in progress along the working process. [Kniberg and Skarin, 2010]

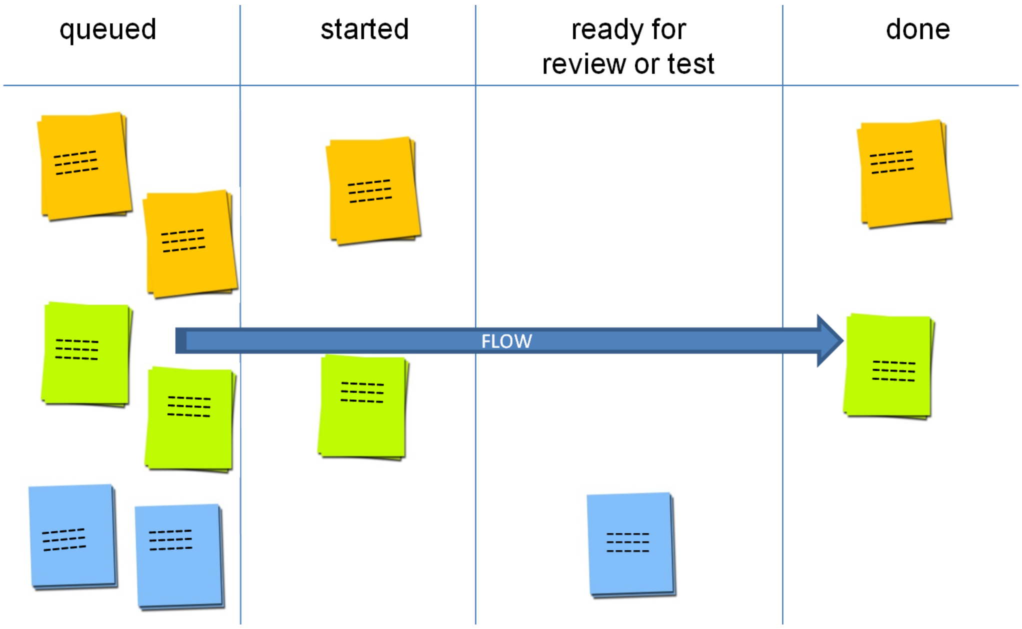


Figure 4.4.: The Kanban board. Compare [Kniberg and Skarin, 2010].

4.3.4. SCRUMBAN

As Scrum and Kanban have things in common, a mixture of the methods exists. Both methods use a board to monitor the status of packages in progress. Differences between the methods exist in the member roles. Both methods limit the work in progress and use a pull principle; with the board they use transparency to make improvements possible. Scrum prioritises the packages, whereas in Kanban, self-defined rules may exist as to which package to take next. [Kniberg and Skarin, 2010]

For the implementation of the concept, Scrumban was chosen. The time was divided into sprints, each about one week long. At first, work packages which should be done by the next sprint were planned. The estimated implementation time helped to find a maximum of packages in the queue for the sprint. The Scrumban board was an essential part of the implementation, which is displayed in Figure 4.5. The left column area contains tasks which are in queue for future weeks. They are placed there as a reminder. The second column holds tasks which have to be done in the current week not started yet. The third column contains tasks which are currently in progress. The fourth column represents finished tasks. Different colour of the post it notes indicate task categories, such as yellow for GUI development. To

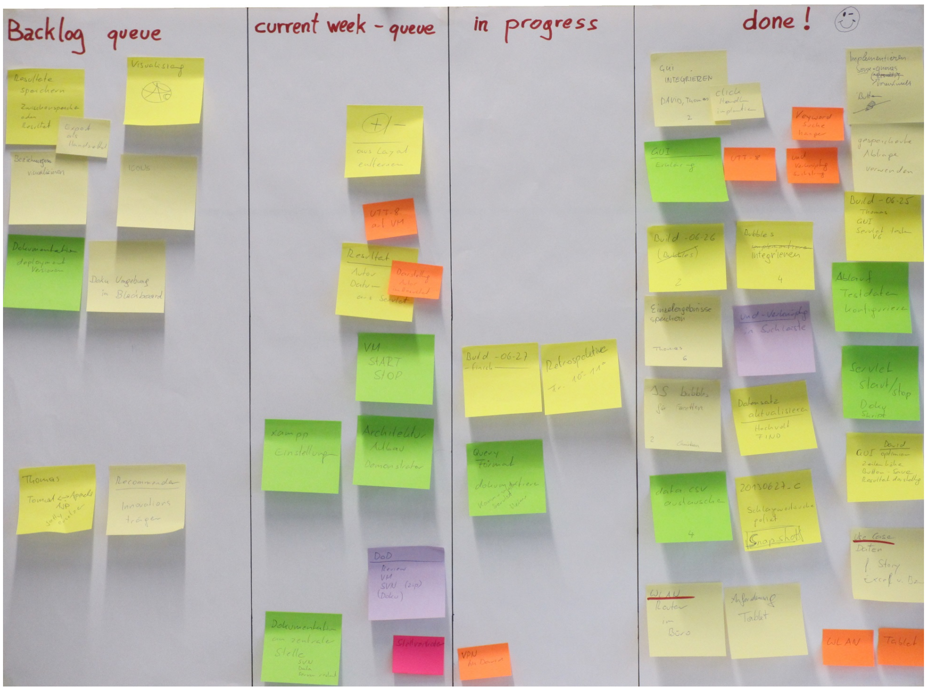


Figure 4.5.: The Scrumban board of the implementation.

be as agile as possible, daily stand-up meetings shorter than half an hour at the board were used to discuss new insights which could possibly lead to changes in work packages. After the initial leadership of the author, the managing role was outsourced to another person, who then led the implementation team of three to four people and kept the discussion with the stakeholders running.

4.4. THE INFORMATION COCKPIT PROTOTYPE

This section describes the implementation of the concept design.

The architecture of the implementation is shown in Figure 4.6. On the left side of the figure, the input for the Java enterprise search engine emulator is mentioned. Above the web servers, the web information cockpit is shown; user interaction leads to HTTP requests to the enterprise search engine emulator and to the django Python web framework, both running on web servers. Django's functionality is to bookmark queries and results and it is planned that the framework will be used for user authentication in a future version.

A comma separated value (CSV) file can be used as input for the result generator. The facets the result generator uses to generate results are loaded with the CSV file. Out of the configuration file, a result generator creates a CSV file containing all results available. This result CSV file is the input for the enterprise search engine emulator. The emulator receives the keywords the user had typed into the GUI and returns those results where the keyword is included. If facets are chosen by the user, then a result is only included in the response if it shares the facet. The search engine emulator runs on a *jetty* web server. Queries sent to the emulator are automatically sent to the django web framework to bookmark the query among the recent queries. The purpose of bookmarking is to implement the saving functionality. However, as the system is deployed as a portable version, results and queries are bookmarked instead of permanently stored. This fulfils the requirement regarding a search history (ID = 5). Also single results and the query which was used for the current set of results can be sent to django to bookmark them, by user input: ticking the checkbox of a result or clicking the *save query* button. This fulfils the requirement about result reproducibility (ID

= 3) as bookmarked results can be reproduced. Shareability has not been implemented yet.

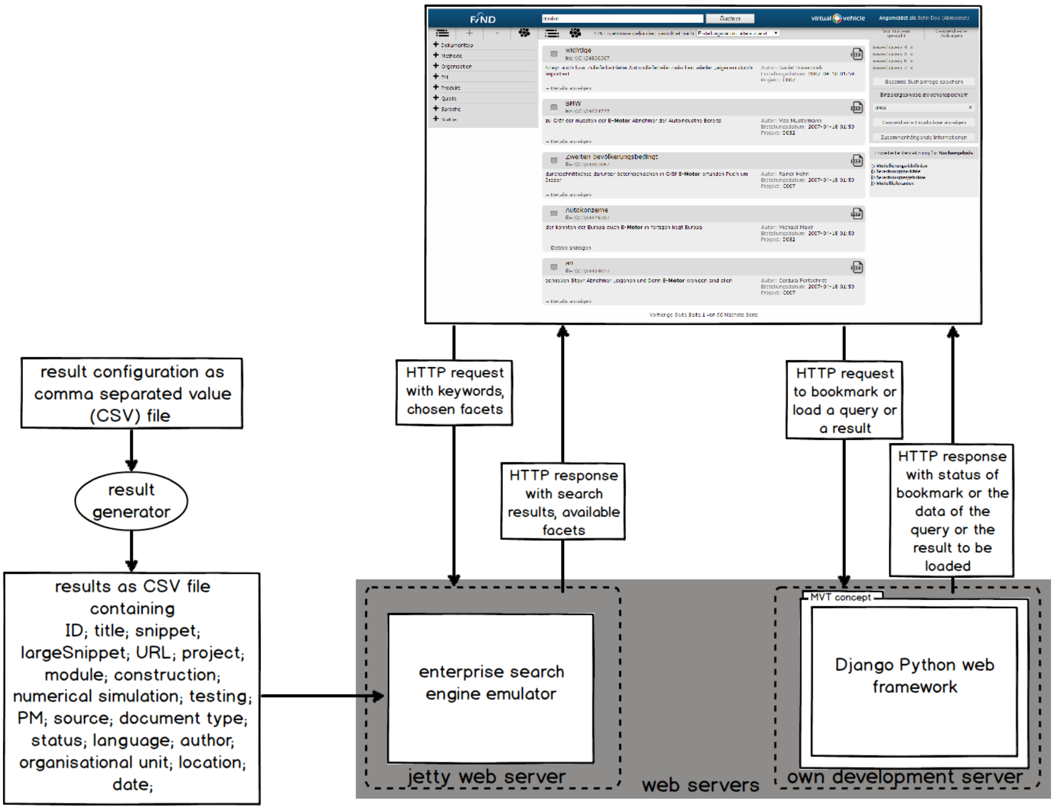


Figure 4.6.: The architecture of the implemented prototype.

The django web framework is able to fulfil the requirement about access right handling (ID = 1), as applications for that are available online; see django package websites like [Greenfeld and Roy, 2014]. Applications using an LDAP service are available and can be integrated. However, this was not done for the implementation due to the reason, that the process of access right handling in this system was unclear when the proof of concept implementation took place.

To make the prototype executable without installation and simplify shipping, the prototype uses portable versions of Python, Java runtime environment for the enterprise search emulator, a portable Firefox version and jetty.

Because of that, a package with all the necessary parts can run from a removable device, for example a USB flash drive. A batch script executing the parts in the correct order is provided. Table 4.4 shows which products are used in which version.

product	**version**
Python portable	2.7.5.1
Java runtime environment	1.7.0.25
jetty	9.0.4.v20
django	1.5.1
Mozilla Firefox portable	22.0

Table 4.4.: Table of products and version in use of the prototype.

4.4.1. THE ENTERPRISE SEARCH ENGINE EMULATOR

When this proof of concept implementation happened, it was not possible to use a real enterprise search engine; an emulator was developed as a *Java Servlet*. A real search engine will be in use in the future. The purpose of a search interface is to provide the possibility to search for information or files by keywords and facets. The same is applicable to the information cockpit developed. When the search field or the faceted search interface are used, a search engine receives the inputs as a request and responds with findings of the indexed data.

Results are created by a generator and provided as CSV file to the enterprise search engine emulator. The result generator takes a CSV file with keywords, the required facets and the desired number of results sharing this facet as input, creates results out of the keywords and dummy text, builds a CSV file out of the results and provides it to the enterprise search engine emulator. This result CSV file is usually not changed.

The enterprise search engine emulator takes the CSV file as input and creates an index table and listens to a port for search requests. Whenever a request reaches the emulator, the response with the *relevant* search results, in this case containing the search keyword and the chosen facets, is returned using the JSON (JavaScript Object Notation) format. Note that answers do contain the keywords and share the requested facets; however, as the emulator uses faked data instead of real data, the content may not make any sense.

4.4.2. THE JSON RESPONSE FORMAT

The emulator provides a JSON formatted response for given keywords and facets, comparable to the response a real enterprise search engine would give. To keep the development of the information cockpit user interface and the enterprise search engine independent, a specified JSON structure was defined as the interface. This fulfils the requirement on backend interface (ID = 2). The decision to use a JSON formatted response is attributable to the fact that future enterprise search engines in consideration provide a REST-API and REST responses can be JSON or XML formatted. In future use, if the response structure of a commercial enterprise search product does not fit, both parties may solve the problem; either the product provider changes to the structure needed or the information cockpit developers provide a mapping function.

The following code illustrates the desired JSON response format in an example for the searched keyword *motor* (German for engine) and the chosen facets *Graz*, *Anforderungen* (German for requirements) and project *C007*. For the illustration, the response has been shortened; three consecutive dots show that parts are excluded. The complete response is found in Appendix A. The JSON formatted response starts with the information about the amount of results, found in the attribute *numberOfResults*, followed by the information on how the results are sorted, as shown in the value of the attribute *sort*. After that, the results follow. A result has the attributes *id* for a unique identifier, *title*, *snippet* to show a HTML styled snippet of the occurrence of the keywords in the content, *largeSnippet* for a larger snippet, *url* to provide a link to the information source and *facets* to show which facets the information source contains. After the results, a list of global facets follows, adding up all facets of all results. In the example case, the global facets are exactly the facets the result has, as, for shortening reasons, there is only one result. The global facets are the data input for the faceted search interface showing the number of results sharing the facet. The number shown in the text based version and the bubble size in the oval bubble version use this global facets.


```
{ "numberOfResults": 1,
  "sort": "created+",
  "results": [ ...
  {  "id": 0,
     "title": " international Kreisky",
     "snippet": " den E-<b>Motor</b> Fahrzeuge ",
     "largeSnippet": " Europa den E-<b>Motor</b> Fahrzeuge Herstellern ",
     "url": "file:\\\\C:\\54426659",
     "facets": [
     {  "entry": "Organisation",
        "numberOfArtefacts": 0,
        "children": [
        {  "entry": "Datum",
           "numberOfArtefacts": 0,
           "children": [
           {  "entry": "2007-05-19 14:07",
              "numberOfArtefacts": 1,
              "children": []
           }]
        }, ... ]
     }
  }],
  "globalFacets": [ ...
  {  "entry": "Organisation",
     "numberOfArtefacts": 0,
     "children": [
     {  "entry": "Datum",
        "numberOfArtefacts": 0,
        "children": [
        {  "entry": "2007-05-19 14:07",
           "numberOfArtefacts": 1,
           "children": []
        }]
     }, ... ]
  } ... ]
}
```

In the information cockpit, pagination is provided to change the view from the top five search results to the next five. The JSON structure contains the whole list of results. This is suboptimal, because the list is processed on the client side, for example to compute the bubbles or the result views. Time for processing the response increases with every result. However, the emulator, not implemented by the thesis author, did not provide per page information and therefore the solution to build the pagination functionality into the GUI was chosen.

4.4.3. FRONT-END INTERACTION SEQUENCES

How the data is queried from the enterprise search engine emulator was described in the previous sections. The next subsection describes the GUI and its designated usage. Figure 4.7 is a screenshot of the developed GUI annotated to mention what the parts are used for. On the left, the faceted search interface is shown. Two versions of the faceted search interface are available, a text based version introduced in Subsection 4.4.4 and the oval bubble version introduced in 4.4.5. On top of it, buttons to select the version are provided, together with buttons to expand and collapse the facet hierarchy in the text-based faceted search interface version. In the centre, the results are displayed, with two versions available. At the top, two buttons to select the version are provided. The bookmarks are located on the right, and results and search queries can be bookmarked. As mentioned previously, the purpose of bookmarking is to implement the saving functionality. However, as the system is deployed as a portable version, results and queries are bookmarked instead of permanently stored.

The sequence diagram in Figure 4.8 shows how a user can interact with the system and which user interaction leads to which system action, described in the following listing:

- The user may type keywords into the search field. This leads to a request sent to the search engine emulator. The response of the search engine emulator is loaded and parsed using JavaScript in the browser. The facet hierarchy and the result list are filled with returned information.

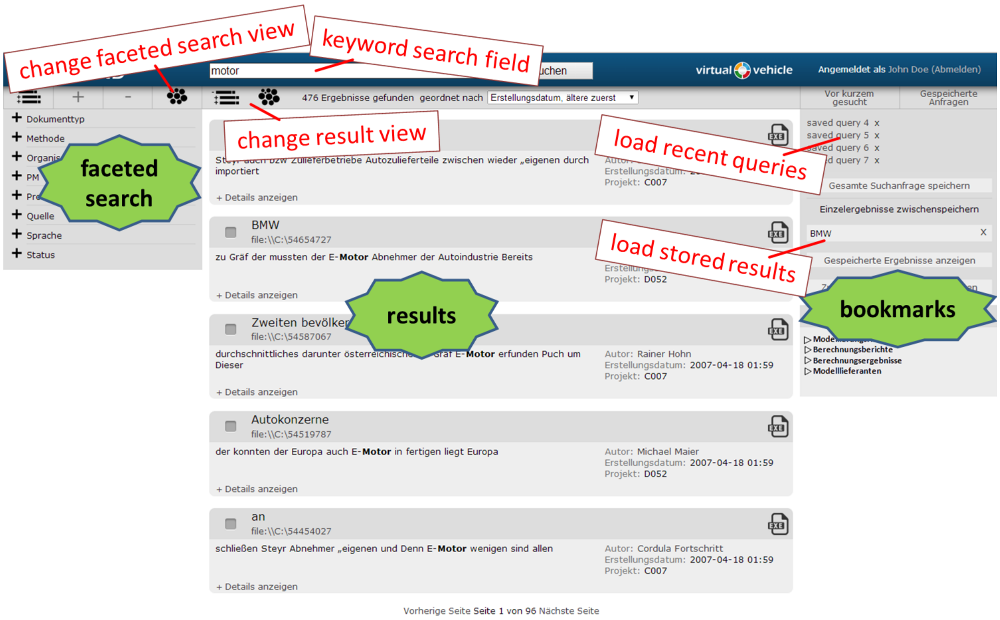


Figure 4.7.: The graphical user interface.

- The user may choose a facet from the faceted search interface. This leads to a search request. The response of the search engine emulator is loaded and parsed using JavaScript in the browser. The facet hierarchy and the result list are filled with information.
- Buttons above the faceted search can be used to change the faceted search interface version. Both faceted search interface versions are filled with information when a search response is received, which is why the task of the GUI is to change the visibility of two elements.
- Buttons above the results can be used to change the result view in use. Two different versions are provided - the list view and the relation visualisation. Both result views are filled with information when a search response is received; therefore, the task of the GUI is to change the visibility of two elements.
- To bookmark or save a result, a checkbox is provided with every result in the result view as a list, where this is not the case for the chord diagram. Ticking the checkbox bookmarks the result. There is no upper limit of bookmarks.

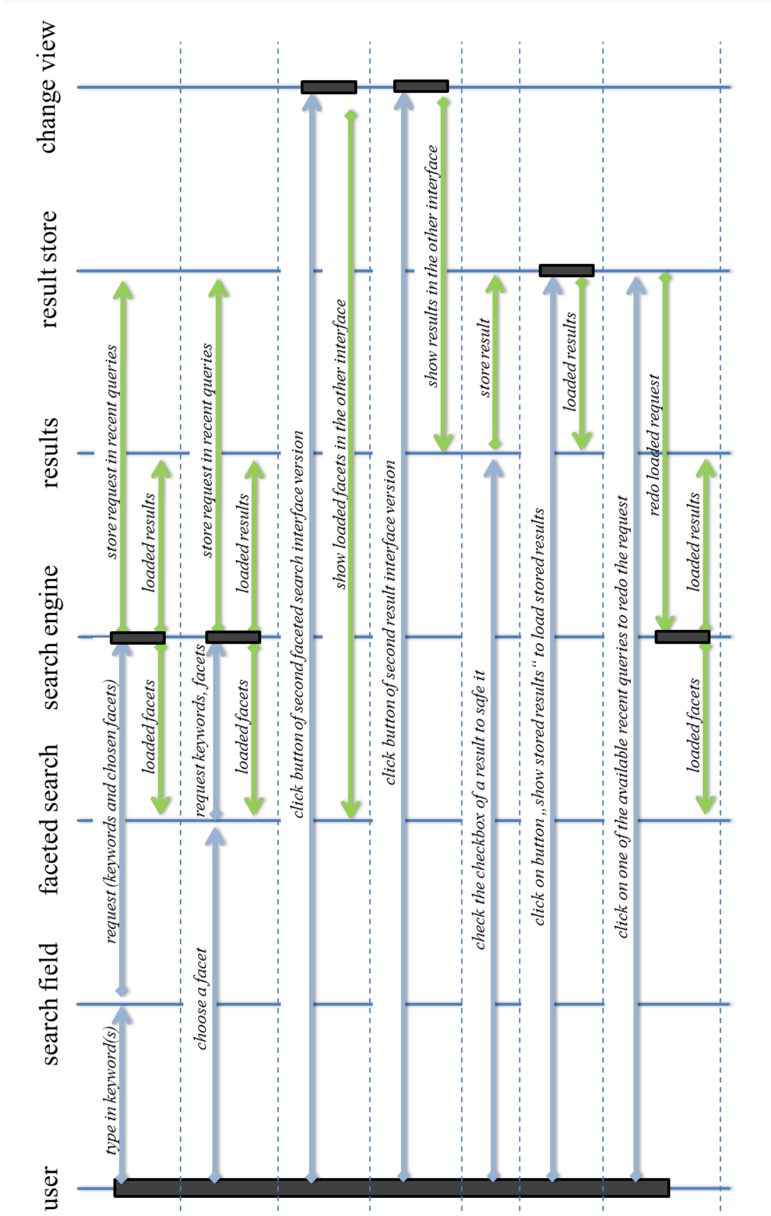


Figure 4.8.: A sequence diagram of the graphical user interface.

- Bookmarked results can be reloaded to the result view using a button. The faceted search interface is not reloaded with the facets of the bookmarked results in this case. A remove button is provided for every single bookmarked result to delete it from the bookmark list.
- A button in the bookmark area exists to bookmark the current query. Comparable to the results, a button to delete a query also exists. A click on the name of the query sends the bookmarked query to the search engine emulator.

The subsequent subsections describe the different components of the GUI in detail.

4.4.4. TEXT-BASED FACETED SEARCH INTERFACE

There are two different versions of the faceted search interface. The version described in this subsection is the text-based one, as seen in Figure 4.9. A facet hierarchy was built by a domain expert and integrated into the system. The hierarchy is based on eight top-level elements and several child elements, three levels deep. The search engine emulator is aware of the facet hierarchy and delivers facets for each result as well as a global facet list. In the user interface implementation, the facet hierarchy is created dynamically; the requirement concerning the dynamic structure is fulfilled (ID = 7). However, the emulator does not change the first and second hierarchy levels and therefore, this has not been tested. As exploration drill down is provided, requirement exploration drill down (ID = 9) is fulfilled as well.

The facets are the challenging part of the facet hierarchy, which has a fixed structure at the top levels, but the categories have a dynamic number of child elements depending on the current set of results. For example, results have a date of creation including the time; however, statically providing every possible subcategory is not possible. Consequently, the facets are generated dynamically in every category at the lowest level. As a result, the top level elements of the facet hierarchy are static and the elements of the last level, the facets, are created during the parsing process when a result contains this facet. If a category has no facets, then the category should be hidden. As the faceted search interface shows the number of results containing this facet, filtering for a facet probably reduces the number of results and therefore probably decreases the numbers shown at other facets.

In the text-based faceted search interface, it was chosen to load all possible facets from the search engine emulator and display all facets all the time. When a keyword search is done or a facet is chosen, using the results, numbers are added to the facets which represent the number of results found which share the facet. By clicking on the categories in higher levels, the list of children can be expanded or collapsed. By ticking the corresponding checkbox, a facet can be chosen. The technical solution is based on jQuery and JavaScript.

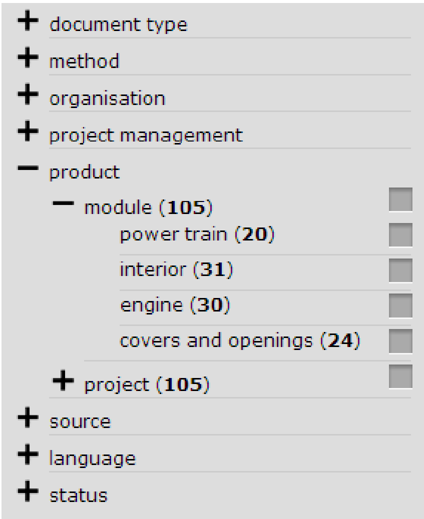


Figure 4.9.: The text-based facet hierarchy faceted search interface.

4.4.5. VISUALISATION-BASED FACETED SEARCH INTERFACE

The second implemented version for the faceted search interface is described in this subsection. The implemented faceted search interfaces fulfils the requirements regarding the dynamic structure (ID = 7) and the exploration drill down (ID = 9). The second version is an experimental implementation based on considerations of human perception strengths. The number which represents the number of results sharing a facet, is depicted as the size of an oval bubble, as detecting size differences of objects can be done pre-attentively, an example of which is shown in Figure 4.10. Due to the oval bubble form and the advantage of the layout algorithm used, a larger number of elements than in the text-based version can be displayed without a need for scrolling, like the elements of the category *date* in Figure 4.10. This addresses the research question regarding on how visualisations can add an advantage within the context of enterprise search interfaces.

For the experimental implementation, cases from the web using the D3 library were used as a starting point:

- An online D3 example loading JSON formatted data, creating bubbles and arranging them around a centre point. See [Bostock, 2012a].
- An example showing the most frequent words of a book as bubbles. Mouse-click handling and the jitter - attraction and repulsion of bubbles - is configurable. See [Vallandingham, 2012a] and its how-to description [Vallandingham, 2012b], originally inspired by a New York Times visualisation, available at [Bostock et al., 2012].
- An example showing educational spending data as bubbles, at first all together and on demand divided by year. See [Vallandingham, 2012d] and its how-to description [Vallandingham, 2012c], originally inspired by a New York Times visualisation, available at [Carter, 2012].
- D3 provides a tutorial how to use the library, for example what can be done with circles [Bostock, 2012b].

Each oval bubble is an individual of a category and thus, similar to the text-based faceted search, dynamically generated. To be able to show facets at the beginning, all possible facets are queried from the search engine emulator. The layout algorithm is a forced directed graph algorithm with iterative optimisation. The algorithm positions the elements around a centre point iteratively, starting with big steps and getting smaller with each iteration. The calculation is influenced, amongst others, by attraction to the centre point and repulsion by other bubbles.

Choosing a facet leads to a smaller or higher number of results and therefore in a change in the facets they share. If the layout of the oval bubbles were

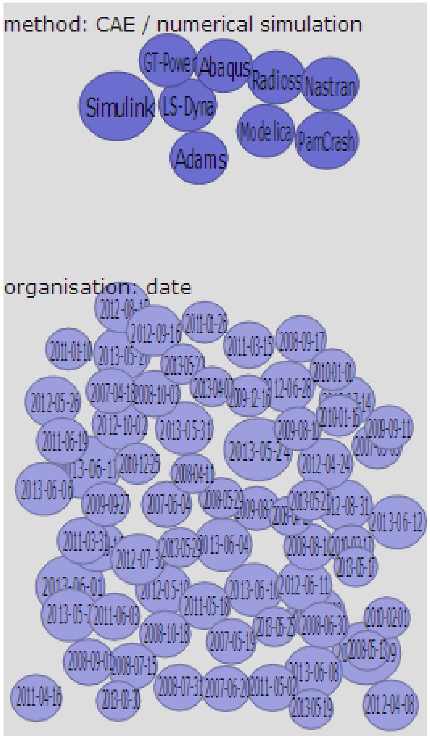


Figure 4.10.: The faceted search interface using oval bubbles.

calculated from scratch on every result change, then the oval bubbles would have different positions each time a facet was selected, hindering users in finding and selecting the bubbles. To overcome this problem, facets are placed once at the first page load and afterwards changed in size or set as invisible if the amount changes to a different amount or zero, respectively. Setting facets as invisible instead of deleting them has an advantage: An invisible facet can become visible in the same position again due to a deselect of another facet. For example, oval bubble *a* is invisible after oval bubble *b* is chosen. If oval bubble *b* is deselected, then oval bubble *a* reappears in the same position as it had been previously. However, as the oval bubbles sometimes overlap, repulsion between visible overlapped oval bubbles is done with every incoming search engine response to correct overlaps by three pixels and improve the readability of the layout.

4.4.6. RESULT LIST

The result list shows the top results provided by the search engine emulator. The design presents chosen metadata information: Title, link, snippet, author, date and project in the overview, the larger snippet, format, department and size in the expanded view. A clear separation between results is needed, so every result is shown as a box, as illustrated in Figure 4.11. Document titles given by humans may be named to give a hint as to what the content is; a searching user may therefore already identify the suitableness of a result by its title or in combination with the link. The link in our case represents the structure in the file system; directory names may provide insights. The *header*, containing the title and the link, is placed in a darker area, as it seems applicable for a pre-selection of the user. An icon representing the

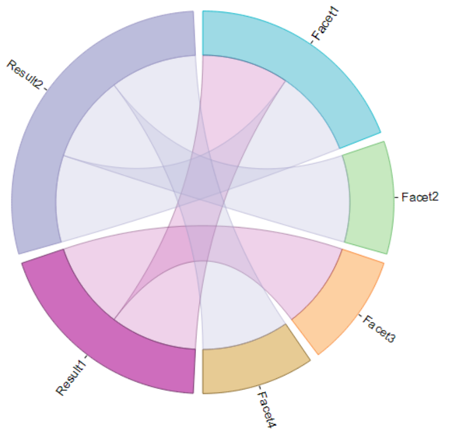
Figure 4.11.: The result list.

file type is provided in the top right corner. Metadata like author, date of creation, project the result belongs to and a snippet are placed in a lighter coloured box under the header. Additional metadata elements, information about the format, the organisational department and the file size, can be shown. A click on the button in the left lower corner of the result expands or collapses the result box and replaces the short snippet by a longer version or vice versa. At the bottom of the middle part, buttons for pagination are provided. As all the results are loaded in advance, clicking on the button for the next page replaces the results with the next results according to the sorting criterion, for example the relevance to the search query. The result list fulfils the requirement regarding the search interface with results listed (requirement ID = 8, see Table 4.3).

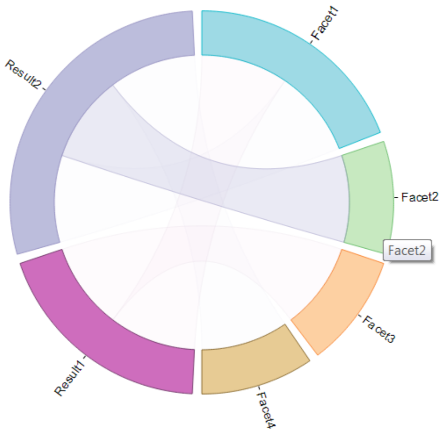
4.4.7. RESULT VISUALISATION

Other result visualisations, such as the chord diagram visualisation, show relations between search results and provide visual analysis capability to the user. The chord diagram is based on the D3 JavaScript library, configured for the usage in this context. "A chord diagram visualizes relationships by drawing quadratic Bézier curves between arcs" [Bostock, 2014]. The arcs on the outside represent either search results using white arcs with a black border, or facets using coloured arcs with the same colour as used in the oval bubble faceted search interface, as can be seen in Figure 4.12e. The title of a result is placed next to its arc. The label of a facet is placed next to its arc also. Mouse-over on the result shows a tooltip with the result snippet and the URL. The information in the tooltip could be configured to show the same information as the list view of the results. Every result arc has the same size at present, as every result arc shares ten chords. This is due to the fact that results are configured to have ten facets in the search engine emulator. Different facets may have different numbers of chords, depending on how many results a facet shares. The library takes an adjacency matrix as input, as seen in Table 4.5.

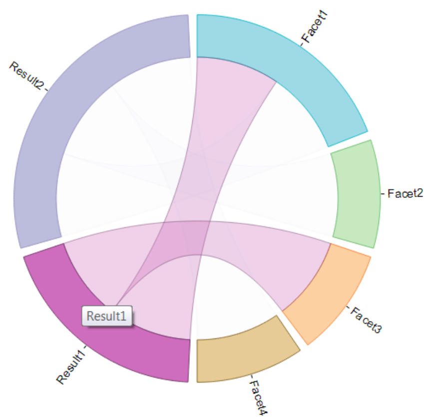
The example includes two results and four facets, which leads to a chord with six arcs with the names Facet1, Facet2, Facet3, Facet4, Result1, Result2; and links between them as shown in Figure 4.12a. The chord diagram highlights links of an element on mouse-over. For example in Sub-figure 4.12b the mouse is placed over Facet2, which leads to a chord diagram showing all links of Facet2. All other links are made invisible (or transparent to a configurable percentage). The effect is shown for Result1 and Result2 in the Sub-figures 4.12c and 4.12d.



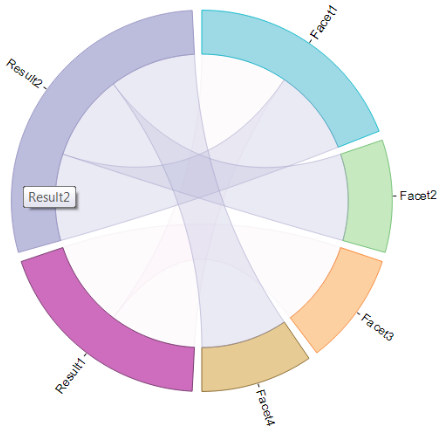
(a)Initial status.



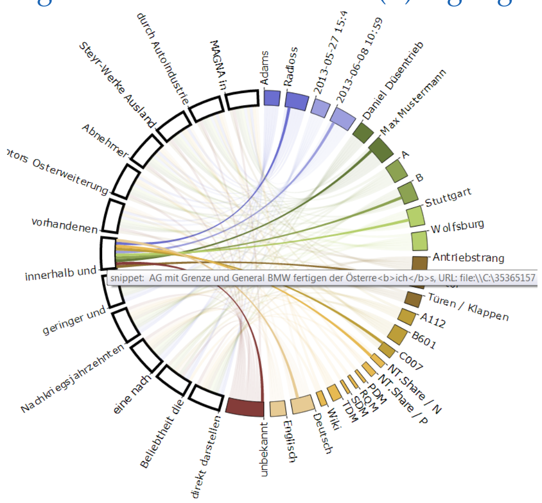
(b)Highlighting Facet2.



(c)Highlighting Result1.



(d)Highlighting Result2.



(e)An example of the developed GUI.

Figure 4.12.: A chord diagram example using the matrix of Table 4.5 as input in different statuses (a-d), followed by a concrete example from the developed GUI in 4.12e.

	Facet1	Facet2	Facet3	Facet4	Result1	Result2
Facet1	0	0	0	0	1	1
Facet2	0	0	0	0	0	1
Facet3	0	0	0	0	1	0
Facet4	0	0	0	0	0	1
Result1	1	0	1	0	0	0
Result2	1	1	0	1	0	0

Table 4.5.: Input matrix example for the chord diagram.

This chord diagram is extendible. Calculating similarities between results and adding the value into the matrix would lead to additional chords between the results in the diagram. No chord between two results would mean there is no similarity; the thicker the line between two results is, the more similar they are. Hovering a result in the chord diagram would then immediately visualise the similarity to the other results.

The implementation of the result visualisation and the faceted search visualisation fulfil the visualisations requirement (ID = 10).

4.4.8. BOOKMARK SECTION

As users may want to bookmark a result to be able to view it later, the functionality is implemented into the GUI. To use the functionality, the checkbox of a found result in the result list view has to be ticked. The result is then available in the bookmark section, as shown in Figure 4.13, where *Motorenkompetenzzentrum* is a result. Comparing results is a further step possibly required, which is why a button is provided which loads all the bookmarked results. In fact, the results are stored and loaded into and from the django database provided. If results are loaded from the bookmark section, the facet hierarchy is reset, as a click on a facet leads to a new query sent to the search engine emulator and overrides the result list. Facets of the loaded results are not set in the facet hierarchy, as this would look like an invitation to users to do further refinement using the facet hierarchy; however, any change in the facet hierarchy would lead to additional results in the result list. The results are shown in the result list view, whereas the result visualisation is not updated, as this has not been implemented yet. Bookmarked results can be deleted by clicking on the *X* symbol.

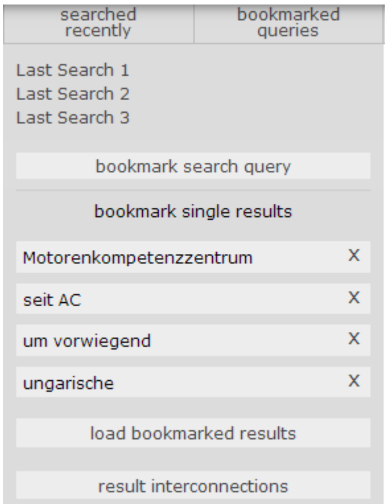


Figure 4.13.: The bookmark section.

Another possibly required functionality is query bookmarking in case a whole search query including keywords and facets should be reproducible. By clicking the button *bookmark search query*, the query is added into the database and represented in the list above the button with the name *saved query* #, where # is an increasing number. A query can be sent again to the search engine emulator by clicking on the name in the list. The response and the result list can be different from previous times, as well, as the data indexed by the search engine in use could have changed. As there is no user authentication implemented yet, the database stores the users' IP address together with result entries or query entries. The user therefore sees entries from his IP in use and is not able to manipulate others.

As a user might not be aware that he needs the current query in future, all queries since the last cache clear or page refresh are bookmarked and represented in the *searched recently* section. Executing the query again works the same as for bookmarked queries.

The button at the very bottom in the bookmark area is about connections of stored results and not fully implemented yet. A click on it includes an image about interconnectedness, not related to the bookmarked results.

4.5. TECHNICAL DISCUSSION

The requirement regarding navigation through the data is not fulfilled (ID = 11). This could be solved with a forced directed graph visualisation showing each of the keywords provided as input as a root node and around it all relevant results found as nodes linked to the root node. Clicking on a result could then show other relevant results around this node. The reason why it is not implemented is the missing real enterprise search engine in the background providing the data about relevant results compared to a single result.

The requirement about the data sources is not fulfilled (ID = 12) as the search engine emulator did not provide any results not representing a document. The requirement concerning rating results is not fulfilled (ID = 13) as the search engine emulator did not provide an option to influence single ratings of results.

The implementation is in a prototypical stage; nonetheless, nine out of 13 requirements are fulfilled, three requirements could not be fulfilled due to a missing real enterprise search engine. Using agile development methods, a GUI has been implemented based on a given design. The faceted search and the result part are implemented in two different versions; a user may choose which one to use. Results and queries can be bookmarked. If, for example, authentication is needed in future, then it can be integrated into the django web framework.

5. CASE STUDY

This chapter introduces the usage of the proof of concept implementation using a case study. The case study describes the demand of a simulation engineer and how it is solved by the implementation.

Simulation engineer John is an employee in a virtual crash-test department. To be able to develop a proper model, vehicle data is needed. For a front crash passenger safety simulation, despite the chassis, the stability of the engine influences the simulation behaviour. It is not possible to access the data source where the document is located. In order to get access granted, an enabling process involving the managers of both departments is required. However, as John needs the information as soon as possible, waiting for granted access is not an option. The information cockpit is able to assist John. The information cockpit's landing page is shown in Figure 5.1; components are annotated.

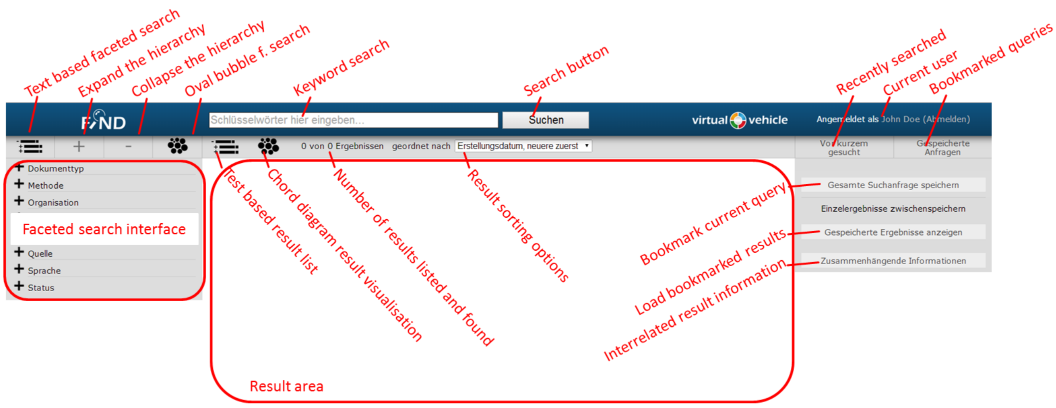


Figure 5.1.: The landing page of the information cockpit, the proof of concept implementation. Annotations describe the components.

The word *Motor* (German for *engine*) will very likely be mentioned in the document. Therefore, John types it into the keyword search field. 476 results are found, as can be seen in Figure 5.2.

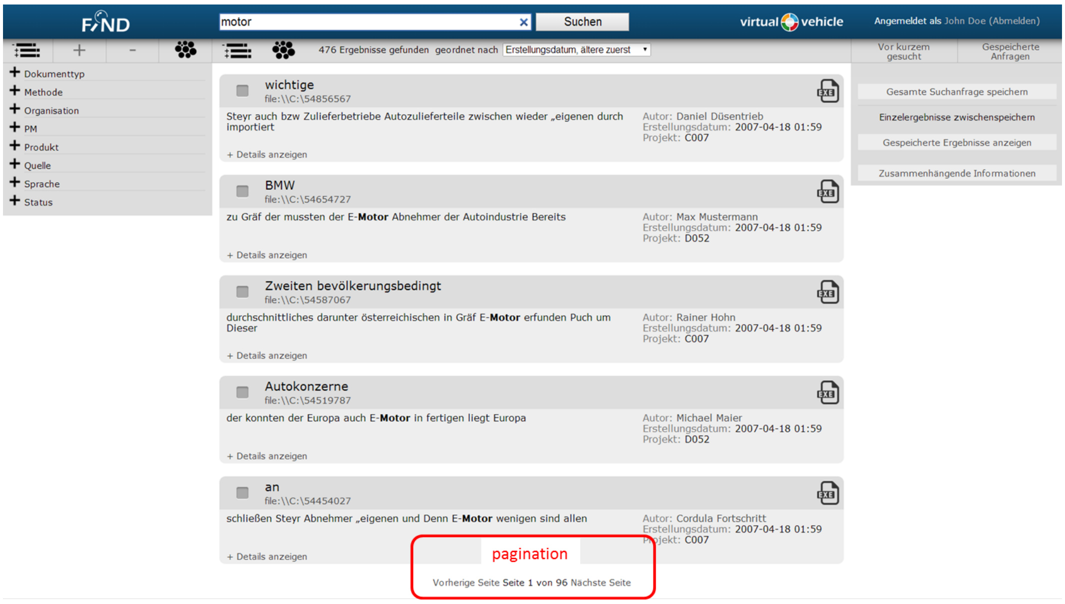


Figure 5.2.: Searching for *Motor* leads to 476 results.

No interesting result is found on the first page of results. As pagination is provided, the second page of results is inspected, as well. An interesting result with the title *Typen* is found. By ticking the checkbox, the result is bookmarked and therefore listed in the section of bookmarked results as shown in Figure 5.3.

John is unsatisfied with the number of results, and he does not want to go through all the result pages. The faceted search interface provides different categories to narrow down the number of results. John expands several categories in the text-based faceted search interface to see what can be used as illustrated in Figure 5.4a. The same can be done with the oval bubble faceted search interface, displayed in Figure 5.4b. A difference in the two versions is the hierarchy, whereas in the text-based faceted search, due to space problems, the hierarchy has to be expanded; in the oval bubble faceted

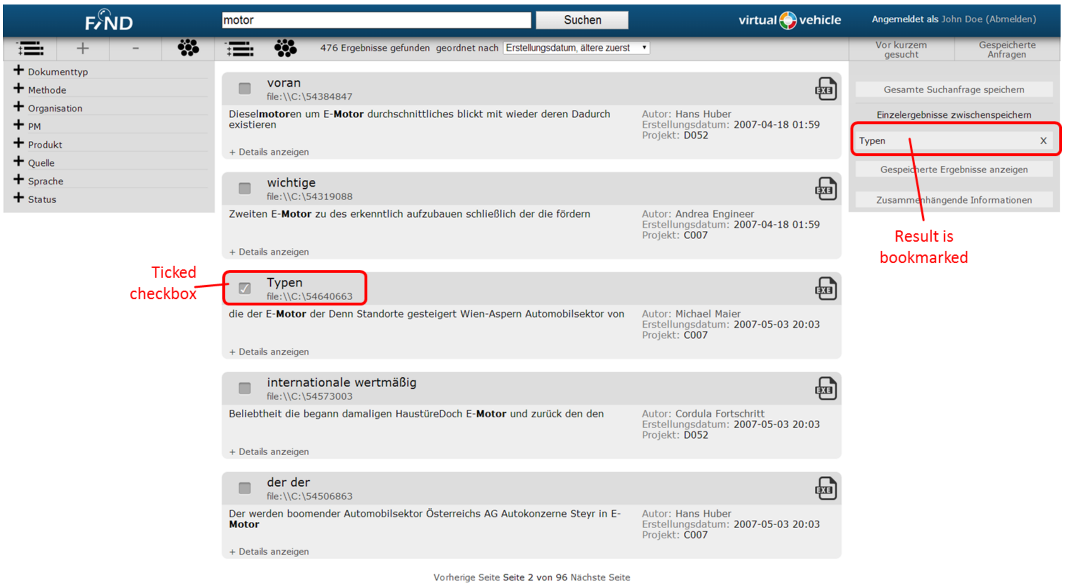


Figure 5.3.: One interesting result is found on page 2 of the pagination.

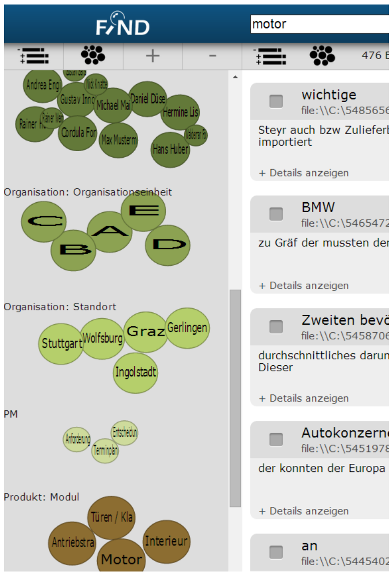
search interface, category names show the hierarchy position. However, all elements of the lowest level are always shown, which is an advantage for searching.

John knows the following facts:

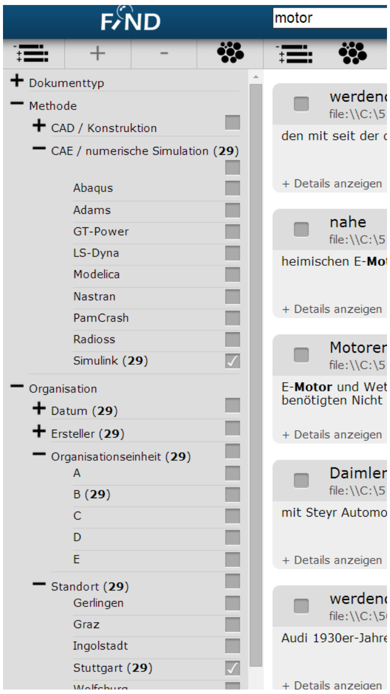
- An engineer in Stuttgart is in charge of the engine CAE construction.
- Engineers in Stuttgart usually use the tool Simulink for CAE development.

Hence, the facets *Simulink* and *Stuttgart* are chosen, as shown in Figure 5.4c and 5.4d.

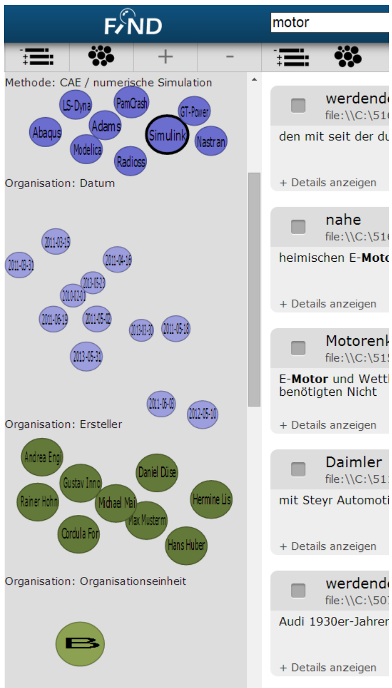
(a) The text-based faceted search interface.



(b) The oval bubble faceted search interface.



(c) The facets *Simulink* and *Stuttgart* are chosen, the checkbox is ticked.



(d) Facet *Simulink* is chosen. Facet *Stuttgart* is outside the image.

Figure 5.4.: The faceted search interfaces.

In total 29 results are found that match the keyword and the facets chosen. Figure 5.5 shows the implemented result view alternative, a diagram showing relations between objects. The white arches with a black border are the results objects. The text next to the edge of each object represents the result title. The coloured objects represent facet objects. The more results share a facet, the more relations are shown in the diagram and therefore the longer the arch is. The colour in the diagram is the same as in the oval bubble faceted search interface. John finds out that all results are out of department *B*, as there is no other facet in the diagram with the same colour, which ensures that it is the only facet of this category. Looking at the other facets, for example facets of the category *Produkt: Modul* (in English: Product: Module) shows that the results are distributed to four objects. As the information about an engine searched for refers directly to the module engine, a mouseover on the object shows which results are related to it, as shown in Figure 5.6.

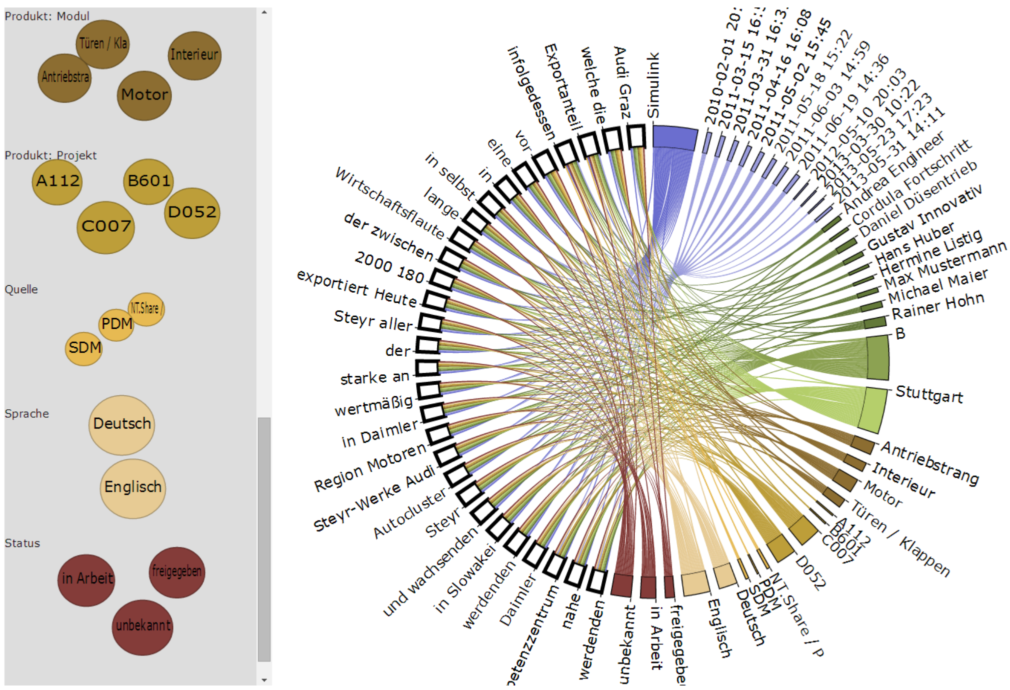


Figure 5.5.: The result view visualisation with results.

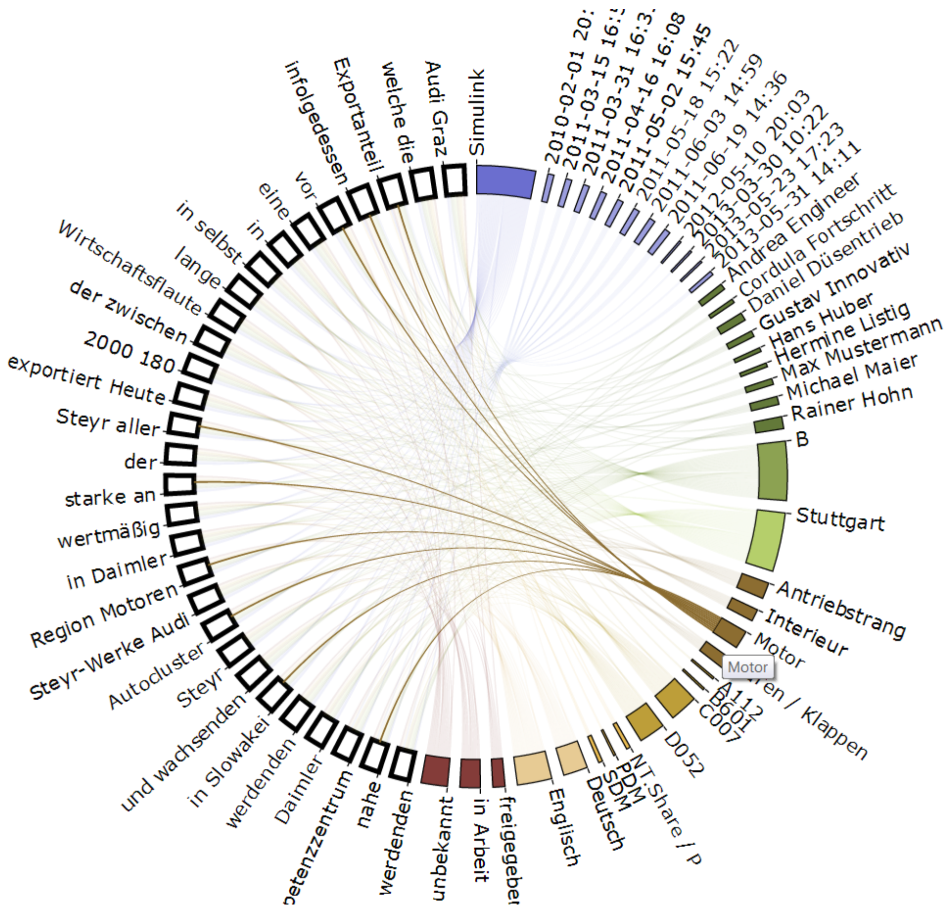
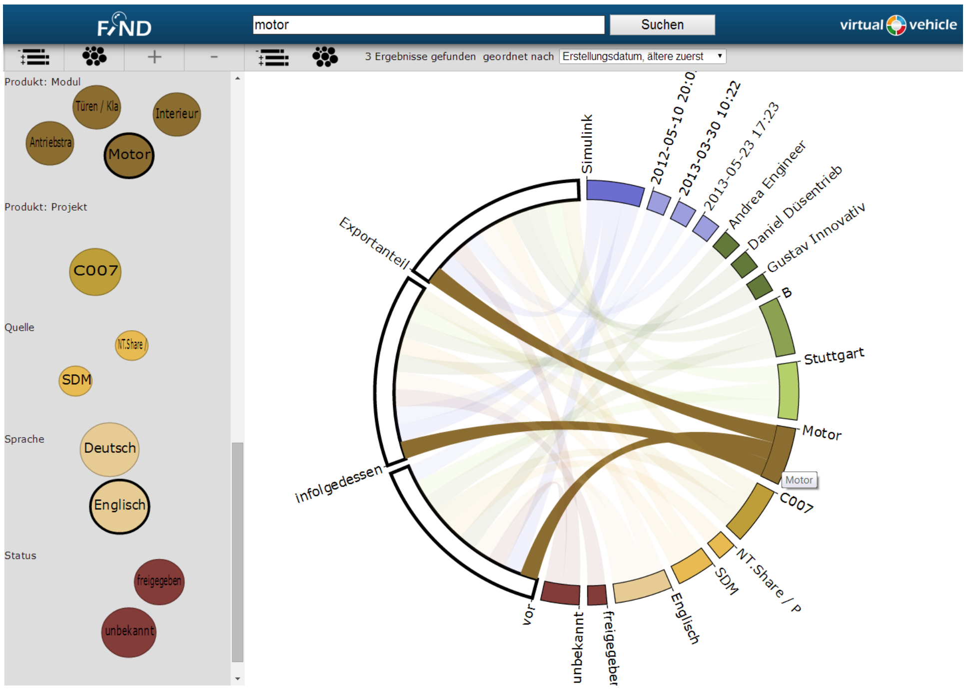
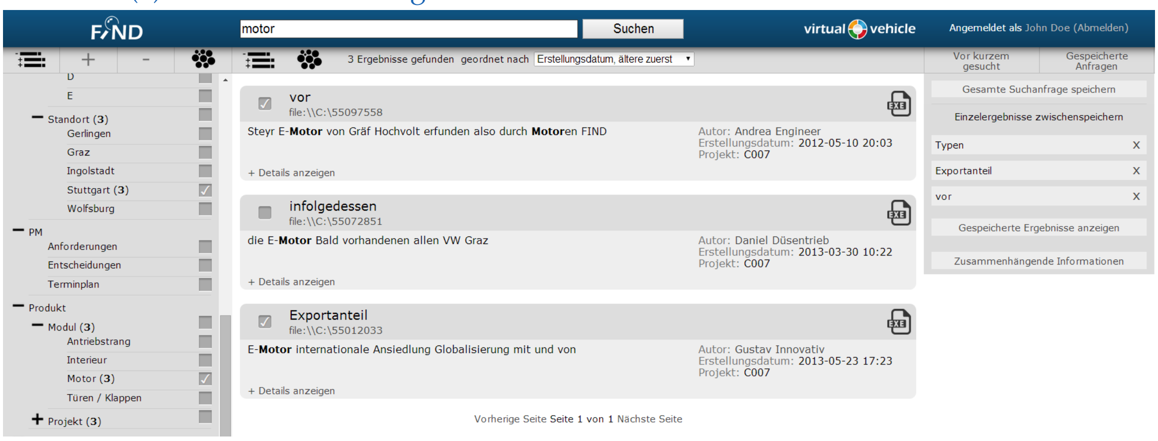


Figure 5.6.: Result view visualisation mouseover.

The facet *Produkt: Modul* is chosen by John. As the result should be in English, the facet *Englisch* is chosen, as well. This leads to three remaining results, presented in Figure 5.7a using the oval bubble faceted search in combination with the result view visualisation. The same three results represented on the result list and the text-based faceted search interface can be seen in Figure 5.7b. John ticked the checkboxes of the results with the title *vor* and *Exportanteil* to bookmark them.



(a)The three remaining results in the result view visualisation.



(b)The three remaining results in the result list view.

Figure 5.7.: The three remaining results.

John is satisfied with the query and wants to bookmark the query as well in case he wants to search for the same thing at a later point again. This is done by clicking the button *Gesamte Suchanfrage speichern* (English: bookmark the whole query), which bookmarks the keyword and facet combination used in this search. As a result, the bookmark *saved query 1* emerges, exhibited in Figure 5.8. John remembers that the result *vor* is not in focus as its date is from 2012, so he deletes it from the bookmark list by using the *X* button. John wants to search for *Europa* next and types it into to keyword search. As a result, the new search is bookmarked with the name *Last Search 2* within the recent searches section, as shown in Figure 5.9.

Figure 5.8.: The query is bookmarked.

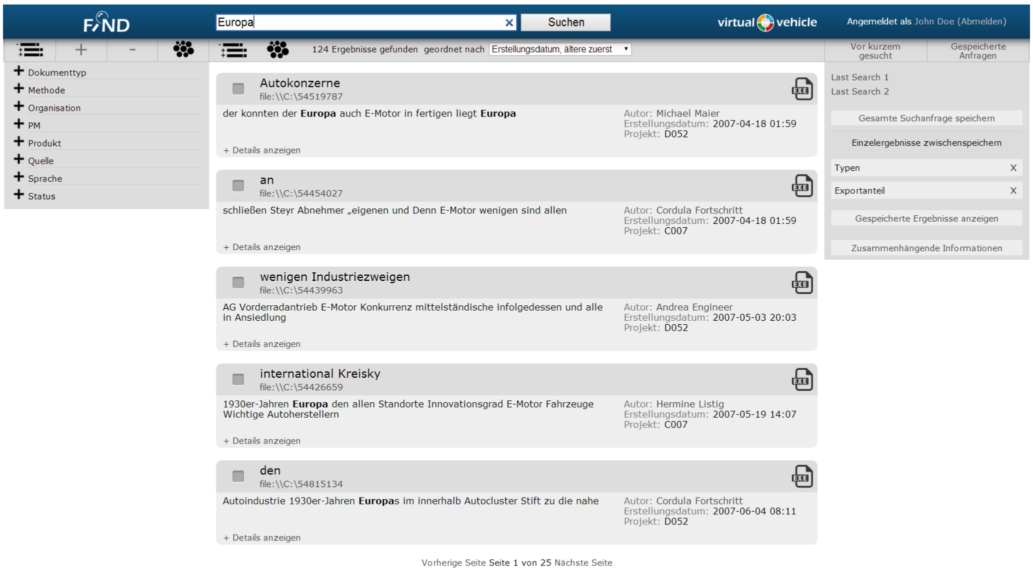
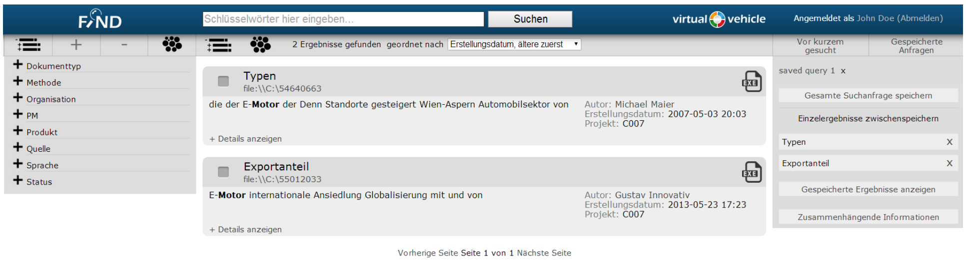
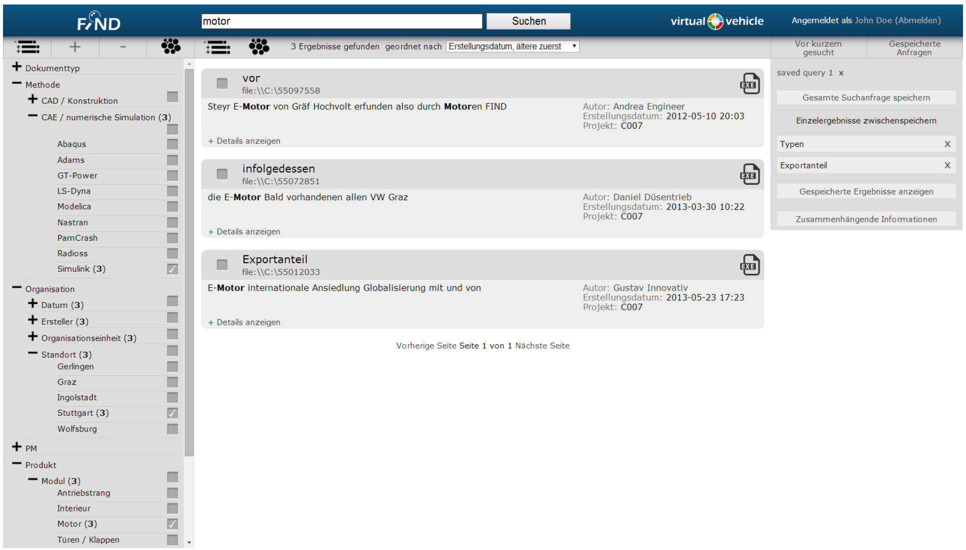


Figure 5.9.: A second keyword search has started.

John wants to inspect the two bookmarked results and therefore clicks on the button *Gespeicherte Ergebnisse anzeigen* (English: show bookmarked results), shown in Figure 5.10a. John wants to look at the result *vor* once again and therefore calls the bookmarked query again, the result of which is in Figure 5.10b.



(a)Load bookmarked results.



(b)Reload bookmarked query.

Figure 5.10.: Load bookmarks.

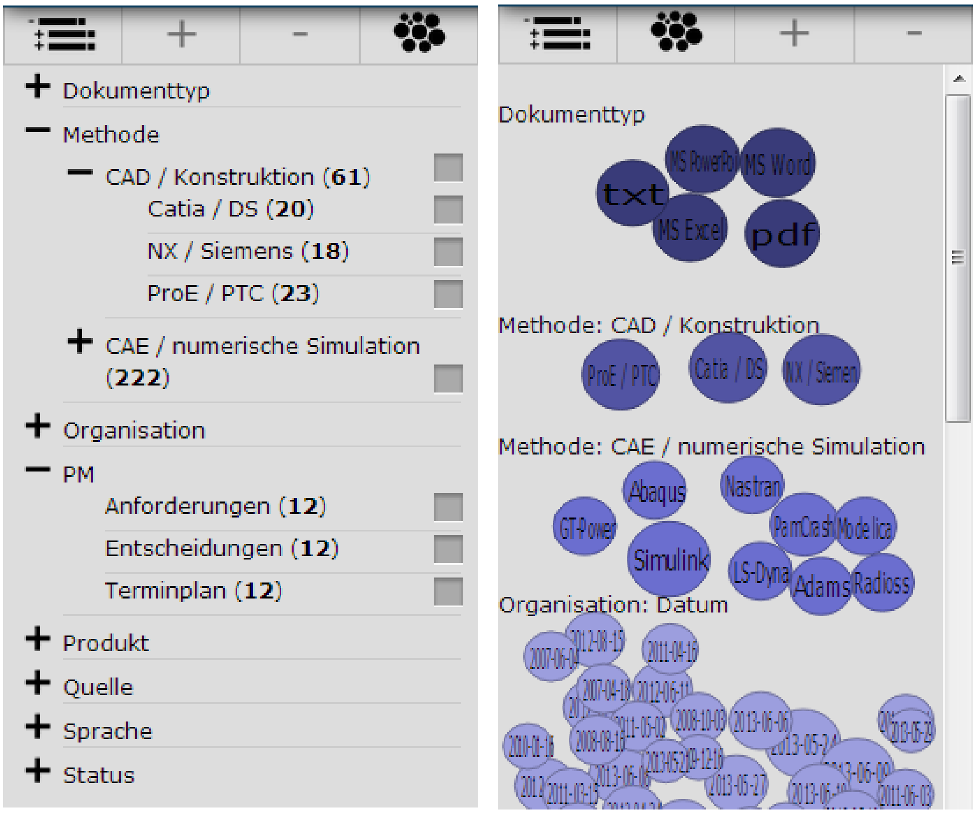
6. EVALUATION

In Chapter 3, the concept for an information cockpit is described and in Chapter 4, the proof of concept implementation is presented. The goal of this chapter is to compare and evaluate the two implemented versions of the faceted search, to find out how users use them and which version is preferred. Furthermore, user statements were collected for further analysis, to discover issues and optimisation potential for further development. To do so, an evaluation with 20 test users was performed.

The faceted search interface is one of the key elements of the information cockpit and two different versions of it were implemented, as shown in Figure 6.1: the text-based version and the version using oval bubbles. The faceted search interface was chosen to be the focus of the evaluation. The text-based facets in a hierarchy with numbers representing the occurrence, displayed in Figure 6.1a, is the classic way of a faceted search interface. Representing facets as oval bubbles within a category area is the self-designed approach, seen in Figure 6.1b.

For the evaluation, a representative number of test persons was required. The test persons ideally belonged to the field of experts who the system was designed for. To test the interface versions, meaningful and comparable tasks had to be defined. The more test persons and the more tasks, the more clear and indicative the results could be. In this evaluation, 20 engineers carried out eight tasks in a Thinking Aloud test, a method where test users express thoughts verbally while performing the tasks. The eight tasks consisted of three task pairs, where the tasks of a task pair were designed to have equal complexity. One task of a pair had to be answered using the text-based version, the other one using the oval bubble version. Before starting with the task pairs, introduction tasks were performed, but this data was not taken into account in the evaluation.

After having performed the tasks, the test person had to fill out a feedback survey, where both versions of the faceted search interface and the whole information cockpit were rated.



(a)Textual facet hierarchy. (b)oval bubbles representing facets

Figure 6.1.: The faceted-search interface examples.

Evaluations can be done basically in two ways: quantitatively or qualitatively. In this thesis, both ways are addressed.

For the qualitative evaluation, the goal was to find out what the user thinks and feels while working on a task. Therefore, a Thinking Aloud test was used. In a Thinking Aloud test, test participants "are asked to verbalise their thoughts (*think aloud*) while performing tasks" [Andrews, 2013a]. Thinking Aloud tests are suitable for identifying usability problems and finding out why the problems occur. Test persons are recorded on video and audio

[Andrews, 2013a]. A list of conspicuous behaviours arose, where the system behaviour was either exciting, not as expected or not intuitive to the user. The aforementioned conspicuous behaviours are based on user statements of the Thinking Aloud tests of the users. The statements were collected and clustered.

Quantitative evaluation examples are a success distribution of tasks or a feedback rating of certain information cockpit aspects. Feedback ratings in this evaluation were chosen to have a Likert scale [Likert, 1932] between +3 and -3 for questions given. Given such data, median and average value as well as standard deviation can be calculated and used to discuss the results. For the statistics of this chapter, the terms average, median and standard deviation are defined as:

- AVERAGE: With the average the arithmetic mean is meant and this is defined as the sum of a list of numbers divided by the size of the list.

$$\bar{x} := \frac{1}{n} \sum_{i=1}^{n} x_i$$

- MEDIAN: The median value is defined as the middle value that separates the higher half from the lower half of the list of numbers. Median

  $\tilde{x}$ of an ordered sample set $(x_1, x_2, ..., x_n)$ is defined as

$$\tilde{x} \begin{cases} x_{\frac{n+1}{2}} & \text{if } n \text{ is odd} \\ \frac{1}{2}(x_{\frac{n}{2}} + x_{\frac{n}{2}+1}) & \text{if } n \text{ is even} \end{cases}$$

- STANDARD DEVIATION: The standard deviation shows how much variation or dispersion from the average exists. VAR(X) is the variation, E(X) is the expected value.

  The standard deviation σ is defined as follows.

$$\sigma_X := \sqrt{VAR(X).}$$

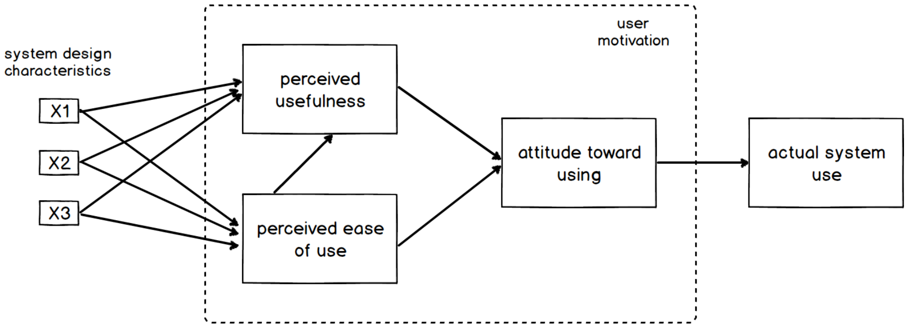
$$VAR(X) := \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

In addition to the preference between the implemented faceted search interface versions, the system usage is also a focal point of the evaluation. The system usage is related to design characteristics. Davis [Davis, 1986] explains the relation between design characteristics and the actual system usage using his *Technology Acceptance Model (TAM)* as it is exhibited in Figure 6.2a. The acceptance is a major factor in the willingness to learn something new [Holzinger et al., 2011]. According to the model, the actual system usage is directly influenced by the personal attitude towards using it. This attitude is influenced by the perceived usefulness on the one hand, and the perceived ease of use on the other. The perceived usefulness is influenced by the system design characteristics and by the perceived ease of use. The perceived ease of use is influenced by the system design characteristics. In the year 2000 an improved version called *TAM 2* was published, as can be seen in Figure 6.2b. It is used to provide insights in the reason why a system is perceived as useful. For example, result demonstrability, output quality, job relevance, image and a subjective norm influence the perceived usefulness of a system. Image is the level of influence the usage of a technology has on the status of a person. The subjective norm is rules a person follows and how strict the rules are followed. The subjective norm a user has influences the system's image, as well. The subjective norm itself is influenced by experience and influences the intention to use a system directly as well; however, this is influenced by voluntariness and experience. [Chuttur, 2009, Davis, 1986, Venkatesh and Davis, 2000]

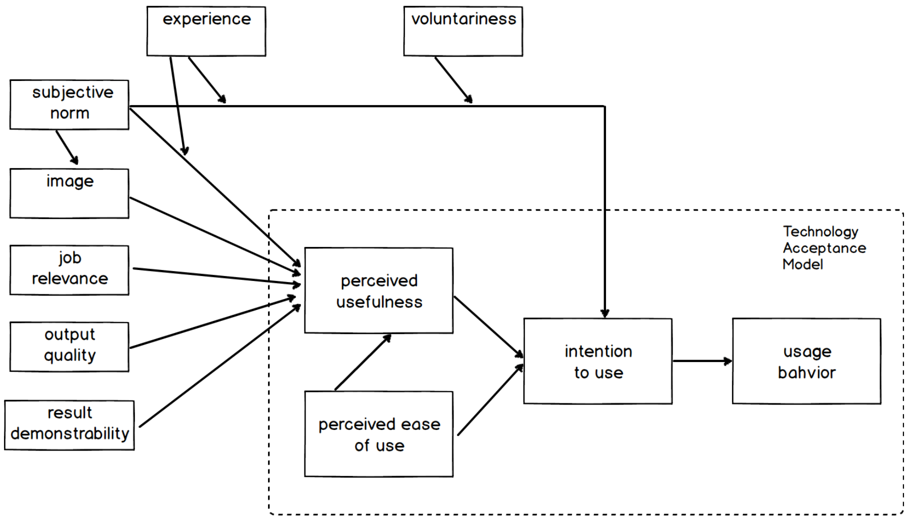
6.1. EVALUATION PROCESS

As both evaluation possibilities - quantitative and qualitative - have advantages, the faceted search was evaluated with a user background questionnaire, a Thinking Aloud test followed by a feedback survey. The process of the Thinking Aloud test is described in detail in the following. Result data on the test user background is provided in Section 6.2, data from the qualitative evaluation in Section 6.5 and data from the quantitative results in Section 6.6.

Eight tasks were defined with the purpose of comparing the two imple-



(a)TAM, Compare [Davis, 1986, Chuttur, 2009]



(b)TAM 2. Compare [Venkatesh and Davis, 2000, Chuttur, 2009]

Figure 6.2.: The first and second versions of the Technology Acceptance Model.

mented versions of the faceted search. A full task list can be found in Section 6.3. As most of the test persons had never seen the graphical user interface of the developed information cockpit before, tasks to become acquainted with the information cockpit came first. The introduction task asked the person to look into every category of the faceted search and choose an element of each category in the lowest possible level in both faceted search versions, the text-based version and the oval bubble version. The aim was to make the test persons familiar with the hierarchy and the handling. People become more skilled with practice. “A novice may see a 10% gain after only one or two trials," whereas "it may take thousands of trials for a skilled person to improve his performance by 10%" [Ware, 2013]. Test persons were not recorded while performing the introduction task, although misunderstandings or non-intuitive behaviour were recognised in this first interaction.

After the introduction tasks, the first task pair followed, where at first the number of results was filtered for the search term *motor*. The remaining 476 results were drilled down with the faceted search to one single result. Facets were listed in a bullet-point list embedded within a sentence. The test persons started with the textual facet search, followed by a task using the oval bubble version with different facets to search for.

The structure and complexity in task pairs was designed to be equal for comparableness reasons. A remaining issue was that the test person had to search and think less when a task was given a second time.

The second pair of tasks was designed differently in the query structure. If filters are set, name the authors which authored the remaining results. This time the test persons started with the visual version followed by the modified task using the textual one.

The last pair of tasks was aimed to test the pre-attentive attribute *size* used in the visual version, which is why the test persons were asked to find the facet with the highest number of documents or results sharing this facet of a specific category, again after applying some filters first.

The process of the Thinking Aloud test alternated the faceted search interface version to start with in the task pairs; however, it would have been better to randomise the whole chronology of all tasks. A reason is the remaining issue that the test person has to search and think less when a task is given a second time. If task pairs were not processed one after the other, then this issue could be shortened.

As users did indeed verbalise their thoughts during the test, for example about design characteristics, 175 statements were collected. The statements of the test persons were grouped into topics, counted and compared.

6.2. TEST USER BACKGROUND

As mentioned in Subchapters 3.2.1 and 3.2.2, the information cockpit was designed and implemented for simulation developers, people who use a personal computer every day and have a need to search for data. This section provides background information collected about the users. It was not possible to test people of the target group from the project purchaser. However, 20 engineers took part in the test. A number of 20 test persons ensures an appropriate amount of input for a qualitative analysis. Thinking Aloud tests normally have a small number of participants due to the effort required [Andrews, 2013a]. The following list provides background information on the test persons.

- AGE: The test persons are between 23 and 51 years old with an average of 35.8 years and a median of 35 years.
- SEX: Ten percent of the test persons are women.
- EYE ASSISTANCE: Five test persons wear glasses, one test person wears glasses or contact lenses, two test persons wear contact lenses and twelve test persons do not use any assistance.
- COLOUR BLINDNESS: All 20 test persons are definitely not colour-blind.
- HIGHEST FINISHED EDUCATION: Two test persons have a doctorate diploma as their highest finished education, three test persons finished high school level, and 15 persons finished one or more academic study.
- PERSONAL COMPUTER USAGE: No test person has been using personal computers for less than ten years, the average being 20.6 years. The median value is 20 years. The weekly average is 44.5 hours of usage. 16 of the test persons use Windows as the operating system, two use Unix and two use Windows and Linux equally.
- INTERNET AND WEB USAGE: Ten test persons use the internet and web mostly at home, five at the office, three equally at home and the office and one person equally at home and on mobile devices. Answers between 0 and 55 hours of web usage per week were given, leading to an average of 19.9 and a mean value of 15 hours per week.
- BROWSER: 19 of the test persons use Firefox, or Firefox in combination with Google Chrome, Internet Explorer or Safari. One person uses Internet Explorer only.

- WEB ADMINISTRATOR EXPERIENCE: Eleven test persons have no experience as a web administrator, whereas nine do. The average is 3.4 years of experience; however, the mean value is zero years.
- USABILITY TEST EXPERIENCE: Every second test person had already been a test person in a usability test before and seven test persons had been part of a test team already.

6.3. TASK DESCRIPTION

In the following list, the translated tasks each test person got are found. The original tasks were provided to the test persons in German.

PREPARATION AND INTRODUCTION TASKS:

1 Please visit the following website: `http://find_frontv1`
You have a few minutes to browse around.

2 Introduction task in both versions: Please choose one facet at the lowest level in every category. For example PDF, ProE / PTC, etc.

TASK PAIR ONE:

3 Please type "engine" (in the original text in German, the term "Motor" was used) into the keyword search; 476 results will be found. Please find the result which shares the following facets by using the text-based faceted search interface only:

  - Status is *approved*
  - The tool in use is called *Catia-DS* from the category CAD construction
  - The author is Hermine Listig
  - And it is related to the module *engine*

4 Please type "engine" into the keyword search; 476 results will be found. Please find the result which shares the following facets by using the oval bubble-based faceted search interface only:

  - Status is *unknown*
  - The tool in use is called *Simulink* from the category CAE numerical simulation

  - The author is Daniel Düsentrieb
  - And it is from the project *C007*

TASK PAIR TWO:

5 Which people in Graz can be related to the CAE numerical simulation tools Abaqus und Nastran? Please find the result by using the oval bubble based faceted search interface.
6 Which people authored time schedules (project management) found in the source RQM in English? Please find the result by using the text-based faceted search interface.

TASK PAIR THREE:

7 Which product of the CAE numerical simulation tools is mostly used in the organisational unit E, if decision criterion is the number of results found using a tool. Please find the result by using the oval bubble based faceted search interface.
8 Which document type is most common in the organisational unit D? Please find the result by using the text-based faceted search interface.

6.4. TEST ENVIRONMENT

In this section, the test environment is described. Morae [TechSmith, 2014] software was used to record screen and user activity. Morae Recorder was installed on the test notebook. Morae Observer and Morae Manager were installed on the observer notebook. The notebooks used a connection via local area network (LAN).

All tests used the following test environment:

- BROWSER: The test persons used Google Chrome in version 28 to solve the tasks.
- OPERATING SYSTEM: Both notebooks run on Windows 7.
- WEBSITE HOST: The developed proof of concept implementation is hosted on a virtual machine on a server within the company Virtual Vehicle Research Center.
- CONNECTION: Ethernet cable.

- NOTEBOOK TEST PERSONS: The test notebook is an Acer Aspire 5745G, Intel Core i7 processor, 8GB RAM.
- TEST NOTEBOOK DISPLAY COLORS: 32 bit.
- TEST NOTEBOOK DISPLAY SIZE: 15.6" 16:9 HD LCD.
- TEST NOTEBOOK DISPLAY RESOLUTION: 1366 x 768.
- RECORDING SOFTWARE: Morae 3.2.1 .
- MICROPHONE: CREATIVE Fatal1ty Pro Series HS-800 Gaming Headset.
- VIDEO CAMERA: The test notebook built-in 1.3 Megapixel camera.
- OBSERVER NOTEBOOK: Fujitsu Siemens Lifebook E-Series, model E8420.
- DATES: 2013-09-03 (nine test persons), 2013-09-09 (seven test persons), 2013-09-10 (four test persons).
- PLACE: Virtual Vehicle Research Center, Inffeldgasse 21a, 8010 Graz, Austria, room 124 ground floor.

6.5. QUALITATIVE EVALUATION RESULTS

This section deals with qualitative results achieved in the evaluation. Thinking Aloud tests are a common method of testing usability. An audio and video recording system was in use, which automatically captures the test persons' screen and data, for example mouse movement or mouse clicks.

This section focuses on the test person statements captured by the audio-recording system during the tasks and the feedback interview. The records were reviewed and every relevant statement was added to a list of statements with some metadata information:

- The statement ID number.
- The user's name (anonymised).
- The timestamp in the recorded video.
- Which area of the information cockpit is meant.

In sum 175 statements were collected. In the next step statements were grouped according to the part of the cockpit mentioned. As a final step, statements which address an equal topic were counted to find out what test persons were mostly talking about and therefore had to think about. The idea behind it is that people often do not have to think about what to do if the system or the procedure is very intuitive. If one topic is mentioned several times by several test persons, then this can be an indicator that it is either not intuitive to use or very impressive.

Tables 6.1 and 6.2 present all statements which were mentioned more than three times. In the last column, the ID numbers of the single statements are listed. The full list of statements is found in Appendix B and Appendix C.

The main findings of Table 6.1 are discussed in the following:

It was stated 33 times that the facet hierarchy used in the text-based faceted search interface is inappropriately designed, as the following test person did after searching heavily for authors within the hierarchy:

> *"Maybe the hierarchy (Organisation - Ersteller / Engl.: organisation - author) is inappropriate."* [Table B.1, ID 54]

Another test person complained about label abbreviations:

> *"Abbreviations in the highest level are irritating, e.g.: PM can stand for Personalmanagement (Engl.: human resources), Projektmanagement (Engl.: project management), Produktmanagement (Engl.: product management),."* [Table B.1, ID 56]

There were 20 test persons producing 33 statements, there were test persons which mentioned this topic more than once. Single statements discuss the category labels and the hierarchical order in use. These statements confirm that a domain expert of the targeted user group has to perform or consult the hierarchy design, as mentioned in Section 2.2.1 already.

19 test persons stated that the facet text in the oval bubble faceted search interface is hardly readable; one test person mentioned:

> *"I had to focus the gaze heavily due to the size of the bubbles or the unsharpness"* [Table B.1, ID 2]

An HTML SVG text style configuration called *lengthAdjust: spacingAndGlyphs* is in use, what takes available space, the bubble size in the x axis, and the text as input, and stretches or compresses the glyphs (alphabetical characters) themselves and the space between them in one axis to fit the text into the available space. To avoid overlapping of glyphs, a minimum bubble size was guaranteed and a label text was restricted to an eleven character fraction to ensure a certain font size displayed in the bubble. The font size is influenced by the available space and the number of characters. As a result, a single character label uses the bubbles size and is displayed big, whereas an eleven character label is smaller. Test persons were confused when searching for a label, e.g. *Daniel Düsentrieb*, and only the label available was *Daniel Düse*. One test person thought that the names searched for were actually Andrea Engl instead of Andrea Engineer and Hermine List instead of Hermine Listig; see Table B.4, ID 152 or ID 93 in Table B.5. For the facet categories differentiation a colour palette was used, as shown in Figure 6.1b at facet *Dokumenttyp*, the dark font colour cannot be seen well on a dark colour. Test persons said:

> *"The text in the bubbles is hard to read."* [Table B.2, ID 36]

One implementation issue is that this also depends on the configuration and the ability of the display in use. Another problem mentioned is the font size of the facet labels, which becomes very small if the bubble size is very small. Lessons learned are that category colours and font colours should be coordinated and the minimum bubble size has to be larger than it actually is.

The oval bubble positioning algorithm was criticised by 15 test persons. One test person explicitly mentioned:

> *"The size difference is nice, but there should be a structure in the placement; otherwise, with more than 8 or 9 elements it gets confusing."* [Table B.2, ID 64]

Bubbles did overlap into other categories or overlapped with other bubbles if there were many bubbles. The reason why this happens is the force directed algorithm used, which positions the elements around a centre point in an iterative positioning calculation starting with huge moving steps and getting smaller iteratively. The calculation is influenced, amongst others, by attraction to the centre point and repulsion by other bubbles. The more bubbles, the longer it takes for the bubbles to find a positioning where no overlaps occur. However, the time for the initial positioning is restricted to about two seconds. Another problem is the zoom function and the display size of the browser. When the positioning algorithm is optimised for a single display size, the place for the bubbles is a fraction of the display size. If this changes, gaps between the categories may occur. The same happens if browser zoom is used. A lesson learned is that a fixed positioning order by the bubble size would probably have been easier to implement and easier for the test persons to understand .

Related to the bubble positioning statements mentioned previously, another 13 statements point out the oval bubbles: Test persons said if there are only a few bubbles, then it is not a problem to find the one searched for.

> *"I want to start with "status" because there are just three possibilities (bubbles) and therefore it is the easiest."* [Table B.1, ID 21]

Another said:

> *"If there are too many bubbles, then you have to start reading again, no advantage, but this might perform better after a learning phase."* [Table B.6, ID 120]

However, if a boundary number of elements is exceeded, then it confuses too much to be able to find the facet easily. This was also discussed by test persons, for example:

> *"More than 5 bubbles (maximum 7) per category leads to irritations or that it has to be read. Then it is nonsense."* [Table B.6, ID 126]

In the text-based faceted search interface, the hierarchy was contracted at the start to show the highest level only. A + symbol indicated that it is expandable; however, test persons had to click on the label to expand it as the + symbol did not trigger any action. To choose a facet, its checkbox had to be ticked, but test persons tried to click on the facet label to do that. Again, this did not trigger the considered action.

> *"Would like to be able to click on the facet name itself to select a facet."* [Table B.7, ID 1]
> *"Would like to be able to click on the plus symbol itself; would be intuitive."* [Table B.7, ID 11]

Lessons learned from those statements are regarding usability. Test persons intuitively expect a specific behaviour, which should be provided.

The last statement of Table 6.1 discussed is a positive one and was mentioned by seven test persons. They think they are faster when using the oval bubble faceted search interface than the text-based faceted search interface. In the text-based version, test persons had to expand the categories, where in the oval bubble version, they did not have to. An *expand all* button was provided in the text based version. A test person pointed out:

> *"Thought at the beginning the bubbles are rubbish or nonsense, but it works! Even if there are still some bugs. Surprised."* [Table B.6, ID 150]

related topic	**#**	**the statement**	**list of IDs**
textual facet hierarchy	33	Problems with the facet hierarchy: things are not found in the expected place or not named appropriately.	9, 17, 37, 41, 47, 49, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 76, 77, 78, 79, 80, 81, 88, 97, 101, 104, 106, 118, 128, 132, 142, 148, 155, 156
oval bubbles	19	Facet text is hard to read.	2, 12, 33, 36, 42, 82, 93, 111, 113, 117, 129, 134, 144, 147, 152, 157, 169, 171
o. bubbles	15	Oval bubble positioning could be better. E.g. overlaps to category texts and in between bubbles. A more structured view is needed.	22, 43, 48, 63, 64, 66, 67, 72, 99, 137, 143, 147, 162, 169, 170
o. bubbles	13	There is a number or a boundary of facets / bubbles where the visualisation of the facets starts to confuse the person searching.	8, 12, 21, 25, 27, 32, 63, 64, 67, 83, 85, 120, 126
textual f.h.	11	Problems with the facet hierarchy: any clicking that doesn't work, e.g. + symbol, click on facet text.	1, 11, 50, 51, 90, 102, 105, 138, 141, 167, 168
o. bubbles	7	Finding the facets is faster than in the textual version.	7, 16, 35, 86, 153, 164, 172

Table 6.1.: Table of most mentioned statements [1/2].

related topic	#	the statement	list of IDs
o. bubbles	6	Size difference detection is very hard, number of documents should be shown explicitly.	5, 14, 147, 158, 160, 161
o. bubbles	6	Provides a good overview or a better one than the textual version at the beginning.	4, 18, 149, 154, 155, 174
o. bubbles	5	Tooltip is hidden by the result list.	34, 99, 130, 135, 144
o. bubbles	5	After the tasks the user thinks the oval bubble version is useful and has advantages. They didn't expect that at the beginning.	19, 137, 146, 150, 173
textual f.h.	5	For (real) search tasks the textual facet hierarchy version is easier to use.	17, 44, 119, 124, 131
faceted search overall	4	Things were named differently in the tasks compared to the hierarchy; this was irritating.	52, 73, 73, 94
f. s. o.	4	It depends on the task which version performs better; both have advantages.	24, 38, 65, 125

Table 6.2.: Table of most mentioned statements [2/2].

6.6. QUANTITATIVE EVALUATION RESULTS

This section deals with quantitative results achieved in the evaluation. There are basically two data sources for the quantitative evaluation in this thesis:

- The captured action data while solving the tasks, e.g. mouse clicks, mouse movement, task times and the success rate of tasks and
- the numeric answers of the test persons in the feedback questionnaire.

6.6.1. THINKING ALOUD TEST RESULTS

This subsection presents quantitative results of the Thinking Aloud test. Task times of Thinking Aloud tests are not suitable for quantitative analysis due to additional time consumed when people are talking. It is therefore not discussed in the evaluation. Only the success distribution can be used.

However, the users were recorded with Morae software. The Morae software captured mouse movement, mouse clicks and time per task. Tasks were defined and reused for every test person. Markers for the start and end of a task were set. The score of a task was set. The screen the user looked at was captured. The face of the user was captured with the notebook's built-in camera, for example to analyse body language. The videos were used to see what a test person was talking about when arguing.

Tasks are organised in pairs; the pairs can be found in the task description in Section 6.3. For each pair, one task had to be solved with the textual faceted search and the other task with the visual faceted search. Two test persons had problems solving task three (using the text-based faceted search interface) and five (using the oval bubble faceted search interface), as shown in Table 6.3.

In task five, Elli should have searched for Abaqus and searched for Simulink instead. She also had problems finding the authors in the categories. Franz could not read the facet text of the bubble, because only the first eleven characters were shown and the tooltip showing the full label text was hidden. Franz could not find it immediately because the positioning algorithm placed it on the border of the category beneath.

Franz misunderstood the question in task seven. The decision criterion was said to be the number of documents and what was implicitly meant was the bubble size; nevertheless, he selected *pdf* at first and needed time to get back on the right path to the answer. Elli did the same and selected *pdf* for a few seconds, unchecked it again and then asked for help. She didn't realise that the bubble size represented the number of results sharing this facet and thought all bubbles were the same size at first.

	Task3	Task4	Task5	Task6	Task7	Task8
Success rate	100%	100%	90%	100%	90%	100%
Completed with difficulty	-	-	5%	-	10%	-
Failed to complete	-	-	5%	-	-	-

Table 6.3.: The task score success distribution of the test persons.

6.6.2. FEEDBACK RESULTS

This subsection presents the results of the feedback questionnaire test persons were asked to fill out after finishing the Thinking Aloud test. In the questionnaire, the test persons had the chance to rate using a Likert scale [Likert, 1932] from +3 to -3. Table 6.4 shows the results. Overall average rating of the overall cockpit impression is 2.2 out of 3 and therefore good. Test persons found the faceted search interface area easily on the website. The graphical design, readability and adjustment of the textual version were rated good.

The cockpit was rated to be consistent, which means it is harmoniously designed without contradictions. The performance in the test environment was good. Test persons expressed the preference for more visualisations in daily work. Also the answers indicate that visualisations make work easier and have a certain *fun* factor. The overall cockpit impression is very good.

On average, the test persons are not sure which version they prefer, but the textual version has a slight advantage. It seems that the test persons would like to use more visualisations in their daily work but they found some issues with the oval bubble version provided. For example, they rated it to be not easy to read and the layout algorithm is not clearly adjusted concerning positioning. This can be explained using the Thinking Aloud statements. Amongst others, the label fractioning to eleven characters and the small font size of small bubbles led to this rating.

question	overall avg. rating	std. dev.
Find and navigate to the faceted search interface area	2.3 (3=very easy)	0.85
Oval bubble version gives advantage to textual faceted search	1.0 (3=strong agree)	1.12
Easy to read in oval bubble version	-0.9 (3=very easy)	1.69
Easy to read in textual version	2.4 (3=very easy)	0.82
Oval bubble version is clearly adjusted (positioning)	0.5 (3=very clear)	1.54
Textual faceted search version is clearly adjusted	1.6 (3=very clear)	1.31
Graphical design of the cockpit, for example colours and icons	1.9 (3=perfect)	1.09
Consistency of the whole cockpit (contradiction free)	2.0 (3=consistent)	0.56
Performance, quickness (cockpit)	1.7 (3=very fast)	1.18
I would prefer more visualisations in daily work	1.5 (3=strongly agree)	1.36
Visualisations make daily work easier	1.5 (3=strongly agree)	1.19
Visualisations provide a certain *fun* factor	1.5 (3=strongly agree)	1.10
I prefer ... (-3 oval bubble version, +3 textual faceted search)	0.3 (3=textual)	1.80
Overall cockpit impression	2.2 (3=perfect)	0.67

Table 6.4.: Table of feedback data from the test persons.

6.7. LESSONS LEARNED

This section discusses lessons learned from the evaluation.

The Thinking Aloud test highlights important issues of the designed approach using oval bubbles to represent facets:

- The layout algorithm which is responsible for the positioning was chosen due to a size-efficient placing. However, as user statements show, it confuses the user that there is no ordering.
- Colours were chosen to be gentle so as not to distress the user in long-term usage. User statements reveal that the facet text is hard to read and one problem might be that a dark font is more difficult to read on darker background colours. Readability is of high relevance, e.g. to people with failing eyesight [Holzinger et al., 2011].
- Again, to ensure size efficiency, facet labels were placed within a bubble. As the bubble size depends on the number of results sharing the facet, the available place might be big or small. As user statements indicate, if bubbles are small, then the facet text is hardly readable. The idea to only show an eleven-character fraction of the label to ensure a minimum font size also led to confusion in a few cases.
- The bubble size should represent the number of results sharing this facet. Due to a variable font size of the label on the bubble, some test persons were confused and stated they thought the bubble was larger because of the label. Therefore, the bubble size did not gain the impact it should have had. The font size probably should be the exact same for every bubble and in an easily readable size. The bubble size differences have to be bigger. A square root function was used to calculate bubble sizes, to not have bubbles as big as the whole area. Probably the number of bubbles shown per category should be limited and a *show more..* button provided.

Respecting the issues, a mockup for a future implementation is displayed in Figure 6.3. The dark colour of the category *Status* is replaced by a lighter one. The bubbles are ordered according to size, starting with the biggest size on the top left, getting smaller to the right. After every three elements, a new row is added. Bubble labels are all of the same font size. Labels which are too long use an abbreviation and the full name is shown in a tooltip on mouse-over. No overlapping bubbles and no overlaps into other categories. This mockup answers the research question on how the proof of concept could be implemented, if findings of the evaluation are respected.

Figure 6.3.: Mockup of an improved oval bubble faceted search interface.

The facet hierarchy in use caused several test persons' displeasure. The hierarchy was introduced early in the development phase by an expert and was not changeable later. An iterative process to find a proper hierarchy should be used. However, the problems led to difficulties finding facets in the hierarchy, especially in the text-based faceted search interface.

7. CONCLUSION

7.1. SUMMARY

In this master's thesis, a user interaction concept for an information cockpit in virtual development is introduced. A proof of concept implementation is evaluated. As visualisations raised attention during literature research, the focus of the evaluation is on the different faceted search interfaces implemented, to find out which version is preferred and why. Additionally, user statements for all parts in use were selected, which can be used as input for further development. The evaluation reveals issues and non intuitive behaviour. The feedback questionnaire shows that users would prefer more visualisations in their daily work. On average, of all ratings, for example, the overall information cockpit impression and the graphical design were rated very high. The implemented oval bubbles is rated that it provides an advantage, but the readability of the facet text is definitely rated negatively.

In summary, one possible interpretation is that for interactive visualisations, an outstanding usability is needed to raise an acceptance level for the willingness of usage wanted by the concept designer. Users are aware of the advantages; however, the barrier for a change in behaviour is at a higher level than expected. Looking back to the implementation phase, instead of implementing a new visualisation for the faceted search based on an available library, taking multiple ready-to-use implementations and adopting them to the use case for a comparing evaluation could have been sufficient. If that had been the case, a version with outstanding usability could possibly be recommended now.

This master's thesis starts with an introduction in Chapter 1, motivating the topic, describing the environment and presenting the research question.

Related work about visual search interfaces can be found in Chapter 2. Visual information processing and faceted search interfaces are discussed.

A mockup of the concept for an information cockpit was drawn and discussed in Chapter 3. The concept was presented and discussed at the International Conference on Web Information Systems and Technologies (WebIST) in Aachen 2013.

The concept is the basis of the proof of concept implementation presented in Chapter 4. It was implemented focusing on aspects of the faceted search interface and result view. Requirements are listed and their completion is referenced. The proof of concept architecture and the development method are explained. Components of the information cockpit implementation are discussed in detail.

The case study in Chapter 5 guides the reader through the information cockpit usage.

Chapter 6 presents the evaluation of the implemented concept. 20 test users were questioned to fulfil search tasks using different versions of the faceted search in a Thinking Aloud test. Quantitative as well as qualitative evaluation results were collected and are presented. In the discussion at the end of the chapter a mockup of an improved version is provided.

7.2. FINDINGS REGARDING THE RESEARCH QUESTIONS

Research question number one asks: *How can visualisations add an advantage within the context of enterprise search interfaces?*

As mentioned in Section 3.3, the concept bears the usage of state of the art visualisations for a faceted search and for the result view in mind. Visualisations can use advantages of human perception and may therefore lead to an increase in performance as pointed out in Sections 2.1.3 and 2.2.3.

This answers the research question regarding on how visualisations can add an advantage within the context of enterprise search interfaces.

Research question number one asks: *How could a proof of concept be implemented?*

Chapter 4 describes the proof of concept implementation of the information cockpit in this thesis.

To answer how especially visualisations which add an advantage in enterprise could be implemented in a proof of concept visualisation, in Section 4.4.5 the following is stated:

The second version is an experimental implementation based on considerations of human perception strengths. The number which represents the number of results sharing a facet, is depicted as the size of an oval bubble, as detecting size differences of objects can be done pre-attentively, an example of which is shown in Figure 4.10. Due to the oval bubble form and the advantage of the layout algorithm used, a larger number of elements than in the text-based version can be displayed without a need for scrolling, like the elements of the category *date* in Figure 4.10. This addresses the research question regarding on how visualisations can add an advantage within the context of enterprise search interfaces.

After evaluating the proof of concept implementation, a mockup of an improved version is presented in Section 6.7 together with a list of lessons learned.

7.3. OUTLOOK

The concept developed within the thesis is a key factor in the project within which the thesis was completed. The insights gained into the concept, the proof of concept implementation and the evaluation serve as basis for further development. The next step is to use real enterprise search engines to run queries. This could be used to perform long-term evaluations with simulation engineers. The concept of the cockpit will be kept. The faceted search interface will be reimplemented into a version using both aspects, facets represented as a text and the number of results sharing the facet as a visualisation, for example by the length of a bar. Usability aspects test users mentioned will be reviewed and adopted for the next proof of concept implementation. As two versions of the result view exist, this would be another research topic ready for evaluation. Within the next version, the challenging topic about combining result views from structured and unstructured data sources provided by different search engines will be addressed. In the topic of system architecture design of next versions, heavy processing components have to be carefully placed or could follow a singleton design pattern to process queries in sequence as in the *APA Labs platform* [Kienreich et al., 2008].

For me personally, I think the topics human computer interaction, usability and information retrieval in combination with engineering and information management are exactly the field of research I am interested in and exactly what I will keep my focus on. I have learned how different engineers from the same department may think and behave, which is one reason why developing interaction concepts or information systems is so challenging. Therefore, humans have to be the centre of every concept. However, when concepts are implemented, requirements and specifications may be interpreted differently by developers involved. As a conclusion, many pitfalls exist when designing and implementing a product and consequently, this will be an ongoing research topic.

It would be interesting to redo the evaluation with a hierarchy fitting perfectly for a target group and a different positioning algorithm, for example the algorithm sketched in the mockup displayed in Figure 6.3.

Another option would be to substitute the oval bubbles by tag clouds, where the font size represents the number of results.

Appendix

Appendix A.

JSON RESPONSE EXAMPLE


```
{
  "numberOfResults": 1,
  "sort": "created+",
  "results": [
  {  "id": 0,
     "title": " international Kreisky",
     "snippet": " 1930er-Jahren Europa den allen Standorte Innovationsgrad E-<b>Motor</b> Fal
     "largeSnippet": " Heute Auf und in 1930er-Jahren Innovationsgrad E-<b>Motor</b> Fahrzeu
     "url": "file:\\\\C:\\54426659",
     "facets": [
     {  "entry": "Dokumenttyp",
        "numberOfArtefacts": 0,
        "children": [
        {  "entry": "MS Word",
           "numberOfArtefacts": 1,
           "children": []
        }]
     },
     {  "entry": "Organisation",
        "numberOfArtefacts": 0,
        "children": [
        {  "entry": "Datum",
           "numberOfArtefacts": 0,
           "children": [
           {  "entry": "2007-05-19 14:07",
              "numberOfArtefacts": 1,
              "children": []
```

```
      }]
    },
    {  "entry": "Ersteller",
       "numberOfArtefacts": 0,
       "children": [
       {  "entry": "Hermine Listig",
          "numberOfArtefacts": 1,
          "children": []
       }]
    },
    {  "entry": "Organisationseinheit",
       "numberOfArtefacts": 0,
       "children": [
       {  "entry": "C",
          "numberOfArtefacts": 1,
          "children": []
       }]
    },
    {  "entry": "Standort",
       "numberOfArtefacts": 0,
       "children": [
       {  "entry": "Graz",
          "numberOfArtefacts": 1,
          "children": []
       }]
    }]
},
{  "entry": "PM",
   "numberOfArtefacts": 0,
   "children": [
   {  "entry": "Anforderungen",
      "numberOfArtefacts": 1,
      "children": []
   }]
},
{  "entry": "Produkt",
   "numberOfArtefacts": 0,
   "children": [
   {  "entry": "Modul",
```

```
      "numberOfArtefacts": 0,
      "children": [
      {  "entry": "Türen / Klappen",
         "numberOfArtefacts": 1,
         "children": []
      }]
    },
    {  "entry": "Projekt",
      "numberOfArtefacts": 0,
      "children": [
      {  "entry": "C007",
         "numberOfArtefacts": 1,
         "children": []
      }]
    }]
  },
  {  "entry": "Quelle",
    "numberOfArtefacts": 0,
    "children": [
    {  "entry": "TDM",
       "numberOfArtefacts": 1,
       "children": []
    }]
  },
  {  "entry": "Sprache",
    "numberOfArtefacts": 0,
    "children": [
    {  "entry": "Deutsch",
       "numberOfArtefacts": 1,
       "children": []
    }]
  },
  {  "entry": "Status",
    "numberOfArtefacts": 0,
    "children": [
    {  "entry": "freigegeben",
       "numberOfArtefacts": 1,
       "children": []
    }]
```

```
    }]
  }],
  "globalFacets": [
  {  "entry": "Dokumenttyp",
     "numberOfArtefacts": 0,
     "children": [
     {  "entry": "MS Word",
        "numberOfArtefacts": 1,
        "children": []
     }]
  },
  {  "entry": "Organisation",
     "numberOfArtefacts": 0,
     "children": [
     {  "entry": "Datum",
        "numberOfArtefacts": 0,
        "children": [
        {  "entry": "2007-05-19 14:07",
           "numberOfArtefacts": 1,
           "children": []
        }]
      },
      {  "entry": "Ersteller",
         "numberOfArtefacts": 0,
         "children": [
         {  "entry": "Hermine Listig",
            "numberOfArtefacts": 1,
            "children": []
         }]
      },
      {  "entry": "Organisationseinheit",
         "numberOfArtefacts": 0,
         "children": [
         {  "entry": "C",
            "numberOfArtefacts": 1,
            "children": []
         }]
      },
      {  "entry": "Standort",
```

```
      "numberOfArtefacts": 0,
      "children": [
      {  "entry": "Graz",
         "numberOfArtefacts": 1,
         "children": []
      }]
    }]
  },
  {  "entry": "PM",
     "numberOfArtefacts": 0,
     "children": [
     {   "entry": "Anforderungen",
         "numberOfArtefacts": 1,
         "children": []
     }]
  },
  {  "entry": "Produkt",
     "numberOfArtefacts": 0,
     "children": [
     {  "entry": "Modul",
        "numberOfArtefacts": 0,
        "children": [
        {  "entry": "Türen / Klappen",
           "numberOfArtefacts": 1,
           "children": []
        }]
     },
     {  "entry": "Projekt",
        "numberOfArtefacts": 0,
        "children": [
        {  "entry": "C007",
           "numberOfArtefacts": 1,
           "children": []
        }]
     }]
  },
  {  "entry": "Quelle",
     "numberOfArtefacts": 0,
     "children": [
```

```
            {  "entry": "TDM",
               "numberOfArtefacts": 1,
               "children": []
            }]
         },
         {  "entry": "Sprache",
            "numberOfArtefacts": 0,
            "children": [
            {  "entry": "Deutsch",
               "numberOfArtefacts": 1,
               "children": []
            }]
         },
         {  "entry": "Status",
            "numberOfArtefacts": 0,
            "children": [
            {  "entry": "freigegeben",
               "numberOfArtefacts": 1,
               "children": []
            }]
         }]
}
```

Appendix B.

TEST PERSON STATEMENTS ON THE FACETED SEARCH INTERFACE

Oval bubble faceted search interface [1/6]				
ID	**TP#**	**Alias**	**statement**	**video timestamp**
2	TP10	"Erwin"	User had to focus the gaze heavily due to the size of the bubbles or the unsharpness	03:53:00
8	TP10	"Erwin"	The bubbles should be more sharp	14:05:00
12	TP13	"Julian"	Some bubbles are hardly readable, minimum bubble size is too small sometimes and for a category with lots of elements (e.g. datetimes) the visualisation is confusing	13:24:00
21	TP16	"Marion"	Says she wants to start with "status" because there are just three possibilities (bubbles) and therefore it is the easiest	02:54:00
22	TP16	"Marion"	some ellipsen-texts are hard to read due to an overlap with category-texts	04:27:00
27	TP16	"Marion"	If there are just a few facets, then identifying which bubble is bigger is much easier. As tradeoff the amount of the facet should be displayed in the tooltip	10:11:00
32	TP9	"Nico"	didn't see if anything has changed	02:17:00

Table B.1.: Test person statements to the oval bubble faceted search interface [1/6].

Oval bubble faceted search interface [2/6]				
ID	**TP#**	**Alias**	**statement**	**video times-tamp**
34	TP9	"Nico"	The tooltip is hidden by the result list	03:59:00
36	TP9	"Nico"	the text in the bubbles is hard to read	07:26:00
42	TP18	"Ernst"	it is hard to read the facet-text	12:16:00
63	TP15	"Christian"	if there are lots of facets, then it is very hard to choose something particular, e.g.: For "Organisationseinheit" the possibilities are "A", "B", "C", ... and therefore the user expects to find it alphabetically ordered	13:44:00
64	TP15	"Christian"	the size difference is nice, but there should be a structure in the placement, otherwise with more than 8 or 9 elements it gets confusing	14:27:00
66	TP15	"Christian"	overlapping bubbles doesn't make it easier	15:35:00
67	TP15	"Christian"	the bubbles don't scale, it is confusing without a structuring	16:13:00
72	TP8	"Elli"	she didn't find the authors in the facet list in the whole task and didn't see that there is pagination	03:27:00 & 05:13:00
82	TP19	"Thomas"	long names make it difficult	08:10:00
83	TP19	"Thomas"	if there are lots of facets, then searching for a particular one is hard	08:20:00
84	TP19	"Thomas"	the size difference of the bubbles representing the amount of results sharing this facet is "super"	08:35:00
85	TP19	"Thomas"	For the date a different solution is needed	08:41:00
91	TP1	"Stuart"	irritated why "Simulink" is found under "Methods". The category name is: Methode: CAE / numerische Simulation	04:10:00
92	TP1	"Stuart"	lots of scrolling is needed	05:58:00
99	TP1	"Stuart"	some overlaps are irritating but that's OK for a prototype	13:20:00

Table B.2.: Test person statements to the oval bubble faceted search interface [2/6].

Oval bubble faceted search interface [3/6]				
ID	**TP#**	**Alias**	**statement**	**video timestamp**
100	TP1	"Stuart"	the colors are remembered very fast, they help to recognize where to find which category, recognition value of the categories is way higher, could be added to the textual version too	13:46:00
111	TP14	"Rene"	there could be a problem if some facets start with the same letters, hard to distinguish	14:00:00
113	TP21	"John"	facet text is hard to read	02:57:00
114	TP21	"John"	bug: if all results have the same facet of an unchosen category, then it doesn't make sense to re-show the other facets after clicking the one	03:37:00
129	TP11	"Oliver"	some ellipses are quite small, hard to read	03:24:00
130	TP11	"Oliver"	tool-tip is hidden by the result list	04:40:00
133	TP7	"Lukas"	on the first view it is missing an overview because you have to scroll and the category-names are hard to read	03:11:00
134	TP7	"Lukas"	the facet text is hard to read	03:38:00
135	TP7	"Lukas"	tool-tip is hidden by the result list	04:50:00
137	TP7	"Lukas"	the visual facet search is new to him, seems confusing on the first view, the placement could be optimized, but seems very useful	07:41:00
143	TP20	"Wolfgang"	text overlaps are suboptimal	03:08:00
144	TP20	"Wolfgang"	Has to search for "Daniel Düsentrieb" and thinks it is an failure because the letters in the bubble say: "Daniel Düse". He can't see the tool tip due to the result-list overlap	03:17:00
147	TP20	"Wolfgang"	what should be better: text overlap, if background-color of ellipse is dark then the text needs a lighter color to be able to read, to detect the size difference (which one is larger) is very hard (my opinion: maybe due to the logarithmic scale or the elliptic form instead of circles)	07:07:00

Table B.3.: Test person statements to the oval bubble faceted search interface [3/6].

Oval bubble faceted search interface [4/6]				
ID	**TP#**	**Alias**	**statement**	**video timestamp**
152	TP17	"Sebastian"	thinks the persons names are Engl and List	03:21:00
157	TP6	"Josef"	hard to read the facet text. Too much text	04:00:00
158	TP6	"Josef"	Would be nice to have the number of documents of the facet would be explicitly shown	07:50:00
160	TP6	"Josef"	wasn't told that the bubble size is related to the number of results found. At the end he got it	08:45:00
164	TP12	"Andreas"	it is faster	02:47:00
169	TP12	"Andreas"	doesn't like the overlaps of the ellipses and that the words are hard to read	10:07:00
170	TP12	"Andreas"	doesn't like the ordering of the bubbles (not structured)	10:14:00
171	TP12	"Andreas"	easy to click, but hard to read	10:14:00
3	TP10	"Erwin"	User thought that the facets that emerge in the same category after a click are subcategories	05:20:00
14	TP3	"Johann"	did not expect that the bubble size is related to the number of results that share this facet and was confused that the number of documents and not the number of results was asked	08:16:00
48	TP2	"Franz"	User thinks there are no facets in this category because the facets are placed beneath in the next category	05:02:00
74	TP8	"Elli"	thinks that all bubbles of the category CAE numerische Simulation are of the same size. She thought the bubble Simulink is slightly bigger due to the length of the word "Simulink"	09:29:00
75	TP8	"Elli"	She thought that bubbles are bigger in size if more space for the word is needed	11:32:00

Table B.4.: Test person statements to the oval bubble faceted search interface [4/6].

Oval bubble faceted search interface [5/6]				
ID	**TP#**	**Alias**	**statement**	**video timestamp**
93	TP1	"Stuart"	thinks the names are Andrea Egl and Hermine List	06:40:00
112	TP21	"John"	thought when he clicks on a bubble, then the subcategories emerge, therefore he is confused because he doesn't know which of the bubbles are the subcategories of one bubble...	02:07:00
117	TP21	"John"	thinks the names are Andrea Egl and Hermine List	10:10:00
7	TP10	"Erwin"	With the bubbles someone can easily choose facets really fast	13:53:00
16	TP3	"Johann"	performs good for overview-questions, e.g. "Which product of the numeric simulation .." due to the fact that the textual facet search needs lots of clicking and searching and the numbers are hard to read and compare	09:40:00
18	TP3	"Johann"	better overview on the visual facet search, but he thinks that some facets and categories are missing	10:15:00
19	TP3	"Johann"	extremely positive event: he didn't believe that there is an advantage of the visual facet search variant before	11:05:00
30	TP9	"Nico"	Have to remember where to find what	01:50:00
31	TP9	"Nico"	thinks the visual facet search needs a bit longer to react	02:01:00
33	TP9	"Nico"	has to search longer for identifying the right name in the bubbles	02:30:00
35	TP9	"Nico"	if the positions of the categories are known then the navigation to the wanted category is faster with the bubbles	07:13:00

Table B.5.: Test person statements to the oval bubble faceted search interface [5/6].

Oval bubble faceted search interface [6/6]				
ID	**TP#**	**Alias**	**statement**	**video times-tamp**
43	TP18	"Ernst"	the visual facet search is not appropriate for him, tag-clouds would be easier due to a certain ground-structure they have. The visual perception is different within persons	19:47:00
46	TP18	"Ernst"	it is not bad/has an advantage to give the possibility of a visual version of the facet search because people are different and the next one might prefer the visual version	23:28:00
95	TP1	"Stuart"	noticeable is that in the visualization more scrolling is needed but at the beginning more things can be seen	11:43:00
96	TP1	"Stuart"	with only one look, how the data is spread and what is going on in the data can be identified	12:04:00
120	TP21	"John"	if there are too much bubbles, then you have to start reading again, no advantage, but this might perform better after a learning phase	13:55:00
126	TP21	"John"	more than 5 bubbles (maximum 7) per category leads to irritations or that it has to be read. He says then it is nonsense	20:10:00
127	TP21	"John"	he thinks that displaying 5 organisational units in bubbles, each with just a single letter, doesn't make sense because you are used to it in a different way	20:34:00
146	TP20	"Wolfgang"	the visual facet search is better than expected at the beginning, but some details have to be fixed	07:00:00
150	TP20	"Wolfgang"	thought at the beginning the bubbles are rubbish or nonsense, but it works! Even if there are still some bugs. Surprised	08:44:00
162	TP6	"Josef"	is not sure if a tree structure is appropriate for the bubbles or if the place where it is should be bigger. Problem is to find a special bubble, because so much text has to be read. Maybe it should increase the size if is mous-overed	11:00:00

Table B.6.: Test person statements to the oval bubble faceted search interface [6/6].

Text based faceted search interface [1/5]				
ID	**TP#**	**Alias**	**statement**	**video timestamp**
1	TP10	"Erwin"	Would like to be able to click on the facet name itself to select a facet	01:33:00
11	TP13	"Julian"	Would like to be able to click on the plus-symbol itself, would be intuitive	13:16:00
26	TP16	"Marion"	if every category is opened, then a lot of scrolling is needed. With the bubbles less scrolling is needed	09:31:00
41	TP18	"Ernst"	irritated that persons are not found on the highest tree-level or that it is asked for "person" and in the tree it is known as "Ersteller"	09:55:00
47	TP2	"Franz"	it is hard to find the category "Modul"	02:13:00
51	TP15	"Christian"	clicks on the name for checking the checkbox what doesn't change anything	05:14:00
54	TP15	"Christian"	maybe the hierarchy (Organisation - Ersteller) is inappropriate	07:49:00
55	TP15	"Christian"	Category name is "Dokumenttyp": Some members are called like the tool name (MS Powerpoint) and some like the file-extension (PDF); what's the difference between word an text	08:15:00
56	TP15	"Christian"	Abbreviations in the highest level are irritating, e.g.: PM can stand for Personalmanagement, Projektmanagement, Produktmanagement,	08:55:00
78	TP19	"Thomas"	some knowledge about the category hierarchy is good to fulfil the tasks. Or it has to be searched	06:23:00
80	TP19	"Thomas"	"Terminplan" fits better to category "Datum"	07:10:00
88	TP5	"Martin"	knowledge about category hierarchy needed	

Table B.7.: Test person statements to the oval bubble faceted search interface [1/5].

Text based faceted search interface [2/5]				
ID	**TP#**	**Alias**	**statement**	**video timestamp**
90	TP1	"Stuart"	it might happen to check all members of the subcategory because it is not recognized that there is a subcategory or that more than one thing was checked	02:21:00
102	TP14	"Rene"	no action on the intuitive click on the + symbol	00:30:00
103	TP14	"Rene"	no progress bar or something similar which shows that the system still works or is in progress	01:00:00
104	TP14	"Rene"	confused that the Tool Catia is found under "Methode"	04:33:00
105	TP14	"Rene"	bug: when a whole category is chosen but then some members are unchecked the checked checkbox of the category should be unchecked	04:45:00
118	TP21	"John"	some knowledge about the category hierarchy is good to fulfil the tasks. Or it has to be searched	13:30:00
132	TP7	"Lukas"	searched for the author first in the categories "Quelle" and then "PM" and said it is non-logical to find it under "Organisation"	02:00:00
138	TP7	"Lukas"	works quite well already, the + symbol should be clickable	07:56:00
141	TP20	"Wolfgang"	+ Symbol	01:18:00
142	TP20	"Wolfgang"	to jump to the Word needed (e.g. Catia) he would use the browser-search	01:32:00
148	TP20	"Wolfgang"	a never-ending list where scrolling leads to death. A filter-input field or a textual search-field just for the facet-list would be good.	07:45:00
163	TP6	"Josef"	you don't know what already was checked! Very negative	11:55:00

Table B.8.: Test person statements to the oval bubble faceted search interface [2/5].

Text based faceted search interface [3/5]				
ID	**TP#**	**Alias**	**statement**	**video timestamp**
167	TP12	"Andreas"	mouseover on a text (on the very last level) shows a "hand" symbol, but clicking doesn't change anything	09:55:00
168	TP12	"Andreas"	should be able to click on the "+" and "-" symbol	10:03:00
9	TP13	"Julian"	Unclear that Catia DS is a tool which will be found under "Methode", user states that "CAD is no method"	01:02:00
39	TP18	"Ernst"	it is not clear if checking a facet-checkbox leads to an AND or OR junction, what is a big problem, because the user has to combine the things in his brain	00:34:00
57	TP15	"Christian"	In category PM some things are missing: "Ziele", "Status" - a project has a start point and an endpoint and lots of gates in between	09:23:00
58	TP15	"Christian"	to find the subcategory "Projekt" in the category "Produkt" doesn't make sense. A project can lead to a product, but then the project would be the master-category of product. Maybe "Produkt" doesn't mean the product itself, but the sum of components. "Projekt" was expected to be found under "PM".	10:30:00
59	TP15	"Christian"	Generic names for the sources are OK for a prototype, but in case of a product this should be rethought. A Wiki might exist in a company but most people might not identify it as a Wiki (more as intranet platform). PDM is Produkt-Daten-Management and therefore the management and not the system	11:25:00

Table B.9.: Test person statements to the oval bubble faceted search interface [3/5].

Text based faceted search interface [4/5]				
ID	**TP#**	**Alias**	**statement**	**video times-tamp**
60	TP15	”Christian”	Typically a following ”S” is used for things like PDM, TDM, etc. if the system is meant	11:52:00
61	TP15	”Christian”	With ”Sprache” it is unclear if the language of the result or the language of a person (searched for) is meant	12:13:00
62	TP15	”Christian”	”Status” is an own high-level category and therefore unclear if it is the status of the documents found or what? Maybe a root-category ”Dokument” is needed where the type, the language and the status is found	12:50:00
77	TP19	”Thomas”	is confused why ”Terminplan” is found under ”PM”	04:05:00
79	TP19	”Thomas”	is concerned if everybody would suggest to find ”Motor” as a ”Modul” und the category ”Produkt”. He thinks that under ”Modul” lots of things can appear, it is not an identifier, e.g.: Software-Modul, Excel-Modul	06:30:00
106	TP14	”Rene”	unclear what a ”Modul” is in the environment	05:55:00
4	TP10	”Erwin”	Checking which persons do have results is ”cooler” with the visual faceted search	09:44:00
10	TP13	”Julian”	the faceted search might be too confusing due to the bigness such categories can get in a real company	11:52:00
15	TP3	”Johann”	The last task has the advantage that after the visual task before it is very clear what is to do	09:20:00
17	TP3	”Johann”	For real search tasks the faceted navigation (textual facet search) was easier, but the facet hierarchy (the tree structure) has to be learned before	10:02:00

Table B.10.: Test person statements to the oval bubble faceted search interface [4/5].

Text based faceted search interface [5/5]				
ID	**TP#**	**Alias**	**statement**	**video timestamp**
81	TP19	"Thomas"	Status, Sprache, Quelle, Projekt and the rest are clear, or distinct, Modul and PM not and in Dokumenttyp it should be consistent (Tool or file extension)	07:35:00
97	TP1	"Stuart"	main categories are seen at the beginning, a guessing game what aspects are hidden in which category starts	12:15:00
151	TP20	"Wolfgang"	he knows a tool where it is possible to write some kind of primitive SQL input, the system is able to parse text, could be interesting for us	09:15:00

Table B.11.: Test person statements to the text based faceted search interface [5/5].

Overall faceted search interface [1/5]				
ID	**TP#**	**Alias**	**statement**	**video timestamp**
28	TP16	"Marion"	a reset-button is missing to start a new search. Clicking F5 is not a solution	11:10:00
29	TP16	"Marion"	an information which tells which facets are already chosen is missing	11:55:00
50	TP2	"Franz"	clicking on the "+" Symbol should be possible,	14:10:00
69	TP15	"Christian"	a basic structure is given in the textual lists	17:45:00
76	TP8	"Elli"	some knowledge about the category hierarchy is needed to fullfill the tasks. Or it has to be searched	11:58:00
87	all		lots of people have problems to find "persons" in the hierarchy	
89	TP5	"Martin"	if the task is not that clear because it is provided by somebody else and not by myself, then the text has to be read very carefully	10:40:00
101	TP1	"Stuart"	the hierarchy structure sometimes seems not logical, PM and Produkt especially	14:21:00
145	TP20	"Wolfgang"	In an overall perspective the things are found really fast	06:56:00
149	TP20	"Wolfgang"	The visual facet search is way more compact. The textual version would be 5-8 display-screens long if all categories are opened. If somebody is new to it he wouldn't find anything in the list	08:14:00
155	TP17	"Sebastian"	learning the working environment is easier with the visual version, but once the tree of the textual version is known then it is equal	07:23:00
161	TP6	"Josef"	was quite easy to find the answers, but there are some specials that has the found out (bubble size, read the bubble text is hard)	10:33:00
174	TP12	"Andreas"	the time-advantage of the visual version is: you don't have to open the categories, because it is already open	11:00:00

Table B.12.: Test person statements to the overall faceted search interface [1/5].

Overall faceted search interface [2/5]				
ID	**TP#**	**Alias**	**statement**	**video timestamp**
49	TP2	"Franz"	questions and cockpit functions were clear, but not the question where a "Produkt" was searched but the name "Produkt" was not in there	13:18:00
52	TP15	"Christian"	It was confusing that the mentioned things in the questions have different names in the cockpit. The questions were weakly / softly verbalized what lead to an interpretation-process	05:36:00
53	TP15	"Christian"	Is the "Ersteller" the "Verfasser" or the person searched for?	06:45:00
73	TP8	"Elli"	2 persons did click on "PDF" when it was asked to search for the highest amount of documents and said to search for elements	08:26:00
94	TP1	"Stuart"	confused because a "Produkt" is asked but he sees Simulink as a "Tool"	09:10:00
98	TP1	"Stuart"	sometimes you have to think where to find the information searched for (in the result view or in the facets)	12:34:00
128	TP21	"John"	CAE and CAD is more and more mixed in recent time, FEM are in Catia, ... confusing!	21:18:00
156	TP6	"Josef"	non-logical to find Catia DS under methods. Above is the category "Dokumenttyp" where he saw lots of tools (like MS Word), therefore he expected the "tool" Catia to be found there	02:00:00
5	TP10	"Erwin"	In Task 8 he prefers the textual version to the visual one due to a big diverse: 2 or 86 documents, maybe the bubble diversity would be bigger when the numbers are more diverse. For exact identifications the textual version is better	12:26:00

Table B.13.: Test person statements to the overall faceted search interface [2/5].

Overall faceted search interface [3/5]				
ID	**TP#**	**Alias**	**statement**	**video timestamp**
6	TP10	"Erwin"	It is easy to learn the usage of both versions very fast	13:02:00
13	TP13	"Julian"	The space reserved for the faceted search could be doubled due to the importance of the part, maybe the part for saving results could be a "context-menu" or results could be displayed in a pop-up window	13:57:00
20	TP3	"Johann"	from a user-interface view the space reserved for the faceted search could be bigger	11:22:00
24	TP16	"Marion"	Some tasks are better solved with the visual version (better overview) and some with the textual version	08:38:00
25	TP16	"Marion"	if there are just a few facets then the visual facet search might perform better because focusing on one is fast; with lots of facets the visual facet search looses the overview (e.g.: datetime).	08:50:00
37	TP9	"Nico"	a good knowledge of the facet categories and what is found within which category is needed	08:11:00
38	TP9	"Nico"	providing a faceted search additional to the text-search is very helpful	08:30:00
44	TP18	"Ernst"	he is a classifier and the classic tree structure is perfect for him. A visual version would be nice but it would have to be perfect to be accepted from him	21:58:00
65	TP15	"Christian"	A combination of visual and textual search can be the right way	15:10:00
68	TP15	"Christian"	Colors are well chosen, high contrasts	17:25:00
71	TP15	"Christian"	the interactivity is very good, choosing aspects without a refresh/reload of the page. Not a feeling of filling out a form	19:28:00

Table B.14.: Test person statements to the overall faceted search interface [3/5].

Overall faceted search interface [4/5]				
ID	**TP#**	**Alias**	**statement**	**video timestamp**
86	TP19	"Thomas"	from his feeling the visual version is faster than the texutal version when fullfilling a task, due to the fact that things are identified faster	08:55:00
115	TP21	"John"	in a real-world scenario he would have filtered for the most (filtering) constricting element first	06:45:00
116	TP21	"John"	if last time the visual facet search was in use, then next time (after page refresh) the visual facet search should be displayed instead of the textual one	08:48:00
119	TP21	"John"	says it is easier for him with the textual version than the visual version	13:48:00
121	TP21	"John"	the question is how much people trust in the structured content. Do they believe that every possible result can be found? The same like with geizhals: I believe it shows me a cheap prize, but there might be a better prize somewhere. He uses geizhals as an indicator but goes to a store to check the price again (store usually is 20% higher). Did Hermine Listig save all her documents in the NT Share where it can be found?	14:46:00
124	TP21	"John"	prefers the textual facet search to the visual version, because it is common. He used tree-structures in textual form for many years	18:45:00
125	TP21	"John"	it might happen that the visual version is preferred, but this needs a learning phase (getting used to it) and the understanding of the advantages of the bubble size, the bubble-color and such things. It is like SAP Users: At the beginning very slow, after some time start to use shortcuts	18:52:00

Table B.15.: Test person statements to the overall faceted search interface [4/5].

Overall faceted search interface [5/5]				
ID	**TP#**	**Alias**	**statement**	**video timestamp**
131	TP11	”Oliver”	some tasks are easier to solve with the textual version, because it seems to be more organized,	08:12:00
153	TP17	”Sebastian”	visual version is faster	06:30:00
154	TP17	”Sebastian”	doesn’t matter in which order to select facets. Both versions (visual and textual) are equal operable but visual version provides a faster overview	06:42:00
172	TP12	”Andreas”	firstly he thought the bubbles are weird because they look like shit, but interestingly he now thinks he was faster or it is faster to work with	10:25:00
173	TP12	”Andreas”	if he had to choose he would use the visual version now	10:40:00
107	TP14	”Rene”	if it is used for a longer time then it can be useful	11:45:00

Table B.16.: Test person statements to the overall faceted search interface [5/5].

Appendix C.

TEST PERSON STATEMENTS ON OTHER INFORMATION COCKPIT AREAS

Result visualisation				
ID	**TP#**	**Alias**	**statement**	**video timestamp**
45	TP18	”Ernst”	he likes the visualisation that represented the results, maybe change the colors, but there the relations get way clearer	22:45:00
110	TP14	”Rene”	after he saw the result visualisation he was satisfied, this would be appropriate	12:55:00

Table C.1.: Test person statements to the result visualisation.

Result list				
ID	**TP#**	**Alias**	**statement**	**video times-tamp**
109	TP14	"Rene"	didn't recognize the pagination	12:23:00
23	TP16	"Marion"	the result list provides pagination, but this isn't clear to the users what leads to confusion why all results are from person A but in the facet-list another person as author is available	05:05:00
159	TP6	"Josef"	didn't see the pagination or the information that says the number of results	08:05:00
70	TP15	"Christian"	it has to be rethought if author, datetime and project must be shown on the first view, or the link to the document. He expects that a click on the icon leads to a download,	18:22:00
108	TP14	"Rene"	would be good to have an additional filter for the result list, e.g. list of all authors, because on the left side there are all the other informations	11:56:00

Table C.2.: Test person statements to the result list.

Icons				
ID	**TP#**	**Alias**	**statement**	**video times-tamp**
40	TP18	"Ernst"	it is unclear which icons are for the faceted search and which are for the result list	05:45:00
166	TP12	"Andreas"	it is weird that the icon for visual and textual facet search are there twice	09:41:00

Table C.3.: Test person statements to the icons.

Overall information cockpit				
ID	**TP#**	**Alias**	**statement**	**video times-tamp**
136	TP7	"Lukas"	the usage is very intuitive	07:35:00
140	TP20	"Wolfgang"	to refresh the site or go back to the landing page normally a click on the logo is a solution. This is missing here	00:13:00
122	TP21	"John"	it assists for getting an orientation whom to ask, but talking to the person personally will still be part of the process because this verifies the results	16:05:00
123	TP21	"John"	if the evaluation of the result list, why and how the list looks like it looks, is obvious or open, then it is easier to trust the result. He wants a relevance factor	17:00:00
139	TP7	"Lukas"	performance was very good, drill down worked fast	08:26:00
165	TP12	"Andreas"	it is rather cool	09:25:00

Table C.4.: Test person statements to the overall information cockpit.

Appendix D.

TEST PERSON BACKGROUND

Test person overview		
Alias	**Sex**	**Age**
"Oliver"	m	23
"Franz"	m	26
"Thomas"	m	28
"Lukas"	m	29
"Rene"	m	30
"Erwin"	m	31
"Christian"	m	32
"Andreas"	m	33
"Josef"	m	33
"Sebastian"	m	34
"Wolfgang"	m	36
"Johann"	m	37
"Elli"	w	38
"Stuart"	m	38
"Julian"	m	40
"Martin"	m	41
"Marion"	w	42
"John"	m	46
"Nico"	m	47
"Ernst"	m	51

Table D.1.: Test person age distribution.

Bibliography

[Amazon.com, 2014] Amazon.com, I. (2014). Amazon.com: Video - camera & photo: Electronics. http://www.amazon.com/s/ref=sr_ex_n_1?rh=n%3A172282%2Cn%3A%2113900861%2Cn%3A%212334090011%2Cn%3A%212334152011%2Cn%3A239390011%2Cn%3A502394%2Cn%3A7161073011&bbn=239390011&ie=UTF8&qid=1395138172.

[Andrews, 2008] Andrews, K. (2008). Evaluation comes in many guises. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08, New York, NY, USA. ACM.

[Andrews, 2013a] Andrews, K. (2013a). Human-computer interaction. http://courses.iicm.tugraz.at/hci/hci.pdf.

[Andrews, 2013b] Andrews, K. (2013b). Information visualisation. http://courses.iicm.tugraz.at/ivis/ivis.pdf.

[Beck et al., 2014] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mallor, S., Shwaber, K., and Sutherland, J. (2014). Agile alliance :: The agile manifesto. http://www.agilealliance.org/the-alliance/the-agile-manifesto/.

[Bilinsky, 2014] Bilinsky, D. (2014). David bilinsky - thoughtful legal management. http://thoughtfullaw.com/2013/05/28/searching-for-a-search-engine/.

[Boehm, 2002] Boehm, B. (2002). Get ready for agile methods, with care. *Computer*, 35(1):64–69.

[Borchers, 2001] Borchers, J. O. (2001). A pattern approach to interaction design. *AI & Society*, 15(4):359–376.

[Borkin et al., 2013] Borkin, M. A., Vo, A. A., Bylinskii, Z., Isola, P., Sunkavalli, S., Oliva, A., and Pfister, H. (2013). What makes a visualization memorable? *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis 2013)*.

[Bostock, 2012a] Bostock, M. (2012a). Bubble chart. http://bl.ocks.org/mbostock/4063269.

[Bostock, 2012b] Bostock, M. (2012b). d3.js three little circles. http://mbostock.github.io/d3/tutorial/circle.html.

[Bostock, 2013] Bostock, M. (2013). Data-driven documents. http://d3js.org/.

[Bostock, 2014] Bostock, M. (2014). Chord layout · mbostock/d3 wiki · github. https://github.com/mbostock/d3/wiki/Chord-Layout.

[Bostock et al., 2012] Bostock, M., Carter, S., and Ericson, M. (2012). At the democratic convention, the words being used. http://www.nytimes.com/interactive/2012/09/04/us/politics/democratic-convention-words.html.

[Bradley, 2011] Bradley, S. (2011). What designers should know about visual perception and memory. http://www.vanseodesign.com/web-design/visual-perception-memory/.

[Branscombe, 2014] Branscombe, M. (2014). Why enterprise search is not internet search — it pro. http://www.itpro.co.uk/608925/why-enterprise-search-is-not-internet-search.

[Carter, 2012] Carter, S. (2012). Four Ways to Slice Obama's 2013 Budget Proposal - Interactive Feature - NYTimes.com. http://www.nytimes.com/interactive/2012/02/13/us/politics/2013-budget-proposal-graphic.html.

[Chuttur, 2009] Chuttur, M. Y. (2009). Overview of the Technology Acceptance Model: Origins, Developments and Future Directions. *Sprouts: Working Papers on Information Systems*, 9(37).

[Cleveland and McGill, 1984] Cleveland, W. S. and McGill, R. (1984). Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387):531–554.

[Cloudera, 2014] Cloudera, I. (2014). Hue sdk documentation. http://cloudera.github.io/hue/docs-2.5.0/sdk/sdk.html.

[Cooper, 1999] Cooper, A. (1999). *The Inmates are Running the Asylum - Why High-Tech Products Drive Us Crazy and How to Restore the Sanity*. SAMS publishing.

[Craft, 2005] Craft, B.; Cairns, P. (2005). Beyond guidelines: what can we learn from the visual information seeking mantra? In *Information Visualisation, 2005. Proceedings. Ninth International Conference on*, pages 110–118.

[Croft, 2007] Croft, J. (2007). Django and mtv. http://jeffcroft.com/blog/2007/jan/11/django-and-mtv/.

[CTSI-Global, 2014] CTSI-Global (2014). Executive dashboard. https://c1.staticflickr.com/3/2849/9727323756_a9f3c15a41_c.jpg.

[CUDO, 2013] CUDO (2013). What is color universal design? http://www.cudo.jp/summary/cudo_e.

[Davis, 1986] Davis, F. D. (1986). *A technology acceptance model for empirically testing new end-user information systems : theory and results*. PhD thesis, Sloan School of Management.

[Davis et al., 2010] Davis, S. B., Bevan, E., and Kudikov, A. (2010). Just in time: defining historical chronographics. pages 1170–1188.

[De Balbin Behrmann, 2014] De Balbin Behrmann, R. (2014). Horses in the tito-bustillo cave 29.000 years old. http://cdn3.spiegel.de/images/image-363852-galleryV9-jpyp.jpg.

[Diderot, 1751] Diderot, D.; d'Alembert, L. R. (1751). Figurative system of human knowledge. http://newqurantranslation.files.wordpress.com/2011/10/enc_systeme_figure1.jpeg.

[Dix et al., 1998] Dix, A. J., Finlay, J. E., Abowd, G. D., and Beale, R. (1998). *Human-Computer Interaction*. Prentice Hall, 2nd edition.

[Django Software Foundation, 2014a] Django Software Foundation (2014a). Contributors to django/django · github. https://github.com/django/django/graphs/contributors.

[Django Software Foundation, 2014b] Django Software Foundation (2014b). Django documentation. https://docs.djangoproject.com/en/1.6/.

[Django Software Foundation, 2014c] Django Software Foundation (2014c). The web framework for perfectionists with deadlines — django. https://www.djangoproject.com/.

[Drucker, 1999] Drucker, P. (1999). Knowledge-Worker Productivity: The Biggest Challenge. *California Management Review*, 41(2):79–94.

[Dublin Core Metadata Initiative (DCMI), 2014] Dublin Core Metadata Initiative (DCMI) (2014). Dublin core metadata element set, version 1.1. http://dublincore.org/documents/dces/.

[Düweke, 2011] Düweke, E.; Rabsch, S. (2011). *Erfolgreiche Websites: SEO, SEM, Online-Marketing, Usability*. Galileo Computing. Galileo Press.

[Dybå and Dingsøyr, 2009] Dybå, T. and Dingsøyr, T. (2009). What do we know about agile software development? *IEEE Software*, 26(5):6–9.

[Eckstein, 2011] Eckstein, R. (2011). *Interactive Search Processes in Complex Work Situations: A Retrieval Framework.* PhD thesis, University of Bamberg.

[Eigner, 2009] Eigner, M.; Stelzer, R. (2009). *Product Lifecycle Management: Ein Leitfaden für Product Development und Life Cycle Management*. Springer.

[Escher, 2013] Escher, M. (2013). M.c. escher - relativity. http://www.mcescher.com/gallery/most-popular/relativity/.

[EURO NCAP, 2014] EURO NCAP (2014). The tests explained — euro ncap - for safer cars crash test safety rating. http://de.euroncap.com/testprocedures.aspx.

[Few, 2004] Few, S. (2004). Tapping the power of visual perception. http://www.perceptualedge.com/articles/ie/visual_perception.pdf.

[Few, 2013] Few, S. (2013). *Information Dashboard Design; Displaying data for at-a-glance monitoring*. Koomey, Jonathan G., 2. edition.

[Fill, 2009] Fill, H. G. (2009). *Visualisation for Semantic Information Systems*. Gabler Verlag, Wiesbaden.

[Friendly, 2014] Friendly, M. (2014). Gallery of data visualization - introducton. http://www.datavis.ca/gallery/.

[Greenfeld and Roy, 2014] Greenfeld, D. and Roy, A. (2014). Django packages : Django packages reusable apps, sites and tools directory. https://www.djangopackages.com/.

[Guardian, 2014] Guardian, N. . M. L. (2014). Government spending by department, 2009-10: full data and visualisation — news — theguardian.com. http://www.theguardian.com/news/datablog/2010/oct/18/government-spending-department-2009-10.

[Haarmann, 2002] Haarmann, H. (2002). *Geschichte der Schrift*. Beck Reihe. Beck.

[Harris and Kamvar, 2014] Harris, J. and Kamvar, S. (2014). We feel fine. http://www.wefeelfine.org/.

[Hawking, 2004] Hawking, D. (2004). Challenges in enterprise search. In *Proceedings of the 15th Australasian Database Conference - Volume 27*, ADC '04, pages 15–24, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

[Healey and Enns, 2012] Healey, C. G. and Enns, J. T. (2012). Attention and visual memory in visualization and computer graphics. *IEEE Transactions on Visualization and Computer Graphics*, 18(7):1170–1188.

[Hearst, 2006] Hearst, M. (2006). Design recommendations for hierarchical faceted search interfaces. In *ACM SIGIR Workshop on Faceted Search*.

[Hearst, 2008] Hearst, M. (2008). UIs for faceted navigation: Recent advances and remaining open problems. In *Workshop on Human-Computer Interaction and Information Retrieval, HCIR*, pages 13–17.

[Hearst, 2009a] Hearst, M. (2009a). *Search User Interfaces*. Cambridge University Press, New York, 1st edition.

[Hearst, 2009b] Hearst, M. A. (2009b). *Search User Interfaces*. Cambridge University Press, 1 edition.

[Heath, 2014] Heath, T. (2014). Images and posters — linked data - connect distributed data across the web. http://linkeddata.org/.

[Hoeber et al., 2009] Hoeber, O., Schroeder, D., and Brooks, M. (2009). Real-world user evaluations of a visual and interactive web search interface. In Banissi, E., Stuart, L. J., Wyeld, T. G., Jern, M., Andrienko, G. L., Memon, N., Alhajj, R., Burkhard, R. A., Grinstein, G. G., Groth, D. P., Ursyn, A., Johansson, J., Forsell, C., Cvek, U., Trutschl, M., Marchese, F. T., Maple, C., Cowell, A. J., and Moere, A. V., editors, *IV*, pages 119–126. IEEE Computer Society.

[Hoffman, 2014] Hoffman, G. (2014). How a car is crash tested. http://autos.aol.com/article/how-crash-tests-work/.

[Holzinger et al., 2013] Holzinger, A., Bruschi, M., and Eder, W. (2013). On interactive data visualization of physiological low-cost-sensor data with focus on mental stress. In *CD-ARES*, pages 469–480.

[Holzinger et al., 2011] Holzinger, A., Searle, G., and Wernbacher, M. (2011). The effect of previous exposure to technology on acceptance and its importance in usability and accessibility engineering. *Universal Access in the Information Society*, 10(3):245–260.

[Huch, 2011] Huch, R.; Jürgens, K. (2011). *Mensch, Körper, Krankheit: Anatomie, Physiologie, Krankheitsbilder ; Lehrbuch und Atlas für die Berufe im Gesundheitswesen ;*. Elsevier, Urban & Fischer.

[ISO 9241-11, 1998] ISO 9241-11 (1998). Ergonomic requirements for office work with visual display terminals - Part 11: Guidance on Usability.

[ISO 9241-210, 2009] ISO 9241-210 (2009). Ergonomics of human system interaction - Part 210: Human-centered design for interactive systems.

[Jentzsch, 2014] Jentzsch, Anja; Cyganiak, R. (2014). Linking open data cloud diagram. http://lod-cloud.net/.

[jQuery Foundation, 2014] jQuery Foundation (2014). jquery. http://jquery.com/.

[Kaiser et al., 2013] Kaiser, C., Softic, S., Sabol, V., and Zechner, M. (2013). A user-interaction approach in an information-cockpit. In *WEBIST 2013, Proceedings of the 9th International Conference on Web Information Systems and Technologies*, pages 635–639, Aachen, Germany. SciTePress.

[Keim et al., 2006] Keim, D., Mansmann, F., Schneidewind, J., and Ziegler, H. (2006). Challenges in visual data analysis. In *Information Visualization, 2006. IV 2006. Tenth International Conference on*, pages 9–16.

[Kienreich et al., 2008] Kienreich, W., Lex, E., and Seifert, C. (2008). Apa labs: An experimental web-based platform for the retrieval and analysis of news articles. In *Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the*, pages 58–62.

[Kirsh and Maglio, 1994] Kirsh, D. and Maglio, P. (1994). On distinguishing epistemic from pragmatic action. *Cognitive Science*, 18(4):513–549.

[Kniberg and Skarin, 2010] Kniberg, H. and Skarin, M. (2010). *Kanban and Scrum: Making the Most of Both*. Enterprise software development series. Lulu Enterprises Incorporated.

[Krastel, 2001] Krastel, M. (2001). STEP basiertes simulationsdatenmanagement - abschlussbericht zur machbarkeitsstudie.

[Krempels and Stocker, 2013] Krempels, K.-H. and Stocker, A., editors (2013). *WEBIST 2013, Proceedings of the 9th International Conference on Web Information Systems and Technologies*, Aachen, Germany. SciTePress.

[Kropp and Meier, 2012] Kropp, M. and Meier, A. (2012). Swiss agile study 2012: Agile software-entwicklung in der schweiz.

[Lee et al., 2009] Lee, B., Smith, G., Robertson, G. G., Czerwinski, M., and Tan, D. S. (2009). Facetlens: exposing trends and relationships to support sensemaking within faceted datasets. In Jr., D. R. O., Arthur, R. B., Hinckley, K., Morris, M. R., Hudson, S. E., and Greenberg, S., editors, *CHI*, pages 1293–1302. ACM.

[Likert, 1932] Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*, 22(140).

[Mackinlay, 1986] Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics TOG*, 5(2):110–141.

[Manning et al., 2008] Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

[Marchionini, 2006] Marchionini, G. (2006). Toward human-computer information retrieval. *Bulletin of the Association for Information Science and Technology*.

[Marchionini and Capra, 2014] Marchionini, G. and Capra, R. (2014). Unc idl relation browser. http://ils.unc.edu/relationbrowser/index.php?page=history.

[Marchionini et al., 2014] Marchionini, G., Plaisant, C., and Komlodi, A. (2014). 3872. http://hcil2.cs.umd.edu/trs/98-01/98-01.html.

[Merriam-Webster, 2014] Merriam-Webster, I. (2014). Iconography - definition and more from the free merriam-webster dictionary. http://www.merriam-webster.com/dictionary/iconography.

[Mouchon, 1780] Mouchon, P. (1780). Table analytique et raisonnée des matieres contenues dans les xxxiii volumes in-folio du dictionnaire des sciences, des arts et des métiers, et dans son supplément. http://seeingcomplexity.files.wordpress.com/2011/02/image008.jpg.

[Mountain Goat Software, 2014] Mountain Goat Software (2014). Scrum images. http://www.mountaingoatsoftware.com/agile/scrum/images/.

[Nation et al., 1997] Nation, D., Plaisant, C., Marchionini, G., and Komlodi, A. (1997). Visualizing websites using a hierarchical table of contents browser: Webtoc. In *In Proceedings of the Third Conference on Human Factors and the Web*.

[Neal, 1997] Neal, M. (1997). Parametric search: Evolving information retrieval for the web. *CADIS Inc*, 1501(2):5.

[Nielsen, 1993] Nielsen, J. (1993). *Usability Engineering*. Interactive technologies. Morgan Kaufmann.

[Peter, 2014] Peter, S. (2014). Informationsvisualisierung in der modernen kunst - medieninformatik. http://www.medien.ifi.lmu.de/lehre/ws0607/mmi1/essays/Silvia-Peter.xhtml.

[Plaisant, 2004] Plaisant, C. (2004). The challenge of information visualization evaluation. In Costabile, M. F., editor, *AVI*, pages 109–116. ACM Press.

[Pohl and Rupp, 2010] Pohl, K. and Rupp, C. (2010). *Basiswissen Requirements Engineering - Aus- und Weiterbildung zum "Certified Professional for Requirements Engineering": Foundation Level nach IREB-Standard (2. Aufl.).* dpunkt.verlag.

[Polowinski, 2009] Polowinski, J. (2009). Widgets for faceted browsing. In Smith, M. and Salvendy, G., editors, *Human Interface and the Management of Information. Designing Information Environments*, volume 5617 of *Lecture Notes in Computer Science*, pages 601–610. Springer Berlin Heidelberg.

[Priestley, 1765] Priestley, J. (1765). A chart of biography, photo by stephen boyd davis. http://chronographics.blogspot.co.at/2009_12_01_archive.html.

[Prieto-Díaz, 1987] Prieto-Díaz, R. (1987). Domain analysis for reusability. In *Proc. of the eleventh annual international computer software and applications conference (COMSPAC 87)*. IEEE computer society.

[Prieto-Díaz, 1990] Prieto-Díaz, R. (1990). Domain analysis: An introduction. *ACM SIGSoft Software Engineering Notes*, 15(2):47–54.

[Roberts, 1989] Roberts, R. M. (1989). *Serendipity: Accidental Discoveries in Science*. Wiley Science Editions. Wiley.

[Rodden, 2014] Rodden, K. (2014). Sequences sunburst. http://bl.ocks.org/kerryrodden/7090426.

[Rosenberger, 2013] Rosenberger, M. (2013). Rattlecad software. http://rattlecad.sourceforge.net/.

[Schabereiter-Gurtner et al., 2002] Schabereiter-Gurtner, C., Saiz-Jimenez, C., Piñar, G., Lubitz, W., and Rölleke, S. (2002). Phylogenetic 16s rrna analysis reveals the presence of complex and partly unknown bacterial communities in tito bustillo cave, spain, and on its palaeolithic paintings. *Environmental Microbiology*, 4(7):392–400.

[Schamber et al., 1990] Schamber, L., Eisenberg, M. B., and Nilan, M. S. (1990). A re-examination of relevance: toward a dynamic, situational definition. *Inf. Process. Manage.*, 26(6):755–776.

[Schulz, 2011] Schulz, H. (2011). Treevis.net: A tree visualization reference. *Computer Graphics and Applications, IEEE*, 31(6):11–15.

[Schulz, 2014] Schulz, H.-J. (2014). treevis.net. treevis.net.

[Shepard, 1990] Shepard, R. N. (1990). *Mind sights: Original visual illusions, ambiguities, and other anomalies, with a commentary on the play of mind in perception and art*. W. H. Freeman, New York.

[Shneiderman, 1998] Shneiderman, B. (1998). *Designing the User Interface – Strategies for Effective Human-Computer-Interaction*. Addison-Wesley Longman, 3 edition.

[Smith et al., 2006a] Smith, G., Czerwinski, M., Meyers, B., Robbins, D., Robertson, G., and D.S., T. (2006a). Facetmap: A scalable search and browse visualization. In *IEEE Transactions on Visualization and Computer Graphics, Vol. 12 (5)*, pages 797–804.

[Smith et al., 2006b] Smith, G., Czerwinski, M., Meyers, B., Robbins, D. C., Robertson, G. G., and Tan, D. S. (2006b). Facetmap: A scalable search and browse visualization. *IEEE Trans. Vis. Comput. Graph.*, 12(5):797–804.

[Smith et al., 2014] Smith, G., Robertson, G., Czerwinski, M., and Tan, D. (2014). Computational user experiences: Facetlens. http://research.microsoft.com/en-us/um/redmond/groups/cue/facetlens/FacetLensb.jpg.

[Stone, 2005] Stone, D. (2005). *User interface design and evaluation*. The Morgan Kaufmann series in interactive technologies. Morgan Kaufmann, Amsterdam, Boston, Paris (etc.).

[TechSmith, 2014] TechSmith, C. (2014). Morae. http://www.techsmith.de/morae.html.

[The Library of Congress, 2014] The Library of Congress (2014). Cason monk-metcalf: Exploratory data vis. http://viewshare.org/views/trow/cason-monk-metcalf/.

[The World Wide Web Consortium (W3C), 2014] The World Wide Web Consortium (W3C) (2014). Html & css - w3c. http://www.w3.org/standards/webdesign/htmlcss.

[Treisman and Gelade, 1980] Treisman, A. M. and Gelade, G. (1980). A feature-integration theory of attention. *Cognit Psychol*, 12(1):97–136.

[Tunkelang, 2009] Tunkelang, D. (2009). *Faceted Search*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers.

[Turkay et al., 2014] Turkay, C., Jeanquartier, F., Holzinger, A., and Hauser, H. (2014). On computationally-enhanced visual analysis of heterogeneous data and its application in biomedical informatics. In *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics*, pages 117–140.

[Tvarozek and Bielikova, 2008] Tvarozek, M. and Bielikova, M. (2008). Personalized view-based search and visualization as a means for deep/semantic web data access. In *Proceedings of the 17th international conference on World Wide Web (WWW'08)*, pages 1023–1024.

[Twyman, 1990] Twyman, M. (1990). *Early lithographed books : a study of the design and production of improper books in the age of the hand press, with a catalogue / by Michael Twyman*. Farrand Press and Private Libraries Association; Distributed in the Western Hemisphere by the Book Press London : Williamsburg, Va.

[Vallandingham, 2012a] Vallandingham, J. (2012a). Bubble cloud. http://vallandingham.me/bubble_cloud/.

[Vallandingham, 2012b] Vallandingham, J. (2012b). Building a bubble cloud. http://vallandingham.me/building_a_bubble_cloud.html.

[Vallandingham, 2012c] Vallandingham, J. (2012c). Creating animated bubble charts in d3. http://vallandingham.me/bubble_charts_in_d3.html.

[Vallandingham, 2012d] Vallandingham, J. (2012d). Gates foundation spending - vallandingham.me. http://vallandingham.me/vis/gates/.

[Velez et al., 2005] Velez, M. C., Silver, D., and Tremaine, M. (2005). Understanding visualization through spatial ability differences. In *IEEE Visualization*, page 65. IEEE Computer Society.

[Venkatesh and Davis, 2000] Venkatesh, V. and Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Manage. Sci.*, 46:186–204.

[W3C, 2014] W3C (2014). Sweoig/taskforces/communityprojects/linkingopendata - w3c wiki. http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData.

[Ward et al., 2010] Ward, M., Grinstein, G., and Keim, D. A. (2010). Interactive data visualization: Foundations, techniques, and application. A. K. Peters, Ltd.

[Ware, 2013] Ware, C. (2013). *Information Visualization: Perception for Design*. Interactive Technologies. Elsevier Science, 3. edition.

[Wei et al., 2013] Wei, B., Liu, J., Ma, J., Zheng, Q., Zhang, W., and Feng, B. (2013). Dft-extractor: A system to extract domain-specific faceted taxonomies from wikipedia. In *Proceedings of the 22Nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 277–280, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

[Weissler, 2014] Weissler, P. (2014). Focus b-pillar 'tailor rolled' to 8 different thicknesses - sae international. http://articles.sae.org/7695/.

[Wilson et al., 2010] Wilson, M. L., Kules, B., m. c. schraefel, and Shneiderman, B. (2010). From keyword search to exploration: Designing future search interfaces for the web. *Foundations and Trends in Web Science*, 2(1):1–97.

[Wolfe and Horowitz, 2004] Wolfe, J. M. and Horowitz, T. S. (2004). What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuroscience*, 5(6):495–501.

[Yahoo!, 2014] Yahoo! (2014). Welcome to flickr - photo sharing. https://www.flickr.com/.

[Zimmerman et al., 2007] Zimmerman, J., Forlizzi, J., and Evenson, S. (2007). Research through design as a method for interaction design research in hci. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 493–502, New York, NY, USA. ACM.