

Subtile Hinweise und automatische Zeitplanung für die verbesserte Benutzerfreundlichkeit in geführten Arbeitsabläufen

EVA LOIDL



MASTERARBEIT

eingereicht am
Fachhochschul-Masterstudiengang

Interactive Media

in Hagenberg

im Juni 2017

© Copyright 2017 Eva Loidl

Diese Arbeit wird unter den Bedingungen der *Creative Commons Lizenz Namensnennung–NichtKommerziell–KeineBearbeitung Österreich* (CC BY-NC-ND) veröffentlicht – siehe <http://creativecommons.org/licenses/by-nc-nd/3.0/at/>.

Erklärung

Ich erkläre eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Hagenberg, am 26. Juni 2017

Eva Loidl

Inhaltsverzeichnis

Erklärung	iii
Gendering	vii
Kurzfassung	viii
Abstract	ix
1 Einleitung	1
1.1 Motivation und Zielsetzung	1
1.2 Struktur und Aufbau	2
2 Aktueller Forschungsstand	4
2.1 Benutzerführung	4
2.2 Webanimation	4
2.2.1 Natürliche, familiäre Bewegungen	5
2.2.2 Visuelle Hinweise	6
2.2.3 Visuelle Rückmeldungen	6
2.2.4 Übergänge und Systemstatus	7
2.3 Cartoon-Animationen	8
2.4 Subtile Hinweise zur Verbesserung der Benutzererfahrung	8
2.4.1 Aktive und passive Hilfesysteme	9
2.4.2 Statische und dynamische Hilfesysteme	11
2.4.3 Einheitliche und individuelle Hilfesysteme	11
2.4.4 Synchron und Asynchrone Hilfesysteme	12
2.5 Beispiele von subtilen Hinweisen	12
2.5.1 Textuelle Hinweise	12
2.5.2 Gitternetzlinien	13
2.5.3 Hilfslinien, Smart Guides und Snapping	14
2.5.4 Farbwechsel	15
2.5.5 Erweitertes Drag & Drop	16
3 Konzeption	20
3.1 Basisfunktionalität und Anforderungen	20

3.1.1	Modulares System	20
3.1.2	Validitätsprüfung	22
3.1.3	Platzieren der Module	23
3.1.4	Bearbeiten der Module	24
3.1.5	Anpassbarkeit und optionale Hilfestellungen	24
3.1.6	Zeitlinie und Agenda	25
3.2	Visualisierungsentwürfe	25
3.2.1	Entwurf 1 – Baum	25
3.2.2	Entwurf 2 – Lego-Steine	27
3.2.3	Entwurf 3 – Post-It's	28
4	Implementierung	30
4.1	Architektur	30
4.2	Verwendete Bibliotheken	31
4.2.1	interact.js	31
4.2.2	Google Charts Timeline	32
4.2.3	AlertifyJS	33
4.2.4	Bootstrap	35
4.3	Die drei Komponenten des Web-Modul-Editors	36
4.3.1	Toolbox	36
4.3.2	Arbeitsfläche	37
4.3.3	Automatisierte Zeitplanung	40
5	Evaluierung	45
5.1	Pilotstudie	45
5.1.1	Struktur des Benutzerfreundlichkeitstests	45
5.1.2	Aufbau und Ablauf der Test-Session	48
5.1.3	Ergebnisse	49
5.1.4	Verbesserungs- und Erweiterungsmöglichkeiten	55
6	Schlussbemerkungen	58
6.1	Zusammenfassung	58
6.2	Resümee	59
6.3	Ausblick	60
A	*.json-Datei	61
B	Fragebogen Pilotstudie	63
C	Inhalt der CD-ROM	67
C.1	PDF-Dateien	67
C.2	Online-Quellen	67
C.3	Abbildungen	67
C.4	Projektdateien	67

Inhaltsverzeichnis	vi
Quellenverzeichnis	68
Literatur	68
Online-Quellen	69

Gendering

Aus Gründen der Lesbarkeit und der sprachlichen Vereinfachung wird in dieser Arbeit auf gendergerechte Didaktik verzichtet. Jegliche Aussagen dieser Arbeit sind als geschlechtsneutral zu betrachten, verwendete Bezeichnungen schließen sowohl die männliche als auch weibliche Form ein, sofern nicht ausdrücklich darauf hingewiesen wird, dass es sich nur um eine der beiden Formen handelt.

Kurzfassung

Durch die stetig steigende Zahl an Webseiten und Online-Anwendungen wird es immer schwieriger Benutzer an neue Interfaces heranzuführen und sie vom Einsatz eines anderen Systems beziehungsweise Umstieg auf ebenjenes zu überzeugen. Findet sich ein Nutzer nicht auf Anhieb mit der Schnittstelle zurecht, so wendet er vorab meist nicht die nötige Zeit auf um das Tool zu erlernen. Eine Möglichkeit die Benutzerfreundlichkeit und Benutzererfahrung zu steigern bietet der Einsatz von subtilen Hinweisen durch Webanimationen. Diese unterstützen den Nutzer unterbewusst und gewährleisten so eine problemlose Interaktion beim Kennenlernen des Interfaces.

In der vorliegenden Diplomarbeit wird neben der Vorstellung unterschiedlicher subtiler Hinweise als Anwendungsbeispiel ein Konzept entwickelt, das Moderatoren bei der Planung von Workshops unterstützt. Es werden diverse Hinweise implementiert, die dem Benutzer ermöglichen, das Tool ohne Einführung durch geschultes Fachpersonal oder zahlreicher textueller Beschreibung, zu bedienen. Im Anschluss folgt eine Pilotstudie mit Testkandidaten, die zur Evaluierung des praktischen Teiles der Arbeit erheblich beiträgt. Dadurch werden potentielle Schwachstellen des Tools sowie Verbesserungsmöglichkeiten ermittelt, die zum einen Teil anschließend implementiert, zum anderen Teil zur Liste der zukünftigen Erweiterungen hinzugefügt wurden.

Abstract

Due to the increasing amount of websites and online applications it becomes more and more difficult to bring users to new interfaces and convince them to change from a known system to a new one. If a user has troubles with interaction in the new interface, he usually does not take the time to get familiar with the tool in advance. One way to increase both usability and user experience is the use of subtle hints, realized with web animations. These support the user subconsciously and thus ensure a smooth interaction when learning the interface.

This thesis deals beyond the introduction of various subtle hints using a web-module-editor as a role model. A concept was created, which is meant to support moderators in planning their workshops. Different subtle hints are implemented. They allow the user to operate the tool without introductions by trained staff or adding numerous textual descriptions. This is followed by a pilot study performed with test candidates, which contribute significantly to the evaluation of the practical part of the present work. As a result potential weaknesses of the tool as well as improvements are determined and either realized or elaborated for future extensions.

Kapitel 1

Einleitung

Benutzerfreundlichkeit manifestierte sich in den letzten Jahren als eine wichtige und mittlerweile unumgängliche Komponente des Web Designs. Der internationale Standard ISO/DIS 9241-11.2 definiert Benutzerfreundlichkeit wie folgt [19]:

“The extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.”

In diesem Fall bedeutet Effektivität die Genauigkeit und Vollständigkeit, mit jener Benutzer ein bestimmtes Ziel in einer gegebenen Umgebung erreichen können. Effizienz misst die Ressourcen, die aufgewendet müssen, in Relation zur Genauigkeit und Vollständigkeit der erreichten Ziele. Hinsichtlich der Zufriedenheit wird der Benutzer, sowie Menschen, die bei der Benützung des Systems betroffen sind, über den Komfort und die Akzeptanz befragt [19].

Nielsen betont, dass es möglich ist, die Lebensqualität der Benutzer mittels Benutzerfreundlichkeit zu verbessern, da diese nicht stetig von der Benutzung eines Computers frustriert und entmutigt werden, sofern eine Webseite einfach zu bedienen ist [5, S. 7].

Nutzer durch einen komplexen Arbeitsablauf zu führen, ohne diese zu überfordern, ist meist nicht so einfach. Eine Möglichkeit bieten subtile Hinweise als Rückmeldung für den Benutzer, die mittels Webanimationen realisiert und in dieser Arbeit genauer diskutiert werden.

1.1 Motivation und Zielsetzung

Bewegung zieht die meiste Aufmerksamkeit auf sich und ist somit ein sehr mächtiges Werkzeug. Aufgrund dieser Tatsache werden Animationen im Web oft als beunruhigend, störend und überflüssig empfunden, sofern diese nicht

mit Bedacht eingesetzt werden. Dies gilt besonders für dekorative Animationen, die bis auf die visuelle Erscheinung keinerlei Nutzen haben. Dennoch können Animationen, vorsichtig und korrekt verwendet, den Benutzer beim Interagieren mit einem System unterstützen und dadurch die Benutzerfreundlichkeit einer Schnittstelle erheblich steigern. Geführte Arbeitsabläufe sind eines der Anwendungsgebiete von funktionalen Rückmeldungen [9, 14].

Als Veranschauungsobjekt dieser Arbeit dient ein Web-Modul-Editor, der Moderatoren bei der Planung ihrer Workshops und Seminare behilflich ist. Da es dem Benutzer ermöglicht werden soll, die Schnittstelle ohne verbalen und textuellen Instruktionen zu benutzen, wird eine Art geführter Arbeitsablauf durch die Implementierung unterschiedlicher subtiler Hinweise realisiert. Diese Hinweise werden diskutiert und erläutert, gefolgt von einer Pilotstudie um herauszufinden wie effektiv sie in der praktischen Anwendung sind. Abschließend werden die Ergebnisse der Studie ausgewertet, wodurch neuer Input und Verbesserungsvorschläge für den webbasierten Editor gewonnen werden. Um die Planung für beispielsweise Workshops effektiver zu gestalten, beinhaltet der Modul-Editor auch eine Zeitplanungskomponente, die vom Moderator direkt als Basis der Agenda übernommen werden kann.

Ziel dieser Arbeit ist es, unterschiedliche Möglichkeiten für Benutzerhinweise durch Webanimation aufzuzeigen, die dazu dienen, dem Nutzer die Verwendung von Webseiten und Webanwendungen zu vereinfachen. Es werden Möglichkeiten gesucht, Benutzern eine visuelle Rückmeldung zu geben, ohne dass diese als störend empfunden wird. Zudem sollten sowohl Anfänger als auch Experten mit der gleichen Schnittstelle angesprochen werden. Als Ergänzung soll ein Weg gefunden werden einfache Zeitplanung zu realisieren.

1.2 Struktur und Aufbau

Die Arbeit gliedert sich in sechs Kapitel. Nachdem sich Kapitel 1 mit der Einleitung, der Motivation und Zielsetzung sowie der Struktur auseinandersetzt, beschreibt Kapitel 2 die Grundlagen der Arbeit. Dazu werden anfänglich Grundbegriffe wie Benutzerführung, Webanimation und Cartoon-Animation erläutert, daraufhin folgt eine Einführung in die unterschiedlichen Hilfesysteme, auf denen subtile Hinweise basieren. Den letzten Teil des Kapitels bilden diverse Beispiele sowie verwandte Arbeiten zum Thema subtile Hinweise.

Kapitel 3 beschäftigt sich mit der Konzeption der Implementierung dieser Arbeit. Nach einer konkreten Beschreibung der Basisfunktionalitäten und Anforderungen werden mögliche Visualisierungsentwürfe vorgestellt und diskutiert. Kapitel 4 beschreibt die eigentliche Implementierung basierend auf die zuvor erläuterte Konzeption. Es werden die verwendeten Bibliotheken vorgestellt, anschließend wird der Aufbau und die Umsetzung des Web-Modul-Editors in drei Abschnitten erläutert. Die Ergebnisse der Implemen-

tierung werden in Kapitel 5 beschrieben und mithilfe einer Pilotstudie evaluiert. Das Kapitel beschreibt neben dem Aufbau und Ablauf der Test-Session die Datenevaluierung und die daraus resultierenden Ergebnisse. Weiters werden Verbesserungs- und Erweiterungsmöglichkeiten diskutiert.

Kapitel 6 enthält neben einer Zusammenfassung eine Schlussfolgerung der gesamten Arbeit sowie einen Ausblick für aufbauende Arbeiten oder Projekte.

Kapitel 2

Aktueller Forschungsstand

2.1 Benutzerführung

Benutzer beschäftigen sich meist nur sehr ungern mit einem System, das sie kostbare Zeit kostet, da es zahlreiche Klicks erfordert, bis das gewünschte Ergebnis erreicht wird. Besonders bei Webseiten und kostenlosen beziehungsweise alternativen Programmen ist die Bereitschaft, sich über längere Zeit mit einer Schnittstelle zu beschäftigen und sich mit dieser vertraut zu machen sehr gering. Um dieser Tatsache entgegenzuwirken wird eine Handhabungsweise benötigt um den Benutzer schnell und behutsam an sein Ziel zu bringen. Eine Option bei linearen Abläufen, bei denen mehrere Schritte der Reihe nach ausgeführt werden müssen, ist, den Anwender durch den gesamten Prozess zu führen. Die größte Herausforderung bei der Benutzerführung besteht darin, unerfahrene Nutzer nicht mit zu vielen Optionen zu überfordern aber gleichzeitig erfahrenen Benutzern eigenwillige Schritte zu ermöglichen [6, S. 329].

Aufgrund der Behinderung einer flüssigen Interaktion sind textuelle Beschreibungen für einzelne Schritte meist eine suboptimale Lösung. Im Idealfall ist der Arbeitsablauf selbsterklärend, einfach und mittels wenigen Instruktionen zu bewältigen.

2.2 Webanimation

Mittlerweile existiert eine Vielzahl an Möglichkeiten zur Implementierung plattformübergreifender Animationen im Netz, ohne dabei auf veraltete Plugins wie Flash angewiesen zu sein. Das Aufkommen von HTML5 und CSS3 ermöglichte einen neuen Weg, um bewegte Objekte auch im Web umzusetzen. Zusätzlich können diese Technologien mit SVG-Grafiken und einer immer größer werdenden Anzahl an *JavaScript* Bibliotheken kombiniert werden, um die Basisfunktionalität zu erweitern.

Bewegung ist eine mächtige Funktionalität um Aufmerksamkeit zu er-

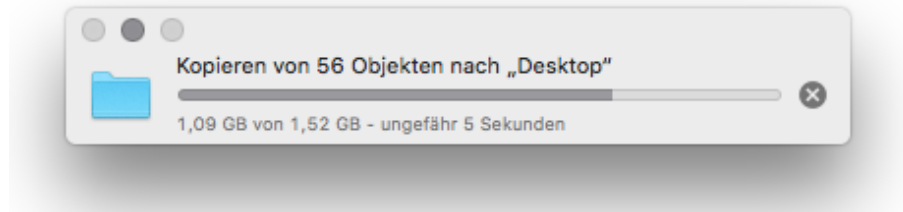


Abbildung 2.1: Fortschrittsindikator auf OS X.

regen. Richtig und behutsam eingesetzt, können Animationen sowohl die Benutzerinteraktion und die Benutzerfreundlichkeit steigern, als auch eine positive Wahrnehmung für eine Applikation vermitteln.

Animation ermöglicht es, durch subtile Bewegungen zusätzliche Informationen zu vermitteln, sofern die Bewegung vom Benutzer selbst ausgelöst wird. Diese sollten im Idealfall beinahe unterbewusst wahrgenommen werden, wenn sie mit Bedacht eingesetzt werden. Sie sollten für den Anwender unauffällig und elegant sowie reaktionsschnell und modern erscheinen [14].

Animation hat sich mittlerweile zu einem wertvollen Werkzeug entwickelt um dem Benutzer sinndienliche Rückmeldungen zu geben. Sehr bekannte Beispiele dafür sind Hover-Effekte auf Buttons und Farbwechsel, wenn diese aktiviert werden. Weitere bekannte Bewegungen sind Fortschrittsindikatoren wie Ladebalken, Sanduhren (findet bei Windows Anwendung) oder sogenannte Beach-Balls (von Mac OS verwendet). Abbildung 2.1 zeigt einen Fortschrittsindikator des Betriebssystems Mac OS. Dieser erscheint, sobald mehrere Dateien kopiert werden sowie das Laden jener mehr als ein paar Sekunden dauert.

Animationen, die in die Benutzerschnittstelle als Teil der Funktionalität eingebunden sind, werden funktionale Animation genannt. Sie unterstützen den Benutzer beim Interagieren indem sie eine humane, erklärende Komponente darstellen. Es existieren einige Kerneigenschaften, die bedacht werden müssen, wenn Animation nicht nur als dekorative Bewegung ohne Nutzen hinzugefügt werden, sondern als funktionale Animation dienen soll.

2.2.1 Natürliche, familiäre Bewegungen

Damit Bewegungen realistisch erscheinen müssen sie einen familiären Charakter haben. Dies erfordert ein gewisses Level an Realismus der Animation um deren Bewegungsart ähnlich jener zu gestalten, wie sie vom Gehirn erwartet wird [24]. Unerwartete Übergänge sollten stets vermieden werden, Objekte steigern im wirklichen Leben ihre Geschwindigkeit am Beginn der Bewegung und werden langsamer bevor sie wieder in den Stillstand übergehen.



Abbildung 2.2: Visueller Hinweis aus dem iOS-Spiel *Pudding Monsters* [29].

2.2.2 Visuelle Hinweise

Visuelle Hinweise sind besonders für Personen von großem Nutzen, die zum ersten Mal mit einer Schnittstelle konfrontiert werden. Ein Weg zum Erlernen möglicher Interaktionen in einem Tool kann mittels Animationen erzielt werden. Dieses Mittel steigert die Aufmerksamkeit des Nutzers und vermeidet Verwirrungen und Frustration [9]. Abbildung 2.2 zeigt eine Möglichkeit eines visuellen Hinweises. Diese Animation ist aus dem iOS-Spiel *Pudding Monsters*. Es wird dem Spieler visuell erklärt, wie er fortfahren muss.

2.2.3 Visuelle Rückmeldungen

In effektiven, benutzerorientierten Schnittstellen ist es sehr wichtig, dass ein Benutzer auf jede Aktion seinerseits eine visuelle Rückmeldung bekommt. Dies vermittelt sowohl das Gefühl, die Kontrolle über das System zu haben, als auch, dass das System die Aktion registriert hat und sie weiter verarbeiten kann.

Elemente wie Buttons, Formularfelder und Kontrollfelder sollen daher immer ein subtiles, visuelles Feedback geben. Bei Formularfeldern hat die Rückmeldung zudem den Vorteil, dass sie auch zur Validierung eingesetzt werden kann. Es ist sowohl eine Bestätigung als auch eine Ablehnung des Inputs visuell umsetzbar [9].

Besonders wertvoll ist die Art der funktionalen Animation bei Touch-Geräten wie Handys und Tablets, da hier Rollover-Effekte nicht eingesetzt werden können.

Das Gebiet der visuellen Rückmeldungen wurde nach der Pilotstudie des praktischen Teils der Arbeit noch einmal essentiell, da sich in der Testsituation andere Gegebenheiten boten als bei der Programmierung dessen. Wie auf diese Erkenntnis reagiert wurde ist in Kapitel 5 unter Abschnitt 5.1

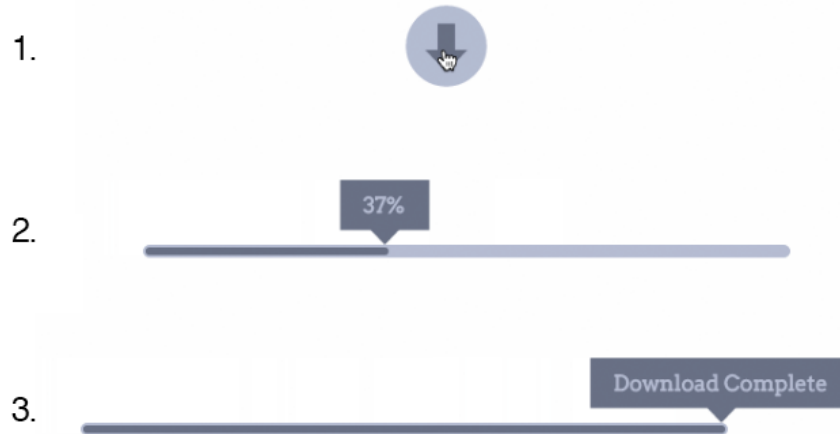


Abbildung 2.3: Ladeanimation, die dem Benutzer zeigt, was gerade im System passiert [10].

nachzulesen.

2.2.4 Übergänge und Systemstatus

Wenn eine Änderung in einer Schnittstelle oder auf einer Webseite gemacht und diese nicht visuell verfolgt werden kann, ist es oft schwer für Benutzer die kausale Verbindung vom alten zum neuen Stand nachzuvollziehen. Um eine zufriedenstellende Benutzererfahrung zu garantieren, ist es daher wichtig, dass der Nutzer durchgehend den Eindruck hat, dass er versteht was das System gerade macht. Es sollte ihm zudem ermöglicht werden, eine Vorstellung davon zu haben, wie der Bildschirm nach einer Aktion aussehen wird. Außerdem muss deutlich gemacht werden, wann eine Aktion beginnt, wie lange sie dauert und wann diese endet. Das Ausführen der Aktion sollte ebenso in einer zufriedenstellenden Zeitspanne geschehen [1, 11].

Abbildung 2.3 zeigt ein Beispiel eines sehr raffinierten Ladebalkens. Er lenkt einerseits den Benutzer durch die Animation ab und verkürzt so auf spielerische Art und Weise die gefühlte Wartezeit für ihn. Andererseits ist ebenso ersichtlich, was zum gegebenen Zeitpunkt im System passiert.

2.3 Cartoon-Animationen

Chang und Ungar (1993) diskutieren in ihrem Paper die Anwendung von Cartoon-Animationen in Benutzerschnittstellen. Sie heben hervor, dass herkömmliche Interfaces meist auf statische Präsentationen aufbauen und dies zu plötzlichen, unerwarteten Änderungen führt, die der Benutzer nicht nachvollziehen kann und wodurch er meist von seinem eigentlichen Vorhaben abgelenkt wird. Cartoon-Animationen hingegen schaffen es, dem Publikum erfolgreich jegliche Bewegungsabfolge zu vermitteln und es zugleich damit zu begeistern. Auch wenn es sich bei Cartoons nur um passive Unterhaltung handelt und Benutzerschnittstellen seriöse, interaktive Tools zum effektiven Arbeiten sind, können Cartoon-Animationen im Web einen kognitiv hohen Nutzen haben.

So treten Cartoon-Objekte niemals überraschend auf oder ändern ohne visueller Verfolgung ihre Position, Form und Größe. Sie ändern ihr Erscheinungsbild und ihren Standort solide und kontinuierlich, wachsen langsam an oder deformieren sich. Auch für Benutzerschnittstellen ist es empfehlenswert, diese Art von Animation zu übernehmen und sich für weiche Übergänge und der Interpretation von Änderungen zu entscheiden. Kleine Objekte wachsen dabei langsam an oder bewegen sich langsam von der Startposition zur Zielposition. Ein weiteres Beispiel wäre ein Menü, das sich gleichmäßig von einem geklickten Button zu einem vollen Benutzermenü umwandelt [1, S. 45 ff.].

Im Falle des praktischen Teiles dieser Arbeit wären Cartoon-Animationen für den einseitigen Editor zu umfangreich, daher wurden Fade-Effekte als Interaktionsmittel gewählt. Für umfangreichere Webseiten sind Cartoon-Animationen aber in jedem Fall sehr empfehlenswert.

2.4 Subtile Hinweise zur Verbesserung der Benutzererfahrung

Selbstbeschreibungsfähig werden laut ISO 9241-10 Schnittstellen und Systeme genannt, die *unmittelbar verständlich* sind [3, S. 6]. Darunter fallen allerdings nur die wenigsten Computersysteme. Daher sollten den Nutzern Hilfestellungen angeboten werden, die zum Beispiel in folgenden Formen auftreten können:

- gedruckte Beschreibungen und Handbücher zum Nachlesen,
- Einschulung oder Training durch qualifiziertes Fachpersonal,
- interaktive Hinweise und Hilfestellungen durch die Schnittstelle selbst.

Dieses Kapitel behandelt die dritte Art, die interaktive Unterstützung des Systems, worauf auch die vorliegende Arbeit aufbaut. Herzeg bezeichnet diese Art als *Hilfesysteme*. Diese sogenannten *Hilfesysteme* unterteilt er auf



Abbildung 2.4: Aktiver Hilfeassistent in *Microsoft Word* [18].

weitere Gesichtspunkte.

2.4.1 Aktive und passive Hilfesysteme

Zum einen unterscheidet Herczeg zwischen *aktiven* und *passiven Hilfesystemen*.

Aktive Hilfesysteme mobilisieren sich selbstständig und schalten sich ein, wann immer eine zusätzliche Abfrage nötig ist, der Benutzer Hilfe benötigen könnte oder einen Fehler macht. Dafür muss das System die Interaktion zwischen Nutzer und Schnittstelle überwachen. Abbildung 2.4 zeigt ein Beispiel für einen aktiven Hilfeassistenten, wobei *Microsoft Word* nach einer bestimmten Benutzeraktion die Hilfe zum Erstellen eines Briefes anbietet. Der Assistent erscheint dabei ohne, dass der Benutzer ihn aktiv aufruft. Zur Realisierung *aktiver Hilfen* bedarf es meist eines sehr hohen Implementierungsaufwandes und der genauen Beobachtung des Benutzerverhaltens [4, S. 153 ff.].

Im Gegensatz dazu gestaltet sich die Umsetzung *passiver Hilfesysteme* wesentlich einfacher. Sie müssen vom Benutzer direkt aktiviert werden und erscheinen meist in Form eines Benutzerhandbuchs, weshalb auch oft der Name *Online-Manual* verwendet wird. Bei der Darstellung dieser Art der Hilfestellung ist es wichtig auf folgende Punkte zu achten:

- Der Text soll so kurz und einfach wie möglich in einer dem Benutzer verständlichen Sprache formuliert werden.
- Er soll sachlich sein.
- Es ist darauf zu achten, gängige Fachbegriffe in den Hilfetexten zu verwenden.
- Eine gute Gliederung und das Hervorheben wichtiger Textteile fördern die Übersichtlichkeit.

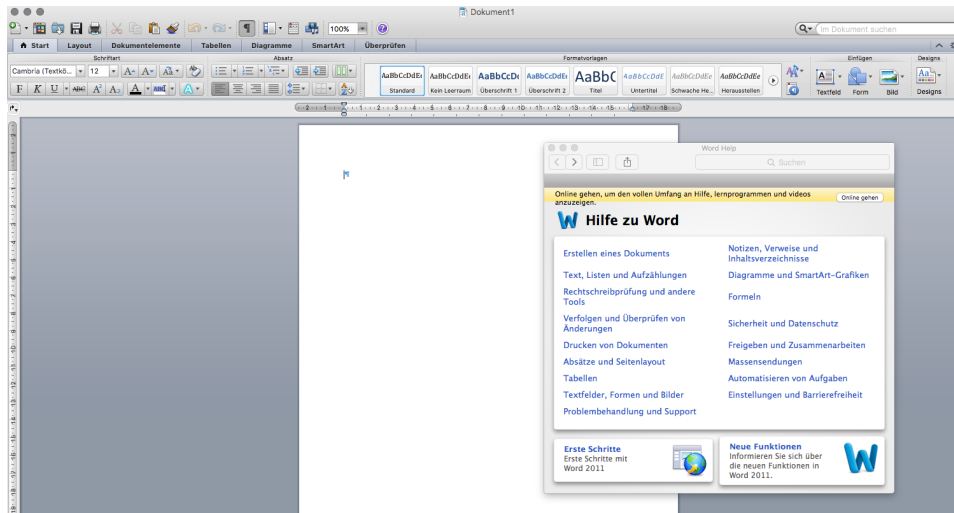


Abbildung 2.5: Passives Hilfesystem von *Microsoft Word*.

- Bilder und Icons können die Verständlichkeit des bloßen Textes erhöhen.

Abbildung 2.5 zeigt die Hilfe in *Microsoft Office*, die nach Klicken des Fragezeichens in der Menüleiste erscheint.

Unglücklicherweise bringen *Online-Manuals* und textuelle Beschreibungen den Nachteil mit sich, dass längere Texte oft nur schwer auf einem Bildschirm zu lesen sind und sich für manche Nutzer der Aufwand, die richtige Erklärung zum gesuchten Problem zu finden, nicht rentiert. Weiters besteht die Möglichkeit, dass bei textuellen Beschreibungen schnell sprachliche Probleme auftreten können, was wiederum die Hilfestellung nicht universell einsetzbar macht. Stichwortsuchen, die meist in *Online-Manuals* inkludiert werden, liefern des Öfteren ebenfalls nicht das gewünschte Ergebnis. Es existieren eine Vielzahl an Möglichkeiten eine Suche zu formulieren, womöglich ist dem Nutzer der Fachbegriff auch nicht geläufig. Dennoch bieten *passive Hilfen* eine schnell realisierbare und oft sehr hilfreiche Informationsquelle für Benutzer eines Systems an [4, S. 153 ff.].

Im praktischen Teil dieser Arbeit werden sowohl aktive als auch passive Hilfesysteme verwendet. Zu den verwendeten, aktiven Hilfesystemen zählen beispielsweise Tooltips, erscheinende Platzhalter oder die unter Abschnitt 3.1.4 beschriebene Hover-Schaltfläche. Ein eingesetztes passives Hilfesystem bildet der Button zum Anzeigen der Modulbeschreibung, der ebenfalls auf der aktiven Hover-Schaltfläche zu finden ist.

2.4.2 Statische und dynamische Hilfesysteme

Weiters teilt Herzeg Hilfestellungen in statische und dynamische Systeme ein.

Als *statische Hilfesysteme* bezeichnet man Hilfestellungen die sich im Laufe der Zeit nicht verändern und immer wieder die gleiche Antwort liefern. Dazu zählen zum Beispiel Handbücher, oftmals als *Online-Manuals* bezeichnet. Aufgrund der Tatsache, dass statische Hilfe häufig zu viele beziehungsweise unpassende Informationen liefert kann es vorkommen, dass der Benutzer zwar über die Hilfsmöglichkeit bescheid weiß, diese jedoch nicht nutzt, weil der Aufwand für ihn zu groß wäre. Um dem entgegen zu wirken kann eine geeignete Zugriffsmethode auf Informationen, beispielsweise durch eine umfangreiche Stichwortsuche, hilfreich sein. Außerdem eignet sich statische Hilfe gut für die Einführung sowie Erklärung der Gesamtstruktur und Basisfunktionalität eines Interfaces [4, S. 153 ff.].

Dynamische, auch *kontextabhängig* genannte, Hilfesysteme verändern sich gemäß des Abfragezeitpunktes. Je nachdem, welche Aktion der Benutzer gerade ausführt, unterstützt das System rationell mit seiner adäquaten Hilfe für genau jene Interaktion. So wird dem Nutzer das Suchen der passenden Lösung in einer großen Wissensbasis erspart und er kann effektiv ohne jegliches Abschweifen sein Ziel verfolgen. Allerdings ist, wie bei *aktiven Hilfesystemen*, auch bei *dynamischen Hilfesystemen* ein großer Mehraufwand für die Umsetzung der Schnittstelle erforderlich [4, S. 153 ff.].

Zur Umsetzung des Web-Modul-Editors wurden sowohl statische als auch dynamische Hilfesysteme verwendet. Das umfangreichste, dynamische Hilfesystem, das implementiert wurde, ist die Generierung der Platzhalter. Zu den eingesetzten, statischen Hilfesystemen zählt unter anderem der Pfeil für den Hinweis auf den Drag & Drop Mechanismus, beschrieben unter Abschnitt 3.1.5.

2.4.3 Einheitliche und individuelle Hilfesysteme

Aufgrund der Individualität einer Hilfe unterteilt Herzeg Hilfe auch in *einheitliche* und *individuelle Hilfe*.

Einheitliche Hilfe bedeutet, dass jeder Benutzer die gleiche Hilfestellung zur gleichen Aktion erhält. Das heißt, dass das System über keinerlei Informationen über den Anwender verfügt und so nicht weiß, ob der Benutzer zur aktuellen Situation schon einmal Hilfe bekommen hat, wodurch diese für ihn vielleicht überflüssig ist [4, S. 153 ff.].

Individuelle Hilfesysteme hingegen wissen über den Benutzer bescheid und können sein Bedürfnis nach gewissen Hilfestellungen besser abschätzen. Diese Benutzerinformationen können entweder direkt von der Person abgefragt oder auch vom System selbst ermittelt werden. Hierbei kann der Benutzer möglicherweise auch den Umfang der gewünschten Hilfe oder die

favorisierte Darstellung jener angeben [4, S. 153 ff.].

Beim praktischen Teil dieser Arbeit wurde bislang nur auf einheitliche Hilfesysteme eingegangen. Eine mögliche Erweiterung oder Fortführung des Projektes könnte jedoch auch zu individuellen Hilfestellungen führen.

2.4.4 Synchron und Asynchrone Hilfesysteme

Als letzten Punkt zur Einteilung von Hilfesystemen unterscheidet Herczeg zwischen *synchronen* und *asynchronen Hilfesystemen*.

Meist tritt Hilfe *synchron* auf, also gleich bei der Ausführung einer Aktion, da der Benutzer diese auch zum Zeitpunkt der Umsetzung benötigt. Vor allem bei *dynamischer Hilfe* ist die Verwendung von *synchronen Hilfesystemen* unumgänglich, da der Nutzer ansonsten keinen Zusammenhang erkennen kann [4, S. 153 ff.].

Hin und wieder kann jedoch auch *asynchrone Hilfe* nützlich sein. Sind Hilfestellungen weniger wichtig oder können diese auch hinausgezögert werden bis ein Benutzer seine Aktion fertig abgeschlossen hat, so macht es auch Sinn diese erst später aufzurufen. Dadurch wird vermieden, dass der Nutzer durch gerade unwesentliche Hinweise abgelenkt wird und sein Vorhaben nicht vollständig ausführen kann [4, S. 153 ff.].

Eine Anwendungsmöglichkeit für *asynchrone Hilfe* zeigt Abbildung 2.6. Es wird ein Kontaktformular dargestellt, das erst beim Klick auf den Button „Senden“ evaluiert wird. Die Warnhinweise werden danach rot dargestellt und helfen dem Benutzer das Formular valide auszufüllen um ein Absenden zu ermöglichen.

Im Zuge des praktischen Teiles dieser Arbeit fanden bislang nur synchrone Hilfestellungen ihre Anwendung. Beispiele daraus sind die Platzhalter sowie die farbliche Kennzeichnung beim möglichen Platzieren von neuen Elementen oder auch der prompte Warnhinweis sobald die verfügbare Workshopzeit überschritten wird.

2.5 Beispiele von subtilen Hinweisen

Im weiteren Abschnitt werden einige Beispiele und verwandte Arbeiten von subtilen Hinweisen vorgestellt. Diese werden zum Teil gänzlich oder in abgewandelter Form im praktischen Teil der Arbeit implementiert.

2.5.1 Textuelle Hinweise

Die einfachste Art von Hinweisen ist wohl der *textuelle Hinweis*. Sie sind sehr rasch und ohne großem Aufwand umzusetzen. In manchen Anwendungsbereichen sind sie effektiv, *textuelle Hinweise* bringen jedoch auch einige Nachteile mit sich. Zum einen kann es mögliche sprachliche Barrieren geben, sei

CONTACT ME

Ich freue mich immer über Fragen und Anregungen

The image shows a contact form with the following elements:

- A text input field containing "Max Muster".
- An email input field containing "max@gmx" with a red error message below it: "Not a valid email address".
- A telephone input field containing "Telefon *" with a red error message below it: "Bitte geben Sie ihre Telefonnummer ein.".
- A large text area for a message containing "Nachricht *" with a red error message below it: "Bitte geben Sie eine Nachricht ein.".
- A dark grey button labeled "SENDEN" at the bottom center.

Abbildung 2.6: Beispiel für eine *asynchrone Hilfe* [15].

es, dass der Benutzer einem anderen Land entstammt und eine andere Sprache spricht, oder dieser womöglich auch gar nicht richtig und sinnerfassend lesen kann. Zum anderen würden für textuelle Hinweise oft viel zu lange Texte nötig sein, wofür sich Nutzer nicht die erforderliche Zeit nehmen jene zu lesen und umzusetzen. Ab und an kann eine textuelle Beschreibung beziehungsweise Hilfe für den Benutzer aber dennoch sehr sinnvoll sein.

Abbildung 2.7 zeigt textuelle Hinweise aus dem Programm *Microsoft PowerPoint*. Dem Benutzer wird dabei erklärt, wo und wie er den Titel und Untertitel seiner Folie hinzufügen kann.

Preim und Dachselts empfehlen eine Kombination aus Bild und Text. Es eignen sich zum Beispiel ein Icon und eine kurze Textuelle Beschreibung sehr gut um besser im Gedächtnis zu bleiben. Dies ist vor allem für Menüleisten eine sehr gängige und gute Alternative zu textlastigen Menüs. Abbildung 2.8 zeigt einen Ausschnitt aus dem kombinierten Menü des Programmes *Microsoft Word* [7, S. 210 f.].

Im praktischen Teil der Arbeit wurde versucht, textuelle Hinweise weitgehend zu vermeiden. Bei Benutzerabfragen, ob Elemente beispielsweise wirklich verschoben oder ersetzt werden sollten, sind diese jedoch unumgänglich und auch sinnvoll.

2.5.2 Gitternetzlinien

Gitternetzlinien werden visuell als feine Striche auf einem Arbeitsblatt dargestellt und ermöglichen dem Benutzer neben dem Unterteilen seines Layouts die genaue Ausrichtung von Elementen. Mittlerweile gibt es zahlreiche Programme, die die Verwendung von *Gitternetzlinien* zulassen. Sie können je nach Präferenz des Benutzers ein- oder ausgeblendet werden. Je nach An-

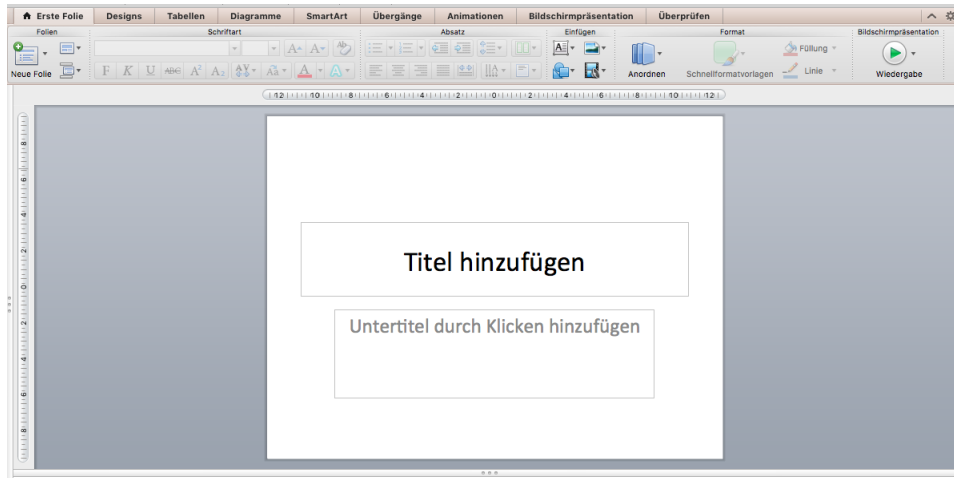


Abbildung 2.7: Textueller Hinweis für den Benutzer aus dem Programm *Microsoft PowerPoint*.

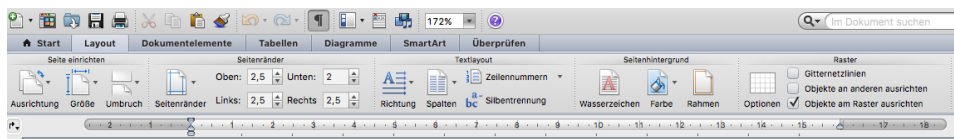


Abbildung 2.8: Kombiniertes Menü durch Text und Bild aus dem Programm *Microsoft Word*.

wendungssoftware können *Gitternetzlinien* teilweise auch selbst angepasst werden. Abbildung 2.9 zeigt die Verwendung von *Gitternetzlinien* im Programm *Microsoft Word*.

Da es im praktischen Teil der Arbeit für die Platzierung der Objekte Platzhalter gibt, die unter Abschnitt 3.1.2 genauer beschrieben sind, kamen keine Gitternetzlinien zum Einsatz. Diese wären jedoch auch eine Lösungsmöglichkeit zur konkreten Platzierung der Module gewesen.

2.5.3 Hilfslinien, Smart Guides und Snapping

Weitere Beispiele für subtile Hinweise sind Smart Guides, Hilfslinien und Snapping. Diese drei Funktionen treten meist in Kombination auf und helfen dem Benutzer Elemente pixelgenau auszurichten. Daher eignen sich die Hinweise besonders gut für Layout-Programme wie etwa *Adobe Illustrator*. Abbildung 2.10 zeigt ein Beispiel in dem Hilfslinien und Smart Guides zum Einsatz kommen.

Die rot gestrichelten Linien sind Hilfslinien und können selbst vom Be-

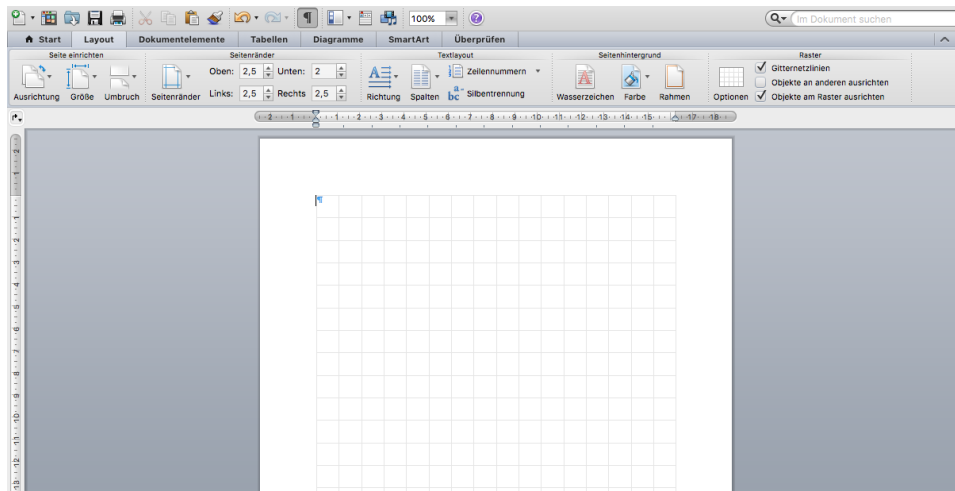


Abbildung 2.9: Gitternetzlinien als Hinweis für den Benutzer aus dem Programm *Microsoft Word*.

nutzer an die gewünschte Position gezogen werden. Dies ermöglicht das simple Einteilen der Arbeitsfläche, anhand dessen die Elemente dann ausgerichtet werden.

Ausgewählte Elemente werden vom System mit acht Angriffspunkten versehen, die die Verwendung von Smart Guides ermöglichen. Zieht der Benutzer ein Element in die Nähe von zuvor gesetzten Hilfslinien oder anderen, bereits platzierten, Elementen, so erscheinen grüne Linien. Dies sind die sogenannten Smart Guides, die nun das genaue Ausrichten der aktiven Elemente zulassen.

Die dritte Art der Hilfestellung ist Snapping, das ebenfalls zur präzisen Platzierung dient und sowohl beim Erscheinen von Smart Guides als auch bei selbst platzierten Hilfslinien wirkt. Dadurch, dass das aktive Element an die geeignete Position gezogen wird, fällt es dem Benutzer leichter es rasch zu platzieren [23].

Eine Abwandlung dieser Arten von subtilen Hinweisen fanden im praktischen Teil der Arbeit ihre Anwendung. Dem Benutzer wird das Positionieren von Elementen erleichtert, indem an validen Stellen Platzhalterobjekte generiert werden, für die ein Snapping-Mechanismus implementiert wurde. Die konkrete Beschreibung dazu ist unter Abschnitt 3.1.2 zu finden.

2.5.4 Farbwechsel

Um dem Benutzer, wie unter Abschnitt 2.2.3 erläutert, eine unmittelbare, visuelle Rückmeldung zu geben, eignen sich beispielsweise Farbwechsel sehr gut. Ein Hover-Effekt mit unterschiedlichsten Farben ist beispielsweise eine

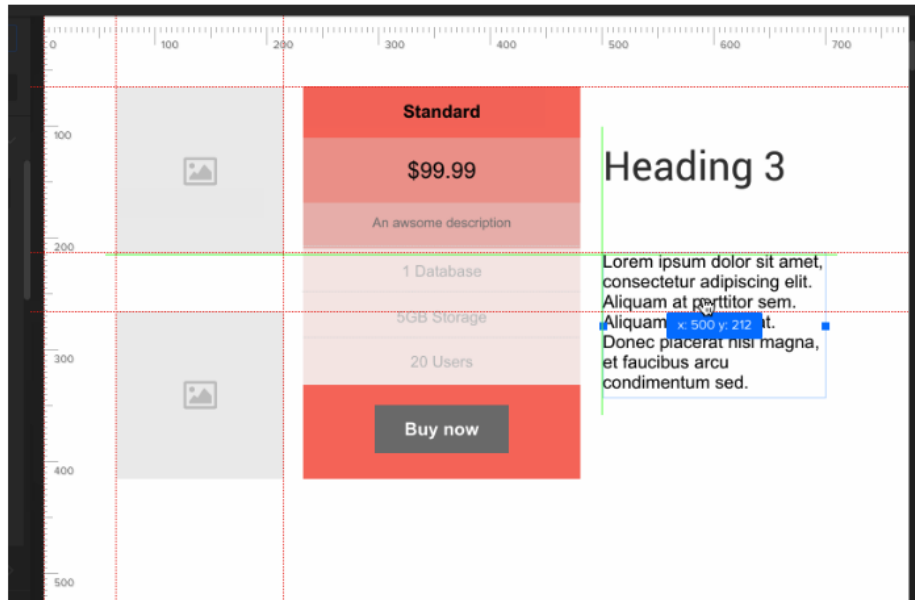


Abbildung 2.10: Beispiel für Smart Guides (grüne Linien) und Einsatz von eigenen Hilfslinien durch den Benutzer (rote Linien) [23].

der einfachsten aber effektivsten Methoden um dem Nutzer zu zeigen, dass sich der Mauszeiger auf einem Link befindet. Außerdem werden Farben häufig sowohl für valide als auch invalide Ausdrücke verwendet. In diesem Fall sind die gewählten Farben meist jenen einer Ampel angenähert. Rot bedeutet somit „Nein“, Grün bedeutet „Ja“. Leider bringt dieser Hinweis aber den Nachteil mit sich, dass für Personen mit einer Rot-Grün-Schwäche eine neue Problematik entsteht und die farbliche Hilfestellung dadurch nutzlos wird.

Auch der Farbwechsel wurde im Web-Modul-Editor als zusätzlichen Hinweis eingebaut. Das genaue Einsatzgebiet ist unter Abschnitt 3.1.2 einzusehen.

2.5.5 Erweitertes Drag & Drop

Collomb, Hascoët, Baudisch und Lee erweiterten die Drag & Drop Technologie auf Wand-Displays. Dabei vereinen sie zum einen die Funktion der Erweiterung der Reichweite des Benutzers aufgrund eines Miniatur-Bildschirmes (*push-and-throw*) und zum anderen die Funktion, dass mögliche Ziele von Drag & Drop zum Benutzer gebracht werden (*drag-and-pop*). Die neue Version nennen die Begründer *push-and-pop* [2, S. 25 ff.]. Abbildung 2.11 zeigt die *push-and-throw* Interaktions-Technik. Der Benutzer möchte das Icon der Bilddatei in den Ordner „My Pictures“ verschieben.

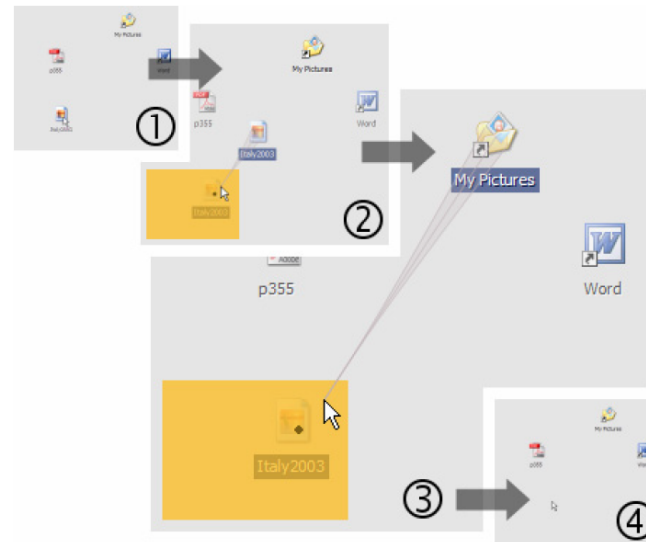


Abbildung 2.11: *Push-and-throw* Technologie für Wand-Displays [2, S. 26].

Sobald die zu verschiebende Datei mit dem Stift ausgewählt wird, erscheint eine Art Miniatur-Bildschirm (hier gelb dargestellt). Außerdem erscheint die Datei als transparentes Icon, das auch auf dem richtigen Bildschirm zu sehen und um die Distanz, die am Miniatur-Bildschirm zurückgelegt wurde, verschoben ist. Dies zeigt eine Vorschau, wo sich das Icon gerade auf dem Wand-Display befinden würde. Das Icon selbst ist mit einer Art Gummiband mit dem Cursor verbunden.

Wird die Bilddatei nun weiter auf dem Miniatur-Bildschirm in Richtung Zielordner verschoben und dieser schlussendlich erreicht, so wird der Ordner auf dem richtigen Bildschirm markiert. Setzt der Benutzer den Stift nun ab, so wird das Objekt tatsächlich verschoben [2, S. 26 f.]. Abbildung 2.12 zeigt die *drag-and-pop* Interaktions-Technik. Der Benutzer möchte die HTML-Datei mittels Drag & Drop in den Papierkorb verschieben.

Bei dieser Technik wird das aktuelle Objekt nicht zum Zielobjekt geschickt, sondern die möglichen Zielicons nähern sich dem aktuellen Objekt an. Sobald der Benutzer also mit dem Ziehen des aktuellen Icons beginnt, werden alle möglichen Zielobjekte, die in der begonnenen Zielrichtung liegen, zum Icon bewegt. Auch hier werden die Zielobjekte mit der eigentlichen Position am wirklichen Bildschirm mittels Gummiband verbunden, sodass der Benutzer weiß woher diese Icons entstammen. Lässt der Nutzer nun die HTML-Datei über dem Papierkorb los, so wird diese gelöscht und alle anderen möglichen Icons zum Ablegen der aktuellen Datei verschwinden.

Der Nachteil von *drag-and-pop* ist, dass aufgrund der Übersichtlichkeit

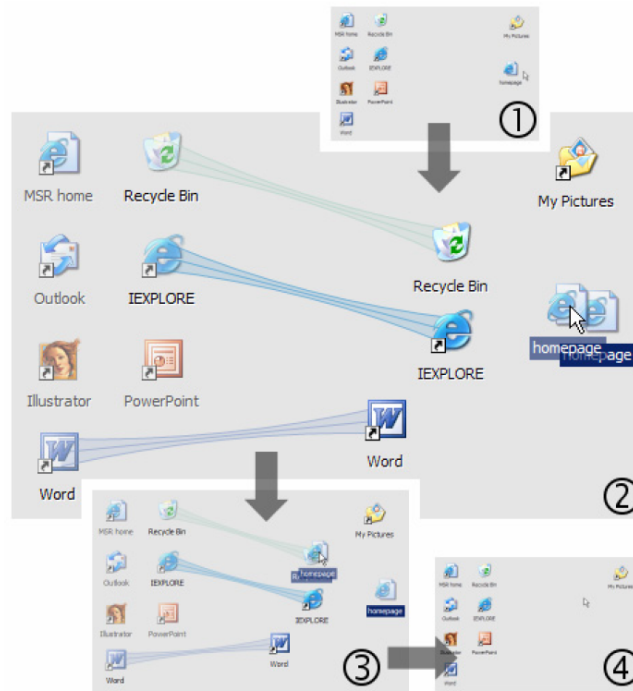


Abbildung 2.12: *Drag-and-pop* Technologie für Wand-Displays [2, S. 27].

nur die validen Icons aufgerufen werden, die in der Richtung des begonnenen Drag & Drops liegen. So wird die Anzahl erheblich eingeschränkt und es kommt zu keinen Überlappungen. Da die Begrenzung aber teilweise recht ungenau ist, kann zum Beispiel eine bogenförmige Bewegung dazu führen, dass das gewünschte Zielicon nicht unter den vorgeschlagenen Icons erscheint [2, S. 27 f.].

Die Version von Collomb, Hascoët, Baudisch und Lee vereint nun die Vorteile der zuvor erklärten Interaktions-Techniken zu *push-and-pop*, die Abbildung 2.13 zeigt. Der Benutzer möchte das Icon der HTML-Datei in den Papierkorb werfen.

Sobald er damit beginnt die HTML-Datei zu ziehen, erscheint wie bei *push-and-throw* ein Miniatur-Bildschirm (hier transparent grün dargestellt). Außerdem werden wie bei *drag-and-pop* die möglichen Zielicons in diesem Miniatur-Bildschirm angezeigt. Zieht der Benutzer nun das aktive Icon über den Papierkorb im Miniatur-Bildschirm, so wird es sowohl dort (mittels feinem Rahmen), als auch auf dem eigentlichen Bildschirm hervorgehoben (blau hinterlegt). Nach dem Loslassen des aktuellen Icons wird im gezogenen Fall die HTML-Datei gelöscht und sowohl der Miniatur-Bildschirm als auch die anderen möglichen Zielicons ausgeblendet [2, S. 28 f.].

Inspiziert von der erweiterten Drag & Drop-Technologie von Collomb,

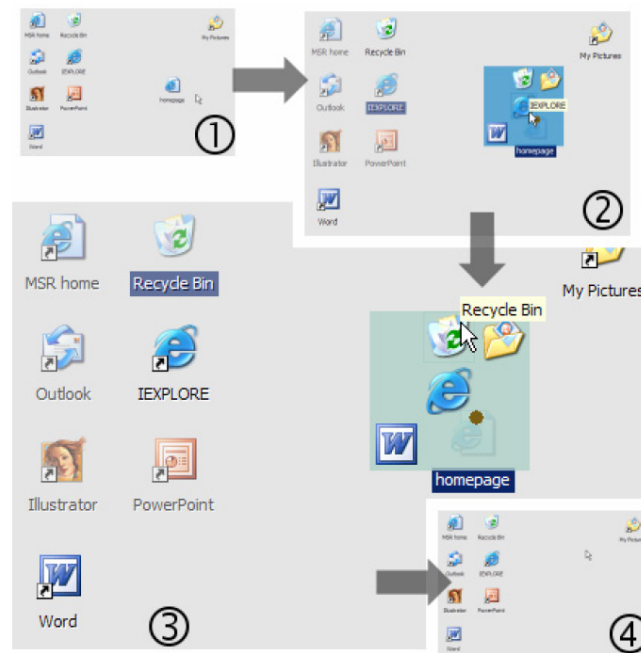


Abbildung 2.13: *Push-and-pop* Technologie für Wand-Displays von Col-lomb, Hascoët, Baudisch und Lee [2, S. 29].

Hascoët, Baudisch und Lee wurde im praktischen Teil der Arbeit eine Hover-Schaltfläche eingebaut, die die möglichen Funktionen der platzierten Objekte direkt beim Modul erscheinen lässt. Die genaue Umsetzung ist unter Abschnitt 3.1.4 zu finden.

Kapitel 3

Konzeption

Zur Veranschaulichung der Benutzerführung anhand der Verwendung subtiler Hinweise wurde ein Web-Modul-Editor erstellt, der die Möglichkeit bietet, einen Workshop oder ein Seminar zu planen. Das folgende Kapitel stellt eine Zusammenfassung der Überlegungen und Konzepte, die bei der Umsetzung berücksichtigt wurden, dar.

3.1 Basisfunktionalität und Anforderungen

Grundidee ist es, den Modul-Editor intuitiv zu gestalten, um ihn ohne Vorwissen verwendbar und einfach, durch den praktischen Einsatz, erlernbar zu machen. Aufgrund der unter Abschnitt 2.1 angeführten Tatsachen darf es nicht erforderlich sein, vor der Verwendung ein Benutzerhandbuch lesen zu müssen oder eine Einführung in das Tool von einem bereits erfahrenen Nutzer zu benötigen.

3.1.1 Modulares System

Der Editor baut auf ein modulares System auf, das vom Benutzer selbst (in diesem Fall dem Moderator des Workshops) zusammengesetzt werden kann. Tabelle 3.1 zeigt die möglichen Modultypen und deren Farben, aus denen der Benutzer die Module für seine Komposition wählen kann. Die zur Verfügung stehenden Module werden zu einem Werkzeugkasten, unterteilt in die Modultypen, zusammengefasst und durch einen Drag & Drop Mechanismus dem Benutzer zugänglich und platzierbar gemacht.

Damit ein schneller Einstiegspunkt für den Nutzer gegeben ist, ist das Design des Editors schlicht gehalten. Nach initialem Laden der verfügbaren Module sind lediglich ein Platzhalterobjekt und die Toolbox mit den Modulen ersichtlich. Der Platzhalter signalisiert dem Benutzer, dass an dieser ein Stelle Modul platziert werden kann.

Findet für einen definierten Zeitraum keine Interaktion seitens des An-

Tabelle 3.1: Fünf unterschiedliche Modultypen des Web-Modul-Editors in die die einzelnen Module eingeteilt werden.

<i>Modul-Typ</i>	<i>Farbe</i>	<i>Beschreibung</i>
Ice Breaker	Lila	Kleine Übungen und Spiele zu Beginn des Workshops, damit sich die Teilnehmer gegenseitig kennenlernen.
Presentation	Blau	Verschiedene Präsentationsmodule wie zum Beispiel Slideshows oder externe Geräte zur Präsentation.
Think & Create	Grün	Kreative Umsetzungsmodule in die die Teilnehmer selbst aktiv werden und Ideen generieren. Beispiele sind Brainstroming, Erstellung von Benutzerprofilen, etc.
Decide	Orange	Entscheidungs-Module, die zum Aussortieren von Ideen dienen. Beispiele sind Voting, Zusammenfassungen, etc.
Break	Grau	Pausen-Modul (sowohl für kurze Zwischen- als auch Mittagspausen).

wenders statt, so ist laut Diskussionsentwurf des *National Institute of Standards and Technology* eine Notifikation an den Benutzer erforderlich [20]. Daher ist für den Editor geplant, dass für das erste Modul in der Toolbox ein Blinken einsetzt, damit der Benutzer weiß, dass eine weitere Aktion mit den Modulen durchzuführen ist. Die Zeitspanne für das Inkrafttreten des Highlightings wird mit sieben Sekunden gewählt. Dies beruht auf der Tatsache, dass laut Nielsen Benutzer zehn Sekunden damit verbringen sich auf einer Webseite zurecht zu finden [17]. Kommen diese nach jener Zeit nicht zum gewünschten Ergebnis, so verlassen sie die Seite ohne Erfolg und werden auch nicht mehr zurückkehren. Sieben Sekunden erscheinen daher als angemessene Zeit, um dem Benutzer die Möglichkeit zu bieten sich selbst einen Überblick zu verschaffen und gegebenenfalls in adäquater Zeit eine Hilfestellung zu bekommen [17, 20].

Jedes Modul verfügt über folgende Parameter:

- Titel,
- Beschreibung,
- Typ,
- Arbeitsmodus, welcher eingeteilt wird in
 - Plenum,
 - Gruppe oder
 - Individuell,
- Dauer,
- Eingangsverbindung und
- Ausgangsverbindung.

Die Parameter *Arbeitsmodus* und *Dauer* sind mit einem Initialwert versehen, können bei Bedarf vom Benutzer aber individuell angepasst werden.

3.1.2 Validitätsprüfung

Da nur eine valide Reihung von Modulen ermöglicht werden soll, ist bereits beim Platzieren der Module eine Prüfung der Ein- und Ausgangsverbindungen erforderlich. Diese Validität muss dem Benutzer zum Zeitpunkt des Platzierens visuell durch subtile Hinweise vermittelt werden.

Im Falle des Modul-Editors treten die Hinweise in Form eines Farbwechsels, wie unter Abschnitt 2.5.4 beschrieben, auf. Außerdem existiert, vor allem für von Rot-Grün-Schwäche betroffene Personen, die Implementierung einer Verschiebung der bereits platzierten, benachbarten Module, wodurch ein Positionieren des neuen Modules ermöglicht oder verwehrt wird. An jeder Stelle, an der das Platzieren des aktuellen Elementes erlaubt ist, erscheint ein Platzhalterelement, das gezogene Element ist von einem grünen Rahmen umgeben. Ist das Einfügen nicht erlaubt, so rücken die benachbarten Elemente zusammen, es erscheint kein Platzhalter, das aktuelle Element ist von einem roten Rahmen umfasst.

Eine Validitätsprüfung ist nicht nur beim Einfügen neuer Module, sondern auch beim Ersetzen sowie beim Löschen bereits platzierter Module erforderlich. Ist ein Ersetzen aufgrund der Ein- und Ausgangsverbindungen der zukünftigen, benachbarten Elemente nicht möglich, so wird das aktuelle Element rot umrahmt, sowie ein Platzieren verweigert. Wird ein Element gelöscht und die danach verbleibenden, benachbarten Module sind nicht kompatibel, ist das Löschen des Modules zwar möglich, es wird ein Platzhalterobjekt anstelle des gelöschten Modules eingefügt.

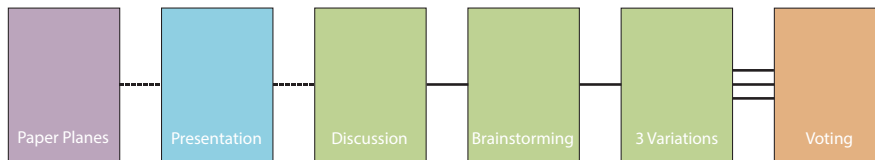


Abbildung 3.1: Verbindungstypen zwischen den Modulen.

3.1.3 Platzieren der Module

Um dem Benutzer das Platzieren der Module so einfach wie möglich zu gestalten, verfügt der Drag & Drop Mechanismus über eine Snapping-Funktion. Wie unter Abschnitt 2.5.3 erläutert, ermöglicht dieser subtile Hinweis ein präzises Einfügen der Module, ohne dass sich der Benutzer mit der exakten Ausrichtung des Elementes beschäftigen muss.

Sobald mindestens zwei Module auf der Arbeitsfläche platziert wurden, werden diese mit Linien verbunden. Für die Verbindung wurden folgende drei Typen festgelegt:

- eine strichlierte Linie,
- eine solide Linie und
- drei solide Linien.

Der Linientyp richtet sich nach den Ein- und Ausgangsverbindungen benachbarten Module.

Eine strichlierte Linie bedeutet, dass zwischen den Modulen kein Informationsaustausch stattfindet. Dies ist zum Beispiel bei einem *Ice-Breaker-Modul* und einer Präsentation der Fall. Das *Ice-Breaker-Modul* hat eine Ausgangsverbindung von 0, das Präsentationsmodul hat eine Eingangsverbindung von 0. Somit werden keine Informationen übergeben und die Beiden werden mittels strichlierter Linie verbunden.

Eine solide Linie erscheint meist bei Ideenfindungsmodulen, sobald genau eine Information übergeben wird. Dies ist beispielsweise zwischen den Modulen *Think-Ë-Create-Modul Discussion* und *Brainstorming* der Fall. *Discussion* hat eine Ausgangsverbindung von $1 - n$. *Brainstorming* verfügt über eine Eingangsverbindung von $0 - 1$, dadurch entsteht eine solide Verbindung, wenn diese beiden Module aufeinander treffen.

Der dritte Typ verwendet drei solide Linien, die entstehen, wenn mehr als eine Information übergeben wird. Zwischen den Modulen *3 Variations* und *Voting* tritt diese Variante auf. Das *Think-Ë-Create-Modul 3 Variations* verfügt über eine Ausgangsverbindung von drei und das *Decide-Modul Voting* hat eine Eingangsverbindung von n . Abbildung 3.1 zeigt die soeben genannten Verbindungstypen und Module.

3.1.4 Bearbeiten der Module

Wurde ein Modul auf der Arbeitsfläche positioniert, existieren dafür unterschiedliche Einstellungsmöglichkeiten. Diese werden direkt mit dem Klick auf das zu verändernde Modul abgerufen. Die erscheinende Parameterübersicht ermöglicht das Einstellen der Dauer, des Arbeitsmodus, der Startzeit sowie das Fixieren eines Modules zur gegebenen Startzeit, um eine Verschiebung durch Variation der Zeitverteilung zu verhindern. Dies ist vor allem für Pausen-Module essentiell. Weiters ist in dieser Optionsfläche auch das Löschen des Modules durchführbar.

Um dem Benutzer einen subtilen Hinweis auf die Einstellungsoptionen zu geben, werden, inspiriert durch Collomb u.a. (beschrieben unter Abschnitt 2.5.5), die möglichen Optionen direkt in Form einer Hover-Schaltfläche zum Element gebracht. Diese Schaltfläche bietet ebenfalls die Möglichkeit zum Löschen und Bearbeiten des Modules sowie zum Anzeigen der Modulbeschreibung.

Eine weitere Option zum Entfernen eines Modules wird mittels Drag & Drop-Technologie geboten. Sobald der Benutzer ein bereits platziertes Element aus dem Workflow herauszieht erscheint das, einen Mülleimer darstellende, Icon, das auch in unterschiedlichen Betriebssystemen wie *Windows* oder *Mac OS* Anwendung findet [16]. Diese Funktionalität wurde von den Nutzern bereits jahrelang in dieser Form verwendet, sodass es mittlerweile für jeden verständlich ist. Sobald das Element auf dem Papierkorb losgelassen wird, wird das Element nach einer Abfrage gelöscht, das Icon verschwindet wieder von der Bildfläche.

Weiters existiert die Möglichkeit ein Element durch ein anderes zu ersetzen. Dies erspart dem Nutzer das Löschen eines Modules, gefolgt vom Einfügen eines anderen an gleicher Stelle. Wird ein Modul über ein bereits platziertes Element gezogen, so vergrößert sich das aktuelle Modul um dem Benutzer ein Ersetzen zu visualisieren. Wird das Modul daraufhin losgelassen, so erscheint eine Abfrage ob das Modul wirklich ersetzt werden soll.

3.1.5 Anpassbarkeit und optionale Hilfestellungen

Da ein Benutzer möglicherweise kein methodisches Wissen über die Planung eines Workshops besitzt, sind sowohl Benutzerführung als auch Erklärungen der einzelnen Module zur Verfügung zu stellen. Der Editor soll aber nicht nur für Anfänger sondern auch für erfahrene Benutzer geeignet sein und deren Bedürfnisse zufrieden stellen. Dies erfordert möglichst unauffällige, optionale Hilfestellungen, die bei Bedarf aufgerufen oder eben übersprungen werden können.

Wenn einem minder erfahrenen Benutzer der Drag & Drop Mechanismus verborgen bleibt und mit einem Klick ein Modul zu selektieren versucht, so wird ein Pfeil auf das Platzhalterelement in der Arbeitsfläche sichtbar.

Dieser Hinweis bleibt möglicherweise einem bereits erfahreneren Benutzer verborgen, da er das Element nicht selektiert, sondern standesgemäß per Drag & Drop auf der Arbeitsfläche platziert.

3.1.6 Zeitlinie und Agenda

Das Ergebnis der Komposition unterschiedlicher Module ist ein valider, linearer Arbeitsablauf, der optisch durch eine strukturierte Liste dargestellt wird. Aufgrund dieser strukturierten Liste wird zeitgleich eine visuelle Zeitlinie zur besseren Zeitplanung sowie eine Agenda für den Workshop erstellt. Später dient die zusammengesetzte Liste auch als Arbeitsmittel des geplanten Workshops.

Die Zeitlinie wird von der Start- bis zur Endzeit des Workshops angesetzt. Elemente werden darin in der Modul-Typ-Farbe und mit dem Modultitel auf einem Zeitstrahl visualisiert. Kommt es zu einer zeitlichen Überschneidung zweier Module, so werden diese untereinander dargestellt, bis die Zeitüberschneidung korregiert wurde. Die Zeitlinie dient neben dem optischen Überblick auch der Beachtung des Zeitlimits. Wird die Workshop-Zeit durch das Platzieren zu vieler beziehungsweise zu langer Module überschritten, so bekommt der Benutzer eine Warnmeldung.

Zur Vereinfachung der Workshop-Planung wird begleitend auch eine Agenda erstellt, die einen Überblick über die geplanten Start- und Endzeiten der Module bietet. Weiters sind die Dauer der Module und die Modul-Titel vermerkt.

3.2 Visualisierungsentwürfe

Es wurden folgende Entwürfe für die Visualisierung der Modulabfolge erstellt.

3.2.1 Entwurf 1 – Baum

Abbildung 3.2 zeigt einen Vorschlag für die Darstellung bei weniger als zehn Modulen. Für mehrere Elemente ist diese Visualisierung weniger geeignet, da durch die baumartige Struktur sehr schnell ein großer Darstellungsbereich notwendig ist.

Entwurf 1 beinhaltet eine Toolbox mit fünf Reiterkarten für die unterschiedlichen Modultypen. Die Module werden darin mittels eines Bildes und der Modul-Bezeichnung dargestellt. Sind Module an der derzeitigen Position aufgrund ihrer Eingangs- und Ausgangsverbindungen nicht möglich, so werden diese bereits im der Toolbox ausgegraut. Unter der Toolbox findet man zwei Formularfelder zur Eingabe der Dauer und der Arbeitsmodi für die einzelnen Module.

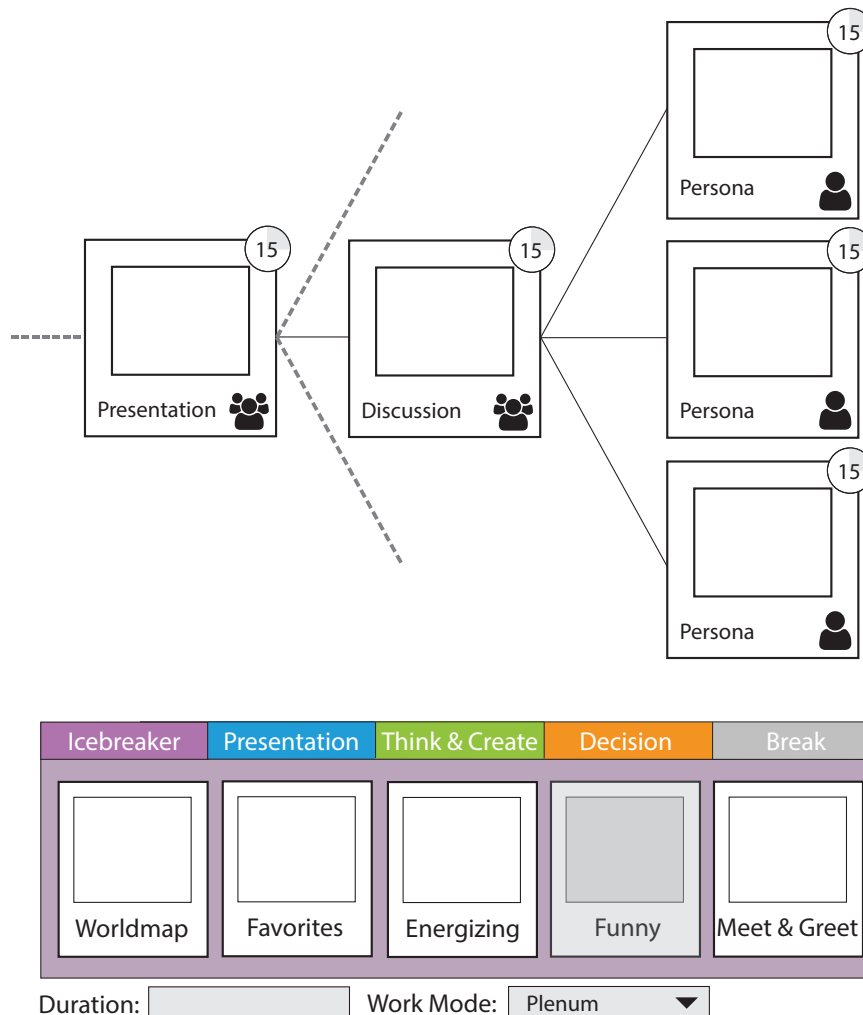


Abbildung 3.2: Entwurf 1 – *Baum* für die Visualisierung der Modulabfolge bei weniger als zehn Modulen.

Zur Auswahl eines Modules wird ein Element mittels Drag & Drop auf die Arbeitsfläche gezogen. Visualisiert wird ein Modul danach ähnlich wie in der Toolbox, jedoch ist auch der Arbeitsmodus mittels Icon und Dauer (in Minuten) ersichtlich. Die platzierten Elemente werden mit einer soliden Linie verbunden. Gibt es noch weitere Möglichkeiten parallel Module einzufügen (beispielsweise durch Änderung des Arbeitsmodus), so werden strichlierte

Linien angezeigt um dem Benutzer ein einfaches Einhängen an dieser Stelle anzuzeigen. Je nach Arbeitsmodus werden auf einer Ebene ein, zwei oder drei Module eingeblendet. Handelt es sich um den Modus *Plenum*, so wird ein Element dargestellt, bei *Gruppe* sind es zwei Module, *Individuell* visualisiert drei Module auf gleicher Ebene.

Dieser Entwurf hat den Vorteil, dass die jeweiligen Arbeitsmodi nicht nur durch das Icon sofort ersichtlich sind, sondern auch durch die Visualisierung der parallelen Elemente. Das Anzeigen der Modulbilder und des Modul-Titels komplettiert die übersichtliche Darstellung, die nicht überladen wirkt. Nachteilig ist allerdings, dass die Baumstruktur sehr schnell umfangreich wird und es dem Benutzer am Anfang möglicherweise nicht klar ist, warum Module auf mehreren Ebenen dargestellt werden. Außerdem ist diese Visualisierungsart für größere Workshops, wie bereits erwähnt, gänzlich ungeeignet, wodurch eine kompaktere Darstellungsform gefunden werden muss.

3.2.2 Entwurf 2 – Lego-Steine

Abbildung 3.3 zeigt eine Darstellungsvariante, die sich auch für eine größere Anzahl von Modulen eignet, da die Visualisierung im Gegensatz zu Entwurf 1 sehr platzsparend ist.

Wie Entwurf 1 beinhaltet auch Entwurf 2 eine Toolbox mit fünf Reiterkarten für die Gliederung der Modultypen. Die Module selbst sind hier aber bausteinförmig und werden nur mittels Modul-Titel beschrieben. Ein Bild pro Modul ist hierbei nicht vorhanden. Module, die an der aktuellen Position nicht valide wären sind auch in diesem Entwurf wieder ausgegraut. Ebenso wie bei Entwurf 1 wird unter der Toolbox ein Formularfeld zur Eingabe der Dauer und ein Dropdown-Menü zur Selektion des Arbeitsmodus angezeigt.

Die visualisierte Form der Module auf der Arbeitsfläche kann von jener in der Toolbox etwas abweichen, da die einzelnen Elemente wie Lego-Steine zusammengebaut werden. Bausteine in der Toolbox haben teilweise strichlierte Linien, wo dabei die Form an die bereits platzierten Objekte individuell angepasst wird. Entwurf 2 zeigt im unteren Teil die jeweiligen Formen der Modultypen wie diese in der Toolbox erscheinen.

Mittels Drag & Drop platzierte Elemente bekommen, wie auch bei Entwurf 1, als zusätzliche Information ein Icon für den Arbeitsmodus und die Zeitangabe in Minuten zur Darstellung der geplanten Dauer. Sollten Elemente zwischen zwei bereits positionierten Modulen eingefügt werden, so gleiten diese auseinander und geben dem Benutzer den Hinweis, dass ein Einfügen an dieser Stelle möglich ist, vorausgesetzt die Eingangs- und Ausgangsverbindungen sind valide. Handelt es sich um eine nicht valide Position, so schaffen die platzierten Module keinen Platz für das neue Element.

Der Vorteil von Entwurf 2 ist, dass diese Darstellungsart eine platzsparende Alternative zu Entwurf 1 bietet. Die Dauer sowie der Arbeitsmodus

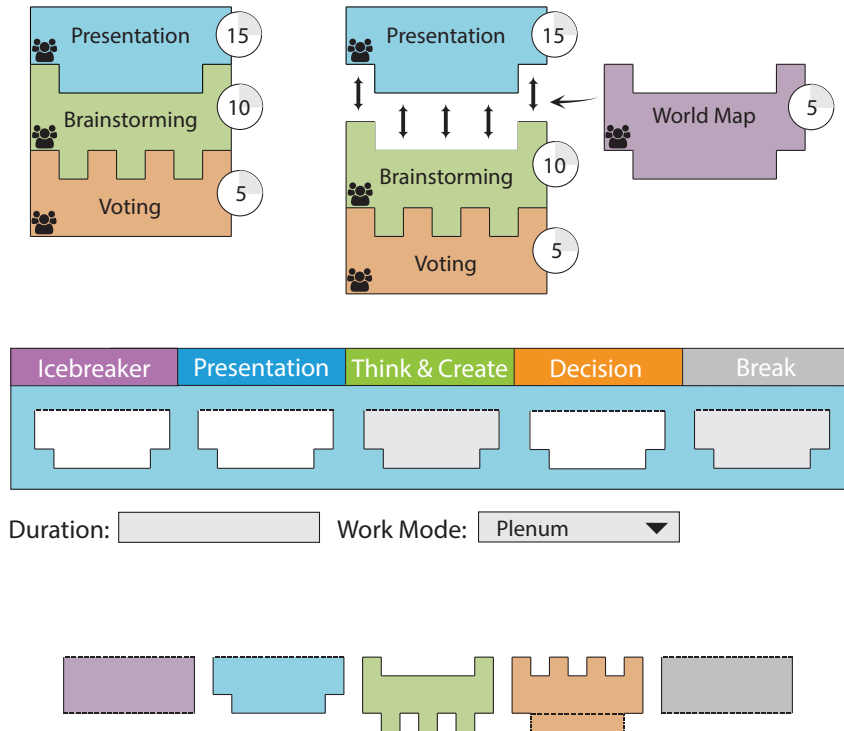


Abbildung 3.3: Entwurf 2 – *Lego-Steine* für die Visualisierung der Modulabfolge bei mehr als zehn Modulen.

ist wie bei Entwurf 1 sofort ersichtlich, dem Benutzer wird durch das Auseinandergleiten positionierter Module deutlich gemacht, wo ein bereits gewähltes Element eingefügt werden kann. Der Nachteil dieser Variante ist, dass die unterschiedlichen Formen der einzelnen Module für den Benutzer nicht verständlich sein könnten und die Toolbox durch die unzähligen, kantigen Objekte sehr unübersichtlich erscheint.

3.2.3 Entwurf 3 – Post-It's

Abbildung 3.4 zeigt einen Entwurf, der einer Tafel mit Post-It's ähnelt. Auf der rechten Seite des Bildschirms wird wie bei den Entwürfen 1 und 2 ebenfalls eine Toolbox geboten. Entwurf 3 verwendet allerdings keine Reiterkarten, jedoch werden einzelnen Modultypen Bereiche zugeteilt, worin sich dann die jeweiligen Module befinden. Dieser Bereich ist je nach Anzahl der Elemente horizontal scrollbar.

Die Arbeitsfläche ist initial mit einem Platzhalterobjekt ausgestattet.

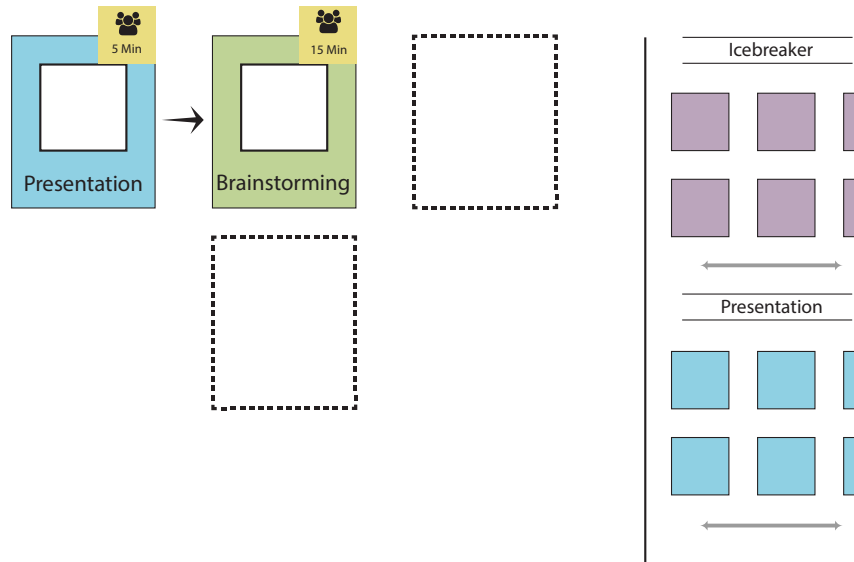


Abbildung 3.4: Entwurf 3 – *Post-It's* für die Visualisierung der Modulabfolge.

Werden Module mittels Drag & Drop darauf platziert, so erscheint ein Dialogfenster, das die Eingabe der Dauer erfordert. Das platzierte Element weist neben einem Modul-Bild und einem Titel noch ein kleines, darauf liegendes Post-It mit der Zeit- und Arbeitsmodus-Information auf. Für alle Elemente wird initial der Arbeitsmodus *Plenum* selektiert.

Platzierte Elemente werden, wie Abbildung 3.4 zeigt, mittels Pfeil verbunden, mögliche weitere Platzhalter werden eingezeichnet. Wird das nächste Modul auf der gleichen Ebene, also unter dem soeben platzierten Element, abgelegt, so ändert sich der Arbeitsmodus der beiden Module auf *Gruppe*. Würde auf dieser Ebene dann auch noch ein drittes Modul abgelegt werden, so wird der Arbeitsmodus jeweils auf *Individuell* geändert.

Der Vorteil von Entwurf 3 ist, dass der Benutzer durch das Platzhalterobjekt genau weiß, an welcher Stelle das nächste Objekt abgelegt werden darf. Die Darstellung der Arbeitsmodi ist, wie auch bei Entwurf 1, sehr schnell ersichtlich, jedoch erfordert dadurch auch die Darstellung dieser Variante sehr viel Platz. Ein weiterer Nachteil ist, dass das Einfügen zwischen bereits platzierten Elementen problematisch werden kann, da in diesem Entwurf keine Platzhalter zwischen den Elementen vorgesehen sind. Außerdem wird die Darstellung unübersichtlich, sobald ein Zeilenumbruch der Elemente aufgrund der Bildschirmbreite erforderlich ist.

Kapitel 4

Implementierung

Das folgende Kapitel erläutert die konkrete Implementierung und die verwendeten Bibliotheken des in Kapitel 3 vorgestellten Web-Modul-Editors.

4.1 Architektur

Die Umsetzung des Web-Modul-Editors ist in unterschiedliche Komponenten und Bereiche unterteilt, die unter Abbildung 4.1 ersichtlich sind.

Die unterste Schicht bilden dabei die externen Bibliotheken, die bei der Umsetzung ihre Anwendung fanden. In der nächsten Ebene sind die Dateien zu erkennen, die als Adapterklassen der Bibliotheken dienen. Darin wurden die jeweiligen Funktionalitäten der JavaScript Bibliotheken implementiert, damit sie später einfach eingebunden werden konnten. Bibliotheken und Da-

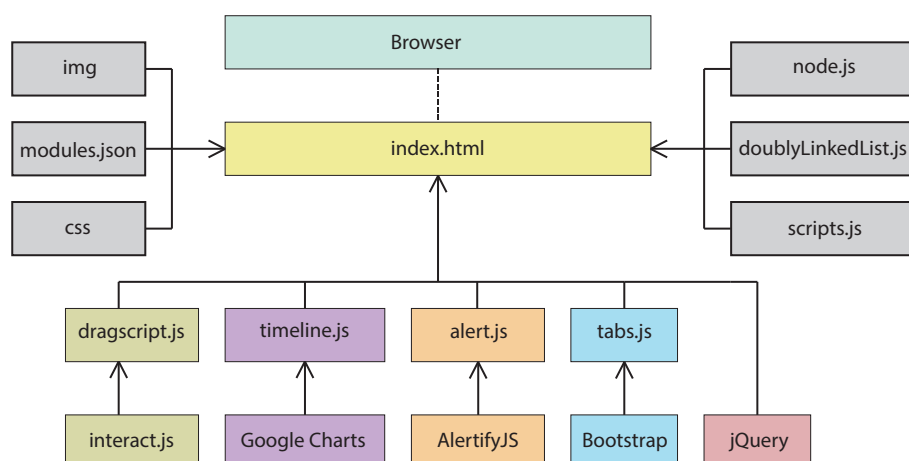


Abbildung 4.1: Aufbau und Zusammenhänge des Web-Modul-Editors.

teilen, die keine zusätzlichen Adapterklassen aufweisen, wurden direkt in der Datei `index.html` eingebunden, die alle Komponenten zusammenfügt und im Browser angezeigt wird. `*.css`-Dateien, Bild-Dateien und die `*.json`-Datei, mit den abgespeicherten Modulen, werden ebenso wie die Datei der gewählten Datenstruktur, der doppelt verketteten Liste, und weiteren Hilfs-`scripts` direkt der Datei `index.html` zugewiesen.

Diese Struktur ergibt einen sehr modularen Aufbau, wodurch Komponenten und einzelne Teile relativ rasch von einem Programmierer ausgetauscht, ersetzt oder erweitert werden können.

4.2 Verwendete Bibliotheken

Für die Umsetzung des Editors wurde die Scriptsprache *JavaScript* gewählt, da sich diese für die Webentwicklung sehr gut anbietet, keine zusätzlichen Installationen notwendig sind und dennoch einen großen Funktionsumfang mit sich bringt. Zur Realisierung des Projektes wurden folgende, in Abbildung 4.1 ersichtliche, frei zugängliche Bibliotheken verwendet:

- *interact.js* für die Umsetzung des Drag & Drop-Mechanismus,
- *Google Charts Timeline* zur Visualisierung der Zeitlinie,
- *AlertifyJS* zur Gestaltung von Browser-Dialogfenster,
- *Bootstrap* für die responsive Gestaltung und zur Umsetzung der Tab-Funktion der *Toolbox*.

Die Bibliotheken und deren gebotene Funktionalitäten werden in den folgenden Unterkapiteln diskutiert.

4.2.1 `interact.js`

Bei *interact.js* handelt es sich um ein JavaScript Modul, das unter Anderem die Realisierung von Drag & Drop-Mechanismen ermöglicht. In der vorliegenden Arbeit ist es dafür verantwortlich, dass die vordefinierten Module aus der *Toolbox* einfach mittels Drag & Drop zusammengesetzt sowie aus dem Workflow genommen und gelöscht werden können. *interact.js* stellt dafür unterschiedliche Funktionen zur Verfügung, die zu einem rasch realisierbaren Ergebnis führen.

Die Kernfunktion von *interact.js* nennt sich `interact`, darauf basieren alle zulässigen Interaktionen. Ihr wird ein bestimmtes Element und ein optionales Objekt für Optionen übergeben. Danach wird sie dafür verwendet, Elemente interagierbar zu machen, indem unterschiedliche Methoden und Eigenschaften zur Konfiguration mitgegeben werden [8].

Ein vereinfachtes Beispiel der Verwendung von `interact` im Löschprozess eines Modules vom Web-Modul-Editor zeigt Programm 4.1. Mittels CSS Selektor wird das gewünschte Element, in diesem Fall die Klasse `recycleBin`,

Programm 4.1: Vereinfachtes Beispiel für die Verwendung der Kernfunktion `interact` im Web-Modul-Editor.

```
1 interact('.recycleBin')
2   .dropzone({
3
4     // only accept elements matching this CSS selector
5     accept: '#dropped',
6
7     // requires an overlap of 75 percent
8     overlap: 0.75,
9
10    ondrop: function (event) {
11      // removes the node
12      removeNode(searchNodeAt(hoverNode));
13    }
14  });
```

in der Schnittstelle ausgewählt und durch bestimmte Methoden und Eigenschaften veränderbar gemacht. Diese Klasse ist dem Mülleimer-Icon zugewiesen. Sobald ein Element darauf abgelegt wird muss es gelöscht werden.

Im unten stehenden Programm wird definiert, dass die Klasse `dropzone` lediglich Elemente mit der ID `dropped` akzeptiert und es eine Überlappung von 75 % geben muss, damit ein Modul darauf abgelegt werden kann. Sind diese Bedingungen erfüllt und wird das Modul darauf losgelassen, so wird die Funktion `removeNode()` aufgerufen und der Benutzer gefragt ob er das Modul tatsächlich löschen möchte.

4.2.2 Google Charts Timeline

Google Charts ermöglicht die Visualisierung unterschiedlichster Daten mittels zur Verfügung gestellter Bibliotheken. Eine mögliche Visualisierungsart ist die Zeitlinie. Nach dem Einbinden der gewünschten Bibliothek wird JavaScript-Code eingebettet, der die Daten beinhaltet und die Grafik individuell anpasst. Daraufhin wird ein Objekt mit einer `id` erzeugt. Zum Visualisieren der Zeitlinie muss daraufhin ein `div` mit dieser `id` eingefügt werden. Es sind keine Plugins oder sonstige Software für das Rendern der Grafiken notwendig, da *Google Charts* auf die HTML5 und SVG-Technologie aufbaut. Dies ermöglicht eine plattformunabhängige Darstellung [12].

Zur Darstellung der Zeitleiste wird von *Google Charts* ein sogenanntes Gantt-Diagramm, ein Balkenplan, verwendet. Es erlaubt eine mehrreihige Visualisierung einzelner Balken über einen bestimmten Zeitraum. Ebenso ist es möglich, Beschriftungen und individuelle Farben für die Balken zu vergeben. Durch ein Hover-Event über die einzelnen Balken erscheinen Details zum jeweilig gewählten Objekt.

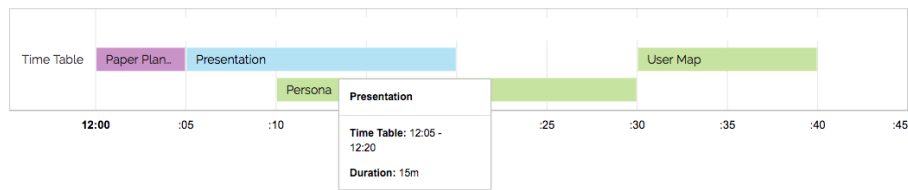


Abbildung 4.2: Zeitlinie des Web-Modul-Editors und Hover-Box mit Details.

Beim Web-Modul-Editor wurde *Google Charts Timeline* dafür verwendet die Zeitplanung eines Workshops visuell darzustellen. So hat der Benutzer die Möglichkeit auf den ersten Blick zu sehen wie viel Zeit der gesamten Workshopdauer er schon verplant hat und ob es Überlappungen der Module gibt. Abbildung 4.2 zeigt die Darstellung der Zeitlinie im Web-Modul-Editor sowie die erscheinende Box mit ergänzenden Informationen bei einem Hover-Event.

Programm 4.2 zeigt eine vereinfachte Version des Codes zur Einbettung der *Google Charts Timeline* in den Web-Modul-Editor. Nach dem Laden der von *Google Charts* benötigten Bibliothek wird in der Funktion `drawChart()` die Grafik modelliert. Mittels `dataTable.addColumn` werden die einzutragenden Spalten festgelegt. In diesem Fall sind es die Beschreibung, der Modulname, sowie die Start- und Endzeit des jeweiligen Modules.

Im Array `row` werden die Parameter für jedes Modul gesondert gespeichert und danach an das Array `rows` übergeben, das die gesammelten Daten aller bereits platzierten Module abspeichert. Mithilfe der Funktion `dataTable.addRow` werden die gesammelten Daten der Module zum visualisieren übergeben. Hier könnte zusätzlich eine Variable `options` hinzugefügt werden, die diverse Anpassungen wie die Farbwahl, Schriftarten oder auch die Darstellung von Überlappungen regelt. Dies ist im vereinfachten Beispiel nicht ersichtlich, jedoch wird in der vollständigen Version für den Web-Modul-Editor von dieser Funktion Gebrauch gemacht.

4.2.3 AlertifyJS

Bei *AlertifyJS* handelt es sich um ein JavaScript-Framework, das gestaltete, responsive Browser-Dialogfenster und Benachrichtigungen erstellt. Dafür stellt es vier unterschiedliche Komponenten zur Verfügung [27]:

- *Alert* zur Erstellung einer gewöhnlichen Alert-Box, die vom Benutzer bestätigt werden muss
- *Confirm* zur Erzeugung einer Bestätigungs-Box wodurch ein Benutzer Aktionen genehmigen oder widerrufen kann
- *Prompt* zum Erzeugen einer Dialog-Box, in die der Benutzer selbst

Programm 4.2: Vereinfachter Code zur Einbettung der Zeitlinie im Web-Modul-Editor mittels *Google Charts*.

```

1 // loads the libraries
2 google.charts.load("current", {packages: ["timeline"]});
3 google.charts.setOnLoadCallback(drawChart);
4
5 function drawChart(node) {
6     var container = document.getElementById('timeline');
7     var chart = new google.visualization.Timeline(container);
8     var dataTable = new google.visualization.DataTable();
9
10    // defines the columns
11    dataTable.addColumn({type: 'string', id: 'Description'});
12    dataTable.addColumn({type: 'string', id: 'Name'});
13    dataTable.addColumn({type: 'date', id: 'Start'});
14    dataTable.addColumn({type: 'date', id: 'End'});
15
16    rows = [];
17    row = [];
18    [...]
19
20    // pushes the single rows to visualize
21    row.push('Time Table', '', new Date(0, 0, 0, parseInt((start) / 60),
22        parseInt((start) % 60), 0), new Date(0, 0, 0, parseInt((wend) / 60)
23        , parseInt((wend) % 60), 0));
24
25    [...]
26    rows.push(row);
27    [...]
28
29    // adds the saved rows to the data table
30    dataTable.addRow(rows);
31    [...]
32
33    // draws the chart
34    chart.draw(dataTable);
35 }

```

einen Wert eintragen kann, der später vom System weiterverarbeitet wird

- *Notifier* zum Erstellen eines kleinen, unaufdringlichen Benachrichtigungsfensters

In der Implementierung des Web-Modul-Editors kommen zwei der vier Komponenten zum Einsatz. Zum einen *Alert* und zum anderen *Confirm*, die in den nächsten beiden Unterpunkten genauer beschrieben werden.

Alert

`alertify.alert(...)` ermöglicht das Erzeugen eines Pop-Up-Dialogfensters, das sicherstellt, dass ein Benutzer eine Nachricht liest. Das Dialogfenster muss durch einen Klick auf OK bestätigt werden, damit eine weitere Interaktion mit dem System möglich ist [26].

Der folgende Code zeigt den Einsatz der Alert-Box im Web-Modul-Editor. Dabei wird beim Klicken auf den Informationsbutton eines Modules seine Modulbeschreibung ausgegeben.

```
alertify.alert(myObj[searchNodeAt(node).type].description);
```

Hier wurde **Alert** in der einfachsten Weise verwendet, indem nur eine Nachricht ausgegeben wird. Weiters wäre es möglich in der Schnittstelle zusätzlich einen Titel sowie eine Funktion zu übergeben, die nach der Bestätigung ausgeführt wird [26].

Confirm

`alertify.confirm(...)` ermöglicht das Erzeugen einer Bestätigungs-Box, mit jener Abfragen durchgeführt werden können. Der Benutzer kann die gewünschte Interaktion entweder durch einen Klick auf OK bestätigen oder durch **Cancel** widerrufen [28].

Der folgende Code zeigt den Einsatz der Bestätigungs-Box im Web-Modul-Editor. In diesem Fall wird abgefragt, ob der Benutzer das Modul wirklich löschen möchte. Wird der Dialog bestätigt, so wird die Funktion `removeTrue()` aufgerufen und der Knoten gelöscht. Es wäre möglich auch noch eine Callback-Funktion anzugeben, hier wird der Vorgang jedoch einfach abgebrochen.

```
alertify.confirm("Do you really want to delete this Node?", function  
() {  
    removeTrue()  
}, function () {});
```

Confirm erlaubt als Übergabeparameter in der Schnittstelle neben einer Nachricht und einer Funktion im Bestätigungsfall auch die Angabe eines Titels sowie der oben bereits erwähnten, möglichen Funktion im Ablehnungsfall [28].

4.2.4 Bootstrap

Bootstrap ist ein sehr bekanntes HTML, CSS und JavaScript Open Source Framework zur Erstellung responsiver Webseiten. Es wird von der Onlineplattform *Twitter* zur Verfügung gestellt und ist seit August 2011 auf *GitHub*, einem Open-Source Hosting-Dienst für Softwareentwicklungs-Projekte, verfügbar. Mittlerweile ist bereits Version 3.3.7 von Bootstrap erhältlich.

Das umfangreiche Framework baut auf ein 12-spaltiges Rastersystem auf und beinhaltet individuell definierbare Umbruchpunkte für unterschiedliche Bildschirmgrößen. Die Breite des Rasters ist mit 940 Pixel standardisiert und in Spalten unterteilt, was eine einfache Layouterstellung ermöglicht. Bootstrap beinhaltet zahlreiche, vordefinierte und -formatierte Elemente wie Labels, Hinweise, Navigationen, Tooltips, Formularfelder, Tabellen und vieles mehr. Es besteht die Möglichkeit eine Einschränkung auf die für die Umsetzung benötigten Module zu treffen, um zu große Datenmengen zu vermeiden und die Performance zu verbessern. Für Neulinge sowie zur schnellen Umsetzung einer Webseite, bietet Bootstrap auch eine Vielzahl an vorgefertigten sowie individuell anpassbaren Vorlagen und Themen an, die sich auch mit Content Management Systemen wie beispielsweise WordPress verknüpfen lassen [21].

4.3 Die drei Komponenten des Web-Modul-Editors

Der Web-Modul-Editor gliedert sich in drei Hauptkomponenten:

- die *Toolbox* mit den vordefinierten Workshop-Modulen,
- die *Arbeitsfläche*, auf der der Workshop zusammengesetzt wird und
- die *Zeitplanungskomponente*.

Die einzelnen Teile werden in den folgenden Unterpunkten näher behandelt und deren Umsetzung genau erläutert.

4.3.1 Toolbox

Die Module, die als Workshop-Elemente dienen, werden in einer MySQL Datenbank gespeichert, als *.json-Datei exportiert und vom Modul-Editor in der Initialisierungsphase importiert. Dies passiert beim Aufruf der Datei *index.html*, die den Einstiegspunkt des Editors darstellt.

Tabelle 4.1 zeigt die Struktur der Datenbank. Jedes Modul verfügt neben einer *id*, einem *title*, einem *type*, einem *work_mode*, einer *duration* und einer *image_url* auch über eine *description*. Das Feld *description* enthält eine kurze Beschreibung zu den jeweiligen Modulen, sodass Benutzer ohne methodischem Wissen bei Bedarf die Informationen aufrufen können. *input* und *output* speichern die Eingangs- und Ausgangsverbindungen der jeweiligen Module. Diese werden in den Variablen mittels Regexnotation abgespeichert, um danach ein möglichst einfaches Prüfen der Validität zu ermöglichen. Der Validitätscheck wird unter Abschnitt 3.1.2 genauer erläutert. Das Feld *tags* wurde vorsorglich eingebunden und dient einer optionalen Suchfunktion in der *Toolbox*, die später bei Bedarf implementiert werden könnte. In der aktuellen Implementierung ist die Suchfunktion aber weder verfügbar, noch geplant.

Tabelle 4.1: Struktur der MYSQL Datenbank.

<i>Name</i>	<i>Typ</i>	<i>Attribute</i>	<i>Extra</i>
int	int(10)	UNSIGNED	AUTO_INCREMENT
title	varchar(100)		
description	text		
type	varchar(100)		
work_mode	varchar(100)		
duration	int(11)		
input	varchar(100)		
output	varchar(100)		
tags	varchar(100)		
image_url	varchar(100)		

Sobald der Editor initialisiert wird, wird die exportierte `*.json`-Datei analysiert, der Inhalt gespeichert sowie die Module in die *Toolbox* geladen. Die initiale `*.json`-Datei ist im Anhang unter Abschnitt A zu finden.

Die *Toolbox* selbst wurde mittels Tab-System von *Bootstrap* (siehe Abschnitt 4.2.4) realisiert und verfügt über fünf Karteireiter für die Modultypen, die in Kapitel 3 diskutiert werden. Das finale Erscheinungsbild der *Toolbox* ist in Abbildung 4.3 zu finden.

4.3.2 Arbeitsfläche

Die *Arbeitsfläche* ist das Kernstück des Modul-Editors mit der umfangreichsten Funktionalität. In erster Linie beinhaltet sie die Fläche zum Platzieren der Module, wodurch die Zusammensetzung des Workshops ermöglicht wird. Sie dient gleichzeitig als grafische Visualisierung des Ablaufes.

Die einzelnen Module werden als Knoten in einer doppelt verketteten Liste abgespeichert. Diese Methode wurde gewählt, da er den Vorteil mit sich bringt, dass das Entfernen und Einfügen einzelner Knoten durch das Verbiegen von Zeigern effizient realisiert werden kann. Durch den dynamischen Ansatz ist kein Umkopieren der Daten und stetiges Allokieren des Speichers notwendig, was sich erheblich auf die Performance auswirkt.

Initial enthält die doppelt verkettete Liste einen Platzhalter-Knoten, der dem Benutzer visualisiert, an welcher Stelle er das erste Modul platzieren kann. Nachdem ein Element zum Workshop hinzugefügt wurde und seine Ein- und Ausgangsparameter signalisieren, dass das Element als Beginn der Liste fungieren kann, wird der initiale Platzhalter-Knoten durch diesen ersetzt. Ab jenem Zeitpunkt wird für jedes Modul, das zum Workshop

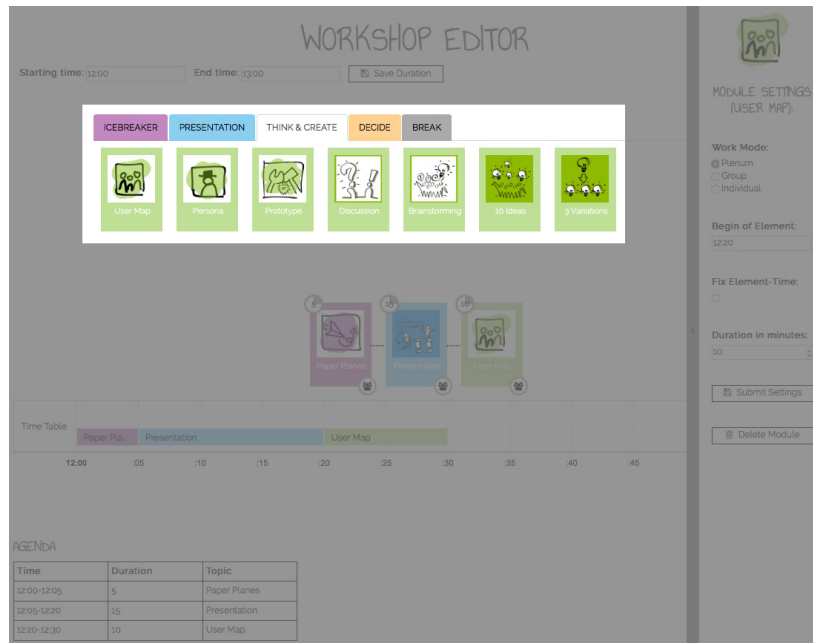


Abbildung 4.3: Erscheinungsbild der *Toolbox*.

hinzugefügt werden sollte, berechnet, wo dieses platziert werden kann. Die Berechnung basiert auf den in der Datenbank spezifizierten Verbindungseigenschaften jedes Modules und bezieht sich auf den jeweils nächsten, bereits platzierten, Knoten der *Arbeitsfläche*. Die Möglichkeit des Hinzufügens wird dem Benutzer durch temporäre Platzhalter-Knoten visualisiert. Das Ablegen eines gezogenen, neuen Modules ist nur auf einem Platzhalter-Knoten möglich. Ist ein Einfügen nicht möglich, so nähern die benachbarten, bereits platzierten Knoten einander an. Zusätzlich zur Generierung des Platzhalters signalisiert die Einfärbung des Rahmens, wie unter Abschnitt 3.1.2 bereits erwähnt, die Möglichkeit des Einfügens.

Die Berechnung des nächsten Knotens erfolgt anhand der Suche der geringsten Vektorlänge im Raum bezogen auf den hinzuzufügenden Knoten. Dies geschieht mittels des Satzes von Pythagoras. Die Distanz zweier Punkte wird durch die Subtraktion der x- und y-Koordinaten bestimmt. Die Suche des nächsten Knotens wird durch die Suche des globalen Minimums realisiert. Die angewandte Formel ist in Programm 4.3 zu sehen.

Die Richtung und Länge des Vektors indizieren, ob der nächste Knoten durch den gezogenen Knoten ersetzt, davor oder danach eingefügt werden soll. Hier bestimmt die Unterschreitung eines definierten Minimalabstandes, dass der Knoten ersetzt wird, eine negative Richtung, dass davor, eine positive Richtung, dass danach eingefügt werden soll. Kann das aktuelle Element platziert werden, so wird ein neuer Knoten generiert und in die Liste ein-

Programm 4.3: Berechnung des nächsten Knotens.

```

1 // offset for minimum function; number which can never be reached
2 offsety = -Math.pow(2, 32)-1;
3 offsetx = -Math.pow(2, 32)-1;
4 closest = node;
5
6 do {
7   // fetches each node from the draw area
8   var nodel = document.getElementsByTagName('itemprop')[parseInt(
9     getNodeIndex(node)) + divs.length];
10
11  // calculates the distance in y and x direction
12  var tmpy = nodel.offsetTop + getNodeHeight() / 2 - (event.pageY +
13    elemOffsetY);
14  var tmpx = nodel.offsetLeft + getNodeWidth() / 2 * 1.2 - (event.pageXX
15    + elemOffsetX);
16
17  // calculates the closest node
18  if (closest == null || Math.sqrt(Math.pow(tmpx, 2) + Math.pow(tmpy, 2))
19    < Math.sqrt(Math.pow(offsetx, 2) + Math.pow(offsety, 2))) {
20    offsety = tmpy;
21    offsetx = tmpx;
22    closest = node;
23  }
24 } while (node != null);

```

gegliedert. Die Verbindung wird danach dem Benutzer, wie unter Abschnitt 3.1.3 erläutert, visualisiert. Abbildung 4.4 zeigt das finale Erscheinungsbild der *Arbeitsfläche*.

Die Platzierung der Elemente wird mit einem Drag & Drop-Mechanismus realisiert, der mittels des *interact.js* JavaScript Modules implementiert wurde (siehe Abschnitt 4.2.1). *interact.js* stellt diverse Handler und Listener zur Verfügung, die an beliebige HTML-Komponenten gebunden werden können. Weiters wird eine individuelle Komponente als *Dropzone* definiert, auf der die Elemente platziert werden können. Sobald ein Modul gezogen wurde, wird ein Callback aktiviert, der die voraussichtliche Position, an der eingefügt werden sollte, berechnet. Die Berechnung wird dem Benutzer mittels Einfügen von Platzhaltern und farblicher Rahmenkennung (Rot oder Grün) visualisiert. Wird das Modul an einer validen Stelle losgelassen, so wird ein neuer Knoten mit entsprechenden Parametern generiert und in die doppelt verkettete Liste eingegliedert.

Zum Löschen mittels Drag & Drop wurde das Papierkorbelement als sekundäre *Dropzone* erklärt. Wird darauf ein Modul platziert, so wird es

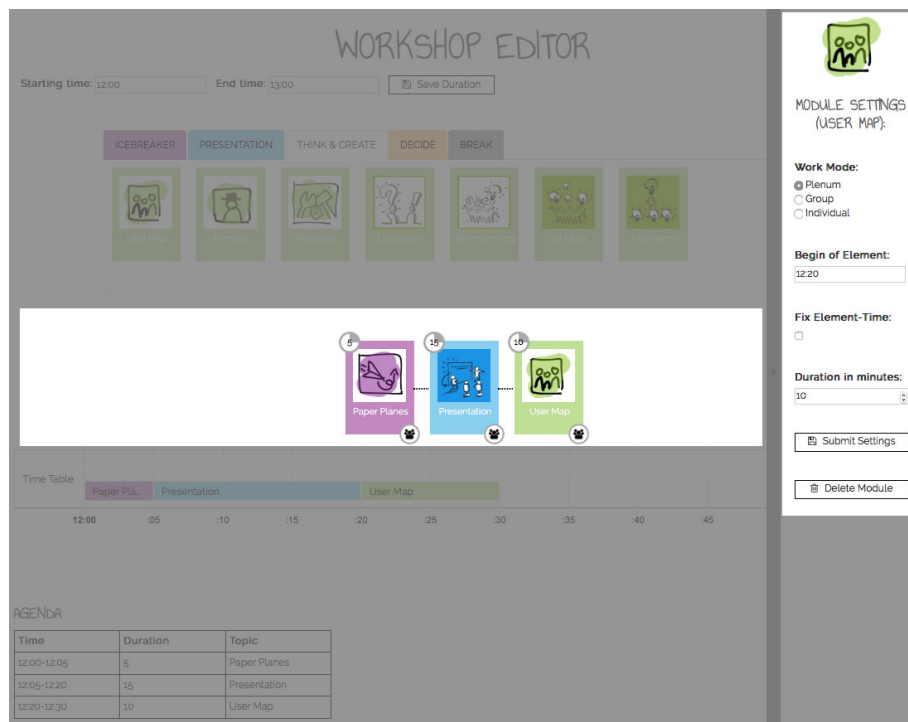


Abbildung 4.4: Erscheinungsbild der *Arbeitsfläche*.

nach positiver Rückmeldung des Benutzers auf die Abfrage gelöscht. Die möglichen Löschvorgänge sind unter Abschnitt 3.1.2 zu finden.

Der Bereich der *Arbeitsfläche* umfasst ebenso die Parameterübersicht des selektierten Modules, die in Abschnitt 3.1.4 diskutiert wird.

4.3.3 Automatisierte Zeitplanung

Die *Zeitplanung* unterteilt sich in zwei Komponenten. Zum einen in den *Zeitstrahl*, der mittels *Google Charts Timeline* (siehe Abschnitt 4.2.2) visualisiert wurde, zum anderen in die *Agenda*, die in Tabellenform dargestellt wird.

Um den *Zeitstrahl* an den gewünschten Zeitraum anzupassen, wird vom Benutzer die Start- und Endzeit des Workshops mithilfe von zwei Formularfeldern konfiguriert und gespeichert. Daraufhin wird die Balkenansicht in der Funktion `drawChart()` generiert.

Zuerst wird geprüft, ob sich das Modul noch innerhalb der definierten Workshop-Zeit befindet. Ist dies nicht der Fall, wird der Benutzer benachrichtigt und die darüber hinausgehende Dauer zum Zeitraum des Workshops addiert.

Mittels `row.push` kann ein Element zum *Zeitstrahl* des Workshops hinzu-

gefügt werden. Hierbei kennzeichnet der erste Eintrag Anfangs- und Endzeit des Workshops. Für alle Workshop-Elemente wird der geplante Start- und Endzeitpunkt berechnet, bevor sie unter Berücksichtigung des Modultitels und der Dauer ebenfalls am *Zeitstrahl* visualisiert werden.

Für die Überprüfung, ob sich Module aufgrund einer Zeitgleichnis überlappen, wurde ein eigenes Level-System eingeführt. Dies prüft, ob eine zeitliche Überschneidung vorliegt. Sobald der Fall eintritt, wird das zweite Modul in ein höheres Level, visualisiert durch das hinzufügen einer zusätzlichen Zeile im *Zeitstrahl*, geschoben. So wird mit jedem Modul verfahren und die Zeitlinie aufgebaut.

Die Zuordnung der Farben pro Modultyp wird mit einem eigenen Farbdictionary vorgenommen. Dies ist notwendig, da das *Google Charts Timeline-Framework* die Farben aus dem Farbbarray sequentiell an die zum *Zeitstrahl* hinzugefügten Module anhand des Modultypes bindet. Vor allem bei Überlappungen ist dafür eine ausführliche Überprüfung notwendig, um die korrekte Zuordnung zu garantieren.

Um dem Workshop-Moderator das Erstellen des Ablaufplanes für die Teilnehmer zu ersparen, kann die generierte *Agenda* als Grundlage übernommen werden. Diese wird in Form einer HTML-Tabelle generiert und beinhaltet die zeitliche Planung, die daraus resultierende Dauer sowie den Titel eines jeden Modules. Abbildung 4.5 zeigt das Erscheinungsbild des *Zeitstrahles* sowie der *Agenda* und der Formularfelder zur Konfiguration der Workshop-Zeit.

WORKSHOP EDITOR

Starting time: 12:00 End time: 13:00 Save Duration

ICEBREAKER PRESENTATION THINK & CREATE DECIDE BREAK

Paper Planes Presentation User Map

Time Table

12:00 :05 :10 :15 :20 :25 :30 :35 :40 :45

AGENDA

Time	Duration	Topic
12:00-12:05	5	Paper Planes
12:05-12:20	15	Presentation
12:20-12:30	10	User Map

MODULE SETTINGS (USER MAP):

Work Mode:
 Plenum
 Group
 Individual

Begin of Element:
12:20

Fix Element-Time:

Duration in minutes:
30

Submit Settings Delete Module

Abbildung 4.5: Erscheinungsbild der Zeitplanungskomponenten.

Programm 4.4: Funktion drawChart() zur Visualisierung der Zeitlinie – Teil 1.

```
1 function drawChart(node) {
2
3     [...]
4
5     // checks if the element fits in the given workshop time
6     if(wend < getWorkshopEndTime()) {
7         alertify.alert("Extended workshop to fit in element");
8         wend = getWorkshopEndTime();
9
10        // sets the new end time of the workshop
11        document.getElementById('endTime').value = (parseInt(wend / 60)
12        < 10 ? "0" : "") + (parseInt(wend / 60)).toString() + ":" + (
13        parseInt(wend % 60) < 10 ? "0" : "") + (parseInt(wend % 60)).
14        toString();
15    }
16
17    row = [];
18    // pushes the single rows to visualize
19    row.push('Time Table', '', new Date(0, 0, 0, parseInt((start) / 60),
20    parseInt((start) % 60), 0), new Date(0, 0, 0, parseInt((wend) / 60)
21    , parseInt((wend) % 60), 0));
22    rows.push(row);
23
24    levels = [];
25    // saves the actual level
26    level = getMaxLevel();
27    var i = 0;
28    var nodeDictionary = [];
29
30    // calculates the levels
31    while (i <= level) {
32        var j = [];
33        nodeDictionary.push(j);
34        i++;
35    }
36
37    // hashmap to build the timeline in respect of the levels
38    currentNode = node;
39    while (currentNode != null) {
40        nodeDictionary[parseInt(currentNode.level)].push(currentNode);
41        currentNode = currentNode.next;
42    }
43}
```

Programm 4.5: Funktion drawChart() zur Visualisierung der Zeitlinie
– Teil 2.

```
1  // builds the timeline
2  var i = 0;
3  var j = 0;
4  while (i < nodeDictionary.length) {
5      j = 0;
6      while (j < nodeDictionary[i].length) {
7          row = [];
8          tmpNode = nodeDictionary[i][j];
9          tmpStart = tmpNode.beginTime;
10         row.push('Time Table', myObj[tmpNode.type].title, new Date
11             (0, 0, 0, parseInt(tmpStart / 60), parseInt(tmpStart % 60), 0), new
12             Date(0, 0, 0, parseInt((tmpStart + tmpNode.duration) / 60), parseInt
13             ((tmpStart + tmpNode.duration) % 60), 0));
14         rows.push(row);
15         colDictionary[myObj[tmpNode.type].title] = 0;
16         j++;
17     }
18     i++;
19 }
20
21 dataTable.addRow(rows);
22
23 // calculates the color per module
24 for (var i = 0; i < dataTable.getNumberOfRows(); i++) {
25     if (!colDictionary[dataTable.getValue(i, 1)]) {
26         colors.push(getColor(getColorByName(dataTable.getValue(i, 1)
27             )));
28         colDictionary[dataTable.getValue(i, 1)] = 1;
29     }
30 }
31
32 // sets additional options like the colors
33 var options = {
34     [...]
35     colors: colors
36 }
37
38 // draws the chart
39 chart.draw(dataTable, options);
40 }
```

Kapitel 5

Evaluierung

Zur Evaluierung des praktischen Teiles der vorliegenden Arbeit wurde eine Pilotstudie durchgeführt, die in diesem Kapitel näher behandelt wird.

5.1 Pilotstudie

5.1.1 Struktur des Benutzerfreundlichkeitstests

Die Pilotstudie war in zwei Teile gegliedert. Zum einen in den praktischen Teil, dem direkten Arbeiten mit dem Tool aufgrund einer konkreten Vorgabe, zum anderen in einen kurzen Reflexionsteil, der in Form der Beantwortung eines Fragebogens durchgeführt wurde.

Die Interaktionen der Probanden wurden mithilfe des Online-Tracking-Tools *mouseflow* aufgezeichnet und während des praktischen Teiles beobachtet. Jede Person wurde im ersten Teil gebeten, angelehnt an die Think-Aloud-Methode, seine Erwartungen, Gefühle, Gedanken und Vorhaben beim Interagieren mit dem Interface laut auszusprechen. Die Äußerungen der Personen wurden notiert, sie dienten danach der Evaluierung des Modul-Editors.

Aufgabenstellung

Es gab keinerlei einführende Erklärungen zum Editor oder zur Aufgabenstellung, da das Tool selbstständig erlernt werden sollte. Alle nötigen Informationen wurden auf dem Blatt mit den konkreten Aufgaben zur Verfügung gestellt.

Die Testpersonen wurden gebeten einen Tagesworkshop mithilfe des Web-Modul-Editors zu planen. Dafür waren konkrete Schritte und Module vorgegeben, die vom Kandidaten zu finden und zusammensetzen waren. Die erforderlichen Einstellungen wie Arbeitsmodus, Dauer und das Fixieren eines Elementes waren ebenfalls vorzunehmen. Im weiteren Schritt mussten Änderungen am bereits erstellten Workshop vorgenommen werden, was die

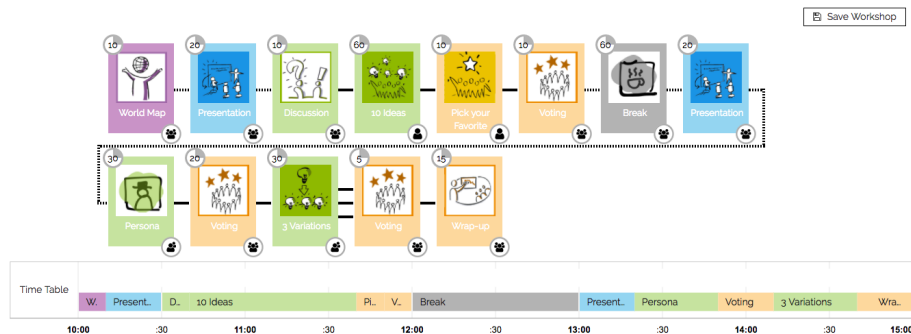


Abbildung 5.1: Lösung für die Aufgabenstellung der Pilotstudie.

Notwendigkeit eines Löschens und Ersetzens von Modulen implizierte. Sobald ein Kandidat glaubte die Aufgabe erfüllt zu haben, wurde er gebeten den erstellten Workshop zu speichern. Die konkreten Punkte der Aufgabenstellung des praktischen Teiles sind in Tabelle 5.1 aufgelistet. Daraufhin gab es noch erforderliche Änderungen durchzuführen, welche in Tabelle 5.2 einzusehen sind.

Für die Dauer der Workshoperstellung wurde mit einer Zeit von zehn bis fünfzehn Minuten gerechnet. Hat ein Kandidat nach zwanzig Minuten seine Lösung noch nicht abgespeichert, so wurde er gebeten dies zu tun und der praktische Teil des Tests wurde abgebrochen.

Abbildung 5.1 zeigt die Lösungsabfolge und die Einstellungen der zu platzierenden Module.

Durch die Aufgabenstellung für die Probanden sollte herausgefunden werden ob die Funktionalitäten des Modul-Editors von Anfang an klar sind oder ob es zu gewissen Punkten Unklarheiten gibt. Es sollte geklärt werden ob zusätzliche, optionale Hilfestellungen, wie unter Abschnitt 3.1.5 beschrieben, wahrgenommen sowie genutzt oder diese als störend empfunden werden. Zudem wird evaluiert ob zu einzelnen Funktionalitäten noch weitere Hilfestellungen wünschenswert beziehungsweise erforderlich sind. Außerdem soll dadurch herausgefunden werden, ob die Nutzer die unterschiedlichen Optionen, beispielsweise zum Löschen eines Modules, wahrnehmen oder welche Methode dafür bevorzugt wird.

Nach dem praktischen Teil folgte der Fragenbogen zur sechsten absolvierten Pilotstudie. Dieser bestand aus Single Choice sowie offenen Fragen. Die Fragen beschäftigten sich neben allgemeinen Daten zur Person hauptsächlich mit der Zufriedenheit des Benutzers während des Arbeitens mit dem Web-Modul-Editor. Diese dienen dazu, herauszufinden ob der Editor trotz fehlender Einführung in das Tool verständlich und selbsterklärend war. Ebenso wurde dadurch nützliches Feedback gewonnen, ob an gewissen Funk-

Tabelle 5.1: Aufgabenstellung der Pilotstudie.

<i>Aufgabe</i>	<i>Zeitvorgabe</i>
Der Workshop dauert insgesamt fünf Stunden.	10:00 – 15:00 Uhr
Ein Modul zum Kennenlernen der Teilnehmer untereinander. Jede Person sollte seine Herkunft kurz erklären.	10 Minuten
Eine Präsentation zur Einführung des Themas.	20 Minuten
Danach wird ein Modul zur Ideenfindung benötigt. Es folgt ein Brainstorming im Plenum um Denkanstöße zum soeben präsentierten Thema zu finden.	10 Minuten
Eine Pause ist erforderlich.	10:40 – 10:55 Uhr
Nach der Pause sollten die Teilnehmer zehn verschiedene Ideen zum besprochenen Thema generieren. Dieser Ideenfindungsprozess sollte als Einzelarbeit durchgeführt werden.	45 Minuten
Aus den generierten Ideen sucht sich nun jeder Teilnehmer selbst seinen Favoriten aus.	10 Minuten
Aus den eigenständig gewählten Favoriten wird danach im Plenum abgestimmt und eine Idee ausgewählt.	10 Minuten
Es folgt die fixe Mittagspause, die auf jeden Fall eingehalten werden muss!	12:00 – 13:00 Uhr
Nach der Mittagspause folgt wieder eine Präsentation zu einem neuen Thema.	20 Minuten
Aufgrund der zuvor gehaltenen Präsentation müssen die Teilnehmer in Gruppen Persona generieren.	30 Minuten
Im Plenum soll danach über die Persona abgestimmt werden.	20 Minuten
Auf Grundlage der abgestimmten Persona sollen nun in der Gruppe drei Variationen dazu generiert werden.	30 Minuten
Danach gibt es eine erneute Abstimmung im Plenum.	5 Minuten
Zum Schluss werden noch einmal alle wichtigen Abstimmungspunkte und Entscheidungen zusammengefasst sowie die nächsten Schritte besprochen.	15 Minuten

Tabelle 5.2: Änderungstabelle der Pilotstudie.

<i>Änderungen</i>
Die kurze Pause am Vormittag kann frei gestaltet werden und muss daher nicht mit einberechnet werden. Sie ist dem Workshop zu entfernen und die übrigen 15 Minuten werden zur individuellen Ideenfindungszeit addiert.
Es wird geraten anstatt des Brainstorming Modules ein Diskussionsmodul zu wählen. Dieses Modul gehört ersetzt.
Am Schluss muss der Workshop abgespeichert werden.

tionalitäten noch subtile Hinweise zur Verdeutlichung notwendig sind, ob gelegentlich schon zu viele Hilfestellungen enthalten oder diese als störend empfunden werden. Der gesamte Fragebogen ist in Anhang B einzusehen.

5.1.2 Aufbau und Ablauf der Test-Session

Testpersonen

Es wurden sieben Personen im Alter von 23 bis 28 Jahren ($M = 24,86$; $SD = 1,95$) ausgewählt, davon sind fünf Kandidaten weiblich und zwei männlich. Bei vier Probanden handelt es sich um Studenten im Bereich Informatik. Die anderen drei sind im technischen Bereich als Webdeveloper, Softwareentwickler und Digital Design Engineer tätig. Zwei der vier Studenten absolvieren eine technische Ausbildung. Sechs der sieben Testkandidaten sind übliche Kreativitätstechniken und Ideenfindungsmodule geläufig, auch wenn sie beruflich und privat nur wenig mit der Planung von Workshops zu tun haben. Die Studie wurde mit keinen Moderatoren durchgeführt, da in erster Linie die Interaktionen unerfahrener Nutzer beobachtet werden sollten.

Testumgebung und Bewegungsaufzeichnung

Die Pilotstudie wurde mit vier Kandidaten in einem ruhigen Raum in den Örtlichkeiten der Fachhochschule Hagenberg durchgeführt. Die anderen drei Studien fanden aus organisatorischen Gründen in einer leisen, privaten Räumlichkeit in Linz statt.

Alle Personen führten die Pilotstudie am gleichen Testgerät durch. Dieses ist ein MacBook Pro (13 Zoll, Mitte 2012) mit dem Betriebssystem OS X El Capitan. Als Browser zum Ausführen des Web-Modul-Editors diente bei allen Testkandidaten Google Chrome, Version 58.0.3029.110 (64-bit).

Die Studie wurde mit jeder Person einzeln und nacheinander durchgeführt. Zur Bewegungsaufzeichnung der Interaktionen durch die Testpersonen

wurde das Online-Tool *mouseflow* verwendet, das unter anderem das Tracken der Mausbewegungen, Klicks und Scrolls ermöglicht [22].

5.1.3 Ergebnisse

Quantitative vs. Qualitative Forschung

Die quantitative Forschung wird meist mit einer großen Anzahl von Personen durchgeführt. Dadurch sollen allgemein gültige Messwerte erzielt werden, die generalisiert werden und zur Repräsentation fungieren. Quantitative Forschung dient dazu eine Hypothese zu bestätigen oder diese zu widerrufen. Methoden dafür sind ein schriftlicher Fragebogen mit vordefinierten Antwortmöglichkeiten oder eine quantitative Befragung. Damit die Daten vergleichbar werden, bekommt jeder Teilnehmer die gleichen Fragen und Antwortmöglichkeiten [13, 25].

Qualitative Forschung wird normalerweise mit einer eher kleineren Anzahl an Personen durchgeführt. Sie ist wesentlich flexibler, da es möglichst keine standardisierten Vorgaben gibt und auf jeden einzelnen Teilnehmer eingegangen wird. Dadurch ist die Streuung der Ergebnisse erheblich größer und diese können nicht einfach, repräsentativ zusammengefasst werden. Qualitative Forschung bringt einen tieferen Einblick an Informationen und ist ideal um Verbesserungsvorschläge sowie Erweiterungsmöglichkeiten zu sammeln. Sie wird oft zur Aufstellung von Hypothesen verwendet und ist daher eine sinnvolle Methode zur Bildung einer Grundlage für quantitative Forschung. Varianten der qualitativen Forschung sind zum Beispiel Gruppendiskussionen oder qualitative Interviews [13, 25].

Die vorliegende Pilotstudie erzielte sowohl quantitative als auch qualitative Ergebnisse. Der Fragebogen enthält neben einigen vordefinierten, quantitativen Single Choice Fragen auch sehr viele offene Fragen, die eher zu qualitativen Ergebnissen führten. Die praktische Aufgabenstellung hat einen quantitativen Charakter, da sie für jede Testperson gleich war. Durch die, an die Think-Aloud-Methode angelehnte, Vorgehensweise während der Durchführung des Tests, erwähnt unter Abschnitt 5.1.1, konnte später aber individuell auf jeden Teilnehmer eingegangen und dessen Problematiken kurz besprochen werden. Dies führte zu weiteren qualitativen Verbesserungsvorschlägen.

Grundsätzliches Empfinden

Abbildung 5.2 zeigt die Ergebnisse des Fragebogens zum Thema Schwierigkeitsgrad der Bedienung des Web-Modul-Editors. Es ist zu sehen, dass lediglich die Optionen *sehr einfach*, mit 57,14 %, und *einfach*, mit 42,86 %, angegeben wurden. Keiner der Testkandidaten wählte die Optionen *geht so*, *eher schwierig* oder *schwierig*. Die Ergebnisse zeigen, dass die Probanden die Bedienung zum größten Teil von Anfang an verständlich empfanden. Dies ist

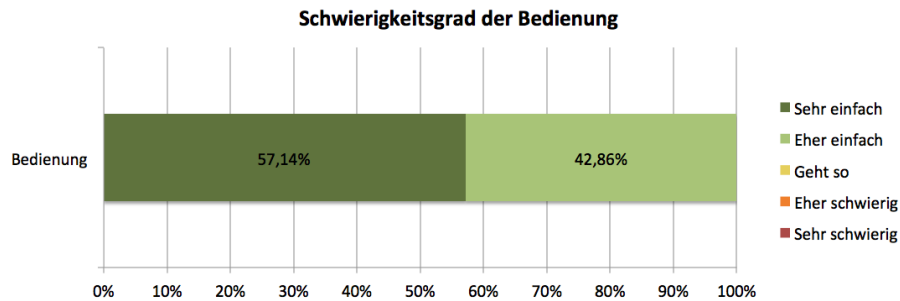


Abbildung 5.2: Ergebnisse der Befragung bezüglich Schwierigkeitsgrad der Bedienung des Web-Modul-Editors.

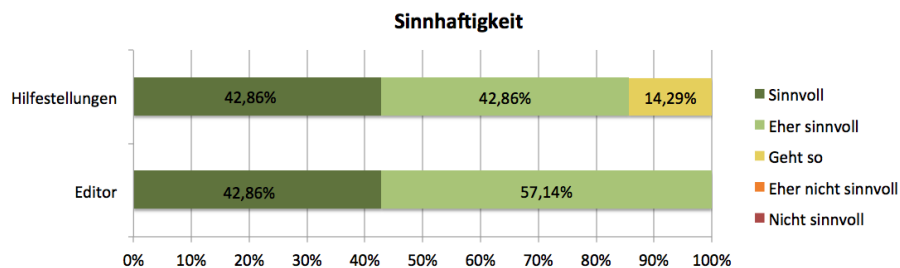


Abbildung 5.3: Ergebnisse der Befragung zur Sinnhaftigkeit der Hilfestellungen und des Editors im Allgemeinen.

bereits ein gutes Zeichen, da trotz fehlender Einführung keine erheblichen Probleme beim Bedienen des Interfaces aufgetreten sind.

In Abbildung 5.3 ist zu sehen, dass alle Kandidaten den Einsatz des Editors zur Planung des Workshops für nützlich empfinden. 42,86 % der Probanden stimmten für *sinnvoll* und 57,14 % für *eher sinnvoll*. Dies zeigt, dass die Bereitschaft, das Tool durch subtile Hinweise zu erlernen, vorhanden ist und weniger Gefahr besteht, dass die Benutzer bereits nach kürzester Zeit abspringen, da sie keinen Nutzen in der Verwendung des Tools sehen.

Die subtilen Hilfestellungen zum Erlernen des Web-Modul-Editors wurden zu gleichen Teilen, mit 42,86 %, als *sinnvoll* und *eher sinnvoll* empfunden. Eine Person, und damit 14,29 %, vermerkte, dass sie unschlüssig ist und wählte daher die Option *geht so*. Somit ist zu verzeichnen, dass der Großteil der befragten Personen die Hilfestellungen als wertvoll beurteilten.

Die Ergebnisse des Fragebogens zum Umfang der subtilen Hilfestellungen im Web-Modul-Editor sind in Abbildung 5.4 ersichtlich. Hierbei gab der Großteil der Probanden, mit 71,43 %, an, dass er genau richtig beziehungsweise angemessen war. 14,29 %, damit eine von sieben Personen, empfand

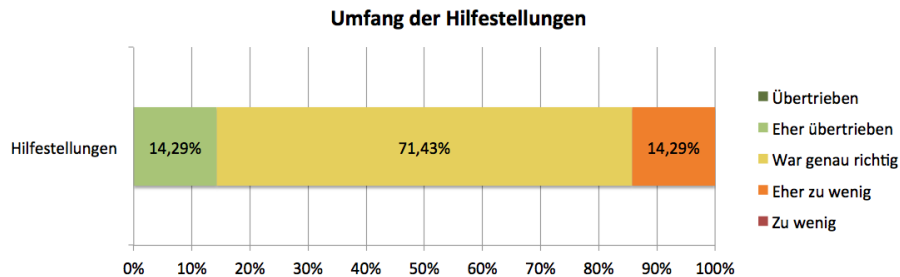


Abbildung 5.4: Ergebnisse der Befragung zum Umfang der subtilen Hilfestellungen im Web-Modul-Editor.

manche Hinweise als *eher übertrieben*. Ebenso eine von sieben Personen, 14,29 % wählte die Option *eher zu wenig*, wobei konkret der Wunsch nach einer besseren Visualisierung der Ersetzungsmöglichkeit geäußert wurde. Zusammenfassend ist daher zu sagen, dass die Hilfestellungen zu mehr als 85 % als hilfreich und nicht störend empfunden wurden.

Einsatz subtiler Hinweise

Nur vier Testpersonen gaben an, dass sie die unterschiedlich visualisierten Verbindungsarten für die Ein- und Ausgangsoptionen der Module wahrgenommen haben. Keine der Personen konnte jedoch den exakten Nutzen beziehungsweise die Bedeutung dieser Verbindungstypen auf Anhieb erkennen. Auf Nachfrage, was diese bedeuten könnten, kamen zwei der vier Testkandidaten auf eine beinahe korrekte Lösung.

*Vielleicht soll es die Anzahl der Ideen darstellen oder den Verlauf der Ideenfindung.*¹

Dies zeigt, dass die Visualisierung der Ein- und Ausgangsverbindungen zumindest für unerfahrene Benutzer keinerlei Nutzen mit sich bringt, der Hinweis könnte durchaus vernachlässigt werden. Da es jedoch auch nicht als störend empfunden wurde und dieser Hinweis möglicherweise für erfahrene Benutzer von Bedeutung sein könnte, wäre es sinnvoll, auf Nachfragen des Nutzers eine Beschreibung im System zu liefern.

Erkenntnis aufgrund der aufgezeichneten Interaktionen

Alle Kandidaten schafften es die Aufgabenstellung innerhalb der zur Verfügung stehenden Zeit von zwanzig Minuten zu vollenden. Der schnellste

¹Kommentar von Teilnehmer 3

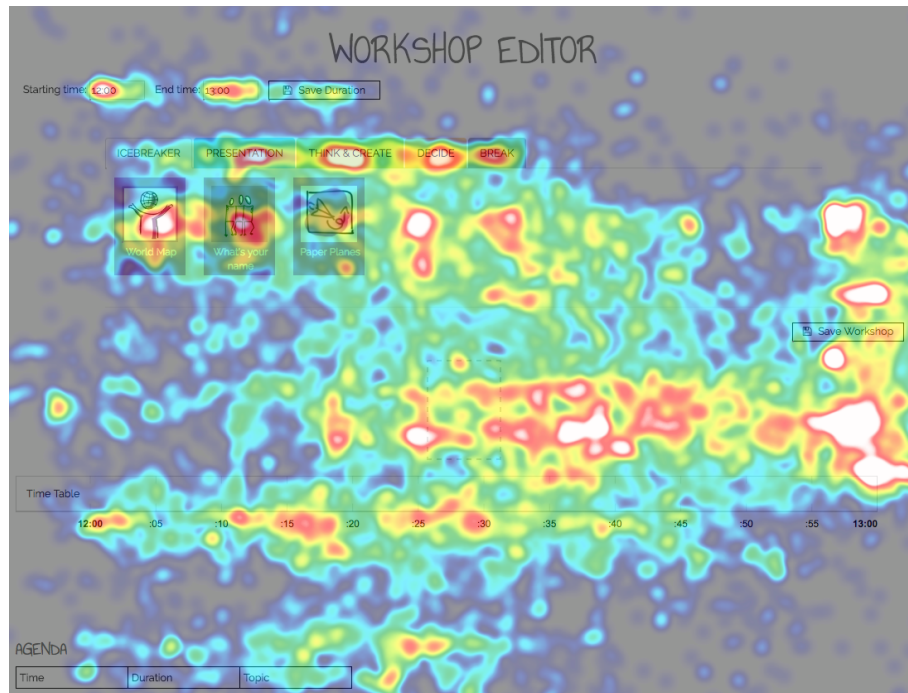


Abbildung 5.5: Mausbewegungen und Klicks der Teilnehmer bei der Pilotstudie.

Kandidat hatte die Aufgabe innerhalb von elf Minuten und siebenundzwanzig Sekunden abgeschlossen, der langsamste Kandidat schaffte es innerhalb von achtzehn Minuten und fünfundfünfzig Sekunden ($M = 13:45$; $SD = 2:42$).

Aufgrund der mittels *mouseflow* aufgezeichneten Daten wurde eine Heatmap erstellt. Die generierte Map, ersichtlich in Abbildung 5.5, zeigt eine Übersicht aller Mausbewegungen und Klicks der sieben Testkandidaten². Wie in dieser Abbildung ersichtlich, fokussieren sich die meisten Interaktionen auf der Toolbox, der die zu verwendenden Module enthält, die Arbeitsfläche, auf der die Module positioniert werden, sowie die Einstellungsoberfläche. Diese ist in Abbildung 5.5 zwar nicht ersichtlich, da sie ausgeblendet ist, jedoch sind die Interaktionen im rechten Bereich des Bildschirms deutlich erkennbar. Überraschend ist, dass auch in der Zeitlinie einige Klicks erfolgten.

Abbildung 5.6 zeigt die Verwendung der unterschiedlichen Optionen zum Löschen und Ersetzen von Modulen. Es ist zu sehen, dass der Großteil der Kandidaten, nämlich 71,43 %, die Hover-Schaltfläche zum Löschen ei-

²Es ist lediglich eine kumulierte Aufzeichnung der Mausbewegungen aller Personen ersichtlich, das das Tool keine Heatmaps pro Person zulässt.

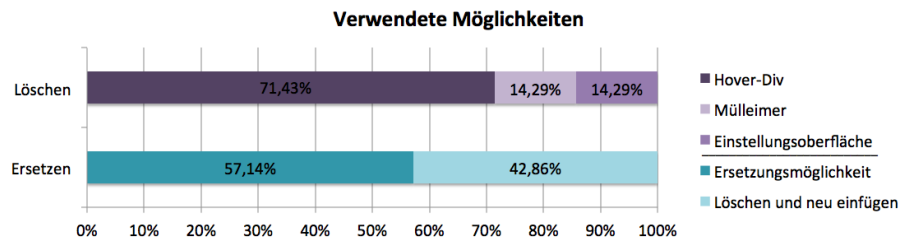


Abbildung 5.6: Ergebnisse der aufgezeichneten Interaktionen zur Verwendung der unterschiedlichen Möglichkeiten.

nes Modules verwendet. Die anderen beiden Optionen, das Löschen mittels Mülleimer-Icon und das Löschen über die Einstellungsfläche, wurden zu gleichen Teilen mit jeweils 14,29 % gewählt. Bei dem Kandidaten, der das Mülleimer-Icon verwendete, war zu beobachten, dass diese Möglichkeit zufällig zuvor schon entdeckt und danach zweimal bevorzugt wurde. Diese Möglichkeit ist zwar etwas versteckt aber vermutlich die schnellste und einfachste Methode zum Löschen eines Modules. Möglicherweise findet es bei geübten Benutzern, die bereits über die Existenz des Icons bescheid wissen, mehr Anklang.

Bezüglich des Ersetzens eines Modules ist die Auswahl der verwendeten Möglichkeiten relativ ausgeglichen. 57,14 % der Probanden entdeckten bereits die Ersetzungsmöglichkeit durch das Darüberziehen eines neuen Elementes über ein altes. Die restlichen 42,86 % löschten erst das alte Modul und fügten danach an der gleichen Stelle das neue Modul wieder ein. Es ist denkbar, dass auch die 42,86 % sich für die Ersetzungsmöglichkeit entscheiden würden, wenn sie nach längerer Interaktion mit dem Web-Modul-Editor diese Funktionalität entdecken.

Erkenntnis aufgrund unterschiedlicher Monitorgrößen

Zur Entwicklung des Web-Modul-Editors wurde ein 27-Zoll-Monitor verwendet, die Anpassungsfähigkeit für andere Bildschirmgrößen wurden durch Verändern der Fensterbreite implementiert. Bei der Durchführung des Pilottests wurde auf einem 13-Zoll-Monitor gearbeitet und dies brachte einige Änderungen mit sich, da die Anpassungsfähigkeit der Fensterhöhe außer Acht gelassen wurde.

Auf dem 27-Zoll-Monitor sind initial alle Bereiche des Editors, inklusive der Zeitplanungskomponenten, auf einen Blick ersichtlich. Wird die Workshopzeit mithilfe der Formularfelder im oberen Bereich des Monitors festgelegt, so spiegelte sich die visuelle Rückmeldung direkt am Zeitstrahl wieder. Dadurch erfährt der Benutzer sofort, dass die Eingabe erfolgreich war. Auf dem 13-Zoll-Monitor, der während der Pilotstudie verwendet wurde,

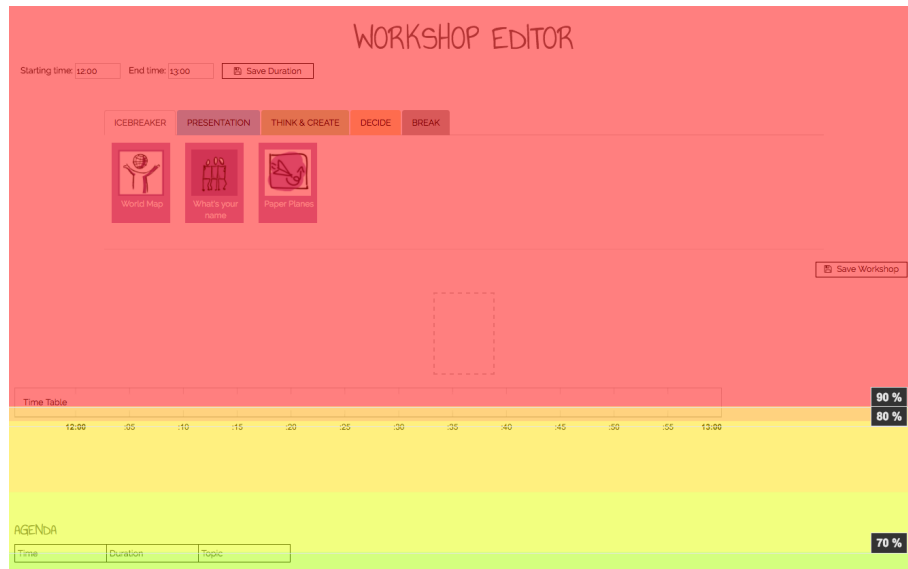


Abbildung 5.7: Scrollbereich der Teilnehmer bei der Pilotstudie.

ist die Zeitplanungskomponente erst durch Scrollen erkennbar. Dadurch gab es beim Abspeichern der Workshopzeit keine direkte Rückmeldung, wie es Abschnitt 2.2.3 erläutert, wodurch des Öfteren eine kurze Verwirrung der Teilnehmer beobachtet werden konnte.

Zudem war zu verzeichnen, da der Bildschirmbereich wesentlich kleiner war, die Zeitplanungskomponenten nur bedingt wahrgenommen wurde. Abbildung 5.7 zeigt den aufgezeichneten Scrollbereich der Teilnehmer. Es ist zu erkennen, dass der direkt auf dem Bildschirm ersichtliche Bereich (rot eingezeichnet) zu 100 % wahrgenommen wurde. Der Bereich der Zeitlinie (orange eingezeichnet) war noch zu zirka 90 – 100 % zu erfassen und der Abschnitt der Agenda (gelb bis hellgrün eingezeichnet) wurde lediglich von rund 65 – 70 % der Teilnehmer registriert und beachtet.

Dieses Ergebnis deckt sich auch mit den Erhebungen mittels Fragebogen. Alle Kandidaten gaben an, dass sie die Zeitlinie wahrgenommen haben und diese auch zur Planung oder Kontrolle sehr hilfreich war. Bei der Agenda gaben drei der sieben Testpersonen an, dass sie diese gar nicht oder erst nach einiger Zeit registriert haben. Zur Planung des Workshops wurde die Agenda von keiner Person verwendet, was jedoch darauf zurückzuführen ist, dass die zu verwendenden Module zeitlich genau vorgegeben wurden. Zudem ist der Hauptnutzen der Agenda erst nach der Planung, nämlich als Basis des Ablaufes für die Teilnehmer, wesentlich.

5.1.4 Verbesserungs- und Erweiterungsmöglichkeiten

Verdeutlichung durch zusätzliche Tooltips

Aufgrund der Problematik, dass einige Testkandidaten bei der Unterscheidung der Arbeitsmodi unschlüssig waren, gab es den Vorschlag, die jeweiligen Typen in Form eines Tooltips zu beschreiben. Diese Beschreibung könnte neben der grundsätzlichen Erklärung auch Beispiele umfassen, da es bei manchen Modulen zu Verwirrungen kam.

*Ich tu mir schwer zwischen Plenum und Gruppe zu unterscheiden. Außerdem weiß ich nicht ob Präsentation jetzt Plenum oder Individual ist, da ja nur eine Person wirklich präsentiert.*³

Zudem gab es von einem Kandidaten den Wunsch, dass auch die Checkbox zum Fixieren von Elementen mit einer Tooltip-Beschreibung versehen wird. Auch dieser Anregung wird nachgegangen, da es zu beobachten war, dass auch die anderen Teilnehmer den Sinn der Checkbox erst verstanden, als sie wirklich angewendet werden musste.

Diese Erweiterungen wurden direkt im Einstellungsmenü vorgenommen, indem die drei Arbeitsmodi sowie die Checkbox zum fixieren der Elemente jeweils mit einer Tooltip-Beschreibung versehen wurden. Die Tooltips selbst wurden mittels vorgesehener Funktion von *Bootstrap* implementiert.

Verbesserungsvorschläge aufgrund der aufgezeichneten Interaktionen

Wie bereits unter Abschnitt 5.1.3 erwähnt und in Abbildung 5.5 ersichtlich, wurden einige ziellose Klicks auf die Module in der Zeitlinie durchgeführt. Daraus ist zu schließen, dass eine weitere Funktionalität im Zeitstrahl zu empfehlen wäre. Durch Klicken auf ein Modul im Strahl ist ein Öffnen der jeweiligen Einstellungsoptionen im rechten Bereich des Bildschirms zu empfehlen. Da diese Erweiterung einen größeren Implementierungsaufwand mit sich bringt, wurde die Änderung noch nicht realisiert. Für einen möglichen Ausbau des Editors ist sie aber durchaus vorzumerken.

Verbesserung zur Berücksichtigung unterschiedlicher Monitorgrößen

Um der Problematik aufgrund unterschiedlicher Monitorgrößen entgegen zu wirken, der unter Abschnitt 5.1.3 erwähnt wurden, empfehlen sich folgende möglichen Verbesserungen.

Beim Festlegen der Workshopzeit wird zusätzlich zur Anpassung des Zeitstrahls ein Icon in Form eines Häkchens direkt neben den zu klickenden Button gesetzt. Dieses erscheint, sobald der Button zum Abspeichern vom Benutzer geklickt wurde und macht dem Benutzer deutlich, dass die Eingabe erfolgreich war.

Als weiteren Verbesserungsvorschlag wurde von einem Kandidaten erwähnt, dass ein Scroll-Down-Button hilfreich wäre. Dieser sollte bereits im ersten Teil, der noch auf dem Bildschirm ersichtlich ist, einen Hinweis auf weiter unter liegende Bereiche geben.

³Kommentar von Teilnehmer 5

*Ich habe erst nicht gesehen, dass es unten noch weiter geht. Vielleicht kann man die Agenda irgendwie hervorheben oder dass es oben einen Hinweis darauf gibt. Eventuell einen Button.*⁴

Jene Hilfestellung wurde durchaus für sinnvoll befunden und deshalb auch in den Web-Modul-Editor integriert.

⁴Kommentar von Teilnehmer 2

Kapitel 6

Schlussbemerkungen

6.1 Zusammenfassung

Aufgrund der immer größer werdenden Anzahl an Webseiten und Online-Anwendungen wird das Thema Benutzerfreundlichkeit und Benutzererfahrung immer wichtiger. Ist ein Benutzer nicht nach kürzester Zeit mit einem neuen System vertraut, so ist die Bereitschaft jenes erst zu erlernen sehr gering. Arbeitsabläufe sollten daher selbsterklärend, einfach und mittels wenigen Instruktionen bewältigt werden können. Die Schwierigkeit besteht zusätzlich darin, unerfahrene Nutzer nicht mit zu vielen Optionen in einem komplexen Arbeitsablauf zu überfordern, aber dennoch erfahrenen Benutzern eigenwillige Schritte zu erlauben.

Webanimation können hilfreich bei der Umsetzung subtiler Hinweise für den Benutzer sein, wodurch der Anwender das Programm direkt beim praktischen Einsatz unterbewusst geleitet wird. Bei der Verwendung von Animationen, besonders bei dekorativen Bewegungen, ist allerdings Vorsicht geboten. Haben Animationen keinerlei Nutzen für den Benutzer, so können diese schnell als überflüssig und störend empfunden werden. Funktionale Animation hingegen kann, richtig eingesetzt, die Interaktion mit dem System unterstützen und dabei erheblich die Benutzerfreundlichkeit der Schnittstelle steigern.

Animationen sollen so konzipiert sein, dass sie unauffällig und elegant sowie reaktionsschnell und modern erscheinen [14]. Sie sollten einen realistischen, familiären Charakter und keine unerwarteten Übergänge vorweisen. Es ist wichtig, dass für jede Aktion des Benutzers eine visuelle Rückmeldung existiert, damit der Benutzer weiß, dass das System die Aktivität registriert hat. Zudem muss gewährleistet werden, dass dem Benutzer zu jedem Zeitpunkt das Gefühl vermittelt wird, dass er weiß was das System gerade macht.

Es existieren unterschiedliche Arten von Hilfesystemen. Herzog unterteilt dabei in aktive und passive, statische und dynamische, einheitliche und individuelle sowie synchrone und asynchrone Hilfestellungen [4, S. 153 ff.].

Beispiele für subtile Hinweise sind textuelle Hinweise, Gitternetzlinien, Hilfslinien, Smart Guides, Snapping, Farbwechsel sowie erweitertes Drag & Drop.

Im praktischen Teil der Arbeit wurde ein Konzept für einen Web-Modul-Editor erstellt und daraufhin implementiert. Dieser Editor bietet Moderatoren die Möglichkeit Workshops zu planen und zugleich einen Zeitplan als Basis für die Agenda zu generieren. Die Webanwendung sollte selbsterklärend und durch den Einsatz von subtilen Hinweisen einfach und rasch zu bedienen sein. Es wurden drei Entwürfe sowie subtile Animationen für unterschiedliche Hilfestellungen konzipiert. Dabei wurde auch berücksichtigt, dass die Online-Anwendung sowohl für erfahrene Benutzer als auch Anfänger geeignet sein soll.

Zur Realisierung des Konzeptes fanden vier Bibliotheken ihre Anwendung. *interact.js* ist für die Umsetzung des Drag & Drop-Mechanismus verantwortlich. *Google Charts Timeline* wurde zur Visualisierung der Zeitlinie eingesetzt und *AlertifyJS* sowie *Bootstrap* wurden zu gestalterischen und responsiven Zwecken implementiert.

Der Web-Modul-Editor ist in drei Teilen aufgebaut. Der Toolbox, die die Module zum Zusammensetzen des Workshops enthält, der Arbeitsfläche, die das Kernstück des Editors darstellt, sowie der Zeitplanungskomponente, die einen Zeitstrahl zur Orientierung für den Benutzer und die Agenda zur weiteren Verwendung umfasst.

Zur Evaluierung der subtilen Hinweise im Web-Modul-Editor wurde eine Pilotstudie mit unterschiedlichen Teilnehmern durchgeführt. Die Testkandidaten wurden gebeten einen Tagesworkshop mithilfe des Web-Modul-Editors zu realisieren, wobei die konkreten Schritte und Module vorgegeben wurden. Im Anschluss folgte eine kurze Erhebung mittels Fragebogen. Die Pilotstudie brachte interessante Ergebnisse und Verbesserungsvorschläge, die zum größten Teil direkt noch umgesetzt wurden.

6.2 Resümee

Zur Umsetzung eines selbsterklärenden Tools haben sich subtile Animationen als Hilfestellungen sowohl angeboten als auch als hilfreich und effizient erwiesen. Dies stellte sich bereits bei der Implementierung des Editors heraus und war auch im Feedback der Probanden in der Evaluierungsphase enthalten. Je nach Anforderung und gewünschtem Implementierungsaufwand können die Hilfen frei gewählt werden, was eine große Bandbreite an Einsatzmöglichkeiten bietet.

Es stellte sich heraus, dass selbst in einem frühen Stadium der Entwicklung Benutzertests und Pilotstudien durchaus empfehlenswert sind. Durch iterative Evaluierungsphasen werden Schwachstellen und Erweiterungsmöglichkeiten aufgezeigt, worauf zu einem frühen Zeitpunkt noch flexibel rea-

giert werden kann.

Werden bereits vorhandene Bibliotheken oder vorgefertigte Elemente eingesetzt, so stellt sich die Anpassungsfähigkeit meist als eine Herausforderung dar. Die unter Abschnitt 5.1.4 in Kapitel 5 erwähnte Erweiterung des Zeitstrahles würde unter Verwendung der *Google Charts* Bibliothek leider nicht möglich sein, da diese Funktionalität nicht vorgesehen ist und es keinerlei Zugriffsmöglichkeiten gibt. Die Realisierung dieser nützlichen Erkenntnis würde daher einen großen Änderungsaufwand mit sich bringen.

6.3 Ausblick

Basierend auf den Ergebnissen der Pilotstudie wurde aufgezeigt, dass durch subtile Hinweise Wissen vermittelt werden kann, das unerfahrene Nutzer möglicherweise noch nicht haben. In diesem Fall war es neben der Steuerung der Online-Anwendung auch die valide Komposition eines Workshops. Es wäre denkbar, dass im heutigen Zeitalter, wo sich jeder über die Einsparung des Budgets für Einschulungen erfreut, komplexe Abläufe und Tools lediglich durch subtile Hinweise erlernbar werden.

In der vorliegenden Arbeit wurde nicht auf den Einsatz in Touch-Geräten geachtet, da es sich ausschließlich um eine Desktop-Anwendung handelt. Die Anpassung des Web-Modul-Editors für alle gängigen Geräte würde mit Sicherheit noch weiteren Implementierungsaufwand sowie das Überdenken mancher Hinweise mit sich bringen. Da dies von Anfang an nicht vorgesehen war, wurde bei der Konzeption der subtilen Hilfestellungen nicht darauf geachtet. Sollte der Wunsch bestehen subtile Hinweise auf allen Geräten verwendbar zu machen, so ist es empfehlenswert dies bereits bei der Konzeption und Auswahl der Hilfestellungen zu beachten.

Anhang A

*.json-Datei

```
1 var myJSON = '[
2 {"id":"1","title":"empty","description":"empty","type":"empty",
   work_mode":"plenum","duration":0,"input":"*","output":"*","tags":
   null,"image_url":"img/empty.png"},
3 {"id":"2","title":"World Map","description":"Every participant indicates
   on an interactive map his country of origin (or his next holiday
   destination, his favorite city, ...).","type":"icebreaker",
   work_mode":"plenum","duration":"5","input":"0","output":"0","tags":
   null,"image_url":"img/world_map.png"},
4 {"id":"3","title":"What's your name","description":"For groupfinding
   purpose, every participant writes his name in one of the boxes.
   Boxes are shuffled, the result are the work groups.","type":"
   icebreaker","work_mode":"plenum","duration":"5","input":"0","output
   ":"0","tags":null,"image_url":"img/whats_your_name.png"},
5 {"id":"4","title":"Presentation","description":"Presentation of prepared
   workshop materials (e.g., slideshows, pictures, websites, videos
   ,...), or ideas/concepts created within the workshop.","type":"
   presentation","work_mode":"plenum","duration":"15","input":"0",
   output":"0","tags":null,"image_url":"img/presentation.png"},
6 {"id":"5","title":"User Map","description":"For a given product/solution
   , participants are asked to write down possible target user groups
   .","type":"thinkcreate","work_mode":"group, individual","duration
   ":"10","input":"?","output":"*","tags":null,"image_url":"img/
   user_map.png"},
7 {"id":"6","title":"Persona","description":"For a given product/solution,
   participants are asked to sketch out a fictive persona and their
   personal characteristics, needs, goals, thoughts,...","type":"
   thinkcreate","work_mode":"group, individual","duration":"20","input
   ":"?","output":"+", "tags":null,"image_url":"img/persona.png"},
8 {"id":"7","title":"Prototype","description":"For a given idea,
   participants create a physical prototype using various prototyping
   materials (e.g., LEGO, paper, clay,...).","type":"thinkcreate",
   work_mode":"group, individual","duration":"30","input":"0","output
   ":"?","tags":null,"image_url":"img/prototype.png"},
9 {"id":"8","title":"Discussion","description":"In a meeting, participants
   can sketch out the most important discussion points and decisions
   .","type":"thinkcreate","work_mode":"group","duration":"10","input
```

```

    ":"*","output":"+","tags":null,"image_url":"img/discussion.png"},
10 {"id":"9","title":"Brainstorming","description":"For a problem/topic,
    participants are brainstorming a multitude of ideas on post-it notes
    .","type":"thinkcreate","work_mode":"group, individual, plenum","
    duration":"5","input":"?","output":"*","tags":null,"image_url":"img/
    brainstorming.png"},
11 {"id":"10","title":"10 Ideas","description":"For a given problem/topic,
    participants are asked to create 10 diverse concepts.","type":"
    thinkcreate","work_mode":"individual, group","duration":"45","input
    ":"?","output":"*","tags":null,"image_url":"img/10_ideas.png"},
12 {"id":"11","title":"3 Variations","description":"For a given idea,
    participants are asked to create three variations.","type":"
    thinkcreate","work_mode":"group, individual","duration":"20","input
    ":"1","output":"3","tags":null,"image_url":"img/3_variations.png"},
13 {"id":"12","title":"Main Problem","description":"For a given persona/
    company/identity, participants are asked to define a main problem to
    be solved.","type":"decide","work_mode":"group, individual","
    duration":"15","input":"0","output":"1","tags":null,"image_url":"img
    /main
14 {"id":"13","title":"Goal Definition","description":"In a meeting,
    participants are asked to define the main goal/purpose.","type":"
    decide","work_mode":"plenum, group, individual","duration":"10","
    input":"0","output":"1","tags":null,"image_url":"img/goal_definition
    .png"},
15 {"id":"14","title":"Wrap-up","description":"In a meeting, participants
    are asked to wrap-up the most important discussion points &
    decisions, as well as next steps in terms of to-do's.","type":"
    decide","work_mode":"plenum, group","duration":"15","input":"*","
    output":"+","tags":null,"image_url":"img/wrap_up.png"},
16 {"id":"15","title":"Voting","description":"For a given number of ideas,
    participants are asked to determine their favorite(s) by means of
    voting.","type":"decide","work_mode":"group, plenum, individual","
    duration":"5","input":"*","output":"*","tags":null,"image_url":"img/
    voting.png"},
17 {"id":"16","title":"Pick your Favorite","description":"For a given
    number of ideas, participants are asked to choose one favorite.","
    type":"decide","work_mode":"group, plenum, individual","duration
    ":"5","input":"*","output":"1","tags":null,"image_url":"img/
    pick_your_favorite.png"},
18 {"id":"17","title":"Summarize","description":"In a workshop,
    participants are asked to summarize their concept/idea, and indicate
    items that are needed for its implementation.","type":"decide","
    work_mode":"group, individual","duration":"15","input":"1","output
    ":"+","tags":null,"image_url":"img/summarize.png"},
19 {"id":"18","title":"Break","description":"Break","type":"break","
    work_mode":"group, individual, plenum","duration":"15","input":"0","
    output":"0","tags":null,"image_url":"img/break.png"},
20 {"id":"19","title":"Paper Planes","description":"In groups of two,
    participants build paper planes, with their non-dominant hand,
    within a time limit of two minutes.","type":"icebreaker","work_mode
    ":"group","duration":"5","input":"0","output":"0","tags":null,"
    image_url":"img/paper_plane.png"}]';

```

Anhang B

Fragebogen Pilotstudie

Fragebogen Web-Modul-Editor

Teilnehmer: ____

1. Wie alt sind Sie?

2. Welches Geschlecht haben Sie?
 Weiblich Männlich
3. Welchen Beruf haben Sie?

4. Haben Sie beruflich oder privat mit der Planung von Seminaren und Workshops zu tun?
 Ja Nein
5. Sind Ihnen übliche Kreativitätstechniken und Ideenfindungsmodule geläufig?
 Ja Nein
6. Wie einfach fanden Sie die Bedienung des Web-Modul-Editors?
 Sehr einfach Eher einfach Geht so Eher schwierig Sehr schwierig
7. Fanden Sie den Editor von Anfang an verständlich?
 Ja Nein
8. Wenn nein: Fanden Sie den Editor nach einer Weile verständlich?
 Ja Nein
9. Was haben Sie nicht von Anfang an verstanden?

10. Was könnte man Ihrer Meinung nach beim Web-Modul-Editor verbessern?

11. Finden Sie den Editor sinnvoll zur Planung von Workshops?
 Ja Eher ja Ich weiß nicht so recht Eher nein Nein
12. Haben Sie die Zeitlinie wahrgenommen?
 Ja Nein
13. War die Zeitlinie nützlich zur Planung des Workshops?
 Ja Nein Bedingt weil: _____
14. Haben Sie die Agenda im unteren Teil wahrgenommen?
 Ja Nein
15. Haben Sie die Agenda im unteren Teil zur Planung verwendet?
 Ja Nein Bedingt weil: _____
16. Haben Sie die unterschiedlichen Verbindungsarten der Module wahrgenommen?
 Ja Nein
17. Glauben Sie die Bedeutung der unterschiedlichen Verbindungsarten verstanden zu haben?
 Ja Nein
18. Wenn ja: Was glauben Sie welche Bedeutung diese haben?

19. Fanden Sie die subtilen Hilfestellungen sinnvoll?
 Ja Eher ja Ich weiß nicht so recht Eher nein Nein
20. Fanden Sie manche der subtilen Hilfestellungen übertrieben oder zu viel?
 Ja Eher ja War genau richtig Nein, eher zu wenig Nein, zu wenig
21. Wenn ja: Welche Hilfestellungen waren übertrieben oder zu viel?

22. Welche Hilfestellungen können Ihrer Meinung nach getrost weggelassen werden?

23. Welche Hilfestellungen sollten Ihrer Meinung nach noch eingebaut werden?

24. Wobei hätten Sie noch eine Hilfestellung benötigt?

25. Wofür könnte der Modul-Editor möglicherweise noch verwendet werden, außer zur Planung von Workshops? (gern auch mit zusätzlichen Erweiterungen und Änderungen?)

26. Möchten Sie sonst noch etwas sagen?

Herzlichen Dank für die Teilnahme! :)

Anhang C

Inhalt der CD-ROM

C.1 PDF-Dateien

Pfad: /

Loidl_Eva_2017.pdf . . . Masterarbeit (Gesamtdokument)

C.2 Online-Quellen

Pfad: /Literatur

*.pdf Kopien der Online-Literatur

C.3 Abbildungen

Pfad: /Abbildungen

*.pdf Vektorgrafiken

*.jpg Diverse Rastergrafiken

*.png Diverse Rastergrafiken

C.4 Projektdateien

Pfad: /Projekt

WebModulEditor.zip . . . Praktische Arbeit Web-Modul-Editor

Quellenverzeichnis

Literatur

- [1] B. Chang und D. Ungar. „Animation: From Cartoons to the User Interface“. In: *Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology*. UIST '93. Atlanta, Georgia, USA: ACM, 1993, S. 45–55. URL: <http://doi.acm.org/10.1145/168642.168647> (siehe S. 7, 8).
- [2] M. Collomb u. a. „Improving Drag-and-drop on Wall-size Displays“. In: *Proceedings of Graphics Interface 2005*. GI '05. Victoria, British Columbia: Canadian Human-Computer Communications Society, 2005, S. 25–32. URL: <http://dl.acm.org/citation.cfm?id=1089508.1089514> (siehe S. 16–19).
- [3] *Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – DTeil 10: Grundsätze der Dialoggestaltung – (ISO 9241-10 : 1995)*. Standard. International Organization for Standardization, Feb. 1995. URL: http://www.interactive-quality.de/site/DE/int/pdf/ISO_9241-10.pdf (besucht am 21.04.2017) (siehe S. 8).
- [4] M. Herczeg. *Interaktionsdesign. Gestaltung interaktiver und multi-medialer Systeme*. München: Oldenbourg Wissenschaftsverlag GmbH, 2006 (siehe S. 9–12, 58).
- [5] J. Nielsen. *Designing Web Usability*. New York, USA: New Riders Publishing, 2000 (siehe S. 1).
- [6] J. Nielsen und H. Loranger. *Prioritizing Web Usability*. Berkeley, CA: New Riders Publishing, 2006 (siehe S. 4).
- [7] B. Preim und R. Dachzelt. *Interaktive Systeme. Band 1: Grundlagen, Graphical User Interfaces, Informationsvisualisierung*. 2. Aufl. Berlin: Springer-Verlag Berlin Heidelberg, 2010 (siehe S. 13).

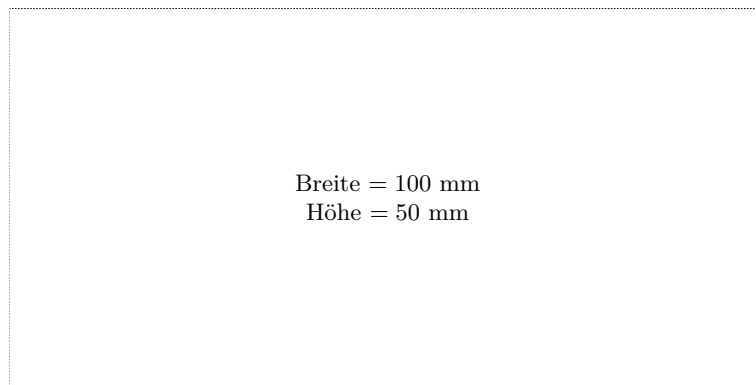
Online-Quellen

- [8] T. Adeyemi. *Documentation interact.js*. 2015. URL: <http://interactjs.io/docs/> (besucht am 01.04.2017) (siehe S. 31).
- [9] N. Babich. *XD Essentials: How Functional Animation Helps Improve UX*. 2016. URL: <https://blogs.adobe.com/creativecloud/xd-essentials-how-functional-animation-helps-improve-ux/> (besucht am 06.01.2017) (siehe S. 2, 6).
- [10] L. Bebbler. *Elastic Progress*. 2015. URL: <https://tympanus.net/codrops/2015/09/23/elastic-progress/> (besucht am 07.01.2017) (siehe S. 7).
- [11] A. Daliot. *Functional Animation In UX Design*. 2015. URL: <https://www.smashingmagazine.com/2015/05/functional-ux-design-animations/> (besucht am 07.01.2017) (siehe S. 7).
- [12] Google Developers. *Timelines*. 2017. URL: <https://developers.google.com/chart/interactive/docs/> (besucht am 02.04.2017) (siehe S. 32).
- [13] netwiss GesmbH. *Quantitative versus qualitative Erhebungen*. 2013. URL: <http://www.abschlussarbeit.at/index.php/sitemap.html?id=44> (besucht am 09.06.2017) (siehe S. 49).
- [14] J. Jacobsen. *Bessere Interfaces dank Animation*. 2014. URL: <http://www.usabilityblog.de/2014/07/bessere-interfaces-dank-animation/> (besucht am 07.04.2017) (siehe S. 2, 5, 58).
- [15] E. Loidl. *Eva Loidl | Design. Graphic, Photographie und Web...* 2017. URL: <http://evaloidl.bplaced.net> (besucht am 23.04.2017) (siehe S. 13).
- [16] Microsoft. *Windows 7: Papierkorb unter Computer (Arbeitsplatz) anzeigen*. 2010. URL: <https://support.microsoft.com/de-de/help/2394916> (besucht am 08.06.2017) (siehe S. 24).
- [17] J. Nielsen. *Zeitebenen des Nutzererlebnisses*. 2009. URL: <http://www.usability.ch/news/zeitebenen-des-nutzererlebnisses.html> (besucht am 18.05.2017) (siehe S. 21).
- [18] W. Schmidt-Sielex. *Word - Hilfreiche Programmeinstellungen*. URL: <http://www.schmidt-sielex.de/word/word02.htm> (besucht am 21.04.2017) (siehe S. 9).
- [19] International Organization for Standardization. *ISO/DIS 9241-11.2*. 2016. URL: http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=63500 (besucht am 02.01.2017) (siehe S. 1).
- [20] National Institute of Standards und Technology. *Timing Issues for Usability*. 2006. URL: <https://www.nist.gov/sites/default/files/documents/itl/vote/TimingIssuesforUsability.pdf> (besucht am 18.05.2017) (siehe S. 21).

- [21] Bootstrap Team. *Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web*. URL: <http://getbootstrap.com> (besucht am 09.05.2017) (siehe S. 36).
- [22] Mouseflow Team. *mouseflow*. URL: <https://mouseflow.com> (besucht am 15.05.2017) (siehe S. 49).
- [23] UXPin. *Guides, smart guides and snapping*. 2016. URL: <https://www.uxpin.com/community/tutorials/guides-smart-guides-and-snapping/> (besucht am 20.04.2017) (siehe S. 15, 16).
- [24] Ltd. Visual Hierarchy. *Improve your UX-Driven Designs with Functional Animations*. 2015. URL: <https://visualhierarchy.co/blog/improve-your-ux-functional-animations/> (besucht am 06.01.2017) (siehe S. 5).
- [25] S. Winter. *Quantitative vs. Qualitative Methoden*. 2000. URL: http://nosnos.synology.me/MethodenlisteUniKarlsruhe/imihome.imi.uni-karlsruhe.de/nquantitative_vs_qualitative_methoden_b.html (besucht am 09.06.2017) (siehe S. 49).
- [26] M. Younes. *Alert Dialog*. URL: <http://alertifyjs.com/alert.html> (besucht am 02.04.2017) (siehe S. 35).
- [27] M. Younes. *AlertifyJS*. URL: <http://alertifyjs.com> (besucht am 02.04.2017) (siehe S. 33).
- [28] M. Younes. *Confirm Dialog*. URL: <http://alertifyjs.com/confirm.html> (besucht am 02.04.2017) (siehe S. 35).
- [29] Limited ZeptoLab UK. *Pudding Monsters*. 2014. URL: <https://itunes.apple.com/us/app/pudding-monsters/id569185650?mt=8> (besucht am 06.01.2017) (siehe S. 6).

Messbox zur Druckkontrolle

— Druckgröße kontrollieren! —



— Diese Seite nach dem Druck entfernen! —