



Fakultät für Maschinenwesen
Lehrstuhl für Ergonomie

Gestaltung eines ergonomischen Interaktionskonzeptes für flexibel einsetzbare und transportable Roboterzellen

Uwe Herbst

Vollständiger Abdruck der von der Fakultät für Maschinenwesen
der Technischen Universität München
zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.) genehmigten Dissertation.

Vorsitzende(r):

Univ.-Prof. Dr.-Ing. Veit St. Senner

Prüfer der Dissertation:

1. Univ.-Prof. Dr. phil. Klaus Bengler

2. Univ.-Prof. Dr.-Ing. Gunther Reinhart

Die Dissertation wurde am 24.06.2015 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 03.11.2015 angenommen.

Danksagung

An dieser Stelle möchte ich mich bei allen Personen bedanken, die mich während der Erstellung dieser Arbeit mit Rat und Tat unterstützt haben.

Insbesondere gebührt mein Dank meinem Doktorvater Professor Klaus Bengler, der mir die Möglichkeit gegeben hat, als wissenschaftlicher Mitarbeiter am Lehrstuhl für Ergonomie der TU München zu promovieren. Diese Arbeit profitiert deutlich von den zahlreichen Anregungen und Ideen, die durch gemeinsame Diskussionen entstanden sind. Einen ähnlich wichtigen Einfluss stellt Professor Heiner Bubb dar, der mein Interesse für die Ergonomie erst geweckt hat und auch in seinem Ruhestand noch stets mit wertvollem Rat zur Verfügung stand. Ebenso möchte ich Professor Gunther Reinhart dafür danken, dass er sich als Zweitprüfer für diese Arbeit zur Verfügung gestellt hat.

Mein ganz besonderer Dank gilt überdies allen Mitarbeitern des Lehrstuhls für Ergonomie (LfE), die nicht nur in Bezug auf den wissenschaftlichen Diskurs und der Projektarbeit sondern auch im Sozialen meine Zeit am Lehrstuhl bereichert haben. Sie haben essenziellen Anteil daran, dass ich auch in Zukunft mit größter Freude an die Zeit am LfE zurückblicken werde. Besonders herausheben möchte ich dabei Dr. Herbert Rausch, Dino Bortot, Jurek Breuninger, Severina Popova, Julia Fridgen, Carsten Dlugosch, Martin Götze, Olaf Sabbah, Jonas Schmidtler und Thomas Weißgerber.

Neben den Mitarbeitern des Lehrstuhls möchte ich aber auch meinen Studenten danken, insbesondere Bernadette Egger, die mich bei der Evaluation der grafischen Benutzerschnittstelle unterstützt hat.

Zu guter Letzt gilt mein Dank natürlich meiner Familie, die mir diesen Weg überhaupt erst ermöglicht, mich dabei unterstützt und stets an mich geglaubt hat.

Abstract

Currently there is a strong demand for flexible and easy-to-program robots, as they offer important advantages for large scale manufacturer as well as small and medium sized enterprises (SME). The former benefit as flexible robot cells are especially suited for the demands of mass customization with its increasing numbers of variants and therefore sinking as well as more fluctuating unit numbers. For SMEs especially the implementing effort, a possibility to reuse the robot system, and the manageable variety of tasks are most important.

One of the main fundamentals for a robot system that meets these demands, concerning major enterprises and SMEs alike, is an ergonomic user interface. The user-centered approach offers the possibility to reduce the time as well as the expertise needed to accomplish the task of programming work processes. Consequently, an ergonomic human robot interface has positive effects on the implementing effort as well as the time needed to reconfigure a system for new tasks. The present work aims to develop a user-centered interface for a transportable robot cell designed for pick-and-place tasks based on the interaction paradigm *task-orientated programming*. Main goal is to enable the on-site personal of SMEs, who can't be regarded as expert users, to program a work task in a maximum of two working days. An important prerequisite to reach this goal is the renunciation from a programming process which is oriented on the functions delivered by the system towards a user orientated concept. This demands an analysis of the mental model of the user clientele regarding the designated manual work places. The results of this study, combined with ergonomic guidelines and a system-ergonomic analysis, are the basis for the design and evaluation of a conceptual graphical user interface. Moreover different technologies for the provision of geometric data are evaluated and the best device is combined with the graphical user interface.

Zusammenfassung

Aktuell ergibt sich ein großer Bedarf an flexiblen, und damit auch leicht zu programmierbaren Robotern. Für größere Unternehmen haben sie den Vorteil, dass sich eine kundenorientierte Produktion, mit einer steigenden Zahl an Varianten, mit einem flexiblen System sinnvoller und kosteneffizienter als mit bisherigen Automatisierungslösungen umsetzen lässt. Für kleine und mittlere Unternehmen stellen insbesondere die Einstiegshürde und die Aufgabenvielfalt wichtige Faktoren dar.

Entscheidende Ausgangskriterien zur Erreichung der gewünschten Flexibilität sind ein geringer Inbetriebnahmeaufwand, niedrige Umrüstzeiten (inklusive der erforderlichen Programmierung) sowie geringe Anforderungen an die Expertise der Mitarbeiter. All diese Aspekte lassen sich durch eine ergonomische Gestaltung der Interaktion der bisher reinen Expertensysteme positiv beeinflussen. Ziel dieser Arbeit ist somit die Gestaltung einer solchen ergonomischen Benutzerschnittstelle für Pick & Place-Arbeitsplätze, welche die benötigte Zeit und Expertise für die Inbetriebnahme bzw. Neu-Programmierung von bestehenden Arbeitsplätzen auf ein Minimum reduziert.

Einer der zentralen Schwerpunkte ist die Abkehr von einer technisch orientierten Prozesssegmentierung hin zu einer nutzerzentrierten Abbildung der Aufgabe. Um dieses Ziel zu erreichen, ist es erforderlich das mentale Modell der Nutzerklientel bezüglich bestehender Handarbeitsplätze zu untersuchen und ein sinnvolles Aufgabenmodell abzuleiten.

Auf Basis dieser Ergebnisse, sowie der bestehenden ergonomischen Richtlinien und einer systemergonomischen Analyse, wird eine grafische Benutzerschnittstelle (GUI) konzipiert und deren Umsetzung evaluiert. Weiterhin werden unterschiedliche Technologien und Konzepte zur Eingabe von Geometriedaten auf ihre Tauglichkeit untersucht und mit der konzeptionellen GUI zu einem Gesamtsystem verknüpft.

Inhaltsverzeichnis

1	EINLEITUNG	1
2	AUFGABENSTELLUNG UND VORGEHENSWEISE	4
3	SYSTEMERGONOMISCHE GRUNDLAGEN	7
3.1	Das Mensch-Maschine-System	7
3.2	Das mentale Modell des Menschen	10
3.3	Nutzerzentrierte Gestaltung der Mensch-Maschine-Interaktion	12
3.3.1	Allgemeine ergonomische Heuristiken	12
3.3.2	Nutzerzentrierter Gestaltungsprozess	13
3.3.3	Untersuchungsmethoden	15
3.3.4	Gestaltung von Touch-Interaktionen	17
4	GRUNDLAGEN UND STAND DER TECHNIK IN DER MENSCH-ROBOTER-INTERAKTION	21
4.1	Allgemeiner Aufbau von Industrierobotern	25
4.2	Verfügbare Interaktionstechnologien für die Eingabe von Koordinaten und Trajektorien	28
4.2.1	Tastensteuerung	29
4.2.2	Mechanische 6D-Eingabegeräte	30
4.2.3	Physische Mensch-Roboter-Interaktion	32
4.2.4	Tracking-Methoden und Gestenerkennung	33
4.2.5	Spracherkennung	35
4.3	Allgemeine Programmiermethoden	36
4.3.1	Offline-Methoden	38
4.3.2	Online-Methoden	42
4.3.3	Aufgabenorientierte Programmierverfahren	46
4.4	Diskussion	52
5	KONZEPTIONELLE UND TECHNISCHE RAHMENBEDINGUNGEN	53
6	UNTERSUCHUNG DES MENTALEN MODELLS DER NUTZERKLIENDEL	58
6.1	Versuchsdesign	59

6.1.1	Versuchsaufbau und Probandenkollektiv	61
6.1.2	Versuchsablauf	64
6.2	Ergebnisse	65
6.3	Diskussion	71
7	GESTALTUNG UND EVALUATION EINER GRAFISCHEN BENUTZEROBERFLÄCHE	72
7.1	Konzeption der grafischen Benutzerschnittstelle	73
7.2	Versuchsdesign	79
7.2.1	Versuchsaufbau	83
7.2.2	Versuchsablauf	85
7.2.3	Probandenkollektiv	85
7.3	Ergebnisse	86
7.4	Diskussion und Optimierungsbedarf	92
8	WAHL EINER GEEIGNETEN EINGABETECHNOLOGIE	93
8.1	Versuchsdesign	93
8.1.1	Versuchsaufbau	96
8.1.2	Versuchsablauf	98
8.1.3	Probandenkollektiv	99
8.2	Auswertung	99
8.2.1	Subjektive Daten	100
8.2.2	Objektive Daten	105
8.3	Ergebnisse	105
8.4	Diskussion	106
9	DISKUSSION DES GESAMTKONZEPTS	108
10	ZUSAMMENFASSUNG UND AUSBLICK	110
11	LITERATURVERZEICHNIS	112

Abkürzungsverzeichnis

ANOVA	<i>Analysis Of Variance</i> (Varianzanalyse)
BHG	Bedienhandgerät
CAD	<i>Computer-Aided Design</i> (Rechnergestütztes Konstruieren)
CAM	<i>Computer-Aided Manufacturing</i> (Rechnergestützte Fertigung)
FP	Flexible Programme
FZI	Forschungszentrum Informatik
GC	<i>Gravity Compensation</i>
GUI	<i>Graphical User Interface</i> (Grafische Benutzerschnittstelle)
LBR	Leichtbauroboter
KMU	Kleine und mittlere Unternehmen
MCI	Mensch-Computer-Interaktion
MMI	Mensch-Maschine-Interaktion
MMS	Mensch-Maschine-System
MRI	Mensch-Roboter-Interaktion
NC	<i>Numerical Control</i> (Numerische Steuerung)
PdV	Programmieren durch Vormachen
PSSUQ	<i>Post-Study System Usability Questionnaire</i>
TCP	<i>Tool Center Point</i>
viEMA	Vernetzte, informationsbasierte Einlern- und Ausführungsstrategien für autonome Montagearbeitsabläufe
VP	Versuchsperson

1 Einleitung

Der Einsatz von Automatisierungslösungen im Allgemeinen und Industrierobotern im Speziellen ist ein bedeutender Grundpfeiler für die Produktion in Großserien bzw. der Massenfertigung. Auf Grund hoher Wirtschaftlichkeit und Effizienz bei großen Stückzahlen leisten Automatisierungslösungen einen essenziellen Beitrag zur Steigerung der Produktivität und Reduzierung der Stückzahlkosten. Ein Industriezweig der ganz speziell vom Einsatz heutiger Automatisierungslösungen profitiert, ist dabei die Automobilbranche (Meyer, 2011).

In den vergangenen Jahren hat sich allerdings in Folge von starker Konkurrenz auf dem globalen Markt sowie einer höheren Nachfrage nach Produktvielfalt ein Trend ergeben, der Unternehmen zu einer Vergrößerung der Variantenvielfalt animiert hat. Diese Entwicklung hat neben anderen Auswirkungen insbesondere eine Erhöhung der Bauteilvarianten und damit der Kosten zur Folge. (Renner, 2007)

Eine Erhöhung der Bauteilvarianten führt bei gleichbleibender Zahl an gefertigten Produkten zu einer Verringerung der Einzelstückzahlen. Wird davon ausgegangen, dass konservative Automatisierungslösungen nicht grundsätzlich in der Lage sind, jegliche Produktvarianten zu fertigen, so ergeben sich erhöhte Kosten auf Grund des stärkeren Einflusses von Investitionskosten auf die reduzierte Zahl an Produkten.

Eine weitere Antwort auf den stark umkämpften globalen Markt ist Renner (2007) zu Folge die Verkürzung von Produktlebenszyklen. Diese hat einen ähnlichen Einfluss wie die Verringerung der Stückzahlen auf die Umlegung der Inbetriebnahme- bzw. Investitionskosten.

Großunternehmen, die bisher Fertigung in Massenproduktion durchgeführt haben, versuchen nun die neuen Herausforderungen durch Einsatz des Produktionskonzeptes *Mass Customization* zu begegnen. Hierbei wird versucht, die Vorteile der Massenproduktion mit den individuellen Wünschen der Kunden zu kombinieren und so die Kosten auf ein Maß zu senken, dass nur minimal über den Kosten der reinen Massenfertigung liegt. (Flavio S. Fogliatto, Giovanni J.C. da Silveira, & Denis Borenstein, 2012; Giovanni Da Silveira, Denis Borenstein, & Flávio S Fogliatto, 2001; Hart, 1995; Suresh Kotha, 1996)

Dabei stellt unter anderem Kotha (1996) die Bedeutung einer Steigerung der Flexibilität der Automation als Voraussetzung für den erfolgreichen Wandel von der Massenproduktion zur *Mass Customization* heraus.

Ein zentraler Aspekt dieser Flexibilität ist die Adaptivität des Systems für neue Aufgabeninhalten inklusive neuer Bauteilvarianten. Konventionelle Automatisierungslösungen sind aber üblicherweise sehr stark auf einen Anwendungsfall zugeschnitten und lassen sich so nur unter großem Aufwand für ein neues Produkt umrüsten bzw. umprogrammieren. (Schraft & Meyer, 2006)

Bezüglich der Flexibilität stellen Handarbeitsplätze aktuell das Maß der Dinge dar. Diese haben den Vorteil, dass sie sehr flexibel bezüglich unterschiedlicher Bauteilvarianten und robust gegenüber Prozessabweichungen sind. Allerdings ist der Einsatz für steigende Stückzahlen nur wenig wirtschaftlich und auch selten praktikabel. Darüber hinaus sind manuelle Arbeitsplätze kritisch für O-Fehler-Anforderungen und es muss besonders darauf geachtet werden, dass die Arbeit für die Mitarbeiter erträglich ist. Vor allem in Zusammenhang mit schweren Bauteilen oder gesundheitsschädlicher Verschmutzung (z. B. Lackieren) sind manuelle Arbeitsplätze aus ergonomischer Sicht kritisch.

Der aktuell fehlende Kompromiss zwischen hoher Flexibilität und geringen variablen Kosten ist dabei keineswegs ein reines Problem von Großunternehmen. Eine Lösung, die die Flexibilität und geringe Kosten vereinen könnte, wäre auch ideal für kleine und mittlere Unternehmen (KMU) (Awais, 2013). Diese profitieren bisher wenig von Automatisierungslösungen, da die Massenfertigung und die dafür entwickelten Systeme ihren Fokus auf Zykluszeiten und weniger, für KMUs bedeutsamer, auf eine einfache Programmierung und Anpassung der Roboteranwendung legen. Für diese ergibt sich aktuell zusätzlich eine große Hürde bei der Anschaffung und Integration von Automatisierungslösungen, da vor einem Betriebseinsatz hohe Investitionen, in Bezug auf Kosten aber auch Zeit und speziell geschultem Personal, getätigt werden müssen. (Schraft & Meyer, 2006)

Die Entwicklung der letzten Jahre zeigt, dass die Problematik der Investitionskosten zwar an Bedeutung verliert (Rückgang der Kosten von 25 % zwischen 1990 und 2004; IFR 2004) aber die Anschaffung eines Systems üblicherweise dennoch erst ab einem Potenzial von drei bis fünf Robotern erfolgt (Colestock, 2005). Neben den reinen Anschaffungskosten des Systems stellt die Verfügbarkeit von Systemexperten zur Programmierung und die Instandhaltung der Automatisierungslösung ein großes Hindernis (Schraft & Meyer 2006, Gottschalt 2001) und ein Risiko dar (Pires, Veiga, & Araújo, 2009). Hierfür muss entweder eine eigene Abteilung geschaffen oder auf externe Experten zurückgegriffen werden. Der erste Ansatz ist ein nicht zu

vernachlässigender Kostenpunkt und letzterer reduziert die Flexibilität und Einsatzgeschwindigkeit deutlich. Dementsprechend bleibt mit der Komplexität des Programmierprozesses eine der großen Hürden bei der Einführung von Automatisierungssystemen, wie eben industriellen Robotern, in kleine und mittelständische Unternehmen bestehen (übersetzt nach Pan, Polden, Larkin, van Duin, & Norrish (2010)).

Im Rahmen des Projekts viEMA¹, das vom Bundesministerium für Wirtschaft und Technologie im Rahmen des Programms *AUTONOMIK* gefördert wurde, soll nun in Zusammenarbeit mit der Robert Bosch GmbH, der ISRA Vision AG, der Faude Automatisierungstechnik GmbH und dem Forschungszentrum Informatik (FZI) eine Roboterzelle geschaffen werden, die eine Schnittstelle zwischen manuellen Arbeitsplätzen und konventioneller Automatisierung schafft.

Um das Problem starker Stückzahlschwankungen zu berücksichtigen, wurde die Zelle als transportables Konzept geplant, das bei Stückzahlhochlauf an einen bestehenden manuellen Arbeitsplatz angedockt wird. Bei Reduktion der Stückzahlen ist es dann möglich auf manuelle Arbeit umzustellen und einen anderen Arbeitsplatz zu unterstützen. Darüber hinaus würde sich eine solche Zelle optimal eignen, um den Schritt von der Musterproduktion, mit geringen Stückzahlen, hin zur Vollautomatisierung zu unterstützen und das Ende des Produktionslebenszyklus, mit sinkenden Stückzahlen, zu begleiten.

¹ www.viema.org

2 Aufgabenstellung und Vorgehensweise

Aus dem Blickwinkel einer industriellen Anwendung gibt es unterschiedliche Anforderungen an eine transportable Roboterzelle. Besonders für kleine und mittlere Unternehmen sind die erforderliche Expertise, insbesondere in der Implementierungsphase, sowie der benötigte zeitliche Bedarf zur Konfiguration und Programmierung entscheidend. Eine teils mehrere Wochen bzw. Monate dauernde Programmierungsphase eines neuen Arbeitsplatzes (Pan et al., 2010) ist im Falle der hier im Fokus stehenden Stückzahlspitzen von wenigen Wochen nicht zielführend.

Die Herausforderungen nach einer Verringerung der Implementierungszeit sowie der benötigten Expertise können als konkurrierend angesehen werden und lassen sich nur durch eine stark nutzerzentrierte Lösung begegnen. Dies bedeutet, dass die Mensch-Roboter-Interaktion (MRI) auf die Fähigkeiten des vorhandenen technischen Personals ausgelegt werden muss. Um dieses Ziel zu erreichen, müssen aber zuvor unterschiedliche Forschungsfragen beantwortet werden.

Wie in Kapitel 3.3 dargestellt, sind Consumer-Produkte in Bezug auf die Entwicklung von Vorgaben für die ergonomische Gestaltung von grafischen Benutzerschnittstellen (GUI) für Touch-Interaktionen den Produktivsystemen voraus. Diese Diskrepanz bei der Qualität der ergonomischen Vorgaben in der Mensch-Computer-Interaktion (MCI) verstärkt sich noch für die Mensch-Roboter-Interaktion, bei der die Gestaltung ergonomischer GUIs bisher nicht im Fokus stand und somit kaum Erkenntnisse vorhanden sind. Insofern ist es sinnvoll, sich für die Gestaltung der GUI in diesem Vorhaben an dem gut erforschten Gebiet der MCI zu orientieren und die relevanten Vorgaben im Zusammenhang mit der MRI zu untersuchen. Ein wesentlicher Aspekt dieser Arbeit ist somit die Untersuchung, welche speziellen Anforderungen die Mensch-Roboter-Interaktion mit sich bringt, die über die Verwendung der Heuristiken und Vorgaben aus dem Bereich der MCI hinausgehen. Als zentral kann hier unter anderem die Frage nach zusätzlichen Plausibilitätsschranken gesehen werden. Denn im Vergleich zur MCI besitzen Fehleingaben bei der Interaktion mit einem Roboter ein deutlich größeres Potenzial für Schäden an Umwelt und System.

Neben der Entwicklung von allgemeinen Gestaltungsvorgaben für die Entwicklung einer MRI muss eine geeignete Prozessprogrammierung entwickelt werden, die den Anforderungen der *aufgabenorientierten Programmierung* genügt. Hier soll, im Sinne

eines nutzerzentrierten Ansatzes, das mentale Modell der Nutzerklientel untersucht und eingebunden werden. Ein Vorgehen, das laut EU Robotics (2011) in dieser Form bisher nicht verfolgt wurde.

Da das Interaktionsparadigma und die Usability des Gesamtsystems ebenso von der verwendeten Eingabetechnologie abhängen, muss darüber hinaus untersucht werden, welches Konzept auf Basis von Usability- und User Experience - Aspekten als sinnvolle Lösung eingebunden werden kann. Wrede et al. (2013) beschreibt sehr anschaulich, dass von einer (physischen) Interaktion erwartet wird, dass sie einfach zu bedienen, komfortabel und nicht bedrohlich sowie selbst erklärend ist. Diesen Parameter gilt es für eine Auswahl sinnvoller Eingabetechnologien zu untersuchen. Für all diese Untersuchungen wird der Fokus auf die *aufgabenorientierte Programmierung* gelegt, zusätzlich wird aber auch die Tauglichkeit in einem Backup-Szenario, sprich ohne Sensorunterstützung, untersucht. Die Interaktionstechnologie zur Darstellung und Interaktion mit der grafischen Benutzerschnittstelle ist in dieser Arbeit nur untergeordnetes Thema.

Um eine Basis für die weiteren Ausführungen zu legen, werden in Kapitel 3 und 4 notwendige ergonomische und technische Grundlagen dargestellt. Kapitel 3 widmet sich dabei der Systemergonomie und stellt das Mensch-Maschine-System (MMS) in den Fokus, das einen Rahmen für den weiteren Theorieteil bildet. An Hand diesem werden die Bedeutung des Konstrukts *mentales Modell* sowie die Grundlagen der ergonomischen Gestaltung von grafischen Bedienoberflächen erläutert. Kapitel 4 widmet sich dem Stand der Forschung im Bereich der Mensch-Roboter-Interaktion und thematisiert Interaktionstechnologien sowie verfügbare Programmiermethoden.

Auf Basis dieser Grundlagen werden in Kapitel 5 die Rahmenbedingungen und Voraussetzungen für den Gestaltungsprozess dargestellt und definiert. Dies ist nötig, da die Gestaltung der MRI nur einen Teil der Konzeptionierung einer Roboterzelle beschreibt und sich wichtige Auswirkungen und Limitierungen durch das Zellenkonzept auf den Gestaltungsprozess ergeben.

Ein zentraler Baustein bei der Entwicklung eines nutzerorientierten Bedienkonzepts ist die Analyse des mentalen Modells der Nutzerklientel in Bezug auf die als Einsatzszenario gewählten Pick & Place-Arbeitsplätze. Da bisherige Programmierlösungen einen sehr funktionsorientierten Ansatz bei der Darstellung von Prozessabläufen verwenden, die aber aus Sicht eines nutzerzentrierten Konzepts ungünstig ist, wird in Kapitel 6 das mentale Modell der Nutzerklientel für ein

typisches Einsatzszenario untersucht. Basierend auf der Untersuchung wird dabei ein Funktionsmodell für die Implementierung der Prozessprogrammierung generiert. Dieses Modell stellt, zusammen mit den Vorgaben aus einer systemergonomischen Analyse sowie der ergonomischen Heuristiken (z. B. Normen und Praxiswissen), die Basis für das in Kapitel 7 dargestellte Benutzerschnittstellenkonzept dar. Neben einer Übersicht der einzelnen Programmoberflächen des Touch-Bedienkonzepts behandelt dieses Kapitel auch die Evaluation zur Absicherung der Nutzerfreundlichkeit und der Fehlerresistenz des Konzepts. Abgerundet wird das Kapitel mit einer Formulierung des abgeleiteten Optimierungsbedarfs.

Neben der Gestaltung und Evaluation der grafischen Benutzeroberfläche ist es außerdem erforderlich eine geeignete Wahl für die Eingabe von Geometriedaten zu treffen. Auf Basis der systemergonomischen Untersuchung sowie der durch das Gesamtkonzept definierten Randbedingungen wird dafür eine Auswahl valider Lösungen in Kapitel 8 dargestellt und evaluiert.

Kapitel 9 diskutiert das aus den Ergebnissen abgeleitete Gesamtsystem. Hierbei wird insbesondere auf die notwendige Einbringung von Plausibilitätsschranken und deren Umsetzung eingegangen.

In Kapitel 10 wird die Arbeit abschließend zusammengefasst und ein Ausblick auf weitere sinnvolle Anschlussuntersuchungen gegeben.

3 Systemergonomische Grundlagen

In folgendem Kapitel sollen die wichtigen systemergonomischen Grundlagen für die weitere Dissertation dargestellt werden. Im Fokus steht dabei vor allem das Konzept des Mensch-Maschine-Systems, das für die weiteren Betrachtungen dieser Arbeit einen geeigneten Rahmen darstellt und die wesentlichen Elemente *mentales Modell*, *ergonomische Gestaltungsgrundlagen* und das, in einem eigenen Kapitel dargestellte, Themengebiet *Robotik* in Zusammenhang bringt.

3.1 Das Mensch-Maschine-System

Das Mensch-Maschine-System (Abbildung 1) ist eines der zentralen Konzepte der Ergonomie. Es beschreibt das Einwirken des Menschen auf eine Maschine zur Erfüllung einer Aufgabe in Form eines Regelkreises. Dies geschieht unter Berücksichtigung des Informationsaustausches zwischen Mensch und Maschine und der induzierten Belastungen auf den Menschen. (Bubb & Schmidtke, 1993)

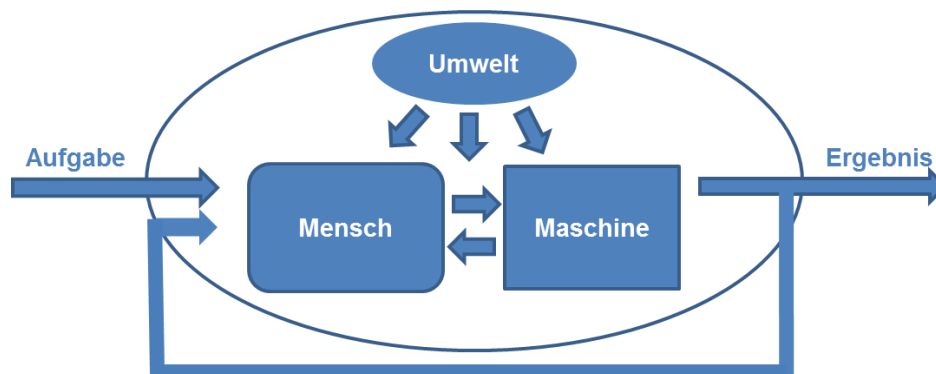


Abbildung 1: Das Mensch-Maschine-System nach Bubb & Schmidtke (1993)

Im MMS wird der Mensch hinsichtlich seiner Fähigkeiten bei der Informationsaufnahme, Informationsverarbeitung und Informationsumsetzung beschrieben.

Die Informationsaufnahme beschreibt dabei alle Prozesse die der Erkennung von Informationen aus der Umwelt, inklusive Rückmeldungen der Maschine sowie der Aufgabenerfüllung dienen. Diese Informationen lassen sich entsprechend der Sinnesrezeptoren unter anderem in visuelle, auditive, vestibuläre, olfaktorische, taktile und kinästhetische Modalitäten gliedern. Verknüpft die Rückmeldung des Systems (aber auch die Informationsumsetzung des Menschen) unterschiedliche

Wahrnehmungskanäle zur Informationsdarbietung wird dies als Multimodalität bezeichnet. Multimodale Systeme eignen sich im Besonderen für eine natürliche Interaktion zwischen Mensch und Computer und weisen, bei geeigneter Umsetzung, eine Reihe von Vorteilen für die Interaktion auf (Vilimek, 2007).

Im Prozess der Informationsverarbeitung werden die Informationen wahrgenommen und basierend darauf ein Abbild der Situation geschaffen. Dieses bildet dann die Grundlage für die Generierung und Auswahl von Handlungsoptionen. Das Vorhandensein eines geeigneten mentalen Modells ist für diesen Prozess förderlich. Die Umsetzung der gewählten Handlungsstrategie findet dann in der Informationsumsetzung statt. Diese kann im z. B. Rahmen der Mensch-Computer-Interaktion manuell oder erkennerbasiert stattfinden. Manuelle Interaktionskonzepte gliedern sich dabei in Texteingabe sowie direkte Manipulation (z. B. per Maus) während erkennerbasierte Systeme verbale (z. B. Sprache) und nicht-verbale (z. B. Gestik) Informationsumsetzungen beschreiben.

Auf Basis dieser menschlichen Informationskanäle findet die Interaktion mit der Maschine statt, die im Idealfall Rückmeldungen an den Benutzer tätigt, aber auch eine körperliche und mentale Belastung ausübt. Das Zusammenspiel von Mensch und Maschine kann mit Hilfe der systemergonomische Analyse (Bubb, 1993) untersucht werden. Diese resultiert aus den Entwicklungen des hier dargestellten MMS und greift die wichtigen ergonomischen Ergebnisse bezüglich der menschlichen Fähigkeiten und der Auswirkungen auf die Gestaltung einer ergonomischen Mensch-Maschine-Schnittstelle auf. Als Bewertungsgrundlage dienen die systemergonomischen Gestaltungsmaximen, die als ein Fundament der ergonomischen Gestaltung von Mensch-Maschine-Interaktionen gesehen werden können (dargestellt in Abbildung 2).

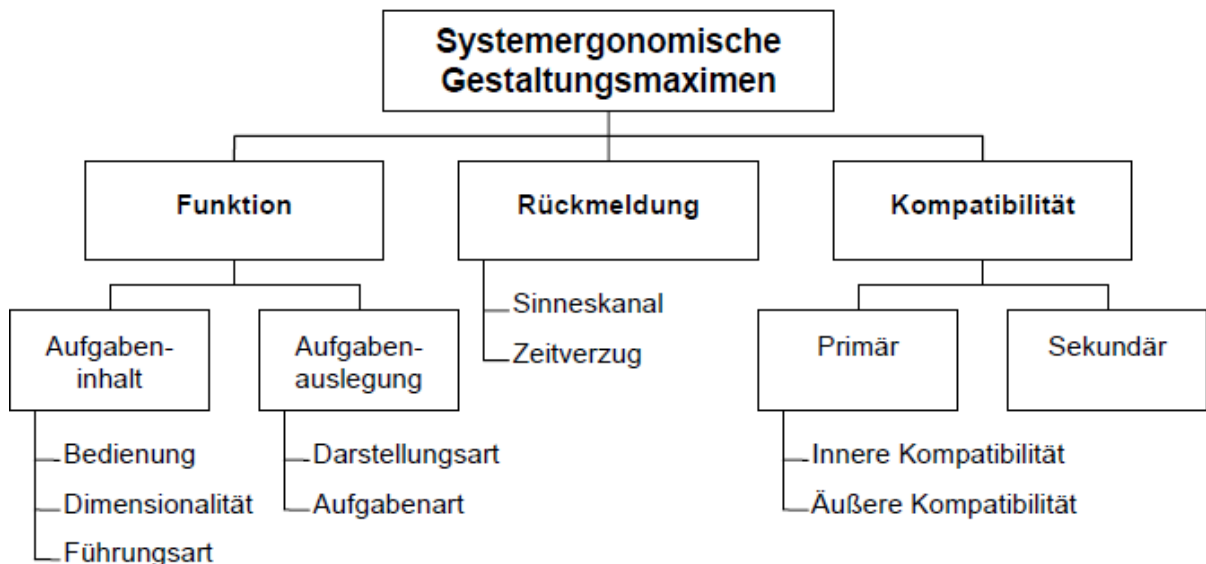


Abbildung 2: Übersicht über die ergonomischen Gestaltungsmaximen (Rassl 2004)

Auf Seiten der Maschine steht in vorliegender Arbeit der zu programmierende Roboter. Aus ergonomischer Sicht steht dabei vor allem das Interaktionsparadigma im Vordergrund, das die Schnittstelle des Systems zum Menschen hin prägt und somit Auswirkungen auf den Informationsfluss aber auch die Belastung des Menschen hat. Das Interaktionsparadigma, das durch das Programmierverfahren bestimmt wird, stellt wesentliche Anforderungen für die Konzeption bzw. Auswahl der graphischen Benutzerschnittstelle sowie des Eingabeelements (siehe Abbildung 3).

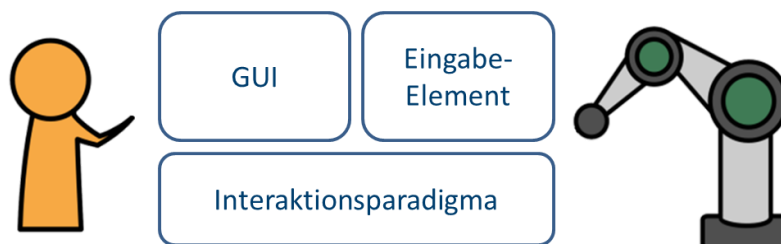


Abbildung 3: Interaktionsparadigma

Die Darstellung des Mensch-Maschine-System (Abbildung 1) bildet, wie bereits erwähnt, im Folgenden den Rahmen für die vertiefte Beschreibung des für diese Arbeit wichtigen Fundaments. Somit soll in folgenden Teilkapiteln das mentale Modell des Menschen, als bedeutendes Konstrukt der menschlichen Informationsverarbeitung, sowie die Grundlagen der nutzerzentrierten Gestaltung dargestellt werden, bevor in Kapitel 4 wesentliche Aspekte der Robotik erläutert werden.

3.2 Das mentale Modell des Menschen

Der Begriff des mentalen Modells entspringt der Kognitionspsychologie und stellte für diese einen Paradigmenwechsel vom Behaviorismus, und damit einem starken Schwerpunkt auf offenes und beobachtbares Verhalten, zu einer Betrachtung von inneren und nicht beobachtbaren Abläufen dar. (Dutke, 1994)

Laut Eberleh (1994) kann der Begriff mentales Modell am besten als „tätigkeitsleitende Vorstellungen und Kenntnisse“ bezeichnet werden, auf Basis derer, durch bewusstes Operieren, das Denken basiert. Mentale Modelle sind also interne Repräsentationen von Systemen, die für bestimmte Wissensdomänen gelten. Sie ermöglichen dem Menschen das Vorhersagen und Erklären physischer Zusammenhänge und werden durch Erfahrung und/oder Wissen gebildet. (Rouse & Morris, 1986; Stagers & Norcio, 1993)

Der Generierungsprozess eines mentalen Modells an sich erfolgt dabei üblicherweise in Analogien (Gentner & Gentner, 1983; Collins & Centner, 1987; Powers & Kiesler, 2006; Stagers & Norcio, 1993). Gentner & Gentner (1983) legen die erzeugende Wirkung von Analogien auf die Schaffung von Modellen in besonders anschaulicher Weise dar. In ihrem Versuch können sie nachweisen, dass es möglich ist, auf Basis von mentalen Modellen in Bezug auf das Verhalten von Wasser oder strömenden Menschenmengen auch das Verhalten von Elektrizität vorherzusagen beziehungsweise zu erklären. Die Resultate weichen dabei aber basierend auf der Fragestellung ab. Menschen, die Elektrizität vor allem mit der Analogie von fließendem Wasser erklären, neigen dazu besonders gut das Verhalten von parallel oder seriell geschalteten Batterien erklären zu können, während die Probanden mit dem Modell sich bewegender Objekte selbigen Sachverhalt besonders gut für Widerstände vorherzusagen konnten.

Damit die Verwendung einer Analogie zur Bildung eines neuen mentalen Modells möglich ist, muss die Analogie als passend erkannt werden. Notwendigkeit ist hier allerdings das Vorhandensein eines Schemas, das sowohl die Basis- als auch die Zieldomäne abdeckt. Ist dieses Schema nicht vorhanden, kann die Analogie nicht entdeckt werden (Gick & Holyoak 1983). Dutke (1994) erklärt diese Verhältnismäßigkeit wie folgt: „Da mentale Modelle auf Analogiebeziehungen aufbauen, müssen sie selbst auf der Grundlage schematischen Wissens konstruiert werden“.

Neben der Generierung des mentalen Modells durch Analogien kann dieses auch durch Interaktion erzeugt werden. Hierbei ist es wichtig, dass das Abbild des Systems, das der Nutzer wahrnimmt, verständlich und konsistent ist. Dies hat also Relevanz für die Qualität der Systemrückmeldung in Mensch-Maschine-Systemen.

Die erzeugten mentalen Modelle sind dennoch häufig fehlerhaft, inkonsistent und „idiosynkratisch“ (Norman, 1983), insbesondere wenn die Systemeigenschaften nicht direkt ersichtlich sind. Dabei lässt sich feststellen, dass Menschen sich der Unvollständigkeit ihrer mentalen Modelle durchaus bewusst sind und diesen in vielen Fällen Aberglaube anhaftet. Beide Eigenschaften führen dazu, dass viele Menschen zusätzliche Handhabungsschritte durchführen, um die eigene Sicherheit bei der Bedienung zu erhöhen und die mentale Belastung zu senken (Norman, 1983; Dutke, 1994).

Ein weiterer Aspekt der, bei der Betrachtung von mentalen Modellen, von Bedeutung ist, ist deren instabiles Wesen. Auf der einen Seite büßen mentale Modelle bei fehlender Nutzung ihre Konsistenz und Vollständigkeit ein. Auf der anderen Seite entwickelt sich das mentale Modell bei häufiger Nutzung aber auch konstant weiter zu einer exakteren Repräsentation des Zielsystems. Begrenzt wird diese Präzisierung durch die Limitierungen der menschlichen Informationsverarbeitung, aber auch durch den technischen Hintergrund vor allem in Bezug auf ähnliche Systeme (Norman, 1983).

Trotz der hier geschilderten Unzulänglichkeiten vieler mentaler Modelle, erfüllen sie aber in den meisten Fällen ihren Zweck und sind dennoch „eine unerlässliche und leistungsbestimmende Grundlage effektiver Arbeit“ (Eberleh, 1994).

Zur wissenschaftlichen Betrachtung dieser Thematik im Zuge der Systemergonomie muss laut Norman (1983) folgende Unterscheidung getroffen werden: das mentale Modell des Anwenders, das konzeptionelle Modell, das Abbild des Systems sowie die wissenschaftliche Konzeptualisierung des mentalen Modells. Letztere ist vor allem Ziel der Forschung, insbesondere der Psychologie, mit dem Ziel das mentale Modell abbilden zu können. Basierend auf diesen Forschungen ergeben sich dann unter anderem Rückschlüsse über die Fähigkeiten und Eigenschaften, die ja auch Grundlage dieses Kapitels sind. Das konzeptionelle Modell ist die Grundlage bei der Schaffung eines neuen technischen Systems und sollte eine angemessene, sprich genaue, konsistente und vollständige, Darstellung des zu entwickelnden Zielsystems liefern. Darüber hinaus muss das konzeptionelle Modell drei essenzielle

Eigenschaften mit sich bringen: Es muss erlernbar, funktional und anwendbar sein. Unter dieser Voraussetzung ist dann die Bildung eines konsistenten mentalen Modells des Anwenders möglich.

Die gezielte Unterstützung der Entwicklung eines konsistenten mentalen Modells kann dann, wie bei Dutke (1994) dargestellt, zu einer deutlichen Verringerung der Trainingszeit (in vorgestellter Studie um 28%) führen.

Gerade dieser starke Effekt auf die Trainingszeit bestätigt die Relevanz der Untersuchung des mentalen und der Entwicklung eines geeigneten konzeptionellen Modells für die vorliegende Arbeit. Da der Fokus dieser Arbeit auf einer Nutzerklientel liegt, die im Bereich der Roboterprogrammierung keine oder nur wenig Erfahrung besitzt, ist im Zweifel das mentale Modell in Bezug auf die Thematik Roboterprogrammierung nicht sehr präzise. Eine Unterstützung kann hier die Trainingszeit deutlich verringern und das Vertrauen bei der Bedienung erhöhen. Insofern kann die Thematik des mentalen Modells als wesentlich für die Entwicklung einer nutzerzentrierten Gestaltung gesehen werden.

3.3 Nutzerzentrierte Gestaltung der Mensch-Maschine-Interaktion

Basierend auf den Erkenntnissen zu den menschlichen Fähigkeiten, Informationen aufzunehmen, zu verarbeiten und umzusetzen, wurden im Forschungsfeld der Ergonomie unterschiedliche Grundlagen für die nutzerzentrierte Interaktionsgestaltung entwickelt. Dieses Teilkapitel stellt zuerst die allgemeine Herangehensweise an diese Thematik dar, bevor es auf die Vor- und Nachteile der Gestaltung von Touch-Konzepten im Speziellen eingeht. Diese haben auf Grund ihrer Eigenschaften in jüngerer Vergangenheit in beinahe allen Bereichen der Entwicklung von Interaktionskonzepten an Beliebtheit gewonnen und werden auch in dieser Arbeit in den Fokus gerückt.

3.3.1 Allgemeine ergonomische Heuristiken

Der wichtigste Begriff zur Bestimmung der Qualität der ergonomischen Umsetzung eines Interaktionskonzepts ist die „Gebrauchstauglichkeit“ bzw. „Usability“. Dieser beschreibt nach DIN EN ISO 9241 - 11 „das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele **effektiv**, **effizient** und **zufriedenstellend** zu erreichen.“ Neben der

Definition des Begriffs Gebrauchstauglichkeit liefert die Norm DIN EN ISO 9241-110 auch Grundsätze für die Gestaltung und Evaluation eines Softwaredialogs:

- Aufgabenangemessenheit
- Selbstbeschreibungsfähigkeit
- Erwartungskonformität
- Steuerbarkeit
- Fehlertoleranz
- Individualisierbarkeit

Erweitern lässt sich dieser Satz an Vorgaben noch durch Shneiderman & Plaisant (2010), die folgenden Postulate formulieren:

- Konsistenz beachten
- Universelle Anwendbarkeit anstreben
- Informative Rückmeldung bei der Gestaltung vorsehen
- Abgeschlossene Dialoge gestalten
- Nutzerfehler durch das Design reduzieren oder vermeiden
- Durchgeführte Aktion auf einfache Weise zurücksetzen lassen
- Die Kontrollüberzeugung des Nutzers unterstützen
- Die Belastung des Kurzzeitgedächtnisses reduzieren

Weitere ähnliche Heuristiken bzw. Gestaltungsempfehlungen für die Software-Ergonomie finden sich bei Nielsen (1994), dem Verband Deutscher Maschinen- und Anlagenbau e.V. (VDMA, 2004), dem Verein Deutscher Ingenieure (VDI 5005) oder auch in Form der bereits dargestellten systemergonomischen Gestaltungsmaximen (siehe Abbildung 2).

3.3.2 Nutzerzentrierter Gestaltungsprozess

Um viele dieser Gestaltungsrichtlinien erfüllen zu können, ist es bedeutsam die Nutzerklientel und den Nutzungskontext des zu entwickelnden Systems zu kennen. Die Richtlinie VDI/VDE 3850 Blatt 1 und Rudlof (2006) schlagen aus diesem Grund den folgenden Ablauf zur benutzergerechten Gestaltung vor:

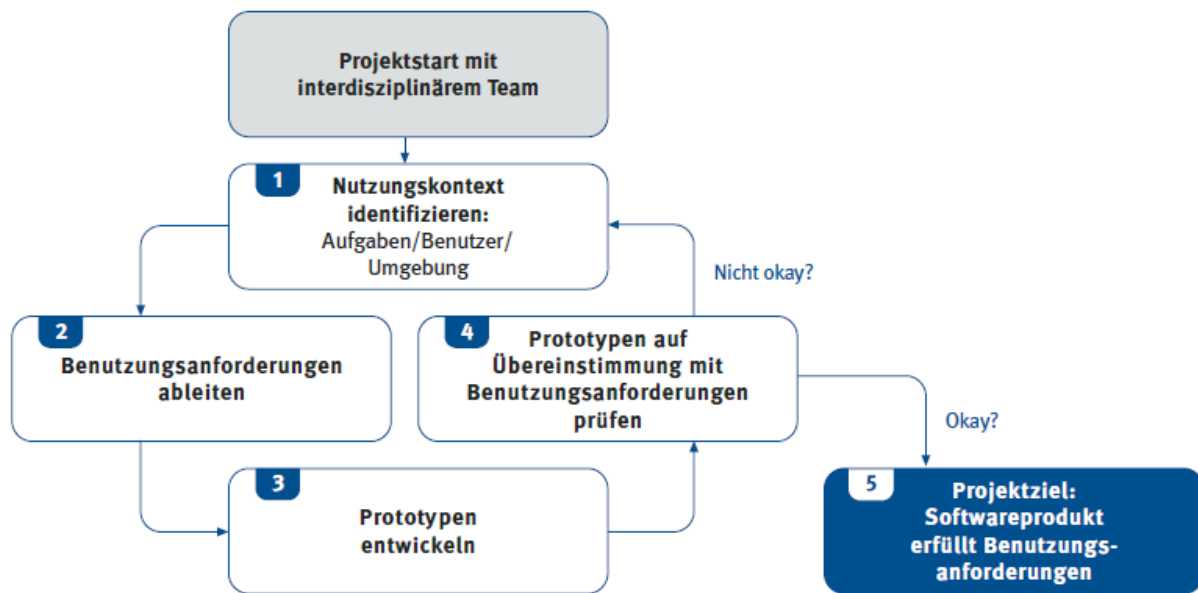


Abbildung 4: Benutzerzentrierter Entwicklungsprozess (Rudlof, 2006)

Dabei werden in einem ersten Schritt der Nutzungskontext, also die zu verrichtenden Aufgabe, die Nutzerklientel und die Umgebungsbedingungen identifiziert. Interessante Unterscheidungsmerkmale sind hier zum Beispiel die Anwendung des Softwareprodukts für den industriellen oder den Consumer-Bereich. Auch die Verwendung der Software auf einem mobilen oder stationären Endgerät hat Einfluss auf die Konzeptionierung. In Bezug auf den Nutzer sind insbesondere die zu erwartende Expertise bzw. Erfahrung, dessen Alter und eine mögliche kulturelle Prägung von Bedeutung.

Im nächsten Schritt werden aus dem Nutzungskontext die Anforderungen an die Benutzung abgeleitet und in einem weiteren Schritt in einem Prototypen umgesetzt, bei dem die ergonomischen Heuristiken aber auch die Implementierung eines konzeptionellen Modells (vgl. Kapitel 3.2) eine zentrale Rolle spielen. In einem letzten Schritt wird der Prototyp auf Übereinstimmung mit den Benutzungsanforderungen geprüft. Diese Evaluierung lässt sich sowohl mit Hilfe einer Probandenstudie als auch mit Hilfe der systemergonomischen Analyse (Bubb, 1993) durchführen. Sind die Anforderungen ausreichend erfüllt, ist das Projektziel erreicht. Üblicherweise ist dieser Vorgang aber iterativ, schließt also einen Verbesserungsprozess mit ein.

3.3.3 Untersuchungsmethoden

Für die erfolgreiche Umsetzung eines Bedienkonzeptes ist es unumgänglich, eine Evaluierung zur Sicherstellung der Gebrauchstauglichkeit durchzuführen. Dafür stehen unterschiedliche Evaluierungsmethoden zur Verfügung, die sich üblicherweise in Verfahren zur Erfassung und Bewertung von objektiven und subjektiven Daten untergliedern.

3.3.3.1 Objektive Bewertungsmethoden

Auf Grund des Fokus dieser Arbeit auf die Entwicklung einer schnell programmierbaren MRI mit geringen Expertiseanforderungen stehen als objektive Messwerte vor allem die benötigte Zeit, die Anzahl sowie Art von Bedienfehler und die Blickstrategien bei der Bearbeitung der Versuchsaufgaben im Vordergrund. Diese geben Aufschluss darüber, inwiefern das Interaktionskonzept den Anforderungen genügt bzw. verbessert werden muss. Die Erfassung der benötigten Zeit aber auch der Bedienfehler lässt sich durch direkte Beobachtung des Versuchs oder nachträgliche Begutachtung von Videoaufnahmen lösen. Die Erfassung der Blickdaten ist diffiziler und erfordert ein Blickerfassungssystem (Abbildung 5).

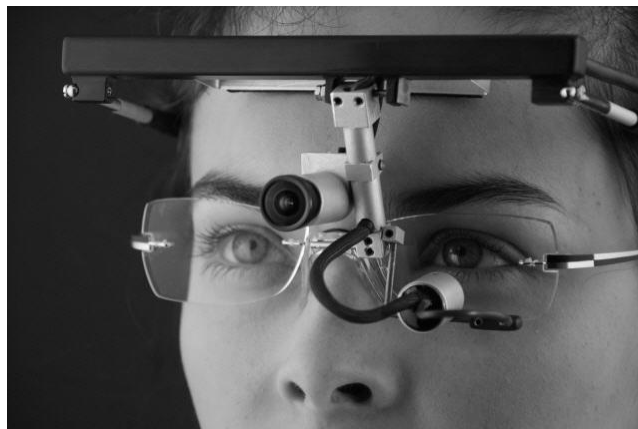


Abbildung 5: Kopfeinheit des DIKABLIS-Systems (Lange, 2010)

Das hier dargestellte DIKABLIS-System (Lange, Wohlfarter, & Bubb, 2009) erfasst das Sichtfeld des Nutzers bzw. Probanden über eine Szenenkamera. Zusätzlich wird mit einer weiteren Kamera die Bewegung der linken Pupille verfolgt und aufgenommen. Durch eine kalibrierte Überlagerung der beiden Videostreams ist es schließlich möglich, die aktuelle Blickposition des Probanden, dargestellt durch das grüne Kreuz in Abbildung 6, abzubilden.



Abbildung 6: Darstellung der überlagerten Blickdaten (grünes Kreuz markiert Blickpunkt)

Das System bietet so, mit geringen Einschränkungen der Sicht sowie der Bewegung, die Möglichkeit Blickpunkte und Blickfolgen während des Versuchs zu bestimmen. Damit lässt sich unter anderem überprüfen, welche Blickstrategien der Proband bei der Suche nach Bedienelementen wählt und dementsprechend die Anordnung eben dieser verbessern.

3.3.3.2 Subjektive Bewertungsmethoden

Für die Erhebung der subjektiven Bewertungen von Probanden werden häufig Fragebögen verwendet. Diese erlauben einen Einblick in die Empfindungen und Erwartungen der Nutzer bzw. Probanden, sind also für Evaluationen in Bezug auf Mensch-Maschine-Interaktion unerlässlich. Allerdings muss dabei berücksichtigt werden, dass Fragebögen stärker zu Verzerrungen neigen, als dies für objektive Verfahren der Fall ist, da sie von menschlichen Faktoren wie Erinnerungsfähigkeit, Aufmerksamkeit und Selbsterkenntnis abhängen. (Bortz & Döring, 2009)

Innerhalb dieser Arbeit kommen sowohl standardisierte als auch selbst gestaltete Fragebögen zur Anwendung. Der Unterschied liegt dabei in der bestätigten Zuverlässigkeit, Validität und Sensibilität der standardisierten Fragebögen, sprich der psychometrischen Qualifikation (Nunally 1978). Dementsprechend sind geeignete standardisierte Fragebögen eigens erstellten Fragebögen vorzuziehen.

Auf Grund seiner Bedeutung für die unterschiedlichen Untersuchungen soll der standardisierte *Post Study System Usability Questionnaire* (PSSUQ) näher

beschrieben werden. Dieser wurde entworfen um die Zufriedenheit der Nutzer mit Computersystemen und Anwendungen zu untersuchen. Der PSSUQ hat seinen Ursprung in einem Projekt der Firma IBM, das zum Ziel hatte, Methoden für die Messung von Leistung, Usability-Problemen sowie Nutzerzufriedenheit zu dokumentieren und zu validieren. In seiner ursprünglichen Version beinhaltete der Fragebogen 18 Teilfragen, die die Bewertung der Charakteristiken *schnelle Aufgabenerfüllung*, *Leichtigkeit des Lernens*, *hochwertige Dokumentation* sowie *Funktionsangemessenheit* erlauben. Nachdem in einer weiteren nicht publizierten Studie der Firma IBM auch die Charakteristik *schnelles Erzielen eines produktiven Einsatzes* als relevantes Merkmal für den Begriff Usability eruiert werden konnte, wurde der PSSUQ in Version 2 um Frage 8 erweitert. In einer weiteren Untersuchung konnten die Fragen 3, 5 und 13 als redundant identifiziert und somit gestrichen werden. (Lewis, 2002; Sauro & Lewis, 2012)

Somit umfasst die aktuelle Version 3 des PSSUQ 16 Fragen, die jeweils über eine siebenstufige Likert-Skala (Bortz et al., 2009) zu beantworten sind. Die Einbindung von Likert-Skalen hat positive Auswirkungen auf die Auswertung der Untersuchungen, da die Skalen als Intervallskalen angenommen werden können und somit zusätzliche statistischer Methoden (wie die Bestimmung des arithmetischen Mittels) angewendet werden dürfen. Die 16 Fragen des PSSUQ lassen die Beurteilung der drei Subklassen *Zweckmäßigkeit des Systems* (Fragen 1-6), *Informationsqualität* (7-12) und *Interface Qualität* (13-15) sowie eine Gesamtbewertung des Systems (1-16) zu. (Lewis, 2002; Sauro & Lewis, 2012)

Auf Grund der großen Auswirkungen der Fragebogengestaltung auf die Qualität der erhobenen Daten ist es sinnvoll, sich bei der Erstellung von Fragebögen an vorhandener Literatur zu orientieren. Hierfür können unter anderem Bühner (2011) und Moosbrugger & Kelava (2012) genannt werden.

3.3.4 Gestaltung von Touch-Interaktionen

Die Gestaltung von Touch-Interaktionen stellt einen Spezialfall der GUI-Entwicklung dar, der in vielerlei Hinsicht spezielle Anforderungen an das Bedienkonzept hat. Dennoch verspricht der Paradigmenwechsel zu Touch-Interaktionen viele Vorteile, die auch im Zusammenhang mit dieser Arbeit relevant sind. Touch-Bedienkonzepte sind leicht zu erlernen, erfordern keinen zusätzlichen Arbeitsraum, sind widerstandsfähig (Sears & Shneiderman, 1991) und werden von Nutzern oft als

„natürlich“ sowie „verlockend“ beschrieben (Forlines, Wigdor, Shen, & Balakrishnan, 2007). Ein weiterer wesentlicher Aspekt bei der Verwendung von Touch-Interaktionen ist der positive Effekt auf die Bildung von mentalen Modellen. Auf Grund der vielfältigen Darstellungsmöglichkeiten kann die Funktionsweise eines Elements, inklusive der Zusammenhänge mit anderen Bedienelementen und Bedingungen für die Nutzung, durch die Gestaltung kommuniziert werden. Ein sehr bekanntes Beispiel für die Verwendung einer Analogie zur Begünstigung der Bildung eines mentalen Modells ist die Verwendung eines typischen Büroarbeitsplatzes als Vorlage für konventionelle Betriebssystemdesktops.

Bei der Touch-Interaktion treten vor allem mobile Geräte bzw. Anwendungen in den Vordergrund, da sie eine nicht ortsgebundene Aufgabenerfüllung ermöglichen und die Verwendung von üblicherweise vertikalen stationären Touch-Oberflächen höhere Belastungen auf das Hand-Arm-System erzeugt (Ahlström, Lenman, & Marmolin, 1992; Sears, 1991).

Ein Problem bei den bisherigen industriellen Umsetzungen von Touch-Konzepten ist die oftmals direkte Konvertierung bisheriger Programmoberflächen bzw. Interaktionskonzepte. Diese erweisen sich als nicht zielführend, da das Konzept der Touch-Interaktion Eigenheiten besitzt, die den Übergang von z. B. auf Maussteuerung ausgelegten Desktopinteraktionen erschweren. Dementsprechend müssen vor allem folgende Aspekte berücksichtigt werden:

- Anpassung der Bedienelementpositionen
- Erhöhung der Bedienelementgrößen
- Kompensation des fehlenden haptischen Feedbacks

Der stärkste Unterschied zu bisherigen Konzepten mit Zeigegeräten, wie Mäusen, ist die starke Verdeckung von Teilen des Bildschirms bei der Eingabe mit dem Finger. Dementsprechend müssen Bedienelemente so angeordnet sein, dass eine Verdeckung wichtiger Informationen durch die Hand bei der Eingabe verringert oder verhindert wird. Da die Haltung bei der Nutzung von mobilen Geräten sehr variabel ist und die Händigkeit der Nutzer nicht vorhergesehen werden kann, sollten wichtige Bedienelemente vor allem am unteren Bildschirmbereich dargestellt werden. In Fällen, in denen der aktuelle Bildschirm für die Bedienung nicht relevant ist (z. B. bei dem Wechsel zwischen Funktionen), können die Bedienelemente am oberen Rand

dargestellt werden. Wird davon ausgegangen, dass der Anteil der Rechtshänder deutlich überwiegt, lässt sich auch eine Anbringung von wichtigen Funktionen an die rechte Bildschirmkante argumentieren. (Breuninger & Popova-Dlugosch, 2013)

Die Anordnung der Elemente wird darüber hinaus auch durch die zweite Anforderung an Touch-Konzepte beeinflusst, der Anpassung der Bedienelementgrößen. Sears & Shneiderman (1991) und Yang, Grossman, Irani, & Fitzmaurice (2011) beschreiben einen Nachteil der Touch-Interaktion bei geringen Bedienelementgrößen gegenüber Zeigegeräten bezüglich Geschwindigkeit und Genauigkeit. Für die Geschwindigkeit kehrt sich dieser Nachteil aber ab einer bestimmten Zielgröße um. Darüber hinaus sorgt eine ausreichende Dimensionierung auch für eine ähnliche bzw. geringere Fehlerrate bei den Touch-basierten Lösungen.

Ein weiterer Unterschied von Touch-Bedienungen zu konventionellen Lösungen ist das Fehlen einer haptischen, also durch den Tastsinn wahrgenommene, Rückmeldung bei Eingaben. Diese Art der Rückmeldung wird für kleine Geräte (insbesondere Smartphones) häufig durch ein Vibrieren des Gerätes simuliert sowie durch visuelle und/oder akustische Reize ersetzt. Das effektivste dieser Konzepte ist die Vibrationsrückmeldung, die durch den Einsatz der Akustik in der wahrgenommenen Perzeption noch verstärkt werden kann (Tikka & Laitinen, 2006). So kann das fehlende haptische Feedback in Form einer Vibrationsrückmeldung nachträglich implementiert werden, diese kann aber nicht die Qualität eines physischen Keyboards oder ähnlicher Interaktionstechnologien erreichen (Hoggan, Brewster, & Johnston, 2008). Für größere Vertreter der Touch-Interaktion, insbesondere im stationären Fall, ist die Implementierung einer haptischen Rückmeldung durch Vibration überdies oftmals nicht sinnvoll möglich. Dementsprechend muss die Gestaltung der Rückmeldung bei der Touch-Bedienung in besonderem Maße beachtet werden, damit der Nachteil einer fehlenden (natürlichen) haptischen Rückmeldung zumindest vermindert werden kann.

Leider findet sich die Entwicklung von Normen, Richtlinien und Vorgaben für die Gestaltung von Touch-Interaktionen im industriellen Umfeld in einem Stadium, in der sie mit der aktuellen Entwicklung aus dem Consumer-Bereich nur schwer mithalten kann. Hier prägen vor allem Unternehmen wie Apple und Google auf Basis ihrer weitverbreiteten mobilen Plattformen die gängigsten Gestaltungsrichtlinien. Sie schaffen Bedienungsumgebungen die eine hohe Akzeptanz und *Joy of Use*, übersetzt

etwa *Freude bei der Benutzung*, der Nutzer erzeugen. (Breuninger, 2014; Breuninger, Popova-Dlugosch, & Bengler, 2012)

Laut Breuninger et al. (2012) ist die Frage, ob die Designkonzepte und -Vorgaben des Consumer-Bereichs direkt auf den industriellen Bereich übertragen werden können, bisher nicht beantwortet. Insofern finden sich in Bezug auf industrielle Umgebungen bisher nur wenige Normen und Richtlinien, die zum Teil, wie im Falle der VDI/VDE 3850 – Blatt 3, auch nicht mehr den aktuellen Stand der Technik widerspiegeln. Eine positive Ausnahme bilden hier Breuninger & Popova-Dlugosch (2013).

4 Grundlagen und Stand der Technik in der Mensch-Roboter-Interaktion

Der Darstellung des MMS (Abbildung 1) folgend, fehlt nun noch die Betrachtung der im System eingebundenen Maschine, also im Kontext dieser Arbeit der Roboter. Der Begriff *Roboter* entstammt dem slawischen bzw. tschechischen Begriff *robota*, welcher üblicherweise mit den Begriffen *Arbeit*, *Zwangsarbeit* und *Fronddienst* übersetzt wird. Die eigentliche Prägung des Begriffs „robot“ geschah durch die Darstellung menschenähnlicher künstlicher Arbeiter im Theaterstück *Rossum's Universal Robots* von Karel Čapek aus dem Jahr 1920. (Goodrich & Schultz, 2007; Schaal, 2007)

Vor dieser Prägung wurde üblicherweise der Begriff *Automat* verwendet. Allerdings war die Schaffung eines neuen Begriffs sinnig, da ein Roboter eine einfache Automatisierungslösung üblicherweise in seinen Fähigkeiten übertrifft und eingesetzt wird, wenn entweder die Bauteilhandhabung oder die Werkstückbearbeitung dreier oder mehr Freiheitsgraden bedarf. (Colestock, 2005)

Laut Vogl (2009) lassen sich Roboter in die folgenden drei Klassen einteilen:

- Industrieroboter
- Serviceroboter
- Humanoide Roboter

Einer Abstraktion dieser Kategorisierung folgen auch Burgard, Christaller, & Knoll (2003), die eine Aufteilung der Robotik in stationäre und mobile Systeme, mit fließendem Übergang zur Servicerobotik, sowie humanoide Roboter sehen. Haun (2007) erweitert diese Klassifizierungen noch um *medizinische Roboter*, *Animatoren und Biorobotik* (Roboter, die das Verhalten von Tieren simulieren oder sogar natürlich-artifizielle Mischwesen darstellen) sowie *Roboter im Alltag und bei Spaß und Spiel*.

Die Klasse der **Industrieroboter** wird nach der Norm DIN EN ISO 8373 als „automatisch gesteuerter, frei programmierbarer Mehrzweckmanipulator [definiert], der in drei oder mehr Achsen programmierbar ist und zur Verwendung in der Automatisierungstechnik entweder an einem festen Ort oder beweglich angeordnet sein kann“. Die VDI-Richtlinie VDI 2860 zeigt die Klassifizierung des

Industrieroboters als Unterkategorie von Bewegungseinrichtungen und erlaubt die Abgrenzung zu Systemen wie zum Beispiel Manipulator oder Telemanipulation:

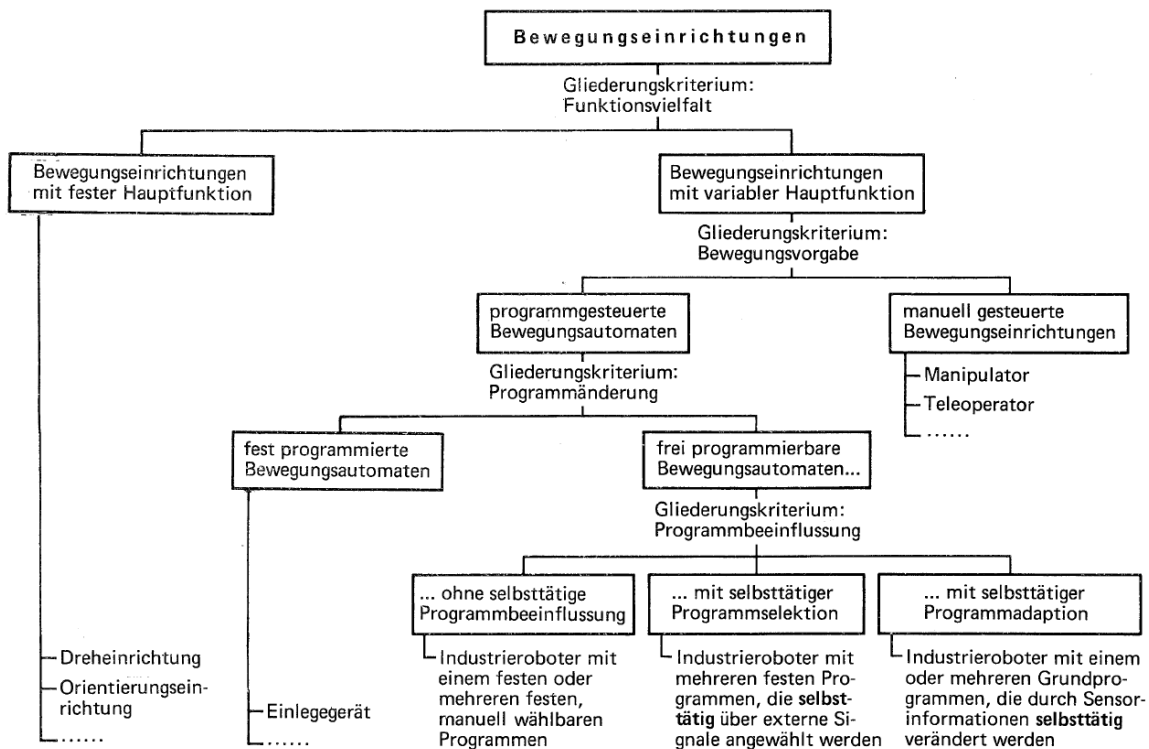


Abbildung 7: Klassifizierung von Bewegungseinrichtung nach VDI 2860

Der Industrieroboter stellt die älteste Roboter-Gattung dar und findet seinen Ursprung in der Verknüpfung der Technologien Telemanipulation und NC-Maschinen, die im ersten programmierbaren Manipulator mündete (Burgard et al., 2003). Auf Basis dieses Prinzips wurde 1960 der erste industrielle Roboter von Unimation, der *Unimate*, an General Motors für einfache Handhabungsaufgaben verkauft. (Burgard et al., 2003; Thrun, 2004)

In Folge wurden Industrieroboter auch für weitere Einsatzgebiete eingesetzt und mit zunehmendem Maße fester Bestandteil des Produktionsumfelds. Die Klasse der Industrieroboter steht im weiteren Verlauf der Arbeit im Vordergrund der Ausführungen, auch wenn sich Methoden und Ergebnisse in Teilen auch auf die anderen Klassen übertragen ließen.

Die Klasse der **Service-Roboter** beschreibt teil- und vollautomatische System die in ihrer Mehrzahl mobil sind und für den Menschen, die Gesellschaft oder auch Einrichtungen nützliche Tätigkeiten außerhalb des industriellen Umfeldes erledigen (DIN EN ISO 8373). Übliche Beispiele hierfür sind die Unterstützung im

Haushaltsbereich (siehe Abbildung 8) oder auch im Weltraum. Eine große Herausforderung bei der Entwicklung von Service-Robotern ist die Navigation in neuen oder auch dynamischen Umgebungen. Hierfür ist der Einsatz umfangreicher Sensorik mit einer ausgeklügelten Sensordatenfusion von Nöten (Burgard et al., 2003). Dies ist auch Grundlage für eine sichere Mensch-Roboter-Koexistenz, also die Anwesenheit von Mensch und Roboter in einer gemeinsamen Arbeitsumgebung ohne direkte Interaktion (Bortot, 2014).



Abbildung 8: Service Roboter "Rollin' Justin" (DLR) und die „DESIRE experimental“ Plattform (Kuehnle et al., 2009)

Humanoide Roboter stellen die am weitesten entwickelte Gattung der Roboter dar und werden vor allem durch die menschenähnliche Gestalt und Informationsverarbeitung definiert. Die Norm DIN EN ISO 8373 beschreibt einen solchen Vertreter als „Roboter, der über einen Körper, einen Kopf und Gliedmaßen verfügt und wie ein Mensch aussieht und sich auch so bewegt“. Auch wenn die Norm dabei die Entwicklung der kognitiven Fähigkeiten außen vor lässt, wird angestrebt menschliche kognitive Prozesse nachzubilden, um so eine bruchlose Kommunikation zwischen Mensch und Roboter, in Form einer Mensch-Roboter-Interaktion auf kognitiver Ebene (Siciliano & Khatib, 2008), zu erlauben und die Schaffungsmöglichkeiten des Roboters zu erweitern. Dieses Ziel, also die Emulation menschlicher sensorisch-kognitiver Fähigkeiten, kann bisher nicht als erreicht betrachtet werden. (Burgard et al., 2003)

Der Aufgabenbereich für humanoide Roboter kann nicht eindeutig von Service- und Industrierobotern getrennt werden. Viel eher bietet die menschenähnliche Form

Vorteile für einige Aufgabenbereiche in privaten aber auch industriellen Umfeld (Tanie, 2003). Bekannte Beispiele sind in Abbildung 9 dargestellt:



Abbildung 9: Humanoide Roboter „Kotaro“ (Mizuuchi et al., 2006), HRP-3 (Kaneko, Harada, Kanehiro, Miyamori, & Akachi, 2008) und ASIMO (Sakagami et al., 2002)

4.1 Allgemeiner Aufbau von Industrierobotern

Für den Einsatz im industriellen Umfeld haben sich unterschiedliche Bauformen etabliert. Siciliano & Khatib (2008) gliedern die Wichtigsten wie folgt:

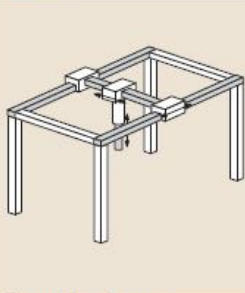
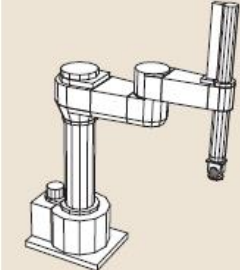
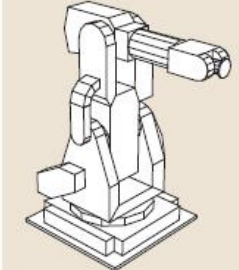
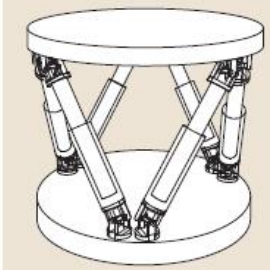
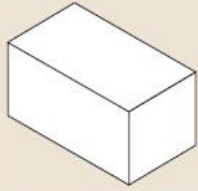
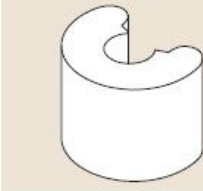
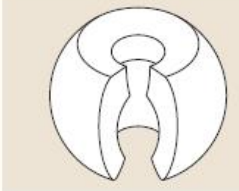




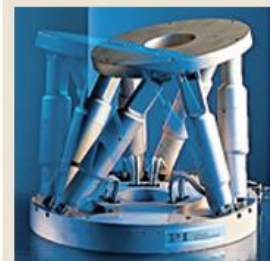
Gantry	SCARA	Serial articulated	Parallel
			
Form of workspace			
			
Product example			
			

Abbildung 10: Übersicht mechanische Strukturen (Siciliano und Khatib 2008)

Neben einer arttypischen Darstellung der einzelnen Systeme stellt Abbildung 10 sowohl die Form der Arbeitsräume als auch reale Beispiele dar.

Der **Portalroboter** (englisch *Gantry-Robot*) stellt die einfachste Form der gezeigten Konfigurationen dar. Dies betrifft nicht nur die mechanische Struktur, sondern auch die Gestaltung der inversen Kinematik. Da die Positionierung des Roboterwerkzeugs in kartesischen Koordinaten stattfindet, was ihm auch die zusätzliche Bezeichnung als kartesischer Roboter einbringt, ist die Steuerung der drei Achsen trivial. Auf Grund der sehr steifen Konstruktion sind außerdem sehr große Exemplare möglich, die insbesondere für den Transport von Lasten, zum Beispiel auch ganzer Fahrzeuge bis zu Flugzeugen, und dem vertikalen Fügen geeignet sind. Dennoch erlaubt die Konfiguration auch bei großen Systemen einen einfachen Zugang zum

Handhabungsort. Allerdings hat die Bauweise einen Nachteil bei notwendigen Änderungen an der Applikation bzw. des Wechsels beteiligter Maschinen, auch des Roboters selbst. Da alle Handhabungsorte innerhalb des vom Roboter aufgespannten Portals liegen müssen, werden Umbauten am System durch die Dimensionen des Roboters beschränkt. (Craig, 2005; Spong, Hutchinson, & Vidyasagar, 2004)

Der **SCARA** (*Selectively Compliant Assembly Robot Arm*) verfügt üblicherweise über vier Freiheitsgrade, bei der die ersten beiden Achsen nur rotatorisch ausgelegt sind und somit eine Bewegung des übrigen Roboterarms in der Planarebene erlauben. Die beiden übrigen Freiheitsgrade werden häufig durch ein einziges Gelenk realisiert. Nachdem die Motorik für die ersten beiden Gelenke in die Basis integriert werden kann, ist es möglich vergleichsweise große Aktuatoren zu verwenden. Diese erlauben dann sehr hohe Bewegungsgeschwindigkeiten des Werkzeugs. Darüber hinaus erlaubt das SCARA-System sehr hohe Präzision bezüglich Position und Trajektorienpfade sowie eine vibrationsfreie Bewegung. Folglich ist er insbesondere für einfache Pick & Place- und vertikale Fügetätigkeiten mit geringen Taktzeiten geeignet. (Craig, 2005; Nof, 1999)

Knickarmroboter verfügen über mindestens drei rotatorische Gelenke die in einer kinematischen Kette angeordnet sind (Bortot, 2014). Auf Grund der Ähnlichkeit zu menschlichen Bewegungen wird diese Klasse häufig auch als anthropomorpher Roboter oder Universalroboter bezeichnet. Grundsätzlich eignet sich dessen mechanische Struktur auf Grund ihrer hohen Flexibilität, der Beweglichkeit und der geringen Eindringung in das Arbeitsfeld insbesondere für räumlich begrenzte Arbeitsplätze, was ihn zu einer beliebten Wahl macht. Als Nachteil kann aufgeführt werden, dass seine Struktur ein Masseausgleich erfordert (Craig, 2005; Hesse & Malisa, 2009; Schilling, 1990) und bezüglich der inversen Kinematik als komplex zu bezeichnen ist.

Die Klasse der **Parallelroboter**, die typischerweise von der speziellen Bauform der Delta-Roboter vertreten wird, besteht aus zwei oder mehr Armen, die den Effektor mit der Basis verbinden. Wie in Abbildung 10 zu sehen, kann der Roboter in dieser Formation nur einen geringen Arbeitsraum abdecken. Allerdings ermöglicht sie ihm auch auf Grund der geringen bewegten Masse hohe Werte bezüglich Beschleunigung und Geschwindigkeit. Darüber hinaus kann auf Grund der guten Steifigkeit eine sehr hohe Genauigkeit erzielt werden. (Stock & Miller, 2003)

Die hier dargestellten Konfigurationen geben nur einen Überblick. Hesse und Malisa (2009) stellen noch weitere Klassifizierungen inklusive möglicher Varianten dar.

Die mechanische Struktur eines Roboters allein ist aber noch nicht in der Lage, industrielle Tätigkeiten durchzuführen. Hierfür sind noch das Roboter-Handgelenk und der Endeffektor erforderlich. Der Endeffektor stellt dabei das letzte Glied in der kinematischen Kette dar und entspricht dem für die Aufgabe benötigten Werkzeug, also beispielsweise Schweißgerät oder Greifer. (Spong et al., 2004)

Darüber hinaus bedarf es für fortgeschrittenere Systeme auch noch der Sensorik, sowie aktiver und passiver Peripherie. Die aktive Peripherie, zum Beispiel eine Spannvorrichtung, grenzt sich zur passiven Peripherie, z. B. ein Schaltschrank, durch seine mechanische Funktionalität ab. (Humburger, 1998)

Das Gesamtsystem all dieser Komponenten wird dann als Roboterzelle bezeichnet und ist die übliche Einsatzform im industriellen Umfeld (Dillmann & Huck, 1991).

Neben der mechanischen Konstruktion des Roboters ist auch die typische Steuerung eines Roboters von Interesse. Diese wird sehr anschaulich durch die Schematisierung von Rahimi & Karwowski (1992) (siehe Abbildung 11) dargestellt.

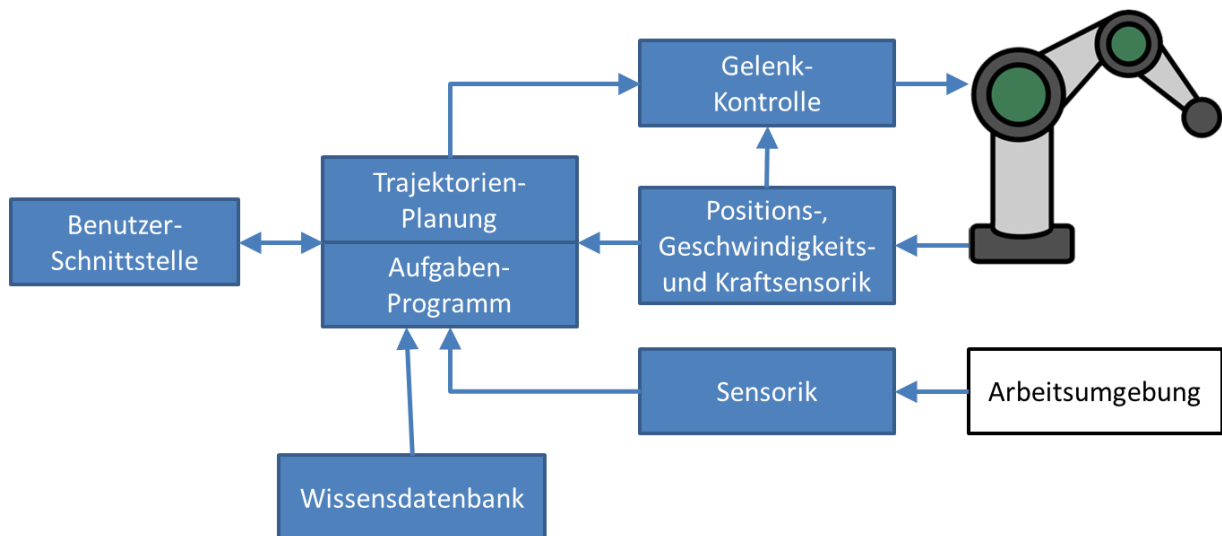


Abbildung 11: Elemente der Robotersteuerung (Darstellung nach Rahimi & Karwowski (1992))

Die Programmierung des Aufgabenprogramms eines definierten Prozess durch den Anwender bzw. Programmieren erfolgt über die *Benutzerschnittstelle*. Basierend auf dem erzeugten *Aufgaben-Programm* übergibt der *Trajektorien-Planer* Vorgaben an die Achsensteuerung bzw. die *Gelenkkontrolle*, die die Position der einzelnen

Roboterachsen steuert und über die Sensorik zur Bestimmung von Position, Geschwindigkeit und Kraft Rückmeldung über den Zustand des Systems erhält.

Über die einfache Programmierung von Trajektorien hinaus kann die Aufgabenplanung komplexe Entscheidungsprozesse für den weiteren Vorgang enthalten. Entscheidungen über die optimale Trajektorie können zum Beispiel durch die Bereitstellung von CAD-Daten erfolgen, die dem System erlauben die schnellste, kollisionsfrei Trajektorie zwischen zwei Punkten zu finden. In solch einem Fall ist die Erfassung der Arbeitsumgebung von Nöten und muss durch geeignete Sensorik bereitgestellt werden.

4.2 Verfügbare Interaktionstechnologien für die Eingabe von Koordinaten und Trajektorien

Die allgemeine Gestaltung der Mensch-Roboter-Interaktion lässt sich in die Konzeptionierung der hardwareseitigen Eingabetechnologie sowie der softwareseitigen grafischen Benutzerschnittstelle trennen.

Ein zentraler Fokus bei der Gestaltung oder Auswahl der Eingabetechnologie ist die Definition von Geometriedaten, wie Positionen des Greifers oder Trajektorien, im mehrdimensionalen Raum. Auf Grund der hohen Zahl an Freiheitsgraden stellt sie eine Herausforderung für den Menschen dar (Bubb, 1993). Mit Hilfe der Entwicklung von automatisierten Methoden zur Erkennung von Bauteilen und der automatisierten Bestimmung von validen Griffen zur Handhabung wird bereits versucht, die Häufigkeit von Eingaben sechsdimensionaler Geometrien zu reduzieren oder ganz zu vermeiden. Allerdings sind viele dieser Entwicklungen noch nicht vollständig ausgereift und somit im Industrieumfeld einsetzbar. Ein Hauptaspekt ist hier die fehlende Robustheit gegen Störungen und damit auch fehlende Prozesssicherheit. In Folge werden aktuell auch weiterhin Eingaben von Zwischen- und Zielposen des Roboters vom Nutzer durchgeführt. Insofern ist die Wahl der Eingabetechnologie für Geometriedaten von großer Bedeutung. Vor diesem Hintergrund sollen im folgenden Konzepte zur Eingabe von Geometriedaten, sowohl Koordinaten als auch Trajektorien, bzw. zum direkten Verfahren des Tool-Center-Point (TCP) dargestellt werden. Der Tool-Center-Point beschreibt dabei einen meist virtuellen günstig gewählten Koordinatenursprung des Werkzeugs und wird zur Bestimmung der Werkzeugposition benutzt.

In den weiteren Ausführungen werden Spezialfälle außen vor gelassen. Hierzu zählen zum Beispiel die Steuerung eines Roboters durch bioelektrische Signale, zu finden unter anderem bei Exoskeletten (Hayashi, Kawamoto, & Sankai, 2005; Yin, Fan, & Xu, 2012), oder auch die Brain-Computer-Interaction (Bell, Shenoy, Chalodhorn, & Rao, 2008). Die unterschiedlichen Eingabegeräte werden in dieser Arbeit wie folgt klassifiziert:

- Tastensteuerung
- Mechanische 6D-Eingabegeräte
- Physische Mensch-Roboter-Interaktion
- Tracking-Methoden und Gestenerkennung
- Spracherkennung

Diese sollen im Folgenden dargestellt und unter anderem auch auf ihre Eignung im Zusammenhang mit dem mentalen Modell des Menschen untersucht werden.

4.2.1 Tastensteuerung

Die konventionellen Interaktionstechnologien für die Programmierung von Geometriedaten sind textuelle Koordinateneingabe, insbesondere als diskrete Werteangaben innerhalb der Programmierung, sowie das inkrementelle Verfahren des Roboter-TCPs mit Hilfe von Folientasten.



Abbildung 12: Steuerung mit KUKA-LBR-Bedienhandgerät

Dabei treten diese Eingabekonzepte nicht zwingend getrennt voneinander auf sondern werden häufig, wie zum Beispiel beim Reis RV20-16², in einem Bedienhandgerät (BHG; siehe Abbildung 12) vereint, um die Nachteile der einzelnen Elemente abzuschwächen. Auf Seiten der textuellen Eingabe lässt sich dabei vor allem die schlechte Abschätzbarkeit von Koordinaten im Raum benennen. Hierbei liefert die Verwendung von Tasten, zum Verfahren des TCP in den translatorischen und rotatorischen Freiheitsgraden, beziehungsweise zur Anpassung der Roboter gelenkwinkel, einen wesentlichen Beitrag.

Für das Verfahren des TCP verfügen BHG üblicherweise über zwei vordefinierte Koordinatensysteme, in denen sich der Roboter verfahren lässt. Das erste dieser Koordinatensysteme ist ein festes kartesisches Weltkoordinatensystem, das zweite ein im TCP des Werkzeugs aufgehängtes Werkzeugkoordinatensystem. Weitere Koordinatensysteme lassen sich üblicherweise auf Wunsch konfigurieren. (Vogl, 2009)

Auch auf Grund der unterschiedlichen Koordinatensysteme des BHG ist die Zuordnung der jeweiligen Bewegungsrichtung, teilweise auch des richtigen Gelenks, nicht zwingend ersichtlich. Insbesondere für unerfahrene Nutzer bedeutet die Zuordnung der Eingabe- zur Bewegungsrichtung, sprich mentale Transformation, eine zusätzliche Belastung. Die Folge ist eine erhöhte wahrgenommene Schwierigkeit der Aufgabe und eine verringerte Leistung des Nutzers (DeJong, Colgate, & Peshkin, 2011). Dies lässt sich auch durch Anwendung der systemergonomischen Gestaltungsregeln erklären, die auf eine Verschlechterung der Usability bei einer höheren Anzahl an Stellteilen als Freiheitsgraden hinweisen. So wird die sechsdimensionale Aufgabe der Ausrichtung des TCP beim BHG durch sechs einzelne Buttonpaare realisiert.

4.2.2 Mechanische 6D-Eingabegeräte

Mechanische Eingabegeräte mit sechs Freiheitsgraden sind in der Lage, die vollständige Positionierung des Roboter-TCPs kombiniert in einer Bedienschnittstelle zu ermöglichen. Somit scheinen sie, insbesondere im Hinblick auf die genannten Nachteile der inkrementellen Tastensteuerung, gut für die Positionierung des TCP geeignet. Zwei Konzepte, die dieser Klassifizierung zugeordnet werden können, sind in Abbildung 13 dargestellt.

² <http://www.reisrobotics.de/de/produkte/roboter/vertikalknickarm/industriroboter-rv20-16>



Abbildung 13: 6D-Maus sowie der Phantom Omni³ der Firma Sensable Technologies

Der bekannteste Vertreter ist dabei die 6D-Maus, die auf Grund ihrer Bedeutung für diese Arbeit in den Vordergrund gestellt werden soll. Sie stellt einen interessanten Ansatz zur Steuerung des TCP dar, da sie nur geringe Einlernzeiten benötigt (Friedrich, Münch, Dillmann, Bocionek, & Sassin, 1996) und, wie erwähnt, im Gegensatz zur Steuerung mit Bedienhandgerät alle sechs Freiheitsgrade der Bewegung in einem Bedienteil kombiniert. Hierfür kann das Stellteil in jede Koordinatenrichtung bewegt bzw. um die Koordinatenachsen gekippt werden.

Wird davon ausgegangen, dass der Nutzer die Orientierung der Maus an die Bewegungsrichtungen des TCP anpassen kann, also entweder durch Kalibrierung oder durch Bewegung der Maus, haben die 6D-Mäuse den Vorteil, dass eine mentale Transformation der Eingabe zur Bewegungsrichtung nicht von Nöten ist. Ein Nachteil bei der Bedienung mit 6D-Maus ist die starke Fokussierung auf die Steuerung des TCP. Gelenkwinkel können mit dieser Interaktionsmethode nicht direkt angesteuert werden und eine Ansteuerung einzelner Gelenke wäre auch auf Grund der Tatsache, dass eine eindimensionale Aufgabe mit einer sechsdimensionalen Interaktion bewältigt wird, nach Bubb (1993) ungünstig. Die Reduzierung der Eingabe auf die Manipulation der TCP- Lage bedeutet, dass die Positionierung der einzelnen Robotersegmente nur noch indirekt erfolgt. Insofern muss darauf vertraut werden, dass der Roboter der TCP-Vorgabe in einer Weise folgt, die nicht zur Kollision des Arms mit der Umgebung führt oder es muss über Ausgleichsbewegungen eine günstige Anfahrt des Roboters an die Zielposition umgesetzt werden. Dementsprechend findet sich diese Steuerung häufig in

³ <http://www.dentsable.com/haptic-phantom-omni.htm>

Kombination mit einem konventionellen Bedienhandgerät, wie zum Beispiel im SmartPAD der Firma KUKA⁴, das die Manipulation der Gelenke, wie in Kapitel 4.2.1 dargestellt, über Buttons zulässt.

Als spezielle Form dieser Technologie stellt die Anbringung einer Maus direkt an den Greifer und damit in die Nähe des TCP dar. Hiermit kann eine sehr direkte Form der Steuerung umgesetzt werden, die vor allem aus Sicht des Nutzers den Vorteil hat, dass die Bewegungsrichtung des Systems in Bezug auf die Eingabe direkt ersichtlich ist. Fehleingaben auf Grund einer fehlerhaften mentalen Transformation sind hier beinahe komplett auszuschließen. Gleichzeitig bleiben auch für dieses System die Probleme des geringen Funktionsumfangs bei der Bewegungssteuerung des Roboters bestehen. Insofern kann die Nutzung einer reinen 6D-Maus-Steuerung vor allem für beengte Umgebungen ungünstig sein. Darüber hinaus erfordert die Steuerung des Roboters in dieser Konfiguration, dass der Nutzer sich sehr nahe am Roboter befindet. Daraus ergeben sich größere Herausforderungen für die Sicherheitsstruktur des Systems. Gleichzeitig erfordert dies auch ein größeres Systemvertrauen beim Nutzer.

4.2.3 Physische Mensch-Roboter-Interaktion

Die physische Mensch-Roboter-Interaktion (Haddadin, Albu-Schäffer & Hirzinger, 2011) bzw. das *kinesthetic teaching* (Akgun, Cakmak, Yoo, & Thomaz, 2012; Siciliano & Khatib, 2008) stellt Interaktionstechnologien dar, die einen direkten Kontakt des Benutzers mit dem Roboterarm oder dem Manipulator abdecken. Dabei steht die Verwendung eines Kraft-Momenten-Sensors am TCP und die *Zero Gravity* (Albu-Schäffer et al., 2007) bzw. *Gravity Compensation* (GC)-Steuerung (Bischoff et al., 2010; Zemiti, Morel, Ormaier, & Bonnet, 2007) im Fokus. Die genannten Interaktionsarten können auch als Kraftregelungstechnologien klassifiziert werden.

Beim Kraft-Momenten-Sensor induziert der Nutzer eine Kraft oder Moment auf den Roboter, der dann versucht der Bewegungsrichtung der Eingabe zu folgen. Da die Krafteinleitung in der Nähe des TCP erfolgt und die Zahl der Freiheitsgrade der Eingabe und der Bewegung übereinstimmen, lässt sich so eine direkte Manipulation des TCP erreichen. Als Besonderheit besitzt das Verfahren mit Kraft-Momenten-Sensor die Eigenschaft, dass bei der Programmierung einer Bahn auf einer Arbeitsoberfläche auch die Prozesskräfte mit erfasst werden können (Meyer, 2011).

⁴ <http://www.kuka-robotics.com/de/products/controllers/smartPAD/>

Dies kann für die Programmierung von Tätigkeiten, wie dem Schleifen von Oberflächen, ein nicht zu vernachlässigender Vorteil sein. Nachteilig an dieser Eingabetechnologie ist, wie schon bei der Verwendung einer 6D-Maus, die ausschließliche Manipulation der Greiferlage. Eine Anpassung der Gelenkpositionen ist nicht Weiteres möglich und erschwert die Verwendung in räumlich beschränkten Umgebungen stark.

Allerdings findet sich in aktuellen Leichtbaurobotern eine konzeptionell erweiterte Anwendung von Kraft-Momenten-Sensoren. Hierbei werden innerhalb eines jeden Gelenks Momenten-Sensoren integriert. Dadurch lässt sich eine Steuerung schaffen, die der durch die Gravitation einwirkenden Kraft in jedem Segment entgegenwirkt und auf von außen induzierte Momente reagiert (Hirzinger et al., 2002). Hiermit ist es möglich, durch menschliches Einwirken auf die unterschiedlichen Segmente des Roboters jede gewünschte Pose direkt anzufahren. Somit entfällt die Beschränkung des Kraft-Momenten-Sensors auf die reine Manipulation des TCP.

Eine Einschränkung beider Ansätze könnte in einer geringen Genauigkeit des Systems gesehen werden, da diese in starkem Maße von den motorischen Fähigkeiten des Menschen und der Latenz der Systems, von Krafteinwirkung zu Bewegung, abhängen.

4.2.4 Tracking-Methoden und Gestenerkennung

Tracking-Methoden oder auch die Gestensteuerung stellen indirekte Interaktionsformen für die Interaktion mit dem Roboter dar. Hierbei gibt es unterschiedliche Möglichkeiten die Bewegungen des Menschen zu verfolgen und in eine Bewegung des Roboters umzusetzen.

Bei der Gestensteuerung werden Bewegungen des Roboters mit einem definierten Satz an Gesten gesteuert. Dabei ist eine Steuerung des TCP üblich. Aus ergonomischer Sicht ist die Gestaltung nutzerfreundlicher Gestik schwierig. Bleibt der Umfang der Befehle auf wenige Gesten beschränkt, so lässt sich eine schnell erlernbare Steuerung bei limitiertem Funktionsumfang umsetzen. Steigt die Funktionszahl an, so erfordert die Bedienung komplexere Gesten, die einen Lernaufwand darstellen und je nach Abstraktion der Gesten auch zu Verwechslungen führen können. Dorau (2011) stellt dabei heraus, dass die Gesten einer Mensch-Maschine-Interaktion (MMI) auch sprachlichen Anforderungen genügen müssen, um dem System die Erkennung dieser eindeutig zu ermöglichen.

Insofern ist es nach Dorau (ebd.) auch notwendig konventionsbasierte Gesten zu verwenden die z. B. der Gebärdensprache deutlich näher sind als natürlichen Bewegungsmustern. Insgesamt scheint die Gestensteuerung, möglicherweise als Teil einer multimodalen MRI, sehr gut für den Einsatz im Bereich der Service Robotik geeignet zu sein (Hasanuzzaman et al., 2004; Nickel & Stiefelhagen, 2007; Waldherr, Romero, & Thrun, 2000). Der Einsatz im industriellen Umfeld insbesondere zur Steuerung des TCP muss auch auf Grund der Genauigkeitsanforderungen diskutiert werden.

Darüber hinaus zeigt die Verwendung eines optischen Trackings starke Unterschiede zwischen Labor- und Produktivumgebungen, da insbesondere die Kontrastwerte zwischen Hand und Umgebung im Labor deutlich besser reguliert werden können (Perzanowski, Adamse, & Schultz, 1998). Als Lösung für dieses Problem bieten sich markergestützte Trackingsysteme, wie z. B. Vicon⁵, an. Im Fall von Vicon wird infrarotes Licht von den Kameras emittiert und den Markern reflektiert. Damit lässt sich ein hochgenaues Tracking der Hand vornehmen, vorausgesetzt, dass Reflexionsquellen in der Umgebung ausgeblendet werden können. Nachteil solcher Systeme ist allerdings der hohe Anschaffungspreis.

Eine Alternative zum optischen Tracking der Hand ist die Verwendung von Datenhandschuhen. Diese bieten diverse Vorteile gegenüber einem optischen Tracking. So kann die Erfassung von Bewegungen über magnetische Tracker realisiert werden (Hendrich et al. 2010). Darüber hinaus kann auch Kraftmess-Sensorik eingebunden werden, die beim Einsatz des Systems mit künstlicher Intelligenz einen Vorteil für die Programmierung der Greifkraft darstellt und ein wichtiger technologischer Schritt für die Entwicklung autonomer Greifprozesse darstellt. (Ishii, Zhao, Inami, Igarashi, & Imai, 2009; Martin, Ambrose, Diftler, Platt, & Butzer, 2004)

Über das Tracking der Hand hinaus, stellen unterschiedliche Veröffentlichungen die Möglichkeit von direkten Eingaben der Trajektorie bzw. Koordinaten über die Verwendung von Zeigeinstrumenten, wie zum Beispiel Laserpointer (Kemp, Anderson, Nguyen, Trevor, & Xu, 2008) oder Stylus (Zaeh & Vogl, 2006), dar. Hiermit lassen sich besonders auf Oberflächen sehr schnelle Eingaben durchführen. Allerdings lassen sich speziell mit dem Laserpointer keine Trajektorien bzw.

⁵ <http://www.vicon.com/System/TSeries>

Koordinaten im 3D-Raum darstellen. Hierfür sind weiterentwickelte Zeigeeinstrumente erforderlich, deren Orientierung getrackt werden kann, um 6D Koordinaten vorgeben zu können. Beide Systeme, also Laserpointer oder ein physischer Pointer, sind abhängig von der Eingabegenauigkeit des Nutzers. Ein weiterer Nachteil fast aller hier dargestellten Tracking-Methoden ist überdies die fehlende haptische Rückmeldung.

4.2.5 Spracherkennung

Bei der Verwendung von Spracherkennung zur Steuerung des TCP wird die Stimme des Menschen über Sensorik erfasst und ausgewertet. Sie ermöglicht theoretisch eine sehr genaue Veränderung des TCP, wenn in Kauf genommen wird, dass der Nutzer diskrete Koordinaten nennt. Dies ist allerdings, wie schon bei der textuellen Eingabe von Koordinatenpunkten dargestellt, eine für den Nutzer wenig greifbare Methode. Anweisung an den Roboter den TCP in eine bestimmte Richtung zu verfahren sind umsetzbar, finden sich aber vor allem in Bereichen niedriger Bewegungsgenauigkeit.

Darüber hinaus weist die Spracherkennung Nachteile im Industrieumfeld in Bezug auf die Prozesssicherheit, aber auch in der bisher erreichten Natürlichkeit der Eingaben, auf (Landhäußer, Hey, & Tichy, 2014). Die Qualität der Spracherkennung ist zusätzlich abhängig von der Anzahl der Nutzer. Darüber hinaus sind im industriellen Umfeld häufig Lautstärken gegeben, die eine klare Erkennung der Spracheingaben des Nutzers schwierig machen. (Perzanowski et al., 1998)

Diese Einflussfaktoren können also die Genauigkeit der Erkennung reduzieren und somit die Prozesssicherheit senken. Fraglich ist in diesem Zusammenhang wie hoch die Wahrscheinlichkeit sowie die Schwere der Folgen eines falsch verstandenen Befehls oder das fälschliche Erkennen eines nichtvorhandenen Befehls ausfallen können. (Bierfreund, Nölle, & Hollmann, 2008)

Die Stärken der Spracherkennung lassen sich deshalb eher in Applikationen mit hoher künstlichen Intelligenz und folglich hohem Abstraktionsgrad ausspielen. Insbesondere im Bereich der Service-Robotik erlaubt die Spracherkennung so eine sehr natürliche Form der Interaktion.

4.3 Allgemeine Programmiermethoden

Unter Programmierverfahren oder –Methoden verstehen Spur & Uhlmann (2012) „das planmäßige Vorgehen zur Erzeugung von Anwenderprogrammen“. Weiterhin teilen sie die möglichen Programmierverfahren in direkte, indirekte und hybride Methoden. Direkte und indirekte Methoden werden auch als online bzw. offline-Prozesse bezeichnet (Pan, Polden, Larkin, van Duin, & Norrish, 2012). Offline-Verfahren erfolgen ohne Verwendung des Robotersystems, üblicherweise mit Hilfe einer Simulationssoftware. Das fertige Programm wird erst zur realen Testphase benötigt (Haun, 2007). Dementsprechend muss die Produktion für die Programmierung des Roboters nicht gestoppt werden und die benötigte Zeit für den Einrichtungsvorgang des Roboters sinkt drastisch (Roos & Behrens, 1997). Wittenberg (1995) beziffert die erzielbare Reduktion des zeitlichen Aufwandes auf 85%. Die geringere Zeitkritikalität erlaubt als weiteren Nebeneffekt mehr Spielraum bei der Optimierung des Programms in Bezug auf Produktionszeit sowie Material- und Energieverbrauch. Zusätzlich erlaubt eine Anbindung an CAD/CAM-Systeme bei Offline-Systemen die Überprüfung auf kollisionsfreie Trajektorien vor der eigentlichen Inbetriebnahme (Nof, 1999). Dabei muss beachtet werden, dass die Genauigkeit der Programmierung sehr stark von der Genauigkeit des Simulationssystems abhängt. Hierbei spielt auch eine Rolle, dass der Roboter sich nicht perfekt repetitiv verfahren kann. Mit Untersuchungen der Wiederholbarkeit und Genauigkeit des Robotersystems kann allerdings Abhilfe geschaffen werden (Roos & Behrens, 1997). Ist die notwendige Genauigkeit der Simulation nicht gegeben müssen im Anschluss an die Offline-Programmierung korrigierenden Parametrisierungsvorgänge getätigt werden, die den Vorteil der geringeren Produktionsunterbrechung einschränken. In vielen Systemen ist die Programmierung zudem in einer anspruchsvollen „prozeduralen“ Programmiersprache nötig (Dixon, 2004).

Online-Methoden beschreiben Programmierverfahren die unter Einbeziehung des Roboters erfolgen. Auf Grund der direkten Steuerung des Roboters können diese als natürlichere, und damit intuitivere, Methoden der Programmierung angesehen werden (Dixon, 2004). Die großen Vorteile der Online-Programmierung sind dabei vor allem die visuelle Perzeption und die niedrigeren Programmierkenntnisse, die üblicherweise von Nöten sind. Neben dem von Dixon (ebd.) angesprochenen Nachteil der Produktionsunterbrechung, die natürlich zusätzliche Kosten verursacht

und das Verfahren in manchen Bereichen nicht anwendbar macht, ist einer der größten Nachteile der Online-Programmierung die Gefährdung durch Falscheingaben. Die daraus resultierenden Fehlbewegungen des Roboters können sowohl den Nutzer als auch die Umgebung gefährden (Rahimi & Karwowski, 1992).

Neben den reinen Online- und Offline-Verfahren gibt es auch eine Reihe von hybriden Programmiermethoden, die Eigenschaften sowohl der Offline- als auch der Online-Programmierung beinhalten.

Eine weitere Unterscheidung die zwischen den Programmiermethoden getroffen werden kann, ist der Grad der Abstraktion (Haun, 2007; Vogl, 2009), der sich in der Literatur auch als *Automation* (Biggs & MacDonald, 2003) oder auch *Sophistication* (Nof, 1999) findet. Für drei Gliederungsstufen lässt sich der Begriff Abstraktion wie folgt beschreiben: Auf der untersten Ebene der Programmierverfahren erstellt der Nutzer exakte Vorgaben für einzelne also elementare Bewegungsabläufe des Roboters. Auf nächsthöherer Ebene umfasst die Programmierung die Steuerung von Bewegungssequenzen sowie der Einbindung der Sensorik. Elementare Bewegungen des Roboters müssen hier nicht mehr vorgegeben werden und die Flexibilität bezüglich Handhabungspositionen und Bauteillagen wird gestärkt. Auf höchster Ebene erfolgt die Programmierung vollständig implizit und nicht länger roboterzentriert. Dabei kann verallgemeinert werden, dass der Anteil zu programmierender expliziter Einzelprozesse mit einem Anstieg des Abstraktionsgrades reduziert wird, da nicht mehr „sämtliche Weginformationen durch den Programmierer eingegeben werden“ (Humburger, 1998) müssen.

Eine Matrix-Darstellung der einzelnen Unterscheidungskriterien und ausgewählter, weil sehr typischer, Programmierverfahren findet sich in Abbildung 14.

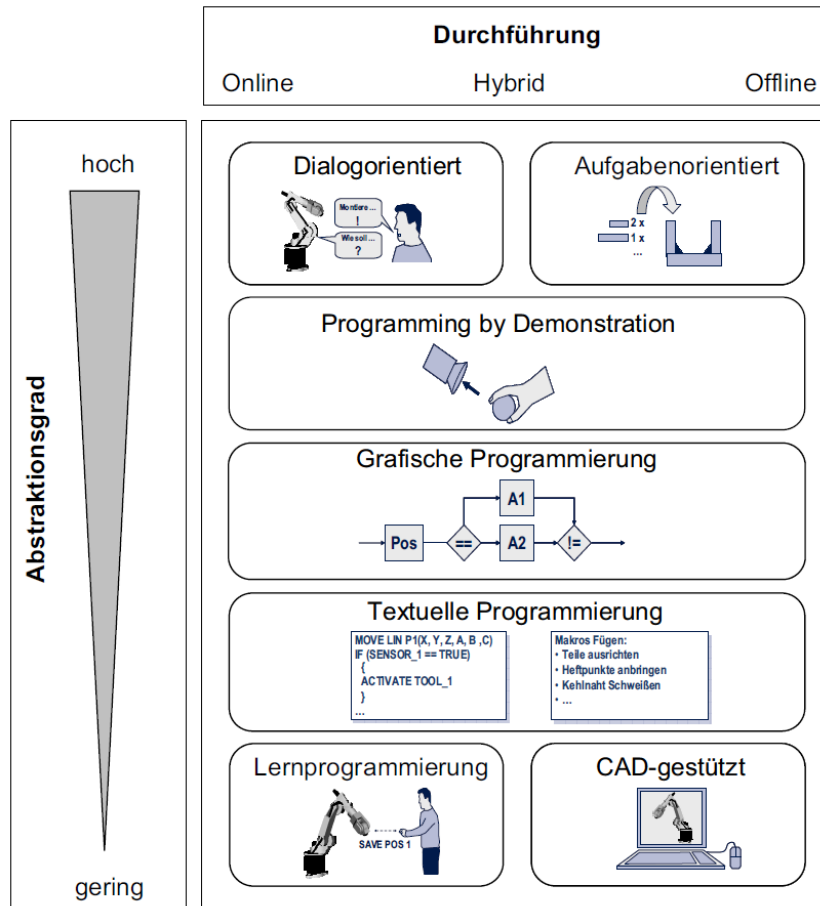


Abbildung 14: Matrix zur Kategorisierung von Programmierverfahren (Vogl 2009)

Zusätzlich zu den hier dargestellten Varianten gibt es noch weitere unterschiedliche Einteilungen der Programmierverfahren. Laut Siegert & Bocionek (1996) sowie Humburger (1998) existiert keine eindeutige und trennscharfe Zuordnung der Vielzahl von Programmiermethoden. Um dennoch eine homogene Gliederung zu schaffen, orientiert sich das weitere Kapitel an der Klassifizierung von Vogl (2009).

4.3.1 Offline-Methoden

In folgendem Kapitel sollen die gängigsten Kategorien zur Offline-Programmierung vorgestellt werden. Dazu zählen im Folgenden:

- Textuelle Programmierung
- Grafische Programmierung
- CAD-unterstützte Programmiersysteme

4.3.1.1 Textuelle Programmierung

Bei der textuellen Programmierung wird das Roboterprogramm unter Verwendung einer spezifischen Programmiersprache in seiner Gesamtheit in Klartext geschrieben. Laut einer Studie aus dem Jahre 1986 des National Bureau of Standards (USA) existieren 95 verschiedene Programmiersprachen, die zur Programmierung von Offline-Systemen verwendet werden können. Diese Roboterprogrammiersprachen lassen sich nach Biggs und MacDonald (2003) in drei unterschiedliche Klassen unterteilen.

Die **controllerspezifischen Programmiersprachen** sind die ursprünglichste Form zur Erstellung von Abläufen und üblicherweise sehr einfach gehalten. Sie verfügen über eine Syntax, die der Programmiersprache BASIC ähnelt, und einfache an den spezifischen Controller angelehnte Befehle zur Steuerung des Roboters sowie des Programmablaufs. Durch die starke Ausrichtung auf die verwendete Hardware sind die erzeugten Programme nicht übertragbar auf Robotersysteme anderer Hersteller. Dies sorgt üblicherweise dafür, dass Unternehmen, die unterschiedliche Robotersysteme im Einsatz haben, Schulungen für jedes der Systeme durchführen müssen.

Die **generischen Hochsprachen** unterscheiden sich zu den controllerspezifischen Sprachen vor allem durch breite Anwendbarkeit. Hierbei werden hoch entwickelte und allgemeinübliche Programmiersprachen, wie zum Beispiel C++, verwendet und um roboterspezifischen Code erweitert. Letzterer wird dabei im günstigsten Fall so erweitert, dass die Abstraktionsstufe über die Manipulation der einzelnen Gelenkpositionen hinausgeht und so zum Beispiel das Verfahren des Manipulators direkt möglich ist (Biggs & MacDonald, 2003). Allerdings ist auch hier, ähnlich wie bei den controllerspezifischen Sprachen, immer noch eine starke Verknüpfung mit dem dazugehörigen Robotersystem vorhanden. Vor allem im Bereich der Forschung wird versucht dieses Problem zu lösen und systemübergreifende Schnittstellen zu schaffen, die die Einbindung unterschiedlichster Systeme erlaubt. In diesem Fall müssen dann nur einzelne Klassen des, üblicherweise objekt-orientierten, Programms an die jeweilige Hardware angepasst werden (Kanayama & Wu, 2000; Vaughan, Gerkey, & Howard, 2003).

Im Gegensatz zu den bisher dargestellten Programmiersprachen verfolgen die **verhaltensbasierten Sprachen** nicht den Ansatz, eine prozedurale Anleitung für das System zu erstellen, sondern viel eher Vorgaben zu schaffen, wie der Roboter auf

bestimmte Bedingungen reagieren soll. Diese Art der textbasierten Programmierung findet sich bisher im Kontext der Industrierobotik nur in Forschungsarbeiten thematisiert. (Wenrui Dai & Kampker, 2000)

Im Bereich der mobilen sensorgeführten Roboterkonzepte finden sich allerdings Beispiele dieser Programmiermethode. Bekannte Vertreter sind dabei die Systeme *Yampa* (Hudak, Courtney, Nilsson, & Peterson, 2003) und *Frob* (Peterson, Hager, & Hudak, 1999; Peterson, Hager, & Serjentov, 2001).

Alle textuellen Programmiersprachen haben den Nachteil, dass der Nutzer über Experten-Wissen bezüglich Programmierung im Allgemeinen und der verwendeten Programmsyntax im Speziellen verfügen muss. Dies ist, wie bereits früher angedeutet, vor allem bei der Verwendung mehrerer unterschiedlicher kommerzieller Systeme eine bisher nicht zu vernachlässigende Hürde. Darüber hinaus erfordert die Verwendung einer rein textbasierten Offline-Programmierung ein hohes Maß an Vorstellungsvermögen und Sicherheit des zu programmierenden Prozesses.

4.3.1.2 Grafische Programmierung

Die grafischen Programmiermethoden stellen eine einfachere Möglichkeit dar, den programmierten Prozess auf die Realität zu projizieren. Hierbei wird das Roboterprogramm nicht durch Programmcode, sondern durch die Anordnung und Parametrisierungen von Graphen, Flow-Chart oder Diagramm-Ansichten dargestellt. Damit ergibt sich eine verständlichere Repräsentation der Algorithmen und der Programmstruktur (Cox & Smedley, 1998; Green & Petre, 1996). Neben der üblicherweise leichter verständlichen Darstellung des Systems, unter anderem weil der Mensch grafische dargestellte Zusammenhänge signifikant schneller aufnehmen kann als beim Lesen von Text (Raeder, 1985), wird durch Verwendung der grafischen Methoden auch der Bedarf an Syntaxwissen reduziert (Bischoff, Kazi, & Seyfarth, 2002; Myers, 1990). Da auch bei der grafischen Umsetzung die Anwenderfreundlichkeit sehr stark von der Umsetzung des Systems abhängt, wurde der Einfluss unterschiedlicher Darstellungsformen und der Bedarf nach Gestaltungsrichtlinien in diversen wissenschaftlichen Publikationen bzw. Projekten thematisiert (Bischoff et al., 2002; Burnett et al., 1995; Green & Petre, 1992; Green & Petre, 1996).

Eine der berühmtesten Umsetzungen dieser Programmiermethode findet sich in der Steuerungsoberfläche des *Mindstorms*-Roboters von Lego :

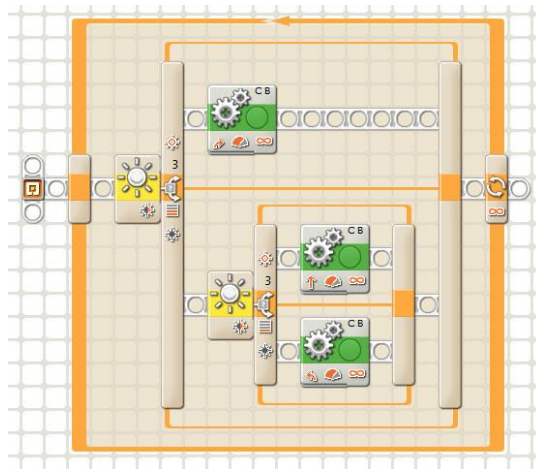


Abbildung 15: Lego Mindstorms NXT-G (Berns & Schmidt, 2010)

Hierbei zieht der Nutzer Elemente aus einem Menü auf die Programmfläche und verknüpft sie mit Hilfe von Ablaufträgern. Eine weitere Variante der grafischen Programmierung, die in ihrer Anwendung deutlich komplexer ist und mit dem Ziel entwickelt wurde, die Programmierung von Industrierobotern zu standardisieren, wird von Microsoft mit dem Microsoft Robotics Developer Studio bereitgestellt (Jackson, 2007).

Die Verwendung vorgegebener grafischer Elemente zur Programmierung ermöglicht bei geeigneter Gestaltung zwar ein, im Vergleich zur textuellen Programmierung, intuitiveres und leichteres Bedienen. Allerdings beschränken die Zahl und die Art der Parametrisierung der Elemente auch die Flexibilität und den Funktionsumfang (Berns & Schmidt, 2010).

4.3.1.3 CAD-unterstützte Programmiersysteme

Die CAD-unterstützten Programmierverfahren sind die logische Weiterführung der zunehmenden Einbindung von geometrischen Modellen in die Entwicklung von Bauteilen und Produkten. Nachdem die Arbeit mit CAD-Modellen sich zunehmender Beliebtheit erfreut hat, beschäftigten sich schon seit 1987 Forschungsarbeiten (Bhanu, 1987) mit der Frage, ob diese Technologie nicht auch der Programmierung von Robotern zu Gute kommen könnte (Neto, Mendes, Araújo, Pires, & Moreira, 2012).

Die Programmierung mit diesen Systemen erfolgt durch Festlegung von anzufahrenden Positionen und Verfahrenswegen. Die Verwendung von Simulationsumgebungen ermöglicht anschließend die Darstellung der Roboterbewegungen (Spur & Uhlmann, 2012). Am besten vergleichen lässt sich die Programmierung der meisten Systeme mit der virtuellen Umsetzung der Online-Methoden direktes und indirektes Teach-In (siehe Kapitel 4.3.2.1 und 4.3.2.2). So kann der Nutzer den Roboter wie beim indirekten Teach-In innerhalb der unterschiedlichen Koordinatensysteme verfahren oder durch direktes Manipulieren des virtuellen TCP Trajektorien aufzeichnen.

Auch der Bezug zu den textbasierten Programmierverfahren ist in einigen Systemen noch vorhanden. Diese lassen sich dann entweder in roboterspezifischen oder auch abstrahierten Sprachen steuern (Vogl, 2009).

Dabei ist es nicht zwingend notwendig, die Programme komplett manuell zu generieren. So zeigt Schäfer (2007) einen Ansatz zur Konvertierung von NC-Programmen, der als Ausgangsbasis für die automatisierte Bearbeitung von Bauteilen dienen kann. Fortschritte im Bereich der Ein- und Wiedergabe sind vor allem bei der Integration von *Virtual Reality* Konzepten zu finden.

Das bedeutendste Manko bei der Verwendung von CAD-unterstützten Tools ist erneut die Genauigkeit der Kalibrierung von Roboter und Umwelt zur Simulationsumgebung. Um eine vollständige virtuelle Umgebung zu schaffen, müssen die Prozessumgebung, der Roboter und die zu handhabenden Bauteile vollständig digital erfasst werden. Bei Abweichungen der Umgebungs- und Bauteilmodelle können fehlerhafte Trajektorien Schäden am Roboter aber auch der Arbeitsumgebung nach sich ziehen (Berns & Schmidt, 2010).

4.3.2 Online-Methoden

Das Paradigma der Online-Programmierung lässt sich in die folgenden Verfahren einteilen:

- Indirekte Teach-In Programmierung
- Direkte Teach-In Programmierung
- Sensorgeführte Programmierung

4.3.2.1 Indirekte Teach-In Programmierung

Indirektes Teach-In, oder auch *Pendant-Teaching*, bezeichnet die Programmierung von Trajektorien auf Basis der Eingabe mit einem Bedienhandgerät (siehe Kapitel 4.2.1). Dabei werden jeweils nur die End-Punkte von Bewegungsabschnitten, auf Anweisung des Programmierers, gespeichert und die eigentliche Trajektorie bei der Ausführung über Interpolationsalgorithmen erzeugt. Hierfür stehen im Allgemeinen zwei Methoden zur Erreichung des Zielpunktes zur Verfügung. Bei der *Point-to-Point*-Bewegung stellt der Roboter ohne Berücksichtigung der eigentlichen Effektor-Bewegung die einzelnen Gelenke auf ihre Zielwerte ein. Bei der *Controlled Path*-Steuerung werden alle Gelenke derart angesteuert, dass der Effektor in einer nachvollziehbaren Trajektorie (z. B. einer Geraden) zum nächsten Trajektorienstützpunkt verfährt (Nof, 1999).

Die Bewegungsgeschwindigkeit des Roboters und die maximale Antriebslast der einzelnen Motoren, die während dieser Programmiermethode aktiviert sind, ist aus Sicherheitsgründen stark reduziert, um den Programmierer, der sich in unmittelbarer Nähe zum System befindet, nicht zu gefährden. Die *indirekte Teach-In* Programmierung wird üblicherweise als eigenständige Programmiermethode verwendet, kann aber auch Teil eines höher entwickelten Programmierverfahrens sein (Rahimi & Karwowski, 1992). Als *Stand-Alone*-Variante verfügt sie über eine geringe Zahl an Zusatzfunktionen, die es dem Nutzer unter anderem erlauben, einfache Strukturierungen des Programmablaufs, wie zum Beispiel Verzweigungen oder zyklische Bewegungsfolgen, vorzunehmen. (Haun, 2007)

Die *indirekte Teach-In* Programmierung ist eine sehr einfache Methode, die keinerlei weiterer Komponenten bedarf. Sie eignet sich gut für das Erstellen einfacher Trajektorien, erfordert aber nach Vogl (2009) „einen geübten Benutzer und ein hohes Maß an Umsicht bei der Programmierung“.

Bei komplexen Prozessen mit hoher Zahl an Teilbewegungen ist diese Methode sehr zeitaufwendig und kaum änderungsfreundlich. Neben dem Zeitaufwand für die Programmierung an sich, müssen im Anschluss an die Erstellung des Programms intensive Testläufe durchgeführt werden, um ein ausreichendes Maß an Sicherheit und Zuverlässigkeit gewährleisten zu können (Pan et al., 2010). Hinzu kommt, dass die Programmierung nur dann zufriedenstellend durchgeführt werden kann, wenn die Trajektorien noch mit Augenkontrolle ausreichend genau festlegbar sind (Haun, 2007). Die Koordination der Roboterbewegungen zusammen mit der gleichzeitigen

Kollisionsüberprüfung, vor allem bei komplexen Bauteilen oder unter starken Bewegungseinschränkungen des Roboters, macht dieses Verfahren üblicherweise sehr mühsam. Unabhängig von den genannten Nachteilen ist diese Programmiermethode in der Praxis am weitesten verbreitet. (Haun, 2007; Pan et al., 2010; Vogl, 2009)

4.3.2.2 Direkte Teach-In Programmierung

Playback oder *direktes Teach-In* unterscheidet sich vom *Pendant-Teaching* vor allem dadurch, dass hier kein BHG von Nöten ist. Der Roboter kann durch direktes Führen der TCP-Position mit Hilfe eines Kraft-Momenten-Sensors oder aber durch direktes Bewegen der einzelnen Arm-Segmente über eine Dreh-Momenten-Steuerung (siehe 4.2.3) an die gewünschte Position verfahren werden. Dabei speichert das System die vom Nutzer abgefahrne Trajektorie nach vorgegebener zeitlicher Taktung oder äquidistant auf Basis des mit dem TCP durchfahrenen drei-dimensionalen Raums. Dabei werden, im Gegensatz zum *indirekten Teach-In*, nicht nur wenige Stützpunkte einprogrammiert, was vor allem die Programmierung von fest definierten Trajektorien begünstigt (Lozano-Perez, 1983). Auf Grund der nutzerfreundlichen Eingabe von Trajektorien stellt die *direkte Teach-In*-Programmierung niedrige Anforderungen an die Programmierkenntnisse des Nutzers. Dieser Vorteil wird noch dadurch verstärkt, dass der Arbeitsraum nicht vermessen werden muss und auch alle Störgrößen wie die konstruktive Ungenauigkeit des Roboters direkt berücksichtigt werden. Andererseits ist es nötig, dass der Nutzer erfahren bezüglich der Arbeitstätigkeit ist und diese mit hoher Präzision ausführen kann. Letzteres ist eine der größten Restriktionen dieses Verfahren. Da der Mensch nicht in der Lage ist, hoch-präzise Trajektorien vorzugeben, liegt der Hauptanwendungszweck vor allem in der Programmierung von Anwendungen mit niedriger Genauigkeit, wie eben der Spritzlackierung sowie Dichtungs- und Bogenschweiß Tätigkeiten (Nof, 1999; Rahimi & Karwowski, 1992a).

Stärker noch als beim *indirekten Teach-In* ist die Anpassbarkeit oder auch die Korrekturfähigkeit der erzeugten Trajektorien beschränkt. Darüber hinaus sind auch keine Analysemöglichkeiten oder Korrekturmaßnahmen der erzeugten Programme auf Basis von Sensordaten vorhanden.

Das *direkte Teach-In* stellt eines der ältesten Verfahren dar, das zum Beispiel für die Programmierung der ersten Lackierroboter verwendet wurde. Allerdings ist diese Art

der Programmierung trotz seiner intuitiven und schnellen Eingabe von räumlichen Bewegungseingaben heutzutage praktisch nicht mehr im Einsatz (Schraft & Meyer, 2006; Vogl, 2009). Eine Wiederbelebung könnte allerdings auf Basis aktueller Leichtbauroboter, wie dem KUKA iiwa⁶ erfolgen. Diese erlauben auf Basis ihrer Gelenkmomenten-Sensorik und Kraftregelung eine schnelle Programmierung von einfachen Trajektorien.

4.3.2.3 Sensorgestützte Programmierverfahren

Laut Spur & Uhlmann (2012) lassen sich *sensorgestützte Programmierverfahren* in *manuell sensor kontrollierte* und *automatisch sensor geführte* Verfahren unterteilen. Ersteres kann dabei als weiteres *Playback*-Verfahren gesehen werden, da hier ein sensorisch erfasstes Eingabetool, üblicherweise in Form eines Stiftes, dazu verwendet wird dem Roboter eine Trajektorie „vorzuzeichnen“. (Pritschow & Tuffentsammer, 1987)

Bei den *automatisch sensor geführten* Verfahren erzeugt der Nutzer nur grobe Bewegungsvorgaben und es erfolgt daraufhin ein automatisches und vom Sensor gesteuertes Scannen des zu bearbeitenden Bauteils oder der Arbeitsfläche. Im, von Pan et al. (2012) dargestellten, Beispiel einer sensor geführten Online-Programmierung ist das System somit in der Lage mit Hilfe von optischer Sensorik einer mit einem Stift vorgegeben Bahn zu folgen. Der Unterschied zur *manuell sensor kontrollierten* Methodik ist hier, dass der Roboter nicht der Bewegung des Stiftes direkt, sondern der gezeichneten Linie folgt (siehe Abbildung 16).

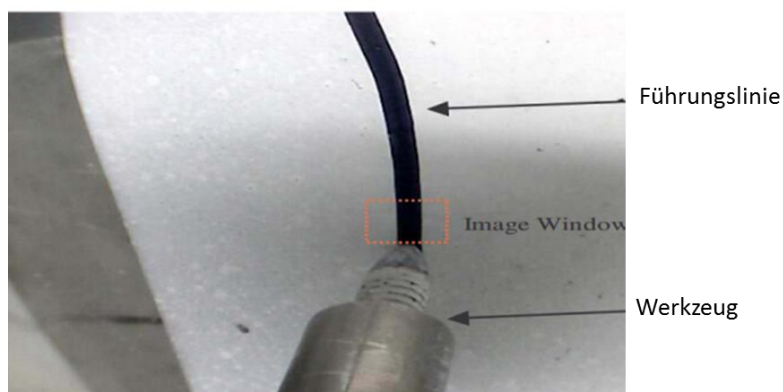


Abbildung 16: Darstellung eines sensorgeführten Verfahrens (Zhang, Chen, Xi, Zhang, & He, 2006) auf Basis visueller Führung

⁶ <http://www.kuka-lbr-iiwa.com/de/>

Neben der Führung auf Basis einer, durch den Nutzer erzeugten, Markierung ist ebenso das Folgen von Bauteilgeometrien denkbar. Unterstützt kann diese Methodik durch den Einsatz eines Kraft-Sensors werden, der das Halten des Kontakts zur Oberfläche erlaubt. Genauso ist auch eine Darstellung von Bewegungs- und Bearbeitungsanweisungen auf dem zu bearbeitenden Bauteil selbst möglich, wie Solvang, Sziebig, & Korondi (2008) zeigen.

Durch die Fähigkeit sich an Gegebenheiten des Arbeitsplatzes oder des zu bearbeitenden Bauteils zu orientieren, können deutlich höhere Genauigkeiten erzielt werden, als dies mit den *Teach-In*-Verfahren möglich wäre. Leider sind die meisten Systeme bisher sehr stark auf einen Anwendungsfall zugeschnitten und erfordern teure Sensorik (Pan et al., 2012).

4.3.3 Aufgabenorientierte Programmierverfahren

Die bisher dargestellten Verfahren zeichnen sich dadurch aus, dass die Bewegungsbahnen des Robotersystems manuell vorgegeben werden müssen. Mit fortschreitenden Entwicklungen im Bereich der künstlichen Intelligenz, insbesondere in Bezug auf die Muster- bzw. Objekterkennung, werden zunehmend Programmierverfahren erforscht, die in der Lage sind, die benötigten Bewegungen und Handhabungsschritte automatisiert zu bestimmen. Diese Verfahren werden als *aufgabenorientierte* bzw. *implizite Programmierverfahren* bezeichnet. Bei diesen Verfahren wird mit Hilfe der künstlichen Intelligenz die Planungsaufgabe erfasst und geeignete Teilhandlungen abgeleitet. Als Voraussetzung dafür muss eine Modellierung der Umwelt, der Handhabungsobjekte und des Roboters gegeben sein. Dargestellt wird die wesentliche Systemstruktur der *aufgabenorientierten Programmierung* in Abbildung 17.

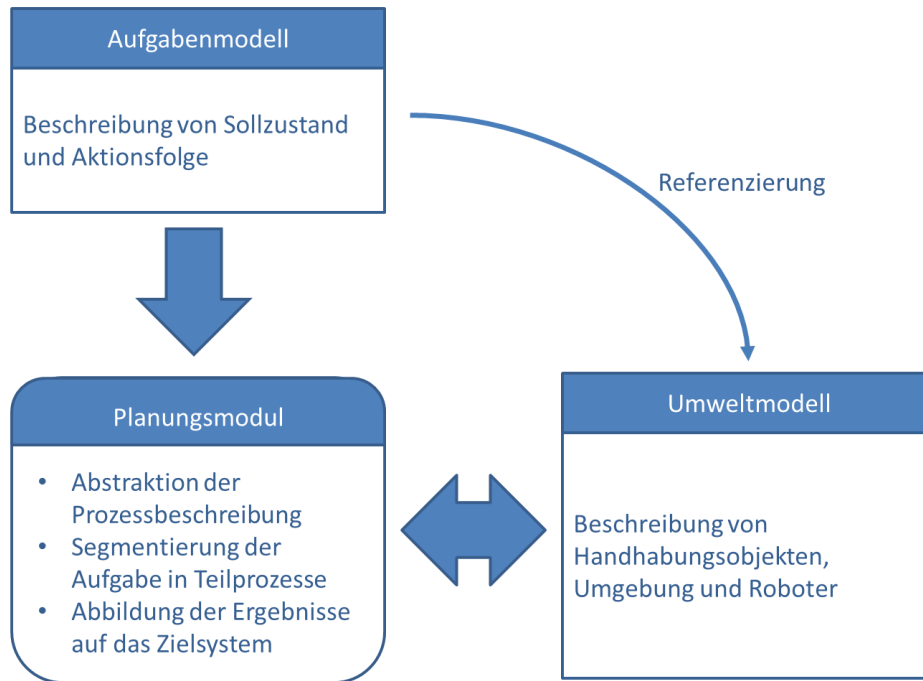


Abbildung 17: Systemstruktur der aufgabenorientierten Programmierung (übersetzt nach Backhaus & Reinhart, 2013; Humburger, 1998)

Das Aufgabenmodell beschreibt dabei den zu erreichenden Sollzustand sowie die dafür notwendige Aktionsfolge des Systems und referenziert dabei die wesentlichen äußeren Bestandteile des Systems. Das Planungsmodul übernimmt die Interpretation bzw. die Abstraktion der Aufgabenbeschreibung und leitet, auch in Abhängigkeit des Zielsystems, elementare Teilhandlungen ab. Somit erlaubt ein solches System die Programmierung stärker vom verwendeten Robotersystem zu abstrahieren. Der Nutzer programmiert folglich nicht die einzelnen Bewegungen des Roboters, sondern spezifiziert die auszuführende Tätigkeit (Aufgabenmodell) auf Basis seines mentalen Modells des Arbeitsplatzes bzw. der auszuführenden Aufgabe. Zusätzlich unterstützt er die Erzeugung des Umweltmodells. Die Vorteile, die sich aus diesem erhöhten Abstraktionsgrads ergeben, sind zum einen die Verringerung der Anforderungen an, vor allem herstellereinspezifische, Programmierkenntnisse sowie zum anderen eine leichtere Übertragbarkeit der erzeugten Programme auf unterschiedliche Systeme (Backhaus & Reinhart, 2013). Darüber hinaus kann vor allem bei ungeübtem Personal davon ausgegangen werden, dass die automatisch generierten Trajektorien Vorteile in Bezug auf Energieeffizienz und Zeit aufweisen. Besonders hervorzuheben ist überdies der ermöglichte Umgang mit verrutschten Handhabungsobjekten, der laut Whitney (1982) eine bedeutende Fähigkeit für den prozesssicheren Betrieb darstellt.

Kugelman (1999) stellt in ihrer Arbeit unterschiedliche Anwendungsszenarien für die *aufgabenorientierte Programmierung* vor. Besonders hervorzuheben sind dabei die Montage, Handhabung von Bauteilen sowie die Laserbearbeitung.

4.3.3.1 Programmieren durch Vormachen (PdV)

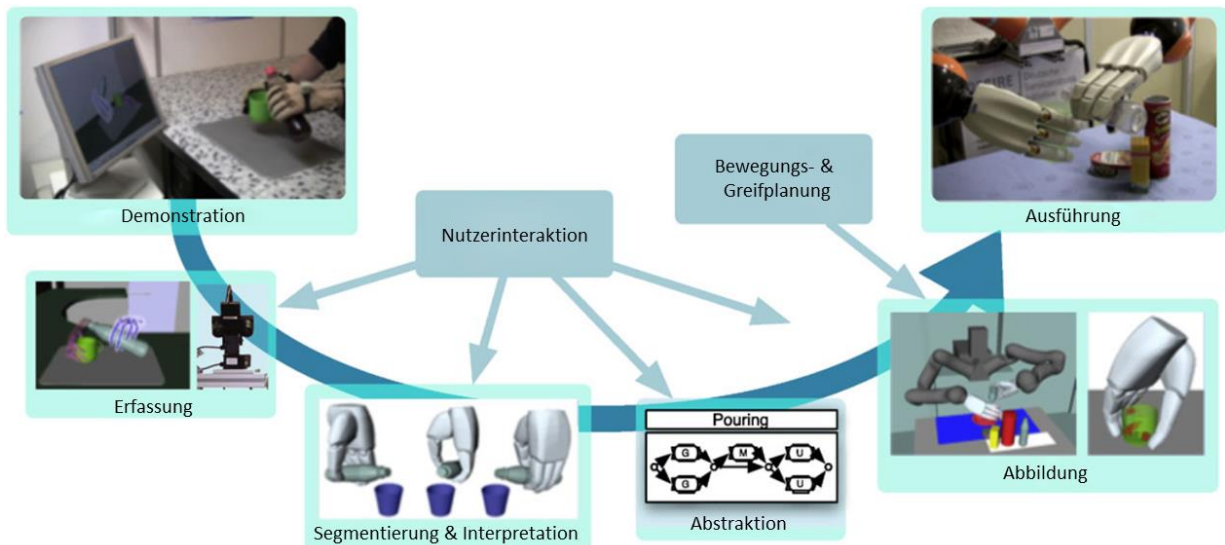


Abbildung 18: Generierung eines neuen Programms mit der Methode PdV (Jäkel, Schmidt-Rohr, Zhixing Xue, Lösch, & Dillmann, 2010)

Das Konzept des *Programmierens durch Vormachen* stellt eine spezielle Variante der *aufgabenorientierten Programmierung* dar. Sie beschreibt das namensgebende Demonstrieren einer Aufgabe, unter passiver Beobachtung des Robotersystems durch den Nutzer, der hier als Experte fungiert (Calinon & Billard, 2007). Der Prozess kann, unter anderem, wie in Abbildung 18 dargestellt werden. Durch die sensorische **Erfassung** des demonstrierten Bewegungsablaufs und der ausgelösten physikalischen Effekte in der Arbeitsumgebung kann das System Rohdaten generieren, die anschließend auf Basis im System vorhandener Elementaroperatoren in Teilhandlungen **segmentiert** werden. Die parametrisierten Teilhandlungen werden anschließend im **Interpretationsschritt** wiederum zu einer Hypothese des Gesamthandlungsablaufs verknüpft. Hierfür sind üblicherweise mehrere Durchläufe von Nöten, bis das System in der Lage ist, die relevante Prozesssegmentierung sowie -Parameter abzuleiten. Durch die Variabilität der Demonstrationen kann das System ableiten, wie genau die Trajektorie in den unterschiedlichen Bereichen des Arbeitsraums gewählt werden muss und welche der

durchgeführten Handlungen tatsächlich relevant sind (Delson & West, 1996). Der Anwender hat anschließend die Möglichkeit die vom System getroffenen Hypothesen und Parameter zu korrigieren. Bei der **Abstraktion** wird dann das erzeugte Handlungswissen in generische Form überführt und in die Wissensdatenbank aufgenommen, sodass es für ähnliche Problemstellungen, z. B. ein anderes Robotersystem, anwendbar ist. Die **Abbildung** auf das eigentliche Zielsystem, also den für die Aufgabe zu verwendenden Roboter, erfolgt erst im nächsten Schritt. Dabei wird die abstrakte Handlung in eine für das System ausführbare Form überführt. Das beinhaltet sowohl die Überführung in für den Roboter ausführbaren Code als auch die Berücksichtigung unterschiedlicher technischer Rahmenbedingungen. Die **Ausführung** findet üblicherweise zuerst virtuell statt. Mit Hilfe der Simulation lässt sich das erzeugte Programm validieren und auf Fehler aus den vorherigen Schritten überprüfen. Die Überprüfung betrifft in diesem Fall nicht nur das erzeugte Programm, sondern auch das Handlungswissen des Systems, sprich die erlernten Lösungsstrategien zu Bewältigung generischer Aufgaben. Finaler Schritt ist dann die Ausführung der realen Aufgabe. Auch hier kann das System noch aus Fehlern lernen. Ein Beispiel dafür kann das Fallenlassen eines Bauteils sein, worauf die Greifkraft in Zukunft erhöht wird. (Zöllner, Rogalla, Ehrenmann, & Dillmann, 2005)

Die Methode *Programmieren durch Vormachen* stellt laut Zöllner et al. (2005) eine der vielversprechendsten Möglichkeiten dar, die Nachteile der bisherigen Programmieransätze in Bezug Komplexität und zeitlichen Aufwand der Programmierung, insbesondere in der Servicerobotik, zu überwinden. Allerdings kommt PdV bisher nicht über die Anwendung in Laborumgebungen und für einfache Aufgaben, wie z. B. Pick & Place-Prozesse, hinaus (Siegert & Bocionek, 1996; Zöllner, Rogalla, Dillmann, & Zöllner, 2002). Eines der größten Probleme bleibt es, die Intention des Nutzers zu erkennen (Pardowitz, Knoop, Dillmann, & Zöllner, 2007). Darüber hinaus beschreiben Siegert & Bocionek (1996), dass, je nach Wahl der Interaktion, ein weiteres Hindernis die häufig nicht vorhandene Aufnahme der Handhabungskräfte ist.

Den genannten Herausforderungen widmeten sich in den vergangenen Jahren einige interessante Forschungsprojekte. Besonders erwähnenswert sind dabei die EU-Projekte SMErobot (Pires, 2006) und ROSETTA (Patel, Hedelind, & Lozan-Villegas, 2012). SMErobot beschäftigte sich mit der Einbindung der *aufgabenorientierten Programmierung* für die namengebenden Small and Medium Enterprises, sprich

KMUs. ROSETTA setzte an, um die Ergebnisse aus dem Projekt SMErobot noch weiter zu verfeinern. Hauptziel war die Entwicklung einer automatischen Programmgenerierung auf Basis einer Wissensdatenbank. Allerdings wurden diese Projekte mit starkem Bezug zum verwendeten Robotersystem durchgeführt und können dementsprechend nicht als vollständig „implizit“ betrachtet werden. (Backhaus & Reinhart, 2013)

Tarchanidis & Lygouras (2003) sowie Zollner et al. (2002) zeigten zudem vor allem die Verwendung von Datenhandschuhen mit taktilem Datenerfassung als sinnvolle Eingabetechnologie auf, die auch die Herausforderung der Erfassung von Greifkräften bewältigen kann.

Eine Untergliederung der einzelnen PdV-Varianten findet sich bei Knoop, Pardowitz, & Dillmann (2008).

4.3.3.2 Instruktive (dialogorientierte) Programmierung

Die *instruktiv dialogorientierte Programmierung* tritt vor allem im Kontext der Servicerobotik und der Mensch-Roboter-Kooperation auf. Mit Berücksichtigung der dialogorientierten Komponente gibt der Nutzer dafür verbale Befehle, die das System in Echtzeit analysiert und interpretiert. Darauf aufbauend wählt es aus einem vorhandenen Set an Elementaroperatoren eine geeignete Strategie zur Aufgabenerfüllung. Allerdings beschränken sich die Fähigkeiten eines instruktiv dialogorientierten Robotersystems nicht zwangsläufig nur auf die Ausführung von Befehlen. Auch die Anweisung von menschlichen Mitarbeitern ist in kooperativen Szenarios denkbar.

Neben der *instruktiv dialogorientierten Programmiermethode* gibt es weiterhin auch instruktive Verfahren die rein auf der Verwendung von Gesten basieren. Biggs & MacDonald (2003) bieten in ihrer Arbeit eine Übersicht unterschiedlicher Ansätze, die sich mit der Verwendung von reiner Gestensteuerung beschäftigen. Aus ergonomischer Sicht ist dabei aber insbesondere die Verknüpfung von beiden Methoden, also der Gesten- sowie der Sprachsteuerung, zu einem multimodalen Verfahren sinnvoll, da auch die menschliche Kommunikation über die Verwendung der Sprache hinausgeht. Foster, By, Rickert & Knoll (2006) beschreiben ein multimodal funktionierendes System bei dem der Roboter JAST in der Lage ist, den Zusammenbau eines kleinen Holzflugzeuges mit einem Mitarbeiter zu koordinieren (siehe Abbildung 19).

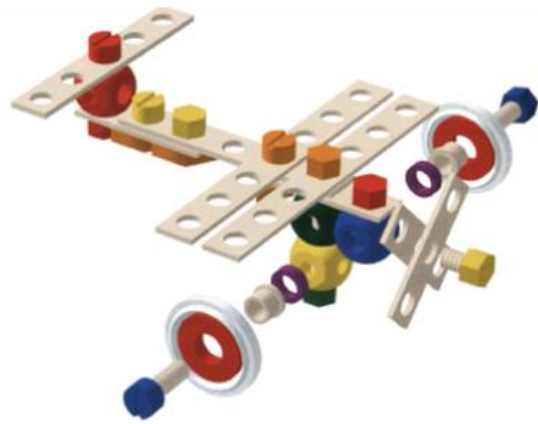


Abbildung 19: Roboter YAST (Foster et al., 2006) und Darstellung einer beispielhaften Aufgabe für die kollaborierende Arbeit von Roboter und Mensch (Rickert et al., 2007)

Hierbei gibt der Roboter Anweisungen und wartet auf die Aufgabenerfüllung durch den menschlichen Mitarbeiter. Das System erlaubt auch die Erfüllung einer gemeinsamen Aufgabe, wenn nur einer der beteiligten Partner das Wissen über die korrekte Vorgehensweise besitzt. Zusätzlich ist auch die Beteiligung mehrerer kollaborierender Partner denkbar ist. So wird in einer Weiterentwicklung des Roboters YAST gezeigt, dass dieser auch in dynamischen Situationen mit mehreren menschlichen Partnern agieren kann. Bei den Untersuchungen übernahm der Roboter die Rollen des Barkeepers und konnte erfolgreich diejenigen Kunden erkennen, die bestellen wollten. Bemerkenswert dabei ist, dass der Roboter auch bei schnell aufeinanderfolgenden Bestellungen unterschiedlicher Getränke diese noch in korrekter Form übergeben kann. (Foster et al., 2012)

Eine industrielle Umsetzung dieser Programmiermethode stellt Pires (2005) dar, die im Funktionsumfang allerdings auf einfache Pick & Place-Aufgaben beschränkt ist und in der dargestellten Weise noch nicht an die im vorherigen Beispiel umgesetzte Natürlichkeit der Sprache heranreicht. In diesem Bereich gibt es dementsprechend auch aktuell noch Forschungsbedarf. Ein Aspekt dabei ist die Erkennung zeitlicher Anordnungen in Befehlen. Landhäußer, Hey, & Tichy (2014) gelangen hier erste Fortschritte.

Trotz des Weiteren Forschungsbedarfs ist diese Programmiermethode sehr vielversprechend und könnte in Zukunft sowohl in der Service-Robotik als auch der

industriellen Anwendung für schnell und intuitiv programmierbare Lösungen sorgen (Vogl, 2009).

4.4 Diskussion

Innerhalb dieses Kapitels wurden die wesentlichen Aspekte der Robotik, inklusive der unterschiedlichen Interaktionstechnologien, für die Eingabe von Geometriedaten, und einer sinnvollen Auswahl an Programmiermethoden dargestellt.

Diese Elemente bilden aus technischer Sicht den Kern aktueller Mensch-Roboter-Interaktionen, wobei die fortschreitende Entwicklung abstrakter Programmierverfahren die Eingabe von Geometriedaten immer stärker in den Hintergrund drängt beziehungsweise obsolet macht. Dies ist insbesondere in Anbetracht der geschilderten Notwendigkeit mentaler Transformationen bei vielen technischen Lösungen und den damit auftretenden Belastungen auf den Nutzer ein sinnvoller Schritt.

Darüber hinaus sollte berücksichtigt werden, dass die Programmierverfahren nur die technische Basis einer nutzerzentrierten MRI darstellen können und eine zusätzliche ergonomische Umsetzung der einzelnen Elemente bzw. des Interaktionskonzepts erforderlich ist, um einen hohen Grad an Gebrauchstauglichkeit zu erreichen. In diesem Zusammenhang kann allerdings allgemein konstatiert werden, dass die Ergonomie noch nicht ausreichenden Anklang bei der Gestaltung von MRI-Konzepten findet. Dautenhahn (2007) beschreibt die Forschung auf dem Gebiet der Mensch-Roboter-Interaktion sogar als in den Kinderschuhen befindlich. Die aktuelle Forschung im Bereich der Mensch-Roboter-Interaktion kann als überwiegend technologiegetrieben beschrieben werden, bei der vor allem auch sicherheitsrelevante Themen eine deutlich größere Rolle spielen als die Entwicklung einer nutzerzentrierten Interaktion (Scholtz, 2003). Damit liefert die Forschung gute und sinnvolle Ansätze, auf deren Basis nun weitere Entwicklungen mit Fokus auf die ergonomische Gestaltung von MRI sowie den Fähigkeiten der Nutzerklientel folgen müssen.

5 Konzeptionelle und technische Rahmenbedingungen

Die aus der Motivation abgeleiteten Anforderungen an die zu entwickelnde Roboterzelle können wie folgt zusammengefasst werden: Die Befähigung des vorhandenen Personals das System in maximal zwei Arbeitstagen (á drei Schichten) so zu konfigurieren und programmieren, dass die Arbeitsaufgabe vollständig und kollisionsfrei durchgeführt werden kann. Um dieses Ziel zu konkretisieren ist es, dem nutzerzentrierten Gestaltungsprozess (Abbildung 4) folgend, erforderlich, in einem ersten Schritt die Nutzerklientel, den Aufgabeninhalt und etwaige Umwelteinflüsse genau zu spezifizieren. Für die Nutzerklientel konnten so im Rahmen einer Voruntersuchung im Projekt viEMA folgende Fähigkeiten als gegeben festgestellt werden:

- Kenntnisse in der Bedienung von Automatanlagen (z. B. Programmauswahl und Umrüsten)
- Allgemeine systematische Fähigkeiten bei der Fehlerdiagnose
- Mechanische und elektrische Grundfertigkeiten für Greiferwechsel, Anpassung des Maschinenschutzes sowie der Schnittstelleninstallation
- Keine Expertise im Bereich der Programmierung sowie Messtechnik

Dementsprechend kann die Nutzerklientel als Vor-Ort-Personal definiert werden, das in Bezug auf die hardwareseitige Konfiguration der Roboterzelle ausreichend befähigt ist, bei der Programmierung aber unterstützt werden muss. Dazu wird als gegeben definiert, dass jeder Nutzer eine einwöchige Schulung als Vorbereitung für den Umgang mit dem System durchläuft.

Als Anwendungsrahmen für die transportable Roboterzelle werden Pick & Place-Arbeitsplätze definiert. Diese eignen sich insbesondere auf Grund ihrer geringen Anforderungen an Werkzeugvariationen, der Positionierung der Zelle und der üblicherweise hohe Flexibilität bezüglich Bauteillagen. Die Zelle soll dabei genutzt werden, um im Falle von Stückzahlspitzen die Tätigkeit für einen menschlichen Mitarbeiter zu übernehmen. Um dieses Ziel zu unterstützen, ist es erforderlich die Umbaumaßnahmen auf ein Minimum zu reduzieren. Die Zelle muss darüber hinaus wirtschaftlichen Anforderungen genügen, die die Gestaltung der Interaktion zumindest beeinflussen. Ein Kriterium der Wirtschaftlichkeit ist die Limitierung der Zielkosten eines solchen Systems auf die Personalkosten, die bei manueller Tätigkeit

innerhalb eines Jahres anfallen würden. Diese Kostengrenze hat bei der Gestaltung einige essentielle Auswirkungen auf die Ausstattung, insbesondere durch die Limitierung der optischen 3D-Erfassung auf zwei Sensoren der Firma ISRA VISION⁷. Jeder dieser Sensoren ist in der Lage eine Erfassung von Bauteilen mit allen sechs Freiheitsgraden durchzuführen. Da in einem üblichen Pick & Place-Szenario davon ausgegangen werden kann, dass sowohl für den Aufnahme- und Ablagebereich ein 3D-Kamerasystem vorhanden sein muss, ist es notwendig eine der beiden Kameras für den Umgebungsscan umzumontieren und auf den Roboterarm der Zelle anzubringen. Damit ist ein höherer zeitlicher Aufwand bei der Konfigurierung der Zelle zu erwarten und es werden Umgebungsscans im automatisierten Prozess unterbunden. Darüber hinaus ist diese wirtschaftliche Limitierung ein entscheidender Faktor bei der Gestaltung des Interaktionsparadigmas und der Auswahl der verwendbaren Eingabekonzepte für Geometriedaten. Für die Auslegung der unterschiedlichen Hardware-Komponenten als auch der Interaktion in der Konzeptionierungsphase der Roboterzelle wurden folgende Handarbeitsplätze als typische Einsatzszenarien definiert (siehe Abbildung 20).



Elektrische Prüfung Anker



Verpackungsplatz Steckerleiste



Beladung CNC-Maschine
(Fräuserschleifen)



O-Ring Montage T-Stück

Abbildung 20: Anwendungsszenarien für die viEMA-Zelle

⁷ www.isravision.com/

Die Szenarien umfassen:

1. Die elektrische Prüfung eines Ankers
2. Das Ablegen von Steckerleisten in Transportkartons
3. Den Wechsel von Fräsköpfen für eine CNC-Maschine sowie
4. Die Montage eines O-Rings auf ein T-Stück

Diese Szenarien stellen die Bandbreite an unterschiedlichen Herausforderungen im Bereich der manuellen Handarbeitsplätze im betrachteten Werk der Robert Bosch GmbH dar und wurden deshalb als Anforderung definiert. Allen Szenarien ist dabei gleich, dass der Roboter nur Pick & Place-Tätigkeiten durchführt und eventuelle Füge-Vorgänge oder die Prüfung eines Werkstücks über ein zusätzliches Blackbox-System erfolgt. Eine Ausnahme bildet dabei die optische Prüfung, die unter anderem in Szenario 2, sprich der Verpackung von Steckerleisten, erforderlich ist. Da die optische Prüfung oft Teil der Verpackungsaufgabe ist, muss diese vom System durchgeführt werden können, da ansonsten das Prinzip einer nur minimalen Änderung des Arbeitsplatzes bei Verwendung der Roboterzelle nicht mehr gegeben wäre. Die verwendete Sensorik von ISRA ist für diese Art der optischen Prüfung geeignet. Der Handhabungsbereich von Bauteilaufnahme und –Ablage erstreckt sich im Rahmen der untersuchten Szenarien auf einen Bereich von 400 x 400 mm. Diese Fläche entspricht auch dem Sensorbereich der 3D-Sensorik, der durch die anvisierte Taktfrequenz der Roboterzelle, die maßgeblich durch die benötigte Zeit eines Bauteilscans beeinflusst wird, und der erforderlichen Auflösung, bei gegebener Bauteilgröße, abgedeckt werden kann.

Auf Basis all dieser Anforderungen wurde eine prototypische Roboterzelle umgesetzt die in Abbildung 21 in den unterschiedlichen Entwicklungsstufen dargestellt wird.

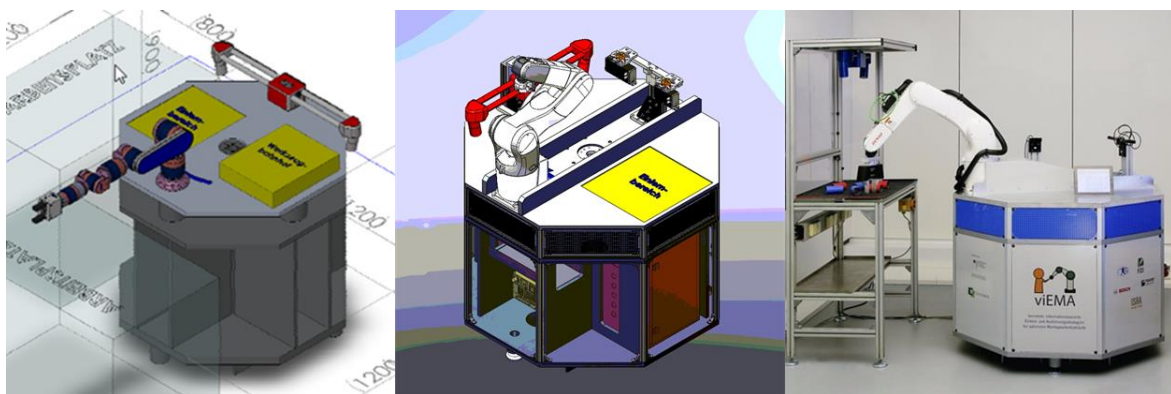


Abbildung 21: Demonstrator viEMA-Zelle

Interessante Aspekte des Demonstrators für das Interaktionskonzept sind die hohe Modularität des Roboters, die eine Verwendung anderer Roboterarme möglich macht, sowie die drehbare Plattform der Zelle. Diese erlaubt es, den Roboter in die Nähe der Handhabungsbereiche zu verfahren und somit den Arbeitsraum zu erhöhen. Weiterhin verfügt das System über einen dedizierten Bereich für Bauteilmodellierung sowie Aufhängungen für Greifer und Kamera.

Auf Basis der verfügbaren Programmiermethoden wurde für die Gestaltung der transportablen Roboterzelle das Interaktionsparadigma *aufgabenorientierte Programmierung* (Kapitel 4.3.3) ausgewählt. Diese Auswahl stellt einen Kompromiss zwischen Prozesssicherheit, hohem Abstraktionsgrad und Kosten dar. Gleichzeitig wird als Back-up-Szenario eine Form der Teach-In-Programmierung einzelner Aktionen vorgesehen, die vor allem einer zusätzlichen Prozesssicherheit und der geringen Anzahl an 3D-Sensoren geschuldet ist. Die großen Vorteile der *aufgabenorientierten Programmierung* bezüglich der Flexibilität, sprich die automatische Generierung von Trajektorien und Griffposen, bleiben für allgemeine Aufgaben, die die Handhabung von Bauteilen auf ebener Fläche oder in bzw. aus Kisten umfassen, erhalten. Für diese Szenarien ist das System weiterhin in der Lage unterschiedliche Bauteillagen zu handhaben und erfordert keine dedizierte Programmierung einzelner Trajektorien und Griffposen durch den Anwender. Im Zuge der hier umgesetzten *aufgabenorientierten Programmierung* muss der Nutzer die korrekten Teilhandlungen sowie das zu handhabende Bauteil und den Handhabungsbereich spezifizieren. Zur Festlegung des Handhabungsbereichs ist es notwendig, eine Eingabe für Geometriedaten vorzusehen. Hierfür sind unterschiedliche Eingabettechnologien zur Markierung geeignet. Da das Gesamtkonzept vorsieht, dass auch Black-Box-Stationen unterstützt werden sollen, die üblicherweise durch die Bauteilaufnahme Probleme mit der Kameraerfassung erzeugen, muss eine Interaktion gewählt werden, die sechsdimensionale Geometrien mit ausreichender Genauigkeit vorgeben kann.

Um das System in die Lage zu versetzen, Griffposen für die zu handhabenden Bauteile zu erzeugen oder diese auf optische Fehler zu überprüfen, muss der Anwender 3D-Modell der Bauteile bereitstellen. Ein Importieren vorhandener Bauteile, zum Beispiel aus der Konstruktion, ist möglich. Darüber hinaus wird vorgesehen, dass die Modellierung als Teil des Programmierprozesses vorgenommen werden kann. Hieraus ergeben sich unterschiedliche Anforderungen

an die Bedienoberfläche, die eine Validierung der Modelle mit dem realen Konterpart in großer Darstellungsqualität erlauben muss.

6 Untersuchung des mentalen Modells der Nutzerklientel

Für die Entwicklung einer nutzerzentrierten MRI ist es sinnvoll ein konzeptionelles Modell zu schaffen, das sich bereits nahe an dem mentalen Modell der Nutzerklientel orientiert, um so die benötigte Zeit zum Erlernen des Systems zu reduzieren. Da im Rahmen der Dissertation eine komplette Neuprogrammierung der Systemarchitektur für den Roboter nicht möglich ist, basiert die Programmierung im Kern weiter auf einem technikorientiertem Ansatz. Die Systemarchitektur wird deshalb um eine Programmebene erweitert, die dem Nutzer eine optimierte Gestaltung der Prozessprogrammierung, zur Definition des Aufgabenmodells (Abbildung 17), präsentiert. Gleichzeitig wird die benötigte Transformation zu den verwendeten *flexiblen Programmen* (FP) des FZI systemtechnisch gelöst.

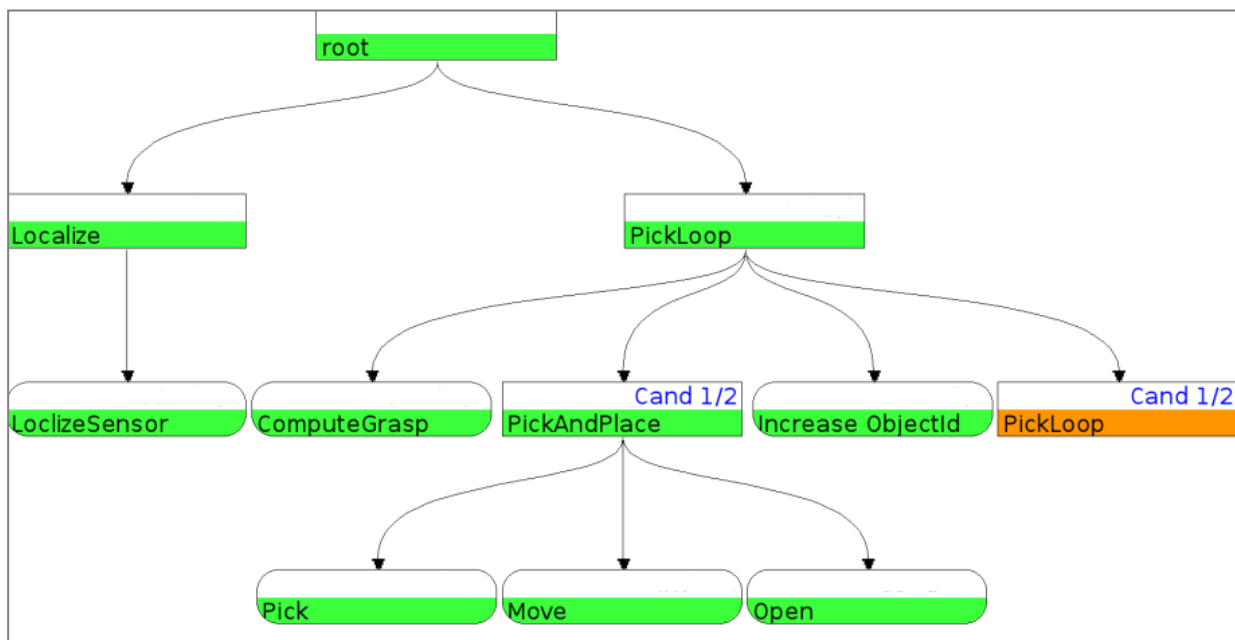


Abbildung 22: Pick & Place-Vorgang im FP des FZI

Abbildung 22 stellt einen einfachen Prozess in der Version der FP dar, die für die transportable Roboterzelle verwendet wird. Es lässt sich erkennen, dass dieses Konzept vor allem als Expertensystem ausgelegt ist und somit zwar ein Höchstmaß an Flexibilität bietet, aber auch dementsprechend komplex ist.

Ziel der in diesem Kapitel dargestellten Untersuchung ist folglich die Entwicklung eines Satzes an Programmfunktionen bzw. einer Programmierstruktur, die an dem mentalen Modell der Nutzer bezüglich Pick & Place-Arbeitsplätzen ausgerichtet ist.

Das erzeugte Modell wird dann für die Umsetzung der Prozessprogrammierung in der zu entwickelnden grafischen Benutzeroberfläche herangezogen. Da die Transformation zwischen nutzerzentriertem Ansatz und bisheriger technikzentrierter Struktur durch das System geleistet werden muss, ist es notwendig, dass der zu entwickelnde Satz an Systemfunktionen konsistent ist und sich auf die flexiblen Programme abbilden lässt. Vorbereitend, als Teil der Versuchskonzeption, wurden verfügbare Arbeitsanweisungen für einzelne Arbeitsplätze der Robert Bosch GmbH analysiert.

6.1 Versuchsdesign

Das Erfassen des mentalen Modells von Arbeitsvorgängen ist kein einfaches Unterfangen, da durch den Versuch Verzerrungseffekte (z. B. Priming (Kahneman, 2014; Tulving, Schacter, & Stark, 1982)) auftreten können. Gleichzeitig fällt es Probanden im Zweifelsfall schwer, ihr mentales Modell des Arbeitsplatzes zu verbalisieren. Aus diesem Grunde liegt dem Versuch das Konzept des Erklärens der Aufgabe von einem Instrukteur für eine ungeschulte Person zugrunde. Dieser Ansatz kann als sehr ursprüngliche und natürliche Form des Lehrens bzw. der Informationsweitergabe (Calinon & Billard, 2007; Silve, 2013) gesehen werden. Diesem Ansatz folgend werden die Probanden für diese Untersuchung in zwei Gruppen unterteilt. Der ersten Gruppe wird der Handarbeitsplatz vor der Untersuchung vorgestellt und sie gilt fortan als *Instrukteursgruppe*. Da die Ergebnisse der Untersuchungen im großen Maße davon abhängen, wie gut Einwirkungen auf die Probanden reduziert oder komplett unterbunden werden können, muss während des Versuchs streng darauf geachtet werden, dass weder Vokabular noch Syntax oder Segmentierung der Gesamtaufgabe durch den Versuchsleiter beeinflusst werden. Um dies zu gewährleisten findet die Darstellung des Arbeitsplatzes mit Hilfe eines Videos statt, das ohne verbale Informationen und Videoschnitte die Pick & Place-Aufgabe darstellt.

In Folge muss sichergestellt werden, dass der Umfang der Tätigkeit durch die Probanden der *Instrukteursgruppe* verinnerlicht wurde. Um dies zu gewährleisten und um Sicherheit bei der Durchführung der Arbeitsaufgabe zu gelangen, bekommen die Probanden die Möglichkeit die Arbeitsaufgabe selbst durchzuführen.

So geschult führt die Expertengruppe die zweite Gruppe, die *Neulinge*, durch einen realen Pick & Place-Prozess und gibt die aus ihrer Sicht notwendigen Informationen

für die Durchführung der einzelnen Handlungsschritte. Dies erlaubt die Erfassung des individuell verwendeten Vokabulars als auch der subjektiv wahrgenommenen Teilhandlungen. Darüber hinaus kann durch mögliches Nachfragen des *Neulings* ein Fehlen wichtiger Informationen oder Zwischenschritte für die Aufgabendurchführung erfasst und dementsprechend ein Unterschied im mentalen Modell abgeleitet werden.

Um die Daten zu verfeinern, wird der Versuch um mehrere vorgelagerte schriftliche Aufgaben (siehe Anhang A.1) erweitert. So soll mit der Erarbeitung einer Arbeitsanweisung ermittelt werden, welche Anweisungen der spätere Nutzer als relevant für die erfolgreiche Erfüllung der Aufgabe erachtet. Da zu diesem Zeitpunkt noch keine Interaktion mit dem zweiten Probanden auftritt und die Wahl der Darstellung frei ist, kann davon ausgegangen werden, dass die Zahl der Störgrößen bei der Beschreibung des Prozessablaufs gering ist. Gleichzeitig bietet diese Aufgabe die Möglichkeit, die Kenntnisse über den Prozess zu vertiefen und sich bereits im Vorfeld Gedanken zu den einzelnen Teilhandlungen zu machen, da hier keinerlei zeitlicher Druck vorhanden ist.

Ein wichtiger Aspekt für die Gestaltung der Prozessprogrammierung in der späteren GUI ist der Segmentierungsgrad des Gesamtprozesses. Für die Evaluierung der optimalen sowie der maximalen Segmentierungstiefe werden die Probanden gebeten den Gesamtprozess nach definierten Gliederungsstufen einzuteilen. Die Festlegung dieser Gliederungsstufen erfolgte auf Basis eines Vorversuchs in die Kategorien 4-5, 6-7 und 8-9 Teilschritte. Eine stärkere Differenzierung erfolgt nicht, da die Aufgabe auch in dieser Ausfertigung schon von den Probanden des Vorversuchs als repetitiv und ermüdend wahrgenommen wurde. Zusätzlich wurden die Probanden gebeten, eine Beschreibung des Prozesses mit maximaler Segmentierungstiefe durchzuführen.

Für die Umsetzung der Prozessprogrammierung ist eine Anordnung der hier zu erarbeitenden Funktionen in einem Flussdiagramm vorgesehen. Dementsprechend wird innerhalb des Fragebogens zusätzlich untersucht, ob die Probanden mit dieser Darstellungsform vertraut sind. Das Verfahren zur Bestimmung eines mentalen Modells und der Darstellung der Ergebnisse mit Hilfe eines Flussdiagramms wurde in ähnlicher Form ebenfalls in einer späteren Untersuchung von Silve (2013) angewendet.

6.1.1 Versuchsaufbau und Probandenkollektiv

Die Untersuchungen zum mentalen Modell wurden mit Hilfe eines realen, aber ausgemusterten, Handarbeitsplatz der Firma Bosch durchgeführt. Dieser entspricht den, durch die Einsatzszenarien, definierten Anforderungen.

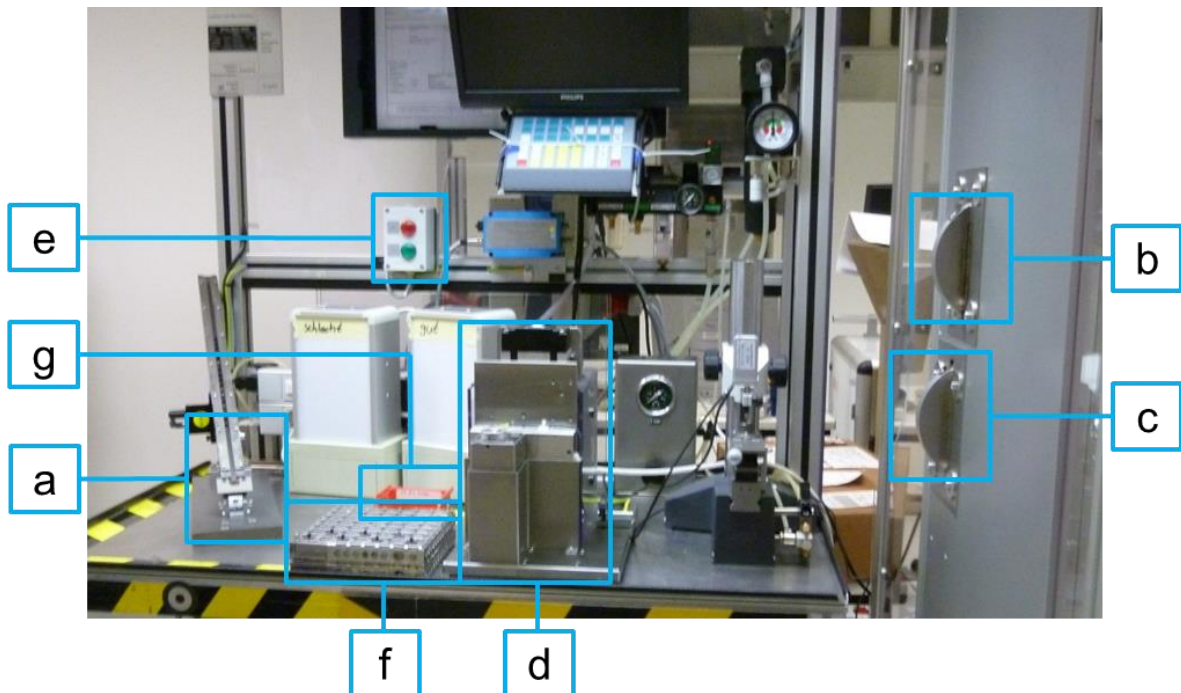


Abbildung 23: Versuchsaufbau zur Untersuchung des mentalen Modells

Der Zweck der, in Abbildung 23 dargestellten, Station ist das automatische Schleifen von sogenannten Ankern und deren anschließende Prüfung. Aufgabe des Werkers ist es, den Anker aus der Zuführrampe bei a) zu entnehmen und dann in die Schleifmaschine bei b) einzulegen. Das fertige Bauteil wird dann bei c) aus der Schleifmaschine entnommen und bei d) in die Prüfstation eingelegt. Die Prüfung erfolgt erneut automatisch und der Worker muss anhand der Rückmeldung bei e) das Bauteil ablegen. Ein grünes Licht bestätigt die korrekte Bearbeitung des Ankers und das Bauteil kann bei f) in die Bauteilaufnahme, nach definiertem Muster (siehe Abbildung 24), abgelegt werden.

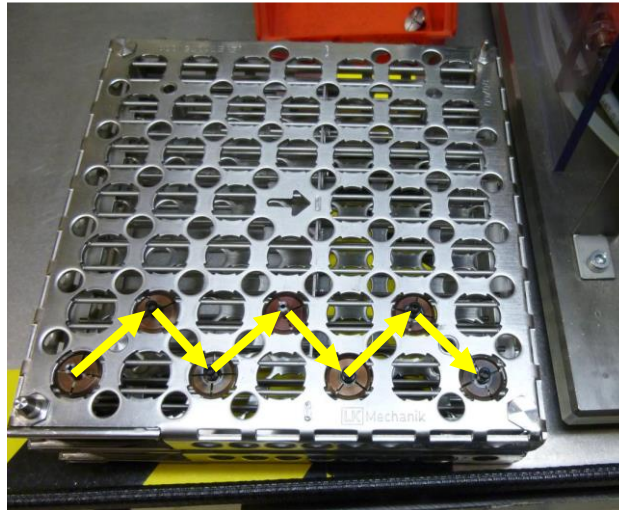


Abbildung 24: Bauteilaufnahme

Wird das Bauteil als fehlerhaft (rotes Licht) dargestellt, muss es in den Ausschuss bei g) gelegt werden. Dabei lassen sich bezüglich der, für den Arbeiter notwendigen, Handlung Ähnlichkeiten zwischen den jeweiligen Entnahme- (bei a und c) und Einlegeprozessen (bei b und d) feststellen. Im ersten Fall entnimmt der Arbeiter von einer fest definierten Position das Bauteil, um sie in letzterem Fall an eine fest definierte Position abzulegen. Im Kontrast stehen hier das Ablegen des Ankers in den Ausschuss, bei dem der Ablageort in Grenzen variabel ist, und die Einordnung in die Bauteilaufnahme, bei der die Ablage des Bauteils zwar pro Durchgang fest zugeordnet ist, aber darüber hinaus in fester Folge variiert.

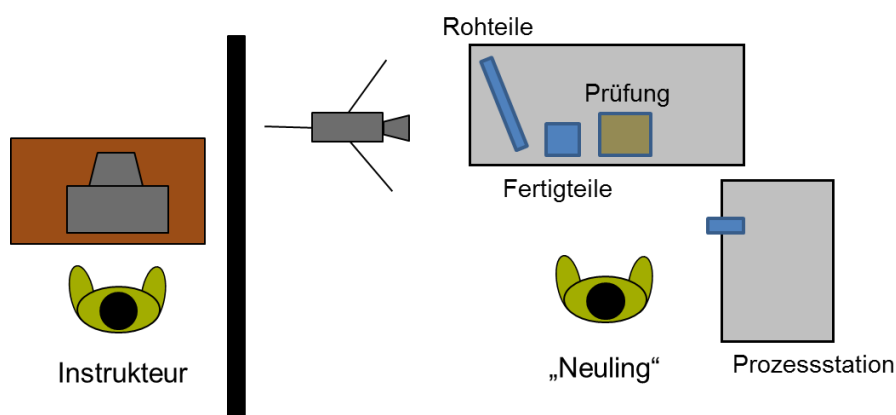


Abbildung 25: Schematische Darstellung des Versuchsaufbaus

Für den Versuch wurden *Instrukteur* und *Neuling* räumlich getrennt. Hiermit sollte sichergestellt werden, dass jegliche non-verbale Kommunikation zwischen den beteiligten Probanden unterbunden wird, die dann die Art und Ausführung der Anweisungen beeinflussen könnte.

Um dennoch sicherzustellen, dass der Instrukteur den aktuellen Stand der Durchführung und die Reaktion des Neulings nachvollziehen kann, war es ihm möglich die Arbeiten sowohl über eine Webcam als auch über die Wiedergabe des DIKABLIS-Systems nachvollziehen (siehe Abbildung 26).

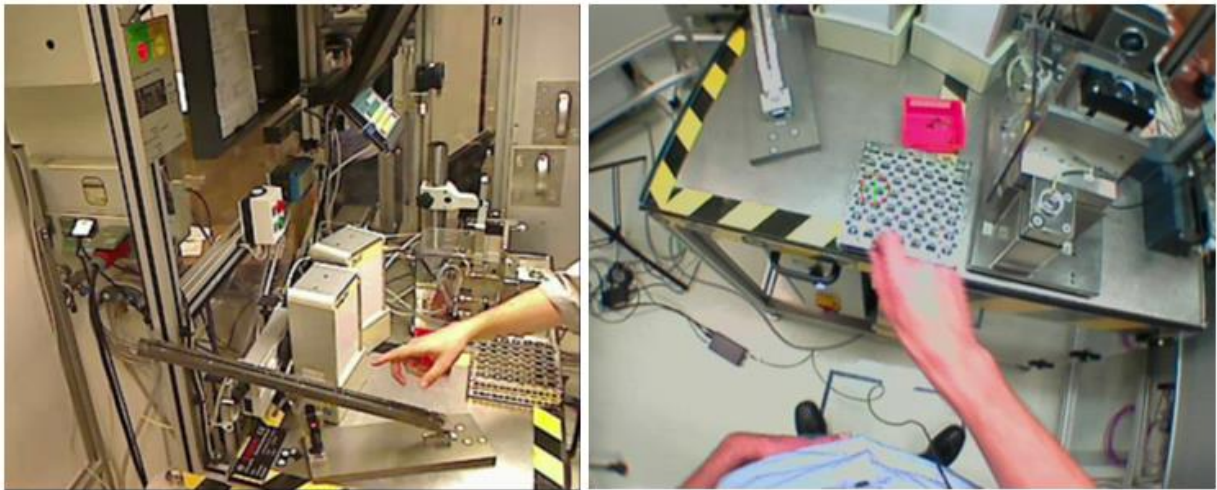


Abbildung 26: Wiedergabe der Aufgabenvollständigkeit (links Webcam, rechts DIKABLIS)

An der Studie zum mentalen Modell nahmen insgesamt zwölf Probanden der Firma Bosch teil, die als typische Nutzerklientel der späteren Roboterzelle ausgewählt worden sind. Hierbei lag der Fokus auf Mitarbeitern, die in ihren Fähigkeiten der Definition aus Kapitel 5 entsprachen und gleichzeitig den zu bearbeitenden Arbeitsplatz höchstens bedingt, also nicht aus eigener Tätigkeit, kannten. Beteiligt an dieser Studie waren in Folge ein weiblicher und elf männliche Probanden, deren durchschnittliches Alter $37,6 \pm 12,3$ Jahre betrug.

6.1.2 Versuchsaufbau

Der Ablauf der Studie zur Erhebung des mentalen Modells der Nutzerklientel ist in Abbildung 27 dargestellt.

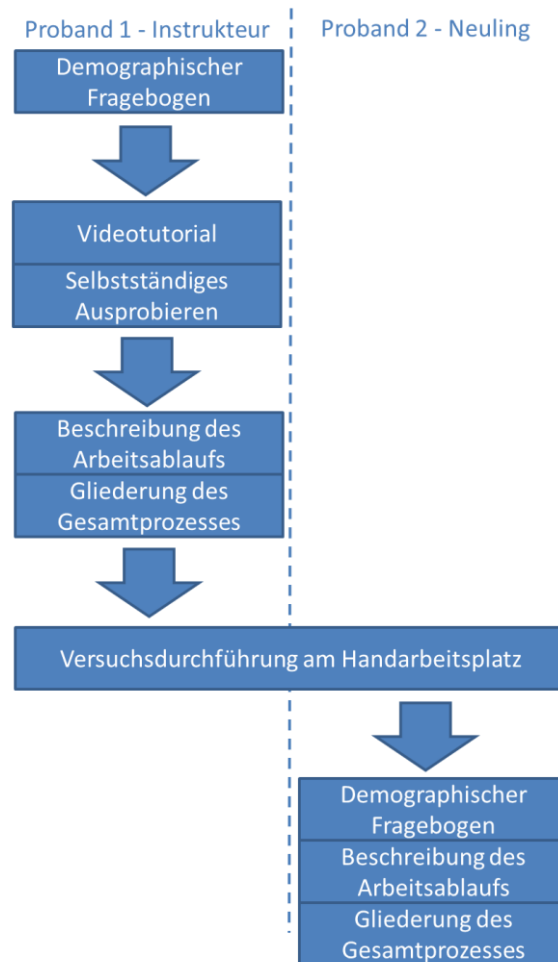


Abbildung 27: Versuchsaufbau „Untersuchung des mentalen Modells“

Zu Beginn wurden die beteiligten *Instrukteure* gebeten einen Fragebogen mit allgemeinen/demografischen Fragestellungen zu beantworten. Anschließend wurde ihnen die Versuchsumgebung dargestellt und der Tätigkeitsablauf mit Hilfe eines Videos präsentiert. Das Video enthielt, wie bereits dargestellt, keinerlei verbale Informationen, sondern beschränkte sich darauf den Ablauf in mehreren Durchläufen zu demonstrieren. Zur Überprüfung der Vollständigkeit und Richtigkeit des Verständnisses des Arbeitsablaufs wurden die Probanden gebeten, die Aufgabe zumindest einmal selbst durchzuführen.

Der erste Aufgabenpunkt bei der eigentlichen Durchführung der Versuchsaufgaben war die Anfertigung einer Arbeitsanleitung unter der Prämisse, dass die zukünftigen neuen Mitarbeiter rein auf Basis dieses Dokuments in der Lage sein würden, den Arbeitsablauf zu erlernen. Hierbei stand es den Probanden frei Stichpunkte oder Fließtext zu verwenden. Als Nächstes wurde den Probanden die Aufgabe gestellt, den gesamten Arbeitsablauf mit den definierten Gliederungsstufen zu beschreiben. Der eigentliche Versuch umfasst dann die Anleitung des *Neulings*. Hierbei wurde die Arbeitsaufgabe sieben Mal, also bis zur Vollendung des dargestellten Ablagemusters durchgeführt. Im Anschluss an die Versuchsdurchführung wurde auch der zweite Proband gebeten, den kompletten Fragebogen zu bearbeiten.

6.2 Ergebnisse

Auf Grund der nur geringen Zahl an verfügbaren Probanden aus der Nutzerklientel werden die Ergebnisse deskriptiv dargestellt. Eines der bedeutendsten Themen dieser Studie ist dabei die Frage nach der geeignetsten Segmentierungstiefe eines typischen Pick-& Place-Arbeitsplatzes, um dem folgend eine geeignete Darstellung der Prozessprogrammierung zu schaffen. Für die Beantwortung dieser Fragestellung ist die maximale Zahl der von den Probanden wahrgenommenen Handlungsschritte von besonderer Bedeutung. Eine Darstellung der Ergebnisse liefert Abbildung 28:

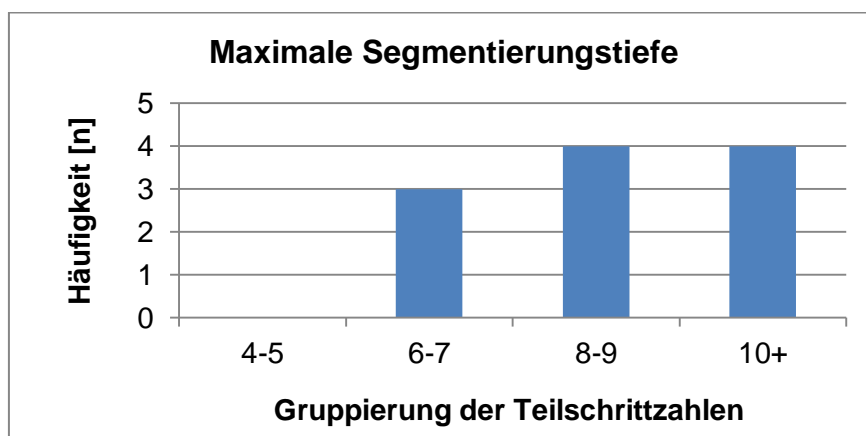


Abbildung 28: Maximale Segmentierungstiefen des Probandenkollektivs

Die Ergebnisse zeigen, dass die maximale Segmentierungstiefe für den Prozess interindividuell sehr stark variiert. Für das betrachtete Szenario lässt sich die untere Grenze bei 6-7 Teilhandlungen darstellen. Das bedeutet, dass drei Probanden den Prozess in nicht mehr als sechs bis sieben Teilhandlungen gliedern können. Da

angenommen werden kann, dass es einfacher ist, einen Prozess zu abstrahieren als ihn zu konkretisieren, soll diese Segmentierungstiefe im Weiteren berücksichtigt werden. Diese Entscheidung lässt sich auch durch die Auswertung der bevorzugten Teilschrittzahl der einzelnen Probanden stützen (siehe Abbildung 29).

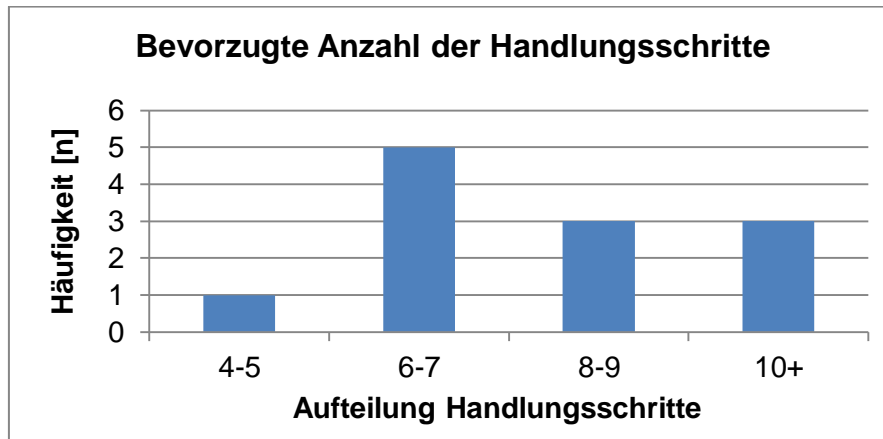


Abbildung 29: Bevorzugte Anzahl an Teilhandlungen

Hierbei lässt sich eine leichte Tendenz der Präferenz bei 6-7 Teilschritten erkennen. Auffällig ist zudem, dass in zwei Fällen eine Diskrepanz zwischen der Zahl der genannten bevorzugten Handlungsschritte und der tatsächlichen Umsetzung der Gliederung auftritt. So werden bei je einem Probanden mit Präferenz 6-7 Teilschritte bzw. 8-9 Teilschritte eine feinere Segmentierung als Wunsch genannt als in den einzelnen Teilversuchen umgesetzt werden konnte. Wird diese Besonderheit berücksichtigt und die präferierte Segmentierungstiefe dahingehend angepasst, so ergibt sich eine stärkere Betonung der Kategorie 6-7 Teilschritte.

Tabelle 1: Matrix-Darstellung der im Versuch genannten Teilprozesse

	VP1	VP1n	VP2	VP3	VP3n	VP4	VP4n	VP5	VP5n	VP6	VP6n
Entnehmen 1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Vorprüfung	x	x	x	x	x	x	✓	x	x	x	x
Einlegen 1	✓	✓	✓	✓	✓	✓	○	✓	✓	✓	✓
Entnehmen 2	✓	✓	○	○	✓	✓	✓	✓	✓	✓	✓
Einlegen 2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Warten/Prüfen	x	✓	✓	○	✓	✓	✓	○	○	✓	✓
Entscheidung	x	x	x	x	x	✓	x	x	x	x	x

Entnehmen 1	✓	x	✓	✓	x	x	o	✓	✓	✓	o
Ablegen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Dieser Argumentation folgend lassen sich die von den Probanden (hierbei steht das „n“ für die Probandengruppe der *Neulinge*) genannten Konstrukte bzw. Teilhandlungen in Tabelle 1 darstellen. Für jede Versuchsperson (VP) wird dabei unterschieden, ob die jeweilige Teilhandlung bei der gewählten (✓) bzw. einer anderen (o) Segmentierungstiefe genannt wird oder bei der Auswertung des Probanden in dieser Form nicht auftritt (x).

Auf Basis der Nennungen, zeigt sich, dass das Entnehmen der Anker aus der Zuführschiene (*Entnehmen 1*) sowie das Einlegen in die Messmaschine (*Einlegen 2*) für alle Probanden relevante und nachvollziehbare Arbeitsschritte sind und somit auch bei der Untersuchung der für den Prozess sinnvoller Konstrukte eine wesentliche Rolle spielen. Das Einlegen (*Einlegen 1*) sowie das Entnehmen (*Entnehmen 2*) des Ankers in die bzw. aus der Schleifmaschine sind ebenfalls bedeutend. Es zeigt sich, dass ein bzw. zwei Probanden diese Schritte innerhalb der 6-7 Teilschritte nicht explizit erwähnen. Allerdings finden sie sich bei anderen Segmentierungstiefen und dementsprechend kann davon ausgegangen werden, dass die Teilhandlungen wahrgenommen und verstanden werden. Ein ähnliches Ergebnis findet sich bei der Teilhandlung Prüfen bzw. Warten. Nur ein Proband impliziert diese Teilhandlung komplett, die übrigen sind sich dieser Teilhandlung bewusst.

Bei der Entnahme des Ankers aus der Prüfvorrichtung lässt sich erkennen, dass dieser Schritt nur in knapp mehr als der Hälfte der Fälle genannt wurde. Auch hier lässt sich anhand der späteren Nennungen erkennen, dass die Handlungen für die meisten Probanden durchaus eine gewisse Bedeutung besitzt. Allerdings fällt der Anteil an Probanden, die diese Teilhandlung in keiner Gliederungsstufe nennen mit 25 % relativ hoch aus. Da die Einbringung dieser Teilhandlung auf Grund der Konsistenz zur übrigen Prozesssegmentierung durchaus sinnvoll ist, wird eine Unterstützung für die spätere Umsetzung vorgesehen. Diese kann dann sowohl in Form eines automatischen Hinzufügens der notwendigen Funktion als auch durch eine Plausibilitätsschranke erreicht werden.

Je zwei Konstrukte wurden überdies nur von je einem Proband als bedeutsam deklariert bzw. überhaupt in der Form wahrgenommen. Diese können als Ausreißer betrachtet und so für die weitere Verwendung der Ergebnisse vernachlässigt werden. Für die Verwendung in einem Funktionsmodell bzw. für eine spätere Prozessprogrammierung muss eine eindeutige Benennung dieser Konstrukte gefunden werden. Auf Grund der geringen Stichprobe für den Gesamtversuch werden die Nennungen auf Basis der Häufigkeitsverteilung untersucht.

Am Beispiel des ersten Konstrukts (siehe Abbildung 30) sollen die unterschiedlichen Nennungen explizit dargestellt werden. Dabei werden die verwendeten Begriffe aus der schriftlichen Arbeitsanweisung, der Prozesssegmentierung und der Syntax während des Versuchs (nur Instruktoren) mit einbezogen.

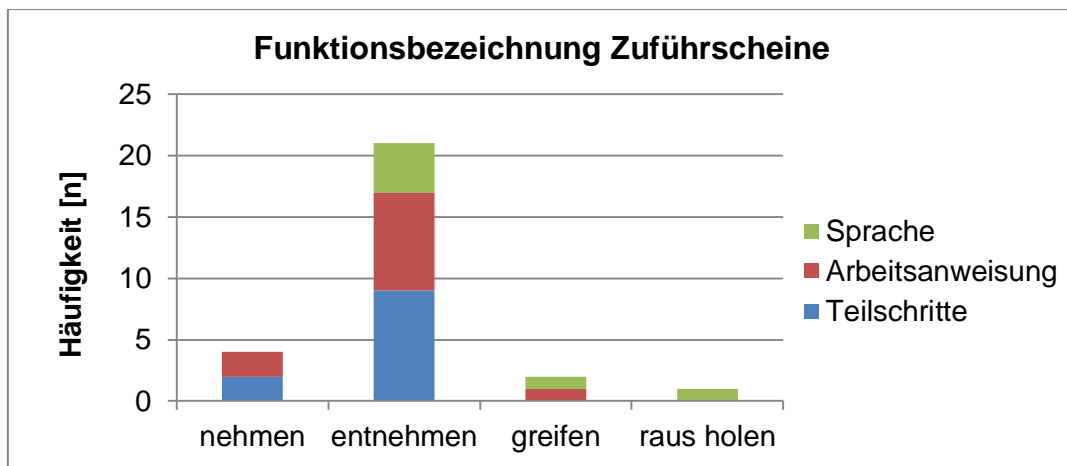


Abbildung 30: Häufigkeitsverteilung der verwendeten Begriffe für die Entnahme des Ankers aus der Zuführschiene

Für die Benennung dieser Teilhandlung zeigt sich ein sehr homogenes Bild. Dementsprechend scheint die Wahl des Begriffs valide für die Bezeichnung des ersten Operators. Dies gilt insbesondere, da der Wortstamm „nehmen“ bis auf eine Ausnahme für die gesamte Stichprobe als angemessen erscheint. Die übrigen Darstellungen finden sich im Anhang B.

Für die Bildung des finalen Funktionssatzes wird in einem Fall von dieser Häufigkeitsverteilung abgewichen. Die Benennung des Zuführens des Ausschuss in die Box bzw. der korrekt produzierten Anker in die Musteraufnahme erfolgt durch die Probanden sehr ähnlich, mit jeweils leichter Tendenz zu „Einlegen“. Da sich die

dazugehörige Handlung aus Sicht des Systems stark unterscheidet, wird hier eine Trennung in *Ablegen* (für Box) und *Einlegen* (für Musteraufnahme) durchgeführt. Auf Basis des resultierenden Satzes an Funktionen ließe sich die Ankerfertigung wie folgt darstellen:

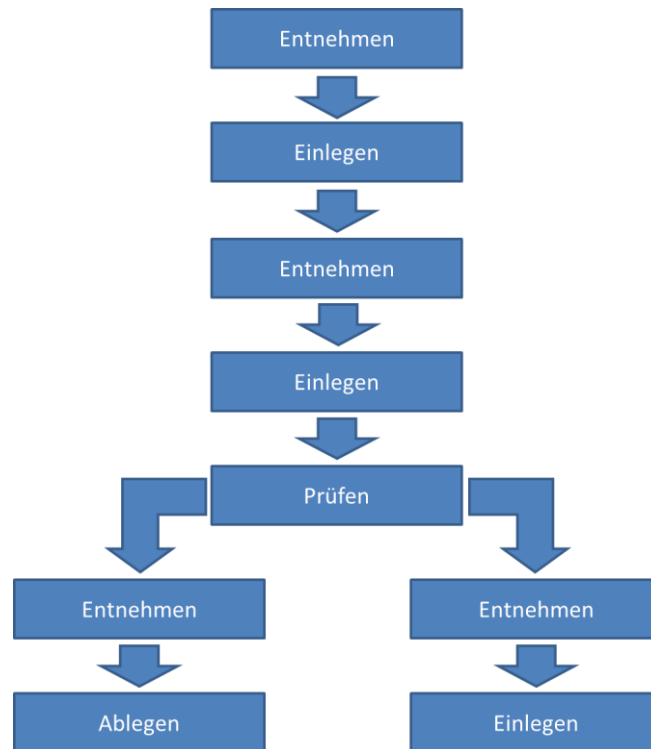


Abbildung 31: Anker-Fertigung mit angepasstem Funktionssatz

Die Fixation auf einen realen Arbeitsplatz hat den Nachteil, dass nicht jede Handhabungsart abgedeckt werden kann. Insbesondere die Handhabung eines frei liegenden Bauteils wurde nicht vollständig berücksichtigt. Für diese Handlung wird der Begriff „Greifen“ eingeführt. Das lässt sich einerseits durch die häufige Verwendung durch die Versuchspersonen und die starke Abgrenzung des Wortstammes zu „Nehmen“ begründen. Letzteres erlaubt eine einfachere Unterscheidung. Insgesamt ergeben sich so acht Funktionen zur Programmierung des Roboters (siehe Abbildung 32).

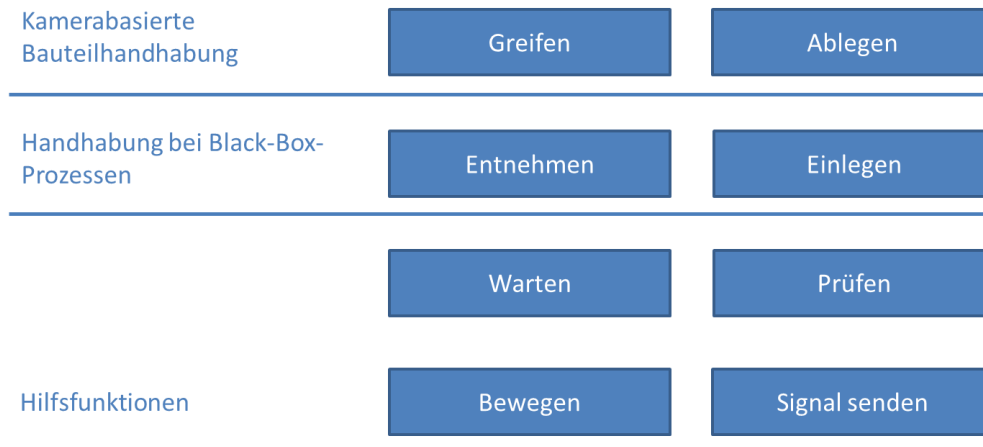


Abbildung 32: Verfügbare Funktionen für die Prozessprogrammierung

Die Funktionen „Greifen“ und „Ablegen“ stellen dabei kameragestützte Prozesse dar, bei denen der Roboter ein Bauteil von einer ebenen Fläche aufnimmt bzw. auf einer ebenen Fläche oder innerhalb einer Kiste ablegt. Entnehmen und Einlegen bezeichnen vor allem die Handhabung zu bzw. von einer fixen Bauteilaufnahme, die vor allem im Zusammenhang mit Blackbox-Prozessen auftreten.

In Bezug auf die *flexiblen Programme* stellt vor allem die programmtechnischen Transformation der Funktion „Greifen“ eine Herausforderung dar. Die Funktion „Greifen“ umfasst dabei grob folgende Teilprozesse:

- Objekt-ID für das Bauteil definieren
- Bauteile lokalisieren
- Überprüfung, ob Bauteile gefunden wurden
- Lage des Bauteils ermitteln
- Ablage-Position bestimmen
- Greifprozess starten
- (eventuell) Roboter in sichere Transportposition bringen
- (eventuell) Den Drehtisch des Roboters anpassen

Interessant an der Auflistung ist die Bestimmung der Ablage-Pose als Teil des Greifprozesses. Bereits für die Wahl des richtigen Griffs ist es erforderlich, dass das System überprüft, ob ein Umgreifen des Bauteils zwingend notwendig ist oder vermieden werden kann. Da ein Umgreifen starken Einfluss auf die Taktzeiten hat, sollte das System, wenn möglich, Umgreifvorgänge vermeiden. Dementsprechend muss die Ablageposition schon bei Planung des Greifvorganges erfasst werden. Auf

Grund der Wahl eines Drehtisches für die Erweiterung des Arbeitsraumes ist es zudem sinnvoll, eine sichere Transportposition des Roboters zu definieren, die eingenommen wird, bevor eine Drehung des Tisches erfolgt.

Die Transformation des hier entwickelten Funktionssatzes wurde vollständig implementiert und auf seine Funktionsfähigkeit überprüft.

Neben der Untersuchung des mentalen Modells galt es, die Vertrautheit der Probanden mit der Darstellungsform des Flussdiagramms zu erfassen. Hierbei ließ sich feststellen, dass jeder der befragten Probanden diese Art der Prozessdarstellung kennt und sich ein Umgang damit zutraut.

6.3 Diskussion

Auch wenn die Ergebnisse vor allem eine große Nähe der gewählten Begriffe der einzelnen Probanden zueinander zeigten, so ist das Ergebnis möglicherweise von regionalen Einflüssen, wie Dialekten, abhängig. Darüber hinaus wäre auch eine Prägung durch das Firmenumfeld und der vorhandenen Arbeitsanleitungen denkbar. Die erarbeiteten Konstrukte entsprechen nämlich in Teilen den verwendeten Begriffen, der vorher untersuchten Arbeitsanweisungen der Fa. Bosch.

Gleichzeitig ist die Untersuchung nur auf eine manuelle Tätigkeit beschränkt. Es ist denkbar, dass die Gestalt der Arbeitsplätze einen Einfluss auf die Segmentierung der Gesamtaufgabe hat. Ein Aspekt ist hier unter anderem, dass die Probanden sowohl bei der Anzahl der präferierten aber auch der maximalen Segmentierungstiefe stark im Bereich der 7 ± 2 psychologischen Einheiten des Arbeitsgedächtnisses bewegen (Miller, 1956).

Da die Ergebnisse dieser Untersuchung in das Konzept der grafischen Benutzeroberfläche implementiert werden, bietet sich im Zuge dessen allerdings die Möglichkeit einer ersten Evaluation der Anwendbarkeit der hier erzielten Ergebnisse.

7 Gestaltung und Evaluation einer grafischen Benutzeroberfläche

Benutzeroberfläche

Eine zentrale Herausforderung der vorgestellten Arbeit ist die Gestaltung einer, für die Kalibrierung sowie Programmierung der Roboterzelle, geeigneten ergonomischen grafischen Benutzerschnittstelle sowie deren Evaluation. Die Durchführung der im folgenden Kapitel vorgestellten Untersuchung fand im Rahmen einer Bachelorarbeit (Egger, 2013) statt. Diese ist Grundlage für die weitere Auswertung, deren Ergebnisse in Kapitel 7.3 dargestellt werden.

Als Hardwareplattform für die Gestaltung und Evaluierung der grafischen Benutzeroberfläche wird auf ein ASUS Tablet vom Typ EeeSlate 121⁸ zurückgegriffen (siehe Abbildung 33).



Abbildung 33: Verwendete Hardware-Plattform

Das Tablet verwendet, wie bei heutigen Geräten dieser Art oder auch bei Smartphones üblich, einen kapazitiven Touchscreen. Dieser ist anwendbar, da für die Einsatzszenarien eine Verwendung von Arbeitshandschuhen ausgeschlossen bzw. unnötig ist. Diese Wahl hat bezogen auf die Nutzerinteraktion den Vorteil, dass sie Multi-Touch-Gesten, also die Interaktion mit mehreren Fingern, erlaubt und so eine direkte Manipulation der notwendigen 3D-Modelle möglich ist (Cooper & Reimann, 2003). Somit stellt die gewählte Hardware-Plattform alle

⁸ http://www.asus.com/Tablets_Mobile/Eee_Slate_EP121

wichtigen Voraussetzungen für die Gestaltung einer ergonomischen grafischen Benutzerschnittstelle dar.

Als Basis für den Gestaltungsprozess kommen die in Kapitel 3.3 dargestellten Grundlagen zum Tragen. Wichtiger Einflussfaktor bei der strukturellen Gestaltung der Bedienoberfläche ist dabei die systemergonomische Analyse (Bubb, 1993). Diese erlaubt die Unterteilung der benötigten Handlungsschritte zur Bewältigung einer Gesamtaufgabe in sequenzielle und simultane Bedienschritte. Dadurch lassen sich Empfehlungen für die Struktur der Bedienung gewinnen. Eine solche wurde in Vorbereitung auf diese Arbeit durchgeführt und die entsprechenden Ergebnisse für den Gestaltungsprozess übernommen. Weiterhin wurde auch das in Kapitel 6 erarbeitete konzeptuelle Modell für die Prozessprogrammierung implementiert. Das Resultat des Gestaltungsprozesses soll im Folgenden dargestellt werden.

7.1 Konzeption der grafischen Benutzerschnittstelle

Der Beginn eines jeden Programmierprozesses wird mit der Anbringung eines eindeutigen Barcodes an den Arbeitsplatz definiert. Hiermit wird eine eindeutige Zuordnung des Arbeitsplatzes zur Aufgabe gewährleistet und eine Konfigurierung des Roboters bei erneutem Einsatz deutlich vereinfacht. Gleichzeitig wird verhindert, dass falsche Programme geladen werden. Nach dem Scannen eines unbekanntes Arbeitsplatzes fordert das System dann zur Erstellung eines neuen Programms auf. Dieser Prozess unterteilt sich in fünf zentrale Prozessschritte: Roboter-Konfiguration, Bauteil-Konfiguration, Prozessprogrammierung, Simulation und Betrieb.

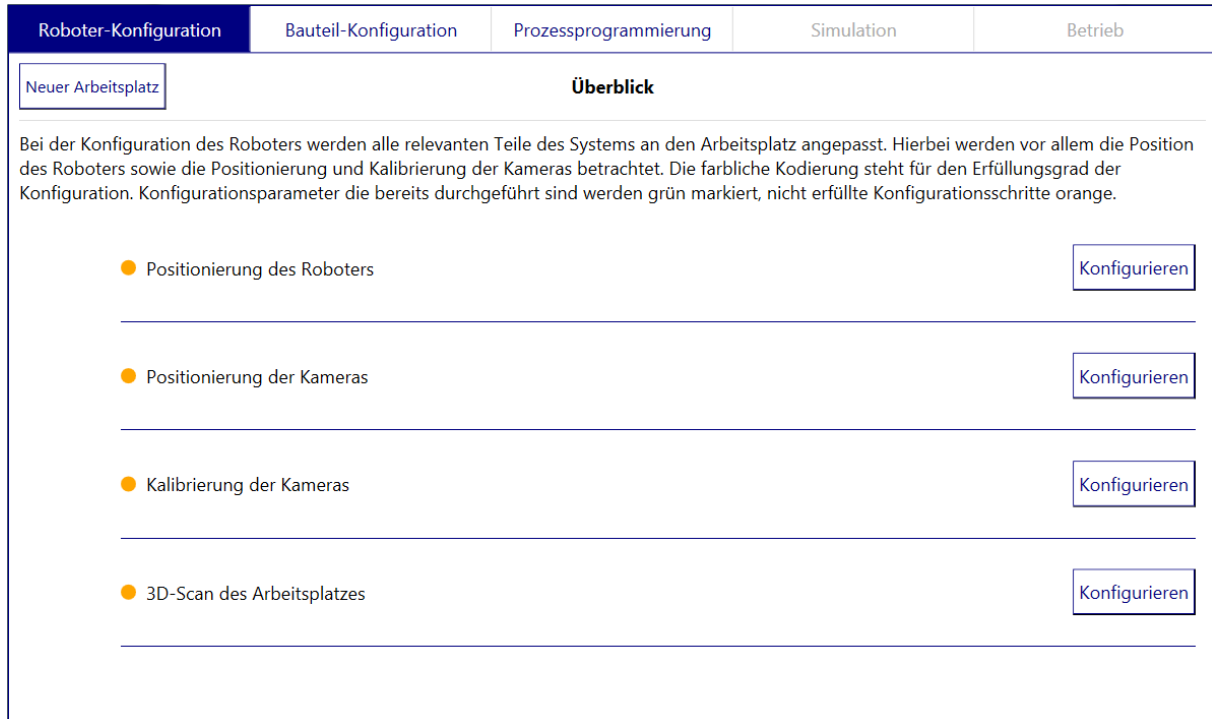


Abbildung 34: Bedienoberfläche „Roboter-Konfiguration“

Die Bearbeitung der ersten drei Prozessschritte ist dabei nicht zwingend sequenziell, wobei eine Vorgehensweise durch die Anordnung der Elemente in der Übersichtsleiste empfohlen wird. Die *Simulation* und der *Betrieb* sind erst zugänglich, wenn die vorherigen Schritte abgeschlossen sind und werden deshalb zu Beginn eines Programmierprozesses farblich ausgegraut.

Der üblicherweise erste Arbeitsschritt, die *Roboter-Konfiguration* (Abbildung 34), beinhaltet die Positionierung der Roboterzelle, die Positionierung und Kalibrierung der beiden 3D-Kameras/Scanner sowie der Darstellung des Umgebungsscans.

Im Prozessschritt *Bauteil-Konfiguration* steht die Bearbeitung der Bauteilliste ((1), siehe Abbildung 35) im Vordergrund.

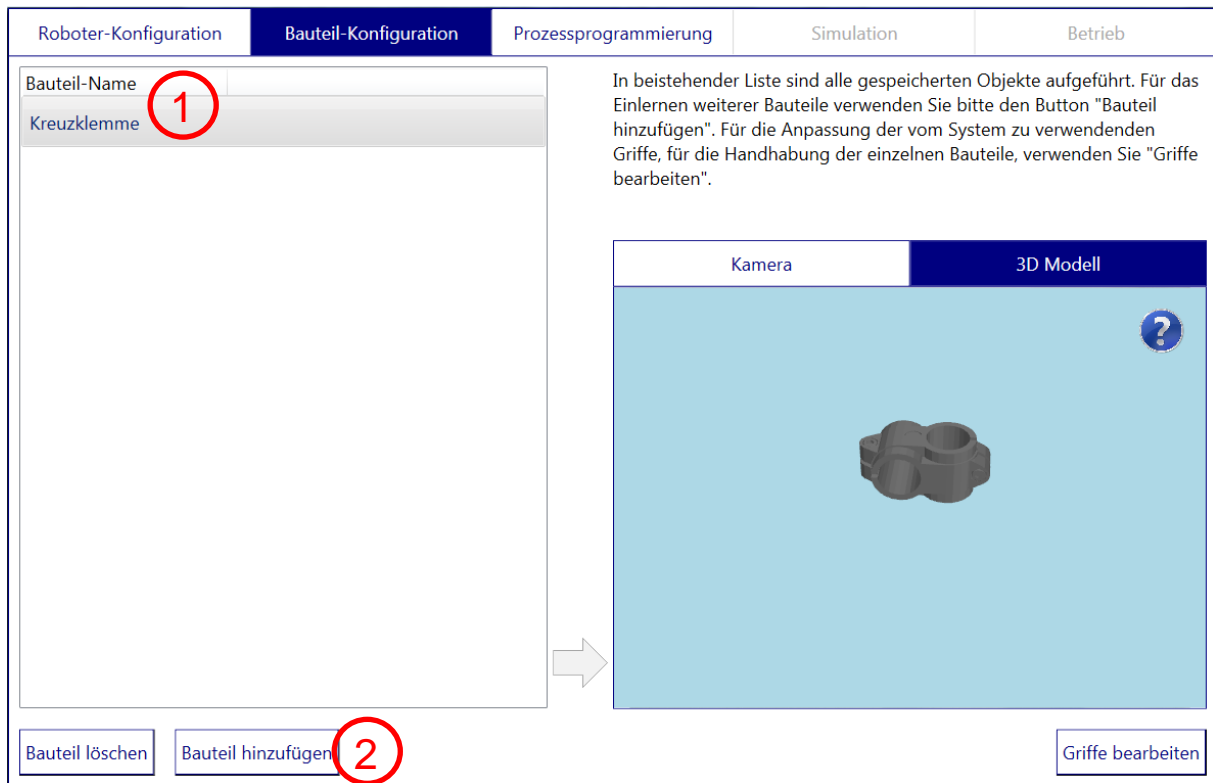


Abbildung 35: Bedienoberfläche „Bauteil-Konfiguration“

Der Anwender überprüft, ob das zu handhabende Bauteil für den Prozess bereits vorhanden ist. Hierfür werden zur eindeutigen Zuordnung über die Bauteilbezeichnung eine Fotoaufnahme sowie ein 3D-Modell bereitgestellt. Damit können auch ähnliche Bauteile im Prozess eindeutig identifiziert werden. Ist das gesuchte Bauteil nicht vorhanden, so muss es über einen gesonderten Scan-Prozess erzeugt werden (*Bauteil hinzufügen* (2)). Bei diesem legt der Anwender das Bauteil auf die dafür vorgesehene Ablage des Roboters und startet den Scan. Ist dieser abgeschlossen muss überprüft werden, ob der Roboter das Bauteil erkennt und das 3D-Modell vollständig ist. Hierfür werden eine Darstellung des Live-Bildes, inklusive Markierung des Bauteils, sowie ein 3D-Fenster zum Betrachten des Modells angeboten. Bei der Erzeugung des 3D-Modells werden vom System zudem automatisch Griffposen für das Bauteil erzeugt. Diese müssen im Anschluss überprüft und instabile bzw. Bauteil beschädigende Griffstrategien aussortiert werden.

Die gesamte Prozedur zur Erstellung eines Bauteilmodells stellt allerdings nur eine Absicherung für den Standardfall dar. Bei diesem wird davon ausgegangen, dass die CAD-Modelle, eine geeignete Musterabbildung und valide Griffe bereits aus der

Konstruktion vorhanden sind und folglich innerhalb einer vom System zugreifbaren Datenbank zur Verfügung stehen.

Im Anschluss an die *Bauteil-Konfiguration* erfolgt die *Prozessprogrammierung*, die auf Basis der Ergebnisse aus dem Versuch zum mentalen Modell entstanden ist und den Arbeitsprozess in Form eines Flussdiagramms präsentiert (siehe Abbildung 36).

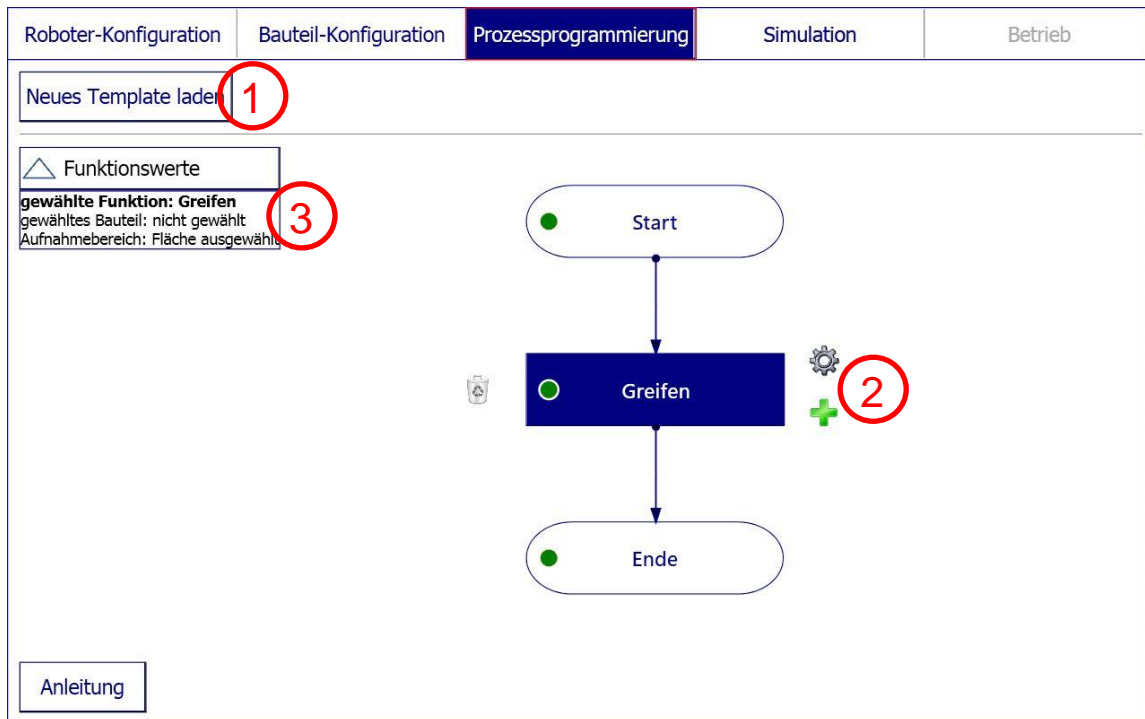


Abbildung 36: Bedienoberfläche „Prozessprogrammierung“

Zur Vereinfachung der Programmerstellung stehen dem Nutzer Templates (1) zur Verfügung, die die gebräuchlichsten Einsatzszenarien abdecken. Zur Anpassung des Templates oder zur Erstellung eines neuen Programms stehen dem Nutzer Funktionsbuttons (2) neben dem aktivierten Operator zur Verfügung. Die Funktionen sind in Abbildung 37 erklärt.

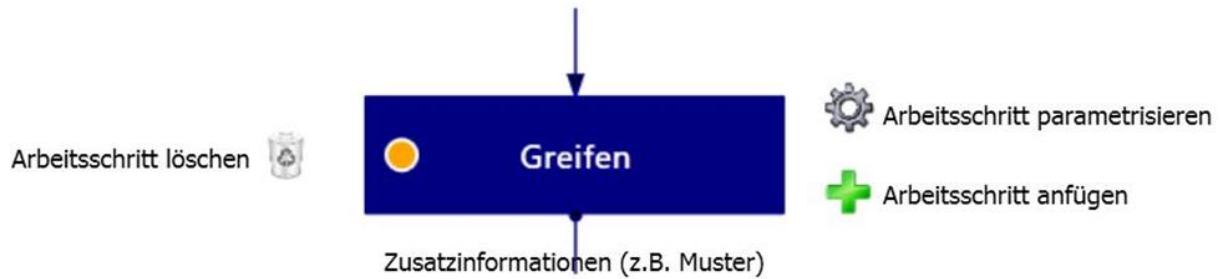


Abbildung 37: Interaktionselemente der Operatoren

Eine wesentliche Funktion dieser Bedienelemente ist der Zugang zur Parametrisierung der jeweiligen Teilhandlung. Für das Beispiel eines Greifprozesses erfolgt so die Definition des Handhabungsbereichs sowie des zu handhabenden Bauteils. Der Erfüllungsgrad eines jeden Elements wird über eine gelbe oder grüne Anzeige im Operator wiedergegeben. Darüber hinaus werden die eingegebenen Parameter für den aktuell angewählten Operator in einer Übersicht ((3) in Abbildung 36) dargestellt.

Im Anschluss an die Programmierung des Prozesses erfolgt eine erste Überprüfung mit Hilfe einer Simulation (siehe Abbildung 38). Dem Nutzer wird hier neben der Darstellung des Roboters (in seiner Arbeitsumgebung) erneut das erstellte Programm angezeigt. So kann die Bewegung des Roboters der aktuellen Teilaufgabe zugeordnet werden, was unter anderem das Kurzzeitgedächtnis des Nutzers entlastet und die Aufgabe damit vereinfacht. Zusätzlich kann die Simulation beliebig verlangsamt werden.

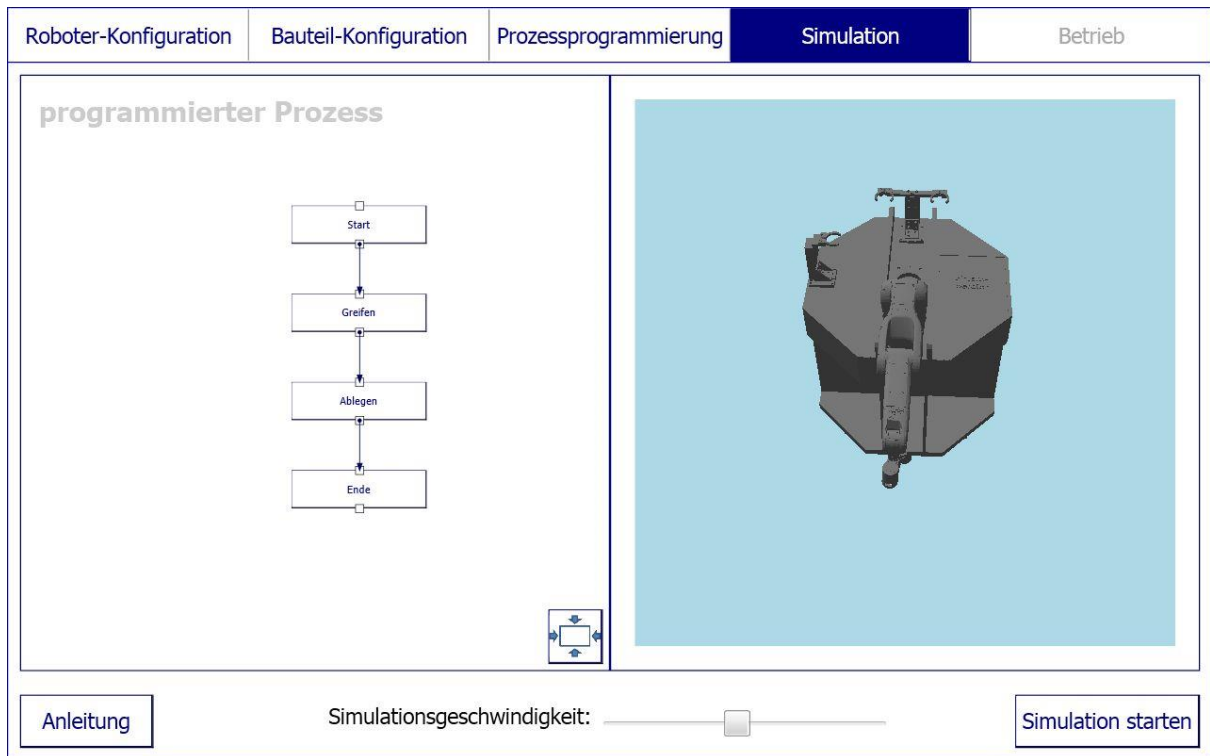


Abbildung 38: Bedienoberfläche „Simulation“

Ist auch diese Simulation erfolgreich absolviert worden, endet der Kalibrier- und Programmierprozess mit der Ausführung der Aufgabe (siehe Abbildung 39).

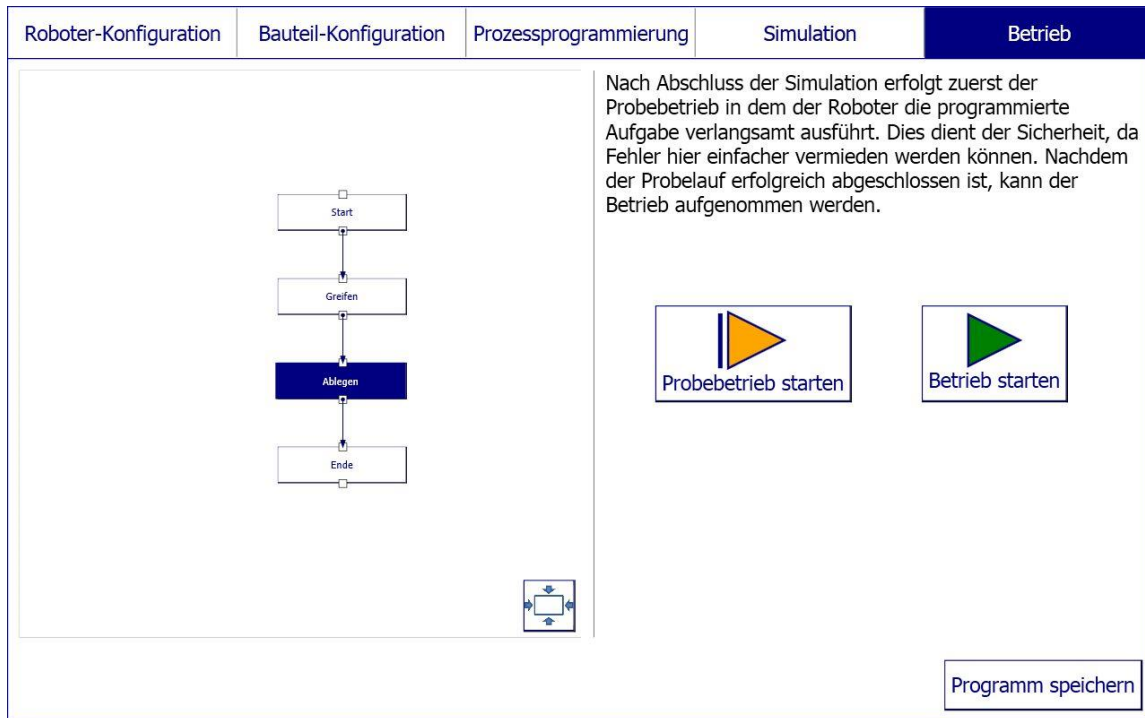


Abbildung 39: Bedienoberfläche "Betrieb"

Um Fehler auszuschließen, die unter anderem im Zusammenhang mit einer ungenauen Umgebungsmodellierung auftreten können, erfolgen die ersten Durchläufe in einem Probetrieb mit deutlich verringerter Geschwindigkeit. Der Probelauf erfolgt in einer Dauerschleife und kann vom Anwender jederzeit abgebrochen werden. Allerdings ist zumindest ein kompletter Probelauf von Nöten, um den eigentlichen Betrieb starten zu können.

Für das gesamte Konzept wird der Umfang von textuellen Eingaben auf ein Minimum reduziert wird, da hier, unter anderem auf Grund der fehlenden haptischen Rückmeldung, eine der großen Schwächen der Touch-Interaktion zu finden ist (Haslbeck, Pecot, Popova, Eichinger, & Bengler, 2011; Haslbeck et al., 2011).

7.2 Versuchsdesign

Im Anschluss an die Konzeption erfolgte eine erste Evaluation des dargestellten GUI-Konzepts im Zusammenhang mit einer Systemergonomie-Expertengruppe, um die Qualität der späteren Ergebnisse aus der Probandenstudie zu verfeinern. Hierbei stand vor allem die Einhaltung von ergonomischen Heuristiken aus dem Bereich der Mensch-Maschine-Interaktion im Zusammenhang mit Touch-Displays im Vordergrund (siehe Kapitel 3.3.4). Dazu zählen zum Beispiel die Beachtung von

minimalen Größen oder die optische Gestaltung von Bedienelementen. Darüber hinaus wurde die logische Struktur des Programms anhand eines gedachten Einsatzszenarios evaluiert.

Neben der Expertenevaluation ist die Absicherung mit Hilfe einer Probandengruppe, die der Nutzerklientel in den Fähigkeiten entspricht, ein wichtiger Faktor. Hierbei steht die Einhaltung der Zielkriterien in Bezug auf die Usability und die erforderliche Zeit im Vordergrund. Die Ergebnisse einer solchen Studie stellen wichtige Ergebnisse für eine weitere Optimierung der grafischen Benutzeroberfläche dar. Die Untersuchung der konzeptionellen GUI erfolgt dabei abstrakt, da die finale Roboterzelle zum Zeitpunkt der Untersuchung noch nicht zur Verfügung stand. Besonders betroffen sind hiervon die Positionierung des Roboters und die Parametrisierung der Handhabungsbereiche bei der Prozessprogrammierung. Um sicherzugehen, dass die Nutzerklientel auf Basis der dargebotenen Informationen zur Aufgabenerfüllung die Notwendigkeit einer genauen Positionierung verstanden hat und den Roboter im realen Anwendungsfall richtig positionieren würden, kam ein Miniaturmodell eines Knickarm-Roboters zum Einsatz. Dieser musste auf einem dargestellten Arbeitsbereich so positioniert werden, dass der Roboter auch in den Randbereichen noch Bauteile greifen kann und diese nicht nur erreicht.

Die Akzeptanz der Probanden wird über einen PSSUQ-Fragebogen (siehe Kapitel 3.3.3.2) bewertet. Um die Ergebnisse dieser Befragung sinnvoll beurteilen zu können, werden die Werte mit den von Sauro und Lewis (2012) postulierten Mittelwerten des PSSUQ verglichen. Sie stellen in ihrer Arbeit dar, dass der Mittelpunkt der Skala, die von Wert 1 bis 7 reicht, nicht beim Wert 4 zu finden ist, sondern über alle Items hinweg niedriger liegt. Um nicht unrealistisch gute Ergebnisse innerhalb dieser Studie zu erzielen, werden dementsprechend die Werte von Sauro und Lewis (ebd.) herangezogen.

Auf Basis der verfügbaren Daten wird für den Vergleich der Ergebnisse aus der Probandenstudie mit den postulierten Werten auf eine Analyse der Mittelwerte und der Konfidenzintervalle zurückgegriffen. Anhand einer Analyse auf Überschneidungen der Konfidenzintervalle kann auf statistisch signifikante Unterschiede geschlossen werden. Abbildung 40 beschreibt die möglichen Ergebnisvarianten der Konfidenzintervallüberlappungen.

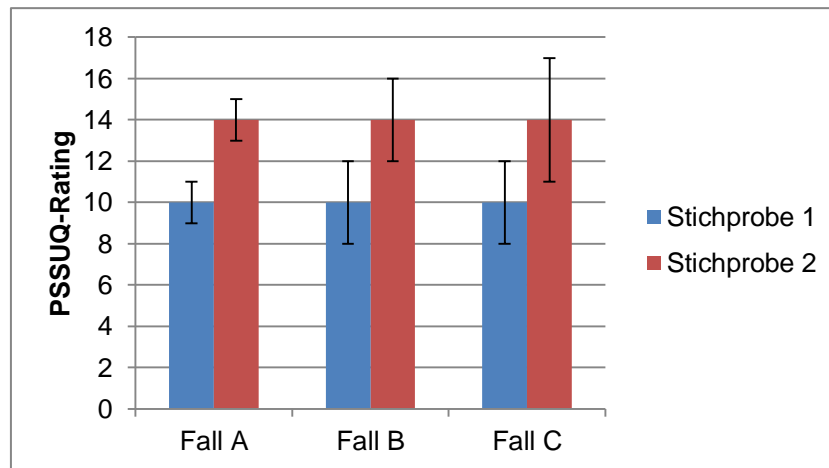


Abbildung 40: Fallbeispiele der Konfidenzintervallüberlappungen

Überschneiden sich die Konfidenzintervalle zweier Stichproben nicht (wie in Fallbeispiel A), so kann nach (Cumming & Finch, 2005; Field, 2013) ein signifikanter Unterschied zwischen den beiden Stichproben angenommen werden ($p < 0,01$). Berühren sich die Konfidenzintervalle, wie in Fallbeispiel B dargestellt, so kann ebenso von einem signifikanten Unterschied ausgegangen werden ($p = 0,01$). Schwierig wird erst die Einordnung bei Überlappung der Konfidenzintervalle. Hier wird häufig angenommen, dass eine Überlappung bereits einen nicht-signifikanten Unterschied beschreibt, allerdings ist dies nicht generell der Fall und muss mit Hilfe der mittleren Fehlergrenze untersucht werden. Die Fehlergrenze hat dabei den Wert des halben Konfidenzintervalls. Für Fall A und Stichprobe 1 würde das Konfidenzintervall also vom unteren Wert 9 bis zur Obergrenze 11 reichen und es ergäbe sich eine Fehlergrenze von eins. Entscheidendes Kriterium für die Beurteilung eines signifikanten Unterschiedes zweier Stichproben ist nun das Verhältnis der mittleren Fehlergrenze zur Größe der Überlappung der Konfidenzintervalle beider Stichproben. Ist der Bereich der Überlappung kleiner als die mittlere Fehlergrenze kann weiterhin von einem signifikanten Unterschied ausgegangen werden ($p < 0,05$). (Cumming & Finch, 2005; Field, 2013)

Für die Evaluation der GUI wird folgende Hypothese aufgestellt:

Hypothese: Die Ergebnisse der neu entwickelten GUI werden die Durchschnittswerte nach Sauro und Lewis (2012) für eine PSSUQ-Befragung signifikant übertreffen.

Weitere Ziele der Untersuchung sind die Ableitung von Optimierungspotenzialen und eine Abschätzung der benötigten Zeit. Hierfür kommt ein DIKABLIS-System zum Einsatz. Die mit dem DIKABLIS-System zu erhebenden Daten sind relevant für die Überprüfung der korrekten Anordnung der Bedienelemente. Häufig auftretende Probleme bei der Suche nach Buttons oder Informationen auf der Bedienoberfläche sprächen für eine ungenügende Anordnung der einzelnen Elemente. Des Weiteren kann das DIKABLIS-System auch bei der Analyse von Bedienfehlern helfen, da alle Bedienhandlungen des Probanden direkt beobachtbar sind. Zusätzlich wird dieser Prozess mit der Methode „Lautes Denken“ (Holzinger, 2005; Shneiderman & Plaisant, 2010) unterstützt. Der Proband wird also dazu aufgefordert, seine Gedankengänge während der Bearbeitung verbal auszudrücken. Die Einbindung dieser Methode reduziert die Qualität der Bewertung der benötigten Zeit für die Gesamtaufgabe auf eine Abschätzung.

7.2.1 Versuchsaufbau

Die Probandenstudie erfolgte, bis auf die Einbindung eines Miniaturmodells eines Knickarm-Roboters, nur durch Interaktion mit der GUI (siehe Abbildung 41).

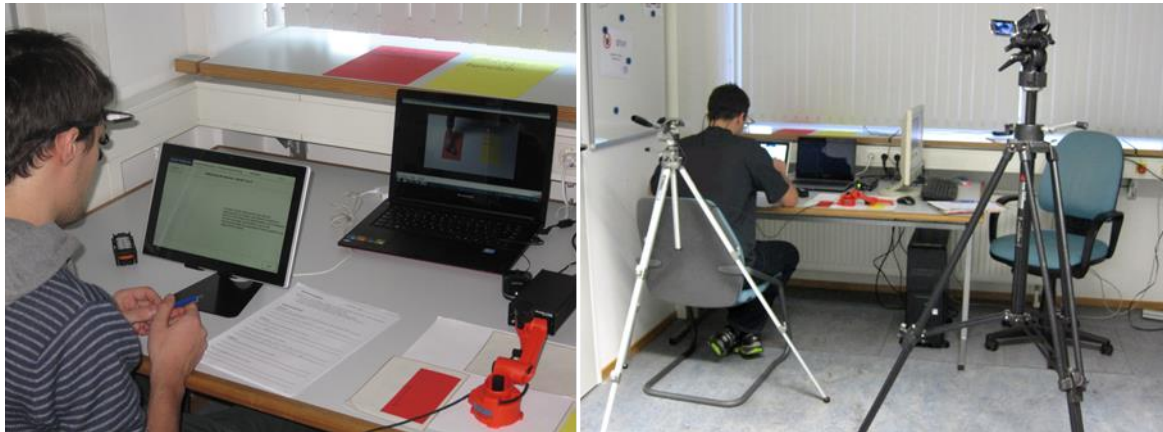


Abbildung 41: Versuchsaufbau

Die Bearbeitung des Konfigurier- und Programmierprozesses wurde dabei durch eine Arbeitsanweisung geführt, die auf Basis einer systemergonomischen Analyse des Gesamtprozesses bzw. einer Tätigkeitsanalyse entwickelt wurde. Die ausgedruckte Arbeitsanweisung wurde in kleinere sinnvoll abgegrenzte Teilaufgaben gegliedert, die den Probanden nach und nach durch das Programm bzw. die Bearbeitung des Gesamtprozesses führten. Diese festen logischen Teilaufgaben reduzieren die mentale Beanspruchung für den Probanden und verbessern die Gewinnung von Daten. So kann nach jedem logischen Handlungsschritt das Verständnis des Probanden bezüglich der Notwendigkeit sowie Bearbeitung des Handlungsschrittes für die Programmierung mit Hilfe eines Fragebogens untersucht werden. Für die Konfigurierprozesse, die zum überwiegenden Teil nur abstrahiert durchgeführt werden konnten, standen dabei vor allem Verständnisfragen im Vordergrund. Diese sollten Aufschluss darüber geben, ob sich der Proband der Notwendigkeit der einzelnen Schritte und deren Ziele bewusst war.

Eine Ausnahme bildete die Konfiguration des im abstrahierten Prozess zu verwendenden Werkstücks, die, bis auf die Ablage des Bauteils, gut nachgebildet werden konnte. Hier stand vor allem die Überprüfung der Vollständigkeit der 3D-Modellierung im Vordergrund. Um die Qualität der Validierung zu überprüfen, wurde das Bauteil sowohl vollständig als auch unvollständig für den Versuch modelliert (siehe Abbildung 42). Der Nutzer erhielt beim ersten Scan die Ansicht des

unvollständigen Bauteils und war dementsprechend dazu angehalten, den Scan des Bauteils erneut durchzuführen. Zur Überprüfung des Modells war es notwendig das Bauteil innerhalb des 3D-Fensters zu drehen, womit auch eine Überprüfung der korrekten Bedienung der 3D-Oberfläche, inklusive der verwendeten Gesten, getestet werden konnte.

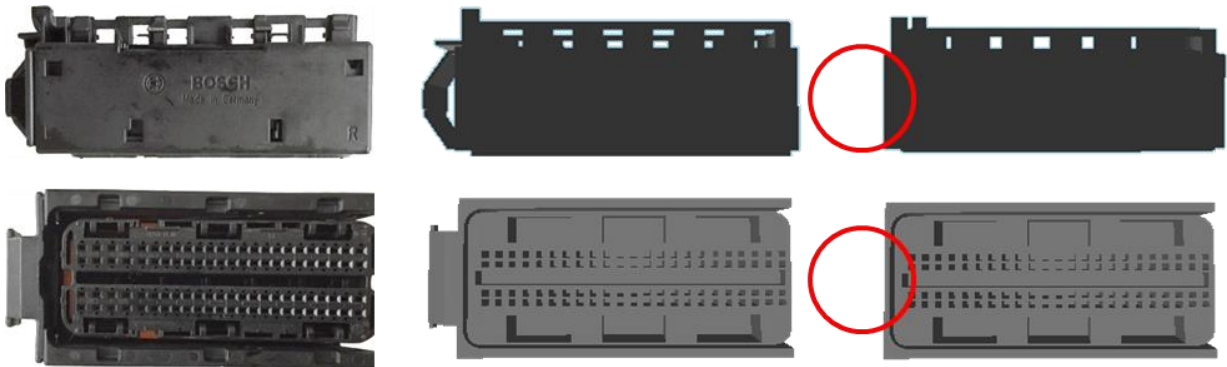


Abbildung 42: Vergleich des realen Werkstücks (links) mit der vollständigen bzw. unvollständigen CAD-Modellierung (rechts)

Überdies wurde der Proband gebeten die vom System berechneten Griffe für das Bauteil zu überprüfen. Hierbei musste der Proband darauf achten, dass keine instabilen Griffe verwendet würden, die entweder zum Entgleiten oder zu einer Beschädigung des Bauteils führen könnten.

Die Darstellung des zu programmierenden Prozesses erfolgte über ein Video, das auf einem separaten Laptop dargestellt wird. Mit dieser Methodik sollte eine Überprüfung der Validität der Ergebnisse aus der Studie zum mentalen Modell ermöglicht werden. Der Ablauf des Prozesses war dabei sehr einfach gehalten und stellte eine einfache Pick & Place-Aufgabe mit einer kurzen Haltezeit dar, die in einem realen Prozess zum Beispiel für einen Scan-Vorgang genutzt werden kann. Zur Unterstützung der Aufgabe stand wiederum der modellierte Arbeitsplatz (inklusive Robotermodell) zur Verfügung, der dieselbe farbliche Codierung aufwies, wie die im Video dargestellten Bereiche.

Die Aufnahme der Blickdaten der Probanden erfolgte über die Einbeziehung eines DIKABLIS-Blickerfassungssystems. Unterstützt wurde das System weiterhin durch eine Videokamera, die die Bedienaktionen und damit sowohl Bedienfehler als auch benötigte Zeit redundant aufnahm.

7.2.2 Versuchsablauf

Zu Beginn der Untersuchung erfolgte die Bearbeitung eines Fragebogens zur Erhebung demografischer Daten. Hierbei standen neben den Fragen zu Alter und Geschlecht vor allem Erfahrungen mit Touch-Geräten und Industrierobotern im Vordergrund. In einem weiteren Schritt wurden die Probanden über das abstrahierte Szenario aufgeklärt und es wurde ihnen Ziel und Zweck dieser Studie dargestellt. Anschließend wurde den Probanden der Fragebogen mit Versuchsaufgaben sowie Verständnisfragen ausgehändigt (siehe Anhang A.2). Nach Erledigung der einzelnen Programmabschnitte hatten die Probanden die Möglichkeit Verbesserungsvorschläge anzubringen. Zum Abschluss des Versuchs bewertet der Proband die Benutzeroberfläche mit Hilfe des PSSUQ und darin ausformulierte Bemerkungen wurden mit dem Versuchsleiter durchgesprochen, um Unklarheiten auszuräumen.

7.2.3 Probandenkollektiv

Als Probandenauswahl standen dieser Usability-Studie zwanzig Studenten und Mitarbeiter des Lehrstuhls für Ergonomie (siebzehn männl. und drei weibl.) zur Verfügung. Das Alter der Probanden erstreckte sich dabei von 19 – 33 Jahren, womit sich ein Durchschnittsalter von 26,3 Jahren mit einer Standardabweichung von 3,8 Jahren ergab. Die Probandenklientel verfügte dabei zum Teil bereits über Erfahrungen mit Robotern, allerdings beschränkte sich diese maximal auf die Programmierung einfacher Trajektorien mit Hilfe von Bedienhandgeräten. Insofern kann in diesem Zusammenhang nicht von Systemexpertise, insbesondere nicht in Zusammenhang mit der thematisierten Programmiermethode *aufgabenorientierte Programmierung*, gesprochen werden.

Die Wahl von Studenten und wissenschaftlichen Mitarbeitern als Repräsentation der Nutzerklientel wird als valide angenommen, da sie ähnliche Fähigkeiten aufweisen. Gleichzeitig kann angenommen werden, dass im ungünstigsten Falle auch Ferienarbeiter und Werksstudenten für die Programmierung eines solchen Systems in Frage kommen. Insofern kann, selbst bei Ablehnung der Annahme vergleichbarer Fähigkeiten, diese Probandengruppe als eine Art Worst-Case-Szenario für die Probandengruppe gesehen werden.

7.3 Ergebnisse

Der zentrale Punkt bei der Bewertung der grafischen Bedienoberfläche ist die korrekte Aufgabenerfüllung innerhalb des in den Forschungszielen definierten Zeitlimits sowie die Akzeptanz der Nutzklientel. Dementsprechend sind für die Beurteilung der konzeptionellen GUI vor allem die subjektiven Bewertungen der Probanden, die Zahl und Art der Fehler sowie die benötigte Zeit für die Durchführung der einzelnen Prozessschritte wesentlich. Die Einhaltung des Zeitlimits von 48h ist allerdings nur im Rahmen einer Gesamtevaluation unter realen Bedingungen vollständig überprüfbar, da viele Aspekte bei einer virtuellen Absicherung unberücksichtigt bleiben und der zeitliche Aufwand zu hoch für übliche Probandenversuche ist. Dementsprechend können die hier dargestellten Werte nur als Abschätzung des reinen Programmieraufwandes betrachtet werden. Dabei konnte der abstrahierte und vereinfachte Konfigurier- und Programmierprozess in einer mittleren Zeit von knapp 23 Minuten durchgeführt werden (siehe Abbildung 43).

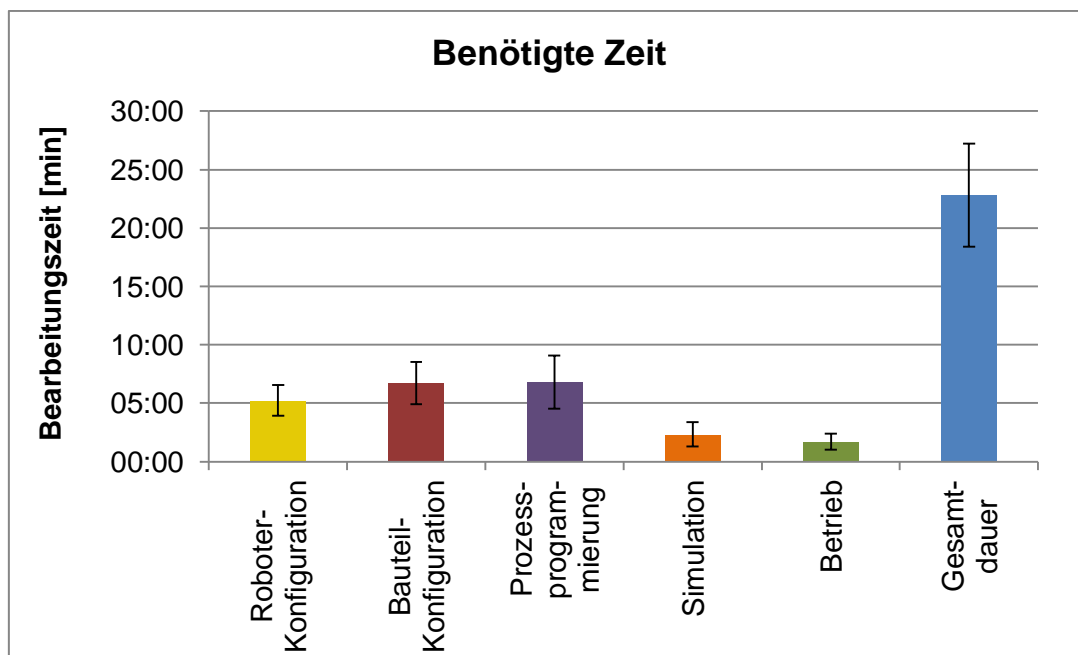


Abbildung 43: Benötigte Zeit zur Aufgabenerfüllung (Fehlerbalken stellen die Standardabweichung dar)

Den Ergebnissen folgend, kann also angenommen werden, dass der zeitliche Aufwand bei der Verwendung der grafischen Benutzerschnittstelle nur einen untergeordneten Einfluss auf die Gesamtdauer des Programmier- und Konfigurierprozess hat.

Für die Beurteilung der Aufgabenerfüllung ist insbesondere die programmtechnische Umsetzung der dargestellten Pick & Place-Aufgabe durch den Probanden relevant. Hierbei steht die Evaluation der Prozessprogrammierung, inklusive der Umsetzung der Ergebnisse zum mentalen Modell, im Vordergrund, die einen wichtigen Indikator für die Effektivität im Umgang mit der GUI darstellt. Abbildung 44 beschreibt die, für diese Fragestellung wichtigen, Resultate.

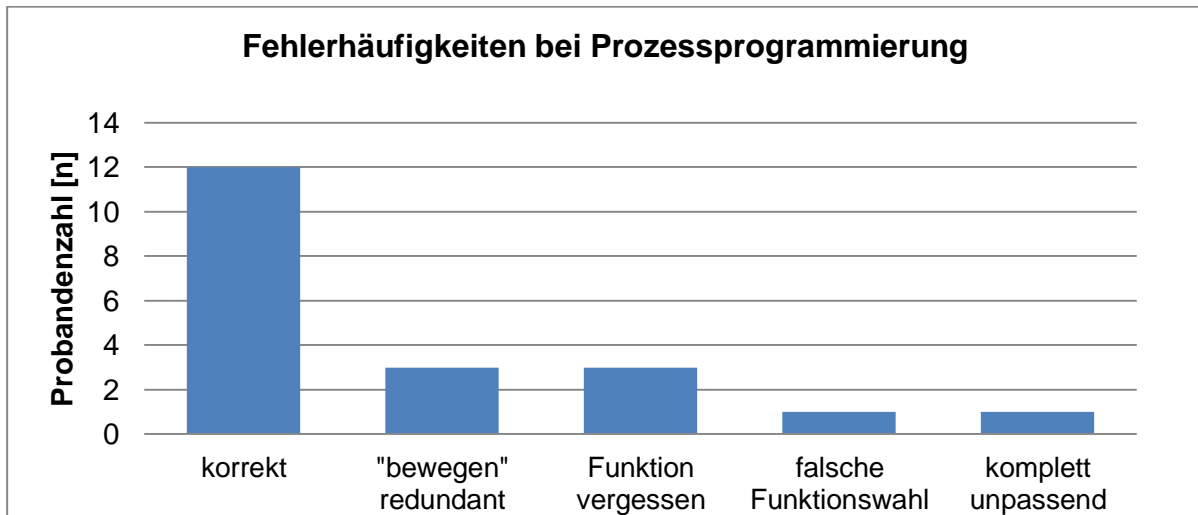


Abbildung 44: Fehlerhäufigkeiten bei der Prozessprogrammierung

Wie in Abbildung 44 dargestellt, waren 12 der 20 Probanden in der Lage den einfachen, per Video gezeigten, Pick & Place- Arbeitsprozess korrekt umzusetzen. Drei weitere Probanden fügten zusätzliche Bewegungsschritte ein, die als redundant betrachtet werden können. Diese verhindern die korrekte Ausführung der Aufgabe nicht, führen aber im Zweifel zu einer Verschlechterung der Systemeffizienz. Da als Teil der Funktionsbeschreibung „Bewegen“ explizit darauf hingewiesen wurde, dass diese Funktion nur als Hilfsfunktion konzipiert ist, weist darauf hin, dass die Erklärungstexte nicht in Gänze korrekt erfasst wurden. Bei drei der in Abbildung 44 dargestellten Probanden, die nicht in der Lage waren die Aufgabe erfolgreich zu programmieren, wurden Funktionen vergessen. In zwei Fällen fehlte der Operator „Warten“ und in einem weiteren Fall das Ablegen des Bauteils zum Ende des Prozesses. Ein Proband hat den Operator „Greifen“ durch „Entnehmen“ ersetzt und ein Ergebnis stand in keinem nachvollziehbaren Verhältnis zur gestellten Aufgabe. Somit kann bei 15 von 20 Personen davon ausgegangen werden, dass die Aufgabe

im Anschluss an den Programmzyklus, bei erfolgreicher Konfiguration der restlichen Prozessschritte, zumindest erfolgreich ausgeführt worden wäre.

Bezüglich der subjektiven Bewertung des GUI-Konzepts steht die Auswertung der Akzeptanz bzw. der subjektiven Einschätzung des GUI-Konzepts im Vordergrund. Diese wurde, wie beschrieben, mit Hilfe des PSSUQ erhoben. Das Ergebnis und die Gegenüberstellung zu den Skalenmittelpunkten nach Sauro und Lewis (2012) ist in Abbildung 45 dargestellt.

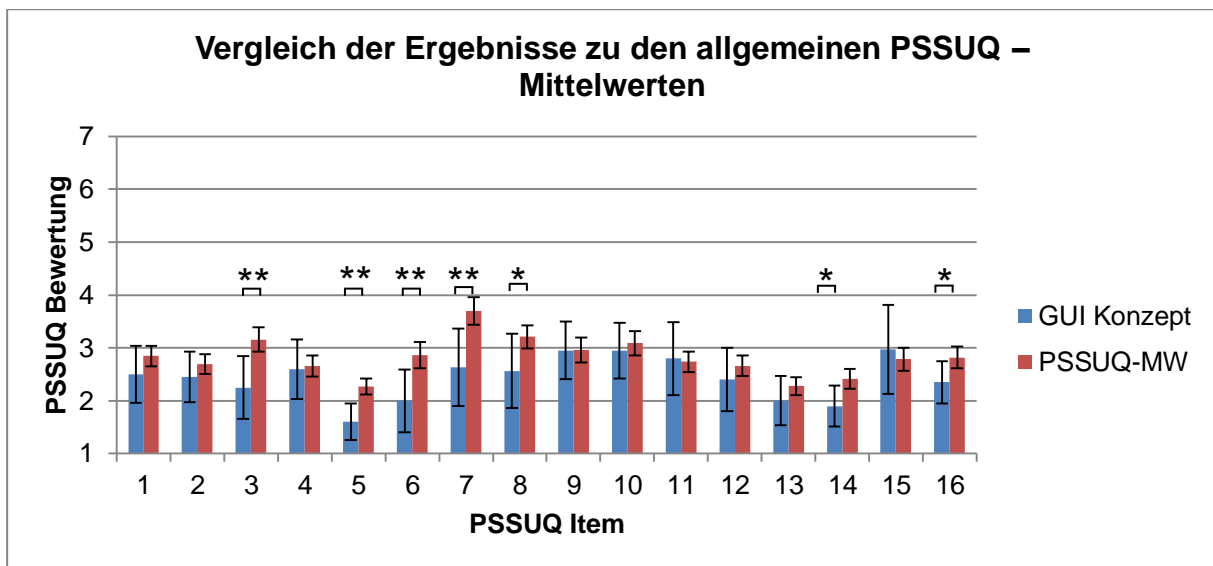


Abbildung 45: Vergleich des PSSUQ-Ratings des GUI und der von Sauro und Lewis (2012) erhobenen Mittelwerten des Fragebogens (Fehlerbalken stellen das 95 % Konfidenzintervall dar)

Dabei lässt sich ablesen, dass ein Großteil der PSSUQ-Items für die konzeptionelle GUI tendenziell überdurchschnittlich bewertet wurde. Signifikante Unterschiede finden sich mit den Items 3 sowie 5-8 vor allem im Bereich der Zweckmäßigkeit des Systems (Frage 1-6) sowie der *Informationsqualität* (Frage 7-12). Die Items 3 sowie 5-7 weisen dabei einen Signifikanzwert von $p < 0,01$ auf. Für die weiteren Items lassen sich signifikante Unterschiede für die Fragen 8, 14 und 16 mit $p < 0,05$ nachweisen.

Beim Vergleich der vier PSSUQ-Interpretationskategorien *Zweckmäßigkeit des Systems*, *Informationsqualität*, *Interfacequalität* und *Gesamtbewertung* verstärkt sich dieses Bild zusätzlich. Dabei zeigen sich signifikanten Unterschiede für alle Kategorien mit Ausnahme der *Interfacequalität*. Insofern kann Hypothese 1 für drei der vier Kategorien bestätigt werden. Da die *Interfacequalität* darüber hinaus die

verwendete Hardware bewertet, kann das Ergebnis als positiv für das GUI-Konzept gewertet werden (Abbildung 46).

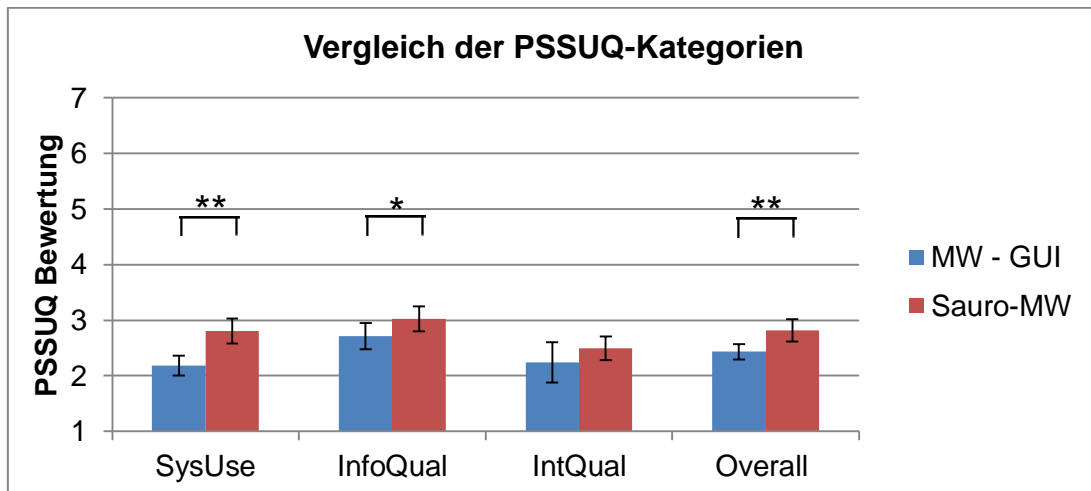


Abbildung 46: Vergleich der PSSUQ-Kategorien zwischen der Probandenstudie und den PSSUQ-Mittelwerten (Fehlerbalken stellen das 95 % Konfidenzintervall dar)

Um eine weitere Verbesserung der grafischen Bedienoberfläche zu erzielen, ist es wichtig, die auftretenden Probleme und die Rückmeldungen der Probanden zu analysieren. Eine wichtige Quelle zur Aufdeckung von Designfehlern und Inkonsistenzen waren der selbstgestaltete Fragebogen und die Methode „Lauter Denken“. Die Auswertung der Anmerkungen beider Methoden wurde zusammengeführt und bezüglich der Benennungshäufigkeit hin untersucht. Ergänzt wurden sie zusätzlich noch um offensichtliche Beobachtungen des Versuchsleiters bzw. bei der Videoanalyse im Zuge der Auswertung. Beispielhaft für die Ergebnisse werden in Tabelle 2 die konzeptübergreifenden Nennungen dargestellt.

Tabelle 2: Darstellung der bereichsübergreifenden Nennungen (Egger, 2013)

	Anz. VPs	%
Allgemein		
Konsistenz bei Anleitungs-Button	2	10
Automatisches Speichern gewünscht	1	5
Feedback über Hardware des Roboters wäre gut	1	5
Reiterpfeile oben gewünscht	1	5
Hauptschritte fortlaufend nummerieren	1	5
Begriffe		
Begriff <i>Konfiguration</i> nicht vertraut	5	25
Begriffe <i>Template</i> und <i>Black-Box</i> zu fachspezifisch	1	5
Zu spezifische Begriffe	1	5
Begriff <i>Funktionsablaufdiagramm</i> unwichtig	1	5
Texte		
Zu lange Texte in Anleitung	3	15
Zu viel Text	1	5
Visuell den Text layouten	1	5
Texte in Stichpunkten, aber Anleitung bei Bedarf	1	5
Tastatur		
Tastatur sollte mit <i>Enter</i> schließen	5	25
Tastatur sollte bei Klick in das Feld öffnen	4	20
Eingabe über Tastatur anzeigen / Tastatur unter Feld	3	15
x bei Tastatur zu klein	2	10
größere Tastatur	1	5
Beobachtung		
VP versuchte Tastatur mit <i>Enter</i> zu schließen	9	45

In allen Fällen wurden die Nennungen zur einfacheren Analyse in logische Kategorien aufgeteilt und auf ihre Plausibilität bzw. Anwendbarkeit hin untersucht. Im betrachteten Fall der allgemeinen Nennungen standen vor allem Probleme mit der Implementierung der virtuellen Tastatur im Vordergrund. Darüber hinaus wurde von einigen Probanden der Umfang der Anleitungstexte als auch das verwendete Vokabular bemängelt. Diese Verbesserungsvorschläge und Kritikpunkte sind Grundlage für einen Optimierungsprozess der Oberfläche.

Nennungen, die auf eine nicht optimale Positionierung von Eingabeelementen hinweisen, wurden mit Hilfe des Blickerfassungssystems DIKABLIS weiter untersucht. Vor allem zwei Auffälligkeiten traten dabei während der Untersuchung auf. Abbildung 47 stellt die Suchstrategien dreier Probanden dar, die bei der Suche nach dem Button für die Hilfe-Funktion des 3D-Fensters (rot eingerahmt) nicht erfolgreich waren.

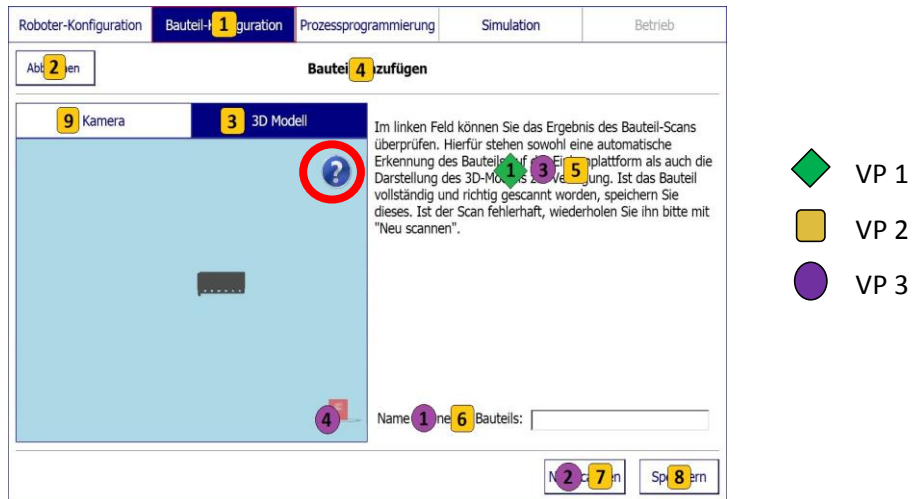


Abbildung 47: Blickstrategien unterschiedlicher Probanden (Egger, 2013)

Dabei lässt sich auf Basis der Blickmuster allerdings keine interindividuell konsistente Strategie beobachten. Im anderen Fall bezieht sich die Problematik auf die Anordnung der Bedienelemente im Bauteil-Konfigurationsfenster (siehe Abbildung 48).

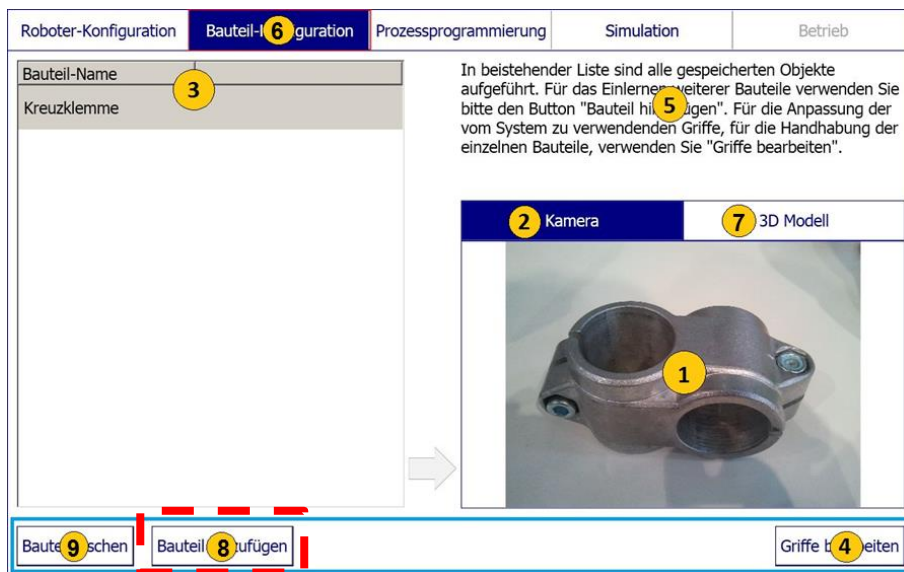


Abbildung 48: Darstellung der gemittelten Blickfolgen für die Bauteilkonfiguration (Egger, 2013)

Ziel dieser Bedienoberfläche ist die Bauteilkonfiguration, bei der es erforderlich sein kann, ein Bauteil zu scannen und damit der Bauteilliste hinzuzufügen. Der Button für diese Aufgabe befindet sich unterhalb der Bauteilliste (rot markiert). Diese Position wird verwendet, um eine logische örtliche Verknüpfung mit der zu bearbeitenden Bauteilliste zu schaffen. Allerdings stellte sich bei Betrachtung der DIKABLIS-Daten

heraus, dass im Mittel sieben Sakkaden (schnelle Blickwechsel) erfolgen, bevor der dazugehörige Button entdeckt wird. 60% der Probanden begannen die Suche mit einem Blick auf die grafische Darstellung des aktuell ausgewählten Bauteils. Eine Überarbeitung dieser Ansicht scheint also sinnvoll und notwendig.

7.4 Diskussion und Optimierungsbedarf

Der Vergleich der PSSUQ-Bewertung mit den von Sauro und Lewis (2012) ermittelten Skalenmittelpunkten ist ungewöhnlich. Der hier dargestellte Versuch ist aber so konzipiert, dass er neben einer Bewertung dieses Konzepts auch als Basis für einen Vergleich mit den Ergebnissen einer Realstudie, im industriellen Umfeld bei realen Einsatzbedingungen, dienen kann. Für diesen Vergleich ist der PSSUQ aus Sicht des Autors insbesondere auch auf Basis der hohen Reliabilität (Bangor, Kortum, & Miller, 2008) sehr gut geeignet.

Bei der Auswertung des PSSUQ zeigt sich ein positives Bild für das GUI-Konzept. Gestützt wird die positive subjektive Einschätzung der Probanden auch durch die benötigte Zeit für die Aufgabenerfüllung. Die Konfiguration des zu verwendenden Bauteils sowie die Programmierung und Überprüfung des Prozesses liegen bei dem hier dargestellten Versuchsdesigns im Mittelwert deutlich unterhalb von 30 Minuten. Dabei bleibt zu berücksichtigen, dass der zu programmierende Prozess in seiner Ausprägung einfach war, also bei höherer Komplexität durchaus davon auszugehen ist, dass der Zeitanteil für die Prozessprogrammierung steigt. Auf Grund der starken Diskrepanz zwischen zur Verfügung stehender und in diesem Versuch benötigter Zeit kann bisher allerdings dennoch von einer Zielerreichung ausgegangen werden.

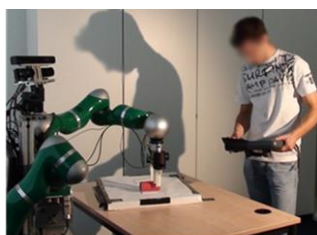
Bezüglich der Umsetzung der Ergebnisse aus der Untersuchung zum mentalen Modell kann angenommen werden, dass diese anwendbar ist. Bereits bei der ersten Nutzung konnte, bei vorher unbekannter Aufgabe und GUI, der Prozess von 15 der 20 Probanden angemessen umgesetzt werden. Darüber hinaus kann das Auftreten der Auslassungsfehler durch die zeitliche Verzögerung zwischen Video-Darstellung der Arbeitsaufgabe und der tatsächlichen Umsetzung aufgetreten sein. Dazu ist wahrscheinlich, dass, bis auf das Auslassen des „Ablegen“-Schritts, keiner der Fehler schädigende Auswirkungen auf das System gehabt hätte. Um allerdings auch diesen Fall abzusichern, muss eine Plausibilitätsschranke eingefügt werden.

8 Wahl einer geeigneten Eingabetechnologie

Neben der Konzeptionierung und Evaluierung der grafischen Bedienschnittstelle muss eine Eingabetechnologie gefunden werden, die für das Interaktionsparadigma geeignet ist. Basierend auf den in Kapitel 5 definierten Anforderungen des Interaktionsparadigmas wird das Eingabekonzept vor allem für die Definition einfacher Handhabungsorte ausgelegt. Auf Grund der Rückfallebene Teach-In bzw. für die Einbeziehung von Black-Box-Prozessen muss das ausgewählte Konzept aber auch die Definition von exakten Koordinaten und Trajektorien erlauben. Dementsprechend werden im Weiteren nur Konzepte untersucht, die das Verfahren des Roboters in sechs Dimensionen erlauben.

8.1 Versuchsdesign

Als Grundlage für die weiteren Untersuchungen muss eine Vorauswahl an wirtschaftlichen und prozesssicheren Technologien vorgenommen werden. Auf Grund der in Kapitel 4.2.4 und 4.2.5 dargestellten Nachteile von Trackingmethoden und Sprachsteuerung werden diese für die weiteren Untersuchungen nicht berücksichtigt. Von den übrigen, der in Kapitel 4.2 dargestellten, Klassifizierungen kommen Vertreter in der dargestellten Studie zur Anwendung. Auf Basis der Anforderung einer flexiblen Roboterzelle inklusive wirtschaftlicher Aspekte wurden dementsprechend folgende vier Konzepte als praktikabel und prozesssicher identifiziert (Abbildung 49).



Bedienhandgerät



6D – Maus am Greifer



6D – Maus



GC-Modus

Abbildung 49: Betrachtete Interaktionskonzepte

Wie in Kapitel 4.2.1 dargelegt, stellt das Bedienhandgerät die am häufigsten verwendete Eingabevariante für die Bedienung eines Roboters dar und eignet sich damit sehr gut als Referenz für die weiteren Eingabetechnologien. Das BHG wird im Rahmen dieser Untersuchung ausschließlich zur Steuerung der TCP-Position (im Werkzeugkoordinatensystem), und nicht der einzelnen Gelenkwinkel, verwendet. Diese Einschränkung reduziert die Variabilität bei der Anpassung der Roboterposition, erlaubt aber einen konsistenteren Vergleich mit den übrigen Konzepten. Darüber hinaus stellt die Steuerung des TCP nach Nof (1999) die nutzerfreundlichere Methode dar.

Neben dem Bedienhandgerät werden zwei 6D-Maus-Konzepte in die Studie mit einbezogen. Das erste 6D-Maus-Konzept verwendet die bereits in Abbildung 13 dargestellte *SpaceMouse Pro* der Firma 3Dconnexion⁹ die üblicherweise vor allem für CAD-Aufgaben angewendet wird. Die Position der Maus ist dem Nutzer freigestellt und ermöglicht bei korrekter Orientierung eine Vereinbarung der Bewegungsrichtungen des Roboters mit den Eingaben der Maus, um eine sonst notwendige mentale Transformation zwischen unterschiedlichen Koordinatensystemen zu vermeiden. Das andere 6D-Maus-Konzept sieht die Anbringung der Maus direkt an den Greifer vor, um so die Bewegungsrichtung des Roboter-TCP direkt an die Bewegungsrichtung der Bedienung zu knüpfen. Mit diesem Konzept soll eine direktere Manipulation des Greifers untersucht werden. Dem Nutzer wird erlaubt in direkter Nähe zum Handhabungsbereich zu arbeiten, was vor allem bei genauer Positionierung des Roboters ohne Ablagemöglichkeit für eine konventionellen 6D-Maus sinnvoll sein kann. Hierfür wird die 6D-Maus *SpaceNavigator* von 3Dconnexion verwendet. Auf Grund der starken technischen Ähnlichkeiten beider Eingabeelemente können Unterschiede bei der Bearbeitung der Aufgaben auf Grund der Maus-Charakteristik ausgeschlossen werden. Das letzte Konzept sieht die Verwendung einer Drehmomenten-Steuerung, in diesem Fall erzielt durch den *Gravity Compensation-Modus* des KUKA LBR4¹⁰, vor.

Die Untersuchung der unterschiedlichen Bedienkonzepte beinhaltet die Definition eines Handhabungsbereichs sowie die Programmierung einer Trajektorie. Die Definition des Handhabungsbereichs entspricht dabei der üblichen Nutzungssituation

⁹ www.3dconnexion.de

¹⁰ www.kuka.de

des Interaktionskonzepts, während das Einlernen einer Trajektorie der Absicherung des Back-up-Szenarios, sprich Programmieren ohne Sensorunterstützung, gilt.

Als objektive Messgröße wird die benötigte Zeit erfasst. Für den Vergleich der subjektiven Bewertungen wird ein Fragebogen verwendet, der folgende Bereiche umfasst:

- Allgemeine Fragen zu Alter, Geschlecht und Erfahrungen mit 6D-Manipulationen
- Einschätzungen zur Eignung der Eingabekonzepte für den jeweiligen Aufgabentypen
- Nennung des besten und schlechtesten Konzepts mit zusätzlicher Nennung von Verbesserungsvorschlägen

Weiterhin wird ein, um Fragen zur Bedienoberfläche gekürzter, PSSUQ-Fragebogen (Lewis, 2002) für jede der getesteten Eingabetechnologien herangezogen (siehe Anhang A.3). Basierend auf diesen Messgrößen wurden folgende drei Hypothesen zur Auswertung aufgestellt:

Hypothese 1a: Die Drehmomenten-Steuerung (*GC-Modus*) erhält die beste subjektive Bewertung.

Hypothese 1b: Der *GC-Modus* erlaubt eine höhere Effizienz bei der Eingabe von Geometriedaten als die übrigen Interaktionstechnologien.

Hypothese 2: Das Konzept *3D-Maus am Greifer* stellt sowohl in Bezug auf die subjektiven als auch objektiven Bewertungen das zweitbeste Konzept dar.

Auf Grund der direkten Manipulation des Roboters bzw. dessen TCP bei der Drehmomenten-Steuerung scheint es sehr wahrscheinlich, dass sie sowohl bezüglich der objektiven Kennwerte aber auch der subjektiven Rückmeldung der Probanden die besten Ergebnisse erzielt. Da das Konzept *3D-Maus am Greifer* ebenfalls keinerlei mentale Transformation des Bewegungskoordinatensystems zum Bedienkoordinatensystem benötigt und die Manipulation des Greifers direkt am Wirkort erlaubt, wird davon ausgegangen, dass dieses System einen guten Ersatz zur Drehmomenten-Steuerung darstellen könne.

8.1.1 Versuchsaufbau

Durch den experimentellen Charakter der Konzepte kam kein realer Handarbeitsplatz zur Anwendung, da eine Beschädigung von Greifer und Arbeitsplatz nicht ausgeschlossen werden konnte und somit eine Fortführung der Untersuchung gefährdet war. Dementsprechend wurden die Interaktionskonzepte in einer Laborumgebung untersucht.

Auf Basis der Einbindung des *GC-Modus* des KUKA LBR wurde dieser Roboter auch für alle anderen Interaktionskonzepte herangezogen, um zusätzliche Einflussfaktoren durch das verwendete Robotersystem zu vermeiden. Durch diese Wahl wurde die Größe der Arbeitsfläche (Abbildung 50) für den Versuch geringfügig verkleinert, unterschreitet die als Mindestanforderung definierten 400 x 400 mm mit ca. 340 x 425 mm aber nur knapp. Durch den prototypischen Charakter der Eingabegeräte war die inverse Kinematik des Roboters in Randbereichen des Arbeitsraums nicht in der Lage die Gelenkwinkel des Roboters so zu wählen, dass der TCP die gewünschte Position erreicht. Dies hat in einem Vorversuch vor allem bei Nutzung der Maus-Konzepte dazu geführt, dass das System sich selbst in einen Schutzzustand versetzte und nicht mehr auf Eingaben reagierte. Um diesen Fehlerfall unter Versuchsbedingungen so weit wie möglich zu vermeiden, wurde die Größe des Arbeitsraumes auf angegebenes Maß verkleinert.

Für die Untersuchung der unterschiedlichen Eingabekonzepte wurden folgende Aufgaben für die Studie definiert:

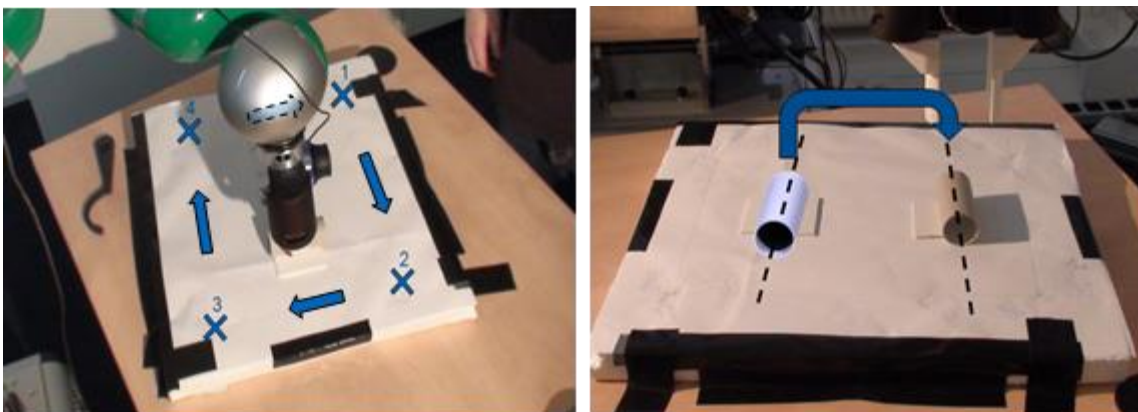


Abbildung 50: Aufgaben für den Interaktionsversuch (Aufgabe 1 links; Aufgabe 2 rechts)

Aufgabe 1 hat das Einlernen von vier vorgegebenen Punkten, also einem gedachten Handhabungsbereich, auf der Arbeitsfläche zum Ziel. Dafür musste der Proband jeden der Punkte mit der markierten Ecke des offenen Greifers anfahren und dann diese Position speichern. Die Reihenfolge der abzufahrenden Positionen war fest definiert.

In Aufgabe 2 wurde der Proband aufgefordert die abgebildete Rolle von einer Position aufzunehmen und passgenau auf die zweite Ablage zu abzulegen. Hierfür war eine minimale Anpassung der Orientierung um die z-Achse des Greifers notwendig. Die Ausführung aller Aufgaben erfolgte immer aus derselben Startposition.

Um Schäden am Greifer zu vermeiden fanden alle Untersuchungen auf einer Styroporplatte statt.

8.1.2 Versuchsablauf

Der Ablauf des Versuchs lässt sich wie folgt darstellen:

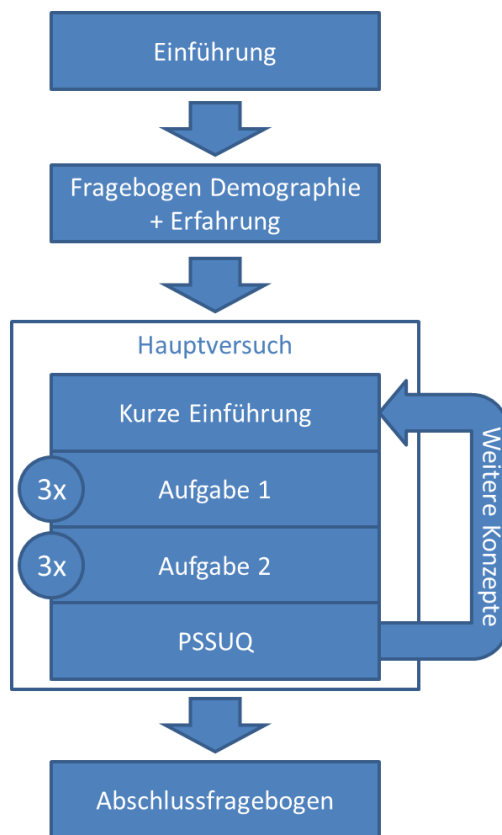


Abbildung 51: Versuchsablauf "Wahl einer geeignet Interaktionstechnologie"

Zu Beginn jedes Versuchsdurchlaufs wurden die Probanden in den Hintergrund und die Bedeutung der Studie eingeführt und darum gebeten einen Fragebogen zu demografischen Daten und der Erfahrung mit 6D-Eingabetechnologien zu beantworten.

Im Anschluss begann die praktische Versuchsdurchführung. Die Versuchsreihenfolge der Interaktionskonzepte wurde für das Probandenkollektiv randomisiert, damit Effekte durch das Wiederholen der Aufgaben minimiert werden konnten. Jedem Proband wurden beide Aufgaben erklärt und er erhielt die Anweisung sich ca. zwei Minuten mit der Bedienung vertraut zu machen. Im ersten Durchlauf erfolgte dann die Bewältigung von Aufgabe 1, also dem Anfahren und Speichern der vier definierten Positionen. Um einmalige Fehler oder Probleme mit der Technik abfangen zu können, wurde die Aufgabe zwei Mal wiederholt.

Nach Vollendung der ersten Aufgabe wurde ein weiteres Mal Aufgabe 2 erklärt und der Nutzer durfte erneut ein Beispiel analog zur eigentlichen Aufgabe als

Vorbereitung durchführen. Anschließend wurde auch Aufgabe 2 drei Mal durchgeführt.

Nach Vollendung der Durchläufe mit einem Interaktionskonzept wurden die Probanden dazu angehalten den vereinfachten PSSUQ -Fragebogen bezüglich der Eigenschaften des getesteten Systems auszufüllen. Nach Durchführung und Bewertung aller Konzepte wurden im Abschlussfragebogen die Eignung der Geräte, persönliche Präferenz und Verbesserungsmöglichkeiten für die persönliche Präferenz abgefragt.

8.1.3 Probandenkollektiv

Für den Versuch stellten sich 2 Frauen und 18 Männer mit einem Altersschnitt von 24,4 und einer Standardabweichung von 3,7 Jahren zur Verfügung. Die Zusammensetzung erfolgte mit Studenten, Auszubildenden und einem wissenschaftlichen Mitarbeiter der beteiligten Projektpartner. Vorherige Erfahrungen mit einem Robotersystem wiesen nur vier der beteiligten Probanden auf. Erfahrungen mit den verwendeten Eingabetechnologien war somit nur in geringem Maße vorhanden, wie Abbildung 52 zeigt.

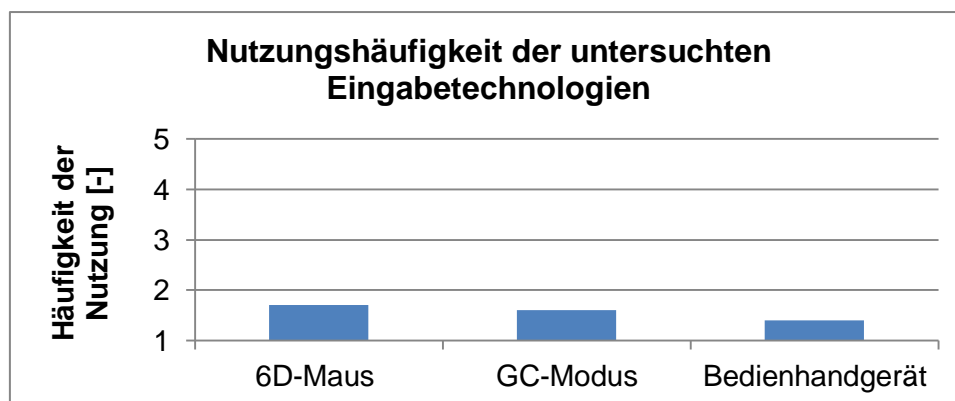


Abbildung 52: Durchschnittliche Häufigkeit der Nutzung der untersuchten Eingabetechnologien (1 - nie bis 5 - häufig)

8.2 Auswertung

Die Studie wurde durchgeführt, um die nach objektiven Gesichtspunkten aber auch bezüglich der subjektiven Kennwerte beste Eingabetechnologie für das in dieser Arbeit dargestellte Konzept einer transportablen Roboterzelle zu finden. Die unabhängige Variable der Studie ist somit die verwendete Eingabetechnologie. Das

folgende Unterkapitel betrachtet, getrennt nach subjektiven und objektiven Kennwerten, die Auswertung der abhängigen Variablen.

Die Verwendung der statistischen Verfahren basiert dabei auch in diesem Versuch auf den Werken von Field (2013) und Bortz et al. (2009). Alle dargestellten Fehlerbalken beschreiben die Standardabweichung.

8.2.1 Subjektive Daten

Um Hypothese 1a (siehe Kapitel 8.1) und den subjektiven Aspekt von Hypothese 2 belegen zu können wurden ein PSSUQ-Fragebogen (Version 2) und ein Fragebogen zur Einschätzung der Anwendbarkeit und Präferenz der Eingabegeräte herangezogen und ausgewertet.

8.2.1.1 PSSUQ

Auf Grund des Fehlens einer grafischen Benutzeroberfläche wurde der PSSUQ um Fragen reduziert, die nicht auf eine reine Hardware-Lösung bezogen werden können. Somit kamen noch die Fragen 1-8 (*Zweckmäßigkeit des Systems*) sowie die Fragen 18 und 19 zum Tragen.

In einem ersten Schritt wurde mit Hilfe des Kolmogorov-Smirnov-Tests die Normalverteilung der Probandenbewertungen untersucht. Da die Probanden eine verbundene Stichprobe darstellten und damit alle Interaktionskonzepte von der gleichen Probandengruppe evaluiert wurden, kam bei der Auswertung anschließend eine einfaktorielle ANOVA zur Anwendung. Um zu überprüfen, ob die Bedingung auf Sphärizität erfüllt wurde, kam der Mauchly-Test zur Anwendung. Wurde die Bedingung nicht erfüllt, wurden die Freiheitsgrade mit dem Greenhouse-Geisser-Korrekturfaktor korrigiert. Um signifikante Unterschiede zwischen den einzelnen Eingabekonzepten zum Vorschein zu bringen, wurde der „Bonferroni Post Hoc“-Test herangezogen. Für alle Analysen wurde dabei ein α -Fehler von $p < 0,05$ angenommen. Um die Ergebnisse übersichtlicher zu gestalten, wurden die Fragen 1-8 nach dem Schema von Lewis (2002) zur *Zweckmäßigkeit des Systems* zusammengefasst. Dabei zeigte die einfaktorielle ANOVA einen signifikanten Zusammenhang zwischen der Wahl des Eingabegerätes und der daraus resultierenden Eignung ($F(2,18; 41,416)=13,203$, $p < 0,001$, $\eta^2=0,410$, $1-\beta=1$). In diesem Fall konnte die Sphärizität allerdings nicht angenommen werden und

dementsprechend mussten die Freiheitsgrade mit dem Greenhouse-Geisser-Korrekturfaktor angepasst werden.

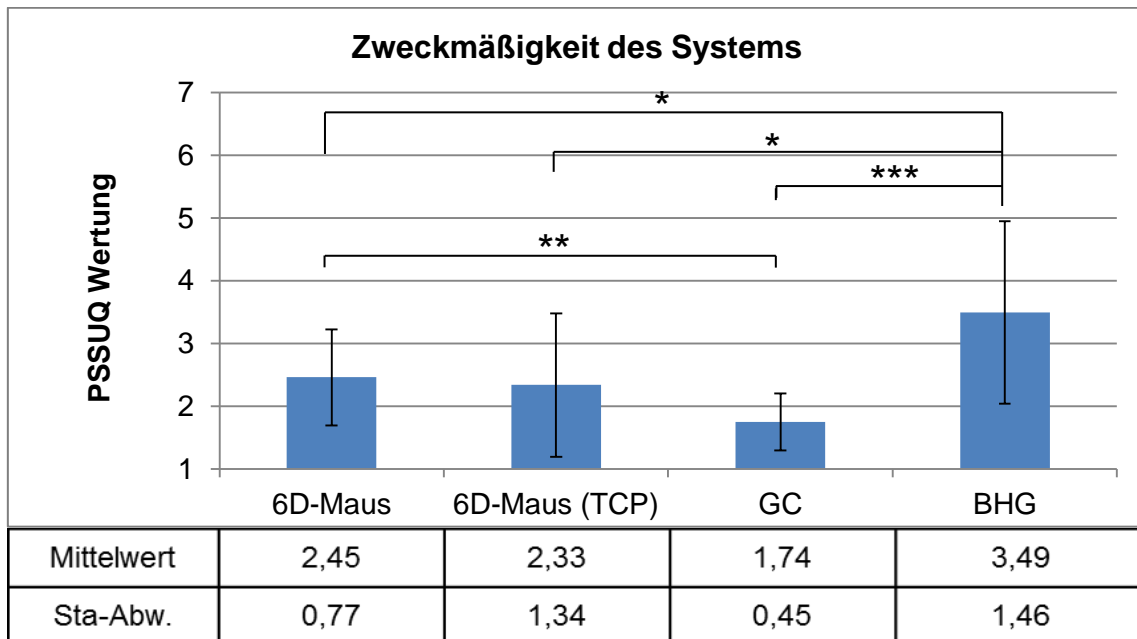


Abbildung: 53 PSSUQ-Bewertung: Eignung der Eingabetechnologien

Wie in Abbildung: 53 dargestellt, konnte mit dem Bonferroni-Test gezeigt werden, dass der *GC-Modus* signifikant besser bewertet wird als die 6D-Maus (Tisch) und das Bedienhandgerät ($p < 0,01$ und $p < 0,001$). Darüber hinaus wurde das Bedienhandgerät auch gegenüber den anderen beiden Bedienkonzepten signifikant schlechter bewertet (je $p < 0,05$). Auffällig bei der Darstellung der Ergebnisse ist die geringe Standardabweichung des *GC-Modus* im Vergleich zu den übrigen Konzepten.

Mit einer weiteren einfaktoriellen ANOVA ($F(3,57)=3,765$, $p < 0,05$, $\eta^2=0,165$, $1-\beta=0,785$) konnte gezeigt werden, dass die Wahl der Eingabetechnologie einen signifikanten Einfluss auf die Einschätzung hat, ob das System über alle erwarteten Funktionen verfügt (PSSUQ-Frage 18).

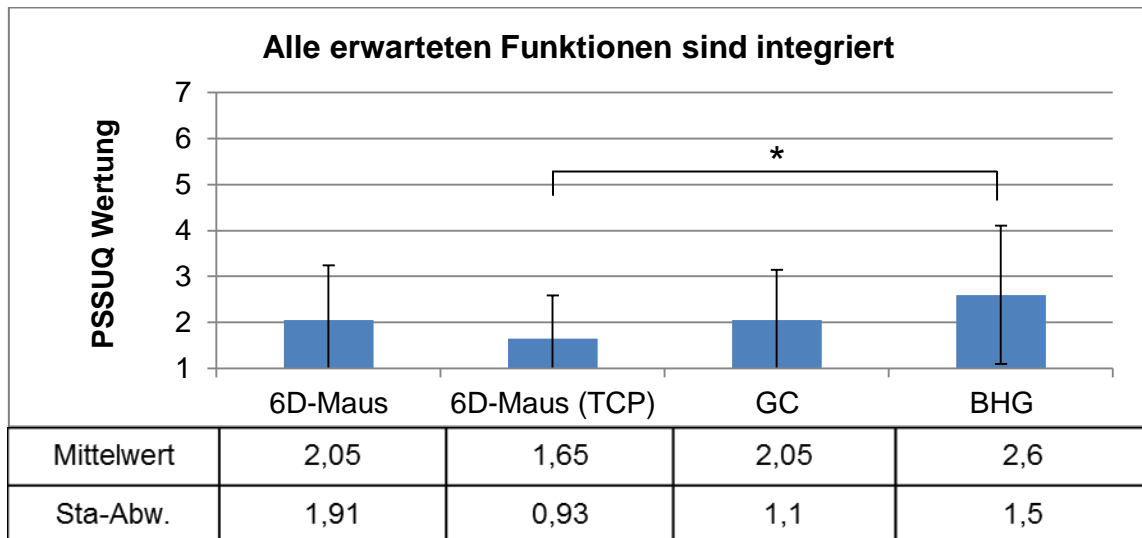


Abbildung 54: Auswertung der PSSUQ-Frage 18

Es lässt sich aus Abbildung 54 erkennen, dass nur der Unterschied zwischen Bedienhandgerät und der 6D-Maus am Greifer (bzw. 6D-Maus (TCP)) signifikant unterschiedlich bewertet wurde ($p < 0,05$). Darüber hinaus zeigen die Unterschiede zwischen *GC-Modus* bzw. 6D-Maus (Arbeitstisch) und dem Bedienhandgerät eine Tendenz auf.

Die letzte Frage des PSSUQ beschäftigt sich mit der Gesamtzufriedenheit der Probanden mit dem verwendeten Eingabekonzept. Auch diese Frage wurde mit einer einfaktoriellen ANOVA untersucht. Ebenso wie bei den vorherigen Fragen lässt sich auch hier eine Abhängigkeit der Bewertung von der zugrunde liegenden Eingabetechnologie feststellen ($F(3,57)=6,236$, $p < 0,001$, $\eta^2=0,247$, $1-\beta=0,953$).

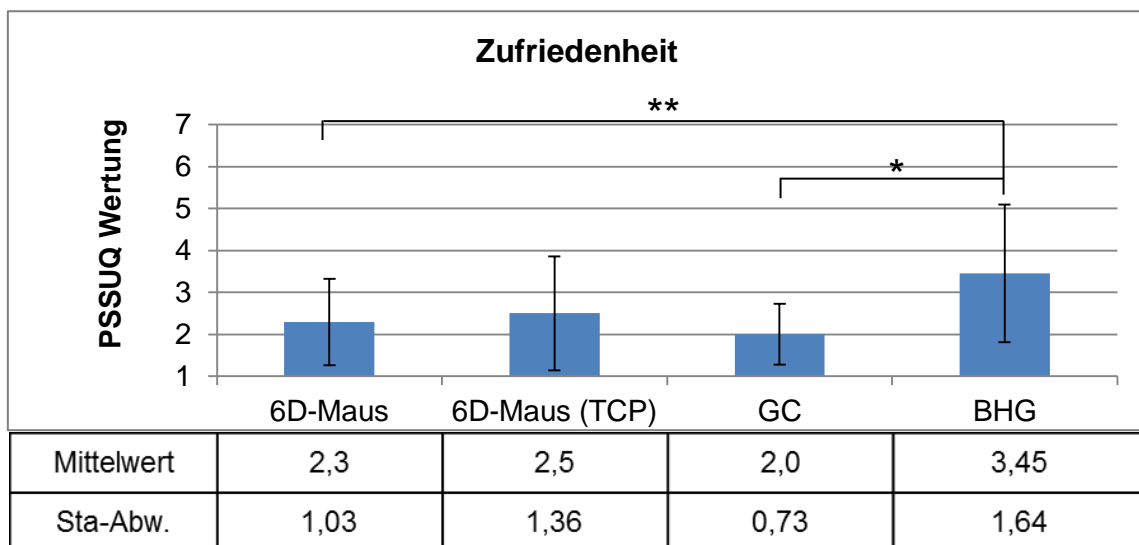


Abbildung 55: Auswertung der Frage 19 aus dem PSSUQ (* $p < 0,05$; ** $p < 0,01$)

Die Gesamtzufriedenheit wird für das Bedienhandgerät am schlechtesten bewertet. Dabei lässt sich beim Vergleich zum 6D-Maus-Konzept (Tisch) und zum *GC-Modus* ein signifikantes Ergebnis nachweisen ($p < 0,01$ bzw. $p < 0,05$). Zusätzlich gibt es eine ausgeprägte Tendenz zwischen BHG und der 6D-Maus am Greifer.

8.2.1.2 Subjektive Einschätzung der Anwendbarkeit und Präferenz der Eingabegeräte

Neben den standardisierten Fragen des PSSUQ erzielt auch der für diesen Versuch entwickelte Fragenbogen wichtige Erkenntnisse über die geeignetste Wahl der Interaktion. Ein wesentlicher Aspekt für die Beurteilung ist die Präferenz der Probanden bei der Wahl der Eingabetechnologien (siehe Abbildung 56).

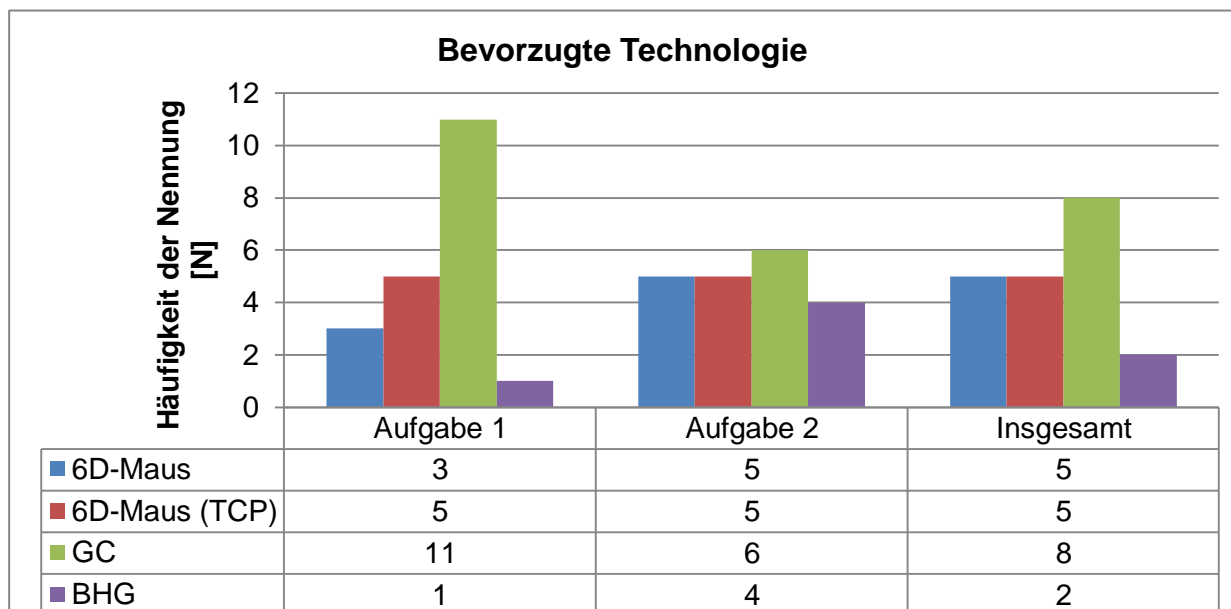


Abbildung 56: Präferenz bei der Eingabetechnologie

Hier zeigen die Bewertungen der Probanden vor allem für Aufgabe 1, also der Definition von vier Punkten auf der Arbeitsfläche, eine starke Tendenz zur Drehmomenten-Steuerung. An zweiter Stelle konnte sich die 6D-Maus am Greifer positionieren. Die am wenigsten präferierte Technologie stellte das Bedienhandgerät dar.

Für Aufgabe 2 ergibt sich ein deutlich homogeneres Bild. Auch in diesem Fall wurde die Drehmomenten-Steuerung mit geringerem Vorsprung als beliebteste Technologie eingestuft. Das BHG findet darüber hinaus bei dieser Aufgabe deutlich mehr

Befürworter als noch bei der ersten Aufgabe und kann somit beinahe komplett zu den übrigen Eingabetechnologien aufschließen.

Die Aussagen zur allgemeinen Präferenz ergeben eine Mischung der Einzelbewertungen. Der *GC-Modus* wird am stärksten präferiert und das Bedienhandgerät erhält erneut die wenigsten Stimmen.

Die Fragen zur Bewertung der Eignung des Systems wurden herangezogen, um die Aussagen zur persönlichen Präferenz noch stärker zu schärfen. Die Auswertung geschah wiederum mit einer einfaktoriellen ANOVA. Für diese ließ sich nachweisen, dass die Bewertung der Eignung der Geräte sowohl für Aufgabe 1 ($F(3,57)=8,173$, $p<0,001$, $\eta^2=0,301$, $1-\beta=0,988$) als auch Aufgabe 2 ($F(3,54)=4,937$, $p<0,05$, $\eta^2=0,215$, $1-\beta=0,891$) abhängig von der Wahl der Interaktionstechnologie ist.

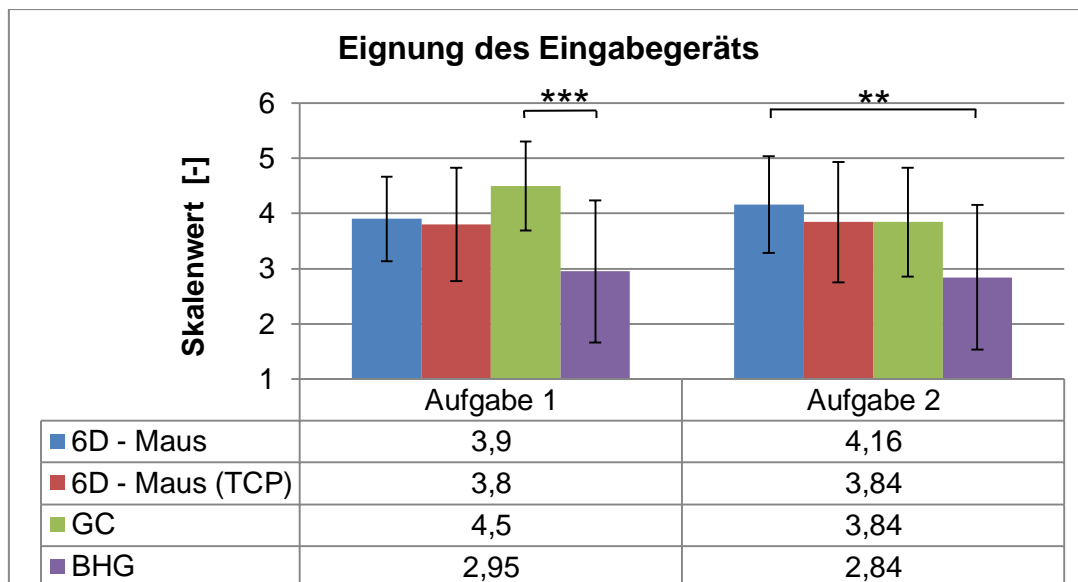


Abbildung 57: Eignungsbewertung der unterschiedlichen Eingabetechnologien bezüglich der untersuchten Aufgaben

So zeigt sich für Aufgabe 1, dass die Probanden den *GC-Modus* für signifikant ($p<0,001$) besser geeignet halten, als das Bedienhandgerät. Darüber hinaus lässt sich auch eine deutliche Tendenz beider Technologien gegenüber den Mauskonzepten zeigen, die sich in ihrer subjektiv wahrgenommenen Eignung für Aufgabe 1 zueinander ausgeglichen im Mittelfeld platzieren.

Für Aufgabe 2 kann gezeigt werden, dass sich signifikante Unterschiede bei der Eignungsbewertung zwischen 6D-Maus (Tisch) und dem Bedienhandgerät ergeben ($p<0,01$). Das Bedienhandgerät wird erneut als ungeeignetstes Konzept bewertet,

während sich die 6D-Maus am Greifer und der *GC-Modus* mit sehr homogenen Werten im Mittelfeld wiederfinden.

8.2.2 Objektive Daten

Zentraler objektiver Parameter bei der Evaluation der unterschiedlichen Eingabegeräte ist die benötigte Zeit zur Definition der benötigten Geometriedaten. Auf Grund technischer Probleme können allerdings nur die Zeiten von Aufgabe 1 sinnvoll verwendet werden.

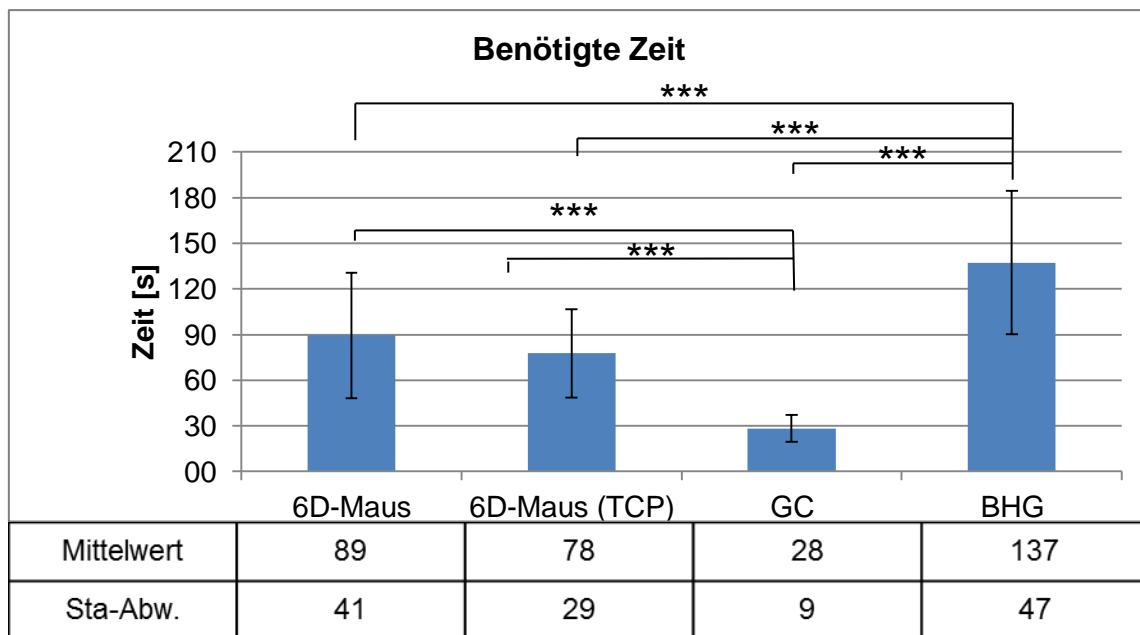


Abbildung 58: benötigte Zeit für Aufgabe 1 „Definition eines Handhabungsbereichs“

Die Auswertung des Versuchs erfolgt, wie auch bei den subjektiven Erhebungen, mit Hilfe einer einfaktoriellen ANOVA. Diese zeigt für die betrachteten Eingabegeräte einen signifikanten Einfluss auf die benötigte Zeit ($F(3,57)=81,220$, $p<0,001$, $\eta^2=0,810$, $1-\beta=1$).

Der Post-Hoc-Test zeigt, dass die zeitlichen Unterschiede zwischen allen Interaktionstechnologien, bis auf den Vergleich der 6D-Maus-Konzepte, signifikant ($p<0,001$) ausfallen. Dabei stellt der *GC-Modus* die schnellste Technologie dieser Untersuchung dar. Das Bedienhandgerät benötigt den höchsten zeitlichen Aufwand.

8.3 Ergebnisse

Basierend auf der Auswertung lässt sich feststellen, dass der „Gravity Compensation“-Modus des KUKA-LBR sowohl bezüglich subjektiver als auch

objektiver Parametern die besten Ergebnisse erzielt. Somit können sowohl Hypothese 1a und 1b angenommen werden. Besonders deutlich ist das Bild bezüglich Aufgabe 1 „Definition von Handlungsbereichen“, für die der *GC-Modus* in beinahe jedem der untersuchten Parameter die beste Technologie darstellt. Eine Ausnahme bildet die Frage nach der Zufriedenheit mit dem Vorhandensein aller erwarteten Funktionen. Besonders deutlich ist der Unterschied zu einem konventionellen Bedienhandgerät, das für alle betrachteten Evaluationsparameter signifikant schlechter abschneidet als der *GC-Modus*. Die beiden Maus-Konzepte sind in ihren subjektiven Bewertungen als auch bezüglich der benötigten Zeit auf einem sehr ähnlichen Niveau und lassen sich basierend auf den erhobenen Messdaten in ihrer Nutzerfreundlichkeit statistisch nicht unterscheiden. Auch bei einer Betrachtung der tendenziellen Unterschiede lassen sich nur innerhalb der einzelnen Fragen Unterschiede feststellen, die sich bei Betrachtung aller Kriterien wieder aufheben. Dementsprechend muss Hypothese 2 abgelehnt werden.

Bezüglich Aufgabe 2 schwindet ein Teil des Vorsprungs vom *GC-Modus* gegenüber den beiden 6D-Maus-Konzepten, die sich auch hier praktisch nicht in ihrer Bewertung unterscheiden. Die Unterschiede in Aufgabe 1 und 2 lassen sich dabei hauptsächlich mit der erhöhten Anforderung an die Eingabegenauigkeit des Eingabeinstruments argumentieren. Ein Indiz für diese These sind die Rückmeldungen der Probanden im Anschluss an den Versuch.

8.4 Diskussion

Auf Grund der technischen Probleme bei der Aufnahme der Bediendauern für Aufgabe 2 ist eine weitere Untersuchung theoretisch sinnvoll. Für die vorliegende Untersuchung steht aber vor allem die Auslegung des Systems auf die *aufgabenorientierte Programmierung* im Vordergrund. Da die benötigten Bedienzeiten für diesen Anwendungsfall vorliegen, ist eine Bewertung möglich.

Ein Vorteil vom *GC-Modus* und 6D-Maus am Greifer, der in dieser Studie nur zu einem Teil in die Bewertung mit eingeflossen ist, ist die große Robustheit gegen Fehlbedienungen auf Grund der nicht benötigten mentalen Transformation der Koordinatensysteme. Es ließ sich vor allem für das Bedienhandgerät in der Eingewöhnungsphase beobachten, dass Nutzer das Koordinatensystem des Systems nicht auf die Eingaberichtungen des Bedienhandgeräts projizieren konnten. Dabei kam es zu vereinzelt Kollisionen mit der Tischoberfläche und diese sind laut

Steinfeld et al. (2006) einer der entscheidenden Faktoren für die Bewertung der Eingabeleistung. Ein weiterer Vorteil des *GC-Modus* ist die direkte Interaktionsmöglichkeit mit allen Achsen des Roboters. Dieser Vorteil wird vor allem dann offenbar, wenn der Roboter sich nicht beliebig an eine Koordinate annähern kann. Diese Möglichkeit bietet zwar auch die Verwendung des Bedienhandgerätes, allerdings werden die einzelnen Achsen des Systems dann separat voneinander gesteuert. Auch wenn hiermit bezüglich der Positionierung ein ähnlich großes Maß an Flexibilität wie beim *GC-Modus* möglich ist, so ist die Bedienung komplex. Auf der einen Seite, weil die Position des TCP nun über eine kinematische Kette der einzelnen Achsen erzeugt werden muss und auf der anderen Seite, da die Drehrichtungen der Achsen schwer abzuschätzen sind. Die verwendeten 6D-Maus-Konzepte sind für diesen Fall nicht ausgelegt und erlauben nur die Steuerung der TCP-Position. Die Empfehlung für den *GC-Modus* entspricht weiteren Autoren (Lopez Infante & Kyrki, 2011; Wrede et al., 2013), die den Einsatz im Zusammenhang mit Nutzern ohne Expertenfähigkeiten als sinnvolle Lösung erachten.

Die direkte physische Manipulation des Roboters ist allerdings auch ein Nachteil in Bezug auf die Bediengenauigkeit. Krauß (2003) beschreibt die Limitierungen menschlicher Fähigkeiten und den Kompromiss zwischen Genauigkeit und benötigter Zeit menschlicher Eingabe. Dementsprechend kann beim Vergleich der subjektiven Bewertungen von Aufgabe 1 und 2 beobachtet werden, dass der Vorsprung des *GC-Modus* bei höheren Genauigkeitsanforderungen abfällt. Da die Bedienung bei den übrigen Konzepten prinzipiell skaliert werden könnte, sprich die Bewegung des TCP bezogen auf den Ausschlag des Bediengeräts, wären mit diesen auch sehr hohe Genauigkeiten problemlos denkbar.

9 Diskussion des Gesamtkonzepts

Auf Basis der dargestellten Untersuchungen konnten sowohl die grafische Benutzerschnittstelle als auch die Eingabekonzepte für Geometriedaten evaluiert und somit die wichtigen Aspekte des Interaktionsparadigmas ausgefüllt werden.

Dementsprechend scheint eine optimierte Umsetzung der evaluierten grafischen Benutzerschnittstelle in Verbindung mit der Drehmomenten-Steuerung eines Leichtbauroboters als vielversprechende Lösung für die in dieser Arbeit definierten Anforderungen.

Die Evaluation der GUI zeigt, dass eine Bedienung des Interaktionskonzepts auch für Nutzer, die keine oder wenig Expertise im Bereich der Roboterprogrammierung aufweisen, möglich ist und diese zumindest einen einfachen Prozess in kurzer Zeit definieren und parametrisieren können. Nachdem die Evaluierung in Form eines abstrahierten Bedienprozesses durchgeführt wurde, müssen die Ergebnisse in einem industriellen Umfeld bestätigt werden. Die Anlehnung der Prozessprogrammierung an das mentale Modell des Nutzers zeigt für einen einfachen Pick & Place-Vorgang sehr vielversprechende Ergebnisse und ist als sinnvoll zu betrachten. Ohne Vorschulung oder vorherige Erklärung der Prozessprogrammierung war es für 15 von 20 Probanden möglich einen Prozessablauf zu erstellen, der eine erfolgreiche Durchführung der Pick & Place-Aufgabe erlaubt hätte. Dementsprechend kann die in Kapitel 7.1 dargestellte Bedienoberfläche in weiten Teilen für das Gesamtkonzept übernommen werden. Die Untersuchung zeigt aber auch, dass stärker noch als bei der MCI im Consumerbereich Plausibilitätsgrenzen eingebaut werden müssen, die eine Fehlbedienung des Systems verhindern. Im Vergleich zur Interaktion mit alltäglichen Mensch-Computer-Interaktionen birgt die Benutzung eines Roboters starke Risiken einer Beschädigung sowohl für den Roboter selbst als auch für die Peripherie. Im Fall der Prozessprogrammierung bedeutet dies, dass eine Plausibilitätsprüfung des eingegebenen Prozesses erforderlich ist, der zum Beispiel Auslassungsfehler unterbindet. Für das in Kapitel 7.1 dargestellte Konzept kann diese Plausibilitätsgrenze wie in Abbildung 59 dargestellt implementiert werden. Das System weist den Nutzer dann darauf hin, dass der Prozess nicht vollständig ist und um eine weitere Funktion erweitert werden muss. Dazu werden diejenigen Funktionen markiert, die zu dieser Warnung beitragen. Im dargestellten Fall wird der Nutzer also darauf hingewiesen, dass das Bauteil zwar gegriffen, aber vor Ende des

Prozesses nicht abgelegt wird. Mit dieser zusätzlichen Hilfestellung ist die Erstellung unvollständiger Programme, wie in Kapitel 7.3 geschildert, nicht länger möglich.

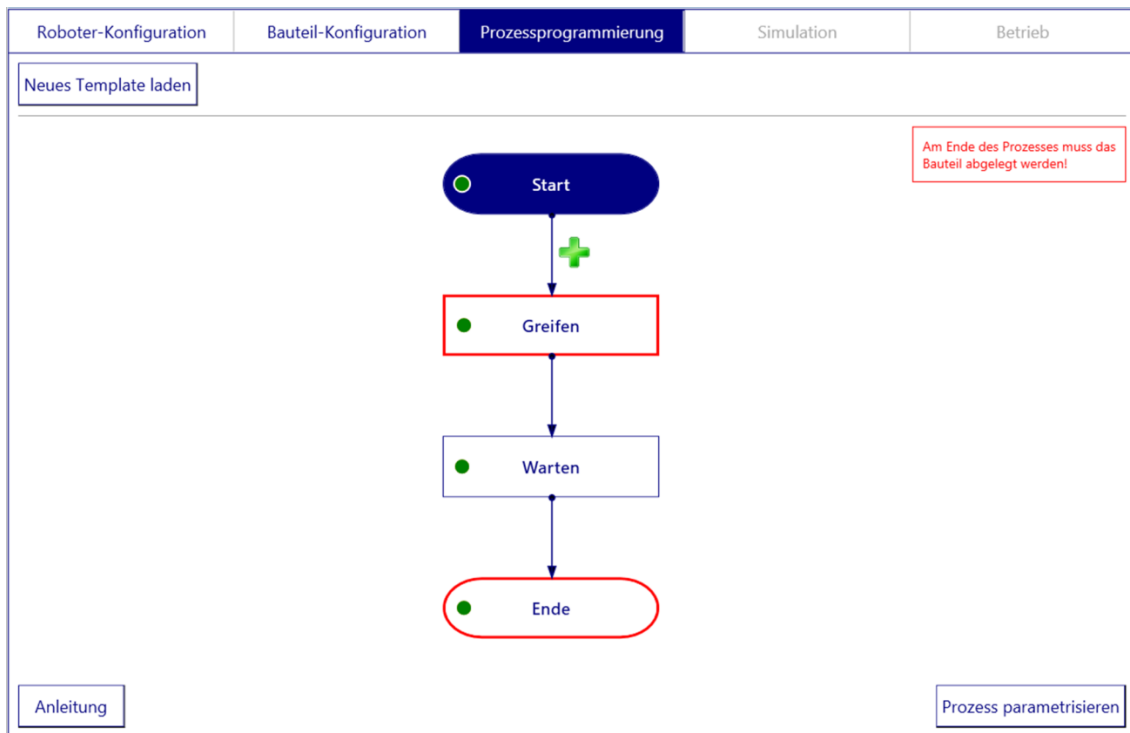


Abbildung 59: Prozessvalidierung

Eine ähnliche Form der Plausibilitätsschranke muss für die Modellierung der Bauteile erfolgen. In Zusammenhang mit der Evaluation der GUI konnte gezeigt werden, dass die Steuerung einer dreidimensionalen Darstellung von Modellen in einer Touch-Umgebung anwendbar ist. Allerdings wurde nur von einem Teil der Probanden die fehlerhafte Modellierung als solche erkannt. Dementsprechend ist es sinnvoll einen erfolgreichen Scan des realen Bauteils als Voraussetzung für die Speicherung des Modells vorzusehen. Die einfachste Möglichkeit einer solchen Implementierung ist die Erstellung eines zusätzlichen Konfigurationsschrittes. Die allgemeine Gestaltungsoptimierung der weiteren Bedienoberflächen folgt den Ergebnissen und Vorschlägen von Egger (2013).

10 Zusammenfassung und Ausblick

Zielsetzung dieser Arbeit war die Gestaltung einer ergonomischen Mensch-Roboter-Interaktion für eine auf Pick & Place-Szenarien ausgerichtete, transportable Roboterzelle. Diese soll vor allem zur Unterstützung bei Stückzahlschwankungen dienen und an bestehende Handarbeitsplätze angedockt werden, was die Nutzungszeiträume auf wenige Wochen bis Monate pro Arbeitsplatz reduziert. Als technisch sinnvolle und, für die nicht-spezialisierte Nutzergruppe, geeignete Programmiermethode wurde dafür die *aufgabenorientierte Programmierung* identifiziert.

Ein erster Meilenstein für die Entwicklung einer nutzerzentrierten Benutzerschnittstelle war die Entwicklung eines geeigneten konzeptuellen Modells mit Fokus auf die Prozessprogrammierung. Hierfür musste die Frage beantwortet werden, wie sich das mentale Modell der Nutzerklientel in Bezug auf Pick & Place-Arbeitsplätze beschreiben und auf das technische System projizieren lässt. Als Ergebnis einer Probandenstudie konnte ein Satz an Programmierfunktionen gewonnen werden, der eine nutzerzentrierte Darstellung und Eingabe für die vom FZI implementierten *flexiblen Programme* des Robotersystems erlaubt.

Zusammen mit den Ergebnissen einer systemergonomischen Analyse floss das konzeptuelle Modell in die Entwicklung einer grafischen Benutzerschnittstelle ein, die sich an den ergonomischen Heuristiken der Mensch-Computer-Interaktion orientiert. Aus ergonomischer Sicht standen hier sowohl die Validierung des konzeptionellen Modells sowie die Untersuchung der Anwendbarkeit von Heuristiken aus der HCI, die vorwiegend im Bereich der Consumer-Produkte entwickelt wurden, im Zentrum des Interesses. Zu diesem Zweck und zur Optimierung des GUI-Konzepts wurden eine Expertenstudie sowie eine Probandenstudie durchgeführt, um Struktur- und Verständnisprobleme bei der Bedienung aufzudecken und Optimierungspotenziale zu generieren. Dabei ließen sich vielversprechende Ergebnisse in Bezug auf die Einbindung der Analyse des mentalen Modells der Nutzerklientel beobachten. Gleichzeitig konnte festgestellt werden, dass die Verwendung von Heuristiken aus dem Consumer-Bereich sinnig ist, allerdings zusätzliche Anstrengungen zur Absicherung der Systemsicherheit, z. B. in Form von Plausibilitätsschranken, unternommen werden müssen.

Um eine, aus ergonomischen Gesichtspunkten, sinnvolle Eingabe von Geometriedaten, sowohl für den Standardanwendungsfall, als auch die Rückfallebene *Teach-In*, zu gewährleisten, wurde eine Probandenstudie durchgeführt. Hierbei wurde eine Auswahl an Interaktionskonzepten in Bezug auf subjektive Rückmeldungen der Probanden als auch die benötigte Zeit hin untersucht. Dabei konnte, den objektiven und subjektiven Bewertungskriterien folgend, der *GC-Modus* des KUKA LBR als sinnvollstes Interaktionskonzept für den betrachteten Kontext herausgearbeitet werden.

Insofern konnte das hier dargestellte Interaktionsparadigma auf Basis der *aufgabenorientierten Programmierung* sowohl mit einer nutzerzentrierten grafischen Bedienoberfläche als auch einer geeigneten Interaktionstechnologie ausgestattet werden. Die Anforderungen an das System, die als Teil der Forschungsziele definiert wurden, können auf Basis der dargestellten Untersuchungen als erreicht betrachtet werden. Zu einer abschließenden Beurteilung des Konzepts wird eine Realstudie im Unternehmensumfeld empfohlen. Hierbei bietet sich ein Vergleich des optimierten Systems mit den bisher dargestellten Ergebnissen an, um die Auswirkungen einzelner Anpassungen zu evaluieren.

Auf Grund der Eignung der hier dargestellten Zelle für ein breites Einsatzspektrum innerhalb von KMUs, aber auch Großunternehmen, wäre zudem eine Erweiterung der grafischen Bedienoberfläche im Hinblick auf die angebotene Funktionsvielfalt sinnvoll. Damit wäre zudem eine interessante Basis für weitere Untersuchung mentaler Modelle in Bezug auf unterschiedliche Arbeitsprozesse gegeben. So ließe sich evaluieren, ob die in dieser Arbeit erzielten homogenen Ergebnisse zum Teil der Komplexität der betrachteten Aufgabe geschuldet sind. Auch könnte ein möglicher Zusammenhang der von den Probanden präferierten Segmentierungstiefe mit den von Miller (1956) beschriebenen 7 ± 2 psychologischen Einheiten des Arbeitsgedächtnisses untersucht werden.

11 Literaturverzeichnis

- VDI 5005 (1990). *Bürokommunikation - Software-Ergonomie in der Bürokommunikation*. Berlin: Beuth Verlag.
- VDI 2860 (1990). *Montage- und Handhabungstechnik; Handhabungsfunktionen, Handhabungseinrichtungen; Begriffe, Definitionen, Symbole*. Berlin: Beuth Verlag.
- DIN EN ISO 9241-11 (1999). *Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten - Teil 11: Anforderungen an die Gebrauchstauglichkeit*. Berlin: Beuth Verlag.
- DIN EN ISO 9241-110 (2008). *Ergonomie der Mensch-System-Interaktion - Teil 110: Grundsätze der Dialoggestaltung*. Berlin: Beuth Verlag.
- DIN EN ISO 8373 (2010). *Roboter und Robotikgeräte - Wörterbuch*. Berlin: Beuth Verlag.
- VDI/VDE 3850 Blatt 1 (2012). *Gebrauchstaugliche Gestaltung von Benutzungsschnittstellen für technische Anlagen - Konzepte, Prinzipien und grundsätzliche Empfehlungen*. Berlin: Beuth Verlag.
- Ahlström, B., Lenman, S., & Marmolin, T. (1992). Overcoming touchscreen user fatigue by workplace design. In D. Wixon (Ed.), *Posters and short talks of the 1992 SIGCHI conference* (p. 101–101).
- Akgun, B., Cakmak, M., Yoo, J. W., & Thomaz, A. L. (2012). Trajectories and keyframes for kinesthetic teaching. In H. Yanco, A. Steinfeld, V. Evers, & O. C. Jenkins (Eds.), *the seventh annual ACM/IEEE international conference* (p. 391–391).
- Albu-Schäffer, A., Haddadin, S., Ott, C., Stemmer, A., Wimböck, T., & Hirzinger, G. (2007). The DLR lightweight robot: design and control concepts for robots in human environments. *Industrial Robot: An International Journal*, 34(5), 376–385. doi:10.1108/01439910710774386
- Awais, M. (2013). *Intuitive Human-Robot Interaction by Intention Recognition*. (Doctoral Dissertation).
- Backhaus, J., & Reinhart, G. (2013). Efficient application of task-oriented programming for assembly systems. In *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)* (pp. 750–755).

- Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An Empirical Evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction*, 24(6), 574–594. doi:10.1080/10447310802205776
- Bell, C. J., Shenoy, P., Chalodhorn, R., & Rao, Rajesh P N. (2008). Control of a humanoid robot by a noninvasive brain-computer interface in humans. *Journal of neural engineering*, 5(2), 214–220. doi:10.1088/1741-2560/5/2/012
- Berns, K., & Schmidt, D. (2010). *Programmierung mit LEGO Mindstorms NXT: Robotersysteme, Entwurfsmethodik, Algorithmen*. eXamen.press. Berlin: Springer.
- Bhanu, B. (1987). CAD-based robot vision. *Computer*, 20(8), 13–16.
- Bierfreund, B., Nölle, K., & Hollmann, R. (2008). Benchmark und Evaluierung der Benutzungsfreundlichkeit von Spracheingabegeräten in Robotersystemen. In M. Schütte (Ed.), *Jahresdokumentation / Gesellschaft für Arbeitswissenschaft e. V.: Vol. 2008. Produkt- und Produktions-Ergonomie. Aufgaben für Entwickler und Planer : Bericht zum 54. Kongress der Gesellschaft für Arbeitswissenschaft vom 9. - 11. April 2008 [an der Technischen Universität München]*. Dortmund: GfA-Press.
- Biggs, G., & MacDonald, B. (2003). A survey of robot programming systems. In *Proceedings of the Australasian conference on robotics and automation* (pp. 1-3).
- Bischoff, R., Kazi, A., & Seyfarth, M. (2002). The MORPHA style guide for icon-based programming. In *IEEE ROMAN 2002. 11th IEEE International Workshop on Robot and Human Interactive Communication* (pp. 482–487).
- Bischoff, R., Kurth, J., Schreiber, G., Koeppe, R., Albu-Schäffer, A., Beyer, A., ... (2010). The kuka-dlr lightweight robot arm-a new reference platform for robotics research and manufacturing. In *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)* (pp. 1-8).
- Bortot, D. (2014). *Ergonomic Human-Robot Coexistence in the Branch of Production. Robotik und Automation*. München: Dr. Hut.
- Bortz, J., Bortz-Döring, & Döring, N. (2009). *Forschungsmethoden und Evaluation: Für Human- und Sozialwissenschaftler ; mit 87 Tabellen* (4., überarb. Aufl., Nachdr). *Springer-Lehrbuch Bachelor, Master*. Heidelberg: Springer-Medizin-Verl.
- Breuninger, J. (2014). Einsatz moderner gestenbasierter Touch-Interaktion in Industrie und Haustechnik: Forschungsprojekt GE3STIK. *Ergonomie Aktuell*, (15).

- Breuninger, J. & Popova-Dlugosch, S. (2013). *Gestaltungsempfehlungen für effiziente Touchscreen-Bedienkonzepte*. bayme vbm - Die bayrischen Metall- und Elektro-Arbeitgeber.
- Breuninger, J., Popova-Dlugosch, S., & Bengler, K. (2012). Einsatz von modernen innovativen Touch-Interaktionskonzepten in Produktivsystemen. In M. Schütte (Ed.): *Vol. 2012. Jahresdokumentation / Gesellschaft für Arbeitswissenschaften e.V., Gestaltung nachhaltiger Arbeitssysteme - Wege zur gesunden, effizienten und sicheren Arbeit. Vom 22. - 24. Februar 2012*. Dortmund: GfA-Press.
- Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry*, 189, 194.
- Bubb, H. (1993). Systemergonomische Gestaltung. *Ergonomie*, 3, 390-420.
- Bubb, H., & Schmidtke, H. (1993). Systemergonomie. *Ergonomie*, 3.
- Bühner, M. (2011). *Einführung in die Test- und Fragebogenkonstruktion* (3., aktualisierte und erw. Aufl.). *Psychologie*. München, Boston [u.a.]: Pearson Studium.
- Burgard, W., Christaller, T., & Knoll, A. (2003). Robotik. *Handbuch der Künstlichen Intelligenz, chapter 0*, 871–939.
- Burnett, I., Baker, M. J., Bohus, C., Carlson, P., Yang, S., & van Zee, P. (1995). Scaling up visual programming languages. *Computer*, 28(3), 45–54.
doi:10.1109/2.366157
- Calinon, S., & Billard, A. (2007). Active Teaching in Robot Programming by Demonstration. In *RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication* (pp. 702–707).
- Colestock, H. (2005). *Industrial robotics: Selection, design, and maintenance*. New York: McGraw-Hill.
- Collins, A., & Centner, D. (1987). How people construct mental models. *Cultural models in language and thought*, 243.
- Cooper, A., & Reimann, R. (2003). *About face 2.0: The essentials of interaction design*. New York, Chichester: Wiley.
- Cox, P. T., & Smedley, T. J. (1998). Visual programming for robot control. In *1998 IEEE Symposium on Visual Languages* (pp. 217–224).
- Craig, J. J. (2005). *Introduction to Robotics*.

- Cumming, G., & Finch, S. (2005). Inference by Eye: Confidence Intervals and How to Read Pictures of Data. *American Psychologist*, 60(2), 170.
- Dautenhahn, K. (2007). Methodology and themes of human-robot interaction: a growing research field. In *International Journal of Advanced Robotic Systems*.
- DeJong, B. P., Colgate, J. E., & Peshkin, M. A. (2011). Mental Transformations in Human-Robot Interaction. In X. Wang (Ed.), *Mixed Reality and Human-Robot Interaction* (pp. 35–51). Dordrecht: Springer Netherlands.
- Delson, N., & West, H. (1996). Robot programming by human demonstration: adaptation and inconsistency in constrained motion. In *IEEE International Conference on Robotics and Automation* (pp. 30–36).
- Dillmann, R., & Huck, M. (1991). *Informationsverarbeitung in der Robotik. Springer-Lehrbuch*. Berlin [u.a.]: Springer.
- Dixon, K. R. (2004). Predictive Robot Programming: Theoretical and Experimental Analysis. *The International Journal of Robotics Research*, 23(9), 955–973. doi:10.1177/0278364904044401
- Donald A. Norman. (1983). Some Observations on Mental Models. In D. Gentner & A. L. Stevens (Eds.), *Cognitive science. Mental models*. Hillsdale, N.J: Erlbaum.
- Dorau, Rainer (2011): *Emotionales Interaktionsdesign. Gesten und Mimik interaktiver Systeme*. Berlin, New York: Springer (X.media.press).
- Dutke, S. (1994). *Mentale Modelle: Konstrukte des Wissens und Verstehens : kognitionspsychologische Grundlagen für die Software-Ergonomie. Arbeit und Technik: Vol. 4*. Göttingen: Verlag für Angewandte Psychologie.
- Eberleh, E. (1994). *Einführung in die Software-Ergonomie: Gestaltung graphisch-interaktiver Systeme: Prinzipien, Werkzeuge, Lösungen* (2., völlig neu bearb. Aufl). *Mensch-Computer-Kommunikation. Grundwissen: Vol. 1*. Berlin, New York: De Gruyter.
- Egger, B. (2013). *Analyse der Nutzerfreundlichkeit einer konzeptionellen Benutzeroberfläche zur Roboterprogrammierung* (unveröffentlichte Bachelorarbeit). Technische Universität München, Garching.
- EU Robotics. (2011). *The EU Robotics Initiative, Report on identified gaps*. Retrieved from http://eurobotics-project.eu/cms/upload/euRobotics_Deliverable_D1.1.1_Report_on_gaps.pdf

- Field, A. P. (2013). *Discovering statistics using IBM SPSS statistics: And sex and drugs and rock 'n' roll* (4th ed). Los Angeles: Sage.
- Flavio S. Fogliatto, Giovanni J.C. da Silveira, & Denis Borenstein. (2012). The mass customization decade: An updated review of the literature. *International Journal of Production Economics*, 138(1), 14–25. doi:10.1016/j.ijpe.2012.03.002
- Forlines, C., Wigdor, D., Shen, C., & Balakrishnan, R. (2007). Direct-touch vs. mouse input for tabletop displays. In M. B. Rosson & D. Gilmore (Eds.), *the SIGCHI conference* (p. 647–647).
- Foster, M. E., By, T., Rickert, M., & Knoll, A. (2006). Human-Robot dialogue for joint construction tasks. In F. Quek, J. Yang, D. Massaro, A. Alwan, & T. J. Hazen (Eds.), *the 8th international conference* (p. 68–68).
- Foster, M. E., Gaschler, A., Giuliani, M., Isard, A., Pateraki, M., & Petrick, R. P. (2012). Two people walk into a bar. In L.-P. Morency, D. Bohus, H. Aghajan, J. Cassell, A. Nijholt, & J. Epps (Eds.), *the 14th ACM international conference* (p. 3–3).
- Friedrich, H., Münch, S., Dillmann, R., Bocionek, S., & Sassin, M. (1996). Robot programming by Demonstration (RPD): Supporting the induction by human interaction. *Machine Learning*, 23(2-3), 163–189. doi:10.1007/BF00117443
- Gentner, D., & Gentner, D. R. (1983). Flowing Waters or Teeming Crowds: Mental Models of Electricity. In D. Gentner & A. L. Stevens (Eds.), *Cognitive science. Mental models*. Hillsdale, N.J: Erlbaum.
- Giovani Da Silveira, Denis Borenstein, & Flávio S Fogliatto. (2001). Mass customization: Literature review and research directions. *International Journal of Production Economics*, 72(1), 1–13. doi:10.1016/S0925-5273(00)00079-7
- Goodrich, M. A., & Schultz, A. C. (2007). Human-Robot Interaction: A Survey. *Foundations and Trends® in Human-Computer Interaction*, 1(3), 203–275. doi:10.1561/11000000005
- Green, T., & Petre, M. (1996). Usability Analysis of Visual Programming Environments: A ‘Cognitive Dimensions’ Framework. *Journal of Visual Languages & Computing*, 7(2), 131–174. doi:10.1006/jvlc.1996.0009
- Green, T. R. G., & Petre, M. (1992). When Visual Programs are Harder to Read than Textual Programs. In *Human-Computer Interaction: Tasks and Organisation*,

- Proceedings of ECCE-6 (6th European Conference on Cognitive Ergonomics)*. GC van der Veer, MJ Tauber, S. Bagnarola and M. Antavolits. Rome, CUD (pp. 167–180).
- Haddadin, Sami; Albu-Schäffer, Alin; Hirzinger, Gerd (2011): Safe Physical Human-Robot Interaction: Measurements, Analysis and New Insights. In: Bruno Siciliano, Oussama Khatib, Frans Groen, Makoto Kaneko und Yoshihiko Nakamura (Hg.): *Robotics Research*, Bd. 66. Berlin, Heidelberg: Springer Berlin Heidelberg (Springer tracts in advanced robotics), S. 395–407.
- Hart, C. W. (1995). Mass customization: conceptual underpinnings, opportunities and limits. *International Journal of Service Industry Management*, 6(2), 36–45.
doi:10.1108/09564239510084932
- Hasanuzzaman, M., Zhang, T., Ainpornaramveth, V., Kiatisevi, P., Shirai, V., & Ueno, H. (2004). Gesture based human-robot interaction using a frame based software platform. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)* (pp. 2883–2888).
- Haslbeck, A., Pecot, K., Popova, S., Eichinger, A., & Bengler, K. (2011). Menschliche Zuverlässigkeit bei der alphanumerischen Eingabe mittels unterschiedlicher Eingabemedien. In VDI Wissensforum GmbH (Ed.), *Technische Zuverlässigkeit 2011. Entwicklung und Betrieb zuverlässiger Produkte*. (25th ed., pp. 87–97). Düsseldorf: VDI-Verlag.
- Haslbeck, A., Popova, S., Krause, M., Pecot, K., Mayer, J., & Bengler, K. (2011). Experimental Evaluations of Touch Interaction Considering Automotive Requirements. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, ... (Eds.), *Lecture Notes in Computer Science. Human-Computer Interaction. Interaction Techniques and Environments* (pp. 23–32). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Haun, M. (2007). *Handbuch Robotik: Programmieren und Einsatz intelligenter Roboter* (1st ed.). VDI-Buch. Berlin: Springer.
- Hayashi, T., Kawamoto, H., & Sankai, Y. (2005). Control method of robot suit HAL working as operator's muscle using biological and dynamical information. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3063–3068).

- Hesse, S., & Malisa, V. (2009). *Taschenbuch Robotik - Montage - Handhabung*. München: Hanser, Carl.
- Hirzinger, G., Sporer, N., Albu-Schaffer, A., Hahnle, M., Krenn, R., Pascucci, A., & Schedl, M. (2002). DLR's torque-controlled light weight robot III-are we reaching the technological limits now? In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on* (Vol. 2, pp. 1710–1716).
- Hoggan, E., Brewster, S. A., & Johnston, J. (2008). Investigating the effectiveness of tactile feedback for mobile touchscreens. In M. Czerwinski, A. Lund, & D. Tan (Eds.), *Proceeding of the twenty-sixth annual CHI conference* (p. 1573–1573).
- Holzinger, A. (2005). Usability engineering methods for software developers. *Communications of the ACM*, 48(1), 71–74. doi:10.1145/1039539.1039541
- Hudak, P., Courtney, A., Nilsson, H., & Peterson, J. (2003). Arrows, Robots, and Functional Reactive Programming. In G. Goos, J. Hartmanis, J. Leeuwen, J. Jeuring, & S. L. P. Jones (Eds.), *Lecture Notes in Computer Science. Advanced Functional Programming* (pp. 159–187). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Humburger, R. (1998). *Konzeption eines Systems zur aufgabenorientierten Roboterprogrammierung* (Als Ms. gedr). *Berichte aus der Produktionstechnik: Bd. 98,4*. Aachen: Shaker.
- Ishii, K., Zhao, S., Inami, M., Igarashi, T., & Imai, M. (2009). Designing Laser Gesture Interface for Robot Control. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, ... (Eds.), *Lecture Notes in Computer Science. Human-Computer Interaction – INTERACT 2009* (pp. 479–492). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Jackson, J. (2007). Microsoft robotics studio: A technical introduction. *IEEE Robotics & Automation Magazine*, 14(4), 82–87. doi:10.1109/M-RA.2007.905745
- Jäkel, R., Schmidt-Rohr, S. R., Zhixing Xue, Lösch, M., & Dillmann, R. (2010). Learning of probabilistic grasping strategies using Programming by Demonstration. In *2010 IEEE International Conference on Robotics and Automation (ICRA 2010)* (pp. 873–880).
- Kahneman, D. (2014). *Schnelles Denken, langsames Denken*. München: Pantheon.

- Kanayama, Y. J., & Wu, C. T. (2000). It's time to make mobile robots programmable. In *2000 ICRA. IEEE International Conference on Robotics and Automation* (pp. 329–334).
- Kaneko, K., Harada, K., Kanehiro, F., Miyamori, G., & Akachi, K. (2008). Humanoid robot HRP-3. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2471–2478).
- Kemp, C. C., Anderson, C. D., Nguyen, H., Trevor, A. J., & Xu, Z. (2008). A point-and-click interface for the real world: laser designation of objects for mobile manipulation. In *Human-Robot Interaction (HRI), 2008 3rd ACM/IEEE International Conference on* (pp. 241–248).
- Knoop, S., Pardowitz, M., & Dillmann, R. (2008). From Abstract Task Knowledge to Executable Robot Programs. *Journal of Intelligent and Robotic Systems*, 52(3-4), 343–362. doi:10.1007/s10846-008-9221-x
- Krauß, L. (2003): *Entwicklung und Evaluation einer Methodik zur Untersuchung von Interaktionsgeräten für Maschinen- und Prozessbediensysteme mit grafischen Benutzungsoberflächen*. Als Ms. gedr. Kaiserslautern: Univ (Fortschritt-Berichte pak Mensch-Maschine-Interaktion, 7).
- Kuehne, J., Verl, A., Xue, Z., Ruehl, S., Zoellner, J. M., Dillmann, R., ... (2009). 6d object localization and obstacle detection for collision-free manipulation with a mobile service robot. In *Advanced Robotics, 2009. ICAR 2009. International Conference on* (pp. 1–6).
- Kugelmann, D. (1999). *Aufgabenorientierte Offline-Programmierung von Industrierobotern (Dissertation)* (Vol. 127): Herbert Utz Verlag.
- Landhäußer, M., Hey, T., & Tichy, W. F. (2014). Deriving time lines from texts. In B. Turhan, A. B. Bener, C. Mericli, A. Miransky, & L. L. Minku (Eds.), *the 3rd International Workshop* (pp. 45–51).
- Lange, C. (2010). Blickgesteuerte Interaktion mit Peripheriegeräten – technische Lösung und ergonomische Absicherung. In K.-H. Pantke (Ed.), *Mensch und Maschine. Wie Brain-Computer-Interfaces und andere Innovationen gelähmten Menschen kommunizieren helfen*. Frankfurt, M.: Mabuse-Verl.
- Lange, C., Wohlfarter, M., & Bubb, H. (2009). Automated Analysis of Eye-Tracking Data for the Evaluation of Driver Information Systems According to ISO/TS 15007-2: 2001. In *Human-Computer Interaction. New Trends* (pp. 105-110). Springer.

- Lewis, J. R. (2002). Psychometric evaluation of the PSSUQ using data from five years of usability studies. *International Journal of Human-Computer Interaction*, 14(3-4), 463-488.
- Lopez Infante, M., & Kyrki, V. (2011). Usability of force-based controllers in physical human-robot interaction. In A. Billard, P. Kahn, J. A. Adams, & G. Trafton (Eds.), *the 6th international conference* (p. 355–355).
- Lozano-Perez, T. (1983). Robot programming. *Proceedings of the IEEE*, 71(7), 821–841.
- Martin, T. B., Ambrose, R. O., Diftler, M. A., Platt, R., & Butzer, M. J. (2004). Tactile gloves for autonomous grasping with the NASA/DARPA Robonaut. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04* (pp. 1713–1718 Vol.2).
- Meyer, C. (2011). *Aufnahme und Nachbearbeitung von Bahnen bei der Programmierung durch Vormachen von Industrierobotern. IPA-IAO Forschung und Praxis: Vol. 507*. Heimsheim: Jost-Jetter.
- Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, 63(2), 81–97. doi:10.1037/h0043158
- Mizuuchi, I., Yoshikai, T., Sodeyama, Y., Nakanishi, Y., Miyadera, A., Yamamoto, T., ... (2006). Development of musculoskeletal humanoid kotaro. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on* (pp. 82–87).
- Moosbrugger, H., & Kelava, A. (2012). *Testtheorie und Fragebogenkonstruktion* (2., aktualisierte und überarbeitete Auflage). *Springer-Lehrbuch*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg.
- Myers, B. A. (1990). Taxonomies of visual programming and program visualization. *Journal of Visual Languages & Computing*, 1(1), 97–123. doi:10.1016/S1045-926X(05)80036-9
- Neto, P., Mendes, N., Araújo, R., Pires, J. N., & Moreira, A. P. (2012). High-level robot programming based on CAD: dealing with unpredictable environments. *Industrial Robot: An International Journal*, 39(3), 294–303. doi:10.1108/01439911211217125

- Nickel, K., & Stiefelhagen, R. (2007). Visual recognition of pointing gestures for human–robot interaction. *Image and Vision Computing*, 25(12), 1875–1884. doi:10.1016/j.imavis.2005.12.020
- Nielsen, J. (1994). *Usability engineering* ([Updated ed.]). San Francisco, Calif.: Morgan Kaufmann Publishers.
- Nof, S. Y. (1999). *Handbook of industrial robotics* (2nd ed). New York: John Wiley.
- Norman, D. A. (1983). Design rules based on analyses of human error. *Communications of the ACM*, 26(4), 254–258.
- Pan, Z., Polden, J., Larkin, N., van Duin, S., & Norrish, J. (2010). Recent Progress on Programming Methods for Industrial Robots. In *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)* (pp. 1–8).
- Pan, Z., Polden, J., Larkin, N., van Duin, S., & Norrish, J. (2012). Recent progress on programming methods for industrial robots. *Robotics and Computer-Integrated Manufacturing*, 28(2), 87–94. doi:10.1016/j.rcim.2011.08.004
- Pardowitz, M., Knoop, S., Dillmann, R., & Zollner, R. D. (2007). Incremental Learning of Tasks From User Demonstrations, Past Experiences, and Vocal Comments. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 37(2), 322–332. doi:10.1109/TSMCB.2006.886951
- Patel, R., Hedelind, M., & Lozan-Villegas, P. (2012). Enabling robots in small-part assembly lines: The "ROSETTA approach"-an industrial perspective. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on* (pp. 1–5).
- Perzanowski, D., Adamse, W., & Schultz, A. (1998). Communicating with a Semi-Autonomous Robot Combining Natural Language and Gestures. In *Workshop on Integrating Robotics Research: Taking the Next Leap* (pp. 55-59).
- Peterson, J., Hager, G., & Serjentov, A. (2001). Composable robot controllers. In *2001 International Symposium on Computational Intelligence in Robotics and Automation* (pp. 149–154).
- Peterson, J., Hager, G. D., & Hudak, P. (1999). A language for declarative robotic programming. In *International Conference on Robotics and Automation* (pp. 1144–1151).

- Pires, J. N. (2006). Robotics for small and medium enterprises: control and programming challenges. *Industrial Robot*, 33(6),
- Pires, J. N., Veiga, G., & Araújo, R. (2009). Programming-by-demonstration in the coworker scenario for SMEs. *Industrial Robot: An International Journal*, 36(1), 73–83. doi:10.1108/01439910910924693
- Powers, A., & Kiesler, S. (2006). The advisor robot: tracing people's mental model from a robot's physical attributes. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction* (pp. 218–225).
- Pritschow, G., & Tuffentsammer, K. (1987). A New Procedure for the Economic Programming of Tooling Robots in the Case of Small Batches. *CIRP Annals - Manufacturing Technology*, 36(1), 339–342. doi:10.1016/S0007-8506(07)62617-1
- Raeder. (1985). A Survey of Current Graphical Programming Techniques. *Computer*, 18(8), 11–25. doi:10.1109/MC.1985.1662971
- Rahimi, M., & Karwowski, W. (1992). *Human-robot interaction*. London, Washington, DC: Taylor & Francis.
- Renner, I. (2007). *Methodische Unterstützung funktions-orientierter Baukastenentwicklung am Beispiel Automobil*. Universität München.
- Roos, E., & Behrens, A. (1997). Off-line programming of industrial robots — Adaptation of simulated user programs to the real environment. *Computers in Industry*, 33(1), 139–150. doi:10.1016/S0166-3615(97)00018-3
- Rouse, W. B., & Morris, N. M. (1986). On looking into the black box: Prospects and limits in the search for mental models. *Psychological bulletin*, 100(3), 349.
- Rudlof, C. (2006). *Handbuch der Software-Ergonomie: Usability Engineering* (2. Auflage). Tübingen.
- Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N., & Fujimura, K. (2002). The intelligent ASIMO: system overview and integration. In *IROS 2002: IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2478–2483).
- Sauro, J., & Lewis, J. R. (2012). *Quantifying the user experience: Practical statistics for user research*. Waltham, Mass: Morgan Kaufmann.
- Schaal, S. (2007). The New Robotics-towards human-centered machines. *HFSP journal*, 1(2), 115–126. doi:10.2976/1.2748612

- Schäfer, T. (2007). *Verfahren zur hämmernden Blechumformung mit Industrieroboter (Dissertation)*. IPA-IAO-Forschung und Praxis: Vol. 452.
- Schilling, R. J. (1990). *Fundamentals of robotics: Analysis and control*. Englewood Cliffs, N.J.: Prentice Hall.
- Scholtz, J. (2003). Theory and evaluation of human robot interactions. In *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the* (pp. 10 pp).
- Schraft, R. D., & Meyer, C. (2006). The need for an intuitive teaching method for small and medium enterprises. *VDI BERICHTE*, 1956, 95.
- Sears, A. (1991). Improving touchscreen keyboards: design issues and a comparison with other devices. *Interacting with Computers*, 3(3), 253–269. doi:10.1016/0953-5438(91)90016-U
- Sears, A., & Shneiderman, B. (1991). High precision touchscreens: design strategies and comparisons with a mouse. *International Journal of Man-machine studies*, 34(4), 593–613. doi:10.1016/0020-7373(91)90037-8
- Shneiderman, B., & Plaisant, C. (2010). *Designing the user interface: Strategies for effective human-computer interaction* (5th ed.). Boston: Addison-Wesley.
- Siciliano, B., & Khatib, O. (Eds.). (2008). *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Siegert, H.-J., & Bocionek, S. (1996). *Robotik: Programmierung intelligenter Roboter*. Berlin [etc.] Springer.
- Silve, S. (2013). Making Instructions for Others: Exploring Mental Models Through a Simple Exercise. *Distinguished Speakers Program*, 74.
- Solvang, B., Sziebig, G., & Korondi, P. (2008). Robot Programming in Machining Operations. *Robot Manipulators, I-Tech Education and Publishing*, 479–496.
- Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2004). *Robot Dynamics and Control*.
- Spur, G., & Uhlmann, E. (2012). Industrieroboter. In K.-H. Grote & J. Feldhusen (Eds.), *Dubbel* (pp. T97). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Staggers, N., & Norcio, A. F. (1993). Mental models: concepts for human-computer interaction research. *International Journal of Man-machine studies*, 38(4), 587–605.

- Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., & Goodrich, M. (2006). Common metrics for human-robot interaction. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction* (pp. 33–40).
- Stock, M., & Miller, K. (2003). Optimal Kinematic Design of Spatial Parallel Manipulators: Application to Linear Delta Robot. *Journal of Mechanical Design*, *125*(2), 292. doi:10.1115/1.1563632
- Suresh Kotha. (1996). From mass production to mass customization: The case of the National Industrial Bicycle Company of Japan. *European Management Journal*, *14*(5), 442–450. doi:10.1016/0263-2373(96)00037-0
- Tanie, K. (2003). Humanoid robot and its application possibility. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. MFI2003* (pp. 213–214).
- Tarchanidis, K. N., & Lygouras, J. N. (2003). Data glove with a force sensor. *IEEE Transactions on Instrumentation and Measurement*, *52*(3), 984–989. doi:10.1109/TIM.2003.809484
- Thomas, U., Hirzinger, G., Rumpe, B., Schulze, C., & Wortmann, A. (2013). A new skill based robot programming language using UML/P Statecharts. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on* (pp. 461–466).
- Thrun, S. (2004). Toward a framework for human-robot interaction. *Human-Computer Interaction*, *19*(1-2), 9–24.
- Tikka, V., & Laitinen, P. (2006). Designing Haptic Feedback for Touch Display: Experimental Study of Perceived Intensity and Integration of Haptic and Audio. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, ... (Eds.), *Lecture Notes in Computer Science. Haptic and Audio Interaction Design* (pp. 36–44). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Tulving, E., Schacter, D. L., & Stark, H. A. (1982). Priming effects in word-fragment completion are independent of recognition memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *8*(4), 336–342. doi:10.1037/0278-7393.8.4.336
- Vaughan, R. T., Gerkey, B. P., & Howard, A. (2003). On device abstractions for portable, reusable robot code. In *2003 IEEE/RSJ International Conference on*

- Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)* (pp. 2421–2427).
- VDMA. (2004). *Leitfaden Software-Ergonomie: Gestaltung von Bedienoberflächen. Software*. Frankfurt am Main: VDMA-Verl.
- Vilimek, Roman (2007): *Gestaltungsaspekte multimodaler Interaktion im Fahrzeug. Ein Beitrag aus ingenieurpsychologischer Perspektive*. Düsseldorf: VDI-Verl. (Fortschritt-Berichte VDI / 22 / VDI, Nr. 26).
- Vogl, W. (2009). *Eine interaktive räumliche Benutzerschnittstelle für die Programmierung von Industrierobotern. Forschungsberichte / IWB: Vol. 228*. München: Utz.
- Voyles, R. M., & Khosla, P. K. (1995). Tactile gestures for human/robot interaction. In *1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots* (pp. 7–13).
- Waldherr, S., Romero, R., & Thrun, S. (2000). A Gesture Based Interface for Human-Robot Interaction. *Autonomous Robots*, 9(2), 151–173.
doi:10.1023/A:1008918401478
- Wenrui Dai, & Kampker, M. (2000). User oriented integration of sensor operations in a offline programming system for welding robots. In *2000 ICRA. IEEE International Conference on Robotics and Automation* (pp. 1563–1567).
- Wittenberg, G. (1995). Developments in offline programming: an overview. *Industrial Robot: An International Journal*, 22(3), 21–23. doi:10.1108/EUM0000000004186
- Wrede, S., Emmerich, C., Grünberg, R., Nordmann, A., Swadzba, A., & Steil, J. (2013). A User Study on Kinesthetic Teaching of Redundant Robots in Task and Configuration Space. *Journal of Human-Robot Interaction*, 2(1), 56–81.
- Yang, X.-D., Grossman, T., Irani, P., & Fitzmaurice, G. (2011). TouchCuts and TouchZoom. In D. Tan, G. Fitzpatrick, C. Gutwin, B. Begole, & W. A. Kellogg (Eds.), *the 2011 annual conference* (p. 2585–2585).
- Yin, Y. H., Fan, Y. J., & Xu, L. D. (2012). EMG and EPP-integrated human-machine interface between the paralyzed and rehabilitation exoskeleton. *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society*, 16(4), 542–549. doi:10.1109/TITB.2011.2178034

- Zaeh, M., & Vogl, W. (2006). Interactive laser-projection for programming industrial robots. In *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality* (pp. 125–128).
- Zemiti, N., Morel, G., Ortmaier, T., & Bonnet, N. (2007). Mechatronic Design of a New Robot for Force Control in Minimally Invasive Surgery. *IEEE/ASME Transactions on Mechatronics*, *12*(2), 143–153. doi:10.1109/TMECH.2007.892831
- Zhang, H., Chen, H., Xi, N., Zhang, G., & He, J. (2006). On-Line Path Generation for Robotic Deburring of Cast Aluminum Wheels. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2400–2405).
- Zöllner, R., Rogalla, O., Dillmann, R., & Zollner, M. (2002). Understanding users intention: programming fine manipulation tasks by demonstration. In *IROS 2002: IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1114–1119).
- Zöllner, R., Rogalla, O., Ehrenmann, M., & Dillmann, R. (2005). Mapping complex tasks to robots: Programming by demonstration in real-world environments. In *Advances in Human-Robot Interaction* (pp. 119–136). Springer.

Anhang A – Fragebögen

A.1 Fragebogen zur Charakterisierung der Nutzerklientel für die Entwicklung einer intuitiven Bedienoberfläche

VP: _____

Datum: _____

Fragebogen zur Charakterisierung der Nutzerklientel für die Entwicklung einer intuitiven Bedienoberfläche

Vielen Dank, dass Sie sich dazu bereit erklärt haben an der Befragung teilzunehmen. Hintergrund der Versuche ist die Entwicklung einer ergonomischen Bedienoberfläche und den dazugehörigen Eingabegeräten für die Programmierung von flexiblen Roboterzellen im Zusammenhang mit Pick- & Place-Aufgaben. Hierbei stehen vor allem eine geringe Einarbeitungszeit und einfache Handhabung im Vordergrund.

Damit diese Ziele erreicht werden können, ist es nötig die Erwartungen, Erfahrungen und Bedürfnisse der möglichen späteren Nutzer mit einzubeziehen. Im folgenden Fragebogen werden Sie darum gebeten allgemeine Fragen zu Ihrer Person sowie speziellere Fragen zu Erfahrungen im Bereich der Computersysteme und Ihre Erwartungen bzw. Herangehensweise zu beantworten.

Im Anschluss daran erfolgt dann die Durchführung des eigentlichen Versuchs.

Wir versichern Ihnen, dass wir alle Angaben und Versuchsdaten streng vertraulich behandeln und die aufgenommenen Videos nur zur Auswertung der Versuche verwenden.

Allgemeine Angaben zur Person:

Alter: _____

Geschlecht: männl. weibl.

Wie lange arbeiten Sie schon für die Robert Bosch AG? _____

In welchem Tätigkeitsbereich sind Sie zurzeit tätig? _____

Erfahrungen im Umgang mit Computersystemen:

Verwenden Sie Computersysteme im privaten Umfeld? ja nein

Wenn ja, welche? (z.B. Smartphones, PC, Spielekonsole)

Welche Software-Programme verwenden Sie in Ihrem Arbeitsumfeld?

Welche Bediengeräte (z.B. Maus, Trackball, Touch-Pad/Screen) verwenden Sie in Ihrem Arbeitsumfeld?

Was ärgert Sie oft im allgemeinen Umgang mit Software-Programmen?

An welche Probleme bzw. Schwierigkeiten können Sie sich im Umgang mit den unterschiedlichen Bediengeräten erinnern?

Welche Eigenschaften im Zusammenhang mit Bedienoberflächen sind Ihnen wichtig?

- | | |
|---|--|
| Hohe Bedien- und Rückmeldegeschwindigkeit | gar nicht <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> außerordentlich |
| Unkomplizierter Einstieg | gar nicht <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> außerordentlich |
| Anzeige aller Daten | gar nicht <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> außerordentlich |
| Hilfssysteme (z.B. Wizard, „Hilfe“) | gar nicht <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> außerordentlich |
| Eindeutige Nutzerführung | gar nicht <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> außerordentlich |
| Übersichtlichkeit | gar nicht <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> außerordentlich |
| Design | gar nicht <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> außerordentlich |
| Ähnlichkeit mit bekannten Systemen | gar nicht <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> außerordentlich |

Was ist Ihnen bei der Bedienung eines Computersystems an Ihrem Arbeitsplatz besonders wichtig?

Was wäre Ihnen bei der Programmierung bzw. Bedienung eines Roboters besonders wichtig?

In welchem Zusammenhang (z.B. beruflich, privat) befanden Sie sich zuletzt in der Situation die Bedienung eines neuen Systems (Software und/oder Bedienteil) erlernen zu müssen und wie gingen Sie dabei vor?

Sind Schwierigkeiten beim Erlernen des neuen Systems aufgetreten? Wenn ja welche?

Aufgabe 2:

Im Folgenden möchten wir Sie nun bitten die Pick & Place-Arbeitsaufgabe (unabhängig von Aufgabe 1) nach unterschiedlichem Raster in einzelne Teilschritte aufzuteilen.

Beispiel:

Teilhandlung 1 → Teilhandlung 2 → ...

Gliedern Sie die Aufgabe bitte in 4-5 Teilschritte.

Gliedern Sie die Aufgabe bitte in 6-7 Teilschritte.

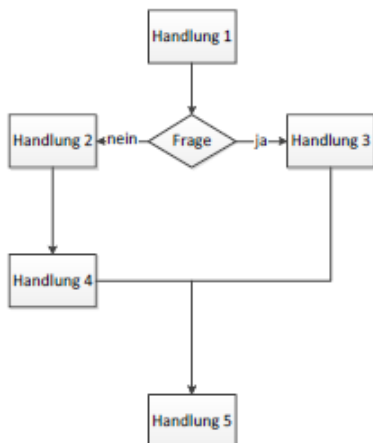
Gliedern Sie die Aufgabe bitte in 8-9 Teilschritte.

Gliedern Sie bitte die Aufgabe in so viele Teilschritte wie es Ihnen möglich ist!

Welche Aufgabensegmentierung entsprach am ehesten Ihren Vorstellungen? (falls keine der bisherigen Möglichkeiten Ihren Vorstellungen entsprach, so tragen Sie bitte eine Gliederung der Arbeitsaufgabe ein, die Ihren Vorstellungen entgegenkommt)

Sind Sie im Umgang mit Funktionsablaufdiagrammen, wie dem Folgenden, vertraut?

ja nein



A.2 Fragebogen und Aufgabensammlung für die Evaluation der grafischen Benutzerschnittstelle

Fragebogen zur Person

1) Versuchsperson-Nummer: _____

2) Alter: _____

3) Geschlecht: männlich weiblich

4) Welche Art von Tätigkeit üben Sie an der Uni aus?

Student Wissenschaftlicher Mitarbeiter Sonstiges: _____

5) Welche der folgenden Touchgeräte benutzen Sie privat oder beruflich?

Smartphone mit Touchscreen nie sehr häufig

Laptop / Netbook mit Touchscreen nie sehr häufig

iPad / Tablet-Computer nie sehr häufig

Sonstiges: _____ nie sehr häufig

6) Wie schätzen Sie ihre Roboterkenntnis ein?

schlecht sehr gut

7) Wie viel Erfahrung besitzen Sie in der Programmierung von Robotern?

keine sehr viel

VP-Nummer _____

Versuchsaufgaben

Bearbeiten Sie die einzelnen Aufgaben der Bereiche Roboter-Konfiguration, Bauteil-Konfiguration, Prozessprogrammierung, Simulation und Betrieb der Reihe nach.

- Lesen Sie zuerst in den Versuchsaufgaben, was Sie tun sollen.
- Bearbeiten Sie anschließend die Aufgabe mit der Programmieroberfläche.
- Versuchen Sie danach die Fragen stichpunktartig, in eigenen Worten, zu beantworten.

Roboter-Konfiguration

Aufgabe 1)

Positionieren Sie das vor ihnen auf dem Tisch stehende Robotermodell richtig. Was müssen Sie bei der Positionierung des Roboters beachten?

Aufgabe 2)

Was müssen Sie bei der Positionierung der Kameras beachten?

Warum ist die Positionierung der Kameras wichtig?

Aufgabe 3)

Wie müssen Sie bei der Kalibrierung der Kameras vorgehen?

Warum ist die Kalibrierung der Kameras wichtig?

Evaluation Roboter-Konfiguration

Was könnte an der Oberflächengestaltung der Roboter-Konfiguration verbessert werden?

Bauteil-Konfiguration

Aufgabe 4)

Fügen Sie als neues Bauteil den „Kontaktträger“ hinzu und überprüfen Sie, ob alle für die Handhabung relevanten Oberflächengeometrien vorhanden sind. Nehmen Sie als Vergleich gerne den Kontaktträger vor Ihnen auf dem Tisch zur Hand.

Aufgabe 5)

Überprüfen sie die vorgegebenen Griffe am Kontaktträger und löschen sie nicht stabile Griffe.

Evaluation Bauteil-Konfiguration

Was könnte an der Oberflächengestaltung der Bauteil-Konfiguration verbessert werden?

Prozessprogrammierung

Aufgabe 6)

Programmieren Sie den gezeigten Prozess aus dem Video. Eine genaue Parametrisierung der Arbeitsschritte ist an dieser Stelle noch nicht nötig.

Aufgabe 7)

Parametrisieren Sie die „Warte“-Funktion aus Ihrem Prozess.

Aufgabe 8)

Laden Sie das Template „Pick- & Place“ und ersetzen Sie die Funktion „Ablegen“ durch die Funktion „Einlegen“.

Evaluation Prozessprogrammierung

Was könnte an der Oberflächengestaltung der Prozessprogrammierung verbessert werden?

Simulation

Aufgabe 9)

Starten Sie die Simulation. Welche Informationen werden dargestellt?

Aufgabe 10)

Erhöhen Sie die Simulationsgeschwindigkeit.

Evaluation Simulation

Was könnte an der Oberflächengestaltung der Simulation verbessert werden?

Betrieb

Aufgabe 11)

Starten Sie den Probetrieb.

Aufgabe 12)

Starten Sie den Betrieb.

Aufgabe 13)

Speichern Sie das Programm unter dem Namen „VP__“.

Evaluation Betrieb

Was könnte an der Oberflächengestaltung des Betriebs verbessert werden?

A.3 Fragebogen für die Wahl einer geeigneten Eingabetechnologie

VP: _____

Datum: _____

Fragebogen zur Auswahl einer geeigneten Steuerungstechnologie für mobile Roboterzellen

Vielen Dank, dass Sie sich dazu bereit erklärt haben an der Studie zur Nutzbarkeit verschiedener Interaktionstechnologien zur Steuerung einer mobilen Roboterzelle teilzunehmen. Hintergrund der Versuche ist die Entwicklung einer ergonomischen Bedienoberfläche und den dazugehörigen Eingabegeräten für die Programmierung von flexiblen Roboterzellen im Zusammenhang mit Pick- & Place-Aufgaben. Hierbei stehen vor allem eine geringe Einarbeitungszeit und einfache Handhabung im Vordergrund. Um letztere zu gewährleisten, werden im folgenden Versuch 4 unterschiedliche Eingabegeräte auf ihre Tauglichkeit in Bezug auf die zwei gängigsten Aufgabentypen getestet. Bevor Sie den jeweiligen Versuch mit einem der Eingabegeräte durchführen haben Sie ein paar Minuten Zeit sich einzugewöhnen.

Wir versichern Ihnen, dass wir alle Angaben und Versuchsdaten streng vertraulich behandeln und die aufgenommenen Videos nur zur Auswertung der Versuche verwenden.

Allgemeine Angaben zur Person:

Alter: _____

Beruf: _____

Geschlecht: männl. weibl.

Haben Sie Erfahrungen im Umgang mit 6D-Manipulationen (hierzu zählen auch CAD-Systeme)? Wenn ja, welche?:

Welche der folgenden Eingabemethoden für 6D-Anwendungen haben Sie bereits verwendet:

6D - Maus	nie <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> sehr häufig
Kraft-Momenten-Sensor	nie <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> sehr häufig
Bedienhandgerät	nie <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> sehr häufig

Instruktion

Dieser Fragebogen gibt Ihnen Gelegenheit, uns Ihre Reaktion auf das System mitzuteilen, das Sie eben verwendet haben. Ihre Antworten werden uns helfen zu verstehen, welche Aspekte des Systems Ihnen kritisch erscheinen und welche Aspekte Sie überzeugen.

Während Sie die Fragen beantworten, denken Sie möglichst an die Aufgaben, die Sie mit dem System bearbeitet haben.

Bitte lesen Sie jede Aussage und geben Sie an, wie stark Sie dieser Aussage zustimmen oder nicht zustimmen, indem Sie die entsprechende Zahl ankreuzen. Wenn eine Aussage nicht zutrifft, kreuzen Sie "nicht zu beantworten" an.

Bitte unterstützen Sie Ihre Antworten durch frei formulierte Kommentare.

Nachdem Sie diesen Fragebogen ausgefüllt haben, werden wir Ihre Antworten mit Ihnen gemeinsam durchsprechen, um sicher zu stellen, dass wir alle Ihre Stellungnahmen verstanden haben.

Danke!

VP: _____ System: _____

1. Insgesamt bin ich zufrieden, wie leicht dieses System zu bedienen ist.

Stimme voll zu	1	2	3	4	5	6	7	Stimme gar nicht zu	nicht zu beantworten
									<input type="checkbox"/>

Anmerkungen: _____

2. Es war einfach, das System zu bedienen.

Stimme voll zu	1	2	3	4	5	6	7	Stimme gar nicht zu	nicht zu beantworten
									<input type="checkbox"/>

Anmerkungen: _____

3. Ich würde die Aufgaben und Szenarien effektiv mit Hilfe des Systems erledigen.

Stimme voll zu	1	2	3	4	5	6	7	Stimme gar nicht zu	nicht zu beantworten
									<input type="checkbox"/>

Anmerkungen: _____

4. Ich konnte Aufgaben und Szenarien schnell mit Hilfe des Systems erledigen.

Stimme voll zu	1	2	3	4	5	6	7	Stimme gar nicht zu	nicht zu beantworten
									<input type="checkbox"/>

Anmerkungen: _____

5. Ich konnte Aufgaben und Szenarien effizient mit Hilfe des Systems erledigen.

Stimme voll zu	1	2	3	4	5	6	7	Stimme gar nicht zu	nicht zu beantworten
									<input type="checkbox"/>

Anmerkungen: _____

6. Ich fühlte mich wohl bei der Bedienung des Systems.

Stimme voll zu	1	2	3	4	5	6	7	Stimme gar nicht zu	nicht zu beantworten
									<input type="checkbox"/>

Anmerkungen: _____

7. Es war leicht zu lernen, das System zu bedienen.

Stimme voll zu	1	2	3	4	5	6	7	Stimme gar nicht zu	nicht zu beantworten
									<input type="checkbox"/>

Anmerkungen: _____

8. Ich glaube ich konnte das System schnell produktiv einsetzen.

Stimme voll zu	1	2	3	4	5	6	7	Stimme gar nicht zu	nicht zu beantworten
									<input type="checkbox"/>

Anmerkungen: _____

9. Dieses System hat alle Funktionen und Fähigkeiten, die ich von ihm erwarte.

Stimme voll zu	1	2	3	4	5	6	7	Stimme gar nicht zu	nicht zu beantworten
									<input type="checkbox"/>

Anmerkungen: _____

10. Insgesamt bin ich mit diesem System zufrieden.

Stimme voll zu	1	2	3	4	5	6	7	Stimme gar nicht zu	nicht zu beantworten
									<input type="checkbox"/>

Anmerkungen: _____

Abschlussfragebogen:

Das Interaktionssystem war für die Erfüllung der Aufgabe „Aufspannen einer Fläche“ geeignet:

- | | |
|--------------------------------|---|
| 3D – Maus | Stimme nicht zu <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Stimme sehr zu |
| 3D – Maus am Roboter | Stimme nicht zu <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Stimme sehr zu |
| Kraft-Momenten-Sensor | Stimme nicht zu <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Stimme sehr zu |
| Benutzeroberfläche / Handgerät | Stimme nicht zu <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Stimme sehr zu |

Welches System war Ihrer Meinung nach am besten für diese Aufgabe geeignet und warum?

Was könnte an diesem System weiter verbessert werden?

Welches System war am wenigsten für diese Aufgabe geeignet und warum?

Das Interaktionssystem war für die Erfüllung der Aufgabe „Pick- & Place“ geeignet:

- | | |
|--------------------------------|---|
| 3D – Maus | Stimme nicht zu <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Stimme sehr zu |
| 3D – Maus am Roboter | Stimme nicht zu <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Stimme sehr zu |
| Kraft-Momenten-Sensor | Stimme nicht zu <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Stimme sehr zu |
| Benutzeroberfläche / Handgerät | Stimme nicht zu <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Stimme sehr zu |

Welches System war Ihrer Meinung nach am besten für diese Aufgabe geeignet und warum?

Was könnte an diesem System weiter verbessert werden?

Welches System war am wenigsten für diese Aufgabe geeignet und warum?

Welches der getesteten Systeme war insgesamt am besten für die Erfüllung beider Aufgabentypen geeignet?

Anhang B Bezeichnung der abgeleiteten Funktionen aus dem mentalen Modell

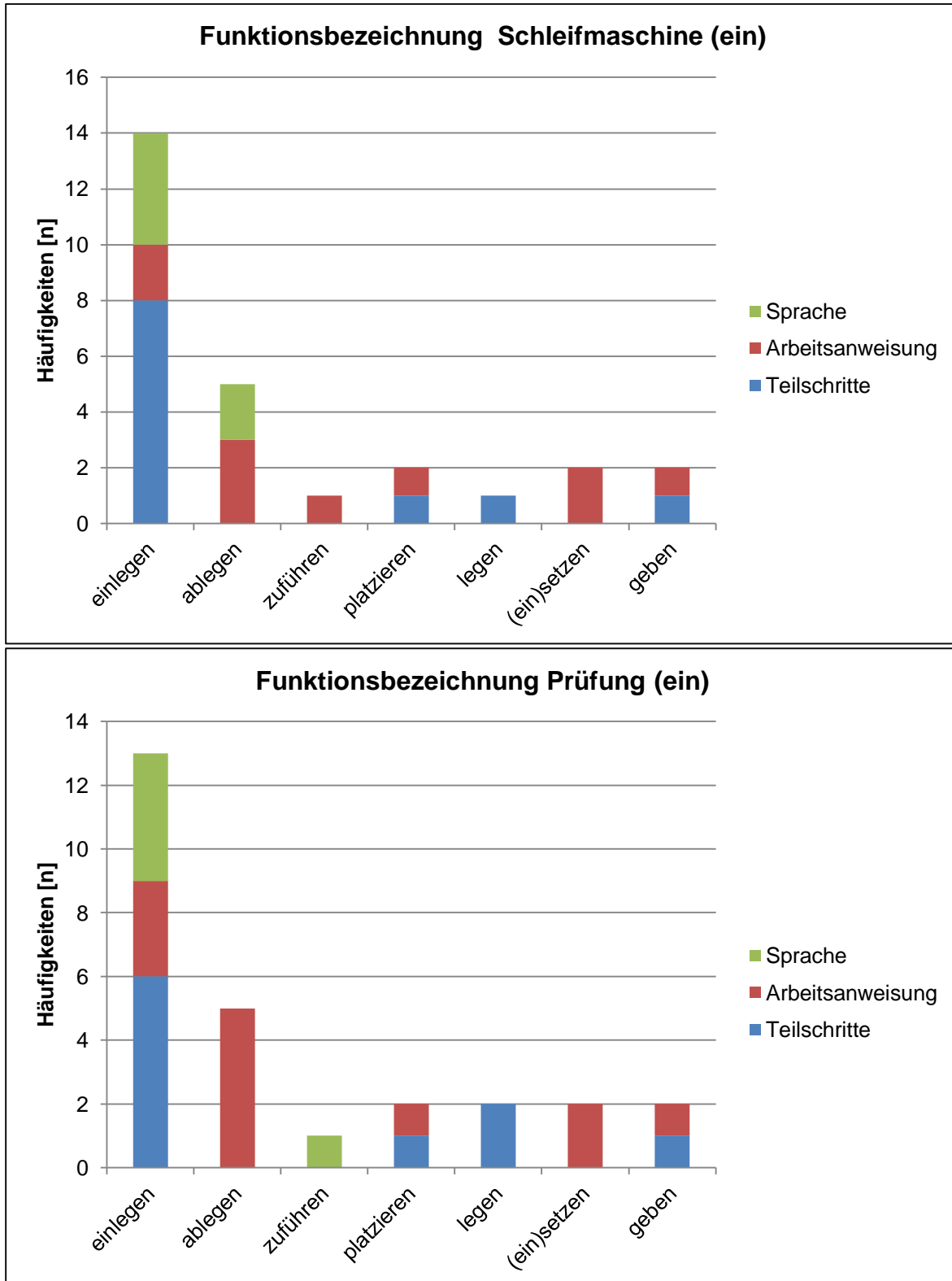


Abbildung 60 Einlegeprozesse

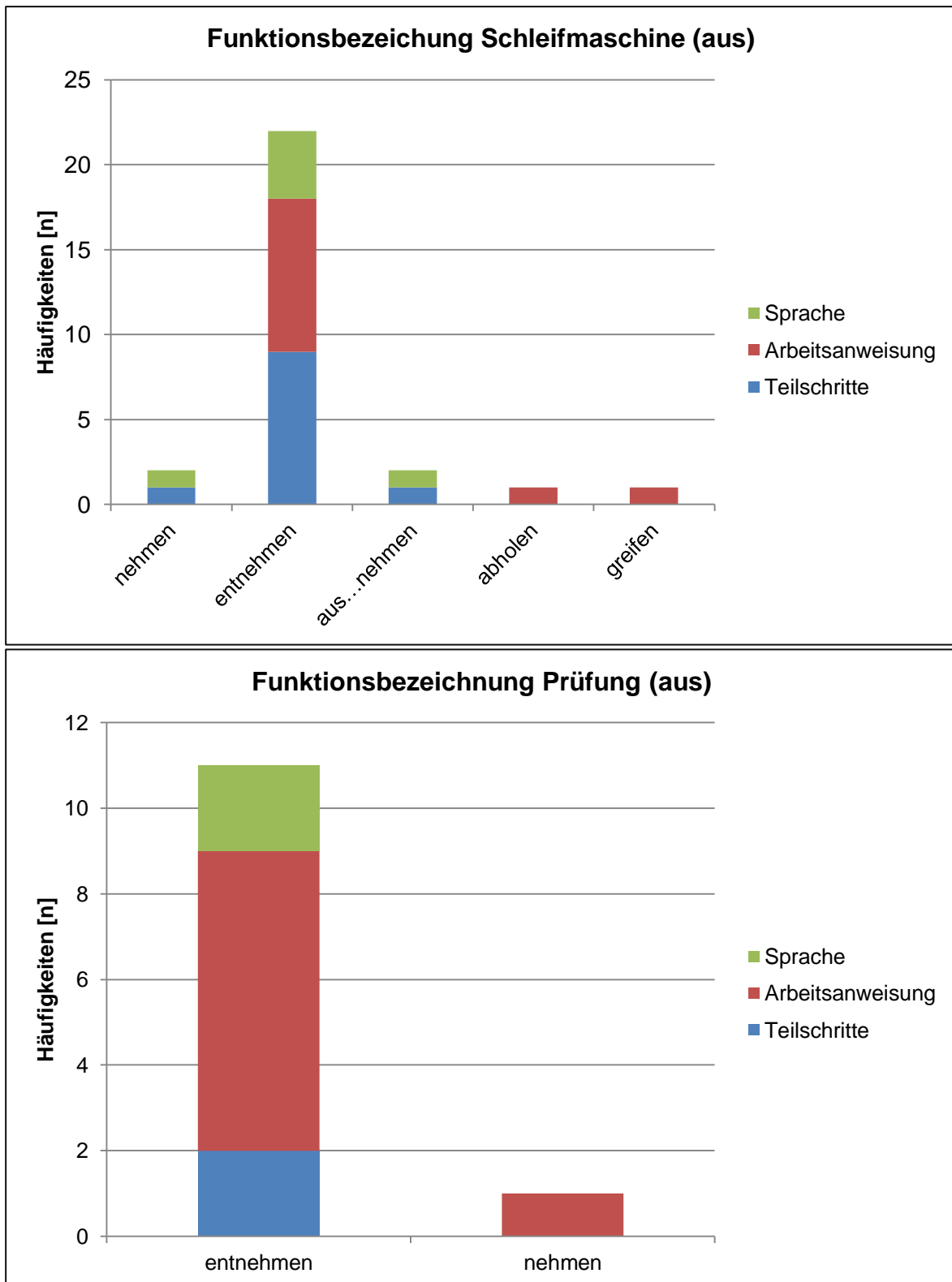


Abbildung 61 Entnahmeprozesse

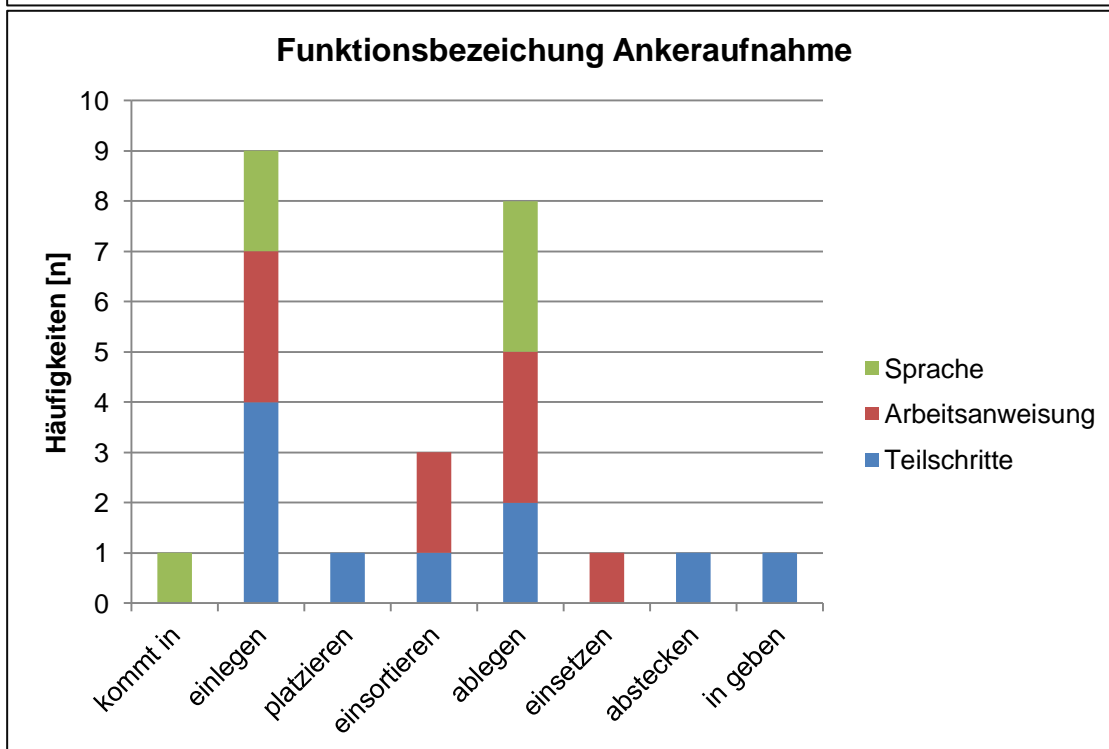
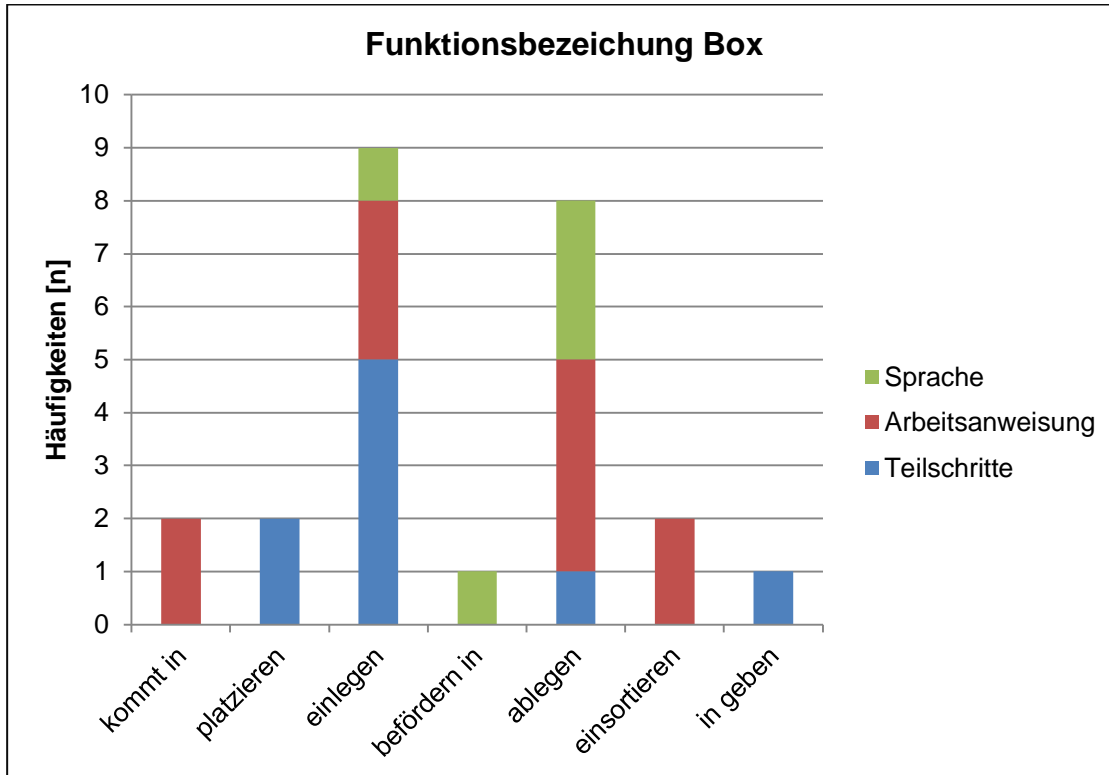


Abbildung 62 Vergleich Ankeraufnahme und Box