

Arbeit zur Erlangung des akademischen Grades
Bachelor of Science

**Graphisch-interaktive
Entwicklungsumgebung für
Robotersysteme zum Einsatz in den
Rehabilitationswissenschaften**

Benjamin Matthias Biehler
geboren in Nürnberg

2021

Lehrstuhl für Computergraphik VII
Fakultät Informatik
Technische Universität Dortmund

Erstgutachter: PD Dr. Weichert
Zweitgutachter: Prof. Dr.-Ing. Bühler
Abgabedatum: 31. September 2021

Kurzfassung

Hier steht eine Kurzfassung der Arbeit in deutscher Sprache inklusive der Zusammenfassung der Ergebnisse. Zusammen mit der englischen Zusammenfassung passt sie auf eine Seite.

Abstract

The abstract is a short summary of the thesis in English, together with the German summary it has to fit on this page.

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Einleitung | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | Struktur der Arbeit | 1 |
| 2 | Mensch-Roboter-Interaktion | 2 |
| 2.1 | Herausforderungen | 2 |
| 2.2 | Stand der Technik | 2 |
| 2.3 | Kollaborations-Schnittstelle | 2 |
| 3 | Interaktive Visuelle Robotersteuerung | 3 |
| 3.1 | Endbenutzer-Entwicklung | 3 |
| 3.2 | No-Code Plattformen | 4 |
| 3.3 | Visuelle Programmierung | 4 |
| 3.4 | Iterativer Entwicklungszyklus | 5 |
| 4 | Roboter-Ansteuerung | 6 |
| 4.1 | Sprachausgabe | 6 |
| 4.2 | Bewegungssteuerung | 6 |
| 4.3 | Spracherkennung | 6 |
| 5 | Benutzerzentrierte Auswertung | 7 |
| 5.1 | Gestaltung der Anforderungserhebung | 7 |
| 5.2 | Instanziierung der Testsettings | 10 |
| 5.3 | Operationalisierung | 10 |
| 6 | Evaluierung | 11 |
| 6.1 | Anforderungsanalyse | 11 |
| 6.2 | Statistische Auswertung | 11 |
| 6.3 | Softwarequalität | 11 |
| 7 | Zusammenfassung und Ausblick | 12 |
| 7.1 | Zusammenfassung | 12 |
| 7.2 | Ausblick | 12 |
| A | Statistik | 13 |

| | |
|------------------------------|-----------|
| Literatur | 14 |
| Abbildungsverzeichnis | 15 |

1 Einleitung

Kurze Einleitung in die Thematik der Bachelorarbeit (Thema vorstellen, Leser*in neugierig machen, Relevanz der BA in Kontext setzen). Das ganze Kapitel sollte auf etwa 3 Seiten machbar sein.

Was macht diese Arbeit interessant?

1.1 Motivation

Wieso ist die entstandene Entwicklungsumgebung entstanden? Was war der Anstoß dazu den Ansatz zu verfolgen? Wer ist die Zielgruppe für diese Anwendung?

1.2 Struktur der Arbeit

Wie ist diese Arbeit aufgebaut? Was ist in den einzelnen Kapiteln zu erwarten? (Wie war die Arbeit aufgebaut?)

2 Mensch-Roboter-Interaktion

2.1 Herausforderungen

Was sind die Probleme, die mit der Anwendung gelöst werden sollen? Was waren die Hindernisse, die sich für diese Arbeit gestellt haben?

2.2 Stand der Technik

Werden ähnliche Tools bereits eingesetzt, wenn ja mit welchem Erfolg? Was war der bisherige Ansatz in dieser Domäne (Pepper-Programmierung)?

2.3 Kollaborations-Schnittstelle

Was ergibt sich aus 2.1 und 2.2 für die Arbeit? Welche grobe Zielsetzung wurde festgelegt?



Abbildung 2.1: Platzhalter für Abbildungsverzeichnis.

3 Interaktive Visuelle Robotersteuerung

1-2 Absätze die dieses Kapitel einleiten (wahrscheinlich längstes Kapitel der Arbeit).

Beschreibung des GUIs und der Prinzipien dahinter. Unterkapitel gerne nochmal durchmischen.

3.1 Endbenutzer-Entwicklung

Endbenutzer-Entwicklung ist, nach Lieberman et al. [3], ein aktuelles Forschungsgebiet der Mensch-Maschine-Interaktion, das sich damit beschäftigt Methoden, Techniken und Werkzeuge zu entwickeln, die es Endnutzern, anstelle von professionellen Software-Entwicklern, ermöglichen in irgendeiner Weise Teile von Software anzupassen, erweitern oder sogar eigenständig zu entwickeln.

Die Domäne der Endbenutzer-Entwicklung erstreckt sich von der Automatisierung von Tabellen in Tabellenkalkulations-Programmen und einfachen Makros in Textbearbeitungsprogrammen [6], über mobile Anwendungen [8] oder Webseiten [5], bis hin zu der Entwicklung von kritischen Anwendungen im Gesundheitssystem [4]. Aktuelle Schätzungen in den USA zufolge arbeiten 90 Millionen Endbenutzer mit einer Form von Endbenutzer-Entwicklung, während nur 3 Millionen professionelle Programmierer dort arbeiten [6]. aktuellere Daten?

Diese Weise Software zu entwickeln bringt jedoch auch viele Probleme, wie etwa Sicherheitslücken, mit sich [1]. Um diese und andere kritische Fehler, die in der Programmierung der jeweiligen Anwendungen vorkommen können, zu verhindern muss die EUD-Programmierungsumgebung die Sicherheit und Korrektheit der entstehenden Software sicher stellen, und mögliche Fehler des Endbenutzers finden oder korrigieren können. Bei der Entwicklung von Anwendungen für soziale Roboter sind Sicherheitsbedenken weniger relevant, da die Roboter nicht auf Daten arbeiten, die von kritischer Bedeutung sind und das mögliche Verhalten der Roboter klar definiert ist.

3.1.1 Arten von Endbenutzer-Entwicklung

Ebenso verschieden, wie die Anwendungsbereiche von Endbenutzer-Entwicklung sind die verschiedenen Ansätze sie zu ermöglichen. Die am weitesten verbreitete Form der Endbenutzer-Entwicklung ist die Verwendung von Makros und Tabellenkalkulation [6]. Hier vereinfachen Endnutzer häufig repetitive Aufgaben mit Hilfe von einfachen Kalkulationen oder automatisierten Schritten. Da dieser Ansatz jedoch meist direkt in eine Anwendung integriert ist, hat er kaum Bedeutung für die Entwicklung von Anwendungen.

Low-Code Plattformen verringern, meistens durch graphische Benutzerschnittstellen, den Bedarf an von Hand geschriebenen Code. Je nach Plattform müssen jedoch einige Teile von Anwendungen noch von Hand geschrieben werden. Dies vereinfacht und beschleunigt die Entwicklung für professionelle Programmierer und senkt die Barriere für Endbenutzer. Im Falle von No-Code Plattformen entfällt der Bedarf Code zu schreiben komplett und macht es Endbenutzern möglich diese Plattformen eigenständig zu benutzen um selbst Anwendungen anzupassen, erweitern oder zu erstellen.

Programming By Example ist..... 10.8.

3.1.2 Endbenutzer-Entwicklung in den Rehabilitationswissenschaften

3.2 No-Code Plattformen

Was sind No-Code-Plattformen? Wo werden diese angewendet?

3.2.1 Prinzipien von No-Code Plattformen

3.2.2 No-Code Plattformen in der Robotik

3.3 Visuelle Programmierung

Was ist Visuelle Programmierung? Wozu ist sie sinnvoll nutzbar? Was sind Schwächen und Herausforderungen bei der Entwicklung einer solchen Schnittstelle?

3.3.1 Eigenschaften von Visueller Programmierung

3.3.2 Block-Programmierung

3.4 Iterativer Entwicklungszyklus

4 Roboter-Ansteuerung

4.1 Sprachausgabe

4.2 Bewegungssteuerung

4.3 Spracherkennung

5 Benutzerzentrierte Auswertung

Das folgende Kapitel führt zunächst in die mögliche Bestimmung der Anforderungen der Nutzer:innen an eine graphisch-interaktive Entwicklungsumgebung ein. Daraufhin wird ein Konzept erläutert, wie eine solche Anwendung mit Hilfe von Nutzertests bewertet und verbessert werden kann. Zu Ende des Kapitels wird erläutert wie diese Konzepte umgesetzt und gezielt an die Zielgruppe verteilt wurden.

5.1 Gestaltung der Anforderungserhebung

Damit die entstehende Entwicklungsumgebung genau auf die Bedürfnisse der Nutzer:innen angepasst ist, müssen anfänglich die Anforderungen der Nutzer abgefragt werden. Dazu bietet sich, besonders durch die Möglichkeit der Einbindung von zusätzlichen Medien, eine gezielte Online-Befragung an [7]. Die Ergebnisse dieser Umfrage werden dann in einen Style-Guide für die Entwicklung der tatsächlichen Anwendung übersetzt.

Neben Fragen zu Gestaltungsentscheidungen, wie die Platzierung von zusätzlichen Informationen oder der Eingabeform von Parametern, ist es auch nützlich die Anwendung mit etablierten Programmierungsumgebungen im Bezug auf die Greifbarkeit und Interesse an der Nutzung zu vergleichen. Des Weiteren können noch Anforderungen der Nutzer:innen an die Anwendung abgefragt werden, um die Relevanz von verschiedenen Funktionen für die Nutzer:innen zu vergleichen und diese entsprechend bei der Entwicklung der Prototypen zu priorisieren. Abschließend ist es hilfreich relevante demographische Informationen der Teilnehmer:innen abzufragen.

Bei jeder der Fragen ist zu beachten, dass diese verständlich, knapp und neutral formuliert ist, sie nicht durch unnötige Verneinungen kompliziert werden, alle Teilnehmer:innen an der Umfrage die Frage beantworten können und ein Informationsgewinn durch die Beantwortung der Frage vorliegt [2]. Um technische Probleme zu vermeiden und die Umfrage möglichst barrierefrei auf verschiedensten Geräten und Browsern darstellen zu können, sollte eine Online-Umfrage-Applikation zu der Erstellung und Durchführung der Umfrage verwendet werden [7]. Die verschiedenen Abschnitte des Fragebogens sollten jeweils auf einer eigenen Seite angezeigt werden, die ebenso den Fortschritt der Umfrage anzeigt. Diese einzelnen Seiten sollten alle

nötigen Informationen beinhalten um die Fragen zu beantworten und ohne scrollen angezeigt werden können [2].

5.1.1 Vergleich mit bestehenden Anwendungen

Ein Vergleich mit bestehenden Benutzeroberflächen ist nötig, um zu erfahren, ob der gewählte Ansatz einen Vorteil gegenüber bereits existierenden Lösungen darstellt. Besonders interessant ist ein Vergleich der Offenheit der Teilnehmer:innen an der Nutzung der verschiedenen Anwendungen.

Um einen unbefangenen Vergleich zwischen den Ansätzen zu ermöglichen, wird erst eine Interaktions-Aufgabe eingeführt, die der Roboter ausführen soll. Im Folgenden werden sowohl die etablierte Software, als auch das erste Mock-Up der Entwicklungsumgebung vorgestellt und erklärt. Abschließend wird die Implementierung der anfänglich eingeführten Aufgabe in beiden Anwendungen abgebildet.

Nach der Einführung in die Anwendungen werden mehrere Aussagen zu der Verständlichkeit und Interesse an der Nutzung einer solchen Benutzeroberfläche aufgeführt. Die Teilnehmer:innen an der Umfrage sollen dann ihre Übereinstimmung mit diesen Aussagen auf jeweils einer 5-Punkte Likert-Skala für alle vorgestellten Nutzeroberflächen beantworten. Abgefragte Einstellungen können Verständlichkeit der Benutzeroberfläche, Einschätzung der eigenen Fähigkeit die jeweilige Anwendung zu nutzen und weiteren Fragen zu spezifischen Teilen der Oberflächen beinhalten.

Die aus diesen Fragen entstehenden Daten können zu einer Einschätzung führen, wie nützlich der gewählte Ansatz ist und ob eine weitere Verfolgung dieses Ansatzes hilfreich ist, oder ein andres Konzept ausgearbeitet werden muss.

5.1.2 Designentscheidungen

Im darauf folgenden Abschnitt des Fragebogens werden verschiedene Funktionen des Roboters vorgestellt um Designentscheidungen am graphischen Interface der Entwicklungsumgebung zu treffen. Dazu werden jeweils mehrere Mock-Ups von Blöcken in Form von Bildern aufgeführt, die die beschriebene Funktion des Roboters umsetzen sollen. Die Teilnehmer:innen werden dazu aufgefordert, das Mock-Up auszuwählen, das am einfachsten zu verstehen ist.

Die Auswahl der Optionen ist so zu treffen, dass die Ergebnisse jeweils Designentscheidungen repräsentieren. Beispielsweise können die Optionen die Nutzung von

Drop-Down-Menüs, Textfeldern oder anderweitiger Eingabe von Parametern unterscheiden. Die hier gesammelten Antworten werden in einen Style-Guide übersetzt, der die Vorlieben der Nutzer:innen repräsentiert.

5.1.3 Funktionelle Anforderungen

Eine weitere wichtige Komponente des Fragebogens ist die Abfrage der Anforderungen, die die Nutzer:innen an die Anwendung haben. Da diese inzwischen eine Einführung in die Entwicklungsumgebungen erhalten haben und in den Fragen zu den Designentscheidungen mit möglichen Mock-Ups der Kontrollelemente vertraut gemacht wurden, können sie hier einzelne Anforderungen einschätzen.

Dies wird ebenfalls in Form von Aussagen umgesetzt, die die Teilnehmer:innen mit einer 5-Punkte Likert-Skala beantworten. Die Aussagen sind jeweils als technische Anforderungen formuliert, die Nutzer:innen von solchen Anwendungen an diese stellen könnten. Beispielsweise kann abgefragt werden, in welcher Sprache die Steuerelemente angezeigt werden sollten, welche weiteren Funktionen in der Oberfläche integriert werden sollten und welche Anforderungen an die Ausführung der Anwendung gestellt werden.

Diese Ergebnisse werden dann teilweise, je nach Art der Aussage, in den Style-Guide oder eine Prioritäten-Liste der Anforderungen eingearbeitet, die die Wichtigkeit der Funktionen für die Prototypen und das endgültige Produkt angibt.

5.1.4 Sonstige Datenerhebung

In einem abschließenden Teil können noch Fragen gestellt werden, die Informationen über die Erfahrungen der Teilnehmer:innen mit Programmierung erheben. Ebenso können für die weiterlaufende Entwicklung relevante demographische Informationen abgefragt werden. Diese Abfrage kann in freier Form passieren.

Um Teilnehmer:innen für die darauf folgenden Nutzertests an den entstehenden Prototypen zu rekrutieren werden diese auf der letzten Seite der Umfrage dazu aufgefordert bei Interesse ihre E-Mail Adresse anzugeben. Da diese dann auch mit den Daten der vorherigen Seiten verknüpft sind, kann später analysiert werden, ob die Teilnehmenden an den Nutzertests repräsentativ für die Befragten stehen.

5.2 Instanziierung der Testsettings

Wie wurden die entstandenen Prototypen getestet? Welche Aufgaben sollten die Nutzer durchführen? Welche Involvierung hatte der Beobachter der Tests?

5.3 Operationalisierung

Welche Nutzer? Etc.

6 Evaluierung

Wie wurden die Ergebnisse dieser Arbeit ausgewertet?

6.1 Anforderungsanalyse

Konzepte zum Auswerten

6.2 Statistische Auswertung

Welche Ergebnisse haben die Anforderungsanalyse und Nutzertests ergeben? Wie kann man diese interpretieren?

6.3 Softwarequalität

Ist es deterministisch? Software-Bewertung

7 Zusammenfassung und Ausblick

Was hat die Auswertung ergeben? Was lässt sich aus den Ergebnissen der Arbeit schließen?

7.1 Zusammenfassung

7.2 Ausblick

War der Ansatz hilfreich? Welche Bedeutung haben die Ergebnisse für ähnliche Projekte? Welche weiteren Veränderungen sollten an diesem Projekt vorgenommen werden, um die Benutzbarkeit zu verbessern?

A Statistik

Statistische Auswertung der Nutzerbefragung

Pseudonymisierte statistische Auswertung der Nutzertests

Literatur

- [1] W. Harrison. „From the Editor: The Dangers of End-User Programming“. In: *IEEE Software* 21.4 (Juli 2004), S. 5–7. ISSN: 1937-4194. DOI: [10.1109/MS.2004.13](https://doi.org/10.1109/MS.2004.13).
- [2] L. Jacobsen J. und Meyer. *Praxisbuch Usability und UX*. Rheinwerk Verlag, 2019. ISBN: 9783836269551.
- [3] H. Lieberman et al. „End-user development: An emerging paradigm“. In: *End user development*. Springer, 2006, S. 1–8.
- [4] H. Lindgren und P. Winnberg. „A model for interaction design of personalised knowledge systems in the health domain“. In: *International Conference on Electronic Healthcare*. Springer. 2010, S. 235–242.
- [5] J. Rode et al. *An end-user development perspective on state-of-the-art web development tools*. Techn. Ber. Department of Computer Science, Virginia Polytechnic Institute & State ..., 2005.
- [6] M. Scaffidi C. und Shaw und B. Myers. „Estimating the numbers of end users and end user programmers“. In: *2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05)*. IEEE. 2005, S. 207–214.
- [7] R. Schnell, P. Hill und E. Esser. *Methoden der empirischen Sozialforschung*. De Gruyter Oldenbourg, 2018. ISBN: 9783110577327.
- [8] D. Wolber et al. *App Inventor*. "O'Reilly Media, Inc.", 2011.

Abbildungsverzeichnis

| | |
|--|---|
| 2.1 Platzhalter für Abbildungsverzeichnis. | 2 |
|--|---|

Eidesstattliche Versicherung

Ich versichere hiermit an Eides statt, dass ich die vorliegende Abschlussarbeit mit dem Titel „Graphisch-interaktive Entwicklungsumgebung für Robotersysteme zum Einsatz in den Rehabilitationswissenschaften“ selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Ort, Datum

Unterschrift

Belehrung

Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50 000,00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden (§ 63 Abs. 5 Hochschulgesetz –HG–).

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird ggf. elektronische Vergleichswerkzeuge (wie z. B. die Software „turnitin“) zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen.

Ort, Datum

Unterschrift