

Konzeption und Entwicklung von Modellierungsansätzen für digitale Therapien mittels Chatbots

Exposé zur Masterarbeit
von

Luisa Andre

Studiengang Informatik (M.Sc)
E-Mail: luisa.andre@student.kit.edu

Lehrstuhl für Pervasive Computing Systeme/TECO
Institut für Telematik
Fakultät für Informatik

Erstgutachter:
Zweitgutachter:
Betreuerin:

Prof. Dr. Michael Beigl
Derzeit noch offen
PD Dr. Andrea Schankin

Projektzeitraum: 01.01.2019 – 30.06.2019

1 Problemstellung

Sie geben Auskunft über das Wetter (vgl. [1]), nehmen Bestellungen entgegen (vgl. [2]) oder wirken als Coach (vgl. [3]) - Chatbots werden bereits vielseitig im Alltag eingesetzt. Auch die Psychologie profitiert von diesen Entwicklungen. 1966 entwickelte Joseph Weizenbaum mit *ELIZA* den ersten Chatbot. *ELIZA* sollte seinem menschlichen Gesprächspartner das Gefühl geben, dass dieser mit einem Psychiater über eine Chatoberfläche kommuniziert. Entwickelt wurde *ELIZA* allerdings nicht mit der Absicht Psychotherapie zugänglich zu machen, sondern um ein Modell zur maschinellen Verarbeitung von natürlicher Sprache zu implementieren (vgl. [4]). Was mit Joseph Weizenbaums *ELIZA* begann, brachte mit der Entwicklung der Forschung und Technik schließlich einige Chatbots, wie *Wysa* (vgl. [3]), *Woebot* (vgl. [5]) und *Tess* (vgl. [6]), im Bereich der psychischen Gesundheit hervor. Sie stellen heutzutage verschiedene Methoden der kognitiven Verhaltenstherapie bereit, die Nutzern helfen können, deren Introspektion zu verbessern. Dabei wirken sie wie ein Coach der jederzeit erreichbar ist (vgl. [7]).

In den 1960-ern hatten nur wenige Zugang zu Computern. Durch ihre Bauweise benötigten diese nicht nur viel Platz, sie waren zu dieser Zeit auch sehr kostspielig (vgl. [8]). Die Technik hat sich allerdings über die Jahrzehnte hinweg stark verändert. Nicht nur wurden Computer erschwinglich und haben eine deutlich größere Rechenleistung, sie begleiten uns mittlerweile auch in Form eines Tablets oder Laptops als Personal Computer durch den Alltag. Seit Apple ihr erstes Smartphone *iPhone* im Jahr 2007 einführte, eröffneten sich durch diese Geräte noch weitere technische Möglichkeiten. Smartphones entwickelten sich zu kleinen, handlichen Geräten die nahezu in jeder Tasche Platz finden (vgl. [9]). Außerdem beinhalten die Geräte heutzutage verschiedene Sensoren, haben Zugriff auf eine Vielzahl von Anwendungen und können sich mit dem Internet verbinden (vgl. [9][10]). Die Handlichkeit und Vielzahl an mitgebrachten Funktionen führte dazu, dass im Jahre 2018 allein in Deutschland 22,74 Millionen Smartphones verkauft wurden (vgl. [11]). Statistiken der *Bitkom Research* ermittelten, dass im Jahr 2017 78 Prozent der Deutschen ein Smartphone verwendeten (vgl. [12]).

Entwickler nutzen die technischen Vorteile der Smartphones und Personal Computer. So begleitet *Woebot* Menschen mit Depressionen oder inneren Unruhen mit Techniken aus der kognitiven Verhaltenstherapie als Selbsthilfe durch den Alltag (vgl. [7]). Der Nutzer kann dabei auswählen, ob dieser über eine *iPhone*-App, *Android*-App oder via *Facebook Messenger* mit *Woebot* kommunizieren möchte (vgl. [5]). Letzteres ist auf jedem browserfähigen Gerät nutzbar.

Eine Studie der Stanford School of Medicine untersuchte den Einsatz des Chatbots *Woebot* hinsichtlich seiner Realisierbarkeit, Nutzerakzeptanz und die vorläufige Wirksamkeit des bereitgestellten Selbsthilfeprogramms. Verglichen wurden dabei zwei Gruppen. Eine dieser Gruppen, bestehend aus 31 Probanden, erhielt Zugriff auf *Woebot*. Die zweite Gruppe, bestehend aus 25 Probanden, erhielt Zugriff auf das Ebook *Depression* des *National Institute of Mental Health*. Die Studiendauer wurde auf zwei Wochen festgelegt. Nach Ablauf der Studie zeigte sich, dass die Mehrheit der *Woebot*-Gruppe beinahe täglich den Chatbot nutzte. Auch konnte bei der Nutzung des *Woebots* im Vergleich zur Nutzung des Ebooks eine größere Zufriedenheit festgestellt werden. Außerdem ließ sich bei dieser Gruppe ein signifikanter,

positiven Einfluss hinsichtlich ihrer Depressionsbewältigung und dem Umgang mit inneren Unruhen messen (vgl. [7]).

Eine weitere Studie testete den Einfluss eines virtuellen Akteurs auf das Nutzerverhalten innerhalb eines klinischen Interviews. In dieser Studie wurden 145 Probanden in zwei Gruppen eingeteilt. 57 dieser Probanden führten einen Dialog mit einem virtuellen Akteur, der von einem Menschen gesteuert wurde. Die restlichen 88 Probanden unterhielten sich mit einem virtuellen Akteur, der mittels künstlicher Intelligenz kommunizierte. Das jeweilige Setting der Gruppen war allen Probanden bekannt. Gemessen wurde, unter anderem anhand eines Fragebogens, die Furcht vor negativer Bewertung (FNE), das Selbstdarstellungsverhalten (IM), die Nutzbarkeit des Systems (SU) sowie die Selbsttäuschung der Probanden (SD). Die Ergebnisse zeigten auf, dass signifikante Unterschiede zwischen den Gruppen gemessen werden konnte. So wurde festgestellt, dass Probanden, die Dialoge mit der künstlichen Intelligenz führte, einen niedrigeren FNE und IM Wert aufweisen (vgl. [13]).

Diese Ergebnisse zeigen auf, dass Chatbots im Bereich der Psychologie und Psychotherapie nützliche Werkzeuge sein können. Allerdings ist das Entwickeln solcher Chatbots für Psychologen noch immer eine Hürde. Zwar gibt es zahlreiche Baukästen zur Entwicklung von Chatbots die keine tiefgreifenden Programmierkenntnisse benötigen. Diese sind jedoch überwiegend auf den Bereich des Marketings ausgerichtet, weshalb sie in ihrem Funktionsumfang meist eingeschränkt sind. Baukästen die mehr Funktionalität bieten, benötigen lange Einarbeitungszeit und Expertenwissen in Bezug auf ihre Programmierung. Eine einfache und schnelle Umsetzung ist daher oft nicht möglich. Auch die Entwicklung eines eigenen Produktes birgt für Psychologen und Softwareunternehmen Probleme. So scheitert die Umsetzung unter anderem an Kommunikationshürden zwischen Entwicklern und Psychologen. Aber auch die komplexen Anforderungen des Medizinproduktegesetzes (MPG), die medizinische Produkte für die Herstellung oder Einführung in den Europäischen Wirtschaftsraum zu erfüllen haben, stellen eine Hürde dar (vgl. [14]).

Das Unternehmen *movisens GmbH* entwickelt derzeit das Projekt *TherapyBuilder* welches Psychologen und Psychotherapeuten die Möglichkeit bieten soll, Chatbots für Studien sowie zur Therapiebegleitung einzusetzen. Im Rahmen dieser Masterarbeit wird für das Projekt *TherapyBuilder* ein Modellierungsansatz *TMA* (Therapy Modelling Approach) entwickelt. Ziel dieses *TMA* ist es, Psychologen die Autonomie zu geben, ohne Expertenwissen Chatbots zu erstellen, um diese in Studien und therapiebegleitend einzusetzen.

2 Zielsetzung & Erkenntnisinteresse

Ziel der Arbeit ist die Konzeption einer Therapiemodellierungsansatz (*TMA*). Diese soll es erlauben, auch technisch wenig versierten Psychologen ihre Therapieideen in einer Art und Weise zu formulieren, die von einer Maschine verarbeitet und ausgeführt werden kann. Dadurch entfällt der hohe und fehleranfällige Abstimmungsaufwand zwischen Forschern und Entwicklern.

Durch den Einsatz der *TMA* sollen *MPG* konforme Anwendungen mit einem Chatbot UI entstehen, welche eine für den Patienten vertraute, dialogähnliche Kommunikation ermöglicht. Dies erlaubt es eine stärkere persönliche Bindung zwischen App und dem Patienten herzustellen, was den Therapieerfolg unterstützen soll.

In der Arbeit gilt es vor allem die komplexen Konfigurationsmöglichkeiten der Domäne einer digitalen Therapie funktional abzubilden. Durch eine Befragung der Anwendergruppen soll Domänenwissen erarbeitet werden und im Folgenden die *TMA* und dessen grafische Repräsentation iterativ entworfen werden. Dabei gilt ein hohes Augenmerk der Usability, um sicherzustellen, dass der Aufwand zur Therapieentwicklung und Studiendurchführung nicht größer ist, als derzeitige Methoden zur Therapieentwicklung und Studiendurchführung.

Es werden zwei Modellierungsansätze entwickelt. Einer dieser Ansätze wird in Form eines komplexen Mockups umgesetzt. Ein weiterer in Form eines Prototypen, welcher aus der Anpassung eines Experience Sampling Tools resultiert. Anschließend erfolgt eine Evaluation der Entwürfe in einer Vergleichsstudie.

3 Stand der Technik

Für die Bearbeitung der Forschungsfrage werden verschiedene Technologien bewertet. Derzeit existiert noch keine bekannte Sprache, die speziell zur Modellierung von Therapien mit Chatbots entwickelt wurden. Aus diesem Grund werden nachfolgend Ansätze betrachtet, die für eine Umsetzung eines *TMA* in Frage kommen. Zunächst werden Plattformen beleuchtet, die eine Erstellung von Chatbots ermöglichen, ohne weitere Programmierkenntnisse zu benötigen.

3.1 Grafische Programmiersprachen

Diese Art der Programmiersprache bedient sich visueller Elemente um Programmstrukturen verständlich abzubilden. Die visuellen Elemente können auf eine bestimmte Domäne zugeschnitten sein (vgl. [15]) oder beschränken sich auf die Visualisierung gängiger Programmanweisungen (vgl. [16]). In den folgenden Abschnitten werden Chatbot-Plattformen und allgemeine grafische Programmiersprachen betrachtet.

3.1.1 Chatbot-Plattformen

Der Konversationsfluss der Chatbots wird auf den Plattformen, wie *Dialogflow* und *IBM Watson* unter anderem als eine Art Baum, ähnlich zur bekannten Ordnerstruktur unter Windows Betriebssystemen, angelegt und dargestellt (vgl. [17] [18]). Chatbot-Plattformen, wie *ManyChat*, *Converse.ai* und *Chatfuel* verwenden Diagramme zur Darstellung eines Chatverlaufes (vgl. [19] [20]) oder Blocksysteme mit Referenzen auf nachfolgende Blöcke (vgl. [21]). Andere Plattformen nutzen keine Darstellung des Verlaufs (vgl. [22]). Im Beispiel von *Botsify* oder *Recast.ai* werden nur Verhaltensweisen angelegt, die durch bestimmte Nutzereingaben getriggert werden.

Auch in der Handhabung der Nutzereingaben gibt es verschiedene Ansätze. So bieten manche Plattformen die Möglichkeit Antworten für den Nutzer des Chatbots vorzugeben (vgl. [21] [20]). Andere hingegen verwenden natürliche Sprachverarbeitung um Schlagwörter einzutrainieren. Der Ersteller des Chatbots legt fest, wie der Chatbot auf die entsprechenden Schlagwörter reagiert (vgl. [22]. [17] [18]) Die Chatbot-Plattform *Chatfuel* verwendet beide Ansätze. So können hier Antworten vordefiniert oder Schlagwörter festgelegt werden (vgl. [21]).

Damit Nutzerdaten abgespeichert und verarbeitet werden können, bieten einige Plattformen Variablen an. Dort können unter anderem Nutzernamen sowie Aufenthaltsort des Nutzers gespeichert und weiterverwendet werden. Der Nutzer kann auf bereits definierte Variablen zurückgreifen oder eigene anlegen (vgl. [21] [19] [17] [18] [20]).

3.1.2 Allgemeine grafische Programmiersprachen

Neben dem Einsatz von grafischer Programmierung von Chatbots, gibt es noch weitere Domänen die ebenfalls die grafische Programmierung verwenden. Die grafische Programmiersprache *Labview* beispielsweise, konzentriert sich auf die Domäne System-, Steuer- und Regelungstechnik (vgl. [15]). Programmiert wird, indem Elemente miteinander kombiniert werden, die als Schaltzeichen aus der Elektrotechnik bekannt sind. Nach diesem Prinzip arbeiten auch die Editoren *Matlab Simulink* und *Choreograph* (vgl. [23] [24]).

Ist keine domänenspezifische grafische Programmiersprache gewünscht oder bekannt, ist es dennoch möglich ohne tiefgreifende Programmierkenntnisse Programme zu entwickeln. Ermöglicht wird dies durch Programmiersprachen, die Programmanweisungen durch Diagramme oder Blöcke visualisieren. Für Diagramme werden unter anderem Zustandsdiagramme oder eine Form von Flussdiagrammen verwendet (vgl. [25] [26] [27]). Durch diese Vorgehensweisen lassen sich Schleifen oder bedingte Anweisungen leicht erkennen. Eine Visualisierung mit Blöcken hingegen folgt dem Steckprinzip. So können Anweisungen in Blockform nebeneinander wie untereinander angeordnet werden. Schleifen oder Bedingungen werden durch Blöcke dargestellt, die andere Blöcke beinhalten. Diese Blöcke stellen Anweisungen dar, die innerhalb dieser Schleife oder jeweiligen Bedingung ausgeführt werden (vgl. [28] [29] [30] [31]). *Lego Mindstorms NXT* verbindet das Steckprinzip der Blöcke mit domänenspezifischen Elementen der Lego Mindstorms Bausätze. Insbesondere Schleifen und Bedingungen werden als eine Art Blocksystem genutzt (vgl. [29]).

3.2 Auszeichnungssprachen

Eine weitere Möglichkeit zur komplexen Programmierung sind die sogenannten vereinfachten Auszeichnungssprachen. Diese arbeiten mit Text der anhand einfacher Befehle formatiert und strukturiert wird. So kann anhand eines vorangehenden Symbols Text als Überschrift definiert werden. Insbesondere *Markdown* verwendet Sonderzeichen um Textabschnitte zu formatieren und strukturieren (vgl. [32]).

Auch *YAML* nutzt Sonderzeichen, um Listen und größere Mengen von Daten zu beschreiben (vgl. [33]). *BBCode* hingegen verwendet einfache Anweisungen die mit eckigen Klammern eingeleitet und abgeschlossen werden. Die Anweisung selbst wird in Form eines Buchstabens angegeben (vgl. [34]).

HTML ist die geläufigste Auszeichnungssprache. Diese wird zur Strukturierung von Websites benötigt. Dabei können verschiedene Bereiche definiert und deren Inhalte strukturiert werden. *HTML* hat die Fähigkeit durch die Verwendung von Tags komplexe Inhalte, wie Texte, Bilder, Listen und Tabellen zu strukturieren und zu formatieren. Die Tags werden mit spitzen Klammern gekennzeichnet. Im Rahmen einer Studie wurde *HTML* eingesetzt um Ambulante Assessment Protokolle zu erstellen, die sowohl vom Menschen lesbar als auch vom Computer ausführbar sind (vgl. [35]).

3.3 Experience Sampling

Psychotherapeuten und Psychologen haben die Möglichkeit anhand bestimmter Experience Sampling Software Fragebögen für mobile Geräte zu entwickeln (vgl. [36]). Hierbei werden auch Lösungen angeboten, die Auszeichnungssprachen verwenden. Die Software *Experience Sampler* verwendet die Auszeichnungssprache *JSON*, aufbauend auf *YAML*, um Fragen, Anzeige- sowie Eingabeformate zu definieren (vgl. [36]). *MyExperience* verwendet einen ähnlichen Ansatz. Als Auszeichnungssprache zur Entwicklung der Fragebögen wird hier auf *XML* zurückgegriffen (vgl. [37]).

Ein weiteres Projekt zur Erstellung von Experience Sampling ist *Jeeves*. Fragebögen werden mit diesem Programm über eine grafische Programmiersprache definiert. Verwendet wird hauptsächlich die grafische Programmierung mit Blöcken. Über eine weitere Oberfläche werden die Eingabeformate der Antworten festgelegt. So ist es möglich Formate wie Likert Skala, Checkboxes, Radiobuttons, Ortsabfragen und weitere zu verwenden (vgl. [38]).

Die Experience Sampling Software *movisensXS* nutzt Diagramme zur Beschreibung des Ablaufs eines Fragebogens. Diese werden nach einem Puzzle-Prinzip angeordnet. Die Fragen selbst, sowie Formate der Antworten, werden separat angelegt und können später im Diagramm ausgewählt werden (vgl. [39]).

3.4 Fazit

Zwar bieten Chatbot-Plattformen bereits einige Funktionen, allerdings fokussieren diese sich vornehmlich auf Marketing, Vertrieb und Support (vgl. [21] [19] [40]). Dies kann die Umsetzung einer komplexen Therapie erschweren. Übliche Elemente, wie visuelle Analogskalen und Likert-Skalen, die häufig in der Psychologie Verwendung finden, können nur schwer oder gar nicht umgesetzt werden. Die Darstellungsformen der Konversationen verschiedener Chatbot-Plattformen haben unterschiedliche Vor- und Nachteile. Bäume und Diagramme bieten eine visuelle und leicht verständliche Übersicht eines Konversationsablaufs. Je größer und komplexer dieser Ablauf allerdings wird, umso unübersichtlicher wird eine Konversation. Bei großen Bäumen und Diagrammen werden in der Gesamtansicht die einzelnen Komponenten und Schriften zu klein und somit schwer lesbar für das menschliche Auge. Ist die Funktion eines hinein- und herauszoomens implementiert, erschwert sich das Verorten der Komponenten im Gesamtsystem (vgl. [41]). Blocksysteme bieten zusätzlich die visuelle Darstellung von Bedingungen, aber auch hier ist ein großer Konversationsablauf schnell unübersichtlich (vgl. [41]). Viele Elemente, die in Chatbot-Plattformen eingesetzt werden, könnten für einen Therapiemodellierungsansatz nützlich sein, da diese leicht nachvollziehbar und leicht in der Handhabung sind. Allerdings ist keine der bisherigen Chatbot-Plattformen derzeit geeignet, um eine komplexe Therapie umzusetzen ohne lange Einarbeitungszeit oder Einschränkungen in der Gestaltung.

Auch eine Umsetzung mit den sogenannten grafischen Programmiersprachen wäre möglich um eigene Chatbots zu entwickeln. Überwiegend gibt es diese für spezielle Domänen wie Elektrotechnik. Das Baukastenprinzip ist hier besonders interessant, da es verschiedene Funktionen visuell darstellt und später in Code übersetzt. Die visuelle Darstellung ist leicht verständlich und schnell zu Erlernen. *Blockly* bedient sich diesem Prinzip, um verschiedene Arten der Programmanweisungen verständlicher darzustellen. Allerdings erhält in dieser Form der Umsetzung ein komplexeres Programm oder System die zuvor genannten Probleme (vgl. [41]). Es gibt ebenfalls

noch keine grafische Programmiersprache, die zur Umsetzung eines Therapiemodellierungsansatzes geeignet wäre.

Für das Beschreiben einer Konversation wäre auch die Nutzung einer Auszeichnungssprache möglich. Das Schreiben eines Konversationsfluss wirkt hier sehr natürlich, da es dem Chatten nahe kommt. Aber auch hier kann man leicht die Übersicht verlieren, da Verzweigungen in Konversationen nicht entsprechend dargestellt werden können, wie es beispielsweise bei Diagrammen möglich ist. Auch liefern nicht alle Auszeichnungssprachen den Umfang von Funktionen um komplexe Therapien darzustellen. Auch die Syntax und Fehlersuche wird zeitaufwändig sofern das genutzte Programm zur Beschreibung keine oder eine rudimentäre Fehlerbehandlung mit sich bringt.

Im Bereich des Experience Samplings werden bereits verschiedene Ansätze verwendet, die eine grafische Programmierung oder das Verwenden einer Auszeichnungssprache integrieren. Hier liegt der Fokus auf der Entwicklung von Fragebögen die zu verschiedenen Zeiten getriggert werden. Derzeit gibt es keine Experience Sampling Software die Therapien gezielt in Form von Konversationen umsetzt. Jedoch können Fragebögen ein wichtiges Stilmittel einer Therapie darstellen.

Aufgrund der verschiedenen Vor- und Nachteile der vorgestellten Ansätze, wäre eine Kombination verschiedener Ansätze denkbar.

4 Konzept

Ziel ist es, ein Modellierungsansatz für digitale Therapien mittels Chatbots zu konzeptionieren und prototypisch umzusetzen. Dabei soll folgende Frage beantwortet werden: Wie lassen sich die Vor- und Nachteile der betrachteten Ansätze zur Modellierung und Strukturierung von Programmen und Fragebögen für die Therapie so einsetzen, damit eine Programmierung von Chatbots ermöglicht wird? Diese soll von einer Maschine verarbeitet und ausgeführt werden. Hierbei soll es möglich sein, auch mit wenigen Programmierkenntnissen, Chatbots zu erstellen um neue Therapiemöglichkeiten zu entwickeln. Die Modellierungssprache soll den Psychologen befähigen ohne tagelange Einarbeitungszeit eine Therapie zu entwickeln. Der Chatbot wird den späteren Nutzern über ein mobiles Endgerät zur Verfügung gestellt. Hierfür sollen zunächst die Anforderungen an eine Modellierungssprache für digitale Therapien mittels Chatbots in Gesprächen mit Psychotherapeuten und Psychiatern ermittelt werden. Anschließend werden verschiedene Konzepte auf dem Papier entwickelt und evaluiert. Das vielversprechendste Konzept soll im Laufe dieser Arbeit in Form eines Mockups umgesetzt und abschließend anhand einer Usability-Studie evaluiert werden.

5 Vorläufige Gliederung

1. Einleitung
 - (a) Problemstellung und Zielsetzung
 - (b) Umfeld
 - (c) Methodisches Vorgehen

- (d) Gliederung
- 2. Grundlagen
 - (a) Definitionen
 - (b) Rahmenbedingungen von Psychotherapien
 - (c) Chatbots
- 3. Stand der Forschung und der Technik
 - (a) Chatbot-Plattformen
 - (b) Grafische Programmiersprachen
 - (c) Auszeichnungssprachen
 - (d) Experience Sampling Software
- 4. Konzeption
 - (a) Anforderungsanalyse
 - (b) Ausarbeitung verschiedener Konzepte
 - i. Beschreibung dieser
 - ii. Evaluation
- 5. Entwicklung der Modellierungsansätze
 - (a) Konzept
 - (b) Umsetzung
 - (c) Evaluation
- 6. Ergebnisse
 - (a) Zusammenfassung
 - (b) Kritische Reflexion
- 7. Ausblick

Bibliography

- [1] *Google Allo - A smart messaging app*. <https://allo.google.com/>. (Accessed on 02/27/2019).
- [2] *Kassen-Schnittstelle Table Duck: Bestellen via Messenger | GASTROFIX*. <https://www.gastrofix.com/de/kassen-schnittstelle-table-duck/>. (Accessed on 02/27/2019).
- [3] *Wysa - your 4 am friend and AI life coach*. <https://www.wysa.io/>. (Accessed on 02/27/2019).
- [4] Joseph Weizenbaum. "ELIZA—A Computer Program For the Study of Natural Language Communication Between Man and Machine". In: (1966), pp. 1–7.
- [5] *Woebot - Your charming robot friend who is ready to listen, 24/7*. <https://woebot.io/>. (Accessed on 02/27/2019).
- [6] *Tess: Artificial Intelligence Mental Health Support*. <https://www.x2ai.com/>. (Accessed on 02/27/2019).
- [7] Kathleen Kara Fitzpatrick, Alison Darcy, and Molly Vierhile. "Delivering Cognitive Behavior Therapy to Young Adults With Symptoms of Depression and Anxiety Using a Fully Automated Conversational Agent (Woebot): A Randomized Controlled Trial". In: *JMIR Mental Health* 4.2 (2017), e19. ISSN: 2368-7959. arXiv: 9809069v1 [arXiv:gr-qc].
- [8] Martin Campbell-Kelly and William Aspray. *Computer : a history of the information machine*. 2. ed. Previous ed.: New York: Basic Books, 1996. Boulder: Westview, 2004. ISBN: 0-8133-4264-3.
- [9] Martin Campbell-Kelly and Daniel D. [author] Garcia-Swartz, eds. *From mainframes to smartphones : a history of the international computer industry*. Includes bibliographical references and index. - Print version record; The aim of this book is to provide a compact and up-to-date business and economic history of the computer industry. The reader we have in mind is someone who wants to make a quick study of the computer industry... In our quest for brevity, we have assumed that our readers are themselves information-technology users and are familiar with the everyday vocabulary of computing (such as operating systems, disk drives, and broadband), and that they are aware of recent gadgetry such as tablet computers and smartphones—Page 1. Cambridge, Massachusetts, 2015.
- [10] *App Store - Apple (DE)*. <https://www.apple.com/de/ios/app-store/>. (Accessed on 03/05/2019).

- [11] *Zukunft der Consumer Technology – 2018*. <https://www.bitkom.org/sites/default/files/file/import/180822-CT-Studie-2018-online.pdf>. (Accessed on 03/06/2019).
- [12] *Smartphone-Markt: Konjunktur und Trends*. <https://www.bitkom.org/sites/default/files/file/import/Bitkom-Pressekonferenz-Smartphone-Markt-Konjunktur-und-Trends-22-02-2017-Praesentation.pdf>. (Accessed on 05/28/2019).
- [13] Jonathan Gratch and Gale Lucas. “It ’ s Only a Computer : The Impact of Human-agent Interaction in Clinical Interviews”. In: Aamas (2014).
- [14] *MPG - nichtamtliches Inhaltsverzeichnis*. <http://www.gesetze-im-internet.de/mpg/>. (Accessed on 05/28/2019).
- [15] *Was ist LabVIEW? - National Instruments*. <http://www.ni.com/de-de/shop/labview.html>. (Accessed on 11/27/2018).
- [16] *Blockly | Google Developers*. <https://developers.google.com/blockly/>. (Accessed on 11/27/2018).
- [17] *Dialogflow*. <https://console.dialogflow.com/api-client/#/login>. (Accessed on 11/27/2018).
- [18] *Katalog - IBM Cloud*. <https://console.bluemix.net/catalog/?category=ai&search=label:lite>. (Accessed on 11/27/2018).
- [19] *Converse.AI*. <http://www.converse.ai/>. (Accessed on 11/27/2018).
- [20] *ManyChat*. <https://manychat.com>. (Accessed on 11/27/2018).
- [21] *Chatfuel dashboard*. <https://dashboard.chatfuel.com/#/bot/5bea8fd30ecd9f4c8c0dd458/structure/5bea8fd30ecd9f4c8c0dd466>. (Accessed on 11/27/2018).
- [22] *Botsify - Create artificial intelligent chatbots without coding*. <https://botsify.com/>. (Accessed on 11/27/2018).
- [23] *Choregraphe Suite — Aldebaran 2.4.3.28-r2 documentation*. <http://doc.aldebaran.com/2-4/software/choregraphe/index.html>. (Accessed on 11/28/2018).
- [24] *Simulink - Simulation und Model-Based Design - MATLAB & Simulink*. <https://de.mathworks.com/products/simulink.html>. (Accessed on 11/28/2018).
- [25] *SwissEduc - Informatik - JavaKara: Einführung in Java*. <https://www.swisseduc.ch/informatik/karatojava/javakara/>. (Accessed on 11/28/2018).
- [26] *DRAKON Editor*. <http://drakon-editor.sourceforge.net/>. (Accessed on 11/28/2018).
- [27] *Pure Data — Pd Community Site*. <https://puredata.info/>. (Accessed on 11/28/2018).
- [28] *Blockly | Google Developers*. <https://developers.google.com/blockly/>. (Accessed on 11/28/2018).
- [29] *NXT Software Download - Downloads - Mindstorms LEGO.com*. <https://www.lego.com/de-de/mindstorms/downloads/nxt-software-download>. (Accessed on 11/28/2018).
- [30] *Snap! (Build Your Own Blocks) 4.2*. <https://snap.berkeley.edu/>. (Accessed on 11/28/2018).

- [31] *squeakland : home of squeak etoys*. <http://www.squeakland.org/>. (Accessed on 11/28/2018).
- [32] *Getting Started | Markdown Guide*. <https://www.markdownguide.org/getting-started/>. (Accessed on 11/28/2018).
- [33] *The Official YAML Web Site*. <http://yaml.org/>. (Accessed on 11/28/2018).
- [34] *BBCode.org, bbcode users guide and tricks on implementing it*. <https://www.bbcode.org/>. (Accessed on 11/28/2018).
- [35] Nikolaos Batalas et al. “Formal representation of ambulatory assessment protocols in HTML5 for human readability and computer execution”. In: *Behavior Research Methods* (2018). ISSN: 15543528.
- [36] *OSF | Sabrina Thai - ExperienceSampler SPSP San Antonio 2017 Draft 1 FINAL To Post.pdf*. <https://osf.io/3am7g/>. (Accessed on 11/28/2018).
- [37] *the MyExperience tool*. <http://myexperience.sourceforge.net/>. (Accessed on 11/28/2018).
- [38] Daniel Rough and Aaron Quigley. “Jeeves - An Experience Sampling Study Creation Tool”. In: *BCS Health Informatics Scotland (HIS)* (2017), pp. 1–10.
- [39] *movisensXS - eXperience Sampling for Android!* <https://xs.movisens.com/>. (Accessed on 11/28/2018).
- [40] *ManyChat - Startseite*. <https://www.facebook.com/ManyChat/>. (Accessed on 03/06/2019).
- [41] Kasper Hornbæk et al. “NAVIGATION PATTERNS AND USABILITY OF ZOOMABLE USER INTERFACES WITH AND WITHOUT”. In: (2003).

