



Bachelorarbeit

**NUTZERSCHNITTSTELLENENTWURF  
FÜR UNTERSCHIEDLICHE  
NUTZUNGSKONTEXTE ZUR  
VISUALISIERUNG VON  
ZEITINFORMATIONEN INNERHALB  
CANVAS-BASIERTER  
DISKUSSIONSBEITRÄGE IN  
GRAPHICUSS**

Anastasia Iljassova  
Mat.-Nr.: 3961303

Verantwortlicher Hochschullehrer  
Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill  
und:  
Dr.-Ing. Tenshi Hara, Dr.-Ing. Iris Braun

Eingereicht am 6. Oktober 2017



## **ERKLÄRUNG**

Ich erkläre, dass ich die vorliegende Arbeit selbständig, unter Angabe aller Zitate und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Dresden, 6. Oktober 2017





## AUFGABENSTELLUNG FÜR DIE BACHELOR-ARBEIT

**THEMA:** Nutzerschnittstellenentwurf für unterschiedliche Nutzungskontexte zur Visualisierung von Zeitinformationen innerhalb Canvas-basierter Diskussionsbeiträge in Graphicuss

Name:	Ijassova, Anastasia	Studiengang:	BA Medieninformatik (PO 2009)
Matrikel-Nummer:	3961303	Projekt/Fokus:	AMCS
verantwortlicher Hochschullehrer:	Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill		
involvierte Mitarbeiter:	Dr.-Ing. Tenshi Hara, Dr.-Ing. Iris Braun		
Beginn:	21. Juli 2017	Einreichung:	6. Oktober 2017

### ZIELSTELLUNG

Am Lehrstuhl Rechnernetze wird mit Auditorium Mobile Classroom Service ein umfangreiches Audience Response System in Form eines Web-Services und diverser Plattformspezifischer Applikationen entwickelt. Eine zukünftige Komponente von AMCS wurde in Form von Graphicuss bereits vorbereitet.

Graphicuss kombiniert Aspekte von klassischen Forensystem (textbasierte Kommunikation) mit denen von virtuellen interaktiven Whiteboards (grafikbasierter Austausch). So ist es insbesondere möglich, an einem beliebigen Punkt im zeitlichen Verlauf eines grafischen Beitrages eine Zitation anzusetzen und Ergänzungen vorzunehmen (temporal korrelierte Zitation). Die bisherigen Arbeiten zu Graphicuss fokussierten dabei auf die eigentliche Erstellung von Beiträgen, das Zitieren und das Speichern in Form von geeigneten Datenmodellen.

Im Rahmen dieser Bachelor-Arbeit sollen geeignete Nutzerschnittstellen untersucht werden, um den temporalen Verlauf intuitiv verständlich zu machen. Die Annahme dabei ist, dass unterschiedliche Nutzungskontexte (textlastige Kommunikation, diagrammlastige Kommunikation, et cetera) unterschiedliche Visualisierungen der Zeitkomponente bedürfen.

Neben einer State-of-the-Art-Analyse soll geklärt werden, welche Möglichkeiten die Ansätze bieten, und welche Ansätze sinnvoll umgesetzt werden können. Die Ergebnisse sollen in einer begründeten Empfehlungstabelle mit Zielvorgaben münden, auf deren Basis in weiteren studentischen Arbeiten Konzepte und Prototypen entwickelt werden können.

Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill  
(verantwortlicher Hochschullehrer)

## **SCHWERPUNKTE**

- Analyse von bestehenden Konzepten,
- Definieren von Kontexten,
- Definieren von Anforderungen,
- Konzeption von Lösungsvorschlägen, und
- Erstellen einer Empfehlungstabelle.

# INHALTSVERZEICHNIS

<b>1</b>	<b>Einleitung</b>	<b>15</b>
1.1	Motivation . . . . .	15
1.2	Zielsetzung . . . . .	15
1.3	Aufbau der Arbeit . . . . .	16
<b>2</b>	<b>Grundlagen</b>	<b>17</b>
2.1	Forensysteme . . . . .	17
2.2	Virtuelle interaktive Whiteboards (V-IWB) . . . . .	19
<b>3</b>	<b>Analyse des aktuellen Standes</b>	<b>21</b>
3.1	Auditorium Mobile Classroom Service (AMCS) . . . . .	21
3.1.1	Studierender als Nutzer . . . . .	22
3.1.2	Lehrender als Nutzer . . . . .	22
3.2	Graphicuss . . . . .	23
3.3	Anforderungen . . . . .	25
<b>4</b>	<b>Analyse</b>	<b>27</b>
4.1	Verwandte Konzepte . . . . .	27

4.1.1	Online Whiteboards . . . . .	27
4.1.1.1	AwwApp . . . . .	27
4.1.1.2	CoSketch . . . . .	28
4.1.1.3	Scribblar . . . . .	29
4.1.1.4	Zusammenfassung . . . . .	30
4.1.2	Undo-Modelle . . . . .	31
4.1.3	Visualisierungsmetaphern für den Aktionsverlauf . . . . .	33
4.1.3.1	Film-Metapher . . . . .	34
4.1.3.2	Karussell-Metapher . . . . .	35
4.1.3.3	Kartenstapel-Metapher . . . . .	36
4.1.4	Zusammenfassung . . . . .	37
4.2	Nutzungskontexte . . . . .	37
<b>5</b>	<b>Konzeption</b>	<b>41</b>
5.1	Einleitung . . . . .	41
5.2	Nutzungskontexte in Graphicuss . . . . .	42
5.3	Bewertung von Kandidatenmetaphern in Bezug auf Nutzungskontexte . . . . .	43
<b>6</b>	<b>Evaluation</b>	<b>49</b>
6.1	Einführung in die Evaluationsphase . . . . .	49
6.2	Durchführung der Evaluation . . . . .	50
6.3	Auswertung der Ergebnisse . . . . .	51
6.3.1	Auswertung der Antworten auf vordefinierte Fragen . . . . .	51
6.3.2	Auswertung verbaler Äußerungen . . . . .	53
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>57</b>

<b>A Anhang</b>	<b>63</b>
A.1 Blank paper . . . . .	63
A.2 Papier-Prototypen . . . . .	68
A.2.1 Zeichenverlauf des Beitrags . . . . .	69
A.3 Evaluation . . . . .	76



# ABBILDUNGSVERZEICHNIS

2.1	Darstellung eines Threads als Baumstruktur . . . . .	18
2.2	Darstellung eines Threads als Baumstruktur und zeitlicher News-on-top-Einordnung	18
2.3	Themen in einem Latex-Forum . . . . .	19
2.4	V-IWB am Beispiel von AwwApp . . . . .	20
3.1	Screenshot von <i>drawing tool</i> -Komponente . . . . .	24
4.1	Zeichenverlauf (grün markiert ist der aktueller Zustand) . . . . .	29
4.2	Tools in Scribblar . . . . .	29
4.3	Ein lineares undo-Modell . . . . .	32
4.4	Beispiel für ein selektives Modell: undo mit einer 'reverse' Kopie . . . . .	33
4.5	Interface des Windows Players . . . . .	34
4.6	Interface des Youtube Players . . . . .	34
4.7	Youtube Player mit der Zustandsansicht . . . . .	34
4.8	Erste 4 Frames eines GIFs . . . . .	35
4.9	Karussell in iTunes . . . . .	35
4.10	Slider mit Pfeilen für die Navigation . . . . .	36
4.11	Slideshow mit Pagination . . . . .	36

4.12	Android Apps . . . . .	36
5.1	Skizze zum Slider-UI mit 2 Navigationsvarianten . . . . .	45
5.2	Skizze zum VideoPlayer-UI . . . . .	45
5.3	Skizze zum Slideshow-UI . . . . .	45
5.4	Skizze zum Karussell-UI . . . . .	45
5.5	Skizze zum Kartenstapel-UI . . . . .	45
6.1	Auswertung der Ergebnisse von F-1 und F-3 . . . . .	52
6.2	Auswertung der Ergebnisse von F-2 . . . . .	52
6.3	Auswertung der Ergebnisse von F-4 . . . . .	52
6.4	Diagramm zum Vergleich von Design-Varianten. (F-1 bis F-6 sind die während der Evaluation gestellten Fragen) . . . . .	53
A.1	Testperson A. . . . .	63
A.2	Testperson B. . . . .	64
A.3	Testperson C. . . . .	65
A.4	Testperson D. . . . .	66
A.5	Testperson E. . . . .	66
A.6	Testperson E. . . . .	67
A.7	Ein erstellter Beitrag als Ausgangspunkt . . . . .	68
A.8	Papier-Prototyp eines Slider-UIs . . . . .	76
A.9	Papier-Prototyp eines VideoPlayer-UIs . . . . .	76
A.10	Papier-Prototyp eines Slideshow-UIs . . . . .	77
A.11	Papier-Prototyp eines Slideshow-UIs mit den zusätzlichen Buttons . . . . .	77

# TABELLENVERZEICHNIS

3.1	Bewertung von Nutzungskontexten bezüglich Anforderungen . . . . .	26
4.1	Funktionsübersicht von meist bekannten V-IWBs . . . . .	31
5.1	Funktionsübersicht von Graphicuss . . . . .	42
5.2	Bewertung von Nutzungskontexten bezüglich Anforderungen . . . . .	47
6.1	Zusammenfassung von Antworten auf vordefinierte Fragen . . . . .	52
6.2	Auswertung der Interviews mit den Probanden . . . . .	55



# 1 EINLEITUNG

## 1.1 MOTIVATION

Die moderne Welt ist ohne Gadgets, Internet und Technik allgemein kaum noch vorstellbar und auch der Bildungsbereich ist keine Ausnahme. Jeder hat mindestens schon einmal den Faden der Vorlesung verloren oder wusste nicht, ob er das Thema richtig verstanden hat. Der Einsatz von interaktiven Werkzeugen, wie z.B. *Audience Response Systems (ARS)* oder *Clicker*, steigert in solchen Fällen die Interaktion zwischen Lehrenden und Lernenden, verbessert die Aufmerksamkeit und sorgt für Echtzeit-Feedback und Erfassung individueller Fortschritte.

Durch immer weitere Verbreitung von Laptops, Smartphones und anderen Geräten mit ständiger Internetverbindung, lassen sich solche Systeme einfach und überall benutzen. Da bspw. Räume nicht mehr extra ausgestattet werden müssen und die hohen Anschaffungskosten von physischen *Clickers* entfallen, die ansonsten für die interaktive Unterrichtsgestaltung benutzt werden. Moderne *ARS* sind häufig software- oder cloudbasiert und können deshalb auf jedem verfügbaren internetfähigen Gerät der Lernenden aufgerufen und ausgeführt werden.

Außerdem bieten moderne softwarebasierte *ARS* viele weitere Möglichkeiten, die bei der Verwendung physischer *Clickers* nicht möglich wären, wie bspw. Kommunikation und individuelle Empfehlungen. Um diese lernfördernden Interaktionsmöglichkeiten für die Nutzer, in diesem Fall für die Lernende, attraktiver zu machen und deren Verwendung zu erleichtern, müssen nutzerfreundliche Schnittstellen vorbereitet und neue, interessante Funktionalitäten eingebaut werden. In dieser Arbeit werden ein *ARS*, das an der TU Dresden entwickelt wurde, und eine sich in der Entwicklung befindliche Komponente vorgestellt, bei der genau das erfolgen soll.

## 1.2 ZIELSETZUNG

Am Lehrstuhl Rechnernetze der Technischen Universität Dresden wurde ein umfangreiches Audience Response System namens Auditorium Mobile Classroom Service (AMCS) entwickelt. Eine

zukünftige Komponente von AMCS wurde in Form von Graphicuss bereits vorbereitet. Graphicuss ist eine Kombination von Elementen klassischer Forensysteme und virtueller interaktiver Whiteboards. Die Grundidee von Graphicuss besteht darin, dass nicht nur Text zitiert werden kann, was für ein klassisches Forum typisch ist, sondern auch graphische Beiträge an einem beliebigen Punkt des Zeichenverlaufs.

Ziel dieser Bachelorarbeit ist eine Untersuchung möglicher Lösungsansätze, der Entwurf einer Nutzerschnittstelle unter der Annahme, dass es unterschiedliche Nutzungskontexte geben könnten, und eine Evaluation dieser Schnittstelle mit potentiellen Endnutzern. Das Ergebnis dieser Untersuchung und der Evaluation soll als Basis für späteres Prototyping und eine konkrete technische Umsetzung der Nutzerschnittstelle dienen.

### **1.3 AUFBAU DER ARBEIT**

Diese Arbeit beinhaltet sechs weitere Kapitel, in welchen zuerst einige, für die weitere Konzeption wichtige, Grundlagen erläutert werden, wie bspw. der Begriff und die Arten von Forensystemen und virtuellen interaktiven Whiteboards. Im Kapitel 3 wird der aktuelle Stand analysiert und Funktionalitäten von Auditorium Mobile Classroom Service (AMCS) und Graphicuss vorgestellt. Anschließend werden Anforderungen an den zukünftigen Zeichenverlauf definiert.

Das 4. Kapitel ist der Analyse von verwandten Konzepten und der Definition von Nutzungskontexten gewidmet. Der Fokus liegt dabei auf bereits existierenden Online Whiteboards und möglichen Visualisierungsmetaphern für den Aktionsverlauf.

Ausgehend von den in Kapitel 3 definierten Anforderungen und den in Kapitel 4 beschriebenen Nutzungskontexten werden im Kapitel 5 Kandidatenmetaphern in Bezug auf Nutzungskontexte in Graphicuss bewertet, Entwürfe erstellt und diskutiert.

Anschließend werden die entstandenen Entwürfe durch potentielle Endnutzer evaluiert und Ergebnisse dieser Evaluation im Kapitel 6 ausgewertet. Im letzten Kapitel folgt die Zusammenfassung der Arbeit und Ausblick.

## 2 GRUNDLAGEN

In diesem Kapitel sollen zunächst einige Grundlagen erklärt werden, die für das Verständnis der Funktionsweise von bereits existierenden Systemen und für die Konzeption der Nutzerschnittstelle wichtig sind.

### 2.1 FORENSYSTEME

Eine der am meisten einsetzbaren Möglichkeiten des Meinungs- bzw. Wissensaustausches im Internet sind Foren. Dabei stellt ein Forum einen Nachrichten- bzw. Diskussionsbereich zur asynchronen Kommunikation dar. In Foren können Nutzer zu selbst gesetzten oder vom Veranstalter vergebenen Themen debattieren. Mehrere Beiträge zum selben Thema, Fragen und Antworten oder Kommentare dazu werden zu Threads zusammengefasst. Der Einsatz eines solchen web-basierten Diskussionsforums bietet zusätzliche Flexibilität hinsichtlich der zeitlichen Koordination der Diskussion.

Der Hauptunterschied zwischen Foren und Kommunikationsmedien wie bspw. Chat oder Videokonferenzen sind neben der textbasierten Kommunikation die Asynchronität, d.h. die Nutzer lesen oder posten die Beiträge zeitversetzt. Aufgrund dieser Asynchronität und der Parallelität der Diskussionsstränge eignen sich Foren vor allem für den sachbezogenen Austausch, weil Beiträge in Ruhe aufgenommen werden können. Die Nutzer können ihre Reaktionen auf Beiträge ausformulieren und inhaltlich absichern.

Man unterscheidet *offene* Diskussionsforen, bei denen beliebige Personen Zugriff auf die Inhalte haben, von *geschlossenen* Diskussionsforen, zu denen nur ein bestimmter Nutzerkreis, z.B. nach dem erfolgreichen Einloggen, Zugang hat. Hinsichtlich der Moderierung lassen sich wiederum zwei Arten unterscheiden. In *moderierten* Foren erfolgt das Einstellen von Beiträgen durch einen Moderator. Dies kann streng reglementiert sein, d.h. jeder Beitrag muss nach einer Überprüfung freigegeben werden oder die bereits existierenden Beiträge können nachträglich auf irrelevante oder sogar ungesetzliche Inhalte geprüft werden. In *nicht moderierten* Foren dagegen werden alle Beiträge aufgenommen.

Basierend auf der Strukturierung von Beiträgen lassen sich zwei Modellierungstypen von Foren unterscheiden: *die klassischen Web-Foren* und *Bulletin Boards*.

**Die klassischen Web-Foren** stellen Beziehungen zwischen Beiträgen innerhalb eines Threads in einer hierarchischen Baumstruktur dar, so dass zu erkennen ist, welcher Beitrag als Antwort auf welchen anderen Beitrag erstellt wurde.

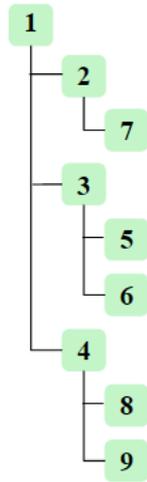


Abbildung 2.1: Darstellung eines Threads als Baumstruktur

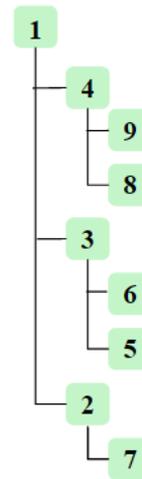


Abbildung 2.2: Darstellung eines Threads als Baumstruktur und zeitlicher News-on-top-Einordnung

Die baumartige Darstellungsform bildet die Struktur der Diskussion ab. Vor allem bei größeren Threads, bei denen mehrere Teildiskussionen existieren, sorgt sie für eine optische Unterstützung bei der Orientierung. Die Erfahrung zeigt jedoch, dass sich Teilnehmer meist sehr schnell an das Schema gewöhnen.

**Ein Bulletin Board** vereint alle Postings eines Themas auf einer Seite. Im Mittelpunkt steht nicht mehr der Thread sondern das Thema. Wie in der Abb. 2.3 zu sehen ist, kann man keine Aussage über die Diskussionsstruktur bzw. die Reihenfolge und Anordnung von Beiträgen treffen. Die grundsätzliche Darstellungsform ist dabei eine Liste aller Beiträge zu einem Thema. Ein Thema kann etwas beliebig Spezielles sein. Da meist mehrere Daten zu einem Thema angezeigt werden, beispielsweise der Name des Teilnehmers, der das Thema eröffnet hat, oder die Anzahl der Beiträge zu dem Thema, wird die Liste zu einer Tabelle. Jede Zeile der Tabelle ist ein Thema, und jede Spalte bietet Informationen zu einer Daten-Eigenschaft, die zu jedem Thema gehört.

Themen	Autor	Antworten	Zugriffe	Letzter Beitrag
<b>Bekanntmachungen</b>				
TUD-Script Schriftinstallation	tudscr	0	3632	Montag 24. August 2015, 10:26 tudscr →
Schriftinstallation für weitere TUD-Klassen	tudscr	0	612	Freitag 20. Juni 2014, 14:02 tudscr →
TUD-Script-Bundle v2.05	tudscr	2	11881	Dienstag 26. Juli 2016, 18:23 tudscr →
<b>Themen</b>				
TUD-Script-Klassen: Installationsprobleme [  Gehe zu Seite: 1, 2, 3, 4, 5 ]	tudscr	48	6823	Mittwoch 14. Juni 2017, 14:18 tudscr →
Assoziiere TU Schriften mit klassischen Befehlen	thewaywewalk	2	138	Mittwoch 14. Dezember 2016, 11:27 thewaywewalk →
Schriften nicht gefunden	p0wl	3	230	Donnerstag 25. August 2016, 10:04 tudscr →
Verzeichnis localtexmf	tokro	5	398	Montag 4. Juli 2016, 22:20 tudscr →
Schriftarten in tudbeamer	Fra	1	460	Sonntag 17. Januar 2016, 21:11 tudscr →
Schriftinstallation unter Mac OS X 10.11 El Capitan	MartinB	1	538	Sonntag 18. Oktober 2015, 22:40 tudscr →
Warnungen beim Installieren, Schriften nicht gefunden	Marsian	4	631	Dienstag 13. Oktober 2015, 10:14 Marsian →
pdf-Umwandlung nicht möglich - Windows 10 - TeXnic	chemical_instinct	4	635	Samstag 12. September 2015, 13:15 chemical_instinct →
Schriftarteninstallation für MikTeX portable [  Gehe zu Seite: 1, 2 ]	ReggaeGandalfVB	14	1759	Dienstag 4. August 2015, 09:39 claimy →
Installationsabbruch Syntax error	DerBastian	2	580	Montag 3. August 2015, 14:22 DerBastian →
Fatal error occurred, no output PDF file produced [  Gehe zu Seite: 1, 2 ]	mobo	15	1640	Montag 29. Juni 2015, 09:20 tudscr →
Schriftinstallation unter TeXlive portable	ReggaeGandalfVB	4	670	Freitag 5. Juni 2015, 13:51 ReggaeGandalfVB →
Fehler bei PDF-Übersetzung des Beispieldocuments	twinky_99	9	1275	Dienstag 2. Juni 2015, 14:55 tudscr →
Mac OS X / TexShop / Schriften werden nicht gefunden	NandM	6	869	Dienstag 19. Mai 2015, 22:02 NandM →
Problem: No file firstmsc.rc. [Linux, TeXlive 2014]	waschmaschine	4	775	Donnerstag 12. Februar 2015, 10:57 tudscr →
LuaLaTeX + TU Dresden Schriftart + fontspec = nicht möglich?	TeXeR	6	1028	Mittwoch 11. Februar 2015, 15:19 tudscr →

Abbildung 2.3: Themen in einem Latex-Forum<sup>1</sup>

## 2.2 VIRTUELLE INTERAKTIVE WHITEBOARDS (V-IWB)

Im Kontext dieser Arbeit ist es wichtig virtuelle interaktive Whiteboards von klassischen digitalen Tafeln zu unterscheiden. Im Vergleich zu bekannten digitalen Whiteboards, die heutzutage immer öfter in den Schulen, Universitäten und sogar Büros zu sehen sind, brauchen V-IWBs kein physisches Display und erlauben voneinander entfernten Nutzern in Echtzeit auf einem Whiteboard zu arbeiten. Also sind die virtuelle interaktive Whiteboards ein Mittel zur kollaborativen Entwicklung und Bereitstellung von Skizzen. Hara T. definiert in seiner Arbeit ein V-IWB als ein kollaboratives System mit (m:n)-Beziehung [Har16]:

- $m=1$  und  $n=0$ : Typ 'Persönliches Notizbuch' (Der Nutzer erstellt den Inhalt und teilt ihn nur mit sich selbst)
- $m=1$  und  $n \in \mathbb{N}_{n \neq 0}$ : Typ 'Persönliches Bulletin' (Der Nutzer erstellt den Inhalt und teilt ihn mit mindestens einem anderen Nutzer im read-only Modus)
- $m \in \mathbb{N}_{>1}$  und  $n = 0$ : Typ 'Geteiltes Notizbuch' (Mehrere Nutzer erstellen den Inhalt und teilen ihn nur mit sich selbst)

<sup>1</sup> <https://latex.wcms-file3.tu-dresden.de/phpBB3/index.php> - - letzter erfolgreicher Aufruf am 08.06.2017

- $m \in \mathbb{N}_{>1}$  und  $n \in \mathbb{N}_{n \neq 0}$ : Typ 'Geteiltes Bulletin' (Mehrere Nutzer erstellen den Inhalt und teilen ihn mit mindestens einem anderen Nutzer im read-only Modus)

Alle definierten Typen von V-IWBs können für die Organisation des Arbeitsraums erfolgreich eingesetzt werden. Die Bulletins, geteilte Bulletins vor allem, ermöglichen die Gruppenarbeit, auch wenn die Teilnehmer sich nicht im selben Raum befinden. Außerdem sind solche Whiteboards nicht nur von eigenen, sondern auch von ausgeliehenen oder auch Universitätsgeräten verfügbar. Als Beispiele für die Hauptfunktionen von V-IWBs kann man in erster Linie Zeichnen, Erstellen und Ausfüllen von Formen, Editieren, Verschieben und Löschen nennen. Abhängig von dem spezifischen Programm erlaubt ein virtuelles Whiteboard in der Regel das Hochladen von Dateien wie z.B. PowerPoint-Präsentationen, Audiodateien, Fotos und Dokumenten, die mit den anderen Teilnehmern geteilt werden können. Neben zuvor genannten Hauptfunktionen kann ein V-IWB auch als Kommunikationsmittel zwischen den verschiedenen Gruppenmitgliedern sein. Dabei werden sowohl Text- als auch Video-Chats unterstützt.

Der größte Teil heutzutage existierender virtuellen Whiteboards wurde nicht für den Einsatz auf Smartphones oder Tablets konzipiert. Allerdings haben einige Hersteller angefangen dieses Problem zu lösen. So ist die AwwApp<sup>2</sup> ein Beispiel für ein virtuelles Whiteboard, auf das über Web-Browser (Computer, Tablet, Smartphone) zugegriffen werden kann.



Abbildung 2.4: V-IWB am Beispiel von AwwApp

<sup>2</sup> <https://awwapp.com/> - letzter erfolgreicher Aufruf am 04.10.2017

# 3 ANALYSE DES AKTUELLEN STANDES

In diesem Kapitel werden existierende Systeme vorgestellt, charakterisiert und als Grundlage für die Formulierung weiterer Anforderungen verwendet.

## 3.1 AUDITORIUM MOBILE CLASSROOM SERVICE (AMCS)

Audience Response Systems (ARS) werden heutzutage im Rahmen von Lehrveranstaltungen oder bei Vorträgen mit zahlreichen Teilnehmern immer häufiger eingesetzt. Sie ermöglichen eine höhere Interaktivität zwischen Dozenten und Zuhörern. Der Dozent wird damit in die Lage versetzt, Fragen verschiedenster Form an die Lernenden zu richten und damit nicht nur deren Kenntnisstand abzufragen, sondern auch die Aufmerksamkeit zu fördern und direktes Feedback zur gehaltenen Lehrveranstaltung zu erhalten. Ein großer Vorteil von ARS ist, dass Abstimmungen anonym sind. Keiner muss Bedenken haben, dass seine Antwort ausgelacht wird oder seine Äußerung zur Lehrveranstaltung ihn später benachteiligt. Als Beispiele für Audience Response Systems können einerseits klassische „Clicker“ und andererseits moderne, webbasierte Lösungen, die mit Hilfe internetfähiger Mobilgeräte der Teilnehmer arbeiten, aufgeführt werden.

Am Lehrstuhl Rechnernetze der Technischen Universität Dresden wird mit Auditorium Mobile Classroom Service (AMCS) ein umfangreiches Audience Response System in Form eines Web-Services und diverser plattformspezifischer Applikationen entwickelt. Ziel des Systems ist es, eine bessere adaptive Passung zwischen den Lehrzielen der Dozenten und den Lernzielen der Studenten sowie eine einfachere Regulierung der individuellen Lernprozesse zu ermöglichen[BKKS15]. AMCS bietet zahlreiche, zusätzlich zu den klassischen, Funktionalitäten und bestimmte Möglichkeiten der Interaktion in der Vorlesung an. Alle Funktionalitäten werden über Smartphones bzw. andere internetfähige Geräte während der Vorlesung zur Verfügung gestellt. Das ganze System kann aus zwei Nutzersichten betrachtet werden: *Studierender* und *Lehrender*.

### 3.1.1 Studierender als Nutzer

Das Einloggen und die Nutzung des Systems von Studenten erfolgt, wie es für ein Audience Response System charakteristisch ist, anonym über die Bildung eines Pseudonyms. Das hat unter anderem den Vorteil, dass jeder Student bei einer mehrfachen Teilnahme an Umfragen im Laufe des Semesters am Ende die Möglichkeit hat, eine Zusammenfassung aller abgegebenen Antworten abzurufen, welche zur Nachbereitung verwendet werden kann.

Zu Beginn der Lernveranstaltung werden einige Fragen über die persönlichen Ziele der Teilnehmer gestellt, die als Grundlage zur Erstellung von Nachrichten und Hinweisen im Laufe und nach der Vorlesung dienen.

Je nach Einstellung der Lehrveranstaltung durch den Lehrenden können die Studenten folgende Funktionen nutzen: *Fragen beantworten* und *Nachrichten erhalten*.

**Fragen beantworten.** Fragen werden auf dem Gerät automatisch freigeschaltet sobald der Lehrende den Zustand bzw. die Folie der Veranstaltung anpasst und hierfür Fragen existieren. Es sind verschiedene Fragetypen möglich, wie bspw. Einfach- und Mehrfachauswahl, Skalenfrage oder Freitextfrage.

**Nachrichten erhalten.** Sofern der Student dem Erhalt von Benachrichtigungen zugestimmt hat und das Gerät auch Benachrichtigungen unterstützt, kann er Nachrichten erhalten. Dadurch können unter anderem weitere Materialien und Links, Hinweise für die Nachbereitung oder vertieftes Lernen im Rahmen der Lehrveranstaltung und je nach zu Beginn der Vorlesung angegebener Präferenz oder auch Hinweise über die Themenwiederholung von zuvor falsch beantworteten Lernaufgaben übermittelt werden.

### 3.1.2 Lehrender als Nutzer

Nachdem der Lehrender die Zugangsdaten übermittelt bekommen hat, kann er folgende Funktionalitäten des Systems nutzen:

**Zustand und aktive Folie der Veranstaltung setzen.** AMCS bietet die Möglichkeit, die Sichtbarkeit der Veranstaltung und verschiedener Fragen zu verwalten. Abhängig von den Fragekontexten kann der Lehrende zwischen vier verschiedenen Zuständen wechseln. Im *Offline*-Zustand sind nur die Kursfragen für Studierende sichtbar. Um sowohl Vorbereitungs- als auch Kursfragen sichtbar zu machen, kann der *vor der Veranstaltung*-Zustand aktiviert werden. Ist der *während der Veranstaltung*-Zustand ausgewählt, sind auch die Fragen der aktiven Folie sichtbar. Nach der Veranstaltung kann der Lehrende den Zustand in *nach der Veranstaltung* wechseln und damit sowohl Nachbereitungs- als auch Kursfragen für Studierende freischalten.

**Fragen an die Studierenden erstellen und auswerten.** Interaktive Lernaufgaben können bei den notwendigen kognitiven und metakognitiven Prozessen unterstützend wirken und für unterschiedliche Ziele eingesetzt werden. Zu einem bestimmten Zeitpunkt der Vorlesung eingesetzte Fragen können zur Vorwissensüberprüfung, Motivation, Verständnisüberprüfung oder Evaluation dienen. Sie ermöglichen den Studierenden eine aktive Auseinandersetzung mit dem Stoff sowie

eine Regulierung des eigenen Lernprozesses. Die Auswertung der Antworten erfolgt in Echtzeit und je nach Fragentyp wird mit Hilfe eines bestimmten Diagrammtyps mit gehöriger Übersicht der Ergebnisse dargestellt, z.B. ein Kuchendiagramm.

**Nachrichten definieren.** Unter Nachrichten versteht man in diesem Kontext die Benachrichtigungen des Browsers oder der App an die Studierenden unter der Voraussetzung, die in der Funktionalitätsbeschreibung 'Nachrichten erhalten' definiert ist. AMCS erweitert das Konzept von Nachrichten um konditionale Freigaben. Damit können Nachrichten (Prompts) in Abhängigkeit von zuvor gegebenen Antworten angezeigt werden. Laut [BKKS15] können auf diese Weise die verschiedenen Nachrichtentypen übermittelt werden. Je nach Präferenz (z.B. Lernzielorientierung, Leistungszielorientierung, Prüfungsvorbereitung oder Interesse am Thema) werden durch *meta-kognitive Prompts* strategische Hinweise für die Vorbereitung und Nachbereitung des Vorlesungsstoffs gegeben. Studierende, die bei Lernaufgaben zu Beginn einer Vorlesung Fehler gemacht haben, können zu einem späteren Zeitpunkt durch *kognitive Prompts* darauf hingewiesen werden, dass das in der Lernaufgabe behandelte Thema noch einmal wiederholt wird. Auch die Bereitstellung von weiterführenden Materialien und Links wird durch *normale Prompts* unterstützt.

## 3.2 GRAPHICUSS

Am Lehrstuhl Rechnernetze der Technischen Universität Dresden wurde eine zukünftige Komponente von AMCS in Form von Graphicuss bereits vorbereitet. Graphicuss stellt eine Kombination der Aspekte klassischer Forensysteme mit denen von virtuellen interaktiven Whiteboards dar. Als ein grafikbasiertes Discussionssystem für Lehrveranstaltungen besitzt die Komponente laut [Che16] drei Basisfunktionalitäten:

**Kursverwaltung (course management).** Der als Lehrender identifizierter Nutzer kann Kurse erstellen und die von ihm erstellten Kurse verwalten. Bei der Erstellung kann dem Kurs ein Name gegeben und ein Hintergrundbild zur Verbesserung der Wiedererkennung hochgeladen werden. Außerdem kann der Lehrende eine konkrete Beschreibung dieses Kurses hinzufügen. Ein neuer Kurs erhält automatisch einen eindeutigen Identifikationscode, mit dessen Hilfe er von Teilnehmern gefunden werden kann. Wenn ein Student sich für einen Kurs interessiert, hat er die Möglichkeit diesen Kurs zu seiner Favoritenliste hinzuzufügen. Das ermöglicht schnelleres Auffinden und Zugreifen auf relevante Veranstaltungen.

**Fragenverwaltung (question management).** Wie zuvor erwähnt wurde, besitzt Graphicuss einige Eigenschaften klassischer Forensysteme, die man in der Fragenverwaltung wiedererkennen kann. Der Studierende hat die Möglichkeit, eigene Fragen bezüglich des Stoffs der Vorlesung zu posten, bereits gestellte Fragen zu ändern oder diese zu löschen, sofern noch keine Kommentare dazu erstellt wurden. Bewertungsmöglichkeiten stellen einen entscheidenden Faktor für den Aufbau von Foren mit qualitativen Inhalten dar. So hat der Nutzer die Möglichkeit abzustimmen, ob eine Frage als hilfreich für andere Teilnehmer empfunden wird oder nicht. Außerdem kann der Autor einer Frage die, seiner Meinung nach, beste und hilfreichste Antwort markieren, sodass diese automatisch ganz oben in der Liste angezeigt wird.

**Antwortverwaltung (answer management).** Die mit dem Thema einer Frage vertrauten Nutzer können diese beantworten. Eigene abgegebene Antworten können im Nachhinein geändert oder gelöscht werden. Wie im Unterabschnitt 'Fragenverwaltung' wird hier die Idee der Bewertung umgesetzt. Die bestbewertete Antwort wird zuerst angezeigt. Außerdem können Antworten, mit dem Ziel, die Originalantwort zu vervollständigen oder auf Fehler hinzuweisen, zitiert werden.

Die Anpassung von Graphicuss an Bildungszwecke und die Erhöhung der Interaktivität des Systems erfolgt durch Integration eines canvas-basierten drawing tools und einer Echtzeit- Funktionalität, die in den folgenden Abschnitten vorgestellt werden.

**Drawing tool.** Natürlich können nicht alle Gedanken mit Hilfe von Worten ausgedrückt werden. Es gibt auch Fragen, die sich deutlicher anhand von Skizzen oder einfachen Diagrammen erklären lassen. Die Zeichenoberfläche des Graphicuss bietet die Möglichkeit nicht nur Text sondern auch geometrische Objekte wie Rechtecke, Kreise, Linien usw. zur Antworten hinzuzufügen. Die gezeichneten Objekte können auch verändert werden: durch Farbwechsel und Größenanpassungen sind Nutzer in der Lage wichtige Teile einer Antwort hervorzuheben. Eine weitere wichtige Komponente der Zeichenoberfläche ist der *Verlauf*. Basierend auf den Annahmen, dass Nutzer während des Zeichnens Fehler unterlaufen, oder diese sich entschließen Teile ihrer graphischen Antwort überarbeiten zu wollen, ist ein Verlauf erforderlich. In Form der Auflistung chronologischer Zustände bietet dieser die Möglichkeit mittels undo- und redo-Funktionen zu einem beliebigen Zustand zurückzuspringen und damit den Zeichenprozess deutlich zu verbessern.

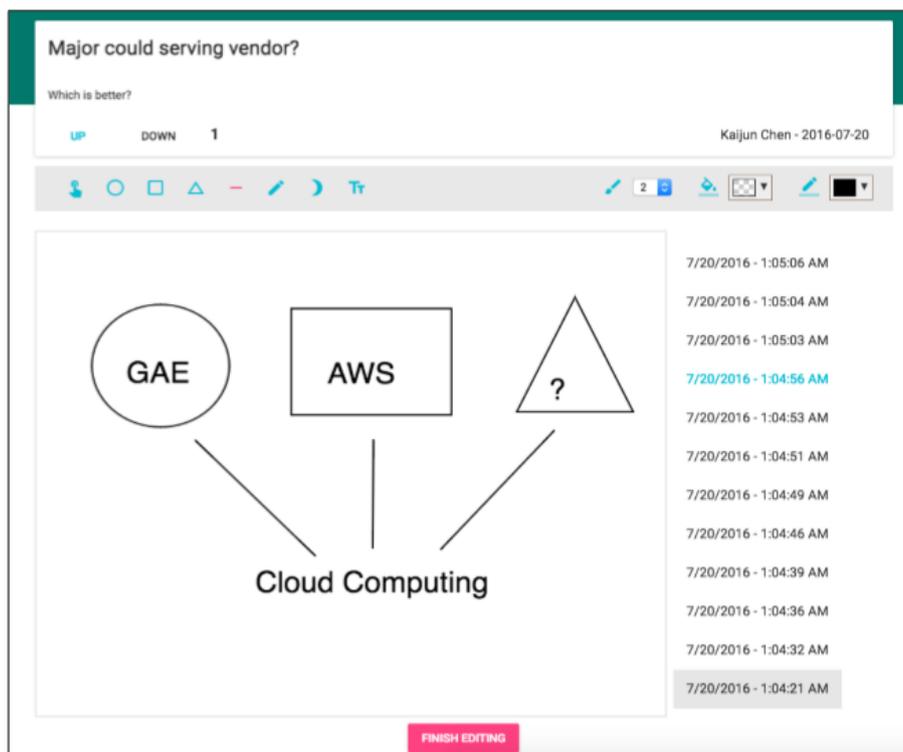


Abbildung 3.1: Screenshot von *drawing tool*-Komponente[Che16]

**Real time.** Ein wichtiger Aspekt für bessere Interaktivität und höhere Motivation der Studierenden ist analog zu den oben diskutierten Forensystemen eine bestimmte Echtzeit-Funktionalität. In Graphicuss sind folgende zwei Varianten definiert:

*Echtzeit-Fragenliste:* Alle neuen Fragen, die von den anderen Nutzer gepostet werden, werden automatisch an den Nutzer geschickt.

*Echtzeit-Antwortenreihenfolge:* Die Antworten werden automatisch nach der Bewertung sortiert. Dafür wird das Neuladen der Seite nicht benötigt.

### 3.3 ANFORDERUNGEN

Im Folgenden werden Anforderungen definiert, die für die weitere Ausarbeitung und Konzeption wichtig sind. Als Grundlage für die Anforderungsdefinition werden die Aufgabenstellung und die in diesem Kapitel vorgestellten Systeme benutzt.

Entsprechend der Aufgabenstellung ist das Ziel dieser Arbeit ein Nutzerschnittstellenentwurf für die Graphicuss-Komponente in Bezug auf unterschiedliche Nutzungskontexte. Die Komponente muss also unterschiedliche Arten der Kommunikation berücksichtigen und nicht nur den zeitlichen Verlauf sondern auch dessen Richtung repräsentieren. Der Nutzer muss erkennen können, wo der Anfang bzw. wo das Ende des Verlaufs ist und in welche Richtung er sich hindurch navigieren kann, um zu einem gewünschten Zustand zu kommen. Bei der Nutzung von Graphicuss können auch große Endbilder entstehen mit einem entsprechend langen Verlauf. Um die Benutzerfreundlichkeit zu fördern, muss die Oberfläche dem Nutzer ermöglichen sich in größeren Schritten, d.h. beispielsweise fünf Bilder vor/zurück, durch den Verlauf zu navigieren.

Da es hier um eine grafische Benutzeroberfläche geht, wird die technische Seite der späteren Umsetzung in dieser Arbeit nicht betrachtet. Der Fokus liegt also auf den Anforderungen zur Nutzerschnittstelle selbst und deren Funktionalitäten. Begriffe wie 'Benutzerfreundlichkeit' und 'Bedienbarkeit' werden stets mit den Anforderungen an eine Benutzeroberfläche verbunden. Die beim Schnittstellenentwurf verwendeten Metaphern müssen verständlich und nachvollziehbar sein. Icons und Bedienelemente müssen sowohl groß genug als auch bekannt sein, d.h. nicht zu Missverständnissen oder Fehlinterpretationen führen. Auftretende Fehler müssen passend und aussagekräftig erklärt sowie kommentiert werden.

Wie bereits erwähnt wurde, können die AMCS-Funktionalitäten über verschiedene internetfähige Geräte zur Verfügung gestellt werden. Das erfordert eine Anpassbarkeit der Anwendung an spezifische Eigenschaften des jeweiligen Gerätes. Die Nutzerschnittstelle muss responsiv sein, da unterschiedliche Geräte unterschiedliche Auflösung haben. Außerdem muss beachtet werden, dass auch Bedienmöglichkeiten unterschiedlich sein können. Wenn es bei einem PC oder einem Laptop um eine Maus-Tastatur-Eingabe geht, handelt es sich bei einem Touchscreen-Gerät um eine Finger-Eingabe. Deswegen müssen Navigationselemente, Icons und Menüs für beide Eingabemodalitäten konzipiert werden. Beispielsweise würde eine klickbare Pagination mit Zahlen für Laptops durch einen Swipe-Vorgang für Smartphones ersetzt werden.

Zusammenfassend lassen sich die aufgestellten Anforderungen in drei Gruppen unterteilen und werden im Folgenden definiert: *Muss-Anforderungen* (was die Nutzerschnittstelle anbieten muss), *Kann-Anforderungen* (was die Nutzerschnittstelle anbieten könnte) und *Abgrenzungsanforderungen* (was die Nutzerschnittstelle nicht anbietet).

<b>Abk.</b>	<b>Anforderungsdefinition</b>
<i>Muss-Anforderungen</i>	
MA-1	Die Nutzerschnittstelle muss mittels verständlicher Metaphern dargestellt werden.
MA-2	Die Nutzerschnittstelle muss unterschiedliche Nutzungskontexte berücksichtigen.
MA-3	Die Nutzerschnittstelle muss den zeitlichen Verlauf und dessen Richtung (z.B. Zeitstempel) visualisieren.
MA-4	Die Nutzerschnittstelle muss die Sichtbarkeit des aktuellen Zustandes im Verlauf gewährleisten.
MA-5	Die Nutzerschnittstelle muss die Möglichkeit anbieten sich in größeren Schritten durch den Verlauf zu navigieren.
<i>Kann-Anforderungen</i>	
KA-1	Die Nutzerschnittstelle muss responsiv sein.
KA-2	Die Nutzerschnittstelle muss große und übersichtliche Icons verwenden.
KA-3	Die Nutzerschnittstelle muss an unterschiedliche Endgeräte anpassbare Bedienelemente verwenden.
KA-4	Die Nutzerschnittstelle muss unterschiedliche Eingabemodalitäten berücksichtigen.
KA-5	Die Nutzerschnittstelle muss bei auftretenden Fehlern passende Fehlermeldungen ausgeben.
<i>Abgrenzungsanforderungen</i>	
AA-1	Es darf keine leere Seite abgeschickt werden.
AA-2	Nach dem Zitieren und anschließendem Editieren wird der alte Verlauf (alles, was nach dem zitierten Zustand folgte) in der zitierten Version durch einen neuen ersetzt. Der Verlauf des Ursprungsbeitrages bleibt unverändert.
AA-3	Die Barrierefreiheit (z.B. Screenreader) wird nicht gewährleistet.

Tabelle 3.1: Bewertung von Nutzungskontexten bezüglich Anforderungen

Die definierten Anforderungen dienen im Folgenden als Grundlage zur Konzeptentwicklung und anschließenden Evaluation.

# 4 ANALYSE

In diesem Kapitel sollen vorhandene Systeme und mögliche Lösungsansätze untersucht werden.

## 4.1 VERWANDTE KONZEPTE

### 4.1.1 Online Whiteboards

#### 4.1.1.1 AwwApp

‘A Web Whiteboard Application’ (AwwApp)<sup>3</sup> ist eine einfache und intuitiv verständliche webbasierte Plattform mit Collaboration-Funktion. Ein wichtiger Vorteil gegenüber anderen V-IWBs, wie im Unterkapitel 2.2 bereits erwähnt wurde, ist die Plattform- und Geräteunabhängigkeit. Der Nutzer kann auf einem beliebigen Gerät und in einem beliebigen Browser Whiteboards erstellen und/oder editieren. Im Free-Modus stehen die folgenden Basisfunktionalitäten zur Verfügung:

*Farbauswahl* für Text oder Stift



*Größenauswahl* für Text, Stift oder Radiergummi



*Werkzeugauswahl:* Stift, Radiergummi, Text, Bild-Upload, Undo, Löschen



<sup>3</sup> <https://awwapp.com/> - letzter erfolgreicher Aufruf am 04.10.2017

Neben den zuvor erwähnten Basisfunktionalitäten bietet die AwwApp dem Nutzer die Möglichkeit erstellte Boards mit anderen Nutzern zu teilen. Dies erfolgt bspw. mittels per Email verschickten URLLinks oder QR-Codes. Die eingeladenen Teilnehmer können ohne Einschränkungen zeichnen und/oder Änderungen an dem Board vornehmen. Sobald für ein Whiteboard mehr als nur ein Nutzer registriert ist, wird die Kommunikationsmöglichkeit mittels eines Text-Chats angeboten.

Die Anmeldung und die Bereitstellung von persönlichen Informationen ist nicht erforderlich, erlaubt aber das Freischalten folgender zusätzlichen Funktionalitäten.

*Zusätzliche Funktionalitäten:* Linie, Rechteck (gefüllt), Ellipse (gefüllt), Position, PDF/PPT-Upload

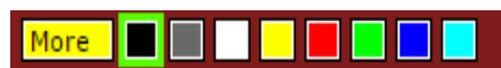


Die AwwApp stellt eine Reihe von Premium-Werkzeugen zur Verfügung, die neben grundlegenden geometrischen Figuren wie Linie, Rechteck oder Ellipse auch weitere organisatorische Optionen eröffnen. So ist es nach der Anmeldung möglich, erstellte Boards zu laden bzw. speichern, eine Videokonferenz zu organisieren oder Teilnehmer im read-only Modus zur Kollaboration einzuladen. Die AwwApp umfasst also alle vier bereits im Kapitel 2.2 definierten Typen von V-IWBs: sowohl persönliche als auch geteilte Notizbücher und Bulletins.

#### 4.1.1.2 CoSketch

CoSketch<sup>4</sup> ist ein anderes Multi-User Online Whiteboard für schnelles Visualisieren und Teilen von Ideen. Alle Basisfunktionalitäten sind, analog zu AwwApp, vorhanden. Der Nutzer kann sowohl andere Teilnehmer durch Verschicken eines Links zur Zusammenarbeit einladen, als auch mit diesen über die im Whiteboard eingebaute Chat-Funktion kommunizieren.

*Farbauswahl* für Text, Stift oder geometrische Figuren



*Größenauswahl* für Stift oder geometrische Figuren



*Werkzeugauswahl:* Linie, Rechteck, Ellipse, Text, Radiergummi, Verschieben



Diese Plattform bietet eine besondere Option, die es erlaubt, Google Maps als Hintergrund der Zeichenoberfläche zu benutzen und direkt auf der Karte zu zeichnen. Außerdem lässt sich das Canvas beliebig skalieren, was Anpassungen an den jeweiligen Bildschirm vereinfacht. Was CoSketch von den anderen betrachteten V-IWB unterscheidet ist der Zeichenverlauf mit sofortiger Vorschau.

<sup>4</sup> <http://cosketch.com/> - letzter erfolgreicher Aufruf am 04.10.2017

Die Abb. 4.1 stellt ein Beispiel für einen Verlauf in Tabellenform dar. In der ersten Spalte wird der Nutzer, der die Änderung vorgenommen hat, vermerkt und in der zweiten was für Änderungen gemacht wurden (Text, Zeichnen usw.) Beim Hovern über die Liste werden Zustandsänderungen sofort angezeigt, was zur Entstehung einer Art animierten Verlaufs führt. Der Nutzer kann einen beliebigen Zustand zu dem aktuellen machen.

Undo-history	
	New Sketch
User4841	Pen
User4841	Pen
User4841	Text
User4841	Pen
User4841	Pen

Abbildung 4.1: Zeichenverlauf (grün markiert ist der aktueller Zustand)

### 4.1.1.3 Scribblar

Scribblar<sup>5</sup> ist ebenfalls eine Plattform für Echtzeitkommunikation und Zusammenarbeit mit Fokus auf Bildung. Neben den Basisfunktionalitäten bietet dieses virtuelle interaktive Whiteboard ein paar spezifische Funktionen. Die WolframAlpha<sup>6</sup>-Integration und andere Tools wie ein Gleichungeditor, ein gemeinsamer Zeiger oder eine Sektion für Unterrichtsmaterialien und Informationen machen Scribblar zu einem mächtigen Werkzeug für fast jede Form von Lernen, Ausarbeiten und Lösen neuer Mathematikprobleme.

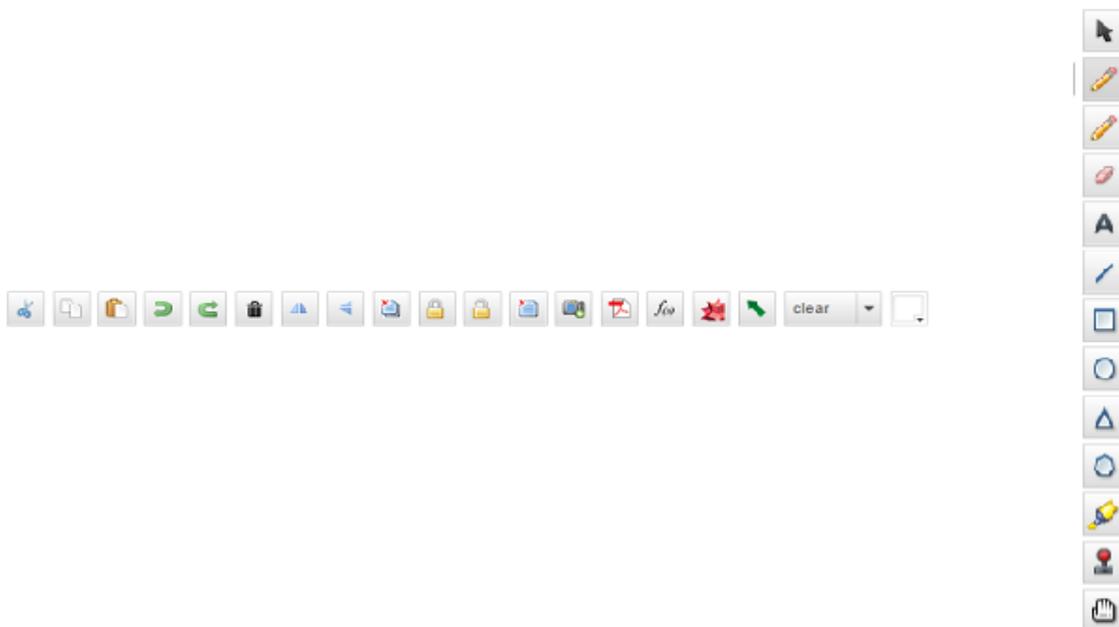


Abbildung 4.2: Tools in Scribblar

<sup>5</sup> <https://scribblar.com/> - letzter erfolgreicher Aufruf am 04.10.2017

<sup>6</sup> WolframAlpha ist ein auf der Software Mathematica basierender Online-Dienst zum Auffinden und Darstellen von Informationen, der von Wolfram Research entwickelt wurde.  
<http://www.wolframalpha.com/> - letzter erfolgreicher Aufruf am 04.10.2017

Für eine einfache und effektive Zusammenarbeit kann sowohl ein Text- als auch ein Audio-Chat benutzt werden. Mit einem Scribblar Pro Konto bekommt der Nutzer Zugang zur API und kann das Whiteboard auf eigenen Seiten einbinden.

#### 4.1.1.4 Zusammenfassung

Im Laufe der Recherche wurden mehrere virtuelle interaktive Online Whiteboards getestet. In der folgenden Zusammenfassung werden die 12 bekanntesten genauer betrachtet. Während der Analyse konnten einige wichtige Kriterien für die Bewertung von V-IWBs definiert werden.

*Farbauswahl:* Diese Funktion bieten alle getesteten V-IWBs an. Für Stift, geometrische Figuren, Text, Hintergrund usw. kann der Nutzer eine Farbe aussuchen. Neben den vordefinierten Farben (4-16 Farben) ermöglichen einige Whiteboards die Auswahl von einer beliebigen Farbe aus dem RGB-Farbraum.

*Größenauswahl:* Hauptsächlich für Stift- und Text-Werkzeuge wird die Möglichkeit angeboten die Größe anzupassen. Deutlich erkennbar sind zwei Darstellungsmuster: Auswahl aus einer festen Anzahl vordefinierter Größen (z.B. Ziteboard<sup>7</sup>) oder in Form eines Sliders (z.B. Realtimeboard<sup>8</sup>).

*Geometrische Primitive:* Die Arbeit mit einem Whiteboard impliziert auch das Erstellen von Diagrammen. Eine der am meisten dafür benutzten Funktionen ist die Einbindung geometrischer Primitive. Die Mehrheit der analysierten V-IWBs unterstützt die Nutzung von von bspw. Kreis-, Rechtecks-, Linien- und Dreiecksformen.

*Zeichenverlauf:* Die Möglichkeit letzte Änderungen rückgängig zu machen, ist eine der wichtigsten Eigenschaften eines interaktiven Whiteboards. Die Recherche hat gezeigt, dass fast alle V-IWBs gut bekannte und intuitiv nachvollziehbare redo/undo-Funktionen zum Einsatz bringen. Die Funktionen werden in Form von Buttons mit nach links für undo bzw. nach rechts für redo zeigenden Pfeilen dargestellt. Unter Umständen kann diese, jedem Computernutzer aus dem alltäglichen Leben bekannte, Funktion ein langes 'Durchklicken' zu gewünschten Zuständen verlangen.

*Bild Upload:* Die absolute Mehrheit von modernen virtuellen Whiteboards unterstützt das Einfügen von Bildern auf ein Board.

*File Upload:* Außer Bildern kann der Nutzer auch externe Dateien wie z.B. PDF, Präsentationen, Dokumente usw. einbinden.

*Chat:* Da die V-IWBs ein Mittel für Kommunikation und Zusammenarbeit sind, spielt das Vorhandensein eines Chats eine große Rolle. Je nach Hersteller und Art des Accounts werden sowohl Text- als auch Audio- und Video-Chats zur Verfügung gestellt.

*Speichern:* Das Exportieren bzw. das Speichern des Inhalts ist eine selbsterklärende Notwen-

<sup>7</sup> <https://ziteboard.com/> - letzter erfolgreicher Aufruf am 04.10.2017

<sup>8</sup> <https://realtimeboard.com/> - letzter erfolgreicher Aufruf am 04.10.2017

digkeit in der Arbeit mit einem Whiteboard. Je nach Anbieter kann der Nutzer diese Funktion sogar im Free-Modus nutzen. Mehr als die Hälfte der V-IWBs erlauben das Herunterladen eines Boards als .png- und/oder .pdf-Datei.

V-IWB	Farbe	Größe	Primitive	Verlauf	Bild Upload	File Upload	Chat	Speichern
AwwApp	Ja	Ja	Ja	Undo/Redo	Ja	.pdf/.ppt	Text/Video	.png/.pdf
CoSketch	Ja	Ja	Ja	Tabelle	Ja	Nein	Text	.png
Scribblar	Ja	Ja	Ja	Undo/Redo	Ja	.pdf/.ppt	Text/Audio	.png/.pdf
Realtimeboard <sup>5</sup>	Ja	Ja	Ja	Undo/Redo	Ja	.pdf/.doc/ Google Drive	Text/Video	Ja
Webwhiteboard <sup>6</sup>	Ja	Ja	Nein	Undo/Redo	Nein	Nein	Nein	.png
Twiddla <sup>7</sup>	Ja	Ja	Ja	Nein	Ja	.pdf/.doc	Text/Audio	Ja
Spacedeck <sup>8</sup>	Ja	Ja	Ja	Nein	Ja	Ja	Text	.pdf/.zip/.txt
Tutorsbox <sup>9</sup>	Ja	Ja	Ja	Undo/Redo	Ja	Ja	Text/Video	.jpg
Whiteboardfox <sup>10</sup>	Ja	Nein	Nein	Undo/Redo	Ja	Nein	Nein	.png
Groupboard <sup>11</sup>	Ja	Ja	Ja	Nein	Ja	Nein	Text	Ja
Ziteboard <sup>12</sup>	Ja	Ja	Ja	Undo/Redo	Ja	.pdf	Text	.png/.pdf/.svg/board
Limnu <sup>13</sup>	Ja	Ja	Nein	Undo/Redo	Ja	Nein	Nein	.png

Tabelle 4.1: Funktionsübersicht von meist bekannten V-IWBs

In der Tabelle 4.1 ist die Tendenz der Bereitstellung von Funktionalitäten deutlich erkennbar. Fast alle der betrachteten V-IWBs bieten ähnliche Funktionen an: Farbe, Größe, geometrische Primitive, Bild Upload und Verlauf gehören zu den Hauptkriterien. Einige Unterschiede sind bei File Upload (sowohl das eigentliche Vorhandensein als auch Art des Files), Chat (nur Text oder Kombination mit Audio/Video) und Speichern zu erkennen.

## 4.1.2 Undo-Modelle

Wie schon im Unterkapitel 4.1.1 diskutiert wurde, basieren die meisten Benutzeroberflächen auf einem **linearen** undo-Modell [Ber94a]. Wenn der Nutzer seinen Fehler rückgängig machen will,

<sup>7</sup> <https://webwhiteboard.com/> - letzter erfolgreicher Aufruf am 04.10.2017

<sup>8</sup> <https://twiddla.com/> - letzter erfolgreicher Aufruf am 04.10.2017

<sup>9</sup> <https://spacedeck.com/> - letzter erfolgreicher Aufruf am 04.10.2017

<sup>10</sup> <https://tutorsbox.com/> - letzter erfolgreicher Aufruf am 04.10.2017

<sup>11</sup> <https://whiteboardfox.com/> - letzter erfolgreicher Aufruf am 04.10.2017

<sup>12</sup> <https://groupboard.com/> - letzter erfolgreicher Aufruf am 04.10.2017

<sup>13</sup> <https://limnu.com/> - letzter erfolgreicher Aufruf am 04.10.2017

kann er nicht zu einem bestimmten Punkt zurück, sondern ihm bleibt nichts anderes übrig solange auf den Button zu klicken bis er den gewünschten Zustand wiederhergestellt hat. Bei einem linearen Modell werden alle ausgeführten Befehle in einer undo-Liste gespeichert. Der letzte Befehl kann rückgängig gemacht werden; er wird dabei aus der undo- in eine redo-Liste verschoben. Der Rückgängig-Vorgang kann solange wiederholt werden, bis die undo-Liste leer ist.

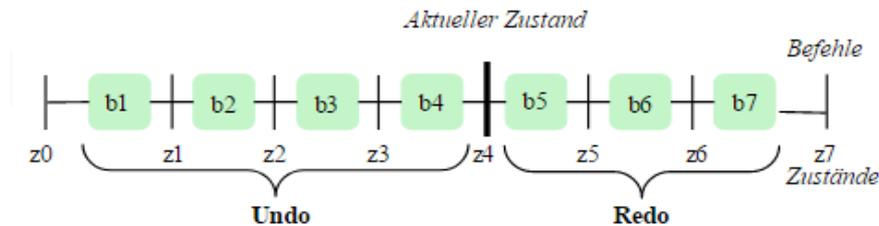


Abbildung 4.3: Ein lineares undo-Modell

Ein **nicht lineares** undo-Modell ist wünschenswert, wenn der Nutzer nicht nur den letzten sondern auch einen beliebigen Befehl aus der undo-Liste rückgängig machen will. Dieser Mechanismus ist besonders für einen grafischen Editor oder Multi-User-Anwendungen sinnvoll, wo Fehler oft auftreten und sie nicht direkt erkennbar sind.

Aus dem nicht-linearen Modell wurden weitere Modelle abgeleitet, wie das *script model* [ACS84], *triadic model* [Thi90] oder das *US&R model*<sup>9</sup> [Yan88]. Alle diese Modelle ignorieren aber das Problem, dass in grafischen Benutzeroberflächen eine vorherige Benutzeraktion keine sinnvolle Interpretation in einem anderen Zustand haben könnte. Der Nutzer hat z.B. einen Kreis auf einem Board gezeichnet und ihn danach verschoben. Das sind zwei eigenständige Aktionen, die der Nutzer rückgängig machen könnte. Wenn er aber ein von oben aufgelisteten Modellen benutzt und die Aktion 'Kreis zeichnen' rückgängig machen will, kann die Aktion 'Kreis verschieben' nicht ausgeführt werden, da der Kreis ja nicht mehr existiert.

Als Lösungsvorschlag definiert Berlage T. in seiner Arbeit ein **direktes selektives undo-Modell** [Ber94a]. Die Grundidee dieses Modells besteht darin, dass statt von der gesamten Befehls-geschichte die Auswirkungen eines Undo-Befehls nur von dem Befehl, der rückgängig gemacht werden soll, und dem aktuellen Zustand der Anwendung abhängig sind. Das direkte selektive Undo-Modell ist eine Erweiterung des linearen Undo-Modells um einen Befehlsbaum<sup>10</sup>. Zusätzlich zum kompletten Funktionsumfang dieser Erweiterung besteht die Möglichkeit, beliebige (selektives Undo unterstützende) Befehle rückgängig zu machen. Befehle, welche selektives Undo unterstützten, müssen in der Lage sein neue Befehle zu erzeugen, die genau die umgekehrten Effekte der ursprünglichen Befehle aufweisen. Bei vielen Befehlen kann dies dadurch erreicht werden, dass die Parameter des entsprechenden Befehls ausgetauscht werden. Diese 'reverse' Befehle werden an das Ende des Befehlsverlaufs gehängt und heben dadurch die Wirkung des ursprünglichen Befehls auf.

<sup>9</sup> Undo, Skip und Redo Modell

<sup>10</sup> Statt einer linearen Liste wird für die Speicherung der Befehls-geschichte ein Baum verwendet. Für einen neuen Befehl wird nach dem aktuellen Zustand ein neuer Zweig in dem Befehlsbaum erzeugt.

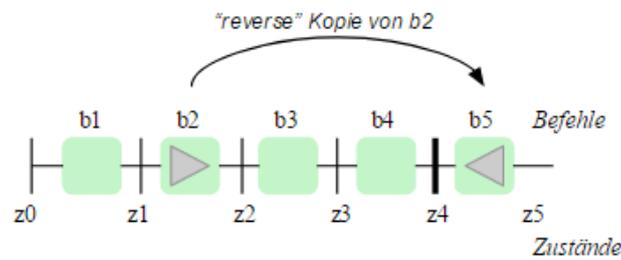


Abbildung 4.4: Beispiel für ein selektives Modell: undo mit einer 'reverse' Kopie

Dank der Nutzung eines baumbasierten linearen undo-Modells werden Standardfälle von einfachen linearen undo- und redo-Funktionen abgedeckt.

### 4.1.3 Visualisierungsmetaphern für den Aktionsverlauf

Der geläufige Begriff '*Metapher*<sup>11</sup>' wird nicht nur von Schriftstellern, sondern auch im alltäglichen Leben, der darin stattfindenden Kommunikation und den Medien oft benutzt. Metaphern erfüllen ihren unmittelbaren Zweck, wenn sie den Rezipienten bekannt sind und korrekt verstanden werden. Diese Eigenschaft soll im Kontext der Mensch-Computer-Interaktion helfen, die Funktionen eines interaktiven Systems aus Benutzersicht geeignet zu benennen, zu gruppieren und z.B. visuell zu repräsentieren [PD10]. Da die Kommunikation zwischen Mensch und Computer nicht nur über explizit ausgetauschte Informationen sondern auch implizit durch Annahmen und Kontextwissen des Benutzers erfolgt, versuchen Nutzer ihre Aktionen in einem unbekanntem System aufgrund ihrer Erfahrung mit anderen Systemen auszuführen. Wichtig ist aber, dass Metaphern nicht übertrieben realistisch nachgebildet sein dürfen und durch Konzepte ergänzt sein müssen.

Metaphern können nicht nur den Umgang mit einem System vermitteln, sondern auch dem Systementwurf dienen. Eingesetzt in der Entwicklungsphase, in der das Zielsystem noch am Entstehen ist, legt eine Metapher eine gewisse Strukturierung dieses Systems nahe, wobei sich diese Struktur, an den den Benutzern bereits vertrauten Konzepten orientiert [PD10]. Obwohl die Entwicklung von Metaphern ein kreativer Prozess ist, kann dieser trotzdem systematisch durchgeführt werden. Laut [Mad94] und [Eri90] hat der metaphorbasierter Entwurfsprozess folgende Hauptbestandteile:

- *Identifikation von Kandidatenmetaphern.* Besonders effektiv ist es, Vorgängersysteme in Bezug auf geeignete Metaphern zu untersuchen oder im Alltagsleben nach Metaphern zu suchen.
- *Bewertung von Metaphern.* Entsprechend dem Einsatzzweck der Metaphern müssen sie hinsichtlich ihrer Bekanntheit und Eignung evaluiert werden.

<sup>11</sup> **Metapher** - (besonders als Stilmittel gebrauchter) sprachlicher Ausdruck, bei dem ein Wort (eine Wortgruppe) aus seinem eigentlichen Bedeutungszusammenhang in einen anderen übertragen wird, ohne dass ein direkter Vergleich die Beziehung zwischen Bezeichnendem und Bezeichnetem verdeutlicht. *Duden - Deutsches Universalwörterbuch, 2015*

- *Entwicklung von Metaphern.* In diesem Schritt wird definiert, welche Aspekte der Metapher und wie nachgebildet werden sollen.

Man kann sich den Verlauf der Aktionen auf einem virtuellen Whiteboard als eine Sequenz von Bildern vorstellen, wo jedes Bild eine vorgenommene Änderung darstellt. Diese Sequenz wird während der Interaktion des Nutzers mit dem System automatisch erstellt. Während der Analysephase wurden einige Kandidatenmetaphern definiert, auf die im Folgenden eingegangen wird.

#### 4.1.3.1 Film-Metapher

Das Grundprinzip der Metaphern im Interface Design ist das gleiche wie in der Sprache und im Denken. Es gibt auch immer eine gewisse Ähnlichkeit zwischen dem realen Objekt einerseits und einer Benutzeroberfläche als Ganzes oder auch einer einzelnen Funktion andererseits. Die in den Abbildungen 4.5 und 4.6 dargestellten Bedienelemente eines Players haben als Quelldomäne entsprechende Elemente von realen Recordern und Playern.



Abbildung 4.5: Interface des Windows Players



Abbildung 4.6: Interface des Youtube Players

Heutzutage werden verschiedene Interface-Arten fürs Abspielen von Videos eingesetzt. Die Mindestanforderung ist aber das Vorhandensein von Play- und Pause-Funktionalitäten. Zusätzlich können Bedienelemente wie Stop, Vor- und Zurückspulen, Ton usw. hinzugefügt werden. Außerdem bieten viele moderne Online-Player die Möglichkeit, sich zu einem bestimmten Zeitpunkt zu navigieren. Dies erfolgt zielgerichtet durch einen Klick auf eine beliebige Stelle der Zeitleiste. Beim Hovern über die Zeitleiste wird eine verkleinerte Ansicht des jeweiligen Zustands in diesem Zeitpunkt angezeigt. Dies sorgt nicht nur für ein einfacheres Auffinden gewünschter Stellen, sondern auch für eine schnellere Wahrnehmung sowie Evaluation des Inhalts.

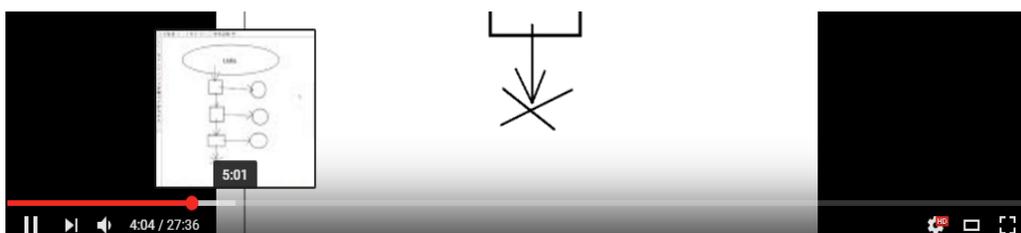


Abbildung 4.7: Youtube Player mit der Zustandsansicht

Eine andere gebräuchliche Art der Darstellung von Bildsequenzen ist ein animiertes GIF<sup>12</sup>. Dabei werden einzelne Bilder zeitverzögert nacheinander abgespielt, was das Darstellen kurzer, film-ähnlicher Dateien ermöglicht. Entwicklern ist es möglich, Kontrollbefehle abzuspeichern. Dies ermöglicht bspw. Verzögerungszeiten zwischen einzelnen Bildern einzufügen, ein GIF zu pausieren oder es abspielen zu lassen.



Abbildung 4.8: Erste 4 Frames eines GIFs

#### 4.1.3.2 Karussell-Metapher

Ein Karussell ist eine Komponente, in der eine Liste mit Elementen (vorzugsweise Bilder) enthalten sind, die sowohl horizontal als auch vertikal dargestellt werden kann. Die Navigation erfolgt in beide Richtungen (links-rechts bzw. oben-unten). Diese Form der Präsentation eignet sich nicht nur für Bildgalerien, auch Nachrichten und andere Inhalte (Tabellen, Flash, Listen, etc.) können auf diese Weise platzsparend und attraktiv dargeboten werden.

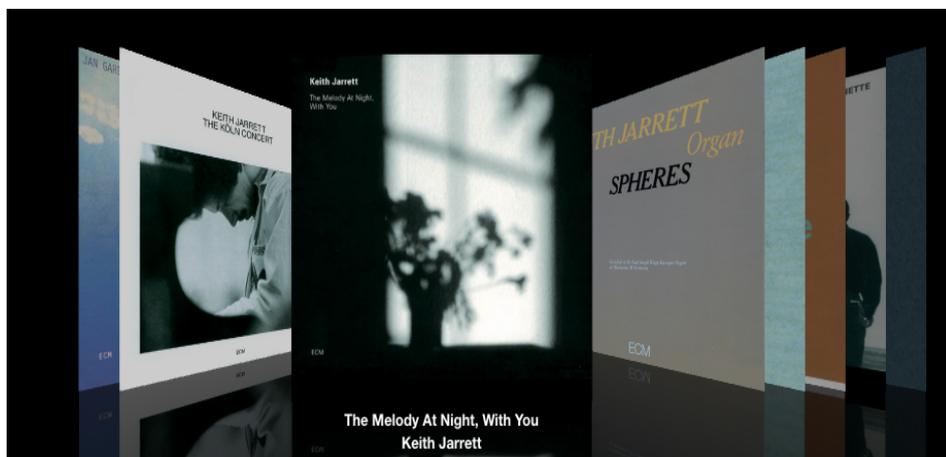


Abbildung 4.9: Karussell in iTunes

Es gibt verschiedene User Interface-Elemente, die man als Karussell bezeichnen kann, z.B. Slider und Slideshow. Das Hauptaufbauprinzip bleibt aber erfüllt: verschiedene Inhalte werden an einer festen Position gezeigt. Diese Inhalte wechseln entweder automatisch oder nach Interaktion. Für das Bild im Fokus gilt, entweder wird nur dieses angezeigt oder die anderen nur verzerrt und unvollständig (s. Abb. 4.9).

<sup>12</sup> Graphics Interchange Format



Abbildung 4.10: Slider mit Pfeilen für die Navigation

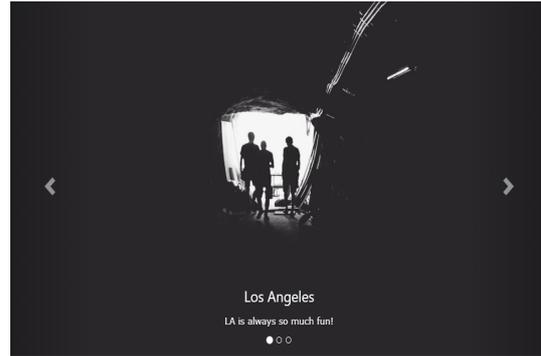


Abbildung 4.11: Slideshow mit Pagination

Slider und Slideshow sind verwandte Konzepte, die auch als eine Karussell-Metapher betrachtet werden können. Der Unterschied besteht darin, dass Slider sich in regelmäßigen Abständen wechselnde Inhalte anzeigen, wohingegen Slideshows nur ein Bild in den Vordergrund setzen. Die Navigation durch Nutzer erfolgt in beiden Fällen entweder durch Swipe-Bewegungen oder Bedienen angebotener Kontrollelemente wie z.B. Pagination und Pfeile. Navigationspfeile werden üblicherweise in Form von verständlichen Richtungszeigern (Klammer oder Pfeil) dargestellt, für die Pagination aber gibt es mehrere Darstellungsvarianten: Zahlen (schrittweise oder aktuelle Seite von der Gesamtanzahl), Richtungspeile oder Indikatoren (Kreise oder Rechtecke).

#### 4.1.3.3 Kartenstapel-Metapher

Eine weitere Metapher, die man teilweise auch als Karussell-Metapher betrachten kann, ist die Kartenstapel-Metapher. Die Wurzeln dieser Metapher liegen in der Vergangenheit, denn sie stammt von HyperCard<sup>13</sup> ab. Der Erfolg der Kartenstapel-Metapher basiert auf ihrem leicht verständlichen Grundprinzip: auf jeder Karte lassen sich verschiedene Sachen darstellen und durch Klicken auf eine Schaltfläche kann von einer Karte zur anderen gesprungen werden. Die Abb. 4.12 ist, vor allem den Android-Nutzern, ein gut bekanntes Beispiel für einen Kartenstapel. Die Karten repräsentieren dabei einzelne Bildschirmseiten oder Apps, die man auswählen oder schließen kann. Wird eine Karte vom Stapel genommen (hier: geschlossen), sieht der Nutzer die darunter liegende Karte vollständig.

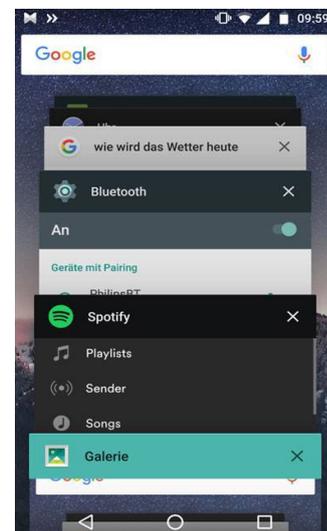


Abbildung 4.12: Android Apps

<sup>13</sup> **HyperCard** ist eine Software für Hypertext mit eigenem Datenformat für eine alte Macintosh-Plattform; entwickelt 1987 für Apple.

#### 4.1.4 Zusammenfassung

In diesem Kapitel wurden bereits existierende virtuelle interaktive Whiteboards vorgestellt und an ausgewählten Beispielen die Hauptfunktionalitäten definiert. Da ein V-IWB ein kreatives Werkzeug für verschiedene Zwecke ist, bedarf es auch bestimmter Möglichkeiten, diese Kreativität zu ermöglichen. Es wurde festgestellt, dass bspw. die *Farbauswahl*-Funktion in allen untersuchten Anwendungen vorhanden war. Unterschiede gab es dabei nur in der Anzahl der wählbaren Farben (vordefinierte Anzahl vs. komplette Palette). Die *Größenauswahl*-Funktion sowie die *geometrischen Primitive* bieten fast alle der betrachteten V-IWBs an. Weitere Funktionalitäten wie z.B. *Bild* und *File Upload*, *Chat* gewinnen heutzutage immer mehr an Bedeutung. Die absolute Mehrheit der in dieser Arbeit untersuchten moderne Tools bringt sie zum Einsatz.

Die wichtigste Funktionalität für diese Arbeit - der *Verlauf* - gehört zu den obligatorischen Merkmalen eines solchen V-IWB. Die Recherche hat gezeigt, dass sie nur mit Hilfe eines linearen undo-Modells repräsentiert wird. Wie im Unterkapitel 4.1.2 bereits diskutiert wurde, ist das Konzept eines nicht lineares undo-Modells für einen grafischen Editor oder Multi-User-Anwendungen sinnvoller. Wichtig ist es hier, dass das folgende Problem berücksichtigt wird. Unter Umständen hat eine vorangegangene Nutzeraktion keine sinnvolle Interpretation in einem Nachfolgezustand, z.B. wie verändern sich die Nachfolgezustände nach dem Löschen einer geometrischen Figur. Dieses Problem könnte ein *direktives selektives undo-Modell*, als eine Erweiterung des linearen Modells, lösen.

In diesem Kapitel wurde der Begriff 'Metapher' eingeführt und Schritte eines metaphorbasiertes Nutzerschnittstellenentwurfs definiert: *Identifikation von Kandidatenmetaphern*, *Bewertung von Metaphern* und *Entwicklung von Metaphern*. Der erste Schritt wurde im Unterkapitel 4.1.3 diskutiert und folgende Kandidatenmetaphern wurden vorgestellt: *Film-*, *Karussell-* und *Kartenstapel-Metapher*. Auf weitere Bestandteile des Entwurfsprozesses wird im Folgenden eingegangen.

## 4.2 NUTZUNGSKONTEXTE

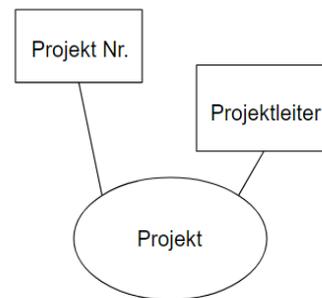
Alle vorgestellten V-IWBs bieten einen großen Spielraum für Kreativität. Das ganze Spektrum an Werkzeugen lässt sich beliebig kombinieren und kann verschiedenen Zwecken dienen: von einfachem Skizzieren bis hin zu komplexen Diagrammen. Je nach Ziel der Nutzung steht oft ein der Werkzeuge im Vordergrund. In dieser Arbeit steht die Verwendung von V-IWBs zu Bildungszwecken im Vordergrund. Für diesen Kontext werden im Folgenden einige der wichtigsten Nutzungskontexte definiert.

**Textlastige Kommunikation.** Einer der nativsten Wege, beliebige Informationen online zu übermitteln ist der Text. Die Text-Funktionalität unterscheidet sich von der Zeichnen-Funktionalität dadurch, dass Nutzer die Eingabe mit Hilfe physischer oder virtueller Tastatur tätigen. Bei der Kommunikation über ein V-IWB dient Texteingabe hauptsächlich den Beschriftungen von Elementen und Kommentaren zu irgendeiner Stelle und wird oft in



Kombination mit anderen Funktionalitäten verwendet. Der Sonderfall der sogenannten 'textlastigen Kommunikation' darf nicht ausgeschlossen werden. Viele Tools für die kollaborative Arbeit stellen keinen Chat zur Verfügung, Kommentare und Meinungs austausch nehmen dann viel Platz in Anspruch. Ein weiteres Beispiel spielt im Bereich der Bildung eine große Rolle: V-IWBs sind ein Mittel Fragen zu stellen und sie zu beantworten, also kann nach einer kurzen Skizze ein Textabschnitt mit der ausführlichen Erklärung folgen.

**Diagrammlastige Kommunikation.** Die geometrischen Primitive, die fast von jedem Whiteboard angeboten werden, eignen sich hervorragend zur Erstellung von Diagrammen und Graphen. Diese Art der Kommunikation wird oft im Bildungsbereich eingesetzt, um bestimmte Inhalte deutlich und schnell darzustellen. Der entscheidende Vorteil von Diagrammen besteht in ihrer Anschaulichkeit und Übersichtlichkeit. Zusätzlich zu den Diagrammen wird Text benutzt, vor allem für die Beschriftungen und Kommentare.



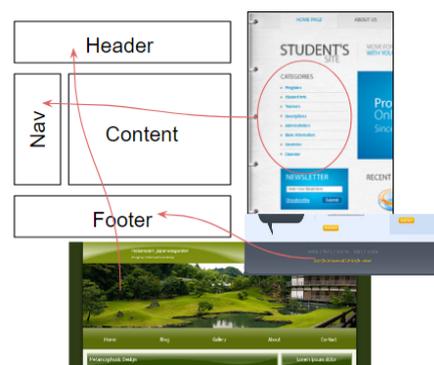
**Skizzenlastige Kommunikation.** Eine der verbreitetsten Einsatzmöglichkeiten eines V-IWBs ist die Erstellung von Skizzen mit dem Stift-Werkzeug. Diese Art der Kommunikation hat Ähnlichkeiten mit der gewöhnlichen Arbeit an einer Tafel, einem klassischen interaktiven Whiteboard oder mit Stift und Papier. Mit einer Skizze sind Konzepte sehr schnell mit hoher Präzision vorgestellt, sofern es sich dabei um Skizzen als eine Art von Ideenpräsentation handelt. Besonders hilfreich sind Skizzen im Bildungsbereich für die Formelumsetzung oder Kommentare direkt auf bereits vorhandenem Inhalt.

$$\textcircled{1} \lim_{x \rightarrow 0} \frac{\sin(x)}{x} = \frac{0}{0}$$

$$\lim_{x \rightarrow 0} \frac{\cos(x)}{1} = \frac{\cos(0)}{1}$$

$$= 1$$

**Bildlastige Kommunikation.** Ein wertvolles, visuelles Kommunikationsmittel sind Bilder. Diese werden schneller erfasst und übermitteln schneller inhaltliche Botschaften als andere Kommunikationsmittel. Außerdem werden Bilder als eine Einheit verstanden und können gedanklich einfacher analysiert und verarbeitet werden. In einem V-IWB werden Bilder oft entweder direkt (normale Screenshots oder Code-Snippets) oder in Kombination mit Zeichenwerkzeugen (Markierung, Unterstreichen, Kommentieren) eingesetzt.



**Dokumentlastige Kommunikation.** Analog zu *Bild Upload* unterstützen viele moderne online Whiteboards auch den *File Upload*. Dieser versetzt Nutzer in die Lage in hochgeladenen Dokumenten bestimmte Stellen zu markieren, eigene Gedanken zu annotieren oder Kommentare zur Verbesserung hinzuzufügen. Wie bei der bildlastigen Kommunikation, werden Dokumente oft in Kombination mit anderen Werkzeugen verwendet.

Im Bildungsbereich könnte als Beispiel die Besprechung eines Tests dienen, bei der eine Person A den ausgefüllten Test der Person B korrigiert. Dabei dienen bspw. markierte Fehler und nützliche Hinweise, in Form von Kommentaren, als Grundlage für die Nachbereitung bzw. Vertiefung des Erlernten der Person B.

zuvor gegebenen Antworten angezeigt werden. Laut [BKKS15] können auf diese Weise die verschiedenen Nachrichtentypen übermittelt werden. Je nach Präferenz (z.B. Lernzielorientierung, Leistungszielorientierung, Prüfungsvorbereitung oder Interesse am Thema) werden durch *metakognitive Prompts* strategische Hinweise für die Vorbereitung und Nachbereitung des Vorlesungsstoffs gegeben. Studierende, die bei Lernaufgaben am Beginn der Vorlesung Fehler gemacht haben, können zu einem späteren Zeitpunkt durch *kognitive Prompts* mitbekommen, wenn das in der Lernaufgabe behandelte Thema noch einmal wiederholt wird. Auch die Bereitstellung von weiterführenden Materialien und Links wird durch *normale Prompts* unterstützt.

### 3.2 GRAPHICUSS

Am Lehrstuhl Rechnernetze der Technischen Universität Dresden wurde eine zukünftige Komponente von AMCS in Form von Graphicuss bereits vorbereitet. Graphicuss stellt eine Kombination der Aspekte von klassischen Forensystemen mit denen von virtuellen interaktiven Whiteboards dar. Als ein grafikbasiertes Discussionssystem für Lehrveranstaltungen besitzt die Komponente laut [Che16] drei Basisfunktionalitäten:

**Kursverwaltung (course management)** Der als Lehrender identifizierter Nutzer kann Kurse erstellen und die von ihm erstellten Kurse verwalten. Bei der Erstellung kann dem Kurs ein Name gegeben und ein Bild als Hintergrund für eine bessere Erkennung hochgeladen werden. Außerdem kann der Lehrende eine konkrete Beschreibung für den Kurs hinzufügen. Ein neuer Kurs bekommt automatisch einen eindeutigen Identifikationscode mit dessen Hilfe er von Teilnehmern gefunden werden kann. Wenn ein Student sich für einen Kurs interessiert, hat er die Möglichkeit diesen Kurs zu seiner Favoritenliste hinzuzufügen. Das ermöglicht schnelleres Auffinden und Zugreifen auf relevante Veranstaltungen.

Es muss beachtet werden, dass alle der definierten Nutzungskontexte nicht nur im reinen, d.h. alleine, ohne Einflüsse anderer Kontexte, Zustand benutzt werden. Selbstverständlich lassen sich mehrere Funktionalitäten eines V-IWBs beliebig kombinieren.



# 5 KONZEPTION

In diesem Kapitel wird der Prozess der Nutzerschnittstellenentwicklung erläutert. Dabei wird auf den Designprozess als eine der wesentlichen Phasen der Entwicklung eingegangen. Im Folgenden werden die Kandidatenmetaphern in Bezug auf Nutzungskontexte bewertet und mittels Sketching und Prototyping mögliche Nutzerschnittstellen definiert, die im weiteren Kapitel durch potentielle Endnutzer evaluiert werden.

## 5.1 EINLEITUNG

In der Arbeit von Preim B. und Dachsel R. über User Interface Engineering (UIE) werden vier Hauptphasen eines UIE-Prozesses definiert: *Analyse-, Design-, Implementierungs- und Evaluationsphase* [PD15]. Alle Phasen sind nicht-hierarchisch, d.h. sie müssen nicht konsequent durchgeführt werden. Dennoch ist eine gewisse Strukturierung des Prozesses sinnvoll. In den bisherigen Kapiteln dieser Arbeit wurden folgende Bestandteile der *Analysephase* behandelt: der aktuelle Stand der Entwicklung und verwandte Konzepte wurden analysiert und Anforderungen an die zu entwickelnde Nutzerschnittstelle des Systems wurden definiert. Die Liste der Anforderungen dient als Ausgangspunkt für den Entwurf geeigneter Lösungen für die Nutzerschnittstelle während der *Designphase*. Für die erfolgreiche Entwicklung einer Nutzerschnittstelle ist das Vorliegen mehrerer Entwürfe wichtig. Da auf deren Basis Vor- und Nachteile besser herausgearbeitet werden können. Nach der Designphase folgt die Umsetzung der entworfenen Lösungen in der *Implementierungsphase* (nicht Teil dieser Arbeit). Die *Evaluation* spielt eine große Rolle für den Entwicklungsprozess, da diese nicht nur als Hauptphase nach der Implementierungsphase auftritt, sondern Bestandteil jeder anderen Phase ist.

## 5.2 NUTZUNGSKONTEXTE IN GRAPHICUSS

In dem Kapitel 3.2 wurde bereits die Komponente des Auditorium Mobile Classroom Services (AMCS) - *Graphicuss* - vorgestellt. Für diese Arbeit ist das canvas-basierte drawing tool von Bedeutung, da es eine Art von V-IWB ist. Basierend auf der Kriterienzusammenfassung untersuchter online Whiteboards (s. Tabelle 4.1) lassen sich die Funktionalitäten von Graphicuss dementsprechend klassifizieren.

V-IWB	Farbe	Größe	Primitive	Verlauf	Bild Upload	File Upload	Chat	Speichern
Graphicuss	Ja	Ja	Ja	Undo/Redo	Nein	Nein	Nein	Nein

Tabelle 5.1: Funktionsübersicht von Graphicuss

Wie in der Tabelle 5.1 zu sehen ist, verfügt Graphicuss über die Basisfunktionalitäten eines V-IWBs: wie bspw. geometrische Primitive (Kreis, Rechteck, Dreieck), Farbauswahl, Größenauswahl und Verlauf.

Die definierten Nutzungskontexte (s. Kapitel 4.2) werden in der Form, in der sie innerhalb eines V-IWBs auftreten, um einiges ergänzt oder gar verändert. Wie es schon zuvor deutlich gemacht wurde, vereint Graphicuss Aspekte virtueller interaktiver online Whiteboards mit denen klassischer Forensysteme. Ausgehend von Nutzungskontexten in einem V-IWB und Eigenschaften eines Forums lassen sich folgende Beitragsarten mit einem bestimmten Schwerpunkt ableiten.

**Text.** Die textlastige Kommunikation tritt in Graphicuss auf eine andere Weise auf. Hier ist die wichtigste Eigenschaft eines Forums zu sehen. Ein Beitrag, sowohl eine Frage als auch eine Antwort, kann als Text geschrieben werden. In Diskussionsforen ist es notwendig, sich beim Antworten auf andere Diskussionsteilnehmer zu beziehen, indem man das Geschriebene zitiert. Unter 'gutem Stil' einer Diskussion, im Sinne einer zwischenmenschlichen Kommunikation, versteht sich das hinreichende Zitieren nur notwendiger und relevanter Stellen. Viele Teilnehmer stört es, wenn mehr als notwendig zitiert wird oder das Zitat als solches nicht kenntlich gemacht wurde. Die kleinste semantisch sinnvolle Informationseinheit für Texte ist ein Wort.

**Diagramm/Graph.** Aufgrund ihrer Anschaulichkeit eignen sich Diagramme bzw. Graphen zur Darstellung von Sachverhalten. Im Bildungsbereich werden sie oft zur Erklärung von Relationen und Strukturen verwendet. Außerdem sind viele Themengebiete nur in dieser Form darstellbar. So ist beispielsweise ein UML-Diagramm nur schwer mit Text beschreibbar. In Graphicuss werden für Diagramme/Graphen einige geometrische Primitive zur Verfügung gestellt. Die Diagrammerstellung ist im Gegensatz zu Bild Upload ein dynamischer Prozess, da Elemente zeitversetzt hinzugefügt werden. Als die kleinste Informationseinheit gilt hierbei häufig ein fertiges Primitiv. Für mögliche Beschriftungen wird analog zum 'Text' ein Wort als Informationseinheit genommen.

**Skizze: Zeichnung.** Skizzenlastige Kommunikation wurde bereits anhand von V-IWBs diskutiert. In der vorliegenden Arbeit wird zwischen Skizzen im Sinne von Zeichnungen und Skizzen im Sinne von Formeln unterschieden. Beide Arten werden identisch realisiert: Freihandzeichnen mit dem Stift-Werkzeug. Der ausschlaggebende Unterschied für diese Untersuchung liegt in der Definition

der kleinsten Informationseinheit. Häufig wird ein Element, welches innerhalb einer Interaktion (bspw. Gedrückthalten der Maustaste) mit dem Canvas entsteht, als kleinste Informationseinheit betrachtet. Für Zeichen-Skizzen erscheint dies, im Gegensatz zu Formel-Skizzen (s. Abschnitt 'Skizze: Formel'), sinnvoll.

**Skizze: Formel.** Analog zu den Zeichen-Skizzen, können auch Formeln in ihrer Entstehung als Skizzen betrachtet werden. Semantisch wichtig ist hier allerdings nicht die einzelne Linie, sondern eine bestimmte Zusammensetzung einzelner Linien (z.B. besteht ein Plus-Zeichen aus zwei sich kreuzenden Linien). Um hierbei unnötige Informationen (eine nicht interpretierbare Abfolge einzelner Striche im Verlauf) zu vermeiden, wird deren sinnvolle (im mathematischen Sinne) Zusammensetzung als kleinste Informationseinheit betrachtet. Damit wäre im Verlauf bezüglich der Skizzenerstellung eines Plus-Zeichens keine Information über zwei einzelne Strichzüge aufzufinden, sondern nur eine Operation, die für das Hinzufügen eines Zeichens steht.

**Bild.** Ein Beitrag, der nur ein Bild enthält, ist, im Gegensatz zu zuvor erwähnten Diagrammen, statischer Natur und beinhaltet keinen zeitlichen Ablauf der Darstellung. Die kleinste Informationseinheit ist das Bild selbst. Diese Art der Kommunikation ist *nicht relevant* für diese Arbeit.

**Kombinationen.** Eine Diskussion bedarf Kreativität für eine bessere Argumentation und verständliche Erklärung. Die breite Werkzeugpalette des Graphicuss unterstützt dabei. Die Freiheit die angebotenen Ausdrucksmittel frei zu verwenden und beliebig einzusetzen liefert viele Kombinationen, z.B. ein Bild kann mit einem Text und einem skizzierten Pfeil ergänzt werden, wenn der Verfasser die anderen Diskussionsteilnehmer auf bestimmte Details aufmerksam machen will. Die kleinsten Informationseinheiten sind in Fällen der Kombination von mehreren Werkzeugen die entsprechenden Einheiten jedes einzelnen Werkzeuges.

## 5.3 BEWERTUNG VON KANDIDATENMETAPHERN IN BEZUG AUF NUTZUNGSKONTEXTE

Ein wichtiger Aspekt des *Designprozesses* ist die Auswahl und konkrete Nutzung von Metaphern [PD15]. Im Unterkapitel 4.1.3 wurden bereits einige Kandidatenmetaphern vorgestellt. Die Bewertung von Metaphern hängt davon ab, ob und wie gut sie den Nutzern vertraut sind, welche Assoziationen die Nutzer haben und wie gut diese Assoziationen mit dem zu entwickelnden System übereinstimmen.

Die Rolle der entwicklungsbegleitenden Evaluation wurde bereits am Anfang dieses Kapitels verdeutlicht. Denn es ist sinnvoll potentielle Endnutzer zu fragen, welche Assoziationen, Vorstellungen und Ideen bezüglich Kandidatenmetaphern sie für das System haben. Zur Evaluation der für diese Arbeit in Frage kommenden Kandidaten-Metaphern wurde die sogenannte 'blank paper'-Methode eingesetzt. Bei dieser Methode werden die Teilnehmer nach einer Einleitung und Erklärung der Aufgabe gebeten, ihre Umsetzungsideen auf einem leeren Blatt Papier festzuhalten (s. Anhang A.1). Für das Thema dieser Arbeit wurden fünf Teilnehmer gebeten, ihre Ideen zur Umsetzung des Verlaufs in Graphicuss zu skizzieren. Im folgenden Abschnitt wird der genauere Ablauf der Methode beschrieben.

Als erstes wurden die Teilnehmer gebeten ein Diagramm zu skizzieren und sich dabei den zeitlichen Verlauf vorzustellen. Vier von fünf Probanden entschieden sich dabei für die Karussell-Metapher in Form eines 'Sliders'. Jeder Zustand sollte durch ein Vorschaubild mit einem Zeitstempel und evtl. einer kurzen Änderungsbeschreibung dargestellt werden. Da die kleinste Informationseinheit ein Primitiv oder die vorgenommene Änderung (Farbe, Größe, Position) ist, entstehen dabei ständig neue Bilder des Diagramms. Unterschiede zwischen mehreren Zuständen sind anhand kleiner Vorschaubilder gut nachvollziehbar. Nicht nur Slider-UI, als Beispiel für eine Karussell-Metapher, sondern auch *Slideshow*- (von einer Testperson vorgeschlagen) und *Karussell-UI* können als geeignete Lösungsansätze für die diagrammlastige Kommunikation in Betracht gezogen werden.

Eine der Hauptcharakteristiken einer Film-Metapher ist die dynamische Darstellung von Bildsequenzen. Diese Darstellung entspricht dem Verlauf der Erstellung eines Diagramms im Kontext dieser Arbeit. Sowohl die Film-Metapher als auch die Kartenstapel-Metapher (aufgrund der Ähnlichkeit zur Karussell-Metapher) können als geeignete Kandidaten für Prototyping und Evaluation betrachtet werden. Die Abbildungen 5.1-5.5 stellen einige initiale Skizzen zur Darstellung von geeigneten Kandidatenmetaphern im Nutzungskontext 'Diagramm/Graph' dar. Ein *Slider-UI* stellt eine Abfolge von Zuständen so dar, dass fünf Zustände gleichzeitig für den Nutzer sichtbar sind und die Navigation horizontal erfolgt (s. Abb. 5.1). Ein *VideoPlayer*- und ein *Slideshow-UI* weisen gewisse Ähnlichkeiten auf: nur ein Zustand ist sichtbar, die Navigation aber erfolgt auf verschiedene Weisen. Bei dem *VideoPlayer-UI* dient eine Leiste mit Buttons der Navigation, bei dem *Slideshow-UI* werden Richtungspfeile als Navigationselemente benutzt (s. Abb. 5.2, 5.3). Ein *Karussell*- und ein *Kartenstapel-UI* (Abb. 5.4, 5.5) sind ebenfalls ähnlich aufgebaut. Es ist jeweils nur ein Zustand für den Nutzer komplett sichtbar, wohingegen andere Zustände nur partiell abgebildet sind (perspektivische Verzerrung beim *Karussell*- und Überdeckung beim *Kartenstapel-UI*).

Die entstandenen Skizzen werden als Grundlage für spätere Diskussionen genutzt, um Vor- und Nachteile verschiedener Varianten zu erörtern [PD15]. Sie dienen weiterhin der Erstellung eines Prototypes, basierend auf anschaulich dargestellten Elementen und Möglichkeiten der Umsetzung. Da sich alle Kandidatenmetaphern für die Erstellung von Diagrammen und Graphen eignen, werden alle anderen Nutzungskontexte anhand der oben abgebildeten Skizzen (s. Abb. 5.1 - 5.5) bewertet. Eine Freichandzeichnung innerhalb eines grafikbasierten Editors stellt letztendlich auch eine Art von Bild dar. Alle Probanden der 'blank paper'-Methode haben auf die konkretisierende Frage bezüglich der Visualisierung des Zeichenverlaufs für eine skizzenlastige Kommunikation geantwortet, dass die auf der Bilderabfolge basierende Darstellungsform sich auch für diesen Nutzungskontext eignet. Man kann also davon ausgehen, dass alle der definierten Kandidatenmetaphern hier auch als geeignet betrachtet werden können.

Der Nutzungskontext 'Text' weist keine Eigenschaften einer Bildfolge auf und musste deshalb mit den Probanden näher diskutiert werden. Alle Testpersonen haben darauf hingewiesen, dass bei textlastiger Kommunikation zusätzliche Informationen in Form eines Zeitstempels eine obligatorische Eigenschaft der Visualisierung sein muss.

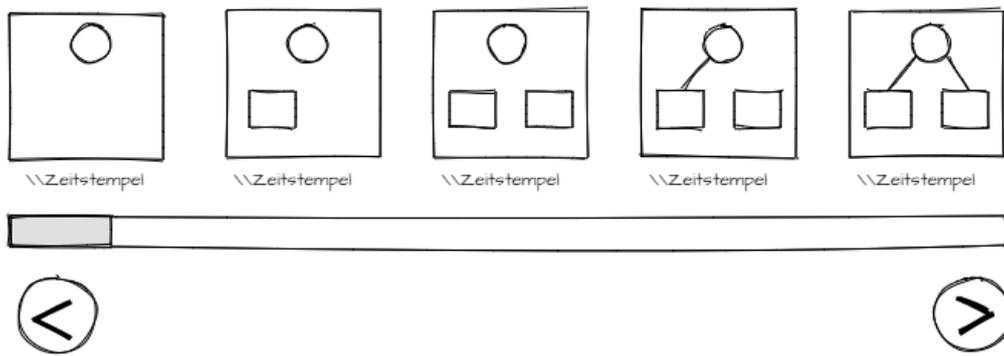


Abbildung 5.1: Skizze zum Slider-UI mit 2 Navigationsvarianten

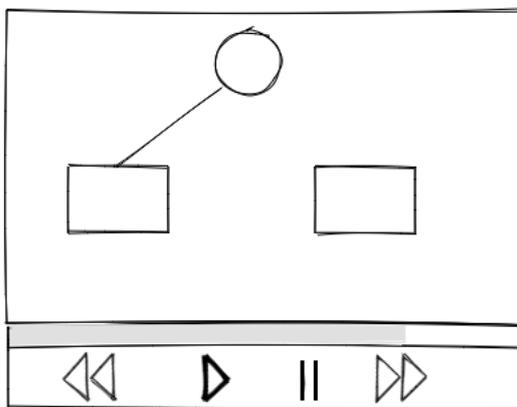


Abbildung 5.2: Skizze zum VideoPlayer-UI

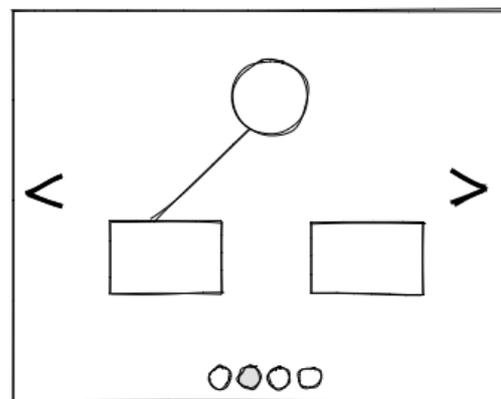


Abbildung 5.3: Skizze zum Slideshow-UI

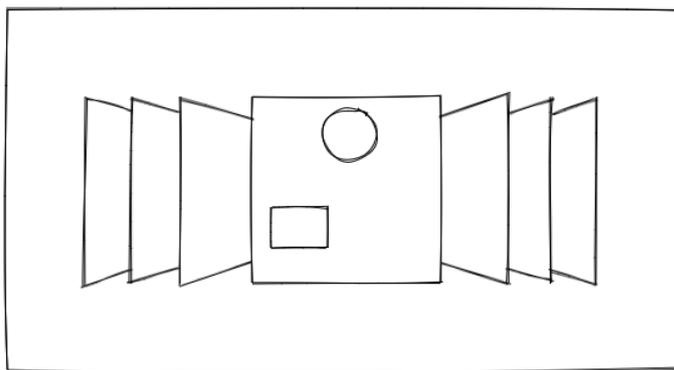


Abbildung 5.4: Skizze zum Karussell-UI



Abbildung 5.5: Skizze zum Kartenstapel-UI

Im Kapitel 3.3 wurden bereits Anforderungen an die Nutzerschnittstelle definiert. Für den Entwicklungsprozess ist es wichtig, dass das zu entwickelnde System letztendlich die gestellten

Anforderungen erfüllt. Im Folgenden werden die Kandidatenmetaphern basierend auf Systemanforderungen (Muss- und Kann-Anforderungen) bewertet.

**MA-1.** *Die Nutzerschnittstelle muss mittels verständlicher Metaphern dargestellt werden.* Die Frage der Verständlichkeit von Metaphern muss den potentiellen Nutzern gestellt werden. In dieser Arbeit wird davon ausgegangen, dass alle hier betrachteten Metaphern bekannt und verständlich sind, da es sich um Assoziationen aus dem mentalen Modell eines modernen Nutzers handelt.

**MA-2.** *Die Nutzerschnittstelle muss unterschiedliche Nutzungskontexte berücksichtigen.* Wie bereits zuvor erwähnt wurde, eignen sich alle der vorgestellten Metaphern zur Darstellung von Bildsequenzen. Nutzungskontexte wie 'Diagramm/Graph', 'Skizze: Zeichnung' und 'Skizze: Formel' werden also berücksichtigt.

**MA-3.** *Die Nutzerschnittstelle muss den zeitlichen Verlauf und dessen Richtung (z.B. Zeitstempel) visualisieren.* Vier von fünf Kandidatenmetaphern erfüllen diese Anforderung unter der Voraussetzung, dass als Richtung die Leserichtung (von links nach rechts) und/oder Zeitstempel als Hilfsmittel zur Richtungsbestimmung betrachtet wird. Die gerichteten Icons und Buttons des VideoPlayers sind selbsterklärend und müssen nicht ergänzt werden.

**MA-4.** *Die Nutzerschnittstelle muss die Sichtbarkeit des aktuellen Zustandes im Verlauf gewährleisten.* Die Sichtbarkeit gewährleisten alle der untersuchten Kandidatenmetaphern. In einem Slider-UI liegen mehrere Zustände gleichzeitig im Fokus des Nutzers und es ist schwer zu unterscheiden, welcher zur Zeit der aktuelle ist. Diese Design-Variante muss also ergänzt werden (z.B. Rahmen oder Hervorhebung des aktuellen Zustandes).

**MA-5.** *Die Nutzerschnittstelle muss die Möglichkeit anbieten, längere Verläufe in größeren Schrittweiten durchzulaufen.* Im Fall eines langen Verlaufs muss der Nutzer die Möglichkeit haben sich zu einem späteren Zustand zu navigieren ohne jeden Zustand durchklicken zu müssen. So wird das bspw. bei Kartenstapel-, Karussell- und Slideshow-UIs überhaupt nicht und bei VideoPlayer- und Slider-UIs nur bedingt angeboten. Die Voraussetzung ist hier das Hinzufügen bei dem Designen und Implementieren von Navigationselementen, die das ermöglichen (z.B. Vorspulbutton mit fünf Frames pro Schritt für VideoPlayer).

**KA-1.** *Die Nutzerschnittstelle muss responsiv sein.* Da die Responsivität nicht Teil dieser Phase ist und erst bei der eigentlichen Entwicklung beachtet und umgesetzt werden muss, ist sie in dieser Phase der Untersuchung für die Bewertung von Kandidatenmetaphern nicht relevant.

**KA-2.** *Die Nutzerschnittstelle muss große und übersichtliche Icons verwenden.* Diese Anforderung bezieht sich auf Prototyping- und Entwurfsphase. Alle dargestellten UI-Skizzen verfügen über eine geringe Anzahl von Icons. Eine maßgebliche Entscheidung über Größe und Darstellung wird erst in einer späteren Phase der Schnittstellen-Entwicklung getroffen.

**KA-3.** *Die Nutzerschnittstelle muss an unterschiedliche Endgeräte anpassbare Bedienelemente verwenden.* Die Bildschirmgröße eines PCs ist natürlich der eines Smartphone nicht identisch. Elemente, die auf einem PC einfach zu bedienen sind, können auf einem Smartphone schwer zu bedienen sein. Ein klassisches Beispiel dafür ist die anpassbare Navigation, die auf einem Smartphone zu der ausklappbaren 'drei-Balken-Navigation' wird. Der Entwickler muss also Eigenschaften von Endgeräten berücksichtigen und die Bedienelemente dementsprechend anpassen.

**KA-4.** *Die Nutzerschnittstelle muss unterschiedliche Eingabemodalitäten berücksichtigen.* Da Graphicuss sowohl auf einem PC/Laptop als auch auf einem Smartphone/Tablet bedienbar sein soll, muss die Nutzerschnittstelle entsprechende Eingabemodalitäten berücksichtigen. So kann man Karussell- und Kartenstapel-UIs aufgrund ihrer Swipe-orientierter Bedienbarkeit ausschließen.

Slider- und Slideshow-UIs lassen sich für beide Gerätearten anpassen, da sie sich sowohl mit Maus-Tastatur-Eingabe als auch mit Finger-Eingabe bedienen lassen. Bei der Entwicklung eines VideoPlayer-UIs muss die Anforderung KA-2 als Voraussetzung erfüllt werden, indem man Buttons und andere Bedienelemente groß und übersichtlich genug für die Finger-Eingabe konzipiert. **KA-5.** Die Nutzerschnittstelle muss bei auftretenden Fehlern passende Fehlermeldungen ausgeben. Da die Fehlermeldungen nicht Teil der diskutierten Schnittstelle sind, sind sie in dieser Phase der Untersuchung für die Bewertung von Kandidatenmetaphern nicht relevant.

Die Tabelle 5.2 stellt die Ergebnisse der Bewertung von Kandidatenmetaphern bezüglich Anforderungen an System zusammenfassend vor.

Anforderung	Slider	VideoPlayer	Slideshow	Karussell	Kartenstapel
MA-1	+	+	+	+	+
MA-2	++	++	++	++	++
MA-3	+	++	+	+	+
MA-4	+	++	++	++	++
MA-5	+	+	-	-	-
KA-1	n.r.	n.r.	n.r.	n.r.	n.r.
KA-2	+	+	+	+	+
KA-3	++	++	++	+	+
KA-4	++	+	++	-	-
KA-5	n.r.	n.r.	n.r.	n.r.	n.r.
Gesamt	3 (++)	4 (++)	4 (++)	2 (++)	2 (++)
	5 (+)	4 (+)	3 (+)	4 (+)	4 (+)
	0 (-)	0 (-)	1 (-)	2 (-)	2 (-)

Legende: ++ = ja, + = ja, unter der Voraussetzung, - = nein, n.r. = nicht relevant

Tabelle 5.2: Bewertung von Nutzungskontexten bezüglich Anforderungen

Aus der Tabelle folgt, dass es keine eindeutige Lösung existiert. Alle Nutzerschnittstellenentwürfe erfüllen mehr oder weniger alle bzw. die Mehrheit der gestellten Anforderungen. Die Karussell- und Kartenstapel-UIs lassen sich ausschließen, aufgrund der Gesamtbewertung (sie erfüllen zwei Anforderungen absolut, vier unter der Voraussetzung und zwei Anforderungen gar nicht). Basierend auf der Auswertung der Tabelle wurde die Entscheidung getroffen, im Rahmen dieser Arbeit die Slider-, VideoPlayer- und Slideshow-UIs für eine Evaluation mit Endnutzern heranzuziehen. Für diesen Zweck sollen entsprechende Papier-Prototypen (Bestandteile werden skizziert und beschriftet) den Umgang mit der Schnittstelle simulieren. Ein Vorteil bei der Verwendung von Papier-Prototypen besteht darin, dass vertraute Werkzeuge genutzt werden können und somit gewährleistet wird, dass sich sowohl Entwickler als auch Testperson voll auf Design und Konzept des Prototypen konzentrieren können (s. Anhang A.2).

Die in diesem Kapitel durchgeführte Bewertung reicht bei weitem nicht für einen erfolgreichen Schnittstellenentwurf. Eine umfassende Bewertung würde den zeitlichen Rahmen dieser Arbeit sprengen und ist daher nicht zielführend. Vielmehr wird der Fokus auf einen anderen wichtigen Aspekt nutzerbasierter Entwicklung - *die Evaluation der Zwischenergebnisse durch potentielle Endnutzer* - gelegt. Das soll Thema des nächsten Kapitels sein.



# 6 EVALUATION

In diesem Kapitel werden sowohl der Aufbau als auch die Ergebnisse der durchgeführten Evaluation dargestellt. Gegenstand dieser Evaluation waren die während der Designphase entstandenen Lösungsansätze und Design-Varianten.

## 6.1 EINFÜHRUNG IN DIE EVALUATIONSPHASE

Am Ende der Designphase wurden drei Nutzerschnittstellenentwürfe als mögliche Lösungen identifiziert. Um eine Entscheidung zu treffen, welche der Design-Varianten zum Entwurf der Nutzerschnittstelle dienen soll, wurde eine empirische (erfahrungsbezogene) Evaluation durchgeführt [PD10]. Im Sinne einer benutzerzentrierten Entwicklung wurden die entstehenden Prototypen mit Endnutzern getestet.

Während der Evaluationsvorbereitung war es wichtig passende Aufgaben auszuwählen, die dazu dienten den Test zu fokussieren, Schwachstellen zu finden oder quantitative Werte zu erheben. Die Definition der Aufgaben wurde auf den aktuellen Stand der Entwicklung angepasst und beinhaltete nur das Testen des zeitlichen Verlaufs und dessen Darstellung. Das fertige Produkt musste am Ende der Entwicklung alle bereits definierten Nutzungskontexte berücksichtigen. Die Testaufgabe definierte also ein Anwendungsfall, bei dem eine Kombination von Nutzungskontexten stattfindet.

**Testaufgabe:** Während einer Prüfungsvorbereitung nutzt ihr und eure Kollegen AMCS, um Fragen zu klären. Gestern hat ein Student einen Beitrag mit einer Frage gepostet. Heute früh habt ihr gesehen, dass jemand die Frage beantwortet hat. Die Antwort wurde mit Hilfe von Graphicuss erstellt und beinhaltet sowohl geometrische Primitive als auch Text und Formeln. Ihr habt einen Fehler in dieser Antwort gefunden und wollt ihn korrigieren ohne alles neu zeichnen zu müssen. Ihr wisst, dass Graphicuss die Möglichkeit anbietet, den Zeichenverlauf eines erstellten Beitrags zu verfolgen und einen gewissen Zustand zu zitieren, um diesen

als Ausgangspunkt euer Korrekturen zu verwenden. Eure Aufgabe besteht darin, die Nutzerschnittstelle zu bedienen und einen beliebigen Zustand auszusuchen. Während des Versuchs ist es wünschenswert, laut zu denken und eure Handlungen zu kommentieren. Nach dem Versuch werden anschließend ein paar Fragen gestellt.

*Anmerkung:* Mögliche Fragen bezüglich der Funktionsweise von AMCS und Graphicuss werden vom Moderator beantwortet.

Nach Abschluss der Vorbereitung wurde die Rekrutierung der Testpersonen durchgeführt. Da AMCS und Graphicuss in erster Linie für Studenten entwickelt worden sind, sollte folglich auch die Evaluation auf diese Personengruppe beschränkt sein. An der Bewertung der Ergebnisse haben 15 Testpersonen teilgenommen, unter ihnen (Medien-)Informatik-, Psychologie-, Chemie-, Elektrotechnik- und Wirtschaftsstudenten. Das Einbeziehen von Studenten anderer Fachrichtungen, neben der Informatik bzw. Medieninformatik, in den Entwicklungsprozess führt zu einem breiteren Spektrum von Erfahrungen und Einsichten.

Ein weiterer Schritt der Vorbereitung der Evaluation war die Erstellung eines Testplans. An diesem Punkt musste geklärt werden, in welcher Form und mit welchen Hilfsmitteln die Evaluation durchgeführt wird. Wie schon zuvor erwähnt wurde, wurde die Evaluation mittels eines Papier-Prototyps und der Wizard-of-Oz-Methode<sup>14</sup> durchgeführt, um die Auswahl zwischen drei Design-Varianten zu treffen (s. Anhang A.3). Vor der Durchführung wurden die Testpersonen gebeten, 'laut' zu denken und alle Handlungen dahingehend zu kommentieren, dass der Moderator zu jedem Zeitpunkt über Stand und Absicht der Testperson informiert ist. Das steigert die Effektivität von Beobachtungen, die bei der Evaluation stattfinden. Abschließend wurden die Teilnehmer mündlich befragt, um durch mehr oder weniger präzise Fragen gezielt Informationen zu erhalten.

## 6.2 DURCHFÜHRUNG DER EVALUATION

Die Evaluation wurde in zwei Iterationen durchgeführt. Beide Durchführungen waren identisch aufgebaut: Einführung, Aufgabenstellung, Aufgabenerfüllung und freies Interview. Die während der Aufgabenerfüllung gegebenen Kommentare wurden protokolliert und die Interviews wurden für eine bessere Auswertung aufgezeichnet. Selbstverständlich wurden alle Teilnehmer im Vorfeld darüber informiert und gaben ihre Zustimmung. Die Fragen, die während des freien Interviews gestellt wurden, ergaben sich hauptsächlich aus dem Gespräch und Kommentaren. Zusätzlich wurden folgende vordefinierte Fragen jedem Probanden gestellt:

**F-1.** *Ist die Darstellungsform und die zugrunde liegende Metapher passend und verständlich?*

**F-2.** *Sind die Icons verständlich?*

**F-3.** *Ist die Richtung des zeitlichen Verlaufs intuitiv nachvollziehbar?*

**F-4.** *Könnten Sie sich diese Design-Variante nicht nur auf dem PC/Laptop sondern auch auf dem Smartphone/Tablet vorstellen?*

<sup>14</sup> Die Wizard-of-Oz-Methode basiert darauf, dass ein Mensch (der Wizard) Teile der Aufgaben übernimmt, die in einem fertigen System automatisch ablaufen[PD10]

**F-5.** Sind zusätzliche Informationen zu dem Zustand notwendig bzw. wünschenswert?

**F-6.** Ist der Unterschied zwischen zwei benachbarten Zuständen erkenntlich?

An dem **ersten Durchlauf** haben 7 Testpersonen teilgenommen. Da die Evaluation 'lautes Denken' beinhaltet, war es sinnvoll ein Einzeltreffen mit jedem Probanden zu vereinbaren. Das hat sich als Vorteil erwiesen, da die Testpersonen sich viel wohler und entspannter gefühlt haben und deren Meinung und Kommentare nicht von den anderen beeinflusst worden waren. Das Ziel der ersten Iteration war es, erste grundsätzliche Ansätze zu testen, grobe Nachteile zu beseitigen und in der zweiten Iteration angepasste und verbesserte Design-Varianten zu präsentieren. Jede der Testpersonen wurde zufälligerweise eine der Design-Varianten zugewiesen: 3 Personen haben das Slider-UI bekommen, 2 Personen das Slideshow-UI und 2 Personen das VideoPlayer-UI. Nach dem Iterationsschritt wurden einige Änderungen vorgenommen, die bei der zweiten Iteration getestet wurden. Zu den erwähnten Verbesserungen gehören bspw. zusätzliche Buttons für größere Schritte in dem Slideshow-UI, Play/Pause-Eingaben in dem VideoPlayer-UI auf ein Button.

An dem **zweiten Durchlauf** haben 8 Testpersonen teilgenommen. Analog zur ersten Durchführung, wurde jedem Probanden eine Design-Variante zur Evaluation ausgehändigt: 3 Personen haben das Slideshow-UI bekommen, 2 Personen das Slider-UI und 3 Personen das VideoPlayer-UI.

Die gesammelte Daten und Gespräche wurden für die nachfolgende Auswertung der Ergebnisse protokolliert und zusammengefasst.

## **6.3 AUSWERTUNG DER ERGEBNISSE**

### **6.3.1 Auswertung der Antworten auf vordefinierte Fragen**

Während der empirischen Evaluation werden viele Daten gesammelt, die oft unstrukturiert sind. Die vordefinierten Fragen, die allen Probanden gestellt wurden, liefern, entgegen den Kommentaren und Meinungen einzelner, eine Datenbasis für die im Folgenden dargestellte Vergleichstabelle und deren Auswertung. Eine Interpretation der Tabellenwerte bezüglich der gestellten Fragen findet sich im Anschluss.

	F-1	F-2	F-3	F-4	F-5	F-6
<b>Slider-UI</b>	++	++	++	-	-	-
	++	+	++	+	++	-
	++	++	++	-	-	-
	++	+	++	-	-	-
	++	+	++	-	-	-
<b>Slideshow-UI</b>	++	++	++	++	-	++
	++	+	++	++	++	-
	++	++	++	++	-	+
	++	+	++	++	++	-
	++	+	++	++	-	-
<b>VideoPlayer-UI</b>	++	++	++	++	-	-
	++	++	++	+	-	-
	++	+	++	+	-	-
	++	+	++	+	-	-
	++	++	++	-	-	-

Legende: ++ = ja, + = ja, unter der Voraussetzung, - = nein

Tabelle 6.1: Zusammenfassung von Antworten auf vordefinierte Fragen

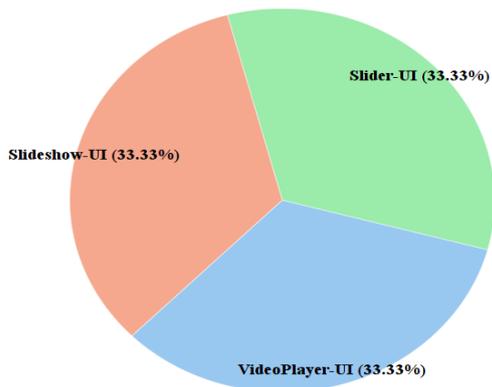


Abbildung 6.1: Auswertung der Ergebnisse von F-1 und F-3

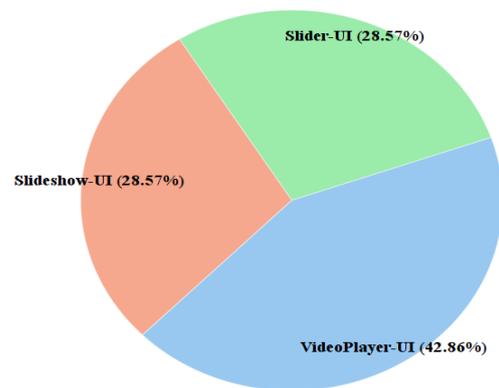


Abbildung 6.2: Auswertung der Ergebnisse von F-2

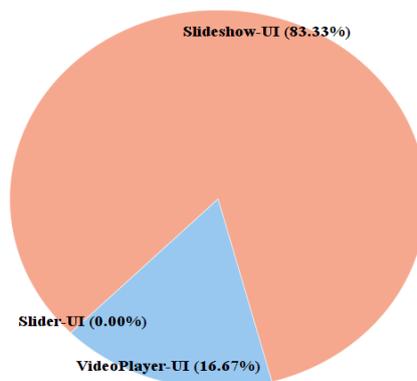


Abbildung 6.3: Auswertung der Ergebnisse von F-4

**F-1.** Die Darstellungsform und die zugrunde liegende Metapher wurden von allen Testpersonen erkannt und als passend empfunden.

**F-2.** Die meisten Icons wurden richtig identifiziert. Die Buttons, die das schnellere Navigieren darstellen sollten, haben 8 von 15 Probanden als missverständlich empfunden.

**F-3.** Die Richtung des zeitlichen Verlaufs war auch für alle intuitiv nachvollziehbar.

**F-4.** Alle fünf Probanden, die das Slideshow-UI getestet haben, können sich diese Lösung auch auf einem mobilen Gerät vorstellen. Was Slider-UI angeht, waren sich die Testpersonen auch einig, dass diese Design-Variante für ein mobiles Gerät eher unpassend ist. Den Testpersonen wurde eine Lösung vorgeschlagen, bei der das Interface auf einem mobilen Gerät nur aus dem Zeichenverlauf (d.h. ohne das Endbild) bestehen würde. Unter dieser Voraussetzung konnten sich drei von fünf Probanden diese Design-Variante auf dem Smartphone vorstellen.

**F-5.** Die Frage, ob zusätzliche Informationen zu dem Zustand notwendig bzw. wünschenswert sind, haben alle Testpersonen, die das VideoPlayer-UI evaluiert haben, mit 'nein' beantwortet, zwei von fünf Slideshow-Tester und ein Slider-Tester mit 'ja'.

**F-6.** Fast alle Probanden (13 von 15) hatten Schwierigkeiten, den Unterschied zwischen zwei Zuständen schnell zu finden. Vor allem ist es schwierig, den Unterschied zwischen sich weiter im Verlauf befindenden Zuständen, die schon viele Elemente beinhalten, zu sehen.

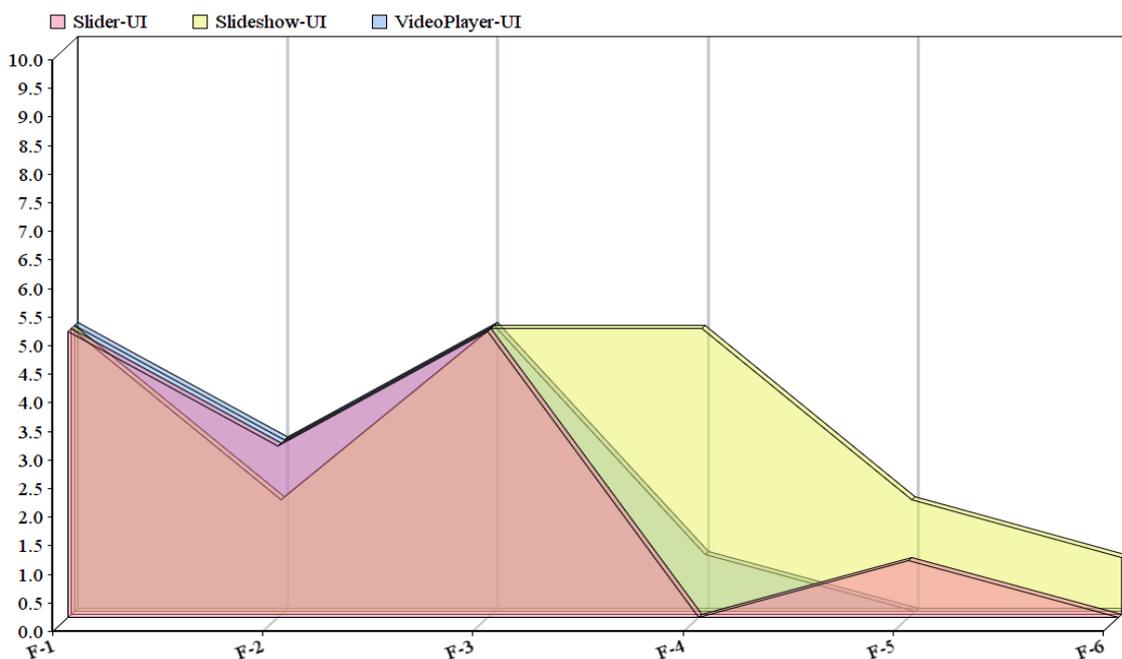


Abbildung 6.4: Diagramm zum Vergleich von Design-Varianten. (F-1 bis F-6 sind die während der Evaluation gestellten Fragen)

### 6.3.2 Auswertung verbaler Äußerungen

Die Auswertung verbaler Äußerungen ist komplizierter und kann daher nicht durch ein universell anwendbares Verfahren gelöst werden[PD10]. Die während der Evaluation gesammelten Meinungen wurden zunächst kategorisiert und auf dieser Basis analysiert.

Die aus den Gesprächen mit den Probanden gesammelten Kommentare lassen sich grob in vier Kategorien zusammenfassen: *Buttons*, *Richtung des Verlaufs*, *Unterschiede zwischen Zuständen*, *Sonstiges*. In der folgenden Tabelle werden Ergebnisse der Evaluation auf Basis von diesen vier Kategorien für jede Design-Variante zusammengefasst.

UI	Kategorie	Beschreibung
Slider-UI	Buttons ◀▶▶	Die Metapher, die für diese Buttons benutzt wurde, stammt von den Bedienelementen eines Videoplayers, sowohl eines physischen als auch eines virtuellen. Die Tatsache, dass diese Metapher für eine schnellere Navigation verwendet wurde, erkannten alle Probanden. Die vordefinierte Schrittgröße (fünf Schritte = fünf sichtbare Zustände) fiel nur zwei von fünf Probanden auf.
	Richtung	Die angebotene Richtung war anhand der ersten fünf sichtbaren Zuständen und üblichen Leserichtung gut nachvollziehbar und wurde als sinnvoll empfunden. Zwei von fünf Probanden haben den Wunsch geäußert, selbst die Abspielrichtung des Verlaufs zu bestimmen, d.h. von dem Endzustand zu dem Startzustand.
	Unterschiede	Da Unterschiede zwischen den Zuständen meist nicht intuitiv nachvollziehbar sind, haben alle Testpersonen von einer Markierung, farblicher Hinterlegung oder Animation gesprochen. Nach längerer Überlegung wurde eine farbliche Hinterlegung ausgeschlossen, da Graphicuss als Editor die Farbauswahl für Werkzeuge anbietet. Im ungünstigsten Fall würden so bspw. graphische Primitive und die letzte Änderung in der gleichen Farbe dargestellt, was zu Missverständnissen führen könnte. Auch die ausschließliche Nutzung von Kreisen oder Rahmen zur Markierung von Änderungen ist nicht weniger problembehaftet, da Graphicuss auch Werkzeuge zur Erstellung dieser Primitive beinhaltet.
	Sonstiges	Alle Probanden empfanden die Schrittweite von fünf für die schnellere Navigation als optimal.
	Buttons ◀▶▶	Die einfachen Navigationselemente waren allen Probanden aus der alltäglichen Internetnutzung bekannt. Die Icons, die für die schnellere Navigation ausgesucht wurden, wurden als Metapher für die erhöhte Geschwindigkeit identifiziert, allerdings war ohne zusätzliche Information nicht eindeutig, was genau beim Betätigen passiert. Zwei von fünf Testpersonen haben darauf hingewiesen, dass, obwohl sie die Buttons richtig identifiziert haben, sie nicht wussten wieviele Schritte auf einmal übersprungen wurden. Drei von fünf Probanden haben die Buttons als Sprung zum Ende bzw. zum Anfang interpretiert. Alle Evaluationsteilnehmer haben als Lösung dieses Problems eindeutige Tooltips und Pagination (in der Form: Nummer des aktuellen Zustands/Anzahl aller Zustände) bevorzugt.

<b>Slideshow-UI</b>	Richtung	Da nur ein Zustand im Fokus des Nutzers liegt und der Verlauf relativ lang sein kann, haben alle fünf Probanden die Möglichkeit vermisst, die Richtung, in der der Verlauf durchschritten wird, selbst zu bestimmen. Im diesem Fall ist es wünschenswert, den Zeichen Verlauf zyklisch zu gestalten, d.h. die Navigation erfolgt von dem Startzustand zu dem Endzustand und nahtlos wieder zu dem Startzustand. Das ermöglicht dem Endnutzer sich direkt zu entscheiden in welche Richtung er den Verlauf abspielen lassen will.
	Unterschiede	Analog zu dem Slider-UI hatten die Testpersonen Schwierigkeiten mit dem Unterschied zwischen zwei Zuständen. Die Lösungsvorschläge waren auch zu den für Slider identisch.
	Sonstiges	Alle Probanden empfanden die Schrittweite von fünf für die schnellere Navigation als optimal.
<b>VideoPlayer-UI</b>	Buttons ◀▶⏮⏭	Dank der zugrunde liegenden Metapher eines Video-players/Videorekorders haben drei von fünf Probanden alle Buttons richtig identifiziert. Die zwei anderen hatten Schwierigkeiten mit Tasten für die schnellere Navigation. Analog zum Slider-UI wurden sie, ohne Tooltips, nicht als Möglichkeit fünf Zustände zu überspringen interpretiert.
	Richtung	Die angebotene Richtung war gut nachvollziehbar und wurde als sinnvoll empfunden.
	Unterschiede	Analog zu den Ergebnissen der anderen beiden Testgruppen, wurden Unterschiede zwischen Zuständen hier auch als 'schwer nachvollziehbar' bewertet. Analog zum Slider-UI war auch hier der Lösungsansatz das Markieren von Änderungen.
	Sonstiges	Vier von fünf Probanden konnten Preview-Bilder, die beim Hovern über die Videoleiste erscheinen, um eine Orientierung zu erleichtern, nur zufällig entdecken. Während der Recherche nach verwandten Konzepten konnte keine sinnvolle und effiziente Lösung für dieses Problem gefunden werden. Alle Probanden empfanden die Schrittweite von fünf für die schnellere Navigation als optimal.

Tabelle 6.2: Auswertung der Interviews mit den Probanden

Ergebnis der durchgeführten Evaluation war, dass es keinen eindeutigen Favoriten zur Umsetzung des Designs der Nutzerschnittstelle gibt. Alle vorgestellten Design-Varianten weisen sowohl Vor- als auch Nachteile auf. Während der Auswertung wurden Aussagen protokolliert, die unabhängig von der späteren Design-Entscheidung, als allgemeine Empfehlungen betrachtet werden können. Diese Aussagen lauten wie folgt:

1. Unterschied zwischen den Zuständen ist schlecht nachvollziehbar und muss deutlich markiert werden.
2. Der Nutzer muss die Möglichkeit haben, die Richtung des Zeichenverlaufs zu bestimmen.
3. Alle Buttons müssen mit Tooltips versehen werden.
4. Fünf Schritte für schnellere Navigation sind optimal.

Die verbesserten und angepassten Design-Varianten (s. Anhang [A.3](#)) entsprechen allen im Kapitel [3.3](#) aufgelisteten Muss-Anforderungen, bei den Kann-Anforderungen muss der Aspekt der Bedienbarkeit auf verschiedenen Endgeräten berücksichtigt werden. Obwohl eine der wichtigsten Eigenschaften von Graphicuss und AMCS die Geräteunabhängigkeit ist, stehen in dieser Arbeit PCs, Laptops, Tablets und Smartphones im Fokus, da vornehmlich diese von Studenten verwendet werden. Wie in der Tabelle [6.1](#) erkennbar ist, entspricht nur das **Slideshow-UI** der Anforderung der geräteunabhängigen Bedienbarkeit und wird ausgehend von der Bewertung bezüglich Anforderungen und Auswertung der Evaluationsergebnisse zur Umsetzung der Nutzerschnittstelle empfohlen.

# 7 ZUSAMMENFASSUNG UND AUSBLICK

Zu Beginn der Arbeit wurden zunächst einige grundlegende Begriffe erklärt, die für das Verständnis der Funktionsweise von bereits existierenden Systemen wichtig sind: Forensysteme und virtuelle interaktive Whiteboards. Anschließend wurde der aktuelle Stand des am Lehrstuhl Rechnernetze der Technischen Universität Dresden entwickelten Systems *AMCS* untersucht. Der Funktionsumfang wurde dabei aus zwei Nutzersichten betrachtet: Studierender und Lehrender. Entsprechend der Aufgabenstellung wurde die zukünftige Komponente von *AMCS* - *Graphicuss* - als Umgebung für den eigentlichen Nutzerschnittstellenentwurf vorgestellt. Dabei wurde sowohl auf drei Basisfunktionalitäten: Kursverwaltung, Fragenverwaltung und Antwortverwaltung, als auch auf weitere Funktionalitäten in Form von *drawing tool* und *real time*, eingegangen. Ausgehend von der Aufgabenstellung und basierend auf bereits gewonnenen Kenntnissen über die Funktionsweise der vorgestellten Systeme, wurden im Kapitel 3.3 Anforderungen an die Nutzerschnittstelle definiert. Die formulierten Anforderungen wurden in der Tabelle 3.1 in drei Gruppen zusammengefasst: *Muss-Anforderungen* (MA-1..MA-5), *Kann-Anforderungen* (KA-1..KA-5) und *Abgrenzungsanforderungen* (AA-1..AA-3).

Nachdem in den ersten Kapiteln einige Grundlagen, die für das Gesamtkonzept wichtig sind, vorgestellt wurden, folgte eine weiterführende Recherche zu verwandten Konzepten. In erster Linie wurden bereits existierende online Whiteboards getestet und verglichen. In der Arbeit wurden die 12 bekanntesten betrachtet. Die Recherche hat aufgezeigt, dass alle untersuchten Whiteboards die gleiche Menge gängiger Werkzeuge anbietet (bspw. Größe- und Farbauswahl). In Tabelle 4.1 wird eine zusammengefasste Funktionsübersicht der bekanntesten V-IWBs dargestellt. Heutzutage existieren mehrere virtuelle interaktive Whiteboards und es ist wichtig zu wissen, wie deren Designer und Entwickler mit dem Problem der Visualisierung von Zeichenverläufen, das Gegenstand dieser Arbeit war, oder vergleichbaren Problemen verfahren bzw. lösen. Wie die Recherche gezeigt hat, werden die Zeichenverläufe in V-IWBs oft auf standardisierte redo-/undo-Funktionen reduziert. Umso wichtiger und interessanter wird dadurch das Ergebnis dieser Bachelorarbeit.

Auf Basis von Literaturrecherchen wurden mögliche undo-Modelle untersucht und im Kapitel 4.1.2 vorgestellt: *ein lineares*, *ein nicht lineares* und *ein direktiv selektives undo-Modell*. Wie zuvor erwähnt wurde, werden Zeichenverläufe moderner V-IWBs linear aufgebaut und werden mit Hilfe einfacher Richtungspfeile dargestellt, d.h. nach dem Zitieren und anschließendem Editieren wird der nachfolgende Verlauf in der neuen Version ersetzt bzw. neu erschaffen. Wichtig ist es hier, dass das folgende Problem berücksichtigt wird. Unter Umständen hat eine vorangegangene Nutzeraktion keine sinnvolle Interpretation in einem Nachfolgezustand, bspw. müssen die Folgen des Löschens einer geometrischen Figur in einem früheren Zustand und deren Auswirkungen auf alle nachfolgenden Zustände berücksichtigt werden. Um diesem Problem zu begegnen, wurde ein direktives selektives undo-Modell vorgestellt. An dieser Stelle sei angemerkt, dass ein Kapitel dieser Arbeit bei weitem nicht ausreicht, um das Thema 'undo-Modelle', für das Treffen einer Entscheidung ausreichend, zu diskutieren.

Anschließend wurde der metaphorbasierten Entwurfsprozess behandelt und Hauptbestandteile dieses Prozesses aufgelistet: *Identifikation* von Kandidatenmetaphern, *Bewertung* von Metaphern und *Entwicklung* von Metaphern. Anschließend wurden im Rahmen des ersten Entwurfsschrittes Kandidatenmetaphern für die zukünftige Nutzerschnittstelle definiert: die *Film*-, *Karussell*- und *Kartenstapel*-Metaphern.

Entsprechend der Aufgabenstellung muss die Nutzerschnittstelle verschiedene Nutzungskontexte berücksichtigen, diese wurden im Kapitel 4.2 definiert: *textlastige*, *diagrammlastige*, *skizzenlastige*, *bildlastige* und *dokumentlastige Kommunikation*. Es muss allerdings beachtet werden, dass Nutzungskontexte in beliebigen Kombinationen auftreten können.

In Kapitel 5 wurden die Nutzungskontexte bezüglich der Eigenschaften eines Forums für Graphicuss angepasst. Folgende Beitragsarten mit jeweils unterschiedlichen Schwerpunkten ließen sich ableiten: *Text*, *Diagramm/Graph*, *Skizze: Zeichnung*, *Skizze: Formel* und *Bild*. Für jede Beitragsart wurde zusätzlich die kleinste sinnvolle Informationseinheit definiert. Im Rahmen des zweiten Schrittes eines metaphorbasierten Designprozesses - *Bewertung von Kandidatenmetaphern* - wurden die drei Kandidatenmetaphern in Bezug auf die definierten Nutzungskontexte bewertet. Dafür wurden zunächst fünf User Interface-Varianten basierend auf den Kandidatenmetaphern skizziert: *Slider-UI*, *Slideshow-UI*, *VideoPlayer-UI*, *Karussell-UI* und *Kartenstapel-UI*. Die Ergebnisse der Bewertung dieser UI-Varianten wurden in der Tabelle 5.2 zusammengefasst. Wie der Tabelle zu entnehmen ist, gibt es keinen eindeutigen Favoriten. Trotzdem ließen sich die Karussell- und Kartenstapel-UIs aufgrund der Gesamtbewertung ausschließen. Basierend auf der Auswertung der Tabelle wurde die Entscheidung getroffen, im Rahmen dieser Arbeit die Slider-, VideoPlayer- und Slideshow-UIs für eine Evaluation mit Endnutzern heranzuziehen.

In Kapitel 6 lag der Fokus auf einem der wichtigsten Aspekte nutzerbasierter Entwicklung - die Evaluation der Zwischenergebnisse durch potentielle Endnutzer. Die Evaluation wurde mittels eines Papier-Prototypen und der Wizard-of-Oz-Methode durchgeführt. Alle Ergebnisse wurden protokolliert und anschließend ausgewertet (s. Tabelle 6.1 und Tabelle 6.2). Basierend auf den Ergebnissen der Evaluation wurden allgemeine Empfehlungen definiert, die von der späteren Design-Entscheidung unabhängig sind. Ausgehend von Bewertungen, die in verschiedenen Phasen der Entwicklung stattgefunden haben, wird das **Slideshow-UI** zur Umsetzung der Nutzerschnittstelle empfohlen.

Das Ergebnis dieser Bachelorarbeit soll als eine Empfehlung betrachtet werden. Während der tatsächlichen Implementierungsphase müssen sowohl hier definierte Anforderungen berücksichtigt als auch neue definiert werden. Da Evaluation ein entwicklungsbegleitender Prozess ist, müssen die Zwischenergebnisse weiter evaluiert werden. Ein wichtiger Aspekt weiterer Untersuchungen ist die Evaluation eines in die Graphicuss-Oberfläche integrierten Zeichenverlaufs.



# LITERATURVERZEICHNIS

- [ACS84] James E. Archer, Jr., Richard Conway, and Fred B. Schneider. User recovery and reversal in interactive systems. *ACM Trans. Program. Lang. Syst.*, 6(1):1–19, January 1984.
- [Ber94a] Thomas Berlage. A Selective Undo Mechanism for Graphical User Interfaces Based on Command Objects. *ACM Transactions on Computer-Human Interaction*, 1(3):269–294, September 1994.
- [Ber94b] Thomas Berlage. A selective undo mechanism for graphical user interfaces based on command objects. *ACM Trans. Comput.-Hum. Interact.*, 1(3):269–294, September 1994.
- [BKKS15] Iris Braun, Felix Kapp, Hermann Körndle, and Alexander Schill. Onlinegestützte audience response systeme: Förderung der kognitiven aktivierung in vorlesungen und eröffnung neuer evaluationsperspektiven. *Wissensgemeinschaften 2015*, pages 157–165, 2015.
- [Che16] Kaijun Chen. Graphical discussion system, 2016.
- [DV99] Alan R. Dennis and Joseph S. Valacich. *Rethinking media richness: Towards a theory of media synchronicity*, page 12. IEEE Comp Soc, 1999.
- [Eri90] T. D. Erickson. Working with interface metaphors. pages 119–154. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.
- [Har16] Tenshi C. Hara. *Analyses on tech-enhanced and anonymous peer discussion as well as anonymous control facilities for tech-enhanced learning*. Dissertation, Technische Universität Dresden, 2016.
- [Kub16] Tommy Kubica. Entwicklung eines Prototyps zur Auswahl und zum Einsatz technischer Werkzeuge/Werkzeugkombinationen in unterschiedlichen Lehrformen. Masterarbeit, Technische Universität Dresden, 2016.

- [Mad94] Kim Halskov Madsen. A guide to metaphorical design. *Commun. ACM*, 37(12):57–62, December 1994.
- [MYIM98] C. Meng, M. Yasue, A. Imamiya, and X. Mao. Visualizing histories for selective undo and redo. In *Proceedings of the Third Asian Pacific Computer and Human Interaction, APCHI '98*, pages 459–, Washington, DC, USA, 1998. IEEE Computer Society.
- [Ost03] B. Ostermann. *Softwareaktionen Wiederholen und Rückgängigmachen: Undo/Redo-Konzepte für interaktive Anwendungen am Beispiel des objektorientierten Rahmenwerkes JWAM*. 2003.
- [PD10] Bernhard Preim and Raimund Dachselt. *Interaktive Systeme. Band 2: User Interface Engineering, 3D-Interaktion, Natural User Interfaces*. Springer Berlin Heidelberg, 2010.
- [PD15] Bernhard Preim and Raimund Dachselt. *Interaktive Systeme. Band 1: Grundlagen, Graphical User Interfaces, Informationsvisualisierung*. Springer Berlin Heidelberg, 2015.
- [Thi90] Harold Thimbleby. *User Interface Design*. ACM, New York, NY, USA, 1990.
- [Yan88] Yiya Yang. Undo support models. *International Journal of Man-Machine Studies*, 28(5):457–481, 5 1988.

# A ANHANG

## A.1 BLANK PAPER

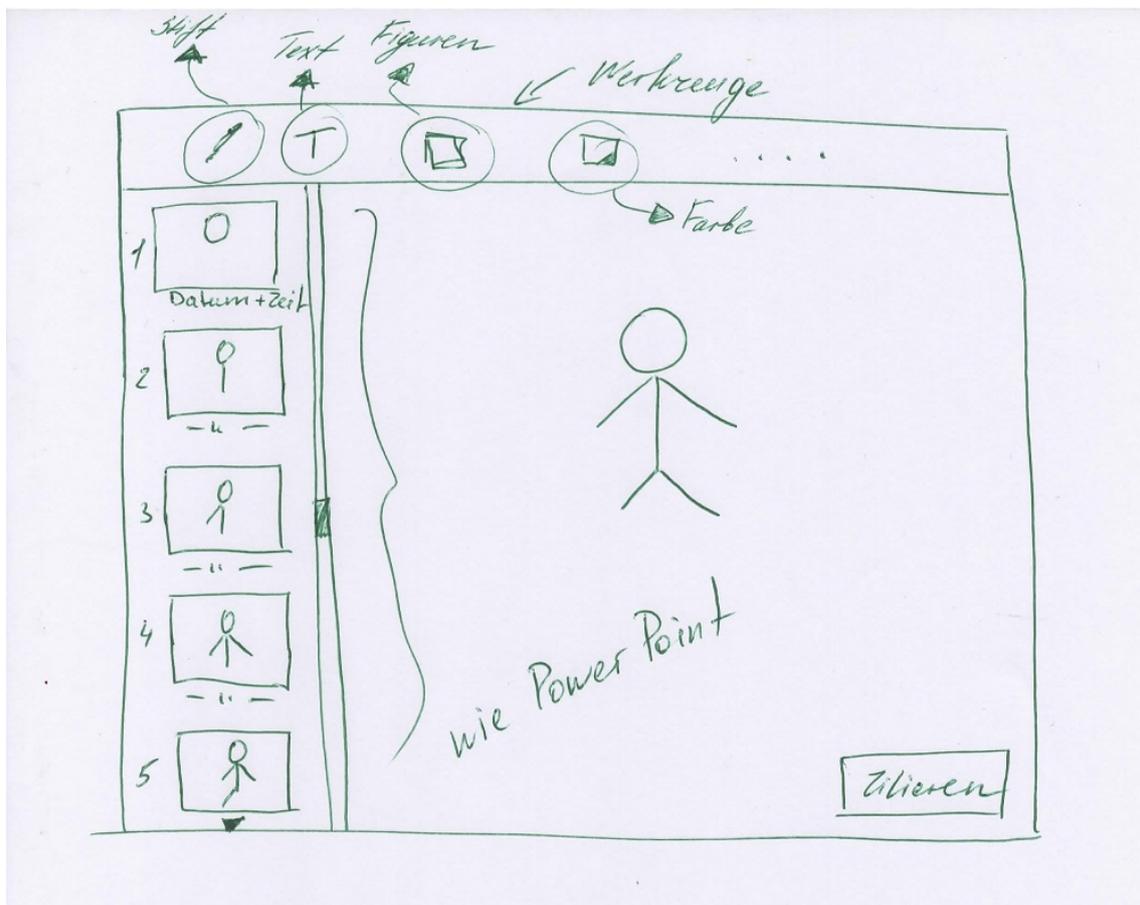


Abbildung A.1: Testperson A.

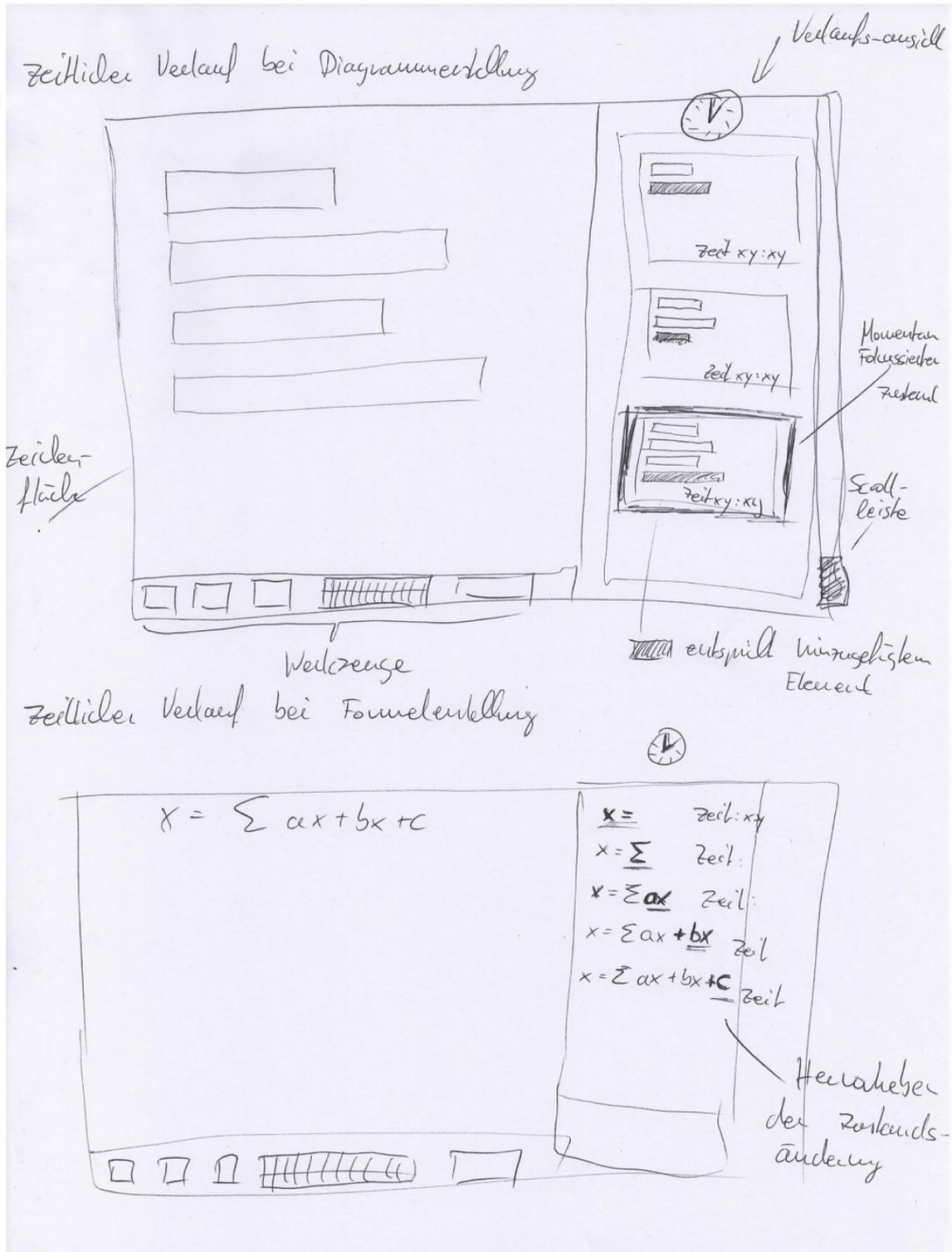


Abbildung A.2: Testperson B.

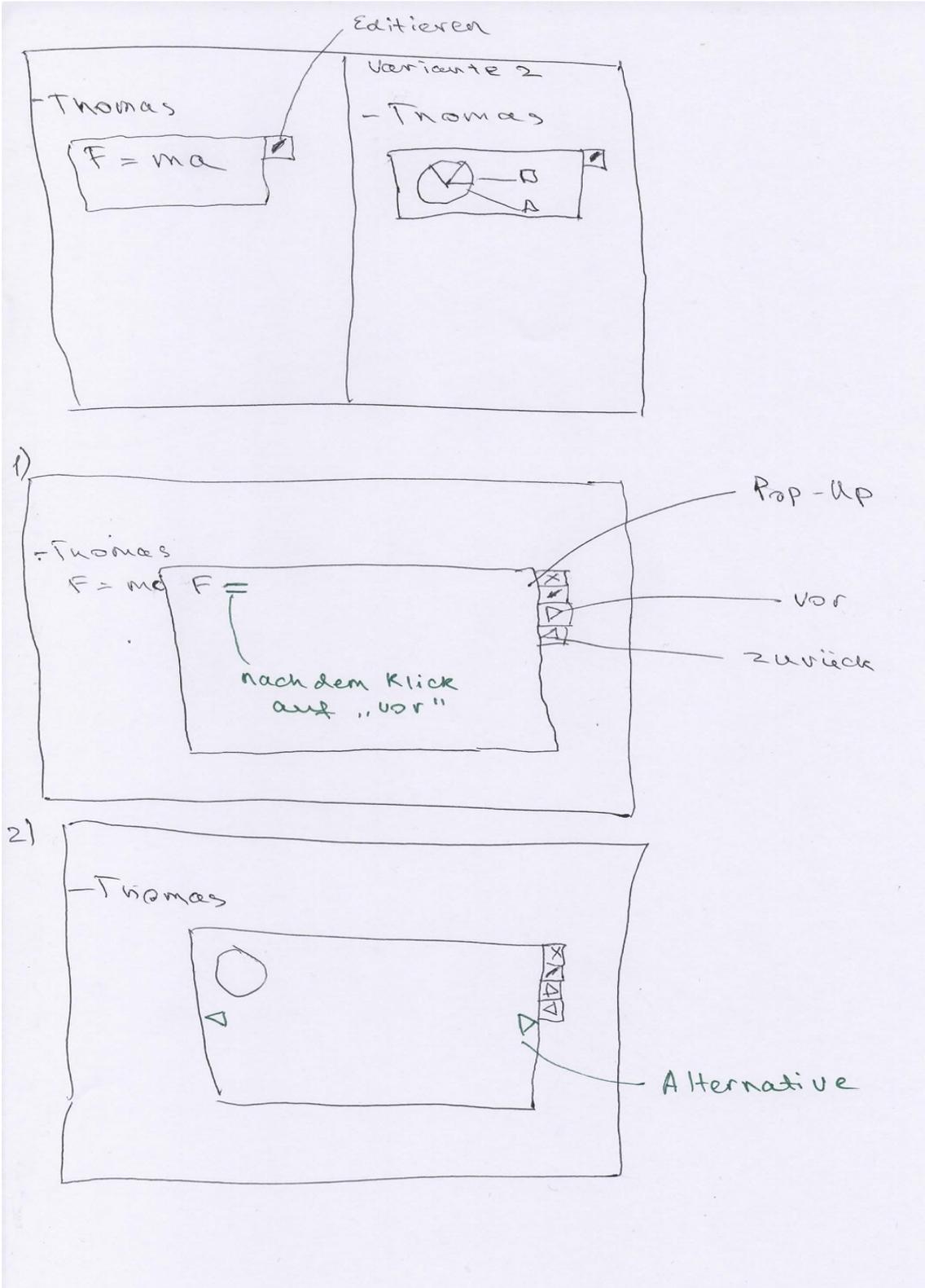


Abbildung A.3: Testperson C.

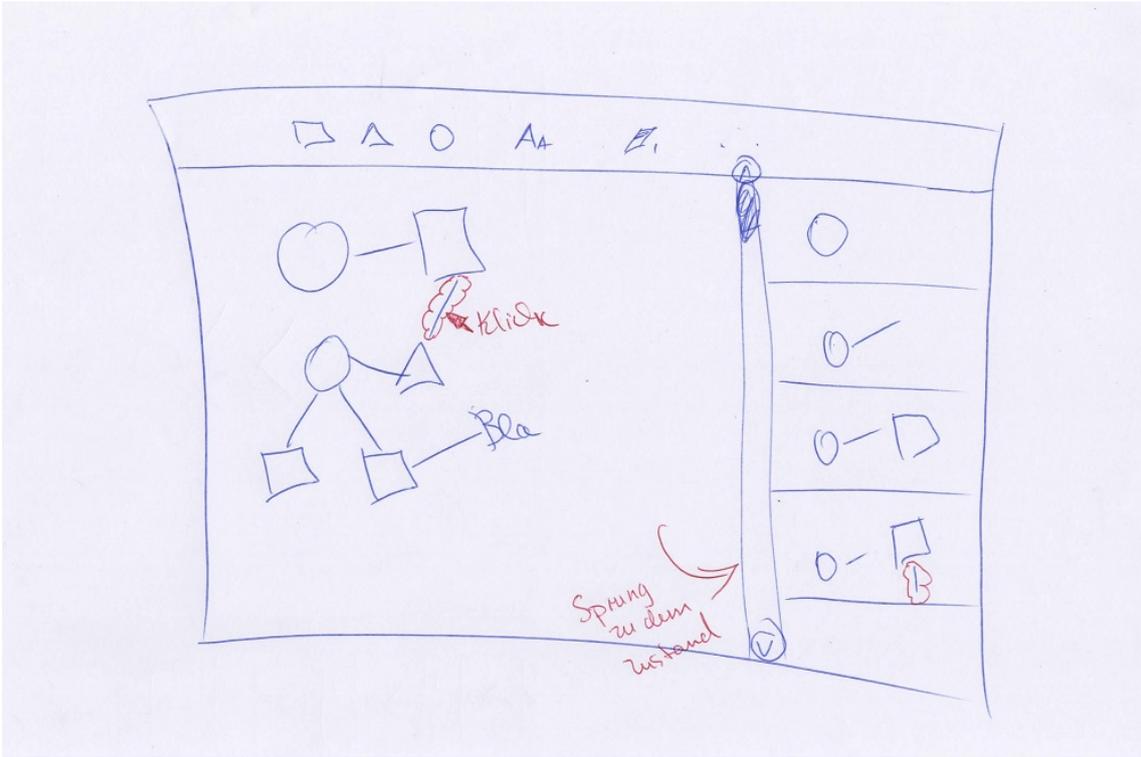


Abbildung A.4: Testperson D.

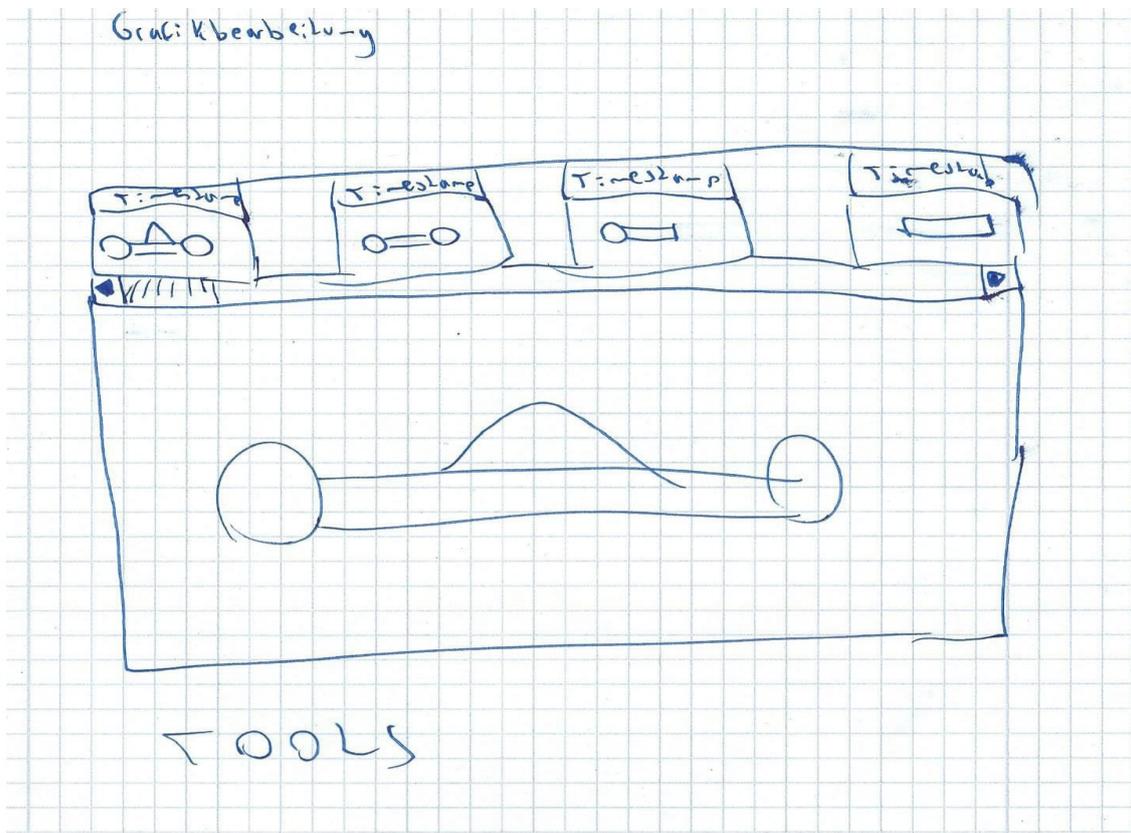


Abbildung A.5: Testperson E.

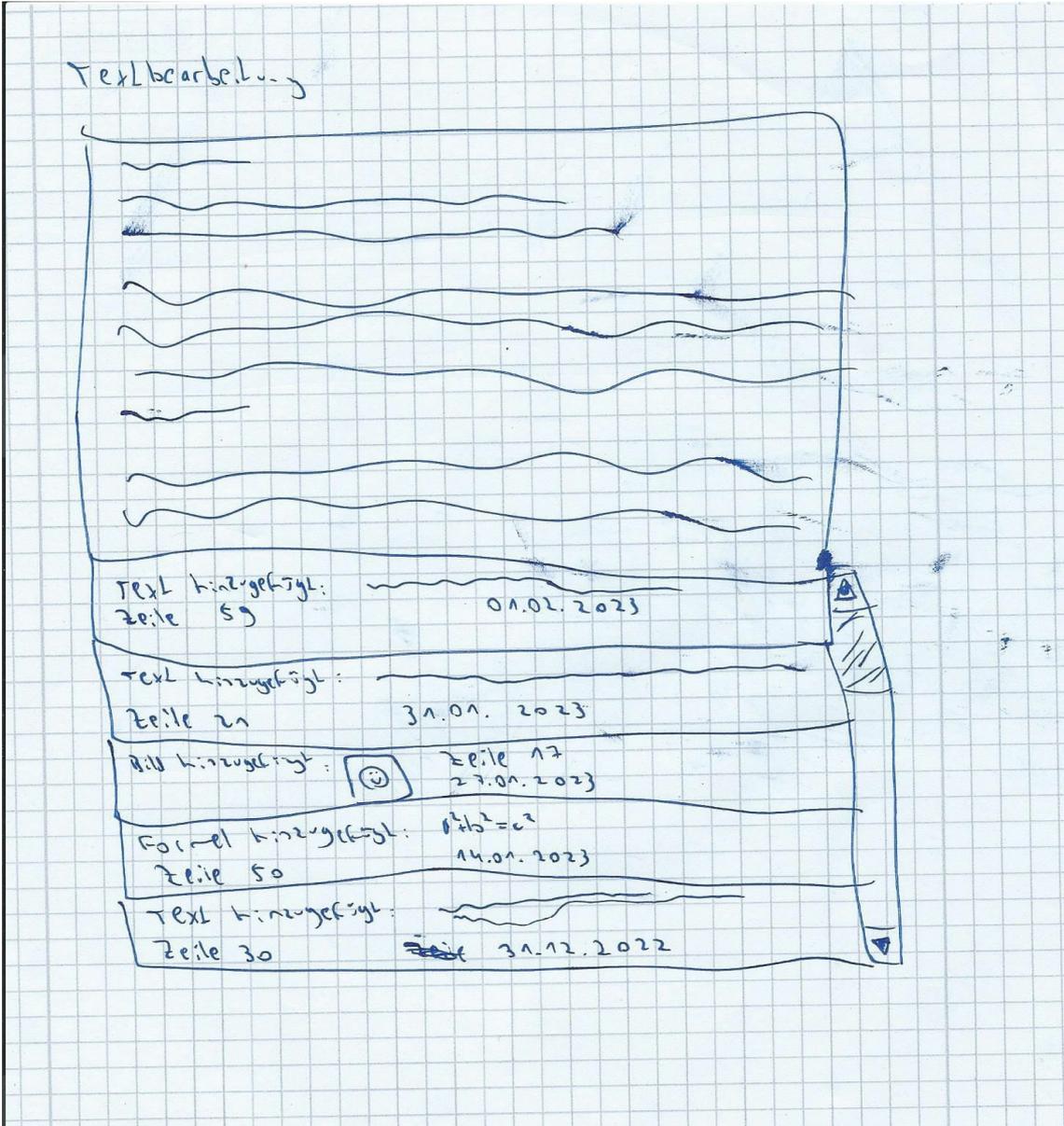


Abbildung A.6: Testperson E.

## A.2 PAPIER-PROTOTYPEN

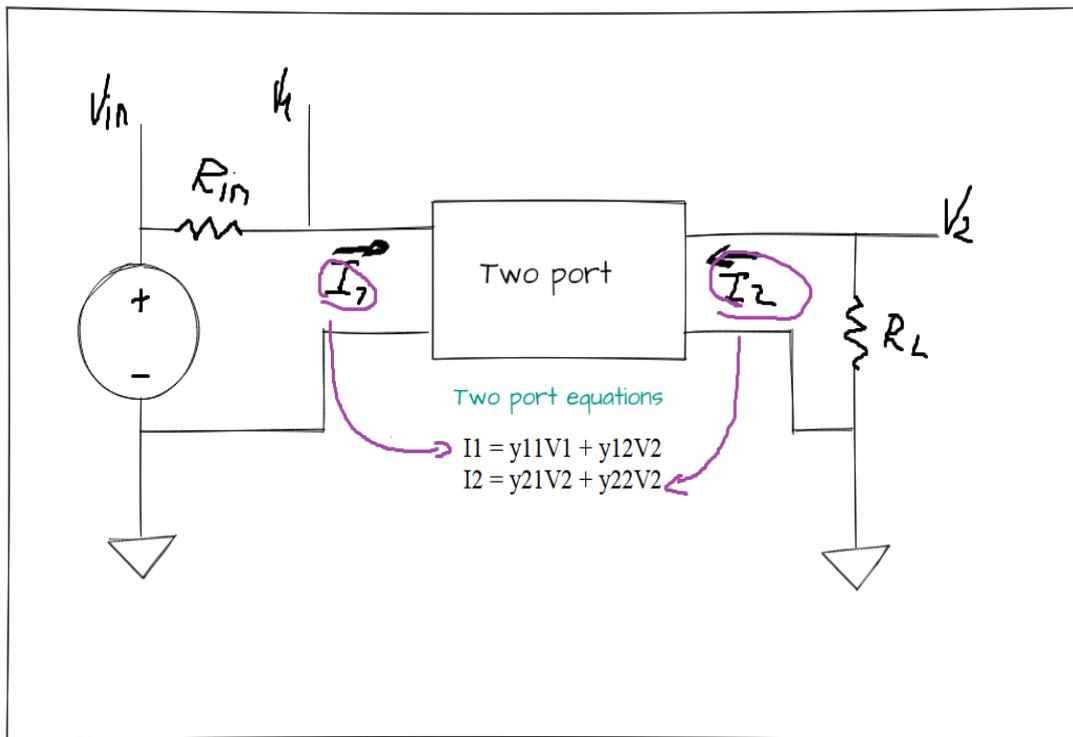
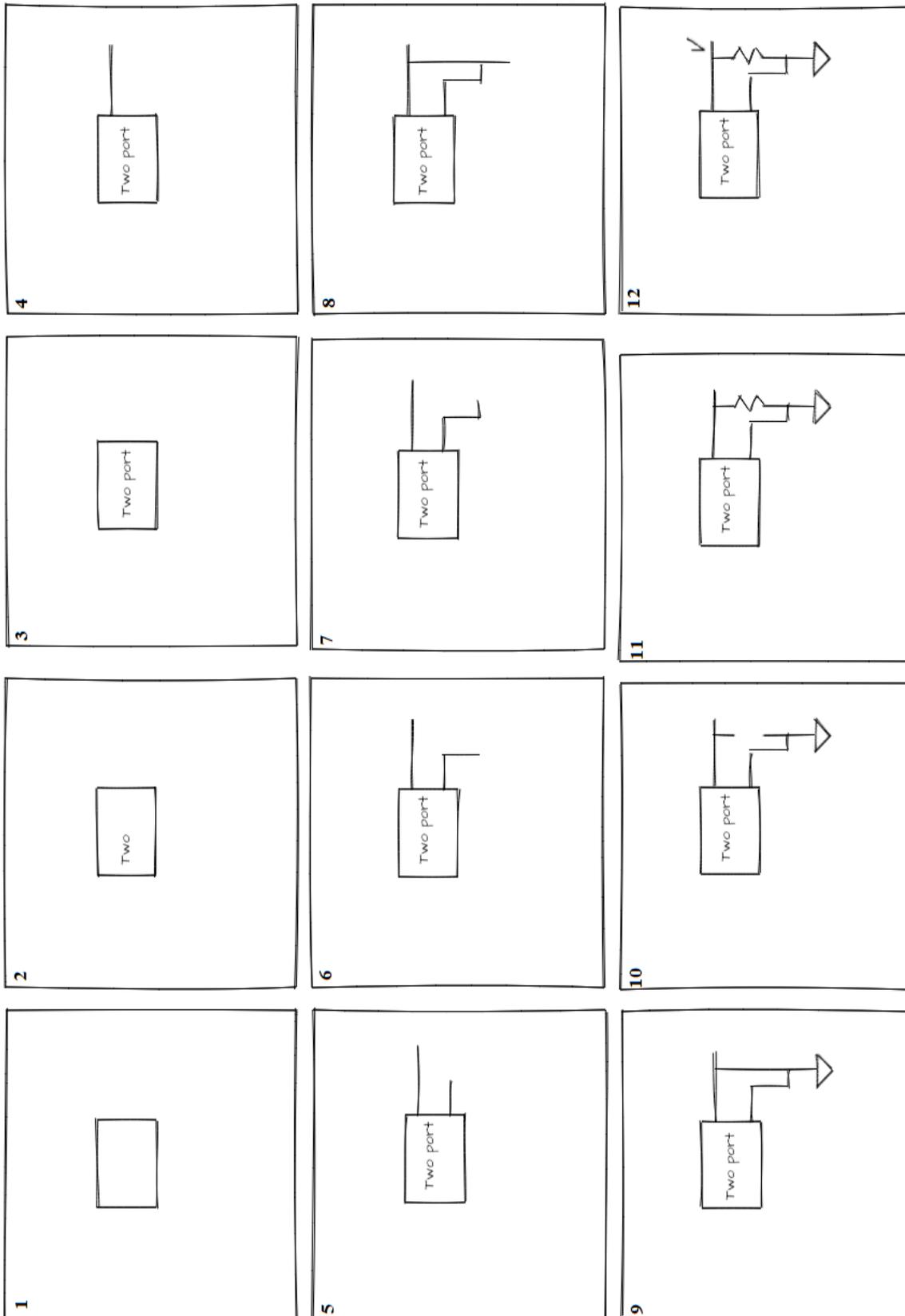
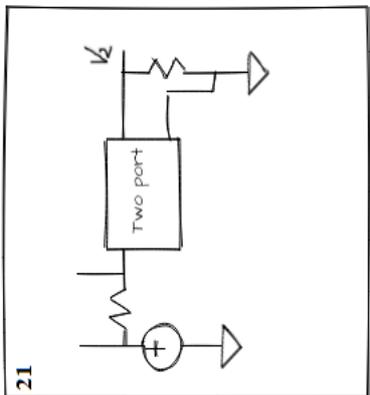
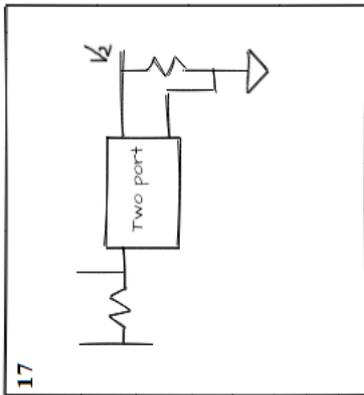
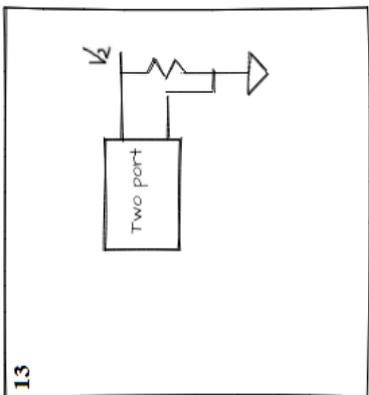
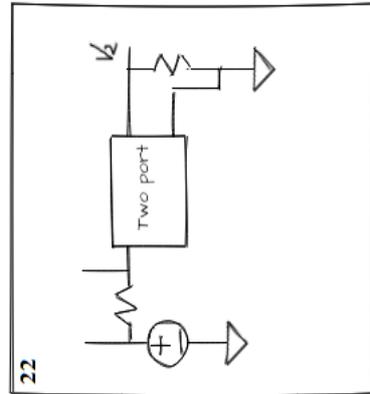
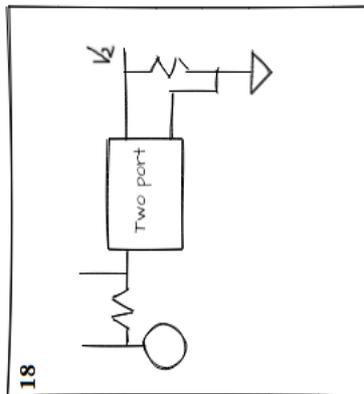
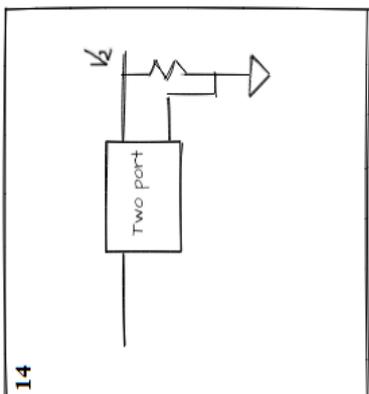
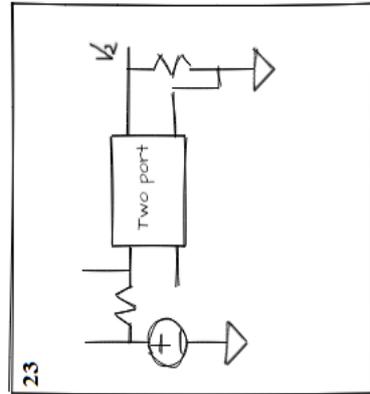
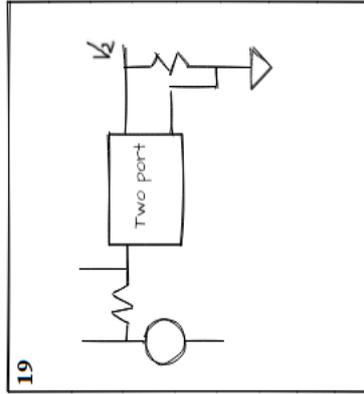
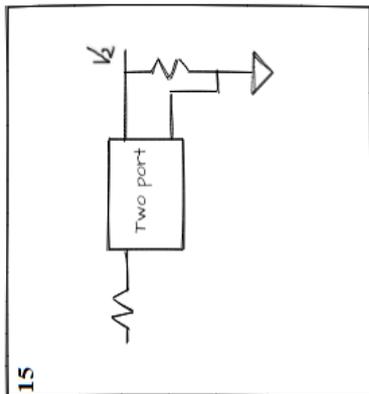
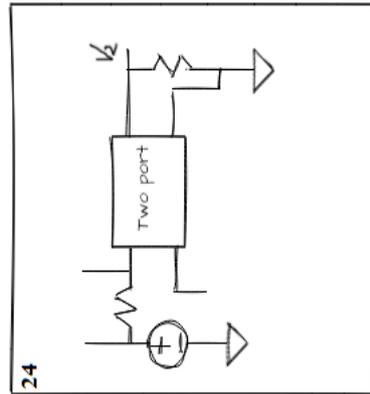
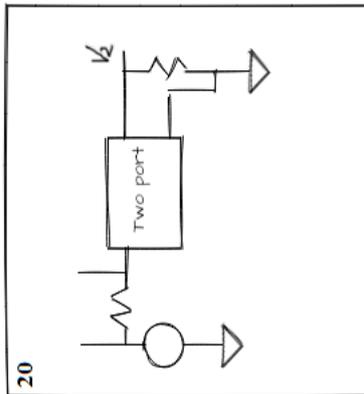
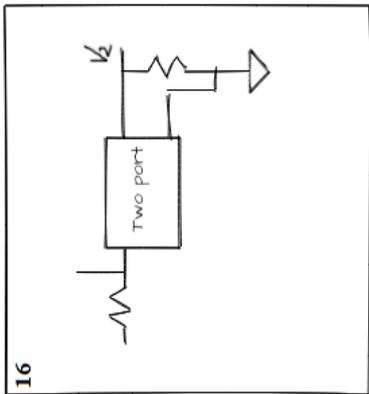
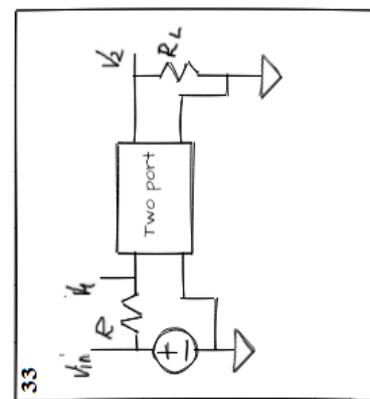
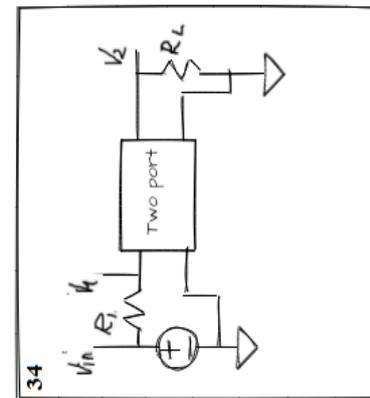
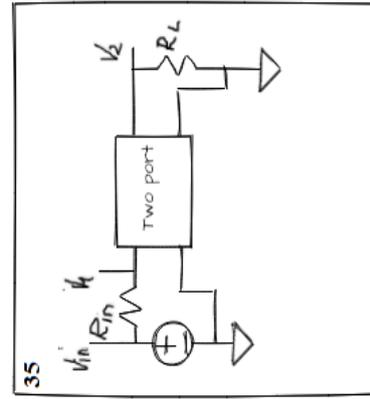
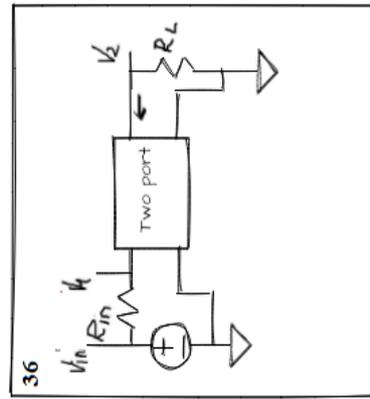
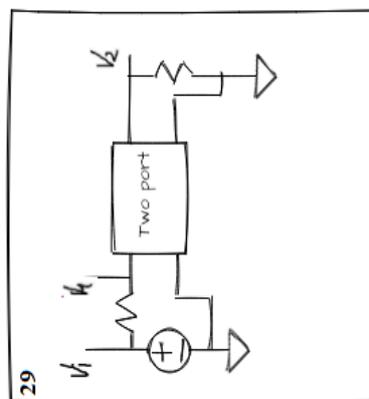
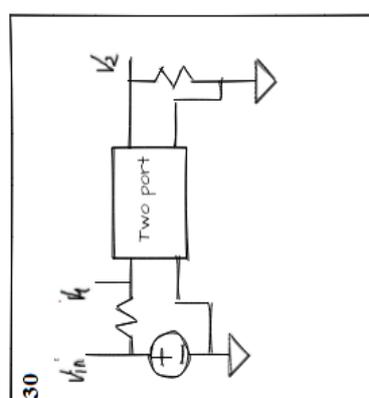
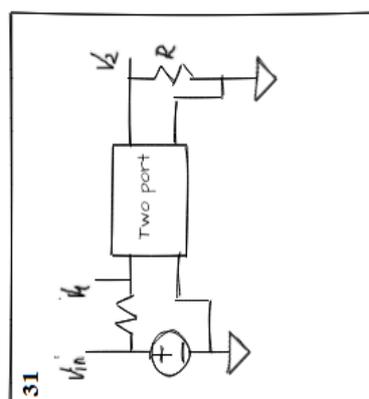
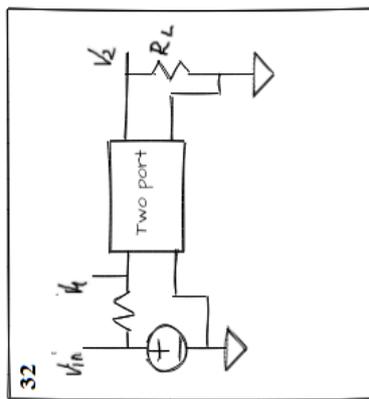
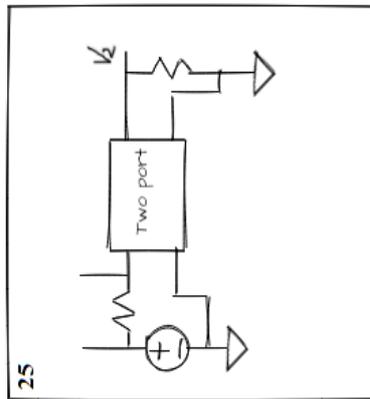
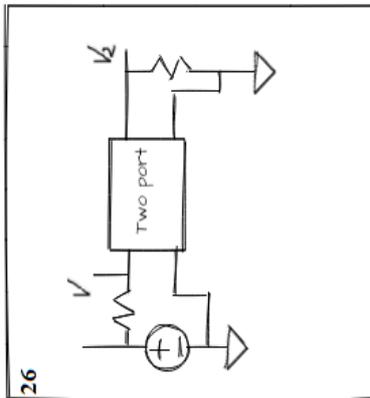
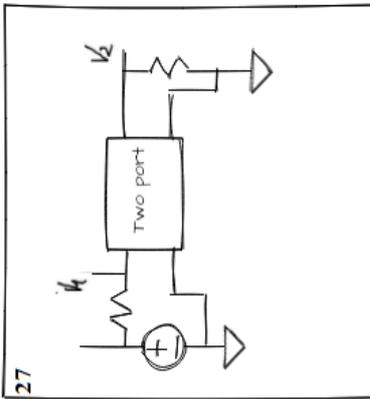
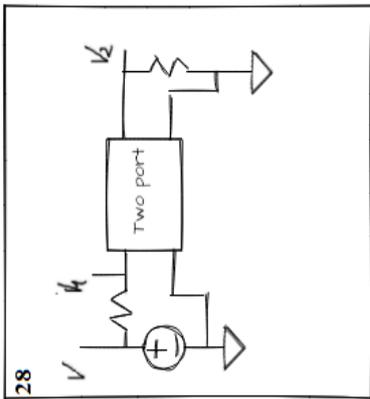


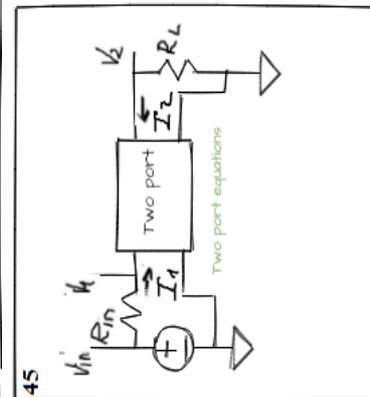
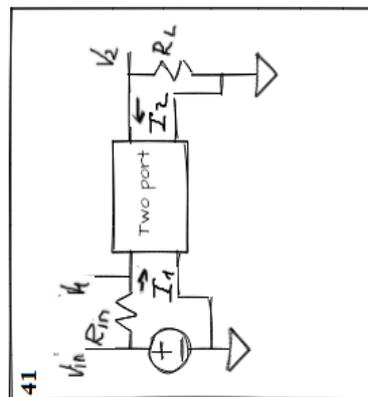
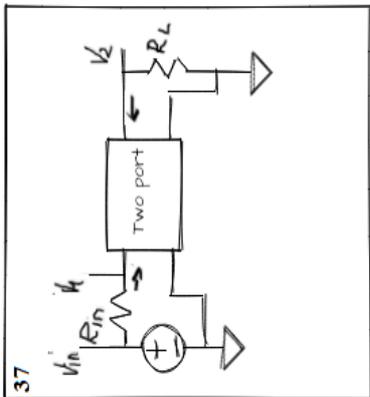
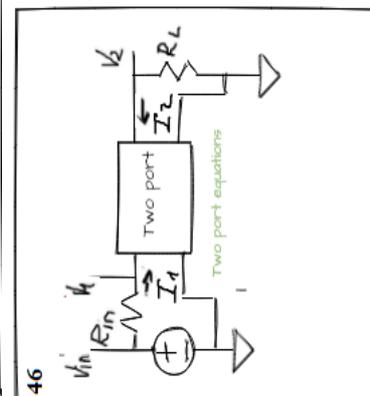
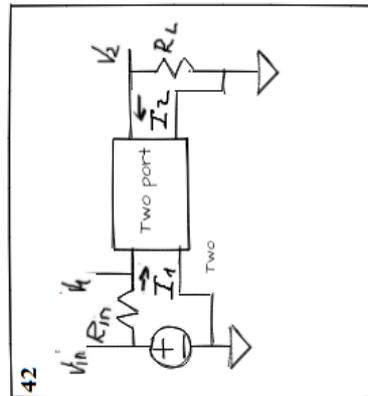
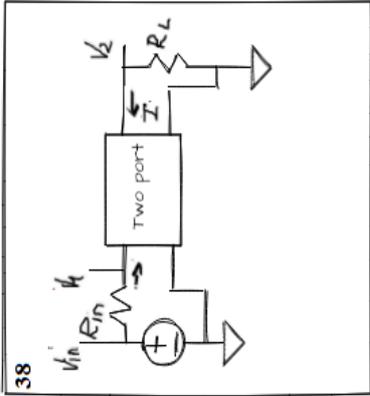
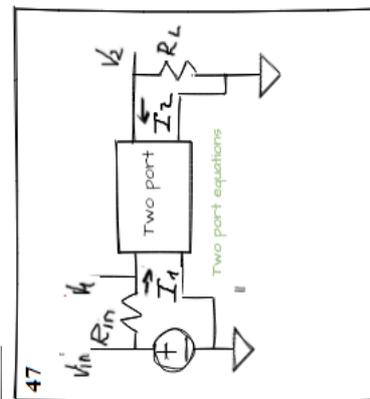
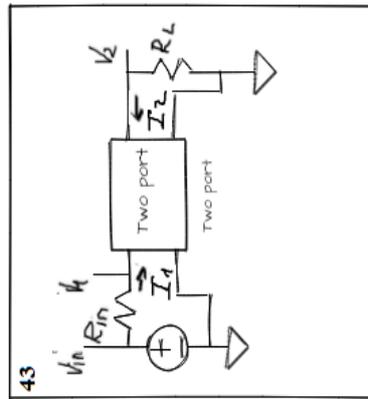
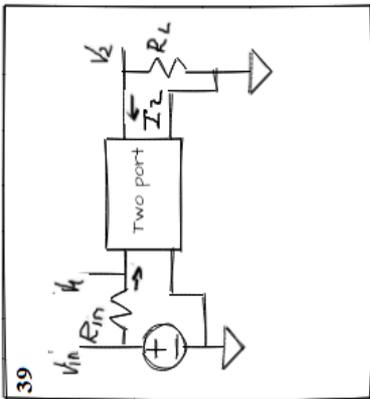
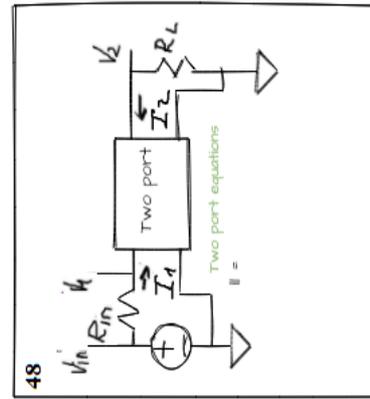
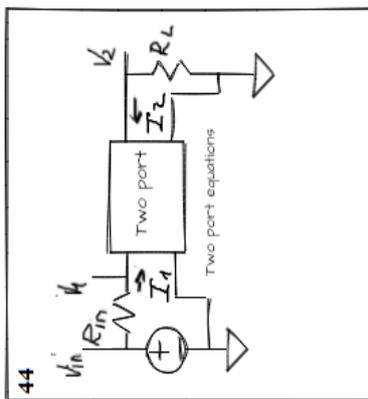
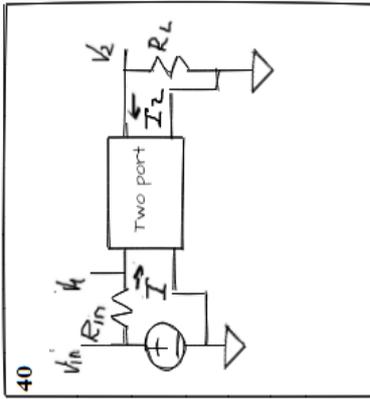
Abbildung A.7: Ein erstellter Beitrag als Ausgangspunkt

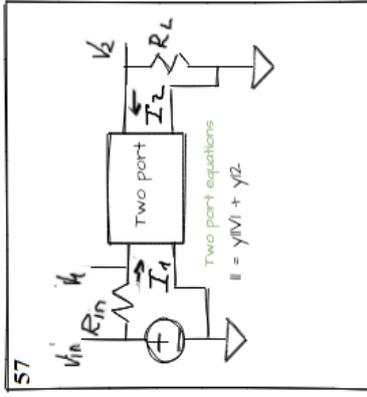
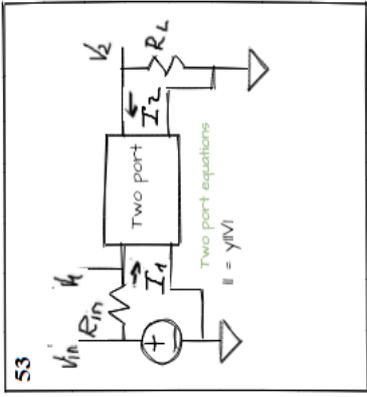
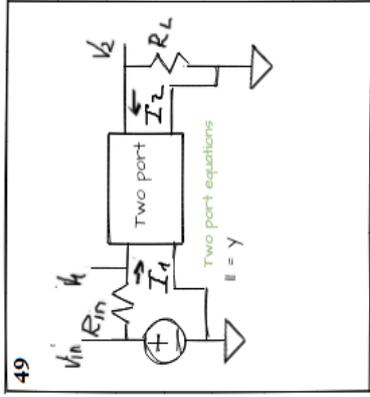
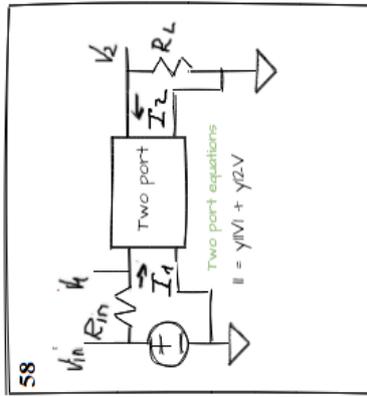
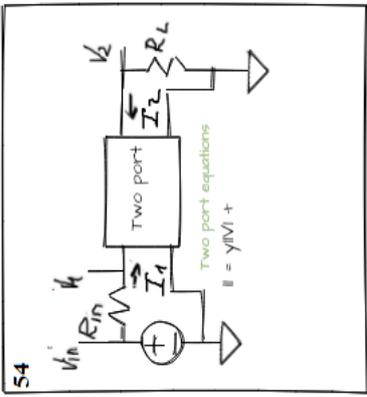
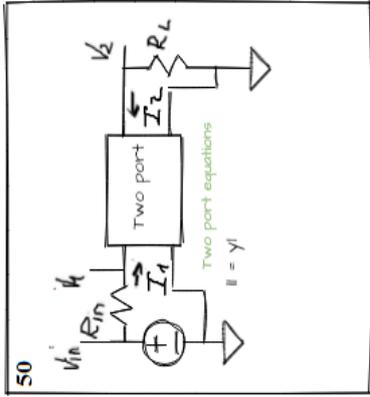
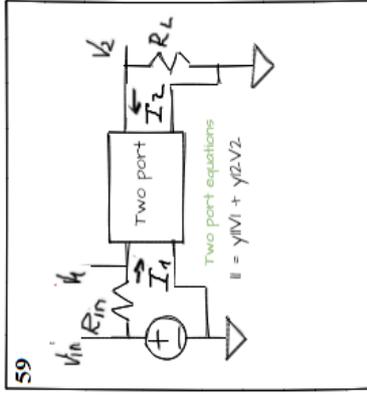
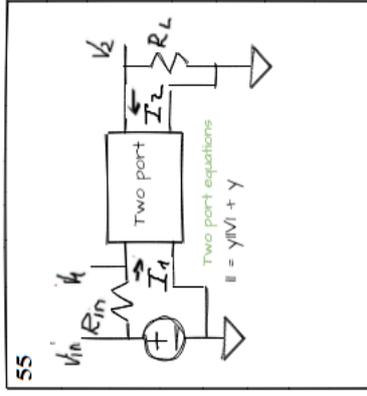
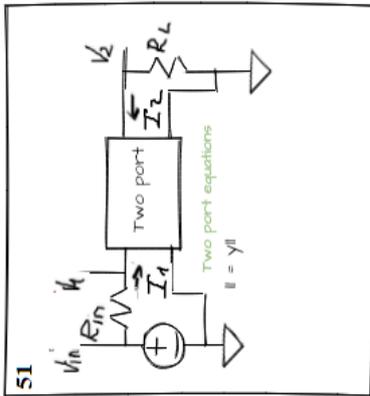
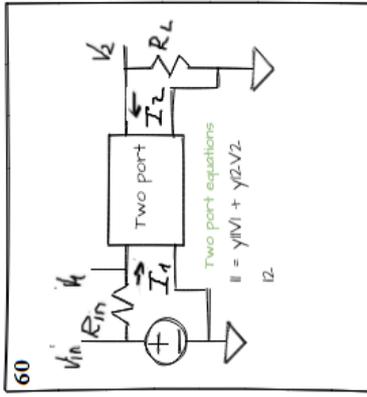
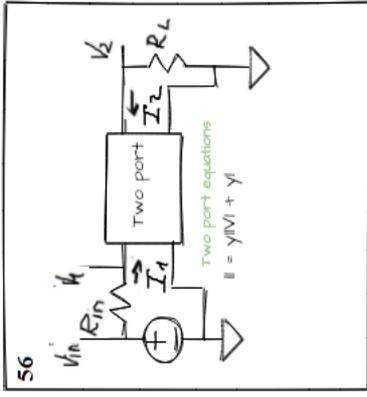
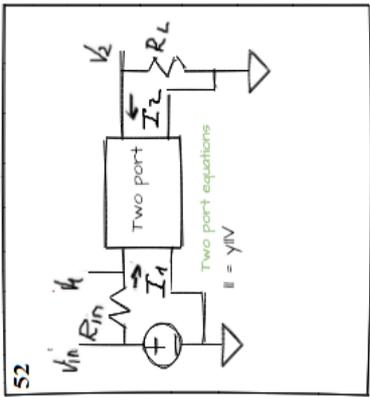
## A.2.1 Zeichenverlauf des Beitrags

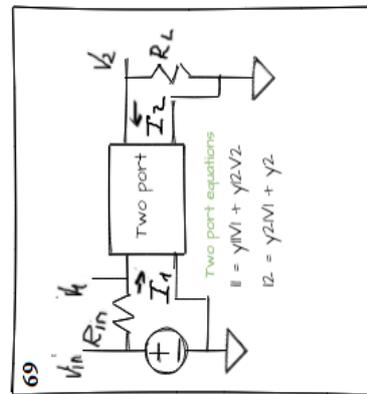
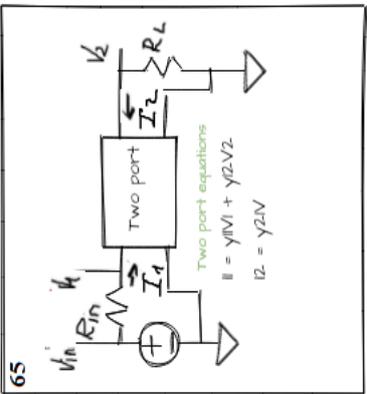
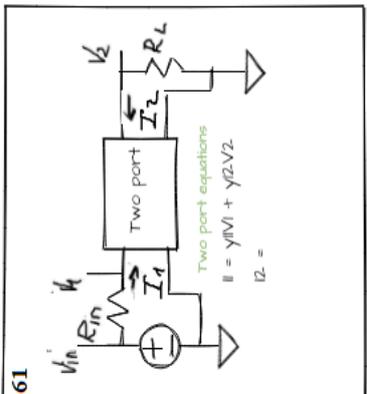
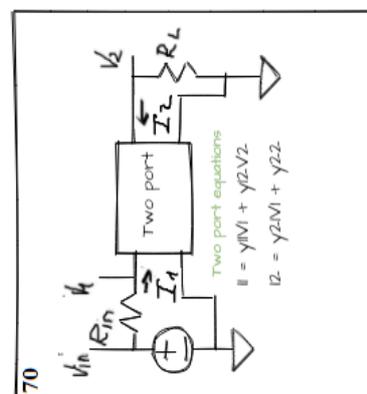
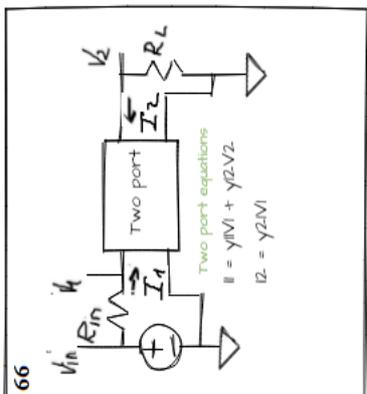
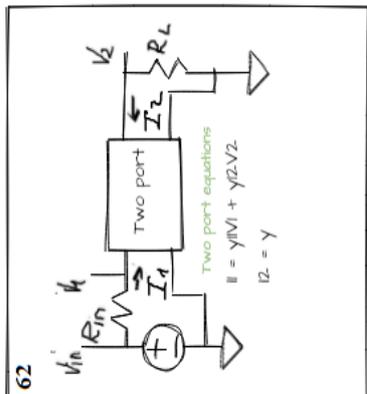
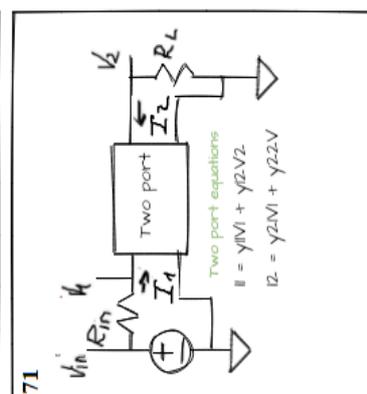
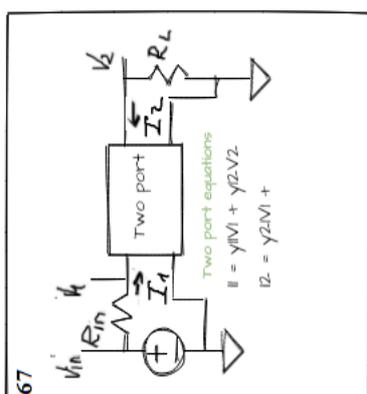
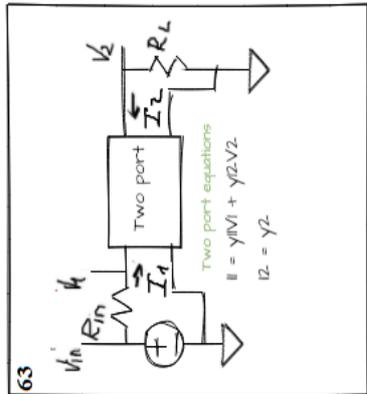
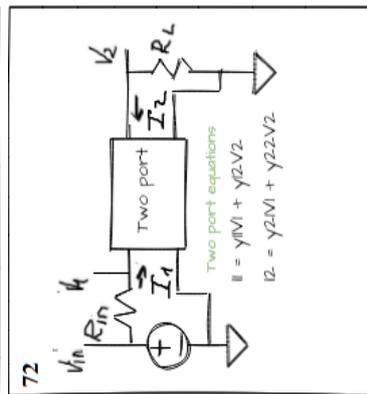
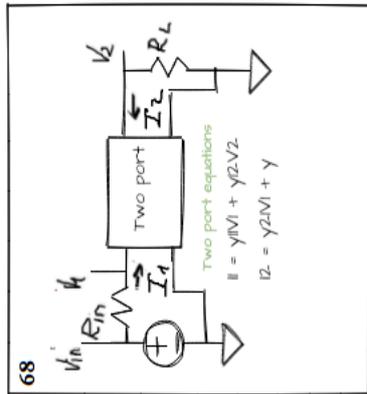
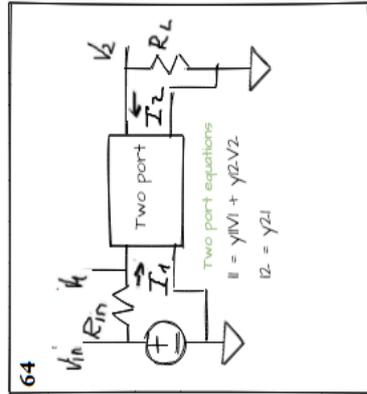


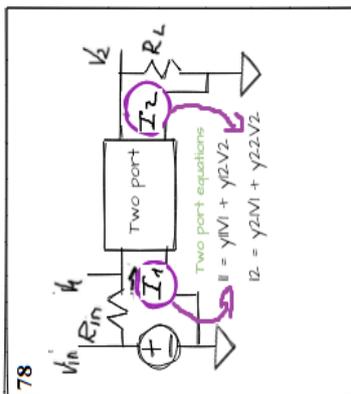
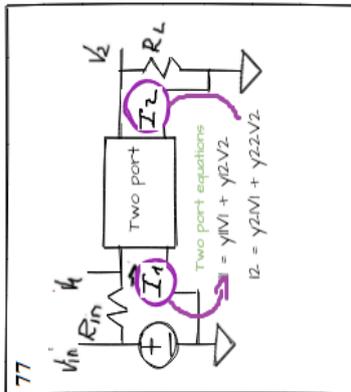
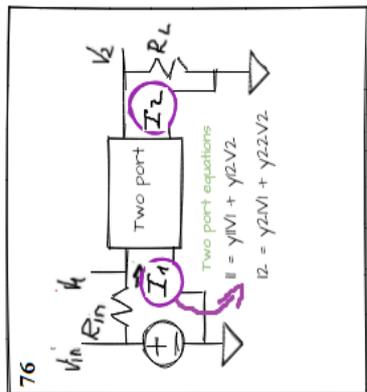
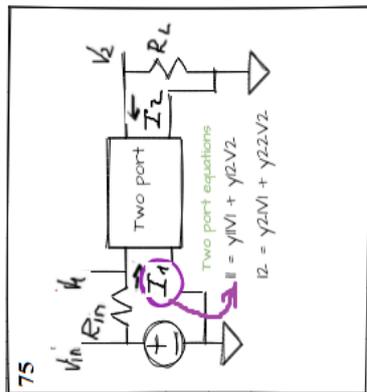
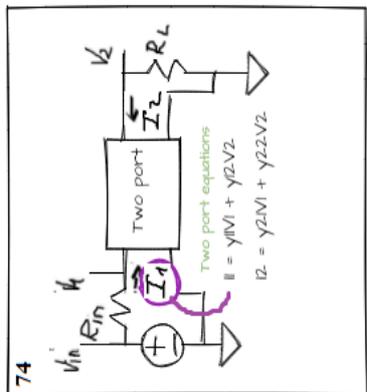
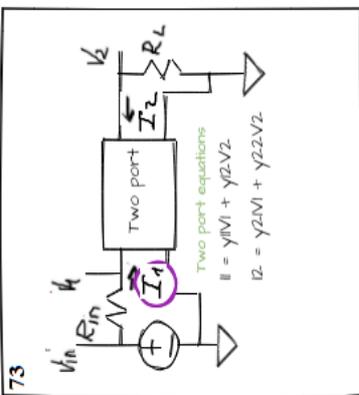












### A.3 EVALUATION

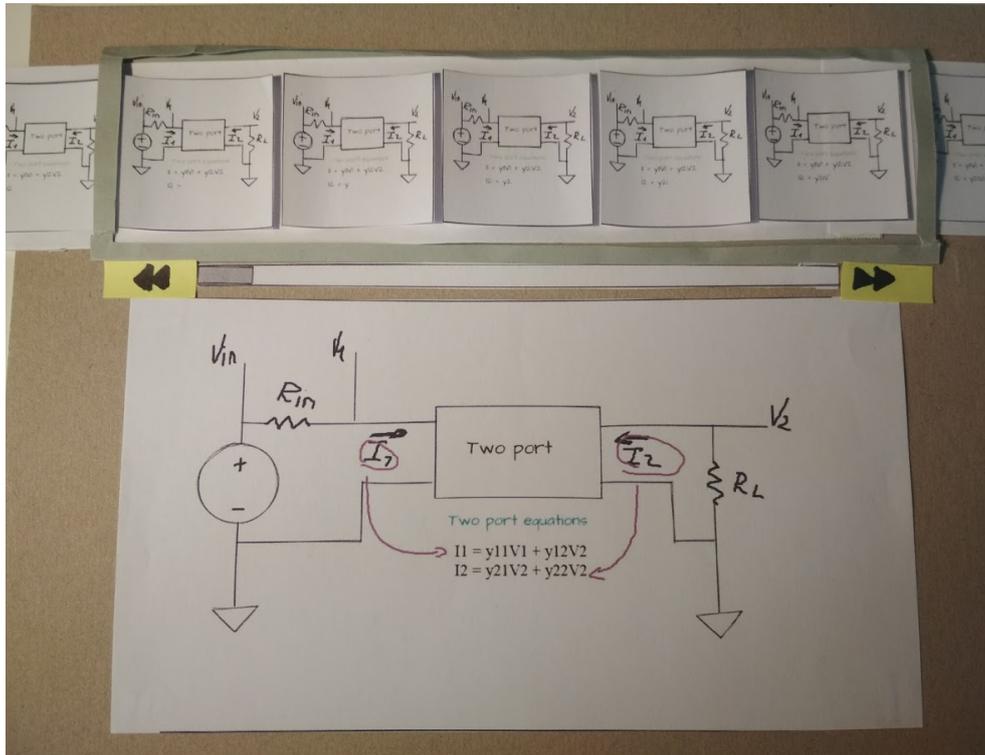


Abbildung A.8: Papier-Prototyp eines Slider-UIs

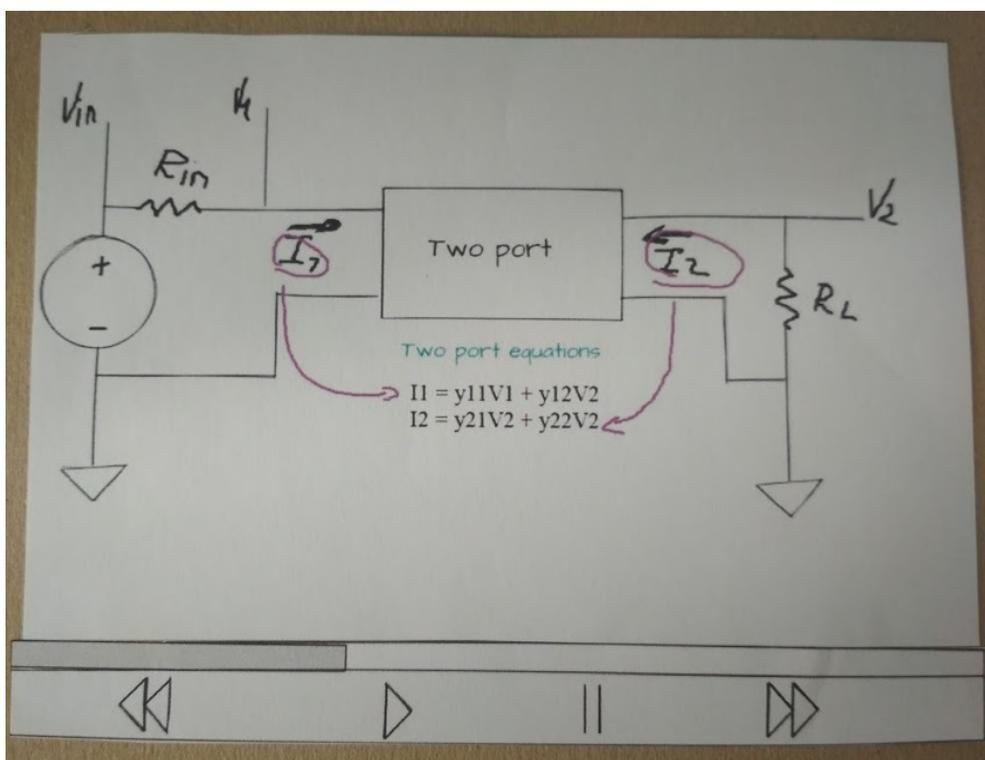


Abbildung A.9: Papier-Prototyp eines VideoPlayer-UIs

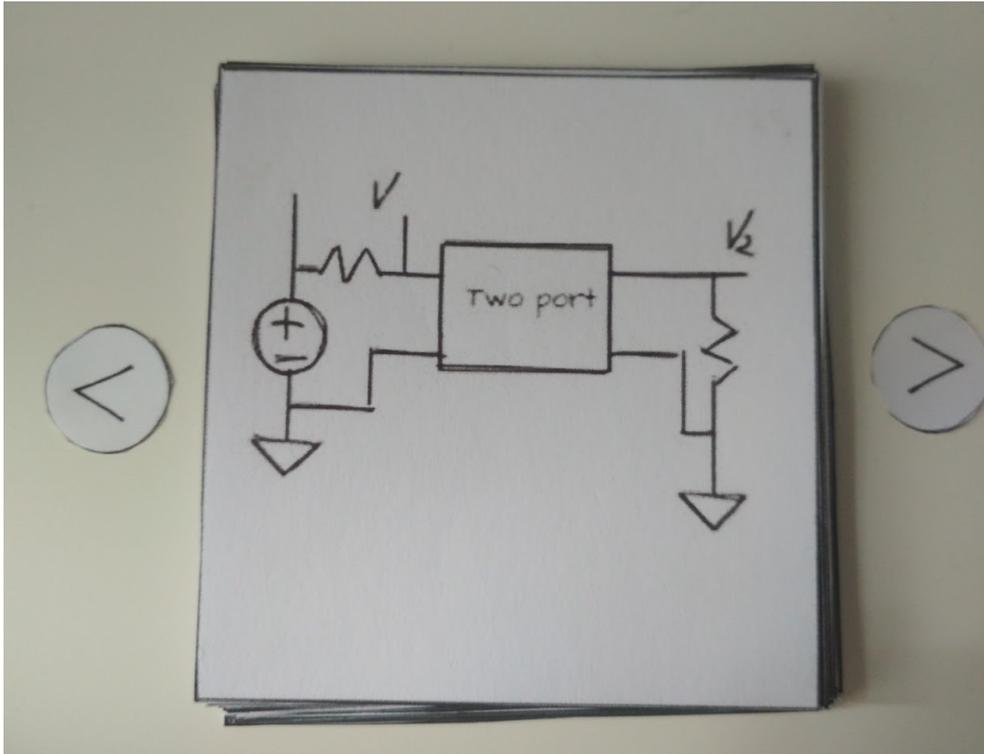


Abbildung A.10: Papier-Prototyp eines Slideshow-UIs

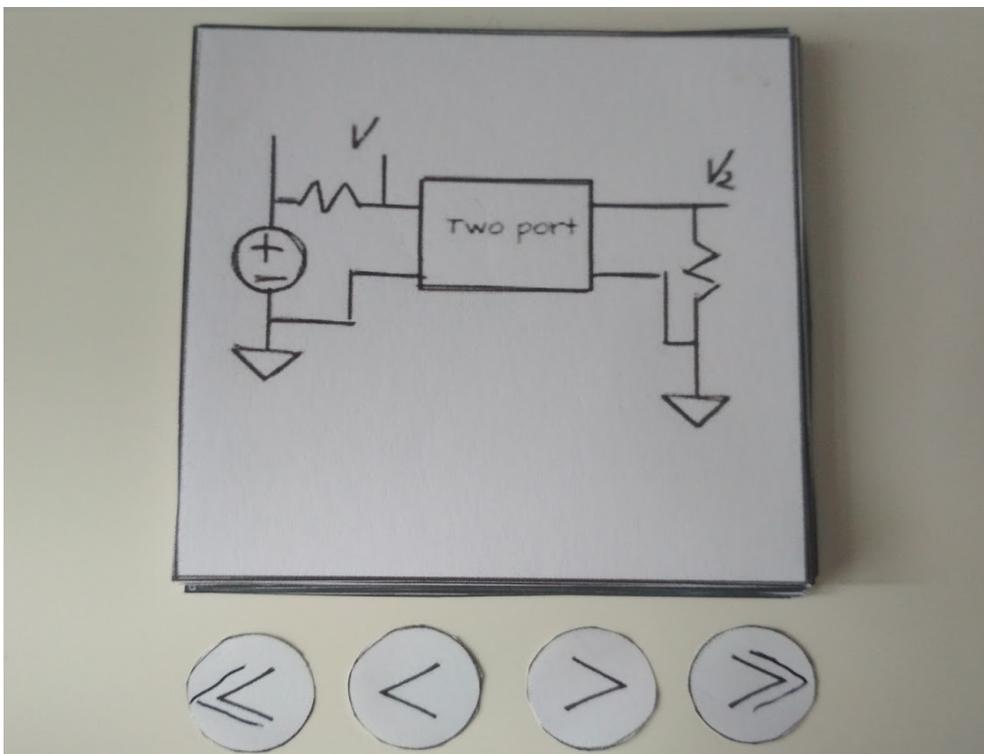


Abbildung A.11: Papier-Prototyp eines Slideshow-UIs mit den zusätzlichen Buttons