

HARTMUT WANDKE, RICHARD OED, EDUARD METZKER, MARKUS VAN BALLEGOOY UND JULIA NITSCHKE
 INSTITUT FÜR PSYCHOLOGIE, HUMBOLDT-UNIVERSITÄT ZU BERLIN UND DAIMLERCHRYSLER AG, FORSCHUNG INFORMATIONSTECHNOLOGIE ULM

DIE ENTWICKLUNG VON USER INTERFACES ALS ARBEITSWISSENSCHAFTLICHER PROZESS UND SEINE UNTERSTÜTZUNG DURCH SOFTWARE-WERKZEUGE

· *Benutzungsschnittstelle* · *Assistenzsysteme* · *Benutzbarkeit* · *Entwicklungsprozess*

ZUSAMMENFASSUNG

In einem BMBF-Leitprojekt zur Mensch-Technik Interaktion in der Wissensgesellschaft - Elektronische Multimediale Bedien- und Service-Assistenz (EMBASSI)¹⁾ - werden intranet-basierte Software-Werkzeuge entwickelt, die den Entwicklern von Assistenzsystemen und anderen interaktiven Systemen helfen sollen, erfolgreich ein aufgaben- und benutzerorientiertes Vorgehen zu praktizieren. Es wird der aktuelle Stand bei der Entwicklung von zwei Komponenten beschrieben: ProUSE unterstützt Organisationen, die an der Entwicklung von User Interfaces beteiligt sind, bei der Analyse, der Spezifikation, dem Entwurf, der Gestaltung, der Bewertung und Entwicklung von Benutzungsoberflächen und Dialogsystemen. GUIDEAS ist ein präskriptives Vorgehensmodell zur Entwicklung von Assistenzsystemen. Die Grundidee beider Komponenten besteht darin, den Entwicklern von User Interfaces im allgemeinen und von EMBASSI-typischen Assistenzsystemen im speziellen arbeitswissenschaftliche Erkenntnisse in Form von hypermedial aufbereiteten Informationstexten, Datenbankeninhalten, Erfahrungsberichten, Methoden, Tools & Templates u. a. zur Verfügung zu stellen und ihnen zugleich eine arbeitswissenschaftlich orientierte Vorgehensweise vorzuschlagen.

DESIGNING USER INTERFACES AS AN ERGONOMIC PROCEDURE SUPPORTED BY SOFTWARE TOOLS

· *user interface* · *assistant systems* · *usability* · *design process*

SUMMARY

In a project called „Electronic Multimedial Operation and Services Assistance“ (German acronym: EMBASSI) are software-tools developed that are based on intranet technology. This project is funded within a national research programme on Human-Machine Interaction in Knowledge Society by the German Ministry of Higher Education and Research. The tools aim at developers of assistance systems and other interactive systems. The tools should support the developers in successfully applying a task- and user-oriented procedure. In this paper we describe the recent status of two components: ProUSE supports organisations involved in the development of user interfaces, especially in stages like analysis, specification, draft, design, evaluation, and implementation. GUIDEAS is a prescriptive process model for the development of assistance systems. The main idea of both components is to provide developers with specially prepared ergonomic knowledge in terms of hypermedia material like information texts, data bases, best practice reports, methods, tools and templates. The tools also suggest to follow an ergonomically-oriented procedure in the user interface development process.

¹⁾ DAS PROJEKT EMBASSI WIRD VOM BMFT UNTER DEM KENNZEICHEN 01IL904I GEFÖRDERT.

1 EINLEITUNG

Technische Systeme werden immer komplizierter. Die Erschließung neuer Anwendungsfelder, Miniaturisierung, Interaktivität, Vernetzung und der Wettbewerb zwischen den Anbietern von zunehmend komplexeren Diensten führen in allen Lebensbereichen dazu, dass immer mehr Personen mit komplizierten Systemen konfrontiert werden, für deren korrekte Benutzung sie kein ausreichendes Wissen besitzen. Eine möglichst einfache und sichere Interaktion zwischen Benutzer und Gerät gewinnt immer mehr an Bedeutung. Während bei professionellen Anwendungsgebieten Benutzer durch spezielle Trainingskurse Wissen z. B. zu SAP-Programmen oder zur Steuerung einer Werkzeugmaschine erwerben können, sind im Alltagsbereich Personen, die sich z. B. ein neues Mobiltelefon gekauft haben, meist auf sich selbst gestellt. User Interface Design rückt daher bei der Entwicklung technischer Systeme stärker als bisher in den Mittelpunkt. User Interface Design erfordert eine Reihe von Kenntnissen über die Benutzer, über menschliche Informationsverarbeitung und Handlung, über die technischen Möglichkeiten der Gestaltung und über Methoden und Vorgehensweisen beim Design. Diese Kenntnisse sind oft verteilt, schwer zugänglich oder bei den Entwicklern interaktiver Systeme nicht vorhanden. Hier soll ein hypermediales Unterstützungssystem Abhilfe schaffen, das von den Autoren im Rahmen des EMBASSI-Projekts geschaffen wird und dessen Inhalte und Funktionsweisen Gegenstand dieses Artikels sind. Während die generelle Gestaltungsfragen von der DaimlerChrysler-Forschung im Modul ProUSE (**PRO**cess **I**ntegrated **U**sability **E**ngineering **E**nvironment) behandelt werden, sind die Arbeiten an der Humboldt-Universität in dem Modul GUID-EAS (**G**uidance **I**n **D**esigning **A**ssistance) auf assistenz-spezifische Fragen gerichtet.

2 USER INTERFACE DESIGN: VON DER KUNST ZUR TECHNOLOGIE

User Interface Design ist eine komplexe Aufgabe, die bei der Entwicklung von technischen Systemen schon immer eine wichtige Rolle gespielt hat, wenngleich sie als eigenständiger Teil der Entwicklung technischer Systeme bis vor wenigen Jahren selten in Erscheinung getreten ist. Allerdings war das User Interface oft nur ein Nebenprodukt der Entwicklung von technischen Systemen. Bekanntestes Beispiel ist wohl die

Sholes- oder QWERTY-Tastatur (im deutschen QUERTZ), bei der die Anordnung der Tasten der Mechanik der Typenhebel und nicht den Erfordernissen menschlicher Hand- und Fingerbewegungen folgt. Ein anderes Beispiel aus der Geräteentwicklung ist die anfangs bevorzugte Anbringung von User Interface Elementen, z.B. Ein- und Ausschaltern, auf der Rückseite (und damit nahe am Netzteil, aber weit weg vom Benutzer) von Computern, Bildschirmen und Druckern.

Auch in der Entwicklung interaktiver Software war zunächst Programmcode zur Interaktion mit Benutzern (z. B. Eingabe von Daten, Anzeigen von Ergebnissen) unmittelbar gemischt mit Programmcode zu Berechnungen oder anderen internen Datenverarbeitungsprozessen. Das User Interface entstand gewissermaßen nebenbei während der Programmierung. Ein separates User Interface hatte in dem sich entwickelnden Software-Engineering zunächst den Vorteil aller Modularisierungen: Es war leichter zu testen, zu verändern und wiederzuverwenden. Erst später kam ein weiterer Vorteil hinzu: ein solch separat erstelltes User Interface kann auch eine bessere Benutzung des Systems (Usability) ermöglichen. User Interface Design wurde zu einem eigenen Spezialgebiet des Software-Engineering. Software-Entwickler spezialisierten sich nicht nur auf bestimmten Anwendungsgebieten oder in Basistechnologien wie Compiler, Datenbanken oder Protokolle, sondern auch auf dem Gebiet des User Interface Design. Bereits zu Beginn der 80er Jahre machte das User Interface ca. 30-35 % der Codezeilen eines Anwendungsprogramms aus (Smith & Mosier 1984). Anfang der 90er Jahre waren es schon ca. 50% (Myers & Rosson 1992). Heute dürften die Funktionen des User Interface oft deutlich mehr als 75% des Programmcodes für interaktive Systeme ausmachen.

User Interface Design kann auf sehr verschiedener Basis betrieben werden: als Kunst, als Handwerk, als Technologie und als automatischer Prozess. User Interface Design als Kunst (hier im Sinne von *Art* gebraucht, vgl. z. B. Fähnrich 1987) geht davon aus, dass es eine kreative - wenn auch durch sehr viel Erfahrung gespeiste - Aktivität ist, ein User Interface zu entwickeln. Ziel einer solchen Entwicklung ist es, ein besonders originelles und unverwechselbares User Interface zu erzeugen. Dafür gibt es keine direkt anwendbaren Regeln, wohl aber die bekannten Kreativitätstechniken, zu denen die Analogiebildung gehört, wie sie bei den

Konzepten „User Interface als Theaterbühne“ (Laurel 1990) oder User Interface als Räume in einem Gebäude (Henderson & Card 1987) benutzt wurde.

Handwerk ist ebenso erfahrungsbasiert, jedoch viel stärker regelorientiert. User Interfaces werden nach bekannten und bewährten Prinzipien entworfen, zu denen Spezialisten in einem längeren Lernprozess Wissen erworben haben. Ein Beispiel ist die systematische Weiterentwicklung der Direkten Manipulation und ihrer Anwendungsfelder durch Shneiderman (1997) und die Arbeiten von Nielsen (1993) und Nielsen & Mack (1994). Vertreter dieser Richtung suchen ständig neue Anwendungsgebiete für die einmal entdeckten Gestaltungsprinzipien, ebenso wie sie um die Vervollkommnung der Regeln bemüht sind. Zum handwerklichen Ansatz gehört auch die Einbeziehung der Kunden, d. h. der zukünftigen Benutzer, in den Entwicklungsprozess, sei es durch einen partizipativen Entwicklungsprozess - insbesondere im sozio-technischen Systemansatz (Ulich 1988a) bzw. als praktizierte industrielle Demokratie (Emery & Thorsrud 1982) - oder durch gründliches User Testing in den entsprechenden Laboren (Nielsen 1994).

Der Übergang zur Technologie ist fließend. Regeln werden gesammelt, geordnet und gewichtet. Es entstehen Standards und Vorschriften (vgl. die Übersicht bei KAN, 1997), Guidelines (z. B. Smith & Mosier 1984; Brown 1988; VDI 1990; Mayhew 1992; Tognazzini 1993), Style Guides (z. B. SUN 1989; IBM 1991; Apple 1992; Microsoft 1992). Style Guides können ergänzt werden durch Entwicklungsumgebungen für User Interfaces (Forbrig et al. 1993; Reiterer 1994), durch standardisierte GUI-Widgets (Myers 1999) und durch Software-Werkzeuge zur Arbeit mit den Guidelines (Hüttner u. a. 1995; Vanderdonck 1999). User Interface Entwicklung wird dadurch sehr einfach. Allerdings steigt die Gefahr der unreflektierten Werkzeuganwendung im Vertrauen darauf, dass dort Usability gewissermaßen „eingebaut“ ist.

Der tatsächliche (teilweise) Einbau wird bei einem automatischen User Interface Design angestrebt. Solche mit KI-Techniken zur Benutzer- und Kontextmodellierung versehenen User Interfaces sollen sich automatisch dem Benutzer und der Situation anpassen. Avatare und andere Agententechnologien sind Beispiele für diesen Ansatz, der vielfach erst noch erprobt wird. (Nass et al. 1995; Sunblad & Sunblad 1998; Uchyigit

et al. 1999; Van Mulken et al. 1999; Nissler 1999).

Keine der Herangehensweisen führt, wenn sie für sich allein praktiziert wird, zu einem befriedigenden Ergebnis. Eine Hauptaufgabe beim User Interface Design ist es, aus den genannten Ansätzen die richtige Auswahl und Kombination zu wählen, die für das konkrete User Interface zu den besten Ergebnissen führt. Genau dies ist das Anliegen, das mit der Ergonomie-Arbeitsumgebung ProUSE und dem Vorgehensmodell GUIDEAS, die in diesem Artikel vorgestellt werden, verfolgt wird.

3 FORMEN DER UNTERSTÜTZUNG DES ENTWICKLUNGSPROZESSES

Es gibt zahlreiche wissenschaftliche Veröffentlichungen zum Thema „Methoden und Vorgehensweisen zur Entwicklung von Benutzungsoberflächen“. Die Gesamtheit wird vornehmlich unter den Stichworten „Usability Engineering“ (Nielsen 1997) und „User/Usage Centered Design“ (Constantine & Lockwood 1999) zusammengefasst. Ziel dieser Methoden und Vorgehensweisen ist die ganzheitliche und durchgängige Betrachtung ergonomischer Aspekte während der Systementwicklung, so dass während der Gesamtlaufzeit eines Projekts das Thema „Usability“ ausreichend berücksichtigt wird. Es sollen sowohl Auftraggeber und Auftragnehmer, als auch die späteren Benutzer der Systeme frühzeitig und kontinuierlich am Gesamtprozess beteiligt werden.

Ziele sind

- die Verbesserung der User Interface-Entwicklung im Rahmen von Systementwicklungsprojekten sowie letztendlich
- die deutliche Steigerung der Qualität von Oberflächen bezüglich Benutzerfreundlichkeit und Aufgabenangemessenheit.

Generell wird *Usability Engineering* (UE) als eine Grundmenge an Aktivitäten angesehen, die idealerweise im Laufe eines Produktlebenszyklus angewandt werden (Nielsen 1997). Der Schwerpunkt von Usability-Aktivitäten liegt dabei in frühen Phasen vor dem eigentlichen Systementwurf. Wesentliche Aktivitäten sind Benutzer-, Funktions- und Aufgabenanalyse, Analyse von existierenden Altsystemen, Definition von Usability-Zielen, Generieren paralleler Gestaltungsvarianten, Benutzerbeteiligung, Anwendung vorhandener Gestaltungsregeln, Prototyping, empirische Usability-Tests und Evaluationen, iteratives Design des Endprodukts sowie Abfrage von Benutzerfeedbacks in Feldversuchen.

Usability Engineering wird sinnvollerweise im Kontext eines typischen Systementwicklungsprozesses angewendet. Oben genannte Aktivitäten des UE werden den verschiedenen Systementwicklungsphasen als so genannte „User Interface Design Tasks“ zugeordnet (Mayhew 1992). Auf einer etwas abstrakteren Ebene wird hier zwischen den Entwicklungsphasen „Projektvorbereitung“, „Anforderungsanalyse“, „Ent-

wurf“, „Entwicklung“ und „Installation/Inbetriebnahme“ unterschieden.

Dieses grundlegende Vorgehensmodell wurde 1996 von der DaimlerChrysler-Forschung aufgegriffen, um es im eigenen industriellen Umfeld praktisch anzuwenden und zu erweitern. Daraus entstand 1999 ein erweitertes Prozessmodell für Usability Engineering. Die einzelnen Systementwicklungsphasen werden durch spezielle Ergonomiebetrachtungen, so genannte „Prozessschritte“ oder „Usability Tasks“, angereichert. Das derzeitige Referenzmodell ist in Bild 1 dargestellt.

4 UNTERSTÜTZUNGSWERKZEUGE

Heute sind über 130 Werkzeuge (Programmierbibliotheken, User Interface Management Systems, User Interface Builder, Icon Builder, Application Frameworks, Toolkits) auf dem Markt (User Interface Tools 2000; Myers 1997), die alle den Zweck verfolgen, den Programmierer bei der Gestaltung und Programmierung der Benutzungsschnittstelle zu unterstützen bzw. in einigen Fällen ihm die Programmierarbeit ganz abzunehmen.

Aber die Gestaltung der Benutzungsschnittstelle beschränkt sich nicht, wie oft fälschlicherweise angenommen, auf das reine Programmieren, sondern ist in den komplexen Prozess des Usability Engineering eingebunden. Praktisch keines dieser 130 Werkzeuge vermittelt das nötige Wissen und unterstützt Entwickler und Programmierer in allen wich-

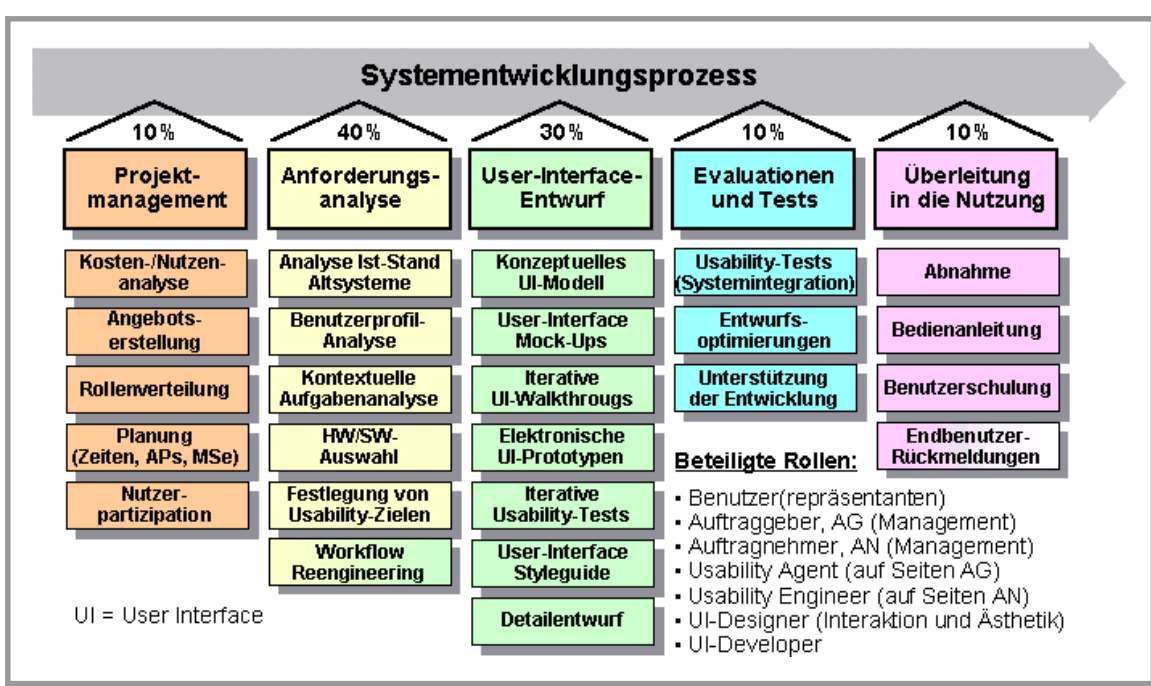


Bild 1:
Ein Referenzmodell für den Usability Engineering Prozess

tigen Phasen dieses Usability Engineering Prozesses.

In der Literatur wird die Notwendigkeit von Werkzeugen beschrieben, die in Anlehnung an CASE-Tools mit dem Begriff CAUSE-Tools (Computer Aided Usability Engineering) eingeführt wurden. Nielsen (1997, S. 264) kommt in seinem Buch „Usability Engineering“ zu dem Schluss: „There are multiple tasks in the usability engineering lifecycle that could be performed more efficiently with computerized tools, there are almost no such tools commercially available...“.

Neben solchen mehr allgemeinen Betrachtungen von Vorgehensweisen gibt es auch eine Reihe von konkreteren Vorgehensmodellen. Ein Beispiel ist *TASK* (Beck & Janssen 1993, Technik der aufgaben- und benutzerangemessenen Software-Konstruktion). Hier wird der Forderung nach stärkerer Aufgabenorientierung und angemessener Beteiligung der Benutzer Rechnung getragen. Ferner werden soziale, organisatorische und technische Anforderungen berücksichtigt.

Ein elektronisches Unterstützungs- und Informationssystem sollte also neben der eigentlichen Aufgabe des Softwareentwurfs ebenso all jene Bereiche abdecken, die damit mittelbar oder unmittelbar verbunden sind. In der Software-Entwicklung kommt solchen so genannten Sekundäraufgaben (Ulich 1988) eine besondere Bedeutung zu. Durch ein hohes Maß an Aufgabenunsicherheit und Aufgabeninterdependenz besteht gerade hier ein größerer Aufwand an Planung, Koordination und projektinterner Kommunikation (Brodbeck 1996). Die im Rahmen des Usability Engineering verstärkte Benutzerbeteiligung kann zudem die Bewältigung zusätzlicher Sekundäraufgaben erforderlich machen und den Entwicklungsprozess erschweren und verzögern (Selig 1986). Diesen Schwierigkeiten kann durch eine Förderung von Kooperation und Kommunikation über die Bereitstellung von adäquaten Unterstützungswerkzeugen entgegen gewirkt werden.

5 PROBLEME IM ENTWICKLUNGSPROZESS

Werden Entwickler von Bedienschnittstellen gefragt, wie ihr Entwicklungsprozess konkret aussieht und wo und in welcher Form sie Unterstützung benötigen, kommt man zu erstaunlichen Ergebnissen.

Die meisten Probleme, die von den Entwicklern genannt werden, betreffen organisatorische Unzulänglichkeiten im Entwicklungsprozess (Oed et al. 2001). Die weitaus meisten Befragten geben Probleme an, die das Projektmanagement betreffen. Häufig werden aber auch fehlende oder unklare Vorgaben hinsichtlich der Anforderungen an das zu entwickelnde System bemängelt. Kritisiert werden auch eine mangelnde oder zähe Kommunikation, ungenügende Absprachen zwischen dem Management und den Entwicklern sowie Kooperationsprobleme mit abhängigen Teilprojekten und Unterbeauftragungen. Gewünscht wird häufig eine sorgfältigere Dokumentation während des Entwicklungsprozesses und ein leichter Zugriff auf entsprechende Ressourcen. Dies betrifft sowohl Informationen über die aktuelle Projektplanung und -koordination sowie die Dokumentation der einzelnen Phasen des Entwicklungsprozesses als auch Erfahrungen aus vorangegangenen Projekten.

Da Entwicklern von Bedienschnittstellen und Benutzungsoberflächen häufig auch das nötige Grundwissen über die software-ergonomische Gestaltung und die Vorgehensweise fehlt, wird auch hierzu Unterstützung gewünscht. Dabei wird aber der Wissensabruf on demand gegenüber organisierten Schullehrangeboten und Tutorien stark bevorzugt.

(Moran 1981, S. 4). Damit wird deutlich, dass sich User Interface und Anwendung zwar unter software-technischen Aspekten trennen lassen, nicht aber unter Benutzungsaspekten. Dem folgt van der Veer (1990, S. 9), der darauf hinweist, dass im Software-Engineering der Begriff User Interface anders gefasst wird als in der Software-Ergonomie. Er schreibt: „User interface (in the restricted sense of software engineering): a system component that manages the dialogue between an application system and an user. It enables interaction on the delegation of tasks to the application systems and it provides facilities for metacommunication about interaction.“ Weiter heißt es: „User interface (in cognitive ergonomic sense): the complete and correct relevant knowledge of the combination of the user interface in the restricted sense and the application interface.“ (van der Veer 1990, S. 9). Obwohl die Beschränkung auf Wissen nicht ganz korrekt ist, denn es kennzeichnet nur den begrifflichen, aber nicht den physische und perzeptiven Kontakt mit dem System, ist diese klare Unterscheidung von van der Veer sehr hilfreich. Ganz in diesem Sinne definiert auch Wandmacher (1993), der die Begriffe *Benutzungsoberfläche* und *Benutzungsschnittstelle* klar voneinander trennt: „Unter Benutzungsoberfläche versteht man alle Einheiten, Formen und Techniken, durch

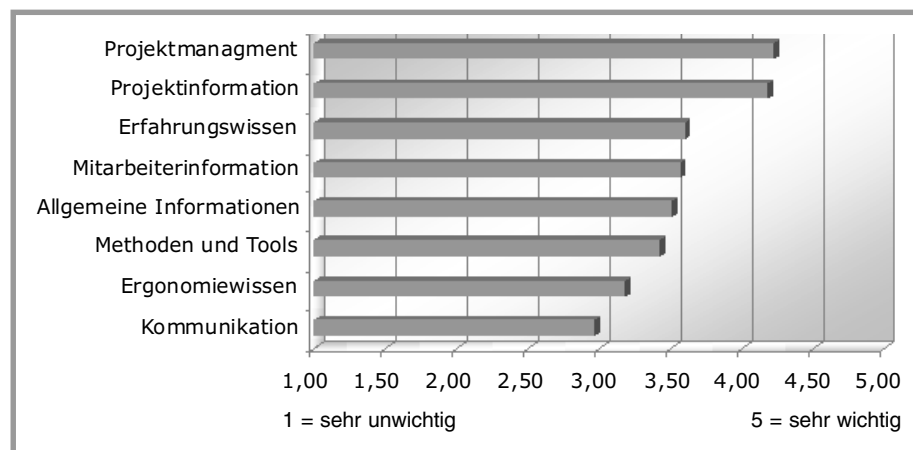


Bild 2: Bewertung der Unterstützungselemente nach Unterstützungsbereichen (n=16 Entwickler aus vier verschiedenen Unternehmen)

6 USER INTERFACE EBENEN

Obwohl der Begriff User Interface sehr häufig benutzt wird, bleibt oft unklar, was der jeweilige Autor oder Entwickler darunter versteht. Eine umfassende und zugleich für den Designprozess hilfreiche Definition hat Moran vorgeschlagen: „The user interface of a system consists of those aspects of the system that the user comes in contact with - physically, perceptually, or conceptually.“

die Benutzer mit dem Computersystem kommunizieren. Die Benutzungsoberfläche ist der zu dem Computersystem gehörende und für die Benutzer erkennbare oder sichtbare Teil der Benutzungsschnittstelle. Benutzungsoberfläche ist damit eine Teilmenge der Benutzungsschnittstelle.“ (Wandmacher 1993, S. 4).

Im Kontrast dazu wird im Software-Engineering eine Unterteilung in verschiedene

Software-Komponenten vorgenommen, so wird z. B. beim weit verbreiteten Seeheim-Modell (Enderle 1984) in eine Präsentationskomponente, eine Dialogsteuerungskomponente und in eine Applikationsschnittstelle unterteilt. Die eigentliche Applikation ist davon separiert. Ein ähnliches Modell ist das IFIP-Modell (Dzida 1983).

In der vorliegenden Arbeit zur Unterstützung von Software-Ingenieuren bei der benutzerorientierten Entwicklung von User Interfaces gehen wir von dem erweiterten User Interface Begriff sensu Moran aus. Die Funktionalität eines Systems (Applikation) wird also nicht ausgeblendet, sondern ist ein wichtiger Bestandteil bei der Entwicklung des User Interface. Damit befinden wir uns übrigens in Übereinstimmung mit zahlreichen Autoren, die im Rahmen der Diskussion von Kriterien der software-ergonomischen Gestaltung den funktionalen Aspekten (z. B. Aufgabenangemessenheit in der DIN EN ISO 9241-10) einen größeren Stellenwert einräumen als interaktionsbezogenen Aspekten wie Steuerbarkeit oder Individualisierbarkeit (Koch et al. 1991; Oppermann et al. 1992).

Sprachlich ist diese Orientierung an dem Übergang vom veralteten Begriff *Benutzerfreundlichkeit* zum gegenwärtig auch in der deutschsprachigen Literatur als terminus technicus benutzten Begriff *usability* festzumachen. Gleichzeitig gewinnt das Kriterium *usability* im Rahmen der Qualitätssicherung bei der Software-Entwicklung zunehmende Bedeutung (DIN EN ISO 9241-11; DIN EN ISO 13407).

User Interfaces sensu Moran bestehen aus mehreren Ebenen, die jeweils unterschiedliche Gestaltungsfragen aufwerfen. Tabelle 1 zeigt die User Interface Komponenten und einige beispielhaft zugeordnete Gestaltungsfragen.

Diese Ebeneneinteilung findet sich in Variationen auch bei anderen Autoren (z. B. Nielsen 1986), die insbesondere linguistische Beschreibungsmittel einsetzen, um die Interaktionsebene und die Ebenen der physischen Komponente weiter zu differenzieren. Da die in früheren Jahren dominierende (schrift-)sprachliche Interaktion jedoch zunehmend durch multimediale Komponenten

ergänzt wird, haben wir auf diese Differenzierungen verzichtet.

Ein solches Ebenenmodell kann im normativen Sinne zur Gestaltung eines User Interface herangezogen werden. Dabei spielen zwei Prinzipien eine Rolle:

1. Die Gestaltung soll von oben nach unten erfolgen.
2. Die Festlegungen sollen in den einzelnen Ebenen so weit wie möglich ausdifferenziert werden, bevor man zur nächsten Ebene übergeht.

Formale Modellierungstechniken wie die Task Action Grammar (Payne & Green 1986), GOMS-Modelle (Card et al. 1983) und die Cognitive Complexity Theorie (Kieras 1988) können eingesetzt werden, das User Interface ebenengerecht zu spezifizieren. Gleichzeitig wird dieses Top-Down-Design jedoch immer durch einen Bottom-Up-Ansatz zu ergänzen sein. So ist z. B. meist schon vor den ersten Gestaltungsentscheidungen aus technischen, ökonomischen, organisatorischen und aufgabenbezogenen Überlegungen bekannt, welche Hardware eingesetzt werden kann und welche nicht.

Die Top-Down-Orientierung beruht auf der Tatsache, dass die oberen Ebenen eines User Interface eine stärkere Bedeutung für die Usability eines Systems haben als die unteren Ebenen. So ist z.B. für einen Benutzer ziemlich wichtig, ob ein Web-Browser komplette Seiten abspeichert oder ob die Grafiken manuell gesondert abgespeichert werden müssen. Weniger wichtig ist dagegen, ob der Back-Button des Browsers rund oder eckig ist.

Wenn User Interface Designer beim Top-Down-Vorgehen unterstützt werden sollen, so wird deutlich, dass es relativ schwierig ist, für die wichtigen oberen Interface-Ebenen allgemeingültige Gestaltungsvorschläge zu geben. Ohne genaue Kenntnisse über die Benutzer (ihre Vorwissen, ihre Ziele, ihre Fähigkeiten und Fertigkeiten, ihre Rolle in einer Organisation usw.) sind kaum Aussagen möglich, die Software-Entwickler bei den Gestaltungsentscheidungen auf den oberen Ebenen unterstützen. Dies trifft ebenso auf die Aufgaben zu. Ohne die konkreten Aufgaben zu kennen, können kaum Hinweise zur Gestaltung formuliert werden. So ist es z.B. nicht möglich, eine generelle direkte Empfehlung für die Funktionen der Störungsdiagnose an einer Werkzeugmaschine

Tabelle 1: Komponenten und Ebenen von User Interfaces nach Moran (1981)

User Interface Komponenten	User Interface Ebenen	Gestaltungsentscheidungen über ...
Konzeptuelle Komponente	Aufgabenebene	... Funktionsteilung Mensch-Computer, Festlegung von Aufgaben, die mit dem System bearbeitet werden sollen, Bildung von Teilaufgaben und Aufgabenhierarchien, Automatisierungsgrad.
	Semantische Ebene	... Systemobjekte und -operationen und Operationen der Benutzer, über Sichten auf das System und über die Verwendung von Metaphern.
Kommunikative Komponente	Syntaktische Ebene	... Regeln zur Anwendung von Operationen auf Systemobjekte, über Systemzustände und Constraints.
	Interaktionsebene	... Merkmale der Interaktion (z. B. Initiative, Direktheit, Granularität, Rückmeldung), Auswahl und Kombination von Interaktionstechniken, wie Commando-, Menü-, Formular- und direkt-manipulative Techniken.
Physische Komponente	Medienebene	... Modalitäten des Informationsaustausches und ihre Kombination (visuell, akustisch, haptisch), z. B. Zeigegeesten, Spracheingabe.
	Räumliche Ebene	... Formate der Informationsausgabe und -eingabe, z. B. Festlegungen zu Fenstergrößen und -anordnungen, Positionen von Interaktionselementen auf dem Bildschirm.
	Kodierungsebene	... Zeichen, Symbole und Icons, Merkmale akustischer Signale, z. B. Form, Größe, Farbe und Bewegung von Objekten, Festlegung lexikalischer Elemente.
	Hardware-Ebene	... Ein- und Ausgabegeräte, z. B. Größe und Auflösung des Bildschirms, Tastatur-, Maus- und Touchfeldeigenschaften, Geräte für Virtual und Augmented Reality

zu geben. Dagegen sieht es bei den Gestaltungsentscheidungen auf den unteren User Interface Ebenen besser aus: Da es hier vor allem um die Anpassung von Systemeigenschaften an perzeptiv-motorische und elementare kognitive Eigenschaften von Benutzern geht, bei denen es wenig interindividuelle Varianz und nur eine geringe Aufgabenabhängigkeit gibt, lassen sich direkt anwendbare Gestaltungsregeln zur Unterstützung des Designprozesses formulieren. So sind z.B. Empfehlungen zur Farbgestaltung (Welche Farben kontrastieren gut? Was ist die symbolische Bedeutung von Rot?) für Geldautomaten, Fahrerassistenzsysteme und natürlich auch Werkzeugmaschinen zutreffend. Allerdings muss beachtet werden, dass es auch auf den unteren User Interface Ebenen Ausnahmen von allgemeingültigen Gestaltungsempfehlungen geben kann. Solche Ausnahmen müssen z.B. bei behinderten Benutzern berücksichtigt

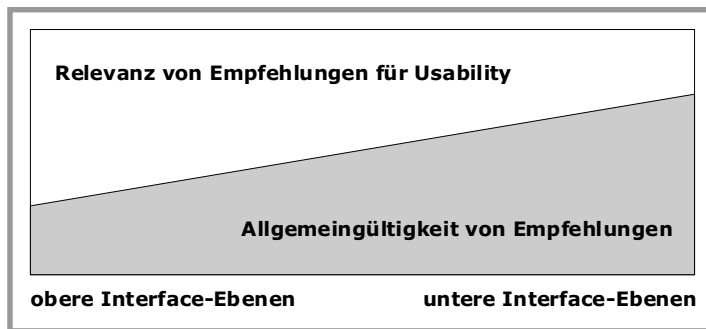
DANN-Regeln). Um sie geben zu können, müssen die Entwickler zunächst spezifizieren, um welche Benutzer es sich handelt, welche Aufgaben in welchen Kontexten diese Benutzer ausführen und welche Ziele mit dem Assistenzsystem verfolgt werden sollen. Diese Angaben werden genutzt, um mit den WENN-Teilen der intern gespeicherten Gestaltungsempfehlungen verglichen zu werden. Um die Anzahl der WENN-DANN-Regeln zu begrenzen und so eine schnelle Verfügbarkeit der Unterstützung zu erreichen, beschränkt sich das Vorgehensmodell GUIDEAS zum einen auf Assistenzfunktionen und berücksichtigt zum anderen nur die im EMBASSI-Projekt untersuchten Anwendungsfelder: Bedienen technischer Systeme im Privathaushalt, von Terminalsystemen in der Öffentlichkeit und bei der Kfz-Benutzung. Bei den Benutzermerkmalen wird das gesamte Spektrum nicht-professioneller Benutzer betrachtet. Die DANN-

tun haben, und alle weiterführenden Gestaltungsfragen werden dann nicht mehr von der GUIDEAS-Komponente, sondern von der ProUSE-Komponente behandelt.

Eine weitere Form der Unterstützung ist sowohl in ProUSE als auch im Vorgehensmodell GUIDEAS implementiert: Wenn es nicht möglich ist, eine direkte Gestaltungsempfehlung zu geben, die sich inhaltlich auf die software-ergonomischen Aspekte eines User Interface bezieht, dann werden methodische Empfehlungen gegeben. Diese methodischen Empfehlungen sind in ProUSE auf das prinzipielle Vorgehen (Projektvorbereitung, Anforderungsanalyse, Prototyping, Evaluationen und Tests) gerichtet, wobei den Entwicklern Informationen und Tools für das Durchführen und das Management dieser Aktivitäten zur Verfügung gestellt werden. Im Vorgehensmodell GUIDEAS werden empirische und experimentelle Methoden speziell für die Anforderungsanalyse und für die Evaluation von User Interfaces (vom konzeptuellen Prototypen bis zum praktisch eingesetzten System) angeboten. Auch hier gibt es Tools für die Durchführung und Auswertung von empirischen Analysen.

Durch die Integration von ProUSE und GUIDEAS kann der gesamte Designprozess für User Interfaces von Assistenzsystemen von der Zieldefinition bis zum „letzten Schliff“ der Oberfläche unterstützt werden, ebenso die Analyse der Benutzungsprozesse nach der Einführung der Systeme in ein Anwendungsgebiet.

Bild 3:
Veranschaulichung des Konflikts zwischen Allgemeingültigkeit und Relevanz von Gestaltungsempfehlungen



werden. Das Bild 3 fasst die Problematik bei der Unterstützung des Designprozesses für User Interfaces zusammen.

Um dem Konflikt zwischen Relevanz und Allgemeingültigkeit gerecht zu werden, haben wir bei der Konzeption der Ergonomie-Arbeitsumgebung ProUSE und des Vorgehensmodells GUIDEAS folgenden Ansatz gewählt: In ProUSE werden vor allem die unteren Interface-Ebenen berücksichtigt. Sie können in zahlreichen Anwendungskontexten eingesetzt werden, so auch beim Design von Assistenzsystemen. Entwickler von Assistenzsystemen, die etwas über die Größe, Anordnung und Farbe von Menüs erfahren wollen, erhalten in ProUSE direkte Empfehlungen. Dies gilt z. B. auch für Fragen der Sprachausgabe oder den Oberflächen-Merkmalen von Avataren wie Geschlecht, Körper, Gesicht, Mimik, Gestik, Outfit. Das Vorgehensmodell GUIDEAS beschränkt sich dagegen auf Gestaltungsempfehlungen für die oberen Interface-Ebenen. Da keine allgemeingültigen Empfehlungen gegeben werden können, sind die Empfehlungen konditional formuliert (in Form von WENN-

Teile beziehen sich nur auf die oberen Ebenen des User Interface: Es werden Empfehlungen zu folgenden Gestaltungsfragen gegeben:

- Welche psychischen Funktionen sollen unterstützt werden?
- In welchem Grad soll die Assistenz erfolgen (von der reinen Information über Bedienschritte bis zur automatischen Ausführung derselben ohne Beteiligung des Benutzers)?
- Wer soll die Initiative bei der Assistenz haben? Soll die Assistenz aktiv erfolgen oder erst auf Anforderung des Benutzers?
- Soll der Inhalt der Unterstützung, der Grad und die Initiative der Assistenz festgelegt sein oder soll der Benutzer (adaptierbar) oder das System (adaptiv) die Assistenz verändern können?

Fragen, die mit der Modalität der Assistenz (z.B. bildliche Darstellung von Handlungsschritten versus sprachlichen Hinweisen) zu

7 DIE ERGONOMIE-ARBEITS-UMGEBUNG ProUSE ZUR PROZESSINTEGRIERTEN UNTERSTÜTZUNG VON ENTWICKLERN

Das Ziel der Ergonomie-Arbeitsumgebung ProUSE ist es, einen wesentlichen Beitrag zur zielgerichteten und effizienten Gestaltung und Entwicklung von Dialogsystemen und User Interfaces zu leisten, die höchsten Qualitätsansprüchen bezüglich der Handhabbarkeit (usability) genügen sollen.

ProUSE unterstützt Organisationen, die an der Entwicklung von User Interfaces beteiligt sind, bei der Analyse, der Spezifikation, dem Entwurf, der Gestaltung, der Bewertung und Entwicklung von Benutzungsoberflächen und Dialogsystemen. Dabei wird generell unterschieden zwischen Organisationen, die komplexe Systeme selbst entwi-

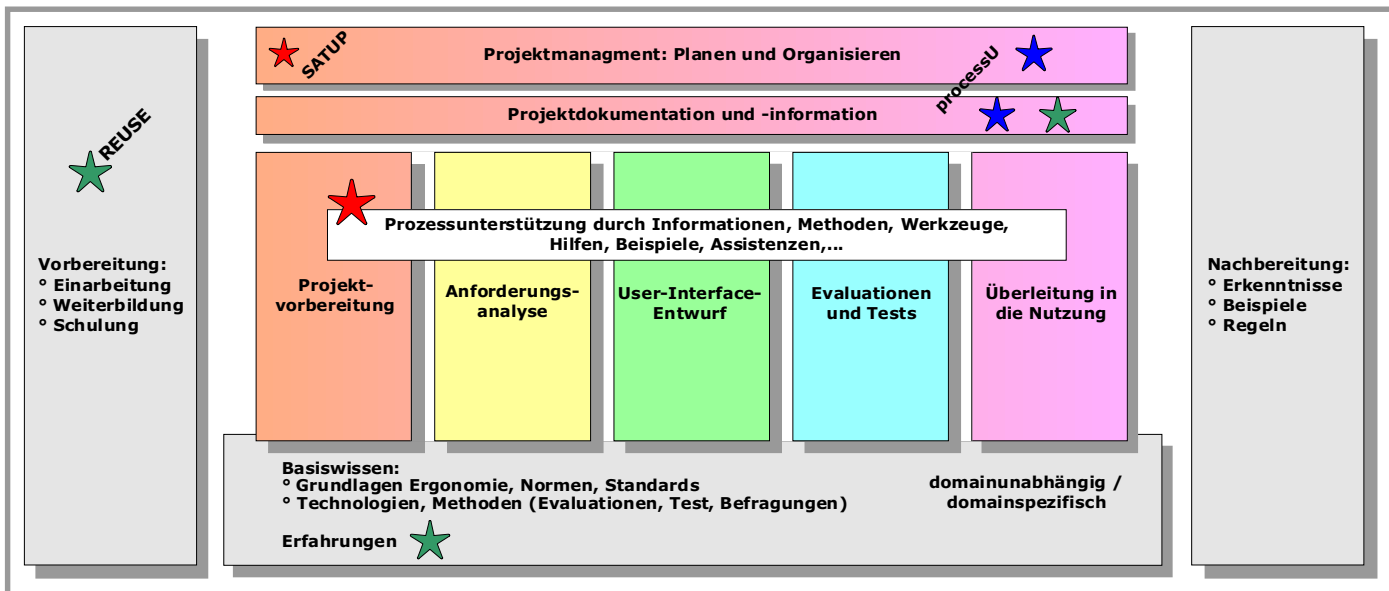


Bild 4: Die fachliche Strukturierung der Ergonomie-Arbeitsumgebung ProUSE

ckeln (typische Entwicklungsorganisation in der Auftragnehmerrolle) und solchen, die im wesentlichen die Entwicklungsarbeiten an andere vergeben (Organisation in der Auftraggeberrolle).

ProUSE stellt Informationen, Werkzeuge und Hilfsmittel zur ganzheitlichen und durchgängigen Entwicklung von User Interfaces nach der Methode des Usability Engineering (UE) zur Verfügung. Bild 1 zeigt das Vorgehensmodell für Usability Engineering, welches dazu als Basis verwendet wird. Es dient als wesentliche konzeptionelle Grundlage für den Aufbau von ProUSE.

Die fachliche Strukturierung von ProUSE (Bild 4) zeigt die wesentlichen Bestandteile des Systems.

Da ProUSE den gesamten Prozess der UI-Entwicklung unterstützen soll, werden Informationen, Methoden und Werkzeuge für alle Entwicklungsphasen beginnend mit der Projektvorbereitung, der Anforderungsanalyse über den UI-Entwurf, Evaluationen und Tests bis zur Überleitung in die Nutzung zur Verfügung gestellt.

Konkrete Ausprägungen dieser Unterstützungswerkzeuge sind:

- Checklisten
- Assistenzen
- Anwendungsbeispiele
- Projektmanagement und Projektdokumentation

- Methoden zu Evaluationen, Benutzertests und Befragungen
- Erfahrungssammlung und Wiederverwendung
- Grundlagenwissen zur Einarbeitung, Schulung, Weiterbildung und zum Lernen bei Bedarf

7.1 Module und Funktionsweise von ProUSE

Der erste Forschungsprototyp der Ergonomie-Arbeitsumgebung besteht aus einzelnen Modulen, die in Form von Java-Applets, HTML-Seiten und Anwendungen zur Verfügung gestellt werden. Über ein gemeinsames Portal sind diese so miteinander verbunden, dass die Entwickler mit ihrem Web-Browser die Module nutzen können. Damit

wird erreicht, dass ProUSE in seiner endgültigen Form auf einem Web-Server laufen kann und somit eine ganze Organisationseinheit unterstützen kann.

Die Module sind im Einzelnen:

- Basiswissen

Hier werden Grundlagen der Gestaltung, der Software-Ergonomie, Normen, Standards und Styleguides übersichtlich und mit vielen Beispielen versehen so dargestellt, dass sich die Benutzer (z.B. neue Mitarbeiter, Interessenten an bestimmten Themen der Gestaltung, ...) informieren, einarbeiten oder selbständig weiterbilden können. Das Basiswissen wird in Form eines interaktiven elektronischen Buches als Web in das Portal integriert.

- SATUP: Setup Assistant for Usability Engineering Processes

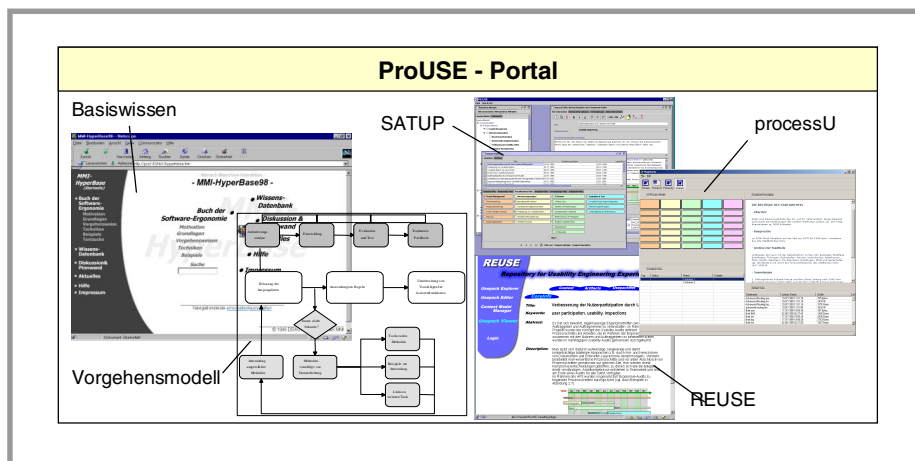


Bild 5: Das ProUSE Portal und die dahinterliegenden Werkzeuge

Wird in einem Projekt ein neuer UE-Prozess aufgesetzt, wird SATUP zur Unterstützung bei Auswahl und projektspezifischem Zugschnitt von Usability Engineering Aktivitäten, Methoden und Werkzeugen (z.B. Usability Maturity Model Assessment) benötigt. Damit kann festgelegt werden, welche Prozessschritte durchlaufen werden und welche UE-Maßnahmen in den praktizierten UE-Prozess aufgenommen werden.

- ♦ REUSE: Repository for Usability Engineering Experiences

Mit REUSE wird die projektübergreifende Erfassung, Organisation und effiziente Wiederverwendung von Prozessprodukten (Methoden, Best Practices, Erfahrungen und weiteren Projektergebnissen) unterstützt. Dazu werden Methoden zur Erfassung und Wiedergewinnung von Erfahrungen entwickelt und mit REUSE in der Ergonomie-Arbeitsumgebung zur Verfügung gestellt. REUSE besteht dazu aus einem Editor zur Erstellung von Erfahrungspaketen, einer Datenbank zur Speicherung der Daten und einer Schnittstelle zur Präsentation und Wiedergewinnung der Information.

- ♦ ProcessU: Prozessunterstützung

Unterstützung bei hochgradig iterativen Entwicklungsprozessen durch einen angepassten UE-Prozess.

- ♦ GUIDEAS Guidance in Developing Assistance

Computerunterstütztes Vorgehensmodell für die Entwicklung von Assistenzsystemen, das im folgenden näher erläutert wird.

7.2 Der Aufbau und die Funktionsweise des Vorgehensmodells GUIDEAS

GUIDEAS soll Entwicklern von Assistenzsystemen in zwei Phasen des Entwicklungsprozesses Unterstützung bieten: In der Phase der Anforderungsanalyse, bei der im wesentlichen ermittelt wird, welchen Assistenzbedarf Benutzer bei der Bedienung eines technischen Systems haben, und in der Evaluationsphase, bei der anhand von Prototypen oder implementierten Assistenzsystemen geprüft werden soll, ob der Assistenzbedarf von Benutzern tatsächlich befriedigt wird und in welchem Umfang dies geschieht.

7.2.1 Phase der Anforderungsanalyse

Wenn Arbeitswissenschaftler gefragt werden, wie denn ein konkretes User Interface gestaltet werden soll, so werden sie in der Regel eine Antwort geben, die mit „Es hängt davon ab, ...“ oder einer ähnlichen Floskel beginnt, oder direkt mit Gegenfragen reagieren. Nur mit der Kenntnis der Aufgaben und Ziele des Benutzers, sowie seines Vorwissens, seiner Erfahrungen, persönlichen Arbeitsstile, aber auch anderer personenbezogener und sozio-biografischer Merkmale bis hin zu möglichen Behinderungen, lässt

sich diese Frage beantworten. Liegen diese Kenntnisse vor, so wird die Antwort selbst auf der Grundlage von Standards, „gesicherten arbeitswissenschaftlichen Erkenntnissen“, Modell- und Theorieanwendungen, persönlicher Erfahrung und weiteren Grundlagen erfolgen. Oft wird die Antwort aber lauten, dass man die Frage nur mit Hilfe von Analysen, Benutzerbeteiligungen, vielleicht sogar empirischen oder experimentellen Untersuchungen beantworten kann. Arbeitswissenschaftler verfügen in der Regel über das methodische Know-how, um solche Analysen durchführen zu können.

Das skizzierte Vorgehen ist auch für die im EMBASSI-Projekt tätigen Psychologen zutreffend. Durch die Beteiligung an den Entwicklungsarbeiten entsteht jedoch zusätzliches spezifisches Wissen über aufgabenorientierte und benutzergerechte Assistenzsysteme. Dieses neu entstandene Wissen soll gemeinsam mit dem ergonomischen Grundwissen dokumentiert und nachnutzbar gemacht werden, so dass auch Entwickler von Assistenzsystemen davon profitieren können, die nicht in dem aktuellen EMBASSI-Projekt beteiligt sind. Die folgende Abbildung zeigt schematisch, wie durch das Zusammenwirken von Entwicklern und Psychologen auf der einen Seite konkrete Assistenzsysteme entstehen und auf der anderen Seite zugleich das Vorgehensmodell GUIDEAS mit Gestaltungswissen gefüllt wird.

Wenn das Vorgehensmodell GUIDEAS soweit entwickelt und mit Gestaltungsregeln gefüllt ist, stellen sich Fragen und Antworten als WENN- und DANN-Teile von Regeln eines Produktionssystems dar, in dem sich die Ergebnisse der Untersuchungen im EMBASSI-Projekt in automatisch nutzbarer Form sammeln.

Folgenden Szenarien sollen im späteren Einsatz des Vorgehensmodells abgedeckt werden:

1. Generierung von Gestaltungsvorschlägen

Der Entwickler eines Assistenzsystems hat Fragen im Zusammenhang mit der Gestaltung der oberen User Interface Ebenen dieses Systems und möchte sich diesbezüglich Vorschläge generieren lassen. Er ruft das Vorgehensmodell GUIDEAS auf und wird zunächst aufgefordert, Angaben zu den Zielen des zu entwickelnden Assistenzsystems, zu den Aufgaben, die unterstützt werden sollen, zu den Kontextbedingungen (z.B. Situation, physische und soziale Umgebung) und zu einigen Merkmalen der zukünftigen Benutzer zu machen. Alle diese Angaben werden durch

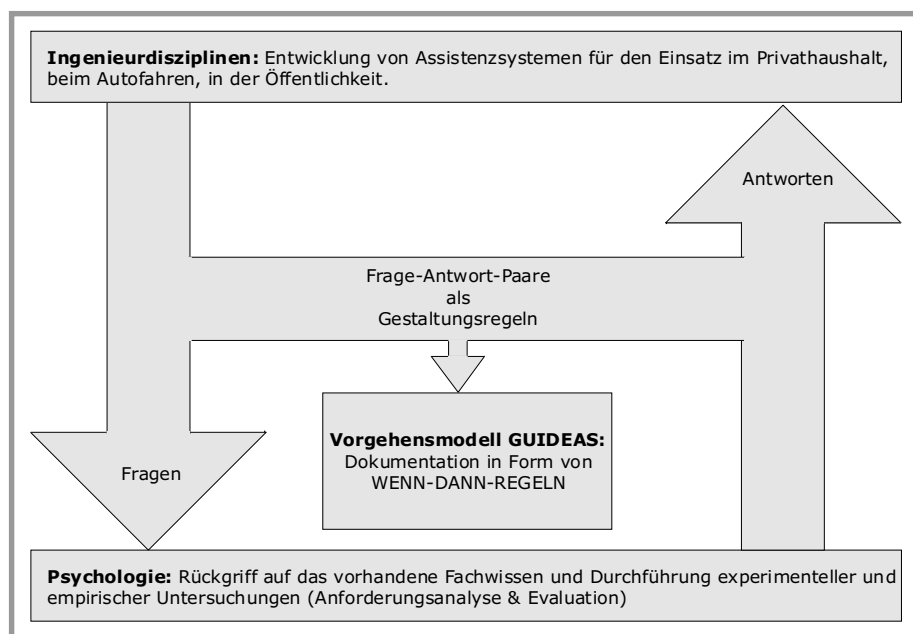


Bild 6: Prozesse der Zusammenarbeit im EMBASSI-Projekt. Ein Teil der Antworten (und damit der Gestaltungsregeln) beruht auf arbeitswissenschaftlichem Fachwissen und muss nicht durch Untersuchungen gewonnen werden. Ob das vorhandene und das erzeugte Wissen hinreichend und valide ist, wird sich in späteren Projektphasen zeigen, in denen die implementierten Assistenzsysteme evaluiert werden.

verschiedene Auswahl-Techniken auf einem HTML-basierten Formular vorgenommen. Je nach Parameter sind Einfach- oder Mehrfachauswahlen möglich.

In allen Fällen hat jedoch der Entwickler eine Abstraktionsleistung zu erbringen. Die im Vorgehensmodell GUIDEAS verwendeten Kategorien sind anwendungsübergreifend formuliert. Die Merkmale von Personen, Aufgaben und Situationen, die hier erfragt werden, sollen geeignet sein, um Assistenzfunktionen beim Bedienen von komplexen Geräten und bei der Inanspruchnahme von Diensten im Haushalt, beim Autofahren und bei öffentlichen Terminalsystemen gleichermaßen vollständig und genau zu beschreiben. Durch die Beschränkung auf einerseits die drei Anwendungsgebiete und andererseits die oberen Ebenen des User Interfaces von Assistenzsystemen, hält sich sowohl der Umfang der zu spezifizierenden Parameter, als auch die geforderte Abstraktionsleistung in Grenzen.

Wenn ein Entwickler eine ausreichende Zahl von Parametern zu Zielen, Aufgaben, Benutzern und Kontext spezifiziert hat, so werden diese Angaben mit den WENN-Teilen der Regeln eines Produktionssystems verglichen. Finden sich zu viele Regeln, die im WENN-Teil mit den Spezifikationen des Entwicklers übereinstimmen, so wird er gebeten, seine Angaben zu präzisieren. Andernfalls bewirken die aktivierten Regeln die Ausgabe von Vorschlägen zur Gestaltung der Assistenz. Die einzelnen Komponenten der Vorschläge sind in den DANN-Teilen der Regeln abgelegt.

Vorschläge werden gemacht zu folgenden Gestaltungsfragen:

- Psychische Funktionen, die unterstützt werden sollen (Wahrnehmen, Entscheidungen ...)
- Automatisierungsgrad der Assistenz (von informierend bis automatisch)
- Initiative der Assistenzfunktion (aktiv - passiv)
- Modalität und Medium (visuell, akustisch, Text, Grafik ...)
- Flexibilität (konstant, adaptierbar, adaptiv)

Da die Vorschläge nicht auf ein konkretes Anwendungsfeld oder Assistenzsystem bezogen sind, muss der Entwickler an dieser Stelle eine Konkretisierung vornehmen und

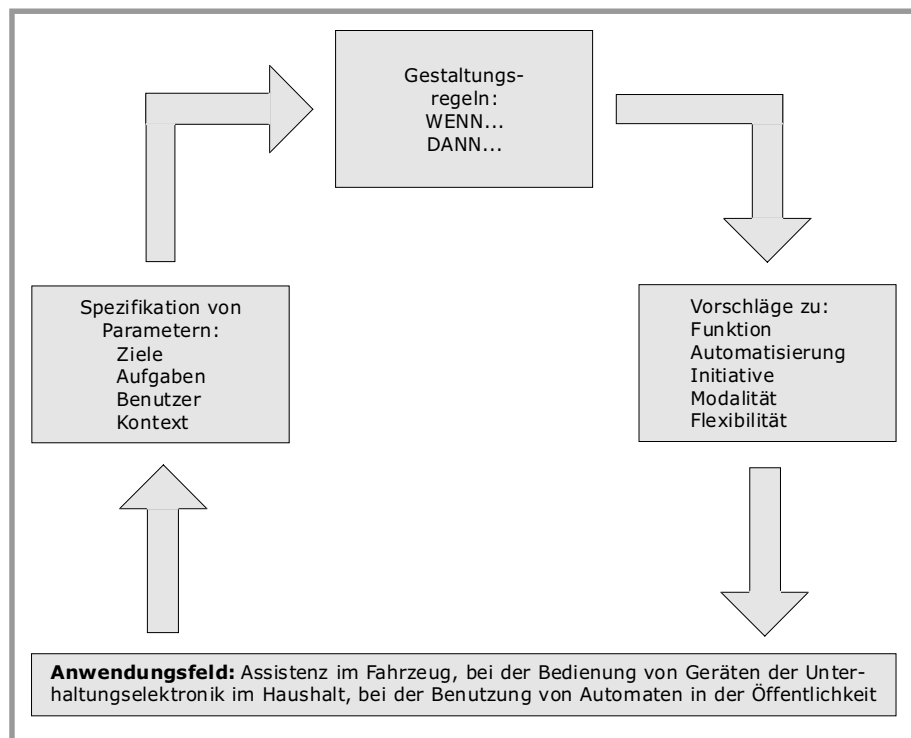


Bild 7: Standardablauf bei der Nutzung von GUIDEAS

die abstrakten Vorschläge instantiiieren. Dadurch schließt sich der Kreis zu der konkreten Entwicklungsaufgabe.

Das Bild 7 zeigt den Prozess, wie er im Szenario Generieren von Gestaltungsvorschlägen abläuft, im Überblick.

Es kann vorkommen, dass Entwickler Schwierigkeiten haben, die Ziele, Aufgaben-, Benutzer-, oder Kontextmerkmale zu spezifizieren. In diesem Fall würden sie, an einer oder mehreren Stellen in dem Bildschirmformular zur Erfassung der Parameter, die Option „Ich weiß nicht.“ auswählen. Dann tritt analog zu der Situation, in der ein Arbeitswissenschaftler zunächst einmal vorschlägt, eine Analyse durchzuführen, das folgende Nutzungsszenario auf.

2. Generierung von Vorschlägen zur Durchführung von Anforderungsanalysen

Viele Parameter können ohne die Anwendung von speziellen Methoden eingegeben werden, dies trifft vor allem auf sozio-biografische Daten der Benutzer und auf Kontextmerkmale der Benutzung zu. Diese Parameter fungieren eher als Erinnerungshilfe für den Entwickler. So kann er z.B. darauf aufmerksam gemacht werden, auch an behinderte Benutzer zu denken, wenn er ein Assistenzsystem entwirft. Besonders bei der Beschreibung der Aufgaben, die unterstützt werden sollen, reicht es oft nicht aus, ad hoc Angaben zu formulieren. Hier wer-

den spezielle Methoden vorgeschlagen, mit denen der Entwickler eine Spezifikation der Merkmale vornehmen kann.

Es werden sowohl Methodenklassen wie Dokumentenanalyse, Beobachtung, Interview, Focusgruppen, Fragebogenverfahren, Strukturlegetechnik, u.a. beschrieben, als auch spezielle Methoden dargestellt wie z. B. die Critical-Incidents-Technik, Fragebogen zur subjektiven Arbeitsanalyse (SAA, Udris & Alioth 1980), Conjoint-Analyse und andere.

Bei den speziellen Methoden wird, wann immer möglich, ein Download von Verfahren, Items, Auswertungstools und weiteren Hilfsmitteln angeboten. Dies trifft in besonderem Maße auf Methoden zu, die von den Mitarbeitern im EMBASSI-Projekt selbst entwickelt wurden, so z.B. ein Fragebogen zur Erfassung der Kontrollüberzeugung im Umgang mit Technik (KUT, Beier 1999).

Der Entwickler wendet die Methoden an und kann dann später auf der Basis der erhobenen Daten die erforderlichen Spezifikationen vornehmen. Bild 8 zeigt diesen Exkurs im Vorgehensmodell GUIDEAS.

3 DIREKTE ERMITTLUNG DES UNTERSTÜTZUNGSBEDARFS

GUIDEAS ist als mitwachsendes System konzipiert. Mit jedem Entwicklungsprojekt,

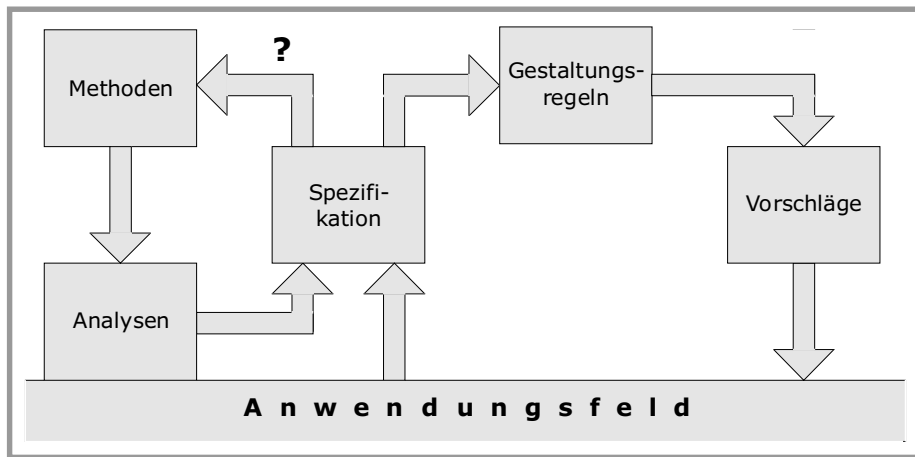


Bild 8: Exkurs in GUIDEAS: Wenn Entwickler nicht wissen, welche Anforderungen an die Benutzer gestellt werden, so werden ihnen Methoden der Anforderungsanalyse vorgeschlagen.

für das es eingesetzt wird, können Gestaltungsregeln überprüft, differenziert, korrigiert, ergänzt und verworfen werden. Die Menge der verfügbaren Gestaltungsregeln ist zunächst sehr begrenzt. Um Entwicklern von Assistenzsystemen dennoch Unterstützung bei der Gestaltung zu geben und auf diese Weise auch neue Gestaltungsregeln zu generieren, sind in GUIDEAS auch spezifische Methoden enthalten, die den Unterstützungsbedarf direkt erfassen. Diese Methoden werden im EMBASSI-Projekt entwickelt und erprobt. Dazu gehören die Funktions-Assistenz-Matrix, die Analyse sozialer Assistenz und die Verwendung von Szenario-Techniken. Diese Methoden liefern direkte Hinweise auf die Gestaltung von Assistenzfunktionen. Aus den Ergebnissen können dann durch Abstraktion Gestaltungsregeln gewonnen werden.

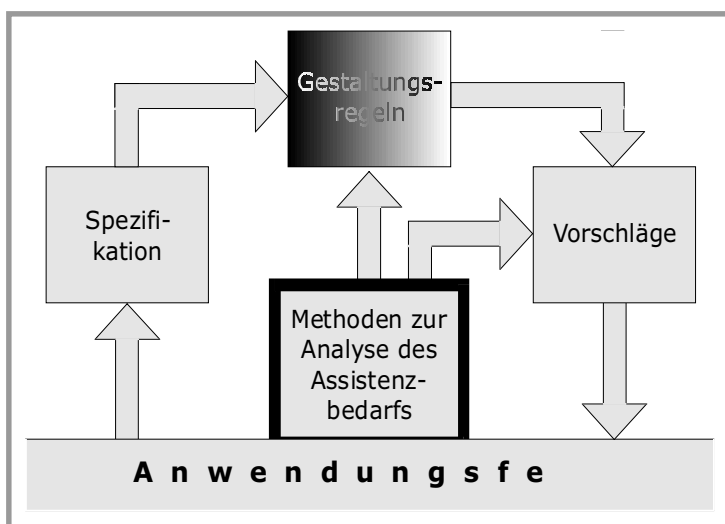
7.2.3 Test- und Evaluationsphase

GUIDEAS soll Entwickler auch bei der Evaluation von Prototypen unterstützen. Nach-

dem auf den drei geschilderten Wegen und durch konkrete Umsetzung der Gestaltungsvorschläge ein Prototyp für ein Assistenzsystem entwickelt wurde, soll dieser Prototyp getestet werden. Die Art von Prototypen reicht von einfachen Papierprototypen über eine Folge von Screenshots bis hin zu ablauffähigen Programmversionen. Vielfach wird es auch nicht nur einen Prototypen geben, sondern mehrere Versionen, die miteinander verglichen werden sollen. GUIDEAS schlägt dann auf der Basis der in der Spezifikationsphase ermittelten Ziele und Anforderungen Evaluationskriterien vor und bietet geeignete Evaluationsmethoden und Tools zur Planung, Durchführung und Auswertung an.

Während der Laufzeit des EMBASSI-Projekts werden alle Ergebnisse von Evaluationsstudien in GUIDEAS dokumentiert. Dies dient zum einen der Sammlung von Beispielen für spätere Benutzer, zum anderen der Überprüfung der Methoden, die durch GUIDEAS empfohlen werden und schließlich der Überarbeitung der Regeln.

Bild 9: Spezielle Methoden zur direkten Analyse des Assistenzbedarfs führen einerseits zu Gestaltungsvorschlägen und ermöglichen es andererseits, die Sammlung von Gestaltungsregeln zu erweitern.



7.2.4 Vorgehen bei der Entwicklung des Vorgehensmodells GUIDEAS

Wichtige Bestandteile von GUIDEAS sind die Regelbasis des Produktionssystems (Gestaltungsregeln) und die Zusammenstellung von Analyse- und Evaluationsmethoden. Beide Bestandteile von GUIDEAS werden auf unterschiedlichen Wegen geschaffen: Durch Analyse der Literatur, durch die Anwendung von Theorien und Modellen, durch die Berücksichtigung von Hintergrundwissen und durch empirische und experimentelle Untersuchungen. Der letzte Weg ist besonders interessant, weil es hier nicht darum geht, vorhandenes arbeitswissenschaftliches Wissen zu sammeln und zu kompilieren, sondern weil hier neues Wissen gewonnen wird. Anhand von zwei ausgewählten Studien soll exemplarisch gezeigt werden, wie die Inhalte von GUIDEAS entstehen. Es handelt sich um die Szenariotechnik für die Ermittlung des Assistenzbedarfs beim Autofahren und die Analyse sozialer Assistenz für die Ermittlung des Assistenzbedarfs beim Bedienen eines Videorekorders.

Analyse des Assistenzbedarfs beim Autofahren durch szenariobasierte Methoden

In der Szenariotechnik nutzt man die menschliche Fähigkeit zur Imagination und lässt die Versuchsperson alle relevanten Merkmale der Situation und auch ihres Verhaltens mental konstruieren. Dieses wird über mündliche oder schriftliche Beschreibungen erreicht, die durch reale Stimuli intensiviert und komplettiert werden können. Hat die Versuchsperson eine hinreichend genaue Vorstellung von der Situation, können ihr Erleben und ihre Verhaltensintentionen in einer Befragung registriert werden. Die Vorteile der Szenariotechnik bestehen darin, dass auf technisch aufwändige bzw. gefährliche Versuchsanordnungen verzichtet werden kann und sogar zukünftige, technisch noch nicht realisierbare Systeme einer Untersuchung zugänglich gemacht werden können. Ein Problem szenariobasierter Untersuchungen besteht in der Übertragung der gefundenen Ergebnisse auf reale Situationen. Der Einsatz der Szenariotechnik erlaubt es, mit geringem Aufwand fast jede beliebige Situation in der Vorstellung herzustellen.

Methodisches Vorgehen

Zwei neuartige Assistenzsysteme wurden in dieser Untersuchung betrachtet: ein System zur Aufmerksamkeitskontrolle des Fahrers

und ein System, das den Fahrer bei der Reaktion auf Störungen im Fahrzeugumfeld unterstützt.

30 Versuchspersonen wurden zu beiden Assistenzsystemen befragt, wobei die eine Hälfte zuerst das System zur Aufmerksamkeitskontrolle des Fahrers bearbeitete und die andere Hälfte das System, das den Fahrer bei der Reaktion auf Störungsmeldungen unterstützt. Der Versuch wurde vollständig als schriftliche Befragung am PC durchgeführt.

Der Versuchsperson wurde zunächst der Funktionsumfang des neuen Assistenzsystems beschrieben. Aufgabe der Versuchsperson war es dann, sich vorzustellen, sie befände sich in einer zuvor schriftlich dargestellten Fahrsituation und ihr Fahrzeug sei mit dem neuartigen Assistenzsystem ausgestattet. Die acht Situationsbeschreibungen pro System enthielten Informationen zu der Fahrumgebung, den kognitiven Anforderungen (Witterungsbedingungen, Verkehrsaufkommen, Ziel der Fahrt) und dem emotionalen Zustand des Fahrers und variierten in drei Faktoren mit je zwei Ausprägungen: Fahrumgebung (Autobahn bzw. Landstrasse), kognitive Beanspruchung (hoch bzw. niedrig) und emotionale Beanspruchung. Dann war ein Fragebogen auszufüllen, dessen Items verschiedene Aspekte kognitiver und emotionaler Beanspruchung erfassten. Damit sollte kontrolliert werden, inwieweit die einzelnen Fahrsituationen tatsächlich unterschiedliche Wirkungen im Erleben der Fahrer erzeugten. Nach einer Aufforderung, sich die beschriebene Situation erneut zu vergegenwärtigen, wurden die Versuchsperson nach ihren Unterstützungswünschen befragt. Es wurden mehrere Assistenzformen unterschiedlichen Automatisierungsgrades vorgeschlagen. Die erlebte Beanspruchung sowie die gewünschte Assistenz wurde auf fünfstufigen Ratingskalen erfasst.

Ergebnisse

Bild 10 zeigt die Ergebnisse für das System zur Fahreraufmerksamkeitskontrolle. Eine Varianzanalyse mit den Faktoren Assistenzform, Fahrumgebung, kognitive Beanspruchung und emotionale Beanspruchung zeigte keine signifikanten Haupteffekte auf die Bewertung der Assistenz. Eher positiv wird Assistenz in Form eines „akustischen Signals“ beurteilt, dicht gefolgt von der „sprachlichen Anweisung“ und der „sprachlichen Anweisung mit anschließender Systemreaktion“. Am wenigsten wünschen sich die Versuchspersonen Unterstützung in Form

einer „automatischen Reaktion“ (Rkt.). Auffällig ist, dass Unterschiede in der Bewertung der Assistenzformen zwischen den einzelnen Situationen bestehen, die Situationen sich jeweils auf den Assistenzwunsch aber gleichförmig auszuwirken scheinen (die Rangreihe der Assistenzformen verändert sich für die verschiedenen Situationen fast nicht). Die Alternative „keine Unterstützung“ wird zu den verschiedenen Assistenzformen exakt gegenläufig bewertet.

Es lässt sich auch ein signifikanter Zusammenhang zwischen der in der Situation erlebten Beanspruchung und dem Wunsch nach Unterstützung erkennen: Je geringer die erlebte Beanspruchung ausfällt, um so weniger Unterstützung wünschen sich die Versuchspersonen.

Bild 11 zeigt die Ergebnisse für das System Reaktion auf Störungsmeldungen. Eine vierfaktorielle Varianzanalyse für Mess-

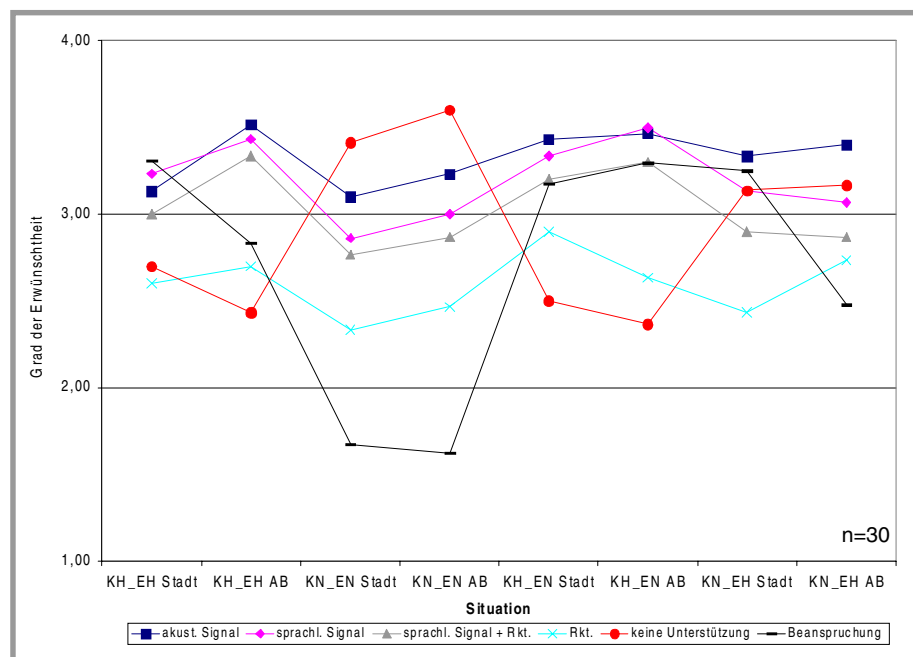


Bild 10: Mittelwerte für die erlebte kognitive bzw. emotionale Beanspruchung (Skala von 1 = sehr geringe Beanspruchung bis 5 = sehr hohe Beanspruchung) und die Erwünschtheit (Skala von 1 = sehr geringe Erwünschtheit bis 5 = sehr hohe Erwünschtheit) unterschiedlicher Assistenzformen bei dem System zur Fahreraufmerksamkeitskontrolle in verschiedenen Fahrsituationen.

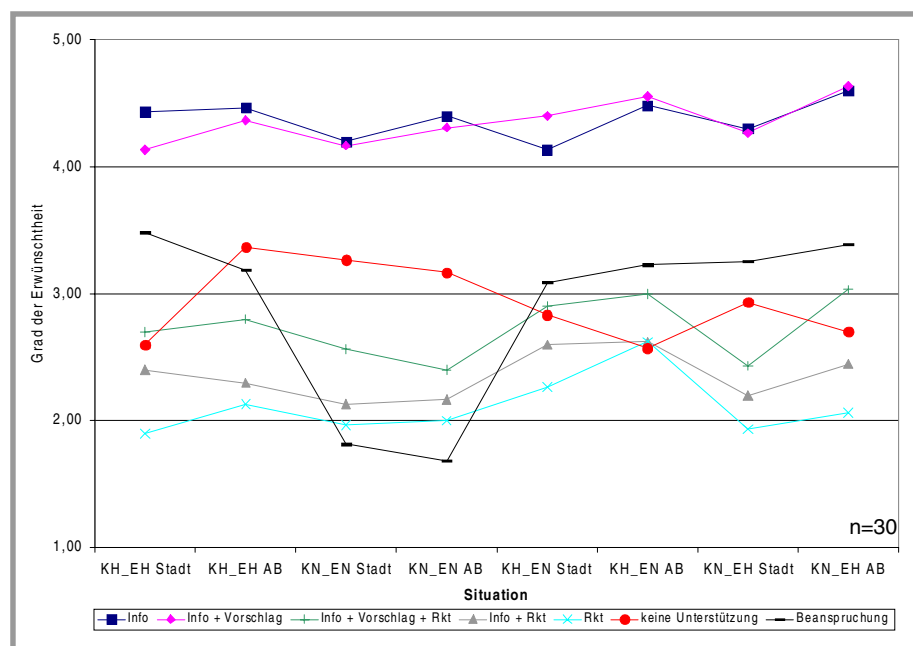


Bild 11: Mittelwerte für die erlebte kognitive bzw. emotionale Beanspruchung (Skala von 1 = sehr geringe Beanspruchung bis 5 = sehr hohe Beanspruchung) und die Erwünschtheit (Skala von 1 = sehr geringe Erwünschtheit bis 5 = sehr hohe Erwünschtheit) unterschiedlicher Assistenzformen bei dem System zur Reaktion auf Störungsmeldungen in verschiedenen Fahrsituationen

wiederholungen mit den Faktoren Assistenzform, Fahrumgebung, kognitive und emotionale Beanspruchung zeigte für den Faktor Assistenzform einen signifikanten Haupteffekt auf die Bewertung der Assistenz. Dies wird auch in Bild 11 deutlich: die Assistenzformen „Information“ und „Information und Handlungsvorschlag“ wurden deutlich bevorzugt. Alle anderen Assistenzformen erzielen geringere Erwünschtheitswerte, die teilweise unterhalb der Skalenmitte (<3) liegen, was insgesamt eine eher negative Bewertung bedeutet.

Schlussfolgerungen

Assistenzformen unterscheiden sich teilweise deutlich in ihrer Bewertung durch Versuchspersonen, wobei die Akzeptanz mit zunehmendem Automatisierungsgrad geringer wird. Möglicherweise stellt die automatische Fahrassistenz einen Eingriff in den Kompetenzbereich und das Kontrollbedürfnis des Fahrers dar, der negativ be-

Tabelle 2: DANN-Teil der Regeln, die aus den Versuchen zur Assistenz bei der Fahreraufmerksamkeit und den Störungsmeldungen abgeleitet wurden

Psychische Funktionen, die unterstützt werden sollen:	Wahrnehmen / Entscheiden
Automatisierungsgrad der Assistenz:	Stufe 1: Benutzer werden informiert
Initiative:	aktiv durch System
Modalität:	akustisch
Flexibilität:	keine Aussage möglich

wertet wird. Eine weitere Ursache für die geringe Akzeptanz der hochautomatisierten Assistenzformen könnte in ihrer Neuartigkeit liegen und in einer generellen, menschlichen Tendenz zur Höherbewertung von Gewohntem und der Ablehnung von Neuem begründet sein.

Für das Vorgehensmodell GUIDEAS lassen sich aus dieser Untersuchung Gestaltungsregeln ableiten, die ausgehend von der Spezifikationen des Autofahrens im WENN-Teil (hier sind für die beiden Szenarien 66 bzw. 64 Bedingungen spezifiziert worden) und den Versuchsergebnissen im DANN-Teil bestehen. Der DANN-Teil einer aus dem Versuch abgeleiteten Regel, kann vereinfacht wie folgt gekennzeichnet werden.

Inwieweit die Szenario-Technik für die Fragestellung Analyse des Unterstützungsbedarfs geeignet ist, kann jetzt noch nicht entschieden werden. Eine Aussage über die Güte der Methode kann erst dann erfolgen, wenn in späteren Phasen (z.B. Test eines Prototypen im Fahrsimulator bzw. im Pkw)

Tabelle 3: Schritte: Abweichungen vom kürzesten Weg in Prozent. * = Diese Werte unterschieden sich in der Varianzanalyse signifikant auf dem $\alpha = 5\%$ -Niveau.

Bedingung	Pretest (ohne Unterstützung)	Haupttest (mit Unterstützung)	Posttest (ohne Unterstützung)
Bedienungsanleitung, n=32	985 %	325 %*	223 %
Soziale Assistenz, n=32	751 %	174 %*	250 %

auch andere Methoden eingesetzt werden können.

Analyse sozialer Assistenz

Die Analyse sozialer Assistenz ermöglicht beides: Die Erfassung des Assistenzbedarfs der Benutzer und die Überprüfung der Auswirkungen verschiedener Arten von Assistenz auf die Leistungen bei der Bedienung. Um diese verschiedenen Arten von Assistenz nicht implementieren zu müssen, wurden menschliche Experten als Assistenten eingesetzt. Durch die physische Anwesenheit einer realen Person können Ausprägun-

So konnten für alle Aufgaben Leistungsdaten wie Anzahl der benötigten Schritte und Zeiten pro Aufgabe ausgewertet werden.

Die Interaktionen zwischen Experten und Versuchspersonen wurden auf Video aufgezeichnet. So konnte posthoc das Expertenverhalten hinsichtlich der Initiative (Assistent ist aktiv, d.h. greift von sich aus ein, oder passiv, d.h. wartet auf Anfrage der Versuchsperson) und der Modalität (Assistent gibt z. B. Erklärungen oder demonstriert Bedienschritte) analysiert werden.

Ergebnisse

Als Leistungsmaße wurden die Lösungszeiten und die Anzahl der Schritte über die nötigen hinaus verwandt. Der Pretest zeigte, dass es keine signifikanten Unterschiede in den Leistungen zwischen den Gruppen gab. Bei der zweiten Aufgabe benötigten die Versuchspersonen mit sozialer Assistenz signifikant weniger Zeit und Schritte als die Versuchspersonen mit der Bedienungsanleitung. In der dritten Bedienaufgabe zeigte sich wiederum kein signifikanter Unterschied in den Leistungen. Tabelle 3 zeigt die durchschnittliche Abweichung an Schritten vom kürzesten Weg in Prozent pro Gruppe und pro Aufgabe (100% entspricht der minimalen Anzahl von Schritten, d.h. dem kürzesten Lösungsweg).

Bild 12 zeigt den Einfluss der Dimension Initiative auf die Leistungen der Versuchspersonen. Es zeigte sich eine signifikante Überlegenheit der Versuchspersonen, die überwiegend aktive Unterstützung erhalten haben.

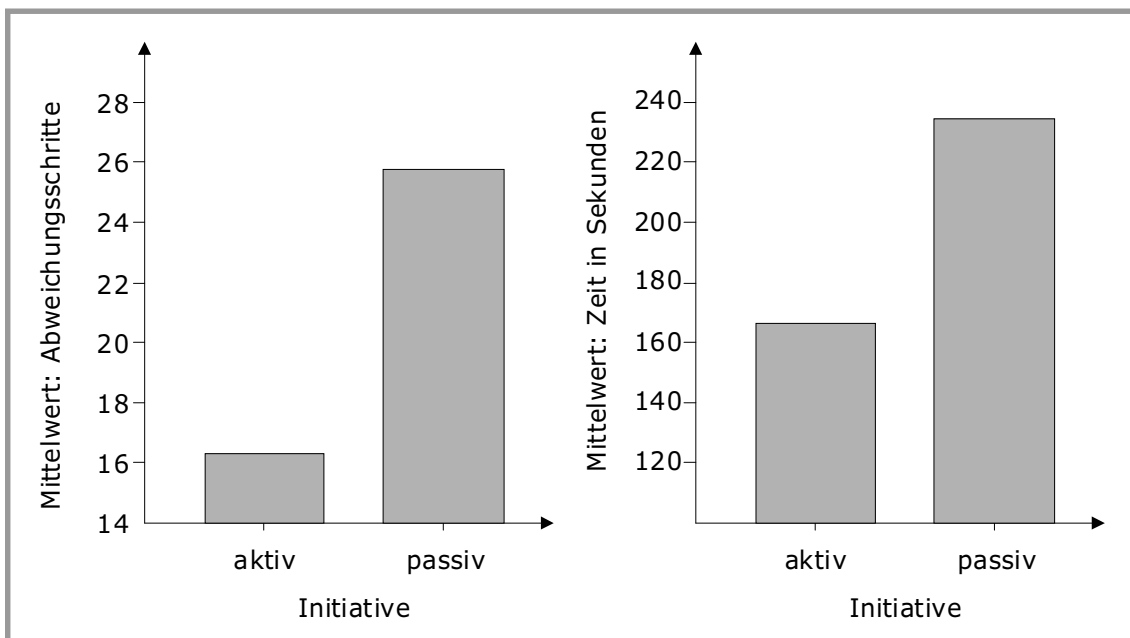
Auf der Dimension Modalität traten zwei Ausprägungen von Arten der Informationsgabe (‘Erklären’ versus ‘Zeigen in Verbindung mit Erklären’) auf. Zeigegesten in Verbindung mit Erklärungen erwiesen sich als signifikant hilfreicher für die Versuchspersonen als Erklärungen allein.

Schlussfolgerungen

Die Ergebnisse zeigen unabhängig von dem konkreten Expertenverhalten in der zweiten Aufgabe eine deutliche Überlegenheit der

Bild 12:

Einfluss der Initiative auf die Leistungen der Versuchspersonen. N (aktiv) = 15, N (passiv) = 17. Die Gruppen unterscheiden sich signifikant auf dem $\alpha = 5\%$ -Niveau.



sozialen Assistenz gegenüber der Bedienungsanleitung als Unterstützung. Das ist nicht überraschend, da ein menschlicher Assistent flexibler auf die Probleme der Versuchspersonen reagieren und aktiv in das Geschehen eingreifen kann. Allerdings überträgt sich dieser Vorteil nicht auf die Leistungen im Posttest. Dieser Befund zeigt, wie wichtig es ist, die Zielstellung für die Entwicklung eines Assistenzsystems festzulegen. Besteht das Ziel des Assistenzsystems darin, das Erlernen des Umgangs mit dem System zu unterstützen, dann scheint ein höherer Automatisierungsgrad und aktive Assistenz nicht geeignet. Besteht das Ziel hingegen darin, die Bedienung zu vereinfachen

WENN-Teil (hier sind 70 Bedingungen spezifiziert worden) und den Versuchsergebnissen im DANN-Teil bestehen. Der DANN-Teil einer aus dem Versuch abgeleiteten Regel, kann dann vereinfacht wie folgt gekennzeichnet werden.

Da diese Regel auf einer eher schwachen Datenbasis beruht (das spontane Verhalten von hilfreichen Personen wurde ausgewertet), sind zu einigen Gestaltungsdimensionen keine Aussagen möglich. Die obige Regel wird ergänzt bzw. korrigiert, sobald neue Ergebnisse vorliegen, die durch systematische Variation der Assistenzdimensionen entstanden sind. So ist z.B. in den bisher-

die Machbarkeit von Assistenztechnologien zu überprüfen. Gegenwärtig und zukünftig geht es darum, zu prüfen, in welchem Maße die technisch machbaren Ansätze auch hilfreich für den Benutzer sind. Die in diesem Zusammenhang durchzuführenden Anforderungs- und Evaluationsanalysen werden eine Vielzahl von Daten liefern, auf deren Grundlage das Vorgehensmodell GUIDEAS mit Inhalten gefüllt werden kann. Mit wachsenden Inhalten wird es zugleich als Werkzeug für die innerhalb des EMBASSI-Projekts tätigen Entwicklungsgruppen interessant, so dass sich hier ein sich selbst verstärkender Kreisprozess aufbauen kann. Nach Abschluss des Projekts sollen ProUse und GUIDEAS soweit fertig gestellt sein, dass eine Übertragung auf andere Anwendungsgebiete, nicht nur Haushalt, Kfz und Terminals, sondern auch professionelle Anwendungen möglich wird.

Tabelle 4: DANN-Teil der Regeln, die aus den Versuchen zur sozialen Assistenz bei der Programmierung eines Videorekorders abgeleitet wurden

Psychische Funktionen, die unterstützt werden sollen:	keine Aussage möglich
Automatisierungsgrad der Assistenz:	keine Aussage möglich
Initiative:	aktiv
Modalität:	akustisch und visuell
Flexibilität:	keine Aussage möglich

chen, dann erweisen sich wiederum gerade diese Eigenschaften als erfolgreich.

Was die Modalität des Expertenverhaltens betrifft, ist die deutliche Überlegenheit von zusätzlichen Zeigegensten ein starkes Argument für den Einsatz anderer Medien als nur der Sprache. Für das Vorgehensmodell GUIDEAS lassen sich aus dieser Untersuchung Gestaltungsregeln ableiten, die ausgehend von der Spezifikationen des Programmieren eines Videorekorders im

gen Versuchen keine automatische Ausführung von Operationen aufgetreten. In nachfolgenden Experimenten werden deshalb hypothesengeleitet weitere Variationen eingeführt.

8 AUSBLICK

Das EMBASSI-Projekt hat mit einer vor allem technologie-orientierten Anlaufphase begonnen. In dieser Phase ging es darum,

LITERATUR

Apple Human Interface Guidelines: Reading MA: Addison-Wesley, 1992

Beier, G.: Kontrollüberzeugungen im Umgang mit Technik. Psychologie Report, Heft 9, S. 684-693, 1999

Beck, A. und Janssen, C.: Vorgehen und Methoden für aufgaben- und benutzer-angemessene Gestaltung von graphischen Benutzungsschnittstellen. In Coy et al. (Hrsg.), Menschengerechte Software als Wettbewerbsfaktor (S. 200-221). Stuttgart: B. G. Teubner, 1993

Brodbeck, F. C.: Kommunikation und Leistung in Projektarbeitsgruppen. Eine empirische Untersu-

chung an Software-Entwicklungsprojekten. Aachen: Shaker Verlag, 1996

Brown, C. M.: Human-Computer Interface Design Guidelines. Norwood: Ablex Publishing Corporation, 1988

Card, S., Moran, T.P. & Newell, A.: The psychology of human-computer interaction. Hillsdale, N.J. : Lawrence Erlbaum Ass. 1983

Constantine, L. L. and Lockwood L. A. D.: Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design. Reading, MA: Addison-Wesley, 1999

Emery, F. und Thorsrud, E.: Industrielle Demokratie. Bern, Stuttgart, Wien: Hans Huber, 1982

Enderle, G.: Seeheim-Workshop on User Interface Management Systems - First Report. Computer Graphics Forum, 3 (2). S. 169-170, 1984

DIN EN ISO 9241-10: Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten - Teil 10: Grundsätze der Dialoggestaltung, Berlin: Beuth-Verlag, 1996

DIN EN ISO 9241-11: Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten - Teil 11: Anforderungen an die Gebrauchstauglichkeit - Leitsätze, Berlin: Beuth-Verlag, 1998

DIN EN ISO 13407: Benutzerorientierte Gestaltung interaktiver Systeme. Berlin: Beuth-Verlag, 2000

Dzida, W.: Das IFIP-Modell für Benutzerschnittstellen. Office Management 31 (Sonderheft). S. 6-8, 1983

Fährnich, K.P.: Software-Ergonomie. München und Wien: R. Oldenbourg Verlag, 1987

Forbrig, P., Gorny, P. und Viereck, A.: Unterstützung des Software-Design-Prozesses durch EXPOSE. In: W. Coy, P. Gorny, I. Kopp und C. Skarpelis (Hrsg.): Menschengerichte Software als Wettbewerbsfaktor. Forschungsansätze und Anwenderergebnisse aus dem Programm „Arbeit und Technik“. Stuttgart: B.G.Teubner, 463-479, 1993

Henderson, D.A. and Card, S.K.: Rooms: The Use of Virtual Workspaces to Reduce Space Contention in a Windows-based Graphical Interface, ACM Transactions on Graphics, 5(3), 211 -243, 1987

Hüttner, J., Wandke, H. und Rätz, A.: Benutzerfreundliche Software. Psychologisches Wissen für die ergonomische Schnittstellengestaltung. Berlin: BMP-Verlag, 1995

IBM Common User Access: (CUA) SC34-4289-00 Cary, NC: IBM, 1991

KAN: Normung im Bereich der Bildschirmarbeit. Kommission Arbeitsschutz und Normung. Sankt Augustin: KAN, 1997

Kieras, D.: Towards a practical GOMS model methodology for user interface design. in: Helander, M. (ed.): Handbook of Human-Computer Interaction. Amsterdam et al. Elsevier, 1988

Koch, M., Reiterer, H. und Min Tjoa, A.: Software-Ergonomie. Gestaltung von EDV-Systemen - Kriterien, Methoden und Werkzeuge. Wien, New, York: Springer-Verlag, 1991

Laurel, B.: The Art of Human Computer Interface Design. Reading, MA: Addison-Wesley, 1990

Mayhew, Deborah J.: Principles and Guidelines in Software User Interface Design. Englewood Cliffs, NJ: Prentice-Hall, 1992

Microsoft: The Windows Interface. Redmont: Microsoft Press, 1992

Moran, T. P.: The Command Language Grammar: a representation for the user interface of interactive computer systems. Int. J. Man-Machine Studies, 15, 3-50, 1981

Myers, B. A.: User Interface Management Systems. Webster, J.G. (ed) Wiley Encyclopedia of Electrical and Electronics Engineering, Volume 23. New York: John Wiley & Sons, pp. 42-58, 1999

Myers, B. A.: User Interface Software Tools. Website Human Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA, URL: <http://www.cs.cmu.edu/afs/cs.cmu.edu/user/bam/www/toolnames.html>. Stand 30. 11. 2000

Myers, B. A. and Rosson, M. B.: Survey on user interface programming. In: Proceedings of SIGCHI'92: Human Factors in Computing Systems, May 3-7, pages 195-202, Monterey, CA: Association for Computing Machinery, 1992

Nass, C., Y. Moon, B. R and Dryer, D. C.: Can computer personalities be human personalities? International Journal of Human-Computer Studies, 43, 223-239, 1995

Nielsen, J.: A virtual protocol model for human-computer interaction. Int. J. Man-Machine Studies, 4, 301-312, 1986

Nielsen, J.: Usability Engineering. Cambridge, MA: Academic Press, 1994

Nielsen, J.: Usability Laboratories. Special Issue of Behaviour and Information Technology, 13, No.1 & 2, 1994

Nielsen, J. and Mack, R.: Usability Inspection Methods. New York: John Wiley & Sons, 1994

Nissler, J., Machate, J. and Hitzges, A.: How to get the Right Outfit for My Agent? Classification- and Design Methodology for a Virtual Shopping Assistant in a 3D World. in: H.-J. Bullinger & J. Ziegler (Eds.), Human-Computer Interaction: Communication, Cooperation, and Application Design, Proceedings of HCI International '99 (the 8th International Conference on Human-Computer Interaction) Volume 2 (S. 162-166). London: Lawrence Erlbaum Associates, 1999

Oed, R., Wetzenstein, E. und Becker, A.: Welche Unterstützung wünschen Softwareentwickler bei der Gestaltung von Benutzungsschnittstellen? In: H. Oberquelle, R. Oppermann, J. Krause (Hrsg.) Mensch & Computer 2001. Stuttgart u.a.: Teubner, 355-364, 2001

Opperman, R., Murchner, B., Reiterer, H. and Koch, M.: Software-ergonomische Evaluation. Der Leitfaden EVADIS II. Berlin und New York: Walter de Gruyter, 1992

Payne, S.J. and Green, T.R.G.: Task-action grammars: A model of the mental representation of languages. Human-Computer Interaction, 2, 93-133, 1986

Reiterer, H.: A user interface design assistant approach. In: Brunnstein, K., Raubold, E. (eds.) Applications and Impacts, Information Processing '94. Proceedings of the IFIP 13th World Computer Congress, Hamburg, Germany, 1994. IFIP Transactions A-52, Volume II, Amsterdam: North-Holland 1994, pp. 180-187, 1994

Selig, J.: EDV-Management. Eine empirische Untersuchung der Entwicklung von Anwendungssystemen in deutschen Unternehmen. Berlin: Springer, 1986

Smith, S. L. and Mosier, J.N.: The user interface to computer-based information systems: a survey of current software design practice. Behaviour and Information Technology. 3, 195-203, 1984

Smith, S. L. and Mosier, J. N.: Guidelines for Designing User Interface Software ESD-TR-86-278 MTR 1009. National Technical Information Service, Springfield, VA: U.S. Department of Commerce, 1986

Tognazzini, B.: TOG on Interface. Reading, MA: Addison-Wesley, 1993

Shneiderman, B.: Designing the User Interface. Strategies for Effective Human-Computer Interaction. Reading, MA: Addison-Wesley, 1997

SUN: Sun Microsystems. Open Look Style Guide, 1989

Sundblad, O. and Sundblad, Y.: OLGA-A Multimodal Interactive Information Assistant. In B. B. Bederson and K. T. Simsarian (Eds.), CHI 98 Video Program, Conference on Human Factors in Computing Systems, 1998

Uchyigit, G., Carlin, B., Quak, E. and Cunningham, J.: Agents in the box. In H.-J. Bullinger and J. Ziegler (Eds.), Human-Computer Interaction: Communication, Cooperation, and Application Design, Proceedings of HCI International '99 (the 8th International Conference on Human-Computer Interaction) Volume 2 (S. 157-161). London: Lawrence Erlbaum Associates, 1999

Udris, I. und Alioth, A.: Fragebogen zur subjektiven Arbeitsanalyse (SAA). in E. Martin, I. Udris, U. Ackermann und K. Ögerli (Hrsg.), Monotonie in der Industrie. Schriften zur Arbeitspsychologie, Band 29. Bern: Huber. S. 61-68, 1980

Ulich, E.: Arbeitspsychologie. Stuttgart: Schäffer-Poeschel 1998

Ulich, E.: Arbeits- und organisationspsychologische Aspekte. In H. Balzert u.a. (Hrsg.), Einführung in die Software-Ergonomie. S. 49-66, Berlin: Walter de Gruyter, 1988

Van Mulken, S., André, E. and Müller, J.: An Empirical Study on the Trustworthiness of Life-Like Interface Agents. In H.-J. Bullinger & J. Ziegler (Eds.), Human-Computer Interaction: Communication, Cooperation, and Application Design, Proceedings of HCI International '99 (the 8th International Conference on Human-Computer Interaction) Volume 2 (S. 152-156). London: Lawrence Erlbaum Associates, 1999

Vanderdonckt, J.: Development Milestones towards a Tool for Working with Guidelines, Interacting with Computers, (12), November, pp. 81-118, 1999

Van der Veer, G.: Human Computer Interaction. Learning, Individual Differences, and Design Recommendation. Alblasterdam: Offsetdrukkerij Haveka B.V., 1990

VDI-Richtlinie 5005: Software-Ergonomie in der Bürokommunikation, Berlin: Beuth Verlag, 1990

Wandmacher, J.: Software-Ergonomie. Berlin und New York: Walter de Gruyter, 1993

ANSCHRIFT DER VERFASSER

Prof. Dr. Hartmut Wandke

Dipl.-Psych. Markus van Ballegooy

Dipl.-Psych. Julia Nitschke

Humboldt-Universität zu Berlin

Institut für Psychologie

Lehrstuhl für Kognitive Ergonomie / Ingenieurpsychologie

Oranienburger Str. 18

D- 10178 Berlin

Dipl.-Ing. (FH) Richard Oed

Dipl.-Inf. Eduard Metzker

DaimlerChrysler AG

Forschung Informationstechnologie

Wilhelm-Runge-Str. 11

D-89081 Ulm