

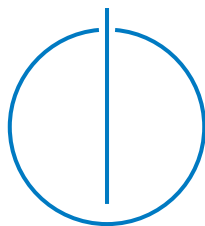


DEPARTMENT OF INFORMATICS
TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

OPTIMIZING THE USER
EXPERIENCE OF A SOCIAL
CONTENT MANAGEMENT
SOFTWARE FOR CASUAL USERS

Florian Katenbrink





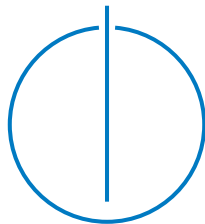
DEPARTMENT OF INFORMATICS
TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

OPTIMIZING THE USER EXPERIENCE OF A
SOCIAL CONTENT MANAGEMENT
SOFTWARE FOR CASUAL USERS

OPTIMIERUNG DER USER EXPERIENCE
EINER SOCIAL CONTENT MANAGEMENT
SOFTWARE FÜR ENDBENUTZER

Author: Florian Katenbrink
Advisor: Thomas Reschenhofer
Supervisor: Prof. Dr. rer.nat. Florian Matthes
Date: August 15th, 2015



Declaration of Authorship

I assure the single handed composition of this bachelor's thesis only supported by declared resources.

Ich versichere, dass ich diese Bachelor's Thesis selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Garching, August 12th, 2015

Florian Katenbrink

Abstract

Today's enterprises often struggle with a mass of growing and diverse data. As content becomes more and more complex data sheets, repositories of data and calendars are not sufficient. In order to keep up with the speed there exist social content management systems such as the hybrid-wiki Tricia.

A crucial factor in the long-term success of software and quality is the design. This thesis presents the design of a social content management software for casual users, thereby optimizing the user experience by using well-proven principles in literature and related software. Content of the designs is based on Tricia's data. The interface is designed with the Material Design language by Google.

The scientific approach is based on a reliable design-science research framework. Designs are therefore periodically assessed and refined with both users and developers of Tricia. Well-chosen principles from Tricia are adopted and tasks are introduced as a central element in the application's design to enhance collaboration.

Contents

1. Introduction	1
1.1. Motivation	1
1.2. Problem Statement	1
1.3. Research Questions	2
1.4. Outline	3
2. Related Work	4
2.1. Foundation: Tricia & SocioCortex	4
2.2. Darwin	6
2.3. The Organic Data Science Framework	7
2.4. Design of Social Software	8
2.4.1. Design Guidelines	8
2.4.2. Material Design	10
3. Research Method	11
4. Design of a Generic Web Application's User Interface	13
4.1. Colors	13
4.2. Typography	14
4.3. Structural Overview	15
4.4. Navigation	17
4.4.1. Data Model	17
4.4.2. Application Bar	17
4.4.3. Sidebar	19
4.5. Wiki Content	24
4.5.1. Data Model	24
4.5.2. Unstructured Information	25
4.5.3. Structured Information	29
4.6. User Profile	35
4.7. Activity Feed	37

Contents

4.8. Search	40
4.9. Task Details	44
4.10. Alerts	44
5. Conclusion & Outlook	46
5.1. Summary	46
5.2. Conclusion	47
5.3. Outlook	49
Bibliography	51
A. Appendix	i
A.1. Evolution of the Navigation	i
A.2. Sidebar Scrolling	vii
A.3. Windowed Designs	x
List of Figures	xxvi

1. Introduction

This chapter gives a brief introduction into the thesis. First the thesis is motivated and the problem illustrated. Then research questions are stated and an outline of the thesis given.

1.1. Motivation

In today's enterprises there exist many different data sources containing multiple data representations (such as dashboards, metrics or emails), which are produced and consumed by again different stakeholders. This information grows with a rapid speed and in order to keep up with it, these enterprises have to use modern tools and methods [EM00]. An information and social integration platform can address this problem, which integrates these different data representations and has to provide stakeholders' relationships, access rights and responsibilities.

In 2005 already, researches saw the need for social software in order to handle these masses of information within enterprises [LM05]. Social content management systems represent an example of a social software. These perform as content management systems with main focus on collecting, managing and publishing information, but also on collaboration of users within this system. A concrete implementation of such an enterprise collaboration platform is Tricia, a Java-based hybrid-wiki system [MNS11].

1.2. Problem Statement

In order to visualize an enterprise collaboration platform, consisting of many different views for end users, different roles and responsibilities, individual demands and groups, an implementation of such a platform has to provide a

respective model-based user-interface for each party and support all capabilities necessary for successful collaboration. There are many different use-cases, types of users and different access points, such as mobile and desktop devices. Therefore many factors need to be considered when designing the platform. One crucial factor in the (long-term) success of software and quality is the design [Hu13]. Content is becoming more and more complex and for example the navigation cannot keep up with the speed of updates [LH11], which leads to a system full of data, but with bad user experience. That is often because most of the developers that create web systems do not follow patterns and conventions in order to create usable web applications [Di13].

Nowadays web applications need to be available for both desktop web and mobile web use, which is just one example of the problems Tricia has. It uses one single user-interface that serves as a mixture for different use-cases. Therefore a new system called SocioCortex is being developed whereas the back-end and front-end are decoupled so any number of user-interfaces can access the data [Sc14]. In order to support casual users, who want to access the system from their desktop, a generic web application has to be provided. Casual users are within one of the two groups:

- Anonymous, non-registered users who want to gather information.
- Registered users who want to change the content, but not the structure.

1.3. Research Questions

As specified earlier in chapter 1.2, there is a need for a generic web application for casual users. Therefore this thesis focuses on usability improvements of a generic web application for the casual user and examines the following research questions, which concern the SocioCortex application:

1. *How can the navigation be structured so that users can easily navigate the web application exploratorily, while more experienced users can use the full potential of it?*
2. *How can the wiki content be displayed in order to make scanning the pages as well as editing content easier?*

3. *What user details are necessary for successful collaboration and how can they be intuitively displayed in a user profile?*
4. *How can a powerful, still easy-to-use activity feed be integrated in the new system?*
5. *How does a search functionality look that assists users in their finding?*

1.4. Outline

CHAPTER 1: INTRODUCTION

The thesis is briefly introduced, the motivation and problem scope are portrayed, whereupon research questions are stated.

CHAPTER 2: RELATED WORK

Related work is presented through the foundation of this thesis and two other wiki systems. A short literature review then states design principles.

CHAPTER 3: RESEARCH METHOD

The research method is presented through a framework for design science research and evaluation steps for analyzing the implementation.

CHAPTER 4: DESIGN OF A GENERIC WEB APPLICATION'S USER INTERFACE

The application of principles and design of the social content management software, optimized in user experience, is demonstrated.

CHAPTER 5: CONCLUSION & OUTLOOK

Concludes the thesis by reviewing through evaluation steps and gives an outlook for further study.

2. Related Work

This chapter takes a look at related work that influences this research. The research's foundation is laid through Tricia and SocioCortex. Then the wiki systems "Darwin" and "The Organic Data Science Framework" is examined. At last design principles are revealed and Material Design is explained.

2.1. Foundation: Tricia & SocioCortex

As mentioned earlier in chapter 1.2, there is the current production version of a hybrid-wiki called Tricia. There is a need of social information hubs, because collaboration not only becomes more and more important in modern enterprises, but also in social and research projects [Ba05]. Therefore problems often are not solved and products and services are not developed by a single person anymore, but by a network of professionals. These projects involve many different stakeholders with different interests and backgrounds. Therefore, social information hubs are supposed to be designed to be usable for any party. Because the projects can be of any area, such systems support many different information representations and integration for many data sources [Sc14].

Tricia was developed after wikis were increasingly used as knowledge repositories for data that was scattered across multiple formats and platforms. Wikis have a problem though. They do not support structured access of the data, however there is a demand to query the wiki or export a spreadsheet of data in knowledge-intensive processes for example. There exist semantic wikis, though their complexity in syntax and modeling concepts make them inappropriate and too hard to learn for most users. Thus the hybrid-wiki Tricia was proposed: a mixture between semantic and standard wikis. In order to offer users a familiar concept, rich text editing of the wiki content is enabled [MNS11].

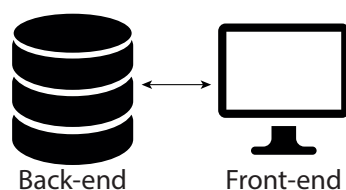


Figure 2.1.: Tricia: strong coupling of back-end and front-end.

After years of use, experience showed that Tricia has many constraints. An analysis with Google Analytics¹ gave additional insight into what features of the wiki system are used more and how it is used. There exists a high coupling between the back-end and the front-end of the system, as shows Figure 2.1. This means there is no possibility to support multiple use-cases. In a wiki system, there exist many users, use-cases and organizations, which make this system too complex when only supporting a single user-interface. Therefore the two parts need to be decoupled. SocioCortex approaches this with front-end applications that access the back-end via REST-API, which means there can be a theoretically unlimited number of web and mobile applications that access the SocioCortex core (see Figure 2.2). Within this core, SocioCortex creates a social, activity and data graph to connect the collaboration, activity and data. On the data-side different sources for data and identity-management are supported. These are not relevant for this thesis though.

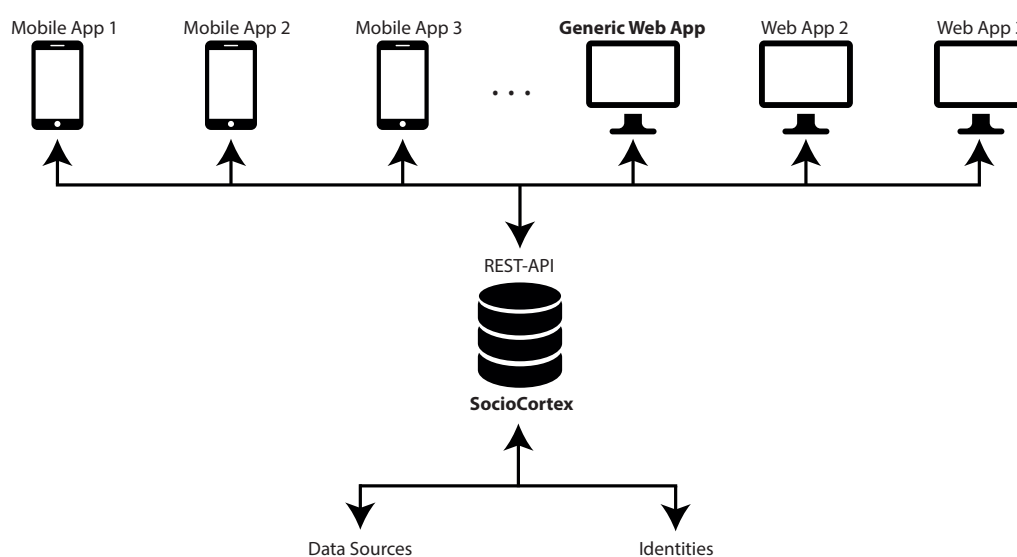


Figure 2.2.: SocioCortex: Separation of back-end and front-end.

¹<http://www.google.com/analytics> (accessed June 22nd, 2015)

Tricia and the SocioCortex concept build the foundation of this thesis. Therefore much of the content in the design is taken from the current Sebis chair website². As the back-end in Tricia and SocioCortex are almost the same, the generic web application's design is fitted to represent the content. The relevant data model is explained in chapter 4.

2.2. Darwin

Darwin is another collaborative wiki, whereas a new concept of tasks was introduced. It provides a framework for knowledge-intensive workflows, which deeply integrates task in collaboration. Tasks are attached to wiki pages in order to coordinate work processes. They can then be decomposed further in order to simplify and reduce the work load. A task not only has metadata such as a start date, end date, the progress and an owner who is delegated to this task, but also includes expertise to be earned. Such a principle encourages users of the system to finish their tasks, because it then reflects their know-how to other people.

For better understanding of the tasks' and subtasks' current progress, a Gantt chart is generated on every wiki page. It is important to mention that the progress is either automatically calculated, based on the completion of attributes or can be edited manually by the users. The progress of each tasks is shown with a circular progress or a bar in case of the Gantt chart. Table 2.3 shows how the colors change based on the state of the task. Darwin also considers inconsistencies, but they are not relevant for this thesis.

Color	Progress
Gray	Task not started
Green	Task completed
Red	Task overdue
Gray & Green	Task partially completed
Light red & Red	Task overdue and partially completed

Figure 2.3.: Colors and progress

As mentioned, tasks can include attributes, which can be a mandatory part of the work load. Attributes have a label and a value. The value can be of

²<https://wwwmatthes.in.tum.de> (accessed August 2nd, 2015)

different data types such as text, boolean, file or date. Based on algorithms, Darwin offers auto-completion when adding a new task. Access rights define, which users are allowed to read and edit the attribute.

Users can define types for wiki pages. They contain information about what content is displayed for each page, such as certain tasks and attributes. An example of such as type could be "Scientific Publication". Pre-defined tasks and attributes are added automatically, though users can later on add new tasks and attributes [HKM15].

The presented concepts will be available in the SocioCortex system and are therefore incorporated in the user experience improvements of the generic web application.

2.3. The Organic Data Science Framework

The Organic Data Science Framework was developed to "support task-oriented self-organizing on-line communities for open scientific collaboration" as science has become more and more collaborative. This work focused on developing a framework which tackles projects with high degrees of "organization and coordination", "retaining users over the long term" and an "incrementally growing community". The system lets users create tasks with a description, which can be decomposed again into smaller subtasks. As the project is open for all contributors and supposed to be ultimately transparent, any user of the system can do so [Mi15b].

In order to refer to tasks from both inside and outside the project, every task has its own page and a unique link. Like the Darwin system, the Organic Data Science Framework has task metadata: start data, end date, owner and expertise. Additional metadata is a field for participants and the type of the task. Optional additional properties can be added to the task, just like attributes to tasks in Darwin. Users can navigate through a tree-structure of tasks, filter or search for it or they can access their personal worklist, which contains all tasks they are the owner or participant of. On the task page, users can switch between a view to show all subtasks and a Gantt chart. A feature that helps users not to forget any tasks is a bell icon for all overdue tasks with the number of overdue tasks attached. On hover it opens a list with all the

overdue tasks. The Organic Data Science Framework offers a context menu in the sidebar for actions, such as renaming, deleting or copying of tasks. As stated earlier, the framework's transparency is applied throughout the whole project. All users have a public profile with their earned expertise, current tasks, future tasks and completed tasks, which makes it easy for others to see what users are working on and specialized in. The task states are handled similar as in Darwin, but offer a powerful representation of inconsistencies and also handle the state of metadata [Mi15a]. As this is not relevant for this thesis, I am not going to analyze this any further.

Many best practices and features of this project are considered in this thesis and integrated in the redesign of the SocioCortex generic web application. One major difference though is that Darwin and the Organic Data Science Framework develop both on the declarative and instantiation layer of tasks, while this thesis only develops the instantiation layer. Therefore modeling is relevant, but not considered in this thesis.

2.4. Design of Social Software

Web design is one of the fastest changing parts, which affects creating modern and robust web applications. Nevertheless there are many principles of design that can be applied independently from the actual design to make a web application more robust and user-friendly, hence improve the user experience.

2.4.1. Design Guidelines

Research on literature about usability improvements of web interfaces and web design revealed many principles that are applied in the design of the generic web application within this thesis. As this thesis' research questions relate to the navigation, wiki content, user profile, activity feed and search, strong effort was made to conduct further research in these five areas. Table 2.4 displays an aggregation of the most important principles, which are also included and applied in the generic web application design, extracted from the examined literature. Each block of principles focuses on the feature part, which the heading states, while the general principles are applied across features.

G. General

- G1* People scan pages rapidly, thus key elements must be visible
- G2* Keep names, titles and descriptions simple
- G3* Make buttons obviously clickable
- G4* Stick to conventions
- G5* The more important elements are, the more prominent they are
- G6* Elements that are related logically are related visually
- G7* Elements are nested visually to show what is part of what
- G8* Pages need to be broken down into clearly defined areas
- G9* Avoid instructions

N. Navigation

- N1* Minimize the amount of jumps, thus content is loaded dynamically
- N2* The navigation displays users where they currently are
- N3* Different pages have slight changes only, so users can easily adapt
- N4* Show movement and progress through changes in the navigation
- N5* People expect the main navigation to lead within the current site
- N6* Breadcrumbs may not increase navigational efficiency and understanding
- N7* The top bar needs to contain: logo, sections, utilities and search
- N8* Make the active tab a different color

W. Wiki Content

- W1* Format text to support scanning
- W2* Page names match the navigation names
- W3* Promote the most important topics on the homepage
- W4* Allow users to tag content

S. Search

- S1* Offer autocomplete
- S2* Allow items to be filtered

F Activity Feed

- F1* Let users subscribe to content
- F2* Users can share content

U. User Profile

- U1* Encourage users to improve their profile by showing their progress
- U2* Do not overload the profile page with unneeded content
- U3* Give users control over how to present themselves
- U4* Allow users to upload one or more images of themselves

Figure 2.4.: Design Principles of User Experience Optimization

"Don't make me think" by Steve Krug [Kr14] states many principles for general user-interface design to improve the user experience. "Designing Web Navigation: Optimizing the User Experience" by James Kalbach [Ka07] gives deep insight into how to design a user-friendly navigation that leads users through a website. "Designing Web Interfaces: Principles and Patterns for Rich Interactions" by Bill Scott and Theresa Neil [SN09] add major principles in order to demonstrate wiki content effectively and demonstrate an activity feed. "Search Patterns" by Peter Morville and Jefferey Callender [MC10] offer design patterns to create a powerful yet easy-to-use search feature. "Designing Social Interfaces: Principles, Patterns, and Practices for Improving the User Experience" by Christian Crumlish and Erin Malone [CM09] uses many examples in order to help improve the social aspect of a web application. Nevertheless this literature is used cross-thematically in order to ensure well-matched principles for a balanced user experience.

2.4.2. Material Design

Material Design is a visual language that was introduced by Google in June 2014. Its goal is to merge principles of classical, good design with today's technology, therefore developing a single underlying system for all devices and platforms. The material environment measures devices in three dimensions, so the design can display different layers and create the illusion of a 3D world. This effect is enhanced by using key and ambient light to cast shadows so as to create visual hierarchies. These shadows are cast, when different layers are on top of one another and block imaginary light sources coming from the z-axis (perpendicular to the display) [Go15a].

This design language is already established in many Google services and Android, the world's most used mobile operating system with a market share of nearly 80% [ID15]. Within one year of production use, Material Design evolved to a most dominant design with more and more websites materializing their design. It offers many design patterns and components that users only have to get used to once and then this knowledge is applicable across devices.

3. Research Method

This chapter illustrates the research method the thesis was performed with.

Information systems are developed to improve an organization in its performance. Research on this is dominated by the paradigms of behavioral science and design science. Hevner et al. recognized that while behavioral science is well covered by many research methods, design science is not. However truth, which is generated by behavioral science and utility, which is generated by design science are inseparable for information systems research. Therefore they developed a framework, which uses a model of both behavioral science and design science. So that the relevancy and accuracy of the research can be measured, seven guidelines "for conducting and evaluating good design-science research" [He04] are included.

The framework (see Figure 3.1) assumes a problem space within businesses. The environment defines this with people, organization and technology "that define business needs as they are perceived by people within the organization" [He04]. These business needs justify the relevancy of research. The business need for this thesis is, as described in chapter 1, the growing mass and diversity in information within enterprises and the resulting system Tricia that has a non-intuitive interface and mixes many use-cases into one.

In information systems research, development and justification are used by the behavioral science method, whereas building and evaluating are the approach of the design science method. This merges in two circular phases, as described by Hevner et al.: Develop/Build, which generates the artifact and Justify/Evaluate, which takes the artifact and results in either problems within the theory or further refinement and reassessment. The final result can thus be used as a solution for the problem space of the environment or as an endorsement for the knowledge base. In the case of this thesis, two weeks of building and developing are followed by justification in front of a focus group, consisting of twelve users of Tricia. Four thereof are also developers of Tricia, three

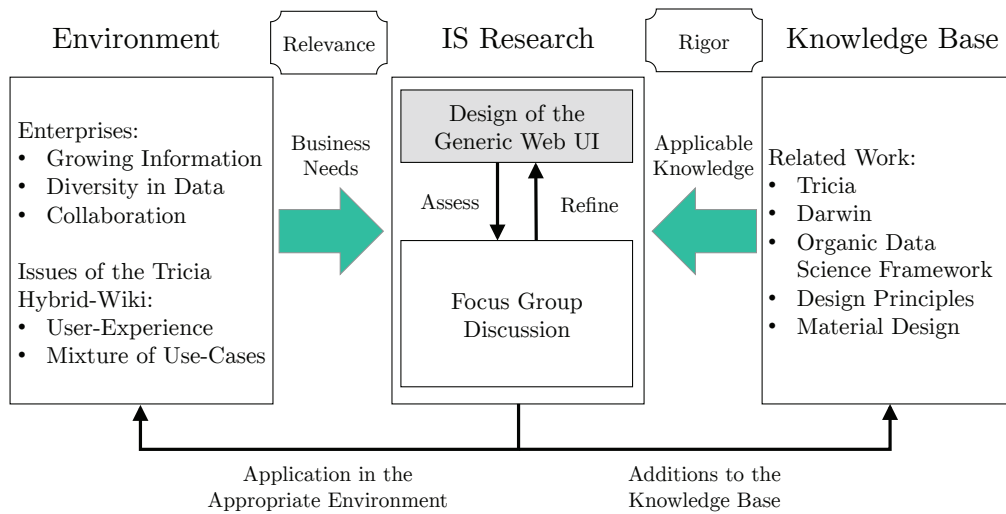


Figure 3.1.: The Adapted Research Framework [He04]

thereof are users of Darwin and one is the creator of the Organic Data Science Framework. Afterwards there was an evaluation by the group. The evolution of a system's component can be viewed in the appendix (see A.1). It displays the navigation, which demonstrates the ten iteration of the Develop/Build and Justify/Evaluate loop.

In order to perform scientific research, the knowledge base provides related research tools and materials. Methodologies offer guidelines that should be considered in the Justify/Evaluate phase. By applying the foundations and methodologies accordingly, rigor in information systems research is ensured. Foundation of this thesis are Tricia, Darwin and the Organic Data Science Framework, as described in chapter 2 and focus group analysis is done in order to ensure rigor.

The complete description of the seven guidelines in order to conduct the framework is illustrated in Hevner et al.: Problem Relevance, Research Rigor, Design as a Search Process, Design as an Artifact, Design Evaluation, Research Contributions and Communication of Research. They are all taken up in chapter 5.2 to draw a conclusion [He04].

4. Design of a Generic Web Application's User Interface

This chapter describes the user interface design of the generic web application of SocioCortex for casual users, while applying the design principles and concepts as described in chapter 2. Each section tries to distinguish the difference to Tricia's approach as clearly as possible by comparing it to the design of the generic web application's user interface. When two graphics are given within one Figure and they are separated by a line, then the upper graphic is the current Tricia UI and the lower one is the design of the generic web application's user interface respectively Tricia left and the generic web application right. The latter is explained in detail, whereas Tricia's UI is only explained, when comparing. Therefore most of the content in the generic web app designs is based on Tricia's current UI. The designs shown are those from after the last iteration of the research cycle. However it is justified why this alternative is chosen eventually. All designs from the generic web app refer to users that is logged in. Users who are not logged in do not have the possibility to edit anything. Therefore they see the same surface, simply with all edit possibilities unavailable. This thesis focuses on optimizing the user experience for casual users in terms of UI optimization, though it is important to specify the underlying data model as it influenced the design decisions. Therefore sections describe the underlying data model beforehand, if necessary. Italic letters with numbers in parentheses identify a design principle that was applied.

4.1. Colors

Material Design suggests the usage of the official color palette as displayed on their website³. They are defined as "unexpected and vibrant", just like the environment and the whole color palette works harmoniously with each

³<https://www.google.com/design/spec/style/color.html> (accessed July 29th, 2015)

other. In order to keep the existing Tricia branding, a similar blue tone was chosen. It is displayed as the first primary color in Figure 4.1. Then two more blue tones are primary colors and a red color the accent color. The second line displays text colors and link color. The third line displays the icon color and background colors. All colors applied together convey a fresh, bright impression. This clear impression is crucial for users to enjoy using the web application. Colors in the generic web app apply even more visual hierarchy and direct the user's attention. This is demonstrated in more detail in later sections. Colors that indicate the task state are explained in section 4.5.3 and colors relating the activity feed are explained in 4.7.

#195B8B	#88C0E8	#F0F7FC	#FF5252
Primary Color	Primary Color	Secondary Color	Accent Color
#FFFFFF	#000000	#555555	#003359
Text Color	Text Color	Secon. Text Color	Link Color
#9B9B9B	#EEEEEE	#FAFAFA	#CCCCCC
Icon Color	Background Color	Background Color	Background Color

Figure 4.1.: The Generic Web Application: Color Palette

4.2. Typography

The font, used throughout the whole generic web application's user interface, is Roboto. It is specified by Material Design as the preferred font for Material Design⁴, as it is refined across a wide set of platforms. The font's official application takes place in Android and many Google applications, also because

⁴<https://www.google.com/design/spec/style/typography.html> (accessed July 29th, 2015)

it is developed by Google. It is a wide and round neo-grotesque sans-serif font. Therefore it offers ideal clarity for both longer texts and headings on devices with different resolutions, which makes it easy for users to read (*G1*). Font sizes range from 10pt to 24pt.

4.3. Structural Overview

Figure 4.2 displays the welcome page of the Chair for Software Engineering for Business Information Systems at the Technische Universität München. The upper image is the current version of Tricia. In direct comparison the lower image is the displayed generic web app. The web application is framed in a window on Google Chrome OS⁵. At any point and on any page of the web application, it is split in three parts, which define the overall structure of the application: The application bar at the top provides global functionality of the website, the sidebar on the left acts as secondary navigation and the wiki content in the center contains all structured and unstructured information about a single page.

The main structure is adapted from Tricia in order to provide a smooth transition for existing users. However both small and large improvements are made, which are demonstrated in the following sections. These improvements are crucial, since studies show that users only need 50 milliseconds to find out if a page appeals to them or not [Li06]. In the generic web application's design users have less options available at direct sight, therefore it makes key elements visible (*G1*). This pattern of reducing features on a single page is applied throughout the whole design. This helps users focus with what they are trying to find. Users who are visiting the application for the first time might be overwhelmed with the mass of features Tricia offers on one page. Another change is the removal of breadcrumbs. As studies show, breadcrumbs may not increase navigational efficiency and understanding (*N6*) [Kr14].

⁵Google Chrome OS is built on top of the open-source Chromium OS project [co15].

4. Design of a Generic Web Application's User Interface

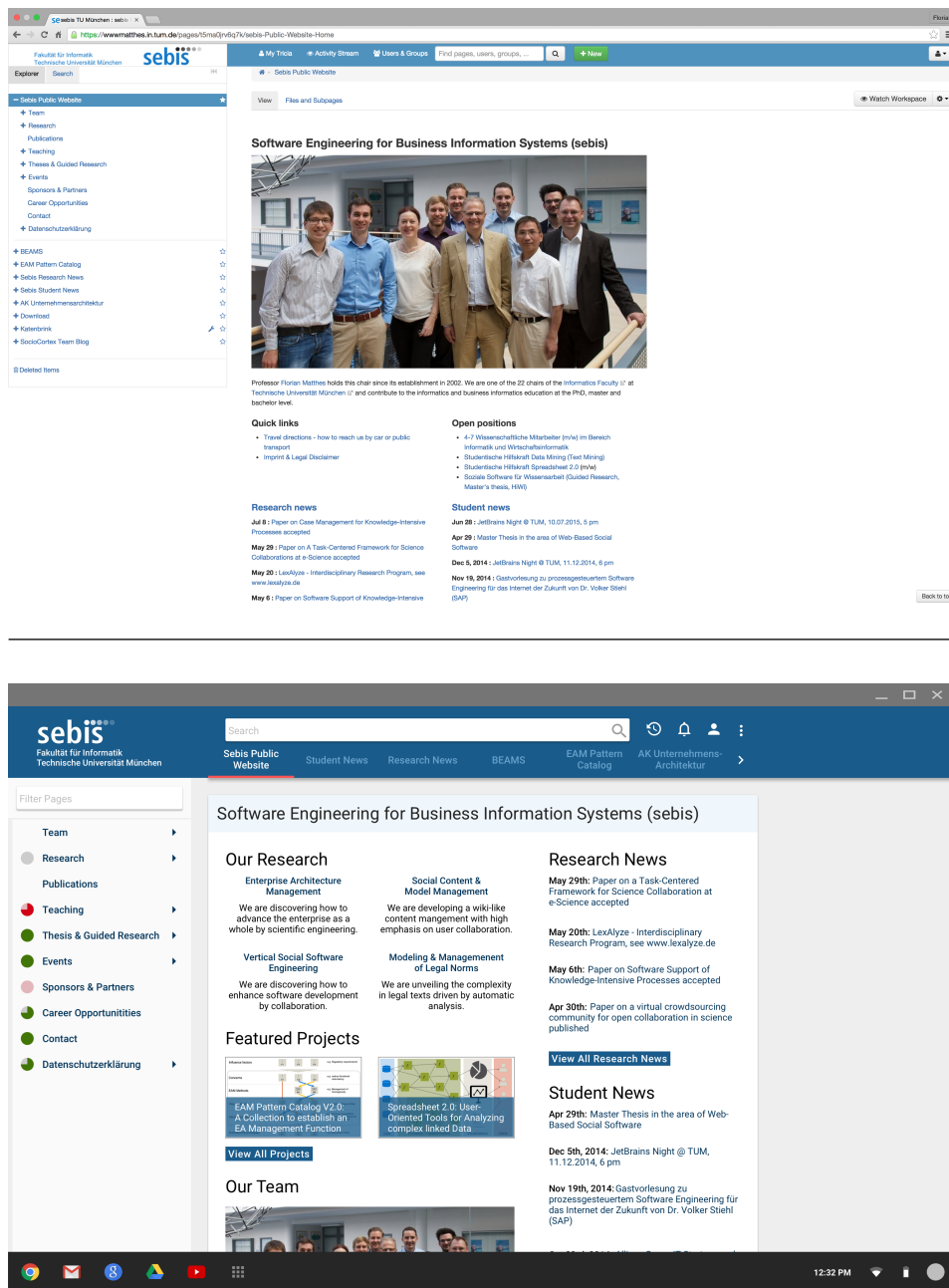


Figure 4.2.: Tricia vs. the Generic Web Application: The Homepage⁶

⁶Tricia's web interface from <https://wwwmatthes.in.tum.de/pages/t5ma0jrv6q7k/sebis-Public-Website-Home> (accessed July 31st, 2015)

4.4. Navigation

The navigation plays a major role in web applications. It guides users through an application, helps them to find what they seek and explore the application. But most importantly it defines the web application's shape and form, however users should not realize it exists.

4.4.1. Data Model

As described in chapter 2, SocioCortex is based on Tricia's data representation. Thus this application uses the same scheme as Tricia: There exist many workspaces, which contain subpages. They act as containers for subpages of a common subject. The subpages are organized in a composite pattern, which generates tree-structured data of subpages. Both workspaces and subpages contain a page that visualizes the current subpage or workspace in the wiki content section. Therefore the wiki content cannot be empty because there is either a workspace respectively underlying subpage selected or some action from the action bar.

4.4.2. Application Bar

This paragraph refers to Figure 4.3. In the top-left part of the application bar there is the chair's logo. Center-aligned at the top, the search bar enables users to search for content and enables experienced users to quickly navigate through the content. Right next to it are four action items. The first item provides access to an activity feed of changes and discussions. The second one shows an alarm indicator when a task is overdue with the number of overdue tasks and the third icon redirects to their own user profile. They are all explained further in the following sections. Below this, a tab section lists all available workspaces horizontally. When users want to browse a different workspace, they need to click on the label that is displayed. The currently selected workspace is underlined in red and stronger white font than the remaining workspaces to make the selection obvious for the user. If there is no currently selected workspace, then all tab labels are a dull white without any red line.

4. Design of a Generic Web Application's User Interface

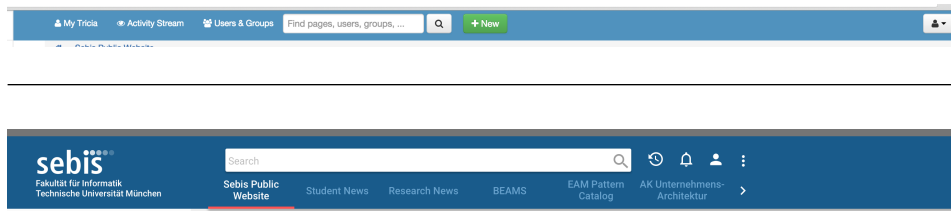


Figure 4.3.: Tricia vs. the Generic Web Application: The Application Bar⁷

As principle *N7* suggests, an application bar needs to contain a logo, sections, utilities and search. It is a conventional way (*G4*) of demonstrating users their choices. With clearly defined areas (*G5*) already in the application bar, users feel comfortable. Tricia has a similar top bar, however it does not contain the logo and the sections. These two are displayed in Tricia's sidebar, where they logically do not belong (*G6*). In Tricia the current workspace is not explicitly highlighted as shown in the next section, but workspaces in the generic web application's user interface are underlined with red (*N8*).

In case there are too many workspaces to fit within the horizontal space, an open, white caret is displayed to the right. Figure 4.4 shows how this caret can be selected, which then opens a menu, where users can select any other available workspace. Users, who are registered and logged in, also have the possibility to reorder the workspaces according to their own needs. They simply drag and drop the workspaces from the list of all available workspaces to the tab section of the action bar and move these again around.

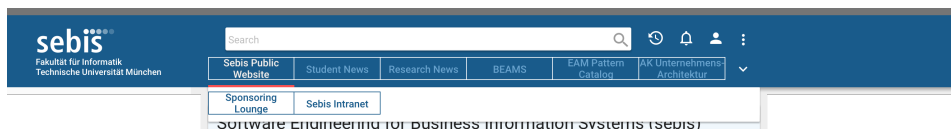


Figure 4.4.: The Generic Web Application: More Workspaces shown

In Tricia, users only have the possibility to pin workspaces to the top. This feature does not enable reordering of the workspaces. It is available in the sidebar, therefore it is covered in the next section.

Four actions, hidden below the three dots (more-icon) at the very right, offer additional functionality. Figure 4.5 shows the actions "New Subpage" to create a new subpage in the current position, "New Workspace" to create a new workspace, "Watch Current Page" to get notified on changes of the page and

⁷Tricia's web interface from <https://wwwmatthes.in.tum.de/pages/t5ma0jrv6q7k/sebis-Public-Website-Home> (accessed July 1st, 2015)

"Share Link" to send an email with the current page's link to someone. Un-registered users only see the profile icon, which they select for registration or log-in. The design of the action bar is defined through Material Design as best practice. It is "used for branding, navigation, search, and actions" [Go15b].

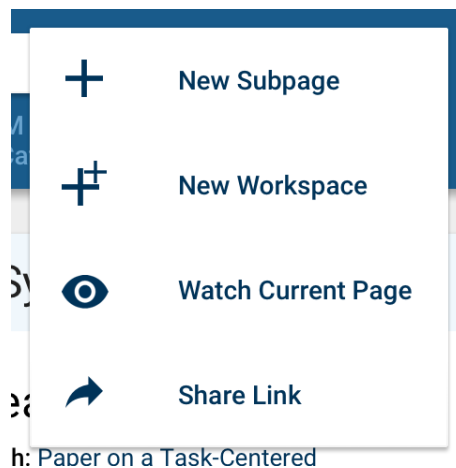


Figure 4.5.: The Generic Web Application: More-Icon clicked

When users choose to create a new subpage in Tricia, they need to press the green "+New" button. It opens a dialog with includes instructions, because they do not initially know what is newly created. By explicitly labeling it "New Subpage" and "New Workspace" users know what this action does (*G9*).

4.4.3. Sidebar

The sidebar contains all subpages of the current workspace. We found out in analyzing Google Analytics of the current Sebis chair web application, users primarily stay within one workspace when they visit the page. Additionally, the search bar is only rarely used. Because this is the case, the sidebar becomes the main choice of navigation within one stay on the website. In Figure 4.6, the first hierarchy of subpages below the workspace "Sebis Public Website" is vertically aligned and displayed. To the left of the labels are circular percentage indicators that show how much of the tasks of the subpage and its underlying subpages has been completed. The color shows the present state of the task. To the right of some subpages, there is a filled caret pointing to the right, which indicates that it again has subpages. A filter box lets users filter all subpages down to the leaves of a hierarchy.

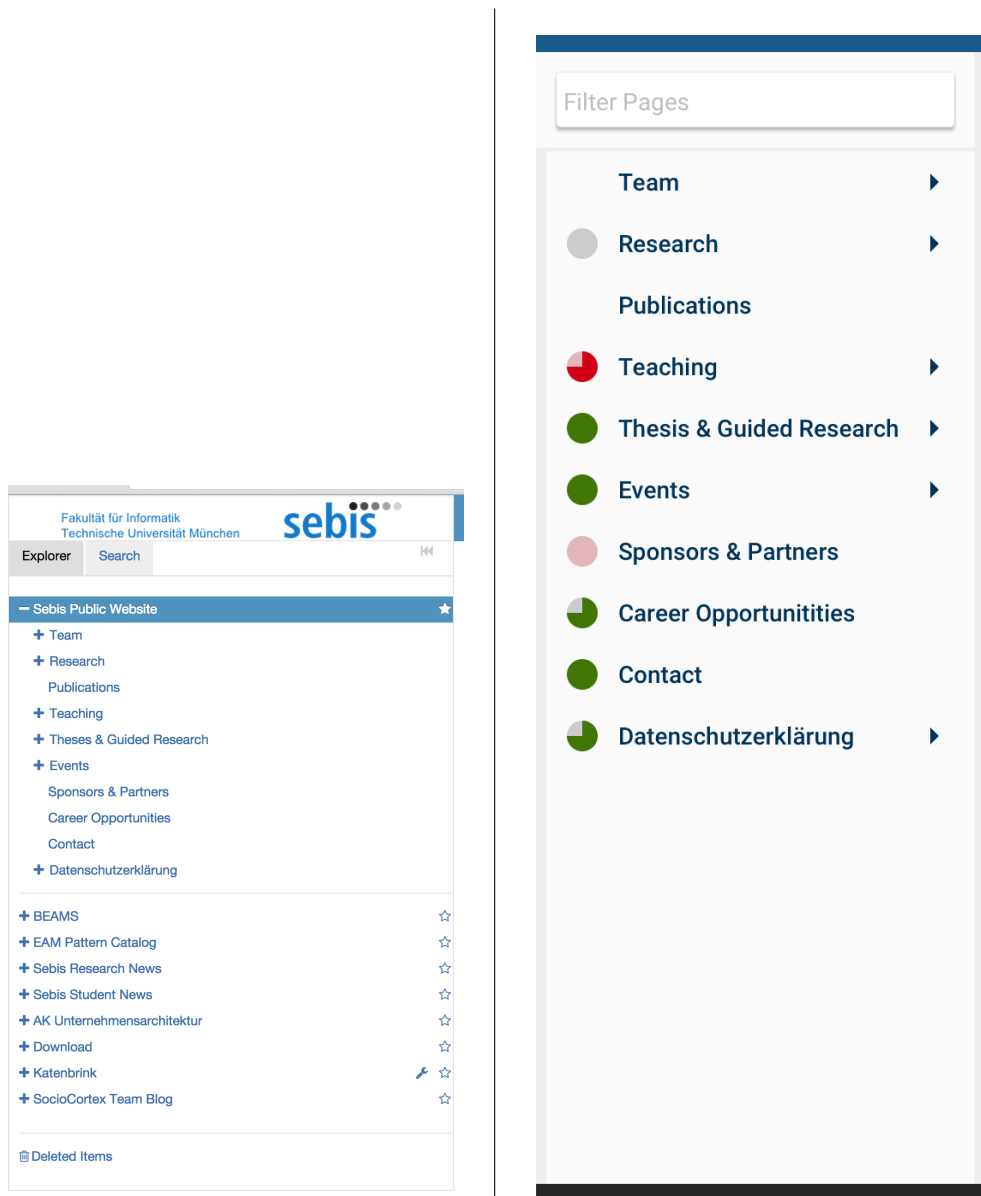


Figure 4.6.: Tricia vs. the Generic Web Application: The Sidebar⁸

As mentioned earlier, workspaces are moved in the generic web application, compared to Tricia, from the sidebar to the application bar and thus making a visual and logical cut ($G6$). This takes the Google Analytics results greatly into account and was decided after many switches in order to, for example, remove a layer of the hierarchy in the sidebar. The filter box on top of the tree navigation is a time-saving shortcut for users who know what they are looking for. This can save a lot of clicks compared to Tricia. Indicating task progress

⁸Tricia's web interface from <https://www.matthes.in.tum.de/pages/t5ma0jrv6q7k/sebis-Public-Website-Home> (accessed July 31st, 2015)

4. Design of a Generic Web Application's User Interface

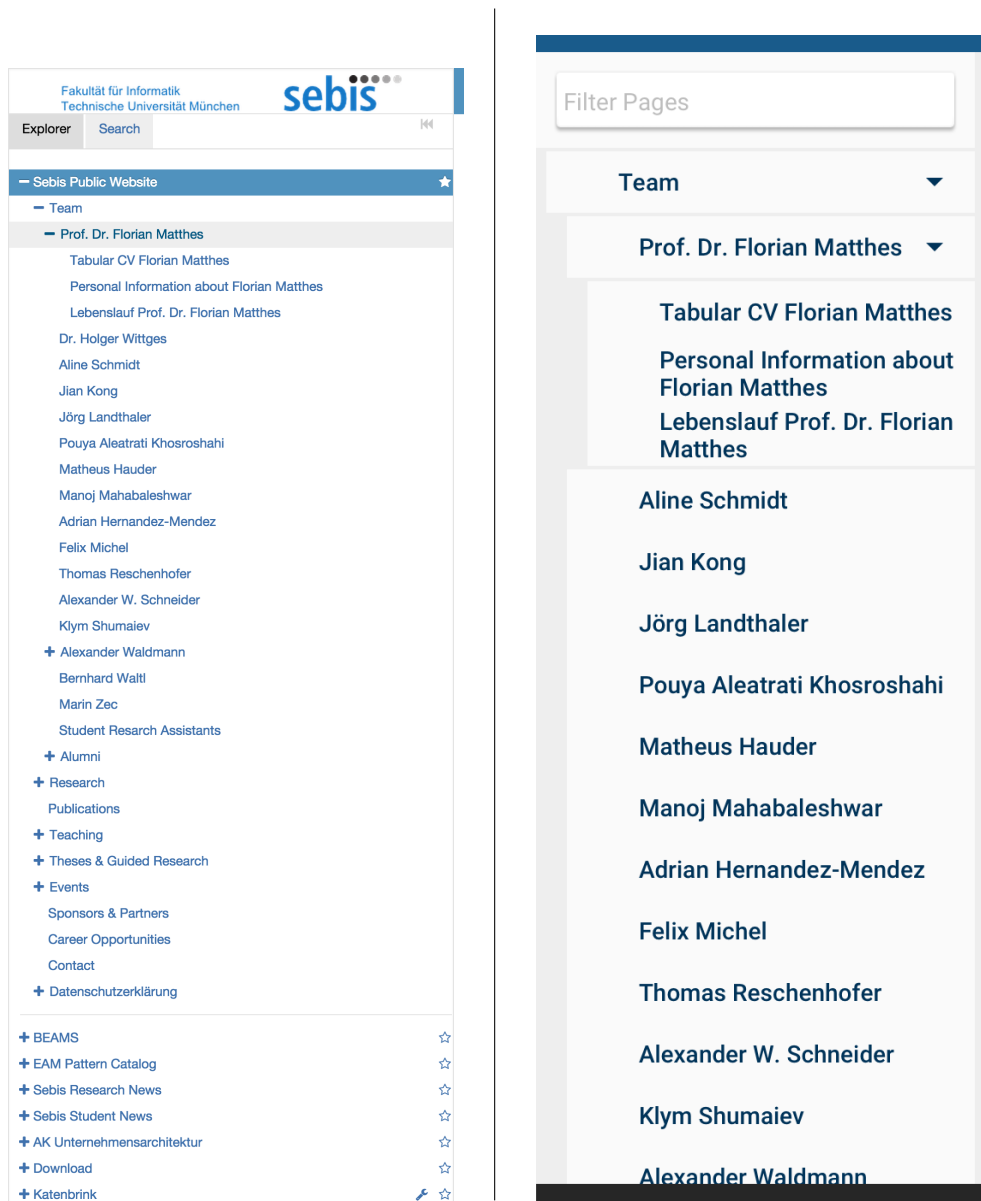


Figure 4.7.: Tricia vs. the Generic Web Application: Caret Navigation⁹

already in the sidebar greatly improves collaboration ($U1$). By seeing this on navigation, users get reminded and see where issues occur.

If users press the caret to the right of the label, the subpage does not load, because it simply acts as convenient navigation through the tree. The caret spins clockwise for 90° , thus pointing to the bottom and a more indented area. So if users press on the "Team" caret, then all team members expand

⁹Tricia's web interface from <https://wwwmatthes.in.tum.de/pages/t5ma0jrv6q7k/sebis-Public-Website-Home> (accessed July 31st, 2015)

below. This pattern is continuously applied to all hierarchies of subpages within a workspace, so when users press on the "Prof. Dr. Florian Matthes" caret next, this splits "Prof. Dr. Matthes" from "Aline Schmidt" and shows the subpages. Figure 4.7 shows how the sidebar looks when this is done. This pattern enables users to quickly orientate themselves in the side bar navigation and see the navigation they pursued.

This change in navigation indicates the user's movement and progress (*N4*). The visual differentiation between different hierarchies is much clearer in the generic web application's user interface than in Tricia, which helps users to find content even faster, even though these change are only small (*N3*). Both Tricia and the generic web app indent lower hierarchies in order to show what is part of what (*G7*). Another behavior that is expected by users is that the main navigation leads within the current side. As both the application bar and sidebar leads users within the generic web app, this is fulfilled (*N5*).

When users click the subpage's label, the subpage is loaded in the wiki content section and the current subpage's background is colored in a strong blue in the sidebar and the indent is filled in red, whereby the font color and caret color change to white. So in case users click on "Prof. Dr. Florian Matthes" it looks like in Figure 4.8.

In comparison to Tricia the active subpage and workspace are presented in the generic web application's user interface more clear. By using the accent color for both the application bar and sidebar, users know immediately where they are (*N2*), making them prominent (*G5*).

The sidebar's scrolling mechanisms are decoupled from the wiki content's scrolling. In order to scroll either one, users have to use scrolling gestures in the according area or hover their mouse over the area and scroll. In the sidebar it works when scrolling down as follows: The lowest hierarchy disappears while the parent hierarchy's lower part pushes up and therefore covers the lowest hierarchy. In this case of Figure 4.8 the lower part starting with "Aline Schmidt" goes up until it fully covers "Tabular CV Florian Matthes" and the two subpages below. If, in the middle of this, users decide to scroll up again, the lowest hierarchy part stretches out vertically again until it expands to its previous height. As soon as the lower part with "Aline Schmidt" hits the upper part, which only contains the single subpage "Prof. Dr. Florian Matthes", the two parts consolidate and the caret to the right of "Prof. Dr.

4. Design of a Generic Web Application's User Interface

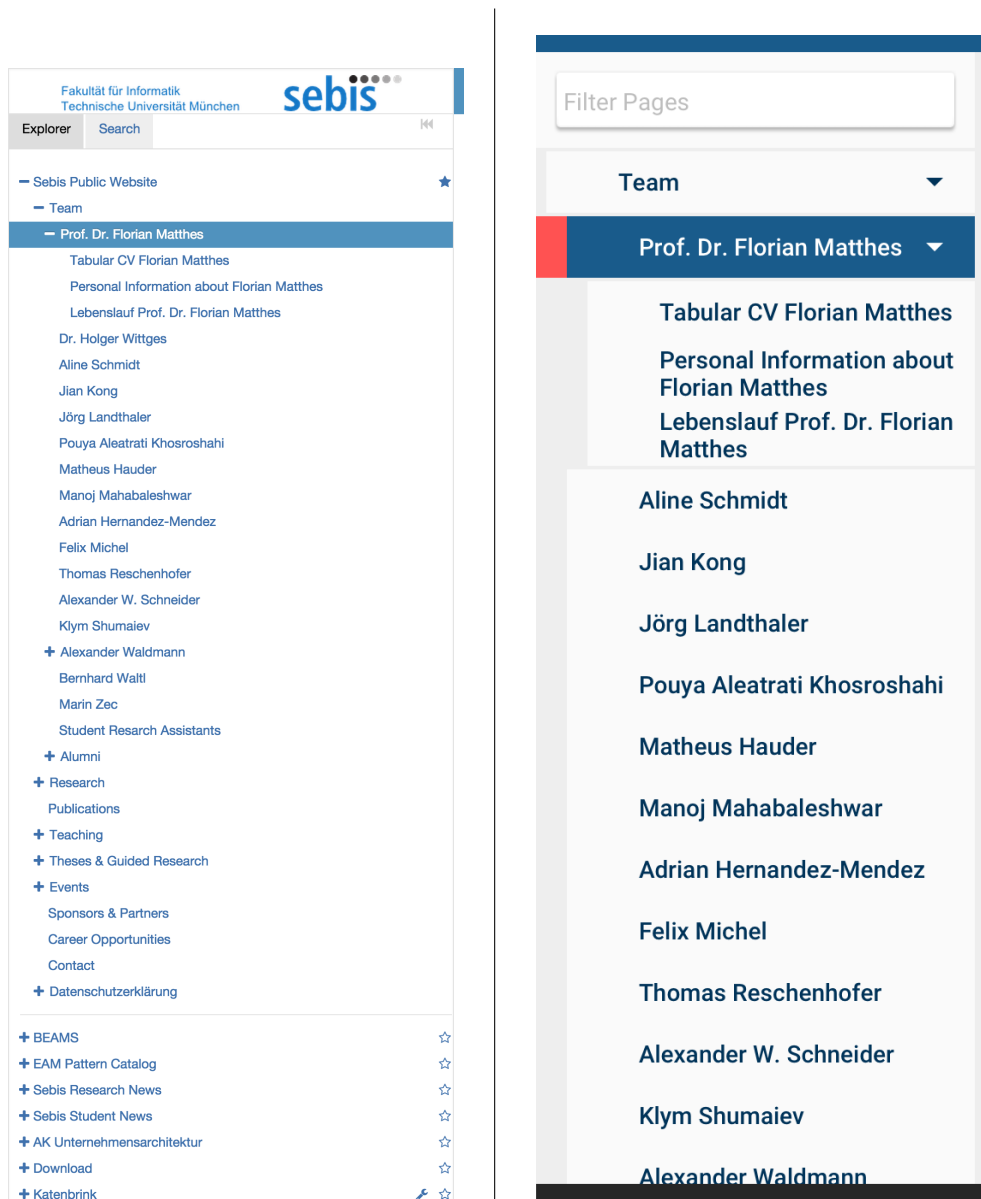


Figure 4.8.: Tricia vs. the Generic Web Application: Sidebar Navigation¹⁰

Florian Matthes" turns counter-clockwise in order to point to the right. Now they cannot be separated by scrolling anymore, but only by clicking the page's label or caret again. This pattern is again applied to all hierarchies. Figure A.2 in the Appendix displays the progress of scrolling in Figure 4.8 in five steps up.

This scrolling mechanism comes along well with the sidebar design. The sidebar scrolls independently from the wiki content and therefore a plain scrolling

¹⁰Tricia's web interface from <https://wwwmatthes.in.tum.de/pages/88bkmvw6y7gx/Prof.-Dr.-Florian-Matthes> (accessed July 31st, 2015)

mechanism as in Tricia is too tedious in usage for users. The separation is useful, though, because many levels of hierarchy would lead to an unclear sidebar, whereas this scrolling mechanism collapses unutilized subpages.

4.5. Wiki Content

In section 4.3, the application's frame was described to be consisting of three parts. In the previous section, the application bar's and sidebar's functions and methods were characterized. This section deals with part three: the wiki content. In the following wiki content is also called pages or wiki pages.

4.5.1. Data Model

Pages are attached to workspaces and subpages. They are based on a type definition, which is instantiated by a page and predefines different aspects such as the structured data that is available by default. Types for pages also have a type definition, which can be instantiated by the page and the instance can be modified. For example attributes can be added or removed. These pages display the current content and contain both structured and unstructured information. Structured information is the list of attributes and/or tasks, if available. Attributes have a name and a value. The value can be of the following type: text, boolean, file, date or a reference. Attributes can be grouped to attribute groups, so they can be displayed in an organized way with a caption. Just like in Darwin and the Organic Data Science Framework, attributes can be attached to tasks. Tasks also contain metadata about the progress, start and end date, the owner and expertise that can be gained upon completion. They can contain subtasks that influence the parent task's percentage and state, hence if for example a subtask is overdue, the parent task is overdue as well. The task's progress can be either automatically calculated or based upon completion of the attached attributes. Completion of metadata is not considered in the calculation, as it is for example the case in the Organic Data Science Framework (see section 2.3). A task in general can have the same states as in Darwin (ignoring inconsistencies) as in Table 4.9. The rest of the page's body content is unstructured information.

The tasks' colors were chosen, because gray, green and red established well in both Darwin and the Organic Data Science Framework and feel natural to users (*G4*).






Color	Progress
	Task not started
	Task completed
	Task partially completed
	Task overdue
	Task overdue and partially completed

Figure 4.9.: The Generic Web Application: Task States

4.5.2. Unstructured Information

When users open the generic web application, the homepage content is displayed like in Figure 4.10, which in this case is the "Sebis Public Website's workspace page.

Every page in the web application consists of a header and a body, sometimes even a footer. The header is the blue area at the top and the body the white. The simplest form of a header is just a light-blue background with a black heading, the same heading of the subpage in the sidebar. If it is a workspace page, the heading does not need to be the same. Every wiki page is editable with a WYSIWYG¹¹ editor in rich-text format, so users can write paragraphs, insert images, add headings and references etc. This makes wiki pages highly interlinked and well readable. If there are any attributes or tasks (structured information) attached, a gray area right inside the page is displayed.

It is crucial that the name of the heading matches the name in the sidebar (*W2*). This assures users that they are on the right page. The generic web application's user interface emphasizes the heading more than Tricia, thus making it easier for users to find the key element (*G5*). The WYSIWYG editor makes it easy for users to create formatted text (*W1*). This is inherited

¹¹WYSIWYG = What you see is what you get

4. Design of a Generic Web Application's User Interface

The screenshot shows the homepage of the Software Engineering for Business Information Systems (sebis) website. At the top, there is a navigation bar with 'View Files and Subpages' and a 'Watch Workspace' button. Below this is a large group photo of the team. Underneath the photo is a paragraph of text about the chair's history and its contribution to education. The main content area is divided into several sections: 'Quick links' with travel directions and legal disclaimers; 'Open positions' listing various roles; 'Research news' with several recent publications; and 'Student news' with information about conferences and theses. A 'Back to top' button is located at the bottom right of the content area.

This screenshot shows a more structured and modern version of the sebis website homepage. The layout is clean and organized into distinct sections. At the top, the title 'Software Engineering for Business Information Systems (sebis)' is prominently displayed. Below the title, there are three main columns. The left column, 'Our Research', features three sub-sections: 'Enterprise Architecture Management', 'Social Content & Model Management', and 'Vertical Social Software Engineering', each with a brief description of the research focus. The middle column, 'Featured Projects', includes two project cards: 'EAM Pattern Catalog V2.0: A Collection to establish an EA Management Function' and 'Spreadsheet 2.0: User-Oriented Tools for Analyzing complex linked Data'. The right column, 'Research News', lists several recent publications with dates and titles. Below the research news, there is a 'View All Research News' button. At the bottom left, there is a 'Our Team' section with a group photo. A 'View All Projects' button is also present. The overall design is more professional and easier to navigate than the previous version.

Figure 4.10.: Tricia vs. the Generic Web Application: Homepage Wiki Content¹²


4. Design of a Generic Web Application's User Interface

View Files and Subpages Versions 0 Comments Watch

Prof. Dr. Florian Matthes

Last modified by Florian Matthes Jun 4

[curriculum vitae](#)
[staff](#)
[sebis](#)
[lebenslauf](#)
[team](#)
[matthes](#)
[photo](#)


Chair Informatics 19
 Software Engineering for Business Information Systems
 Institut für Informatik
 TU München
 Boltzmannstrasse 3
 D-85748 Garching bei München
 How to get to Garching
 Room: 1.12.54 (click for campus maps)

Florian Matthes holds the chair **Software Engineering for Business Information Systems** at Technische Universität München. The current focus of his research is on technologies driving the digital transformation of enterprises and societies: Enterprise architecture management, social content and model management, and semantic modeling of legal texts (LexAlyze).

As head of the software architecture working group of the Gesellschaft für Informatik, member of the advisory board of the Ernst Denert-Stiftung für Software Engineering and organizer of several workshops and conferences in the area of enterprise architecture he puts special emphasis on the cooperation between practitioners and scientists in informatics and information systems.

Since 2014 he is extending this theory-based and practice-oriented cooperative work to also include scientists and practitioners from the legal domain to foster a better shared understanding of the interaction between informatic, economic and legal models of an increasingly digital society.

He is co-founder and chairman of **CoreMedia** (1996) and **infoAsset** (1999), co-founder of further small software and service university spin-off, and scientific advisor of **UnternehmerTUM**, the center of innovation and business creation at TU München.

Earlier stations of his academic career are the **Goethe-University Frankfurt** (Diploma 1988) the **University of Hamburg** (PhD 1992), the **Digital Systems Research Center** (now HP SRC Classic) in Palo Alto, USA (Researcher 1992-1993), and the **Technical University Hamburg-Harburg** (Associate Professor 1997-2002).

Until 2010 he served as dean of studies at the **Faculty for Informatics** and member of the teaching board of TU München.

Florian and his wife Nastaran Matthes are proud parents of two daughters.

Academic Information


- German curriculum vitae
- Tabular curriculum vitae
- Publications: sebis, DBLP, Google Scholar
- Current research projects
- Completed enterprise architecture management projects
- Completed social software engineering projects
- Completed persistent programming language projects
- Technology transfer
- Teaching experience

Follow me Back to top

Attributes of this Team Member	
Position	Full Professor
E-Mail	matthes [at] in.tum.de
Phone	+49 89 289 17132
Fax	+49 89 289 17136
Room	01.12.054
Secretary	Aline Schmidt
LinkedIn	http://de.linkedin.com/pub/florian-matthes/21/52a/855
Xing	https://www.xing.com/profile/Florian_Matthes
Skype	f1matthes
Twitter	@matthes
Image File	Florian Matthes 2014.JPG

Prof. Dr. Florian Matthes

[sebis](#)
[staff](#)
[curriculum vitae](#)
View


Chair Informatics 19
 Software Engineering for Business Information Systems
 Institut für Informatik
 TU München
 Boltzmannstrasse 3
 D-85748 Garching bei München
 How to get to Garching
 Room: 1.12.54 (click for campus maps)

Florian Matthes holds the chair **Software Engineering for Business Information Systems** at Technische Universität München. The current focus of his research is on technologies driving the digital transformation of enterprises and societies: Enterprise architecture management, social content and model management, and semantic modeling of legal texts (LexAlyze).

As head of the software architecture working group of the Gesellschaft für Informatik, member of the advisory board of the Ernst Denert-Stiftung für Software Engineering and organizer of several workshops and conferences in the area of enterprise architecture he puts special emphasis on the cooperation between practitioners and scientists in informatics and information systems.

Since 2014 he is extending this theory-based and practice-oriented cooperative work to also include scientists and practitioners from the legal domain to foster a better shared understanding of the interaction between informatic, economic and legal models of an increasingly digital society. He is co-founder and chairman of **CoreMedia** (1996) and **infoAsset** (1999), co-founder of further small software and service university spin-off, and scientific advisor of **UnternehmerTUM**, the center of innovation and business creation at TU München.

Earlier stations of his academic career are the **Goethe-University Frankfurt** (Diploma 1988) the **University of Hamburg** (PhD 1992), the **Digital Systems Research Center** (now HP SRC Classic) in Palo Alto, USA (Researcher 1992-1993), and the **Technical University Hamburg-Harburg** (Associate Professor 1997-2002).

Until 2010 he served as dean of studies at the **Faculty for Informatics** and member of the teaching board of TU München.

Florian and his wife Nastaran Matthes are proud parents of two daughters.

Team Member	
Attributes	
Position	Full Professor
E-Mail	matthes [at] in.tum.de
Phone	+49 89 289 17132
Fax	+49 89 289 17136
Room	01.12.054
Secretary	Aline Schmidt
LinkedIn	http://de.linkedin.com/...
Xing	https://www.xing.com/...
Skype	f1matthes
Twitter	@matthes
Attribute name	Attribute value

Figure 4.11.: Tricia vs. the Generic Web Application: The Wiki Page¹³

¹³Tricia's web interface from <https://wwwmatthes.in.tum.de/pages/88bkmvw6y7gx/Prof.-Dr.-Florian-Matthes> (accessed July 31st, 2015)

from Tricia and many other web applications. The clear distinction between header, body and the structured information makes it easier for users to understand the difference between structured and unstructured content (*G8*). The differentiation is weak in Tricia. The clear segmentation and formatting of content makes it a lot easier for users to find what they are looking for. The most important topics are research, projects, news and the team. Therefore they are promoted on the homepage (*W3*).

Figure 4.11 shows the team member's page "Prof. Dr. Florian Matthes". Unstructured information floats around the structured information. This page has a more advanced header, as there are tags and different views available.

The floating of unstructured information in the generic web app is possible through the clear differentiation of the two, which has the great effect of saving expensive vertical space, which is lost in Tricia. Keeping the tags in the header makes it clear that they are not unstructured information (*W4*). Again this is possible through the clearly defined areas.

When users select the "View" button, they can redirect to see the attachments, different versions of the page, comments made to this page or change the settings (see Figure 4.12).



Figure 4.12.: The Generic Web Application: View-Menu selected

The consolidation of these four items in a menu cleaned up the header. In Tricia all options are visible at the same time, which again overwhelms users with information (see Figure 4.11). As they are not primarily accessed it is helpful for users if they are hidden (*G1*).

¹²Tricia's web interface from <https://www.matthes.in.tum.de/pages/t5ma0jrv6q7k/sebis-Public-Website-Home> (accessed August 1st, 2015)

4. Design of a Generic Web Application's User Interface

Finally, Figure 4.13 displays the footer of a page, which consists of a "last modified by" ending with a reference to the user, who modified it.

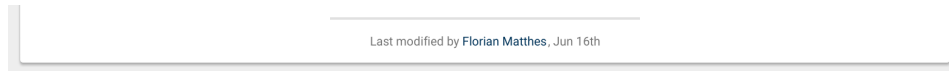


Figure 4.13.: The Generic Web Application: The Last modified Footer

4.5.3. Structured Information

Different from the homepage is that the page "Prof. Dr. Florian Matthes" has attributes attached. They float in the gray area right to the wrapping text (see Figure 4.14. At the top of the area, the type name is displayed, so users can categorize this page and understand faster what kind of page it displays. On the left of the attribute box are the names and on the right the values of the attribute. At the very right, these attributes can be edited and at the bottom, they can be added to this page.

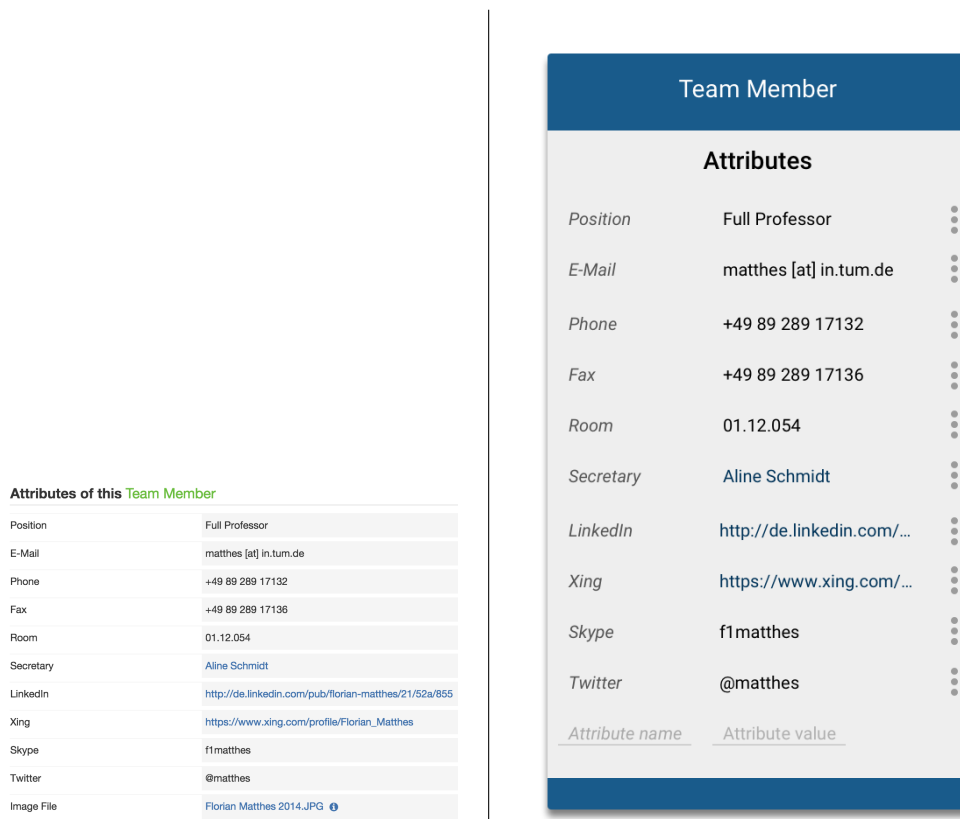


Figure 4.14.: Tricia vs. the Generic Web Application: Attributes¹⁴

4. Design of a Generic Web Application's User Interface

Tricia uses a lot of unused horizontal space for the attribute names. In the generic web application's user interface it is easy to modify, add and remove attributes. For users who need these features a lot, it is valuable to have it within so few clicks and for users, who do not need it, it is so unobtrusive (*G5*) that it does not distract from the content. They are self-explanatory and therefore do not need any further instructions (*G9*).

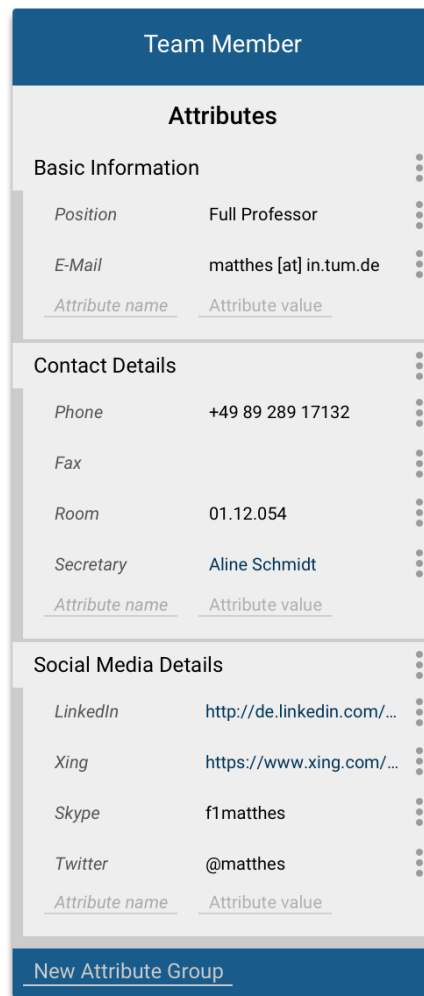


Figure 4.15.: The Generic Web Application: Attribute Groups

The example of Figure 4.15 shows again the "Prof. Dr. Florian Matthes" page, but this time ordered in three attribute groups: "Basic Information", "Contact Details" and "Social Media Details", which again contain different attributes of the types text, reference and date. When users would like to add new attributes to an attribute group, they can add the name in the "Attribute

¹⁴Tricia's web interface from <https://wwwmatthes.in.tum.de/pages/88bkmvw6y7gx/Prof.-Dr.-Florian-Matthes> (accessed July 31st, 2015)

4. Design of a Generic Web Application's User Interface

name" field, the value in the "Attribute value" field and submit this change by pressing enter. This is the same pattern as with attribute groups and tasks.

Attribute groups aid users in ordering and later finding attributes, which logically belong together (G6). The indenting shows what is part of what (G7). In Tricia there is not way to group attributes, which can become confusing as pages sometimes have even more attributes.

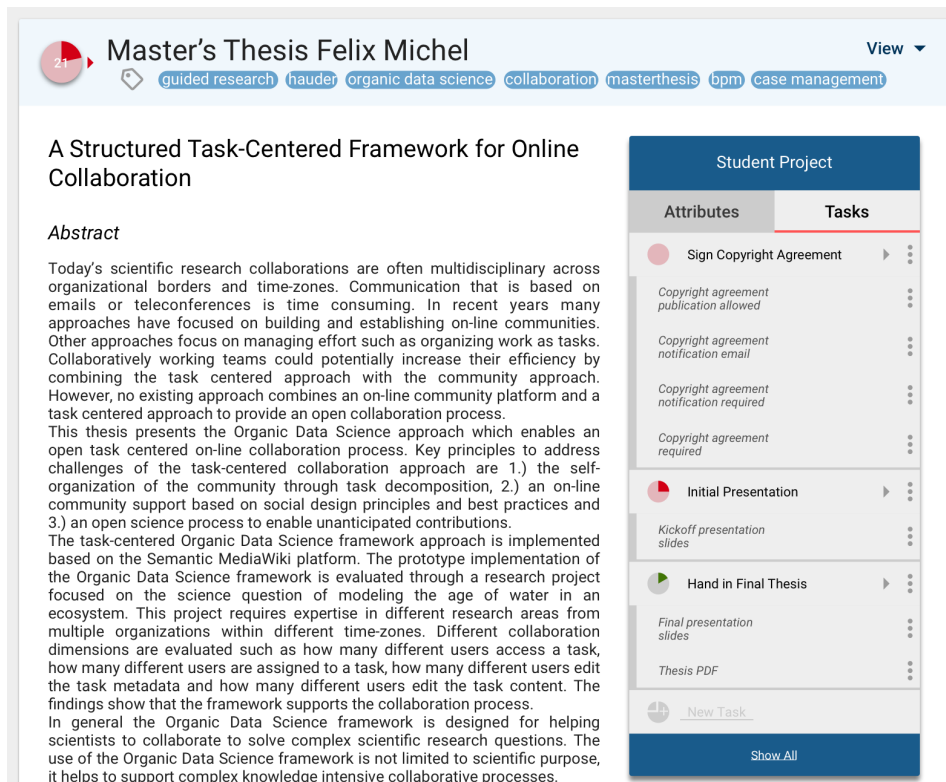


Figure 4.16.: The Generic Web Application: Master's Thesis

Figure 4.16 displays another wiki page of someone's master's thesis. In this case, the page has not only attributes, but also tasks attached. On the left side of the header there is an overall task percentage indicator that aggregates the tasks and subtasks of this page and calculates a total. It also has another utilization, which is shown later. On the right there is again the gray attribute box, but this time it is displayed with tasks, as the task tab is selected. this view shows only the current tasks. Clicking on the attributes tab shows all attributes belonging to this page instead of the tasks. If there are any subtasks, then a reference is included in the attributes at the bottom of the list. At the bottom of the area users can create new tasks and attach them directly to the wiki page.

4. Design of a Generic Web Application's User Interface

The clear separation of tasks and attributes, but still their visual proximity suggests users that they are connected in some way (*G6*). As the model states attributes can be attached to tasks, therefore the indentation shows they are a part (*G7*). Tasks has a similar layout as the attribute groups (*N3*), which makes it easy for users to find what they are looking for. Grouping them in tabs, instead of creating several pages, reduces the amount of jumps (*N1*).

When users press a task name or its caret, which then spins clockwise and the name becomes bold, then it expands and shows both the metadata and a line to add attributes to the task below the attributes. Figure 4.17 shows the task "Show Copyright Agreement", when it is clicked. The elements on the white background display the metadata.

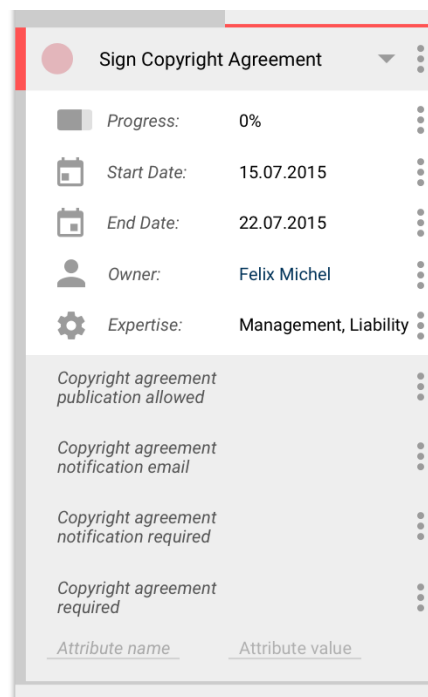


Figure 4.17.: The Generic Web Application: Selecting a task

First, only a task's name, its progress/state and the attached attributes are shown, because it enhances the clarity and users are not confronted with too much data at once (*G1*). Metadata is only shown on click, as it is not the predominant data users want to access. The selected task is highlighted with the accent color next to the header of the task. It uses the same pattern as subpages and workspaces, which helps users find the active element (*G4*, *N8*).

Figure 4.18 shows how the generic web app supports users by suggesting them with auto-completion what attributes are most likely to be used here (*S1*). Because this is a client for the end-user, there is no field for users to specify the type. The application tries to guess the type instead.

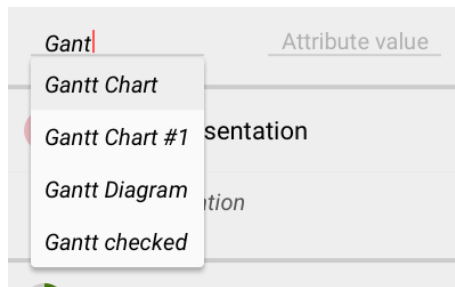


Figure 4.18.: The Generic Web Application: Autocompleting an Attribute

For editing the task content there are three possibilities: edit tasks, edit metadata and edit attributes. Figure 4.19 shows the menus for all three. A menu is shown when users click on the more-icon next to the content. The first menu shows the available functions for a task: "Edit" for editing the task name, "Delete" for deleting the task, "Delegate" for forwarding the task to another user and "Skip" for not having to complete the task. The second menu shows functions for attributes, which can also be edited and deleted, but when "Edit" is clicked both the name and label can be edited. The last menu is for metadata. The five fields for metadata are always attached to a task, therefore users can only edit the values by clicking "Edit". Edit changes are applied by pressing enter.

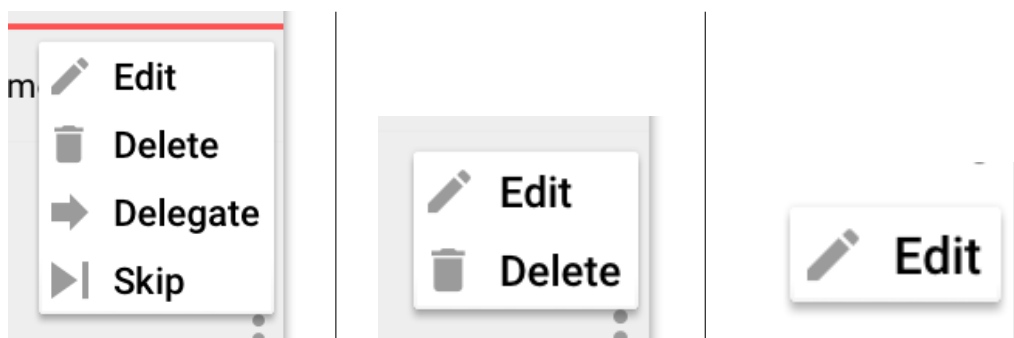


Figure 4.19.: The Generic Web Application: The Edit Menus

The menu entries show both text and an icon, as it makes it is the fastest way for users to recognize their options and avoids ambiguity [Be15]. Even though

4. Design of a Generic Web Application's User Interface

metadata has only one option, it is important to keep the same pattern of showing a menu by clicking on the more-icon (G_4).

Current Tasks: Tasks where today is greater than start date and incomplete

Future Tasks: Tasks where today is smaller than start date and incomplete

Completed Tasks: Tasks that are completed

At the bottom of the gray box, there is a "Show All" button, which users can select to show all three categories of tasks. Figure 4.20 shows how they are apportioned in the three categories. It is separated in two columns, as it would be too high otherwise.

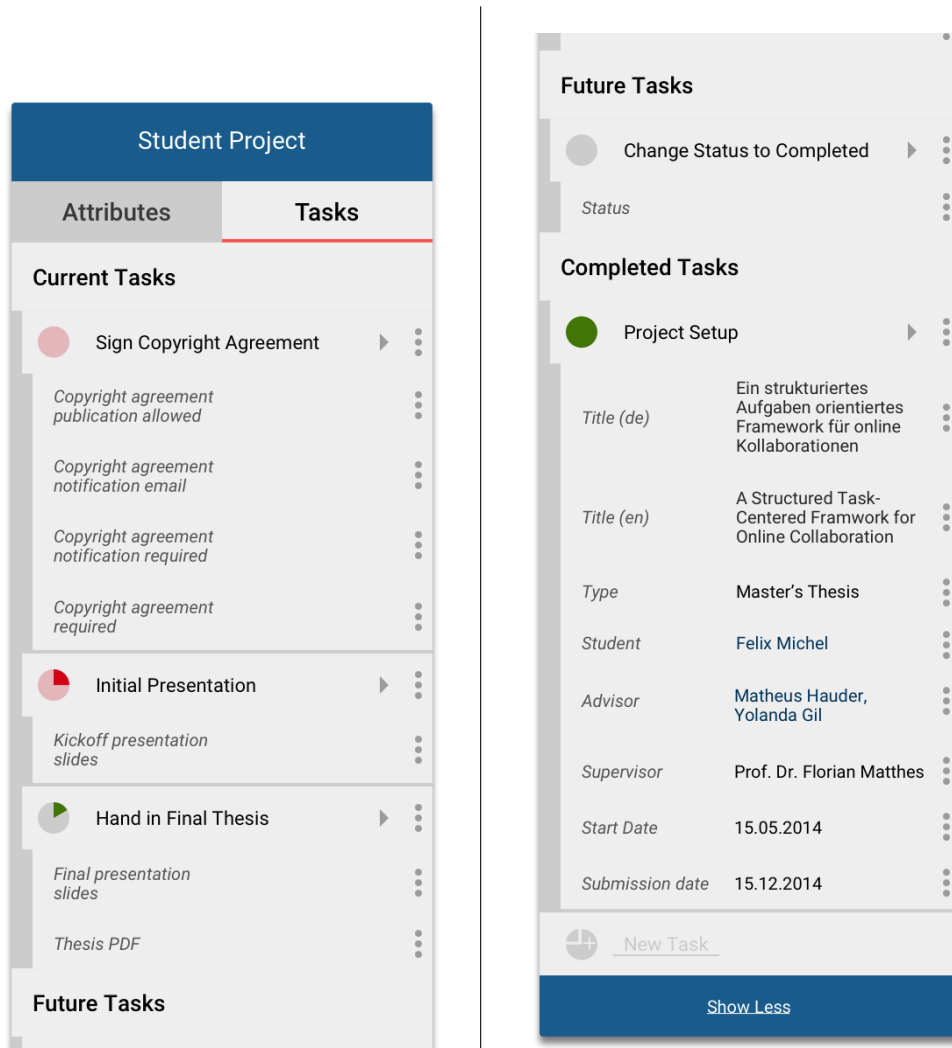


Figure 4.20.: The Generic Web Application: All Tasks shown

The generic web application includes the possibility to visualize the process of all tasks in a Gantt chart. This feature is displayed in Figure 4.21 and derived from the Organic Data Science Framework and Darwin (see chapter 2). If users want to toggle this feature, they need to press the page's percentage indicator in the header. This also automatically opens the task tab and shows all tasks. In this example of a Gantt chart, there are five tasks. The first is completed, the second overdue and not started, the third is overdue and started, the fourth is started and the fifth is not started yet. Users are guided with additional help of "Today" and the vertical indicator below. When every part of a bar left to the indicator is green, they are in time, whereas the start and end date define the length and position of a bar. If the indicator is right of a task and the task is not completed, then it is overdue.

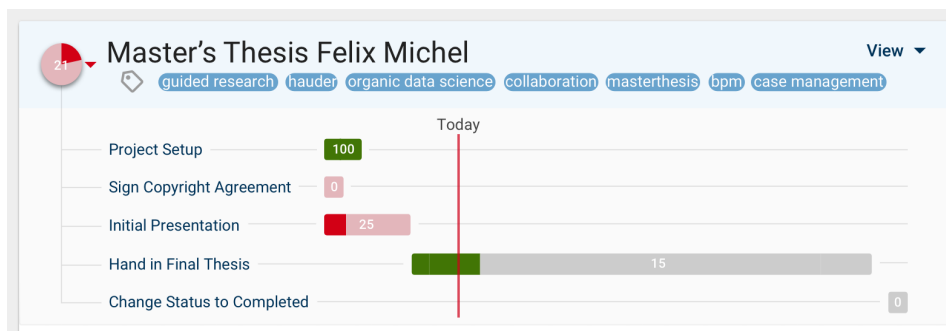


Figure 4.21.: The Generic Web Application: The Gantt Chart

This feature greatly improves the user's understanding of the page's overall progress. Users can see which task should take how much time and therefore partition their work time according to this or catch up with work.

4.6. User Profile

The user profile portrays the first feature that is not displayed within the tree hierarchy of pages and workspaces. Figure 4.22 shows how this feature has a different sidebar than the previous wiki pages and contains a wiki page. By pressing the user-icon in the action bar the own profile page is opened (and the user-icon highlighted in red). In the sidebar, there is the profile picture, the user's name, the email and when the user logged in the last time. Below there is a pie chart of the expertise, which the user earned by completing tasks. As central piece of information, the three task sections "Current Tasks", "Future

4. Design of a Generic Web Application's User Interface

Tasks" and "Completed Tasks" give references to the pages where these tasks are attached to. This makes the user profile the repository of task information and achievements for a single user. The tasks in every section are displayed with their progress and a link to where the task is used as well as the parent page. The last piece of information is about when it is due, it starts or it was completed, which is the most relevant information users need concerning the start and end date.

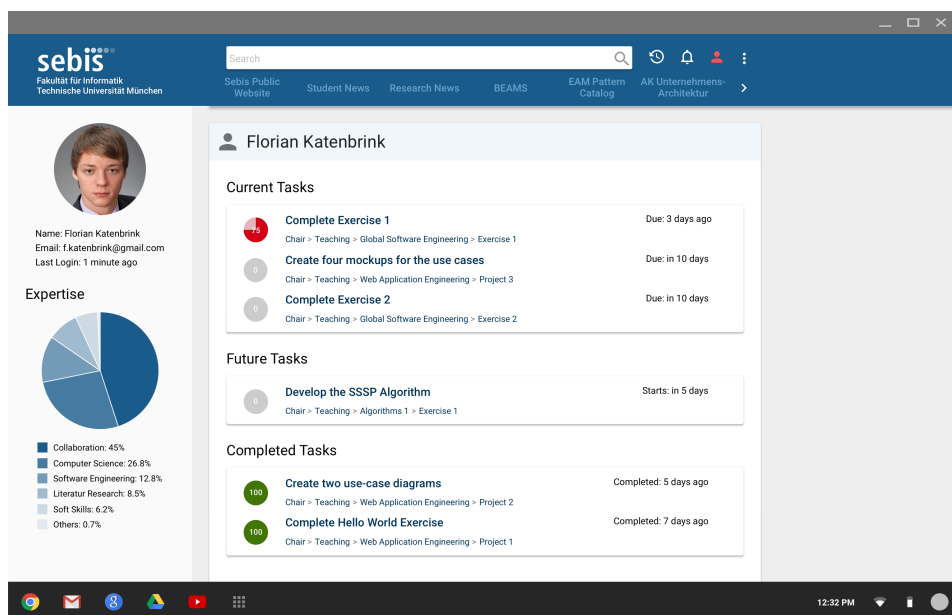
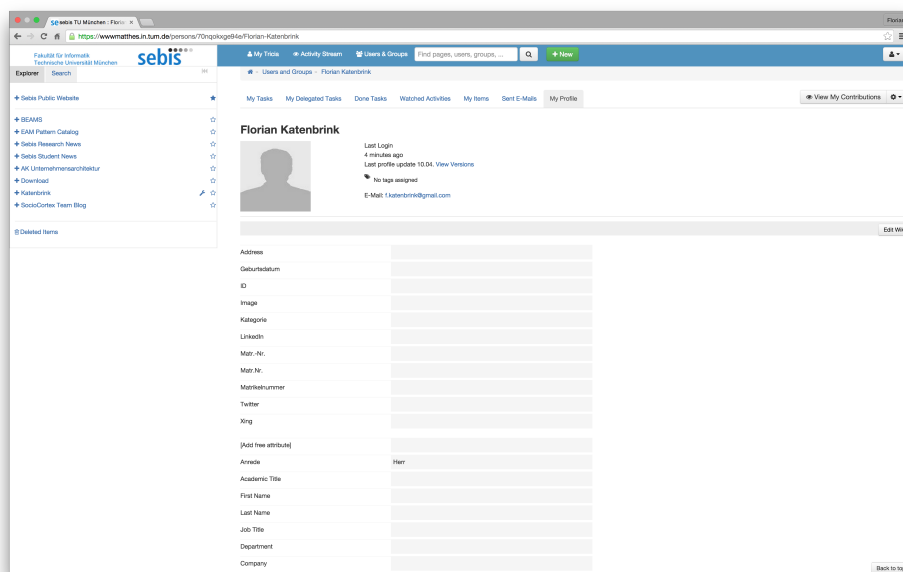


Figure 4.22.: Tricia vs. the Generic Web Application: The User Profile¹⁵

Name, email and last login are enough attribute information for the use case of this web application. In Tricia there is a large amount of (redundant) attributes on the user profile, even though it is not used (*U2*). The three information in the generic web app are guaranteed to be always there, because they are demanded on registration (*U1*). The different views in Tricia are not used and therefore the user profile in the generic web application's user interface is displayed in a single view without those views. Indicating in the application bar that the user profile is open may seem not helpful at first. However all users have a profile and if the own user profile is open, the indicator appears in the accent color. Uploading an image of themselves gives users control over how to present themselves in such a professional environment (*U3*, *U4*). Showing a set of expertise has been adapted from Darwin and The Organic Data Science Framework, where they proved as a successful factor in encouraging users to fulfill their tasks.

4.7. Activity Feed



Figure 4.23.: The Generic Web Application: The Activity Feed Indicator

In a system with so much interlinked content, it is impossible to keep track of all the data by simply checking the pages, so the activity feed offers users to stay up-to-date with the content they are interested in. As stated in section 4.4.2, users can subscribe to content by pressing the eye-icon (*F1*). They get notified on changes by a red indicator on the activity feed icon in the application bar (see Figure 4.23) and a click on this icon leads to the activity feed. There users find the option "Show watched activities only" at the bottom, which makes the feed show "watched" information only. Figure 4.24 shows the actual feed as a wiki page with chronologically ordered activities in reverse order and in the sidebar there is filter functionality. A single piece of information, shown in the feed, is called an activity. In general, there is a distinction between three types of activities that appear on the feed. Discussions by users are displayed

¹⁵Tricia's web interface from <https://wwwmatthes.in.tum.de/persons/70nqokxge94e/Florian-Katenbrink> (accessed August 2nd, 2015)

4. Design of a Generic Web Application's User Interface

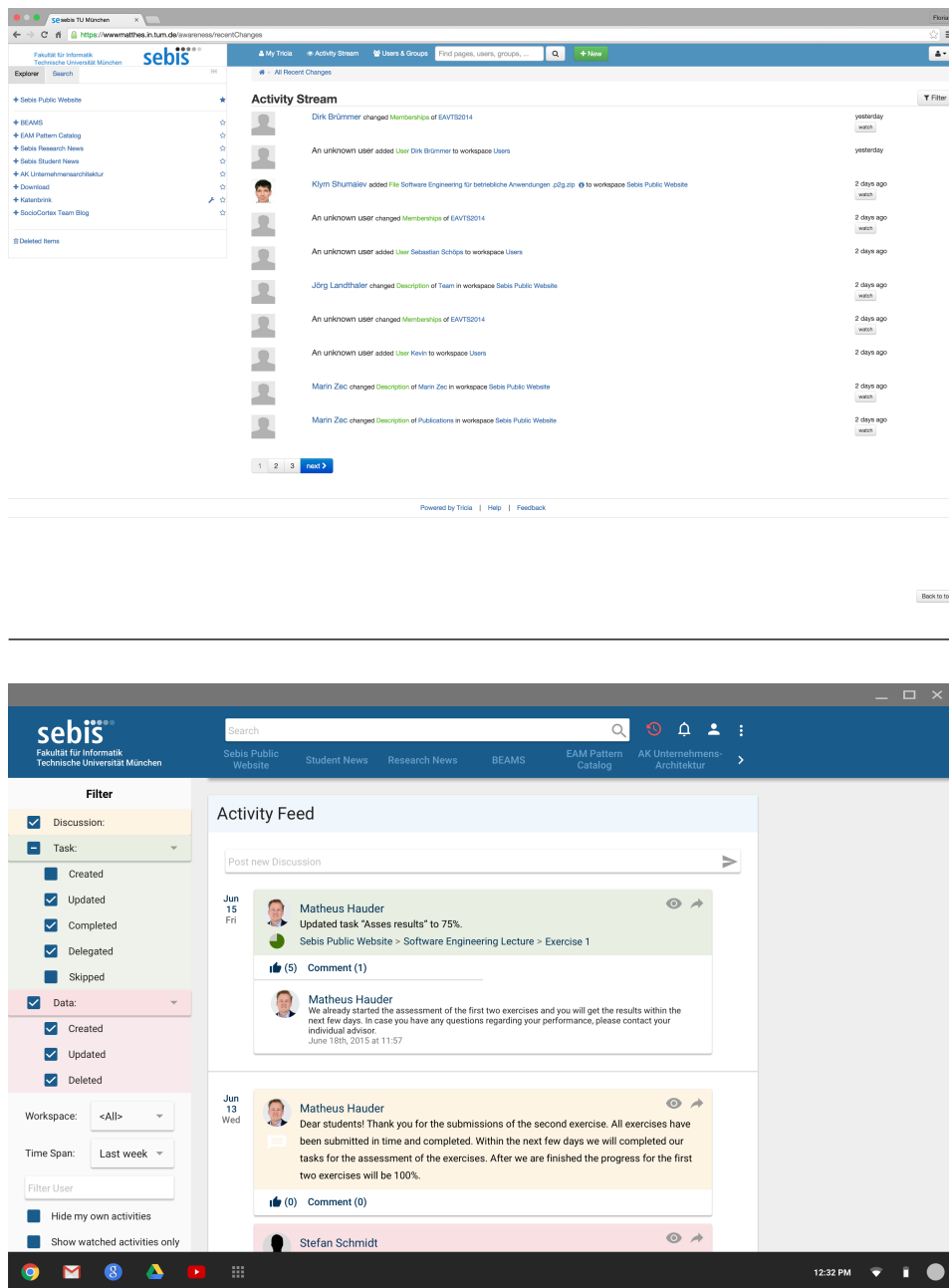


Figure 4.24.: Tricia vs. the Generic Web Application: The Activity Feed¹⁶

in green, changes in task's metadata in yellow and changes in both wiki pages and attributes in red.

Users can start a new discussion by using the "Post new Discussion" text field in the center and pressing enter or the caret on the right. Discussions are mostly publicly available and therefore any user can respond to a discussion

¹⁶Tricia's web interface from <https://wwwmatthes.in.tum.de/awareness/recentChanges> (accessed August 1st, 2015)

by using the "Comment" button on an activity. Apart from that, users can "like" a discussion by pressing the thumbs-up-icon. At the top right of every activity there is a button to watch or share it. The share feature sends an email to a user relating this activity.

Discussions are important in order to enhance collaboration and communication. In Tricia they do not exist because there is only one activity on the feed: data. The level of collaboration as in the generic web application is lower in Tricia, therefore communication often has to be performed in ways that are not part of the application. A like-button is a feature many feeds on the web have, so users are comfortable with this way of communication (*G4*). Sharing content is another way of enhancing collaboration (*F2*). Tricia shows the activities in pagination, whereas the generic web application's user interface supports infinite scrolling. This makes it more comfortable, when looking at many changes.

The sidebar offers users to turn on/off discussions (yellow), tasks (green) and data (red) activities. The separation by colors helps users associate them with their filter options (*G6*). They are clearly separated by colors. The task and data activities can be filtered further. Tasks can be filtered according to five different categories of metadata changes: Created, Updated, Completed, Delegated and Skipped. Data can be filtered by Created, Updated or Deleted. In both cases there can be multiple fields selected or none. If task or data is deselected then the respective colored area below folds up. General filter functionality at the bottom is to show activities within a certain workspace, show activities within a time span or activities that are issued by a certain user. Users also have the option to hide their own activities so they can focus on other activities. In the generic web application, the filter sidebar is right at the hand of users. This presence of filtering functionality encourages users to make use of it. However it is more important than in Tricia, because there are two new types of activities, which add again more information to the feed. In Tricia users have to press the filter icon and then the filter options expand.

For data activities there are two ways of presentation. Figure 4.25 shows two different data activities. When there is a change in wiki pages, users have the option to redirect to a Diff¹⁷ of changes, but the activity also shows three lines

¹⁷A Diff tool shows differences between the old and the new version of data

4. Design of a Generic Web Application's User Interface

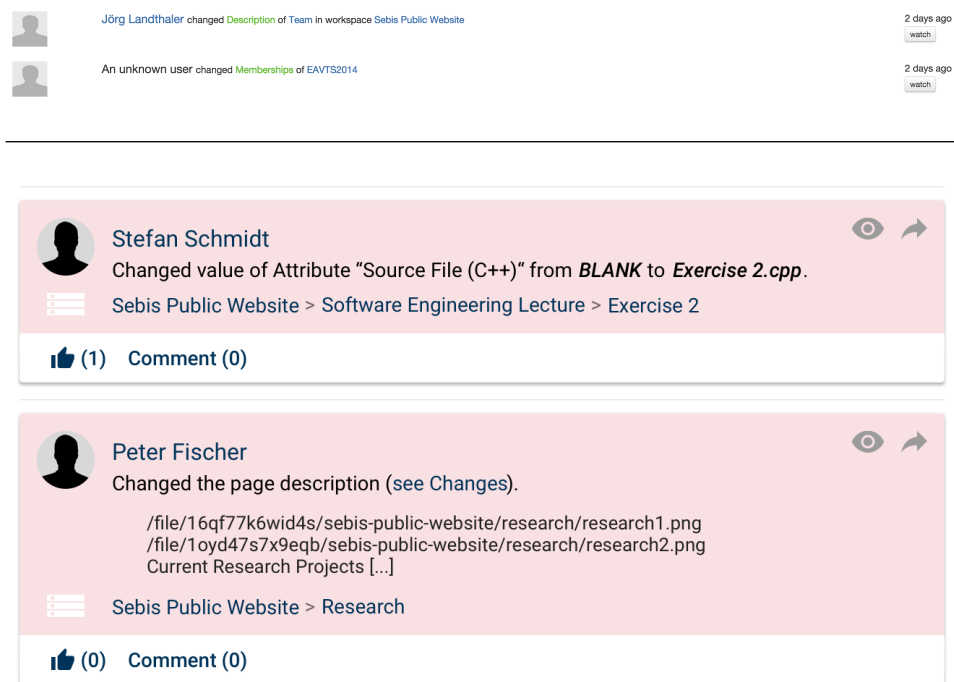


Figure 4.25.: Tricia vs. the Generic Web Application: Data Activities¹⁸

of the Diff. If there is a change in attributes, the activity displays the changes immediately.

As data activities are the only activity shown in Tricia it can be well compared to the generic web application. In Tricia the feed only tells users when, by whom and where the changes were made. It does not show the value that is changed. This causes users to make a click for every single activity, when they want to see the value. In the generic web app users have changed values and sometimes even the old value right at their sight. This reduces the amount of jumps ($N1$).

4.8. Search

Studies have shown that experienced users often use search functionality [Bu05], therefore it is important to offer a powerful interface to support users in this way of information seeking as well. Therefore there is always a search bar in the Application Bar of the generic web app. When users type, auto-completion

¹⁸Tricia's web interface from <https://wwwmatthes.in.tum.de/awareness/recentChanges> (accessed August 1st, 2015)

4. Design of a Generic Web Application's User Interface

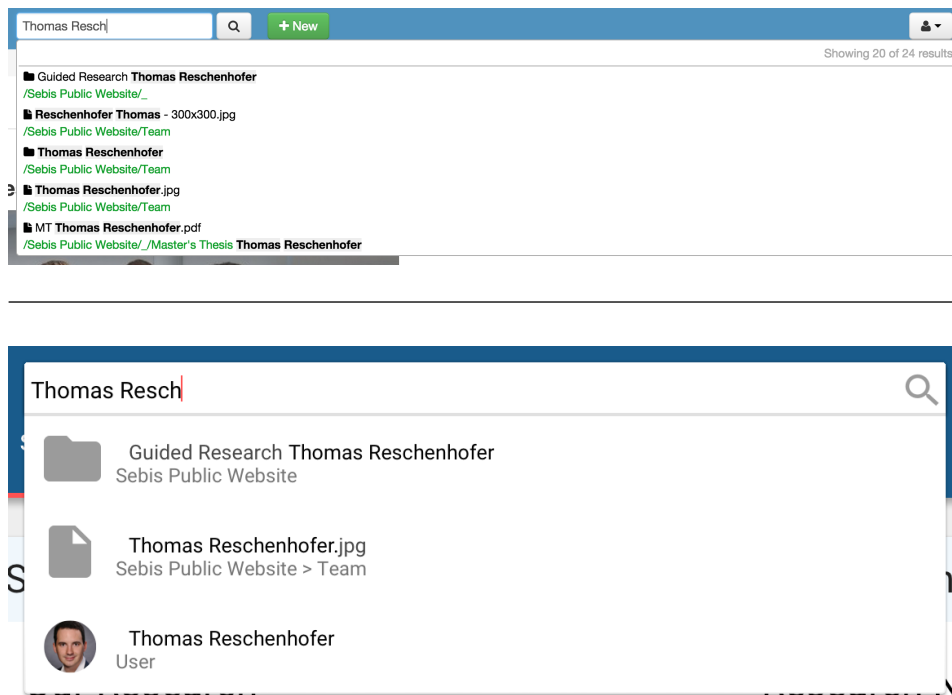


Figure 4.26.: Tricia vs. the Generic Web Application: Using the Search Bar¹⁹

suggests possible hits by displaying a type icon, the name and path to the object (see Figure 4.26).

The search bar in the generic web application is wider than in Tricia, because longer search terms cannot be seen in Tricia anymore. As user profiles become more important in the generic web app, the user's profile picture is shown in the search results (*U1*).

Users can redirect to the result directly by selecting it in the list or show more search results by clicking the magnifying glass. Figure 4.27 displays the complete list of search results in a page and the sidebar that offers filter functionality. They can be sorted according to relevance, last modification date and name. In case users only want to see, for example, tasks, they can filter the content type. Another powerful filter possibility is to specify the overlying workspace of this content. The last three filter features are for more advanced users: filtering of the page type, attributes like "created at", "parent page", etc. and special filtering, which shows results for pages that contain invalid links or values.

¹⁹Tricia's web interface from <https://wwwmatthes.in.tum.de/pages/t5ma0jrv6q7k/sebis-Public-Website-Home> (accessed August 2nd, 2015)

4. Design of a Generic Web Application's User Interface

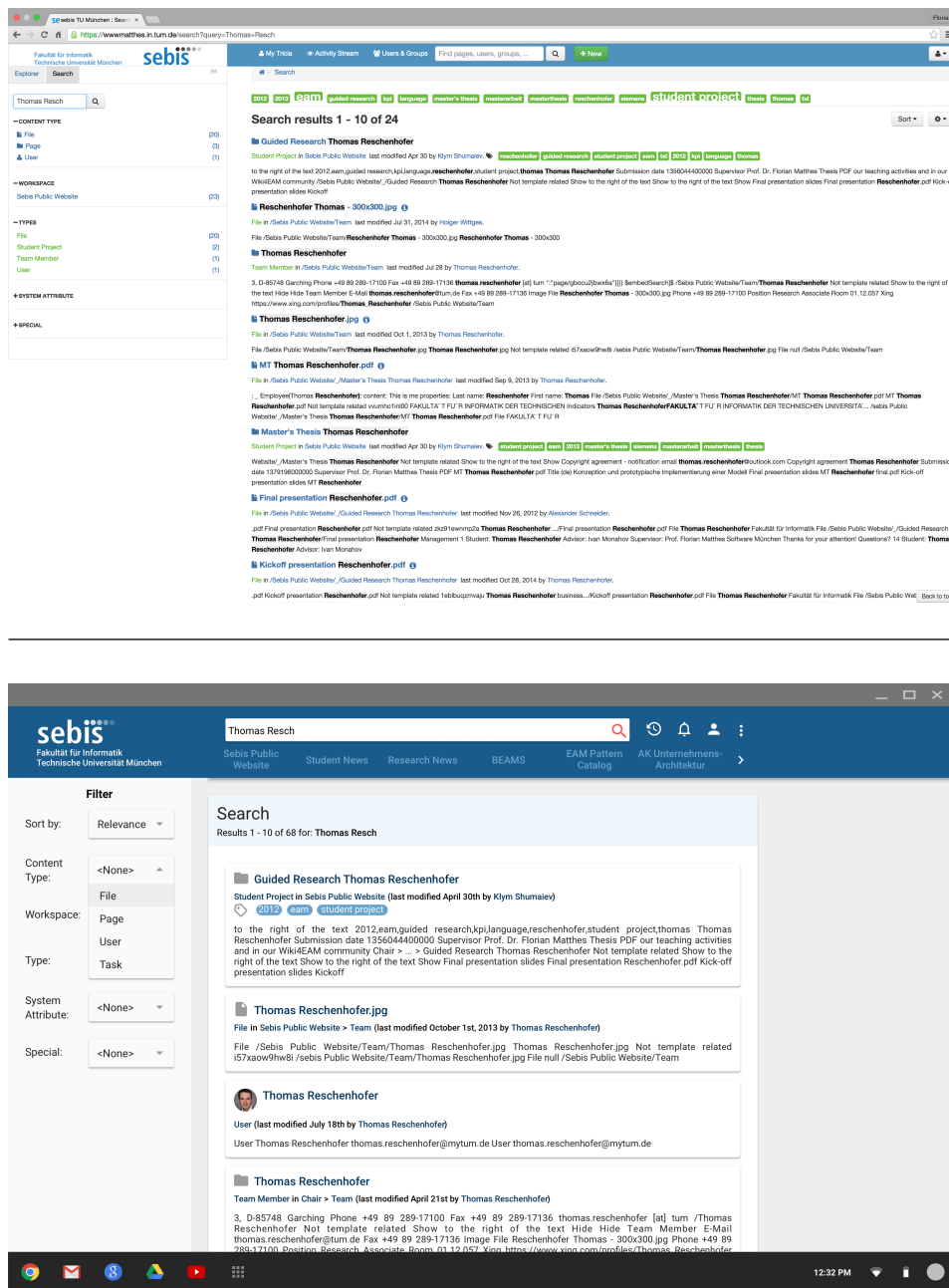


Figure 4.27.: Tricia vs. the Generic Web Application: Search Results²⁰

In comparison to Tricia, the search results for the generic web application's user interface are displayed in a clear overall structure. In Tricia it is hard to distinguish where a search result starts and where it ends. Because of cards the generic web app has a clear area for each result (*G8*) and prominent result titles (*G1*). It is confusing in Tricia, which search bar users should use,

²⁰Tricia's web interface from <https://wwwmatthes.in.tum.de/search?query=Thomas+Resch> (accessed August 2nd, 2015)

4. Design of a Generic Web Application's User Interface

which is unified into a single search bar in the generic web application's user interface.

In the wiki page's header there is displayed how many search results there are and for what search term. In the page's body there is a card for every result, arranged in vertical order. There are four types of content that can be searched for and essentially displayed: files, pages, users and tasks, whereas all three results are displayed in the same overall structure. Figure 4.28 shows the single result for a file. The small icon on the top-left indicates the type, a file, as well as the text below. To the right there is the file name, which opens the file on click. Below this is the path to the page where the file is attached, the data of last modification and by whom it was modified last. Then follows the description of the context in which the file was found according to the search term.

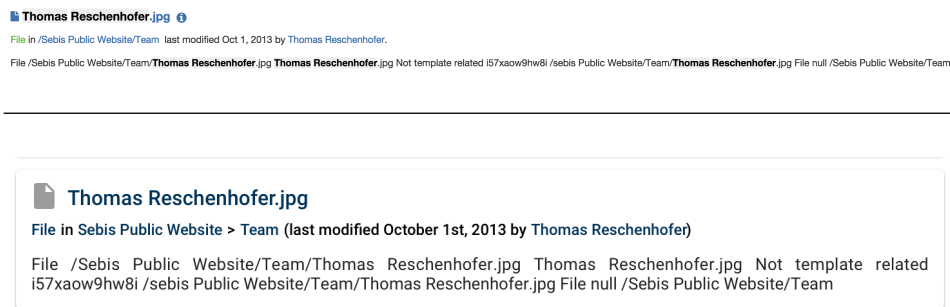


Figure 4.28.: Tricia vs. the Generic Web Application: The File Search Result²¹

The individual search results in the generic web application's user interface have a clearer structure than results in Tricia. The title and information, such as the path and last modification date, has fewer spacing in between as to the results content. This lets users scan results even more quickly (*G1*, *W1*).

The other three activity results are structured in the same pattern. A page result has a page icon, the page type and the page title in the heading. The title leads to the page whereas the path that is shown below leads to the parent page. As tags can be attached to pages, they are also shown in the heading, but not as prominent as in Tricia, because they are not majorly important (*G5*). A click on such a tag filters all the SocioCortex content according to the tag. Below this is again the description of the page. For the third type, a user profile

²¹Tricia's web interface from <https://wwwmatthes.in.tum.de/search?query=Thomas+Resch> (accessed August 2nd, 2015)

result, there is a thumbnail in the top-left corner of the profile picture instead of an icon. The title (user name) leads to the user profile. Lastly Figure 4.29 displays a task result. The heading leads to the task details page. As content this result shows some aggregated information: in how many pages this task is used and how many of these are completed, in progress, overdue or not started yet. The task is used in many pages, therefore aggregated information is most useful for users. For detailed information they need to go to the task detail page.

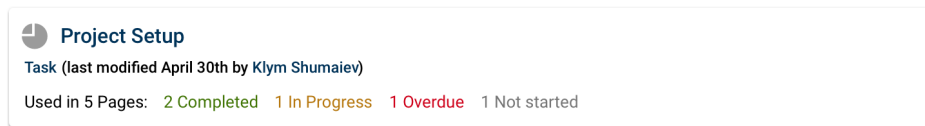


Figure 4.29.: The Generic Web Application: The Task Search Result

4.9. Task Details

In order to show task results of a search term in a more detailed manner, users can, as mentioned in the previous section, click on the result's heading to get to the task details (see Figure 4.30). A task definition is always stored in a workspace and the sidebar displays the task's related workspace's subpage hierarchy. The page is titled with the task name. Users find a list of all pages below the task's title. Each list entry starts with the task's specific progress icon, followed by the page name title and a caret to the right. When users press the page name they will be redirected to the page that contains this specific task, whereas when they click on the caret the list entry expands. The white expanded area then displays all attributes in the left column and the metadata in the right column of the area. This gives users an overview, as well as details, over specific tasks. They can for example examine tasks that are often overdue and perhaps too difficult or tasks, which users do not like to carry out.

4. Design of a Generic Web Application's User Interface

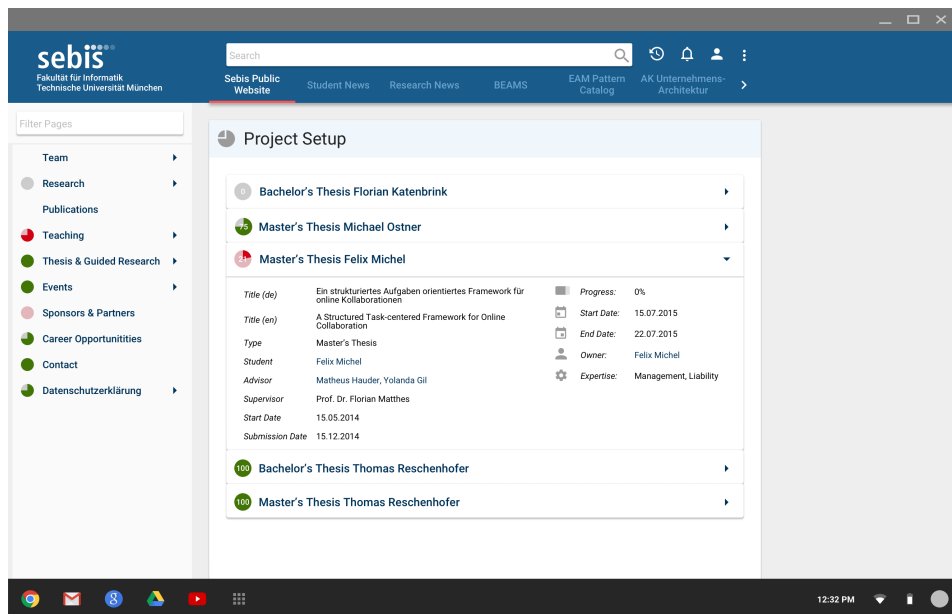


Figure 4.30.: The Generic Web Application: Task Details

4.10. Alerts

Alerts are features that help users in completing their tasks. Whenever users have one or more overdue tasks, a red indicator is shown in the application bar above the red alerts icon. Figure 4.31 shows the indicator with seven undue tasks.



Figure 4.31.: The Generic Web Application: The Alert Indicator



Figure 4.32.: The Generic Web Application: Alerts

On click a menu opens with a list of undue tasks (see Figure 4.32). Each list item shows the task's progress, title, when it was due and the attached page's

4. Design of a Generic Web Application's User Interface

path. When users press an item, they are redirected to the page showing the task, whereas the task section is opened immediately. If users press the "Show All Tasks" button below, they will be redirected to their own user profile page.

5. Conclusion & Outlook

5.1. Summary

The aim of this thesis was to optimize the user experience for a social content management software. Thereby focus was on optimizing the navigation, wiki content, user profile, activity feed and search.

The foundation was laid by Tricia, the existing hybrid-wiki and the new decoupled approach SocioCortex. These systems evolved out of the realization that modern enterprises need a way to deal with the mass of diversely growing data. Both Darwin and the Organic Data Science Framework introduced strong concepts that are not existent in Tricia, but are now applied in the generic web application's design. The interface is built on clear design principles that assure robustness and it is based on the Material Design visual language.

The applied research method for design-science in information systems ensured a framework for designing and evaluating in periods of two weeks. An analysis of the research is conducted in the conclusion by a defining how each of the guidelines the framework specifies was successfully conducted.

The design of the generic web application was demonstrated. A color palette and typography definitions are given. The design covers four core parts of Tricia: navigation, wiki content, user profile and search results. An activity feed as well as tasks, alerts and a page for task details are furthermore integrated in the design, which have its origin in the Darwin and the Organic Data Science Framework. By showing how Tricia implemented the design and directly comparing this to the generic web application's user interface improvements in the user experience were demonstrated.

5.2. Conclusion

A framework for design-science research was introduced in chapter 3. This framework provides, as mentioned, guidelines in order to evaluate the scientific relevance of a research. Figure 5.1 demonstrates how the seven guidelines were successfully applied.

Problem Relevance: The problem relevance is given through the introduction. Enterprises gain more and more diverse information. Therefore Tricia, a hybrid-wiki, was developed, which has many constraints and does not provide a good user experience. Enterprises and further researchers find valuable applied principles to improve a social content management software in this research.

Research Rigor: The generic web application's design is based on three existing wiki systems. They are scientifically sound, whereas the design of the generic web application's user interface is optimized using design principles from profound literature. It is accomplished with a design language established within the last year.

Design as a Search Process: The thesis demonstrates improvements in user experience by comparing Tricia with the new generic web application's user interface. The artifact was developed in many iterations of refinement and assessment, which improved the user experience step by step.

Design as an Artifact: The artifact is the stack of designs of the generic web application. A complete list of the windowed designs is given in the Appendix (see A.3).

Design Evaluation: The assessment-refinement cycle was executed in two-weekly focus group meetings. An example of the results for the Navigation by this research cycle is given in the Appendix (see A.1).

Research Contributions: The design-science contributions of this research are the demonstrated artifact. Moreover the demonstration of how to improve the user experience by applying the guidelines and following a research framework.

Communication of Research: This research is aimed at the users and developers of a social content management software. Nonetheless this thesis shows managers also how improvements can be made. It implies that the managers have a good understanding of their own system.

Figure 5.1.: An Evaluation of the 7 Research-Framework Guidelines

The navigation has been greatly improved in comparison to Tricia's navigation. As described in section 4.4, navigation is now split up in two parts: the application bar and the sidebar. This creates a logical segregation of workspaces and subpages, which helps users to understand the underlying model intuitively. Additionally it takes into consideration that users in Tricia usually stay within one workspace, when visiting the web application. It has a helpful side effect of removing one hierarchy layer from the sidebar to the top bar. The sidebar navigation is therefore not as overloaded anymore and the application bar combines primary elements. However more experienced users can still navigate quickly through the tree view and examine the task progress of subpages in the sidebar.

On the first impression, the wiki content might not have changed as much. This is because users still have the freedom of designing unstructured information in their own way. These slight changes have a large impact on the user experience though. By hiding the different views of a page beneath a View-menu and moving both the watch and share functionality up to the application bar, users are confronted with less options at once. This makes it easier for them to focus on the content. Pages now save vertical space by differentiating structured and unstructured information in a clear way and therefore making unstructured content float around the structured. The included elements to view and edit tasks are installed right in place and therefore avoid unnecessary jumps in navigation. Tasks and attributes offer as a combined element a way that increases collaboration and completeness of pages as tasks can be used to complete both structured and unstructured data.

By reducing the user profile attributes down to name, email and last login it is now 100% filled in for all users as this information is gathered on registration. In Tricia, hardly any user filled in his or her user profile. The use of the profile picture in different areas of the application, such as the activity feed or the search encourages users to upload an image. This information is sufficient for successful collaboration, as details about tasks are more important. Because the profile now informs about task achievements and expertise that can be gained, this is the main content. Users can quickly locate anyone's current, future and completed tasks as well as their expertise through the profile.

As a new feature, the activity feed combines Tricia's activity stream with tasks and the collaborative discussion element. Tasks and collaboration be-

come more and more important, therefore their integration into the feed enable users to even keep track of changes in larger systems. activities in the generic web application's activity feed are clearly separated, therefore making scanning the activities easier. Simple, though powerful filtering of content is encouraged through the user-interface, which helps users to focus on their individual interests in data.

Search in the generic web application's user interface becomes more intuitive. Small changes, such as unifying two search bars into one, showing profile picture thumbnails in the results or making the search bar wider improve the users user experience. Tags are not as prominent as in Tricia, because they are not main elements. Users mainly scan search results for certain keywords. By changing separation within a search result item and among items, scanning of search results is easier.

5.3. Outlook

The design of the generic web application's user interface in this thesis strongly focuses on the representation for users who view the content. Another important aspect is the representation of how users can edit the Wiki Content, create new subpages and workspaces, edit the tree structure of a workspace, etc. These features are part of a web application for the casual user, therefore further research needs to go more into detail in this field.

As the design is optimized for a web application for the casual user, it is moreover interesting to research how for example a mobile client could look or an interface for the power users of the generic web application, who needs to work a lot with the system's metadata.

Another aspect that is not covered in this thesis is the implementation of the generic web application. It is advisable to stick to a framework that highly supports the Material Design language and dynamic front-end functionality, such as Angular Material²².

The design principle G2 in Figure 2.4 suggests to use simple and meaningful names and G4 suggests to stick to conventions. Therefore an analysis on how

²²<https://material.angularjs.org> (accessed August 2nd, 2015)

5. Conclusion & Outlook

the Social Content Management Software's content structure, labels, names, etc. influences the user experience can be an interesting research.

The representation of tasks in this thesis considers neither inconsistencies nor metadata influences. These two factors might give users a valuable notice about the task. However before applying them in a design it is interesting to find out beforehand, whether they are valuable for users or not.

There are users who are either not interested or included in processing tasks. For these users there needs to be an option show hide tasks entirely or only partially (for example the sidebar only). Providing a mechanism for is is not within the scope of this research though.

An individualization of the interface is not covered in the designs. Because users have different taste in color and fonts, the individualization of these two by users can improve their personal user experience.

Bibliography

- [Ba05] Barabasi, A. L.: *SOCIOLOGY: Network Theory-the Emergence of the Creative Enterprise*. *Science*. 308(5722):639–641. Apr 2005. 2.1
- [Be15] Bedford, A.: *Icon Usability*. Aug 2015 (accessed August 1st, 2015). 4.5.3
- [Bu05] Buchanan, G.; Cunningham, S. J.; Blandford, A.; Rimmer, J.; Warwick, C.: *Information seeking by humanities scholars*. In (Rauber, A.; Christodoulakis, S.; Tjoa, A. M., Ed.): *Research and Advanced Technology for Digital Libraries*. volume 3652 of *Lecture Notes in Computer Science*. pages 218–229. Berlin. 2005. ECDL. Springer. 4.8
- [CM09] Crumlish, C.; Malone, E.: *Designing Social Interfaces*. O'Reilly. Beijing. 2009. 978-0-596-15492-9. 2.4.1
- [co15] contributors, C. O.: *The Chromium Projects*. Jul 2015 (accessed July 19th, 2015). 5
- [Di13] Dias, A. L.; Fortes, R. P. d. M.; Masiero, P. C.; Watanabe, W. M.; Ramos, M. E.: *An Approach to Improve the Accessibility and Usability of Existing Web System*. In *Proceedings of the 31st ACM International Conference on Design of Communication*. SIGDOC '13. pages 39–48. New York, NY, USA. 2013. ACM. 1.2
- [EM00] Edmunds, A.; Morris, A.: *The Problem of Information Overload in Business Organisations: A Review of the Literature*. *Int. J. Inf. Manag.* 20(1):17–28. Feb 2000. 1.1
- [Go15a] Google: *Material Design - Introduction (May 2015 release)*. May 2015 (accessed July 17th, 2015). 2.4.2
- [Go15b] Google: *Material Design - Layout > Structure*. Jul 2015 (accessed July 19th, 2015). 4.4.2

- [He04] Hevner, A. R.; March, S. T.; Park, J.; Ram, S.: *Design Science in Information Systems Research*. *MIS Q.* 28(1):75–105. Mar 2004. 3, 3.1, 3, 5.3
- [HKM15] Hauder, M.; Kazman, R.; Matthes, F.: *Empowering End-Users to Collaboratively Structure Processes for Knowledge Work*. In *Business Information Systems - 18th International Conference, BIS 2015, Pozna, Poland, June 24-26, 2015, Proceedings*. pages 207–219. 2015. 2.2
- [Hu13] Hu, C.: *The Nature of Software Design and Its Teaching: An Exposition*. *ACM Inroads*. 4(2):62–72. Jun 2013. 1.2
- [ID15] IDC: *Smartphone OS Market Share, Q1 2015*. May 2015 (accessed July 15th, 2015). 2.4.2
- [Ka07] Kalbach, J.: *Designing web navigation - optimizing the user experience*. O’Reilly. 2007. 978-0-596-52810-2. 2.4.1
- [Kr14] Krug, S.: *Don’T Make Me Think, Revisited: A Common Sense Approach to Web Usability*. New Riders Publishing. Thousand Oaks, CA, USA. 3rd edition. 2014. 0321965515, 9780321965516. 2.4.1, 4.3
- [LH11] Lunn, D.; Harper, S.: *Improving the Accessibility of Dynamic Web Content for Older Users*. In *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility*. W4A ’11. pages 16:1–16:2. New York, NY, USA. 2011. ACM. 1.2
- [Li06] Lindgaard, G.; Fernandes, G.; Dudek, C.; Brown, J. M.: *Attention web designers: You have 50 milliseconds to make a good first impression!* *Behaviour & IT*. 25(2):115–126. 2006. 4.3
- [LM05] Lehel, V.; Matthes, F.: *User-Centered Social Software - Der Wissensarbeitsplatz der Zukunft?* In *Konferenz zum Einsatz von Knowledge Management in Wirtschaft und Verwaltung (KnowTech)*. München. 2005. BitKom. 1.1
- [MC10] Morville, P.; Callender, J.: *Search Patterns - Design for Discovery*. O’Reilly. 2010. 978-0-596-80227-1. 2.4.1
- [Mi15a] Michel, F.: *A structured task-centered framework for online collaboration*. Jul 2015 (accessed July 22nd, 2015). 2.3

- [Mi15b] Michel, F.; Gil, Y.; Ratnakar, V.; Hauder, M.: *A Task-Centered Interface for On-Line Collaboration in Science*. In *Proceedings of the 20th International Conference on Intelligent User Interfaces Companion*. IUI Companion '15. pages 45–48. New York, NY, USA. 2015. ACM. 2.3
- [MNS11] Matthes, F.; Neubert, C.; Steinhoff, A.: *Hybrid Wikis: Empowering Users to Collaboratively Structure Information*. In *ICSOFT (1)'11*. pages 250–259. 2011. 1.1, 2.1
- [Sc14] Schneider, A.: *sebis Research Profile*. Nov 2014. 1.2, 2.1
- [SN09] Scott, B.; Neil, T.: *Designing Web Interfaces - Principles and Patterns for Rich Interactions*. O'Reilly. 2009. 978-0-596-51625-3. 2.4.1

A. Appendix

A.1. Evolution of the Navigation



Figure A.1.: Evolution of the Navigation: Part I

A. Appendix

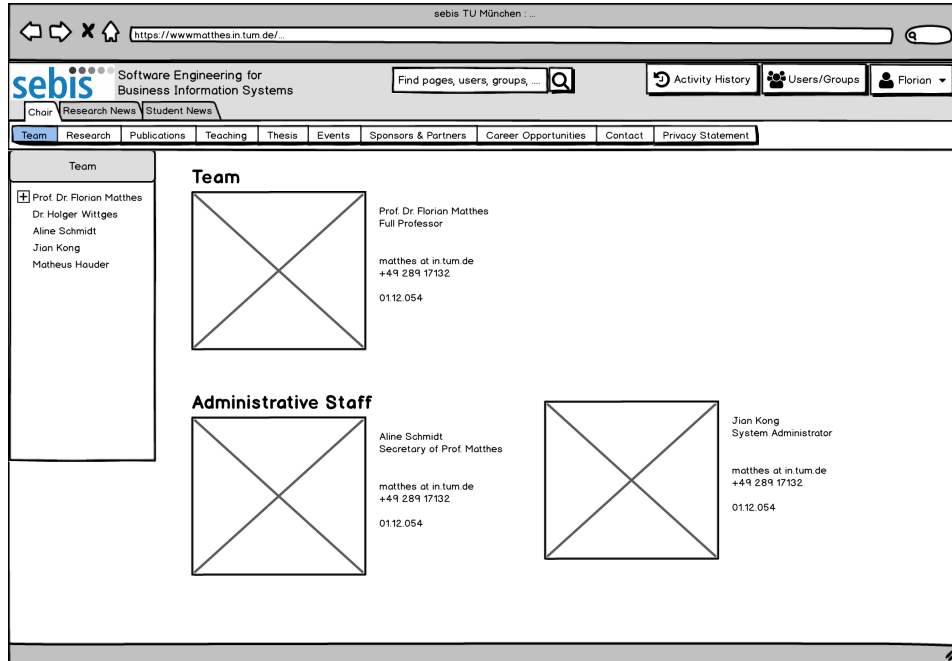


Figure A.2.: Evolution of the Navigation: Part II



Figure A.3.: Evolution of the Navigation: Part III

A. Appendix

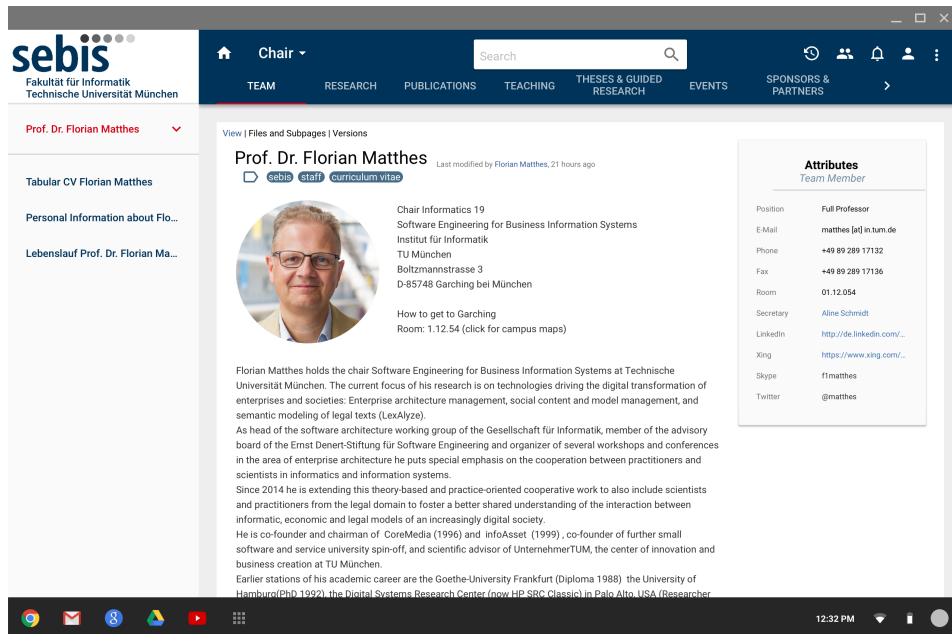


Figure A.4.: Evolution of the Navigation: Part IV

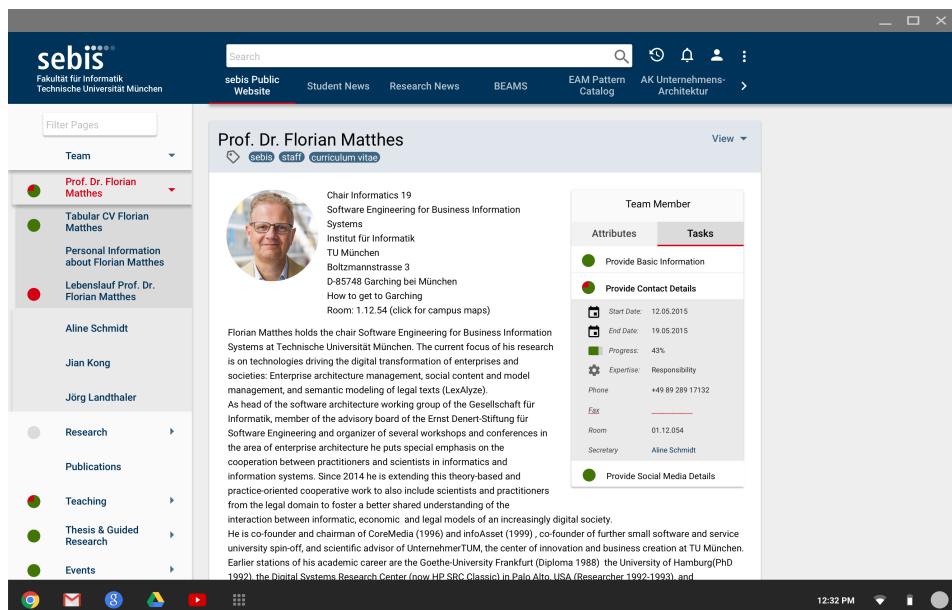


Figure A.5.: Evolution of the Navigation: Part V

A. Appendix

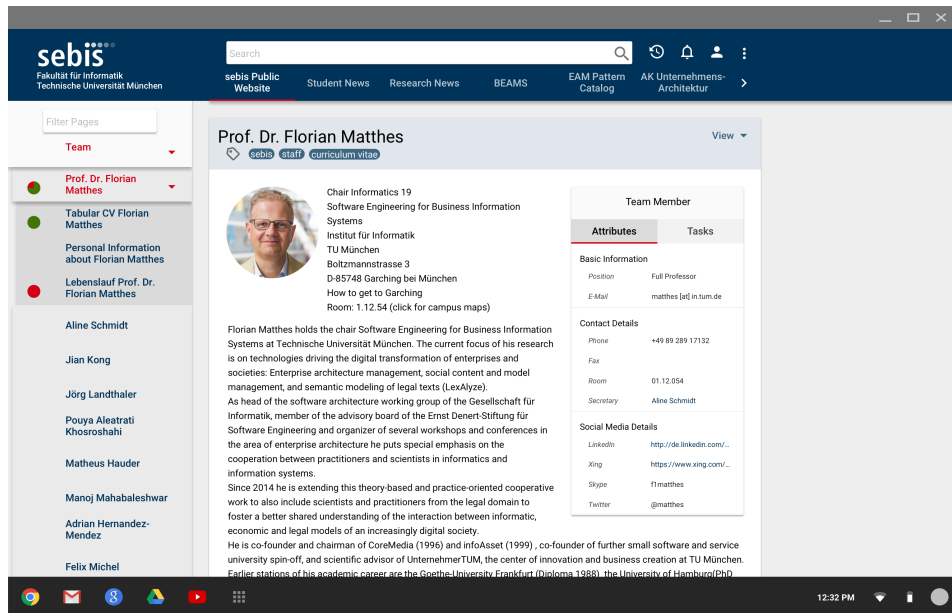


Figure A.6.: Evolution of the Navigation: Part VI

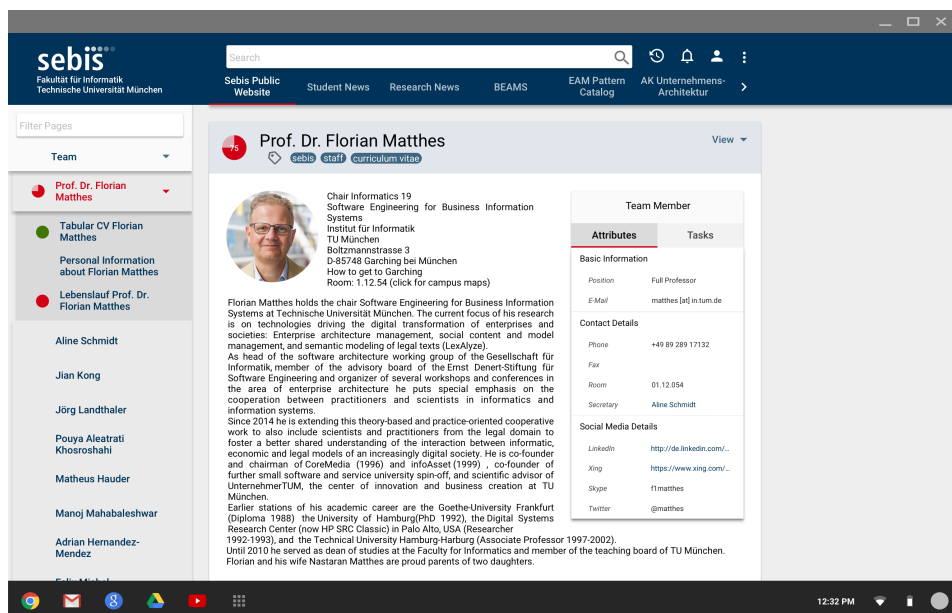


Figure A.7.: Evolution of the Navigation: Part VII

A. Appendix

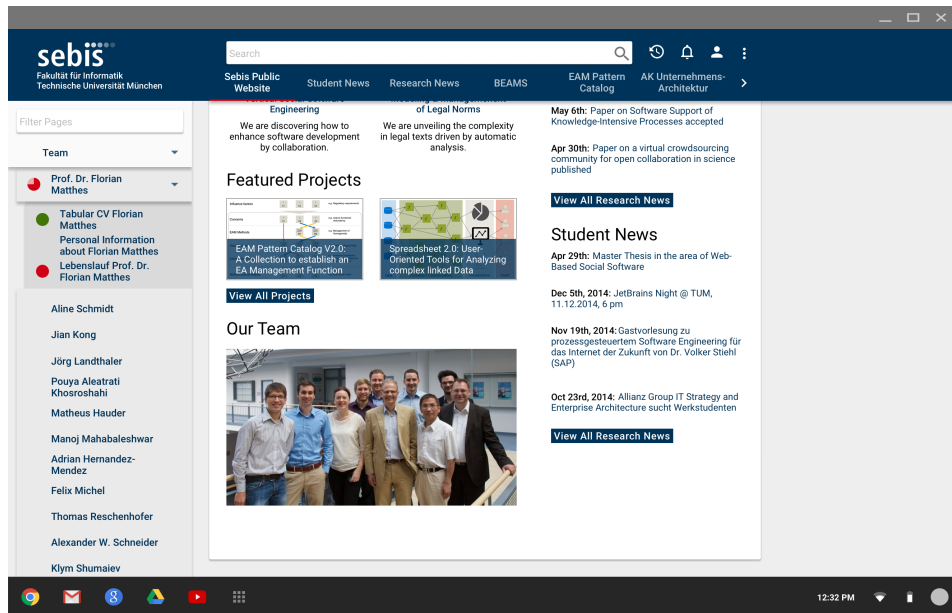


Figure A.8.: Evolution of the Navigation: Part VIII

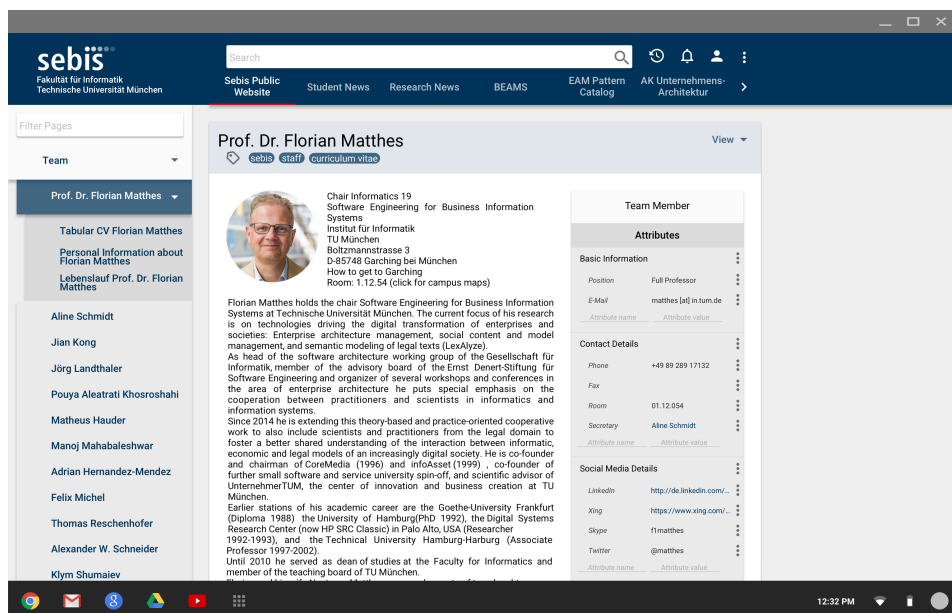


Figure A.9.: Evolution of the Navigation: Part IX

A. Appendix

The screenshot displays the website for the Chair of Informatics 19 at TU München. The header includes the 'sebis' logo and navigation links for 'Sebis Public Website', 'Student News', 'Research News', 'BEAMS', 'EAM Pattern Catalog', and 'AK Unternehmens-Architektur'. A search bar is located in the top right. The main content area features a profile for Prof. Dr. Florian Matthes, including a photo, contact information, and a detailed biography. A sidebar on the left lists other team members, and a right sidebar shows a list of attributes for the selected team member.

sebis
Fakultät für Informatik
Technische Universität München

Search

Sebis Public Website | Student News | Research News | BEAMS | EAM Pattern Catalog | AK Unternehmens-Architektur

Filter Pages

Team

- Prof. Dr. Florian Matthes
- Tabular CV Florian Matthes
- Personal Information about Florian Matthes
- Lebenslauf Prof. Dr. Florian Matthes
- Aline Schmidt
- Jian Kong
- Jörg Landthaler
- Pouya Aletrati Khoshroshahi
- Matheus Hauder
- Manoj Mahabaleswar
- Adrian Hernandez-Mendez
- Felix Michel
- Thomas Reschenhofer
- Alexander W. Schneider
- Klym Shumaiev
- Alexander Waldmann

Prof. Dr. Florian Matthes

Chair Informatics 19
Software Engineering for Business Information Systems
Institut für Informatik
TU München
Boltzmannstrasse 3
D-85748 Garching bei München
How to get to Garching
Room: 1.12.54 (click for campus maps)

Florian Matthes holds the chair Software Engineering for Business Information Systems at Technische Universität München. The current focus of his research is on technologies driving the digital transformation of enterprises and societies: Enterprise architecture management, social content and model management, and semantic modeling of legal texts (LexAlyze).
As head of the software architecture working group of the Gesellschaft für Informatik, member of the advisory board of the Ernst Denert-Stiftung für Software Engineering and organizer of several workshops and conferences in the area of enterprise architecture he puts special emphasis on the cooperation between practitioners and scientists in informatics and information systems.
Since 2014 he is extending this theory-based and practice-oriented cooperative work to also include scientists and practitioners from the legal domain to foster a better shared understanding of the interaction between informatic, economic and legal models of an increasingly digital society. He is co-founder and chairman of CoreMedia (1996) and infoAsset (1999) - co-founder of further small software and service university spin-off, and scientific advisor of UnternehmerTUM, the center of innovation and business creation at TU München.
Earlier stations of his academic career are the GoetheUniversity Frankfurt (Diploma 1988) the University of Hamburg (PhD 1992), the Digital Systems Research Center (now HP SRC Classic) in Palo Alto, USA (Researcher 1992-1993), and the Technical University Hamburg-Harburg (Associate Professor 1997-2002).
Until 2010 he served as dean of studies at the Faculty for Informatics and member of the teaching board of TU München.
Florian and his wife Nastaran Matthes are proud parents of two daughters.

Team Member

Attributes	
Position	Full Professor
E-Mail	matthes@in.tum.de
Phone	+49 89 289 17132
Fax	+49 89 289 17136
Room	01.12.054
Secretary	Aline Schmidt
LinkedIn	http://de.linkedin.com/...
Xing	https://www.xing.com/...
Skype	fmatthes
Twitter	@fmatthes
Attribute name	Attribute value

Figure A.10.: Evolution of the Navigation: Part X

A.2. Sidebar Scrolling

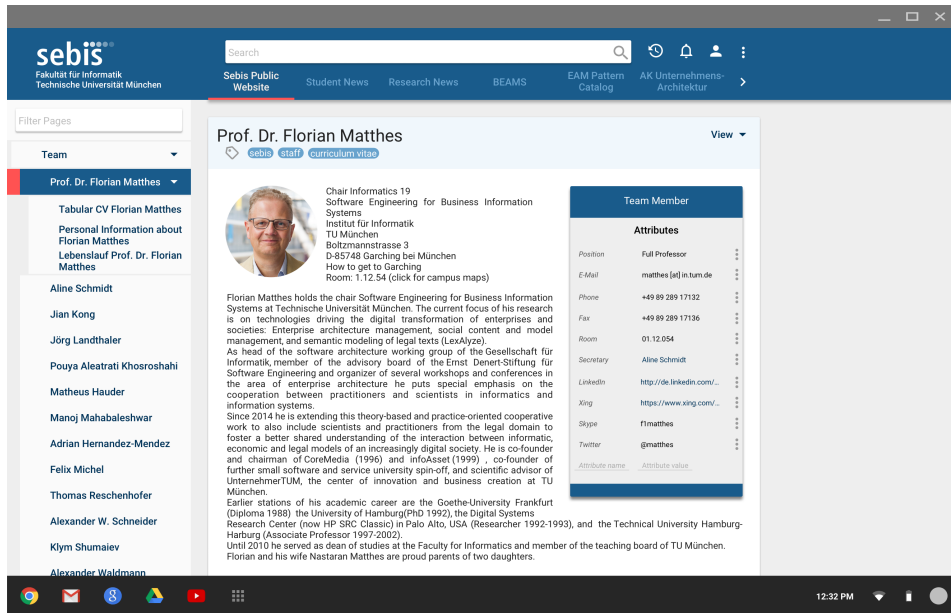


Figure A.11.: Sidebar Scrolling: Part I

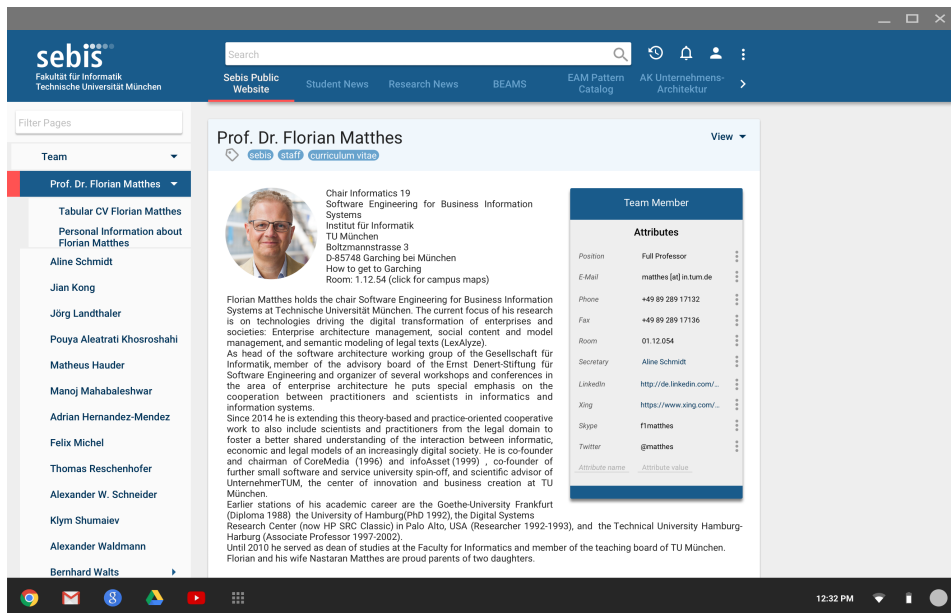


Figure A.12.: Sidebar Scrolling: Part II

A. Appendix

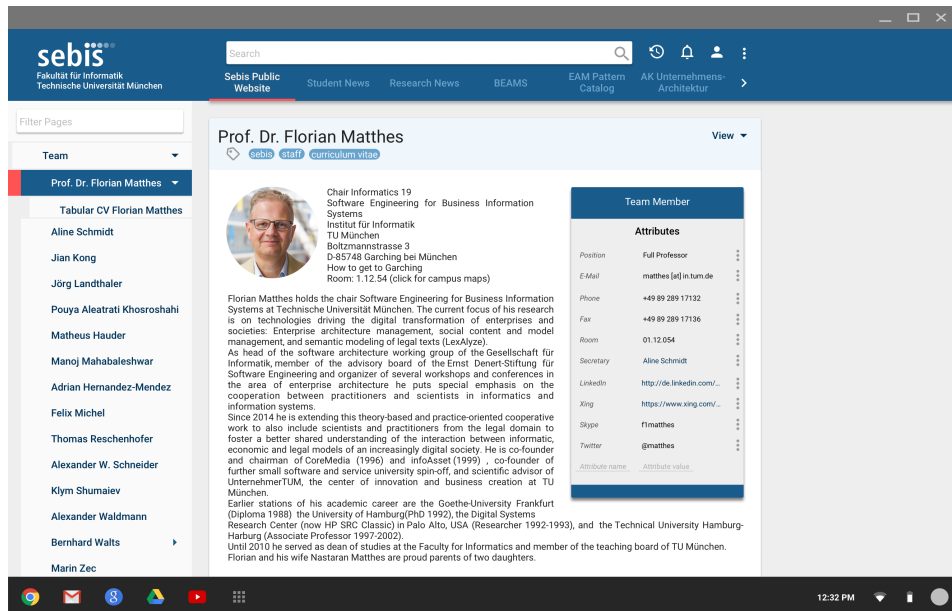


Figure A.13.: Sidebar Scrolling: Part III

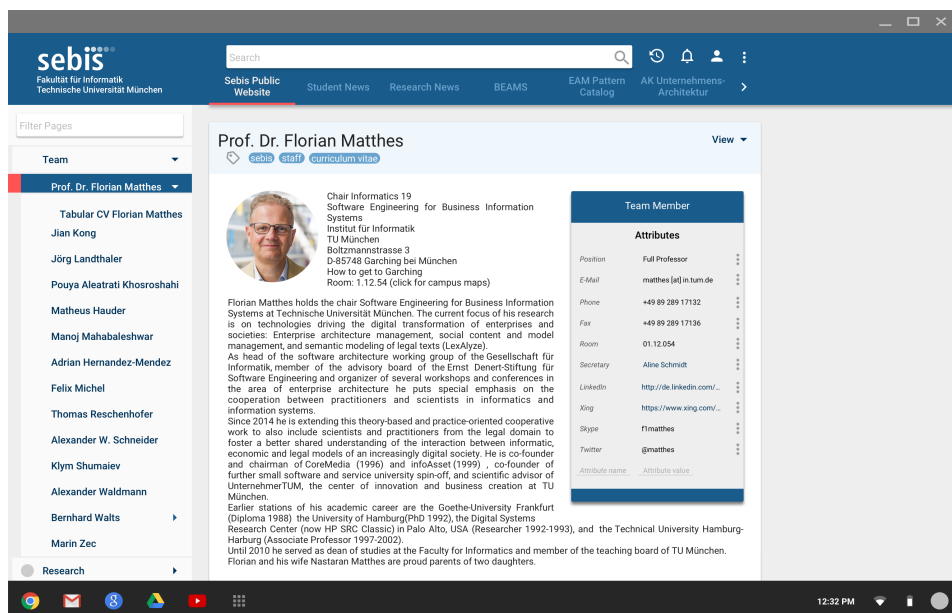


Figure A.14.: Sidebar Scrolling: Part IV

A. Appendix

The screenshot shows the website for Prof. Dr. Florian Matthes at the sebis (Software Engineering for Business Information Systems) group at TU München. The page features a blue header with the sebis logo and navigation links. A sidebar on the left contains a 'Team' section with a scrollable list of names. The main content area displays the profile of Prof. Dr. Florian Matthes, including a photo, contact information, and a detailed biography. A 'Team Member' box on the right lists his attributes such as position, email, phone, and room.

sebis
Fakultät für Informatik
Technische Universität München

Search

Sebis Public Website | Student News | Research News | BEAMS | EAM Pattern Catalog | AK Unternehmens-Architektur

Filter Pages

Team

- Aline Schmidt
- Jian Kong
- Jörg Landthaler
- Pouya Alestrati Khosroshahi
- Matheus Hauder
- Manoj Mahabaleshwar
- Adrian Hernandez-Mendez
- Felix Michel
- Thomas Reschenhofer
- Alexander W. Schneider
- Klym Shurnaiev
- Alexander Waldmann
- Bernhard Waits
- Marin Zec

Publications

Prof. Dr. Florian Matthes

Chair Informatics 19
Software Engineering for Business Information Systems
Institut für Informatik
TU München
Boltzmannstrasse 3
D-85748 Garching bei München
How to get to Garching
Room: 1.12.54 (click for campus maps)

Florian Matthes holds the chair Software Engineering for Business Information Systems at Technische Universität München. The current focus of his research is on technologies driving the digital transformation of enterprises and societies: Enterprise architecture management, social content and model management, and semantic modeling of legal texts (LexAllyze).
As head of the software architecture working group of the Gesellschaft für Informatik, member of the advisory board of the Ernst Denert-Stiftung für Software Engineering and organizer of several workshops and conferences in the area of enterprise architecture he puts special emphasis on the cooperation between practitioners and scientists in informatics and information systems.
Since 2014 he is extending this theory-based and practice-oriented cooperative work to also include scientists and practitioners from the legal domain to foster a better shared understanding of the interaction between informatic, economic and legal models of an increasingly digital society. He is co-founder and chairman of CoreMedia (1996) and infoAsset (1999) - co-founder of further small software and service university spin-off, and scientific advisor of UnternehmerTUM, the center of innovation and business creation at TU München.
Earlier stations of his academic career are the GoetheUniversity Frankfurt (Diploma 1988) the University of Hamburg (PhD 1992), the Digital Systems Research Center (now HP SRC Classic) in Palo Alto, USA (Researcher 1992-1993), and the Technical University Hamburg-Harburg (Associate Professor 1997-2002).
Until 2010 he served as dean of studies at the Faculty for Informatics and member of the teaching board of TU München.
Florian and his wife Nastaran Matthes are proud parents of two daughters.

Team Member

Attributes	
Position	Full Professor
E-Mail	matthes@in.tum.de
Phone	+49 89 289 17132
Fax	+49 89 289 17136
Room	01.12.054
Secretary	Aline Schmidt
LinkedIn	http://de.linkedin.com/...
Xing	https://www.xing.com/...
Skype	fmatthes
Twitter	@fmatthes
Attribute name	Attribute value

Figure A.15.: Sidebar Scrolling: Part V

A.3. Windowed Designs

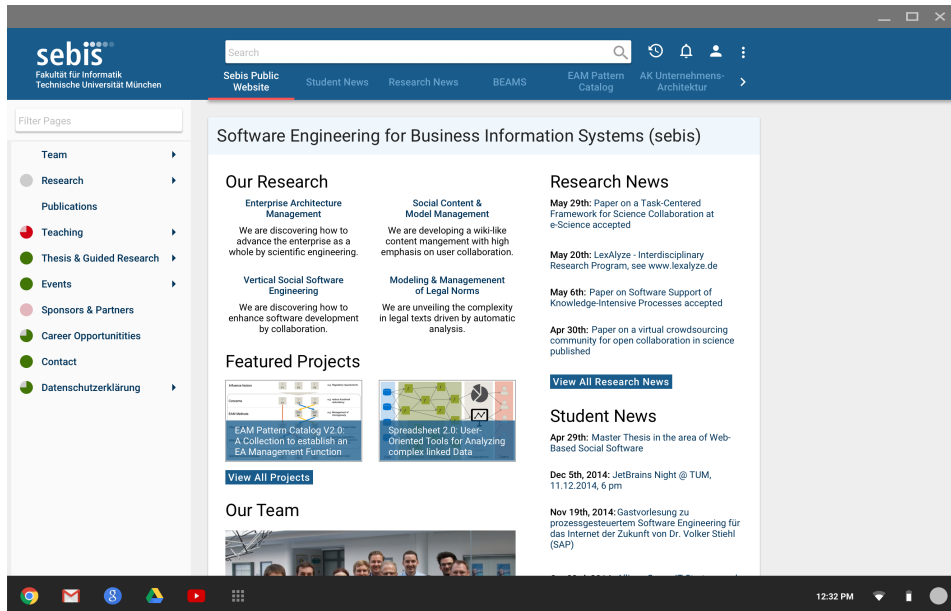


Figure A.16.: Windowed Design: Homepage

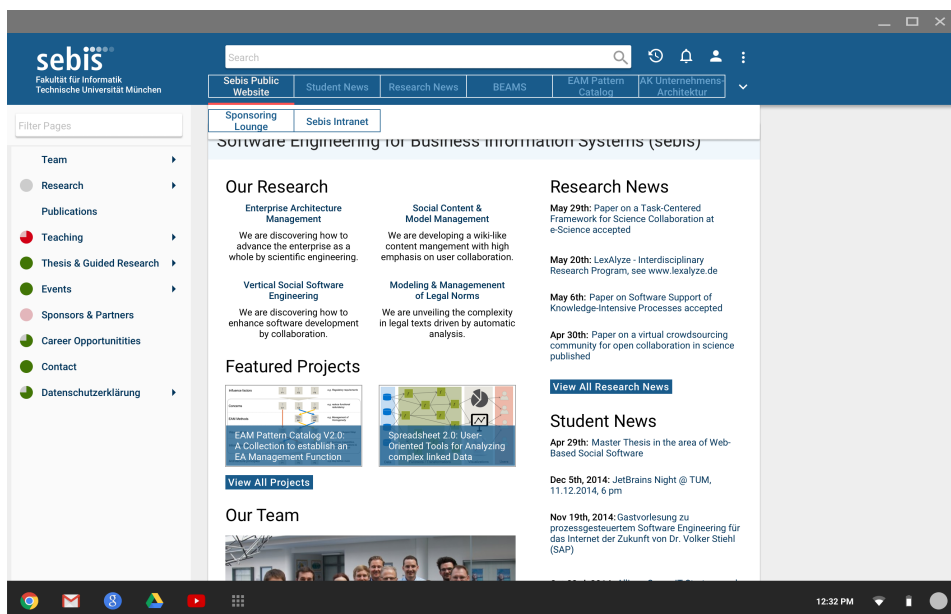


Figure A.17.: Windowed Design: Homepage - showing more Workspaces

A. Appendix

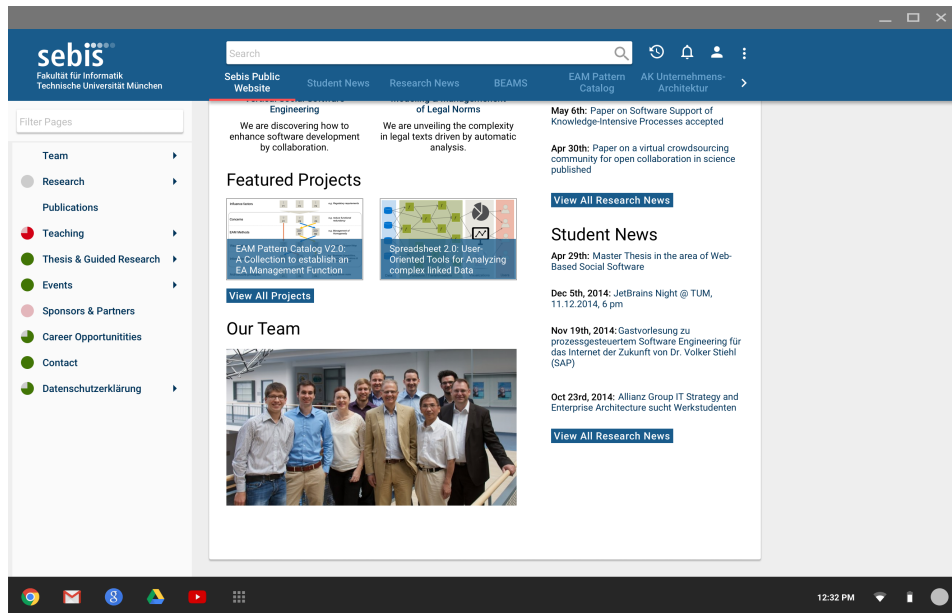


Figure A.18.: Windowed Design: Homepage - scrolled

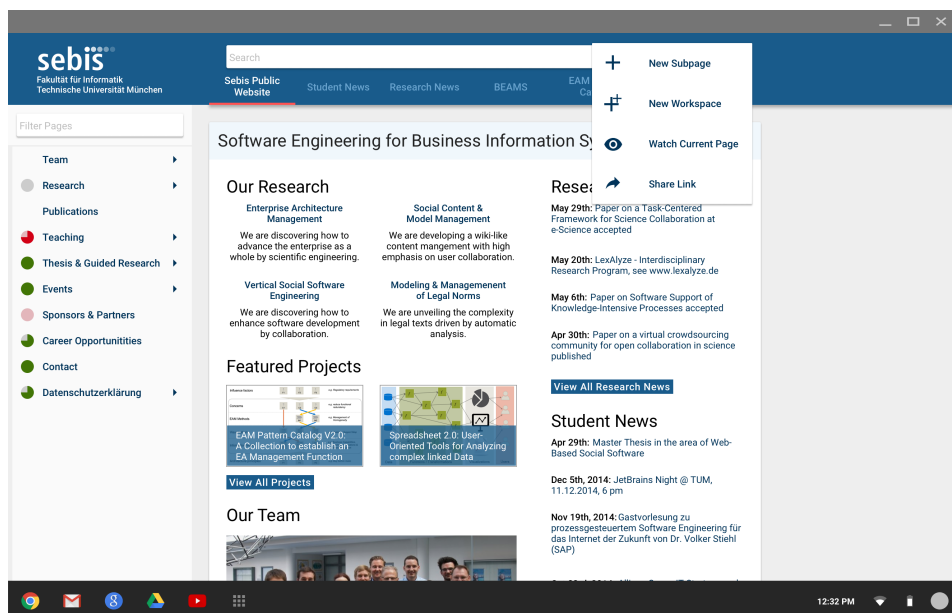


Figure A.19.: Windowed Design: Homepage - More-Icon clicked

A. Appendix

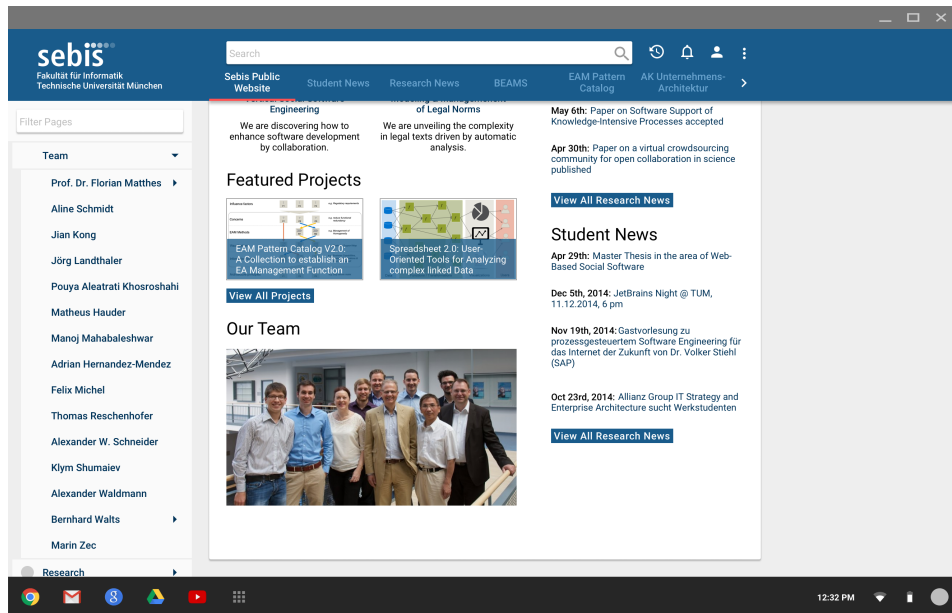


Figure A.20.: Windowed Design: Homepage - Team-Caret clicked

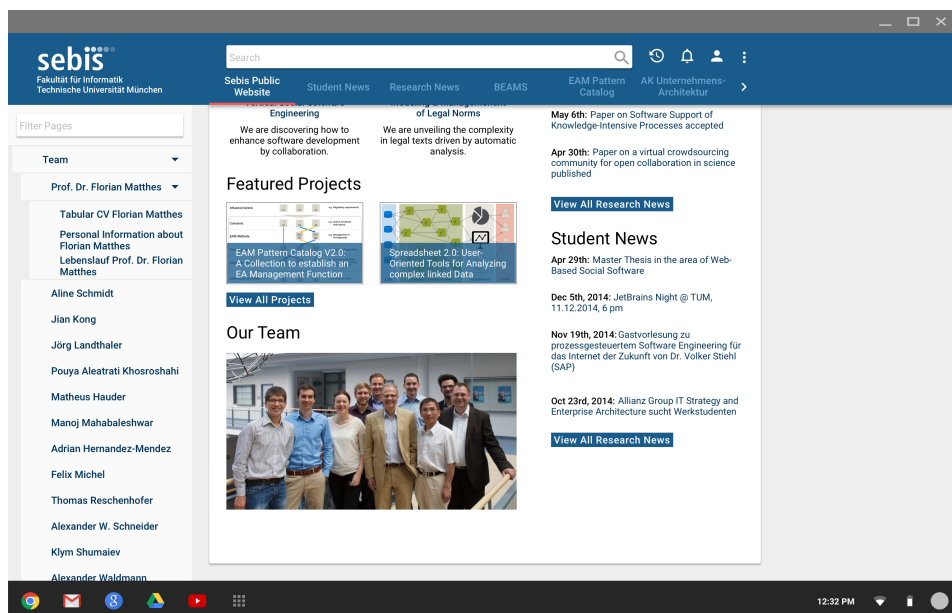


Figure A.21.: Windowed Design: Homepage - Prof. Matthes Caret clicked

A. Appendix

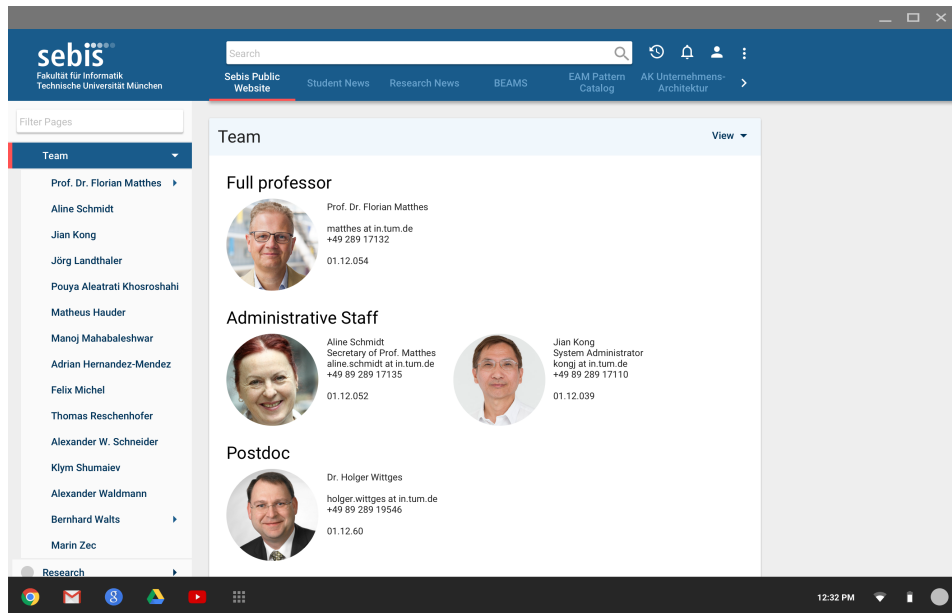


Figure A.22.: Windowed Design: Team

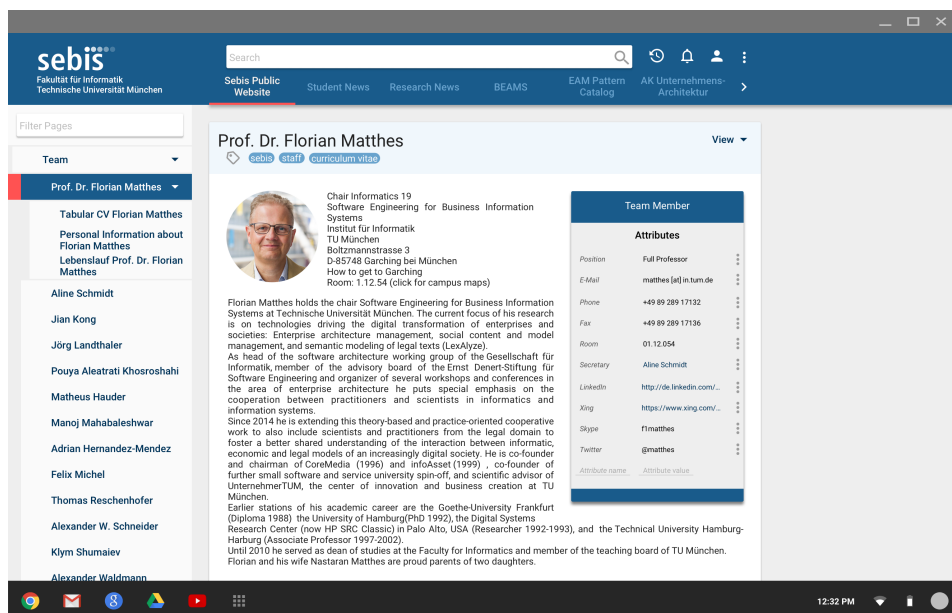


Figure A.23.: Windowed Design: Prof. Matthes

A. Appendix

The screenshot shows the 'sebis' website interface. The top navigation bar includes 'Sebis Public Website', 'Student News', 'Research News', 'BEAMS', 'EAM Pattern Catalog', and 'AK Unternehmens-Architektur'. The main content area is titled 'Prof. Dr. Florian Matthes' and includes a 'View' dropdown menu. The left sidebar lists team members, with 'Prof. Dr. Florian Matthes' selected. The main profile area features a photo of Prof. Matthes, his title 'Chair Informatics 19 Software Engineering for Business Information Systems', and a detailed biography. The right sidebar, titled 'Team Member', contains an 'Attributes' table with the following data:

Attributes	
Basic Information	
Position	Full Professor
E-Mail	matthes[at]in.tum.de
Contact Details	
Phone	+49 89 289 17132
Fax	+49 89 289 17136
Room	01.12.054
Secretary	Aline Schmidt
Social Media Details	
LinkedIn	http://de.linkedin.com/...
Xing	https://www.xing.com/...
Skype	fmatthes
Twitter	@matthes

Figure A.24.: Windowed Design: Prof. Matthes - Attribute Groups

This screenshot is similar to Figure A.24, showing the same profile page for Prof. Dr. Florian Matthes. The layout and content are identical, but it includes a 'last modified' timestamp at the bottom of the main profile area: 'Last modified by Florian Matthes, Jun 16th'.

Figure A.25.: Windowed Design: Prof. Matthes - last modified

A. Appendix

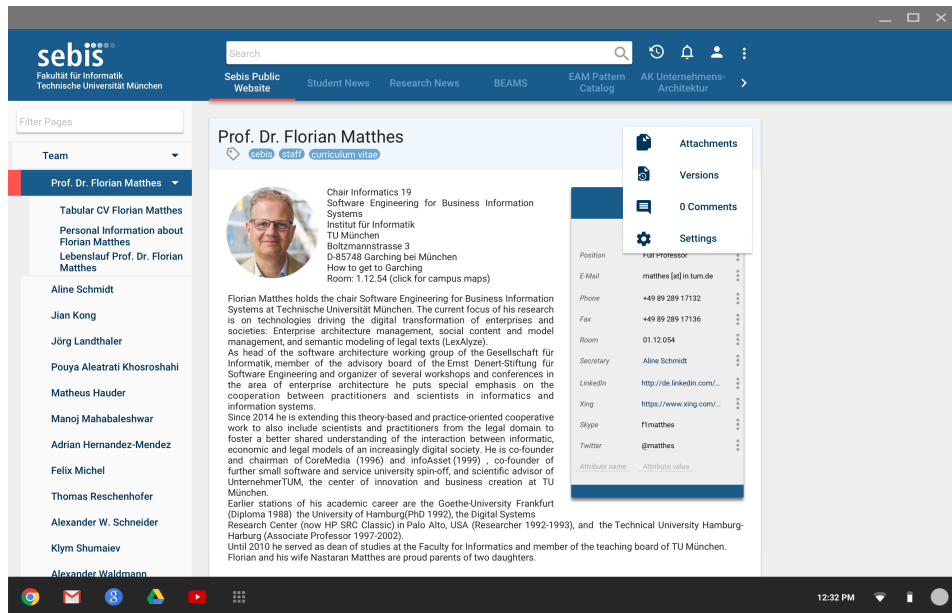


Figure A.26.: Windowed Design: Prof. Matthes - View clicked

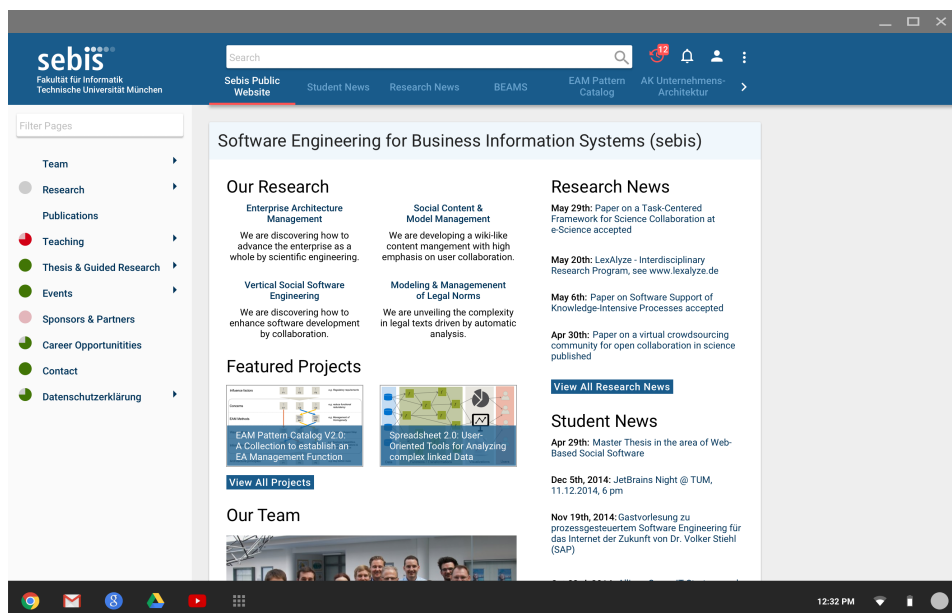


Figure A.27.: Windowed Design: Homepage - Feed Indicator

A. Appendix

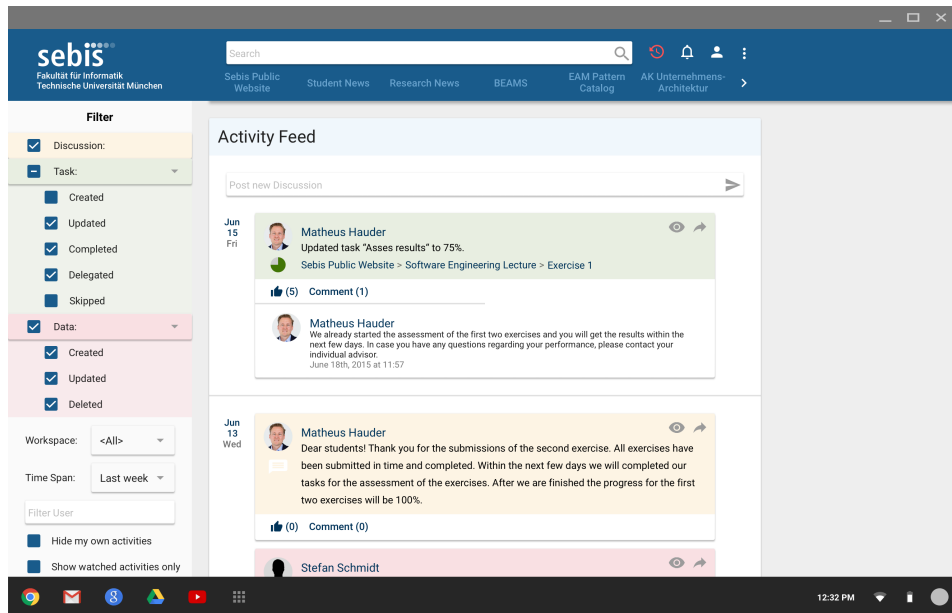


Figure A.28.: Windowed Design: Activity Feed

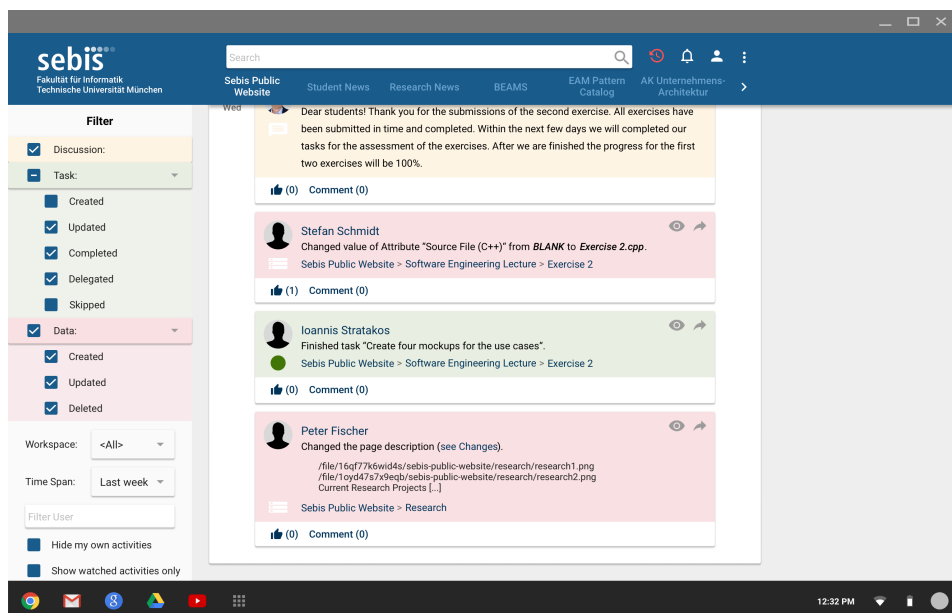


Figure A.29.: Windowed Design: Activity Feed - scrolled

A. Appendix

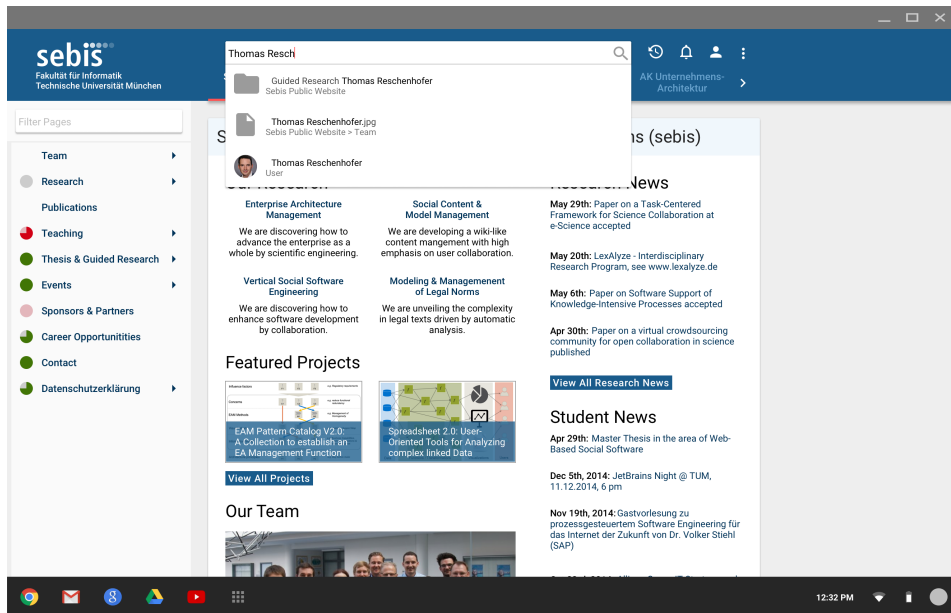


Figure A.30.: Windowed Design: Homepage - Search Bar used

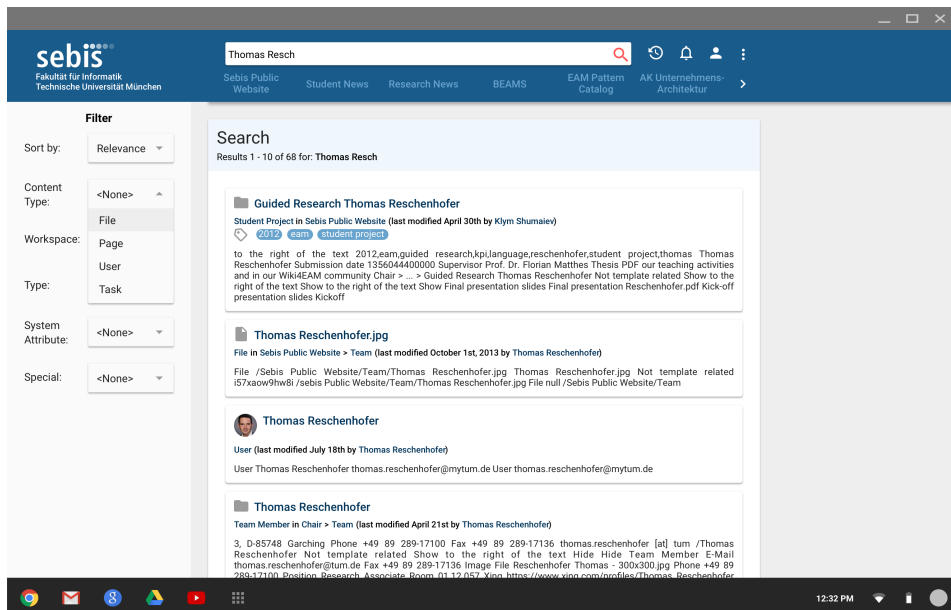


Figure A.31.: Windowed Design: Search Results

A. Appendix

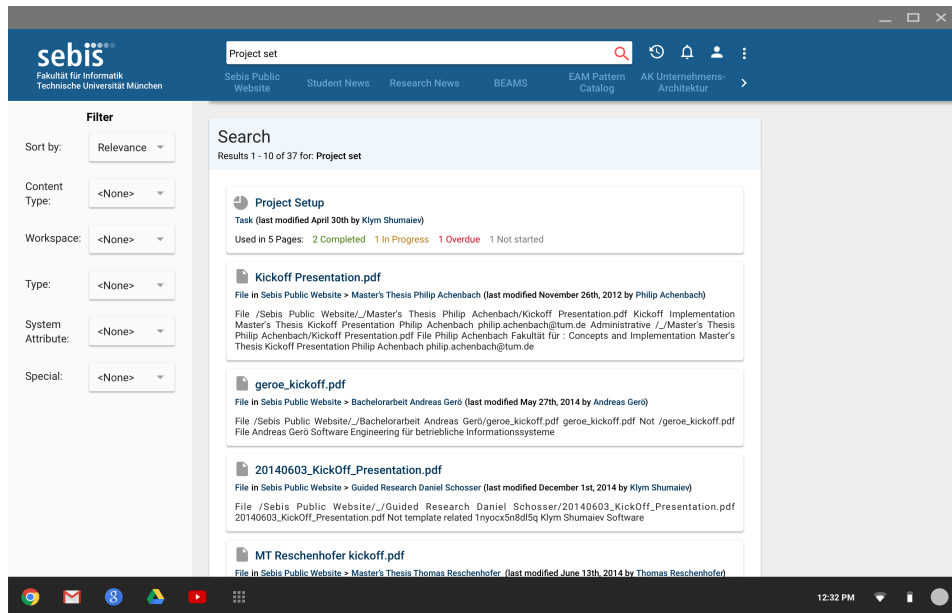


Figure A.32.: Windowed Design: Search Results - Task

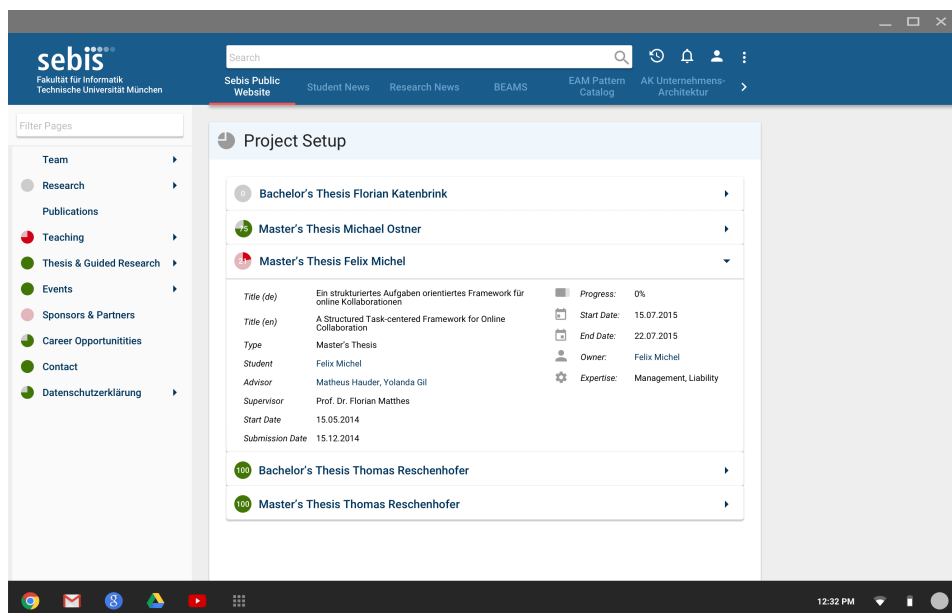


Figure A.33.: Windowed Design: Task Details

A. Appendix

sebis
Fakultät für Informatik
Technische Universität München

Search

Sebis Public Website | Student News | Research News | BEAMS | EAM Pattern Catalog | AK Unternehmens-Architektur

Florian Katenbrink

Name: Florian Katenbrink
Email: f.katenbrink@gmail.com
Last Login: 1 minute ago

Expertise

- Collaboration: 45%
- Computer Science: 26.8%
- Software Engineering: 12.8%
- Literatur Research: 8.5%
- Soft Skills: 6.2%
- Others: 0.7%

Current Tasks

- Complete Exercise 1** (75%)
Chair > Teaching > Global Software Engineering > Exercise 1
Due: 3 days ago
- Create four mockups for the use cases** (0%)
Chair > Teaching > Web Application Engineering > Project 3
Due: in 10 days
- Complete Exercise 2** (0%)
Chair > Teaching > Global Software Engineering > Exercise 2
Due: in 10 days

Future Tasks

- Develop the SSSP Algorithm** (0%)
Chair > Teaching > Algorithms 1 > Exercise 1
Starts: in 5 days

Completed Tasks

- Create two use-case diagrams** (100%)
Chair > Teaching > Web Application Engineering > Project 2
Completed: 5 days ago
- Complete Hello World Exercise** (100%)
Chair > Teaching > Web Application Engineering > Project 1
Completed: 7 days ago

Figure A.34.: Windowed Design: User Profile

sebis
Fakultät für Informatik
Technische Universität München

Search

Sebis Public Website | Student News | Research News | BEAMS | EAM Pattern Catalog | AK Unternehmens-Architektur

Master's Thesis Felix Michel

quid research | audio | organic data science | collaboration | masterthesis | open | case management

A Structured Task-Centered Framework for Online Collaboration

Abstract

Today's scientific research collaborations are often multidisciplinary across organizational borders and time-zones. Communication that is based on emails or teleconferences is time consuming. In recent years many approaches have focused on building and establishing on-line communities. Other approaches focus on managing effort such as organizing work as tasks. Collaboratively working teams could potentially increase their efficiency by combining the task centered approach with the community approach. However, no existing approach combines an on-line community platform and a task centered approach to provide an open collaboration process. This thesis presents the Organic Data Science approach which enables an open task centered on-line collaboration process. Key principles to address challenges of the task-centered collaboration approach are 1.) the self-organization of the community through task decomposition, 2.) an on-line community support based on social design principles and best practices and 3.) an open science process to enable unanticipated contributions. The task-centered Organic Data Science framework approach is implemented based on the Semantic MediaWiki platform. The prototype implementation of the Organic Data Science framework is evaluated through a research project focused on the science question of modeling the age of water in an ecosystem. This project requires expertise in different research areas from multiple organizations within different time-zones. Different collaboration dimensions are evaluated such as how many different users access a task, how many different users are assigned to a task, how many different users edit the task metadata and how many different users edit the task content. The findings show that the framework supports the collaboration process. In general the Organic Data Science framework is designed for helping scientists to collaborate to solve complex scientific research questions. The use of the Organic Data Science framework is not limited to scientific purpose, it helps to support complex knowledge intensive collaborative processes.

Student Project	
Attributes	Tasks
Title (de)	Ein strukturiertes Aufgaben orientiertes Framework für online Kollaborationen
Title (en)	A Structured Task-Centered Framework for Online Collaboration
Project	Darwin
Type	Master's Thesis
Student	Felix Michel
Advisor	Matheus Haeder, Yolande Gil
Supervisor	Prof. Dr. Florian Matthes
Start Date	15.05.2015
Contributor Agreement signed on	
Checklist filled	Yes
Submission date	15.12.2015
Kickoff presentation slides	
Final presentation slides	
Thesis PDF	

Figure A.35.: Windowed Design: Master's Thesis - Attributes

A. Appendix

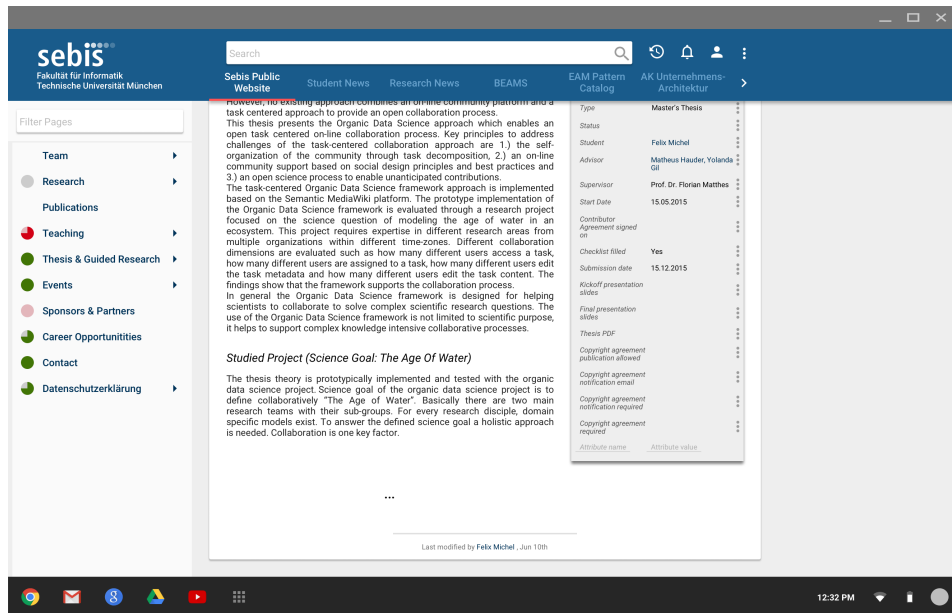


Figure A.36.: Windowed Design: Master's Thesis - scrolled

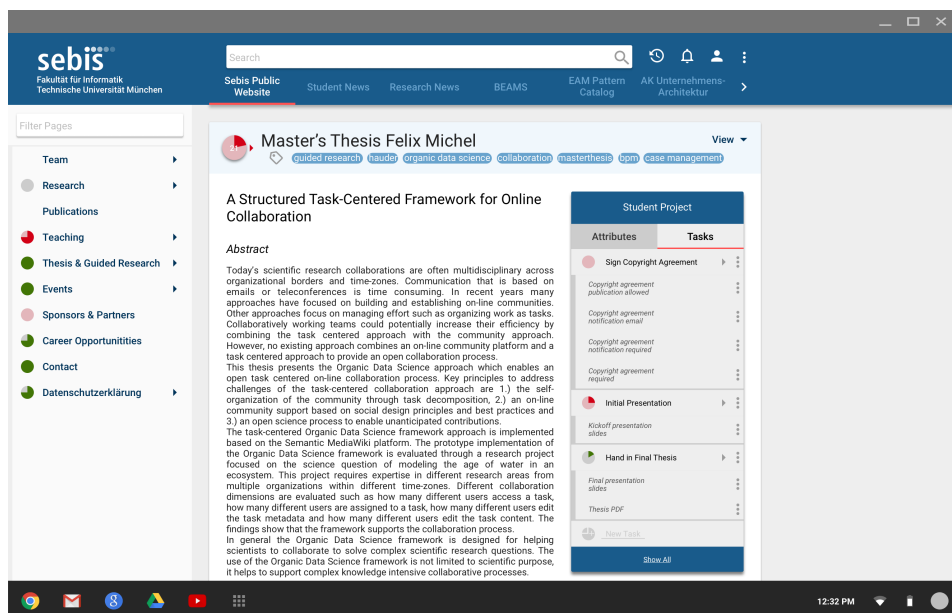


Figure A.37.: Windowed Design: Master's Thesis - Tasks

A. Appendix

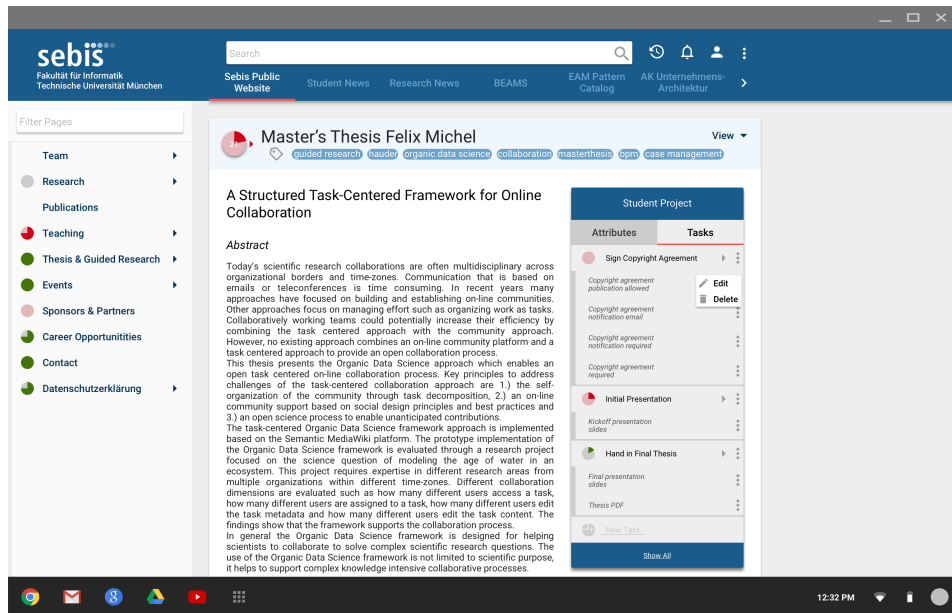


Figure A.38.: Windowed Design: Master's Thesis - Edit Attribute

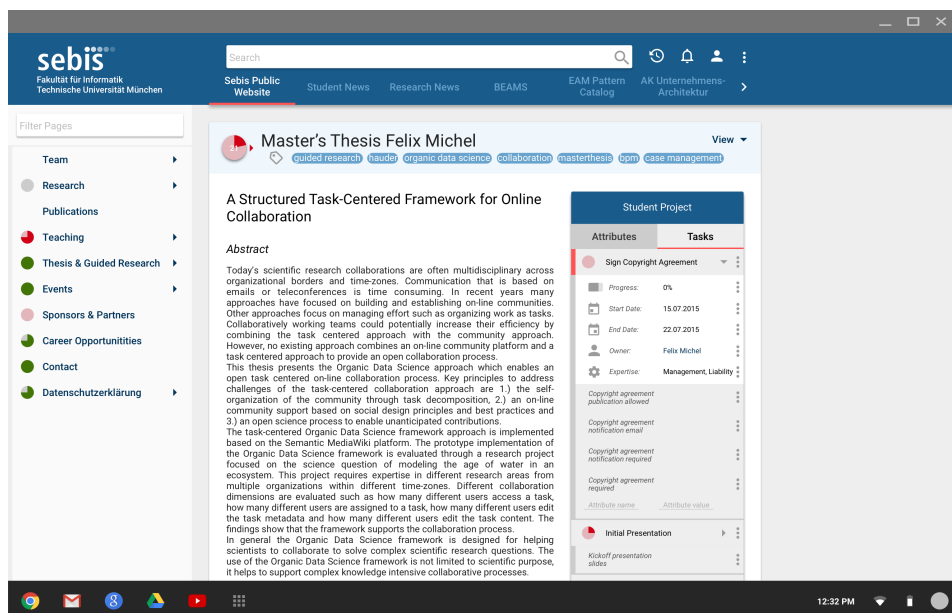


Figure A.39.: Windowed Design: Master's Thesis - Task clicked

A. Appendix

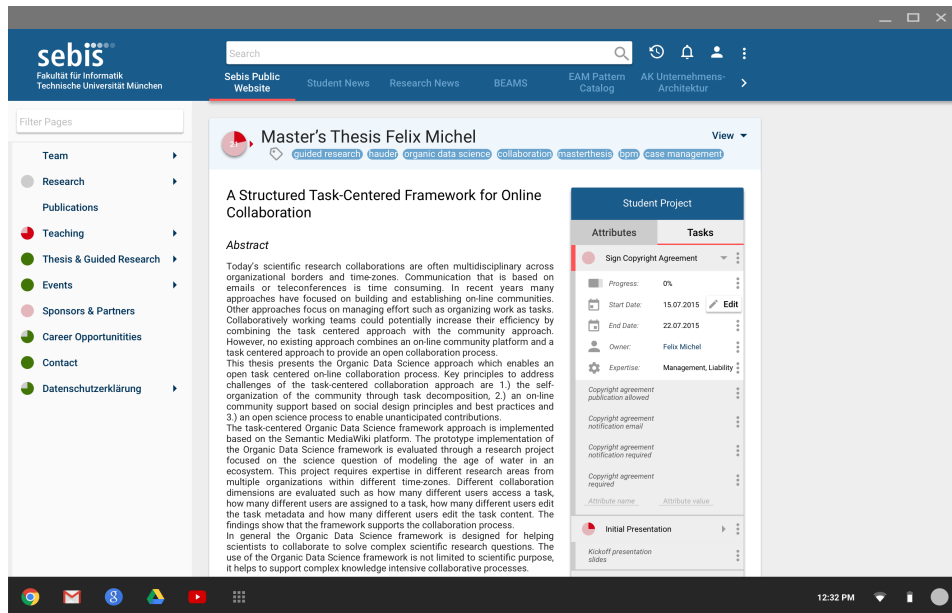


Figure A.40.: Windowed Design: Master's Thesis - Edit Metadata

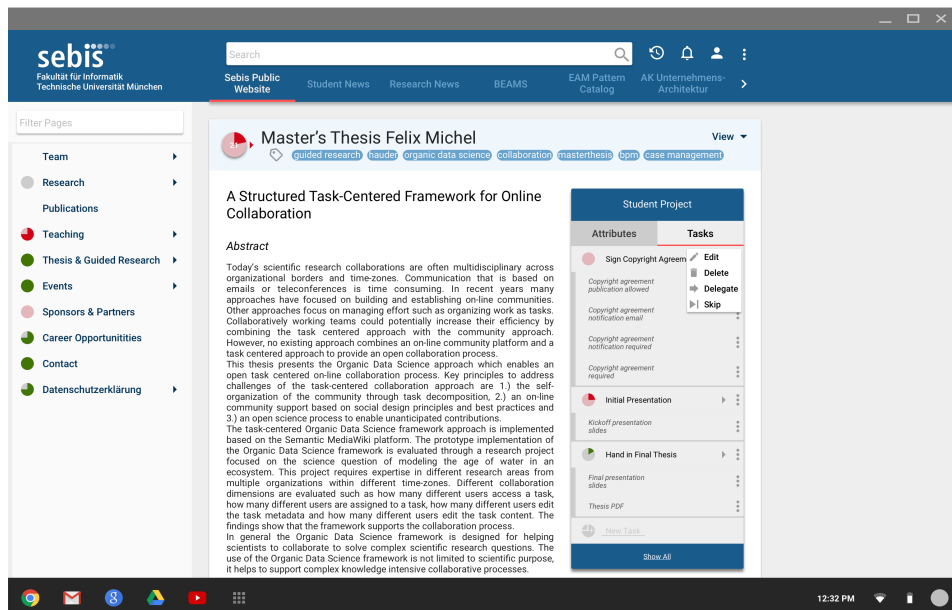


Figure A.41.: Windowed Design: Master's Thesis - Edit Task

A. Appendix

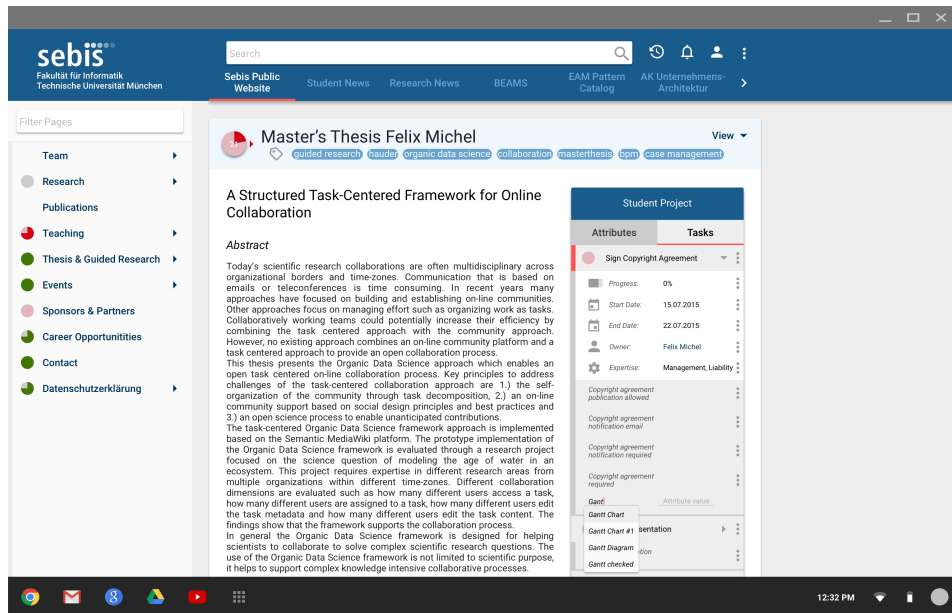


Figure A.42.: Windowed Design: Master's Thesis - Autocompletion

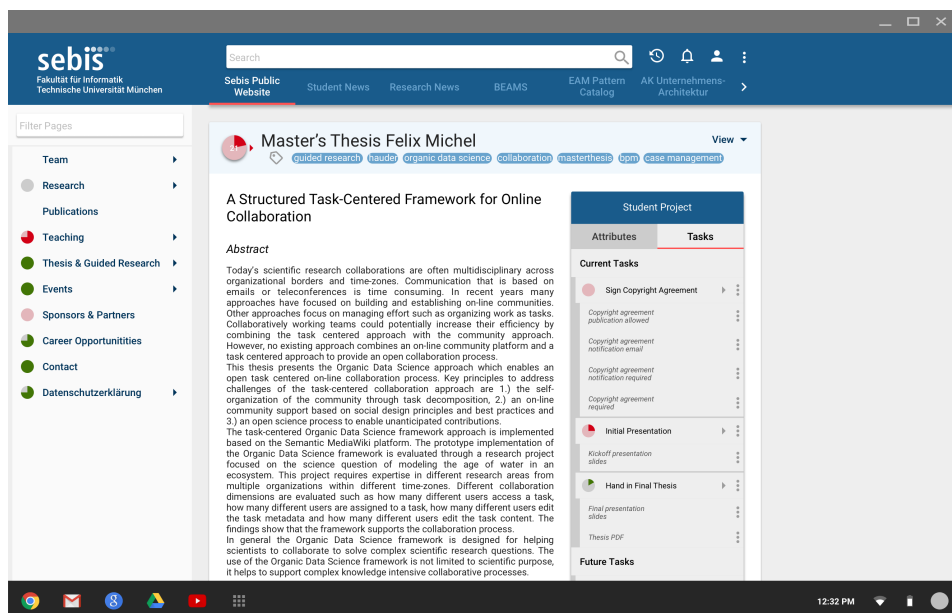


Figure A.43.: Windowed Design: Master's Thesis - Show all Tasks

A. Appendix

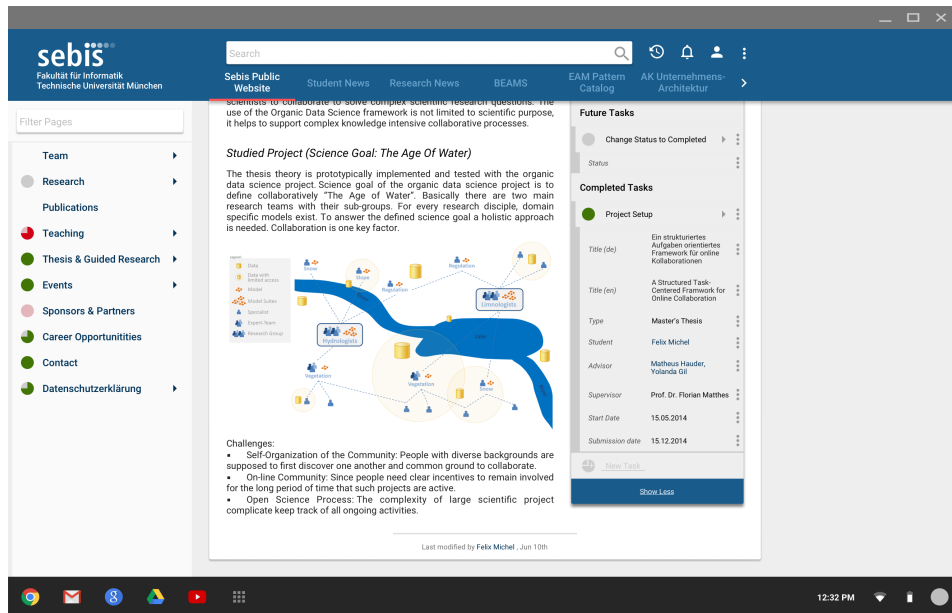


Figure A.44.: Windowed Design: Master's Thesis - Show all Tasks, scrolled

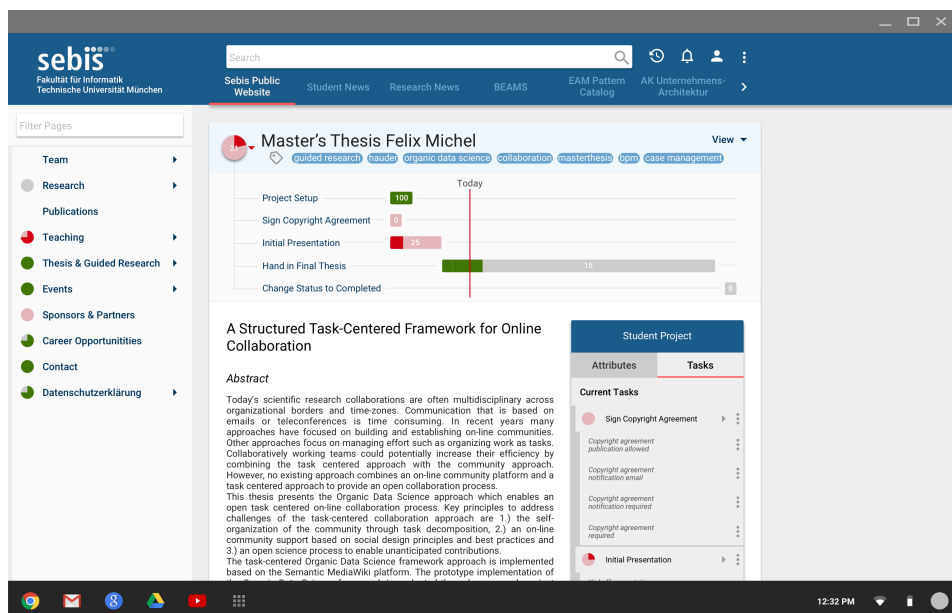


Figure A.45.: Windowed Design: Master's Thesis - Gantt chart

A. Appendix

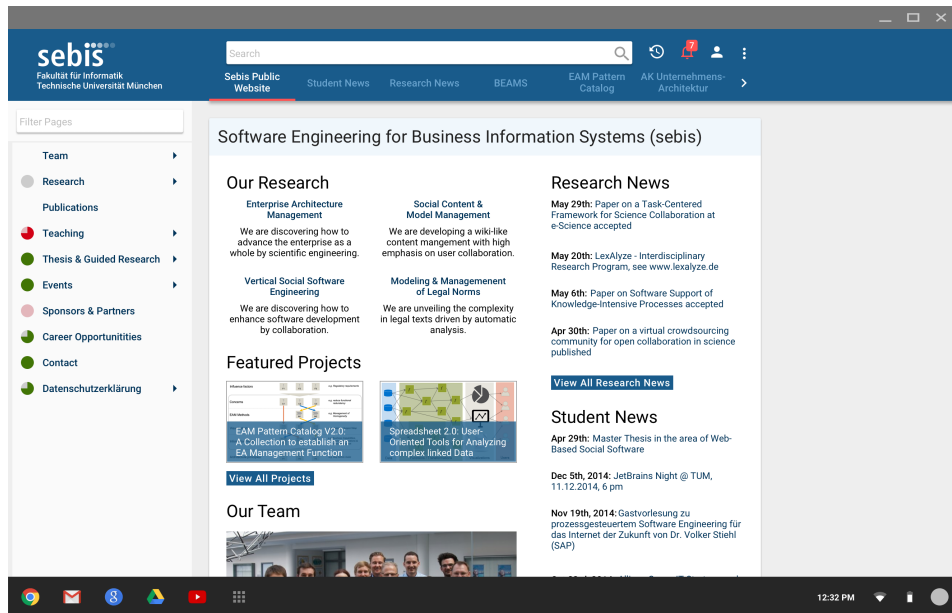


Figure A.46.: Windowed Design: Master's Thesis - Alerts Indicator

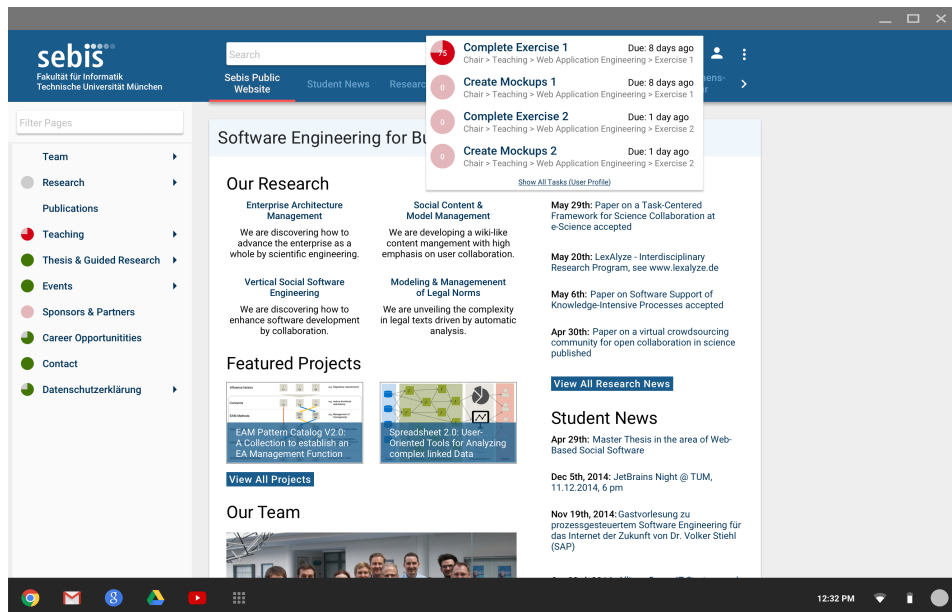


Figure A.47.: Windowed Design: Master's Thesis - Alerts

List of Figures

2.1. Tricia: strong coupling of back-end and front-end.	5
2.2. SocioCortex: Separation of back-end and front-end.	5
2.3. Colors and progress	6
2.4. Design Principles of User Experience Optimization	9
3.1. The Adapted Research Framework [He04]	12
4.1. The Generic Web Application: Color Palette	14
4.2. Tricia vs. the Generic Web Application: The Homepage ²³	16
4.3. Tricia vs. the Generic Web Application: The Application Bar ²⁴ .	18
4.4. The Generic Web Application: More Workspaces shown	18
4.5. The Generic Web Application: More-Icon clicked	19
4.6. Tricia vs. the Generic Web Application: The Sidebar ²⁵	20
4.7. Tricia vs. the Generic Web Application: Caret Navigation ²⁶ . .	21
4.8. Tricia vs. the Generic Web Application: Sidebar Navigation ²⁷ .	23
4.9. The Generic Web Application: Task States	25
4.10. Tricia vs. the Generic Web Application: Homepage Wiki Con- tent ²⁸	26
4.11. Tricia vs. the Generic Web Application: The Wiki Page ²⁹	27
4.12. The Generic Web Application: View-Menu selected	28
4.13. The Generic Web Application: The Last modified Footer	29
4.14. Tricia vs. the Generic Web Application: Attributes ³⁰	29
4.15. The Generic Web Application: Attribute Groups	30
4.16. The Generic Web Application: Master's Thesis	31
4.17. The Generic Web Application: Selecting a task	32
4.18. The Generic Web Application: Autocompleting an Attribute . .	33
4.19. The Generic Web Application: The Edit Menus	33
4.20. The Generic Web Application: All Tasks shown	34
4.21. The Generic Web Application: The Gantt Chart	35
4.22. Tricia vs. the Generic Web Application: The User Profile ³¹ . . .	36

List of Figures

4.23. The Generic Web Application: The Activity Feed Indicator . . .	37
4.24. Tricia vs. the Generic Web Application: The Activity Feed ³² . . .	38
4.25. Tricia vs. the Generic Web Application: Data Activities ³³ . . .	40
4.26. Tricia vs. the Generic Web Application: Using the Search Bar ³⁴	41
4.27. Tricia vs. the Generic Web Application: Search Results ³⁵	42
4.28. Tricia vs. the Generic Web Application: The File Search Result ³⁶	43
4.29. The Generic Web Application: The Task Search Result	44
4.30. The Generic Web Application: Task Details	45
4.31. The Generic Web Application: The Alert Indicator	45
4.32. The Generic Web Application: Alerts	45
5.1. An Evaluation of the 7 Research-Framework Guidelines	47
A.1. Evolution of the Navigation: Part I	i
A.2. Evolution of the Navigation: Part II	ii
A.3. Evolution of the Navigation: Part III	ii
A.4. Evolution of the Navigation: Part IV	iii
A.5. Evolution of the Navigation: Part V	iii
A.6. Evolution of the Navigation: Part VI	iv
A.7. Evolution of the Navigation: Part VII	iv
A.8. Evolution of the Navigation: Part VIII	v
A.9. Evolution of the Navigation: Part IX	v
A.10. Evolution of the Navigation: Part X	vi
A.11. Sidebar Scrolling: Part I	vii
A.12. Sidebar Scrolling: Part II	vii
A.13. Sidebar Scrolling: Part III	viii
A.14. Sidebar Scrolling: Part IV	viii
A.15. Sidebar Scrolling: Part V	ix
A.16. Windowed Design: Homepage	x
A.17. Windowed Design: Homepage - showing more Workspaces	x
A.18. Windowed Design: Homepage - scrolled	xi
A.19. Windowed Design: Homepage - More-Icon clicked	xi
A.20. Windowed Design: Homepage - Team-Caret clicked	xii
A.21. Windowed Design: Homepage - Prof. Matthes Caret clicked . . .	xii
A.22. Windowed Design: Team	xiii
A.23. Windowed Design: Prof. Matthes	xiii
A.24. Windowed Design: Prof. Matthes - Attribute Groups	xiv

A.25.Windowed Design: Prof. Matthes - last modified	xiv
A.26.Windowed Design: Prof. Matthes - View clicked	xv
A.27.Windowed Design: Homepage - Feed Indicator	xv
A.28.Windowed Design: Activity Feed	xvi
A.29.Windowed Design: Activity Feed - scrolled	xvi
A.30.Windowed Design: Homepage - Search Bar used	xvii
A.31.Windowed Design: Search Results	xvii
A.32.Windowed Design: Search Results - Task	xviii
A.33.Windowed Design: Task Details	xviii
A.34.Windowed Design: User Profile	xix
A.35.Windowed Design: Master's Thesis - Attributes	xix
A.36.Windowed Design: Master's Thesis - scrolled	xx
A.37.Windowed Design: Master's Thesis - Tasks	xx
A.38.Windowed Design: Master's Thesis - Edit Attribute	xxi
A.39.Windowed Design: Master's Thesis - Task clicked	xxi
A.40.Windowed Design: Master's Thesis - Edit Metadata	xxii
A.41.Windowed Design: Master's Thesis - Edit Task	xxii
A.42.Windowed Design: Master's Thesis - Autocompletion	xxiii
A.43.Windowed Design: Master's Thesis - Show all Tasks	xxiii
A.44.Windowed Design: Master's Thesis - Show all Tasks, scrolled . .	xxiv
A.45.Windowed Design: Master's Thesis - Gantt chart	xxiv
A.46.Windowed Design: Master's Thesis - Alerts Indicator	xxv
A.47.Windowed Design: Master's Thesis - Alerts	xxv

OPTIMIZING THE USER
EXPERIENCE OF A SOCIAL
CONTENT MANAGEMENT
SOFTWARE FOR CASUAL USERS

Florian Katenbrink