

Institut für Visualisierung und Interaktive Systeme
Universität Stuttgart
Universitätsstraße 38
70569 Stuttgart
Germany

Diplomarbeit Nr. 3672

**Visuelle Analyse
von Eyetracking-Experimenten
mit interaktiven Stimuli**

Leonard Maximilian Bruder

Studiengang:	Softwaretechnik
Prüfer:	Prof. Dr. Thomas Ertl
Betreuer:	Dipl.-Inf. Tanja Blascheck
begonnen am:	16.06.2014
beendet am:	16.12.2014
CR-Klassifikation:	H1.1, H.1.2, H.5.2, I.3.6

Kurzfassung

Mit Eye-Tracking-Experimenten können Fragestellungen auf dem Forschungsgebiet der Visualisierung beantwortet werden. Bisherige Experimente befassen sich vor allem mit statischen Stimuli oder dynamischen Inhalten wie Videos. Obwohl Interaktion ein wesentlicher Bestandteil vieler Visualisierungen ist, wurde sie bisher in Erfassung und Auswertung nicht ausreichend berücksichtigt. In dieser Arbeit stelle ich ein Konzept vor, wie Eye-Tracking-Experimente und deren Analyse erweitert werden können, um interaktive Stimuli wie Visualisierungen oder Webseiten zu untersuchen.

Zur Entwicklung eines solchen Konzepts wurden zuerst die besonderen Eigenschaften interaktiver Stimuli analysiert. Diese Eigenschaften haben maßgeblich Einfluss auf die Daten, die zusätzlich aufgezeichnet werden müssen. Es wird gezeigt, wie die in einer Studie verwendete Software zu diesem Zweck angepasst werden kann. Des Weiteren habe ich eine Visualisierung entwickelt, die für die Auswertung der in erweiterten Experimenten erfassten Daten, also der Eye-Tracking-Daten und der zusätzlich protokollierten Daten mehrerer Probanden, geeignet ist. Die prototypische Implementierung dieser Visualisierung ermöglicht den Vergleich dieser Daten. Neben dem Aufbau der GUI des Prototyps und dessen Datenstruktur habe ich außerdem die Demonstration der Funktionsweise in dieser Arbeit dokumentiert.

Abstract

Eye tracking can be used to answer different questions concerning computer-based visualization. So far, research focuses either on static stimuli or dynamic content, with a few exceptions. Even though interaction is an important part of many visualizations, it rarely has been taken into account appropriately. In this thesis I present a method to enrich eye-tracking experiments with the objective to analyze data of interactive stimuli like visualizations and user interfaces.

In order to develop an appropriate concept, I determined those characteristics of interactive stimuli which differ from static stimuli. These characteristics influence to a great extent the additional data which have to be logged. I further illustrated how to adapt the software used in a study to log this data. My solution is a newly developed visualization for the analysis of both, the eye-tracking data and the additionally logged data, for multiple participants of a study. A software tool using this visualization has been implemented and allows to compare the data. Finally, I documented the structure, the user interface and the functionality of this prototype in my thesis.

Geschlechtsneutrale Formulierungen in diesem Werk

Aus Gründen der einfacheren Lesbarkeit wird auf eine geschlechtsneutrale Differenzierung verzichtet. Entsprechende Begriffe gelten im Sinne der Gleichbehandlung grundsätzlich für alle Geschlechter.

Inhalt

1	Einleitung	1
1.1	Gliederung	1
2	Grundlagen	3
2.1	Eye-Tracking	3
2.1.1	Augenbewegung	3
2.1.2	Metriken	4
2.1.3	Technologie zur Blickerfassung	5
2.2	Visualisierung	7
2.2.1	Die Visualisierungspipeline	7
2.2.2	Elemente einer Visualisierung	8
2.2.3	Visualisierung von Eye-Tracking-Daten	9
2.2.4	Interaktive Stimuli	14
2.2.5	Visual Information Seeking Mantra	14
2.3	Grafische Benutzungsschnittstelle	15
2.3.1	Elemente einer GUI	16
2.3.2	Grundsätze der Dialoggestaltung	17
2.4	Usability	18
2.4.1	Interaktion	20
3	Aufgabenstellung und Lösungsansatz	23
3.1	Hintergrund	23
3.2	Aufgabenstellung	23
3.3	Lösungsansatz	24
4	Relevante Arbeiten	26
5	Lösungskonzept	31
5.1	Anforderungen	31
5.1.1	Daten und Datenherkunft	32
5.1.2	Anforderungen aus bisherigen Erkenntnissen	36
5.2	Erfasste Daten	37
5.3	Visuelle Analyse	38
5.3.1	Visual Information Seeking Mantra	44
5.4	Erfüllung der Anforderungen	46
6	Implementierung eines Prototyps	49
6.1	Die Benutzungsschnittstelle von Inseyet	49
6.1.1	Aufbau des Hauptfensters	49
6.1.2	Bereiche des Programmfensters	51

6.2	Architektur.....	61
6.3	Datenstruktur	63
6.4	Datenimport	67
6.5	Verwendung der Daten.....	70
6.6	Benutzungsschnittstelle und Interaktion.....	71
6.6.1	Overview First.....	71
6.6.2	Zoom & Filter.....	71
6.6.3	Details on Demand	72
6.7	Daten zur Entwicklung von Inseyet.....	74
7	Demonstration des Prototyps	76
7.1	Darstellung von Daten als Übersicht	76
7.2	Beispiel für eine Analyse.....	77
8	Zusammenfassung und Ausblick	85
8.1	Fazit	85
9	Quellenverzeichnis	87

Abbildungsverzeichnis

Abbildung 1: Zwei externe Eyetracker des Herstellers Tobii	5
Abbildung 2: Die zweite Version der Tobii Glasses	6
Abbildung 3: Ausschnitt eines Stimulus mit eingezeichneten Eye-Tracking-Daten	6
Abbildung 4: Die Visualisierungspipeline	8
Abbildung 5: Beispiel für eine Heatmap für einen Stimulus	10
Abbildung 6: Beispiel für die Darstellung mehrerer Scanpaths für einen Stimulus	11
Abbildung 7: Beispiel für die Parallel Scan Path Visualization.....	12
Abbildung 8: Beispiel für ein Circular Heat Map Transition Diagram.....	13
Abbildung 9: Beispiel für die Umsetzung eines Scarf Plots	14
Abbildung 10: Inhalte dieser Diplomarbeit.....	24
Abbildung 11: Snapshot des Programms KiEV	27
Abbildung 12: Der Interaktionsgraph nach Groth und Murphy.....	28
Abbildung 13: Ausschnitt der Logdatei eines Probanden	29
Abbildung 14: Die Funktionsweise von DynAOI.....	29
Abbildung 15: Die Programmoberfläche von VarifocalReader nach Programmstart	33
Abbildung 16: Programmoberfläche von VarifocalReader nach der Interaktion	34
Abbildung 17: Entwurf für die Darstellung der Daten.....	40
Abbildung 18: Radialer Entwurf für die Darstellung der Daten	42
Abbildung 19: Entwurf eines Programmfensters für die Präsentation der Daten	43
Abbildung 20: Hauptfenster unmittelbar nach dem Start des Programms.....	50
Abbildung 21: Hauptfenster von Inseyet mit Graph	51
Abbildung 22: Menü- und Reiterleiste von Inseyet	52
Abbildung 23: Bedienelemente zur Steuerung der Visualisierung in Inseyet.....	53
Abbildung 24: Zusammenhang zwischen Zoomfaktor und Größe des Versatzreglers.....	54
Abbildung 25: Buttons zum Anpassen der Visualisierung in Inseyet.....	54
Abbildung 26: Liste der Probanden in Inseyet	56
Abbildung 27: Dialog zum Einlesen der Daten eines Probanden	56
Abbildung 28: Darstellung der Daten dreier Probanden mit Inseyet	58
Abbildung 29: Interaktionsprotokoll aller dargestellten Probanden	60
Abbildung 30: Das Architekturprinzips Model-View-ViewModel	61
Abbildung 31: Überblick über die interne Datenstruktur.....	63
Abbildung 32: Aufbau des Eintrags für eine Fixation	64
Abbildung 33: Aufbau der Einträge für Mausbewegung und -fixierung sowie AOI.....	65
Abbildung 34: Aufbau der Einträge für Interaktion, Stimulus und AOI	66
Abbildung 35: Struktur eines Eintrag im Interaktionsprotokoll.....	66

Abbildung 36: Tooltip für einen Probanden im Graphen	72
Abbildung 37: Tooltips für die verschiedenen Elemente im Graphen.....	73
Abbildung 38: Ausschnitt aus einem Snapshot einer Videoaufzeichnung	74
Abbildung 39: Darstellung von 30 Probanden mit Inseyet	76
Abbildung 40: Visualisierung eines Datensatzes von sechs Probanden mit Inseyet	77
Abbildung 41: Visualisierung der Daten (sortiert) von fünf Probanden mit Inseyet	78
Abbildung 42: Visualisierung der Daten dreier Probanden mit Inseyet	79
Abbildung 43: Interaktionsprotokoll mit Einträgen zu drei Probanden	80
Abbildung 44: Vergleich der Scroll-Interaktionen mithilfe von Tooltips.....	81
Abbildung 45: Visualisierung ohne Interaktionsdaten.....	82
Abbildung 46: Visualisierung mit Zoom, Zeitversatz und ohne Interaktionsprotokoll	83
Abbildung 47: Der Stimulus, wie er dem Probanden angezeigt wurde	84

Tabellenverzeichnis

Tabelle 1: Gegenüberstellung ursprünglicher Anforderungen und deren Umsetzung.....	47
Tabelle 2: Beim Einlesen der Eye-Tracking-Daten verwendete Spalten der Quelldatei und deren Bedeutung.....	68
Tabelle 3: Beim Einlesen der Interaktionsdaten verwendete Spalten der Quelldatei und deren Bedeutung.....	69
Tabelle 4: Beim Einlesen der Daten zur Mausbewegung verwendete Spalten der Quelldatei und deren Bedeutung.....	69

1 Einleitung

Auf dem Forschungsgebiet der Visualisierung beschäftigt man sich in vielen Experimenten mit der Frage, wie geeignet eine bestimmte Visualisierung am Computer für die Bearbeitung einer konkreten Aufgabenstellung ist. In diesen Experimenten werden Techniken zur Blickerfassung eingesetzt, um festzustellen, wie Probanden eine Darstellung wahrnehmen und welche Informationen sie daraus gewinnen. Viele solcher Darstellungen können vom Probanden interaktiv verändert werden. Das bedeutet, dass sich diese Darstellungen für die einzelnen Probanden in einem Experiment erheblich unterscheiden können.

Die Menge der Daten, die bei solchen Experimenten erfasst wird, ist beträchtlich. Eine Auswertung solcher Datenmengen kann durch eine geeignete Darstellung der erfassten Daten deutlich vereinfacht werden. Derartige Visualisierungen existieren bereits und unterscheiden sich vor allem durch ihre Eignung für konkrete Aufgabenstellungen. Meiner Recherche zufolge beschränkt sich ein Großteil der Visualisierungen auf die Darstellung von Daten zur Blickerfassung. Bei diesen wird die Möglichkeit, diese mit den Daten zur Interaktion in Verbindung zu setzen, außer Acht gelassen. Das liegt unter anderem daran, dass viele der Visualisierungen auf statische Datenrepräsentationen in einem Experiment ausgelegt sind.

Vergleicht man bei einer Analyse die visualisierten Daten verschiedener Probanden, lassen sich eventuelle Zusammenhänge zwischen den Daten herausfinden. Durch die zusätzliche Anzeige von erfassten Daten zur Interaktion, neben denen zur Blickerfassung, können neue Muster entdeckt und interpretiert werden.

Das Ziel dieser Diplomarbeit ist es, eine Visualisierung zu entwickeln, die es erlaubt, die Daten zur Blickerfassung verschiedener Probanden – trotz der während des Experiments individuell angepassten Darstellung – miteinander vergleichen zu können. Bisherige Visualisierungen, die Daten zur Blickerfassung gemeinsam mit denen der Interaktion darstellen, eignen sich nur bedingt für diesen Zweck.

Das in meiner Arbeit vorgestellte Konzept ermöglicht es, die Daten mehrerer Probanden gleichzeitig so darzustellen, dass für jeden einzelnen zu erkennen ist, wie er die ihm angezeigte Darstellung wahrgenommen und durch Interaktion verändert hat. Somit können die angezeigten Daten der Blickerfassung in den richtigen Kontext eingeordnet werden, weil erkenntlich ist, was der Proband in einer interaktiven Datenrepräsentation zu sehen bekommen hat.

1.1 Gliederung

Diese Diplomarbeit ist in acht Kapitel unterteilt. Nach der Einleitung in das Thema *Visuelle Analyse von Eye-Tracking-Experimenten mit interaktiven Stimuli* sind in den folgenden Kapiteln die erforderlichen Grundlagen, die konkrete Aufgabenstellung, eine Auswahl von anderen Arbeiten mit Themenbezug, das Konzept der visuellen Analyse selbst, die Beschreibung der Implementierung eines Prototyps, die Demonstration des Prototyps und eine Zusammenfassung der Arbeit enthalten.

Kapitel 2 – Grundlagen definiert die für diese Arbeit wichtigen Begriffe. Es werden unterschiedliche Konzepte vorgestellt, die für meine Aufgabenstellung, den Lösungsansatz und das Verständnis der nachfolgenden Inhalte relevant sind. Hierbei geht es insbesondere um die

Themen Blickerfassung, Visualisierung von bei der Blickerfassung entstandenen Daten und den Aufbau von Benutzungsschnittstellen sowie Visualisierungen.

Kapitel 3 – Aufgabenstellung und Lösungsansatz befasst sich mit Hintergrundinformationen zum Thema der Diplomarbeit, der eigentlichen Aufgabenstellung und skizziert einen Lösungsansatz.

Kapitel 4 – Relevante Arbeiten gibt einen Überblick über relevante Arbeiten bezüglich der Erfassung, Verarbeitung und Darstellung von Daten aus Experimenten zu interaktiven Visualisierungen und geht auf deren Bedeutung für meine Arbeit ein.

Kapitel 5 – Lösungskonzept enthält das von mir entwickelte Konzept zur visuellen Analyse von Eye-Tracking-Experimenten mit interaktiven Stimuli. Es wird sowohl auf die Anforderungen an eine Visualisierung als auch auf die Entstehung und den Aufbau einer konkreten Visualisierung eingegangen.

Kapitel 6 – Implementierung eines Prototyps dokumentiert die Arbeit am Prototyp, der zum Zweck der Evaluation sowie Demonstration des Konzepts implementiert wurde. Zum einen wird die interne Programmstruktur aus technischer Sicht beschrieben und zum anderen wird die Aufteilung der Benutzungsschnittstelle in verschiedene Bereiche, auch aus Sicht eines Anwenders, formuliert. Darüber hinaus gibt das Kapitel Auskünfte zum Datenimport und zum entstandenen Programmcode.

Kapitel 7 – Demonstration des Prototyps führt den Leser durch eine Datenanalyse mit dem entwickelten Prototyp, um die Stärken und Schwächen des entstandenen Visualisierungskonzepts und dessen Umsetzung im Prototyp aufzuzeigen.

Kapitel 8 – Zusammenfassung und Ausblick fasst die Ergebnisse der Diplomarbeit zusammen. Ein Ausblick enthält Vorschläge, wie das entwickelte Konzept verbessert und erweitert werden kann.

2 Grundlagen

Dieses Kapitel soll dem fachfremden Leser helfen, die Inhalte dieser Diplomarbeit verstehen und beurteilen zu können. Insbesondere wird auf die Themen Blickerfassung (Eye-Tracking), Visualisierung, Benutzerfreundlichkeit (Usability), grafische Benutzungsschnittstelle (GUI) und Interaktion eingegangen. Das Ziel ist es, gängige Konzepte zu vermitteln und wichtige Fachbegriffe zu definieren.

2.1 Eye-Tracking

Eye-Tracking ist eine wichtige Methode zur Analyse der Interaktion eines Menschen mit seiner Umwelt und insbesondere der Interaktion mit Computersystemen. Eye-Tracking wird zum Beispiel bei der Evaluierung von Benutzungsschnittstellen eingesetzt, um zu erkennen, wie der Mensch diese wahrnimmt. Ein Eyetracker ist das Gerät, mit dem die Aufzeichnung der Augenbewegung eines Menschen vorgenommen wird.

Ein gesunder Mensch nimmt einen Großteil der Informationen aus seiner Umgebung mit den Augen wahr (Zühlke 2012). Aus diesem Grund liegt es nahe, dass Computerprogramme die meisten Informationen in visualisierter Form bereitstellen – im Unterschied zu auditivem oder taktilem Feedback. Die Analyse der Augenbewegung eines Menschen lässt folglich Rückschlüsse auf dessen Art der Informationsaufnahme und -verarbeitung zu (Just und Carpenter 1980). Sie ist gleichzeitig die Analyse der visuellen Aufmerksamkeit einer Person, weil Menschen natürlich auf interessante, reizvolle oder wichtige Objekte blicken, die auf Grund präattentiver Wahrnehmung direkt fokussiert werden. Präattentive Wahrnehmung bedeutet in diesem Kontext, dass ein Sinnesreiz vom Nervensystem einer Person zwar bereits wahrgenommen wurde, jedoch nicht bis ins Bewusstsein der Person gelangt ist.

Bei der Blickerfassung werden die Koordinaten der Punkte eines Stimulus registriert, die sich in Blickrichtung eines Probanden befinden. Diese Punkte werden zusammen mit dem zugehörigen Zeitstempel und gegebenenfalls weiteren Informationen gespeichert. Daraus lassen sich die Augenbewegungen rekonstruieren. Ein *Stimulus* ist dabei das betrachtete Objekt, also zum Beispiel ein Gegenstand, ein Bild, ein Video, eine Website oder ein Programmfenster.

Zur Analyse der Eye-Tracking Daten werden ausgewählte Bereiche des Stimulus als *Area(s) of Interest (AOI)* definiert. Das geschieht entweder im Voraus, wenn diese Bereiche beispielsweise besonders relevant für eine Benutzerstudie sind, oder im Nachhinein, um die Bereiche zu kennzeichnen, die häufig oder für einen längeren Zeitraum von vielen Probanden betrachtet wurden. Sind die AOI bekannt, kann zum Beispiel berechnet werden, wie lange eine AOI betrachtet wurde oder wie oft zwischen verschiedenen AOI hin- und hergeschaut wurde. Einen solchen Übergang bezeichnet man auch als *Transition*.

2.1.1 Augenbewegung

Es gibt verschiedene Bewegungen der Augen (Joos, Röttin und Velichkovsky 2003), die für die Auswertung von Daten zur Augenbewegung des Menschen von Bedeutung sind. Diese lassen sich wie folgt unterscheiden:

Blickpunkt: Als Blickpunkt wird üblicher Weise die zu einem beliebigen Zeitpunkt ermittelte Position der Augen auf einem Stimulus bezeichnet.

Fixation: Bei einer Fixation wird das Auge auf ein ruhendes Objekt ausgerichtet. Aus technischer Sicht setzt sich eine Fixation aus aufeinander folgenden Blickpunkten zusammen. Diese können bezüglich ihrer Position innerhalb eines geringen, definierten Rahmens voneinander abweichen. Es kann außerdem eine zeitliche Begrenzung für die Dauer einer Fixation verwendet werden. Weil die Nervenzellen des Auges auf Veränderung reagieren, unterliegt das Auge während einer Fixation sogenannten *Mikrobewegungen*, einer Art Zittern des Auges, um die Aufnahme neuer Reize zu gewährleisten.

Sakkade: Sakkaden sind sehr schnelle, zielsuchende Bewegungen des Auges zwischen zwei Fixationen. Die Dauer einer Sakkade beträgt zwischen 10 und 100 Millisekunden (Duchowski 2007). Sie sind meistens ein Hinweis auf die Verschiebung der visuellen Aufmerksamkeit. Während einer Sakkade – also einer raschen Augenbewegung – ist die visuelle Wahrnehmung eingeschränkt.

Verfolgung: Mit Verfolgung oder *Smooth Pursuit* bezeichnet man eine gleichmäßige Augenbewegung, bei der die Augen ein sich langsam bewegendes Objekt verfolgen. Bewegt sich das Objekt sehr schnell, sind Sakkaden erforderlich, um es mit dem Auge verfolgen zu können.

Scanpath: Der aus Fixationen (Verharren des Blickpunkts) und Sakkaden (Bewegung des Blickpunkts) resultierende Pfad wird als Scanpath bezeichnet.

2.1.2 Metriken

Für die Interpretation der in Kapitel 2.1.1 *Augenbewegung* genannten Daten sind verschiedene Metriken bekannt (Jacob und Karn 2003). Die gängigsten darunter werden nachfolgend beschrieben.

Die Anzahl der Fixationen insgesamt: Die Gesamtzahl der Fixationen ist immer in Abhängigkeit von der Interaktionsdauer zu betrachten. Dennoch deutet eine hohe Anzahl Fixationen darauf hin, dass sich der Proband nicht zurechtgefunden hat. Ursache hierfür kann unter anderem eine schlechte Anordnung grafischer Elemente in einem Programm oder einer Visualisierung sein.

Die Dauer, die eine einzelne AOI betrachtet wurde: Der prozentuale Anteil an der gesamten Interaktionsdauer, in der eine AOI betrachtet wurde, könnte deren Wichtigkeit widerspiegeln. Allerdings setzt sich die Dauer aus einzelnen Fixationen zusammen. Das bedeutet, diese Dauer lässt sich in *Häufigkeit der Fixationen* und *durchschnittliche Fixationsdauer* zerlegen. Während die Häufigkeit ein Indikator für die Wichtigkeit einer AOI ist, ist eine hohe durchschnittliche Fixationsdauer ein Hinweis auf erschwerte Informationsgewinnung. Es ist außerdem denkbar, dass sehr kurze Fixationen einer AOI darauf hinweisen, dass die gesuchten Informationen dort nicht wie erwartet aufzufinden waren.

Scanpath: Ein Scanpath kann ebenfalls als Metrik herangezogen werden. Er gibt Hinweise darauf, wie sinnvoll die einzelnen Elemente respektive AOI auf dem Stimulus angeordnet wurden.

2.1.3 Technologie zur Blickerfassung

Die zeitgemäßen Verfahren zum Erfassen der Augenbewegung sind kamerabasierte Lösungen. Bei diesen Verfahren werden üblicherweise Kameras mit Infrarotlichtquellen kombiniert, um anhand der Reflexion des Lichts auf der Hornhaut die Position der Pupillen festzustellen. Damit lässt sich auch das Zentrum der Pupillen ermitteln, wodurch wiederum die vom Menschen betrachteten Punkte bestimmt werden können.

Zwei verbreitete Methoden hierfür sind kopfgetragene, also brillenähnliche Systeme und stationäre beziehungsweise externe Systeme. Letztere werden zum Beispiel an einem Monitor platziert, der den Stimulus anzeigt.



Abbildung 1: Zwei externe Eyetracker des Herstellers Tobii

Die Modelle Tobii T60XL (links) und Tobii X2-60 (rechts) mit einer Abtastrate von jeweils 60 Hz. Während der Eyetracker beim Tobii T60XL fest im Bildschirm verbaut ist, kann der Tobii X2-60 frei positioniert werden. Bildquelle: (www.tobii.com)

Der Vorteil von stationären Systemen ist, dass sie die Augenbewegung eines Probanden ermitteln können, ohne am Probanden befestigt werden zu müssen. Solche Systeme sind entweder wie in *Abbildung 1: Zwei externe Eyetracker des Herstellers Tobii* zu sehen fest am Monitor verbaut oder können frei zwischen Proband und Stimulus platziert werden. Weil stationäre Systeme in Kombination mit Computern eingesetzt werden, muss der Stimulus in der Regel nicht durch eine weitere Kamera aufgezeichnet werden. Es wird stattdessen direkt der angezeigte Stimulus gespeichert, der in vielen Fällen bereits im Voraus bekannt ist.

Der Nachteil solcher Systeme ist, dass der Proband durch starke Kopfbewegungen die Resultate der Blickerfassung beeinflussen kann. Eine Hilfe für den Probanden, um solche Bewegungen zu vermeiden, sind Stützen, auf die das Kinn abgelegt werden kann.

Ein kopfgetragenes System, wie in *Abbildung 2: Die zweite Version der Tobii Glasses* dargestellt, wird dem Probanden aufgesetzt und hat den Vorteil, dass zusätzlich zur Augenbewegung auch das Sichtfeld des Probanden durch eine Kamera erfasst werden kann. Damit wird die Blickrichtung relativ zur Ausrichtung des Kopfes erfasst. Diese Art von Eyetracker kann in einer beliebigen Umgebung eingesetzt werden, weil sowohl der Stimulus beziehungsweise das Sichtfeld als auch die Augenbewegung erfasst werden. Folglich kann mit einem solchen

System die Interaktion mit einem im Vorfeld unbekanntem Stimulus aufgezeichnet und im Anschluss analysiert werden.



Abbildung 2: Die zweite Version der Tobii Glasses

Der brillenähnliche Eyetracker des Herstellers Tobii verfügt zusätzlich zu den Aufzeichnungsmöglichkeiten für die Augenbewegung über eine Kamera zur Aufnahme des Sichtfelds des Trägers. Es ist außerdem ein Mikrophon zur Aufnahme akustischer Signale eingebaut. Bildquelle: (www.tobii.com)

Eine wichtige Rolle bei der Erfassung und Auswertung von Eye-Tracking-Daten spielt die Qualität der Werte. Diese hängt vor allem von der Genauigkeit der Messungen, der verwendeten Abtastrate und dem Blickwinkel ab.

Die Abtastrate gibt das Intervall an, in dem ein Eyetracker die einzelnen Blickpunkte aufzeichnet. Eine Abtastrate von 30 Hz entspricht einem Intervall von ca. 33 Millisekunden und ist mit nahezu allen herkömmlichen Geräten möglich. Aufgrund der gegenwärtig sehr viel höheren Rechenleistung von externen Systemen gegenüber der mobiler Eyetracker, ermöglichen externe Systeme eine deutlich höhere Abtastrate.

Die Genauigkeit variiert von Eyetracker zu Eyetracker und gibt an, welche maximale Abweichung bei der Berechnung eines Blickpunkts auftreten kann. Weil diese Angabe in Grad erfolgt, spielt die Distanz zwischen Proband und Stimulus ebenfalls eine erhebliche Rolle.



Abbildung 3: Ausschnitt eines Stimulus mit eingezeichneten Eye-Tracking-Daten

Bildausschnitt mit erkennbarer Ungenauigkeit der aufgezeichneten Fixationen. Der Proband, in diesem Fall der Autor selbst, hat den Inhalt des Textfelds gelesen. Die vom Eyetracker eingezeichneten Fixationen weisen einen leichten Versatz auf.

Für den Eyetracker Tobii T60XL wird zum Beispiel eine Genauigkeit der Messung von $0,5^\circ$ bei entsprechender Kalibrierung gewährleistet (T60XL TechnicalSpecification). Bei einem Abstand von bis zu 70 cm zum Monitor entspricht das einer maximalen Abweichung von $70 \text{ cm} * \tan(0,5^\circ) = 0,61 \text{ cm}$. Diese Länge überschreitet die Größe vieler grafischer Elemente

auf Webseiten oder in Programmen. Es ist also denkbar, dass ein Eyetracker nicht korrekt aufzeichnet, dass ein Proband eine bestimmte Textzeile liest (wie in *Abbildung 3: Ausschnitt eines Stimulus mit eingezeichneten Eye-Tracking-Daten* zu sehen) oder ein kleines Bedienelement, wie beispielsweise einen Button, für kurze Zeit betrachtet.

2.2 Visualisierung

Der Begriff *Visualisierung* bezeichnet das Veranschaulichen von Daten. Das Ziel dabei ist, eine geeignete Darstellung der zugrundeliegenden Daten zu erzeugen, die es dem Menschen ermöglicht, diese oft abstrakten Daten mental zu erfassen. Zwischen den dargestellten Daten bestehen in der Regel viele verschiedene Zusammenhänge, die erst durch die Visualisierung sichtbar oder verständlich werden.

Die Anwendung dient häufig dem Auffinden von Mustern und Regelmäßigkeiten, der Entwicklung von Hypothesen aus diesen oder dem Bestätigen bereits bestehender Hypothesen anhand von Visualisierungen.

Bei der Analyse von Daten sind vor allem komplexe Zusammenhänge interessant. Um das Verständnis dieser Zusammenhänge zu vereinfachen, muss die kognitive Last, die durch die Visualisierung entsteht, auf ein Minimum reduziert werden. Das wird unter anderem dadurch erreicht, dass irrelevante Informationen ausgeblendet werden und bei der Visualisierung auf rein optische Aspekte verzichtet wird. Optimal ist ein freier und ungetrübter Blick auf die wesentlichen Informationen. Dazu gehört auch, dass die Überschneidung von dargestellten Elementen und ihre zu enge Anordnung vermieden werden. Gelingt das nicht und lassen sich die Elemente einer Visualisierung nicht mehr voneinander unterscheiden oder sind nicht mehr eindeutig identifizierbar, spricht man von *Visual Clutter*.

2.2.1 Die Visualisierungspipeline

Die Visualisierungspipeline in *Abbildung 4: Die Visualisierungspipeline* fasst die einzelnen Schritte sowie die daraus resultierenden Zwischenstände im Visualisierungsprozess zusammen und lässt sich gut am Beispiel einer Eye-Tracking-Studie beschreiben.

Nachdem die Daten beim Eye-Tracking erfasst wurden (*Data Aquisition*) müssen die entstandenen Rohdaten (*Raw Data*) gefiltert werden (*Filtering*), um nicht relevante Daten zu entfernen. Diese können zum Beispiel durch eine Unterbrechung und dem damit verbundenen Verlust des Blickkontakts entstanden sein.

Aus dem Filtern der Rohdaten entstehen die zu visualisierenden Daten (*Visualization Data*). Diese müssen wiederum auf – wenn auch zunächst abstrakte – geometrische Objekte und deren Attribute abgebildet werden (*Mapping*). Konkret könnte das zum Beispiel die Zuordnung von Fixationsdaten auf Kreise sein, deren Radius proportional zur Fixationsdauer ist. Die bei diesem Schritt erzeugten Geometriedaten sind bereits eine sinnvolle Repräsentation (*Renderable Representation*) der gefilterten Daten. Aus ihnen werden im letzten Schritt, dem Rendering, die eigentlichen Bilddaten generiert, die zum visuellen Ergebnis (*Visualization*) führen.

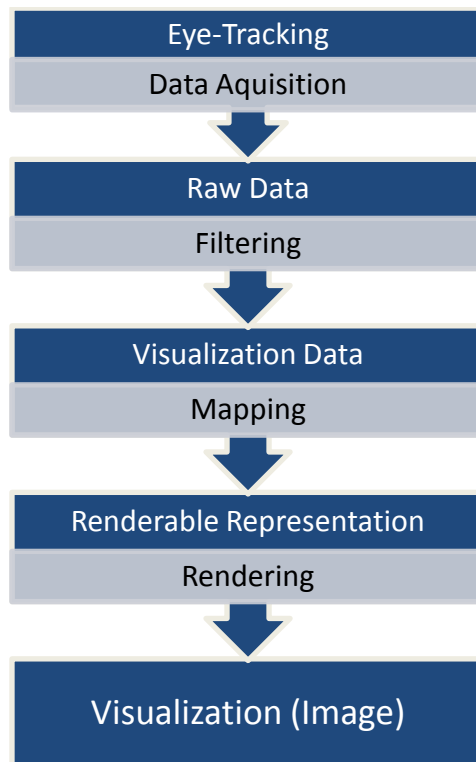


Abbildung 4: Die Visualisierungspipeline

Die einzelnen Schritte der Visualisierungspipeline, die von der Datengenerierung bis zu einer sichtbaren Visualisierung führen.

Nachfolgend werden verschiedene Elemente von visuellen Darstellungen auf abstrakter Ebene definiert.

2.2.2 Elemente einer Visualisierung

Für die Beurteilung von Eye-Tracking-Daten zu verschiedenen Visualisierungen ist es nützlich, enthaltene Elemente zu kategorisieren. Um durch die Analyse von Eye-Tracking-Daten verstehen zu können, wie ein Proband mit einem interaktiven Stimulus gearbeitet hat, müssen die von ihm betrachteten Elemente unterschieden und eingeordnet werden können.

Hintergrund: Der Hintergrund einer Visualisierung von Eye-Tracking-Daten kann der zugrundeliegende Stimulus sein. Das ist zum Beispiel bei Heatmaps der Fall (siehe Kapitel 2.2.2 *Visualisierung von Eye-Tracking-Daten*). Allerdings muss die Sinnhaftigkeit eines Hintergrunds für jede Visualisierung individuell geprüft werden. Beispielsweise ist die Verwendung eines Stimulus als Hintergrund nur dann möglich, wenn er für verschiedene Probanden gleich ist.

Beschriftungen: Die Beschriftung von Elementen ist eine hilfreiche Stütze für das schnelle Verständnis der angezeigten Visualisierung. Weil die Werte einzelner Elemente meistens durch deren Eigenschaften zum Ausdruck gebracht werden, ist eine explizite Angabe dieser Werte durch eine Beschriftung überflüssig. Es sollte außerdem darauf geachtet werden, nur die Beschriftungen automatisch anzuzeigen, die für ein generelles Verständnis oder den Überblick über die Daten erforderlich sind.

Achsen: Bedingt durch die zweidimensionale Darstellung von Informationen auf Monitoren und Leinwänden ist die Verwendung zwei- beziehungsweise dreidimensionaler Koordinatensysteme naheliegend. In diesem Fall existieren explizit oder implizit mindestens eine vertikale und eine horizontale Achse. Oft wird auch die zugehörige Hierarchie visualisiert und als Achse verwendet. Dabei ist es wichtig, zu verstehen, dass die Anordnung der Elemente in einer Hierarchie (zum Beispiel: alphabetisch) maßgeblichen Einfluss auf die damit verbundene Visualisierung und somit auf das Verständnis dieser haben kann.

Legende: Eine Legende gibt Aufschluss über das Mapping der Visualisierung und ist damit entscheidend für das Verständnis einer grafischen Repräsentation der Daten. Erfahrene Anwender sind oft in der Lage, eine Visualisierung auch ohne Legende korrekt entschlüsseln zu können. Deshalb ist es sinnvoll, eine Möglichkeit zum Ausblenden der Legende bereitzustellen.

Datenelemente: Datenelemente stellen die zugehörigen Werte durch ihre Eigenschaften dar. Zum Beispiel kann die Größe eines Kreises in einem Scanpath die Dauer der zugehörigen Fixation repräsentieren. Gleichzeitig codiert die Farbe des Kreises den Probanden, dem die Fixation zuzuordnen ist. Die Verbindung zwischen Werten und Ausprägung eines Elements ist durch die jeweilige Visualisierung gegeben, kann aber unter Umständen durch den Anwender angepasst werden.

Interaktionselemente: Interaktionselemente sind Bedienelemente, die die in Abschnitt *Interaktionskategorien* in Kapitel 2.4.1 *Interaktion* beschriebenen Interaktionen ermöglichen. Ein Datenelement kann gleichzeitig ein Interaktionselement sein.

Tooltip: Ein Tooltip ist ein kleines Fenster mit der Beschreibung eines Elements, das eingeblendet wird, sobald der Mauszeiger kurzzeitig über diesem Element verweilt. Ein Tooltip kann neben der Beschreibung eines Elements durchaus weitere Informationen enthalten.

2.2.3 Visualisierung von Eye-Tracking-Daten

Nachfolgend werden verschiedene Techniken vorgestellt, wie sich Daten aus Eye-Tracking-Experimenten visualisieren lassen. Zuerst werden zwei klassische Darstellungsmöglichkeiten beschrieben, die vor allem auf Grund ihrer Anschaulichkeit hier aufgeführt sind. Danach werden neuere Visualisierungen beschrieben, die für bestimmte Zwecke und unter Berücksichtigung bekannter Vor- und Nachteile entwickelt wurden. Darüber hinaus existiert natürlich eine Vielzahl anderer Visualisierungen, die in dieser Arbeit nicht berücksichtigt wurden.

Klassische Visualisierungen

Daten, die beim Erfassen von Augenbewegungen gesammelt wurden, werden klassischer Weise als sogenannte *Heatmaps* oder in Form von *Scanpath-Visualisierungen* dargestellt.

Eine Heatmap zeigt die Verteilung der visuellen Aufmerksamkeit einer beliebigen Anzahl von Probanden auf einem Stimulus über die Zeit aggregiert (siehe *Abbildung 5: Beispiel für eine Heatmap für einen Stimulus*). Dadurch ist die Heatmap eine rein räumliche Darstellung von Daten. Es wird die Intensität, mit der verschiedene Bereiche des Stimulus von den Probanden fixiert wurden, dargestellt. Die Intensität wird anhand der Dauer der Fixationen und der Häufigkeit von Fixation desselben Bereichs ermittelt.

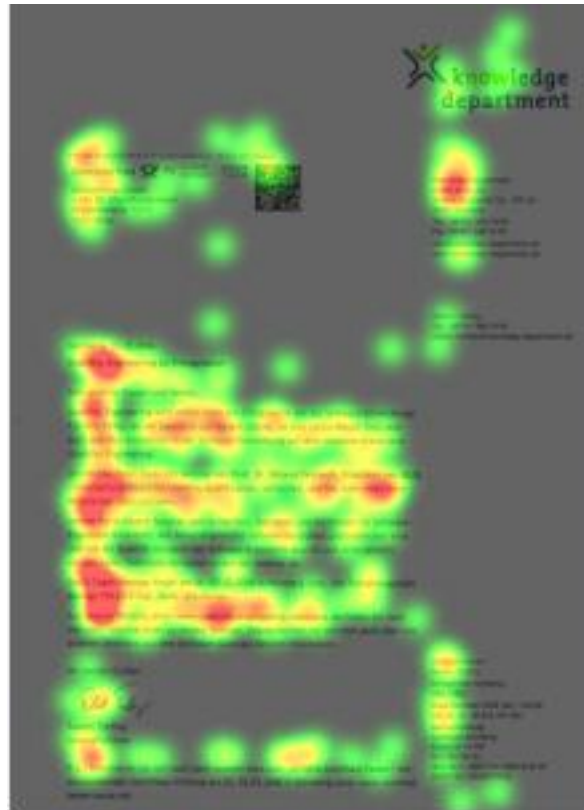


Abbildung 5: Beispiel für eine Heatmap für einen Stimulus

Die Heatmap verwendet den Stimulus selbst als Hintergrund. Bereiche, die von den Probanden überdurchschnittlich oft beziehungsweise lange betrachtet wurden, wurden rot eingefärbt. Bereiche, denen durchschnittlich viel Aufmerksamkeit geschenkt wurde sind gelb markiert. Die Farbe Grün kennzeichnet die Bereiche, die kaum visuelle Aufmerksamkeit erhalten haben. Bereiche ohne Einfärbung wurden von den Probanden nicht beachtet. Bildquelle: (www.usability.at)

Die Farben zur Darstellung der Intensität richten sich dabei in der Regel nach der Anzahl der Fixationen eines bestimmten Bereichs oder der Dauer, die ein solcher Bereich insgesamt fixiert wurde. Häufig werden die Farben Grün oder Blau für niedrige, Gelb für durchschnittliche und Rot für hohe Werte eingesetzt. Dadurch erweckt eine Heatmap den Eindruck, als würde die Wärmeverteilung auf einer Landkarte dargestellt.

Eine solche Darstellung verzichtet auf die Visualisierung von Sakkaden und der Reihenfolge, in der die einzelnen Fixationen aufgezeichnet wurden. Entsprechend können diese Eigenschaften mithilfe einer Heatmap nicht analysiert werden. Außerdem ist eine Heatmap nicht für interaktive und nur bedingt für dynamische Stimuli geeignet, weil sie die Daten in Abhängigkeit vom Stimulus präsentiert. Allerdings lassen sich mithilfe einer Heatmap AOI für statische Stimuli bestimmen. Je nachdem, welche Bereiche eines Stimulus von den Probanden überdurchschnittlich oft fixiert wurden, sollten diese Bereiche als mögliche AOI in Betracht gezogen werden.

Eine Scanpath-Visualisierung veranschaulicht die Bewegung der Augen in einer zeitlichen Abfolge (siehe *Abbildung 6: Beispiel für die Darstellung mehrerer Scanpaths für einen Stimulus*). Das gelingt, indem solche Visualisierungen sowohl die Fixationen als auch die Bewe-

gung zwischen diesen als Knoten (Fixationen) und Kanten (Sakkaden) in einem Diagramm verknüpfen. Es ist üblich, die Größe der Knoten an die Dauer der zugehörigen Fixation anzupassen. Aus dieser Visualisierung sind dann die Sequenz aller Augenbewegungen und die daraus resultierende Strategie eines Probanden ersichtlich. Die Strategie spiegelt die Reihenfolge der Betrachtung einzelner Bereiche eines Stimulus und den Wechsel zwischen diesen wider.



Abbildung 6: Beispiel für die Darstellung mehrerer Scanpaths für einen Stimulus

Die farblich (mit grün beziehungsweise rot) codierten Scanpaths zweier Probanden mit dem zugehörigen Stimulus als Hintergrund. Der Radius einer Kreisfläche gibt die Dauer der Fixation an. Mit Zahlen in den Kreisen könnte man außerdem die Reihenfolge der einzelnen Fixationen eines Probanden repräsentieren. Die Verbindung zwischen zwei Kreisen stellt die Sakkade zwischen den zugehörigen Fixationen dar. Bildquelle: (www.usability.at)

Bei einer solchen Scanpath-Visualisierung bleibt der Bezug zum Stimulus, ähnlich wie bei einer Heatmap, erhalten. Dadurch ist sie ebenfalls vor allem für statische Stimuli geeignet. Trotzdem kann die zeitliche Dimension durch die Nummerierung der Fixationen eines Probanden zusätzlich visualisiert werden. Hilfreich für die Darstellung der zeitlichen Abfolge sind außerdem die Verknüpfungen der Fixationen durch die eingezeichneten Sakkaden.

Solche Scanpath-Visualisierungen dienen nur bedingt als Vergleichsbasis verschiedener Probanden. Eine Schwierigkeit ist, einschätzen zu können, wie zwei Scanpaths zeitlich zueinander in Verbindung stehen. Werden die Daten zu vieler Probanden zeitgleich visualisiert, ist *Visual Clutter* außerdem unvermeidlich. Auch die Dauer der gezeigten Interaktion ist ein wesentlicher Faktor zur Vermeidung von *Visual Clutter*: Je größer der gezeigte Zeitabschnitt ist, desto mehr Elemente werden auf dem Stimulus eingezeichnet. Die Überschneidungen von Elementen für lange Zeiträume sind unvermeidlich.

Weitere Visualisierungen

Mittlerweile gibt es eine Vielzahl verschiedener und interessanter Visualisierungen von Eye-Tracking-Daten. Diese versuchen meistens, bekannte Probleme vorhandener Visualisierungstechniken zu lösen.

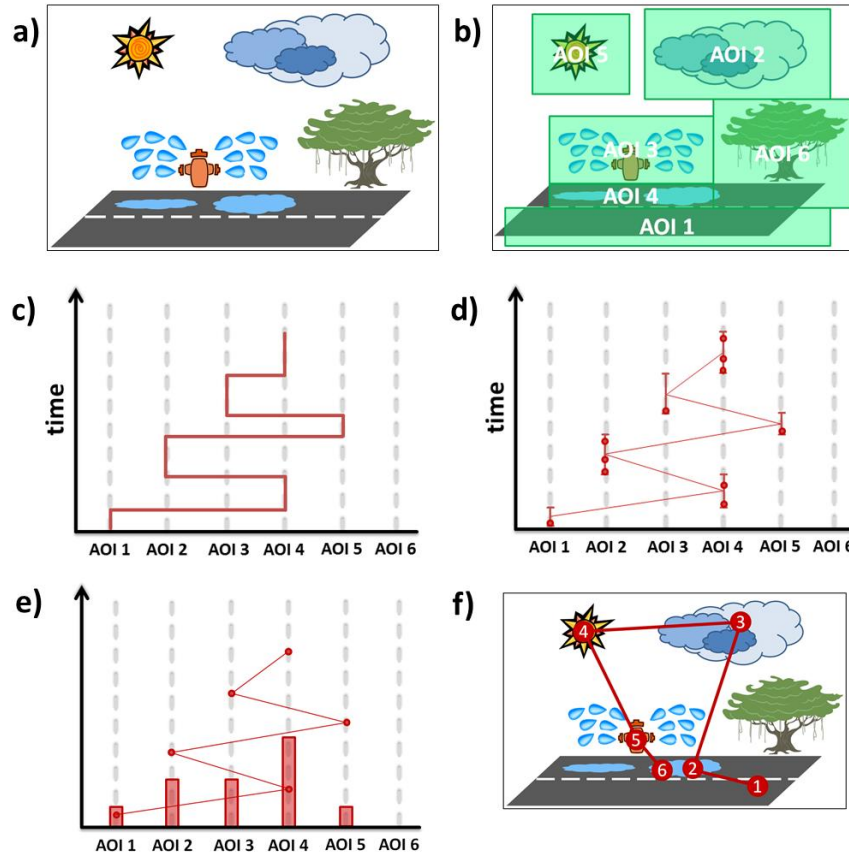


Abbildung 7: Beispiel für die Parallel Scan Path Visualization

Die Abbildung zeigt in a) den Stimulus und in b) dessen AOI. Darunter befindet sich in c) die Darstellung des Scanpaths, aus dem ersichtlich ist, wie lange und in welcher Reihenfolge ein Proband verschiedene AOI betrachtet hat. In d) werden einzelne Fixationen visualisiert. Es ist zudem dargestellt, welche Fixationen zusammengefasst wurden. In e) wird der gleiche Scanpath erzeugt, wie er in d) vorhanden ist. Allerdings ist in e) zusätzlich der prozentuale Anteil der Betrachtungsdauer einer AOI zu sehen. Die vertikalen Achsen in c), d) und e) dienen als Zeitachsen. Dadurch werden die Augenbewegungen zwischen den AOI über die Zeit erkenntlich. In f) ist der Stimulus mit eingezeichnetem Scanpath zu sehen. (Raschke, Chen und Ertl 2012)

Eine weitere Möglichkeit ist beispielsweise die parallele Darstellung der Augenbewegung mehrerer Probanden in einer einzigen Grafik. Mit der sogenannten *Parallel Scan Paths Visualization*, die in *Abbildung 7: Beispiel für die Parallel Scan Path Visualization* gezeigt ist, lassen sich verschiedene Informationen wie Fixationen, Scanpaths oder die Verteilung der Blickdauer auf die einzelnen AOI geeignet veranschaulichen (Raschke, Chen und Ertl 2012).

Hierbei wird der Stimulus in AOI aufgeteilt, um die Abhängigkeit der angezeigten Informationen vom Stimulus zu reduzieren. Diese AOI werden auf einer horizontalen Achse angeordnet und zu der Zeit auf der vertikalen Achse in Relation gesetzt. Ein großer Vorteil gegenüber

herkömmlichen Scanpath-Visualisierungen ist, dass hiermit auch große Zeitabschnitte ohne die Entstehung von *Visual Clutter* dargestellt werden können.

Visualisierungen, die Daten mithilfe von AOI unabhängig vom Stimulus darstellen, verwenden häufig ein zweidimensionales Koordinatensystem bestehend aus einer Achse mit den AOI und einer Zeitachse. Allerdings besteht auch die Möglichkeit, die Zeit zu aggregieren und die AOI wie die Bereiche eines Stimulus in einer Heatmap entsprechend der gesamten Fixationsdauer farblich zu codieren. Ein Beispiel für eine solche Darstellung ist ein *Circular Heat Map Transition Diagram* (Blascheck, Raschke und Ertl 2013).

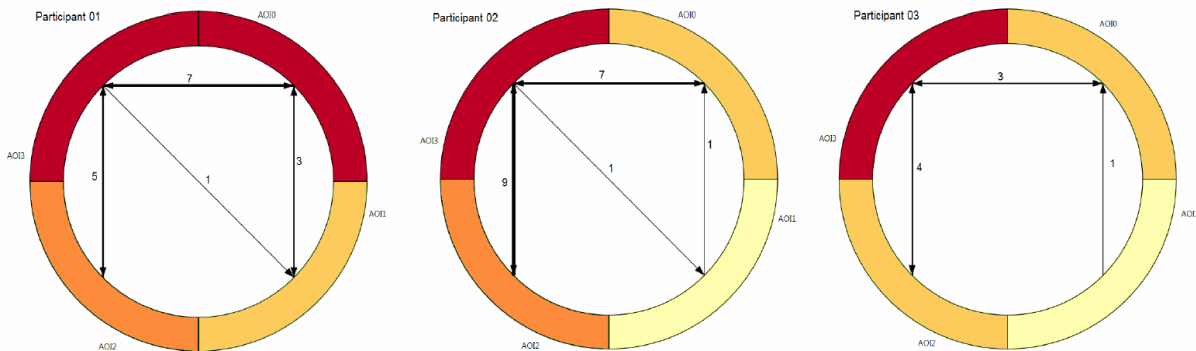


Abbildung 8: Beispiel für ein Circular Heat Map Transition Diagram

Verschiedene AOI eines Stimulus für drei Probanden. Die Farbgebung richtet sich jeweils nach der Anzahl der Fixationen. Zusätzlich sind die Transitionen zwischen den AOI durch Pfeile gekennzeichnet. Die Dicke des Pfeils ist dabei abhängig von der Anzahl der Transitionen. (Blascheck, Raschke und Ertl 2013)

Aus einem *Circular Heat Map Transition Diagram* lässt sich schnell erschließen, welche AOI am meisten Aufmerksamkeit erhalten haben (siehe *Abbildung 8: Beispiel für ein Circular Heat Map Transition Diagram*). Darüber hinaus kann anhand der Transitionen überprüft werden, wie gut die AOI angeordnet sind, beziehungsweise welche Anordnung der AOI sinnvoll ist. Allerdings führt eine Hohe Anzahl AOI in Verbindung mit einer großen Anzahl Transitionen auch bei dieser Visualisierung zu *Visual Clutter*.

Für eine gemeinsame Darstellung von AOI und Probanden bieten sich neben einem einfachen Koordinatensystem so genannte *Scarf Plots* an. Ein *Scarf Plot*, wie er in *Abbildung 9: Beispiel für die Umsetzung eines Scarf Plots* dargestellt ist, enthält für jeden Probanden eine Zeile, die dem zeitlichen Verlauf entspricht. In dieser Zeile werden die betrachteten AOI als Balken eingezeichnet. Die Dicke der Balken richtet sich nach der Dauer, für die die jeweilige AOI betrachtet wurde. Um welche AOI es sich handelt, ist farblich codiert.

Diese Art von Visualisierung eignet sich gut für eine geringe Anzahl von AOI, weil die einzelnen Farben dann einfach unterschieden werden können. Für mehr als ein Dutzend AOI fällt es dem Betrachter vermutlich schwer, die einzelnen Farben voneinander zu trennen.

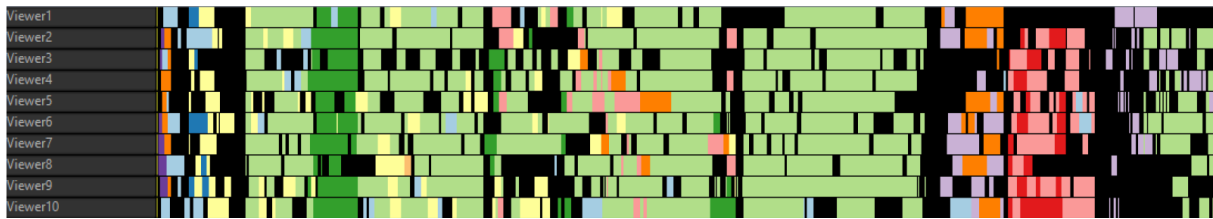


Abbildung 9: Beispiel für die Umsetzung eines Scarf Plots

Eine Liste von zehn Probanden. Für jeden dieser Probanden zeigt die zugehörige Zeitleiste die Zeitspannen der betrachteten AOI. Die Farbe eines Zeitblocks gibt dabei Aufschluss darüber, um welche AOI es sich handelt. Wurde für einen beliebigen Zeitraum keine AOI betrachtet, wird der entsprechende Bereich schwarz dargestellt. (Kurzhaus, Heimerl und Weiskopf 2014)

Außerdem kann mit einer solchen Visualisierung vergleichsweise einfach festgestellt werden, inwieweit sich das Blickverhalten verschiedener Probanden ähnelt.

2.2.4 Interaktive Stimuli

Die im vorigen Abschnitt aufgeführten Beispiele zeigen jeweils einen statischen Stimulus. Das bedeutet, dass es sich dabei um während der Betrachtung durch einen Probanden unveränderte Bilder handelt. Ein interaktiver Stimulus kann dagegen durch verschiedene Interaktionen verändert werden. Als dynamische Stimuli bezeichnet man darüber hinaus zum Beispiel Videos.

Die Visualisierungen von Daten zu den drei genannten Kategorien unterscheiden sich auf Grund der Eigenschaften dieser Daten. Während Bilder und Videos für alle Probanden zum selben Zeitpunkt den gleichen Inhalt haben, ist das bei interaktiven Stimuli nicht der Fall. Weil die einzelnen Probanden den Stimulus beliebig verändern können, lassen sich die für die verschiedenen Probanden aufgezeichneten Eye-Tracking-Daten nicht mehr direkt miteinander vergleichen. Das liegt unter anderem daran, dass die Positionen der AOI interaktiver Stimuli variieren können. Selbst wenn für verschiedene Probanden ähnliche Daten vorliegen, können diese Daten nur dann sinnvoll interpretiert werden, wenn der zugehörige Stimulus zu jedem Zeitpunkt bekannt ist.

Für interaktive Stimuli reicht es demnach nicht mehr aus, lediglich verschiedene Bereiche als *Areas of Interest* zu definieren. Vielmehr müssen alle dargestellten Elemente als Objekte betrachtet werden, deren Position bei jeder Veränderung des zugehörigen Stimulus mit aufgezeichnet wird. Im Grunde sollte jedes Objekt, das in seiner Darstellung durch Interaktion verändert werden kann, als AOI definiert werden. Damit würde man auch für interaktive Stimuli eine Grundlage schaffen, mit welcher sich Scanpaths verschiedener Probanden vergleichen lassen.

2.2.5 Visual Information Seeking Mantra

Eine Visualisierung von Eye-Tracking Daten, wie sie in Kapitel 2.2.1 *Die Visualisierungspipeline* beschrieben wurde, soll von einem Anwender vollständig erfasst und verstanden werden können. Dafür ist es wichtig, dass der Anwender einen Überblick über die vorhandenen Daten bekommt und gleichzeitig die Möglichkeit hat, sich interessante Bereiche genauer anzuschauen. Es ist sinnvoll, dass der Kontext dabei nicht verloren geht, also klar ist, welcher Bereich aus den vorhandenen Daten untersucht wird.

Zu diesem Zweck wurde das Prinzip aller Designrichtlinien für Visualisierungen formuliert. Schneiderman hat es als *Visual Information Seeking Mantra* zusammengefasst und mit folgenden Worten prägnant beschrieben: „*Overview first, zoom and filter, then details on demand.*“ (Schneiderman 1996)

Um als Betrachter eine Visualisierung verstehen zu können, ist es demnach sehr hilfreich, mit dieser auf folgende Art und Weise interagieren zu können:

Overview: Es muss möglich sein, einen Überblick über die vorhandenen Daten zu bekommen. Am besten wird unmittelbar nach Abschluss der Visualisierung eine Übersicht präsentiert. Diese sollte alle Informationen mit hinreichendem Abstraktionsgrad enthalten. Das Ziel ist erreicht, wenn der Anwender in der Übersicht Muster erkennen kann, die durch das Analysieren einzelner Bereiche nicht oder nur schwer hätten erkannt werden können.

Zoom and Filter: Es ist sinnvoll, die Visualisierung für den Anwender durch Vergrößern und Filtern auf die wesentlichen und interessanten Aspekte zu reduzieren. In welchem Rahmen das geschieht, sollte dem Anwender überlassen werden. Das heißt, der Anwender sollte den Grad der Vergrößerung frei wählen können und selbst bestimmen, welche Daten er angezeigt bekommen möchte. Dadurch entsteht die Möglichkeit, *Visual Clutter* interaktiv zu reduzieren oder ganz zu vermeiden.

Details on Demand: Wichtige Informationen und Details sollten eine Visualisierung nicht überladen. Auf der anderen Seite sollte es auch nicht notwendig sein, die Ansicht zu wechseln, um die gesuchten Informationen zu erhalten. Eine Möglichkeit, die diese Anforderung erfüllt, besteht darin, die gewünschten Informationen erst auf Nachfrage beziehungsweise durch entsprechende Interaktion zur Verfügung zu stellen. Eine gängige Vorgehensweise ist das Einblenden von Tooltips. Es ist auch denkbar, zusätzliche Informationen in einem weiteren Fenster anzuzeigen, das der Anwender neben dem Hauptfenster platzieren kann.

2.3 Grafische Benutzungsschnittstelle

Eine *grafische Benutzungsschnittstelle* (engl. Graphical User Interface, abgekürzt: GUI) ist der Teil eines Systems, der einem Anwender zur Interaktion mit diesem System bereitgestellt wird. Die GUI wird hierfür auf einem Monitor dargestellt. Mit den in einer GUI vorhandenen Interaktionselementen (grafischen Symbolen) können Befehle an das zugehörige System übermittelt werden. Die Maus ist das bekannteste Eingabegerät zur Bedienung der grafischen Interaktionselemente.

Außerdem ist innerhalb einer GUI bekannt, welches Element bei der nächsten Interaktion relevant ist, also den Fokus der Anwendung besitzt. Dieses Element wird entsprechend seiner Eigenschaft hervorgehoben, um dem Anwender die Interaktion mit dem Programm durch Tastatureingaben zu vereinfachen.

Weil beim Eye-Tracking üblicher Weise ein einzelnes Programm oder eine Website untersucht wird, ist mit GUI in diesem Kontext die grafische Schnittstelle dieser Software gemeint. Bei einem Eye-Tracking-Experiment sollte darauf geachtet werden, dass sich der Anzeigebereich über den gesamten Monitor erstreckt. Dadurch werden überflüssiges Protokollieren der Position des Anzeigebereichs auf dem Desktop und das damit verbundene Umrechnen der Koordinaten vermieden.

2.3.1 Elemente einer GUI

An dieser Stelle werden herkömmliche Steuerelemente einer GUI beschrieben. Für die Analyse von Eye-Tracking-Daten und der zugehörigen Interaktion eines Probanden ist es wichtig, diese Elemente und damit die Vielzahl der Interaktionsmöglichkeiten mit einem System zu kennen. Einfache Basiselemente werden dabei immer wieder neu zu komplexeren Strukturen zusammengesetzt.

Label: Ein Label ist ein Feld, das aus einfachem Text zur Beschreibung von grafischen Elementen besteht. Es kann in der Regel nicht mit Eingabegeräten fokussiert werden.

Icon: Ein Icon ist ein grafisches Symbol, das meistens eine Funktion – oft aber auch eine Datei oder eine Anwendung – repräsentiert. Icons werden häufig in anderen Elementen aufgegriffen, um diese zu symbolisieren.

Button: Ein Button ist eine Schaltfläche mit verknüpfter Funktion, die durch Klick mit der primären Maustaste auf diese Fläche ausgeführt wird. Die Funktion wird durch ein Icon auf dem Button symbolisiert und / oder mit einem Label (ebenfalls auf dem Button) genauer beschrieben.

Ein Button kann auch als **Schalter (Toggle Button)** fungieren. In diesem Fall wird durch das Betätigen des Buttons ein Zustand aktiviert (eingeschaltet) oder deaktiviert (ausgeschaltet).

Textfeld: Ein Textfeld ist in der Regel ein Feld zur Eingabe von Schriftzeichen. Es wird häufig auch zur textuellen Ausgabe benutzt. Teilweise dient es gleichzeitig der Ein- beziehungsweise Ausgabe, indem zum Beispiel ein Vorschlag des Systems im Textfeld vom Benutzer angepasst werden kann. Form und Größe eines Textfeldes variieren je nach Aufgabenstellung stark.

Menü: Ein Menü ist eine Auswahl der wesentlichen Funktionen eines Programms für den Anwender. Es wird oft als Drop-Down-Liste dargestellt und kommt in der Regel nur einmal in einer Anwendung vor. Die Drop-Down-Liste besteht oft aus untereinander gereihten Buttons, deren verknüpfte Funktion durch Labels repräsentiert wird. Ein Menü kann weitere Untermenüs enthalten.

Das Programmmenü steht in den meisten Fällen während der gesamten Programmausführung zur Verfügung. Entsprechend ist ein Interaktionselement mit fixer Position zum Aufruf des Menüs vorhanden, wodurch sich das Menü stets über denselben Anzeigebereich erstreckt.

Kontextmenü: Als Kontextmenü bezeichnet man eine Auswahl der wesentlichen Funktionen zu einem konkreten Element (beispielsweise einer Grafik). Es wird gewöhnlich mit einem Klick der sekundären Maustaste auf dieses Element aufgerufen und ebenfalls in Form einer Drop-Down-Liste angezeigt. Folglich ist die Position eines Kontextmenüs erst bei der Anzeige bekannt und hängt sowohl von der Position des zugehörigen Elements als auch der Anzahl bereitstehender Funktionen zu diesem ab. Der Grund dafür ist, dass sich die Drop-Down-Liste flexibel nach oben beziehungsweise unten öffnen lassen muss, falls sich das Element am unteren beziehungsweise oberen Rand des Anzeigebereichs befindet.

Symbolleiste: Eine Symbolleiste ist häufig ein schmaler Bereich, der aus Symbolen beziehungsweise grafischen Interaktionselementen besteht, die den Aufruf einzelner Programmfunktionen ermöglichen. Eine Symbolleiste kann an Stelle eines Menüs oder zu dessen Vervollständigung eingesetzt werden. In vielen Fällen erstreckt sich eine Symbolleiste über die gesamte Horizontale beziehungsweise Vertikale am Rand des Anzeigebereichs eines Programms.

Statusleiste: Eine Statusleiste dient der Darstellung des Programmzustands beziehungsweise Programmfortschritts am äußeren, meist unteren Rand des Anzeigebereichs.

Bildlaufleiste: Eine Bildlaufleiste (*Scrollbar*) kann zum vertikalen oder horizontalen Verschieben des angezeigten Inhalts eingesetzt werden. Sie befindet sich in der Regel am Rand des zugehörigen Anzeigebereichs. Die Ausrichtung der Bildlaufleiste entspricht dabei der durch die Interaktion betroffenen Verschieberichtung.

Bildlaufleisten werden auch zum stufenlosen Verstellen von Werten verwendet. In diesem Kontext spricht man von *Schiebereglern* (*Slider / Slide Control*). Beispielsweise lässt sich so der Grad der Vergrößerung einer Darstellung bestimmen.

Kombinationsfeld: Ein Kombinationsfeld ermöglicht die Auswahl eines durch ein Label repräsentierten Objekts aus einer Liste. Es ist je nach Kontext möglich, die Liste durch Eingabe in ein Textfeld zu filtern, zu ergänzen oder zu durchsuchen. Die mehrzeilige Liste wird dabei unter dem Textfeld dynamisch ein- bzw. ausgeblendet und verfügt oft über eine vertikale Bildlaufleiste, um die Zahl gleichzeitig angezeigter Elemente in großen Listen zu begrenzen.

Tooltip: Ein Tooltip blendet ein kleines Fenster mit der Beschreibung eines Elements ein, wenn der Mauszeiger kurzzeitig über diesem Element verweilt. Als interaktives Element, das dynamisch an verschiedenen Positionen ein- beziehungsweise ausgeblendet wird und dabei andere AOI überdeckt, erschwert es die Analyse von Eye-Tracking Daten. Die Länge des beschreibenden Textes bedingt außerdem die Größe des Fensters.

Dialog: Als Dialog werden in der Regel Fenster bezeichnet, die während der Programmausführung eingeblendet werden, um Benutzereingaben einzuholen. Dialoge sind meistens formularartig gestaltet und bestehen aus den bereits vorgestellten Elementen. Charakteristisch sind Buttons, die das Navigieren zwischen einzelnen Schritten eines Dialogs oder das Bestätigen bzw. Abbrechen des Dialogs ermöglichen.

Viele Dialoge sind modal. Das heißt, ein Dialog verhindert die weitere Ausführung des Programms bis der jeweilige Dialog beendet wurde. Dabei spielt es keine Rolle, ob der Dialog vollständig durchlaufen oder vorzeitig abgebrochen wurde.

Canvas: Der Begriff Canvas entstammt dem englischen Wort für Leinwand und bezeichnet einen digitalen Zeichenbereich. Auf diesen kann unter anderem das Ergebnis einer Visualisierung gezeichnet werden. Darüber hinaus können Objekte auf einem Canvas verschoben, rotiert oder skaliert werden.

2.3.2 Grundsätze der Dialoggestaltung

Die vorhandenen Normen zur Gestaltung von Dialogen sind auch für die Gestaltung von Benutzungsschnittstellen sinnvoll. Diese Schnittstellen können in erweitertem Sinne ebenfalls

als Dialog verstanden werden. Diese Normen gewährleisten eine zum Großteil einheitliche Anordnung von Elementen in verschiedenen Programmen.

Teil 110 der DIN EN ISO 9241 beschreibt die Grundsätze der Dialoggestaltung. Diese sind erforderlich, um dem Benutzer eine einfache Bedienung der Schnittstellen von Webseiten und Software zu ermöglichen und ihm damit die Durchführung seiner Aufgaben zu vereinfachen.

Aufgabenangemessenheit: Aufgabenangemessenheit ist dann gegeben, wenn die Software so an die Aufgabe angepasst wurde, dass die Funktionalität und die Interaktionsmöglichkeiten den Eigenschaften der Aufgabe gerecht werden. Wichtig ist, dass der Anwender durch die Software sinnvoll unterstützt wird. Das kann zum Beispiel durch das Vorschlagen geeigneter Standardwerte erfolgen, die der Benutzer dann nicht mehr erneut eingeben muss. Generell gilt, dass unnötige Interaktionen vermieden werden sollen.

Selbstbeschreibungsfähigkeit: Der Benutzer muss stets mühelos erkennen können, in welchem Dialog und an welcher Stelle in diesem Dialog er sich befindet, welche Funktionen zur Verfügung stehen und wie diese Aktionen ausgelöst werden können. Das kann durch eine klare Anzeige von Zustandsänderungen und erforderlichen Schritten erreicht werden.

Lernförderlichkeit: Man spricht von Lernförderlichkeit, wenn der Benutzer beim Erlernen der Interaktionsmöglichkeiten einer Software geführt und unterstützt wird. Um das zu gewährleisten, müssen die verwendeten Konzepte eingehalten werden. Dadurch verhält sich das Programm wie erwartet und vorhandene Funktionen können rasch gefunden werden.

Steuerbarkeit: Es ist wichtig, dass der Anwender eine Interaktionsfolge starten und diese beeinflussen kann. Dazu gehört auch, dass einzelne Eingaben und Aktionen rückgängig gemacht werden können.

Erwartungskonformität: Hier geht es unter anderem um die Einhaltung gängiger und allgemein anerkannter Konventionen sowie Konsistenz in Gestaltung und Bedienung. Für eine hohe Erwartungskonformität müssen vor allem das Arbeitsgebiet und die Kenntnisse der Anwender berücksichtigt werden.

Individualisierbarkeit: Individualisierbarkeit ist vorhanden, wenn es dem Benutzer möglich ist, die Interaktion und die Darstellung von Informationen seinen Bedürfnissen und Fähigkeiten entsprechend anzupassen. Hat der Anwender beispielsweise die Möglichkeit, Tastenkürzel für den Aufruf bestimmter Funktionen zu definieren oder seine Menüs mit für ihn relevanten Buttons selbst zu gestalten, so kann er die Software individualisieren.

Fehlertoleranz: Für den Benutzer ist es von Vorteil, wenn er seine Arbeit trotz fehlerhafter Eingaben ohne oder mit nur sehr geringem Korrekturaufwand fortsetzen kann.

Wurden diese Grundsätze bei der Programmentwicklung berücksichtigt, sind die Voraussetzungen für ein positives Erlebnis während der Arbeit mit der Software geschaffen.

2.4 Usability

Usability (Benutzerfreundlichkeit) wird als ein Qualitätsmerkmal von Software verstanden, das ein Benutzer bei der Interaktion mit dem Programm wahrnimmt. Verhält sich die Anwen-

dung dabei wie erwartet und lässt sie sich zudem einfach bedienen, gilt sie als benutzerfreundlich. Im Vordergrund steht also, wie ein Benutzer die Software erlebt und wie zufrieden er bei und nach der Interaktion mit dem Programm ist. Dieses Verständnis geht über die reine Gebrauchstauglichkeit als Mindestanforderung hinaus.

Eine Schwierigkeit ist, dass bei der Entwicklung der Software die Bedingungen, die beim späteren Einsatz vorherrschen, nicht vollständig bekannt sind. Für ein positives Erlebnis ist entscheidend, welche konkrete Aufgabe bearbeitet wird, welche Kenntnisse der jeweilige Anwender hat und welche äußeren Gegebenheiten vorliegen. Beispielsweise kann sich die zu große Entfernung zum Monitor insofern auf die Wahrnehmung auswirken, dass angezeigte Inhalte nur schwer zu erkennen sind. Gleichzeitig bedingen die Gewohnheiten verschiedener Anwender unterschiedliche Erwartungshaltungen. Dazu zählt zum Beispiel die Erreichbarkeit aller Funktionen per Button, Menüeintrag, Kontextmenü und Tastaturkürzel.

Software hat als Werkzeug für eine bestimmte Aufgabe das Potential, die Aufgabe erheblich zu verändern. Früher war es üblich, Musikdateien nach Interpreten sortiert in Ordnern abzulegen. Eine Synchronisation dieser Dateien zwischen verschiedenen Computern war durch manuelles Kopieren ausgewählter Ordner möglich. Moderne Programme synchronisieren diese Inhalte dagegen automatisch. Allerdings muss die Synchronisation jetzt verwaltet werden, weil nicht jede Änderung auf allen Geräten erwünscht ist oder die Synchronisation zu einem bestimmten Zeitpunkt erfolgen soll. Der Anwender muss sich jetzt also nicht mehr um das Kopieren der Dateien kümmern, sondern um das Verwalten der Automatismen.

Essentiell für die Interaktion mit einem System ist unter anderem das richtige Feedback. Dem Anwender muss also mitgeteilt werden, dass er eine Interaktion gestartet oder eine bestimmte Funktion ausgelöst hat. Beispielsweise wird das Laden eines Programms von älteren Betriebssystemen durch eine Art virtuelle Sanduhr dargestellt. Dadurch wird dem Benutzer verdeutlicht, dass die gewünschte Programmausführung vorgenommen wird. Das wiederum unterbindet die mehrfache Ausführung des Programms. Der Anwender muss erkennen können, was passiert und warum etwas passiert. Außerdem sollte gleiches Verhalten des Anwenders gleiches Verhalten des Systems bedingen. Der Anwender soll die Interaktion reproduzieren können.

Ein wesentlicher Aspekt für die Benutzerfreundlichkeit ist außerdem die Berücksichtigung der Fähigkeiten und Fertigkeiten der Anwender. Eine natürliche und vertraute Interaktion mit einer Software lässt sich herstellen, indem verbreitete Methoden aufgegriffen und verwendet werden. Beispiele hierfür sind ein Menü am oberen Rand einer Anwendung oder eine hierarchische Struktur am linken Rand einer Visualisierung. Durch die Verwendung verbreiteter Methoden wird ein Erlernen neuer Ansätze vermieden.

Wie weit das gehen kann, macht die Verwendung einer herkömmlichen und weit verbreiteten QWERTY- beziehungsweise QWERTZ-Tastatur deutlich. Die Anordnung der Tasten ist nicht optimal, weil oft verwendete Buchstaben weit auseinander liegen. Dennoch wird die nahezu gleiche Tastenanordnung in verschiedenen Sprachräumen verwendet. Es wurden einige Tastaturen entwickelt, bei denen dieses Defizit für eine bestimmte Sprache behoben wurde. Darunter fällt die Dvorak-Tastaturbelegung, mit der gleiche Aufgaben möglicherweise effizienter bearbeitet werden können. Allerdings konnten sich solche angepassten Tastaturen bisher nur

bedingt oder gar nicht durchsetzen. Ein Grund dafür könnte die enorme, erforderliche Umgewöhnung für die Anwender sein.

Es ist wichtig zu erkennen, dass Benutzerfreundlichkeit nicht einschichtig ist und es kein pauschales Rezept gibt, das einfach angewandt werden kann. Um dem Anwender eine zufriedenstellende Lösung in Form eines Programms oder einer Visualisierung zu bieten, kann ein erster Lösungsansatz in einem iterativen Prozess durch die Evaluation mit Eye-Tracking ständig verbessert werden. Eine zufriedenstellende Lösung für die Problemstellung aus dieser Diplomarbeit besteht darin, dem Anwender eine uneingeschränkte und komfortable Analyse der vorhandenen Daten zur Augenbewegung und Interaktion zu ermöglichen (siehe Kapitel 3 *Aufgabenstellung und Lösungsansatz*).

Eye-Tracking kann gezielt zum Erfassen der Benutzerfreundlichkeit einer Software eingesetzt werden, indem zum Beispiel gezeigt wird, wie die Anwender mit den zur Verfügung stehenden Funktionen umgehen und ob sie diese überhaupt wahrnehmen. Die im Rahmen dieser Diplomarbeit entwickelte Visualisierung und deren Implementierung berücksichtigen wesentliche Aspekte aus dem Bereich Usability. Auch das in Kapitel 2.2.4 *Visual Information Seeking Mantra* erwähnte Prinzip ist dabei erklärtes Ziel und förderlich, um eine hohe Benutzerfreundlichkeit zu gewährleisten.

2.4.1 Interaktion

Menschen in unserer Gesellschaft interagieren täglich mehrfach mit verschiedensten Systemen. Vom *Drücken* des Weckers über automatische Schiebetüren, der Verwendung von Apps auf mobilen Endgeräten, der Arbeit mit und am Computer bis zur Bestellung von Waren auf einer Online-Plattform unterscheiden sich diese Interaktionen durch ihre Dauer und Komplexität erheblich. Während das Betätigen eines Weckers nicht unbedingt als Interaktion mit einem System wahrgenommen wird, kann die Bedienung eines Ticketautomaten die Benutzer haltlos überfordern. Die Wahrnehmung einer Interaktion hängt dabei stark von ihrer Gestaltung ab.

Die wesentliche Fragestellung für die Entwicklung eines Systems sollte zum Ziel haben, dass sich dieses System so verhält, wie der Anwender es erwartet. Die Gestaltung kann dabei so ausgereift sein, dass die Interaktion gar nicht mehr als solche wahrgenommen wird. Beispielsweise öffnet sich eine Schiebetür rechtzeitig bevor man hindurchgeht und schließt sich ohne eigenes Zutun.

Überträgt man den Gedanken auf die Visualisierung von Eye-Tracking-Daten, kann man sich eine Lösung vorstellen, bei der die relevanten Details zu einem interessanten Objekt immer genau dann verfügbar sind, wenn diese für den Anwender hilfreich sind. Ausgehend davon, dass ein Anwender den Mauszeiger über solche Objekte bewegt, kann man relevante Informationen entweder direkt am Objekt einblenden oder in einem dafür vorgesehen Bereich dynamisch anpassen, zum Beispiel in einer Detailansicht am rechten Rand. Das heißt, der Anwender kann mit den Objekten interagieren, hat aber immer, wenn er in die Detailansicht schaut, die zugehörigen Informationen zum richtigen Objekt.

Die Aufzeichnung der Interaktion gestaltet sich zum Teil analog zum Eye-Tracking – sie muss nur stattdessen die Position des Mauszeigers protokollieren. Diese Position ist einem System während der Bedienung durch einen Anwender zu jedem Zeitpunkt bekannt und kann

ohne zusätzliche Hardware erfasst werden. Genau wie eine Fixation mit dem Auge, kann auch mit der Maus ein bestimmter Bereich fixiert werden. Im Rahmen dieser Arbeit wird das gezielte Fixieren des Mauszeigers auf einer AOI als *Mausfixierung* oder *Fixation mit der Maus* bezeichnet. Die tatsächliche Interaktion geht selbstverständlich über das Bewegen des Mauszeigers hinaus und beinhaltet das Bedienen angezeigter Steuerelemente. Die Bedienung kann auch über andere Eingabegeräte erfolgen – zum Beispiel über eine Tastatur.

Die Interaktion mit einem Stimulus dient in der Regel der Informationsgewinnung. Wird dieser gedankliche Prozess eines Probanden von ihm laut geäußert und das Gesagte dokumentiert, bezeichnet man das daraus entstandene Ergebnis als *Think-Aloud-Protokoll*.

Interaktionskategorien

Für eine gelungene Umsetzung des in Kapitel 2.2.4 *Visual Information Seeking Mantra* genannten, gleichnamigen Prinzips, ist die Verwendung bekannter Interaktionsmöglichkeiten hilfreich. Eine zielführende Informationsvisualisierung sollte so mit Interaktionsmöglichkeiten angereichert sein, dass die Visualisierung die Erledigung der damit verbundenen Aufgaben erheblich erleichtert. Gemeint ist die Anpassung der Repräsentation vorhandener Daten entsprechend der Eingaben des Benutzers. Erst wenn der Benutzer die Informationen erhält, die er zur Analyse benötigt, ist die zugrundeliegende Visualisierung erfolgreich. Die Auswahl eines interessanten Bereichs aus einer Übersicht durch die Interaktionen Zoomen und Filtern und das anschließende Bereitstellen von Details zu Elementen in diesem Bereich entspricht der *Visual Information Seeking Mantra*. Es liegt insbesondere die beschriebene Anpassung der Datenpräsentation vor.

Welche Möglichkeiten zur Interaktion mit Visualisierungen bestehen, lässt sich in Kategorien zusammenfassen. Yi et al. unterscheiden die verschiedenen Kategorien anhand der vom Anwender verfolgten Absicht wie folgt (Yi, et al. 2007):

Selektieren: Selektieren bezeichnet das Auswählen und Markieren eines interessanten Objekts, um dieses Objekt im Blick behalten zu können. Das ist vor allem dann hilfreich, wenn sich die Visualisierung durch weitere Interaktionen verändert. Auf diese Weise lässt sich zum Beispiel ein Objekt, dessen Position in einer Visualisierung für verschiedene Zeitschritte variiert, über diese Zeitschritte hinweg verfolgen.

Explorieren: Weil ein Anwender durch Interaktionen wie Zoomen oft nur eine Untermenge der vorhandenen Daten betrachtet, ist es immer wieder erforderlich, diese Untermenge neu zu wählen oder zu wechseln. Das heißt, der angezeigte Bereich muss bei gleichbleibender Vergrößerung verschoben werden können, um dem Anwender einen anderen Bereich zeigen zu können.

Umgestalten: Mit dieser Interaktionstechnik kann der Benutzer die Anordnung grafischer Elemente einer Präsentation verändern. Das ermöglicht eine neue Sicht auf die angezeigten Daten, welche das Auffinden neuer Zusammenhänge zwischen den Daten erlaubt.

Kodieren: Das Kodieren erlaubt den Wechsel auf eine andere visuelle Repräsentation als die bisherige. Das betrifft in erster Linie die Eigenschaften der angezeigten Elemente. Dazu zählen unter anderem Farbe, Form und Größe der Elemente. Der Wechsel von einem Balkendiagramm zu einer Visualisierung der gleichen Daten als Kreisdiagramm ist ein Beispiel für ein solches Kodieren. Ähnlich wie beim Umgestalten erlaubt das Kodieren eine neue Sicht auf die Daten und das damit verbundene Auffinden bisher unbemerkter Zusammenhänge.

Abstrahieren / Zusammenfassen: Durch Abstrahieren beziehungsweise Zusammenfassen kann der Grad des Informationsgehalts bestimmt werden. Genauer ausgedrückt, kann damit zum Beispiel von einer abstrakten Darstellung der Daten in einer Übersicht auf die Darstellung einzelner Bereiche bis hin zur Darstellung der Details eines Elements übergegangen werden. Konkret können also Details ein- beziehungsweise ausgeblendet werden. Ein einfaches Beispiel für diese Interaktionskategorie ist das Zoomen – vorausgesetzt, dass die Präsentation dabei in ihrer Art nur unwesentlich verändert wird.

Filtern: Das Filtern dient dem Anpassen der angezeigten Daten und ermöglicht unter anderem das Ausblenden nicht relevanter Informationen und damit eine stärkere Fokussierung auf wichtige Aspekte. Im Wesentlichen werden beim Filtern Kriterien festgelegt, die die einzelnen Daten erfüllen müssen, um angezeigt zu werden. Ein einfaches Beispiel ist das Festlegen einer Zeitspanne, wodurch nur die Daten angezeigt werden, die innerhalb dieser Zeitspanne anzuordnen sind.

Verknüpfen: Das Verknüpfen ist besonders für die Verwendung mehrerer, verschiedener Ansichten oder Präsentationen des gleichen Datenbestands geeignet (Baldonado, Woodruff und Kuchinsky 2000). Damit können interessante Objekte durch Verknüpfungen in anderen Darstellungen oder Ansichten wiedererkannt und hervorgehoben werden (*Brushing & Linking*). Außerdem können zusammenhängende Objekte gekennzeichnet werden, wie beispielsweise alle Kanten eines bestimmten Knotens.

Diese Interaktionskategorien lassen sich als „*High-Level-Interaktion*“ verstehen, indem sie die Ziele des Benutzers beschreiben. Einzelne Aktionen wie das Anklicken eines Objekts zu dessen Selektion oder das eigentliche Scrollen in einer Präsentation, um den Datenbestand zu explorieren, sind dagegen die entsprechenden „*Low-Level-Interaktionen*“.

Bei der Aufzeichnung von Eye-Tracking-Daten ist es sinnvoll, Ereignisse wie ein Klicken mit der Maus zu protokollieren, um später auf die damit verbundenen *High-Level-Interaktionen* schließen zu können. Bei der Interpretation muss allerdings darauf geachtet werden, dass es durchaus möglich ist, dass zwei unterschiedliche *Low-Level-Interaktionen* der gleichen *High-Level-Interaktion* entsprechen.

3 Aufgabenstellung und Lösungsansatz

3.1 Hintergrund

Im Zentrum aktueller Forschungsfragen auf dem Gebiet der Visualisierung stehen mit zunehmendem Maße Eye-Tracking-Experimente, die verschiedene Visualisierungstechniken auf ihre Benutzerfreundlichkeit und Aufgabenangemessenheit hin untersuchen oder miteinander vergleichen. Interaktive Stimuli wie zum Beispiel interaktive Knoten-Kanten-Diagramme, Webseiten oder komplexe Interaktionsmechanismen stellen dabei eine Herausforderung dar, weil die Eye-Tracking-Daten mehrerer Benutzer nicht ohne weiteres miteinander verglichen werden können. Ein Vergleich der Scanpaths ist aufgrund der Interaktion nicht mehr möglich, weil jeder Benutzer sie individuell durchführt. Deshalb werden neue visuelle Analysemethoden benötigt, um diesen Stimulustyp zu analysieren.

3.2 Aufgabenstellung

Das Ziel meiner Arbeit ist, ein geeignetes Visualisierungskonzept für die Analyse interaktiver Stimuli zu entwickeln. Dazu wird zunächst eine Recherche zu aktuellen Visualisierungstechniken für Eye-Tracking-Daten durchgeführt. Auf Basis einer Evaluation existierender Techniken wird ein Konzept für eine Visualisierungstechnik für interaktive Stimuli entwickelt. Anschließend werden die wichtigsten Aspekte des Konzepts prototypisch implementiert, um die Funktionsweise des Konzepts zu demonstrieren.

Der Diplomarbeit liegen folgende Arbeitsschritte zu Grunde:

- Recherche zu existierenden Ansätzen der visuellen Analyse und Techniken
- Entwicklung eines Konzepts zur visuellen Analyse von Eye-Tracking-Experimenten mit interaktiven Stimuli
- Implementierung des Ansatzes in Form eines Prototypen
- Demonstration des Prototypen anhand eines Szenarios

Um ein schlüssiges Konzept entwickeln zu können, muss außerdem ermittelt werden, welche Rohdaten sinnvoll kombiniert werden können. Es muss geklärt werden, ob beziehungsweise wie diese Daten für interaktive Stimuli erfasst werden können.

In *Abbildung 10: Inhalte dieser Diplomarbeit* werden die durchzuführenden Schritte als Bausteine betrachtet und entsprechend zusammengesetzt. Zuerst muss festgelegt werden, welche Daten beim Eye-Tracking für interaktive Stimuli erfasst werden können und müssen. Daraus entsteht das Datenmodell, für das eine Visualisierung gesucht wird. Ein anhand dieses Modells entwickeltes Visualisierungskonzept wird dann prototypisch implementiert. Darüber hinaus gilt es zu zeigen, dass der Prototyp das Potential besitzt, die gesammelten Daten so aufzubereiten, dass die Zusammenhänge zwischen den Daten sinnvoll analysiert werden können.

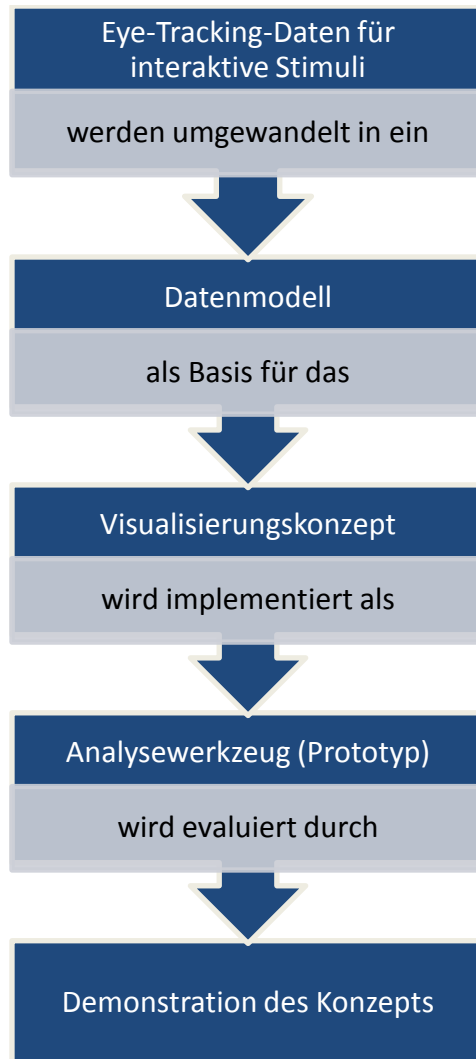


Abbildung 10: Inhalte dieser Diplomarbeit

Die erforderlichen Zwischenstände und die Übergänge zwischen diesen in einer sinnvollen Abfolge. Die Zwischenergebnisse sind dunkelblau hinterlegt, die durchzuführenden Schritte zum Erreichen dieser mit hellem blau gekennzeichnet.

3.3 Lösungsansatz

Die Entwicklung eines Visualisierungskonzepts für interaktive Stimuli erfordert, dass neben den Eye-Tracking-Daten auch die Daten zur Interaktion vorhanden sind, mit denen sich die Veränderung des Stimulus rekonstruieren lässt. Erst die Berücksichtigung der Veränderung des Stimulus ermöglicht es, eine Basis für den Vergleich der Eye-Tracking-Daten verschiedener Probanden einer Studie zu schaffen. Deshalb ist es notwendig, eine Eye-Tracking-Studie zu interaktiven Stimuli so zu erweitern, dass die Daten zur Interaktion mit erfasst werden. Dafür muss in der Regel das Programm angepasst werden, dessen Anwendung durch den Benutzer mit Eye-Tracking beobachtet wird. Es muss die erforderlichen Daten zur Interaktion und zu den gezeigten Inhalten protokollartig aufzeichnen. Zu diesem Zweck ist eine Identifikation aller Objekte, die durch Interaktion im Stimulus angezeigt werden können, erforderlich. Es bietet sich an, jedes Objekt als AOI zu betrachten, deren Größe, Position und Sichtbarkeit im Laufe der Zeit verändert werden kann.

Ein interaktiver Stimulus kann nicht vom Eyetracker in AOI unterteilt werden, weil der Eyetracker die gezeigten Inhalte für gewöhnlich nicht erkennen und klassifizieren kann. Es ist erforderlich, dass die erfassten Eye-Tracking-Daten auf die einzelnen AOI abgebildet werden können. Das bedeutet, die aufgezeichneten Daten müssen alle Informationen enthalten, die notwendig sind, um die Position der Augen auf dem zum jeweiligen Zeitpunkt gültigen Stimulus korrekt den einzelnen AOI zuordnen zu können.

Für frühere Studien, auf deren Daten im Rahmen dieser Arbeit zugegriffen werden konnte, liegen solche zusätzlich erforderlichen Daten nicht vor. Deshalb musste eine anstehende Studie so erweitert werden, dass die geforderten Daten bei der Durchführung der Studie mit erfasst werden.

Die nachfolgende Aufzählung gibt einen Überblick über die während des Interaktionszeitraums zu erfassenden Daten. Dabei wird für alle zu erfassenden Daten stets die Systemzeit hinterlegt, um eine Synchronisation der verschiedenen Datentypen zu ermöglichen.

- Eye-Tracking-Daten
 - Zeitstempel und Koordinaten des Blickpunkts
- Low-Level-Interaktionen (Tastatur- und Mauseingaben)
 - Zeitstempel und Koordinaten des Mauszeigers
 - Objekt-ID der betroffenen AOI
- Objektmodell aller durch Interaktion entstandenen Stimuli
 - Gültigkeitszeitraum bestehend aus Zeitstempeln für Start und Ende
 - Koordinaten aller Objekte beziehungsweise AOI
- Think-Aloud-Protokoll
 - Zeitstempel und Aufzeichnung

Für eine Analyse verschiedener Visualisierungen mittels der genannten Daten muss das Konzept, neben einer Gegenüberstellung der Positionen des Mauszeigers und der Blickpunkte eines Probanden für den gesamten Interaktionszeitraum, auch den Vergleich dieser Informationen für mehrere Probanden unterstützen. Das bedingt die Darstellung aller von den Probanden hervorgerufenen Veränderungen des ihnen gezeigten Stimulus. Werden die verschiedenen Interaktionsmöglichkeiten visuell aufbereitet, können aus ähnlichen Abfolgen von Interaktionen und den daraus entstandenen Zuständen des Stimulus Muster zum Vorgehen bei der Informationsgewinnung abgeleitet werden.

Um diese Vorstellungen zu realisieren, muss das Konzept, neben einer Übersicht über die Daten, auch die in Abschnitt *Interaktionskategorien* in Kapitel 2.4.1 *Interaktion* genannten Interaktionsmöglichkeiten zur Umsetzung des in Kapitel 2.2.4 *Visual Information Seeking Mantra* beschriebenen Prinzips zur Informationsgewinnung berücksichtigen.

4 Relevante Arbeiten

In diesem Kapitel werden vorhandene Arbeiten vorgestellt, die sich mit ähnlichen Aspekten befassen haben, wie sie in meiner Arbeit betrachtet werden. Thematik meiner Diplomarbeit ist die Vergleichbarkeit von Eye-Tracking-Daten zu interaktiven Stimuli mit einer geeigneten Visualisierung. Das beinhaltet zum einen die zusätzliche Erfassung und Auswertung von Daten zur Interaktion und zum anderen die Visualisierung von Interaktionsdaten selbst. Wichtig ist insbesondere die Synchronisation von Eye-Tracking- und Interaktionsdaten. Darüber hinaus wird die geforderte Vergleichbarkeit vor allem durch eine Visualisierung ermöglicht, die die Daten mehrerer Probanden umfasst.

Ein weiterer Aspekt ist die Verwendung von AOI gegenüber einer Betrachtung einzelner Blickpunkte. Für Stimuli wie interaktive Visualisierungen sind vor allem die von den Probanden fixierten Objekte respektive AOI von Bedeutung. Der Vergleich einzelner Punkte ist hier nur bedingt hilfreich. Demnach sind vor allem die Visualisierungstechniken interessant, die sich mit der Darstellung der von mehreren Probanden fixierten AOI befassen (siehe Kapitel 2.2.2 *Visualisierung von Eye-Tracking-Daten*).

In einem ersten Schritt zur gleichzeitigen Erfassung von Maus- und Augenbewegung haben Chen et al. den Zusammenhang beider Bewegungen beim Surfen im Web untersucht (Chen, Anderson und Sohn 2001). Ergebnisse der Arbeit waren vor allem statistische Auswertungen über diesen Zusammenhang. Eine Visualisierung der Daten anhand der Aufzeichnung eines einzelnen Probanden erfolgte, indem jeweils ein Scanpath beziehungsweise Cursorpfad für Maus- und Augenbewegungen des Probanden eingezeichnet wurde. Ziel der Arbeit war weniger das Auffinden von Mustern bei der Informationsgewinnung, wie es für meine Arbeit von Bedeutung ist, als vielmehr der direkte Zusammenhang zwischen Maus- und Augenbewegung. Eine Erkenntnis war die hohe Wahrscheinlichkeit dafür, dass Blickpunkt und Position des Mauszeigers nah beieinander liegen, wenn der Mauszeiger auf einem wichtigen Element verweilt. Die Autoren folgerten daraus, dass die Aufzeichnung der Daten zur Mausbewegung als Alternative zum Eye-Tracking für die Bewertung von *Usability* und die Beurteilung der Gestaltung von Benutzungsschnittstellen eingesetzt werden könnte.

Mit dem Tool KiEV (siehe *Abbildung 11: Snapshot des Programms KiEV*) wurde eine Möglichkeit zur gleichzeitigen Visualisierung von Eye-Tracking-Daten und Tastenanschlägen geschaffen (Špakov und Rähä 2008). Die Daten dafür entstanden beim Übersetzen von Texten. Das Übersetzen besteht sowohl aus einer lesenden als auch aus einer schreibenden Tätigkeit. Dabei werden die erkannten Fixationen mit den einzelnen Wörtern des zugrundeliegenden Texts verknüpft. Ziel war es, Zusammenhänge zwischen der Augenbewegung und dem Tippen zu untersuchen. Dafür sollten vor allem solche Daten hervorgehoben werden, die sich von denen des gewöhnlichen Übersetzens des Textes unterscheiden.

Um eine Datenbasis zur Auswertung mit KiEV zu schaffen, wurden die Logdateien der zur Übersetzung verwendeten Programme ausgelesen. Darin waren der zu übersetzende Text sowie alle Tastenanschläge des Übersetzers vermerkt. Zusätzlich wurde ein Programm verwendet, um die erfassten Blickpunkte noch während der Aufnahme den angezeigten Wörtern zuzuordnen und diese mit den entsprechenden Eye-Tracking-Daten zu protokollieren. Nach Angaben der Autoren war es allerdings nicht gelungen, die Fixationen auf Wörter eines Textes

abzubilden, der dynamisch verändert wurde. Weitere Informationen zu den einzelnen Wörtern können mithilfe von Tooltips eingeblendet werden. Dazu zählen unter anderem die Anzahl der Fixationen, die durchschnittliche Fixationsdauer und die Dauer der ersten Fixation.

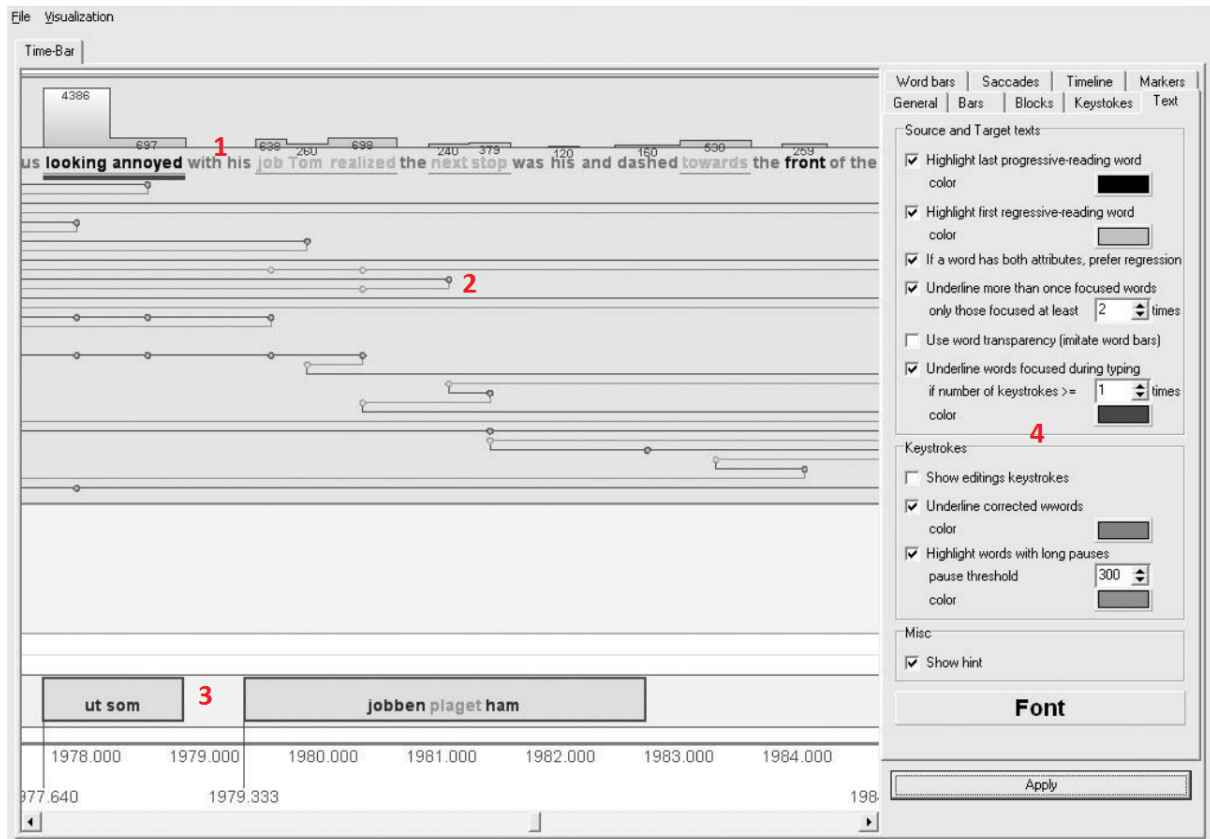


Abbildung 11: Snapshot des Programms KiEV

Programmfenster in KiEV mit auf einer Zeitleiste angeordneter, horizontaler Balken. Auf der rechten Seite befinden sich diverse Einstellungsmöglichkeiten (4). Die Balken repräsentieren Daten zu den Fixationen von Wörtern des Ursprungstextes (1), des übersetzten Textes (3) und des Fortschritts der Übersetzung. Der große Bereich in der Mitte zeigt die Fixationen der einzelnen Wörter (2). (Špakov und Riihä 2008)

Mit KiEV wurde eine Visualisierung realisiert, mit der die Fixationen auf verschiedenen AOI in einem Stimulus und die damit verbundenen Interaktionen gleichzeitig dargestellt werden. Ein ähnliches Ziel soll auch mit meiner Arbeit erreicht werden.

Eine Möglichkeit, Interaktionsschritte bei der Informationsgewinnung aus Visualisierungen grafisch darzustellen wurde bereits von Groth und Murphy prototypisch implementiert (Groth und Murphy 2004). Dieser Prototyp enthält eine Visualisierung, die zeigt, wie die Informationen gewonnen wurden. Ein Ausschnitt davon ist in *Abbildung 12: Der Interaktionsgraph nach Groth und Murphy* zu sehen. Jeder Interaktionsschritt kann hierfür zusätzlich mit textuellen Anmerkungen annotiert werden – es wird eine Art *Think-Aloud-Protokoll* in die Visualisierung integriert. Dadurch kann man die Vorgehensweisen bei der Informationsgewinnung darstellen, um diese später auf andere Datensätze anzuwenden. Das ist eine Basis dafür, verschiedene solcher Vorgehensweisen bewerten und vergleichen zu können.

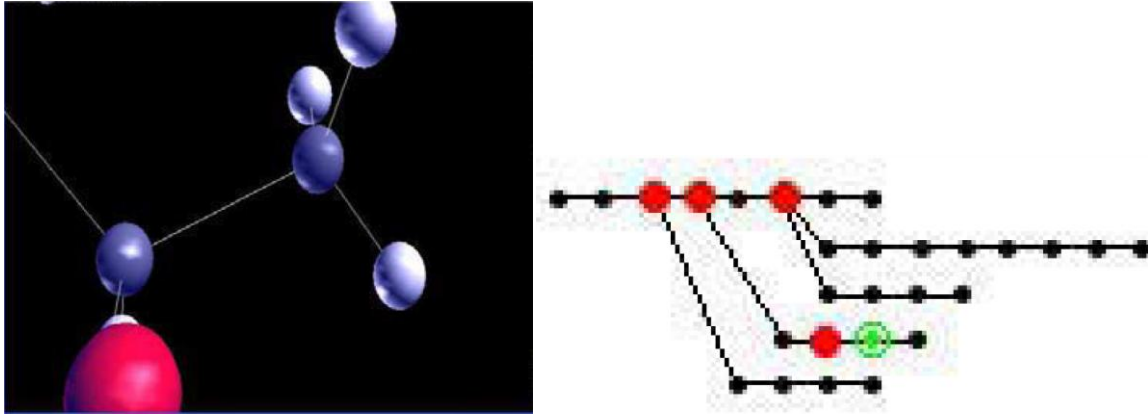


Abbildung 12: Der Interaktionsgraph nach Groth und Murphy

Ausschnitte der Darstellung des Interaktionsgraphen im Prototyp von Groth Murphy. Die Knoten symbolisieren die verschiedenen Zustände einer Visualisierung, während die Kanten die Übergänge zwischen diesen Zuständen anzeigen. Jeder Übergang entspricht dabei einem einzelnen Interaktionsschritt, wie zum Beispiel Zoomen in der Visualisierung. Der Graph auf der rechten Seite dient als Übersicht und erlaubt die Annotation der Zustandsknoten. (Groth und Murphy 2004)

Auch wenn sich eine solche Darstellung gut für Interaktionsdaten zu interaktiven Stimuli eignet, ist sie nicht für die Visualisierung von Daten mehrerer Probanden oder den direkten Vergleich dieser Daten gedacht.

Pohl et al. haben eine ausgiebige Auswertung von Logdateien zur Interaktion mit Informationsvisualisierungen durchgeführt (Pohl, Wiltner und Miksch 2010). Die Einträge der Logdateien enthielten neben dem Zeitstempel nicht nur die vom Benutzer durchgeführte Interaktion, sondern auch Informationen darüber, wie diese durchgeführt wurde. Die zugehörige Visualisierung, die den Probanden zur Verfügung stand, erlaubte die Anordnung von grafischen Elementen, die Personen und Fragen repräsentierten. Die Interaktionen umfassten dabei zum Beispiel das Aktivieren beziehungsweise Deaktivieren verschiedener Visualisierungsmethoden, wie und wohin einzelne Elemente bewegt wurden und wann ein Tooltip ein- oder ausgeblendet wurde. Gleichzeitig umfasste die Protokollierung, ob die Interaktion über einen Menüeintrag aufgerufen oder direkt am Element (zum Beispiel durch Drag & Drop) durchgeführt wurde. Die Elemente wurden anhand ihrer Bezeichnung unterschieden.

Allerdings erfolgte die Auswertung der Resultate nicht primär durch eine zusammenfassende, grafische Visualisierung der ermittelten Daten sondern vor allem mithilfe errechneter Werte. Trotzdem wurden alle Logdateien einzeln untersucht, um vorhandene Muster erkennen zu können (siehe *Abbildung 13: Ausschnitt der Logdatei eines Probanden*). An dieser Stelle machte sich wohl das Fehlen einer geeigneten Visualisierungstechnik für Interaktionsdaten, bemerkbar – es wurden lediglich die vorhandenen Protokolldateien mit Farben für die verschiedenen Interaktionskategorien angereichert. Ungeachtet dessen führte die Untersuchung der Dateien zu der Erkenntnis, dass alle Probanden ähnliche Abfolgen von Interaktionen angewandt hatten, um die gleiche Aufgabe zu lösen.



Abbildung 13: Ausschnitt der Logdatei eines Probanden

Die einzelnen Zeilen der Logdatei sind entsprechend dem Typ der Interaktion unterschiedlich eingefärbt. Die entstandenen Blöcke (am linken Rand) können auf diese Weise zu Abfolgen zusammengefasst und miteinander verglichen werden. (Pohl, Wiltner und Miksch 2010)

Mit DynAOI (siehe *Abbildung 14: Die Funktionsweise von DynAOI*) wurde ein Werkzeug entwickelt, mit welchem AOI in dynamischen Stimuli erkannt und verfolgt werden können (Papenmeier und Huff 2012). Dazu wird für jeden dynamischen Stimulus (in diesem Fall für jedes Video) ein dreidimensionales Modell des gezeigten Inhalts hinterlegt. Die erfassten Eye-Tracking-Daten können so auf die sich bewegenden Objekte im Video abgebildet werden. Weil es sich bei den gezeigten Videos in der Regel um Animationen handelte, waren solche Modelle bereits vorhanden.

Die Verwendung eines Modells zur Identifikation von AOI in dynamischen oder interaktiven Stimuli ist eine Möglichkeit, vorhandene Eye-Tracking-Daten auf diese Inhalte abzubilden. Es ist gleichzeitig ein effektives Mittel für interaktive Stimuli, um eine Vergleichsbasis der Eye-Tracking-Daten verschiedener Probanden zu schaffen.

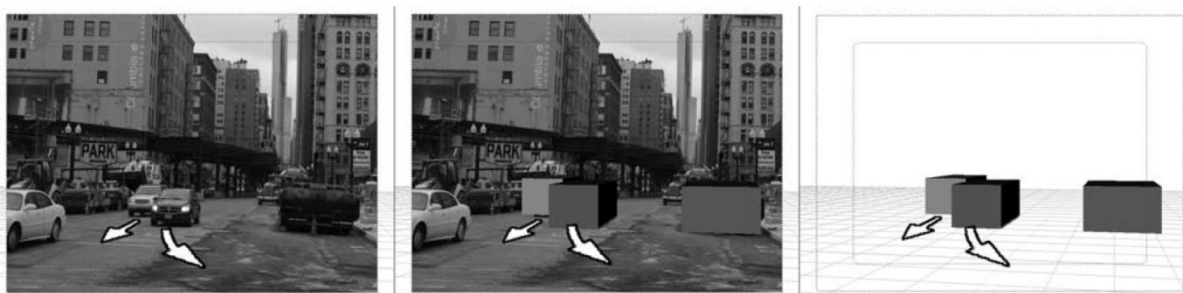


Abbildung 14: Die Funktionsweise von DynAOI

Der Inhalt eines Videos (links) mit einer realen Szene, die Abbildung relevanter Inhalte auf dreidimensionale Objekte (mittig) und die Klassifizierung dieser Objekte in einer Modelldarstellung als statische beziehungsweise dynamische (durch Pfeile symbolisiert) AOI (rechts). (Papenmeier und Huff 2012)

Die vorgestellten Arbeiten beinhalten verschiedene Lösungen für die Analyse von Daten, wie sie bei der Interaktion mit interaktiven Stimuli anfallen. In jeder dieser Arbeiten wurde ein für meine Arbeit wichtiger Aspekt behandelt. Für mein Konzept waren die Berücksichtigung von Zusammenhängen zwischen Augen- und Mausbewegungen (Chen, Anderson und Sohn 2001), die gleichzeitige Visualisierung von Eye-Tracking- und Interaktionsdaten (Špakov und Rähä 2008), die geeignete Darstellung der durch Interaktion erreichten Zustände eines interaktiven Stimulus und der Übergänge zwischen diesen (Groth und Murphy 2004), die aussagekräftige Protokollierung und Auswertung von Interaktionsdaten (Pohl, Wiltner und Miksch 2010), und das Erstellen eines Modells zur Abbildung von Fixationen auf dynamische (beziehungsweise interaktive) Stimuli (Papenmeier und Huff 2012) relevant.

5 Lösungskonzept

In diesem Kapitel stelle ich das im Rahmen meiner Diplomarbeit erarbeitete Konzept für die Analyse von Eye-Tracking-Daten und Interaktionsdaten für interaktive Stimuli vor. Entscheidend für die Analyse ist, dass diese Daten gemeinsam in einer einzigen Visualisierung erkennbar sind. Dadurch sollen Zusammenhänge zwischen der Interaktion eines Probanden und den von ihm aufgenommenen Informationen aufgezeigt werden. Des Weiteren ist es essentiell, dass die visualisierten Daten verschiedener Probanden vergleichend gegenüber gestellt werden können. Erst ein solcher Vergleich ermöglicht das Auffinden von Mustern in den Daten zur Interaktion und zum Eye-Tracking.

Zuerst erläutere ich die Idee des Konzepts sowie wichtige Voraussetzungen für die Umsetzung. Teil dieser Voraussetzungen ist auch die Analyse des interaktiven Stimulus, zu dem die Daten für eine Visualisierung erfasst werden. Anschließend erfolgt die Beschreibung der Entstehung des Konzepts unter Berücksichtigung der Voraussetzungen und Beachtung sinnvoller Interaktionsmöglichkeiten. Die Gegenüberstellung der Anforderungen an das Konzept und den Merkmalen des entwickelten Konzepts, bildet den Abschluss des Kapitels.

5.1 Anforderungen

Neben den durch Eye-Tracking aufgezeichneten Daten müssen die Daten zur Interaktion erfasst werden. Zu letzteren gehört unter anderem die Dokumentation der Mausposition zu jedem Zeitpunkt. Diese Daten können nicht vom Eyetracker erfasst werden, weil der gezeigte, interaktive Stimulus ständig verändert werden kann. Aus diesem Grund ist die Definition von AOI für einen interaktiven Stimulus am Eyetracker nicht möglich. Vielmehr muss ein Modell für alle Zustände des Stimulus erstellt werden, das eine Abbildung der vom Eyetracker erfassten Blickpunkte auf die im Stimulus gezeigten Objekte erlaubt. Diese Objekte müssen identifiziert und sinnvoll in AOI unterteilt werden, um die erfassten Daten in einer AOI-basierten Visualisierung auf solche Objekte abzubilden.

Die für diese Arbeit wichtigste Kategorie interaktiver Stimuli sind interaktive Visualisierungen. Im Grunde soll das entwickelte Konzept auf alle Arten von interaktiven Stimuli wie Webseiten, Videos oder sonstigen Benutzungsschnittstellen angewendet werden können. Der Fokus meiner Arbeit liegt allerdings auf Programmen, die zur Analyse von Daten mithilfe von Visualisierungen gedacht sind.

Zu diesem Zweck wurde das Programm *VarifocalReader* (Koch, et al. 2014), das die genannten Anforderungen erfüllt, so erweitert, dass es die erforderlichen Daten bereitstellt. Dazu zählt neben den eigentlichen Daten zur Interaktion auch das zugrundeliegende Objektmodell, das durch AOI definiert ist. Ohne dieses Modell müssten die entstandenen Stimuli nachträglich ausgewertet und in ein solches Modell übertragen werden.

Der Vorteil bei der Erweiterung eines Programms gegenüber der nachträglichen Erstellung eines Modells besteht darin, dass das Programm die Elemente, die es visualisiert, genau kennt. Es ist also in der Lage, für jeden neuen Zustand der Visualisierung alle Objekte anzugeben, die es darstellt. Zusätzlich können die geometrischen Daten für jedes Objekt protokolliert werden. Unter der Voraussetzung, dass der Quellcode angepasst werden kann, ist das eine einfach zu implementierende Lösung. Sie hat außerdem den Vorteil, dass ein Proband nicht

mitbekommt, dass die Interaktion protokolliert wird. Das heißt, der Proband wird nicht durch die zusätzliche Erfassung der Daten beeinflusst.

Steht der Quellcode des Programms nicht zur Verfügung, ist es äußerst aufwändig, ein Objektmodell der Visualisierung zu erstellen. Sofern dieser Prozess nicht automatisiert werden kann, ist das Erstellen eines Modells vermutlich nicht praktikabel.

Wurde für jedes Objekt eine eindeutige ID vergeben, ist es im Grunde irrelevant, wenn das Objekt durch eine Interaktion verdeckt oder ausgeblendet wird – solange klar ist, ob ein und falls ja, welcher Teil des Objekts an welcher Stelle zu sehen ist. Sollte beispielsweise ein Tooltip ein anderes Objekt teilweise verdecken, ist ein neuer Zustand des Stimulus entstanden, bei dem das überdeckte Objekt andere geometrische Daten aufweist als zuvor. Das bedeutet, der tatsächlich sichtbare Teil eines Objekts ist der für eine Zuordnung zu Eye-Tracking-Daten relevante Teil und damit der für die Aufzeichnung erforderliche.

Für eine Visualisierung der Interaktionsdaten zusammen mit den Eye-Tracking-Daten ist die Qualität der aufgezeichneten Daten von besonderer Bedeutung. Die Abbildung der Augenbewegungen auf die AOI wird deutlich erleichtert, wenn bereits die Quelldaten ausschließlich aus den im Stimulus dargestellten Elementen mit den entsprechenden, geometrischen Eigenschaften bestehen. Ist für alle Objekte die Position auf der *Z-Achse* hinterlegt, lassen sich diese Eigenschaften rekonstruieren. Die Position auf der *Z-Achse* gibt an, welche Priorität ein Element bei der Darstellung hat. Ein Element mit hoher Priorität wird im Vordergrund angeordnet und überdeckt somit die anderen Elemente.

5.1.1 Daten und Datenherkunft

Nachdem festgestellt wurde, welche Daten für die gesuchte Visualisierung vorhanden sein müssen, wird in diesem Abschnitt beispielhaft beschrieben, wie ein existierendes Programm für die genannten Zwecke analysiert werden kann. Das Ziel ist es, herauszufinden, was bei der Datenerfassung berücksichtigt werden muss und auf welche Weise protokolliert wird. Es muss entschieden werden, welche Ereignisse aufgezeichnet werden sollen.

Dafür wurde in einem ersten Schritt ermittelt, wie das bestehende Programm *VarifocalReader* aufgebaut ist und wie es eingesetzt werden kann. Es sollte unter anderem geprüft werden, wie einzelne AOI definiert werden können. Nachfolgend wird der Aufbau des Programms näher beschrieben.

Beschreibung des Programms VarifocalReader

Die zu visualisierenden Daten für diese Diplomarbeit wurden während der Verwendung des Programms *VarifocalReader* (Koch, et al. 2014) von diesem erfasst. Mit *VarifocalReader* werden die Inhalte eines Buchs so dargestellt, dass sie schnell durchsucht werden können (siehe *Abbildung 15: Die Programmoberfläche von VarifocalReader nach Programmstart*). Ein Buch wird dabei in seine Kapitel (*Chapters*), Unterkapitel (*Subchapters*), Seiten (*Pages*) und Paragraphen (*Paragraphs*) unterteilt. Für jede dieser Kategorien wird eine eigene Spalte zur Darstellung verwendet (von links nach rechts in genannter Reihenfolge). Zusätzlich wird der eigentliche Text angezeigt und wichtige Stellen, wie Namen und Zitate, werden darin farblich hervorgehoben.

Jedes Element einer Kategorie wird durch einen zugehörigen Block repräsentiert – also zum Beispiel jedes Kapitel der Kategorie *Chapters*. Jeder Block enthält eine eindeutige Bezeichnung, wie den Namen des zugehörigen Kapitels oder die Seitenzahl der entsprechenden Seite. Die Größe der einzelnen Blöcke innerhalb einer Spalte repräsentiert dabei die Länge der entsprechenden Abschnitte im Buch. Man erkennt in *Abbildung 15: Die Programmoberfläche von VarifocalReader nach Programmstart* zum Beispiel auf den ersten Blick, dass die Einleitung (linke Spalte, ganz oben) offensichtlich sehr viel kürzer als die nachfolgenden Kapitel (linke Spalte) ist, weil der zugehörige Block deutlich schmaler ist.

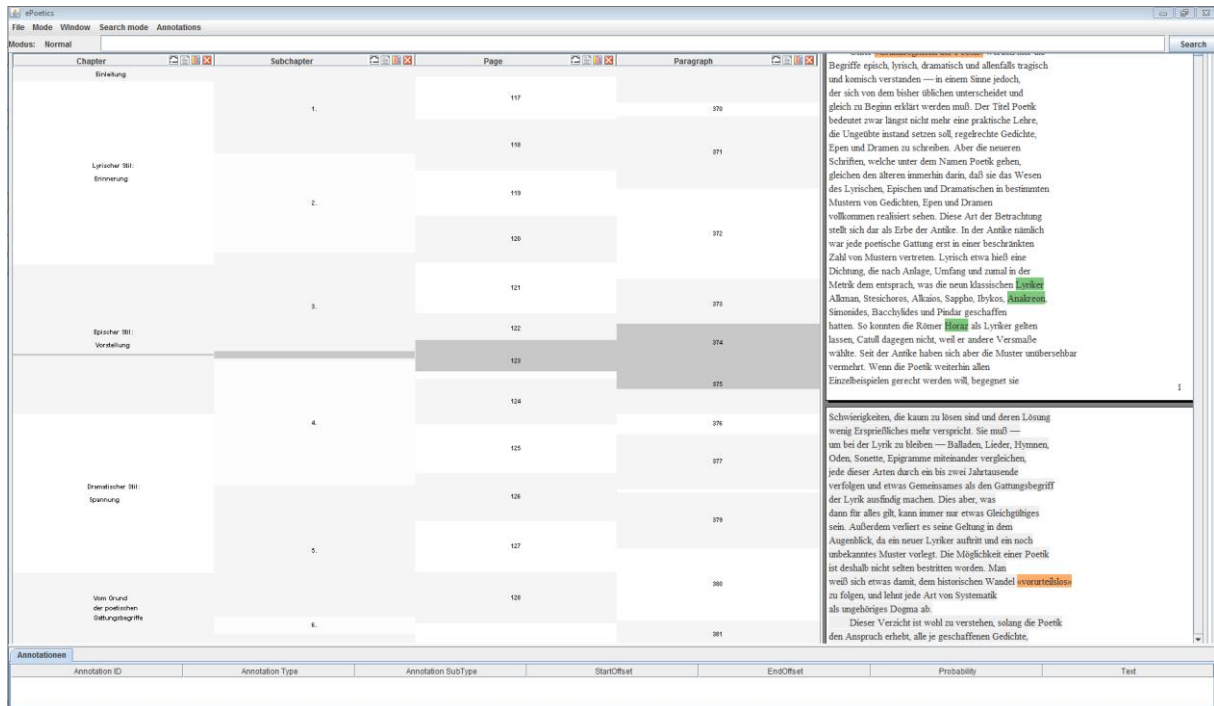


Abbildung 15: Die Programmoberfläche von VarifocalReader nach Programmstart

Die Darstellung eines Buchtextes mit dem Programm VarifocalReader unmittelbar nach Programmstart. Dabei enthält die linke Spalte die Repräsentation der Kapitel. Danach kommen in dieser Reihenfolge die Spalten für Unterkapitel, Seiten, Paragraphen und den eigentlichen Text (ganz rechts).

Außer dem Scrollen als wichtigste Interaktion lassen sich zusätzliche Informationen zu jedem Block in den einzelnen Spalten einblenden. In *Abbildung 16: Programmoberfläche von VarifocalReader nach der Interaktion* ist zu sehen, wie diese Informationen angezeigt werden. Insbesondere sind sogenannte *Word Clouds*, Balkendiagramme und Piktogramme zu erkennen. Alle drei Darstellungen geben auf unterschiedliche Weise Auskunft über das Vorkommen verschiedener Wörter im Text. Zum Beispiel über wichtige Begriffe in diesem Kapitel (*Word Clouds*), deren Häufigkeit (Balkendiagramme) oder an welcher Stelle sie zu finden sind (Piktogramme).

Word Clouds geben Aufschluss darüber, welche Wörter im jeweiligen Buchabschnitt signifikant häufiger als in anderen Teilen des Werkes vorkommen. Außerdem wird die Häufigkeit dieser Wörter durch deren Anordnung und deren jeweilige Schriftgröße dargestellt. Dabei haben die Wörter in den oberen Zeilen einer *Word Cloud* einen höheren Wert als die Wörter

in den darunter liegenden Zeilen. Gleiches gilt für Wörter innerhalb einer Zeile, die weiter links stehen.

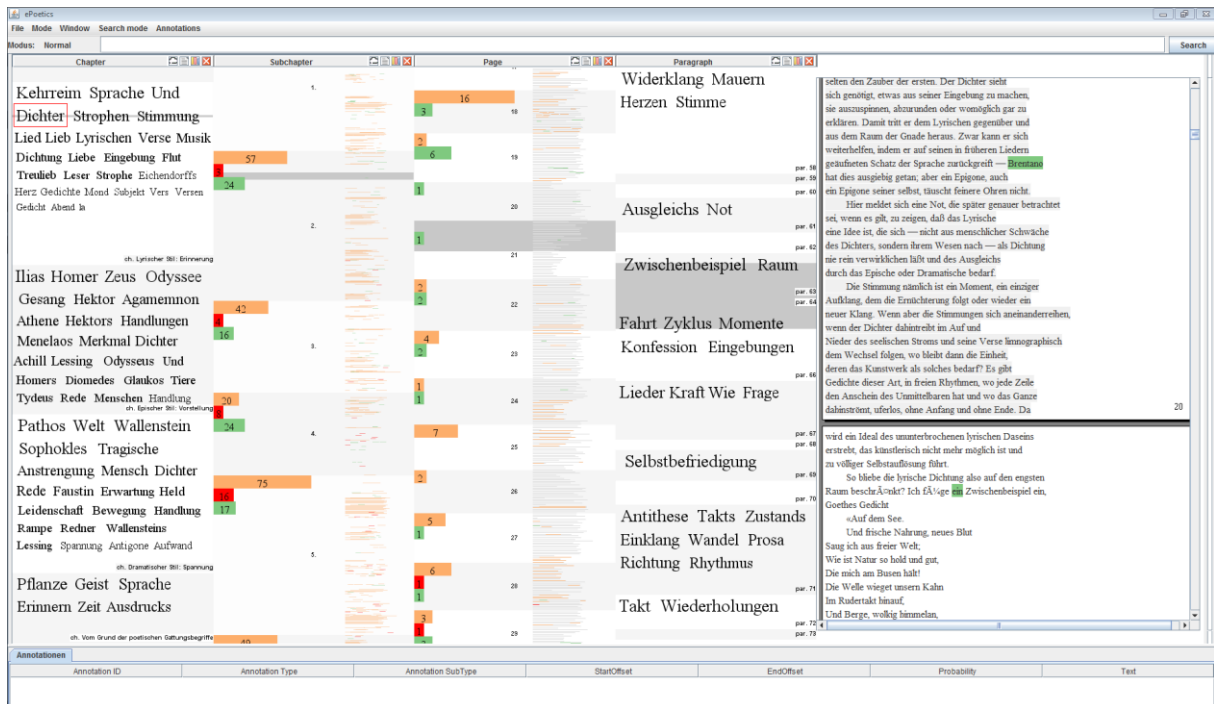


Abbildung 16: Programmoberfläche von VarifocalReader nach der Interaktion

Programmfenster von VarifocalReader. Es wurden verschiedene Informationen als Word Clouds, Balkendiagramme und Piktogramme in den einzelnen Spalten eingeblendet. Die linke Spalte zeigt für jedes Kapitel eine Word Cloud. Die mittleren Spalten für Unterkapitel und Seiten enthalten Balkendiagramme zur Anzeige der Häufigkeit von Zitaten (orange), Namen (grün) und des in einer Word Cloud markierten Wortes (rot) im jeweiligen Abschnitt. Zusätzlich enthalten beide Spalten Piktogramme. Die Spalte für die Paragraphen enthält ebenfalls Word Clouds.

Die einzelnen Balkendiagramme lassen erkennen, wie oft ein bestimmtes Wort (oder Name beziehungsweise Zitat) im jeweiligen Abschnitt auftritt. Worum es sich handelt ist dabei farblich kodiert. Unterschieden wird zwischen Namen (grün), dem selektierten Wort (rot) und Zitaten (orange). Piktogramme lassen darüber hinaus erkennen, an welcher konkreten Stelle im Text sich solche Begriffe befinden: Sie sind eine Miniaturansicht des Texts mit farblicher Hervorhebung der entsprechenden Stellen.

Analyse eines Programms am Beispiel VarifocalReader

Die Beschreibung des Programms VarifocalReader im letzten Abschnitt erfolgte, um nachvollziehen zu können, wie ein interaktiver Stimulus aufgebaut sein kann. Die auf einem solchen, konkreten Aufbau basierende Analyse eines Programms wird in diesem Unterkapitel beispielhaft für VarifocalReader beschrieben. Neben der Struktur der Visualisierung wurden auch die zur Verfügung stehenden Interaktionsmöglichkeiten betrachtet.

Es war naheliegend, jeden einzelnen Block einer Spalte als AOI zu definieren, weil jeder dieser Blöcke einen eigenen, konkreten Abschnitt des Buches repräsentiert. Dadurch kann später verglichen werden, welcher Pfad von den verschiedenen Probanden bei der Informationsgewinnung jeweils eingeschlagen wurde. Allerdings kann der Anwender den Inhalt der einzel-

nen Blöcke spaltenweise durch das Anzeigen zusätzlicher Informationen verändern. Folglich könnte auch jeder Balken eines Balkendiagramms, jedes Piktogramm und jedes Wort einer *Word Cloud* als AOI definiert werden. Im Rahmen dieser Arbeit wurde vorerst auf eine derartig detaillierte Unterscheidung verzichtet – auch deshalb, weil das Programm noch in der Entwicklung steckte und die für eine umfassende Protokollierung erforderliche Anpassung nicht allzu umfangreich werden durfte.

Die Interaktionsmöglichkeiten für die Visualisierung in *VarifocalReader* sind ebenfalls überschaubar. Neben dem Explorieren der Daten durch vertikales Scrollen erlaubt das Programm lediglich das Ein- beziehungsweise Ausblenden der in Abschnitt *Beschreibung des Programms VarifocalReader* in Kapitel 5.1.1 *Daten und Datenherkunft* beschriebenen Balkendiagramme, Piktogramme und *Word Clouds*. Dabei ist anzumerken, dass das Aktivieren der *Word Clouds* zur Folge hat, dass eventuell angezeigte Balkendiagramme und Piktogramme nicht mehr dargestellt werden. Zudem besteht nur noch die Möglichkeit, in einer *Word Cloud* angezeigte Wörter zu selektieren. Diese Selektion wirkt sich vor allem auf die Darstellung der Balkendiagramme und Piktogramme in den anderen Spalten aus. Beide stellen zusätzliche Informationen zum selektierten Wort bereit. Das Balkendiagramm jedes Blocks gibt Auskunft über die Häufigkeit des ausgewählten Worts in diesem Block. Das Piktogramm zeigt, an welchen Stellen das entsprechende Wort im zum Block zugehörigen Text auftritt.

Ferner existieren keine Dialoge oder Tooltips, die durch die Interaktion mit der Visualisierung hervorgerufen werden können. Das erleichtert die Abbildung der erfassten Eye-Tracking-Daten auf die einzelnen Blöcke insofern, als diese Blöcke zu keinem Zeitpunkt von anderen Elementen überdeckt werden. Einzige Ausnahme ist, dass die Blöcke durch Scrollen so verschoben werden können, dass die Blöcke am oberen beziehungsweise unteren Rand einer Spalte nicht mehr vollständig angezeigt werden. Demnach muss – wie bereits vorgeschlagen – bei der Aufzeichnung darauf geachtet werden, nicht einfach die anhand der Länge des zugehörigen Texts berechnete Größe eines Blocks zu protokollieren sondern die tatsächlich dargestellte. Ist das nicht der Fall, könnte die Fixation eines Probanden auf der Spaltenüberschrift fälschlicherweise auf die AOI des obersten Blocks dieser Spalte abgebildet werden.

Für *VarifocalReader* verändern also alle verfügbaren Interaktionsmöglichkeiten den Stimulus, auch wenn nicht jede Interaktion zur Verschiebung der Blöcke führt. Existieren für ein Programm Interaktionen, die kein Neuzeichnen der Visualisierung auslösen, muss individuell entschieden werden, ob und wie solche Ereignisse protokolliert werden.

Bereits während der Analyse von *VarifocalReader* ist ein Anwendungsfall aufgetreten, der besondere Aufmerksamkeit verdient: Eine wie in *Abbildung 16: Programmoberfläche von VarifocalReader nach der Interaktion* gezeigte Darstellung kann zum Beispiel verwendet werden, um mithilfe der eingeblendeten Balkendiagramme zu entscheiden, auf welcher Seite des Buches am meisten Namen (orangefarbene Balken) zu finden sind. Diese Information kann mit einer einzigen Scroll-Bewegung eingeholt werden. Dazu reicht es aus, sich die Ausdehnung des längsten Balkens beziehungsweise die Position, bis zu welcher der Balken reicht, zu merken. Wird während des Scroll-Vorgangs ein längerer Balken entdeckt, wird die Information umgehend ersetzt.

Kritisch ist hierbei, dass bereits während des Scrollens unzählige Stimuli erzeugt und dem Probanden angezeigt wurden, ohne, dass diese Informationen aufgezeichnet wurden. Es han-

delt sich einerseits um nützliche Daten und andererseits um schwer darzustellende Informationen. Es ist nahezu unmöglich, die im beschriebenen Fall entstandenen Stimuli festzuhalten und sinnvoll zu verarbeiten, zumal ohne eine Art *Think-Aloud-Protokoll* nicht eindeutig ist, ob einfach nur gescrollt wurde oder der Proband dabei Informationen gewonnen hat. Deshalb wurden von *VarifocalReader* für alle genannten Interaktionsmöglichkeiten nur die jeweils entstandenen Stimuli nach Abschluss der Interaktion aufgezeichnet.

Sind in einem Programm Tooltips für die Visualisierung oder Dialoge vorhanden, muss unbedingt deren tatsächliche Position und Größe dokumentiert werden. Vor allem Tooltips bestehen oft aus dynamisch erzeugten Inhalten und werden am zugehörigen Element angezeigt. Je nach Position des Elements variiert auch die Ausrichtung des Tooltips. Beispielsweise wird der Tooltip eines Elements am unteren Rand der Visualisierung in der Regel oberhalb des Elements geöffnet, während der gleiche Tooltip für das Element am oberen Rand der Visualisierung unterhalb des Elements angezeigt wird. Ändert sich der Inhalt eines Tooltips ist es wahrscheinlich, dass auch die Größe des Tooltips davon betroffen ist.

Eine ähnliche Schwierigkeit besteht für Dialoge, deren Position und Größe oft vom Anwender bestimmt und verändert werden kann. In beiden Fällen muss entschieden werden, wie damit umgegangen wird beziehungsweise, ob diese Faktoren bereits bei der Aufzeichnung oder erst bei der Auswertung berücksichtigt werden. Erfolgt die Protokollierung in besonders kurzen Zeitintervallen von wenigen Millisekunden, muss geprüft werden, ob die Berechnung der durch Dialoge und Tooltips beeinflussten AOI parallel zur Protokollierung durchgeführt werden kann.

Ein weiteres Problem in Bezug auf Tooltips ist, dass ein Tooltip auch unbeabsichtigt hervorgerufen werden kann. Es kann nicht allein anhand der Daten entschieden werden, ob der Anwender den Mauscursor nur zufällig über ein Element bewegt hat oder ob er das bewusst getan hat, um Informationen zu erhalten. Auskunft darüber können zum Beispiel *Think-Aloud-Protokolle* geben.

5.1.2 Anforderungen aus bisherigen Erkenntnissen

Zusammengefasst ergeben sich aus den Vor- und Nachteilen der in Kapitel 4 *Relevante Arbeiten* vorgestellten Techniken nachfolgende Anforderungen für die angestrebte Visualisierung.

Die Sortierung der Anforderungen stellt keine Priorisierung der Anforderungen dar, die Reihenfolge richtet sich aber nach den Erfordernissen des *Visual Information Seeking Mantra*.

- Eine Übersicht, die den Vergleich der Daten mehrerer Probanden ermöglicht, muss vorhanden sein.
- Es muss möglich sein, die Daten ausgewählter Probanden aus dieser Übersicht gegenüberzustellen.
- Die Daten eines einzelnen Probanden müssen individuell analysiert werden können.
- Die Daten müssen so dargestellt werden, dass ein Bezug zwischen den Blickpunkten und der Interaktion beziehungsweise der Bewegung des Mauszeigers hergestellt werden kann, falls ein solcher existiert.

- Es müssen alle Zustände und Übergänge zwischen diesen erkennbar sein. Das bedeutet, dass dargestellt werden muss, welche Interaktionen zu welchem Zeitpunkt welchen Zustand des Stimulus erzeugt haben.
- Alle relevanten Informationen zu den angezeigten Elementen müssen abgerufen werden können.

Ferner ist es erstrebenswert, diese Anforderungen unter Berücksichtigung des *Visual Information Seeking Mantra* umzusetzen. Erfreulich ist, dass sich manche der Anforderungen auf dieses Prinzip übertragen lassen, weil sie ebenfalls eine Übersicht, die Möglichkeit einer gezielten Analyse und die Abfrage konkreter Informationen einfordern.

5.2 Erfasste Daten

Um eine Visualisierung zu ermöglichen, die Eye-Tracking-Daten und Interaktionsdaten eines interaktiven Stimulus miteinander verknüpft, wurden nachfolgend beschriebene Daten berücksichtigt.

Zuerst wurden die vom Eyetracker aufgezeichneten Daten auf die wesentlichen Informationen reduziert. Für jeden Eintrag waren der Zeitpunkt, die zugehörige Fixation sowie deren Koordinaten und Dauer von Bedeutung.

Damit die Informationen zur Augenbewegung den Daten zur Interaktion verhältnismäßig gegenübergestellt werden konnten, wurden gleichartige Einträge zur Mausbewegung zusammengestellt. Diese Einträge bestehen ebenfalls aus einem Zeitstempel und den zugehörigen Koordinaten des Mauszeigers. Darüber hinaus enthalten sie eine Beschreibung der AOI, die der Position des Mauszeigers zu diesem Zeitpunkt zuzuordnen ist. Die Angaben zur AOI bestehen aus Position, Größe, Typ und ID. Obwohl im Grunde die ID ausreicht, um eine AOI eindeutig zu identifizieren, ist die Angabe des Typs durchaus sinnvoll. Diese zusätzliche Information bietet die Möglichkeit, die AOI entsprechend ihrer Darstellung im Programm zu klassifizieren, ohne dabei den genauen Aufbau des zugrundeliegenden Programms zu kennen. Betrachtet man die Daten zur Mausbewegung, lässt sich erkennen, dass daraus ebenfalls Fixationen – allerdings Fixationen mit der Maus – berechnet werden können.

Weil die Interaktionsdaten nicht nur aus der Mausbewegung eines Probanden bestehen, werden außerdem alle vom Probanden vorgenommenen Interaktionen zusammen mit dem daraus entstandenen Zustand des interaktiven Stimulus erfasst. Dieser Zustand beschreibt alle im Stimulus zu diesem Zeitpunkt sichtbaren AOI. Entsprechend sieht ein Eintrag für eine Interaktion vor, dass außer der Informationen, wie sie für einen Eintrag zur Mausbewegung festgehalten werden, der aktuell gültige Stimulus definiert wird. Konkret besteht ein solcher Interaktionseintrag also aus den Daten zur Mausbewegung für den Zeitpunkt der Interaktion, aus dem aufgetretenen Ereignis respektive der Interaktion und den Daten zum Stimulus bestehend aus Position, Größe, Typ und ID aller sichtbaren AOI.

Zur eindeutigen Definition des Zustands eines Stimulus reicht es aus, jeweils nur die Elemente anzugeben, die tatsächlich verändert werden können. Sind in einem interaktiven Stimulus statische Bereiche vorhanden, können diese durch die einmalige Angabe des ursprünglichen Zustands ermittelt werden und müssen nicht für jeden neuen Zustand mit protokolliert und ausgewertet werden.

Abgesehen von den aus vielen Einträgen bestehenden Protokolldateien ist es für die Analyse zweifellos hilfreich, die gesamte Interaktion als Video aufzuzeichnen und bereitzustellen. Das wird in der Regel bereits vom Eyetracker erledigt und muss von einer Visualisierung lediglich aufgegriffen werden. In einem solchen Video sind sowohl die einzelnen Fixationen der Augen als auch die Interaktionen auf dem interaktiven Stimulus zu sehen.

Darüber hinaus kann eine Videoaufzeichnung mit Ton als *Think-Aloud-Protokoll* verstanden werden, sofern der Proband seine Gedanken bei der Bearbeitung laut mitgeteilt hat. Das bedeutet, die Aufzeichnung enthält die Erkenntnisse (*Insights*), die der Proband während der Interaktion gewonnen hat. Die Aufzeichnung kann neben einem solchen Audio-Mitschnitt auch durch textuelles Protokollieren erfolgen.

Solche Protokolle können entweder während der Bearbeitung einer Aufgabe durch einen Probanden aufgezeichnet oder im Nachhinein erfasst werden, in dem der Proband eine Videoaufzeichnung der eigenen Bearbeitung gezeigt bekommt und entsprechend kommentiert. Letztere Methode hat den Vorteil, dass sie die Bearbeitung nicht beeinflussen kann. Allerdings ist es möglich, dass der Proband nicht in der Lage ist, den Gedankengang im Nachhinein exakt wiederzugeben, wodurch Informationen verfälscht werden oder verloren gehen könnten. In jedem Fall gewährt ein *Think-Aloud-Protokoll* weitere Einblicke in den Prozess der Informationsgewinnung. Deshalb ist es fester Bestandteil der vollständigen Interaktionserfassung.

Die einzelnen Erkenntnisse, die ein Proband während der Interaktion erlangt hat, lassen sich allerdings nur schwer durch ein automatisiertes Verfahren festhalten oder nachträglich aus einer Aufzeichnung extrahieren. Aus diesem Grund müssen die erfassten Daten manuell angereichert werden. Die präzise Zuordnung zu einem konkreten Interaktionsereignis, einer Fixation oder einem Zeitstempel ist eine schwierige Herausforderung.

Eine Möglichkeit ist, das Erfassen von Einträgen in einem *Think-Aloud-Protokoll* mit dem Konzept zu unterstützen. Das heißt, die übrigen Daten werden visualisiert und es ist durch das Einfügen und Positionieren von Elementen möglich, Ereignisse der Informationsgewinnung aus dem *Think-Aloud-Protokoll* dem korrekten Zeitabschnitt zuzuordnen. Diese Elemente müssen durch Annotation mit dem vorhandenen Eintrag verknüpft werden können.

5.3 Visuelle Analyse

Auf Grund der geforderten Vergleichbarkeit der Probanden ist die Verwendung von Scanpaths zur Darstellung der Daten tendenziell ungeeignet. Das gilt insbesondere, weil für jeden Probanden zwei Scanpaths verwendet werden müssten: einer für die Augen- und einer für die Mausbewegung. Das dadurch entstehende *Visual Clutter* wäre vom Anwender nur schwer zu entschlüsseln. Auch die auf Heatmaps basierenden Techniken können dieser Anforderung nicht gerecht werden, weil sie weder den zeitlichen Verlauf berücksichtigen noch zwischen den Daten einzelner Probanden unterscheiden. Generell erscheint es wenig sinnvoll, eine Visualisierung zu verwenden, die den Stimulus selbst mit einschließt, weil dieser durch jeden Probanden unterschiedlich verändert wird.

Durch die in Kapitel 5.1 *Anforderungen* genannten Aspekte ist die Verwendung von AOI ohnehin vorgegeben. Eine geeignete Visualisierung der AOI für mehrere Probanden wurde mit *Scarf Plots* bereits in Abschnitt *Weitere Visualisierungen* in Kapitel 2.2.2 *Visualisierung von*

Eye-Tracking-Daten vorgestellt. Weil die AOI bei der Datenerfassung typisiert wurden, kann die in *Scarf Plots* verwendete, farbliche Codierung der AOI gezielt eingesetzt werden.

Die Schwierigkeit bei der Verwendung von *Scarf Plots* besteht darin, die AOI-Verläufe beider Datentypen (Augen- und Mausbewegung) darzustellen. Der naive Ansatz, einfach zwei Zeilen für jeden Teilnehmer zu verwenden, sorgt entweder für die Entkopplung der Daten voneinander – wenn erst alle Zeilen für die Augen- und danach alle Zeilen für die Mausbewegungen aufgeführt werden – oder für die Unterbrechung der Datentypen durch den jeweils anderen – wenn für jeden Probanden die Zeilen für beide Datentypen untereinander angeordnet werden. Dazu kommt, dass die eigentliche Interaktion und der durch die einzelnen Interaktionsschritte veränderte Stimulus in einer solchen Darstellung noch nicht berücksichtigt werden. Folglich können vergleichbare Visualisierungstechniken für AOI-Verläufe, unter Beachtung der genannten Punkte, nur bedingt eingesetzt werden.

Das führt zu der Überlegung, dass neben der Einteilung der Darstellung in Abschnitte für jeden Probanden eine weitere Einteilung dieser Abschnitte in Unterabschnitte für die AOI beziehungsweise Typen von AOI sinnvoll ist. Es soll also statt eines Stimulus das zugrundeliegende Objektmodell visualisiert werden. Dieses Modell kann so angezeigt werden, dass es für alle Probanden gleich ist und dadurch den Vergleich der darin eingezeichneten Elemente ermöglicht. Jeder so entstandene Unterabschnitt muss abermals unterteilt werden: es bedarf jeweils eines Abschnitts für jeden Datentyp oder einer unterschiedlichen Kodierung der Eigenschaften jedes Datentyps.

Durch eine parallele Anordnung aller Unterabschnitte können die Veränderungen des Stimulus durch Interaktion mit einem zu diesen Abschnitten orthogonalen Element unterbrochen werden – zum Beispiel einem Querbalken. Schließlich sind in der Regel alle Abschnitte von der Änderung betroffen. Die Eigenschaften des Elements, das die Veränderung darstellt, können dabei ebenfalls die Art der Änderung verschlüsseln. Eine Unterscheidung der Interaktionen, die diese Änderungen verursachten, erfolgt anhand der Absicht des Probanden. Eine mögliche Klassifizierung dieser Absichten nach Yi et al. wurde in Abschnitt *Interaktionskategorien* in Kapitel 2.4.1 *Interaktion* vorgestellt. Wird beispielsweise der gleiche Schalter zum Ein- beziehungsweise Ausblenden von Informationen in einer Visualisierung zwei Mal hintereinander gedrückt, unterscheidet sich in der Regel die Absicht dahinter. Beim ersten Klick ist die Absicht das Einblenden von Informationen und beim zweiten Klick das Ausblenden, obwohl technisch gesehen dieselbe Interaktion vorliegt.

Fixationen durch Auge und Maus sowie vorhandene Einträge eines *Think-Aloud-Protokolls* sind jeweils einer AOI und einem Zeitstempel zugeordnet. Entsprechend können sie gleich behandelt und eingezeichnet werden. Für das *Think-Aloud-Protokoll* wird keine weitere Unterteilung der AOI benötigt, weil Einträge hierfür nur spärlich über die Bearbeitungsdauer verteilt anfallen und nicht in regelmäßigen Abständen auftreten.

Ein erster Entwurf besteht also aus einer tabellarischen Darstellung, die sich einzelner Spalten für jeden Probanden bedient. Diese Spalten sind weiter in die verschiedenen Typen von AOI unterteilt. Jeder Typ von AOI enthält dabei jeweils einen Bereich für Daten zur Augen- und einen für Daten zur Mausbewegung. Horizontale Linien in jeder Spalte stehen für eine Änderung am Stimulus durch Interaktion. Mit einem solchen Aufbau entspricht jede Spalte einer

eigenen Zeitleiste für den jeweiligen Probanden. Um welche Interaktion es sich handelt, wird durch die farbliche Kodierung der horizontalen Linien ausgedrückt.

Bei der Farbgebung ist es wichtig, darauf zu achten, dass die Farben gut voneinander unterschieden werden können. Außerdem müssen die Farben für die Interaktionen heller sein, als die für Fixationen. Das ist deshalb sinnvoll, weil die Elemente für die Interaktionen von den Elementen einzelner Fixationen überdeckt werden sollen. Für Interaktionen sollten, falls vorhanden, Farben verwendet werden, die mit diesen assoziiert werden können: zum Beispiel Grün für das Hinzufügen von Elementen oder Rot für das Entfernen. Für besondere Interaktionen, wie das Selektieren einzelner Elemente oder das Wechseln der Ansicht, sind Farben geeignet, die sich von den übrigen abheben.

Abbildung 17: Entwurf für die Darstellung der Daten zeigt exemplarisch, wie die Daten für drei Probanden tabellenartig, mit einer Spalte für jeden Probanden, angeordnet werden können. Jede Spalte repräsentiert eine eigene Zeitleiste und enthält dabei eine weitere Unterteilung in die verschiedenen AOI-Typen. Für jede Type existiert ein vertikaler Strang für die Fixationen mit dem Auge (links in einer solchen Spalte) und für die Fixierungen einer AOI mit der Maus (rechts in einer solchen Spalte). Um welchen Datentyp es sich handelt, ist an den Symbolen über den Einträgen zu erkennen.























Proband 1			Proband 2			Proband 3		
AOI Typ1	AOI Typ2	AOI Typ3	AOI Typ1	AOI Typ2	AOI Typ3	AOI Typ1	AOI Typ2	AOI Typ3
 	 	 	 	 	 	 	 	 
x x 				x x				x x
	x x			x x		x x		
	x -	- x		x x 			x x	
		x x 	x x					x x 

Abbildung 17: Entwurf für die Darstellung der Daten

Drei Spalten mit den vorhandenen Typen von AOI (Typ1, Typ2 und Typ3). Die Spalten sind durch die Augen- und Maussymbole für jeden Typ von AOI weiter unterteilt. Jeder Eintrag entspricht einer Fixation der jeweiligen AOI mit den Augen (x) beziehungsweise mit der Maus (x). Interaktionen, die zur Veränderung des jeweiligen Stimulus geführt haben, sind durch horizontale Balken vermerkt. Die Warnsymbole deuten auf Erkenntnisse bei der Informationsgewinnung hin.

Die verschiedenen Datentypen, also Fixationen mit dem Auge beziehungsweise mit der Maus, werden abwechselnd dargestellt. Folglich ist es umständlich, den Verlauf der Blickpunkte

verschiedener Probanden miteinander zu vergleichen. Das Gleiche gilt für den Verlauf der Mausbewegung. Um das zu vermeiden, kann die Visualisierung mit Interaktionsmöglichkeiten wie dem Aus- beziehungsweise Einblenden der jeweiligen Datentypen angereichert werden. Denkbar ist auch das Kennzeichnen unterschiedlicher Datentypen durch die Form oder Farbe des repräsentierenden Elements.

Um den Vergleich der Spalten beziehungsweise der Probanden zu erleichtern, kann es je nach Aufgabenstellung hilfreich sein, die Länge der Spalte und damit der Zeitleiste unterschiedlich zu nutzen. Es kann entweder für jeden Probanden die eigene Interaktionszeitspanne verwendet und alle Elemente entsprechend dieser Dauer auf der Zeitachse respektive Spalte angeordnet werden. Dieses Vorgehen ist im Grunde eine Normalisierung der Werte. Oder es wird ermittelt, welcher Proband die längste Zeitspanne aufweist und diese Dauer wird verwendet, um die Elemente aller Probanden anzuordnen. Letztere Methode erlaubt den direkten zeitlichen Vergleich, weil alle Elemente auf der gleichen Horizontale demselben Zeitstempel entsprechen. Der Nachteil ist, dass bei großen Differenzen zwischen den Bearbeitungszeiten der einzelnen Probanden der zur Verfügung stehende Platz nicht ausgenutzt wird.

Prüft man diese Form der Darstellung auf ihre Eignung als Übersicht, stellt man fest, dass die Breite der Spalten zwar variiert werden kann, um eine größere Anzahl an Probanden darzustellen, diese Skalierung aber schnell an ihre Grenzen stößt. Eine ähnliche Beobachtung trifft zu, wenn man daran denkt, dass es kein Limit für die Interaktionszeitspanne gibt. Zwangsläufig muss die Größe der dargestellten Fixationen in einer Übersicht von der Anzahl der Probanden und der Dauer der Interaktionen abhängen. Eine Alternative ist, Scroll-Mechanismen anzubieten, um zu vermeiden, dass die Elemente zu klein oder gar nicht mehr angezeigt werden. Allerdings lässt sich dann nur noch bedingt von einer Übersicht sprechen.

Statt der Scroll-Mechanismen kann auch eine weitere Reihe von Spalten für die Probanden unterhalb der bisher angedachten Zeile hinzugefügt werden. Dadurch können zwar doppelt so viele Probanden dargestellt werden, allerdings wird es deutlich schwieriger, die Daten übereinanderliegender Darstellungen zu vergleichen. Außerdem entsteht dadurch ein Bruch in der Darstellung, der zusätzliche Orientierung erfordert, weil jetzt eine zusätzliche „Kopfzeile“ mit den Bezeichnungen der Probanden vorhanden ist.

Eine radiale Anordnung der Spalten hat den Vorteil, dass die Darstellung nicht zweigeteilt ist und eine unterschiedliche Anzahl Probanden immer auf der gleichen Kreisfläche angezeigt werden kann (siehe *Abbildung 18: Radialer Entwurf für die Darstellung der Daten*). Für jeden Proband wird ein Kreissegment reserviert, dessen Größe von der Anzahl der gezeigten Probanden abhängt. Die einzelnen Segmente für die AOI innerhalb dieses Kreissegments hängen folglich nicht nur von der Anzahl der AOI-Typen, sondern auch von der Anzahl der dargestellten Probanden ab. Demnach stößt diese Darstellung für eine große Anzahl Probanden oder Typen von AOI ebenfalls an ihre Grenzen. Je kleiner der Winkel eines Kreissegments wird, desto schwieriger wird es, die darin enthaltenen Elemente zu erkennen und miteinander zu vergleichen.

Der Datentyp eines Elements für eine Fixation kann auch in einer radialen Darstellung durch die Farbe, die Form oder die Position des zugehörigen Elements ausgedrückt werden. Die Position darf dabei nur so verändert werden, dass der Abstand zum Kreismittelpunkt unverändert bleibt und damit die zeitliche Zuordnungsmöglichkeit gewahrt ist. Der Bereich des Gra-

phen unmittelbar um den Kreismittelpunkt wirkt bereits für eine geringe Anzahl von Elementen *überladen*, vor allem, weil sich hier die Trennstriche vereinen. Ein Mindestabstand aller Elemente zum Kreismittelpunkt würde Abhilfe schaffen.

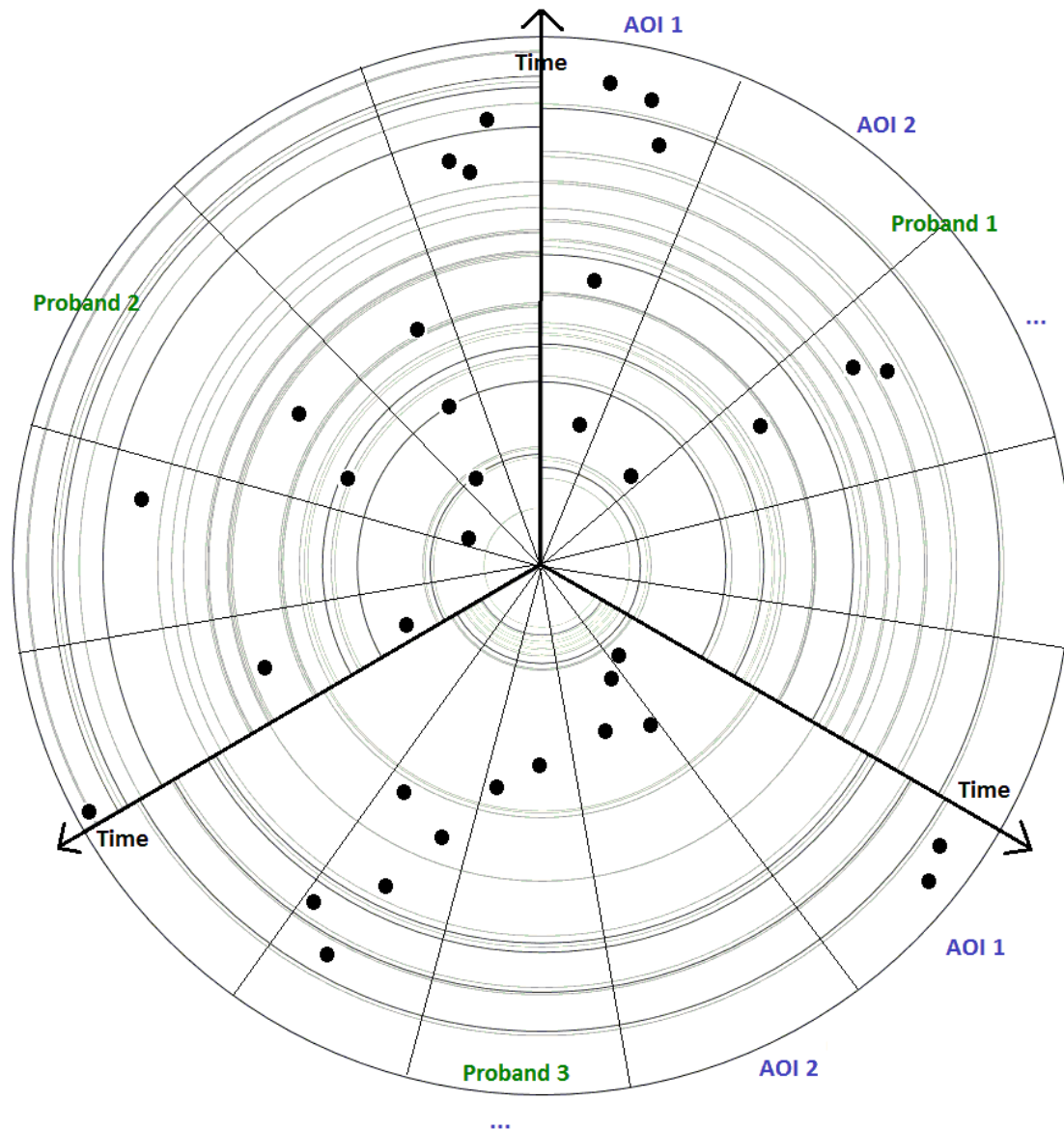


Abbildung 18: Radialer Entwurf für die Darstellung der Daten

Radialer Entwurf als alternative Darstellung der Daten. Die dicken Trennstriche, die vom Kreismittelpunkt ausgehen, kennzeichnen die Kreissegmente der einzelnen Probanden und dienen gleichzeitig als Zeitachse. Die dünnen Trennstriche dazwischen begrenzen den Anzeigebereich für die einzelnen AOI. Die hellen Kreisbögen innerhalb des Kreises repräsentieren die verschiedenen Interaktionen und damit die Veränderung des Stimulus. Die unterschiedlichen Graustufen kodieren die Art der Interaktion. Fixationen durch Auge und Maus werden als dunkle Kreisflächen dargestellt.

Ergänzend zu einem in *Abbildung 18: Radialer Entwurf für die Darstellung der Daten* gezeigten Graphen ist eine Übersicht über alle Teilnehmer einer Studie in Form einer Liste hilfreich, falls die Anzeige der Daten einzelner Probanden im Graphen durch Filtern ausgeblendet wird.

Außerdem können in einer solchen Liste Metadaten zu den Probanden angezeigt werden, ohne durch die Anzeige dieser Informationen *Visual Clutter* im Graphen zu verursachen.

Damit die im Graphen eingezeichneten Interaktionsschritte bei der Analyse besser erfasst werden können, eignet sich eine zusätzliche Liste mit Einträgen zu den Interaktionen. Eine solche Liste könnte neben dem Typ der Interaktion auch den Zeitstempel und den Probanden zur Interaktion enthalten. Wird der angezeigte Bereich durch Zoomen verändert, lassen sich die neu angezeigten Informationen mithilfe der zusätzlichen Informationen in der Liste schneller einordnen.

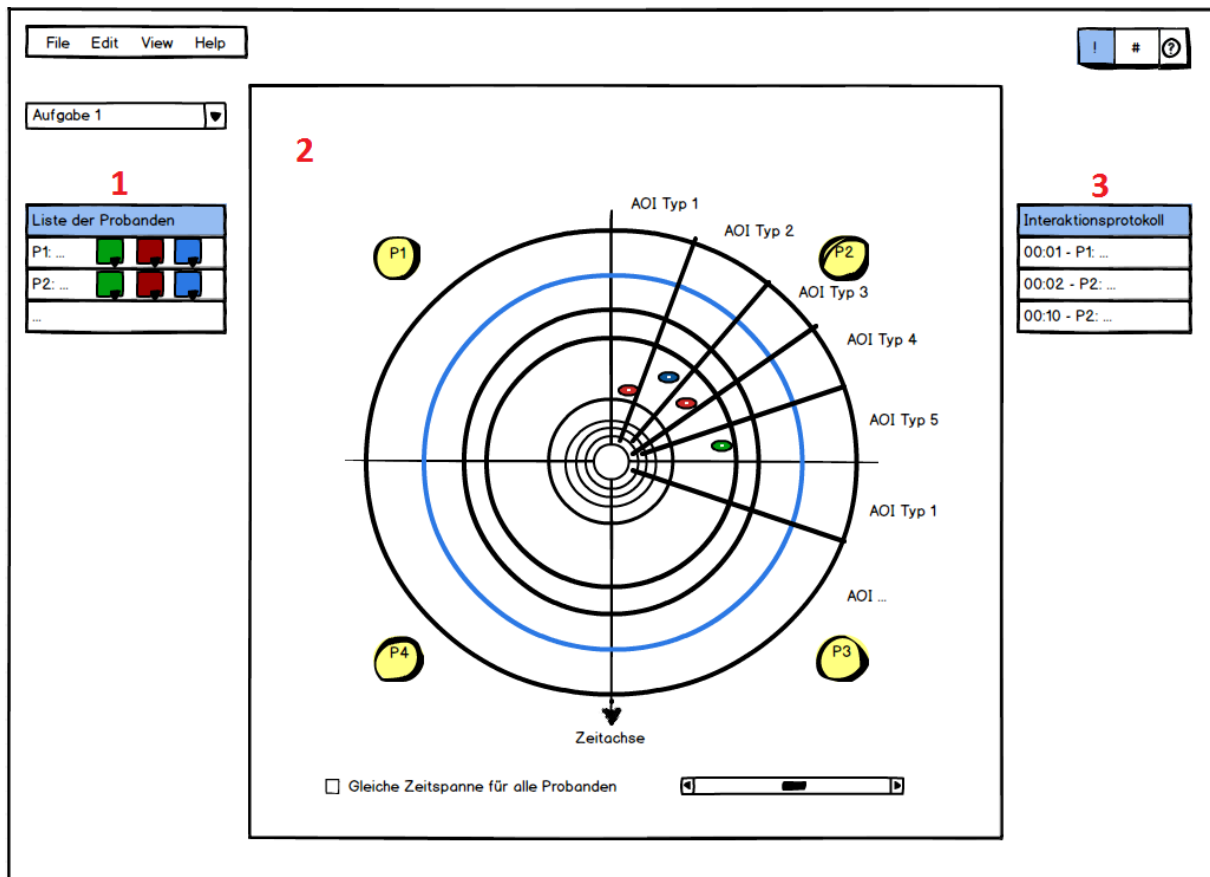


Abbildung 19: Entwurf eines Programmfensters für die Präsentation der Daten

Das Mockup eines Programmfensters, das neben den sinnbildlichen Menü- und Symbolleisten aus einer Liste der Probanden (1), der Präsentation der zu den Probanden gehörenden Daten (2) und einem Interaktionsprotokoll (3) besteht. Unter dem angedeuteten Graphen (2) und in der Liste der Probanden (1) sind außerdem verschiedene Bedienelemente demonstrativ untergebracht.

Führt man die genannten Intentionen in einem gedachten Programmfenster zusammen, könnte dieses wie in *Abbildung 19: Entwurf eines Programmfensters für die Präsentation der Daten* aussehen. Das beschriebene Konzept soll insbesondere dazu führen, erfasste Daten mit einem explorativen Verfahren analysieren zu können. Das heißt, Muster in der Repräsentation der Daten sollen das Aufstellen von Vermutungen über die Zusammenhänge zwischen den Daten fördern. Daraus entstehende Hypothesen können dann durch geeignete Studien und Experimente bestätigt werden. Der Fokus der Visualisierung liegt also nicht darauf, Belege für bestehende Hypothesen in den visualisierten Daten zu finden.

Um das gewünschte Vorgehen zu erleichtern, ist es sinnvoll, den Vergleich vorhandener Daten auf verschiedene Arten zu unterstützen. Für das Auffinden von Mustern ist es hilfreich, sowohl die Daten einzelner Probanden als auch die Daten verschiedener Aufgaben aus einer oder mehreren Studien gegenüberstellen zu können. Sofern eine Visualisierung ausschließlich die Daten einer einzigen Aufgabe umfasst, können verschiedene Aufgaben respektive Visualisierungen auf einzelne Programmfenster verteilt werden. Das ermöglicht den direkten Vergleich der Präsentationen verschiedener Aufgaben miteinander.

Im nachfolgenden Text wird darauf eingegangen, wie die in Kapitel 5.1 *Anforderungen* genannten Bedingungen mit dieser Visualisierung erfüllt werden. Die Beschreibung orientiert sich am *Visual Information Seeking Mantra*.

5.3.1 Visual Information Seeking Mantra

Es ist wichtig, die aus einer Übersicht (*Overview*) gewonnenen Informationen und Annahmen anhand detaillierterer Datenpräsentationen überprüfen und verfeinern zu können. Um zu solchen, weniger abstrakten Ansichten zu gelangen, müssen verschiedene Interaktionsmöglichkeiten bereitgestellt werden. Neben dem Ausblenden irrelevanter Datenelemente (*Filtern*) sollen wichtige Elemente detaillierter dargestellt werden können (*Zoomen*) und wichtige Informationen zu diesen abgefragt werden können (*Details on Demand*). Wie diese Interaktionen mit dem Konzept in Einklang gebracht werden, ist in den nachfolgenden Abschnitten genauer beschrieben.

Um den beschriebenen Wechsel zwischen verschiedenen Abstraktionsgraden zu steuern, sind unterschiedliche, konkrete Interaktionsmöglichkeiten notwendig. Es sind Bedienelemente erforderlich, um das Zoomlevel für die Anzeige der Daten, den darzustellenden Zeitabschnitt und die Verwendung der Zeitachse – als absolute beziehungsweise relative Zeit – auszuwählen zu können. Außerdem muss festgelegt werden können, welche Datentypen für welche Probanden visualisiert werden sollen.

Für das Einblenden von Details sind weitere Anzeigeflächen in Form von Tooltips (für konkrete Werte) oder neuen Programmfenstern (für das Abspielen von Aufzeichnungen) erforderlich. Auf diese Weise werden die neuen Informationen im richtigen Kontext übermittelt, weil die ursprüngliche Darstellung der Daten weiterhin vorhanden ist.

Overview first

In *Abbildung 19: Entwurf eines Programmfensters für die Präsentation der Daten* ist zu erkennen, dass die Anzeige der Daten aller Probanden (aus der gezeigten *Liste der Probanden*) im Graphen eine Übersicht über vorhandene Daten sein kann. Es sind sowohl die Eye-Tracking-Daten als auch die Interaktionsdaten und die Daten zur Mausbewegung sämtlicher Probanden dargestellt. Es können bereits in dieser Übersicht Analogien, Ausreißer oder andere Muster mittels der Anordnung der Datenelemente festgestellt werden. Welche Daten für die einzelnen Probanden zur Verfügung stehen, wird in der Liste der Probanden durch Kennzeichen entsprechender Elemente für jeden Datentyp angezeigt und ergänzt damit die Übersicht. Die genaue zeitliche Abfolge aller Interaktionen für die dargestellten Probanden ist im Interaktionsprotokoll zu sehen. Die Zuordnung der angezeigten Datenelemente zu den Probanden und den verschiedenen Typen von AOI ist durch die Verschlagwortung der entsprechenden Kreissegmente gegeben. Einzelne Bedienelemente, wie zum Beispiel die für die Einstellung des Abstraktionsgrads oder den Wechsel zwischen absoluter und relativer Darstel-

lung der Zeit, geben Aufschluss darüber, wie die einzelnen Elemente angeordnet wurden und zu interpretieren sind.

Es ist denkbar, dass die Darstellung vieler Typen von AOI für eine große Anzahl Probanden einen höheren Abstraktionsgrad bei der Visualisierung erfordert, um diese Daten übersichtlich aufzubereiten. Während sich die AOI durch neue, übergreifende Typisierung abstrahieren lassen, können die Daten einzelner Probanden nicht direkt zusammengefasst werden. Die maximale Zahl der Probanden, für die im Rahmen einer Studie Daten zu einer Aufgabe erfasst werden, wurde bei der Analyse der Aufgabenstellung mit einem Wert von 20-30 festgehalten. Ob sich das Konzept auch für diese Anzahl Probanden uneingeschränkt als Übersicht eignet, kann bei der Demonstration meines Prototyps festgestellt werden.

Zoom & Filter

Durch Filtern der Daten müssen uninteressante Bereiche von der Darstellung ausgeschlossen werden können. Eine erste Entscheidung, welche Daten als relevant beziehungsweise irrelevant eingestuft werden, wird in der Regel bereits beim Betrachten einer Übersicht über vorhandene Daten getroffen. Weil diese Entscheidung bei der Analyse des Graphen getroffen wird, ist es sinnvoll, wenn die Möglichkeit, bestimmte Bereiche nicht mehr darzustellen, direkt im Graphen angeboten wird. Deshalb existiert im Graphen für jeden Probanden, dessen Daten im Graphen dargestellt sind, ein Bedienfeld, über dessen Bedienelemente die gewünschte Anpassung angestoßen werden kann. Sind nur die Daten eines einzigen Probanden von besonderem Interesse, kann über das entsprechende Bedienfeld auch veranlasst werden, dass nur noch seine Daten im Graphen angezeigt werden.

Gleichzeitig muss die Auswahl, für welche der Probanden Daten visualisiert werden sollen, auch in der *Liste der Probanden* getroffen werden können, weil die Liste ebenfalls die Probanden visuell darstellt. Die Liste enthält einen Eintrag für jeden Probanden. Die Auswahl kann also durch Selektion beziehungsweise Deselektion der Einträge vorgenommen werden. Ist die Auswahl beendet, kann die Visualisierung der Daten zu den selektierten Einträgen neu angestoßen werden.

Außer durch die Möglichkeit, alle Daten eines oder mehrerer Probanden ein- beziehungsweise auszublenden, kann eine auf interessante Bereiche beschränkte Darstellung auch durch Zoomen erreicht werden. Mit einem entsprechenden Bedienelement lässt sich zum Beispiel der genaue Wert für die Vergrößerung der Darstellung gleichzeitig festlegen und ablesen. Das Resultat ist, dass nur noch ein entsprechender Ausschnitt des ursprünglich sichtbaren Bearbeitungszeitraums der Probanden angezeigt wird. Weil dieser Ausschnitt am Anfang, am Ende des Zeitraums oder dazwischen liegen kann, ist ein weiteres Bedienelement erforderlich, mit dem sich dieser Bereich auswählen lässt.

Es gibt Szenarien, in denen nicht alle Datentypen (namentlich Eye-Tracking-Daten, Interaktionsdaten und Daten zur Mausbewegung) von Interesse sind. Aus diesem Grund kann für jeden Eintrag in der Liste der Probanden individuell festgelegt werden, welche Datentypen angezeigt werden sollen. Prinzipiell werden für alle Einträge alle vorhandenen Daten dargestellt, sofern nichts anderes festgelegt wurde. Darüber hinaus ist es möglich, die Anzeige von Daten der einzelnen Datentypen für alle Probanden in der Liste zu unterbinden.

Details on Demand

Ist der angezeigte Bereich für die Analyse relevant, gewinnen zusätzliche Informationen an Bedeutung. Ungeachtet dessen müssen diese stets zur Verfügung stehen, weil sie bereits bei einem hohen Abstraktionsgrad der dargestellten Daten von Belang sein können. Wichtig ist, dass jedes dargestellte Element Träger von relevanten Informationen sein kann. Aus diesem Grund ist auch für jedes dieser Elemente eine Fläche zur Anzeige dieser Informationen vorhanden.

Für eine Fixation sind beispielsweise der genaue Zeitpunkt, die Dauer und die betroffene AOI von Bedeutung. Für eine Veränderung des Stimulus ist unter anderem interessant, welche Art von Interaktion zu dieser geführt hat, zu welchem Zeitpunkt diese stattgefunden hat oder welche AOI damit in Verbindung stehen beziehungsweise davon betroffen sind. Je nach Art der Interaktion können weitere Details von Bedeutung sein – wie zum Beispiel die Dauer einer Scroll-Bewegung. Zu den Daten eines Probanden sind vor allem Meta-Informationen relevant, wie beispielsweise die Bearbeitungsdauer oder Daten zum dargestellten Ausschnitt der Bearbeitung (Beginn, Ende, Zeitspanne).

Jede im Graphen angezeigte Veränderung eines Stimulus ist gleichzeitig im Interaktionsprotokoll vermerkt. Hier lassen sich auch Informationen wie Zeitstempel und Art der Interaktion ablesen. Obwohl diese Informationen permanent zur Verfügung stehen, ermöglicht das Selektieren einer solchen Interaktion – durch das damit verbundene Hervorheben dieser im Graphen und im Protokoll – das rasche Auffinden der Interaktion in beiden Ansichten. Durch dieses *Brushing & Linking* werden die Informationen im entscheidenden Moment (bei der Selektion) farblich hervorgehoben und können dadurch dem dargestellten Element im Graphen beziehungsweise im Interaktionsprotokoll zugeordnet werden.

Weitere wichtige Details stecken im durch Interaktion erreichten Zustand des Stimulus. Weil der Stimulus im Laufe der Bearbeitung durch einen Probanden verändert wurde, ist es hilfreich zu wissen, wie der Stimulus zum Zeitpunkt einer Fixation oder Interaktion tatsächlich ausgesehen hat. Der Klick auf ein beliebiges, im Graphen dargestelltes Element sollte also eine Anzeige des für den entsprechenden Zeitstempel gültigen Stimulus zur Folge haben. Darüber hinaus ist es sinnvoll, die Positionen von Blickpunkt und Mauszeiger zu diesem Zeitpunkt ebenfalls einzuzeichnen, falls diese Fixationen nicht ohnehin schon eingezeichnet sind – zum Beispiel durch den Eyetracker bei der Aufzeichnung. Wie bereits angedeutet, ist es sinnvoll, den Stimulus in einem neuen Programmfenster zu öffnen, damit ein Bezug zu den vorhandenen Elementen hergestellt werden kann.

Eine Interaktion kann oft nur bedingt visualisiert werden. Das gilt auch für tatsächliche Einblicke in die Informationsgewinnung. Ist anhand der Daten ersichtlich, dass eine andauernde Interaktion vorliegt oder das Ziel der Aufgabe vom Probanden erreicht wurde, kann das zugehörige *Think-Aloud-Protokoll* Auskunft darüber geben. Wird die zum jeweiligen Probanden gehörende Aufzeichnung aus diesem Grund abgespielt, kann das folglich als das Anfordern von Details verstanden werden.

5.4 Erfüllung der Anforderungen

Als Zusammenfassung des vorgestellten Konzepts und zum Abgleich der Merkmale des Konzepts mit den Anforderungen an dasselbe werden die in Kapitel 5.1 *Anforderungen* genannten

Punkte einer Beschreibung der Umsetzung dieser Anforderungen tabellarisch gegenübergestellt.

Anforderungen an die Visualisierung	Umsetzung im gezeigten Mockup
<ul style="list-style-type: none"> • Eine Übersicht, die den Vergleich der Daten mehrerer Probanden ermöglicht, muss vorhanden sein. 	Werden alle Probanden im Graphen angezeigt, ist eine solche Übersicht vorhanden. Diese wird durch die Liste der Probanden und das Interaktionsprotokoll ergänzt.
<ul style="list-style-type: none"> • Es muss möglich sein, die Daten ausgewählter Probanden aus dieser Übersicht gegenüberzustellen. 	Durch das Ein- beziehungsweise Ausblenden der Daten einzelner Probanden ist eine solche Gegenüberstellung möglich.
<ul style="list-style-type: none"> • Die Daten eines einzelnen Probanden müssen individuell analysiert werden können. 	Es ist möglich, nur die Daten eines Probanden anzuzeigen. Diese lassen sich dann mithilfe vorhandener Interaktionsmöglichkeiten gezielt analysieren.
<ul style="list-style-type: none"> • Die Daten müssen so dargestellt werden, dass ein Bezug zwischen den Blickpunkten und der Interaktion beziehungsweise Bewegung des Mauszeigers hergestellt werden kann, falls ein solcher existiert. 	Die Fixationen von Auge- und Maus werden für jeden Probanden und jeden Typ von AOI nebeneinander dargestellt. Im Graphen ist außerdem eingezeichnet, wann eine Interaktion erfolgte und um welche es sich dabei handelt.
<ul style="list-style-type: none"> • Es müssen alle Zustände und Übergänge zwischen diesen erkennbar sein. Konkret bedeutet das, dass dargestellt werden muss, welche Interaktionen zu welchem Zeitpunkt welchen Zustand des Stimulus erzeugt haben. 	Alle Zustandsübergänge des Stimulus eines Probanden sind im Graphen eingezeichnet und im Interaktionsprotokoll vermerkt, sofern sie innerhalb der gewählten Zeitspanne liegen. Der exakte Stimulus für einen Probanden kann durch Interaktion für jeden präsentierten Zeitpunkt angezeigt werden.
<ul style="list-style-type: none"> • Alle relevanten Informationen zu den angezeigten Elementen müssen abgerufen werden können. 	Die Verwendung von Tooltips für alle im Graphen eingezeichneten Elemente gewährleistet die Verfügbarkeit relevanter Details. Der exakte Stimulus für jeden Zeitpunkt innerhalb des präsentierten Bearbeitungszeitraums kann in einem neuen Fenster angezeigt werden.

Tabelle 1: Gegenüberstellung ursprünglicher Anforderungen und deren Umsetzung

Tabellarische Gegenüberstellung der genannten Anforderungen und der Eigenschaften der konzipierten Visualisierung. Es wird beschrieben, wie diese Visualisierung die geforderten Punkte umsetzt.

Für jede Anforderung wurde in *Tabelle 1: Gegenüberstellung ursprünglicher Anforderungen und deren Umsetzung* darauf eingegangen, wie diese im Konzept realisiert wurde. Auch wenn das Konzept die geforderten Eigenschaften in vielerlei Hinsicht aufweist, lässt sich erst durch die Implementierung und anschließende Anwendung des Konzepts feststellen, wie geeignet die Umsetzung der einzelnen Anforderungen für die genannten Zwecke ist.

Ein Problem ist, dass das Konzept für eine überschaubare Anzahl an Probanden ausgelegt ist. Sowohl die Liste der Probanden als auch die Darstellung dieser im Graphen ist nur für eine begrenzte Anzahl Probanden übersichtlich.

Die Gegenüberstellung der Daten einzelner Probanden ist dagegen durch die gleichartige Darstellung im Kreis und die vielfältigen Auswahlmöglichkeiten der darzustellenden Probanden vorteilhaft gelöst.

Die Interaktionsmöglichkeiten zum Zoomen und Filtern der Daten und zur Bereitstellung detaillierter Informationen sind vorgesehen und ermöglichen die Analyse der Daten eines einzelnen Probanden. Die Verwendung einer einzelnen Zeitachse für die Positionierung aller Datenelemente macht die einfache Zuordnung dieser untereinander möglich.

Es sind zwar alle Interaktionen dargestellt, allerdings ist nicht direkt ersichtlich, wie lange einzelne Scroll-Interaktionen andauern. Mit den im Graphen dargestellten Informationen ist es außerdem nur bedingt möglich, Rückschlüsse auf die Veränderungen am Stimulus oder dessen Zustand zu schließen.

Alle vorhandenen Informationen werden verwertet und können als Ergänzung zu den zugehörigen Elementen abgerufen werden. Das Auslesen des Zustands eines Stimulus zu einem konkreten Zeitpunkt ist für die Analyse der Interaktion mit interaktiven Stimuli besonders hilfreich.

Insgesamt konnten mit dem Konzept ein Großteil der Anforderungen erfolgreich umgesetzt werden. Der Mangel an Darstellungsvermögen für eine große Anzahl an Probanden und das Defizit einer Visualisierung der Zustände verschiedener Stimuli zeigen die Grenzen des Konzepts. Für den detaillierten Vergleich der Daten weniger Probanden und das Erkennen von Mustern in diesen Daten scheint das Konzept hingegen gut geeignet zu sein.

6 Implementierung eines Prototyps

Dieses Kapitel enthält die bei der Realisierung des Visualisierungskonzepts berücksichtigten Ideen und beschreibt wesentliche Aspekte des Programms *Inseyet*, dem Prototyp zur Umsetzung meines Konzepts. Dabei wird zunächst das Ergebnis, also *Inseyet* selbst, mithilfe der Programmoberfläche vorgestellt. Anschließend wird die Visualisierungspipeline durchlaufen und dokumentiert, wie sich diese in der programminternen Datenverarbeitung widerspiegelt: Angefangen bei der Datengenerierung, wie sie bereits in Kapitel 5 *Lösungskonzept* beschrieben wurde, über den Transport der Daten in eine interne Datenstruktur und die sinnvolle Verknüpfung dieser bis zur Anzeige der Informationen im Programm.

Die prototypische Umsetzung des Konzepts erfolgte mit Microsoft Visual Studio 2013 in der Programmiersprache C#. Dabei wurden verschiedene Komponenten des .NET Frameworks 4.5 verwendet. Dazu zählt insbesondere die *Windows Presentation Foundation* (WPF), welche sich gut für die Erstellung von grafischen Benutzungsschnittstellen eignet.

Das Programm wurde intern entsprechend dem Architekturmuster *Model-View-ViewModel* (MVVM) gestaltet, ein Pattern das mit WPF einhergeht (siehe Kapitel 6.2 *Architektur*).

Durch die Arbeit am Konzept ist bekannt, welche Daten verarbeitet werden und wie diese in der Programmoberfläche dargestellt werden sollen. Zudem zeigt das Konzept, welche Einstellungen durch einen Anwender vorgenommen werden sollen. Deshalb ist es möglich, die Benutzungsschnittstelle unabhängig von der internen Datenstruktur zu entwickeln und zu beschreiben. Ich beginne mit der Beschreibung der GUI.

6.1 Die Benutzungsschnittstelle von Inseyet

Inseyet wurde in seiner Erscheinung bewusst schlicht gehalten, wie es die in Kapitel 2.3.2 *Grundsätze der Dialoggestaltung* beschriebenen Normen vorschlagen. Ziel war es, die Bedienung so zu gestalten, dass keine Einarbeitung erforderlich ist, sofern die mit diesem Programm zu lösende Aufgabe bekannt ist.

Alle Bedienelemente der Benutzungsschnittstelle wurden ungeachtet dessen mit Tooltips versehen. Diese geben entweder an, welcher Befehl durch die Interaktion mit den Bedienelementen ausgeführt wird oder beinhalten zusätzliche Informationen, Hinweise oder Beispiele für die Verwendung der Elemente. So ist für den Anwender zum Beispiel ersichtlich, dass fast alle vorhandenen Bedienelemente nicht nur die Art der Visualisierung beeinflussen, sondern auch ein Neuzeichnen auslösen.

6.1.1 Aufbau des Hauptfensters

Bei erstmaligem Öffnen des Programms ist bereits ein leerer Reiter (*Tab*) enthalten (siehe *Abbildung 20: Hauptfenster unmittelbar nach dem Start des Programms*). Weil zu diesem Zeitpunkt noch keine Daten im Programm vorhanden sind, ist nur das Feld für die Einstellungen von Bedeutung. Der Button *Add Participant* in diesem Feld ist das größte und meines Erachtens auffälligste Element. In der Mitte des Fensters ist außerdem der Hinweis „*Please add a participant [...]*“ zu finden. Alle anderen Bedienelemente führen in diesem Stadium zu keiner für den Anwender sichtbaren Veränderung.

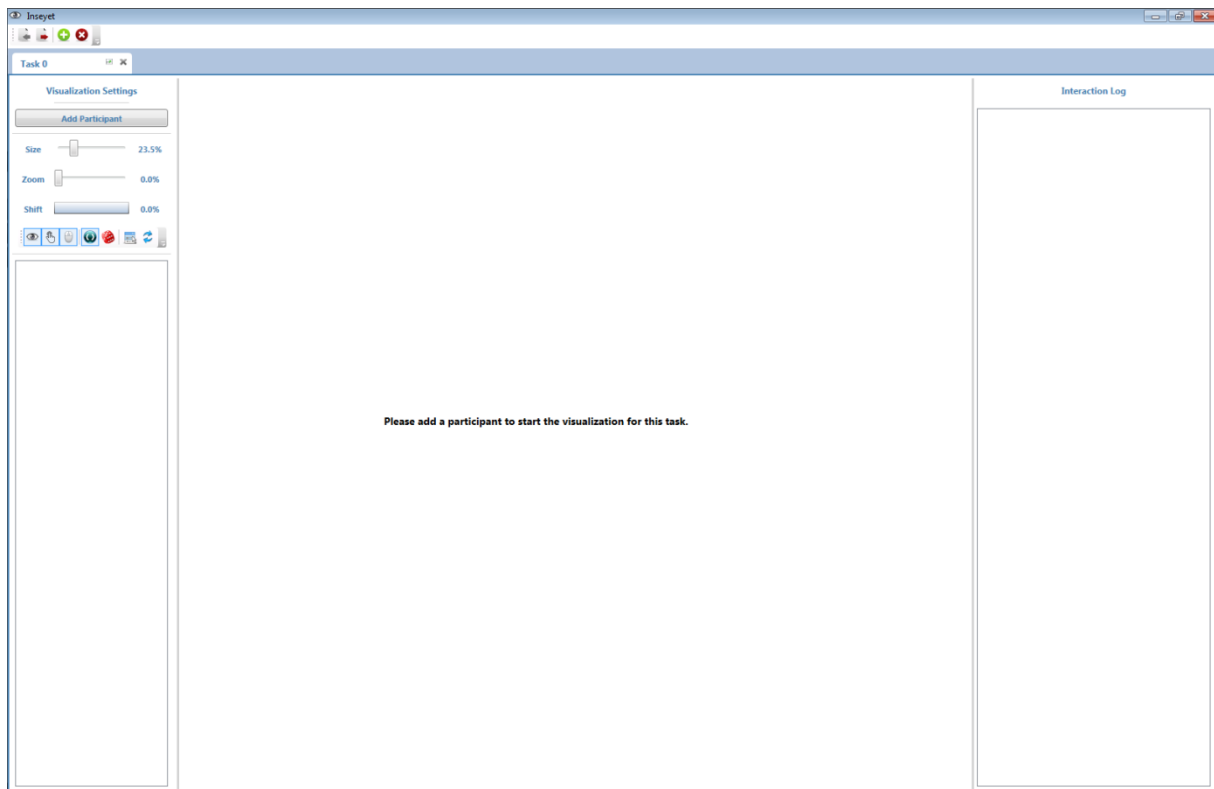


Abbildung 20: Hauptfenster unmittelbar nach dem Start des Programms

Das Programmfenster von Inseyet direkt nach dem Start der Anwendung. In diesem Fenster ist neben der Menüleiste ein Reiter zu sehen, der die Bedienelemente zur Konfiguration der Visualisierung enthält.

Hat der Benutzer erfolgreich einen oder mehrere Probanden durch Interaktion über den Button *Add Participant* hinzugefügt, müssen diese selektiert werden, damit die zugehörigen Daten visualisiert werden. Ein entsprechender Hinweis wird ebenfalls in der Mitte des Fensters angezeigt. Sofern für keinen der ausgewählten Probanden Daten vorhanden sind, wird der Anwender an gleicher Stelle aufgefordert Daten für diese einzulesen. Der Hinweis enthält dann eine genaue Beschreibung des Vorgehens zum Einlesen von Daten für einen Probanden.

Der Anwender muss auf der einen Seite nur unbedingt erforderliche Schritte vornehmen (es reicht beispielsweise das Einlesen der Daten zur Interaktion, um einen Graphen anzuzeigen). Auf der anderen Seite kann er im Grunde nichts falsch machen oder etwas tun, das ihn nicht zum gewünschten Ergebnis führt, weil es sich um wenige, klar formulierte Schritte handelt.

Wie in *Abbildung 21: Hauptfenster von Inseyet mit Graph* zu sehen, besteht die Programmoberfläche von *Inseyet* aus wenigen Bereichen, von denen der Graph (3) mit zugehörigem Interaktionsprotokoll (5) den größten Teil beansprucht. Menü- und Reiterleiste (1) stehen im Hintergrund zur Verfügung, weil sie in der Regel nur selten gebraucht werden.

Oben links im Reiter (2) sind sämtliche Einstellungsmöglichkeiten für die Visualisierung zu finden. Es wird erwartet, dass dieser Bereich zuerst fixiert wird, weil westliches Lesen von links nach rechts und von oben nach unten erfolgt. Aus diesem Grund wurde der Button zum Hinzufügen von Probanden in voller Breite dort platziert. Des Weiteren sind dort drei verschiedene Schieberegler zum Einstellen des Anzeigebereichs vorhanden.

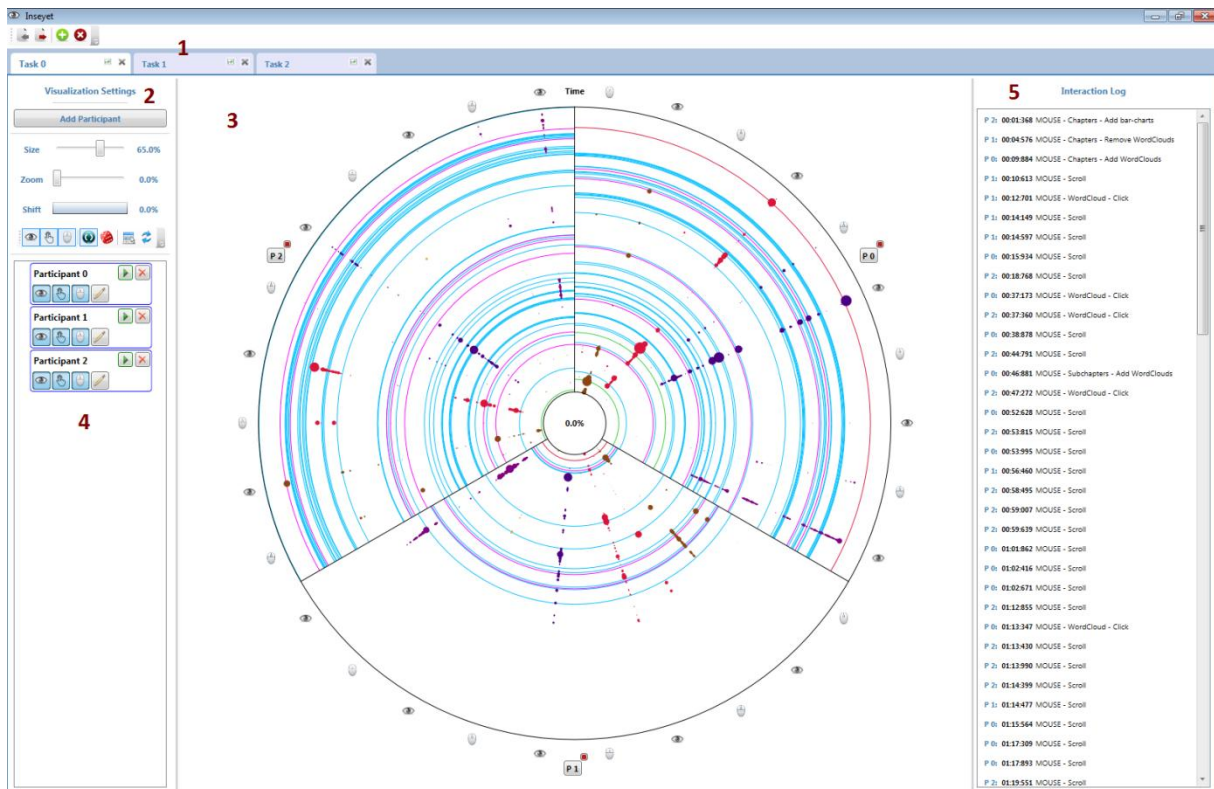


Abbildung 21: Hauptfenster von Inseyet mit Graph

Das Programmfenster nach dem Einlesen und Visualisieren verschiedener Daten. Gezeigt sind Menüleiste und Reitersteuerung (1), Einstellungsmöglichkeiten für die Visualisierung (2), die Darstellungsfläche für die Visualisierung (3), eine Liste der Probanden (4) und eine Liste relevanter Einträge aus den Ereignisprotokollen der in der Visualisierung berücksichtigten Probanden (5).

Es kann festgelegt werden, welche Informationen visualisiert werden sollen. Der Bereich unterhalb der Einstellungen (4) zeigt vorhandene Probanden an und gibt Auskunft darüber, welche Daten vorhanden sind und welche bei der Selektion des jeweiligen Listenelements visualisiert werden.

6.1.2 Bereiche des Programmfensters

Nachfolgend werden die einzelnen Bereiche des Hauptfensters und deren Bedienelemente genauer beschrieben und der Nutzen dieser für den Anwender verdeutlicht.

Mit den Buttons in grüner und roter Farbe in der Menüleiste können neue Reiter hinzugefügt oder bestehende Reiter entfernt werden (siehe *Abbildung 22: Menü- und Reiterleiste von Inseyet*). Jeder Reiter stellt dabei eine einzelne Aufgabe (*Task*) einer Studie dar. Ein Reiter enthält idealer Weise die Daten aller Probanden, die diese Aufgabe im Rahmen der Studie durchgeführt haben. Diese eins zu eins Beziehung von Reiter und Aufgabe ist deshalb naheliegend, weil in der Regel die Daten von verschiedenen Probanden, die die gleiche Aufgabe absolviert haben, verglichen werden sollen.

Menüleiste und Reiterleiste

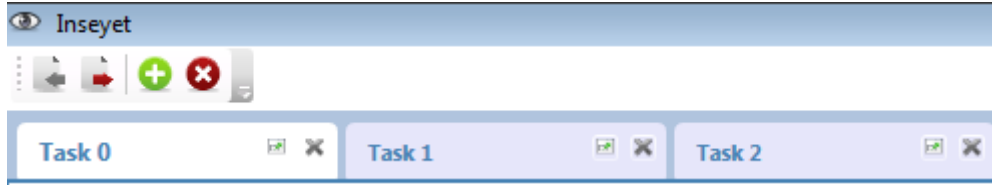


Abbildung 22: Menü- und Reiterleiste von Inseyet

Die Menüleiste enthält in dieser Reihenfolge Buttons für den Im- beziehungsweise Export von Projekten und das Hinzufügen beziehungsweise Entfernen einzelner Reiter. Die Reiterleiste befindet sich unmittelbar unter der Menüleiste und enthält drei Reiter, die neben einem frei wählbaren Titel auch Buttons zum Öffnen als Fenster und zum Schließen des jeweiligen Reiters enthalten.

Die anderen Buttons in der Menüleiste dienen dem Im- beziehungsweise Export von Projekten im programmeigenen Datenformat, wobei ein Projekt schlicht die Summe aller offenen Reiter (also Aufgaben) und deren Inhalt (also Probanden) ist. Weil das Programm die verwendeten Daten nur visualisiert und nicht verändert, sind Funktionen wie Undo (Eingabe rückgängig machen) und Redo (Eingabe wiederholen) oder explizites Speichern nicht zwingend erforderlich. Gleichzeitig ist die zu jeder Operation inverse Aktion denkbar einfach: Aktiviert man einen Schalter, kann man ihn auf die gleiche Weise wieder deaktivieren. Verändert man die Position eines Reglers, kann man ihn genauso wieder auf die Ursprungsposition bewegen.

Dem Anwender steht es frei, die Reiter selbstständig umzubenennen, um die damit repräsentierten Aufgaben erkenntlich zu machen. Die entsprechende Interaktion wird über einen Doppelklick auf den Kopf des Reiters angestoßen. Der anfängliche Name spiegelt lediglich die interne ID wider. Außerdem lässt sich jeder Reiter auch aus der Reiterleiste ausgliedern und gleichzeitig in einem separaten Fenster öffnen. Das ermöglicht die gleichzeitige Darstellung der Daten zu mehreren Aufgaben nebeneinander. Natürlich kann das entstandene Fenster jederzeit wieder als Reiter eingegliedert werden.

Die Einstellungen

Neben dem bereits erwähnten Button *Add Participant* zum Hinzufügen von Probanden zu einer Aufgabe gibt es zehn weitere Bedienelemente, die als Gruppe von Einstellungsmöglichkeiten zur Verfügung stehen. Darunter sind drei Schieberegler und sieben Buttons, die alle der Konfiguration der Visualisierung dienen. Die ersten fünf von ihnen fungieren als Schalter (siehe *Abbildung 23: Bedienelemente zur Steuerung der Visualisierung in Inseyet*, von links nach rechts). Sie lassen sich also durch einen einfachen Klick an- beziehungsweise abstellen.

Der oberste Schieberegler mit der Beschriftung *Size* verändert die Größe des gesamten Graphen im entsprechenden Reiter beziehungsweise Fenster, also die Größe aller dargestellten Elemente in gleichem Maß. Das ist deshalb sinnvoll, um verschiedenen Bedingungen wie Bildschirmgröße, -auflösung oder der Anzeige unterschiedlicher Graphen in mehreren Fenstern gerecht zu werden.

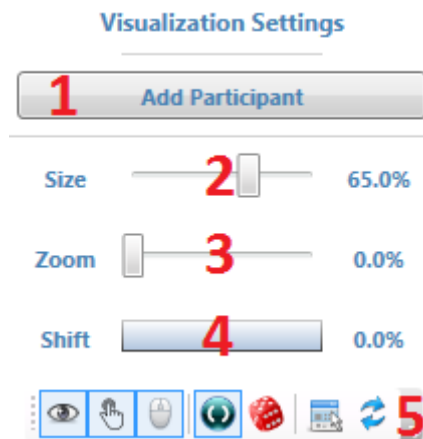


Abbildung 23: Bedienelemente zur Steuerung der Visualisierung in Inseyet

Die Einstellungsmöglichkeiten von Inseyet für die Visualisierung. Neben einem Button zum Hinzufügen von Probanden (1) und drei Schiebereglern zum Einstellen der Größe des Graphen (2), angezeigter Interaktionsdauer (3) und Interaktionsstartzeitpunkt (4) sind sieben weitere Buttons in einer Leiste (5) zu sehen. Die ersten fünf der Buttons (mit den Symbolen Auge, Hand, Maus, An/Aus und Würfel) dienen als Schalter.

Der mittlere Schieberegler *Zoom* verringert die angezeigte Zeitspanne bei gleichbleibender Größe des Graphen. Das bedeutet, dass nur ein Ausschnitt aller vorhandenen Daten und damit auch weniger Elemente dargestellt wird beziehungsweise werden. Weil die Größe der einzelnen Elemente von der angezeigten Zeitspanne abhängt, werden alle Elemente innerhalb dieser Zeitspanne entsprechend größer angezeigt. Benutzt man diesen Regler, liegt der angezeigte Bereich zwischen dem Start der Interaktion, also dem Nullpunkt und dem Ende der Interaktion abzüglich des durch den Regler ermittelten prozentualen Anteils der Gesamtspanne zwischen Start und Ende der Interaktion. Dies lässt sich mit folgender Formel zum Ausdruck bringen:

$$\text{Angezeigte Zeitspanne} = \text{Start Interaktion bis (Ende Int. - Ende Int. * Zoomfaktor)}$$

Der Startwert bei diesem Beispiel liegt stets beim Nullpunkt der Interaktion. Um auch solche Bereiche vergrößert darstellen zu können, die näher am Ende der Interaktion als am Anfang liegen, bedarf es einer Möglichkeit, den Startwert zu verändern. Zu diesem Zweck existiert der dritte Schieberegler mit der Bezeichnung *Shift*. Er ermöglicht das Anpassen des Startwerts der Interaktion und damit die Darstellung verschiedener Bereiche mit erhöhtem Zoomfaktor. Dieser Zeitversatz wirkt sich wie folgt auf obige Formel aus:

$$\text{Angezeigte Zeitspanne} = \text{Start} + \text{Versatz bis (Ende} + \text{Versatz - Ende * Zoomfaktor)}$$

Die Größe des Reglers für den Zeitversatz ist dabei abhängig vom Wert des Zoomreglers. Ist der Zoomfaktor bei 0,0%, dann wird bereits alles angezeigt und der Regler hat die volle Größe. Nimmt der Zoomfaktor zu, verkleinert sich der Regler *Shift*. Die Verschiebung des Startzeitpunkts bei einem Zoomfaktor von 0,0% hätte zur Folge, dass einerseits der zur Verfügung stehende Platz nicht voll genutzt würde, während andererseits vorhandene Informationen nicht zu sehen wären. Deshalb bietet der Regler für den Versatz in diesem Fall keinen Spielraum zum Verschieben (siehe *Abbildung 24: Zusammenhang zwischen Zoomfaktor und Größe des Versatzreglers*).



Abbildung 24: Zusammenhang zwischen Zoomfaktor und Größe des Versatzreglers

Die Veränderung der Breite des unteren Reglers (Shift) in Abhängigkeit des Werts des oberen Reglers (Zoom). Dabei entspricht die Breite des unteren Reglers stets 100-X% seiner maximalen Breite, wobei X der Wert des oberen Reglers ist.

Ein ähnliches Phänomen tritt immer dann auf, wenn der Wert des Versatzreglers den Wert des Zoomreglers übersteigt. Das bedeutet, dass der anzuzeigende Bereich über das Ende der Interaktion hinausgeht. Aus diesem Grund wird dem Anwender bereits durch die Größe des Reglers suggeriert, dass dieser nur so weit verschoben werden kann, wie es der Zoomfaktor erlaubt.

Werden die Daten mehrerer Probanden in Abhängigkeit voneinander dargestellt, orientiert sich der Zoomfaktor für alle Elemente zusätzlich an der längsten Zeitspanne. Diese entspricht der Differenz zwischen einem gemeinsamen Startwert und dem spätesten Ende einer Interaktion unter diesen Probanden.

Die verbleibenden sieben Bedienelemente wurden alle in einer Art Werkzeugleiste nebeneinander angeordnet.



Abbildung 25: Buttons zum Anpassen der Visualisierung in Inseyet

Die Buttons zur Konfiguration der Visualisierung. Die Konfiguration betrifft dabei die Anzeige von Augendaten (Auge), Interaktionsdaten (Hand) und Mausdaten (Maus). Des Weiteren kann die Skalierung der angezeigten Daten angepasst werden (An-/Aus-Schalter), eine zufällige Reihenfolge für die Anzeige der Probanden gewählt werden (Würfel) oder die Visualisierung neu veranlasst werden (Pfeile). Letzteres kann auch mit vorheriger Selektion aller Probanden ausgeführt werden (Mauselektion).

Bei den ersten fünf Buttons (von links nach rechts) handelt es sich um Schalter mit Symbolen für Augenbewegung, Interaktion, Mausbewegung, Zustand und Zufall (siehe *Abbildung 25: Buttons zum Anpassen der Visualisierung in Inseyet*). Der Zustand *ein* wird dabei durch eine quadratische, farbliche Umrandung hervorgehoben. Die ersten drei dieser Buttons sind standardmäßig aktiviert und steuern, welche Daten bei der Visualisierung berücksichtigt werden sollen. Es gibt drei Arten von Daten, die damit gesteuert werden können: Daten zur Augenbewegung, Daten zur Interaktion und Daten zur Mausbewegung.

Wird einer dieser drei Schalter deaktiviert, dann werden die Daten des entsprechenden Typs nicht mehr visualisiert. Weil diese Einstellung auch für jeden Probanden einzeln vorgenommen werden kann, sind die Daten für diesen genau dann zu sehen, wenn sie sowohl in der allgemeinen Einstellung als auch für den einzelnen Probanden selbst aktiviert sind. Die Einstellung für einen einzelnen Probanden wird ignoriert, wenn die allgemeine Einstellung deaktiviert ist.

tiviert ist. Diese Konfigurationsmöglichkeiten sind hilfreich, um *Visual Clutter* zu reduzieren, falls nicht alle Datentypen für die Analyse der angezeigten Daten von Belang sind.

Der nächste Schalter ist in initialem Zustand ebenfalls aktiviert und steuert, ob für die Darstellung der Daten eines Probanden der längste Interaktionszeitraum unter allen darzustellenden Probanden verwendet werden soll. Das hat Einfluss darauf, wie die Zeitachse beziehungsweise der Radius verwendet wird und interpretiert werden muss. Für den Fall, dass der Schalter im Zustand *ein* ist, richtet sich das Maximum der Zeitachse nach dem Ende der Interaktion des Probanden, der für die Bewältigung der Aufgabe am meisten Zeit beansprucht hat. Folglich gilt, dass der Abstand zum Kreismittelpunkt für Elemente mit gleichem Zeitstempel für alle Probanden gleich ist. Gleiches gilt für den Start der Interaktion, wobei dieser generell als Nullpunkt zu verstehen ist. Betrachtet werden dabei natürlich nur die Probanden, deren Daten auch visualisiert werden.

Werden Start- beziehungsweise Endzeitpunkt des entsprechenden Probanden durch Zoom oder Zeitversatz verändert, betrifft das auch alle anderen Probanden. Für diese hängt der Anzeigebereich dann direkt vom Anzeigebereich des Probanden ab, durch den das Maximum der Zeitachse definiert wurde. Das Betätigen des Schalters stößt die Visualisierung der Daten mit den entsprechenden Einstellungen an.

Ist der Schalter für den Wechsel zwischen absoluter und relativer Zeit im Zustand *aus*, stellt die Zeitachse einen relativen Bezug zu den einzelnen Interaktionszeiträumen der Probanden dar. Das heißt, dass beispielsweise jedes Element, das einem Betrag von 90% auf der Zeitachse zuzuordnen ist, Daten mit einem Zeitstempel nach Ablauf von 90% der gesamten Interaktionsdauer des entsprechenden Probanden repräsentiert. Die Elemente verschiedener Probanden, auf die das zutrifft, können demnach verschiedene Zeitstempel haben.

Weil die Reihenfolge, in der die Probanden an der Studie mitwirkten oder in der sie im Programm erfasst wurden, keinen Einfluss auf die Interpretation der Daten haben sollte, kann man die Reihenfolge der Probanden in der Visualisierung randomisieren. Ist der zugehörige Schalter (an fünfter Stelle von links) aktiviert, wird bei jeder Aktualisierung des Graphen eine zufällige Reihenfolge für die Darstellung der Probanden generiert.

Die letzten beiden Buttons dienen der Aktualisierung des Graphen, indem sie die Visualisierung erneut anstoßen. wobei der erste der beiden zuerst alle vorhandenen Probanden selektiert. Letzterer bedient sich der aktuellen Auswahl der Probanden.

Die Probandenliste

Jeder hinzugefügter Proband wird durch ein eigenes Feld, das aus einem Label und verschiedenen Buttons zusammengesetzt ist, in der Liste repräsentiert (siehe *Abbildung 26: Liste der Probanden in Inseyet*). Das Label enthält dabei den frei wählbaren Namen des Probanden. Die ersten drei Buttons (von links nach rechts) fungieren als Schalter und steuern, welche Daten bei der Visualisierung für diesen Probanden berücksichtigt werden sollen. Sie tragen die gleichen Symbole für Augenbewegung, Interaktion und Mausbewegung, wie die Schalter in den allgemeinen Einstellungen. Die Funktionsweise unterscheidet sich ebenfalls nur insofern, dass sich das Betätigen einer der Schalter lediglich auf die Visualisierung der Daten eines einzigen Probanden auswirkt.



Abbildung 26: Liste der Probanden in Inset

Drei Einträge in der Liste der Probanden. Jeder Eintrag enthält die Bezeichnung des Probanden, einen Button „Play“ (grüner Pfeil) zum Abspielen des zugehörigen Videos, einen Button „Remove“ (rotes X) zum Entfernen des Eintrags, einen Button „Edit“ (Bleistift) zum Einlesen von Dateien sowie drei Buttons zur Steuerung der Anzeige von Augendaten (Auge), Interaktionsdaten (Hand) und Mausdaten (Maus).

Ein weiterer Button dient dem Bearbeiten der Daten eines Probanden (vierter von links) beziehungsweise dem Öffnen des Dialogs für das Einlesen der Daten. Dieser wird durch einen Bleistift symbolisiert.

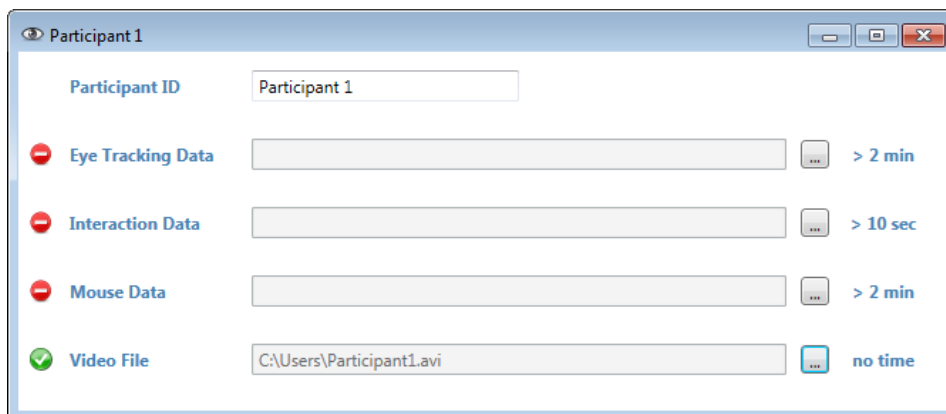


Abbildung 27: Dialog zum Einlesen der Daten eines Probanden

Der Dialog besteht neben einem Textfeld zum Anpassen der Bezeichnung des Probanden aus einer Zeile für jeden Datentyp. Diese enthalten von links nach rechts jeweils ein Statussymbol, den Datentyp, falls vorhanden einen Dateipfad, einen Button zum Einlesen der entsprechenden Datei und eine Anzeige für die voraussichtlich für das Einlesen benötigte Zeit.

Benutzt der Anwender diesen, öffnet sich das in *Abbildung 27: Dialog zum Einlesen der Daten eines Probanden* gezeigte Dialogfenster. An oberster Stelle befindet sich das Textfeld zum Anpassen der Bezeichnung des entsprechenden Probanden. Neben diesem sind für jeden relevanten Datentyp eine Zeile aus Statussymbol, Namen des Typs, Anzeigefeld, Button und Zeitangabe zu sehen. Das Statussymbol gibt an, ob bereits Daten des jeweiligen Typs erfolgreich eingelesen wurden (grün) oder noch nicht vorhanden sind (rot). Wurde die zugehörige Datei frisch eingelesen enthält das Anzeigefeld außerdem den Dateipfad. Beim Programmstart wird die Existenz der einzelnen Dateien erneut geprüft. Mit dem Button kann der Dialog

zur Auswahl eines Dateipfads geöffnet werden. Die Zeitangabe soll als Orientierung für die benötigte Dauer zum Einlesen der jeweiligen Datei auf einem herkömmlichen Computer dienen.

Wurden für einen Probanden erfolgreich Daten eines Datentyps eingelesen, wird der zugehörige Schalter zum Steuern der Visualisierung für diesen Probanden im Zustand *ein* freigeschaltet. Der Anwender kann jetzt das Visualisieren der eingelesenen Daten für diesen Probanden aktivieren beziehungsweise deaktivieren. Die betroffenen Datentypen sind Daten zur Augenbewegung, zur Interaktion sowie zur Mausbewegung. Der Zustand eines Schalters wird ignoriert, wenn die allgemeine Einstellung für den jeweiligen Datentyp auf *aus* gesetzt wurde.

Wurde ein Dateipfad für die Videoaufzeichnung der Interaktion eines Probanden angegeben, wird der entsprechende Button zum Abspielen des Videos ebenfalls freigeschaltet. Diese Aufzeichnungen werden in der Regel von einem Eyetracker erstellt, weshalb die erfassten Fixationen üblicherweise im Video eingezeichnet sind. Des Weiteren kann jeder Proband durch einen Klick auf den Button zum Entfernen (durch ein rotes X gekennzeichnet) wieder aus der Liste gelöscht werden.

Der Graph

Der große Anzeigebereich in der Mitte eines Reiters ist für den Graphen vorgesehen (siehe *Abbildung 21: Hauptfenster von Inseyet mit Graph*). Unmittelbar damit verbunden ist die in Abschnitt *Das Interaktionsprotokoll* beschriebene Liste von Interaktionen. Im Nachfolgenden werden die einzelnen Elemente des Graphen sowie verschiedene Interaktionsmöglichkeiten beschrieben.

Der in *Abbildung 28: Darstellung der Daten dreier Probanden mit Inseyet* dargestellte Graph besteht aus einem Kreis, der in drei gleichartige Kreissegmente unterteilt ist. Jedes dieser Segmente ist einem einzigen Probanden zuzuordnen. Um welchen Probanden es sich jeweils handelt, wird durch die Buttons außerhalb des Kreises gekennzeichnet. Sie tragen die Aufschrift *P 0*, *P 1* und *P 2*, was jeweils dem ersten und den letzten beiden Zeichen der Bezeichnung des zugehörigen Probanden entspricht. Für diese Zeichenkombination wird nachfolgend auch der Begriff *Initialen* verwendet.

Der Button selbst dient einem weiteren Zweck: Klickt der Anwender auf diesen, veranlasst er eine erneute Visualisierung, die ausschließlich den entsprechenden Probanden berücksichtigt. Oberhalb dieses Buttons befindet sich stets ein weiterer, sehr viel kleinerer, roter Button, der ebenfalls eine Aktualisierung des Graphen anstößt. Bei dieser werden alle bisher dargestellten Probanden berücksichtigt – mit Ausnahme des ausgewählten.

Die anderen Symbole außerhalb des Kreises kennzeichnen den Typ der Daten auf dem Pfad vom Kreismittelpunkt bis zum jeweiligen Symbol. Es handelt sich dabei entweder um Daten zur Augenbewegung (durch ein Auge symbolisiert) oder um Daten zur Mausbewegung (durch eine Computermaus symbolisiert). Übersteigt die Anzahl der Probanden einen für die Darstellung der Symbole erforderlichen Grenzwert, werden die Hilfssymbole nicht mehr mit eingezeichnet. Dadurch wird vermieden, dass der Graph unnötig überladen wird. Der Grenzwert wird dynamisch aus der Größe des Graphen und der Anzahl der dargestellten Probanden ermittelt.

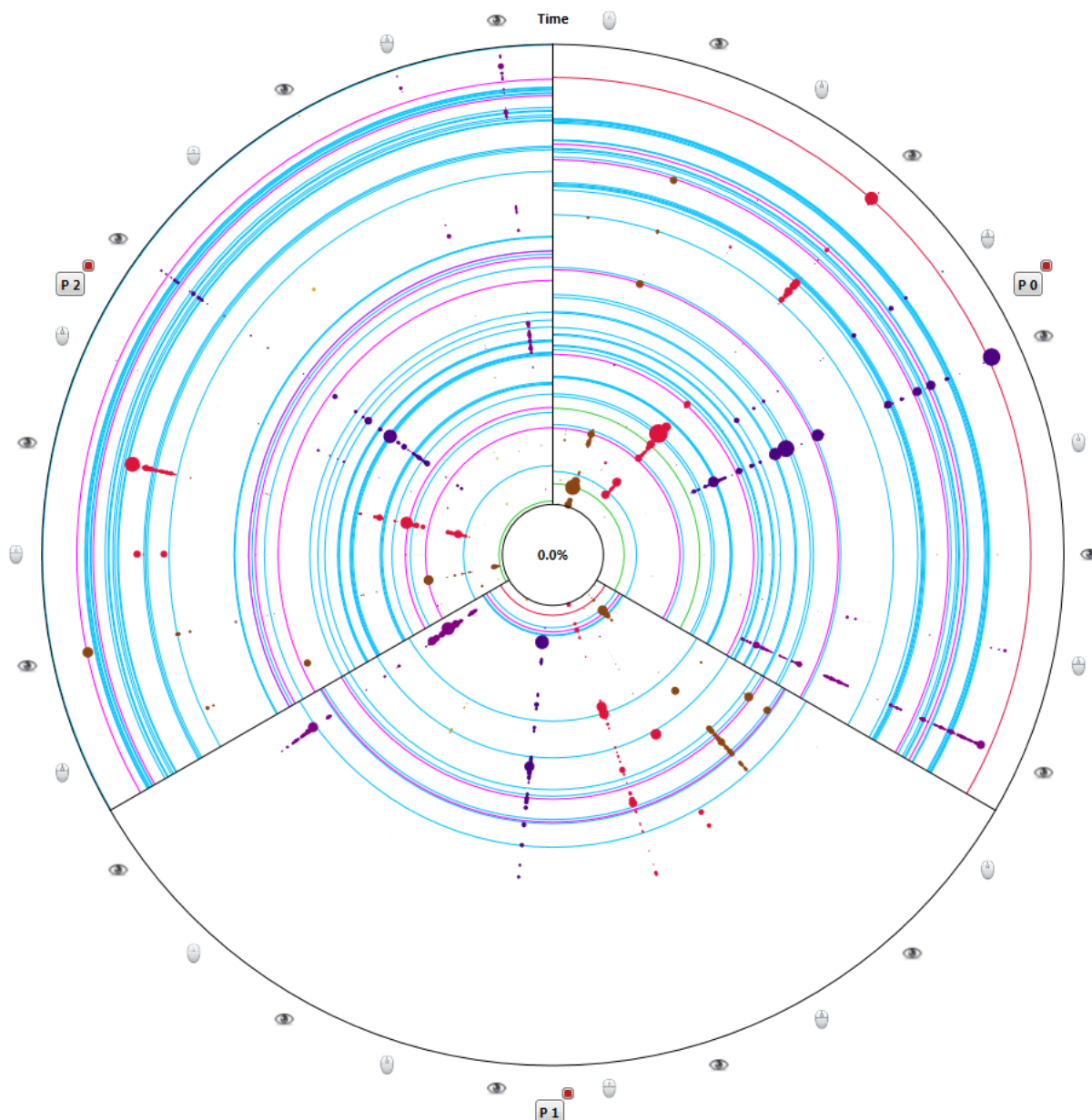


Abbildung 28: Darstellung der Daten dreier Probanden mit Inseyet

Es wurden die Daten dreier Probanden visualisiert. Für alle drei werden sowohl die Daten zur Interaktion als auch die zur Maus- und Augenbewegung dargestellt. Die farbigen Kreisbruchstücke zeigen dabei Interaktionen an, die zur Veränderung des Stimulus geführt haben. Die „Farbtupfer“ stellen die Fixationen von Auge und Maus dar, die anhand der Symbole außerhalb des Kreises unterschieden werden können. Die Buttons außerhalb eines Kreissegments kennzeichnen den zu den Daten gehörenden Probanden in diesem Segment.

Aus dem gleichen Grund wird das Zentrum des Kreises nicht zur Anzeige von Daten verwendet sondern bleibt – bis auf eine Prozentanzeige – leer. Resultat ist, dass auch bei einer größeren Anzahl Probanden die Informationen im Inneren des Graphen entnommen werden kann. Wie das Label *Time* am oberen Ende des Graphen signalisiert, handelt es sich bei den Trennstrichen zwischen den Bereichen für die einzelnen Probanden gleichzeitig um Zeitachsen. Der Startzeitpunkt im Inneren des Kreises wird durch einen Prozentwert angegeben. Der Wert entspricht dem Wert des Schiebereglers *Shift* und gibt damit an, dass die Achse eine Zeitspanne, beginnend bei dem entsprechenden prozentualen Wert der gesamten Interaktionsdauer,

darstellt. Weil sich die Interaktionsdauer jedes Probanden unterscheiden kann, werden konkrete Werte nur im Tooltip für den jeweiligen Probanden angezeigt. Dieses erscheint, wenn der Mauszeiger über dem zugehörigen Button außerhalb des Kreissegments des Probanden verweilt.

Die farbigen Kreisbögen in den Kreissegmenten repräsentieren eine Veränderung des Stimulus. Jede Farbe steht dabei für eine bestimmte Interaktion. Die Farbe Grün steht für alle Interaktionen, bei denen der Stimulus mit Informationen angereichert wird, zum Beispiel durch das Einblenden von zusätzlichen Informationen oder Navigationshilfen. Im Gegenzug bedeutet die rote Farbe, dass Informationen oder Hilfen ausgeblendet wurden. Um welche Informationen es sich dabei konkret handelt kann entweder dem zugehörigen Tooltip oder dem Interaktionsprotokoll (siehe Abschnitt *Das Interaktionsprotokoll* in Kapitel 6.1.2 *Bereiche des Programmfensters*) entnommen werden. Ein helles Blau bedeutet, dass der Stimulus durch Scrollen verändert wurde. Dabei zeigt der Kreisbogen den Beginn der Interaktion. Mit Magenta eingefärbte Kreisbögen signalisieren einen einfachen Klick auf ein Element des Stimulus.

Bei der Farbgebung für die Kreisbögen wurde darauf geachtet, Grundfarben mit hoher Sättigung zu verwenden. Ein Grund ist, dass die Kreisbögen sich im Gegensatz zu den anderen Elementen über die gesamte Breite des Kreissegments erstrecken. Das bedeutet, sie sind auch noch zu sehen, wenn sie teilweise von Elementen für Fixationen zum gleichen Zeitpunkt überdeckt werden. Ein anderer Grund ist, dass eine Vielzahl von Interaktionen theoretisch die gesamte Fläche des Kreissegments bedecken könnte. Die konkreten Farben für die verschiedenen Interaktionen wurden ob ihrer gängigen Assoziation gewählt: Grüne Farbtöne kennzeichnen in der Regel das Hinzufügen von Elementen, während rote Farbtöne für das Entfernen stehen. In diesem Sinne dienen blaue Farbtöne dazu, *neutrale* Interaktionen wie Scrollen auszudrücken. Besondere Ereignisse, wie die Selektion eines Elements durch einen Klick, sollten sich dagegen von den anderen Interaktionen abheben und wurden mit Magenta kodiert. Gelb wurden vermieden, weil es auf weißem Hintergrund zu kontrastarm ist.

Der Abstand zwischen zwei Kreisbögen beschreibt die Anzeigedauer des durch die letzte, vorausgegangene Interaktion entstandenen Stimulus. Der Abstand ist dabei proportional zum prozentualen Anteil des Stimulus an der im Graphen gezeigten Zeitspanne. Entspricht der Abstand also 20% der Zeitachse, war der entsprechende Stimulus 20% der Zeit innerhalb des gezeigten Ausschnitts zu sehen.

Die in dunkleren Farben eingezeichneten und in gleicher Farbe ausgefüllten Kreise repräsentieren Fixationen mit dem Auge und mit der Maus. Worum es sich jeweils handelt, lässt sich an den Symbolen außerhalb des Kreises oder am jeweiligen Tooltip erkennen. Die verschiedenen Farben kodieren dabei den Typ beziehungsweise die Ebene der betroffenen AOI (siehe Kapitel 5.1.1 *Daten und Datenherkunft*).

Die Ebene *Chapters* wird durch die Farbe Braun dargestellt. Die Farbe Purpur vertritt die Ebene *Subchapters*. Indigo wurde zum Kennzeichnen der Ebene *Pages* eingesetzt, Orange steht für *Paragraphs* und Lila stellt die Ebene *Text* dar. Zusätzlich lässt sich jede Ebene auch über den Winkel innerhalb eines Kreissegments ermitteln. Wichtig ist, dass die Daten für Maus- und Augenbewegung innerhalb einer Ebene mit der gleichen Farbe visualisiert werden.

Die Elemente für die Daten zur Mausbewegung gleicher Ebene sind stets auf einer Linie mit einem kleineren Winkel (im Uhrzeigersinn) angeordnet.

Für die Fixationen aller AOI-Typen wurden Mischfarben mit hoher Sättigung gewählt, damit sie deutlich von den Farben für die Interaktionen unterschieden werden können. Die Farben sind außerdem dunkler als die für die Interaktion, damit sie bei einer Überdeckung besser zu erkennen sind. Die AOI-Typen werden, im Gegensatz zu Interaktionen wie Hinzufügen (grün) oder Entfernen (rot), in der Regel nicht mit einem konkreten Farbton assoziiert. Weiterhin wurde darauf geachtet, dass sich nebeneinanderliegende Farben voneinander unterscheiden lassen.

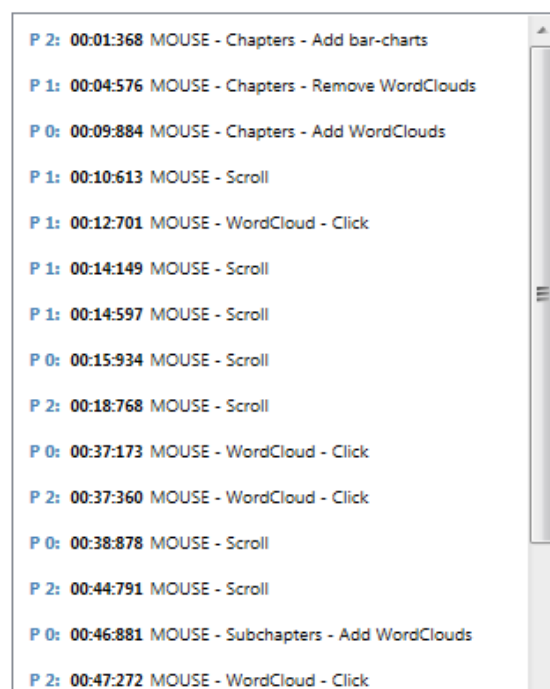
Die Größe der eingezeichneten Kreisflächen orientiert sich vor allem am Verhältnis der Dauer der Fixation gegenüber der insgesamt angezeigten Dauer. Zudem werden Faktoren wie die Anzahl der Probanden, der aktuelle Zoomwert und die Abweichung der Werte zur Bestimmung einer Fixation mit dem Auge beziehungsweise der Maus berücksichtigt.

Für die Positionierung der Kreise wurde – wie für alle anderen eingezeichneten Elemente auch – der zugehörige Zeitstempel verwendet. Die Entfernung zum Kreismittelpunkt hängt also vom gezeigten Ausschnitt ab und richtet sich nach dessen Start- und Endwerten beziehungsweise der daraus resultierenden Dauer.

Das Interaktionsprotokoll

Das Interaktionsprotokoll enthält die Einträge aller im Graphen dargestellten Probanden.

Interaction Log



P 2:	00:01:368	MOUSE - Chapters - Add bar-charts
P 1:	00:04:576	MOUSE - Chapters - Remove WordClouds
P 0:	00:09:884	MOUSE - Chapters - Add WordClouds
P 1:	00:10:613	MOUSE - Scroll
P 1:	00:12:701	MOUSE - WordCloud - Click
P 1:	00:14:149	MOUSE - Scroll
P 1:	00:14:597	MOUSE - Scroll
P 0:	00:15:934	MOUSE - Scroll
P 2:	00:18:768	MOUSE - Scroll
P 0:	00:37:173	MOUSE - WordCloud - Click
P 2:	00:37:360	MOUSE - WordCloud - Click
P 0:	00:38:878	MOUSE - Scroll
P 2:	00:44:791	MOUSE - Scroll
P 0:	00:46:881	MOUSE - Subchapters - Add WordClouds
P 2:	00:47:272	MOUSE - WordCloud - Click

Abbildung 29: Interaktionsprotokoll aller dargestellten Probanden

Der dargestellte Ausschnitt eines Interaktionsprotokolls enthält Einträge dreier Probanden denen die Initialen P0, P1 und P2 zugeordnet sind. Jeder Eintrag enthält außerdem einen Zeitstempel und Schlagworte zur eigentlichen Interaktion.

Ein Eintrag im Interaktionsprotokoll besteht aus den Initialen des zugehörigen Probanden, einem Zeitstempel und ein paar Schlagworten zur Interaktion (siehe *Abbildung 29: Interaktionsprotokoll aller dargestellten Probanden*). Die Schlagworte enthalten das für die entsprechende Interaktion verwendete Eingabegerät, die Art der Interaktion und, falls bekannt, das Element, mit dem interagiert wurde. Das Interaktionsprotokoll zeigt genau die Einträge, die auch im Graphen zu sehen sind. Es werden demnach der tatsächliche Anzeigebereich (also Zoom und Zeitversatz) und die Sichtbarkeitseinstellung der einzelnen Probanden berücksichtigt.

Weil jeder Eintrag mit dem entsprechenden Element im Graphen verknüpft ist, werden bei der Selektion eines Eintrags sowohl der Eintrag selbst als auch das damit verknüpfte Element farblich hervorgehoben. Das ermöglicht dem Anwender eine rasche Zuordnung der Inhalte des Interaktionsprotokolls zu den Kreissegmenten im Graphen. Sowohl De- als auch Mehrfachselektion sind möglich. Für jede Kombination aus Proband und Zeitstempel kann basierend auf den Einträgen in der Logdatei nur ein Eintrag existieren.

6.2 Architektur

Bei der Entwicklung des Prototyps wurde das Architekturmuster *Model-View-ViewModel* (MVVM) verwendet (siehe *Abbildung 30: Das Architekturprinzips Model-View-ViewModel*). Dabei bilden die bei der Studie entstandenen Protokolldateien das Model, das durch das Einlesen dieser Dateien in die interne Datenstruktur, dem ViewModel, konvertiert wird. Ziel ist es, die Daten in diesem Schritt bereits soweit aufzubereiten, dass es zur Anzeige der Daten aus dem ViewModel in einer View keiner weiteren Bearbeitung mehr bedarf.

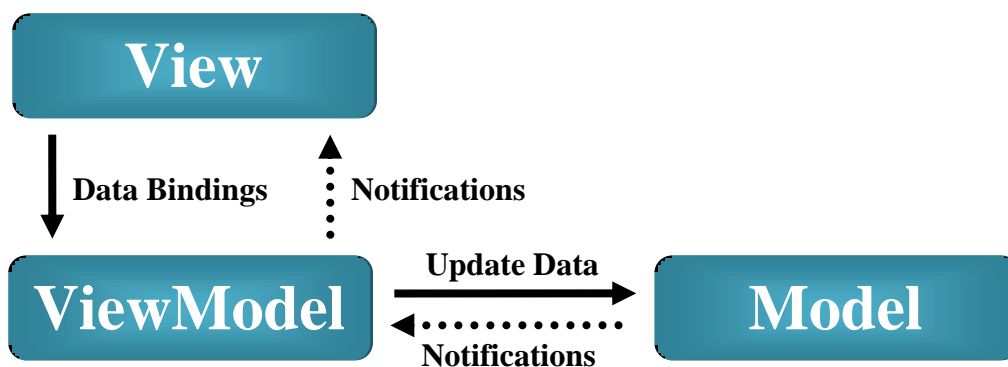


Abbildung 30: Das Architekturprinzips Model-View-ViewModel

Skizze des Patterns Model-View-ViewModel (MVVM). Die Pfeile zeigen die Kommunikation zwischen den Komponenten View und ViewModel beziehungsweise Model und ViewModel. Die View verwendet die Daten aus dem ViewModels mithilfe von Bindings. Das ViewModel informiert die View über Änderungen an den Daten. Das ViewModel kann die Daten im Model aktualisieren und wird von diesem über Änderungen in Kenntnis gesetzt.

Als Model werden dabei die eigentlichen Rohdaten bezeichnet, die für gewöhnlich in Datenbanken oder einzelnen Dateien vorliegen. Letzteres war der Ausgangspunkt für die in *Inseyet* verwendeten Daten: Die Daten zur Interaktion wurden in herkömmlichen *.xlsx-Dateien (Ex-

cel Spreadsheet) gespeichert, die Daten zum Eye-Tracking in *.tsv-Dateien (*Tab-Separated Values*), welche direkt in *.xlsx-Dateien umgewandelt werden können.

Diese Dateien werden dann durch gezieltes Einlesen in eine programmeigene Datenstruktur konvertiert, dem ViewModel. In diesem Schritt werden die Daten bereits so aufbereitet, dass sie später von einer oder mehrerer Views in der GUI dargestellt werden können. Wird eine Datenbank zum Speichern der Daten verwendet, gilt diese ebenfalls als Model. Die Daten müssen dabei zur Ablage in den Tabellen entsprechend dieser aufbereitet werden.

Wesentliches Element des MVVM-Patterns ist die Anbindung der Daten im ViewModel an die Darstellung in den verschiedenen Views:

Beispielsweise stellt das ViewModel alle Probanden, die dieselbe Aufgabe bearbeitet haben, in einer Liste bereit. Eine View selbst zeigt den Inhalt dieser Liste dann auf ihre Weise unverändert an. Jede Veränderung an dieser Liste, durch Hinzufügen oder Entfernen einzelner Probanden oder dem Ändern der Attribute eines solchen Probanden, erfolgt stets im ViewModel. Das ViewModel gibt dann öffentlich bekannt, dass sich diese Liste geändert hat. Wichtig an dieser Stelle ist, dass ein ViewModel die damit verknüpften Views nicht *kennt*. Es weiß also nicht, welche Views die eigenen Daten in welcher Form repräsentieren und kann diese auch nicht steuern oder beeinflussen.

Eine View hat genau ein ViewModel und kennt dieses. Die Datenstruktur muss also so gestaltet sein, dass alle anzuzeigenden Daten in diesem ViewModel vorhanden sind. Wird eine View wie oben beschrieben über eine Aktualisierung informiert, fragt sie die betroffenen Daten erneut ab und zeigt diese dann an. Dadurch zeigen alle Views, die die oben genannte Liste anzeigen, stets die aktuellste Version dieser Liste an.

Für alle voneinander unabhängigen Bereiche der Programmoberfläche von *Inseyet* wurden eigene Gruppen aus View(s) und ViewModel im Code angelegt. Beispiele hierfür sind der Graph, die einzelnen Probanden in der Liste eines Reiters oder sogar ein Reiter selbst. Die einzelnen Views verschiedener Gruppen wurden also teilweise ineinander verschachtelt: Jeder Proband ist in der Liste eines Reiters enthalten, der Graph und die Probandenliste sind im Reiter enthalten und der Reiter selbst wird in der Reiterleiste des Hauptfensters eingebunden.

Jede Gruppe wird dabei im Code durch eine Schnittstelle repräsentiert, die das ViewModel und alle vorhandenen Views für dieses kennt. Sie ermöglicht die Verknüpfung einer oder mehrerer Views mit dem bereits initialisierten ViewModel.

Am Beispiel Probanden lässt sich die Funktionsweise genauer beschreiben: Soll ein neuer Proband zur Liste hinzugefügt werden, wird ein neues Objekt der Gruppe Proband erzeugt. Das neue Objekt wird dann dem ViewModel der Probandenliste hinzugefügt. Diese Veränderung löst dann eine Aktualisierung der View dieser Liste aus. Die Liste zeigt jetzt für jeden Eintrag die entsprechende View aus dessen Gruppe an. Wird in einer der Probanden-Views der Button *Edit* (siehe Abschnitt *Die Probandenliste* in Kapitel 6.1.2 *Bereiche des Programmfensters*) gedrückt, wird eine weitere View für diesen Probanden angezeigt. In diesem Fall handelt es sich um den Dialog zum Einlesen der Daten für diesen Probanden, welcher das gleiche ViewModel verwendet wie die View für den Eintrag in der Liste. Weil sich die Views untereinander nicht kennen und auch das ViewModel die verwendenden Views nicht kennt,

erfolgt der beschriebene Aufruf einer View aus einer anderen heraus über das übergeordnete ViewModel, also dem der Liste.

Auch im Graphen ist für jeden angezeigten Probanden eine View für diesen enthalten. Die View besteht unter anderem aus einem Button mit den Initialen des Probanden (siehe Abschnitt *Der Graph* in Kapitel 6.1.2 *Bereiche des Programmfensters*) und dient damit auch der Zuordnung der angezeigten Daten zu den Probanden in der Liste. Dadurch können im Graphen Interaktionsmöglichkeiten zu den einzelnen Probanden angeboten werden.

Alle anderen Klassen, für die es keine Repräsentation in einem der Programmfenster von *Inseyet* gibt, sind von dieser Strukturierung ausgenommen. Sie stehen als statische Funktionen sowohl den Views als auch den ViewModels zur Verfügung. Beispiele hierfür sind vor allem Klassen für das Konvertieren von Werten, sogenannte Konverter.

Durch die beschriebene Trennung zwischen View, ViewModel und Model ist es einerseits möglich, die Programmoberfläche mit verhältnismäßig wenig Aufwand umzugestalten. Auf der anderen Seite bleibt die GUI von einer Änderung der Struktur der Protokolldateien beziehungsweise des Modells vollkommen unberührt. Die Entkopplung von Präsentationslogik und Datenmodell des Modells ist also gegeben.

6.3 Datenstruktur

Abbildung 31: Überblick über die interne Datenstruktur skizziert die interne Datenstruktur.

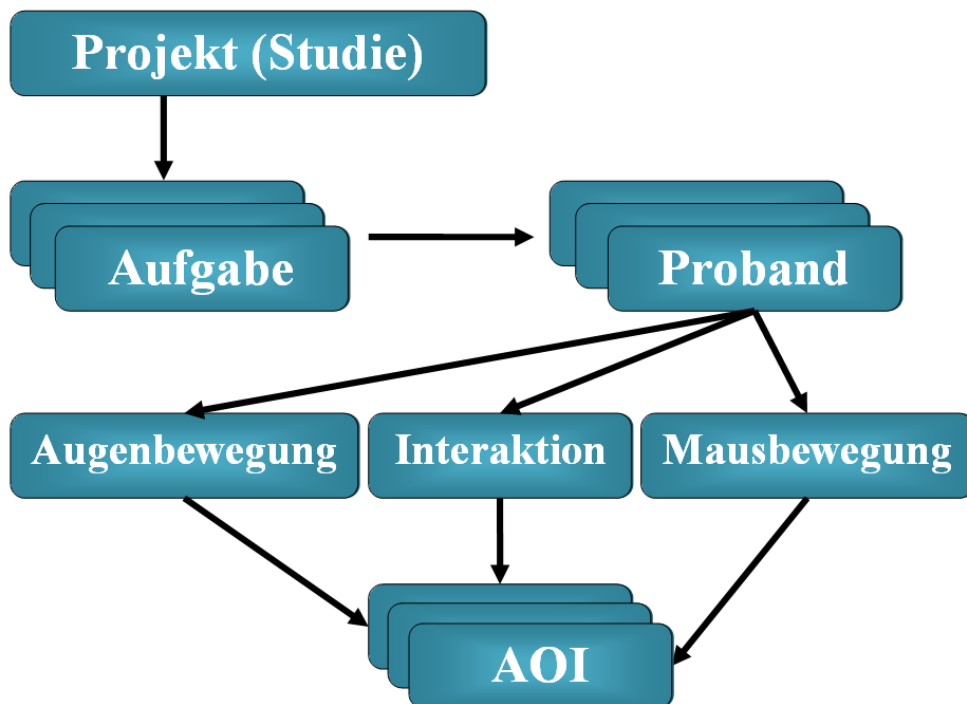


Abbildung 31: Überblick über die interne Datenstruktur

Für jede Studie können mehrere Aufgaben erfasst werden. Jeder Aufgabe kann eine beliebige Anzahl Probanden zugeordnet werden. Für jeden Probanden können Daten zur Augenbewegung, zur Interaktion und zur Mausbewegung hinterlegt werden. Diese Daten bestehen unter anderem aus den einzelnen AOI.

Damit die vorhandenen Daten effizient abgefragt und verarbeitet werden können, werden sie direkt beim Einlesen (siehe Kapitel 6.4 *Datenimport*) gefiltert und strukturiert. Weil jeder Reiter eine eigene Aufgabe einer Studie repräsentiert, kann die Summe aller Reiter in einem Projekt als die Gesamtheit der Daten einer Studie verstanden werden. Das heißt, dass für jede neue Studie auch ein neues Projekt angelegt werden kann. Jeder Reiter enthält eine Liste mit Probanden, für die jeweils Daten zu der durch den Reiter verkörperten Aufgabe vorhanden sind und eingelesen werden können. Informationen zur Datenherkunft sind in Kapitel 5.1.1 *Daten und Datenherkunft* enthalten. Dabei wird jedem neuen Probanden beim Anlegen mitgeteilt, zu welcher Aufgabe er gehört. Eine Zuordnung von Probanden zur Aufgabe ist also jederzeit möglich.

Neben einem Namen und einer innerhalb der Aufgabe eindeutigen ID verfügt jeder Proband über drei Objekte mit den Daten zur Augenbewegung, zur Interaktion und zur Mausbewegung (siehe *Abbildung 31: Überblick über die interne Datenstruktur*). Die Dateipfade zur jeweiligen Quelldatei sind im entsprechenden Objekt hinterlegt und behalten ihre Gültigkeit, solange die angegebene Datei existiert. Gleiches gilt für den Pfad zur Videoaufzeichnung der gesamten Interaktion.

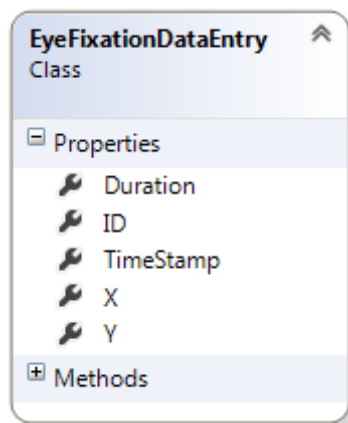


Abbildung 32: Aufbau des Eintrags für eine Fixation

Ein Eintrag für eine Fixation (EyeFixationDataEntry) besteht neben einer ID (ID) aus einem Zeitstempel (TimeStamp), deren Dauer (Duration), und den X- und Y-Koordinaten der Fixation (X, Y).

Die Daten zur Augenbewegung werden in einer Liste gehalten, die die Daten aller Fixationen aus der gewählten Logdatei enthält. Jeder Eintrag besteht aus einer ID, einem Zeitstempel, den X- und Y-Koordinaten der Fixation zu diesem Zeitpunkt und der Fixationsdauer (siehe *Abbildung 32: Aufbau des Eintrags für eine Fixation*). Der Zeitstempel enthält den Startzeitpunkt der Fixation.

Analog dazu werden auch die Daten zur Mausbewegung gehalten (siehe *Abbildung 33: Aufbau der Einträge für Mausbewegung und -fixierung sowie AOI*). Für diese gibt es zwei Listen – eine für die Einträge selbst, eine für die Mausfixierung innerhalb einer AOI. Als Mausfixierung wird dabei eine andauernde Mausbewegung innerhalb einer einzigen, definierten AOI verstanden. Ein Eintrag für die Position des Mauszeigers besteht aus einer ID, einem Zeitstempel, den X- und Y-Koordinaten der Position zu diesem Zeitpunkt und der ID der zu-

geordneten Mausfixierung. Ein Eintrag für die Mausfixierung wurde gegenüber dem Eintrag für eine Fixation erweitert.

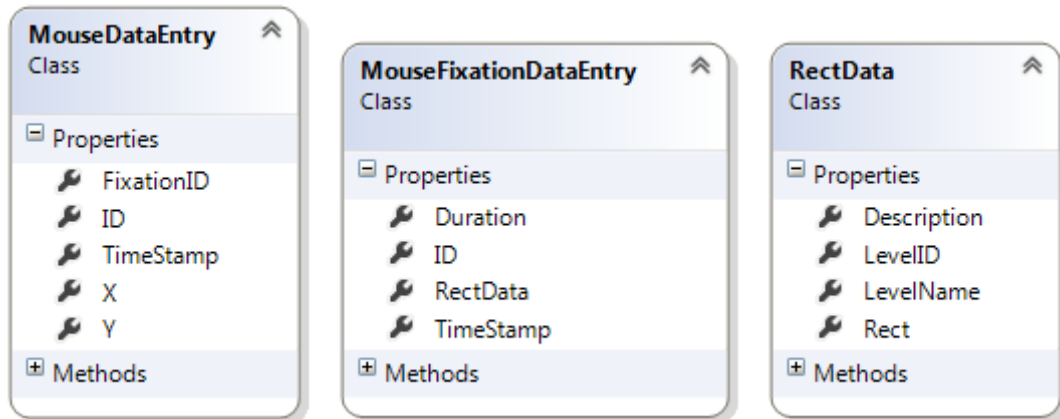


Abbildung 33: Aufbau der Einträge für Mausbewegung und -fixierung sowie AOI

Ein Eintrag für die Position des Mauszeigers (MouseDataEntry, links) besteht neben einer ID (ID) aus einem Zeitstempel (TimeStamp), den X- und Y-Koordinaten der Position (X, Y) und der ID der zugewiesenen Fixierung (FixationID). Ein Eintrag für eine Mausfixierung (MouseFixationDataEntry, mittig) enthält neben der ID (ID) einen Zeitstempel (TimeStamp), die Dauer (Duration), und die Daten der betroffenen AOI (RectData). Ein Eintrag für eine AOI (RectData) besteht neben den geometrischen Daten (Rect) aus dem Typ der AOI (LevelID, LevelName) und einer eindeutigen ID (Description).

Anstatt zweier Koordinaten wird dem Eintrag eine AOI (RectData) mit ihren Eigenschaften und der aktuellen Position hinzugefügt. Die Eigenschaften einer AOI identifizieren diese eindeutig: Level-ID und -Name geben den Typ beziehungsweise die Ebene der AOI an (siehe Kapitel 5.1.1 *Daten und Datenherkunft*). Die Beschreibung ermöglicht dann eine genaue Zuordnung innerhalb dieser Ebene. Weil der Stimulus stets aus Rechtecken besteht und sich die AOI daran orientieren, lässt sich die Position einer AOI zu einem bestimmten Zeitpunkt durch ein solches Rechteck exakt beschreiben.

Die Daten zur Interaktion enthalten neben einer Liste mit den Einträgen zur Interaktion selbst auch eine Liste mit allen durch Interaktion neu entstandenen Stimuli (siehe *Abbildung 34: Aufbau der Einträge für Interaktion, Stimulus und AOI*). Weil nicht jede Interaktion einen neuen Stimulus erzeugt, ist diese Unterteilung sinnvoll. Außerdem verwendet *VarifocalReader* eine Struktur, bei der alle AOI der obersten Ebene statisch sind (Kapitel 5.1.1 *Daten und Datenherkunft*). Das heißt, im Gegensatz zu allen anderen Typen von AOI bleiben diese während der Interaktion bezüglich Position und Sichtbarkeit unverändert. Entsprechend werden die AOI der obersten Ebene nicht für jeden neuen Stimulus erneut ausgelesen oder abgespeichert sondern werden einmalig als Liste in den Interaktionsdaten selbst hinterlegt (für eine solche AOI wird ebenfalls die oben beschriebene Struktur verwendet).

Ein Eintrag zur Interaktion besteht aus einer ID, einem Zeitstempel, dem Typ der Interaktion, der Mausposition, der Position der betroffenen AOI (als Rechteck) und der ID des zu diesem Zeitpunkt gültigen Stimulus. Weil die betroffene AOI für jeden Eintrag in der Logdatei bereits enthalten ist, ist es für eine einfache Abfrage nicht erforderlich, die Mausposition mit dem Objektmodell des aktuellen Stimulus (bestehend aus den einzelnen AOI) abzugleichen.

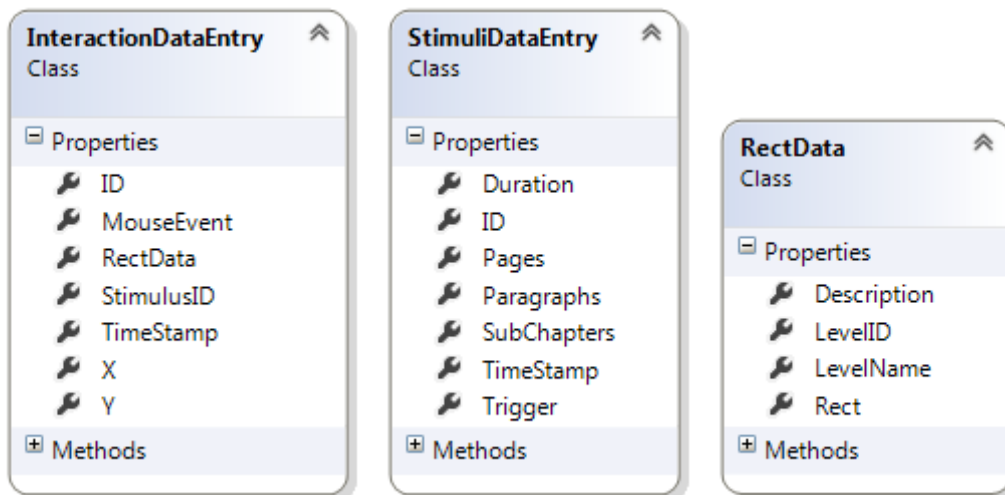


Abbildung 34: Aufbau der Einträge für Interaktion, Stimulus und AOI

Ein Eintrag für eine Interaktion (*InteractionDataEntry*, links) enthält ID (ID), Zeitstempel (*TimeStamp*), Position des Mauszeigers (X, Y), Interaktionstyp (*MouseEvent*), betroffene AOI (*RectData*) und ID des daraus entstandenen Stimulus (*StimulusID*). Ein Eintrag für einen Stimulus (*StimuliDataEntry*, mittig) enthält (ID), Zeitstempel (*TimeStamp*), Gültigkeitsdauer (*Duration*), vorausgehende Interaktion (*Trigger*) und Listen für verschiedene AOI-Typen (*SubChapters*, *Pages*, *Paragraphs*).

Ein Eintrag für einen Stimulus besteht aus einer ID, einem Zeitstempel, der Gültigkeitsdauer des Stimulus, dem Typ der auslösenden Interaktion und einer Liste für jede Type aller in diesem Stimulus sichtbaren AOI. Die Listen für Unterkapitel, Seiten und Paragraphen enthalten dabei die bereits beschriebenen Einträge für AOI. Das auslösende Ereignis für einen Stimulus gleicht immer dem Mausereignis der zugehörigen, auslösenden Interaktion.

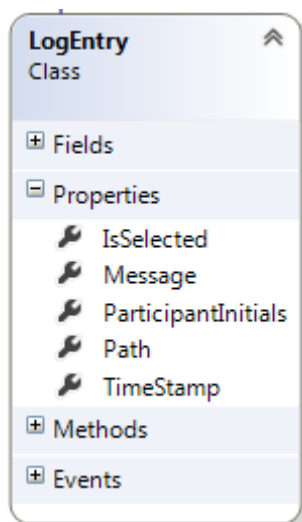


Abbildung 35: Struktur eines Eintrag im Interaktionsprotokoll

Ein Protokolleintrag (*LogEntry*) enthält Zeitstempel (*TimeStamp*), Interaktionstyp (*Message*), Initialen (*ParticipantInitials*) und das Element im Graphen (*Path*, *IsSelected*).

Weil jeder Stimulus sowohl im Graphen selbst als auch im Interaktionsprotokoll repräsentiert werden kann, gibt es außerdem eine Klasse für die Einträge in diesem Protokoll. Sie enthält die Initialen des zugehörigen Probanden, den Zeitstempel, ab dem der Stimulus gültig ist, und eine Kurzbeschreibung der Interaktion, aus dem der Stimulus resultiert. Die Initialen eines Probanden bestehen dabei aus dem ersten und den beiden letzten Zeichen seiner Bezeichnung. Der Grund dafür ist die Nummerierung der Probanden anhand von zweistelligen IDs, die der Bezeichnung in der Regel angefügt werden.

Jeder Eintrag im Interaktionsprotokoll ist über den im Graphen für den zugehörigen Stimulus eingezeichneten Pfad mit diesem verknüpft und weiß, ob dieser selektiert ist. (siehe *Abbildung 35: Struktur eines Eintrag im Interaktionsprotokoll*).

6.4 Datenimport

Der Import der für einen Probanden erfassten Daten erfolgt ausschließlich über einen entsprechenden Dialog (siehe Abschnitt *Die Probandenliste* in Kapitel 6.1.2 *Bereiche des Programmfensters*). Es können nacheinander die Daten zur Augenbewegung, zur Interaktion und zur Mausbewegung geladen werden. Voraussetzung dabei ist, dass die Dateien einem bestimmten Format entsprechen. Tun sie das nicht, führt das in der Regel zum Abbruch des Prozesses unter entsprechender Fehlermeldung. In Ausnahmefällen kann es passieren, dass die Daten trotzdem korrekt eingelesen werden. Es obliegt dem Anwender selbst, für korrekte Quelldateien zu sorgen. Eine umfassende Fehlerbehandlung durch das Programm ist natürlich sinnvoll, liegt aber außerhalb des Rahmens dieser Arbeit.

Die Eye-Tracking-Daten liegen in einer *.tsv-Datei (*Tab-Separated Values*) vor und müssen durch einmaliges Öffnen und anschließendes Speichern als *.xlsx-Datei (*Excel Spreadsheet*) mit Microsoft Office Excel konvertiert werden. Ist dieser Schritt erfolgt, liegen sämtliche Daten (für die Augenbewegung, die Interaktion und die Mausbewegung) als *Excel Spreadsheet* vor.

Für jeden Datentyp existiert eine statische Funktion zum Einlesen der Daten aus einem *Excel Spreadsheet*. Diese Funktionen orientieren sich an bestimmten Zellen in der Datei und lesen dann zeilenweise die Daten ein. Es wird beispielsweise im Voraus nach dem tatsächlichen Start und dem Ende der Aufzeichnung gesucht, um das Einlesen auf die relevanten Zeilen zu beschränken. Mit Kenntnis der Logdatei können Start und Ende anhand identifizierender Einträge festgestellt werden. Dadurch werden bereits in diesem Schritt Protokolleinträge gefiltert, die vor dem Beginn oder nach dem Ende der eigentlichen Aufzeichnung liegen. Des Weiteren werden Einträge übersprungen, die als fehlerhaft oder unvollständig erkannt werden.

Die nachfolgenden Tabellen listen für jeden Datentyp die Spalten und deren Bedeutung auf, die beim Import der Daten berücksichtigt werden.

Spalte (Eye-Tracking-Daten)	Bedeutung
DateTimeStamp	Der Zeitstempel der Systemzeit für diesen Eintrag
FixationIndex	Die ID der zugehörigen Fixation
MappedFixationPointX	Die X-Koordinate der zugehörigen Fixation
MappedFixationPointY	Die Y-Koordinate der zugehörigen Fixation
FixationDuration	Die Dauer der Fixation

Tabelle 2: Beim Einlesen der Eye-Tracking-Daten verwendete Spalten der Quelldatei und deren Bedeutung

Für jede Fixation werden ID (FixationIndex), Zeitstempel (DateTimeStamp), Koordinaten (MappedFixationPointX, MappedFixationPointY) und Dauer (FixationDuration) eingelesen.

Die in *Tabelle 2: Beim Einlesen der Eye-Tracking-Daten verwendete Spalten der Quelldatei und deren Bedeutung* aufgeführten Daten werden für die Einträge der Fixationen verwendet (siehe Kapitel 6.3 *Datenstruktur*). Ein Eintrag für eine Fixation enthält deren ID, den Zeitstempel, X- und Y-Koordinaten und Dauer (siehe *Abbildung 32: Aufbau des Eintrags für eine Fixation*).

Weil der Eyetracker die Fixationen bereits aus den Daten der Augenbewegung berechnet hat, können diese wie beschrieben direkt aus der Quelldatei entnommen werden.

Für die Interaktionsdaten werden aus den Daten in *Tabelle 3: Beim Einlesen der Interaktionsdaten verwendete Spalten der Quelldatei und deren Bedeutung* zwei verschiedene Typen von Einträgen erzeugt. Zum einen wird zu jeder eingelesenen Zeile ein neuer Interaktions-Eintrag generiert und zum anderen wird für jeden neuen Stimulus ein entsprechender Eintrag erstellt. Beide Einträge verfügen dabei über den Zeitstempel und eine Beschreibung der ausgeführten Interaktion. Handelt es sich um eine Selektion, wird zusätzlich das davon betroffene Element innerhalb einer AOI mit angegeben.

Jede Interaktion wird über eine entsprechende ID mit dem aus der Interaktion entstandenen Stimulus verknüpft. Darüber hinaus werden die X- und Y-Koordinaten der Position des Mauszeigers und die davon betroffene AOI festgehalten. Für einen Stimulus-Eintrag wird hinterlegt, wie lange dieser Stimulus gültig war. Außerdem enthält der Eintrag alle AOI, die im Stimulus vorhanden sind. Die AOI der obersten Ebene sind hiervon ausgenommen, weil sie für alle Stimuli gleich sind und deshalb nur einmalig ausgelesen werden (siehe *Abbildung 34: Aufbau der Einträge für Interaktion, Stimulus und AOI*).

Spalte (Interaktionsdaten)	Bedeutung
DateTimeStamp	Der Zeitstempel der Systemzeit für diesen Eintrag
MouseEvent	Beschreibung der Interaktion
MousePointX	Die X-Koordinate der Position des Mauszeigers
MousePointY	Die Y-Koordinate der Position des Mauszeigers
LevelName	Die Ebene der betroffenen AOI
Description	Die ID der betroffenen AOI
SelectedWord	Das in der betroffenen AOI selektierte Wort, falls vorhanden
RectData	Die Koordinaten der betroffenen AOI
StimulusData*	Die Daten aller sichtbaren AOI

Tabelle 3: Beim Einlesen der Interaktionsdaten verwendete Spalten der Quelldatei und deren Bedeutung

Für jede Interaktion werden Interaktionstyp (MouseEvent), Zeitstempel (DateTimeStamp), Koordinaten (MousePointX, MousePointY), Bezeichnung (LevelName, Description) und geometrische Eigenschaften (RectData) der betroffenen AOI sowie das Objektmodell des entstandenen Zustands des Stimulus (StimulusData) eingelesen. Handelt es sich bei der Interaktion um eine Selektion, wird außerdem das selektierte Element (SelectedWord) gespeichert.

Wichtig ist, dass nicht durch jede Interaktion direkt ein neuer Stimulus entsteht. Die Logdatei enthält zum Beispiel jeweils einen Low-Level-Eintrag für den Start einer Scroll-Interaktion und das Ende dieser Interaktion. Weil nur der Stimulus nach Abschluss der Interaktion relevant ist, werden beide Interaktions-Einträge zu einem einzigen Eintrag für einen neuen Stimulus zusammengefasst.

Spalte (Mouse-Tracking-Daten)	Bedeutung
DateTimeStamp	Der Zeitstempel der Systemzeit für diesen Eintrag
MousePointX	Die X-Koordinate der Position des Mauszeigers
MousePointY	Die Y-Koordinate der Position des Mauszeigers
LevelName	Die Ebene der betroffenen AOI
Description	Die ID der betroffenen AOI
RectData	Die Koordinaten der betroffenen AOI

Tabelle 4: Beim Einlesen der Daten zur Mausbewegung verwendete Spalten der Quelldatei und deren Bedeutung

Für jede Fixation mit der Maus werden Zeitstempel (DateTimeStamp), Koordinaten (MousePointX, MousePointY) und die Bezeichnung (LevelName, Description) sowie die geometrischen Eigenschaften (RectData) der betroffenen AOI eingelesen.

Auch die Daten zur Mausbewegung werden in zwei verschiedene Sorten von Einträgen unterteilt. Dazu werden die in *Tabelle 4: Beim Einlesen der Daten zur Mausbewegung verwendete Spalten der Quelldatei und deren Bedeutung* aufgeführten Attribute Zeitstempel, X-Koordinate und Y-Koordinate zusammen mit einer ID für die zugeordnete Fixierung als Eintrag für eine Mausbewegung zusammengelegt. Der Eintrag zur Mausfixierung selbst enthält neben einer ID und dem Zeitstempel auch die beim Einlesen ermittelte Dauer der Fixierung und die Daten der betroffenen AOI.

Beim Einlesen der Daten zur Mausbewegung wird für jeden neuen Eintrag geprüft, ob die ausgelesene AOI mit der AOI des vorhergehenden Eintrags übereinstimmt. Ist das nicht der Fall, wird ein neuer Eintrag für eine Mausfixierung erstellt und mit dem neuen Eintrag zur Mausbewegung verknüpft. Stimmen die AOI überein wird die Dauer der aktuellen Mausfixierung auf die Differenz zwischen den Zeitstempeln gesetzt.

6.5 Verwendung der Daten

Für die sinnvolle Anzeige der Daten ist es erforderlich, die eingelesenen Daten aufeinander abzustimmen. Deshalb ist es beispielsweise für das Anzeigen von Daten zur Augenbewegung unerlässlich, bereits Interaktionsdaten eingelesen zu haben. Dabei geht es vor allem um die Synchronisation der Daten bezüglich der Zeit. Weil die Logdateien für die Interaktion und für das Eye-Tracking von unterschiedlichen Programmen erstellt wurden, muss dieser Abgleich im Nachhinein erfolgen. Für diese Arbeit kommen die Eye-Tracking Daten vom Eyetracker selbst, während die Interaktionsdaten von *VarifocalReader* protokolliert werden. Dafür werden beim Protokollieren alle Zeitstempel an der Systemzeit ausgerichtet und bei der internen Verarbeitung in Millisekunden umgewandelt.

Insbesondere muss der Stimulus zu jeder Fixation ermittelt werden, um die AOI zu bestimmen, die vom Probanden betrachtet wurde. Weil es sich um einen interaktiven Stimulus handelt, konnte der Eyetracker die AOI während der Aufzeichnung nicht kennen. Die AOI sind während der Aufnahme also nur *VarifocalReader* bekannt gewesen (siehe Kapitel 5 *Lösungskonzept*). Weil die Daten zur Mausbewegung ebenfalls von *VarifocalReader* stammen, entfällt dieser Schritt für diese.

Weil die Fixationen bereits vom Eyetracker bestimmt werden und die Mausfixierungen erst beim Einlesen der Daten entstehen, unterscheiden sich beide in ihrer Bedeutung. Eine Fixation mit dem Auge wird anhand der Distanz verschiedener Blickpunkte ermittelt. Zudem ist ein Grenzwert für die Dauer der Fixation festgelegt. Für eine Fixierung der Maus auf eine AOI wird lediglich geprüft, ob und wie lange sich die Maus dort befindet.

Aus diesem Grund ist die durch eine einzelne Mausfixierung erfasste Fläche sehr viel größer, als die von einer Fixation mit dem Auge betroffene. Die erfasste Fläche für eine Mausfixierung gleicht der Fläche der umliegenden AOI, in der der Mauszeiger verweilt, bis er auf eine andere AOI bewegt wird. Der Mangel eines Grenzwerts für die Abweichung der Position des Mauszeigers ist der Grund dafür. Hinzu kommt, dass die Augen ständig in Bewegung sind, während der Proband die Maus durchaus für einen beliebigen Zeitraum unberührt lassen kann. Diese Abweichung muss bei der Visualisierung insofern berücksichtigt werden, als beide Datentypen für den Anwender gut erkennbar sein sollen. Das heißt, es muss eine Skalierung ge-

funden werden, die nicht nur die Dauer der Fixation beziehungsweise Fixierung in Relation zur angezeigten Zeitspanne verwendet, sondern auch die oben genannten Faktoren.

Darüber hinaus richten sich die angezeigten Daten stets nach der aktuell ausgewählten Zeitspanne der Interaktionsdaten, die wiederum von den gewählten Einstellungen für die Visualisierung abhängig ist.

Um diese Faktoren in konkreten Zahlen abzubilden, wurde die Tatsache ausgenutzt, dass alle AOI im Stimulus die gleiche Breite aufweisen. Dividiert man diese durch den vom Eyetracker verwendeten, konstanten Grenzwert für den Abstand bei der Berechnung einer Fixation, erhält man einen Wert, der das beschriebene Größenverhältnis widerspiegelt. Der in *Inseyet* verwendete Wert ist 9,4. Das bedeutet: Ohne entsprechende Behandlung werden die Daten für die Mausbewegung fast zehn Mal größer dargestellt, als vergleichbare Daten für die Augenbewegung. Dieser Effekt wird durch Interaktionen wie zum Beispiel das Zoomen noch verstärkt.

6.6 Benutzungsschnittstelle und Interaktion

In diesem Abschnitt werden die Interaktionsmöglichkeiten der Benutzungsschnittstelle für den Anwender beschrieben. Auch wenn einzelne Funktionen bereits erklärt wurden (siehe Abschnitt *Die Einstellungen* in Kapitel 6.1.2 *Bereiche des Programmfensters*) werden sie hier aus einem zweckgebundenen Blickwinkel betrachtet. Entsprechend des *Visual Information Seeking Mantra* ist es möglich, sich durch den Graphen zuerst einen Überblick zu verschaffen, danach bestimmte Bereiche genauer zu betrachten und abschließend relevante Informationen zu diesem Bereich abzufragen (siehe Kapitel 2.2.4 *Visual Information Seeking Mantra*). Dieses Vorgehen wird nachfolgend beschrieben. Es wird vorausgesetzt, dass bereits Daten eingelesen wurden und im Programm zur Verfügung stehen.

6.6.1 Overview First

Durch die vorhandenen Reiter entsteht bereits ein erster Eindruck über den Umfang der vorhandenen Fragen. Auch die Probandenliste lässt die Anzahl der Probanden und sogar die Verfügbarkeit von Daten zu diesen erschließen. Trotzdem besteht die eigentliche Übersicht vor allem aus dem Graphen selbst.

Durch die Selektion mehrerer oder aller Probanden und der anschließenden Visualisierung erhält der Anwender einen Überblick über die Daten der ausgewählten Probanden. Ausreißer fallen dabei zum Beispiel durch den im Graphen ungenutzten Platz auf. Der Anwender kann sofort erkennen, ob für alle Probanden valide Daten vorliegen oder noch geladen werden müssen.

6.6.2 Zoom & Filter

Es ist möglich, einzelne Probanden entweder durch die Deselektion in der Liste oder durch einen Klick auf den entsprechenden Button im Graphen selbst auszublenden. Das reduziert die angezeigten Daten auf die vom Anwender als relevant eingestuft. Auf ähnliche Weise kann der Anwender die Anzeige der Daten auf die eines einzelnen Probanden beschränken: Das geschieht entweder durch Deselektion aller anderen Probanden oder durch einen einzigen Klick auf den Button des Probanden im Graphen.

Des Weiteren kann der Anwender entscheiden, welche Datentypen er angezeigt haben möchte. So kann er unter anderem die Anzeige der Daten zur Mausbewegung für einzelne oder für

alle Probanden unterbinden. Gleiches gilt für die Daten zur Augenbewegung und für die Daten zur Interaktion.

Darüber hinaus ist es möglich, nur einen Teil der vorhandenen Daten aller Probanden anzuzeigen. Welcher Bereich das sein soll oder wie groß dieser ist, kann der Anwender mit den entsprechenden Schiebereglern festlegen. Das Resultat ist die Anzeige weniger Daten im Graphen unter Ausschöpfung des gesamten Platzes. Auf diese Weise werden beispielsweise Fixationen zu Tage gefördert, die bei der Anzeige aller Daten nicht zu sehen waren. Das liegt daran, dass die berechnete Größe zu klein für eine Anzeige auf dem Monitor war. Die Größe entspricht der Bedeutung einzelner Einträge, die für kürzere Interaktionsausschnitte zunimmt. Die Dauer eines Elements hat einen größeren Anteil an einem kürzeren Ausschnitt als an der gesamten Interaktionszeitspanne.

6.6.3 Details on Demand

Hat der Anwender die Daten dann auf die für ihn interessanten beschränkt, kann er sich verschiedene Informationen beschaffen. Das geschieht entweder über einzelne Tooltips oder durch die Selektion von Elementen und dem damit verbundenen *Brushing & Linking*. Welche Tooltips dabei welche Informationen bereitstellen, ist nachfolgend aufgeführt. Ein Tooltip erscheint immer dann, wenn der Mauscursor für kurze Zeit über dem jeweiligen Element verweilt.

Participant 0	
Visible From:	00:09:652
Visible To:	03:02:640
Visible Duration:	02:52:988
Total Duration:	03:33:288
Global Duration:	03:45:715

Abbildung 36: Tooltip für einen Probanden im Graphen

Der Tooltip für einen Probanden gibt Auskunft über den für ihn angezeigten Interaktionszeitraum (Visible Duration). Dieser berechnet sich aus den gewählten Zeitpunkten für Start (Visible From) und Ende (Visible To). Dieser Zeitraum ist stets ein Ausschnitt des erfassten Interaktionszeitraums (Total Duration) der innerhalb des längsten Interaktionszeitraums aller Probanden (Global Duration) liegt.

Der Tooltip für einen Probanden enthält stets dessen vollständige Bezeichnung, damit dieser identifiziert werden kann. Darunter befinden sich die Informationen zur aktuellen Darstellung im Graphen. Diese bestehen unter anderem aus den Werten für Beginn, Ende und Dauer der angezeigten Zeitspanne für diesen Probanden. Zusätzlich sind die erfasste Interaktionsdauer für diesen Probanden und die längste der insgesamt erfassten Interaktionszeiträume angegeben. Beträgt der aktuelle Zoomfaktor 0%, sind die Werte für die Dauer und das Ende der angezeigten Zeitspanne sowie für die erfasste Zeitspanne identisch. Der Wert für die längste Zeitspanne wird nur angezeigt, wenn die längste Interaktionszeitspanne aller Probanden für die Darstellung der Elemente verwendet wird. Er gibt dann diesen Zeitraum an.

Für jeden Probanden wurde der damit verknüpfte Button mit einem wie in *Abbildung 36: Tooltip für einen Probanden im Graphen* gezeigten Tooltip versehen. Darin sind alle Daten über den angezeigten Ausschnitt aus den Daten des Probanden wie Startzeitpunkt, Ende und Dauer enthalten.

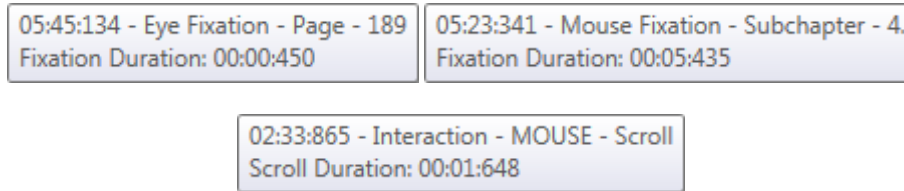


Abbildung 37: Tooltips für die verschiedenen Elemente im Graphen

Der Tooltip oben links beschreibt eine Fixation mit dem Auge (Eye Fixation), das rechts daneben eine Fixierung mit der Maus (Mouse Fixation) und das unterste eine Interaktion (Interaction). Alle drei bestehen aus einem Zeitstempel, einer Beschreibung des Typs des zugehörigen Elements, der Ebene der betroffenen AOI und einer identifizierenden Beschreibung dieser. Darüber hinaus wird die Dauer der Fixation beziehungsweise Interaktion angegeben.

Für jedes farbige Element im Graphen werden Informationen zu diesem in einem Tooltip bereitgestellt. *Abbildung 37: Tooltips für die verschiedenen Elemente im Graphen* zeigt eine Auswahl von Tooltips für die wichtigsten Elemente im Graphen. Sie gleichen sich im Aufbau und enthalten neben Zeitangaben auch Informationen bezüglich der von den repräsentierten Daten betroffenen AOI. Dazu gehören der Zeitstempel, der Datentyp, die Bezeichnung des Typs der AOI und der AOI selbst sowie die Dauer der Fixation beziehungsweise der Interaktion.

Weitere Informationen können dem Interaktionsprotokoll entnommen werden. Dieses wurde so mit dem Graphen verknüpft, dass es stets die Interaktionen enthält, die im Graphen dargestellt werden. Selektiert man ein Element im Graphen (mit der sekundären Maustaste), wird sowohl das Element als auch der damit verlinkte Eintrag im Interaktionsprotokoll durch dunkelblaue Einfärbung hervorgehoben. Diese Verbindung besteht in beide Richtungen: Selektiert man einen Eintrag im Protokoll, wird das entsprechende Element im Graphen ebenfalls hervorgehoben. Dadurch ist es möglich, die zeitliche Abfolge der Einträge im Protokoll den Elementen im Graphen zuzuordnen. Das Beschriebene gilt dabei auch für die Mehrfach- und die Deselektion.

Für jeden Probanden kann außerdem über den entsprechenden Button (siehe Abschnitt *Die Probandenliste* in Kapitel 6.1.2 *Bereiche des Programmfensters*) das zugehörige Video in einem eigenen Fenster abgespielt werden, sofern die Videodatei hinterlegt wurde. Das ermöglicht dem Anwender, sich die Videoaufzeichnung der Interaktion an den Stellen anzuschauen, die er durch die Analyse der Daten als interessant eingestuft hat. In diesem Video sind sowohl die Mausposition als auch die Fixationen des Probanden zu sehen. Außerdem ermöglicht die Wiedergabe der Aufzeichnung dem Anwender, sich den Monolog des Probanden (*Think-Aloud-Protokoll*) beziehungsweise den Dialog zwischen Studienleiter und Proband an einer beliebigen Stelle in der Aufzeichnung anzuhören.

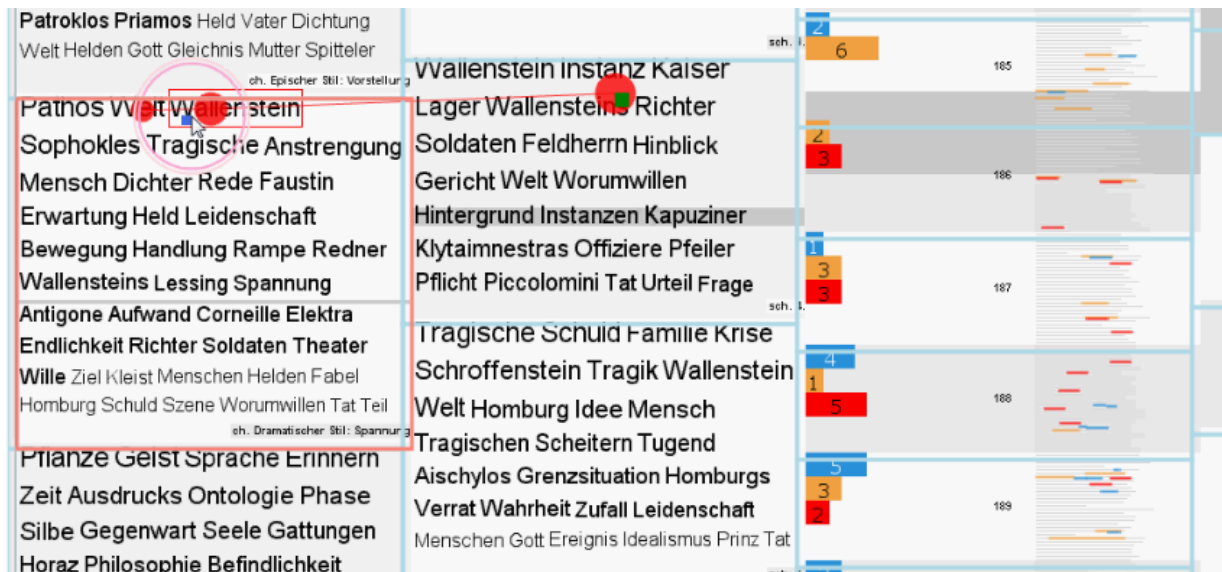


Abbildung 38: Ausschnitt aus einem Snapshot einer Videoaufzeichnung

Im gezeigten Ausschnitt eines Einzelbilds aus der Videoaufzeichnung eines Probanden sind die erfassten Daten eingezeichnet. Dazu gehören die Position des Mauszeigers (dunkelblaues Quadrat), die aktuelle Fixation (grünes Quadrat), das Objektmodell (hellblaues Rechtecke) sowie die mit der Maus fokussierte AOI (orangefarbenes Rechteck). Der Rest des Bilds (Stimulus, rote Kreisflächen für Fixationen) wurde dem vom Eyetracker bearbeiteten Video entnommen.

Um sich genau den Zustand des Stimulus anzuschauen, den der Proband zu einem bestimmten Zeitpunkt angezeigt bekommen hat, kann der Anwender auf jedes farbige Element im Graphen klicken. Sofern eine Videodatei vorhanden ist, wird das Bild aus dem Video extrahiert und angezeigt, das dem Zeitstempel des angeklickten Elements entspricht (siehe *Abbildung 38: Ausschnitt aus einem Snapshot einer Videoaufzeichnung*). In diesem Bild werden dann alle vorhandenen Daten für diesen Zeitpunkt eingezeichnet. Sind Daten zur Mausbewegung vorhanden, wird die Position des Mauszeigers für diesen Zeitpunkt mit einem dunkelblauen Quadrat gekennzeichnet. Gleiches gilt für die Fixation aus den Eye-Tracking-Daten (grünes Quadrat). Aus den Interaktionsdaten wird der gültige Zustand des Stimulus ausgelesen, um die sichtbaren AOI mit hellblauen Linien einzurahmen.

Wie in *Abbildung 38: Ausschnitt aus einem Snapshot einer Videoaufzeichnung* zu erkennen ist, kann der Anwender mithilfe des Snapshots feststellen, welche Teile des Stimulus als AOI definiert wurden. Des Weiteren lässt sich die Güte der protokollierten Informationen herausfinden. In *Abbildung 38: Ausschnitt aus einem Snapshot einer Videoaufzeichnung* kann zum Beispiel festgestellt werden, dass die Definition der AOI minimal von den vorhandenen Blöcken abweicht.

6.7 Daten zur Entwicklung von Inseyet

Die nachfolgenden Werte wurden mit den von Microsoft Visual Studio 2013 bereitgestellten Analysewerkzeugen ermittelt. Es sind vor allem Angaben zum Umfang des Programmcodes enthalten, weil andere Metriken für das entwickelte Programm – wegen des geringen Umfangs – meines Erachtens nur bedingt geeignet sind. Beispielsweise unterscheiden sich die Komplexität einzelner Klassen und die damit verbundene Wartbarkeit erheblich. Gerade für

den Datenimport und das Zeichnen der Elemente im Graphen sind Verschachtelungen von Schleifen und Abfragen unvermeidlich – diese wurden auf ein Minimum reduziert. Die genaue Kenntnis der geometrischen Daten einzelner Elemente oder der Struktur der importierten Dateien kann nur schwer im Code festgehalten werden.

Inseyet umfasst ca. 2100 ausführbare Codezeilen. Etwas mehr als diese Anzahl an Zeilen kommt durch Formatierungen und Kommentare hinzu (also insgesamt ca. 5000 Zeilen). Nicht enthalten sind außerdem die für das Layout, für die Verknüpfung von View und ViewModel, die für die Positionierung der Elemente und die für das Erscheinungsbild verantwortlichen Codezeilen (etwa 750 Zeilen) in den *.xaml-Dateien.

Von den 2100 ausführbaren Zeilen Code werden 600 für den Graphen insgesamt und davon 500 für die Darstellung der Informationen alleine verwendet. Das entspricht 28,5% (600 Zeilen) beziehungsweise 23,8% (500 Zeilen) des Gesamtvolumens. Weitere 450 ausführbare Programmzeilen (21,4%) fallen für die Datenhaltung, also für die interne Verwaltung von Daten zur Augenbewegung, Interaktion und Mausbewegung, an. Für das Einlesen dieser Daten und die Übertragung in die interne Datenstruktur werden 200 Zeilen (9,5%) benötigt.

Der Aufwand für die Implementierung betrug ca. 180 Stunden.

7 Demonstration des Prototyps

In diesem Kapitel wird unter anderem gezeigt, wie der bestehende Datensatz einer Studie mit dem in Kapitel 5 *Lösungskonzept* beschriebenen Programm *VarifocalReader* analysiert werden kann. Die Analyse basiert auf dem im selben Kapitel vorgestellten Konzept und erfolgt mithilfe meines in Kapitel 6 *Implementierung eines Prototyps* beschriebenen Prototyps. Zur Demonstration ist zudem eine Übersicht über die Beispieldaten für 30 Probanden abgebildet.

7.1 Darstellung von Daten als Übersicht

In *Abbildung 39: Darstellung von 30 Probanden mit Inseyet* ist dargestellt, wie ein Graph für die Daten von dreißig Probanden zu einer Aufgabe aussehen könnte.

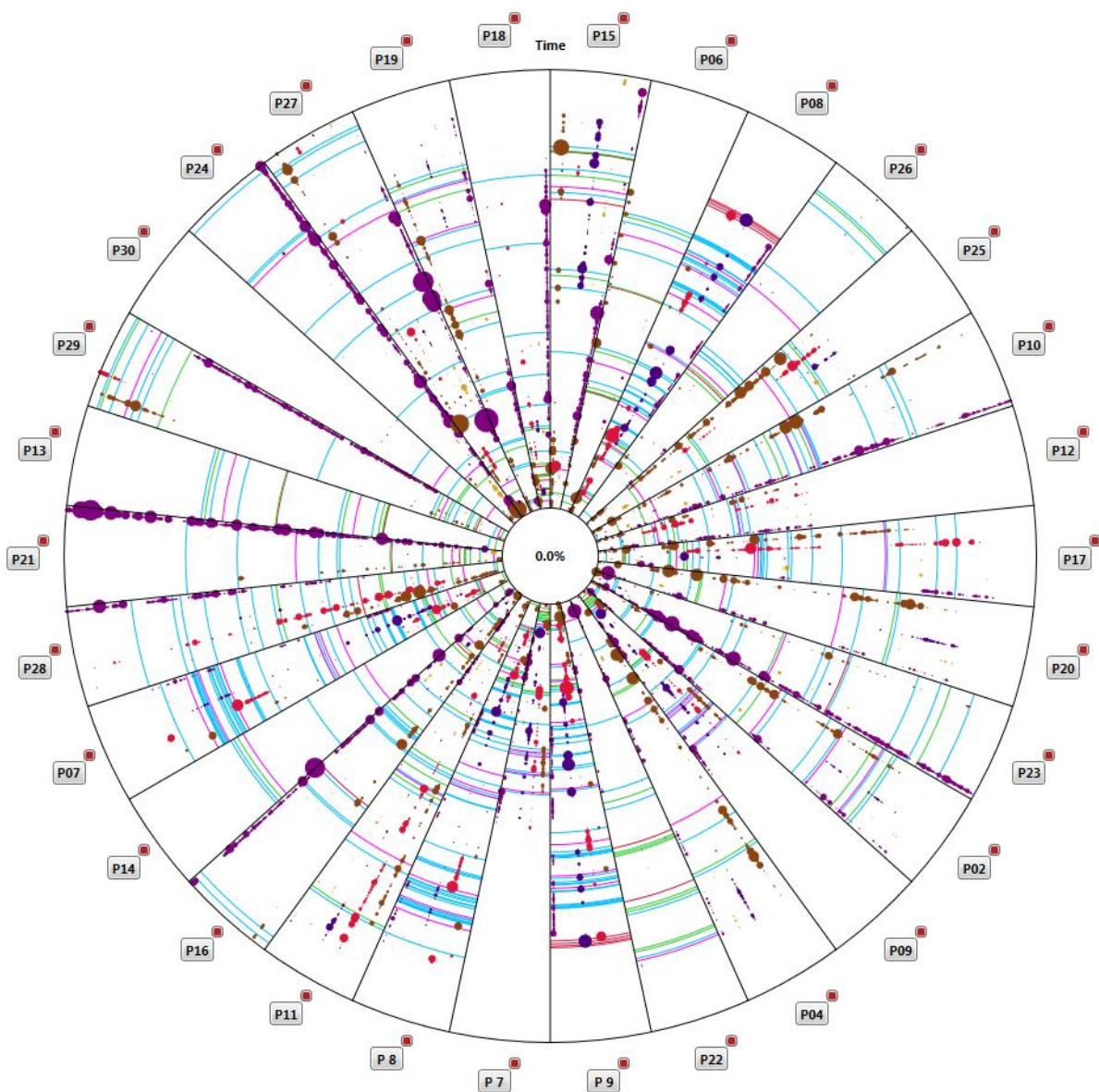


Abbildung 39: Darstellung von 30 Probanden mit Inseyet

Übersicht, wie sie für die Daten von 30 Probanden in einem Graphen aussehen könnte. Jedes Kreissegment entspricht der gleichen Zeitspanne.

Für den in *Abbildung 39: Darstellung von 30 Probanden mit Inseyet* gezeigten Graphen wurden, wegen des Mangels an einem solchen Datensatz, verschiedene Beispieldaten verwendet. Das bedeutet, dass diese Übersicht nicht geeignet ist, um Muster darin zu suchen oder die Daten einzelner Probanden miteinander zu vergleichen. Zweck dieser Abbildung ist, dass sich der Leser ein Bild davon machen kann, wie gut sich das Konzept und in Folge der Prototyp als Übersicht über die Daten für eine Vielzahl Probanden eignen.

7.2 Beispiel für eine Analyse

Die für die Analyse verwendeten Daten wurden bereits im Programm geladen und eine erste Visualisierung (siehe *Abbildung 40: Visualisierung eines Datensatzes von sechs Probanden mit Inseyet*) zeigt die vorhandenen Daten zu sechs Probanden. Die Größe des Graphen wurde mit dem obersten Schieberegler entsprechend der Bildschirmauflösung so angepasst, dass der vorhandene Platz genutzt wird. Die Daten wurden gemäß der zugehörigen Aufgabe auf verschiedene Reiter verteilt.

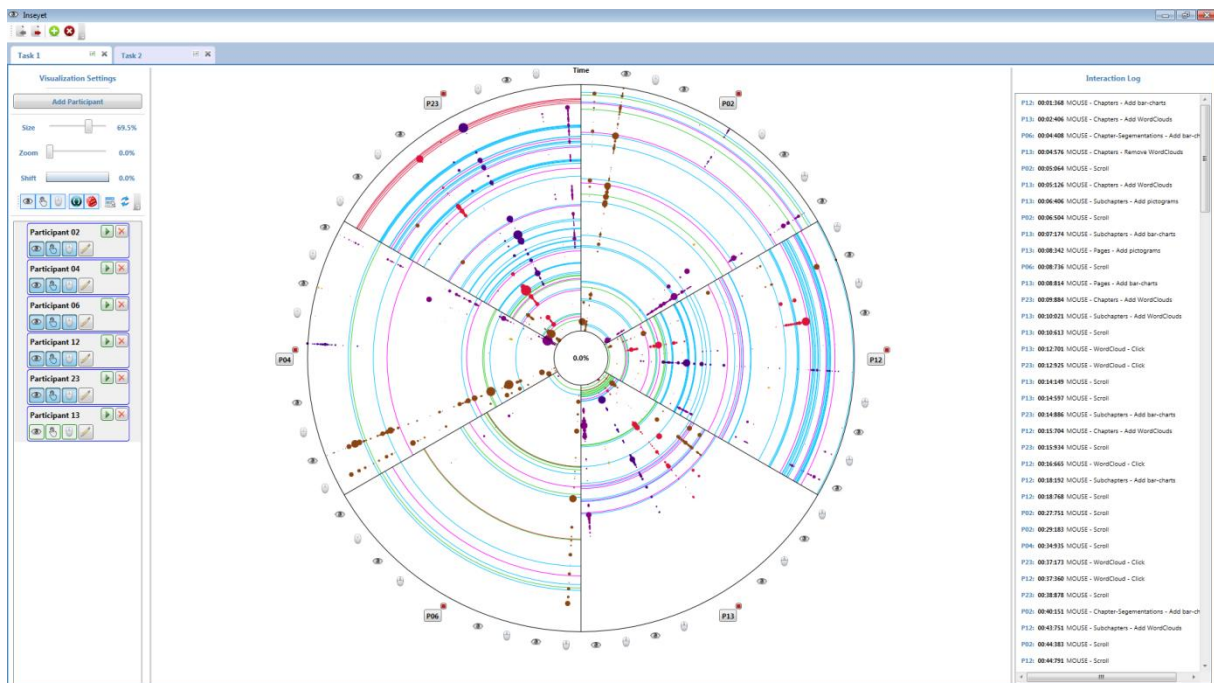


Abbildung 40: Visualisierung eines Datensatzes von sechs Probanden mit Inseyet

Das Hauptfenster von Inseyet. Der angezeigte Reiter enthält die Visualisierung der Daten von sechs Probanden.

Der in *Abbildung 40: Visualisierung eines Datensatzes von sechs Probanden mit Inseyet* dargestellte Graph dient als Übersicht über die eingelesenen Daten. Die Anordnung der Elemente im Graphen ist normiert. Das heißt, es wird für alle Probanden die tatsächlich verstrichene Zeit mit zunehmendem Abstand zum Kreismittelpunkt dargestellt. Die einzelnen Kreissegmente sind in jeweils fünf Abschnitte für die verschiedenen Typen von AOI unterteilt. Die Abschnitte repräsentieren in dieser Reihenfolge Kapitel, Unterkapitel, Seiten, Paragraphen und Originaltext. Die Information wird einerseits in Tooltips bereit gestellt und ist andererseits für den erfahrenen Anwender aus der Farbgebung der Elemente ersichtlich. Wichtig an dieser Stelle ist, dass sich die Farben aller AOI-Typen voneinander unterscheiden.

Ungeachtet dessen ist im gezeigten Graphen bereits auf den ersten Blick zu sehen, dass sich die Daten eines einzelnen Probanden (P13) bezüglich der Interaktionsdauer deutlich von denen der anderen unterscheiden. Der *Ausreißer* kann bei Bedarf aus der Darstellung entfernt werden.

Außerdem sieht es so aus, als ob sich die verbliebenen Daten anhand der gezeigten Interaktionshäufigkeit in zwei Gruppen unterteilen lassen. Die Datenrepräsentationen der beiden Probanden (P12, P23) im Graphen weisen eine deutlich höhere Anzahl und eine dadurch bedingte, dichtere Anordnung von Elementen auf, als die der anderen. Eine Sortierung der präsentierten Daten entsprechend dieser Eigenschaft verdeutlicht eine solche Einteilung, wie in *Abbildung 41: Visualisierung der Daten (sortiert) von fünf Probanden mit Inseyet* zu erkennen ist.

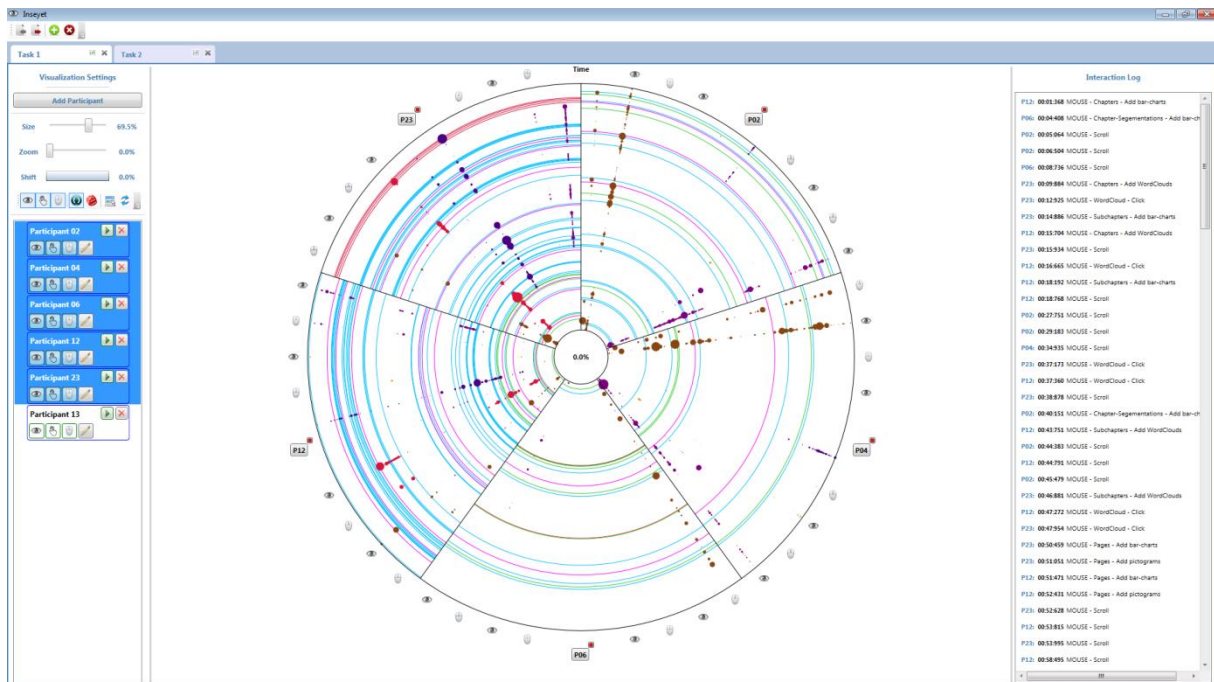


Abbildung 41: Visualisierung der Daten (sortiert) von fünf Probanden mit Inseyet

Eine gefilterte Präsentation vorhandener Daten. Proband (13) ist nicht in der Liste (links) selektiert und deshalb auch nicht im Graphen (mittig) zu sehen. Die Dichte der angezeigten Daten unterscheidet sich für zwei der Probanden (P12, P23) erheblich von der der anderen Probanden (P02, P04, P06).

Abbildung 41: Visualisierung der Daten (sortiert) von fünf Probanden mit Inseyet verdeutlicht, dass die Daten für drei der Probanden erheblich weniger Einträge für Interaktionen enthalten, als die der anderen beiden Probanden. Auch die Fixationen weisen darauf hin, dass die Informationsgewinnung in einer Gruppe mithilfe anderer AOI erfolgte, als es in der zweiten Gruppe der Fall war. Die Gründe dafür können vielfältig sein und lassen sich anhand der Visualisierung nur bedingt feststellen. Eine Erklärung für die unterschiedliche Interaktionsdichte könnte der Grad an Erfahrung sein, den ein Proband mit vergleichbaren Anwendungen hat.

An diesem Punkt kann es sinnvoll sein, die Daten weiter zu filtern und die beiden Gruppen separat voneinander zu betrachten. Um Parallelen zwischen den unterschiedlichen Gruppen

festzustellen, können die einzelnen Gruppen auch in verschiedenen Reitern beziehungsweise Programmfenstern dargestellt werden.

Für die Analyse soll zunächst die größere Gruppe genauer untersucht werden. Durch Selektieren der Einträge in der Liste der Probanden wird so gefiltert, dass nur noch die ausgewählten Daten angezeigt werden. Die verbliebenen Daten werden jetzt entsprechend der zugehörigen Probanden in drei Kreissegmente aufgeteilt. Auch das *Interaktionsprotokoll* wird, unter Berücksichtigung der selektierten Einträge in der *Liste der Probanden*, neu mit Einträgen angereichert.

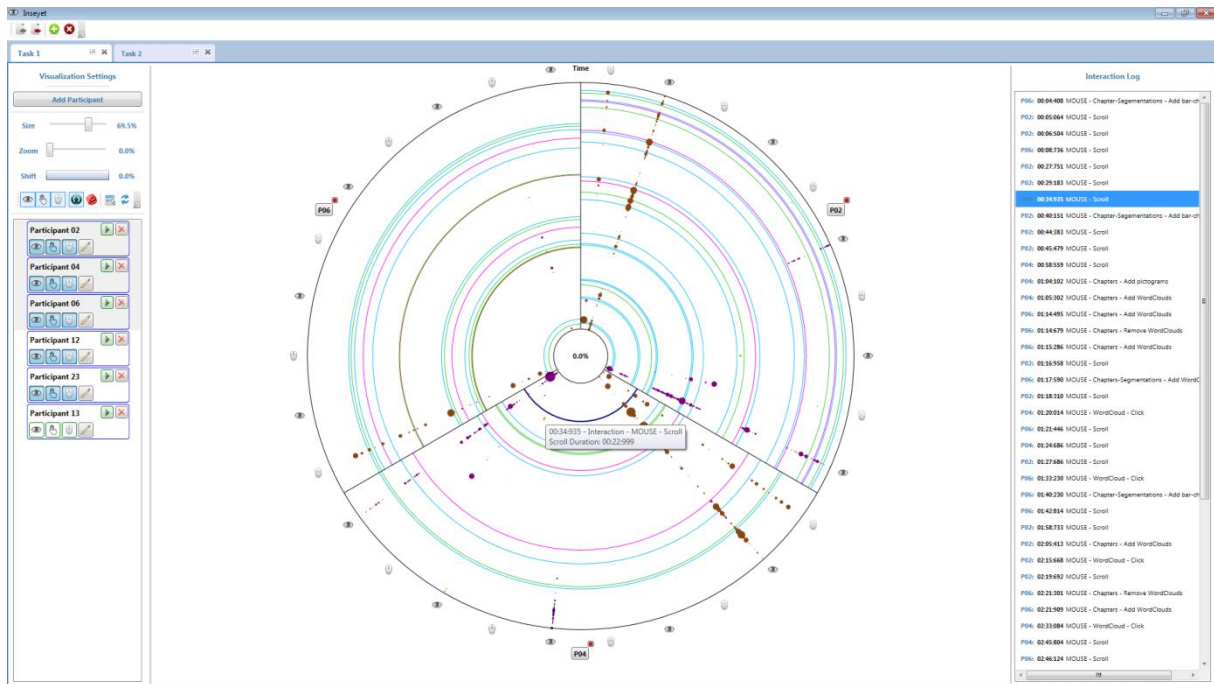


Abbildung 42: Visualisierung der Daten dreier Probanden mit Inseyet

Eine gefilterte Präsentation vorhandener Daten. Nur drei der in der Liste (links) aufgeführten Probanden wurden bei der Visualisierung berücksichtigt. Der Eintrag für eine Interaktion ist selektiert und das entsprechende Element im Graphen wurde farblich hervorgehoben. Zusätzlich ist das Tooltip zu diesem Element eingezeichnet.

Die entstandene Visualisierung ist in *Abbildung 42: Visualisierung der Daten dreier Probanden mit Inseyet* dargestellt. In dieser ist zu erkennen, dass die Verteilungen der Fixationen aller Probanden auf dem Stimulus ähnlich sind und sich im Wesentlichen auf den ersten (*Chapters*) und den letzten Typ (*Text*) von AOI beschränken. Es fällt außerdem auf, dass für einen Probanden (P06) keine Fixationen für die Augenbewegung angezeigt werden, obwohl die Einstellungen die Darstellung dieser Daten erlauben. Diese Einstellung kann nur vorgenommen werden, wenn erfolgreich Daten eingelesen wurden. Deshalb ist es wahrscheinlich, dass die eingelesenen Daten in irgendeiner Weise fehlerhaft sind und nicht sinnvoll ausgewertet werden können. Auf Grund der technischen Unabhängigkeit der Daten voneinander, kann davon ausgegangen werden, dass die Daten zur Interaktion und zur Mausbewegung für denselben Probanden davon nicht betroffen sind. Diese Daten können für die weitere Analyse verwendet werden.

Die angezeigten Interaktionen lassen erkennen, dass einer der Probanden (P02) zu Beginn der Bearbeitung signifikant mehr Scroll-Interaktionen durchgeführt hat, als die anderen beiden. Dieser Sachverhalt ist auch im *Interaktionsprotokoll* zu erkennen (siehe *Abbildung 43: Interaktionsprotokoll mit Einträgen zu drei Probanden*). Für eine Interpretation dieser Auffälligkeit kann mithilfe von Tooltips auf die Details der jeweiligen Interaktionen zugegriffen werden.

Interaction Log		
P06:	00:04:408	MOUSE - Chapter-Segementations - Add bar-ch
P02:	00:05:064	MOUSE - Scroll
P02:	00:06:504	MOUSE - Scroll
P06:	00:08:736	MOUSE - Scroll
P02:	00:27:751	MOUSE - Scroll
P02:	00:29:183	MOUSE - Scroll
P04:	00:34:935	MOUSE - Scroll
P02:	00:40:151	MOUSE - Chapter-Segementations - Add bar-ch
P02:	00:44:383	MOUSE - Scroll
P02:	00:45:479	MOUSE - Scroll
P04:	00:58:559	MOUSE - Scroll
P04:	01:04:102	MOUSE - Chapters - Add pictograms
P04:	01:05:302	MOUSE - Chapters - Add WordClouds
P06:	01:14:495	MOUSE - Chapters - Add WordClouds
P06:	01:14:679	MOUSE - Chapters - Remove WordClouds
P06:	01:15:286	MOUSE - Chapters - Add WordClouds
P02:	01:16:958	MOUSE - Scroll
P06:	01:17:590	MOUSE - Chapters-Segementations - Add WordC
P02:	01:18:310	MOUSE - Scroll
P04:	01:20:014	MOUSE - WordCloud - Click
P06:	01:21:446	MOUSE - Scroll
P04:	01:24:686	MOUSE - Scroll
P02:	01:27:686	MOUSE - Scroll

Abbildung 43: Interaktionsprotokoll mit Einträgen zu drei Probanden

Das Interaktionsprotokoll für die ausgewählten Probanden. Die Einträge für die Scroll-Interaktionen von P02 sind in der Liste hervorgehoben. Die Zeitstempel lassen erkennen, dass der Proband P02 häufig in kurzen Zeitabständen je zweimal hintereinander gescrollt hat.

Für jede Scroll-Interaktion gibt der zugehörige Tooltip Auskunft über die Dauer der Interaktion (siehe *Abbildung 44: Vergleich der Scroll-Interaktionen mithilfe von Tooltips*). Konkret lässt sich diese Dauer so interpretieren, dass der Proband die sekundäre Maustaste zum Start der Interaktion gedrückt und für die gesamte Scroll-Dauer in gedrückter Position gehalten hat. Welche AOI er beim Scrollen durchlaufen hat, kann durch Interpretation der Mausbewegung näher bestimmt werden.

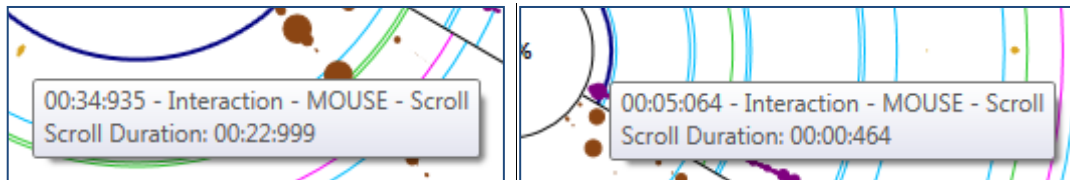


Abbildung 44: Vergleich der Scroll-Interaktionen mithilfe von Tooltips

Die Tooltips der jeweils ersten Interaktion zweier Probanden „P04“ (links) und „P02“ (rechts).

Das Auslesen der Daten aus den Tooltips gibt Aufschluss über das Scroll-Verhalten der einzelnen Probanden: Während zwei der Probanden jeweils lang andauernde Scroll-Interaktionen (10 - 50 Sekunden) vorgenommen haben, hat einer der Probanden vor jedem längeren (15 - 30 Sekunden) Scrollen einmal relativ kurz (~0,5 Sekunden) gescrollt.

Es ist davon auszugehen, dass die Probanden während einer über mehrere Sekunden andauernden Scroll-Bewegung innegehalten haben, um den entstandenen Stimulus zu begutachten. Grund für diese Annahme ist, dass eine durchgängige Scroll-Bewegung auf Kapitelebene (diese Auskunft liefern die Daten zur Mausbewegung), über den gesamten Inhalt, deutlich weniger Zeit beansprucht. Allerdings haben sie die für die Scroll-Interaktion verwendete Maustaste nicht losgelassen, wodurch die Interaktion nicht unterbrochen und damit auch kein neuer Eintrag protokolliert wurde. Anhand der Interaktionseinträge ist außerdem nicht ersichtlich, in welche Richtung gescrollt wurde. Eine Analyse der Mausfixationen im – durch Startzeitpunkt und Dauer der Interaktion – festgelegten Zeitraum ermöglicht derartige Einblicke.

Eine Alternative dazu bietet das Abspielen der Videoaufzeichnung für den ermittelten Zeitraum. Allerdings ist es nicht Ziel der Visualisierung, sich dermaßen auf die Aufzeichnungen aller Probanden zu stützen. Dieser Zwiespalt ist ein Hinweis darauf, dass sich die Visualisierung nur bedingt für die vom *VarifocalReader* aufgezeichneten Daten zu angebotenen Scroll-Interaktion eignet. Eine Anpassung des Verhaltens beim Protokollieren der Interaktionsdaten in dem Sinne, dass eine lang anhaltende Scroll-Interaktion als mehrere, kurze Interaktionen verstanden und aufgezeichnet wird, ist eine Lösungsmöglichkeit. Mit dieser Lösung könnte das vorgeschlagene Konzept weiter unverändert verwendet werden. Zudem wäre in der Visualisierung durch dicht aneinandergereihte Elemente für die Interaktion direkt die Dauer einer solch umfassenden Scroll-Interaktion ersichtlich.

Möchte man sich bei der Analyse auf die Daten zu Augen- und Mausbewegungen der Probanden konzentrieren, hilft das Ausblenden der Elemente für die Interaktionsdaten, das *Visual Clutter* zu reduzieren. Eine solche Visualisierung ist in *Abbildung 45: Visualisierung ohne Interaktionsdaten* dargestellt. Ohne die Anzeige von Interaktionsdaten im Graphen enthält auch das *Interaktionsprotokoll* keine Einträge.

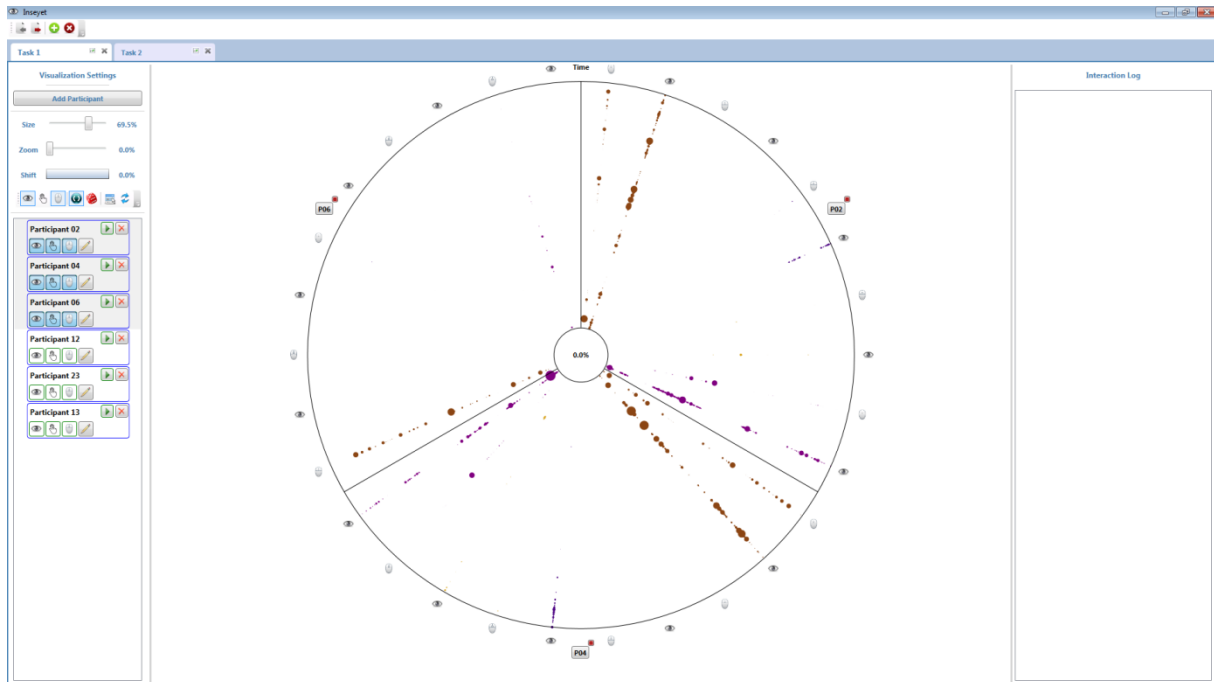


Abbildung 45: Visualisierung ohne Interaktionsdaten

Die Visualisierung der Daten zur Augen- und zur Mausbewegung dreier Probanden. Die Interaktionsdaten wurden nicht visualisiert und sind folglich auch nicht im Interaktionsprotokoll (rechts) aufgeführt.

Bei genauer Betrachtung des in *Abbildung 45: Visualisierung ohne Interaktionsdaten* gezeigten Graphen entdeckt man neben den Elementen für Fixationen in den Farben Braun und Lila auch orangefarbene Elemente für zwei der Probanden (P02, P04). Die dadurch präsentierten Fixationen sind allerdings ausschließlich den Augen zuzuordnen. Um diese Anomalie genauer untersuchen zu können, kann der betroffene Abschnitt des Interaktionszeitraums für alle Probanden vergrößert dargestellt werden.

Auch nach der Vergrößerung können nahezu alle dargestellten Elemente entweder den Kapitel-AOI (braun) oder den Text-AOI (lila) zugeordnet werden (siehe *Abbildung 46: Visualisierung mit Zoom, Zeitversatz und ohne Interaktionsprotokoll*). Die dazwischen liegenden Typen von AOI (Unterkapitel, Seiten und Paragraphen) wurden während der gesamten Bearbeitung von allen Probanden praktisch ignoriert. Einzige Ausnahme bilden die in der Abbildung gekennzeichneten Fixationen. Die Tooltips geben nur Auskunft über die Dauer der Fixation und den Typ der betrachteten AOI (Paragraph), nicht jedoch darüber, warum diese Abweichung bei beiden Probanden vorhanden ist. Ob sie für den dritten Probanden auch zutrifft, lässt sich auf Grund des Datenmangels zu dessen Augenbewegung nicht beurteilen.

Für eine präzise Beurteilung ist demnach ein Blick auf den Stimulus, wie ihn die Probanden zum Zeitpunkt der abweichenden Fixationen präsentiert bekommen haben, erforderlich. Der Stimulus in diesem Zustand lässt sich mithilfe des Zeitstempels der Fixationen aus der hinterlegten Videoaufzeichnung entnehmen. Der Klick auf eines der, im Graphen sichtbaren, Elemente öffnet ein Fenster, in dem der Stimulus wie beschrieben zu sehen ist (siehe *Abbildung 47: Der Stimulus, wie er dem Probanden angezeigt wurde*).

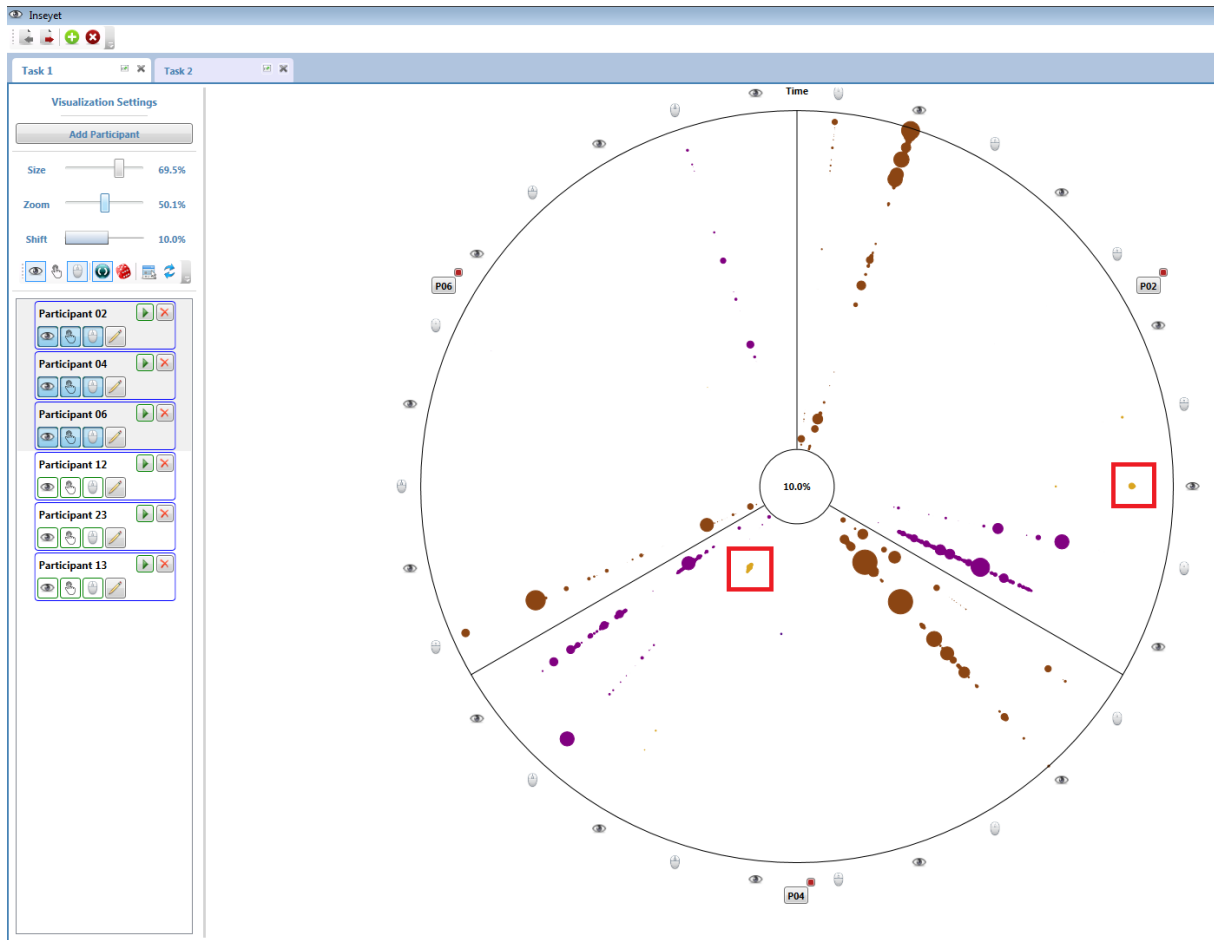


Abbildung 46: Visualisierung mit Zoom, Zeitversatz und ohne Interaktionsprotokoll

Es ist deutlich zu erkennen, dass neben den häufig auftretenden Fixationen von Kapitel-AOI (braun) und Text-AOI (lila) auch vereinzelt orangefarbene Fixationen (Paragraphen-AOI) im Graphen zu sehen sind (nachträglich durch rote Umrandung hervorgehoben). Das ohnehin leere Interaktionsprotokoll ist hier zum Zwecke der Anschaulichkeit nicht mit abgebildet.

Im gezeigten Stimulus ist zu erkennen, dass der Proband die gestellte Frage in der Textbox oberhalb der Spalte für Paragraphen gelesen hat und nicht, wie in der Visualisierung dargestellt, einen Block für einen Paragraphen angeschaut hat. Diese Textbox enthält alle zu beantwortenden Fragen einer Aufgabe. Zwischen den einzelnen Fragen kann interaktiv gewechselt werden. Warum diese Fixation dem AOI-Typ *Paragraph* zugeordnet wurde, ist ebenfalls dargestellt. Alle von *VarifocalReader* protokollierten AOI wurden mit eingezeichnet. Der oberste Block in der Spalte der Paragraphen wurde dabei nicht berechnet, sondern in voller Größe angegeben. Folglich wurden die Fixationen oberhalb der Spalte nicht dem Textfeld für die Frage, sondern dem obersten Block in der Spalte der Paragraphen zugeordnet.

Das Textfeld für die gestellte Aufgabe beziehungsweise Frage wurde bei der Analyse des Programms *VarifocalReader* in Kapitel 5.1 *Anforderungen* nicht als potentielle AOI in Betracht gezogen. Das liegt daran, dass zum Zeitpunkt der Analyse und dem damit verbundenen Erstellen des Aufbaus der Protokolldatei das Feld noch nicht vorhanden oder angedacht war. Das Feld wurde in einer unabhängigen Weiterentwicklung unmittelbar vor Beginn der Studie eingefügt, aus der die verwendeten Beispieldaten stammen. Kurz gefasst: Es wäre ein neuer,

sechster Typ von AOI zu definieren und bei der Umsetzung des Konzepts zu berücksichtigen, dem Fixationen auf diesem Textfeld zugeordnet werden können. Das würde ermöglichen, zu erkennen, wann ein Proband die nächste Teilfrage eingblendet hat. Daraus könnte wiederum ein Rückschluss auf den Fortschritt bei der Bearbeitung der gesamten Aufgabe gezogen werden.

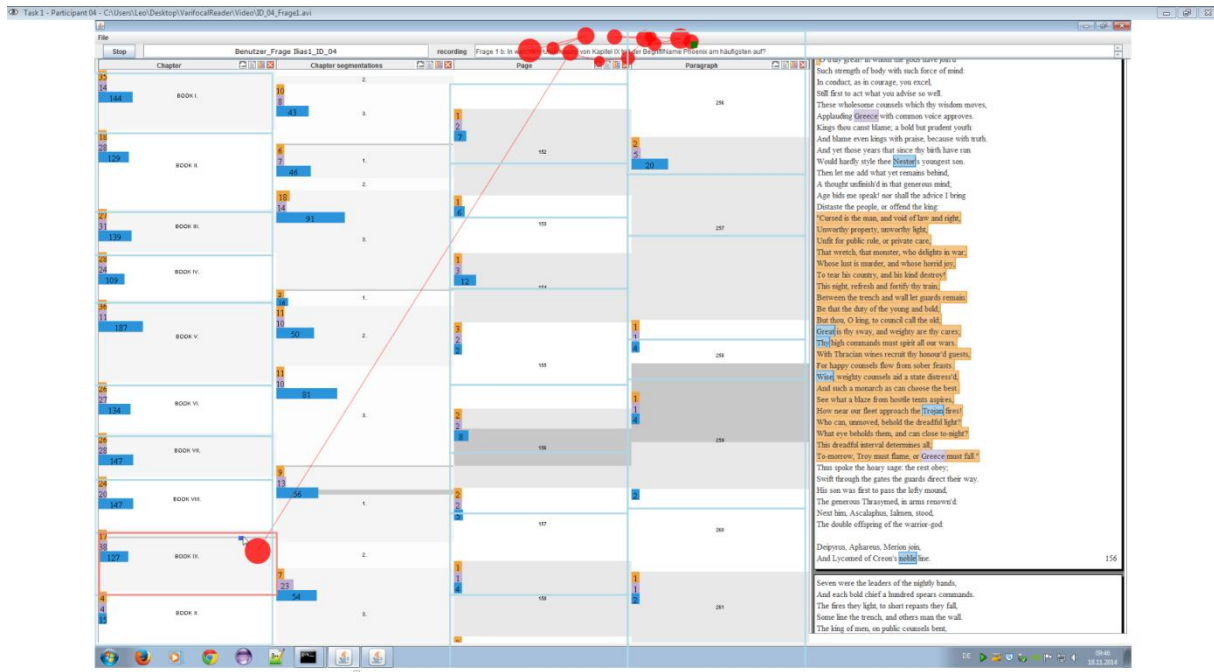


Abbildung 47: Der Stimulus, wie er dem Probanden angezeigt wurde

Der Stimulus in dem Zustand, wie ihn ein Proband zu einem bestimmten Zeitpunkt präsentiert bekommen hat. Zusätzlich eingezeichnet sind die im Programm vorhandenen Daten zu diesem Stimulus. Dazu gehören die Position des Mauszeigers (Mauszeiger beziehungsweise dunkelblaues Quadrat), die Fixationen innerhalb einer geringen Zeitspanne um den konkreten Zeitpunkt herum (rote Kreisflächen) und die registrierten AOI (hellblaue Umrandung).

Trotz des Fehlens eines eigenen AOI-Typs wurde auf diese Weise festgestellt, dass die Probanden sich die im Textfeld angezeigte, zu beantwortende Frage durchgelesen haben. Folglich haben die ausgewählten Probanden zur Informationsgewinnung bei dieser Aufgabe tatsächlich nur die Blöcke für Kapitel und Text verwendet. Je nach Aufgabenstellung für die Probanden lässt sich diese Erkenntnis entsprechend verwerten beziehungsweise interpretieren.

Die Anwendung des Prototyps hat einige Vor- und Nachteile dessen sowie des damit umgesetzten Visualisierungskonzepts verdeutlicht. Eine Auswertung dieser Erkenntnisse befindet sich als Fazit am Ende des Dokuments (siehe Kapitel 8.1 Fazit).

8 Zusammenfassung und Ausblick

In meiner Diplomarbeit habe ich gezeigt, wie man die in einer Eye-Tracking-Studie mit interaktiven Stimuli erfassten Daten zur Augenbewegung, zur Mausbewegung und zur Interaktion eines Probanden gemeinsam visualisieren kann. Diese Stimuli können sich aufgrund der individuellen Interaktionen einzelner Probanden erheblich voneinander unterscheiden. Mein Ziel war es, einen Vergleich der Daten durch eine geeignete grafische Darstellung dennoch zu ermöglichen.

Dafür habe ich die wissenschaftlichen Grundlagen beschrieben, die wichtige Technologien, Konzepte, Visualisierungen und Klassifizierungen für das Thema dieser Arbeit enthalten. Aus einer Konkretisierung der Aufgabenstellung habe ich einen Lösungsansatz entworfen. Durch diesen habe ich andere Arbeiten gefunden, in denen einzelne Aspekte meines Ansatzes behandelt wurden. Die Auswertung dieser Arbeiten diente mir als Ausgangspunkt und Grundlage für die Umsetzung einzelner Ansätze in meinem Konzept für die Visualisierung. Aus dem vorhandenen Material konnte ich weitere, konkrete Anforderungen an eine Lösung ableiten.

Die Entwicklung meines Visualisierungskonzepts wurde anhand der gestellten Anforderungen schrittweise erläutert. Ich habe erklärt, wie die unterschiedlichen Anforderungen entstanden sind, welche Daten bei einer Studie erfasst werden müssen, damit die Anforderungen erfüllt werden können, und wie diese Daten erfasst werden können. Für die Datenerfassung wurde am Beispiel eines Programms ausgeführt, wie diese erfolgen kann und welche Punkte zu berücksichtigen sind. Bei der Formulierung der Gedankengänge, die zu einer geeigneten Visualisierung der Daten führten, habe ich mich an der gegebenen Struktur der erfassten Daten orientiert. Für die Gestaltung der visuellen Darstellung wurde das *Visual Information Seeking Mantra* verwendet. Die Einbeziehung dieses Mantra ergänzt die Beschreibung des Konzepts. Anhand einer tabellarischen Übersicht habe ich zusammengefasst, inwiefern die Anforderungen an das Konzept erfüllt werden.

Zur Demonstration der entwickelten Visualisierung wurde ein Prototyp mit dem Namen *Inseyet* entwickelt, der in diesem Dokument ausführlich beschrieben ist. Es wurden die einzelnen Elemente der Benutzungsschnittstelle, die Funktionsweise des Programms und die Architektur sowie die interne Datenstruktur erklärt. Dabei habe ich den Import und die Verwendung der erfassten Daten veranschaulicht. Informationen zur Entwicklung des Prototyps ergänzen die Beschreibung. Um die Möglichkeiten der entwickelten Visualisierung aufzuzeigen, habe ich ein Beispiel für die Datenanalyse mit *Inseyet* umfassend dokumentiert.

8.1 Fazit

Die Entwicklung des Konzepts und dessen Umsetzung als Prototyp haben gezeigt, dass es möglich ist, die verschiedenen Datentypen, die bei der Interaktion mit interaktiven Stimuli erfasst werden können, gemeinsam so zu visualisieren, dass sie bei einer Analyse miteinander verknüpft werden können. Das gilt auch für den Vergleich solcher Daten zwischen verschiedenen Probanden. Somit können Muster in der Darstellung als Verhaltensweisen von Probanden bei der Interaktion mit einer Visualisierung interpretiert werden. Das ermöglicht Rückschlüsse auf die unterschiedlichen Arten der Informationsgewinnung bei der Analyse interaktiver Stimuli.

Sowohl bei der prototypischen Realisierung des Konzepts als auch bei der Anwendung des Prototyps wurde klar, dass dieses Konzept und der Prototyp Weiterentwicklungspotential bieten. Das Konzept könnte dahingehend erweitert werden, dass *Think-Aloud-Protokolle* als fester Bestandteil bei der Datenerfassung etabliert und klassifiziert werden. Beispielsweise kann ein Proband seine Erkenntnisse über die Funktionsweise einer Visualisierung äußern. Er kann auch seinen Fortschritt während der Informationsgewinnung bezüglich der Aufgabenstellung nennen. Beides sind Inhalte eines *Think-Aloud-Protokolls* und können durch eine Klassifizierung unterschieden werden. Eine andere Erweiterung wäre, die unterschiedliche Dauer von Interaktionen (insbesondere einer Scroll-Interaktion) bereits in einer Übersicht darzustellen.

Im Prototyp selbst werden *Think-Aloud-Protokolle* in der Darstellung der Daten als Graph bislang nicht berücksichtigt. Ein Grund dafür ist, dass solche Protokolle während der Entwicklung des Prototyps nicht vorlagen und nicht klar war, in welcher Form diese vorliegen würden. Eine Möglichkeit ist, solche Protokolle mithilfe des Prototyps selbst an die Daten eines Probanden anzufügen. Zum Zeitpunkt der Fertigstellung des Prototyps war noch keine geeignete Lösung für das automatische Extrahieren einzelner Protokolleinträge und die Zuordnung dieser zu einem konkreten Zeitstempel vorgesehen.

Der Prototyp könnte außerdem mit weiteren Interaktionsmöglichkeiten angereichert werden. Die Gestaltung könnte optimiert werden, indem die darzustellenden Elemente durch eine dynamische Anpassung der Gestaltung besser berücksichtigt werden. Das Einblenden einer Legende zur Erklärung der Visualisierung würde die erstmalige Analyse mit diesem Programm erleichtern. Fortgeschrittene Anwender würden mit der Möglichkeit, die Farben für die verschiedenen Elemente selbst festlegen zu können, weitere Unterstützung erhalten. Für eine große Anzahl Probanden könnte die Anzeigefläche dynamisch insofern angepasst werden, als der Radius der inneren, freien Kreisfläche diese Anzahl berücksichtigt. Damit würde *Visual Clutter* im Inneren des Graphen vermieden.

Die Umsetzung dieser Ideen und Anregungen für den Prototyp konnten im Rahmen meiner Diplomarbeit nicht mehr vorgenommen werden. Eine Realisierung in naher Zukunft ist ange-dacht.

9 Quellenverzeichnis

Baldonado, Michelle Q. Wang, Allison Woodruff, und Allan Kuchinsky. „Guidelines for using multiple views in information visualization.“ *Proceedings of the working conference on Advanced visual interfaces (AVI '00)*, 2000: 110-119.

Blascheck, Tanja, Michael Raschke, und Thomas Ertl. „Circular heat map transition diagram.“ *Proceedings of the 2013 Conference on Eye Tracking South Africa*. New York, NY, USA: ACM, 2013.

Chen, Mon-Chu, John R Anderson, und Myeong-Ho Sohn. „What can a mouse cursor tell us more? Correlation of eye/mouse movements on web browsing.“ *Proceeding CHI '01 Extended Abstracts on Human Factors in Computing Systems*, 2001: 281-282.

In *Eye Tracking Methodology*, von Andrew T. Duchowski, 42. Springer, 2007.

Groth, D. P., und B. W. Murphy. „Tracking User Interaction Within Visualizations.“ *IEEE Symposium on Information Visualization 2004*, Oktober 2004.

Jacob, Robert J. K, und Keith S Karn. *Eye tracking in human-computer interaction and usability research: Ready to deliver the promises*. 2003.

Joos, M., M. Röttin, und B. M. Velichkovsky. „Spezielle Verfahren I: Bewegungen des menschlichen Auges: Fakten, Methoden und innovative Anwendungen.“ In *Psycholinguistik: Ein internationales*, von Gert Rickheit, Theo Herrmann und Werner Deutsch, 142-168. Walter de Gruyter, 2003.

Just, M. A., und P. A. Carpenter. „A theory of reading: From eye fixations to comprehension.“ *Psychological Review*, 1980: 329-354.

Koch, Steffen, Markus John, Michael Wörner, Andreas Müller, und Thomas Ertl. „VarifocalReader - In-Depth Visual Analysis of Large Text Documents.“ *IEEE Transactions on Visualization and Computer Graphics vol.20*, 31. December 2014: 1723-1732.

Kurzhaus, K, F Heimerl, und D Weiskopf. „ISecCube: Visual analysis of gaze data for video.“ *Proceedings of the 2014 Symposium on Eye Tracking Research & Applications*, 2014: 43-50.

Papenmeier, Frank, und Markus Huff. „DynAOI: A tool for matching eye-movement data with dynamic areas of interest in animations and movies.“ *Behavior Research Methods, Volume 42, Issue 1*, Februar 2012: 179-187.

Pohl, Margit, Sylvia Wiltner, und Silvia Miksch. „Exploring Information Visualization - Describing Different Interaction Patterns.“ *Proceedings of the 3rd BELIV'10 Workshop: BEyond time and errors: novel evaluation methods for Information Visualization*, 2010: 16-23.

Raschke, Michael, Xumei Chen, und Thomas Ertl. „Parallel scan-path visualization.“ In *Proceedings of the Symposium on Eye Tracking Research and Applications*, 165-168. 2012.

Schneiderman, Ben. „The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations.“ *IEEE Visual Languages*, 1996: 336-343.

Špakov, Oleg, und Kari-Jouko Rähä. „KiEV: A Tool for Visualization of Reading and Writing Processes in Translation of Text.“ *Proceedings of the 2008 symposium on Eye tracking research & applications*, 2008: 107-110.

Tobii. *T60XL Technical Specification*. 2014.

www.tobii.com/Global/Analysis/Marketing/Brochures/ProductBrochures/Tobii_T60XL_TechnicalSpecification_Leaflet.pdf (Zugriff am 01. Dezember 2014).

—. *www.tobii.com*. 2014. www.tobii.com/eye-tracking-research/global/products (Zugriff am 01. Dezember 2014).

Usability.at. „www.usability.at.“ 2014.

www.usability.at/download/Produktblatt_Eyetracking.pdf (Zugriff am 08. September 2014).

Yi, J. S., , Y. ah Kang, J. T. Stasko, und J. A. Jacko. „Toward a Deeper Understanding of the Role of Interaction in Information Visualization.“ *IEEE Transactions on Visualization and Computer Graphics*, 2007: 1224-1231.

In *Nutzergerechte Entwicklung von Mensch-Maschine-Systemen*, von Detlef Zühlke, 7. Springer, 2012.

Erklärung

Hiermit versichere ich, diese Arbeit selbstständig verfasst und nur die angegebenen Quellen benutzt zu haben. Wörtliche und sinngemäße Übernahmen aus anderen Quellen habe ich nach bestem Wissen und Gewissen als solche kenntlich gemacht.

Stuttgart, den 16. Dezember 2014 _____