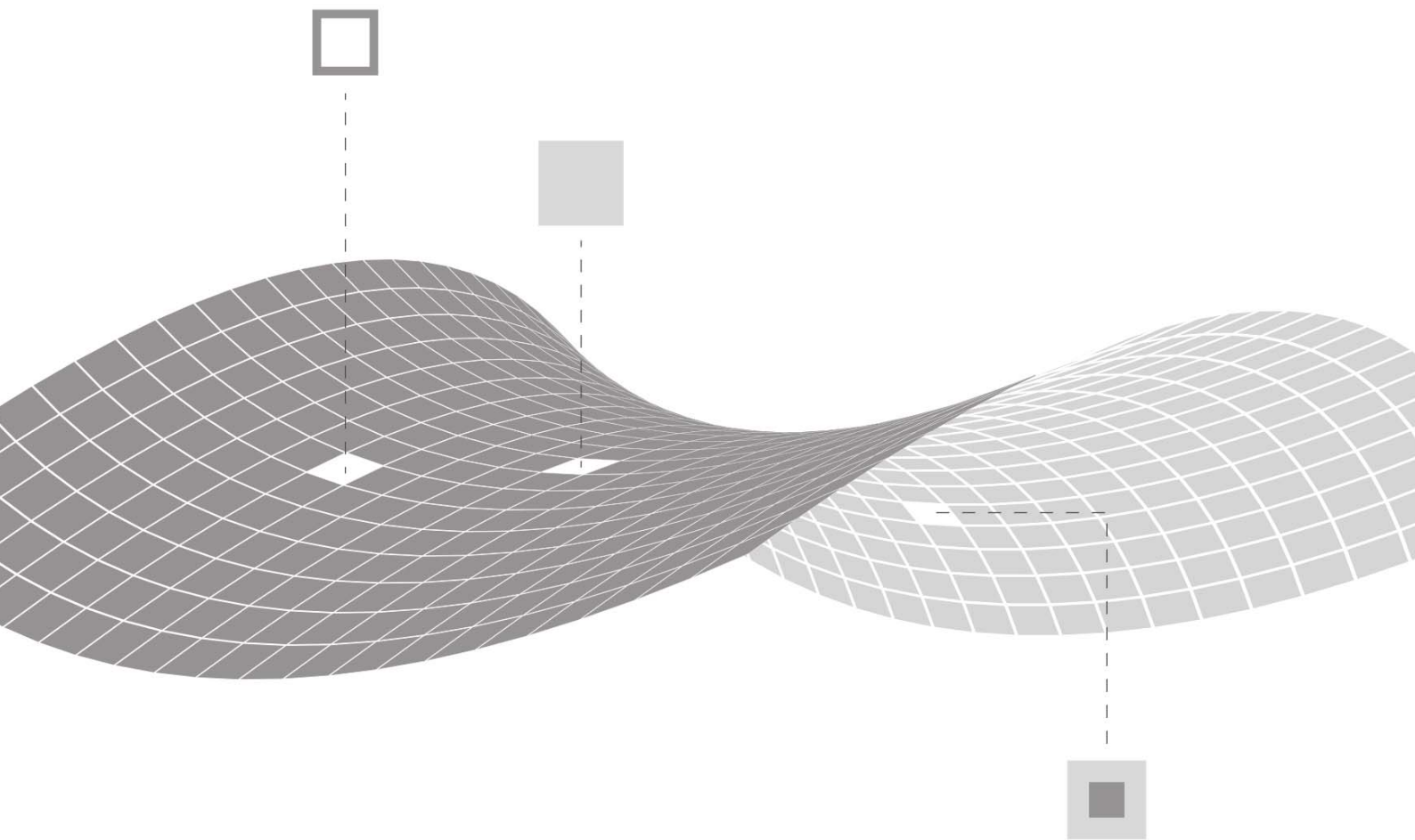


Matthias Peissner

Entwurfsmusterbasierter Ansatz für adaptive Benutzungsschnittstellen zur Überwindung von Nutzungsbarrieren



SCHRIFTENREIHE ZU ARBEITSWISSENSCHAFT UND TECHNOLOGIEMANAGEMENT

Herausgeber

Univ.-Prof. Dr.-Ing. Dr.-Ing. E. h. Dr. h. c. Dieter Spath

Univ.-Prof. Dr.-Ing. habil. Prof. e. h. mult. Dr. h. c. mult. Hans-Jörg Bullinger

Institut für Arbeitswissenschaft und Technologiemanagement IAT
der Universität Stuttgart, Stuttgart

Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO, Stuttgart

Band 12

Matthias Peissner

Entwurfsmusterbasierter Ansatz für adaptive Benutzungsschnittstellen zur
Überwindung von Nutzungsbarrieren

Impressum

Kontaktadresse:

*Institut für Arbeitswissenschaft
und Technologiemanagement IAT
der Universität Stuttgart und
Fraunhofer-Institut für Arbeitswirtschaft
und Organisation IAO
Nobelstraße 12, 70569 Stuttgart
Telefon +49 711 970-01, Fax -2299
www.iat.uni-stuttgart.de
www.iao.fraunhofer.de*

*Schriftenreihe zu Arbeitswissenschaft
und Technologiemanagement*

Herausgeber:

*Univ. Prof. Dr.-Ing. E.h. Dr. h. c. Dieter Spath
Univ. Prof. Dr.-Ing. habil. Prof. e.h. mult.
Dr. h.c. mult. Hans-Jörg Bullinger*

*Institut für Arbeitswissenschaft
und Technologiemanagement IAT
der Universität Stuttgart und
Fraunhofer-Institut für Arbeitswirtschaft
und Organisation IAO*

Bibliografische Information der

*Deutschen Nationalbibliothek:
Die Deutsche Nationalbibliothek verzeichnet
diese Publikation in der Deutschen National-
bibliografie; detaillierte bibliografische Daten sind
im Internet über www.dnb.de abrufbar.*

ISSN 2195-3414

ISBN 978-3-8396-0769-5

D 93

Zugl.: Stuttgart, Univ., Diss., 2014

Druck und Weiterverarbeitung:

*IRB Mediendienstleistungen
Fraunhofer-Informationszentrum
Raum und Bau IRB, Stuttgart*

*Für den Druck des Buches wurde chlor-
und säurefreies Papier verwendet.*

© by FRAUNHOFER VERLAG, 2014

*Fraunhofer-Informationszentrum
Raum und Bau IRB
Postfach 800469, 70504 Stuttgart
Nobelstraße 12, 70569 Stuttgart
Telefon +49 711 970-2500, Fax -2508
E-Mail verlag@fraunhofer.de
<http://verlag.fraunhofer.de>*

Alle Rechte vorbehalten

*Dieses Werk ist einschließlich aller seiner Teile ur-
heberrechtlich geschützt. Jede Verwertung, die über
die engen Grenzen des Urheberrechtsgesetzes hin-
ausgeht, ist ohne schriftliche Zustimmung des Ver-
lages unzulässig und strafbar. Dies gilt insbesondere
für Vervielfältigungen, Übersetzungen, Mikroverfil-
mungen sowie die Speicherung in elektronischen
Systemen. Die Wiedergabe von Warenbezeichnun-
gen und Handelsnamen in diesem Buch berechtigt
nicht zu der Annahme, dass solche Bezeichnungen
im Sinne der Warenzeichen- und Markenschutz-
Gesetzgebung als frei zu betrachten wären und des-
halb von jedermann benutzt werden dürften. Soweit
in diesem Werk direkt oder indirekt auf Gesetze,
Vorschriften oder Richtlinien (z.B. DIN, VDI) Bezug
genommen oder aus ihnen zitiert worden ist, kann
der Verlag keine Gewähr für Richtigkeit, Vollständig-
keit oder Aktualität übernehmen.*

Geleitwort

Grundlage der Arbeiten am Institut für Arbeitswissenschaft und Technologiemanagement IAT der Universität Stuttgart und am kooperierenden Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO ist die Überzeugung, dass unternehmerischer Erfolg in Zeiten globalen Wettbewerbs vor allem bedeutet, neue technologische Potenziale nutzbringend einzusetzen. Deren erfolgreicher Einsatz wird vor allem durch die Fähigkeit bestimmt, kunden- und mitarbeiterorientiert Technologien schneller als die Mitbewerber zu entwickeln und anzuwenden. Dabei müssen gleichzeitig innovative und anthropozentrische Konzepte der Arbeitsorganisation zum Einsatz kommen. Die systematische Gestaltung wird also erst durch die Bündelung von Management- und Technologiekompetenz ermöglicht. Dabei wird durch eine ganzheitliche Betrachtung der Forschungs- und Entwicklungsthemen gewährleistet, dass wirtschaftlicher Erfolg, Mitarbeiterinteressen und gesellschaftliche Auswirkungen immer gleichwertig berücksichtigt werden.

Die im Rahmen der Forschungsarbeiten an den Instituten entstandenen Dissertationen werden in der »Schriftenreihe zu Arbeitswissenschaft und Technologiemanagement« veröffentlicht. Die Schriftenreihe ersetzt die Reihe »IPA-IAO Forschung und Praxis«, herausgegeben von H. J. Warnecke, H.-J. Bullinger, E. Westkämper und D. Spath. In dieser Reihe sind in den vergangenen Jahren über 500 Dissertationen erschienen. Die Herausgeber wünschen den Autoren, dass ihre Dissertationen aus den Bereichen Arbeitswissenschaft und Technologiemanagement in der breiten Fachwelt als wichtige und maßgebliche Beiträge wahrgenommen werden und so den Wissensstand auf ein neues Niveau heben.



Dieter Spath



Hans-Jörg Bullinger

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO, Stuttgart.

Mein herzlicher Dank gilt Herrn Prof. Dr.-Ing. Dr.-Ing. E.h. Dr. h.c. Dieter Spath für die wissenschaftliche Unterstützung und die wohlwollende Förderung dieser Arbeit. Herrn Prof. Dr.-Ing. Thomas Maier danke ich für sein Interesse an der Arbeit und die Übernahme des Mitberichts.

Ganz besonders möchte ich mich bei Frau apl. Prof. Dr.-Ing. habil. Anette Weisbecker bedanken. Ihre inhaltliche Begleitung der Arbeit und ihre konstruktiven Hinweise haben mir sehr geholfen.

Weiterhin danke ich den Partnern des MyUI-Projekts und allen Kolleginnen und Kollegen des Competence Centers Human-Computer Interaction, die zum Gelingen der Arbeit beigetragen haben.

Schließlich möchte ich mich bei meiner Frau Tanja und unseren Söhnen Otis und Valentin bedanken, die mich liebevoll unterstützt und motiviert haben.

Stuttgart, im Juli 2014
Matthias Peissner

**Entwurfsmusterbasierter Ansatz für adaptive
Benutzungsschnittstellen zur Überwindung von
Nutzungsbarrieren**

Von der Fakultät Konstruktions-, Produktions- und Fahrzeugtechnik
der Universität Stuttgart
zur Erlangung der Würde eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Abhandlung

Vorgelegt von
Dipl. Psych. Matthias Peissner
aus Hirschau

Hauptberichter: Univ.-Prof. Dr.-Ing. Dr.-Ing. E.h. Dr. h.c. Dieter Spath
Mitberichter: Univ.-Prof. Dr.-Ing. Thomas Maier

Tag der mündlichen Prüfung: 24. April 2014

Institut für Arbeitswissenschaft und Technologiemanagement IAT
der Universität Stuttgart

2014

Inhalt

| | |
|---|-----------|
| Abkürzungsverzeichnis | 11 |
| Abbildungsverzeichnis | 13 |
| Tabellenverzeichnis | 15 |
| 1 Einleitung..... | 17 |
| 1.1 Adaptive Benutzungsschnittstellen zur Überwindung von Nutzungsbarrieren..... | 17 |
| 1.2 Aufbau der Arbeit..... | 19 |
| 2 Stand der Forschung und Technik | 21 |
| 2.1 Adaptierbare und adaptive Benutzungsschnittstellen | 23 |
| 2.2 Selbstständige Anpassungen während der Nutzung..... | 26 |
| 2.3 Transparenz und Kontrollierbarkeit bei systeminitiierten Anpassungen..... | 28 |
| 2.3.1 Transparenz von systeminitiierten Anpassungen | 28 |
| 2.3.2 Kontrollierbarkeit von systeminitiierten Anpassungen | 31 |
| 2.4 Umfassende Anpassungen zur Überwindung von Nutzungsbarrieren..... | 34 |
| 2.5 Modulare, erweiterbare und offene Adaptationsmechanismen | 37 |
| 2.6 Grundlegende Voraussetzungen für Akzeptanz in der Software-Entwicklung..... | 39 |
| 2.6.1 Effiziente Entwicklung einer adaptiven Benutzungsschnittstelle | 40 |
| 2.6.2 Einstiegshürde und Kompetenzanforderungen auf Seiten der Entwickler..... | 43 |
| 2.6.3 Nachvollziehbarkeit und Beherrschbarkeit der Prozesse zur Erzeugung der resultierenden Benutzungsschnittstelle | 45 |
| 2.7 Zusammenfassung: Anforderungen und Defizite bestehender Ansätze | 46 |
| 3 Überblick des Ansatzes..... | 51 |
| 3.1 Selbstlernende und dynamische Adaptationen nach dem Vorbild der zwischenmenschlichen Kommunikation..... | 52 |
| 3.2 Adaptierbarkeit sowie Transparenz und Kontrollierbarkeit des Anpassungsverhaltens | 54 |
| 3.3 Entwurfsmuster als modulare Bausteine der adaptiven Benutzungsschnittstelle..... | 54 |
| 3.4 Umfassende Anpassungen in einem dreistufigen Prozess zur Generierung und Adaptation der Benutzungsschnittstelle | 55 |
| 3.5 Model-View Controller als konzeptionelle Basis der Software-Architektur | 57 |
| 3.6 Einfache und effiziente Applikationsmodellierung mit erweiterten Zustandsdiagrammen | 58 |
| 4 Entwurfsmuster für adaptive Benutzungsschnittstellen..... | 60 |
| 4.1 Adaptive Benutzungsschnittstellen durch Komposition von Entwurfsmustern..... | 60 |
| 4.2 Typen von Entwurfsmustern..... | 61 |
| 4.3 Entwurfsmustersprache..... | 63 |
| 4.3.1 Die Relation substituiert..... | 64 |
| 4.3.2 Die Relation benötigt..... | 65 |
| 4.3.3 Die Relation wird benötigt von..... | 65 |
| 4.3.4 Die Relation setzt <Variable> wie benötigt von..... | 66 |

| | | |
|----------|---|------------|
| 4.3.5 | Die Relation benötigt <Variable> wie gesetzt von..... | 66 |
| 4.3.6 | Die Relation setzt <Variable> wie genutzt von | 67 |
| 4.3.7 | Die Relation nutzt <Variable> wie gesetzt von | 68 |
| 4.3.8 | Die Relation erweitert | 68 |
| 4.3.9 | Die Relation wird erweitert von..... | 69 |
| 4.4 | Entwurfsmusterrepositorium | 69 |
| 4.4.1 | Entwurfsmuster in natürlicher Sprache und als Softwarekomponenten | 69 |
| 4.4.2 | Erweiterbares Repositorium | 71 |
| 4.4.3 | Öffentlich zugängliches Repositorium | 74 |
| 5 | Parametrierung der adaptiven Benutzungsschnittstelle | 76 |
| 5.1 | Gerätemanager..... | 77 |
| 5.2 | Geräteprofil (DP: Device Profile) | 78 |
| 5.3 | Gerätespezifische Muster (Device-Specific Patterns)..... | 80 |
| 5.4 | Nutzer- und Umgebungsprofil (UP: User Profile)..... | 82 |
| 5.5 | Benutzungsschnittstellenprofil (User Interface Profile)..... | 84 |
| 5.6 | Individualisierungsmuster (Individualization Patterns)..... | 86 |
| 5.7 | Customization-Profil (CP: Customization Profile) | 89 |
| 6 | Vorbereitung der adaptiven Benutzungsschnittstelle..... | 91 |
| 6.1 | Interaktionsmuster (Interaction Patterns) | 92 |
| 6.2 | Bedien- und Anzeigeelemente (User Interface Elements) | 94 |
| 7 | Generierung und Adaptation der adaptiven Benutzungsschnittstelle | 97 |
| 7.1 | Generierung der Benutzungsschnittstelle | 98 |
| 7.2 | Rückkopplung zum Nutzer- und Kontextmanagement..... | 99 |
| 7.3 | Adaptationen während der Nutzung..... | 100 |
| 7.4 | Adaptationsausführungsmuster (Adaptation Rendering Patterns)..... | 101 |
| 7.5 | Adaptationsdialogmuster (Adaptation Dialogue Patterns)..... | 104 |
| 7.5.1 | Automatische Adaptation mit impliziter Nutzerzustimmung | 106 |
| 7.5.2 | Explizite Nutzerbestätigung vor der Adaptation | 108 |
| 7.5.3 | Explizite Nutzerbestätigung nach der Adaptation..... | 109 |
| 7.6 | Mechanismen zur nutzerinitiierten Anpassung der adaptiven Benutzungsschnittstelle..... | 110 |
| 7.6.1 | Nutzerinitiierte Anpassungen über das UP | 110 |
| 7.6.2 | Nutzerinitiierte Anpassungen über das UIP..... | 111 |
| 8 | Das Abstrakte Applikations-Interaktions-Modell (AAIM) zur Modellierung der adaptiven Benutzungsschnittstelle | 112 |
| 8.1 | Zielsetzung und Rolle des AAIM | 112 |
| 8.2 | Zustandsdiagramme zur abstrakten Modellierung der Benutzungsschnittstelle..... | 113 |
| 8.3 | Interaktionssituationen als Grundlage von Adaptationen..... | 114 |
| 8.4 | Interaktionssituationen in Zuständen..... | 116 |
| 8.5 | Interaktionssituationen an Zustandsübergängen | 117 |
| 8.6 | Interaktionssituationen und ihre Syntax..... | 118 |
| 8.6.1 | Eingangsterm der IS..... | 119 |
| 8.6.2 | Ausgangsterm der IS..... | 120 |
| 8.7 | AAIM im Entwicklungsprozess einer adaptiven Benutzungsschnittstelle | 120 |

| | | |
|-----------|--|------------|
| 9 | Evaluierung im Anwendungsfeld barrierefreier Applikationen für interaktives Fernsehen | 122 |
| 9.1 | Evaluierungsplan | 124 |
| 9.2 | Technische Evaluierung anhand einer Referenzimplementierung | 126 |
| 9.2.1 | Evaluierung der automatischen Adaptivität | 132 |
| 9.2.2 | Evaluierung der gemischten Initiative | 135 |
| 9.2.3 | Evaluierung der kontinuierlichen Anpassungsfähigkeit während der Nutzung | 137 |
| 9.2.4 | Evaluierung der umfassenden Anpassungsfähigkeit | 141 |
| 9.2.5 | Evaluierung der Modularität | 144 |
| 9.2.6 | Evaluierung der Erweiterbarkeit | 146 |
| 9.2.7 | Evaluierung der Möglichkeit, die Adaptationsmechanismen zu veröffentlichen und für externe Beiträge zu öffnen | 148 |
| 9.2.8 | Evaluierung des sukzessiven Effizienzgewinns bei der Entwicklung weiterer adaptiver Benutzungsschnittstellen | 151 |
| 9.3 | Empirische Evaluierung der Transparenz und Kontrollierbarkeit mit potenziellen Endnutzern ... | 157 |
| 9.3.1 | Fragestellung | 158 |
| 9.3.2 | Methode | 158 |
| 9.3.3 | Ergebnisse | 163 |
| 9.3.4 | Interpretation und Diskussion | 166 |
| 9.4 | Empirische Evaluierung mit Software-Entwicklern und Entscheidungsträgern aus Software-Unternehmen | 168 |
| 9.4.1 | Fragestellung | 168 |
| 9.4.2 | Methode | 168 |
| 9.4.3 | Ergebnisse | 171 |
| 9.4.4 | Interpretation und Diskussion | 174 |
| 9.5 | Zusammenfassung der Evaluierung | 175 |
| 10 | Zusammenfassung und Ausblick | 177 |
| 10.1 | Zusammenfassung | 177 |
| 10.2 | Ausblick | 178 |
| 11 | Summary | 181 |
| 12 | Anhang | 182 |
| 12.1 | Bildschirmansichten der Referenzimplementierung | 182 |
| 12.2 | Überblick der im Rahmen der Referenzimplementierung erstellten Entwurfsmuster | 185 |
| 12.3 | Beispiele für konkrete Entwurfsmusterbeschreibungen | 187 |
| 12.4 | Bildschirmansichten des Szenarios zur Evaluierung der automatischen Adaptivität | 193 |
| 12.5 | Bildschirmansichten der Szenarien zur Evaluierung der kontinuierlichen Anpassungsfähigkeit während der Nutzung | 197 |
| 13 | Literatur | 199 |

Abkürzungsverzeichnis

| | |
|-------|---|
| A | Menge der Adaptationsmuster |
| AA | Menge der Adaptationsausführungsmuster |
| AAIM | Abstract Application Interaction Model |
| ABS | Adaptive Benutzungsschnittstelle |
| AD | Menge der Adaptationsdialogmuster |
| AI | Automatische Adaptation mit impliziter Nutzerzustimmung (Adaptationsdialogmuster) |
| AUI | Abstract User Interface |
| B | Menge der Bedien- und Anzeigeelemente |
| BNF | Backus-Naur Form |
| CMYK | Cyan, Magenta, Yellow, Key |
| CP | Customization Profile |
| CSS | Cascading Style Sheets |
| CUI | Concrete User Interface |
| D | Menge der gerätespezifischen Entwurfsmuster |
| DIMDI | Deutsches Institut für Medizinische Dokumentation und Information |
| DP | Device Profile |
| DTB | Digital Talking Book |
| EN | Explizite Nutzerbestätigung nach der Adaptation (Adaptationsdialogmuster) |
| EPG | Electronic Program Guide |
| EU | Europäische Union |
| EV | Explizite Nutzerbestätigung vor der Adaptation (Adaptationsdialogmuster) |
| FUI | Final User Interface |
| GOMS | Goals, operators, methods, selection rules |
| GUI | Graphical User Interface |
| HTML | Hypertext Markup Language |
| HTTP | Hypertext Transfer Protocol |
| I | Menge der Individualisierungsmuster |
| ICF | International Classification of Functioning, Disability and Health |
| ID | Identifizier |
| IS | Interaktionssituation |

| | |
|---------|---|
| ISE | Erweiterung einer Interaktionssituation |
| ISO | International Organization for Standardization |
| iTV | Interaktives Fernsehen |
| K | Menge der Interaktionsmuster |
| MVC | Model-View-Controller |
| OMG | Object Management Group |
| PIS | Primäre Interaktionssituation |
| RGB | Red, Green, Blue |
| RPC | Remote Procedure Call |
| TV | Television |
| UI | User Interface |
| UIDL | User Interface Description Language |
| UIML | User Interface Markup Language |
| UIP | User Interface Profile |
| UP | User Profile |
| UML | Unified Modeling Language |
| VUMS | Virtual User Modeling and Simulation |
| WHO | World Health Organization |
| WYSIWYG | What You See Is What You Get |
| XML | Extensible Markup Language |
| XML-RPC | Remote Procedure Call (in Extensible Markup Language kodiert) |

Abbildungsverzeichnis

| | | |
|--------------|--|-----|
| Abbildung 1 | Aufbau der Arbeit | 20 |
| Abbildung 2 | Beitrag der vorliegenden Arbeit zum MyUI-Gesamtsystem..... | 51 |
| Abbildung 3 | Selbstlernende und selbstadaptive Benutzungsschnittstellen in MyUI..... | 53 |
| Abbildung 4 | Dreistufiger Prozess zur Generierung und Adaptation der Benutzungsschnittstelle | 57 |
| Abbildung 5 | Verschiedene Entwurfsmustertypen im Adaptationsprozess..... | 61 |
| Abbildung 6 | Generisches Modell für Erweiterungen des MyUI-Entwurfsmusterrepositoriums | 71 |
| Abbildung 7 | Profile und Entwurfsmuster im Prozess der ABS-Parametrierung..... | 76 |
| Abbildung 8 | Der Gerätemanager steuert die Erstellung und Aktualisierung des Geräteprofils..... | 78 |
| Abbildung 9 | Die Individualisierungsmuster des Bündels DISPLAY MODE transformieren die Werte der UP-Variablen <i>languageReception</i> und <i>understandingAbstractSigns</i> in geeignete <i>displayMode</i> -Einstellungen im UIP..... | 87 |
| Abbildung 10 | Wesentliche Konzepte im Prozess der ABS-Vorbereitung..... | 91 |
| Abbildung 11 | Wesentliche Konzepte im Prozess der ABS-Generierung und Adaptation | 97 |
| Abbildung 12 | Gestaltungsraum der MyUI-Adaptationsdialogmuster..... | 105 |
| Abbildung 13 | Transparenz- und Kontrollmechanismen bei der automatischen Adaptation mit impliziter Nutzerzustimmung..... | 106 |
| Abbildung 14 | Ablauf einer automatischen Adaptation mit impliziter Nutzerzustimmung..... | 106 |
| Abbildung 15 | Bildschirmsequenz während einer automatischen Adaptation mit impliziter Nutzerzustimmung..... | 107 |
| Abbildung 16 | Transparenz- und Kontrollmechanismen bei der expliziten Nutzerbestätigung vor der Adaptation..... | 108 |
| Abbildung 17 | Ablauf des Adaptationsdialogmusters „Explizite Nutzerbestätigung vor der Adaptation“ | 108 |
| Abbildung 18 | Adaptationsdialogmuster „Explizite Nutzerbestätigung vor der Adaptation“: Überblendung der aktuellen Benutzungsschnittstelle durch ein Dialogfenster mit Vorschau der geplanten Anpassung | 109 |
| Abbildung 19 | Transparenz- und Kontrollmechanismen bei der expliziten Nutzerbestätigung nach der Adaptation..... | 109 |
| Abbildung 20 | Ablauf des Adaptationsdialogmusters „Explizite Nutzerbestätigung nach der Adaptation“ | 110 |
| Abbildung 21 | Rolle des AAIM im MyUI-Rahmenmodell | 113 |
| Abbildung 22 | Einfacher Zustand im AAIM..... | 116 |
| Abbildung 23 | Zusammengesetzter Zustand im AAIM | 117 |
| Abbildung 24 | Transition mit einer IS-Erweiterung (ISE) in der verkürzten AAIM-Schreibweise (oben) und in der UML-konformen Schreibweise (unten)..... | 118 |
| Abbildung 25 | Implementierungskonzept für den Adaptationsprozess bei Laufzeitanpassungen (Sequenzdiagramm) | 128 |

| | | |
|--------------|---|-----|
| Abbildung 26 | MyUI-Referenzimplementierung auf iTV und iPhone..... | 129 |
| Abbildung 27 | Ausschnitt des AAIM für die E-Mail-Applikation | 131 |
| Abbildung 28 | Web-Applikation zur gezielten Manipulation des Nutzer- und Umgebungsprofils (UP) ... | 132 |
| Abbildung 29 | Hilfsmittel zur systematischen Variation des Nutzens von automatischen Adaptationen in der Nutzerstudie..... | 161 |
| Abbildung 30 | Anzahl positiver Nutzerbewertungen für Transparenz, Kontrollierbarkeit, Nutzungskomfort und Akzeptanz der verschiedenen Dialogbedingungen | 164 |
| Abbildung 31 | Ergebnisse der Befragung von Entscheidungsträgern von Software-Unternehmen und Entwicklern zur praktischen Eignung des Ansatzes für die kommerzielle Software-Entwicklung (Mittelwerte der Bewertungen auf fünfstelligen Likert-Skalen)..... | 172 |
| Abbildung 32 | Bildschirmansichten der Referenzimplementierung | 184 |
| Abbildung 33 | Überblick der für die Referenzimplementierung dokumentierten Entwurfsmuster. Entwurfsmuster innerhalb eines Bündels sind nicht durch Linien getrennt. Die Entwurfsmusterbeschreibungen finden sich unter http://myui-patterns.iao.fraunhofer.de | 186 |
| Abbildung 34 | Beschreibung des Interaktionsmusters „Main Menu – 4x4 Tile Style (iTV Default)“ | 188 |
| Abbildung 35 | Beschreibung des Interaktionsmusters „Main Menu – 3x3 Tile Style“ | 191 |
| Abbildung 36 | Beschreibung des Individualisierungsmusterbündels „ <i>DISPLAY MODE</i> “ – kopiert aus dem MyUI-Entwurfsmusterrepositorium unter http://myui-patterns.iao.fraunhofer.de/index.php/Display_Mode . Das zusätzliche Feld „State“ wurde von den Entwicklern während der Implementierung eingefügt, um den Stand der inhaltlichen und technischen Realisierung zu kennzeichnen..... | 192 |
| Abbildung 37 | Bildschirmansichten des Testszenarios zur Evaluierung der automatischen Adaptivität (Beschreibung in Abschnitt 9.2.1.2)..... | 196 |
| Abbildung 38 | Bildschirmansichten der Testszenarien zur Evaluierung der kontinuierlichen Anpassungsfähigkeit (Beschreibung in Abschnitt 9.2.3.2)..... | 198 |

Tabellenverzeichnis

| | | |
|------------|--|-----|
| Tabelle 1 | Zusammenfassende Bewertung aktueller Ansätze anhand der hergeleiteten Anforderungen. | 50 |
| Tabelle 2 | Beispiel eines Geräteprofils (DP) für die Konfiguration eines interaktiven Fernsehgeräts | 79 |
| Tabelle 3 | Vorlage für die Beschreibung eines Bündels von gerätespezifischen Mustern | 81 |
| Tabelle 4 | Vorlage für die Beschreibung eines gerätespezifischen Musters | 81 |
| Tabelle 5 | Variablen des Nutzer- und Umgebungsprofils UP | 84 |
| Tabelle 6 | Variablen des Benutzungsschnittstellenprofils (UIP) | 86 |
| Tabelle 7 | Vorlage für die Beschreibung eines Bündels von Individualisierungsmustern | 88 |
| Tabelle 8 | Vorlage für die Beschreibung eines Individualisierungsmusters | 89 |
| Tabelle 9 | Variablen des Customization-Profiles (CP) | 90 |
| Tabelle 10 | Vorlage für die Beschreibung eines Bündels von Interaktionsmustern | 93 |
| Tabelle 11 | Vorlage für die Beschreibung eines Interaktionsmusters | 94 |
| Tabelle 12 | Vorlage für die Beschreibung eines Bedien- und Anzeigeelements | 96 |
| Tabelle 13 | Vorlage für die Beschreibung eines Adaptationsausführungsmusters | 103 |
| Tabelle 14 | Das ABS-Rahmenmodell stellt einige generische Interaktionssituationen zur Verfügung (alphabetische Reihenfolge) | 115 |
| Tabelle 15 | Fragestellungen für die Evaluierung des Ansatzes auf Grundlage der hergeleiteten Anforderungen | 125 |
| Tabelle 16 | Testscenario zur Evaluierung der automatischen Adaptivität | 133 |
| Tabelle 17 | Verschiedene Formen der gemischten Initiative im MyUI-System | 136 |
| Tabelle 18 | Technische Daten des Testaufbaus zur Messung der Prozesszeiten für Laufzeitanpassungen | 139 |
| Tabelle 19 | Testscenarien zur Evaluierung der kontinuierlichen Anpassungsfähigkeit während der Nutzung | 139 |
| Tabelle 20 | Ergebnisse der Zeitmessungen zur Evaluierung der kontinuierlichen Anpassungsfähigkeit während der Nutzung. Die dargestellten Prozesszeiten ergeben sich nach Abzug der Zeiten für Animationen und Nutzerbestätigungen von den gemessenen Gesamtzeiten. | 140 |
| Tabelle 21 | Bewertung der Wiederverwendbarkeit der Komponenten der ABS-Infrastruktur | 154 |
| Tabelle 22 | Interaktionssituationen und Interaktionsmuster, die zur Realisierung des Wiederverwendungsszenarios benötigt würden. Interaktionsmuster, die bereits bestehen und wiederverwendet werden können, sind in Fettschrift hervorgehoben. | 155 |
| Tabelle 23 | Nutzerpräferenzen für Adaptationsdialogmuster und Baseline (absolute Häufigkeiten) | 164 |
| Tabelle 24 | Können Adaptationsdialogmuster den Nutzer unterstützen, die Kosten automatischer Adaptationen zu kontrollieren? Die Tabelle zeigt absolute Häufigkeiten von Fällen, in denen die Probanden geringe subjektive Kosten berichten, obwohl sie gerade in einer Testbedingung mit hohen intendierten Kosten sind ($Kosten_{subj.} < Kosten_{int.}$) | 165 |
| Tabelle 25 | Absolute Häufigkeiten der Nutzerpräferenzen für die Dialogbedingungen unter verschiedenen experimentellen Kosten- und Nutzenbedingungen | 166 |

| | | |
|------------|--|-----|
| Tabelle 26 | Anzahl der Probanden (nach Rolle und Ort) zur Evaluierung der praktischen Eignung des Ansatz für die kommerzielle Software-Entwicklung | 168 |
| Tabelle 27 | Ablauf der Testsitzungen mit Entscheidungsträgern in Software-Unternehmen und Software-Entwicklern | 169 |
| Tabelle 28 | Deutsche Übersetzung der Fragebogen-Items, anhand derer Entscheidungsträger von Software-Unternehmen und Entwickler die praktische Eignung des Ansatzes für die kommerzielle Software-Entwicklung bewerten (Original in Englisch)..... | 171 |
| Tabelle 29 | Mittelwerte aller Bewertungen von Entscheidungsträgern und Entwicklern zu den betrachteten Kriterien. Die rechte Spalte stellt die Mittelwerte dar, wenn die negativ formulierten Aussagen nicht in die Auswertung eingehen. | 172 |
| Tabelle 30 | Ergebnisse der Evaluierung des Ansatzes anhand der hergeleiteten Anforderungen | 175 |

1 Einleitung

1.1 Adaptive Benutzungsschnittstellen zur Überwindung von Nutzungsbarrieren

Adaptive Benutzungsschnittstellen (ABS) ermöglichen eine individuelle Anpassung der Informationsdarstellung und Interaktionsmechanismen eines interaktiven Systems an unterschiedliche Nutzer¹ und unterschiedliche Nutzungsbedingungen. Im Gegensatz zu adaptierbaren Systemen erfolgt die Anpassung dabei automatisch und auf Initiative des Systems (vgl. Oppermann, 1994).

Trotz der erhöhten Komplexität in der Entwicklung von ABS und möglicher Probleme, die durch dynamische Veränderungen der Benutzungsschnittstelle entstehen können, bieten ABS insbesondere dann große Vorteile (vgl. Kühme & Schneider-Hufschmidt, 1993), wenn ...

- ... ein System von Nutzern mit unterschiedlichen Anforderungen genutzt wird. Beispiele umfassen u.a. Systeme zur Produktionsplanung und –steuerung, die von Ingenieuren, Technikern und Facharbeitern und ungelerten Hilfskräften mit unterschiedlichen Aufgaben, Kompetenzen und Vorkenntnissen genutzt werden; oder Konsumentenprodukte, die in verschiedenen Kulturkreisen von Nutzern mit unterschiedlichen kognitiven, motorischen und perzeptuellen Voraussetzungen oder mit Behinderungen genutzt werden.
- ... ein System von Nutzern genutzt wird, deren Anforderungen sich über die Zeit verändern. Beispiele sind zunehmende Nutzungskompetenzen beim Erlernen eines Systems, Entwicklung des Lernfortschritts bei E-Learning Applikationen und veränderliche kognitive, motorische und perzeptuelle Voraussetzungen im Zusammenhang mit Krankheitsverläufen oder altersbedingten Entwicklungsprozessen.
- ... ein System unter verschiedenen oder wechselnden Kontextbedingungen genutzt wird. Beispielsweise können je nach Interaktionssituation unterschiedliche Hilfestellungen (kontextsensitive Hilfe) oder das Angebot unterschiedlicher Handlungsoptionen besonders nützlich sein. Außerdem empfehlen sich möglicherweise je nach Nutzungssituation völlig unterschiedliche Ein- und Ausgabemodalitäten und –formate für ein und dasselbe System. Ähnliche Problemstellungen treten auf, wenn dieselbe Funktionalität auf verschiedenen Endgeräten angeboten wird.

Die technische Grundlage für die Realisierung einer ABS kann als Zusammenspiel von drei funktionalen Komponenten beschrieben werden (vgl. Weibelzahl, 2002, zusammengefasst nach Jameson, 2001 und Oppermann, 1994)²:

¹ Der Begriff „Nutzer“ wird in dieser Arbeit geschlechtsneutral verwendet und schließt stets weibliche und männliche Personen ein, die ein interaktives System nutzen.

² Eine ähnliche Gliederung treffen Kobsa, Koenemann und Pohl (2001) für die Personalisierung von Hypermedia-Anwendungen: Sie nennen die drei Funktionen „Akquisition“, „Repräsentation und sekundäre Inferenz“ und „Produktion“.

- Afferenz: Sammlung von verfügbaren und beobachtbaren Daten über den Nutzer und den Nutzungskontext
- Inferenz: Rückschluss auf relevante Eigenschaften des Nutzers und des Nutzungskontexts aufgrund der gesammelten Rohdaten
- Efferenz: Entscheidung über das Adaptationsverhalten des Systems und Ausführung der Anpassung

Die vorliegende Arbeit konzentriert sich auf die efferente Komponente. Es wird ein technischer Ansatz entwickelt, der es ermöglicht, auf der Grundlage einer vorhandenen Wissensbasis über Nutzer und Nutzungskontext eine individualisierte Benutzungsschnittstelle zu generieren und diese während der Laufzeit dynamisch an Veränderungen der Wissensbasis anzupassen. Die konkreten Mechanismen der Datenerhebung (Afferenz) sowie der Erstellung und Aktualisierung eines Nutzer- und Kontextprofils (Inferenz) werden dabei nur auf der Ebene von Abhängigkeiten und Wechselwirkungen mit der Gestaltung des Adaptationsverhaltens behandelt. Darüber hinaus werden nur Adaptationen der in Software realisierten Teile von Bediensystemen betrachtet. Veränderungen der Form, Größe oder Position physischer Eingabegeräte wie sie zum Beispiel bei Petrov (2012) oder Petrov und Maier (2009) unter dem Schlagwort „adaptiv variable Stellteile“ behandelt werden, sind nicht Gegenstand dieser Arbeit.

Eine weitere Fokussierung des Untersuchungsfelds dieser Arbeit betrifft die Zielsetzung der Adaptivität. Der zu entwickelnde Ansatz konzentriert sich auf Anpassungen einer Benutzungsschnittstelle, die auf die Überwindung von Nutzungsbarrieren zielen. Der Begriff der Nutzungsbarrieren ist eng mit dem Konzept der Zugänglichkeit (Accessibility) verbunden. In der deutschen Fassung der entsprechenden ISO Norm (DIN EN ISO 9241-171) wird Zugänglichkeit als die „Gebrauchstauglichkeit eines Produktes, einer Dienstleistung, einer Umgebung oder einer Einrichtung für eine in Bezug auf ihre Fähigkeiten möglichst weit gefasste Gruppe von Menschen“ (ISO, 2008a) definiert.

Der Bezug zur Gebrauchstauglichkeit (Usability) bringt das Ziel zum Ausdruck, für die gesamte Bandbreite möglicher individueller Fähigkeiten ein möglichst hohes Ausmaß an Effektivität, Effizienz und Zufriedenstellung im jeweiligen Nutzungskontext zu erreichen (vgl. ISO, 2010). Außerdem wird explizit darauf hingewiesen, dass der Begriff der Zugänglichkeit den gesamten Bereich der Fähigkeiten der Benutzer betrifft und nicht auf Benutzer beschränkt ist, die formell als behindert gelten. Neben Menschen mit Behinderungen werden insbesondere auch „ältere Menschen, die (...) in ihren physischen, sensorischen oder kognitiven Fähigkeiten eingeschränkt sind“ und „Menschen mit zeitweise auftretenden Beeinträchtigungen, wie z. B. Personen mit gebrochenem Arm oder solche, die ihre Brille vergessen haben“ in die Zielgruppe einbezogen. Schließlich noch „Menschen, die in bestimmten Situationen Schwierigkeiten haben, wie z. B. Personen, die in lauten Umgebungen arbeiteten oder aufgrund ihrer Beschäftigung momentan keine Hand frei haben“ (vgl. ISO, 2008a).

Damit ist das Konzept der Zugänglichkeit nicht auf eine vermeintliche Minderheit potenzieller Nutzer beschränkt, sondern besitzt eine hohe gesellschaftliche und wirtschaftliche Relevanz. Auch für interaktive Systeme, die in der Industrie als Arbeitsmittel eingesetzt werden, ist die Zugänglichkeit ein wesentlicher Erfolgsfaktor. Insbesondere die aktuelle Diskussion um eine Erhöhung des Renteneintrittsalters und die zunehmende Verbreitung von mobilen Geräten in verschiedenen Arbeitsumgebungen machen deutlich, wie wichtig die Beseitigung von Nutzungsbarrieren auch aus der Sicht der Arbeitgeber ist. Zusätzlich ergeben sich durch die

kürzer werdenden Produktzyklen heutiger Geräte immer häufiger Situationen, in denen sich die Nutzer mit neuen, ungewohnten Darstellungs- und Interaktionsformen konfrontiert sehen.

Demnach beschränkt sich der Gegenstandsbereich dieser Arbeit auf Anpassungen einer Benutzungsschnittstelle mit dem Ziel einer möglichst hohen Gebrauchstauglichkeit trotz Beeinträchtigungen durch individuelle Einschränkungen des Benutzers oder durch bestimmte Kontextbedingungen, die eine Bedienung erschweren. Die behandelten Kontextbedingungen umfassen dabei Merkmale der Nutzungsumgebung und Merkmale der technischen Ausstattung wie beispielsweise die Bildschirmgröße oder verfügbare Eingabegeräte. Zusammenfassend dienen Anpassungen der Benutzungsschnittstelle im Rahmen dieser Arbeit stets der Überwindung von Nutzungsbarrieren, die durch eine oder mehrere der folgenden Faktoren bedingt sind:

- Perzeptuelle Einschränkungen des Benutzers
- Motorische Einschränkungen des Benutzers
- Kognitive Einschränkungen des Benutzers
- Besondere Anforderungen durch die Nutzungsumgebung
- Besondere Anforderungen durch die technische Ausstattung des Benutzers

1.2 Aufbau der Arbeit

Abbildung 1 zeigt den Aufbau der Arbeit. Nach dieser Einleitung werden in Kapitel 2 auf Grundlage der aktuellen Fachliteratur die Anforderungen an ABS zur Überwindung von Nutzungsbarrieren hergeleitet. Bestehende Lösungsansätze werden vorgestellt und vor dem Hintergrund der beschriebenen Anforderungen diskutiert. In Abschnitt 2.7 wird die Analyse des Stands der Forschung und Technik zusammengefasst. In einer tabellarischen Übersicht wird dargestellt, in welchem Ausmaß ausgewählte Lösungsansätze die Anforderungen erfüllen.

In den Kapiteln 3 bis 8 wird der Ansatz beschrieben. Nach einem Überblick der zugrundeliegenden Prinzipien und Kernkonzepte in Kapitel 3 werden die Details des Lösungsansatzes vertieft.

In Kapitel 4 geht es um die Entwurfsmuster, die in diesem Ansatz als Bausteine der ABS-Generierung dienen. Es werden verschiedene Typen von Entwurfsmustern eingeführt. Relationen zwischen den Mustern werden definiert. Das Entwurfsmusterrepositorium wird beschrieben.

Danach wird der Prozess der Generierung und Anpassung der ABS im Detail behandelt. Für die drei Prozessschritte der ABS-Parametrierung (Kapitel 5), ABS-Vorbereitung (Kapitel 6) und ABS-Generierung und Adaptation (Kapitel 7) werden die wesentlichen Konzepte sowie die beteiligten Komponenten, Entwurfsmustertypen und Mechanismen beschrieben.

In Kapitel 8 wird das Abstrakte Applikations-Interaktions-Modell (AAIM) zur Modellierung der ABS vorgestellt. Nach einer Erläuterung der Funktionen, die das AAIM im Rahmen des Ansatzes erfüllt, wird die Notation des AAIM beschrieben, die eine Erweiterung der UML 2 – Zustandsdiagramme darstellt. Besondere Berücksichtigung findet dabei das Konzept der „Interaktionssituation“, das die Zustände des AAIM mit den Entwurfsmustern in Beziehung setzt. Schließlich wird ein Ausblick auf die Implementierung des AAIM in einer Entwicklungsumgebung gegeben.

Die Evaluierung des Ansatzes in Kapitel 9 erfolgt anhand der in Kapitel 2 hergeleiteten Anforderungen. Neben einer technischen Evaluierung mit einer Referenzimplementierung umfasst die Evaluierungsmethodik empirische Studien mit potenziellen Endnutzern und ABS-Entwicklern. Die Zusammenfassung der Evaluierungsergebnisse in Abschnitt 9.5 ermöglicht eine Bewertung des Fortschritts des Ansatzes gegenüber dem aktuellen Stand der Technik.

Kapitel 10 bietet eine Zusammenfassung der Arbeit. Ein Ausblick auf weitere Anwendungsfelder und zukünftige Arbeiten zur Weiterentwicklung des Ansatzes wird gegeben.

Im Anhang finden sich ergänzende Informationen und Illustrationen zur Veranschaulichung konkreter Entwurfsmuster und der Referenzimplementierung.

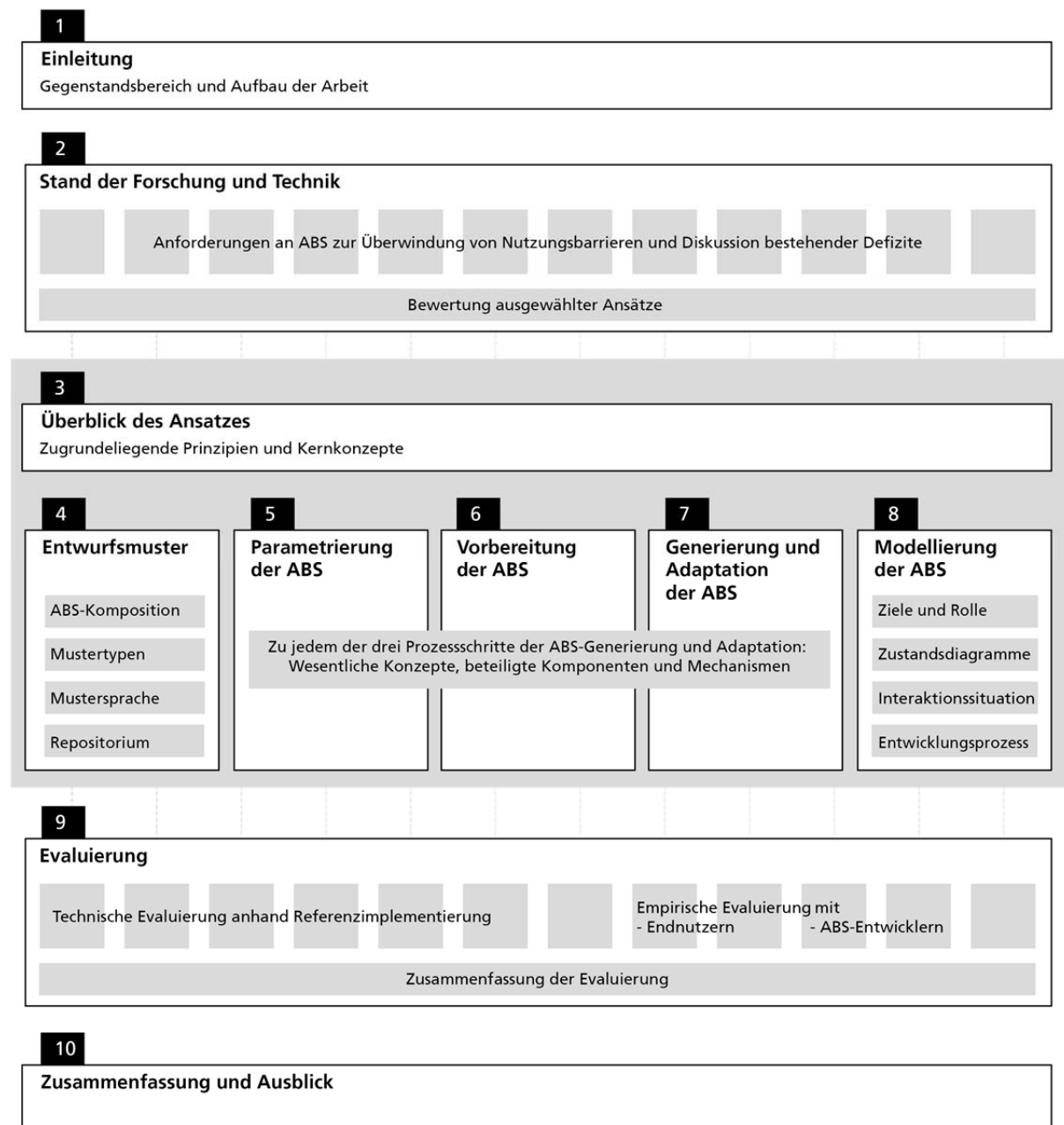


Abbildung 1 Aufbau der Arbeit

2 Stand der Forschung und Technik

ABS haben bereits eine lange Forschungstradition (vgl. z.B. Nielsen, 1993). Ihre Potenziale zur Steigerung der Gebrauchstauglichkeit und Zugänglichkeit (Barrierefreiheit) von interaktiven Systemen werden nach wie vor hoch eingeschätzt (vgl. Harper et al., 2008; Ringbauer et al., 2007; Aarts & Encarnação, 2006; und Savidis & Stephanidis, 2004). In ihrem ABILITY-BASED DESIGN – Ansatz nennen Wobbrock und seine Kollegen von der University of Washington die Adaptivität von Benutzungsschnittstellen als ein wesentliches Gestaltungsprinzip für barrierefreie Computersysteme (Wobbrock et al., 2011). Benutzungsschnittstellen, die sich dynamisch an individuelle Bedürfnisse sowie relevante Kontextbedingungen anpassen, können Nutzungsbarrieren möglicherweise schon beseitigen, bevor sie stören. Einige Autoren betrachten Adaptierbarkeit und Adaptivität heute als die einzigen realisierbaren Ansätze, um der enormen Heterogenität der Benutzer und dem damit verbundenen Anspruch personalisierter Benutzungsschnittstellen gerecht zu werden (vgl. Weld et al., 2003; Findlater & McGrenere, 2004). Trotz der großen Potenziale und der langen Forschungstradition sind heutige Ansätze jedoch noch weit von einer größeren Marktdurchdringung entfernt (vgl. Peissner et al., 2012a). Um auch jenseits des wissenschaftlichen Diskurses erfolgreich zu sein, werden Lösungsansätze für ABS zur Überwindung von Nutzungsbarrieren insbesondere die folgenden grundlegenden Anforderungen erfüllen müssen:

- *Adaptierbare und adaptive Benutzungsschnittstellen*
ABS zur Überwindung von Nutzungsbarrieren benötigen adaptive Mechanismen zur automatischen Anpassung an individuelle Bedürfnisse. Zwar ist die Adaptierbarkeit durch den Nutzer ein wichtiges Akzeptanzkriterium. Sie ermöglicht es dem Nutzer, aktiv zu einer optimalen Personalisierung der ABS beizutragen (vgl. Jameson, 2009; Fischer, 1993). Alleine reicht sie jedoch nicht aus. Denn gerade die Menschen, die aufgrund bestimmter Einschränkungen nicht in der Lage sind, ein System überhaupt zu nutzen, werden in einem rein adaptierbaren System wohl auch die Konfigurationsdialoge zur Beseitigung der Nutzungsbarrieren nicht bedienen können. Daher ist es für ABS zur Überwindung von Nutzungsbarrieren wichtig, Konfigurationsmöglichkeiten zur aktiven Einflussnahme durch den Nutzer (adaptierbar) mit automatischen system-initiierten Anpassungen (adaptiv) zu kombinieren. So können Nutzer und System zugunsten einer optimalen Anpassung kooperieren.
- *Selbstständige Anpassungen während der Nutzung*
Beschränkt sich das adaptive Verhalten auf die Phase der Initialisierung der ABS (vor der Nutzung), können trotz Anpassungen schwerwiegende Bedienprobleme während der Nutzung auftreten. Insbesondere wenn das zugrundeliegende Wissen über Nutzer, Umgebung und technische Ausstattung nicht vollständig oder nicht korrekt ist oder wenn sich die Bedingungen der Anpassung über die Zeit verändern – wie zum Beispiel veränderliche Licht- und Geräuschbedingungen oder abnehmende Fähigkeiten im Verlauf des Alterwerdens (vgl. Sloan et al., 2010). Daher ist es eine wesentliche Anforderung an ABS zur Überwindung von Nutzungsbarrieren, dass während der Nutzung die Wissensbasis permanent aktualisiert wird und sich die ABS unmittelbar entsprechend adaptiert. Nur so können auftretende Bedienprobleme und Nutzungsbarrieren direkt beseitigt werden.

- *Transparenz und Kontrollierbarkeit bei system-initiierten Adaptationen*
System-initiierte Adaptationen während der Nutzung können den Nutzern Probleme bereiten. Unerwartete Veränderungen der Benutzungsschnittstelle und das Gefühl des Kontrollverlusts können die Gebrauchstauglichkeit einschränken und zu Akzeptanzproblemen führen (Weld et al., 2003). Daher sind Mechanismen notwendig, die dafür sorgen, dass automatische Adaptationen für die Nutzer sowohl erkennbar und nachvollziehbar (transparent) als auch voll kontrollierbar sind.
- *Umfassende Anpassungen zur Überwindung von Nutzungsbarrieren*
ABS zur Überwindung von Nutzungsbarrieren benötigen einen konzeptionellen Rahmen und eine technische Infrastruktur für Adaptationen an verschiedene Nutzereigenschaften, Umgebungsbedingungen und technische Ausstattungen. Dies erfordert Anpassungen der Informationspräsentation, der Navigation und der Eingabemechanismen. Ansätze, die sich nur auf bestimmte Anpassungen oder bestimmte Einschränkungen beziehen sowie Ansätze für Benutzungsschnittstellen für multiple Plattformen lösen nur Teile des Problems, da sie nicht in der Lage sind, einen universellen Zugang für eine möglichst weit gefasste Gruppe von Menschen zu bieten.
- *Modulare, erweiterbare und offene Adaptationsmechanismen*
Derart umfangreiche Adaptationen erfordern eine große Wissensbasis oder komplexe Adaptationsalgorithmen. Dies wird mit einem monolithischen System kaum realisierbar sein. Modularität und Erweiterbarkeit des Ansatzes werden notwendig sein, um die Komplexität beherrschbar zu halten und um mit einer zunächst überschaubaren Teilmenge von Gestaltungslösungen zu starten, die dann sukzessive ausgebaut werden kann. Die Erweiterbarkeit der Adaptationsmechanismen sollte sich dabei nicht auf ein abgeschlossenes Team beschränken, sondern offen sein für externe Expertenbeiträge. Nur so kann mittelfristig ein Großteil der notwendigen Anpassungen ermöglicht werden, um dem Ziel einer weitest gehenden Zugänglichkeit nahe zu kommen.
- *Grundlegende Voraussetzungen für Akzeptanz in der Software-Entwicklung*
Nicht nur bezüglich der Nutzer, sondern auch bezüglich der Entwickler von ABS sind Akzeptanzhindernisse und Skepsis zu berücksichtigen (Gajos et al., 2010). Für den Erfolg eines Ansatzes für ABS ist es demnach auch wichtig, wesentlichen wirtschaftlichen und praktischen Anforderungen der kommerziellen Software-Entwicklung zu entsprechen (vgl. Lin & Landay, 2008). Neben einer effizienten Erstellung neuer adaptiver Applikationen umfasst dies insbesondere, dass die Entwicklungsansätze leicht erlernbar sind und den aktuellen Kompetenzprofilen und Entwicklungspraktiken in den Unternehmen entsprechen. Zusätzlich ist wichtig, dass die Entwickler die resultierende ABS gezielt selbst beeinflussen können und die zugrundeliegenden Mechanismen der ABS-Generierung nachvollziehen können.

In den folgenden Abschnitten dieses Kapitels werden diese wesentlichen Erfolgsfaktoren und Anforderungen von ABS zur Überwindung von Nutzungsbarrieren hergeleitet und detailliert beschrieben. Diese Anforderungen dienen als Struktur zur Darstellung des heutigen Standes der Forschung und Technik. Dabei werden insbesondere die Defizite der aktuellen Ansätze herausgearbeitet. Eine Kurzdarstellung der wichtigsten aktuellen Ansätze findet sich in Abschnitt 2.7.

2.1 Adaptierbare und adaptive Benutzungsschnittstellen

Für die Anpassung von Benutzungsschnittstellen existieren grundsätzlich zwei Ansätze: Je nachdem, ob eher der Benutzer oder das System die Anpassung kontrollieren, spricht man von *adaptierbaren* oder *adaptiven* Benutzungsschnittstellen (vgl. z.B. Dieterich et al., 1993; Kobsa et al., 2001; Weld et al., 2003; Findlater & McGrenere, 2004)³.

Für eine weitere Differenzierung unterscheiden Dieterich et al. (1993) zwischen der Initiative, dem Vorschlag, der Entscheidung und der Ausführung einer Anpassung, die jeweils entweder beim Benutzer oder beim System liegen können. Wenn alle diese vier aufeinander folgenden Stufen einer Anpassung vom Benutzer kontrolliert werden, handelt es sich um eine *adaptierbare* Benutzungsschnittstelle. Hingegen führt bei einer *adaptiven* Benutzungsschnittstelle das System all diese Schritte selbstständig aus (vgl. Kobsa et al., 2001). Auch beliebige Zwischenformen sind möglich. Ein Beispiel sind Systeme, die eine Adaptation vorschlagen, während die Entscheidung über die Ausführung beim Benutzer liegt („Nutzer-kontrollierte Adaptivität“). Wird die Kontrolle über die Anpassungen zwischen System und Benutzer geteilt, wird häufig von *gemischter Initiative* („mixed-initiative“) gesprochen (Findlater & McGrenere, 2004). Im Sinn der Definition von Horvitz (1999) wird in dieser Arbeit von Systemen mit gemischter Initiative gesprochen, wenn intelligente Automatismen und der Nutzer kooperieren, um die Ziele des Nutzers zu erreichen - wenn also Adaptierbarkeit und Adaptivität in einem System kombiniert sind.

Im Vergleich zu adaptiven Systemen sind adaptierbare Benutzungsschnittstellen deutlich weniger aufwändig in der Implementierung. Allerdings können nutzerinitiierte Änderungen der Benutzungsschnittstelle unter Umständen zu Inkonsistenzen führen. Der wichtigste Vorteil gegenüber adaptiven Systemen liegt darin, dass der Benutzer selbst die volle Kontrolle über eventuelle Anpassungen besitzt. So kann die Benutzungsschnittstelle genau auf individuelle Bedürfnisse und Präferenzen eingestellt werden und die selbst getätigten Veränderungen sind für den Benutzer stets nachvollziehbar (vgl. Fischer, 1993). Dieser Vorteil der vollen Kontrollierbarkeit bringt jedoch gleichzeitig Probleme mit sich. Die wenigsten Nutzer nehmen Anpassungsmöglichkeiten aktiv wahr (vgl. Gajos et al., 2010). Neben fehlender Zeit und mangelndem Interesse sind hierfür insbesondere auch Bedienschwierigkeiten verantwortlich (Mackay, 1991). Menschen mit wenig Erfahrung im Umgang mit Computern und Menschen mit individuellen Einschränkungen würden am meisten von Anpassungen der Benutzungsschnittstelle profitieren, da Standardkonfigurationen gerade ihnen häufig Probleme bereiten. Für sie stellen Konfigurationsdialoge eine zusätzliche, teilweise unüberwindbare Barriere dar.

Damit ist Adaptivität eine notwendige Eigenschaft von flexiblen Benutzungsschnittstellen, deren Anpassungen der Überwindung von Nutzungsbarrieren dienen sollen. Nur über systeminitiierte adaptive Mechanismen können Nutzungsbarrieren überwunden werden ohne mit zusätzlichem Interaktionsbedarf in Personalisierungsdialogen, die möglicherweise selbst Probleme bereiten, neue Hürden zu schaffen.

Benutzungsschnittstellen, deren Anpassungen ausschließlich auf systeminitiierten und intelligenten Adaptationen beruhen, bringen jedoch auch Probleme der Gebrauchstauglichkeit (Usability) mit sich. Als Beispiele hierfür nennt Jameson (2009) u.a. Probleme im Zusammenhang

³ Stephanidis et al. (1998b) beschreiben ein anderes Verständnis von adaptierbar und adaptiv (s. weiter unten in diesem Abschnitt)

mit einer Lernphase des Systems, sowie Kontrollverlust, mangelnde Verständlichkeit und mögliche Systemfehler. Eine gezielte Einbeziehung des Nutzers in die Anpassungsprozesse betrachtet er dabei als eine wesentliche Maßnahme zur Verbesserung der Effektivität von intelligenten Benutzungsschnittstellen (Jameson, 2009). Die Erkenntnis, dass Anpassungsstrategien mit gemischter Initiative am besten geeignet sind, teilen auch andere Autoren. Findlater und McGrenere (2004) betrachten vor allem die Heterogenität der adressierten Nutzergruppen als wesentliches Argument hierfür. In ihrer Nutzerstudie zum Vergleich von adaptierbaren und adaptiven Menüs finden sie heraus, dass die Mehrheit der Nutzer adaptierbare Menüs aufgrund ihrer besseren Kontrollierbarkeit bevorzugen, andere jedoch die automatische Anpassung klar favorisieren (Findlater & McGrenere, 2004). Weld et al. (2003) stellen zusammenfassend fest, dass neben automatischen Anpassungen an bekannte Kontextbedingungen wie z.B. Gerät, Ort und Aktivität, auch nutzergesteuerte Mechanismen angeboten werden sollten, um dem Nutzer stets die Kontrolle über den Anpassungsprozess zu ermöglichen. Zu einem ähnlichen Schluss kommen auch McGrenere und Kollegen aufgrund ihrer Feldstudie, in der die Teilnehmer dasselbe System (MS Word 2000) in einer adaptiven Version, bei der die Menüs je nach individueller Nutzung automatisch angepasst werden, und zugleich in einer adaptierbaren Version, bei der die Nutzer ausgehend von einem minimalen Funktionsumfang selbst neue Funktionen hinzufügen und wieder entfernen können, über einen Zeitraum von sechs Wochen nutzen (McGrenere et al., 2002).

Eine mögliche Ausprägung von gemischter Initiative ist die Unterstützung nutzerinitiiertener Personalisierungsaktivitäten durch Empfehlungen des Systems. Ein modernes Beispiel hierfür ist MICA („mixed-initiative customization assistance“), das die Personalisierung von MS Word 2003 unterstützt. Sobald der Nutzer einen Personalisierungsvorgang startet, bietet MICA Empfehlungen, die auf der Grundlage simulierter Interaktionen des jeweiligen Nutzers errechnet werden. Der Nutzer kann dann selbst entscheiden, welche der empfohlenen Anpassungen durchgeführt werden sollen (Bunt et al., 2009). Für die Unterstützung von Personalisierungsaktivitäten zeigt MICA interessante Möglichkeiten der Verbindung von Adaptierbarkeit und Adaptivität auf. Allerdings wird ein System, dessen Anpassungen die Zugänglichkeit für eingeschränkte Nutzer erhöhen soll, deutlich mehr Initiative übernehmen müssen, um über eine stärkere Benutzerführung und proaktives Systemverhalten auftretende Nutzungsbarrieren zu überwinden.

Im Bereich Barrierefreiheit gibt es noch keine systematischen Ansätze für gemischte Initiative. Die wenigen bestehenden adaptiven Systeme bieten teilweise keine oder nur unzureichende Mechanismen für eine sinnvolle Aufteilung der Kontrolle zwischen System und Nutzer.

Eines der am häufigsten zitierten adaptiven Systeme mit dem Ziel einer breiten Zugänglichkeit ist der AVANTI Web Browser. Er beinhaltet Mechanismen zur automatischen Anpassung an motorische Einschränkungen und Blindheit. Nutzerinitiierte Anpassungs- oder Kontrollmechanismen sind jedoch nicht vorgesehen. Dies drückt sich auch in der von den Autoren verwendeten Terminologie aus. Sie verwenden „adaptierbar“ im Zusammenhang mit systemgesteuerten Anpassungen zu Beginn einer Nutzungssequenz, während „adaptiv“ für systeminitiierte Anpassungen während einer Nutzungssequenz gebraucht wird. Das oben

eingeführte Konzept der „Adaptierbarkeit“ im allgemeinen Verständnis wird gar nicht verwendet (Stephanidis et al., 1998b).

Auch das Projekt GUIDE⁴ („Gentle user interfaces for elderly people“) konzentriert sich auf system-initiierte Anpassungen. Die Initialisierung des Nutzerprofils erfolgt über eine systemgeführte Applikation mit Fragen und Aufgaben zur Feststellung der Nutzerfähigkeiten und Präferenzen. Für die verwendeten Anpassungsprinzipien „Erweiterung“ (augmentation: Ergänzung der ursprünglichen Benutzungsschnittstelle durch alternative Modalitäten), „Adjustierung“ (adjustment: Anpassungen der allgemeinen visuellen Darstellung) und „Ersetzung“ (replacement: Anpassung der Inhalte, Layoutgestaltung und Präsentationsformate) werden keinerlei Möglichkeiten der Einflussnahme durch den Nutzer beschrieben (Coelho et al., 2011). Jedoch werden in GUIDE erste Ansätze entwickelt, um eingeschränkte Nutzer bei der Konfiguration ihrer Benutzungsschnittstelle durch interaktive Spiele oder einen geführten Dialog zu unterstützen. Einen ähnlichen Ansatz verfolgt auch das Forschungsprojekt MyUI⁵ („Mainstreaming Accessibility through Synergistic User Modelling and Adaptability“). Ergebnisse zur Validierung stehen jedoch in beiden Projekten noch nicht zur Verfügung.

Das von der Universität Washington entwickelte System SUPPLE bietet Anpassungen an die genutzten Geräte und an Nutzungsgewohnheiten sowie körperliche Einschränkungen des Nutzers. Die Adaptation erfolgt dabei während der Generierung der Benutzungsschnittstelle, bei der alle möglichen Designvarianten bezüglich ihres zu erwartenden Interaktionsaufwands für den jeweiligen Nutzer bewertet werden. Die Variante mit den geringsten zu erwartenden „Kosten“ wird ausgewählt und automatisch generiert. Während der Nutzung bietet SUPPLE aktive Einflussmöglichkeiten für den Nutzer. Elemente der Benutzungsschnittstelle können gelöscht und verschoben werden oder durch andere passende Bedienelemente ersetzt werden. Die vom Nutzer vorgenommenen Änderungen werden dann in einem „Customization Plan“ gespeichert, der auch später wieder vom Nutzer bearbeitet werden kann (Gajos et al., 2010). So stehen dem Nutzer umfangreiche Änderungsmöglichkeiten zur Verfügung, die ohne wiederholte Konfiguration auf verschiedene Geräte angewendet werden können. Andererseits erfordert diese Art der Adaptierbarkeit auf Seiten des Nutzers ein hohes Maß an Fertigkeiten im Umgang mit dem System und genaue Vorstellungen bezüglich der gewünschten Benutzungsschnittstelle. Für einen Ansatz, der die generelle Zugänglichkeit für *alle* Nutzergruppen erhöhen möchte, sind die vorgeschlagenen Konfigurationsmechanismen daher weniger geeignet.

Zusammenfassend kann man feststellen, dass es bislang nur wenige Ansätze gibt, die automatische systeminitiierte Anpassungen mit Möglichkeiten der Konfiguration und Einflussnahme durch den Nutzer kombinieren. Die bereits bestehenden Ansätze scheinen jedoch für den Anwendungsbereich der Barrierefreiheit nur bedingt geeignet. Insbesondere fehlen Lösungen, die durch ein hohes Maß an Systeminitiative den Nutzer von Konfigurationsaufgaben entlasten und zugleich umfassende Einflussmöglichkeiten bieten, die auch von Nutzern mit geringen Fähigkeiten beherrschbar sind.

⁴ <http://www.guide-project.eu/>

⁵ <http://www.myui.eu/>

2.2 Selbstständige Anpassungen während der Nutzung

Ein weiteres Unterscheidungsmerkmal betrifft die Dynamik und den Zeitpunkt der Anpassung. Typische Strategien reichen von einer *Anpassung vor der ersten Nutzung* (häufig auch konfigurierbare Systeme genannt) über eine *Anpassung zwischen zwei Nutzungssequenzen* bis hin zu einer *kontinuierlichen Anpassung während der Nutzung* (vgl. Dieterich et al., 1993 und Kobsa et al., 2001). Die letztgenannte Strategie der kontinuierlichen Anpassung ist zwar technisch am anspruchsvollsten, jedoch für die Sicherung einer universellen Zugänglichkeit die interessanteste. Die ersten beiden Strategien sind bezüglich ihrer Anforderungen an die effiziente Komponente (Entscheidung und Ausführung der Anpassung) kaum zu unterscheiden, da die Anpassung jeweils *vor der Nutzung* erfolgt. Ihr wesentlicher Unterschied liegt in der Dynamik des zugrundeliegenden *Nutzer- oder Kontextmodells*.

Gemäß der Definitionen des VUMS-Clusters⁶ wird ein *Nutzermodell* in dieser Arbeit als Menge von Variablen verstanden, die bestimmte Nutzereigenschaften repräsentieren, um den Nutzer eines Produkts hinreichend zu beschreiben. Ein Nutzermodell wird formal in einem maschinenlesbaren und einem menschenlesbaren Format beschrieben. Eine Instanz eines Nutzermodells wird *Nutzerprofil* genannt. Ein Nutzerprofil repräsentiert entweder einen spezifischen realen Nutzer oder einen Stellvertreter einer Gruppe realer Nutzer. Ein *Kontextmodell* enthält neben den Informationen über den Nutzer weitere Informationen über den Nutzungskontext, für die ein Einfluss auf die Interaktion zwischen Nutzer und Applikation angenommen wird. Damit ist neben dem Nutzermodell auch ein Umgebungsmodell und ein Gerätemodell Teil des Kontextmodells (Peissner et al., 2011a).

ABS, die während der Nutzung keine Anpassungen mehr vornehmen, sind zur Überwindung von Nutzungsbarrieren bei Weitem nicht so gut geeignet. Denn sie sind zu träge, um dynamisch auf kritische Ereignisse während der Nutzung und Veränderungen der Bedingungen, an die sich die ABS anpasst, zu reagieren. Der Bedarf einer kontinuierlichen Anpassung wird insbesondere anhand der folgenden beiden Argumente deutlich:

1. Adaptivität erfordert zuverlässiges Wissen über die Bedingungen, an die sich die Benutzungsschnittstelle anpassen soll, z.B. ein Nutzermodell oder ein Kontextmodell (vgl. Fischer, 1993; Paramythis et al., 2001). In der Praxis muss davon ausgegangen werden, dass dieses Wissen fehlerhaft oder unvollständig ist. Damit werden auch die Adaptationen, wenn sie auf falschen oder unvollständigen Annahmen beruhen, nicht immer in der Lage sein, die tatsächlich bestehenden Bedingungen und Anforderungen optimal zu unterstützen. Als Folge werden Probleme und Nutzungsbarrieren trotz bestmöglicher Anpassungsmechanismen immer wieder auftreten. Ein effektiver Ansatz für barrierefreie und adaptive Benutzungsschnittstellen sollte daher auftretende Probleme während der Interaktion erkennen und sofort durch eine entsprechende Anpassung darauf reagieren können (vgl. Stephanidis, 1998b).

⁶ VUMS steht für „Virtual User Modeling and Simulation“. Das VUMS-Cluster ist ein Verbund der europäischen Forschungsprojekte GUIDE, MyUI, VERITAS und VICON. All diese Projekte arbeiten daran, die Zugänglichkeit verschiedener Produkte und Anwendungen zu verbessern. Dabei pflegen die Projekte eine enge Kooperation, insbesondere in den Bereichen der Interoperabilität und der Standardisierung von Nutzer- und Kontextmodellen für eine verbesserte Zugänglichkeit (s. <http://www.veritas-project.eu/vums/>).

2. Ein weiteres Argument für Laufzeitanpassungen sind entwicklungsbedingte Veränderungen der individuellen Fähigkeiten und Bedürfnisse der Nutzer. Vorübergehende Einschränkungen - beispielsweise als Folge eines Unfalls oder eines Schlaganfalls - haben einen dynamischen Verlauf. Die plötzlich eintretenden Nutzungsbarrieren als auch das allmähliche Abklingen der Probleme sollte durch dynamische Anpassungsmechanismen der Benutzungsschnittstelle abgedeckt werden können. Ähnliche Anforderungen ergeben sich durch abnehmende physische, sensorische und kognitive Fähigkeiten im Verlauf des Älterwerdens (vgl. Sloan et al., 2010).

Adaptive Systeme mit einer derart hohen Volatilität erfordern dynamische Prozesse zur Nutzer- und Kontextmodellierung, die relevante Interaktionsereignisse erkennen und im Sinne einer Aktualisierung der Wissensbasis interpretieren. Diese Funktionen der Afferenz und Inferenz stehen jedoch nicht im Mittelpunkt dieser Arbeit. Dennoch erfordern Systeme, die in der oben beschriebenen Weise selbstlernende und selbstadaptive Mechanismen eng miteinander verbinden, einen intensiven Austausch zwischen Afferenz, Inferenz und der efferenten Komponente, die für die Anpassung der Benutzungsschnittstelle verantwortlich ist. Daher werden diese integrierten Mechanismen gemeinsam mit der Fähigkeit, individualisierte Benutzungsschnittstellen während der Laufzeit dynamisch anzupassen oder zu generieren, im Folgenden mit betrachtet.

In ihrer UNIFIED USER INTERFACE (UUI) Software-Architektur skizzieren Savidis und Stephanidis (2001) ein System, das sich bei der Initialisierung an ein bestehendes Nutzer- und Kontextprofil anpasst und mit Laufzeitanpassungen auf relevante Interaktionsereignisse reagiert. Grundlage der Anpassungen sind alternative *Dialogue Patterns*, die für eine bestimmte Teilaufgabe alternative Lösungen für unterschiedliche Nutzer- und Kontextausprägungen anbieten. Werden während der Interaktion in einer bestimmten Teilaufgabe Bedienprobleme detektiert, benachrichtigt der sogenannte User Information Server die Decision-Making Komponente, die dann unter bestimmten Umständen die aktuell aktivierten Dialogue Patterns durch entsprechende Alternativen ersetzt (Savidis & Stephanidis, 2001). Damit sind in der UUI Software-Architektur Anpassungen zur Laufzeit grundsätzlich vorgesehen. Jedoch gibt es eine Reihe von Defiziten bezüglich ihrer Umsetzung in adaptiven barrierefreien Anwendungen. Einerseits ist die von Savidis und Stephanidis beschriebene spezielle *Unified User Interface Design Methode* (Savidis & Stephanidis, 2004) in ihrer Implementierung aufwändig (s. Abschnitt 2.6.1). Das größte Defizit im Rahmen der oben diskutierten Anforderungen ist jedoch andererseits, dass zur Laufzeit keine Änderungen des Nutzerprofils und damit keine Anpassungen der Benutzungsschnittstelle an ein aktualisiertes Nutzerprofil vorgesehen sind. Während die Anpassungen bei der Initialisierung der Benutzungsschnittstelle auf einem eher „statischen“ Benutzerprofil basieren („*adaptability cycle*“), beziehen sich die Anpassungen während der Laufzeit auf die Feststellung von Ereignissen während der Interaktion, die keinen Niederschlag im Nutzerprofil finden („*adaptivity cycle*“) (Savidis & Stephanidis, 2001). Beispiele finden sich in der Beschreibung des AVANTI Systems, das eine Implementierung des UUI-Ansatzes darstellt (vgl. Stephanidis et al., 1998b). Damit kann zwar kurzfristig auf erkannte Bedienprobleme reagiert werden, die Erkennung hoher Bedienfehlerraten oder einer Desorientierung des Nutzers wird jedoch nicht im Sinn eines selbstlernenden Systems dem Nutzerprofil zugeführt, um für zukünftige Interaktionen die Bedienung zu erleichtern.

Auch im Zusammenhang mit PLASTIC USER INTERFACES werden Mechanismen einer dynamischen Anpassung der Benutzungsschnittstelle an sich ändernde Rahmenbedingungen

beschrieben. Zwar konzentriert sich dieser Ansatz im Wesentlichen auf die Unterstützung verschiedener, teilweise gleichzeitig genutzter Endgeräte. Dennoch sind die beschriebenen Konzepte und Ansätze auch für das Anwendungsfeld von barrierefreien ABS interessant. Das zugrundeliegende Rahmenmodell CAMELEON beinhaltet Komponenten wie den „*Situation Synthesizer*“ zur Interpretation der aktuell anliegenden Sensorinformationen sowie die „*Evolution Engine*“ und den „*Configurator*“ zur Planung und Durchführung von dynamischen Adaptationen der Benutzungsschnittstelle (Balme et al., 2004). Ein wesentliches Mittel für Laufzeitanpassungen stellen die sogenannten *Comets* („*Context of use Mouldable widgets*“) dar. *Comets* sind Interaktionselemente, die sich zur Laufzeit in unterschiedlichen Varianten zur Anzeige bringen oder ausgetauscht bzw. ein- oder ausgeblendet werden können, um die Benutzungsschnittstelle an veränderte Anforderungen anzupassen (Calvary et al., 2004).

Ein grundlegendes Problemfeld dynamischer Anpassungen während der Laufzeit stellt die zeitliche Performanz der Adaptationen dar. Das bereits oben genannte System SUPPLE zum Beispiel verfügt über ausgefeilte Mechanismen, um für die aktuellen Kontext- und Nutzervoraussetzungen das am besten geeignete Design zu ermitteln. Die eingesetzten Algorithmen zur Minimierung der jeweils entstehenden Kosten der Interaktion werden jedoch schnell so umfangreich, dass die resultierenden Ausführungszeiten bei bis zu 10-20 Minuten liegen (Gajos et al., 2010) und damit nicht für eine Neu-Generierung der Benutzungsschnittstelle während der Nutzung geeignet sind.

Zusammenfassend lässt sich feststellen, dass nur wenige Ansätze für eine kontinuierliche Adaptation während der Nutzung bestehen. Die meisten veröffentlichten Ansätze für adaptive Benutzungsschnittstellen konzentrieren sich lediglich auf Anpassungen während der Initialisierung: z.B. EGOKI (Abascal et al., 2011), OASIS (Leuteritz et al., 2009) und PERSONAL UNIVERSAL CONTROLLER (PUC, Nichols & Myers, 2009), der beispielsweise in HUDDLE eine Anwendung zur Laufzeit-Generierung von Benutzungsschnittstellen für verschiedene Endgeräte findet (Nichols et al., 2006b). Eine dynamische Anpassung während der Nutzung an eine kontinuierlich aktualisierte Wissensbasis bezüglich relevanter Eigenschaften des Nutzers, seiner Umgebung und der verwendeten Geräte ist bislang noch in keinem System umgesetzt.

2.3 Transparenz und Kontrollierbarkeit bei systeminitiierten Anpassungen

Gemessen an dem technischen Entwicklungsstand adaptiver Benutzungsschnittstellen, wurde bereits viel über die möglichen Probleme von system-initiierten Anpassungen veröffentlicht. In einem Überblicksartikel fasst Jameson die Diskussion über mögliche Usability-Probleme von intelligenten Benutzungsschnittstellen zusammen (Jameson, 2009). Zwei wesentliche Problemfelder, die auch von anderen Autoren häufig beschrieben werden, betreffen dabei die *Transparenz* und die *Kontrollierbarkeit* von systeminitiierten Anpassungen aus der Sicht der Nutzer (z.B. Fischer, 1993; Weld et al., 2003; Wobbrock et al., 2011).

2.3.1 Transparenz von systeminitiierten Anpassungen

Unter dem Begriff der *Transparenz* werden in dieser Arbeit die Eigenschaften einer ABS zusammengefasst, die dazu führen, dass automatische Anpassungen ausreichend erkennbar und nachvollziehbar sind und nicht zur Verwirrung oder Desorientierung des Nutzers führen. Ein wichtiger Aspekt ist dabei, dass der Nutzer erkennt, was sich verändert hat und die Beziehung

zwischen der Benutzungsschnittstelle vor der Anpassung und der Benutzungsschnittstelle nach der Anpassung klar wird (vgl. Dessart et al., 2011). Darüber hinaus kann es auch sinnvoll sein, den Nutzer in die Lage zu versetzen, die Ursachen einer Anpassung zu verstehen, um die Vorhersagbarkeit des Systemverhaltens zu unterstützen (vgl. Jameson, 2009).

Dabei können sich verschiedene Adaptationsstrategien in ihrer Verständlichkeit und Vorhersagbarkeit stark unterscheiden. Am Beispiel adaptiver Menüs beschreiben Findlater und Gajos mögliche Adaptationsstrategien, die von Veränderungen der Reihenfolge der einzelnen Menüitems, ihrer Größe, ihrer farblichen Darstellung bis hin zum Ausblenden oder zum verzögerten Einblenden selten genutzter Items oder der zusätzlichen Darstellung häufig genutzter Items in einem hervorgehobenen Bereich reichen. Die Transparenz der verschiedenen Lösungen hängt dabei unter anderem von der räumlichen Stabilität der Darstellung ab und davon, ob der Nutzer die Adaptation ignorieren kann („elective adaptation“) oder ob die Adaptation eine veränderte Nutzerreaktion erfordert („mandatory adaptation“). Beide Aspekte haben mit der Notwendigkeit einer Neuorientierung des Nutzers zu tun (Findlater & Gajos, 2009).

Dieser Effekt der Neuorientierung scheint in einem engen Zusammenhang mit der Zuverlässigkeit der Anpassungsmechanismen zu stehen. Fehler bei der Erkennung der Rahmenbedingungen, an die die Anpassung erfolgen soll (Afferenz und Inferenz), sind eine inhärente Gefahr für die Transparenz eines adaptiven Systems. Diese können leicht zu Adaptationen führen, die weder verständlich noch nachvollziehbar sind (vgl. Findlater & Gajos, 2009; Jameson, 2009). Tsandilas und Schraefel (2005) zeigen in einer Vergleichsstudie, dass je nach Gestaltung der Adaptation der Einfluss der Erkennungsgenauigkeit auf die Nutzerperformanz unterschiedlich groß sein kann. Bei einer perfekten Anpassung erreichen die Teilnehmer ihrer Studie die Interaktionsziele in adaptiven Auswahllisten schneller, wenn nicht benötigte Listenitems ausgeblendet werden, als wenn lediglich die vorgeschlagenen Items farblich hervorgehoben werden. Mit zunehmenden Fehlerquoten des Anpassungsmechanismus sinkt die Nutzerperformanz in beiden Adaptationsbedingungen. Dieser Effekt ist jedoch in der Bedingung mit den ausgeblendeten Items deutlich stärker ausgeprägt. Neben der Notwendigkeit, fälschlich ausgeblendete Items explizit wieder zur Darstellung bringen zu müssen, trägt auch die reduzierte Verlässlichkeit des Systems und das damit reduzierte Nutzervertrauen zu diesem Effekt bei. Eine erneute visuelle Orientierungsreaktion ist die Folge. So benötigen die Probanden bei geringerer Systemverlässlichkeit mehr Zeit, um das gesuchte Item wieder in der Liste aufzufinden - auch wenn es direkt in der angebotenen Auswahl enthalten ist (Tsandilas & Schraefel, 2005). Als mögliche Konsequenz ließe sich hieraus für das Design von Anpassungsmechanismen die Empfehlung ableiten, dass je nach Zuverlässigkeit der Kontexterkenkung unterschiedliche Gestaltungsvarianten vorgehalten werden sollten.

Die Bedingungen und die möglichen Probleme der Transparenz von ABS sind in der Literatur weitgehend beschrieben. Konkrete Gestaltungsansätze, die eine hohe Transparenz gewährleisten, finden sich dagegen nur vereinzelt. Diese widmen sich bislang nur einzelnen Teilaspekten des Problems. In dem bislang einzigen veröffentlichten Ansatz, um Nutzer auf systeminitiierte Adaptationen während der Laufzeit aufmerksam zu machen und ihnen die Veränderungen während der Anpassung zu verdeutlichen, schlagen Dessart und seine Kollegen animierte Übergänge vor (Dessart et al., 2011). Sie entwerfen eine Taxonomie von Adaptationsoperationen („Adaptation Operations“) wie z.B. Veränderungen der Größe oder Ersetzungen einzelner Bildelemente durch andere Elemente und ordnen den einzelnen Adaptationsoperationen geeignete Animationen zu. Zur Begründung ihrer Empfehlungen

verweisen sie auf Usability Guidelines, Erkenntnisse aus der kognitiven Psychologie und die aktuelle Praxis in Präsentationssoftware. Die Ergebnisse ihrer Nutzerstudie zur Validierung ihres Ansatzes deuten darauf hin, dass die animierten Übergänge helfen können, Desorientierung zu vermeiden und das Nutzervertrauen gegenüber dem System zu steigern (Dessart et al., 2011). Auch wenn bislang noch nicht auf gezielte Studien zur Validierung mit eingeschränkten Nutzern zurück gegriffen werden kann, scheinen die von Dessart und Kollegen vorgestellten animierten Übergänge zunächst geeignet, um auch in der vorliegenden Arbeit bei systeminitiierten Anpassungen während der Laufzeit für Kontinuität und Transparenz zu sorgen.

Eine weitere Möglichkeit zur Verbesserung der Transparenz sehen die Autoren darin, Erklärungen zu den vorgenommenen Adaptationsoperationen anzubieten (Dessart et al., 2011). Bunt und ihre Kolleginnen versuchen in ihrem MICA-System durch Erläuterungen für die vom System generierten Personalisierungsempfehlungen die Verständlichkeit und Nachvollziehbarkeit der intelligenten Benutzungsschnittstelle zu fördern (Bunt et al., 2009). Dabei wird zunächst erläutert, warum und mit welchem Ziel eine Empfehlung gegeben wird und welche Zeitersparnis durch die empfohlene Personalisierung möglich ist. Danach wird der Empfehlungsprozess kurz erklärt und angegeben, welche Faktoren dabei berücksichtigt werden. Die heterogenen Nutzerreaktionen im Benutzertest relativieren allerdings die Wichtigkeit der angebotenen Erläuterungen. Einige Nutzer wissen diese Form der Transparenz sehr zu schätzen und berichten ein gesteigertes Gefühl des Vertrauens, der Verständlichkeit und der Vorhersagbarkeit. Andere jedoch sehen keinen besonderen Mehrwert gegenüber der Systemvariante ohne Erläuterungen und kritisieren den zusätzlichen Zeitaufwand, den die ausführlichen Erklärungen mit sich bringen. Sie sind insbesondere an einer effizienten Bearbeitung ihrer primären Aufgabe interessiert und möchten für die Personalisierung möglichst wenig Zeit aufbringen (Bunt et al., 2009). Für adaptive Systeme, deren Anpassungen auf die Überwindung von Nutzungsbarrieren zielen, scheint daher ein eher sensibler und pragmatischer Umgang mit derartigen Transparenzmechanismen wichtig zu sein. Wie bereits in Abschnitt 2.1 am Beispiel des SUPPLE-Systems (Gajos et al., 2010) ausgeführt, müssen auch die Mechanismen, die Transparenz und Kontrollierbarkeit sicherstellen sollen, so gestaltet sein, dass sie für möglichst alle Nutzer komfortabel zugänglich sind und nicht breite Nutzerschichten überfordern.

Dies gilt insbesondere auch für Ansätze, die eine noch höhere Transparenz von adaptiven Systemen fordern. Kühme (1993) skizziert Eigenschaften eines Systems, die darauf abzielen, den Nutzer dabei zu unterstützen, ein angemessenes mentales Modell der Adaptationsmechanismen und –strategien zu entwickeln. Er fordert unter anderem, dass den Nutzern die Existenz eines Nutzermodells verdeutlicht wird und dass die Nutzer sowohl ihr Nutzerprofil als auch die Anpassungsmechanismen einsehen und nachvollziehen können. Mit dem Prinzip der Offenheit dieser Informationen und Strukturen verbindet Kühme auch die Forderung nach möglichst einfachen und leicht verständlichen Modellen und Mechanismen (Kühme, 1993).

Der aktuelle Stand der Forschung und Technik zum Thema Transparenz lässt sich nur schwer einheitlich zusammenfassen. Für die Verdeutlichung von system-initiierten Laufzeitanpassungen kann auf eine Taxonomie von animierten Übergängen zurückgegriffen werden (vgl. Dessart et al., 2011). Weiterführende Transparenzanforderungen können jedoch für den Anwendungsbereich der Barrierefreiheit kaum konkretisiert werden. Es ist noch völlig offen, in wie weit sich die Forderungen nach einem tiefen Nutzerverständnis mit den Fähigkeiten und Bedürfnissen der heterogenen Nutzergruppen am besten in Einklang bringen lassen. Eine leicht verständliche Darstellung der Nutzerprofile ist insbesondere im Bereich Barrierefreiheit auch aus ethischen und

datenschutzrechtlichen Gründen wünschenswert (vgl. auch Jameson, 2009). In einigen Forschungsprojekten würde dies jedoch bereits an der Komplexität und dem Vokabular der Nutzermodelle scheitern (vgl. Peissner et al., 2011a). Selbst bei dem pragmatischen Nutzermodell von MyUI, das auf medizinische Fachbegriffe und Einheiten verzichtet (vgl. Strnad et al., 2012b), stellt eine für alle Nutzer leicht verständliche Darstellung eine Herausforderung dar. Soweit möglich sollte eine adaptive Benutzungsschnittstelle Anhaltspunkte für die Gründe und Zusammenhänge von Laufzeitanpassungen zur Verfügung stellen. Eine vollständige Offenlegung der Adaptionsmechanismen kann für besonders interessierte und technikaffine Nutzer wertvoll sein, sollte jedoch die eigentliche Nutzung des Systems nicht erschweren. Da eine Darstellung der Adaptionsmechanismen insbesondere für die Transparenz auf Seiten der Entwickler von adaptiven Anwendungen wichtig ist, wird dieser Aspekt in Abschnitt 2.6.3 näher behandelt.

2.3.2 Kontrollierbarkeit von systeminitiierten Anpassungen

In Abschnitt 2.1 wird beschrieben, wie wichtig es für den Erfolg eines adaptiven Systems ist, eine geeignete Balance von Systeminitiative und Nutzerkontrolle zu finden. In einem adaptiven System mit gemischter Initiative sind effektive Einflussmöglichkeiten des Nutzers ein wesentliches Merkmal zur Schaffung der angestrebten Kontrollierbarkeit. Neben den bereits diskutierten nutzerinitiierten Personalisierungsmöglichkeiten spielt dabei die Gestaltung der systeminitiierten, automatischen Anpassungen eine wesentliche Rolle, da automatische Anpassungen leicht zu einem subjektiven Kontrollverlust auf Seiten des Benutzers führen können (vgl. Jameson, 2009).

Unter dem Begriff der *Kontrollierbarkeit* werden in dieser Arbeit die Eigenschaften einer ABS zusammengefasst, die es dem Nutzer ermöglichen, Kontrolle über systeminitiierte Anpassungen auszuüben. Dazu zählen beispielsweise Möglichkeiten, Anpassungen zu verhindern bzw. explizit zu befürworten, sowie Anpassungen in einer Vorschau zu betrachten, sie rückgängig zu machen oder durch eigene Anpassungen zu überschreiben (vgl. Wobbrock et al., 2011⁷).

Norman sieht bereits 1994 die größten Herausforderungen intelligenter Agententechnologie nicht in der technischen Weiterentwicklung, sondern in der Gestaltung der Interaktion zwischen Agent und Nutzer. Das Gefühl des Nutzers die Systemintelligenz kontrollieren zu können, ist für ihn ein zentraler Erfolgsfaktor, weil anders keine Nutzerakzeptanz zu erwarten ist (Norman, 1994). Auch die Vertreter der sogenannten PLASTIC USER INTERFACES erheben die Forderung nach unbedingter Nutzerkontrolle zu einem der Hauptprinzipien ihres Ansatzes für ABS. Alle Vorgänge an der Benutzungsschnittstelle sollen letztlich durch den Nutzer kontrollierbar bleiben – insbesondere auch systeminitiierte Anpassungen (Vanderdonck et al., 2008).

Trewin relativiert die Bedeutung der Kontrollierbarkeit automatischer Adaptationen, wenn sie der Überwindung von Nutzungsbarrieren dienen (Trewin, 2003). Sie weist darauf hin, dass sich die meisten Veröffentlichungen auf Adaptationen beziehen, die auf eine noch komfortablere oder noch effizientere Nutzung bereits vorhandener Funktionen abzielen. Im Anwendungsfall der Barrierefreiheit hingegen ermöglichen personalisierte Adaptationen jedoch häufig erst die Nutzung bestimmter Funktionen. Menschen mit perzeptuellen, kognitiven oder motorischen

⁷ Wobbrock und seine Kollegen fassen diese Eigenschaften gemeinsam mit den in Abschnitt 2.3.1 behandelten Transparenzeigenschaften unter dem Schlagwort „Transparency“ zusammen (Wobbrock et al., 2011)

Einschränkungen empfinden möglicherweise nur ein sehr geringes Maß an Kontrolle über eine nicht personalisierte Benutzungsschnittstelle. In diesen Fällen ist Trewin überzeugt, dass Anpassungsmechanismen, die nur wenig Eingreifen des Nutzers erfordern, die subjektiv empfundene Kontrollierbarkeit sogar erhöhen und nicht beeinträchtigen (Trewin, 2003).

Allerdings sind auch ABS zur Überwindung von Nutzungsbarrieren fehleranfällig, da bezüglich der Bedingungen, an die die Benutzungsschnittstelle angepasst wird, immer eine gewisse Ungewissheit besteht. Weld et al. (2003) weisen darauf hin, dass insbesondere fehlerhafte Adaptationen für den Nutzer mit „Kosten“ verbunden sind, die bei der Gesamtbetrachtung eines adaptiven Systems zusammen mit den potenziellen Vorteilen berücksichtigt werden müssen (Weld et al., 2003). Gajos, Czerwinski, Tan und Weld (2006) systematisieren die Kosten-Nutzen-Betrachtung von Adaptationen. Kosten entstehen demnach nicht nur bei fehlerhaften, sondern möglicherweise auch bei korrekten Anpassungen, z.B. durch die Instabilität der Darstellung und die damit verbundene Notwendigkeit einer Neuorientierung (Gajos et al., 2006; vgl. auch Findlater & Gajos, 2009). Findlater und Gajos stellen fest, dass ABS nur dann erfolgreich sein können, wenn sie dem Nutzer ein günstiges Kosten-Nutzen-Verhältnis bieten können, wenn also der subjektive Nutzen der Anpassungen die empfundenen Kosten überwiegt (Findlater & Gajos, 2009). Automatische Anpassungen können dabei einen wesentlichen Beitrag leisten - insbesondere wenn sie helfen, Nutzungsbarrieren zu überwinden. Jedoch stoßen intelligente Automatismen bei der Optimierung individueller Kosten-Nutzen-Verhältnisse an ihre Grenzen. Kontrollmechanismen, die dem Nutzer zur Verfügung gestellt werden, um das Adaptationsverhalten des Systems zu beeinflussen, können so als effektives Mittel zur Optimierung des subjektiven Kosten-Nutzen-Verhältnisses durch den Nutzer selbst verstanden werden und damit wesentlich zum Erfolg eines adaptiven Systems beitragen. Wichtige Voraussetzung hierfür ist jedoch, dass der Nutzer zum Zeitpunkt seiner Entscheidung Kosten und Nutzen der in Frage stehenden Anpassungen beurteilen kann.

Zur Gestaltung der Kontrollmechanismen von automatischen Adaptationen stehen bislang nur wenige wissenschaftliche Befunde zur Verfügung. Allerdings lassen sich einige Ansätze zur Systematisierung der Gestaltungsmöglichkeiten zusammenfassen:

Kobsa et al. (2001) unterschieden drei Ebenen der Nutzerkontrolle: (1) Auf der *generellen Ebene* kann der Nutzer alle Adaptationen generell ablehnen oder befürworten. (2) Auf der *Typen-Ebene* kann der Nutzer bestimmte Arten von Anpassungen ablehnen oder befürworten, z.B. ersetze Videos immer durch Standbilder. (3) Auf der *fallbasierten Ebene* entscheidet der Nutzer in jeder Situation einzeln über mögliche Anpassungen (Kobsa et al., 2001).

In dem pStars-Modell für PLASTIC USER INTERFACES unterscheiden Calvary et al. (2011) zwischen drei Ausprägungen der Kontrollierbarkeit: (1) Der Ausgangszustand beschreibt eine Situation, in der der Adaptationsprozess zwar beobachtet werden kann und dem Nutzer transparent ist, jedoch keine Einflussmöglichkeiten durch den Nutzer vorgesehen sind (*keine Kontrolle*). (2) In der mittleren Ausprägung kann der Nutzer eine vom System vorgeschlagene Adaptation bestätigen oder ablehnen („*Approbation*“). (3) Die höchste Form der Kontrollierbarkeit zeichnet sich durch eine vollständige Kontrolle des Anpassungsprozesses durch den Nutzer aus („*Specification*“) (Calvary et al., 2011).

Durch Jamesons Vorschlag, einfaches Nutzerfeedback nach einer erfolgten Adaptation einzuholen („*I don't like what just happened*“, Jameson, 2009), lässt sich die oben als zweite Variante aufgeführte „*Approbation*“ weiter unterteilen in eine Bestätigung vor und eine Bestätigung nach der Ausführung einer Adaptation.

Das Konzept der PLASTIC USER INTERFACES sieht ein „Supra-UI“ (Calvary et al., 2011) oder „Meta-UI“ (Coutaz et al., 2007) vor, über das der Nutzer das Adaptationsverhalten des Systems beeinflussen kann. In seiner extremen Ausprägung kann dieses Meta-UI als Entwicklungsumgebung für den Endnutzer betrachtet werden, mit dem er vorhandene Adaptationsregeln ändern und sogar neue Adaptationsregeln spezifizieren kann (Coutaz, 2010).

Lavie und Meyer vergleichen in einer empirischen Studie vier verschiedene Adaptationsstrategien mit unterschiedlichem Grad der Nutzerkontrolle (Lavie & Meyer, 2010): (1) Die manuelle Bedingung entspricht einem adaptierbaren System, bei dem alle Anpassungsoperationen vom Nutzer durchgeführt werden. (2) In der Bedingung „User Selection“ kann der Nutzer aus einer vom System vorgeschlagenen Vorauswahl von Handlungsoptionen wählen. (3) Die Bedingung „User Approval“ sieht eine explizite Bestätigung einer vom System initiierten Handlungsoption vor. (4) Die letzte Ausprägung entspricht einem voll adaptiven System, bei dem alle Stufen der Anpassung vom System initiiert und ausgeführt werden. Der Nutzer hat lediglich die Möglichkeit, den laufenden Adaptationsprozess abzubrechen. Die vier Strategien erweisen sich in der Unterstützung von begleitenden Aufgaben in einer Fahrsimulatorstudie als unterschiedlich effektiv. Bei Routine-Aufgaben, in denen das System eine korrekte Adaptation anbieten kann, bietet die voll adaptive Variante die größten Vorteile. In Situationen, in denen das System mit einer fehlerhaften Adaptation reagiert, sind jedoch die Varianten, die eine aktive Nutzerinteraktion erfordern (1, 2 und 3), überlegen. Vor dem Hintergrund der Fehleranfälligkeit adaptiver Mechanismen stellen die Autoren den Vorteil der Nutzerbeteiligung auch bei korrekten Anpassungen heraus (Varianten 2 und 3). Durch die Interaktionssituationen, die sich dadurch für die Nutzer ergeben, sind sie im Falle einer falschen Adaptation besser auf die manuelle Interaktion bzw. die Fehlerbehebung vorbereitet. Das beschriebene Kosten-Nutzen-Muster der verschiedenen Kontrollmechanismen zeigt sich in verstärkter Form in einer Vergleichsstudie, die mit älteren Nutzern durchgeführt wird. Ältere Nutzer profitieren im Fall korrekter Adaptationen stärker von automatischen Anpassungen, erleiden jedoch bei unangemessener Anpassung größere Nachteile (Lavie & Meyer, 2010).

Auch wenn das Experiment von Lavie und Meyer Adaptivität im Sinne einer automatisierten Aufgabenerledigung untersucht, lassen sich die Ergebnisse dennoch auf ABS zur Überwindung von Nutzungsbarrieren übertragen. Für Nutzer, denen die Interaktion schwer fällt, scheint der Nutzen einer korrekten Anpassung deutlich stärker ins Gewicht zu fallen als die Kosten des Kontrollverlusts über das Adaptationsverhalten des Systems. Dies deckt sich mit Trewins Behauptung einer geringeren Bedeutung der Kontrollierbarkeit für automatische Adaptationen, die die Barrierefreiheit verbessern (s. oben, vgl. Trewin, 2003). Allerdings wird der Erfolg von ABS zur Überwindung von Nutzungsbarrieren umso mehr davon abhängen, ob geeignete Mechanismen zur Verfügung stehen, die es Nutzern mit perzeptuellen, motorischen oder kognitiven Einschränkungen ermöglichen, fehlerhafte oder unangemessene Anpassungen zu verhindern oder rückgängig zu machen. Die Gestaltung und der passgenaue Einsatz dieser Kontrollmechanismen stellt eine große Herausforderung dar. Hierzu finden sich bislang keine anwendbaren Empfehlungen oder Ergebnisse in der Literatur.

2.4 Umfassende Anpassungen zur Überwindung von Nutzungsbarrieren

Effektive Ansätze für ABS zur Überwindung von Nutzungsbarrieren erfordern umfassende Anpassungen an unterschiedlichste

- Nutzerfähigkeiten und Bedürfnisse,
- Nutzungsumgebungen und
- technische Ausstattungen.

In der Regel wird dieses Tripel aus Nutzer, Umgebung und Plattform auch als Nutzungskontext („Context of Use“) zusammengefasst (vgl. Calvary et al., 2011). Zur Modellierung relevanter Nutzereigenschaften wird im Bereich der Barrierefreiheit häufig auf das internationale ICF-Klassifizierungsschema („International Classification of Functioning, Disability and Health (ICF)“) der WHO (World Health Organization) zurückgegriffen⁸. Das Umgebungsmodell umfasst die Menge der peripheren Objekte, Personen und Ereignisse, die einen Einfluss auf das System oder Nutzerverhalten haben können. So müssen zum Beispiel Umgebungsgeräusche berücksichtigt werden, wenn das System akustisches Feedback gibt (vgl. Calvary et al., 2003). Schließlich wird die technische Plattform bzw. technische Ausstattung modelliert. Meskens und Kollegen definieren eine technische Plattform als Kombination aus einem Hardware-Gerät, einem Betriebssystem und einem „User Interface Toolkit“, das die Technologien für die Implementierung der Benutzungsschnittstelle bereit stellt (Meskens et al., 2008). Jedoch muss berücksichtigt werden, dass möglicherweise mehrere Ein- und Ausgabegeräte parallel oder alternativ zur Verfügung stehen können. Daher ist es sinnvoll, wenn das Plattformmodell eher als Beschreibung aller verfügbaren Rechner-, Sensor-, Netzwerk- und Interaktionsressourcen verstanden wird (vgl. Calvary et al., 2011).

Insbesondere durch die Berücksichtigung von möglichen perzeptuellen, motorischen und kognitiven Einschränkungen der Nutzer wird klar, dass auf Ebene der Benutzungsschnittstelle umfangreiche Anpassungen nötig sein können, um entsprechende Nutzungsbarrieren zu überwinden. Diese umfassen⁹:

- Anpassungen der *Informationsausgabe*, u.a. Schriftgrößen und Farben, Bildschirmaufbau, Ausgabemodalitäten, etc.,
- Anpassungen der *Navigation*, d.h. der Pfade, die der Nutzer durch die Applikation nehmen kann, um eine bestimmte Funktion oder einen bestimmten Inhalt zu erreichen, und
- Anpassungen der *Eingabemechanismen*, wobei unterschiedliche Eingabegeräte und Eingabemodalitäten unterstützt werden können oder die Interpretation der Nutzereingaben angepasst werden kann.

Dabei können die Begriffe „Ausgabemodalität“ und „Eingabemodalität“ zu dem Begriff „Interaktionsmodalität“ (oder Modalität *M*) zusammengefasst werden, der in dieser Arbeit

⁸ <http://www.who.int/classifications/icf/en/>

⁹ in Anlehnung an gängige Taxonomien von Ebenen der Anpassung wie z.B. in Kobsa et al., 2001; Brusilovsky & Maybury, 2002 und Peng & Silver, 2007

gemäß der Definition von Vanderdonckt et al., 2008 als Tupel einer Interaktionssprache (L) und eines physischen Geräts (d) verstanden wird: $M ::= \langle L, d \rangle$. Dabei definiert die Interaktionssprache eine Menge wohlgeformter Ausdrücke mit einer bestimmten Bedeutung. So ist zum Beispiel die Modalität Spracheingabe durch das Tupel $\langle \textit{pseudo-natürliche Sprache NL, Mikrofon} \rangle$ bezeichnet, wobei NL durch eine spezifische Grammatik definiert ist (Vanderdonckt et al., 2008). Umfasst ein System zwei oder mehrere Eingabemodalitäten in Kombination mit zwei oder mehreren Ausgabemodalitäten, spricht man von einem „multimodalen System“ (vgl. Oviatt, 2003)

In dieser Arbeit wird in der Folge immer dann von „umfassenden Anpassungen“ gesprochen, wenn sich die Anpassungsmechanismen einer ABS auf perzeptuelle, motorische und kognitive Nutzerfähigkeiten sowie auf Eigenschaften der Nutzungsumgebung und der technischen Ausstattung beziehen und dabei Veränderungen in der Art der Informationsausgabe, der Navigation und der Eingabemechanismen bewirken können.

Bei derart vielfältigen Anpassungen ist problematisch, dass häufig Wechselbeziehungen zwischen Anpassungen in den verschiedenen Gestaltungsbereichen bestehen können.

Adaptationsmechanismen, die einen bestimmten Interaktionsaspekt verbessern sollen, können oft in anderen Gestaltungsbereichen zu Einbußen im Bedienkomfort oder gar zu Bedienproblemen führen (vgl. Findlater & Gajos, 2009). Die Verwendung größerer Bildschirmschriften als Reaktion auf eine festgestellte Sehbeeinträchtigung kann beispielsweise dazu führen, dass eine Auswahlliste auf zwei Bildschirmansichten aufgeteilt werden muss, was wiederum dazu führt, dass sich die Anforderungen an Aufmerksamkeit und Gedächtnis des Nutzers erhöhen (vgl. Peissner et al., 2011c).

Die Gestaltung und Realisierung von ABS, die sich während der Laufzeit in allen Gestaltungsbereichen umfassend an individuelle Anforderungen der Nutzer sowie ihrer technischen Ausstattung und Nutzungsumgebung anpassen, ist demzufolge eine hoch komplexe Herausforderung. Ansätze, die hierfür sowohl einen konzeptionellen Rahmen als auch eine technische Infrastruktur bieten, um umfassende Laufzeitanpassungen in allen genannten Gestaltungsbereichen zu steuern und zugleich Inkonsistenzen zu vermeiden, stehen heute nicht zur Verfügung.

AVANTI deckt alle drei der oben genannten Anpassungsebenen ab. „Syntactic Adaptability/Adaptivity Rules“ sind dabei für Anpassungen der Navigation zuständig, während „Lexical Adaptability/Adaptivity Rules“ Anpassungen der Ausgabe und Eingabemechanismen bewirken (Stephanidis et al., 1998b). Grundsätzlich wird auch der Anspruch vertreten, dabei sowohl auf Eigenschaften des Nutzers, der Umgebung und der technischen Ausstattung einzugehen. Die Existenz eines User Profils wird ebenso angenommen wie eine vergleichbare Beschreibung von Attributen der Umgebung und der technischen Plattform (Savidis & Stephanidis, 2004). Allerdings besitzt das bei Stephanidis und Kollegen dargestellte AVANTI Framework nur einen „User Modelling Server“ zur Pflege von Nutzerprofilen (Stephanidis et al., 1998a) und die beschriebenen Anpassungsmechanismen werden nur auf Nutzereinschränkungen und einige Ereignisse angewandt, die während der Interaktion erfasst werden, um auf Probleme des Nutzers hinzuweisen, wie z.B. Fehlerraten oder Zeiten ohne Nutzeraktion (Stephanidis et al., 1998b). Auch der Umfang der behandelten Nutzereinschränkungen ist unvollständig. Lediglich Nutzer mit motorischen Einschränkungen und Blinde werden berücksichtigt (Stephanidis, 1998b). Allerdings weisen die Autoren darauf hin, dass das den Adaptationen zugrundeliegende Regelwerk leicht erweitert oder angepasst werden kann (Stephanidis et al., 1998a).

Im EU-Projekt ASK-IT (Ambient Intelligence System of Agents for Knowledge-based and Integrated Services for Mobility Impaired Users) wurde eine Taxonomie von Interaktionsanforderungen durch verschiedene Nutzereinschränkungen, Umgebungs- und Nutzungsbedingungen und technischer Geräte erarbeitet, die als Ausgangspunkt einer Auswahl geeigneter Gestaltungselemente einer ABS dienen kann (vgl. Ringbauer et al., 2007). Im Folgeprojekt OASIS (Open architecture for Accessible Services Integration and Standardisation) wurde ein Rahmenmodell für ABS entwickelt, das im Wesentlichen auf einer *Adaptive Widget Library* basiert, die für einzelne Bedienelemente wie Buttons oder Menüs und umfangreicheren User Interface Komponenten wie einem Image Viewer oder einem File Upload Element veränderliche Attribute definiert, die dann individuell angepasst werden können. Die Anpassung struktureller Aspekte der Interaktionsgestaltung wie beispielsweise Anzahl und Folge von Bediensritten oder Anzahl von Optionen in einem Menü ist jedoch im OASIS-Ansatz nicht vorgesehen (vgl. Leuteritz et al., 2009).

Auch im Konzept der PLASTIC USER INTERFACES sind umfangreiche Anpassungen vorgesehen. Die Plastizität von Benutzungsschnittstellen ist definiert als deren Fähigkeit, sich an den Nutzungskontext anzupassen und dabei eine hohe Usability zu erhalten (z.B. Sottet et al. 2007b). Dabei wird der Nutzungskontext als ein strukturierter Informationsraum verstanden, der ein Nutzermodell, ein Modell der sozialen und physischen Umgebung und ein Modell der genutzten technischen Plattform umfasst (Sottet et al. 2007b; Calvary et al., 2003). Damit deckt der konzeptionelle Rahmen der PLASTIC USER INTERFACES alle für die Barrierefreiheit relevanten Aspekte ab. Die detaillierte Ausarbeitung sowie die beschriebenen Szenarien und Anwendungsbeispiele beziehen sich jedoch ausschließlich auf eine Plastizität, die auf eine komfortable Nutzung unterschiedlicher technischer Geräte und komplexer Interaktionsräume abzielt. Schon die Beschreibung des zugrundeliegenden technischen Referenzmodells CAMELEON-RT weist klar auf das ursprünglich adressierte Problemfeld des „Ubiquitous Computing“ hin. Dies wird unter anderem durch die Prinzipien der „UI Distribution“ (Verteilung einer Benutzungsschnittstelle auf mehrere parallel genutzte Interaktionsressourcen) und „UI Migration“ (vollständige oder teilweise Übertragung einer Benutzungsschnittstelle von einer technischen Plattform auf eine andere) deutlich (Balme et al., 2004). Auch die beschriebenen Fallstudien gehen in diese Richtung: CamNote bzw. CamNote++ (verschiedene Versionen einer Präsentationssoftware) und I-AM („Interaction Abstract Machine“) ermöglichen die Migration und Distribution zwischen PC und mobilen Geräten mit kleinen Bildschirmen bzw. zwischen Computern mit unterschiedlichen Betriebssystemen (Balme et al., 2004; Demeure et al., 2008). HHCS ist eine Heizungssteuerungssoftware für unterschiedliche Zielgeräte wie ein Wanddisplay, ein mobiles Gerät oder eine Java-basierte Armbanduhr (Sottet et al. 2007a). In einer neueren Fallstudie über adaptive E-Government Applikationen werden zwar einleitend die Anforderungen einer barrierefreien Technikgestaltung angeschnitten. Der beschriebene Demonstrator ermöglicht hingegen wieder nur die verteilte Nutzung eines Internet-Services über einen Computer und ein iPhone (Calvary et al., 2011). Mechanismen für Anpassungen an verschiedene Fähigkeiten und Bedürfnisse unterschiedlicher Nutzer werden in keiner Veröffentlichung zu PLASTIC USER INTERFACES näher beschrieben.

Viele andere Ansätze konzentrieren sich von vorne herein auf einen spezifischen Zielbereich der Anpassung. GUIDE beschränkt sich auf unterschiedliche Anforderungen und Einschränkungen älterer Nutzer (Coelho et al., 2011). Jason und Kollegen (Jason et al., 2010) berichten von einer ABS, die Call Center Agenten je nach individueller Nutzungskompetenz mehr oder weniger Benutzerführung und Bedienhilfen anbietet. Anpassungen an unterschiedliche Umgebungs- und

Nutzungsbedingungen werden von Kane, Wobbrock und Smith beschrieben (Kane et al., 2008), die eine mobile Applikation testen, deren Bedienelemente sich vergrößern, wenn der Nutzer sich fortbewegt. Lehmann und Kollegen (Lehmann et al., 2010) stellen eine Laufzeitumgebung für die Realisierung von Kontextanpassungen vor, die zum Beispiel auf Veränderungen im Abstand des Nutzers vom Bildschirm mit einer dynamischen Vergrößerung der Schrift reagieren kann. Auch für personalisierte Bedienhilfen werden Kontextinformationen genutzt. Neuere Beispiele hierfür sind AUGUR, eine kontext-sensitive Assistenz für formularbasierte Webapplikationen (Hartmann et al., 2009) und der Kalenderassistent SmartCal (Krzywicki et al., 2010). Schließlich gibt es eine ganze Reihe von ABS für verschiedene Geräte und Interaktionsmodalitäten (z.B. Ali et al., 2002; Nichols & Myers, 2009; Paterno et al., 2008). Das bereits oben erwähnte SUPPLE umfasst Anpassungen an verschiedene Endgeräte (Gajos & Weld, 2004) und körperlich-motorische Einschränkungen (Gajos et al., 2008).

Umfassende Anpassungen werden bislang lediglich in Form von konzeptionellen Rahmenmodellen behandelt. Duarte und Carriço (2006) stellen mit FAME („Model-based Framework for Adaptive Multimodal Environments“) eine Architektur für adaptive multimodale Applikationen vor. Neben einem Nutzermodell, einem Plattform- und Gerätemodell und einem Umgebungsmodell sehen die Autoren ein Interaktionsmodell vor, das die möglichen Darstellungs- und Interaktionskomponenten der Benutzungsschnittstelle beschreibt. Adaptationen werden nach Veränderungen in den Modellen durchgeführt und folgen definierten Adaptationsregeln. Jedoch beschränkt sich der Ansatz auf die Anpassung der Eingabe- und Ausgabemodalitäten. Anpassungen der Navigation werden nicht erwähnt. Ein weiteres Rahmenmodell für umfassende ABS wird von Motti (2011) skizziert. Allerdings ist ihre Beschreibung noch vage und konzentriert sich auf die wesentlichen Anforderungen und Grundzüge des Modells. Es wird deutlich, dass ihr Ansatz auf den PLASTIC USER INTERFACES basieren wird und dass die Anpassungen in einem Template beschrieben werden sollen, das an Entwurfsmuster für Benutzungsschnittstellen erinnert (vgl. Borchers, 2001).

Zusammenfassend lässt sich feststellen, dass noch kein Ansatz für umfassende Adaptationen an Nutzer, Umgebungsbedingungen und technische Ausstattung beschrieben wurde, der zugleich Veränderungen der Darstellung, Interaktion und Navigation ermöglicht. Einige konzeptionelle Rahmenmodelle wie AVANTI, ASK-IT oder PLASTIC USER INTERFACES skizzieren eine geeignete Infrastruktur, jedoch beziehen sich die konkreten Ausarbeitungen und Demonstratoren jeweils nur auf Teilbereiche eines umfassenden Ansatzes für ABS zur Überwindung von Nutzungsbarrieren. Damit fehlt es auch an konkreten Konzepten für das Management der komplexen geforderten Adaptationen und möglicher Zielkonflikte aufgrund der unterschiedlichen Anpassungsziele und betroffenen Gestaltungsbereiche.

2.5 Modulare, erweiterbare und offene Adaptationsmechanismen

Individualisierte Benutzungsschnittstellen zur Überwindung von Nutzungsbarrieren erfordern ein breites Spektrum unterschiedlicher Lösungen. Um für zahlreiche unterschiedliche Nutzeigenschaften, Umgebungsbedingungen und Zielgeräte passgenaue Lösungen anbieten zu können, muss ein geeigneter Ansatz für ABS in der Lage sein, die Interaktion mit ein und demselben System in ganz unterschiedlichen Ausprägungen anzubieten. Anpassungen in diesem Umfang erfordern eine große Basis an Adaptationsregeln oder komplexe Adaptationsalgorithmen. Ein monolithisches System wird diesen Anforderungen kaum gerecht werden können. Vielmehr bieten sich *modulare* Ansätze an, bei denen durch die Kombinationen

wiederkehrender Module eine hohe Lösungsvielfalt erzielt werden kann. Für die Ausarbeitung und Implementierung des Ansatzes ist zudem die *Erweiterbarkeit* der Adaptationsmechanismen wichtig. So kann zunächst mit einer überschaubaren Menge von Gestaltungslösungen begonnen werden, die dann nach und nach erweitert werden kann.

Die Formulierung passgenauer Gestaltungslösungen für unterschiedliche Anforderungen und Nutzereinschränkungen kann detailliertes und vielfältiges Expertenwissen erfordern. Die heute verfügbaren Gestaltungsnormen und Designrichtlinien im Bereich der Barrierefreiheit beziehen sich im Wesentlichen auf grundlegende Eigenschaften einer Benutzungsschnittstelle, die den Zugang für möglichst viele verschiedene Nutzer ermöglicht (z.B. ISO, 2008a; Caldwell et al. 2008). Viel schwieriger ist es, spezielles Gestaltungswissen für spezifische individuelle Besonderheiten in Erfahrung zu bringen. Daher ist es besonders interessant, für die Gestaltung, Verfeinerung und Vervollständigung der Adaptationsmechanismen einer Infrastruktur für barrierefreie ABS auch relevantes Expertenwissen von außerhalb des Entwickler- und Designerteams nutzen zu können. Dies erfordert neben der Erweiterbarkeit auch eine *Offenheit* der Adaptationsmechanismen, die sowohl eine öffentliche Zugänglichkeit als auch eine Offenheit für externe Beiträge beinhaltet.

Kein bislang veröffentlichter Ansatz kann all diese Anforderungen erfüllen. Regelbasierte Systeme bieten grundsätzliche gute Voraussetzungen für Modularität und Erweiterbarkeit. In den meisten regelbasierten adaptiven Systemen fehlt jedoch eine Struktur im Regelwerk, um Gruppen von Regeln zu funktionalen Modulen zusammenzufassen und die Interaktion zwischen den funktionalen Modulen über definierte Schnittstellen zu kontrollieren.

So werden in der *Decision Making Specification Language* (DMSL) der UNIFIED USER INTERFACES zwar syntaktische von lexikalen Anpassungsregeln unterschieden, die jeweils auf unterschiedliche Gestaltungsbereiche wirken, und „Adaptivity“-Regeln von „Adaptability“-Regeln, die zu unterschiedlichen Zeitpunkten (Initialisierung bzw. Laufzeit) ausgeführt werden (vgl. Savidis et al., 2005 und Stephanidis et al., 1998b). Eine weiterführende funktionale Kapselung innerhalb dieser Kategorien wird jedoch nicht beschrieben. Diese wäre jedoch wichtig, um bei der Erweiterung des Regelwerks die Kompatibilität zu sichern oder zumindest zu unterstützen.

Ähnlich verhält es sich bei FAME. Da für jede adaptierbare Interaktionskomponente der ABS eine eigene Adaptationsmatrix erstellt wird, ist grundsätzlich eine Modularität und Erweiterbarkeit gegeben. Jedoch werden Wechselwirkungen zwischen den Komponenten (und damit den Modulen, hier Adaptationsmatrizen) nicht behandelt. Mechanismen zur Sicherung einer Konsistenz über alle Komponenten einer ABS stehen damit nicht zur Verfügung (Duarte & Carriço, 2006).

Auch im Ansatz der PLASTIC USER INTERFACES stellt die Anpassung einzelner Darstellungs- und Bedienelemente („widgets“) ein wesentliches Element des Adaptationsverhaltens dar. Demeure, Calvary und Coninx (2008) beschreiben eine Benutzungsschnittstelle als einen Graph von *Comets* („COntext of use Mouldable widgETs“), die unterschiedliche Darstellungsformen für definierte Interaktionsaufgaben ermöglichen. In einem semantischen Netzwerk werden die Beziehungen der Adaptationsmechanismen über verschiedene Abstraktionsebenen der ABS-Beschreibung aufgegliedert. Dadurch wird das Gestaltungswissen für verschiedene Kontextbedingungen expliziert und zugleich wird die Möglichkeit geschaffen, durch karthetische Produktbildung zwischen verschiedenen Gestaltungsräumen („dialog spaces“) vielfältige Lösungen zu komponieren (Demeure et al. 2007). Auch bezüglich der Offenheit der Adaptationsmechanismen bieten die Arbeiten zu PLASTIC USER INTERFACES interessante Konzepte, die jedoch angesichts

der oben genannten Anforderungen nicht ausreichend erscheinen. Einerseits sehen die beschriebenen Ansätze und Tools weitreichende Einflussmöglichkeiten durch Applikationsentwickler und sogar Endnutzer vor. Der „Comets Inspector“ bietet beispielsweise neben der Möglichkeit, die verfügbaren Designalternativen (Comets) zu priorisieren, auch noch weitere Möglichkeiten, das Adaptationsverhalten des Systems während der Entwicklung und während der Laufzeit zu beeinflussen (Demeure et al., 2007). Darüber hinaus erweitert Coutaz das Konzept der „Open Adaptiveness“, bei dem das Adaptationsverhalten eines Systems externe Daten oder externe Mechanismen einbezieht (Calvary et al., 2004), auf eine aktive Beteiligung des Endnutzers bei der Verfeinerung und Erweiterung des Regelwerks (Coutaz, 2010). Andererseits sind jedoch die Adaptationsregeln selbst nicht öffentlich zugänglich. In der Literatur finden sich keine Hinweise auf eine öffentliche Möglichkeit, die Adaptationsregeln explizit zu bearbeiten oder neue Regeln oder Comets beizusteuern.

Im SUPPLE-System werden Anpassungen der Benutzungsschnittstelle nicht durch Regeln sondern durch einen Algorithmus zur Optimierung einer Kostenfunktion gesteuert. Die Kostenfunktion und deren Komponenten werden von Gajos, Weld und Wobbrock (2010) ausführlich und nachvollziehbar offengelegt. Die Kostenfunktion kann in der beschriebenen Form Eigenschaften der genutzten Interaktionsgeräte sowie individuelle Gewohnheiten und motorische Einschränkungen der Nutzer berücksichtigen. Erweiterungen der Kostenfunktion, um auch perzeptuelle und kognitive Nutzereigenschaften oder Umgebungsbedingungen abzudecken, wären grundsätzlich möglich. Die bereits diskutierten Performanzprobleme (vgl. Abschnitt 2.2) schränken derartige Erweiterungen jedoch stark ein.

Insgesamt haben die behandelten Eigenschaften der Modularität, Erweiterbarkeit und Offenheit in der Literatur zur Generierung von ABS bisher noch wenig Beachtung gefunden (vgl. Motti, 2011). Während modulare und erweiterbare Strukturen in einigen Systemen teilweise angelegt sind (neben den oben genannten auch ASK-IT, vgl. Ringbauer et al., 2007 und OASIS, vgl. Leuteritz et al., 2009), fehlt es insbesondere an einer Offenheit der Adaptationsregeln und –mechanismen, um gezielt externe Experten an deren Verfeinerung und Vervollständigung zu beteiligen.

2.6 Grundlegende Voraussetzungen für Akzeptanz in der Software-Entwicklung

Neben den in Abschnitt 2.3 beschriebenen Akzeptanzhindernissen auf Seiten der Endnutzer besteht auch bei Gestaltern und Entwicklern eine gewisse Skepsis gegenüber adaptiven und automatisch generierten Benutzungsschnittstellen (vgl. Gajos et al., 2010). Die in der Literatur ausführlich behandelten Herausforderungen umfassen hierbei insbesondere die in den folgenden Unterabschnitten näher beschriebenen Aspekte:

- Effizienz der Entwicklung einer adaptiven Benutzungsschnittstelle,
- Einstiegshürde und Kompetenzanforderungen auf Seiten der Entwickler und
- Nachvollziehbarkeit und Kontrollierbarkeit der Prozesse zur Erzeugung der resultierenden Benutzungsschnittstelle.

Überwiegend hängen diese Herausforderungen mit dem Ansatz modellbasierter Benutzungsschnittstellen zusammen (vgl. Szekely, 1996), der in der Regel bei der Entwicklung von ABS zum Einsatz kommt (vgl. Cockton, 1987). Dabei dient ein abstraktes Modell der Benutzungsschnittstelle als Grundlage für die Generierung der verschiedenen möglichen

Varianten einer ABS. Das abstrakte Modell beschreibt die Gemeinsamkeiten aller möglicher Varianten und ist damit unabhängig von konkreten Darstellungs- und Interaktionsmechanismen.

Insbesondere der wachsende Bedarf an Benutzungsschnittstellen, die auf verschiedenen Endgeräten und Zielplattformen laufen, hat in den letzten Jahren die Entwicklung von abstrakten Beschreibungssprachen für generierte Benutzungsschnittstellen vorangetrieben. Die wichtigsten Ansätze umfassen dabei MARIA (Paterno et al., 2009) und das früher veröffentlichte TERESA (Paterno et al., 2008) mit CONCUR TASK TREES (Paterno, 1999), PERSONAL UNIVERSAL CONTROLLER (PUC, Nichols & Myers, 2009) wie beispielsweise in HUDDLE (Nichols et al., 2006b) und UNIFORM (Nichols et al., 2006a) eingesetzt, sowie UIML (User Interface Markup Language) und CANONICAL ABSTRACT PROTOTYPES (CAP) mit den neuesten Weiterentwicklungen in CAP3 (Van den Bergh et al., 2011) (nach Peissner et al., 2012a).

Die Arbeit mit abstrakten Modellen ist teilweise nur schwer mit der heutigen Praxis bei der Gestaltung und Entwicklung von Benutzungsschnittstellen zu vereinbaren (vgl. Lin & Landay, 2008). Daher wird der Erfolg eines Ansatzes für ABS auch davon abhängen, inwieweit er grundlegende Anforderungen von Software entwickelnden Unternehmen erfüllen kann.

2.6.1 Effiziente Entwicklung einer adaptiven Benutzungsschnittstelle

Eine wesentliche Voraussetzung für die Adoption neuer Software-Entwicklungsansätze in der Praxis ist ihre Wirtschaftlichkeit und insbesondere ihre ressourceneffiziente Umsetzbarkeit. Grundsätzlich versprechen modellbasierte Ansätze zur Generierung von Benutzungsschnittstellen enorme Einsparpotenziale. Vor allem wenn verschiedene Varianten einer Benutzungsschnittstelle benötigt werden, können Automatismen dem Entwickler wesentliche Aufgaben abnehmen (vgl. Myers et al., 2000 für unterschiedliche Plattformen und Endgeräte). Die Benutzungsschnittstelle wird nur einmal in einer abstrahierten Form spezifiziert, die die Grundlage für die automatische Generierung der verschiedenen konkreten Ausprägungen darstellt. Auch nachträgliche Änderungen am Modell können automatisch auf alle Varianten der Benutzungsschnittstelle übertragen werden (Meskens et al., 2008).

Um die Effizienzpotenziale von automatisch generierten Benutzungsschnittstellen voll ausschöpfen zu können, sind insbesondere die folgenden beiden Anforderungen relevant:

- Prägnanz und Sparsamkeit
Der Ansatz sollte die manuell zu erstellenden Artefakte (Modelle) bezüglich ihres Umfangs und ihrer Komplexität auf ein Minimum reduzieren (vgl. Trewin et al., 2002). Redundanzen in den zu erstellenden Modellen sind zu vermeiden.
- Wiederverwendbarkeit
Die Generierung und Adaptation der Benutzungsschnittstelle sollte möglichst weitgehend auf generische und wiederverwendbare Mechanismen und Komponenten zurückgreifen. Damit müssen wiederkehrende Bestandteile nicht für jede neue Applikation neu erstellt werden, sondern können im abstrakten Modell lediglich referenziert werden. Eine wesentliche Grundvoraussetzung hierfür ist eine saubere Trennung der Applikationslogik von den Adaptationsmechanismen der Benutzungsschnittstelle.

Damit ist die Kernfrage dieses Abschnitts, wie umfangreich der notwendige Beitrag des Entwicklers für den Prozess der ABS-Generierung ist. Mit steigendem Automatisierungsgrad der ABS-Generierung ist ein geringerer Entwicklungsaufwand zu erwarten. Andererseits kann ein

hoher Automatisierungsgrad sowohl die ergonomische als auch ästhetische Qualität der generierten Benutzungsschnittstelle beeinträchtigen (vgl. Myers et al., 2000; Gajos et al., 2010). Darüber hinaus können sich die Entwickler in ihren Einflussmöglichkeiten zu stark eingeschränkt fühlen. Diese Problematik wird in Abschnitt 2.6.3 behandelt.

Bei den meisten bisherigen Ansätzen für ABS stehen die beiden oben genannten Anforderungen nicht im Mittelpunkt der Bemühungen. Insbesondere die UNIFIED USER INTERFACE DESIGN METHODE (Savidis & Stephanidis, 2004) erfordert einen hohen Gestaltungsaufwand. Ihr Prinzip des „Polymorphismus“ reicht bis in die hierarchische Aufgabenzerlegung, die zu Beginn der Gestaltungsaktivitäten durchgeführt wird. Für jede unterstützte Nutzerinteraktion einer zu entwickelnden Anwendung ist eine sogenannte polymorphe Aufgabenzerlegung (*Polymorphic Task Decomposition*) vorgesehen, die die Grundlage verschiedener Interaktionsdesignvarianten für unterschiedliche Nutzer- und Kontextbedingungen darstellt. Damit ergibt sich für den Designer oder Entwickler einer neuen adaptiven Anwendung die anspruchsvolle Aufgabe, nicht mehr nur eine Lösung zu erarbeiten, sondern einen gesamten Designraum verschiedener Lösungen für jede einzelne Nutzeraufgabe. Durch die enge Bindung der Designlösungen an die speziellen Nutzeraufgaben muss dieser aufwendige Gestaltungsprozess für jede neue Anwendung erneut durchgeführt werden. Ein generischer Ansatz mit wiederverwendbaren Lösungskomponenten könnte einen deutlichen Effizienzgewinn erzeugen.

Ähnlich verhält es sich mit FAME. Zur Definition der Adaptationsregeln erstellt der Anwendungsentwickler für jede adaptierbare Komponente eine „Verhaltensmatrix“ (*Behavioral Matrix*), die das jeweilige Adaptationsverhalten beschreibt und Informationen über das bisherige Nutzerverhalten speichert, um eine Optimierung der Anpassungsregeln zu unterstützen. Diese Verhaltensmatrizen sollen helfen, das Adaptationsverhalten des Systems für den Entwickler zu veranschaulichen und übersichtlicher darzustellen. Jedoch müssen für jeden denkbaren Nutzungsfall nach einer Identifikation der anpassbaren Komponenten die spezifischen Verhaltensmatrizen erstellt werden (Duarte & Carriço, 2006). Mechanismen für eine Wiederverwertbarkeit oder Konsistenzsicherung der Verhaltensmatrizen werden nicht beschrieben.

Bei SUPPLE ist der Ausgangspunkt der Generierung einer ABS eine *Functional Interface Specification*, die die Menge der auf der Benutzungsschnittstelle darzustellenden Objekte („*Interface Elements*“) mit Ihren Datentypen und Wertebereichen sowie weitere vom Gestalter definierte Beschränkungen („*Interface Constraints*“) enthält (Gajos et al., 2010). Durch gerätespezifische *Widgets* werden die abstrakten *Interface Elements* in konkrete Bedien- und Anzeigeelemente einer Benutzungsschnittstelle überführt. Dabei werden bereits die spezifischen Bedingungen des verwendeten Geräts berücksichtigt. Die Transformation der *Functional Interface Specification* in eine angepasste Benutzungsschnittstelle ist in der Regel nicht eindeutig, da es für jedes *Interface Element* mehrere konkrete Darstellungsmöglichkeiten geben kann. Die Auswahl der am besten geeigneten Darstellungsform erfolgt über die Minimierung einer Kostenfunktion, die in Abhängigkeit des aktuellen Nutzungskontexts den erwarteten Interaktionsaufwand beschreibt. Bei der Entwicklung einer neuen Applikation beschränkt sich der Aufwand lediglich auf die Definition einer *Functional Interface Specification*. Die gerätespezifischen *Widgets* sind hingegen generisch und können für jede neue Anwendung wiederverwendet werden. Unter der Voraussetzung einer ausreichenden Menge von *Widgets* muss sich der Entwickler einer neuen ABS somit nicht mit den verschiedenen Anpassungsmöglichkeiten auseinandersetzen (Gajos & Weld, 2004).

Einen ähnlichen Ansatz verfolgt OASIS mit einer *Adaptive Widget Library*. Diese Bibliothek wieder verwendbarer adaptiver Bedienelemente befreit die Entwickler von der Notwendigkeit, sich über das adaptive Verhalten der Benutzungsschnittstelle Gedanken zu machen. Die Entwickler können die *Adaptive Widgets* wie herkömmliche Bedienelemente in ihr Programm einbinden. Zusätzlich deuten Leonidis und Kollegen (2012) die Möglichkeit an, dass erfahrenere Entwickler das Adaptationsverhalten der Bedienelemente modifizieren können (Leonidis et al., 2012).

In vielen Ansätzen zur Generierung von ABS erfolgt die Modellierung der Benutzungsschnittstelle auf drei verschiedenen Abstraktionsebenen: (1) Ein *Task and Object Model* bzw. *Task and Concepts Model* beschreibt die Aufgaben des Nutzers und die wesentlichen vom Nutzer manipulierten Objekte bzw. Konzepte des Anwendungsbereichs. (2) Ein *Abstract User Interface (AUI) Model* repräsentiert die Struktur und den Inhalt einer Benutzungsschnittstelle unabhängig von den Interaktionsmodalitäten der Ein- und Ausgabe. Dabei wird auf abstrakte Interaktionsobjekte bzw. Interaktoren zugegriffen, die nur die funktionale Bedeutung eines Bedienelements festlegen, dessen Erscheinungsbild und Interaktionsverhalten jedoch nicht spezifizieren. (3) Ein *Concrete User Interface (CUI)* stellt eine Konkretisierung eines AUI für einen spezifischen Nutzungskontext dar. Es definiert das Erscheinungsbild und Verhalten einer Benutzungsschnittstelle vollständig. Im Unterschied zum *Final User Interface (FUI)* ist es jedoch noch unabhängig von einer konkreten Software-Technologie zur Implementierung (vgl. z.B. Mori et al., 2004; Paterno et al., 2009; Szekely, 1996 oder Sottet et al., 2007a)

Ansätze wie TERESA (Mori et al., 2004), die die manuelle Erstellung all dieser drei Modelle erfordern, erscheinen aufwändig. Insbesondere wenn für verschiedene Kontextbedingungen mehrere CUIs zur Konkretisierung des AUI erstellt werden müssen. Mit den sogenannten *Comets* („COntext of use Mouldable widgETs“) steht bei den PLASTIC USER INTERFACES ein Mechanismus zur Verfügung, der hier eine deutliche Entlastung ermöglicht. Ein Comet umfasst verschiedene konkrete Interaktionsobjekte, die dieselbe Nutzeraufgabe auf unterschiedliche Weise realisieren. Während der Laufzeit ermöglichen Comets eine Anpassung der Benutzungsschnittstelle auf Ebene einzelner Interaktionsobjekte (Calvary et al., 2004; Demeure et al., 2008) und führen damit die Generierung eines CUI bzw. FUI automatisch aus (vgl. Coutaz, 2010). Insgesamt lassen die Autoren der PLASTIC USER INTERFACES jedoch (noch) offen, wie ein konkreter modellbasierter Entwicklungsprozess aussehen sollte. Transformationen zwischen den Modellen sind grundsätzlich vertikal in beide Richtungen sowohl automatisch als auch manuell möglich (Calvary et al., 2004). Erst kürzlich wurden erste Ansätze für eine Designmethode (Ceret, 2011) und ein Entwicklungswerkzeug (Frey et al., 2012) veröffentlicht, die eine Konkretisierung bzw. eine Effizienzsteigerung der Vorgehensweise zum Ziel haben.

Neben der Analyse des Entwicklungsaufwands verlangt eine Effizienzbetrachtung auch die Berücksichtigung der Effektivität der heutigen Ansätze und damit u.a. die Fragen, welche Adaptationen die einzelnen Ansätze ermöglichen und welche Art und Komplexität von interaktiven Systemen sie abbilden können. Myers, Hudson und Pausch (2000) beschreiben in ihrem Überblicksartikel „Past, Present, and Future of User Interface Software Tools“ einen Zusammenhang zwischen „Threshold“ (Hürde des Erlernens und der Nutzung, vgl. Abschnitt 2.6.2) und „Ceiling“ (Mächtigkeit und Effektivität) eines Ansatzes. Sie stellen fest, dass die meisten bis dahin erfolgreichen Systeme in beiden Aspekten entweder gleichzeitig als hoch oder in beiden Aspekten gleichzeitig als gering einzuschätzen sind. Während in der Vergangenheit zu hohe Hürden häufig eine größere Marktdurchdringung verhindert haben, sehen die Autoren für die Zukunft weiter steigende Anforderungen und eine größere

Diversifizierung von Benutzungsschnittstellen vorher. Daher sind „high ceiling – low threshold“-Systeme nach wie vor ein wichtiges Ziel für die Zukunft (Myers et al., 2000).

Am Beispiel des EU-Projekts GUIDE lässt sich der beschriebene Zusammenhang von Hürde und Effektivität gut nachvollziehen. Der dort entwickelte Ansatz ist bezüglich seines Entwicklungsaufwands besonders interessant, da Adaptationen der Benutzungsschnittstelle nicht auf einem neu zu erstellenden abstrakten Modell, sondern auf speziellen Annotationen im HTML-Code einer bestehenden Web-Applikation basieren. Dadurch entfällt der Aufwand einer abstrakten Applikationsmodellierung komplett. Andererseits ist jedoch der Umfang der möglichen Anpassungen dadurch stark eingeschränkt. Anpassungen können nur „lokal“ auf der aktuell dargestellten Seite erfolgen (Coelho & Duarte, 2011).

Zusammenfassend kann man feststellen, dass heutige Ansätze für ABS den Entwicklungsaufwand zur Erstellung einer adaptiven Applikation meist nur am Rande betrachten. Wiederverwendbare generische Komponenten finden beispielsweise bei SUPPLE (gerätespezifische Widgets) und den PLASTIC USER INTERFACES („Comets“) Anwendung. Am Beispiel von GUIDE lässt sich erkennen, dass ohne ein gewisses Maß an Modellierung keine allzu umfangreichen Adaptationen möglich sind. Umfassende Anpassungen bei gleichzeitig geringem Entwicklungsaufwand sind heute noch mit keinem System realisierbar. Dennoch scheint es nötig und möglich, die Attraktivität von ABS durch eine gezielte Steigerung der Entwicklungseffizienz zu verbessern.

2.6.2 Einstiegshürde und Kompetenzanforderungen auf Seiten der Entwickler

Die Erstellung eines Modells, das eine Benutzungsschnittstelle unabhängig von ihrem konkreten Erscheinungsbild und den verwendeten Eingabemechanismen beschreibt, erfordert ein gewisses Abstraktionsvermögen auf Seiten des Entwicklers. Diese abstrakte und teilweise zeitaufwändige Arbeitsweise der modellbasierten Entwicklung von Benutzungsschnittstellen entspricht üblicherweise nicht der heutigen Praxis – auch wenn einige Systeme, wie z.B. TERESA, einen grafischen Editor für die Erstellung der Modelle bieten (Paterno et al., 2008, nach Gajos et al., 2010). Grafische, konkrete Repräsentationen der zu entwickelnden Benutzungsschnittstelle sind heute fester Bestandteil in allen Phasen des Gestaltungsprozesses (vgl. Newman & Landay, 2000; Lin & Landay, 2008). Meistens werden auch zur Entwicklung der Benutzungsschnittstelle grafische Editoren und GUI-Builder mit WYSIWYG¹⁰-Funktionalität eingesetzt. Schließlich wird das Erlernen einer neuen Sprache zur Spezifikation der Modelle häufig als weitere Hürde genannt (Myers et al., 2000).

GUMMY (Meskens et al., 2008) und DAMASK (Lin & Landay, 2008) sind interessante Ansätze, um die modellbasierte Entwicklung von Benutzungsschnittstellen mit der heutigen Praxis und den üblichen Kompetenzen und Vorlieben der Entwickler und Designer in Einklang zu bringen. Beide Werkzeuge ermöglichen den Entwicklern die Arbeit an einem konkreten Design für ein bestimmtes Endgerät, das dann Ausgangspunkt für eine spätere Abstraktion der Benutzungsschnittstelle darstellt. Für die Erstellung eines konkreten Gestaltungsentwurfs bietet GUMMY einen GUI-Builder mit plattformspezifischen Bedienelementen. GUMMY erzeugt aus der

¹⁰ WYSIWYG: What You See Is What You Get

manuell erstellten konkreten Benutzungsschnittstelle eine UIML-Beschreibung (UIML: User Interface Markup Language, s. Helms & Abrams, 2008), die dann als Grundlage für die Generierung eines initialen Gestaltungsentwurfs für eine Benutzungsschnittstelle derselben Applikation auf einer anderen technischen Plattform dient. Der Designer kann dann die generierte Benutzungsschnittstelle weiter verändern und verfeinern. Nachträgliche manuelle Veränderungen an der neuen konkreten Benutzungsschnittstelle werden automatisch im UIML-Modell übernommen (Meskens et al., 2008). DAMASK ist ein Prototyping-Werkzeug, mit dem Designer eine Benutzungsschnittstelle für eine spezifische Plattform skizzieren und dabei auf bereits bestehende Entwurfsmuster (Design Patterns) zurückgreifen. In dem *DAMASK Pattern Browser* kann der Gestalter geeignete Entwurfsmuster auswählen und in die Arbeitsfläche des DAMASK Editors ziehen. Die verwendeten Entwurfsmuster können bei Bedarf für die spezielle Applikation angepasst werden. Einerseits nehmen die Autoren an, dass die Arbeit mit Entwurfsmustern den Entwicklern und Gestaltern gut geläufig ist, da sie heute eine weit verbreitete Methode zur Kommunikation und Verwaltung bewährter Designlösungen darstellen. Andererseits dienen die Entwurfsmuster als wesentlicher Mechanismus, eine konkrete Benutzungsschnittstelle auf andere Plattformen zu übertragen, indem DAMASK für ein und dasselbe Gestaltungsproblem mehrere plattformspezifische Entwurfsmuster anbietet (Lin & Landay, 2008). Beide Werkzeuge sind nicht direkt geeignet, um ABS zur Überwindung von Nutzungsbarrieren zu entwickeln. Ihr Anwendungsbereich ist auf die Gestaltung von Benutzungsschnittstellen für verschiedene technische Plattformen beschränkt. Der von DAMASK gewählte Ansatz über Entwurfsmuster ist dennoch vielversprechend. Die Sammlung der Entwurfsmuster könnte wahrscheinlich je nach Bedarf erweitert werden. Jedoch erscheinen die in dem Artikel von Lin und Landay genannten Beispiel-Patterns nicht generisch, sondern eher applikationsspezifisch (z.B. „Clean Product Details“ für eine E-Commerce Plattform). Dadurch wäre eine Erweiterung des Pattern-Katalogs auf beliebige Anwendungen und Anpassungen an Nutzer, Umgebung und technische Plattform aufwändig und umfangreich. Schließlich erzeugt DAMASK Gestaltungsentwürfe und keine vollständigen Benutzungsschnittstellen. Dynamische Adaptationen werden in beiden Werkzeugen nicht adressiert (vgl. Peissner et al., 2012a).

Nichols und Myers (2009) zeigen mit ihrem PERSONAL UNIVERSAL CONTROLLER (PUC), dass auch nicht-grafische Beschreibungssprachen für Benutzungsschnittstellen einfach lernbar und leicht nutzbar sein können. Die von ihnen zusammengefassten Gestaltungsprinzipien ihrer XML-basierten Sprache sowie wesentliche Konzepte wie beispielsweise die *Smart Templates*, die bewährte Gestaltungslösungen für verschiedene Plattformen aufgreifen, können bei der Entwicklung eines geeigneten Spezifikationsformats für barrierefreie ABS herangezogen werden (Nichols & Myers, 2009).

Damit lässt sich zusammenfassend feststellen, dass die oben genannten Anforderungen und Probleme in neueren Forschungsarbeiten zu generierten Benutzungsschnittstellen aufgegriffen werden. Eine vollständige Lösung steht jedoch noch nicht zur Verfügung. Insbesondere der in DAMASK (Lin & Landay, 2008) skizzierte Ansatz der Arbeit an einem konkreten Design mit Entwurfsmustern in einem grafischen Editor erscheint aussichtsreich. Eine Erweiterung dieses Ansatzes auf die Generierung von ABS und auf das gesamte Feld der Barrierefreiheit ist ein denkbarer Weg für die Lösung der in dieser Arbeit adressierten Problemstellung.

2.6.3 Nachvollziehbarkeit und Beherrschbarkeit der Prozesse zur Erzeugung der resultierenden Benutzungsschnittstelle

Entwickler möchten die automatischen Prozesse zur Generierung einer ABS verstehen und beherrschen können. Myers und seine Kollegen jedoch weisen darauf hin, dass es in vielen Fällen schwierig sein kann, den Zusammenhang zwischen der Spezifikation einer Benutzungsschnittstelle und dem finalen generierten Ergebnis nachzuvollziehen und gezielt beeinflussen zu können (Myers et al., 2000).

Ein Grund für den Wunsch nach Einflussnahme auf die resultierenden Benutzungsschnittstellen liegt darin, dass automatisch generierte Benutzungsschnittstellen in der Vergangenheit häufig keine allzu hohen ästhetischen Ansprüche erfüllen konnten (vgl. Myers et al., 2000; Gajos et al., 2010). Zwar konnten Nichols und Kollegen für ihren Ansatz des PERSONAL UNIVERSAL CONTROLLERS (PUC) zeigen, dass durch eine automatische Generierung die Gebrauchstauglichkeit (Usability) von Benutzungsschnittstellen für multiple Geräte verbessert werden kann (Nichols et al., 2007). Ein Schwachpunkt sind aber dennoch häufig die fehlenden Mechanismen zur firmen- oder projektspezifischen Anpassung grafischer Eigenschaften der Benutzungsschnittstelle („Customizing“) (vgl. Nichols et al., 2007; Peissner et al., 2012a). Szekely skizziert bereits 1996 in seinem Überblicksartikel eine flexible Kooperation zwischen automatischer Generierung und manueller Bearbeitung der finalen Benutzungsschnittstelle. Dabei fordert er eine vollständige Beherrschbarkeit aller Eigenschaften der Benutzungsschnittstelle durch den Entwickler und eine Offenlegung aller verwendeten Modelle (Szekely, 1996). Derart umfangreiche Einflussmöglichkeiten auf die generierte Benutzungsschnittstelle werden dem Entwickler von kaum einem der Ansätze geboten.

Wie bereits in Abschnitt 2.5 unter dem Schlagwort der Offenheit beschrieben, sind die Mechanismen zur automatischen Erstellung der ABS bei SUPPLE ausführlich und nachvollziehbar veröffentlicht. Darüber hinaus sieht SUPPLE für den Endnutzer gewisse Möglichkeiten vor, die Benutzungsschnittstelle während der Laufzeit anzupassen. Allerdings beschränken sich diese Anpassungen lediglich auf die Veränderung der Auswahl und Kombination der vom System zur Verfügung gestellten Bedien- und Anzeigeelemente. Die Elemente an sich und deren vom System vorgeschlagenen Anordnung können jedoch nicht verändert werden – weder durch den Endnutzer noch durch den Entwickler (vgl. Gajos & Weld, 2004; Gajos et al., 2010).

Im Zusammenhang mit PLASTIC USER INTERFACES bieten der *Comets Inspector* (Demeure et al., 2006; Demeure et al., 2007) und das Werkzeug *PlastiXML* (Collignon et al., 2008) Möglichkeiten, die Mechanismen der Generierung und Anpassung der ABS zu untersuchen und zu verstehen sowie diese gezielt zu beeinflussen. Aus den Veröffentlichungen geht zwar hervor, dass zwischen alternativen Darstellungs- und Interaktionsvarianten gewählt werden kann und gewisse Darstellungseigenschaften wie z.B. Farbe frei eingestellt werden können. Jedoch bleibt unklar, in wie weit ein typisches Bearbeitungsszenario zur Anpassung der ABS an die grafische Identität einer Marke oder eines Unternehmens unterstützt wird.

An den Beispielen der beiden Ansätze UNIFIED USER INTERFACES (UII) und FAME wird die Wechselwirkung zwischen der Wiederverwertbarkeit einzelner Module und der Beherrschbarkeit der generierten Benutzungsschnittstelle deutlich. Beide Systeme erfordern eine manuelle Ausgestaltung und Spezifikation der verschiedenen Lösungsvarianten einer Benutzungsschnittstelle – bei FAME in Form einer *Behavioral Matrix* (Duarte & Carriço, 2006), bei UII als *Design Space* (Savidis & Stephanidis, 2004). Damit hat der Entwickler zwar alle

Einflussmöglichkeiten auf die generierten Benutzungsschnittstellen. Andererseits erfordert die Entwicklung jeder neuen Applikation einen hohen Aufwand (vgl. Abschnitt 2.6.1). Ideal wäre ein kombinierter Ansatz, der bei der Generierung von applikationsspezifischen Benutzungsschnittstellen auf einer Sammlung von Designlösungen aufbaut, die je nach Wunsch des Entwicklers noch angepasst werden können. Bei der Frage, ob die Anpassungen des Entwicklers dann jeweils die gesamte wiederverwendbare Gestaltungslösung verändern oder nur fallspezifisch angewendet werden, erscheinen Ansätze interessant, die - wie z.B. DAMASK - verschiedene Anwendungsebenen unterschieden. So gelten die Entwurfsmuster bei DAMASK je nach entsprechender Ebene entweder für die Benutzungsschnittstellen aller oder nur bestimmter Geräte (vgl. Lin & Landay, 2008).

2.7 Zusammenfassung: Anforderungen und Defizite bestehender Ansätze

Als Zusammenfassung des aktuellen Stands der Forschung und Technik lässt sich feststellen, dass vor dem Hintergrund der in den Abschnitten 2.1 bis 2.6 dargestellten Anforderungen an ABS zur Überwindung von Nutzungsbarrieren ein signifikanter Handlungsbedarf besteht. Dieses Fazit wird in diesem Abschnitt stellvertretend anhand ausgewählter Lösungsansätze begründet, die bereits in den vorausgehenden Abschnitten hinsichtlich der spezifischen Aspekte von ABS zur Überwindung von Nutzungsbarrieren untersucht worden sind. Nach einer knappen Charakterisierung der einzelnen ausgewählten Ansätze findet sich in Tabelle 1 eine vergleichende Gegenüberstellung der Ansätze bezüglich der geforderten Eigenschaften:

- DAMASK
DAMASK ist ein Werkzeug für die Gestaltung von Benutzungsschnittstellen, die auf mehreren Geräten verwendet werden sollen. Während der Designer eine Benutzungsschnittstelle für ein Gerät skizziert, erstellt DAMASK ein abstraktes Modell, das dann zur Generierung weiterer Benutzungsschnittstellen für andere Geräte genutzt wird. Ermöglicht werden diese Mechanismen durch die Nutzung einer Sammlung von Entwurfsmustern (Design Patterns), die für die verschiedenen unterstützten Geräte Gestaltungslösungen für eine Menge von Designproblemen vorhalten. Diese Entwurfsmuster werden vom Gestalter während des Entwurfs der Benutzungsschnittstelle referenziert. Zwar werden barrierefreie Gestaltungslösungen von DAMASK nicht explizit adressiert und ebenso fehlen in diesem Design-Werkzeug die Mechanismen zur Generierung einer Benutzungsschnittstelle. Dennoch können insbesondere der Pattern-zentrierte Ansatz und die grafische, praxisnahe Arbeitsweise des Gestalters gut auf die Problemstellung dieser Arbeit übertragen werden.
Quelle: Lin & Landay, 2008
- FAME
FAME („Model-based Framework for Adaptive Multimodal Environments“) ist ein modellbasiertes Rahmenmodell für adaptive multimodale Benutzungsschnittstellen. Es umfasst eine Architektur sowie Empfehlungen für die Gestaltung eines Entwicklungsprozesses für adaptive Applikationen. Im Zentrum des Ansatzes steht eine Verhaltensmatrix, mit der der Entwickler das Verhalten einzelner Bedien- und Anzeigeelemente unter verschiedenen Bedingungen spezifiziert. Als Referenzapplikation wird die Implementierung eines digitalen Hörbuchs (Digital Talking Book: DTB) beschrieben, das sich an das genutzte Gerät, Umgebungsbedingungen und das

Nutzerverhalten anpassen kann.

Quelle: Duarte & Carrico, 2006

– GUIDE

Das europäische Forschungsprojekt GUIDE¹¹ („Gentle user interfaces for elderly people“) läuft parallel zu den in dieser Arbeit beschriebenen Entwicklungen. Durch multimodale und adaptive Benutzungsschnittstellen soll GUIDE zur Steigerung der Barrierefreiheit von interaktiven TV-Applikationen beitragen. Die Anpassungen der Benutzungsschnittstellen konzentrieren sich im Wesentlichen auf die „Erweiterung“ (Augmentation) der ursprünglichen Benutzungsschnittstelle durch das Angebot zusätzlicher Ein- und Ausgabemodalitäten. Mit dem Ziel einer möglichst geringen Einstiegshürde für Anbieter und Entwickler von adaptiven Systemen verzichtet GUIDE auf die Erstellung eines abstrakten Modells der Benutzungsschnittstelle. Bestehende Web-Applikationen werden durch Annotationen im HTML-Code nachträglich adaptiv gemacht.

Quellen: Biswas et al., 2011; Coelho & Duarte, 2011; Coelho et al., 2011

– MICA

MICA steht für „Mixed-Initiative Customization Assistance“ und ist ein beispielhaftes System für eine intelligente Unterstützung des Nutzers bei der Personalisierung komplexer Benutzungsschnittstellen. Dabei wird ein kooperativer Dialogstil (gemischte Initiative) gewählt, der durch adaptive Unterstützungsfunktionen des Systems gekennzeichnet ist und gleichzeitig dem Nutzer die volle Kontrolle über den Konfigurationsprozess überlässt. Grundlage der automatisch generierten Konfigurationsvorschläge ist eine GOMS-Analyse (Card et al., 1983), die während der Nutzung die voraussichtlich benötigte Bearbeitungszeit für verschiedene Konfigurationsalternativen berechnet. Die für diese Arbeit relevanten Aspekte des MICA-Systems umfassen vor allem die bislang einzigartigen Ansätze der gemischten Initiative sowie die Mechanismen zur Schaffung von Transparenz und Kontrollierbarkeit des intelligenten Systemverhaltens aus Sicht des Nutzers.

Quellen: Bunt et al., 2009; Bunt et al., 2007

– OASIS und ASK-IT

OASIS¹² („Open architecture for Accessible Services Integration and Standardisation“) kann als Folgeprojekt des europäischen Projekts ASK-IT¹³ („Ambient Intelligence System of Agents for Knowledge-based and Integrated Services for Mobility Impaired Users“) betrachtet werden. In ASK-IT wurden personalisierte und kontextsensitive Services entwickelt, die die Mobilität von mobilitätseingeschränkten Personen unterstützen sollen. Ein wesentlicher Beitrag dieses Projekts ist die Entwicklung eines konzeptionellen Rahmens für umfangreiche Anpassungen einer Benutzungsschnittstelle an verschiedene Nutzereinschränkungen in einem modularen, regelbasierten System. Darauf baut OASIS auf. Der wesentliche Kern der Anpassungen in OASIS liegt in der „Adaptive Widget Library“, einer Bibliothek anpassbarer Bedien- und Anzeigeelemente. Die Entwicklung von OASIS-Applikationen wird durch die Integration der Adaptive Widget Library in den NetBeans GUI Builder unterstützt. Aufgrund der konzeptionellen Nähe zum Ansatz der

¹¹ <http://www.guide-project.eu/>

¹² <http://www.oasis-project.eu>

¹³ <http://www.ask-it.org>

Unified User Interfaces (UII) sowie der zahlreichen Referenzen und der direkten Beteiligung wesentlicher Urheber der UII vom ICS FORTH aus Kreta, könnte man diesen Ansatz auch als Weiterentwicklung der UII interpretieren.

Quellen: Leonidis et al., 2012; Leuteritz et al., 2009; Ringbauer et al., 2007

– Personal Universal Controller (PUC) und HUDDLE

PUC ist eine Beschreibungssprache für Benutzungsschnittstellen (UIDL: User Interface Description Language), die als Grundlage für die Generierung von grafischen und sprachbasierten Benutzungsschnittstellen für unterschiedliche Endgeräte dient. HUDDLE baut auf PUC auf und generiert automatisch Benutzungsschnittstellen zur Steuerung von dynamisch zusammengestellten und verbundenen audio-visuellen Geräten in einem Heimkino-Szenario. Der Ansatz wurde an der Carnegie Mellon University in Pittsburgh, USA entwickelt. PUC und HUDDLE sind für diese Arbeit relevant, da sie eines der ganz wenigen praktisch validierten Systeme zur automatischen Generierung von Benutzungsschnittstellen darstellen. Die in PUC gewählte Herangehensweise einer abstrakten UIDL sowie die Idee von intelligenten Bedienelementen (Smart Templates), die sich je nach Zielgerät unterschiedlich darstellen und verhalten, können zur Lösung des in dieser Arbeit beschriebenen Problems herangezogen werden.

Quellen: Nichols & Myers, 2009; Nichols et al., 2007; Nichols et al., 2006a; Nichols et al., 2006b

– PLASTIC USER INTERFACES (PUI)

Seit mehr als zehn Jahren veröffentlichen insbesondere Wissenschaftler aus Grenoble und Louvain zahlreiche Arbeiten unter dem Schlagwort der „User Interface Plasticity“. Während der Ausgangspunkt der Arbeiten in den Bereichen des Ubiquitous Computing und der Nutzung mehrerer verschiedener technischer Endgeräte liegt, sind die zugrundeliegenden Konzepte und Technologien ausreichend generisch, um auf jede Art von Änderungen des Nutzungskontexts angewendet zu werden. Neben dem starken und umfassenden konzeptionellen Rahmen wie beispielsweise dem Referenzmodell „CAMELEON-RT“ sind insbesondere die sogenannten „Comets“, die eine Laufzeitanpassung einzelner Bedienelemente ermöglichen, sowie die Idee eines Meta-UIs zur Kontrollierbarkeit des Systemverhaltens durch den Nutzer interessant.

Quellen: Frey et al., 2012; Calvary et al., 2011; Ceret, 2011; Dessart et al., 2011; Coutaz, 2010; Collignon et al., 2008; Demeure et al., 2008; Vanderdonckt et al., 2008; Coutaz et al., 2007; Demeure et al., 2007; Coutaz, 2006; Demeure et al., 2006; Balme et al., 2004; Calvary et al., 2004; Calvary et al., 2003

– SUPPLE

An der University of Washington wurde mit SUPPLE eines der einflussreichsten adaptiven Systeme zur Überwindung von Nutzungsbarrieren entwickelt. Die automatisch generierten Benutzungsschnittstellen minimieren den Interaktionsaufwand unter Berücksichtigung der genutzten Geräte, der bearbeiteten Aufgaben sowie angenommener Nutzerpräferenzen und motorischer sowie perzeptueller Einschränkungen. Grundlage hierfür sind komplexe Kostenfunktionen, die alle für die Anpassung relevanten Aspekte in eine Modellierung der Interaktion einbeziehen. Interessant sind darüber hinaus die Mechanismen, die SUPPLE den Endnutzern bietet, um die generierten Benutzungsschnittstellen nach ihren Vorstellungen selbst anzupassen.

Quellen: Wobbrock et al., 2011; Gajos et al., 2010; Gajos et al., 2008; Gajos & Weld, 2004; Weld et al., 2003

- Unified User Interface (UUI) und AVANTI
Der AVANTI Webbrowser ist die bekannteste Implementierung des UNIFIED USER INTERFACE (UUI) – Ansatzes, der hauptsächlich von Mitarbeitern des ICS Forth in Kreta entwickelt und veröffentlicht worden ist. Die UUI-Software-Architektur ermöglicht Anpassungen der ABS an ein Nutzer- und Kontextprofil bei der Initialisierung und während der Interaktion. Grundlage der Anpassungen stellen alternative Entwurfsmuster (Design Patterns) dar, die alternative Lösungen für unterschiedliche Ausprägungen des Nutzer- und Kontextprofils beschreiben. Die Unified User Interface Design Methode beschreibt die Vorgehensweise zur Entwicklung einer adaptiven Applikation. Aufgrund seines umfangreichen und soliden konzeptionellen Rahmens gilt UUI noch heute als einer der einflussreichsten Ansätze für ABS zur Überwindung von Nutzungsbarrieren.
Quellen: Savidis et al., 2005; Savidis & Stephanidis, 2004; Kobsa et al., 2001; Savidis & Stephanidis, 2001; Stephanidis et al., 1998a; Stephanidis et al., 1998b

Aus Tabelle 1 wird deutlich, dass trotz der langen Forschungstradition kein Ansatz besteht, der den dargestellten Anforderungen vollständig in der gewünschten Form entspricht. Die festgestellten Defizite begründen den in dieser Arbeit adressierten Handlungsbedarf. Es wird ein Ansatz benötigt, der sowohl die notwendigen funktionalen Eigenschaften als auch grundlegende Voraussetzungen für die Akzeptanz bei Nutzern und Entwicklern mitbringt, um die Potenziale von ABS für maximal zugängliche interaktive Systeme zu nutzen und erfolgreich in den Markt bringen zu können. Durch die Adressierung der aufgestellten Anforderungen soll der zu entwickelnde Ansatz einen Beitrag zur Weiterentwicklung von ABS zur Überwindung von Nutzungsbarrieren leisten. Damit konkretisiert sich die Problemstellung der vorliegenden Arbeit. Die dargelegten Anforderungen bilden einerseits die richtungsweisende Grundlage für die Entwicklungen im Rahmen dieser Arbeit. Andererseits dienen sie als Wertesystem für die in Kapitel 9 dargestellte Validierung und Erfolgsbewertung des Lösungsansatzes.

| Anforderungen an Lösungsansatz | DAMASK | FAME | GUIDE | MICA | OASIS / ASK-IT | PUC / Huddle | Plastic UI | SUPPLE | UUI/ AVANTI |
|--|--------|------|-------|------|----------------|--------------|------------|--------|-------------|
| Adaptivität: automatische systeminitiierte Anpassungen (2.1) | | ● | ◐ | ● | ◐ | ● | ● | ● | ● |
| Adaptivität und Adaptierbarkeit in einem System (2.1) | | ◐ | ○ | ◐ | ○ | ○ | ◐ | ◐ | ○ |
| Kontinuierliche Anpassung an ein selbstlernendes Profil während der Nutzung (2.2) | | ◐ | ○ | ◐ | ◐ | ○ | ◐ | ◐ | ◐ |
| Transparenz: erkennbare, verstehbare und nachvollziehbare Adaptationen (2.3.1) | | ○ | ◐ | ◐ | ○ | | ◐ | ◐ | ○ |
| Kontrollierbarkeit automatischer Adaptationen (2.3.2) | | ○ | ◐ | ● | ○ | | ◐ | ◐ | ○ |
| Anpassungen an Nutzer, Umgebung und technische Ausstattung (0) | ◐ | ◐ | ● | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ |
| Anpassungen der Informationsdarstellung, Interaktion und Navigation (0) | ◐ | ◐ | ◐ | ◐ | ◐ | ● | ● | ● | ● |
| Modulare und erweiterbare Adaptationsmechanismen (2.5) | | ◐ | ● | ○ | ◐ | ◐ | ● | ● | ◐ |
| Adaptationsmechanismen öffentlich zugänglich und offen für externe Beiträge (2.5) | | | ◐ | ○ | ◐ | ◐ | ◐ | ◐ | ◐ |
| Effiziente Entwicklung: wiederverwendbare Komponenten und geringer Spezifikationsaufwand bei ausreichender Mächtigkeit (2.6.1) | ◐ | ○ | ◐ | | ◐ | ◐ | ● | ● | ○ |
| Geringe Einstiegshürde und Kompetenzanforderungen auf Seiten der Entwickler (2.6.2) | ● | ◐ | ● | | ● | ● | ◐ | ○ | ○ |
| Generierung der ABS für Entwickler nachvollziehbar und beherrschbar (2.6.3) | | | | | ● | ◐ | ◐ | ○ | ◐ |

Legende:

● Breite Abdeckung, vertiefte Konzepte

◐ Teilweise Abdeckung, vertiefte Konzepte

◐ Breite Abdeckung, einfache Konzepte

○ Geringe Abdeckung, einfache Konzepte

Tabelle 1 Zusammenfassende Bewertung aktueller Ansätze anhand der hergeleiteten Anforderungen

3 Überblick des Ansatzes

Der im Folgenden beschriebene Ansatz für ABS zur Überwindung von Nutzungsbarrieren wurde vom Autor dieser Arbeit im Rahmen des europäischen Forschungsprojekts MyUI¹⁴ („Mainstreaming Accessibility through Synergistic User Modelling and Adaptability“) entwickelt. Das MyUI-Gesamtsystem bietet eine vollständige Infrastruktur mit allen benötigten Komponenten einer Entwicklungs- und Laufzeitumgebung für ABS.

Die Darstellung des Lösungsansatzes in den Kapiteln 3 bis 8 bezieht sich ausschließlich auf den originären Beitrag des Autors dieser Arbeit zum MyUI-Gesamtsystem. Dieser Beitrag umfasst den konzeptionellen Rahmen des Gesamtsystems (Kapitel 3) und die effiziente Komponente mit allen Artefakten und Mechanismen zur Generierung und Adaptation der ABS während der Laufzeit (Kapitel 4 bis 8). Die weiteren Komponenten des MyUI-Gesamtsystems sind die Infrastruktur zur Nutzer- und Kontextmodellierung und die ABS-Entwicklungsumgebung. Diese wurden auf Grundlage des konzeptionellen Rahmens dieser Arbeit von anderen Projektpartnern implementiert und werden in dieser Arbeit nicht näher behandelt. Abbildung 2 zeigt das MyUI-Gesamtsystem. Der in dieser Arbeit beschriebene Ansatz bezieht sich auf die umrahmten Komponenten.

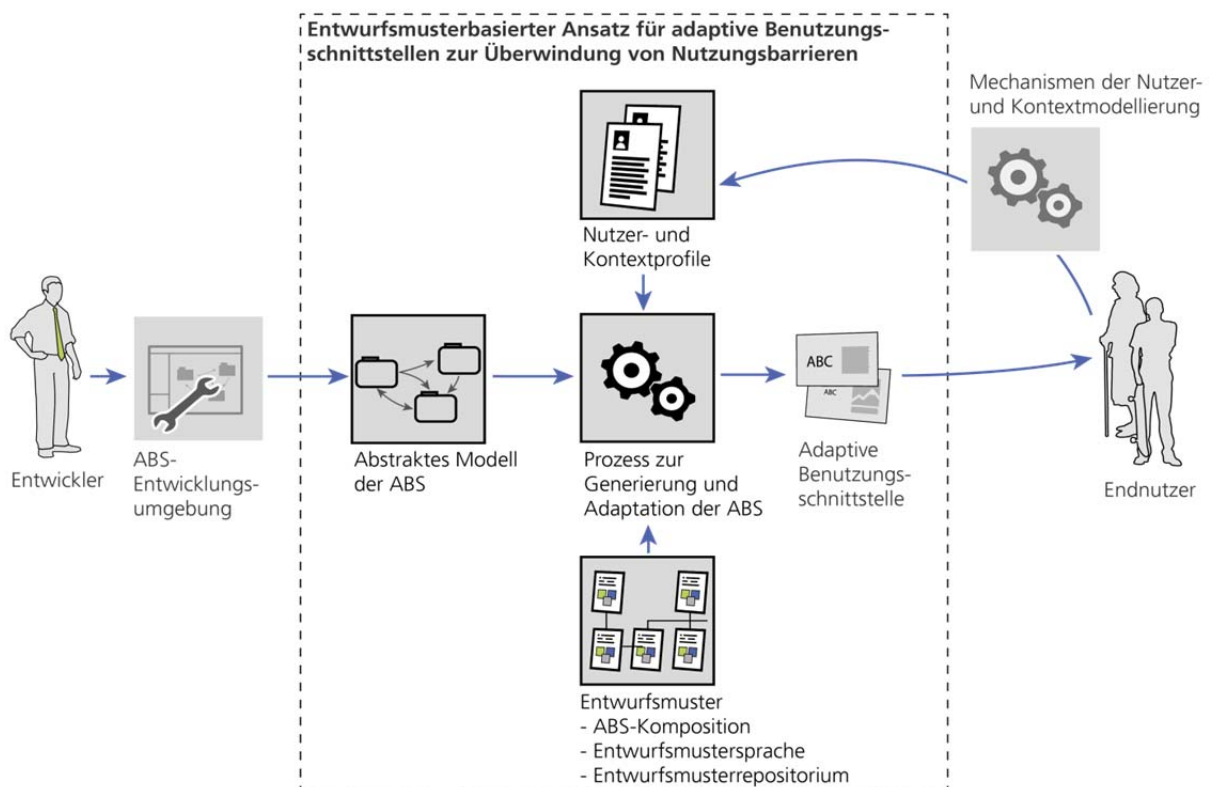


Abbildung 2 Beitrag der vorliegenden Arbeit zum MyUI-Gesamtsystem

¹⁴ <http://www.myui.eu/>

Dieses Kapitel bietet einen Überblick der grundlegenden Prinzipien und Kernkonzepte des Lösungsansatzes für ABS zur Überwindung von Nutzungsbarrieren. Die meisten hier skizzierten Konzepte werden in späteren Kapiteln dieser Arbeit noch genauer beschrieben. Wesentliche Teile der folgenden Darstellungen in Kapitel 3 bis 8 sind aus öffentlichen Projektdokumenten (Peissner et al., 2011b) und Publikationen (Peissner et al., 2012a; Peissner et al., 2012b; Peissner et al., 2011c), die der Autor dieser Arbeit selbst verfasst hat, übernommen, angepasst und ins Deutsche übersetzt.

3.1 Selbstlernende und dynamische Adaptationen nach dem Vorbild der zwischenmenschlichen Kommunikation

Als Vorbild für die Adaptationen der Benutzungsschnittstellen dient die natürliche und allmähliche gegenseitige Anpassung zweier menschlicher Kommunikationspartner. Zu Beginn sind beide Partner in der Regel eher vorsichtig und verhalten sich neutral. Mit der Zeit lernen sich beide Partner immer besser kennen. Sie entwickeln ein gegenseitiges Verständnis, indem sie die Reaktionen des Partners auf ihr eigenes Kommunikationsverhalten wahrnehmen, wie z.B. explizite Äußerungen, emotionale Gesichtsausdrücke, Mimik und Gesten. Das zunehmende gegenseitige Verständnis und Kennen des Partners hilft den beiden, ihr eigenes Verhalten an die individuellen Besonderheiten des Gegenübers anzupassen. Ohne eine ausdrückliche Befragung zu persönlichen Vorlieben und Einstellungen passen sich die Kommunikationspartner während der Interaktion schnell und natürlich aneinander an. Dieser sanfte und natürliche Anpassungsprozess soll auf die Individualisierung von ABS übertragen werden.

Vor der Nutzung einer adaptiven Applikation auf Basis des MyUI-Ansatzes ist eine Erstkonfiguration des Nutzerprofils nicht notwendig. Bei der ersten Interaktion zwischen einer MyUI-Applikation und einem neuen Nutzer startet das System mit einem Standard-Nutzerprofil. Während der Nutzung sammelt und speichert das MyUI-System relevante Informationen über den Nutzer und seine Umgebung. Dieses neu erworbene Wissen wird nun dazu genutzt, die Benutzungsschnittstelle sofort und während der Interaktion so anzupassen, dass die angenommenen individuellen und situativen Anforderungen optimal unterstützt werden. In einem Regelkreis wird in einem iterativen Prozess das gespeicherte Wissen über Nutzer und Umgebung nach und nach verfeinert und aktualisiert. Im Gleichschritt werden entsprechende Anpassungen an der Benutzungsschnittstelle durchgeführt. Abbildung 3 illustriert den beschriebenen Regelkreis.

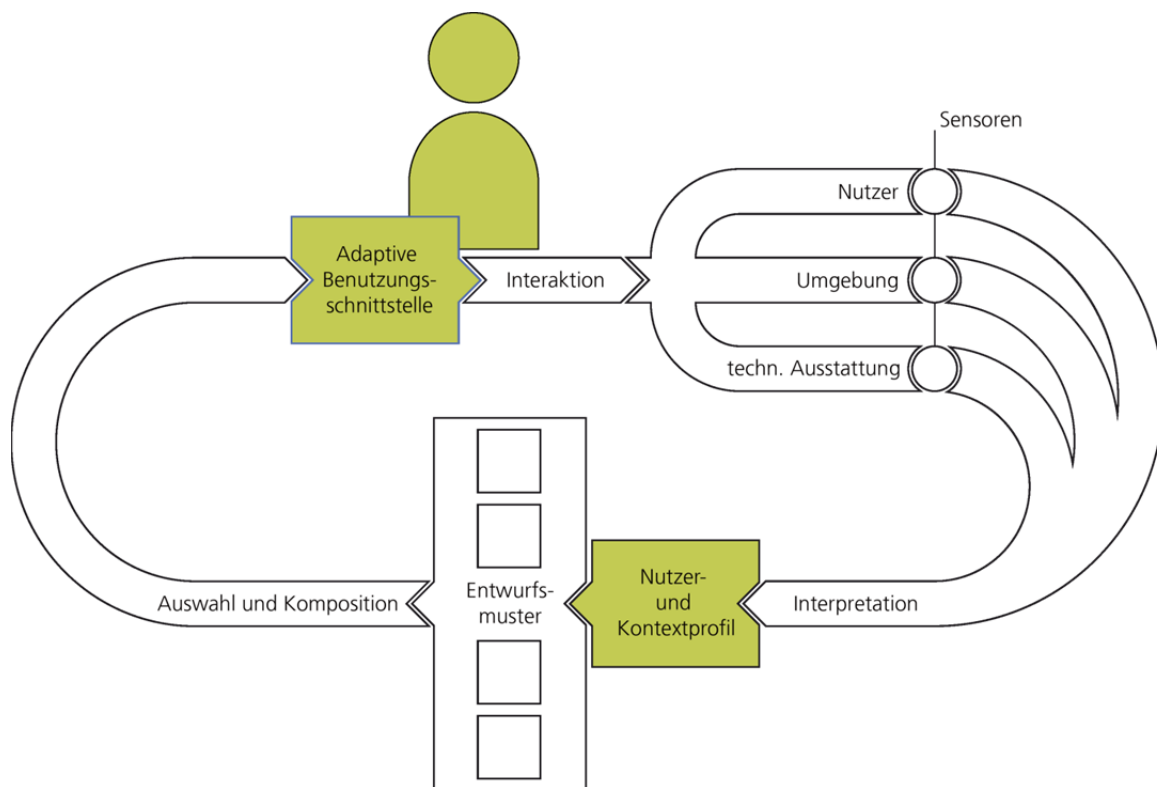


Abbildung 3 Selbstlernende und selbstadaptive Benutzungsschnittstellen in MyUI

Dieses Modell der Adaptation entspricht den von Paramythis, Totter und Stephanidis (2001) beschriebenen Grundzügen der Adaptation in ABS, die auf Totterdells Zwei-Ebenen-Architektur der Adaptation beruhen (vgl. Totterdell & Rautenbach, 1990).

Die meisten für das MyUI-Nutzerprofil relevanten Informationen werden direkt aus der Erkennung bestimmter Verhaltensweisen des Nutzers bei der Interaktion mit der Benutzungsschnittstelle gezogen. Beispiele für Ereignisse mit derartigem Informationsgehalt sind u.a. Time-Out (der Nutzer reagiert nicht auf eine Eingabeaufforderung des Systems innerhalb eines bestimmten Zeitfensters), wiederholtes Rückgängigmachen (der Nutzer scheint mehrmals aus Versehen die falsche Option gewählt zu haben) und Umwege. MyUI ermöglicht auch den zusätzlichen Einsatz von Hardware-Sensoren wie zum Beispiel Blickbewegungsregistrierung, um das Aufmerksamkeitsniveau des Nutzers festzustellen. Aufgrund der damit für den Nutzer verbundenen Hürden zusätzlicher zu vernetzender Geräte werden jedoch Software-Sensoren bevorzugt, die bestimmte Ereignisse aus der Interaktion zwischen Nutzer und ABS aufnehmen. Die Interpretation dieser Ereignisse im Sinne entsprechender Veränderungen des Nutzerprofils wird auf Ebene der Benutzungsschnittstelle ausgeführt und in Abschnitt 7.2 dargestellt. Darüber hinaus gehende Mechanismen der Nutzer- und Kontextmodellierung sind nicht Gegenstand dieser Arbeit und werden in Hernández et al., 2011 und Strnad et al., 2012a beschrieben.

Wie bereits in Abbildung 3 angedeutet, beruht die Generierung und Adaptation der Benutzungsschnittstellen auf einer Auswahl und Komposition von Entwurfsmustern, die den aktuell bekannten Anforderungen des Nutzers und der weiteren Kontextbedingungen am besten gerecht werden. Der Prozess zur Generierung und Adaptation der Benutzungsschnittstellen wird in den Kapiteln 5, 6 und 7 detailliert erläutert.

3.2 Adaptierbarkeit sowie Transparenz und Kontrollierbarkeit des Anpassungsverhaltens

Neben den automatischen Methoden der Nutzer- und Kontextmodellierung während der Interaktion und den entsprechenden automatischen Adaptationen bietet MyUI auch umfangreiche Konfigurationsmöglichkeiten. Die Nutzer können zu jedem Zeitpunkt die aktuellen Systemeinstellungen und die im Nutzerprofil gespeicherten Daten einsehen und verändern. Damit können sie die Benutzungsschnittstelle über verschiedene Mechanismen nach ihren Anforderungen und Präferenzen konfigurieren. Machen sie Angaben über ihre eigenen Fähigkeiten bzw. Einschränkungen im Nutzerprofil, werden die aktualisierten Daten an die MyUI-Infrastruktur zur Nutzer- und Kontextmodellierung zurückgemeldet und es erfolgt direkt eine entsprechende Anpassung der Benutzungsschnittstelle an das neue Nutzerprofil. Ändern die Nutzer direkt die Einstellungen der Benutzungsschnittstelle (z.B. Auswahl einer größeren Schrift), wird sowohl die Benutzungsschnittstelle als auch das Nutzerprofil entsprechend angepasst.

Für die automatischen systeminitiierten Adaptationen sind umfangreiche Mechanismen zur Sicherung der Transparenz und der Kontrollierbarkeit durch die Nutzer vorgesehen. Zweifellos ist es für die Gebrauchstauglichkeit und die Akzeptanz einer ABS unerlässlich, dass die Nutzer die automatischen Anpassungen erkennen und verstehen können, und das adaptive Systemverhalten kontrollieren können (vgl. Abschnitt 2.3). Die vorliegende Arbeit nimmt sich dieses wichtigen Themenfeldes in besonderer Weise an. Gezielte Mechanismen zur Schaffung von Transparenz und Kontrollierbarkeit der automatischen Adaptationen sind ein fester Bestandteil des MyUI-Systems.

In Anlehnung an die Terminologie der PLASTIC USER INTERFACES (vgl. Coutaz, 2006) könnte man die Funktionalitäten zur Steuerung des Anpassungsverhaltens als „Meta-Benutzungsschnittstelle“ zusammenfassen. In dieser Arbeit werden sie im Abschnitt 7.5 als *Adaptationsdialogmuster* beschrieben. Darüber hinaus tragen auch die in Abschnitt 7.4 behandelten *Adaptationsausführungsmuster* zu einer erhöhten Transparenz der Anpassungen bei.

3.3 Entwurfsmuster als modulare Bausteine der adaptiven Benutzungsschnittstelle

Für die Generierung und Adaptation von Benutzungsschnittstellen wird ein modularer Ansatz verfolgt. Dabei basiert die Generierung der ABS auf der Komposition von multimodalen Entwurfsmustern für Benutzungsschnittstellen. Jedes Muster bezieht sich auf spezifische Nutzereigenschaften und Kontextbedingungen, für die das Muster eine geeignete Gestaltungslösung bietet. So wird im Einzelfall die Zugänglichkeit eines Systems gewährleistet, indem verschiedene Entwurfsmuster kombiniert werden, die für den spezifischen Nutzer und Nutzungskontext geeignet sind. In Kapitel 4 wird das Konzept der Entwurfsmuster für ABS detailliert beschrieben.

Nach Borchers' funktionaler Klassifizierung von Entwurfsmustern für Benutzungsschnittstellen können Entwurfsmuster entweder hauptsächlich die Ausgabe (*perception*), die Eingabe (*manipulation*) oder die *Navigation* durch das System behandeln (vgl. Borchers, 2000). Die MyUI-Entwurfsmuster decken alle diese Bereiche ab und ermöglichen Anpassungen der Informationspräsentation, der Navigation und der Eingabemechanismen. Jedoch wird im Repositorium der MyUI-Entwurfsmuster eine andere Taxonomie angewendet. MyUI-Muster werden nach ihrer

Funktion im Adaptationsprozess kategorisiert. Mit dem Ziel einer hohen Konsistenz über die gesamte Benutzungsschnittstelle und um Konflikte im Adaptationsprozess zu vermeiden, wirken verschiedene Mustertypen auf verschiedenen Ebenen und auf verschiedene Aspekte der Benutzungsschnittstelle. Eine detaillierte Beschreibung der verschiedenen Mustertypen und ihrer jeweiligen Rollen findet sich in Abschnitt 4.2.

Neben der Modularität, die durch die Struktur der MyUI-Entwurfsmustersprache unterstützt wird, bietet der musterbasierte Ansatz auch beste Voraussetzungen für die Erweiterbarkeit. Durch das Hinzufügen neuer Entwurfsmuster kann ein Zwischenstand des Musterrepositoriums leicht ergänzt werden, um beispielsweise neue Geräte und Interaktionstechniken oder zusätzliche Einschränkungen und Umgebungsbedingungen abzudecken. Darüber hinaus wird das MyUI-Repositorium als öffentlich zugängliches MediaWiki¹⁵ gepflegt¹⁶. Damit sind alle Entwurfsmuster öffentlich einsehbar und können von anderen Fachexperten verfeinert und erweitert werden.

3.4 Umfassende Anpassungen in einem dreistufigen Prozess zur Generierung und Adaptation der Benutzungsschnittstelle

Der Prozess zur Generierung und Adaptation von ABS in MyUI gliedert sich in drei Phasen (s. Abbildung 4). Diese werden in den Kapiteln 5, 6 und 7 genauer behandelt:

1. *Parametrierung der ABS*

Das Ergebnis des ersten Schrittes ist das *Benutzungsschnittstellenprofil* (UIP: User Interface Profile). Das UIP besteht aus einer Menge von globalen Variablen, die bestimmte Eigenschaften der gesamten Benutzungsschnittstelle definieren. Beispiele hierfür sind Schriftgrößen oder der Darstellungsmodus (*displayMode*) mit möglichen Werten wie „text only“, „mainly text“ oder „graphics only“, der das Verhältnis von Text zu grafischen Elementen auf dem Bildschirm bestimmt. Die Einstellungen des UIP gelten über die gesamte Benutzungsschnittstelle und alle Interaktionssituationen der Applikation hinweg. Die Variablen des UIP werden so gesetzt, dass sie zu einer optimalen Anpassung an den jeweiligen Nutzer, seine Umgebung und die gerade verwendeten Ein- und Ausgabegeräte führen. Darüber hinaus werden im Rahmen der ABS-Parametrierung Einstellungen des Entwicklers bezüglich des gewünschten Erscheinungsbildes der ABS im Sinne einer Marke oder einer Firma (Customization) berücksichtigt.

Damit werden während der ABS-Parametrierung Informationen aus drei verschiedenen Quellen verarbeitet:

- Informationen über den Nutzer und die aktuelle Nutzungsumgebung aus dem *Nutzer- und Umgebungsprofil* (UP: User Profile),
- Informationen über die technische Ausstattung mit den verfügbaren Ein- und Ausgabegeräten aus dem *Geräteprofil* (DP: Device Profile) und

¹⁵ <http://www.mediawiki.org/wiki/MediaWiki/de>

¹⁶ <http://myui-patterns.iao.fraunhofer.de>

- Einstellungen zur Anpassung des Erscheinungsbildes an eine Marke oder eine Firma, wie sie vom Entwickler im *Customization-Profil*¹⁷ (CP: Customization Profile) definiert werden.

Die ABS-Parametrierung kann als Transformation dieser drei Profile in das UIP betrachtet werden. Das UIP wird zu Beginn einer neuen Interaktionssequenz¹⁸ mit einer MyUI-Anwendung initialisiert. Eine wiederholte Parametrierung (d.h. eine Aktualisierung des UIP) wird ausgelöst, wenn sich die verfügbaren Informationen über den Nutzer, die Umgebung oder die verfügbaren Geräte signifikant verändern.

2. Vorbereitung der ABS

Bei der ABS-Vorbereitung werden die für die aktuelle Situation am besten geeigneten Bedien- und Anzeigekomponenten ausgewählt. Als Ergebnis dieses Schrittes liegt eine Auswahl von Bedien- und Anzeigekomponenten vor, aus denen dann (im dritten Schritt) die Benutzungsschnittstelle erzeugt wird. Die Eingangsgrößen dieses Prozessschrittes umfassen

- eine abstrakte Beschreibung der Interaktion mit der ABS (das AAIM: Abstract Application Interaction Model, s. Abschnitt 3.6), die die Interaktionsmöglichkeiten eines Nutzers in einer Applikation modelliert und dabei jedem Zustand der Applikation eine Menge von Interaktionssituationen zuordnet, und
- das UIP, das die individuellen Anforderungen des Nutzers, seiner Umgebung und technischen Ausstattung in Begriffen spezieller Merkmalsausprägungen der Benutzungsschnittstelle widerspiegelt.

Die ABS-Vorbereitung wird jedes Mal ausgelöst, wenn auf Seiten der Applikation ein Zustandswechsel erfolgt (im AAIM) oder wenn signifikante Änderungen im UIP festgestellt werden.

3. Generierung und Adaptation der ABS

Nun werden die ausgewählten Bedien- und Anzeigekomponenten in eine vollständige Benutzungsschnittstelle überführt. Ausgangspunkt dieses Schrittes ist die in der ABS-Vorbereitung getroffene Auswahl. Das Ergebnis ist eine vollständige Benutzungsschnittstelle, die zu jedem Zeitpunkt das aktuell verfügbare Wissen über Nutzer, Umgebung und Ausstattung reflektiert. Während der Interaktion werden spezielle Ereignisse festgestellt, die auf Barrieren, Bedienprobleme oder neue Anforderungen hinweisen können. Diese Ereignisse werden der MyUI-Kontextmanagementinfrastruktur zurückgemeldet, um das UP und das DP zu aktualisieren und einen erneuten Durchlauf des dreistufigen Prozesses zu starten. Die resultierenden Laufzeitanpassungen können als

¹⁷ Für den englischsprachigen Begriff „Customization“ gibt es keine angemessene deutsche Übersetzung. Der Begriff der Anpassung wäre im Rahmen dieser Arbeit zu allgemein. Das eingedeutschte Wort „Kustomisierung“ wird zwar gelegentlich genutzt, ist jedoch nicht so gebräuchlich, dass es im Dudenkorpus enthalten wäre (vgl. <http://www.duden.de/>; Zugriff am 7. Juni 2013). Aus diesem Grund wird in dieser Arbeit das gebräuchliche englische Fremdwort „Customization“ verwendet.

¹⁸ Eine Interaktionssequenz (oder Nutzungssequenz) mit einer MyUI-Applikation beginnt mit dem Start der Applikation und endet mit dem Schließen der Applikation. Zu Beginn einer Interaktionssequenz wird die ABS initial generiert. Während der Interaktionssequenz kann es zu mehreren Anpassungen der ABS kommen.

eine wiederholte Generierung der Benutzungsschnittstelle betrachtet werden – jedoch mit speziellen Mechanismen, um von einer Instanz der ABS zur nächsten zu wechseln.

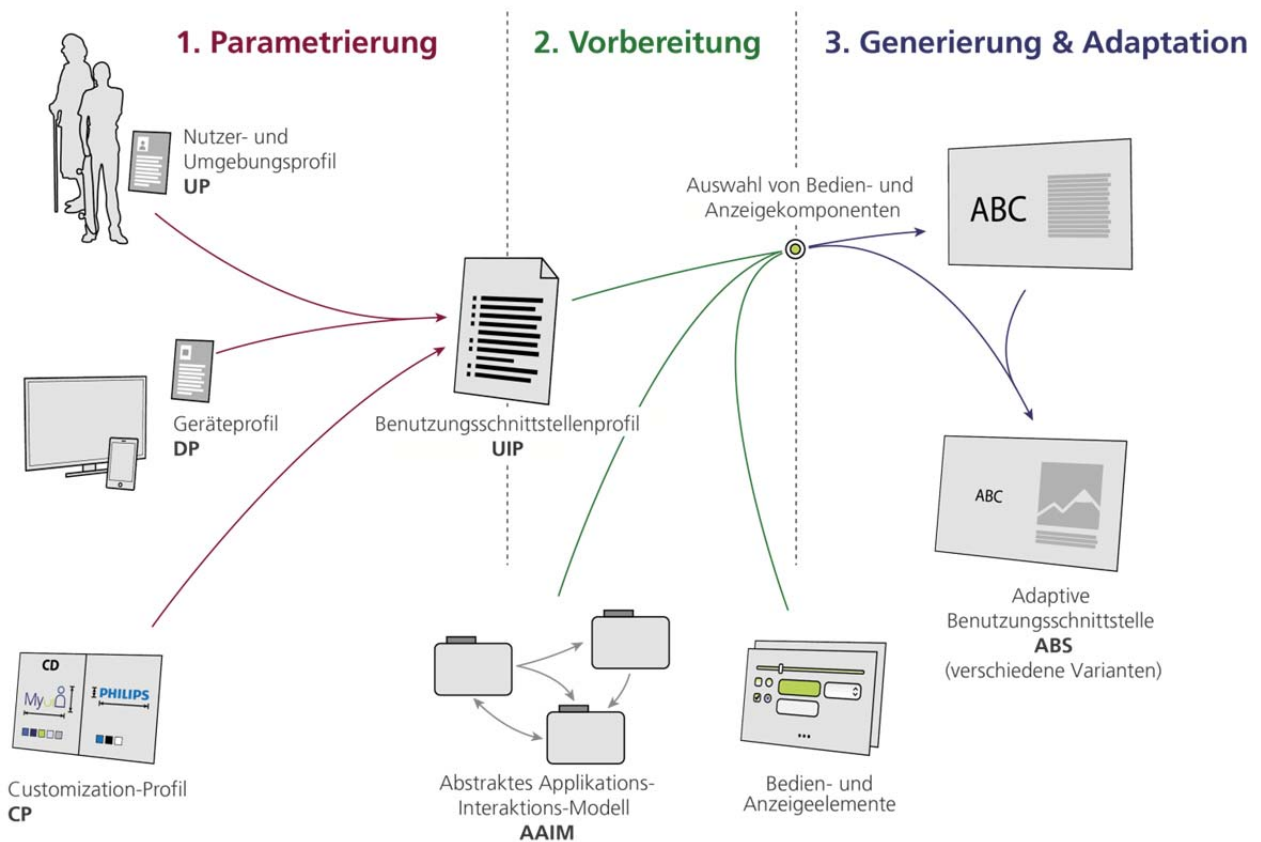


Abbildung 4 Dreistufiger Prozess zur Generierung und Adaptation der Benutzungsschnittstelle

3.5 Model-View Controller als konzeptionelle Basis der Software-Architektur

ABS erfordern eine strikte Trennung der Benutzungsschnittstelle von der Applikationslogik, damit ein und derselbe Zustand der Applikation in unterschiedlicher Art und Weise dargestellt und vom Nutzer manipuliert werden kann. Die Software-Architektur des Lösungsansatzes basiert auf dem Paradigma des Model-View-Controller (MVC, vgl. Gamma, Helm, Johnson & Vlissides, 1994), bei dem das *Model* die zugrundeliegenden Daten vorhält, die *View* die für den Nutzer wahrnehmbare Informationsausgabe repräsentiert und der *Controller* die Eingaben verarbeitet und spezifische Funktionen ausführt, die zu einer Veränderung des Modells und einer Aktualisierung der View führen können. Durch eine *Publish/Subscribe-Struktur* (vgl. Eugster et al., 2003) können die Abhängigkeiten zwischen Model und View weiter reduziert werden. So kann beispielsweise die Aktualität der View dadurch sichergestellt werden, dass die View unmittelbar benachrichtigt wird, sobald der Controller das Model verändert hat.

Im Lösungsansatz dieser Arbeit findet sich eine MVC-Struktur auf zwei Ebenen:

- *MVC auf Ebene der Gesamtapplikation*
Das Model der Gesamtapplikation repräsentiert die Datenbasis der Applikation (z.B. eine Wetterdatenbank für eine Wettervorhersage-Applikation). Der Controller nimmt Eingaben entgegen (z.B. die Auswahl eines neuen Ortes, für den Wetterdaten angezeigt werden sollen), manipuliert die Datenbasis gegebenenfalls und sorgt für eine entsprechende Aktualisierung der View (z.B. Anzeige der Wettervorhersage für den neu ausgewählten Ort), die die ABS der Applikation darstellt.
- *MVC auf Ebene der ABS*
Die ABS kann wiederum selbst als MVC-Struktur beschrieben werden. Hier umfasst das Model die Datenbasis, die der Generierung und Adaptation der ABS zugrundeliegt, insbesondere das AAIM sowie das Nutzer- und Umgebungsprofil (UP), das Geräteprofil (DP) und das Benutzungsschnittstellenprofil (UIP). Der Controller besteht aus der MyUI-Adaptationsengine und gewissen Adaptationsmechanismen auf der Ebene einzelner Entwurfsmuster. Der Controller ist u.a. verantwortlich für die Transformation zwischen den verschiedenen Modellen, z.B. der Transformation von UP zu UIP. Sobald Veränderungen im UIP festgestellt werden, wird die aktuelle Instanz der ABS, die als Komposition einer Auswahl von Bedien- und Anzeige Komponenten als View bezeichnet werden kann, benachrichtigt und von der Adaptationsengine (Controller) angepasst bzw. neu generiert.

3.6 Einfache und effiziente Applikationsmodellierung mit erweiterten Zustandsdiagrammen

Zur abstrakten Spezifikation einer adaptiven ABS wird ein eigens entwickeltes Format, das Abstrakte Applikations-Interaktions-Modell (AAIM: Abstract Application Interaction Model) verwendet. Das AAIM stellt eine Erweiterung der UML 2 State Machine Diagramme dar (vgl. OMG, 2011; oder Jeckle et al., 2004). Im AAIM sind die einzelnen Zustände einer Applikation und mögliche Übergänge zwischen den Zuständen repräsentiert. Für jeden Zustand gibt das AAIM an, welche Interaktionsoptionen dem Nutzer aktuell zur Verfügung stehen.

Die Vorteile einer ABS-Modellierung mit Zustandsdiagrammen umfassen unter anderem:

- Im Vergleich zu vielen anderen Beschreibungssprachen für Benutzungsschnittstellen (vgl. Moustakas et al., 2011) arbeiten Zustandsdiagramme auf einem höheren Abstraktionsniveau. Sie ermöglichen es, die Interaktion zwischen einem Nutzer und einer Applikation zu spezifizieren, ohne Aussagen über das Erscheinungsbild und die konkreten Interaktions- und Navigationsmechanismen der Benutzungsschnittstelle zu treffen. Das ist notwendig, da diese Aspekte der Benutzungsschnittstelle Gegenstand der Anpassungen zur Überwindung von Nutzungsbarrieren darstellen. So bildet das AAIM die Grundlage für alle möglichen Varianten der ABS, die durch Generierung und Adaptation aufgrund verschiedener Kontextbedingungen entstehen können.
- Zustandsdiagramme stellen inzwischen eine gängige Form der Spezifikation von Benutzungsschnittstellen dar (vgl. Harel, 1987 oder Horrocks, 1999). Durch die Verankerung in der UML (Unified Modeling Language) hat sie heute eine weite Verbreitung in der Praxis der Software-Entwicklung gefunden. Zusätzlich sind

Zustandsdiagramme aufgrund ihrer grafischen Notation leicht zu erlernen und können auch Laien eine gut verständliche Übersicht über ein interaktives System vermitteln. Damit bietet das AAIM gute Voraussetzungen für eine hohe Akzeptanz bei Software-Entwicklern.

Das AAIM bildet somit einerseits den Ausgangspunkt für die Generierung und alle Adaptionen der ABS. Andererseits bestimmt es weitgehend die Arbeit zur Entwicklung einer neuen adaptiven Applikation mit MyUI-Technologien, da die AAIM-Erstellung zur wesentlichen Aufgabe des Applikationsentwicklers wird. Mit dem Ziel einer hohen Akzeptanz bei Software entwickelnden Unternehmen ist im Rahmen des MyUI-Projekts eine gesamte Entwicklungsumgebung für ABS erstellt worden. Die einzelnen Werkzeuge und Mechanismen der Entwicklungsumgebung sind nicht zentraler Bestandteil dieser Arbeit. Sie können allerdings als Umsetzung des AAIM verstanden werden, da ihre Arbeitsweise stark von den Konzepten des AAIM bestimmt ist. Das AAIM wird in Kapitel 8 detailliert beschrieben.

4 Entwurfsmuster für adaptive Benutzungsschnittstellen

4.1 Adaptive Benutzungsschnittstellen durch Komposition von Entwurfsmustern

Um die große Heterogenität von Nutzern, Umgebungen und Geräten abzudecken, wird ein modularer Ansatz für ABS verfolgt, der auf der Komposition von multimodalen Entwurfsmustern beruht. Die Entwurfsmuster tragen das benötigte Gestaltungswissen, um den beschriebenen dreistufigen Prozess der ABS-Generierung und Adaptation auszuführen.

Jedes MyUI-Entwurfsmuster ist mit einer ausführbaren Softwarekomponente assoziiert. So wird die Zugänglichkeit einer ABS dadurch gewährleistet, dass diejenigen Entwurfsmuster ausgeführt werden, die bewährte Gestaltungslösungen für die aktuelle Interaktionssituation sowie die spezifischen Nutzereigenschaften, Umgebungsbedingungen und genutzten technischen Geräte bieten. Jede Instanz einer MyUI-ABS ist daher das Ergebnis einer Komposition von mehreren Entwurfsmustern, das heißt, das Ergebnis einer Komposition von Softwarekomponenten, die den Entwurfsmustern entsprechen. Eine Adaptation der ABS wird dadurch erreicht, dass ein gerade aktives Entwurfsmuster durch ein anderes Muster ersetzt wird, von dem angenommen wird, dass es für das beschriebene Problem im aktuellen Kontext das am besten geeignete Entwurfsmuster ist.

In Anlehnung an Borchers' Ansatz für Entwurfsmuster zur Gestaltung von Benutzungsschnittstellen (Borchers, 2001), der auf Alexander et al. (1977) basiert, werden alle Entwurfsmuster nach einer definierten Struktur beschrieben (vgl. Abschnitt 4.4). Der Hauptteil der Beschreibung jedes MyUI-Entwurfsmusters besitzt die folgenden drei wesentlichen Bestandteile:

1. Das *Problem* beschreibt ein Gestaltungsproblem oder eine bestimmte Situation der ABS, für das eine Gestaltungslösung benötigt wird. Damit charakterisiert das Problem den „Interaktionszweck“ des Musters, d.h. die Funktion, die das Muster für die ABS erfüllt. Für ein und dasselbe Problem können mehrere Entwurfsmuster existieren, die jeweils unterschiedliche Lösungen bieten. Der Problemausdruck kann damit als Auslöser eines Entwurfsmusters oder einer Menge von Entwurfsmustern interpretiert werden.
2. Der *Kontext* konkretisiert das Problem unter Bezug auf spezifische Anforderungen der Zugänglichkeit. Es wird genau angegeben, für welche spezifischen Eigenschaften des Nutzers, der technischen Ausstattung oder der Umgebung das vorliegende Muster eine Lösung für das beschriebene Problem bietet. Durch die Einschränkungen des Kontext-Arguments wird die Menge der für die aktuelle Situation bzw. für das aktuelle Problem passenden Lösungen auf genau ein geeignetes Muster beschränkt.
3. Die *Lösung* beschreibt in einer möglichst generischen Form, wie das Problem im spezifizierten Kontext gelöst werden kann.

Damit können die beiden Felder „Problem“ und „Kontext“ als ein durch UND verbundenes If-Argument verstanden werden. Und das Lösungsfeld entspricht dem Then-Statement des Entwurfsmusters. Eine Kombination aus einem bestimmten Gestaltungsproblem und einem Kontext wird in der Folge auch als „Gestaltungsbedingung“ bezeichnet.

4.2 Typen von Entwurfsmustern

Das MyUI-Entwurfsmusterrepository enthält verschiedene Kategorien von Entwurfsmustern. Jeder Mustertyp erfüllt unterschiedliche Funktionen im Adaptationsprozess (s. Abbildung 5):

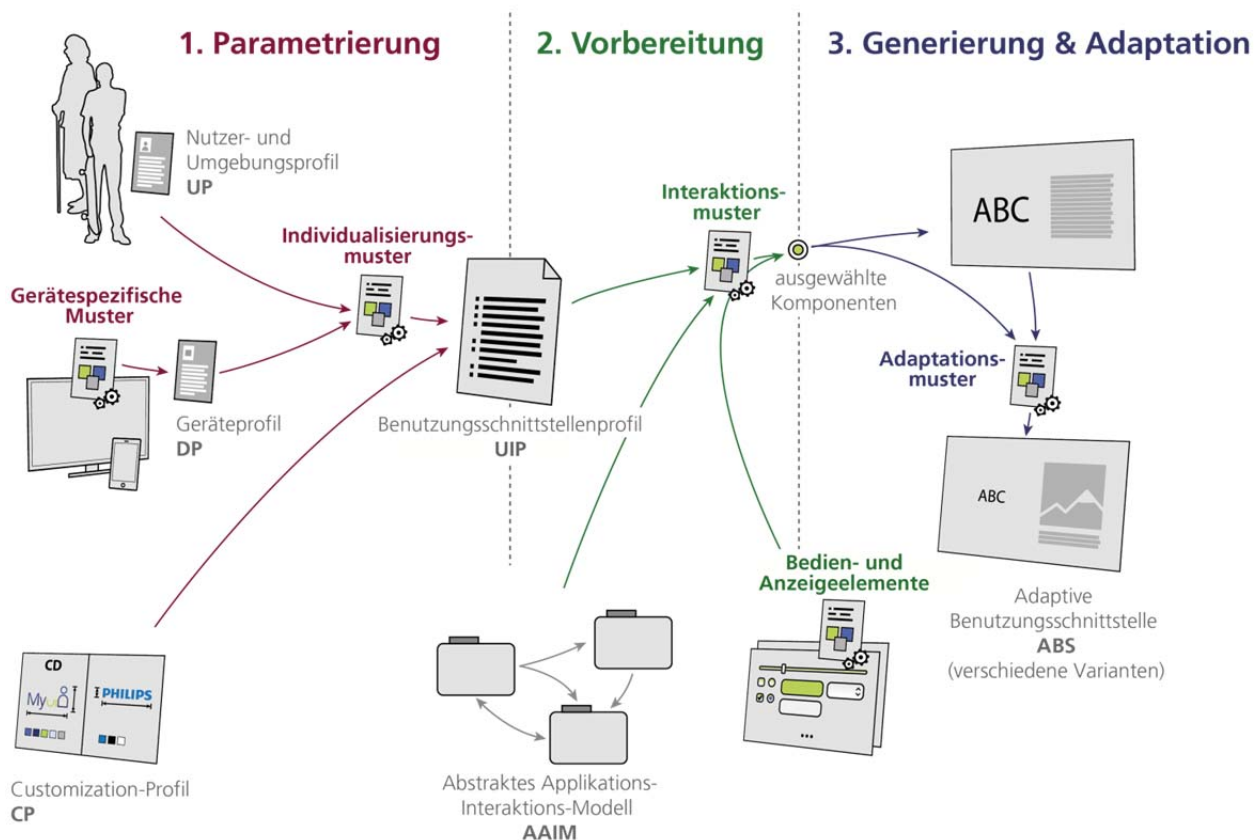


Abbildung 5 Verschiedene Entwurfsmustertypen im Adaptationsprozess

- *Gerätespezifische Muster (Device-specific Patterns)* erzeugen das Geräteprofil (DP: Device Profile) auf der Grundlage der von den verwendeten Ein- und Ausgabegeräten mitgeteilten Gerätemerkmale. Das DP besteht aus einer Menge von globalen Variablen, die die ABS-Parametrierung vorbereiten, indem sie Wertebereiche für Einstellungen der Benutzungschnittstelle vorgeben, aus denen die Individualisierungsmuster die am besten geeigneten Werte auswählen (z.B. mögliche Schriftgrößen für die aktuellen Ausgabegeräte). Gerätespezifische Muster definieren Variablen, die während der gesamten Interaktion mit derselben technischen Ausstattung stabil bleiben. Wird die technische Ausstattung während einer Interaktionssequenz verändert, dann wird das DP entsprechend aktualisiert. So kann beispielsweise durch das Hinzufügen eines neuen Ausgabegeräts während einer Interaktionssequenz eine Laufzeitanpassung der ABS angestoßen werden.
- *Individualisierungsmuster (Individualization Patterns)* definieren globale Einstellungen der ABS zur Passung an individuelle und situative Bedürfnisse, z.B. große Schrift für Nutzer mit Sehbeeinträchtigungen oder überwiegend ikonografische Darstellungsformen für Nutzer mit Problemen der Sprachwahrnehmung.

Da sich das verfügbare Wissen über Nutzerbedürfnisse und andere Kontextbedingungen während einer Interaktionssequenz ändern kann, spielen Individualisierungsmuster eine wichtige Rolle für Adaptationen während der Nutzung.

Diese ersten beiden Musterkategorien legen die Grundlage für die Generierung der ABS. Gemeinsam setzen und verändern sie globale Variablen, die die gesamte ABS über alle Interaktionssituationen der Applikation hinweg beeinflussen. Ihr Beitrag zum Prozess der ABS-Generierung und Anpassung schlägt sich im *Benutzungsschnittstellenprofil* (UIP: User Interface Profile) nieder und wird in Kapitel 5 (Parametrierung der ABS) näher beschrieben. Ein zusätzlicher Einfluss auf das UIP kommt vom Customization-Profil (CP: Customization Profile), über das der Entwickler das Erscheinungsbild der ABS an die Identität eines Unternehmens, eines Projekts oder einer Marke anpassen kann. Die Einstellungen des CP werden im UIP ohne weitere Transformation direkt übernommen. Daher werden für das Customizing keine eigenen Entwurfsmuster benötigt.

- *Interaktionsmuster (Interaction Patterns)*
stellen Interaktionskomponenten wie zum Beispiel Auswahllisten oder Menüs zur Verfügung. Interaktionsmuster beziehen sich auf die Interaktionssituationen, wie sie im AAIM referenziert werden, indem jedes Interaktionsmuster eine konkrete Lösung für eine Interaktionssituation bietet. Für jede Interaktionssituation gibt es ein Bündel von verschiedenen Interaktionsmustern. Alle Interaktionsmuster eines Bündels beziehen sich auf dieselbe Interaktionssituation und dienen damit demselben Zweck in der Interaktion. Jedoch unterscheiden sich die konkreten Lösungen bezüglich ihres Erscheinungsbilds und der verwendeten Eingabemechanismen, um jeweils unterschiedliche Nutzerbedürfnisse und Kontextbedingungen optimal zu unterstützen. Die Auswahl des jeweils am besten geeigneten Interaktionsmusters erfolgt auf der Basis spezifischer Variablen des UIP.
- *Bedien- und Anzeigeelemente (User Interface Elements)*
sind die generischen Bausteine, aus denen sich die in den Interaktionsmustern beschriebenen Komponenten zusammensetzen. Zum Beispiel erfordert ein Interaktionsmuster, das eine Auswahlliste zur Verfügung stellt, Schaltflächen (Buttons) für die einzelnen Optionen der Auswahlliste. Neben verschiedenen Typen von Buttons umfassen die Bedien- und Anzeigeelemente u.a. Felder zur Texteingabe und Textausgabe sowie einfache Darstellungselemente wie Beschriftungen oder Fortschrittsanzeigen. Darüber hinaus umfassen die Bedien- und Anzeigeelemente auch generische Interaktionsmechanismen, die in verschiedenen Interaktionsmustern wiederkehren. Beispiele hierfür sind akustische Interaktionsmöglichkeiten wie die Sprachausgabe und Spracheingabe gerade verfügbarer Optionen oder die Definition der Funktionsweise grundlegender Bedienmechanismen wie der Cursor-Navigation.

Interaktionsmuster und Bedien- und Anzeigeelemente stellen die Bausteine dar, aus denen eine konkrete ABS zusammengesetzt wird. Die Auswahl der am besten geeigneten Lösungen erfolgt durch die Interaktionsmuster auf Grundlage der im AAIM definierten Interaktionssituationen und des UIP. Jedes Interaktionsmuster ruft eine Menge benötigter Bedien- und Anzeigeelemente auf. Der Auswahlprozess der am besten geeigneten Komponenten und Elemente wird in Kapitel 6 näher beschrieben. Die Erzeugung und Aktualisierung der Benutzungsschnittstelle wird in Kapitel 7 ausgeführt.

Die beiden folgenden Mustertypen werden unter dem Begriff *Adaptationsmuster* zusammengefasst. Sie beschreiben die dynamischen Aspekte der Anpassungen und definieren die Mechanismen, mit denen zwischen verschiedenen Instanzen einer ABS gewechselt wird.

- *Adaptationsausführungsmuster (Adaptation Rendering Patterns)* definieren die grafische Ausführung des Übergangs von einer Instanz der ABS zur nächsten. Dabei wird sowohl der Wechsel zwischen zwei Interaktionsmustern eines Bündels als auch die Veränderung von Bedien- und Anzeigeelementen behandelt.
- *Adaptationsdialogmuster (Adaptation Dialogue Patterns)* spezifizieren den Dialog, der zwischen Nutzer und ABS rund um eine Adaptation stattfindet. Insbesondere bei automatischen Anpassungen, die durch eine Erkennung von Kontextveränderungen ausgelöst werden, sollen Adaptationsdialogmuster sicherstellen, dass der Benutzer die Adaptation bemerkt und das Adaptationsverhalten des Systems steuern kann. Durch ihren Beitrag zur Transparenz und Kontrollierbarkeit der automatischen Anpassungen werden sie als eine wesentliche Voraussetzung für die Nutzerakzeptanz betrachtet.

Eine detailliertere Beschreibung der verschiedenen Mustertypen findet sich in den Kapiteln 5 bis 7, in denen die drei Phasen der ABS-Generierung und Adaptation genauer beschrieben werden.

4.3 Entwurfsmustersprache

Die Entwurfsmuster stehen in engen wechselseitigen Beziehungen. In einer hierarchisch aufgebauten Entwurfsmustersprache (vgl. Borchers, 2001) verweisen Muster höherer Ebenen bei der Beschreibung einer Lösung auf kleinteiligere Muster, die einzelne Aspekte dieser Lösung detaillierter ausarbeiten. Und die Muster, die auf tieferen Ebenen arbeiten, können wiederum nur in einem bestimmten Kontext eingesetzt werden, der erst durch die Anwendung höhergelegener Muster geschaffen wird. Dabei spielen die Beziehungen zwischen den Mustern eine wichtige Rolle. Die in diesem Abschnitt beschriebenen Relationen werden definiert, um eine systematische Referenzierung dieser Beziehungen sicherzustellen.

Im Folgenden werden Relationen durch die Schriftart `Courier New` hervorgehoben. Einzelne Entwurfsmuster werden durch Kleinbuchstaben bezeichnet. Mengen M von Mustern werden durch kursive Großbuchstaben referenziert. Durch die beschriebenen Mustertypen ergeben sich folgende Mengen:

- D bezeichnet die Menge der gerätespezifischen Muster.
- I bezeichnet die Menge der Individualisierungsmuster.
- K bezeichnet die Menge der Interaktionsmuster.
- B bezeichnet die Menge der Bedien- und Anzeigeelemente.
- A bezeichnet die Menge der Adaptationsmuster, mit den Teilmengen
 - AA für die Menge der Adaptationsausführungsmuster und
 - AD für die Menge der Adaptationsdialogmuster.

Ein Paar von Entwurfsmustern desselben Typs wird *homogen* genannt:
 $\{a, b\}$ ist homogen gdw. $a, b \in M$ mit $M \in \{D, I, K, B, A\}$.

Darüber hinaus werden globale Variablen der Profile DP, UP und UIP durch kursive Kleinbuchstaben bezeichnet. Mengen von Variablen werden durch kursive Großbuchstaben in Fettschrift gekennzeichnet. **DP** bezeichnet die Menge der im DP gesetzten Variablen. Entsprechend umfassen **UP** und **UIP** die Mengen der Variablen von UP und UIP.

4.3.1 Die Relation substituiert

Für die beiden Entwurfsmuster a und b gilt:

a substituiert b, wenn

- die Entwurfsmuster a und b homogen sind und demselben Interaktionszweck dienen, d.h. dasselbe Problem adressieren, und wenn
- beide Muster sich gegenseitig ausschließen, d.h. wenn beide Muster niemals gleichzeitig aktiv sein können.

Die Relation substituiert ist für homogene Musterpaare aus D, I, K und AD definiert:

$\text{substituiert} \subseteq M \times M$ mit $M \in \{D, I, K, AD\}$.

Die Relation substituiert ist symmetrisch und transitiv. Damit gilt:

- a substituiert b gdw. b substituiert a.
- Wenn a substituiert b und b substituiert c, dann a substituiert c.

Durch die Relation substituiert wird das Entwurfsmusterrepositorium in *Bündel* von Entwurfsmuster strukturiert. Alle Muster, die denselben Interaktionszweck erfüllen, jedoch unterschiedliche Nutzerbedürfnisse oder Kontextbedingungen unterstützen, werden durch die Relation substituiert zu einem *Bündel* zusammengefasst. So setzen beispielsweise alle Individualisierungsmuster eines Bündels eine oder mehrere bestimmte Variablen des UIP jeweils auf unterschiedliche Werte. Und die Interaktionsmuster eines Bündels stellen jeweils unterschiedliche Bedien- und Anzeigekomponenten für dieselbe Interaktionssituation zur Verfügung.

Manche Bündel besitzen ein *Default-Muster*, das angewendet wird, wenn im referenzierten Profil die Standardeinstellungen bestehen. Die weiteren Entwurfsmuster eines Bündels werden *Varianten* genannt. Für ein Bündel $BÜ_m$ gilt:

- $BÜ_m \subseteq M$ mit $M \in \{D, I, K, AD\}$
- $BÜ_m := \{m \mid m \text{ substituiert } n \wedge n \in BÜ_m\}$
- $BÜ_m \cap BÜ_n = \{\}$ für alle $m \neq n$ und $\neg (m \text{ substituiert } n)$

4.3.2 Die Relation benötigt

Für die beiden Entwurfsmuster a und b gilt:

a benötigt b,

wenn das Entwurfsmuster b Teile der von a beschriebenen Lösung näher detailliert.

In diesem Fall erfordert die Ausführung des Musters a, das eine Lösung für ein übergeordnetes Problem bietet, den Aufruf des Musters b zur näheren Spezifikation eines bestimmten Teilaspekts seiner Lösung. Die Aktivierung des Musters b ist damit ein Teil der Lösung von a.

$\text{benötigt} \subseteq K \times B$.

Die Relation `benötigt` ist für geordnete Musterpaare aus der Menge K der Interaktionsmuster und der Menge B der Bedien- und Anzeigeelemente definiert. Interaktionsmuster referenzieren auf Bedien- und Anzeigeelemente für die Beschreibung generischer Elemente der ABS. Zum Beispiel kann ein Interaktionsmuster, das eine Titelzeile definiert, zur Beschreibung der verwendeten Schaltflächen auf das Bedien- und Anzeigeelement „Navigation Button“ zurückgreifen. Damit kann u.a. gewährleistet werden, dass für die Navigation in allen Interaktionssituationen einheitlich dieselben Bedien- und Anzeigeelemente verwendet werden.

Die Relation `benötigt` dient damit der vertikalen Strukturierung des Musterrepositoriums. Entwurfsmuster höherer Ebene (Interaktionsmuster), die gesamte Komponenten der ABS beschreiben, werden durch sie mit Entwurfsmustern niedrigerer Ebenen (Bedien- und Anzeigeelemente) verbunden, die kleinere, generische Aspekte und Elemente der ABS beschreiben.

Außerdem ist die Relation `benötigt` auf dem Kreuzprodukt der Menge AA der Adaptationsausführungsmuster definiert:

$\text{benötigt} \subseteq AA \times AA$.

In diesem Fall definiert ein übergeordnetes Adaptationsausführungsmuster, das direkt einem Interaktionsmusterbündel zugeordnet ist, welche weiteren Adaptationsausführungsmuster in welcher Reihenfolge aufgerufen werden, um die aktuellen Bedien- und Anzeigeelemente während einer Laufzeitanpassung zu verändern.

4.3.3 Die Relation wird benötigt von

Die Relation `wird benötigt von` ist die Umkehrrelation von `benötigt` (4.3.2). Es gilt:

– $\text{wird benötigt von} \subseteq B \times K$

oder

$\text{wird benötigt von} \subseteq AA \times AA$.

– a wird benötigt von b, wenn das Entwurfsmuster a Teile der von b beschriebenen Lösung näher detailliert.

4.3.4 Die Relation setzt <Variable> wie benötigt von

Für die beiden Entwurfsmusterbündel $BÜ_i$ und $BÜ_j$ und die Variable v gilt:

$BÜ_i$ setzt v wie benötigt von $BÜ_j$,

wenn die Muster des Bündels $BÜ_i$ den Wert einer globalen MyUI-Variablen v setzen, der Teile der Entwurfsmusterlösungen des Bündels $BÜ_j$ spezifiziert. Dem entsprechend wird die Variable v in den Lösungsbeschreibungen (Then-Statement) der Entwurfsmuster des Bündels $BÜ_j$ referenziert.

setzt <Variable> wie benötigt von $\subseteq BÜ_i \times v \times BÜ_j$,

mit $BÜ_i \subseteq D$, $v \in \mathbf{DP}$ und $BÜ_j \subseteq I$.

In diesem Fall unterstützt die Relation setzt <Variable> wie benötigt von den Prozess der ABS-Parametrierung, indem sie Bündel von gerätespezifischen Mustern mit Bündel von Individualisierungsmustern verbindet. Schrittweise werden Einstellungen der ABS im DP zunächst auf die aktuellen Geräte angepasst und dann auf die individuellen Nutzer- und Umgebungsbedingungen. So werden zum Beispiel durch die gerätespezifischen Muster des Bündels *DEVICE FONT SIZES* zunächst kleine, mittlere und große Schriftgrößen definiert, die für das aktuelle Gerät geeignet sind. Ob schließlich die kleine, mittlere oder große Schrift angezeigt wird, wird dann durch die Individualisierungsmuster des Bündels *FONT SIZE* bestimmt, die dabei auf das im UP verfügbare Wissen über die Sehstärke des Nutzers zurückgreifen.

Des Weiteren gilt für das Entwurfsmusterbündel $BÜ_i$, die Variable v und das Entwurfsmuster m :

$BÜ_i$ setzt v wie benötigt von m ,

wenn die Muster des Bündels $BÜ_i$ den Wert einer globalen Variablen v setzen, der Teile der Lösung des Entwurfsmusters m spezifiziert. Dem entsprechend wird die Variable v in der Lösungsbeschreibung (Then-Statement) des Entwurfsmusters m referenziert.

setzt <Variable> wie benötigt von $\subseteq BÜ_i \times v \times m$,

mit $BÜ_i \subseteq I$, $v \in \mathbf{UIP}$ und $m \in \{AA; B\}$.

Im zweiten Fall werden Einstellungen des UIP in den Lösungen von Adaptationsausführungsmustern ($m \in AA$) bzw. Bedien- und Anzeigeelementen ($m \in B$) referenziert. Ein Beispiel ist die *adaptationRenderingTime* (UIP-Variable), die je nach *processingSpeed* (UP-Variable) durch Individualisierungsmuster eingestellt wird und deren Wert später in fast allen Adaptationsausführungsmustern als Parameter zur Steuerung der Animationsgeschwindigkeit verwendet wird.

4.3.5 Die Relation benötigt <Variable> wie gesetzt von

Die Relation benötigt <Variable> wie gesetzt von ist die Umkehrrelation von setzt <Variable> wie benötigt von (4.3.4). Es gilt:

- benötigt <Variable> wie gesetzt von $\subseteq BÜ_i \times v \times BÜ_j$,
mit $BÜ_i \subseteq I$, $v \in \mathbf{DP}$ und $BÜ_j \subseteq D$.
- $BÜ_i$ benötigt v wie gesetzt von $BÜ_j$,
wenn die Entwurfsmuster des Bündels $BÜ_i$ in ihren Lösungsbeschreibungen auf eine Variable v zurückgreifen, deren Wert in den Entwurfsmustern des Bündels $BÜ_j$ gesetzt wird.

Oder:

- benötigt `<Variable>` wie gesetzt von $\subseteq m \times v \times BÜ_i$,
mit $m \in \{AA; B\}$, $v \in \mathbf{UIP}$ und $BÜ_i \subseteq I$.
- m benötigt v wie gesetzt von $BÜ_i$,
wenn das Entwurfsmuster m in seiner Lösungsbeschreibung auf eine Variable v zurückgreift, deren Wert in den Entwurfsmustern des Bündels $BÜ_i$ gesetzt wird.

4.3.6 Die Relation `setzt <Variable> wie genutzt von`

Für die beiden Entwurfsmusterbündel $BÜ_i$ und $BÜ_j$ und die Variable v gilt:

$BÜ_i$ setzt v wie genutzt von $BÜ_j$,

wenn die Muster des Bündels $BÜ_i$ den Wert einer globalen MyUI-Variablen v setzen und v genutzt wird, um das am besten geeignete Muster aus dem Bündel $BÜ_j$ auszuwählen. Demzufolge wird die Variable v im Lösungsfeld (Then-Statement) der Musterbeschreibungen des Bündels $BÜ_i$ und im Kontextfeld der Musterbeschreibungen (If-Statement) des Bündels $BÜ_j$ referenziert.

`setzt <Variable> wie genutzt von` $\subseteq BÜ_i \times v \times BÜ_j$,

mit $BÜ_i \subseteq D$, $v \in \mathbf{DP}$ und $BÜ_j \subseteq D$

oder $BÜ_i \subseteq D$, $v \in \mathbf{DP}$ und $BÜ_j \subseteq I$

oder $BÜ_i \subseteq I$, $v \in \mathbf{UIP}$ und $BÜ_j \subseteq K$.

Damit erfüllt die Relation `setzt <Variable> wie genutzt von` verschiedene Funktionen:

Einerseits verbindet sie verschiedene Bündel von gerätespezifischen Mustern, um Abhängigkeiten zwischen den Variablen des DP zu unterstützen. So wird beispielsweise je nach Ausprägung der Gerätekategorie (DP-Variable *deviceCategory*), die von den Mustern des Bündels *CATEGORIZE DEVICE* gesetzt wird, das geeignete gerätespezifische Muster aus dem Bündel *LAYOUT GRID AND STRUCTURE* ausgewählt, um die Layoutstruktur der ABS festzulegen.

Zweitens spielt sie eine Rolle bei der schrittweisen Parametrierung der ABS, indem die Auswahl der geeigneten Individualisierungsmuster u.a. von Parametern des DP abhängt, die von gerätespezifischen Mustern gesetzt worden sind. So wird beispielsweise das Individualisierungsmuster „Tonal Audio Output – on“ nur dann aktiviert, wenn ein Lautsprecher verfügbar ist, d.h. wenn die DP-Variable *speakerAvailable*, für die das Musterbündel *AVAILABLE SPEAKER* verantwortlich ist, den Wert „true“ besitzt.

Schließlich unterstützt sie den dreistufigen Prozess der ABS-Generierung, indem sie die ABS-Parametrierung mit der der ABS-Vorbereitung verbindet. Bündel von Individualisierungsmustern, die für die Erstellung und Aktualisierung des UIP verantwortlich sind, werden über diese Relation mit Bündeln von Interaktionsmustern verknüpft, aus denen anhand der gesetzte Variablen v die am besten geeignete Variante ausgewählt wird. So bestimmen beispielsweise die verschiedenen Individualisierungsmuster des Bündels *SCREEN COMPLEXITY* den Wert der UIP-Variablen *screenComplexity*, um die Anzahl der gleichzeitig dargestellten Bildelemente an die individuelle Aufmerksamkeitskapazität des Nutzers (UP-Variable *attention*) anzupassen. Für das Hauptmenü zum Beispiel wird dann je nach Wert der UIP-Variablen *screenComplexity* aus dem Bündel *Main Menu* ein Interaktionsmuster ausgewählt, das dieser Anforderung genügt.

4.3.7 Die Relation nutzt <Variable> wie gesetzt von

Die Relation nutzt <Variable> wie gesetzt von ist die Umkehrrelation von setzt <Variable> wie genutzt von (4.3.6). Es gilt:

- nutzt <Variable> wie gesetzt von $\subseteq B\ddot{U}_i \times v \times B\ddot{U}_j$,
mit $B\ddot{U}_i \subseteq D$, $v \in \mathbf{DP}$ und $B\ddot{U}_j \subseteq D$
oder $B\ddot{U}_i \subseteq K$, $v \in \mathbf{UIP}$ und $B\ddot{U}_j \subseteq I$.
- $B\ddot{U}_i$ nutzt v wie gesetzt von $B\ddot{U}_j$,
wenn der Wert der Variablen v des DP von den gerätespezifischen Mustern des Bündels $B\ddot{U}_j$ bestimmt wird und wenn diese Variable v genutzt wird, um aus dem Musterbündel $B\ddot{U}_i$ das am besten geeignete gerätespezifische Muster auszuwählen,
oder wenn der Wert der Variablen v des UIP von den Individualisierungsmustern des Bündels $B\ddot{U}_j$ bestimmt wird und wenn diese Variable v genutzt wird, um aus dem Entwurfsmusterbündel $B\ddot{U}_i$ das am besten geeignete Interaktionsmuster auszuwählen.

4.3.8 Die Relation erweitert

Für die beiden Entwurfsmuster a und b gilt:

a erweitert b,

wenn die Lösung des Entwurfsmusters a die Lösung des übergeordneten Muster b vollständig umfasst und diese um spezifische Lösungskomponenten ergänzt. In diesem Fall wird bei der Ausführung des Musters a, das eine Lösung für ein spezifisches Problem bietet, auf die allgemeinere Lösungsbeschreibung des Musters b zurückgegriffen.

erweitert $\subseteq B \times B$.

Die Relation erweitert ist für geordnete, homogene Musterpaare aus der Menge B der Bedien- und Anzeigeelemente definiert. Im Gegensatz zu den anderen Relationen, beschreibt diese Relation keinen wesentlichen Aspekt des Zusammenspiels verschiedener Entwurfsmuster während der ABS-Generierung und Adaptation. Durch eine Strukturierung der Menge B der Bedien- und Anzeigeelemente hilft sie, eine Konsistenz zwischen ähnlichen Bedien- und Anzeigeelementen der ABS zu sichern und Redundanzen in der Implementierung zu vermeiden. So verbindet die Relation erweitert beispielsweise verschiedene Bedien- und Anzeigeelemente, die spezifische Buttons wie z.B. Menübuttons, Navigationsbuttons, Aktionsbuttons, etc. definieren, mit einem übergeordneten Muster, das die Gemeinsamkeiten aller Buttontypen beschreibt. Damit greift die Relation erweitert das Konzept der Vererbung aus der objektorientierten Programmierung auf (Booch et al., 2007). Während die einzelnen spezifischen Buttontypen direkt von Interaktionsmustern über die Relation benötigt aufgerufen werden, kann das übergeordnete Muster nie alleine verwendet werden. Nach den Konventionen der objektorientierten Programmierung könnte man diese Muster abstrakte Entwurfsmuster nennen.

4.3.9 Die Relation wird erweitert von

Die Relation `wird erweitert von` ist die Umkehrrelation von `erweitert` (4.3.8). Es gilt:

- `wird erweitert von` $\subseteq B \times B$.
- `a` `wird erweitert von` `b`, wenn die Lösung des abstrakten Entwurfsmusters `a` vollständig in `b` enthalten ist und durch die Lösungsbeschreibung des Musters `b` um spezifische Aspekte ergänzt wird.

4.4 Entwurfsmusterrepositorium

Das gesamte Wissen, das benötigt wird, um die ABS zu generieren und zu adaptieren, wird in Form von Entwurfsmustern beschrieben. Diese Entwurfsmuster werden in dem MyUI-Entwurfsmusterrepositorium gespeichert und vorgehalten.

4.4.1 Entwurfsmuster in natürlicher Sprache und als Softwarekomponenten

Jedes Entwurfsmuster des Repositoriums ist in natürlicher Sprache beschrieben. Diese Beschreibung folgt einem definierten Format. Die Grundstruktur aller Musterbeschreibungen basiert auf den Arbeiten von Borchers (2001) und enthält die folgenden Abschnitte:

- Name
Ein aussagekräftiger Name verdeutlicht die Funktion des Musters und dient zur eindeutigen Referenz¹⁹.
- Problem
Das Problemargument beschreibt das Gestaltungsproblem, für das das Muster eine Lösung bietet.
- Kontext
Die Kontextbeschreibung konkretisiert die Problemstellung. Es wird genau angegeben, unter welchen Kontextbedingungen dieses Muster die passende Lösung für das Problem bietet. Die Kontextbeschreibung ist wichtig, wenn mehrere Muster dasselbe Problem adressieren, d.h. wenn Musterbündel bestehen. Für Mustertypen, die keine Bündel besitzen, d.h. für die die Relation `substituiert` nicht definiert ist, ist das Kontextfeld nicht relevant und wird in der Beschreibung weggelassen.
- Lösung
Hier wird die Lösung beschrieben.
- Code-Referenz
Ein technologieunabhängiger Pseudo-Code beschreibt die Lösung in einer technischen, formalen Weise und bietet eine Referenz für die Implementierung.

¹⁹ Sollten natürlichsprachliche Namen nicht ausreichend trennscharf erscheinen, kann zusätzlich ein alphanumerischer Identifikator verwendet werden, um die Eindeutigkeit zu garantieren, auch wenn mehrere Personen parallel Entwurfsmuster erstellen.

- Diagramm
Eine grafische Darstellung bietet eine Illustration der Lösung.
- Rationale
erläutert und begründet die Lösung, z.B. durch Literaturverweise oder theoretische Überlegungen.
- Bewertung
Eine Bewertung auf einer ordinalen Skala gibt an, wie verlässlich und fundiert das Entwurfsmuster ist. So können auch weniger gut validierte Lösungsvorschläge in das Repositorium aufgenommen und entsprechend gekennzeichnet werden. Die folgenden drei Bewertungskategorien werden angewendet:
 - * Das Muster ist nicht oder nur unzureichend validiert. Weitere empirische Forschung wird benötigt.
 - ** Die Validierungsergebnisse für dieses Muster sind entweder weniger aussagekräftig (z.B. kleine Stichprobe, Mängel im Untersuchungsdesign, etc.) oder stammen nicht direkt aus demselben Untersuchungsfeld und können daher nicht ohne weiteren Transfer zur Bestätigung dieses Musters herangezogen werden.
 - *** Das Muster ist gut validiert. Sauber durchgeführte Studien belegen die Gültigkeit für den Anwendungsbereich und die adressierten Nutzergruppen des Musters.

Zusätzliche Felder der Musterbeschreibungen beziehen sich auf Relationen zu anderen Mustern, wie sie in Abschnitt 4.3 dargestellt sind. Je nach Mustertyp enthalten die Musterbeschreibungen zusätzliche Felder nur für die Relationen, die für den jeweiligen Mustertyp definiert sind.

Damit die Entwurfsmuster zur Generierung und Adaptation einer ABS verwendet werden können, besitzt jedes Muster des Repositoriums eine software-technische Entsprechung. Die Muster sind mit wiederverwendbaren Softwarekomponenten assoziiert, die für die Generierung und Adaptation der ABS verantwortlich sind. So lässt sich jede Situation einer Applikation durch eine Menge von Gestaltungsproblemen beschreiben, für die das Repositorium jeweils eine oder mehrere Lösungen (d.h. Entwurfsmuster) vorhält. Durch einen Abgleich der Kontextbedingungen der in Frage kommenden Muster mit dem Geräteprofil (DP) und dem Nutzer- und Umgebungsprofil (UP) werden die am besten geeigneten Muster ausgewählt. Durch eine Ausführung der mit den ausgewählten Mustern assoziierten Softwarekomponenten wird dann die ABS generiert bzw. adaptiert. Durch diese Verbindung von menschenlesbaren Entwurfsmustern und maschinenlesbaren Softwarekomponenten schafft das MyUI-Entwurfsmusterrepositorium eine Brücke zwischen Empfehlungen und Richtlinien für barrierefreie Interaktionsgestaltung einerseits und generativen Benutzungsschnittstellen andererseits.

4.4.2 Erweiterbares Repository

Das Repository wäre vollständig, wenn es für jede denkbare Interaktionssituation und jede beliebige Kombination von individuellen Voraussetzungen des Nutzers, der Umgebung und der technischen Ausstattung Entwurfsmuster zur Komposition einer barrierefreien Benutzungsschnittstelle bereithalten würde. Da eine derartige Vollständigkeit in der Praxis nicht erreichbar ist, ist die Erweiterbarkeit des Repositorys von großer Bedeutung.

Die Erweiterbarkeit des MyUI-Musterrepositorys spiegelt sich in zwei wesentlichen Eigenschaften wider: Einerseits können neue, zusätzliche Entwurfsmuster dem Repository hinzugefügt werden, ohne dass eine komplette neue Version eines monolithischen Musterkatalogs erstellt werden müsste. Damit können Erweiterungen des Regelwerks realisiert werden, ohne umfangreiche Änderungen an bestehenden Mustern zu erfordern. Andererseits können einzelne bestehende Entwurfsmuster erweitert und verfeinert werden, ohne vielfältige und schwer kontrollierbare Einflüsse auf andere Gestaltungsbereiche und Muster befürchten zu müssen. Diese Modularität ist der Struktur der MyUI-Entwurfsmustersprache zu verdanken, die sich aufgrund der verschiedenen Mustertypen (Abschnitt 4.2) und der Relationen zwischen den Mustern (Abschnitt 4.3) ergibt.

Der Anlass für Erweiterungen des Musterrepositorys sind in der Regel neue Gestaltungsbedingungen, die durch die bestehenden Muster noch nicht abgedeckt sind. Das heißt, es handelt sich entweder um ein neues Problem, für das noch kein Musterbündel besteht oder um neue Kontextbedingungen, die noch von keinem Muster adressiert werden. Erweiterungen des Repositorys verlaufen grundsätzlich nach dem in Abbildung 6 dargestellten Schema:

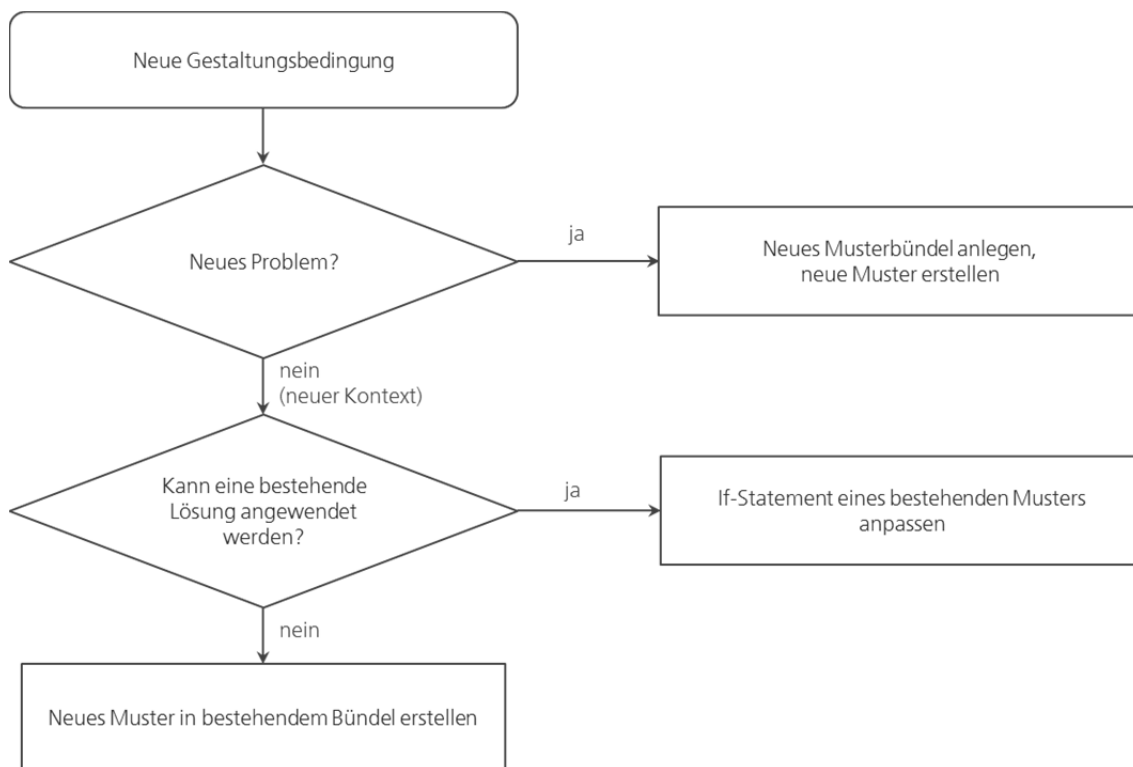


Abbildung 6 Generisches Modell für Erweiterungen des MyUI-Entwurfsmusterrepositorys

- Soll ein neues Problem abgedeckt werden, wird grundsätzlich ein neues Musterbündel benötigt, das einzelne Muster für unterschiedliche wichtige Kontextbedingungen enthält.
- Wird lediglich eine neue Kontextbedingung adressiert, so genügen Veränderungen in bereits bestehenden Bündeln. Dabei können mehrere Musterbündel betroffen sein.
 - Wenn zur Abdeckung der neuen Kontextbedingungen auf ein bereits bestehendes Muster des Bündels zurückgegriffen werden kann, genügt es, das If-Statements des Musters entsprechend anzupassen.
 - Andernfalls wird ein neues Muster erstellt.

Trotz der modularen Struktur des Repositoriums können bei Erweiterungen gewisse Wechselwirkungen entstehen, die eine Bearbeitung von bestehenden Mustern erfordern. Dabei können Wechselwirkungen *innerhalb* und *zwischen* den verschiedenen Mustertypen auftreten:

- *Wechselwirkungen innerhalb eines Mustertyps*
Innerhalb eines Mustertyps beschränken sich die Wechselwirkungen neuer bzw. veränderter Entwurfsmuster mit bestehenden Mustern auf die Mengen von Entwurfsmustern, die durch die Relation `substituiert` zu Musterbündeln zusammengefasst werden (vgl. Abschnitt 4.3.1). Jedes Mal, wenn neue Muster hinzugefügt oder bestehende Muster verändert werden, ist darauf zu achten, dass für jede mögliche Kontextausprägung genau ein Entwurfsmuster des jeweiligen Bündels aktiviert wird. Das heißt, die If-Statements der Muster eines Bündels müssen genau so formuliert sein, dass der Wertebereich der im Kontext referenzierten Variablen vollständig und überschneidungsfrei abgedeckt ist.
- *Wechselwirkungen zwischen Mustertypen*
Die Gliederung in verschiedene Mustertypen bewirkt eine funktionale Kapselung. Jeder Mustertyp leistet einen spezifischen Beitrag zur Generierung und Adaptation der ABS, der sich klar vom Beitrag der anderen Mustertypen unterscheidet (vgl. Abschnitt 4.2). Die möglichen Wechselwirkungen zwischen den Mustertypen ergeben sich aufgrund der Relationen `benötigt`, `setzt <Variable> wie benötigt von` und `setzt <Variable> wie genutzt von` und deren Umkehrrelationen. Diese Relationen beschreiben das Zusammenspiel der verschiedenen Mustertypen, die im dreistufigen Prozess der ABS-Generierung und Adaptation nacheinander in Aktion treten. Dabei dient das Ergebnis eines Musters des einen Typs häufig als Ausgangspunkt für die Arbeit des nächsten Musters eines anderen Typs. Die Variablen der Profile DP und UIP können dabei als Schnittstellen für den Datenaustausch zwischen verschiedenen Mustertypen betrachtet werden.

Die Vorgehensweise bei den folgenden wichtigsten Erweiterungsszenarien veranschaulicht den Umgang mit möglichen Wechselwirkungen. Dabei wird der generische Ablauf für Erweiterungen (Abbildung 6) teilweise mehrmals in verschiedenen Mustertypen durchlaufen:

- *Erweiterungen bei Abdeckung eines neuen Geräts*
Soll die Nutzung eines zusätzlichen Geräts unterstützt werden, werden die folgenden Schritte durchlaufen:
 1. Wenn bereits ein ähnliches Gerät unterstützt wird, können geringfügige Ergänzungen in den If-Statements bestimmter gerätespezifischer Muster ausreichen, um das neue Gerät mit bereits bestehenden Lösungen abzudecken.

Weitere Schritte sind dann nicht mehr erforderlich. Andernfalls – z.B. wenn es sich um einen neuen Gerätetyp handelt - müssen neue gerätespezifische Muster ($d_i \in D$) erstellt werden und möglicherweise muss das Geräteprofil DP um neue Variable erweitert werden. Diese neuen Muster bzw. Variablen repräsentieren die mit dem neuen Gerät verbundenen Ein- und Ausgabekapazitäten. In diesem Fall müssen die weiteren folgenden Schritte durchlaufen werden.

2. Zum Abgleich der Möglichkeiten der neuen technischen Ausstattung mit den Eigenschaften des Nutzers und der Umgebung können Veränderungen der Individualisierungsmuster / erforderlich werden. Wenn bereits bestehende Individualisierungsmuster zufriedenstellende Lösungen für die neuen Bedingungen bieten, genügt es, die If-Statements dieser Individualisierungsmuster entsprechend zu erweitern. Andernfalls werden neue Muster erstellt. Wenn einer bestehenden UIP-Variablen ein neuer Wert zugeordnet werden soll, genügt ein neues Muster in einem bestehenden Bündel. Soll eine neue UIP-Variable eingeführt werden, wird ein neues Musterbündel in / angelegt. Im letzteren Fall bleiben alle bestehenden Individualisierungsmuster von den Erweiterungen unberührt.
 3. Um eine neue UIP-Variable angemessen zu berücksichtigen, sind in der Regel auch Arbeiten in der Menge K der Interaktionsmuster notwendig. Im einfachsten Fall genügt hierfür eine Anpassung der If-Statements der bestehenden Interaktionsmuster. Die If-Statements bestehender Interaktionsmuster müssen auch angepasst werden, wenn durch neu zu schaffende Bedien- und Anzeigekomponenten die neuen Ein- und Ausgabemöglichkeiten voll ausgeschöpft werden sollen. Ein Beispiel hierfür wäre eine neue Komponente zur sprachgestützten Texteingabe nachdem das Anschließen eines Mikrofons ermöglicht wurde. Ebenso können dann neue Bedien- und Anzeigeelemente erforderlich werden, die hierfür die entsprechenden generischen Bausteine und Mechanismen liefern (z.B. freie Spracherkennung oder ein Schieberegler zur Anpassung der Textausgabegeschwindigkeit).
- *Erweiterungen bei Abdeckung zusätzlicher Nutzer- und Umgebungsbedingungen*
Wenn das adaptive Verhalten der ABS auf zusätzliche Eigenschaften des Nutzers oder der Umgebung erweitert werden soll, ergibt sich folgender Ablauf:
1. Das Nutzer- und Umgebungsprofil UP wird um eine oder mehrere Variablen erweitert, um das neu abzudeckende Wissen über den Nutzer und seine Umgebung entsprechend zu repräsentieren.
 2. Nun ist zu prüfen, ob sich die beabsichtigten Anpassungen der ABS an das neue Wissen über Nutzer und Umgebung bereits über das bestehende Benutzungsschnittstellenprofil UIP durchführen lassen. Wenn das nicht der Fall ist, werden neue UIP-Variablen erstellt. Jede neue UIP-Variable erfordert in der Regel ein neues Bündel von Individualisierungsmustern, die dafür sorgen, dass die neuen ABS-Merkmale (UIP-Variablen) so gesetzt werden, dass das zusätzliche Wissen über Nutzer und Umgebung möglichst optimal genutzt wird.
 3. Ein erweitertes UIP zieht auch Änderungsbedarf in der Menge K der Interaktionsmuster nach sich. Wenn neue Interaktionskomponenten geschaffen werden sollen, die gezielt bestimmte Besonderheiten in den neu abgedeckten

Nutzer- und Umgebungsbedingungen adressieren, werden neue Interaktionsmuster in bestehenden Musterbündeln angelegt. Andernfalls wird geprüft, in welchen Bündeln von Interaktionsmustern die neuen UIP-Variablen zur Auswahl des am besten geeigneten Musters herangezogen werden sollen. In beiden Fällen wird eine Überarbeitung der If-Statements in den betroffenen Bündeln notwendig. Dabei ist insbesondere auf die vollständige und überschneidungsfreie Abdeckung des Wertebereichs der Kontextvariablen zu achten.

– *Erweiterungen bei Abdeckung neuer Interaktionssituationen*

Wenn die Mustersammlung eine neue Interaktionssituation abdecken soll, wird im Wesentlichen nur ein neues Bündel von Interaktionsmustern erforderlich:

1. Ein neues Bündel von Interaktionsmustern, die die neue Interaktionssituation in ihrem Problem-Argument referenzieren, wird angelegt.
2. Möglicherweise erfordern die neuen Interaktionsmuster auch neue Bedien- und Anzeigeelemente, die neu erstellt werden müssen. Andernfalls ergeben sich keine weiteren Konsequenzen für bereits bestehende Muster oder andere Mustertypen.

– *Erweiterungen zur Verfeinerung des Entwurfsmusterrepositoriums*

Bei der Ausdifferenzierung der Gestaltungslösungen kann der Wunsch nach neuen Lösungsvarianten aufkommen. Im Wesentlichen können zwei Fälle unterschieden werden:

- Wenn bestehende Bündel von gerätespezifischen Mustern, Individualisierungsmustern, Interaktionsmustern oder Adaptationsdialogmustern durch neue Lösungen erweitert werden, ist lediglich auf die vollständige und überschneidungsfreie Abdeckung des Wertebereichs der Kontextvariablen zu achten.
- Sollen neue Bedien- und Anzeigeelemente genutzt werden, müssen diese entsprechend in den Lösungsbeschreibungen der Interaktionsmuster, die die neuen Elemente nutzen sollen, aufgerufen werden.

4.4.3 Öffentlich zugängliches Repository

Ein Entwurfsmusterrepositorium für barrierefreie ABS erfordert umfangreiches und vielfältiges Fachwissen über mögliche Einschränkungen und effektive Gestaltungsmechanismen. Neben der Erweiterbarkeit ist daher die öffentliche Zugänglichkeit eine wichtige Eigenschaft des MyUI-Entwurfsmusterrepositoriums. Der Lösungsansatz sieht vor, dass jede Person, die sich für die Regeln und Mechanismen der ABS-Anpassungen interessiert, die dafür verantwortlichen Entwurfsmuster einsehen und kommentieren kann, sowie Vorschläge für Veränderungen der bestehenden Muster oder für neue Entwurfsmuster beitragen kann. So soll eine möglichst große Gruppe von Fachexperten in die Verfeinerung und Erweiterung des Repositoriums einbezogen werden.

Für die praktische Umsetzung des Entwurfsmusterrepositoriums bietet sich eine internetbasierte Kollaborationsplattform an²⁰. Auch wenn die Details der Gestaltung und Implementierung dieser Plattform nicht im Fokus dieser Arbeit stehen, lassen sich dennoch Anforderungen an eine geeignete technische Lösung benennen:

- Der Zugang zum Repository wird online über einen Internetbrowser ermöglicht. Die Nutzung des Repositorys stellt keine besonderen Anforderungen an die Computerausstattung und erfordert keine Installation weiterer Programme.
- Neben dem lesenden Zugriff besteht für die Nutzer des Repositorys die Möglichkeit, neue Muster anzulegen, bestehende Muster zu verändern und Kommentare zu bestehenden Mustern zu hinterlassen.
- Um möglichen Schaden durch Missbrauch von Schreibrechten zu minimieren, bietet die technische Plattform eine Nutzerverwaltung, die nur registrierten und mit Authentifizierung eingeloggten Nutzern schreibenden Zugriff ermöglicht.
- Des Weiteren bietet die Plattform einen History-Mechanismus, der alle Änderungen des Repositorys nachvollziehbar und reversibel macht.
- Die Darstellung und Bearbeitung der Entwurfsmuster erfolgt in dem oben skizzierten Format (vgl. Abschnitt 4.4.1). Dies erfordert das Einbinden von grafischen Darstellungen und idealerweise auch von Animationen oder kurzen Videosequenzen im Abschnitt „Diagramm“ der Musterbeschreibung.
- Für das Anlegen neuer Muster werden Vorlagen zur Verfügung gestellt, die den Autor sowohl unterstützen als auch zwingen, Muster im standardisierten Format zu erstellen.
- Die technische Plattform des Repositorys bietet Hypermedia- (oder zumindest Hypertext-) Funktionalität, damit die Beziehungen zwischen Mustern, Variablen und Profilen durch Verlinkungen gut verdeutlicht werden können.

²⁰ Die erste Version des MyUI-Entwurfsmusterrepositoriums wird als MediaWiki gepflegt und ist unter <http://myui-patterns.iao.fraunhofer.de> öffentlich zugänglich.

5 Parametrierung der adaptiven Benutzungsschnittstelle

Die Parametrierung der ABS ist der erste Prozessschritt der ABS-Generierung und Adaptation. Das Ergebnis dieses ersten Schrittes ist das Benutzungsschnittstellenprofil (UIP: User Interface Profile). Die Variablenwerte des UIP definieren globale Eigenschaften der ABS und gelten für die gesamte Benutzungsschnittstelle in allen Interaktionssituationen der Applikation. Das UIP wird so parametriert, dass eine möglichst gute Anpassung der ABS an das aktuelle Wissen über die technische Ausstattung, den Nutzer und seine Umgebung erreicht wird. Zusätzlich können Maßgaben des gewünschten Erscheinungsbildes einer Marke, eines Projekts oder einer Firma berücksichtigt und im UIP abgebildet werden.

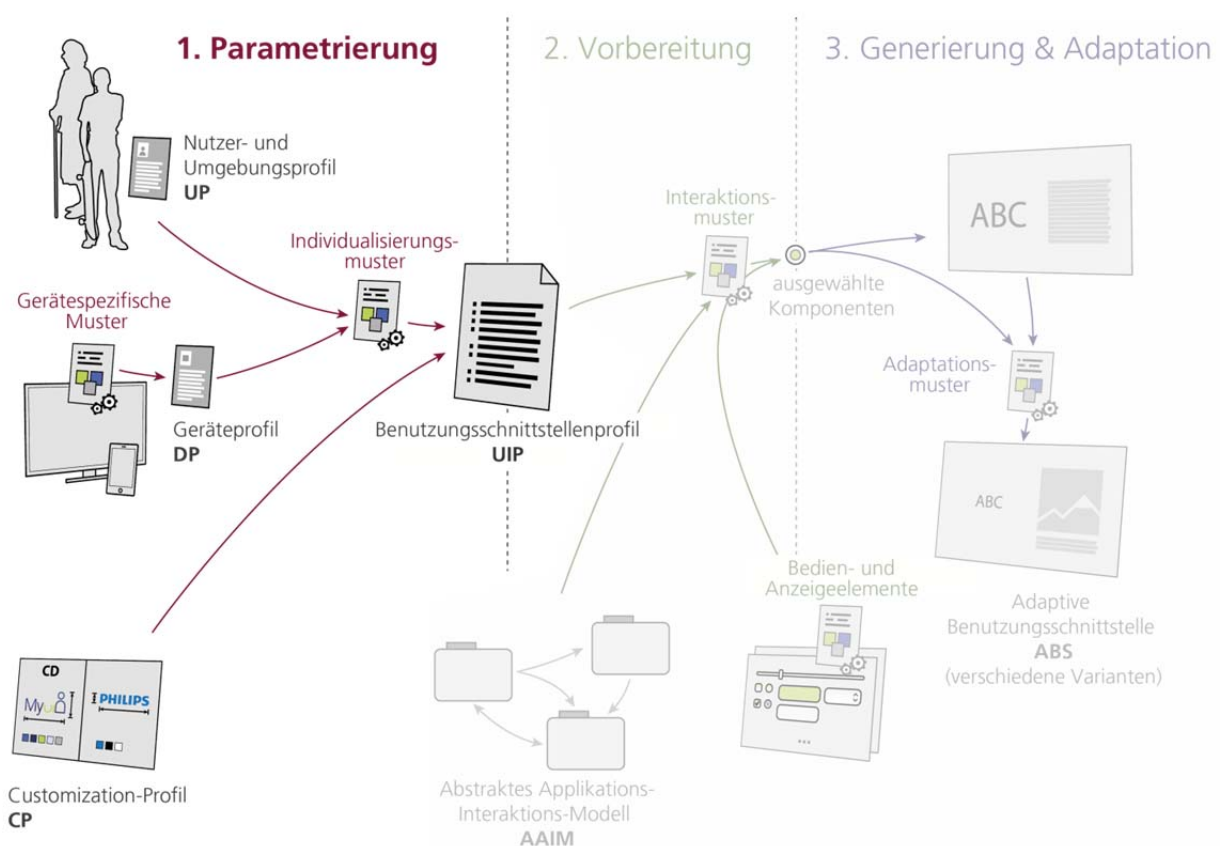


Abbildung 7 Profile und Entwurfsmuster im Prozess der ABS-Parametrierung

Die wesentlichen Konzepte der ABS-Parametrierung umfassen die folgenden Profile und Entwurfsmustertypen (vgl. Abbildung 7):

- Der *Gerätmanager* (nicht abgebildet) verwaltet und steuert die Erstellung und Aktualisierung des Geräteprofils.
- Das *Geräteprofil* (DP: Device Profile) stellt Informationen über die aktuell verfügbaren und genutzten Ein- und Ausgabegeräte bereit. Das DP stellt ein Zwischenergebnis der ABS-Parametrierung dar. Es ist Output der von den gerätespezifischen Mustern

durchgeführten Transformationen und dient als Input für die Arbeit der Individualisierungsmuster.

- *Gerätespezifische Muster* erzeugen und aktualisieren das DP auf der Grundlage der im Gerätemanager verfügbaren generischen Informationen über die aktuellen und genutzten Ein- und Ausgabegeräte.
- Das *Nutzer- und Umgebungsprofil* (UP: User Profile) stellt Informationen über Eigenschaften des Nutzers und der Nutzungsumgebung bereit. Das UP ist das Produkt der in dieser Arbeit nicht näher behandelten MyUI-Kontextmanagementinfrastruktur (vgl. Hernández et al., 2011; Strnad et al., 2012a) und dient als eine wesentliche Eingangsgröße für die Arbeit der Individualisierungsmuster.
- Das *Benutzungsschnittstellenprofil* (UIP: User Interface Profile) ist das zentrale Ergebnis der ABS-Parametrierung. Zu großen Teilen ist es das Ergebnis der Individualisierungsmuster. Einige Variablen des UIP werden direkt vom Customization-Profil übernommen.
- *Individualisierungsmuster* erzeugen und aktualisieren das UIP. Dabei dienen das DP und das UP als Eingangsgrößen.
- Das *Customization-Profil* (CP: Customization Profile) beinhaltet Variablen, die der ABS eine firmen-, projekt- oder markenspezifische Erscheinung verleihen. Das CP kann direkt vom Entwickler einer MyUI-Applikation erstellt und verändert werden.

5.1 Gerätemanager

Der Gerätemanager erfüllt zwei Aufgaben (s. Abbildung 8):

1. Zunächst werden die verfügbaren Ein- und Ausgabegeräte registriert. Der Gerätemanager verfügt über ein Geräterepositorium, das Spezifikationen zu allen von MyUI unterstützten Geräten bereithält. Zuerst empfängt der Gerätemanager die direkt vom Gerät zur Verfügung gestellten Informationen wie z.B. Typenbezeichnung, Bildschirmauflösung, etc. Dann werden zusätzlich die detaillierteren Informationen und Spezifikationen aus dem Geräterepositorium abgerufen.
2. In einem zweiten Schritt werden die erfassten Geräteinformationen in ein DP transformiert. Dazu ruft der Gerätemanager gerätespezifische Muster auf, die die verfügbaren Geräte kategorisieren und die Variablen des DP entsprechend setzen. Eine detaillierte Beschreibung des DP und seiner Variablen findet sich im nächsten Abschnitt.

Ähnlich zur MyUI-Kontextmanagementinfrastruktur (vgl. Hernández et al., 2011; Strnad et al., 2012a) arbeitet der Gerätemanager in einem kontinuierlichen Prozess. So werden die beiden Prozessschritte des Gerätemangers sowohl zu Beginn einer neuen Interaktionssequenz als auch bei der Detektion einer Änderung in der technischen Ausstattung ausgeführt. Auf diese Weise kann das DP permanent aktualisiert werden, um beispielsweise auf den Austausch oder die Ergänzung eines Interaktionsgeräts oder das Drehen des Bildschirms von hochkant zu quer entsprechend reagieren zu können und über eine erneute ABS Parametrierung eine Aktualisierung des UIP zu erreichen.

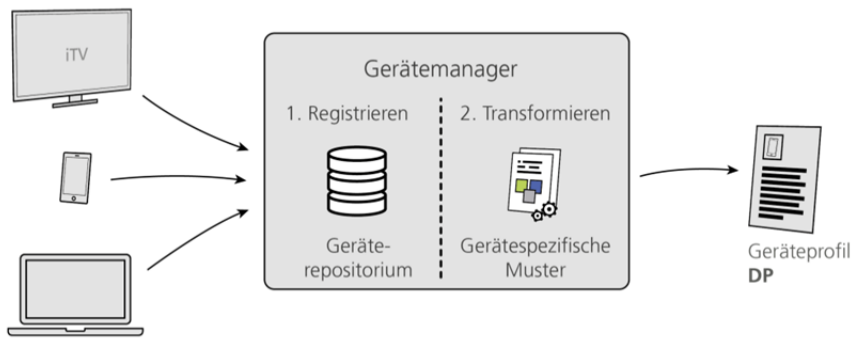


Abbildung 8 Der Gerätemanager steuert die Erstellung und Aktualisierung des Geräteprofils

5.2 Geräteprofil (DP: Device Profile)

Das Geräteprofil (DP: Device Profile) spezifiziert wesentliche Eigenschaften der genutzten und verfügbaren Geräte in einem standardisierten Format. Die Einstellungen des DP sind globale Variablen, die über die gesamte Applikation hinweg in allen Interaktionssituationen und unabhängig vom jeweiligen Nutzer gelten. Die Erstellung und Aktualisierung des DP wird vom Gerätemanager gesteuert und von den gerätespezifischen Entwurfsmustern ausgeführt. Das DP enthält die folgenden Variablen:

- Kategorie
Die Kategorie dient einer groben Klassifizierung der Gesamtkonfiguration. Sie kann Werte wie „Smart Phone“, „Touch Panel / Tablet“, „iTV“, „Desktop Computer“, etc. annehmen. In der Regel spezifiziert sie den Typ des Geräts, auf dem die MyUI-Applikation läuft. Durch die Kategorie sind grundlegende Interaktions- und Kontextparameter wie zum Beispiel die Entfernung zum Bildschirm oder die Bildschirmausrichtung festgelegt.
- Variablen zur Schriftgröße
Je nach Gerätekategorie werden mehrere Schriftgrößen (klein, mittel, groß und sehr groß) für die Schrifttypen Titelschrift, Textkörperschrift und Ergänzungstextschrift definiert, die für die aktuelle technische Ausstattung in Frage kommen. Die Entscheidung, welche der Schriftgrößen schließlich ins UIP übernommen und in der ABS angewendet wird, wird von den Individualisierungsmustern aufgrund der Werte im UP getroffen.
- Variablen zum Layout
In Abhängigkeit der aktuellen Darstellungsmöglichkeiten werden im DP bestimmte Layout-Variablen gesetzt. Diese umfassen beispielsweise Variablen zur Definition der Größe und Anzahl von Layoutrasterzellen (*cellSize*, *cellCountX* und *cellCountY*), der Bildschirmausrichtung und der Koordinaten der einheitlich verwendeten Bildschirmbereiche (z.B. Titelzeile, Inhaltsbereich, Funktionsbereich, etc.).
- Variablen zu Eingabegeräten
Diese Variablen geben darüber Auskunft, welche Eingabegeräte verfügbar sind. Je nach den Einstellungen in diesen Variablen können dem Nutzer unterschiedliche Eingabemechanismen angeboten werden. Wenn beispielsweise die Variablen *microphoneAvailable* und *keyboardAvailable* aktiviert sind, kann das System Eingaben über Sprache, alphanumerische Tasten und Pfeiltasten anbieten.

- Variablen zu Ausgabegeräten
Diese Gruppe definiert die verfügbaren Ausgabegeräte. Aktuell werden hier zwei Variablen geführt, die angeben, ob ein grafisches Display und ein Lautsprechersystem verfügbar sind. Sowohl hier als auch in der Gruppe der Variablen zu den Eingabegeräten können weitere Variablen hinzugefügt werden, um speziellere Hilfsgeräte (Assistive Technologies) wie zum Beispiel Braille-Displays oder Apparaturen für Blicksteuerung zu unterstützen.

Tabelle 2 zeigt ein Beispiel eines MyUI-Geräteprofils für ein interaktives Fernsehgerät:

| DP Variable | Wert | Beschreibung |
|------------------------------------|---|---|
| <i>deviceCategory</i> | iTV | Klasse des Geräts, auf dem die Applikation läuft |
| Variablen zur Schriftgröße | | |
| <i>titleFontSizes</i> | {small: 36 pt; medium: 55 pt; large: 83 pt; xlarge: 110 pt} | Mögliche Schriftgrößen für Titelemente |
| <i>bodyTextFontSizes</i> | {small: 24 pt; medium: 34 pt; large: 51 pt; xlarge: 68 pt} | Mögliche Schriftgrößen für Textkörperelemente |
| <i>complementaryTextFontSizes</i> | {small: 18 pt; medium: 21 pt; large: 32 pt; xlarge: 42 pt} | Mögliche Schriftgrößen für ergänzende Textelemente, z.B. zusätzliche Erläuterungen oder weniger wichtige Kommentare |
| Variablen zum Layout | | |
| <i>cellSize</i> | 5 px | Anzahl der Pixel für Höhe und Breite einer Rasterzelle |
| <i>cellCountX</i> | 128 | Anzahl der Rasterzellen in horizontaler Richtung auf dem gesamten Bildschirm |
| <i>cellCountY</i> | 96 | Anzahl der Rasterzellen in vertikaler Richtung auf dem gesamten Bildschirm |
| <i>interactiveElementMinSizes</i> | {small: 9; medium: 14; large: 22; xlarge: 30} | Mögliche Mindestgrößen (in Rasterzellen) für interaktive Bildelemente |
| Variablen zu Eingabegeräten | | |
| <i>microphoneAvailable</i> | true-external | Verfügbarkeit und Typ eines Mikrofons |
| <i>touchAvailable</i> | false | Verfügbarkeit und Typ eines Touchdisplays |
| <i>pointingDeviceAvailable</i> | false | Verfügbarkeit und Typ eines Zeigegeräts |
| <i>keyboardAvailable</i> | true-QWERTY | Verfügbarkeit und Typ einer Tastatur |
| <i>remoteControlAvailable</i> | true-standard | Verfügbarkeit und Typ einer Fernbedienung |
| <i>cameraAvailable</i> | true | Verfügbarkeit einer Kamera |
| Variablen zu Ausgabegeräten | | |
| <i>displayAvailable</i> | {available: true; size: 42"; resolution: 1280:720} | Verfügbarkeit, Größe und Auflösung eines Bildschirms |
| <i>speakerAvailable</i> | true | Verfügbarkeit eines Lautsprechersystems |

Tabelle 2 Beispiel eines Geräteprofils (DP) für die Konfiguration eines interaktiven Fernsehgeräts

5.3 Gerätespezifische Muster (Device-Specific Patterns)

Gerätespezifische Entwurfsmuster erstellen und aktualisieren das DP. Sie transformieren einfache Gerätedaten in Variablen des DP. Der Input dieser Transformation wird vom Gerätemanager zur Verfügung gestellt und kommt einerseits direkt von den angeschlossenen Geräten, die von sich aus grundlegende Informationen über sich bekannt geben. Andererseits stammen die Gerätedaten aus dem Gerätepositorium (vgl. Abschnitt 5.1). Gerätespezifische Muster werden zu Beginn einer neuen Interaktionssequenz aufgerufen und wenn der Gerätemanager während einer Interaktionssequenz eine Änderung der technischen Ausstattung erkennt.

Gerätespezifische Muster können die folgenden Relationen zu anderen Mustern besitzen (vgl. Abschnitt 4.3):

- `substituiert`
Diese Relation fasst zwei oder mehrere gerätespezifische Muster zu Bündeln zusammen.
- `setzt <Variable> wie benötigt von`
Diese Relation unterstützt den zweistufigen Prozess der ABS-Parametrierung, indem sie Bündel von gerätespezifischen Mustern mit Bündel von Individualisierungsmustern verbindet. Die Erstellung und Aktualisierung des DP ist dabei ein Zwischenschritt auf dem Weg zum UIP. Einige gerätespezifische Muster legen im DP zunächst mehrere Variablen an, die die technischen Gegebenheiten auf unterschiedliche Weise gut unterstützen. Im zweiten Schritt greifen Individualisierungsmuster dann auf diese Variablen des DP zurück: Je nach Einstellungen des UP werden diejenigen DP-Variablen in den Lösungsfeldern der aktivierten Individualisierungsmuster referenziert, die dem aktuellen Nutzer die bestmögliche Unterstützung gewährleisten (vgl. Abschnitt 4.3.4).
- `setzt <Variable> wie genutzt von`
Diese Relation kommt bei den gerätespezifischen Mustern in zwei Ausprägungen zum Einsatz. Einerseits bildet sie Abhängigkeiten zwischen Variablen des DP ab. Einige Bündel gerätespezifischer Muster referenzieren bei der Auswahl des geeigneten Musters auf andere DP-Variablen. Beispiele von DP-Variablen, die selbst wieder als Inputvariablen für andere gerätespezifische Muster dienen, umfassen die Geräteklasse (*deviceCategory*) und die Bildschirmauflösung als Bestandteil der mehrstelligen Variable *displayAvailable* (vgl. Abschnitt 4.3.6). Andererseits setzen einige gerätespezifische Muster Werte im DP, die zur Auswahl geeigneter Individualisierungsmuster herangezogen werden (ein Beispiel ist im Abschnitt 4.3.6 beschrieben).
- `nutzt <Variable> wie gesetzt von`
Für den oben zuerst beschriebenen Fall, dass die Relation `setzt <Variable> wie genutzt von` zwei Bündel von gerätespezifischen Mustern verbindet, ist auch diese Relation als deren Umkehrrelation entsprechend definiert.

Um Redundanzen in den Beschreibungen der gerätespezifischen Muster eines Bündels zu vermeiden, sieht das MyUI-Entwurfsmusterrepositorium eine Beschreibung der Gemeinsamkeiten aller Muster eines Bündels in einem definierten Format vor (s. Tabelle 3). Um eine möglichst gute internationale Zugänglichkeit des Musterrepositoriums zu ermöglichen, sind alle Musterbeschreibungen und Vorlagen in englischer Sprache angelegt. Die Instruktionen und Erläuterungen zum Ausfüllen der Vorlagen sind in dieser Arbeit in Deutsch gehalten. Des Weiteren wird zur formalen Beschreibung der Syntax für einige Felder der Vorlagen die von der

Object Management Group (OMG, 2011) verwendete Variante der Backus-Naur-Form (BNF) benutzt. Um eine bessere Lesbarkeit zu gewährleisten und die Notationskonventionen dieser Arbeit nicht zu brechen, werden nur Variablennamen in kursiv gesetzt.

| Device-specific pattern bundle | <Name des gerätespezifischen Musterbündels> |
|--------------------------------|---|
| Problem | <Beschreibung der DP-Variable(n) und ihres Zweckes, die durch die Muster des Bündels gesetzt werden soll(en), um unterschiedliche Eigenschaften verschiedener Geräte durch unterschiedliche DP-Parameter in einem normierten Format auszudrücken> |
| Sets variables | <Auflistung der von den Mustern des Bündels gesetzten DP-Variablen> |
| Patterns | <Auflistung der Entwurfsmuster des Bündels> |

Tabelle 3 Vorlage für die Beschreibung eines Bündels von gerätespezifischen Mustern

Die Standardvorlage für die Beschreibung eines gerätespezifischen Musters hat die folgenden Felder (s. Tabelle 4):

| Device-specific pattern | <Name des gerätespezifischen Musters> |
|-------------------------------|---|
| Problem | <Beschreibung des von diesem Muster adressierten Gestaltungsproblems, das darin besteht, (eine) bestimmte DP-Variable(n) so zu setzen, dass die hierfür relevanten Besonderheiten eines bestimmten Geräts oder mehrerer ähnlicher Geräte in einem normierten Format ausgedrückt werden> |
| Context | 'IF' <Prüfung gegen bestimmte Werte der Gerätespezifikation> ['AND' 'OR' <DP-Variable> = <Wert>] * |
| Solution | 'THEN' 'Set' <DP-Variable> '=' <Wert(e)> ['set' <DP-Variable> '=' <Wert(e)>] * |
| Code Reference | <Soweit möglich werden Teile der Lösung in einer programmierbaren Form angegeben. Hierfür kann z.B. ein technologieunabhängiger Pseudocode verwendet werden.> |
| Diagram | <Soweit möglich wird die Gestaltungslösung in einer schematischen Darstellung veranschaulicht.> |
| Rationale (references) | <Erläuterung bzw. Begründung der Lösung; falls vorhanden können Querverweise zu entsprechender Fachliteratur oder Normen und Standards angegeben werden.> |
| Ranking | <Bewertung der Reife des Musters aufgrund des Grads der Validierung auf einer dreistufigen Skala.> ::= '***' '**' '*' |
| Pattern Bundle | <Bündel, dem dieses Muster angehört> |
| Substitutes | <Auflistung der anderen Muster des Bündels, d.h. der Muster, die in Relation substituiert mit dem beschriebenen Muster stehen> |
| Uses <variable> | <Auflistung der DP-Variablen, die im Kontextfeld des beschriebenen Musters referenziert werden> |

Tabelle 4 Vorlage für die Beschreibung eines gerätespezifischen Musters

5.4 Nutzer- und Umgebungsprofil (UP: User Profile)

Das Nutzer- und Umgebungsprofil (UP: User Profile) speichert Informationen über relevante Nutzereigenschaften und Umgebungsbedingungen. Die Mechanismen zur Erstellung und Aktualisierung des UP sind Funktionen der MyUI-Kontextmanagementinfrastruktur. Ebenso wie die Konstruktion des UP sind diese Mechanismen nicht Gegenstand dieser Arbeit. Sie werden bei Hernández et al. (2011) näher beschrieben. Für die Auswahl der UP-Variablen gilt der Grundsatz, dass nur diejenigen Nutzer- und Umgebungsbedingungen berücksichtigt werden, die einen potenziellen Einfluss auf Nutzungsbarrieren haben. Die meisten nutzerbezogenen Variablen basieren auf dem ICF-Klassifizierungsschema der WHO²¹. Die umgebungsbezogenen Variablen sind momentan auf Beleuchtungsbedingungen (*ambientLight*) und Umgebungslärm (*ambientNoise*) beschränkt. Die MyUI-Kontextmanagementinfrastruktur ist auf eine leichte Erweiterbarkeit des UP ausgelegt (vgl. Hernández et al., 2011). Auch die in dieser Arbeit fokussierten Adaptationsmechanismen sind gut erweiterbar. Werden dem UP neue Variablen hinzugefügt, so ist der in Abschnitt 4.4.2 beschriebene Prozess zu durchlaufen.

Alle Variablen des UP, die das Ausmaß einer graduell ausdrückbaren, interaktionsrelevanten Nutzerfähigkeit oder Umgebungsbedingung beschreiben, arbeiten einheitlich mit einem Wertebereich der reellen Zahlen zwischen 0 und 4 einschließlich beider Grenzen. Dabei steht der Wert 0 für optimale Bedingungen ohne Einschränkung. Der Wert 4 steht für maximale Einschränkungen bzw. geringstmögliche Fähigkeiten. Diese Skalierung ist am Beurteilungsvorschlag der WHO²² (DIMDI, 2004) angelehnt. Darüber hinaus zielt diese Vereinfachung darauf ab, ethische Probleme der Datensicherheit zu minimieren und die Vielzahl der ansonsten üblichen medizinischen Maßeinheiten im Sinne der Auswirkung auf die Interaktion zu normieren (vgl. Hernández et al., 2011). Die Ausgangsversion des MyUI-UP ist in Tabelle 5 dargestellt:

| UP Variable | Wertebereich | Beschreibung |
|---|--------------|---|
| Variablen zu perzeptuellen Einschränkungen | | |
| <i>visualAcuity</i> | [0,4] | Einschränkungen der visuellen Wahrnehmung |
| <i>fieldOfVision</i> | [0,4] | Einschränkungen des Sehfelds des Nutzers |
| <i>ambientLight</i> | [0,4] | Einschränkungen durch zu helle Lichtbedingungen |
| <i>ambientNoise</i> | [0,4] | Einschränkungen durch Umgebungslärm |
| <i>hearing</i> | [0,4] | Einschränkungen der Wahrnehmung akustischer Systemausgaben |
| Variablen zu kognitiven Einschränkungen | | |
| <i>languageReception</i> | [0,4] | Einschränkungen des Verständnisses geschriebener oder gesprochener Sprache |
| <i>languageProduction</i> | [0,4] | Einschränkungen der Nutzerfähigkeit, geschriebene oder gesprochene Sprache zu produzieren |

²¹ <http://www.who.int/classifications/icf/en/>

²² <http://www.who.int/classifications/icf/training/icfchecklist.pdf>

| | | |
|---|----------------------------|---|
| <i>understandingAbstractSigns</i> | [0,4] | Einschränkungen des Verständnisses abstrakter Zeichen und Piktogramme |
| <i>attention</i> | [0,4] | Einschränkungen der Nutzerfähigkeit, gleichzeitig mehreren Dingen Aufmerksamkeit zu widmen bzw. sich auf eine Sache voll zu konzentrieren |
| <i>processingSpeed</i> | [0,4] | Einschränkungen der Geschwindigkeit, mit der der Nutzer Informationen verarbeiten kann |
| <i>workingMemory</i> | [0,4] | Einschränkungen der Nutzerfähigkeit, eine genaue Abfolge von Prozessschritten zu erinnern und sich in diesem Prozess zu orientieren |
| <i>longTermMemory</i> | [0,4] | Einschränkungen der Nutzerfähigkeit, Informationen zu lernen und für eine lange Zeit zu erinnern |
| <i>ICTLiteracy</i> | [0,4] | Einschränkungen des Nutzers aufgrund geringer Erfahrung mit moderner Informationstechnologie |
| <i>handEyeCoordination</i> | [0,4] | Einschränkungen der Nutzerfähigkeit, Handbewegungen mit Gesehenem zu koordinieren |
| Variablen zu motorischen Einschränkungen | | |
| <i>speechArticulation</i> | [0,4] | Einschränkungen des Nutzers bei der Artikulation von Sprache |
| <i>fingerPrecision</i> | [0,4] | Einschränkungen des Nutzers bei der präzisen Bewegung der Finger |
| <i>handPrecision</i> | [0,4] | Einschränkungen des Nutzers bei der präzisen Bewegung der Hand |
| <i>armPrecision</i> | [0,4] | Einschränkungen des Nutzers bei der präzisen Bewegung der Arme |
| <i>contactGrip</i> | [0,4] | Einschränkungen des Nutzers, einen Kontaktgriff auszuführen |
| <i>pinchGrip</i> | [0,4] | Einschränkungen des Nutzers, einen Zufassungsgriff auszuführen |
| <i>clenchGrip</i> | [0,4] | Einschränkungen des Nutzers, einen Umfassungsgriff auszuführen |
| Allgemeine Variablen | | |
| <i>name</i> | | Name, ID oder Kurzname des Nutzers |
| <i>emailAddress (optional)</i> | | E-Mail-Adresse des Nutzers |
| <i>preferredLanguage</i> | {English, German, Spanish} | Vom Nutzer präferierte Sprache für Benutzungsschnittstellen |
| <i>successfulInteractions</i> | N | Anzahl erfolgreicher Interaktionen mit dem System |
| <i>stateTransitions</i> | N | Anzahl von Zustandsübergängen, die der Nutzer mit dem MyUI-System ausgeführt hat |
| <i>MyUIExperience</i> | [0, 4] | Grad der Erfahrung des Nutzers mit dem MyUI-System; diese Variable dient als Indikator dafür, wie gut der Nutzer mit den Adaptationsmechanismen vertraut ist und wie lange sich das MyUI-UP bereits auf den Nutzer „einstellen“ konnte. |

| | | |
|-------------------------------|---|---|
| <i>preferenceInputDevice</i> | {microphone; touch; pointingDevice; keyboard; remoteControl; camera} | Geordnete Liste der vom Nutzer präferierten Eingabegeräte |
| <i>preferenceTonalOutput</i> | {True; False} | Die Variable gibt an, ob der Nutzer nichtsprachliche akustische Ausgaben als Ergänzung zur grafischen Anzeige präferiert. |
| <i>preferenceSpeechOutput</i> | {True; False} | Die Variable gibt an, ob der Nutzer Sprachausgaben als Ergänzung zur grafischen Textanzeige präferiert. |

Tabelle 5 Variablen des Nutzer- und Umgebungsprofils UP

5.5 Benutzungsschnittstellenprofil (User Interface Profile)

Das Benutzungsschnittstellenprofil (UIP: User Interface Profile) ist das zentrale Ergebnis der ABS-Parametrierung. Während das DP und das UP Eigenschaften des Nutzungskontexts repräsentieren, repräsentiert das UIP Eigenschaften der ABS. Das UIP besteht aus einer Menge von globalen Variablen, die das Aussehen und die Funktionsweisen der gesamten Benutzungsschnittstelle bestimmen. Durch dieses zentral verwaltete UIP kann eine optimale Anpassung an die Anforderungen des Nutzungskontexts und eine maximale Konsistenz innerhalb einzelner Anwendungen und über verschiedene Anwendungen hinweg gewährleistet werden. So wirken Adaptationen, die aus neu verfügbaren Informationen über den Nutzer oder die Umgebung resultieren, nicht nur punktuell auf den aktuellen Dialog oder einzelne Bedienelemente, sondern auf die gesamte Benutzungsschnittstelle des MyUI-Systems. Auch das UIP ist erweiterbar. Die aktuellen Variablen des UIP sind in Tabelle 6 aufgeführt.

| UIP Variable | Wertebereich | Beschreibung |
|---------------------------|---|---|
| <i>titleBarArea</i> | $(x; y; b; h; visible)$ mit $x, b \in \mathbb{N} \mid 0 \leq x, b \leq cellCountX,$ $y, h \in \mathbb{N} \mid 0 \leq y, h \leq cellCountY$ und $visible \in \{true; false\}$ | Ausgangskordinaten, Breite und Höhe sowie Anzeigeeoptionen der Titelzeile |
| <i>contentArea</i> | $(x; y; b; h; visible)$ mit $x, b \in \mathbb{N} \mid 0 \leq x, b \leq cellCountX,$ $y, h \in \mathbb{N} \mid 0 \leq y, h \leq cellCountY$ und $visible \in \{true; false\}$ | Ausgangskordinaten, Breite und Höhe sowie Anzeigeeoptionen des Inhaltsbereichs |
| <i>contentControlArea</i> | $(x; y; b; h; visible)$ mit $x, b \in \mathbb{N} \mid 0 \leq x, b \leq cellCountX,$ $y, h \in \mathbb{N} \mid 0 \leq y, h \leq cellCountY$ und $visible \in \{true; false\}$ | Ausgangskordinaten, Breite und Höhe sowie Anzeigeeoptionen des Inhaltsteuerungsbereichs |
| <i>functionsArea</i> | $(x; y; b; h; visible)$ mit $x, b \in \mathbb{N} \mid 0 \leq x, b \leq cellCountX,$ $y, h \in \mathbb{N} \mid 0 \leq y, h \leq cellCountY$ und $visible \in \{true; false\}$ | Ausgangskordinaten, Breite und Höhe sowie Anzeigeeoptionen des Funktionsbereichs |

| | | |
|-----------------------------------|---|--|
| <i>adaptationArea</i> | $(x; y; b; h; visible)$ mit $x, b \in \mathbb{N} \mid 0 \leq x, b \leq cellCountX,$ $y, h \in \mathbb{N} \mid 0 \leq y, h \leq cellCountY$ und $visible \in \{true; false\}$ | Ausgangskordinaten, Breite und Höhe sowie Anzeigeoptionen des Adaptationssteuerungsbereichs |
| <i>titleFontSize</i> | $\{titleFontSizes.small;$ $titleFontSizes.medium;$ $titleFontSizes.large;$ $titleFontSizes.xlarge\}$ | Schriftgröße aller Titelemente durch Referenz auf eine ausgewählte DP-Variable (s. Wertebereich) |
| <i>bodyTextFontSize</i> | $\{bodyTextFontSizes.small;$ $bodyTextFontSizes.medium;$ $bodyTextFontSizes.large;$ $bodyTextFontSizes.xlarge\}$ | Schriftgröße aller Textkörperelemente durch Referenz auf eine ausgewählte DP-Variable (s. Wertebereich) |
| <i>complementaryTextFontSize</i> | $\{complementaryTextFontSizes.small;$ $complementaryTextFontSizes.medium;$ $complementaryTextFontSizes.large;$ $complementaryTextFontSizes.xlarge\}$ | Schriftgröße aller ergänzenden Textelemente durch Referenz auf eine ausgewählte DP-Variable (s. Wertebereich) |
| <i>screenComplexity</i> | $\{unconstrained; medium; low\}$ | Maximale Gesamtanzahl der im Inhaltsbereich oder Funktionsbereich gleichzeitig dargestellten interaktiven Bildelemente |
| <i>eyeCandy</i> | $\{on; off\}$ | An-/Ausschalter für die Anzeige schmückender Bildelemente |
| <i>displayMode</i> | $\{text\ only;$ $mainly\ text;$ $text\ and\ graphics;$ mainly graphics; $graphics\ only\}$ | Mengenverhältnis von dargestellten Text- und Grafikelementen |
| <i>luminance</i> | $\{low; medium; high\}$ | Leuchtdichte der grafischen Darstellung |
| <i>interactiveElementsMinSize</i> | $\{interactiveElementsMinSizes.small;$ $interactiveElementsMinSizes.medium;$ $interactiveElementsMinSizes.large;$ $interactiveElementsMinSizes.xlarge\}$ | Mindestgröße von interaktiven Bildelementen durch Referenz auf eine ausgewählte DP-Variable (s. Wertebereich) |
| <i>audioOutputVolume</i> | $\{low; medium; high; silent\}$ | Lautstärkeniveau und An-/Ausschalter für akustische Ausgaben |
| <i>speechAudioOutput</i> | $\{on; off\}$ | An-/Ausschalter für akustische Sprachausgabe |
| <i>speechAudioSpeed</i> | $\{100; 75\}$ | Geschwindigkeitsniveau der akustischen Sprachausgaben in % der ursprünglichen Aufnahmegeschwindigkeit |
| <i>tonalAudioOutput</i> | $\{on; off\}$ | An-/Ausschalter für nicht-sprachliche akustische Ausgaben |
| <i>adaptationRenderingTime</i> | $\{300; 500\}$ | Zeitspanne in ms, in der eine Adaptation während der Interaktion durchgeführt wird |
| <i>timeOut</i> | $\{5000; 8000\}$ | Zeitspanne in ms, in der das System Nutzerreaktionen auf eine explizite Handlungsaufforderung registriert |

| | | |
|--------------------------|----------------|--|
| <i>confirmationLevel</i> | {medium; high} | Ausmaß, in dem das System explizites Feedback und Bestätigungen auf Nutzeraktionen gibt |
| <i>cursorInput</i> | {on; off} | An-/Ausschalter für Nutzereingaben über die Pfeiltasten |
| <i>numericInput</i> | {on; off} | An-/Ausschalter für Nutzereingaben über Nummerntasten |
| <i>touchInput</i> | {on; off} | An-/Ausschalter für Nutzereingaben über ein Touchdisplay oder Touchpad |
| <i>pointingInput</i> | {on; off} | An-/Ausschalter für Nutzereingaben über ein Zeigegerät |
| <i>keyboardInput</i> | {on; off} | An-/Ausschalter für Nutzereingaben über eine alphanumerische Tastatur |
| <i>voiceInput</i> | {on; off} | An-/Ausschalter für Nutzereingaben über gesprochene Sprache |
| <i>skin</i> | {MyUI; ...} | Diese Variable definiert, welches Customization-Profil verwendet werden soll. Sie gilt jeweils für eine Applikation und wird vom Entwickler gesetzt. |

Tabelle 6 Variablen des Benutzungsschnittstellenprofils (UIP)

5.6 Individualisierungsmuster (Individualization Patterns)

Individualisierungsmuster sind das Kernstück der automatischen ABS-Individualisierung (vgl. Peissner et al., 2012b). Sie führen die Kontextinformationen aus DP und UP zusammen, um das UIP zu erstellen und zu aktualisieren. Man könnte sagen, Individualisierungsmuster „übersetzen“ die im DP und im UP repräsentierten Eigenschaften der technischen Ausstattung, des Nutzers und der Umgebung in allgemeine Gestaltungsmerkmale der Benutzungsschnittstelle. Damit tragen die Individualisierungsmuster einen wesentlichen Teil des im MyUI-System hinterlegten Gestaltungswissens für barrierefreie Benutzungsschnittstellen. Die Formulierung von Individualisierungsmustern erfordert ganz konkrete Antworten auf Fragen wie „Welche Gestaltungsmerkmale machen eine Benutzungsschnittstelle auf einem kleinen Bildschirm auch für stark sehbehinderte Nutzer zugänglich?“ oder „Wie können Nutzer mit Aufmerksamkeitsdefiziten optimal bei der Erreichung ihrer Interaktionsziele unterstützt werden?“.

Individualisierungsmuster sind in Bündeln organisiert. Alle Muster eines Bündels wirken auf dieselben Variablen des UIP. Diese UIP-Variable dient in der Regel als Namensgeber für das Musterbündel. Abbildung 9 illustriert beispielsweise, wie verschiedene Muster des Bündels *DISPLAY MODE* je nach Wertekombinationen in den UP-Variablen *understandingAbstractSigns* und *languageReception* die UIP-Variable *displayMode* setzen könnten, um das Verhältnis von textueller zu bildhafter Informationsdarstellung optimal den individuellen Anforderungen des Nutzers anzupassen. Die farbliche Kodierung in den Zellen der Tabelle und vor den einzelnen Individualisierungsmustern verdeutlicht den Zusammenhang der Eingangswerte der UP-Variablen (If-Statements) mit den gesetzten Werten der UIP-Variable (Then-Statement).

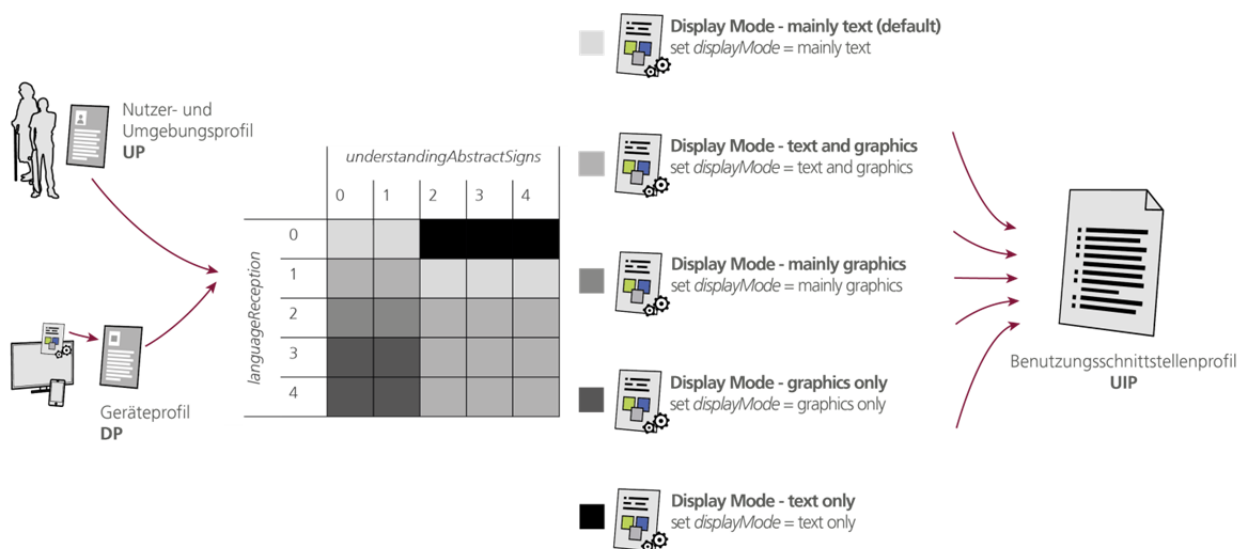


Abbildung 9 Die Individualisierungsmuster des Bündels DISPLAY MODE transformieren die Werte der UP-Variablen *languageReception* und *understandingAbstractSigns* in geeignete *displayMode*-Einstellungen im UIP

Wird die erste Interaktionssequenz eines Nutzers ohne Erstkonfiguration des UP gestartet, so stehen alle Variablen des UP auf ihren Default-Werten und alle Default-Individualisierungsmuster werden aktiviert. Dies führt zu einem Default-UIP. Jedes Mal, wenn Änderungen im UP registriert werden, werden die Regeln aller Individualisierungsmuster neu auf Gültigkeit geprüft und die Muster, deren If-Statement zutrifft werden aktiviert. Dadurch ergeben sich unter Umständen Veränderungen in den globalen Variablen des UIP. Damit ist die ABS-Parametrierung ein kontinuierlicher Prozess während der Interaktion. Veränderungen im UP oder dem DP werden permanent überwacht und lösen über die Aktivierung von neuen Individualisierungsmustern Aktualisierungen der UIP-Parameter aus.

Individualisierungsmuster können die folgenden Relationen zu anderen Mustern besitzen (vgl. Abschnitt 4.3):

- **substituiert**
Diese Relation fasst zwei oder mehrere Individualisierungsmuster zu Musterbündeln zusammen.
- **benötigt <Variable> wie gesetzt von**
Diese Relation verbindet Bündel von Individualisierungsmustern mit Bündeln von gerätespezifischen Mustern über Variablen des DP. Die Individualisierungsmuster „benötigen“ die von den gerätespezifischen Mustern gesetzten DP-Variablen, indem sie bestimmten UIP-Variablen die DP-Variablen als Werte zuweisen. Die Relation entspricht der Umkehrrelation der in Abschnitt 5.3 im Zusammenhang mit den gerätespezifischen Mustern beschriebenen Relation **setzt <Variable> wie benötigt von**.
- **setzt <Variable> wie benötigt von**
Diese Relation verbindet den Output von Individualisierungsmusterbündeln mit einzelnen Adaptationsausführungsmustern und Bedien- und Anzeigeelementen. Die von Individualisierungsmustern gesetzten UIP-Parameter werden in der Spezifikation der Lösungen der anderen Muster referenziert.

- nutzt <Variable> wie gesetzt von
Diese Relation verbindet Individualisierungsmusterbündel zur Auswahl des geeigneten Individualisierungsmusters mit Bündeln von gerätespezifischen Mustern über die Variablen des DP. Sie entspricht der Umkehrrelation der in Abschnitt 5.3 bei den gerätespezifischen Mustern beschriebenen Relation setzt <Variable> wie genutzt von.
- setzt <Variable> wie genutzt von
Diese Relation verbindet den Output von Individualisierungsmusterbündeln mit Bündeln von Interaktionsmustern. Je nach Einstellungen im UIP werden jeweils unterschiedliche Interaktionsmustervarianten ausgewählt.

Für die Beschreibung von Individualisierungsmusterbündeln ist das folgende Format vorgesehen (Tabelle 7):

| Individualization pattern bundle | <Name des Individualisierungsmusterbündels> |
|----------------------------------|---|
| Problem | <Beschreibung eines Gestaltungsproblems, das für die gesamte Benutzungsschnittstelle einheitlich gelöst werden soll und dessen Lösung je nach Kontextbedingungen in UP und DP unterschiedlich ausfallen soll> |
| Sets variables | <Auflistung der von den Mustern des Bündels gesetzten UIP-Variablen> |
| Patterns | <Auflistung der Entwurfsmuster des Bündels> |

Tabelle 7 Vorlage für die Beschreibung eines Bündels von Individualisierungsmustern

Für die Beschreibung einzelner Individualisierungsmuster gilt die in Tabelle 8 dargestellte Vorlage.

| Individualization pattern | <Name des Individualisierungsmusters> |
|-------------------------------|--|
| Problem | <Beschreibung des von diesem Muster adressierten Gestaltungsproblems, das darin besteht, (eine) bestimmte UIP-Variable(n) so zu setzen, dass mögliche Nutzungsbarrieren, die aus den spezifischen Kontextbedingungen aus UP und DP dieses Musters resultieren, möglichst gut kompensiert werden> |
| Context | 'IF' <DP-Variable> <UP-Variable> '=' <Wert> [('AND' 'OR') (<DP-Variable> <UP-Variable>) '=' <Wert>] * |
| Solution | 'THEN' 'set' <UIP-Variable> '=' <Wert(e)> ['set' <UIP-Variable> '=' <Wert(e)>] * |
| Code Reference | <Soweit möglich werden Teile der Lösung in einer programmnahen Form angegeben. Hierfür kann z.B. ein technologieunabhängiger Pseudocode verwendet werden.> |
| Diagram | <Soweit möglich wird die Gestaltungslösung in einer schematischen Darstellung veranschaulicht.> |
| Rationale (references) | <Erläuterung bzw. Begründung der Lösung; falls vorhanden können Querverweise zu entsprechender Fachliteratur oder Normen und Standards angegeben werden.> |

| | |
|----------------------------------|---|
| Ranking | <Bewertung der Reife des Musters aufgrund des Grads der Validierung auf einer dreistufigen Skala.> ::= '***' '**' '*' |
| Pattern Bundle | <Bündel, dem dieses Muster angehört > |
| Substitutes | <Auflistung der anderen Muster des Bündels, d.h. der Muster, die in Relation substituiert mit dem beschriebenen Muster stehen> |
| Requires <variable> | <Auflistung der DP-Variablen, die im Lösungsfeld des beschriebenen Musters referenziert werden, um einer UIP-Variablen einen Wert zuzuweisen> |
| Uses <variable> | <Auflistung der DP-Variablen, die im Kontextfeld des beschriebenen Musters referenziert werden> |

Tabelle 8 Vorlage für die Beschreibung eines Individualisierungsmusters

5.7 Customization-Profil (CP: Customization Profile)

Das Customization-Profil (CP: Customization Profile) leistet keinen Beitrag zur Überwindung von Nutzungsbarrieren oder zur Steigerung des individuellen Nutzungskomforts für die Nutzer. Vielmehr trägt das CP der Anforderung des Marktes nach einem firmenspezifischen oder markenspezifischen Auftreten von interaktiven Produkten und Services Rechnung. Das CP beinhaltet eine Sammlung von Variablen, die bestimmte Eigenschaften der Benutzungsschnittstelle definieren, die als wesentliche Komponenten eines unterscheidbaren Erscheinungsbilds eines Unternehmens (Corporate Design) oder einer Marke gelten. Das CP umfasst die folgenden Bereiche:

- Farbschema,
- Schriftart,
- verwendete Bilder und Icons und
- verwendete Klänge.

All diese Bereiche können vom Entwickler frei gestaltet werden, um den Auftritt der Applikation zu beeinflussen. Die speziellen CP-Einstellungen spiegeln sich dann in allen möglichen angepassten Instanzen der generierten ABS wider. Customization-Profile werden von Entwicklern angelegt, verändert und unter frei gewählten Namen gespeichert. Der Entwickler gibt für jede Applikation an, welches CP verwendet werden soll. Diese Information wird in der UIP-Variable *skin* gespeichert. So gehen die Werte des gewählten CP direkt in die ABS-Parametrierung ein. Unter dem *skin*-Namen „MyUI“ steht ein Default-CP zur Verfügung, das dem Erscheinungsbild des MyUI-Projekts entspricht. Wird der Wert der *skin*-Variable nicht verändert, so wird dieses Default-CP genutzt.

In Tabelle 9 findet sich ein Überblick der Variablen des CP. Die zahlreichen Variablen des Farbschemas bieten weitreichende Möglichkeiten für farbliche Unterscheidungen. Wird eine schlichtere Gestaltung bevorzugt, können mehreren Variablen (z.B. *primaryObjectsColour* und *secondaryObjectsColour*) dieselben Farbwerte zugewiesen werden. Im CP werden alle Farbwerte in einem gängigen Farbmodell wie RGB („Red, Green, Blue“: additives Farbmodell mit den drei Grundfarben rot, grün und blau) oder CMYK („Cyan, Magenta, Yellow, Key“: substraktives Farbmodell mit den drei Farbkomponenten cyan, magenta und gelb sowie dem Schwarzanteil „Key“) angegeben.

| CP Variable | Wertebereich | Beschreibung |
|--|------------------------|--|
| Variablen zum Farbschema | | |
| <i>mainBackgroundColour</i> | <Farbwerte> | Farbe des Bildschirmhintergrunds |
| <i>titleBarBackgroundColour</i> | <Farbwerte> | Farbe des Hintergrunds der Titelleiste |
| <i>contentBackgroundColour</i> | <Farbwerte> | Farbe des Hintergrunds des Inhaltsbereichs |
| <i>primaryObjectsColour</i> | <Farbwerte> | Farbe der Objekte, die durch primäre Interaktionssituationen ²³ aufgerufen werden (typischerweise im Inhaltsbereich oder in Overlays dargestellt) |
| <i>secondaryObjectsColour</i> | <Farbwerte> | Farbe der Objekte, die durch weitere Interaktionssituationen aufgerufen werden (typischerweise im Funktionsbereich dargestellt) |
| <i>supplementaryObjectsColour</i> | <Farbwerte> | Farbe der Objekte, die der Meta-Interaktion dienen, wie z.B. Bestätigung in Dialogfenstern, Scrollen und Blättern |
| <i>dialogueBoxBackgroundColour</i> | <Farbwerte> | Hintergrundfarbe der Dialogfenster |
| <i>confirmationPositiveColour</i> | <Farbwerte> | Farbe für Elemente, die eine positive Rückmeldung signalisieren (in der Regel ein Grünton) |
| <i>confirmationNegativeColour</i> | <Farbwerte> | Farbe für Elemente, die eine negative Rückmeldung signalisieren (in der Regel ein Rotton) |
| <i>textInputBackgroundColour</i> | <Farbwerte> | Hintergrundfarbe der Texteingabefelder und Tooltips |
| <i>textOutputBackgroundColour</i> | <Farbwerte> | Hintergrundfarbe der Textausgabefelder |
| <i>primaryTextColour</i> | <Farbwerte> | Farbe der Textausgabeelemente; sie wird auf fast allen Hintergrund- und Objektfarben verwendet. Ausnahmen: <i>textInputBackground</i> und <i>dialogueBoxBackground</i> . |
| <i>secondaryTextColour</i> | <Farbwerte> | Farbe der Textelemente, die mit einer Nutzereingabe assoziiert sind; diese Farbe wird auf den Hintergrundfarben <i>textInputBackgroundColour</i> und <i>dialogueBoxBackgroundColour</i> dargestellt. |
| Variablen zu Schrift, Grafikdateien und Klängen | | |
| <i>fontStyle</i> | <Name des Schrifttyps> | Bezeichnung des zu verwendenden Schrifttyps, z.B. „Verdana“ oder „Myriad Pro“ |
| <i>graphicsLibrary</i> | <absoluter Pfadname> | Alle zu verwendenden Bilder und Icons werden unter festgelegten Dateinamen in einem Verzeichnis abgelegt, dessen Pfad dieser Variablen als Wert zugewiesen wird. |
| <i>soundsLibrary</i> | <absoluter Pfadname> | Alle zu verwendenden Klänge werden unter festgelegten Dateinamen in einem Verzeichnis abgelegt, dessen Pfad dieser Variable als Wert zugewiesen wird. |

Tabelle 9 Variablen des Customization-Profiles (CP)

²³ Eine Erläuterung der „primären Interaktionssituation“ findet sich in Abschnitt 8.4.

6 Vorbereitung der adaptiven Benutzungsschnittstelle

Der zweite Prozessschritt der ABS-Generierung und Adaptation ist die Vorbereitung der ABS. Während dieses Schrittes wird eine Auswahl von Interaktionsmustern zusammengestellt, die die aktuellen Interaktionssituationen vor dem Hintergrund der angenommenen Kontextbedingungen in bestmöglicher Weise zugänglich machen. Die ausgewählten Interaktionsmuster wiederum rufen Bedien- und Anzeigeelemente auf, die kleinere, generische Aspekte der Gesamtlösung – wie zum Beispiel einzelne Buttons - beschreiben. So liegt als Ergebnis der ABS-Vorbereitung eine Menge von Interaktionsmustern und mit ihnen verbundenen Bedien- und Anzeigeelementen vor.

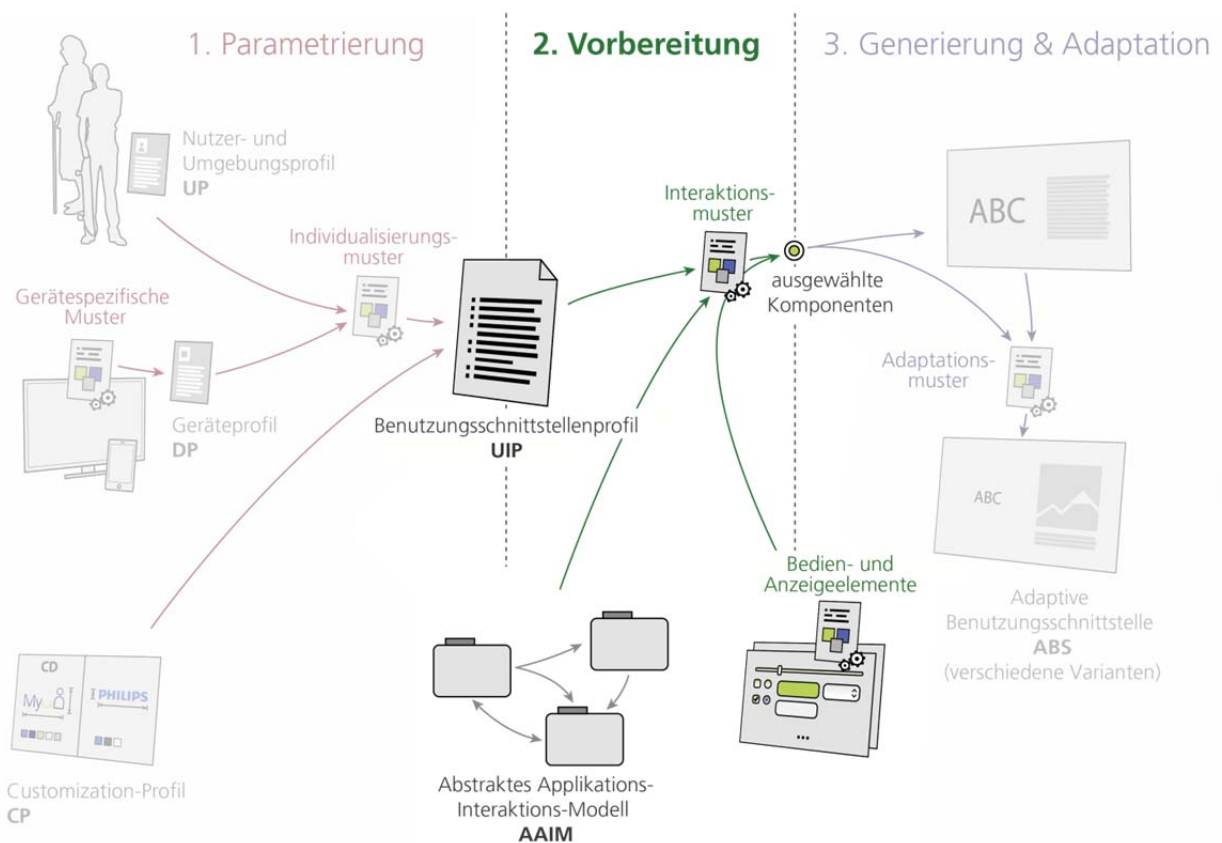


Abbildung 10 Wesentliche Konzepte im Prozess der ABS-Vorbereitung

Die maßgebenden Konzepte der ABS-Vorbereitung umfassen (vgl. Abbildung 10):

- Das *Abstrakte Applikations-Interaktions-Modell* (AAIM, vgl. Kapitel 8) liefert zu jedem Zeitpunkt eine Menge von Interaktionssituationen, die beschreiben, welche Interaktionsmöglichkeiten sich dem Nutzer im aktuellen Zustand der Applikation bieten. Diese Interaktionssituationen bestimmen, welche Interaktionsmusterbündel zur Generierung der ABS benötigt werden.
- Das *Benutzungsschnittstellenprofil* (UIP) stellt neben dem AAIM den zweiten wesentlichen Input zur Auswahl der zu verwendenden Interaktionsmuster dar. Die Variablen des UIP

bestimmen, welche Interaktionsmuster der vom AAIM vorausgewählten Bündel zum Einsatz kommen sollen.

- *Interaktionsmuster* beschreiben Darstellungs- und Interaktionskomponenten der ABS. Sie bieten Gestaltungslösungen für die im AAIM referenzierten Interaktionssituationen und verfügen über das notwendige Regelwissen zur Auswahl der am besten geeigneten Bedien- und Anzeigeelemente.
- *Bedien- und Anzeigeelemente* werden von den Interaktionsmustern durch die Relation *benötigt* aufgerufen, um Teile der von den Interaktionsmustern beschriebenen Lösungen durch generische Elemente zu detaillieren.

6.1 Interaktionsmuster (Interaction Patterns)

Interaktionsmuster stellen den Dreh- und Angelpunkt der ABS-Vorbereitung dar. Sie besitzen das Wissen, um für die aktuellen Interaktionssituationen und Kontextbedingungen die Elemente auszuwählen, aus denen die ABS erzeugt werden soll. Für jede Interaktionssituation gibt es nicht nur ein Interaktionsmuster, sondern ein ganzes Bündel von Mustern. In Abhängigkeit der aktuellen Werte des UIP wird jeweils das Interaktionsmuster aus dem Bündel ausgewählt, das die individuellen Einschränkungen und Kontextbedingungen am besten abdeckt. Dieser Auswahlprozess wird während einer Interaktionssequenz mit einer MyUI-Anwendung mehrmals durchgeführt. Ausgelöst wird er durch zwei Ereignisse: (1) wenn im AAIM ein Zustandsübergang erfolgt, d.h. wenn neue Interaktionssituationen aktiviert werden, und (2) wenn Änderungen im UIP registriert werden.

Interaktionsmuster besitzen keine Referenzen zum UP. Ihre Eingangsgrößen beschränken sich auf die Interaktionssituationen des AAIM und die Parameter des UIP. Als ein Ergebnis ihrer Arbeit rufen Interaktionsmuster Bedien- und Anzeigeelemente zur Detaillierung bestimmter Teilaspekte ihrer Lösung auf.

Die folgenden Relationen beschreiben die Beziehungen, in denen Interaktionsmuster zu anderen Mustern stehen können (vgl. Abschnitt 4.3):

- *substituiert*
Diese Relation fasst zwei oder mehrere Interaktionsmuster zu Musterbündel zusammen.
- *benötigt*
Diese Relation verbindet Interaktionsmuster, die ganze Interaktionskomponenten (User Interface Widgets wie zum Beispiel ein Menü) beschreiben, mit Bedien- und Anzeigeelementen, die kleinere, generische Bestandteile dieser Lösung (wie zum Beispiel Menü-Buttons) spezifizieren.
- *nutzt <Variable> wie gesetzt von*
Diese Relation zwischen Bündeln von Interaktionsmustern und Individualisierungsmustern drückt die nutzerindividuelle und kontextspezifische Auswahl der geeigneten Variante eines Interaktionsmusterbündels aus. So wird zum Beispiel die Auswahl einer Menüvariante durch mehrere Variablen des UIP bestimmt, die von Individualisierungsmustern aufgrund der Darstellungskapazität des Geräts, der Aufmerksamkeitsspanne des Nutzers und der genutzten Eingabemechanismen gesetzt werden.

Über die Interaktionssituationen des AAIM sind Interaktionsmuster mit den Inhalten und Funktionen der Applikation verbunden. Interaktionsmuster bringen Applikationsinhalte zur Darstellung auf der ABS und bieten den Nutzern die Möglichkeit, Applikationsinhalte zu manipulieren und eventuell neue Inhalte zu erstellen. Darüber hinaus stellen Interaktionsmuster den Nutzern Mechanismen zur Verfügung, mit denen sie Applikationsfunktionen auslösen können. Da alle Interaktionsmuster eines Bündels demselben Zweck innerhalb der ABS dienen, besitzen sie dieselben Schnittstellen für den Datenaustausch mit der Applikationslogik. Die Datenstrukturen für die Eingangs- und Ausgangsparameter sind daher im Standardformat der Beschreibung eines Interaktionsmusterbündels enthalten (s. Tabelle 10).

| Interaction pattern bundle | <Name des Interaktionsmusterbündels> |
|----------------------------|--|
| Problem | <Beschreibung des Zwecks der Interaktionskomponenten, die durch die Muster des Bündels bereitgestellt werden; dabei handelt es sich um eine Interaktionssituation oder eine Gruppe von ähnlichen Interaktionssituationen.> 'Interaction Situation =' <Interaktionssituation des AAIM> ['OR' <Interaktionssituation des AAIM>] * |
| Parameters | 'IN' <Datenstruktur und Beschreibung der Eingangsparameter der Muster dieses Bündels, v.a. zur Spezifikation des Inhalts der Interaktionskomponente> 'OUT' <Datenstruktur und Beschreibung der Ausgangsparameter der Muster dieses Bündels, z.B. um die vom Nutzer ausgewählte Option einer Auswahlkomponente (z.B. Auswahlliste) zu spezifizieren> |
| Patterns | <Auflistung der Entwurfsmuster des Bündels> |

Tabelle 10 Vorlage für die Beschreibung eines Bündels von Interaktionsmustern

Die einzelnen Interaktionsmuster werden nach der in Tabelle 11 dargestellten Vorlage beschrieben.

| Interaction pattern | <Name des Interaktionsmusters> |
|---------------------|---|
| Problem | <Beschreibung des von diesem Muster adressierten Gestaltungsproblems und Benennung der Interaktionssituation(en), für die das Muster eine Interaktionskomponente bietet> 'Interaction Situation =' <Interaktionssituation des AAIM> ['OR' <Interaktionssituation des AAIM>] * |
| Context | 'IF' <UIP-Variable> '=' <Wert> [('AND' 'OR') <UIP-Variable> '=' <Wert>] * |
| Solution | 'THEN' <Beschreibung der Lösung des Musters, indem eine Interaktionskomponente mit allen wesentlichen Eigenschaften in natürlicher Sprache spezifiziert wird; dabei werden die ggf. genutzten Bedien- und Anzeigeelemente benannt.> |

| | |
|-------------------------------|---|
| Code Reference | <Soweit möglich werden Teile der Lösung in einer programmierbaren Form angegeben. Hierfür kann z.B. ein technologieunabhängiger Pseudocode verwendet werden.> |
| Diagram | <Soweit möglich wird die Gestaltungslösung in einer schematischen Darstellung veranschaulicht.> |
| Rationale (references) | <Erläuterung bzw. Begründung der Lösung; falls vorhanden können Querverweise zu entsprechender Fachliteratur oder Normen und Standards angegeben werden.> |
| Ranking | <Bewertung der Reife des Musters aufgrund des Grads der Validierung auf einer dreistufigen Skala.> ::= '***' '**' '*' |
| Pattern Bundle | <Bündel, dem dieses Muster angehört > |
| Substitutes | <Auflistung der anderen Muster des Bündels, d.h. der Muster, die in Relation <i>substituiert</i> mit dem beschriebenen Muster stehen> |
| Requires | <Auflistung der von diesem Muster in der Lösungsbeschreibung referenzierten Bedien- und Anzeigeelemente, d.h. Auflistung der Muster aus der Menge <i>B</i> der Bedien- und Anzeigeelemente, die in Relation <i>wird benötigt von</i> mit dem beschriebenen Muster stehen> |
| Uses <variable> | <Auflistung der <i>UIP-Variablen</i> , die im Kontextfeld des beschriebenen Musters referenziert werden> |

Tabelle 11 Vorlage für die Beschreibung eines Interaktionsmusters

6.2 Bedien- und Anzeigeelemente (User Interface Elements)

Bedien- und Anzeigeelemente sind die generischen Bausteine, aus denen die Interaktionsmuster zusammengesetzt werden. So benötigt beispielsweise die Darstellung einer Auswahlliste (Interaktionsmuster) mehrere Optionsbuttons (Bedien- und Anzeigeelement). Bedien- und Anzeigeelemente unterscheiden sich in ihrer Struktur und Funktionsweise von den anderen Entwurfsmustern. Sie sind nicht in Bündeln strukturiert, d.h. die Relation *substituiert* ist nicht auf sie anwendbar und es gibt keine Varianten. Damit ist die Auswahl der Bedien- und Anzeigeelemente vergleichsweise einfach und erfolgt ohne die Prüfung komplexer Regeln. Die gerade aktivierten Interaktionsmuster rufen die entsprechend benötigten Bedien- und Anzeigeelemente direkt auf. Diese Aufrufe erfolgen über die Relation *benötigt*. Dabei weisen sie ihnen wesentliche Parameter wie Größe und Position auf dem Bildschirm zu. So kann die Menge *B* der Bedien- und Anzeigeelemente als eine Bibliothek von generischen und parametrierbaren ABS-Elementen verstanden werden.

Die Lösungsbeschreibungen der Bedien- und Anzeigeelemente sind dafür etwas komplexer. Für jedes Bedien- und Anzeigeelement ist ein über alle Instanzen des Musters einheitlich genutztes, allgemeines Aussehen und Interaktionsverhalten definiert. Jedoch richten sich einzelne Details des Aussehens nach den im UIP abgebildeten Kontextbedingungen. Beispielsweise hängt die genaue Ausführung von Button-Beschriftungen vom Wert der UIP-Variable *displayMode* ab. Je nachdem ob eher bildhafte oder textuelle Darstellungsformen genutzt werden sollen, wird auch das Text-Icon-Verhältnis einer Buttonbeschriftung angepasst. Im Gegensatz zu anderen Mustern sitzen die dafür zuständigen Anpassungsregeln bei den Bedien- und Anzeigeelementen direkt in der Lösungsbeschreibung des Musters. Damit können Bedien- und Anzeigeelemente als adaptive Elemente betrachtet werden, die in bestimmten Grenzen ihr Aussehen selbst verändern können. In dieser Hinsicht ähneln die Bedien- und Anzeigeelemente den sogenannten „Comets“ (Calvary

et al., 2004), die einen wesentlichen Beitrag zu den Laufzeitanpassungen der PLASTIC USER INTERFACES leisten (vgl. Kapitel 2).

Für Bedien- und Anzeigeelemente sind die folgenden Relationen definiert:

- `wird benötigt von`
Diese Relation verbindet Bedien- und Anzeigeelemente mit Interaktionsmustern, von denen sie aufgerufen (d.h. `benötigt`) werden, um Teilaspekte ihrer Lösung zu spezifizieren.
- `benötigt <Variable> wie gesetzt von`
Diese Relation verbindet Bedien- und Anzeigeelemente mit Bündel von Individualisierungsmustern. Die betroffenen Bedien- und Anzeigeelemente referenzieren in der Lösungsbeschreibung auf Variablen des UIP, die von Individualisierungsmustern gesteuert werden.
- `erweitert`
Diese Relation etabliert Vererbungsmechanismen in der Menge *B* der Bedien- und Anzeigeelemente. Sie verbindet spezifische Elemente mit abstrakten generischen Mustern, die die Gemeinsamkeit einer ganzen Klasse von ähnlichen Bedien- und Anzeigeelementen definieren.
- `wird erweitert von`
Diese Relation ist die Umkehrrelation von `erweitert`.

Die Vorlage zur Beschreibung von Bedien- und Anzeigeelementen findet sich in Tabelle 12. Nachdem in der Menge *B* der Bedien- und Anzeigeelemente keine Bündel bestehen, ist eine Beschreibung des Kontexts nicht nötig. Die Vorlage enthält daher kein Kontextfeld.

| User interface element | <Name des Bedien- und Anzeigeelements> |
|------------------------|---|
| Problem | <p><Beschreibung des von diesem Muster adressierten Gestaltungsproblems und Benennung der Interaktionsmuster, für die das Muster ein Bedien- und Anzeigeelement zur Verfügung stellt></p> <p>'is required by' <Auflistung der Interaktionsmuster, die dieses Bedien- und Anzeigeelement in ihrer Lösungsbeschreibung referenzieren, d.h. Auflistung der Muster aus der Menge <i>K</i> der Interaktionsmuster, die in Relation <code>benötigt</code> mit dem beschriebenen Muster stehen></p> |
| Solution | <p>['extends' <Bedien- und Anzeigeelement, zu dem das Muster in Relation <code>erweitert</code> steht>]</p> <p><Beschreibung der Lösung des Musters, indem ein Bedien- oder Anzeigeelement mit allen wesentlichen Eigenschaften in natürlicher Sprache spezifiziert wird; erbt das Muster die Eigenschaften eines durch die Relation <code>erweitert</code> referenzierten Bedien- und Anzeigeelements, so werden nur die Erweiterungen beschrieben. Wenn das Bedien- und Anzeigeelement je nach Bedingungen des UIP unterschiedlich dargestellt werden soll, werden die Bedingungen und jeweiligen Ausführungen beschrieben.></p> |
| Code Reference | <Soweit möglich werden Teile der Lösung in einer programmierbaren Form angegeben. Hierfür kann z.B. ein technologieunabhängiger Pseudocode verwendet werden.> |
| Diagram | <Soweit möglich wird die Gestaltungslösung in einer schematischen Darstellung veranschaulicht.> |

| | |
|-------------------------------|--|
| Rationale (references) | <Erläuterung bzw. Begründung der Lösung; falls vorhanden können Querverweise zu entsprechender Fachliteratur oder Normen und Standards angegeben werden.> |
| Ranking | <Bewertung der Reife des Musters aufgrund des Grads der Validierung auf einer dreistufigen Skala.> ::= '***' '**' '*' |
| Extended by | <Auflistung der Bedien- und Anzeigeelemente, die das Muster in ihrem Lösungsfeld referenzieren, um es spezifisch zu erweitern, d.h. Auflistung der Muster, die in Relation <i>erweitert</i> mit dem beschriebenen Muster stehen; bei nicht-abstrakten Bedien- und Anzeigeelementen bleibt dieses Feld leer.> |

Tabelle 12 Vorlage für die Beschreibung eines Bedien- und Anzeigeelements

7 Generierung und Adaptation der adaptiven Benutzungsschnittstelle

In diesem letzten Schritt wird auf Grundlage der in der ABS-Vorbereitung ausgewählten Komponenten und Elementen die konkrete ABS generiert. Die ABS wird permanent konsistent zum UIP gehalten und ist damit zu jedem Zeitpunkt optimal an das aktuell verfügbare Wissen über Nutzer, Ausstattung und Umgebung angepasst. Ergeben sich während der Interaktion Änderungen in der Wissensbasis, d.h. Aktualisierungen in den MyUI-Profilen, so wird unmittelbar eine neue Benutzungsschnittstelle generiert. Um diese Laufzeitanpassungen zu unterstützen, bieten Adaptionsmuster spezielle Mechanismen zum Wechsel von einer Instanz der ABS zur nächsten.

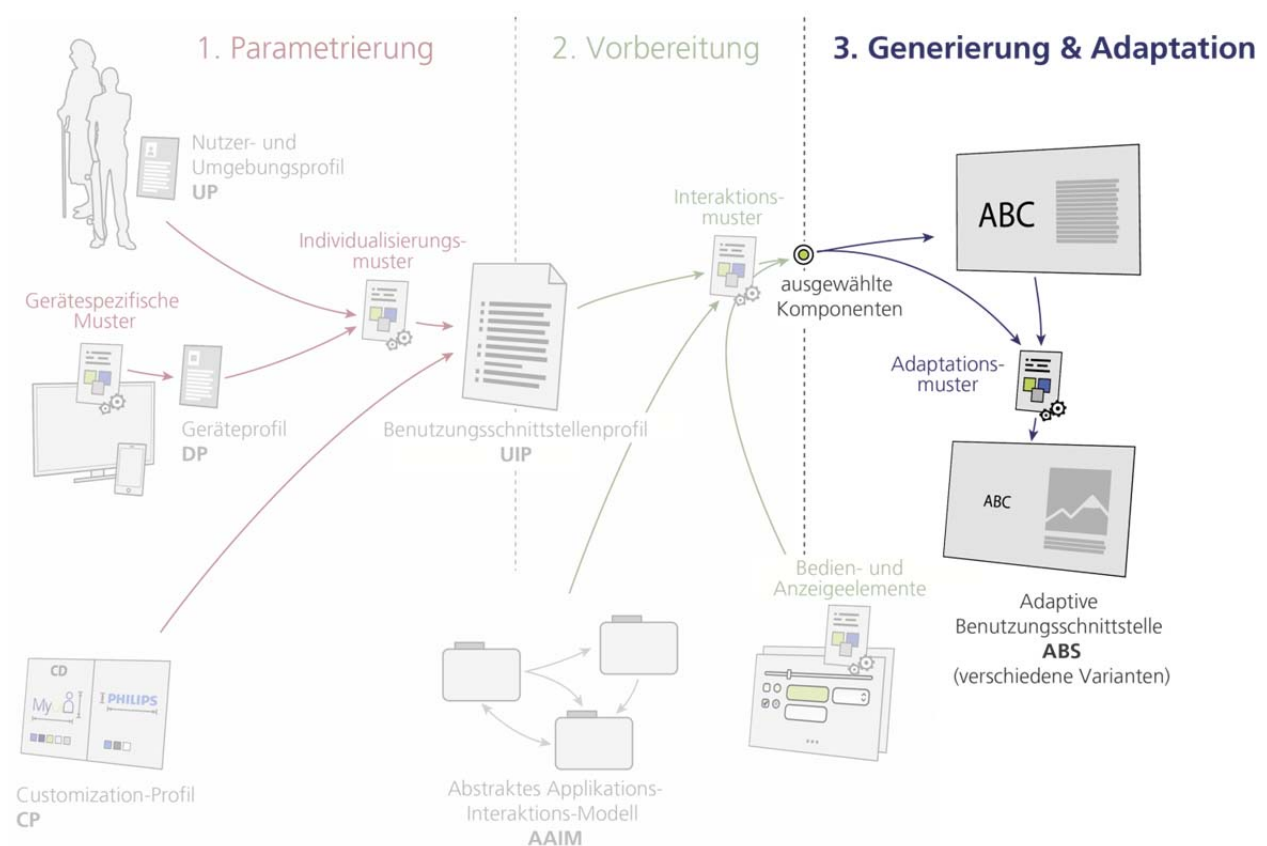


Abbildung 11 Wesentliche Konzepte im Prozess der ABS-Generierung und Adaptation

Wie in Abbildung 11 dargestellt, basiert die ABS-Generierung und Adaptation insbesondere auf den folgenden Konzepten:

- Die in der ABS-Vorbereitung *ausgewählten Bedien- und Anzeigekomponenten* stellen den Ausgangspunkt der ABS-Generierung dar. Sie sind die Bausteine der ABS.
- Die *adaptive Benutzungsschnittstelle (ABS)* ist das Ergebnis der ABS-Generierung und Adaptation.

- *Adaptationsmuster* spielen eine tragende Rolle bei Anpassungen während der Interaktion. Sie steuern Laufzeitanpassungen, indem sie die Übergänge von einer Instanz der ABS zur nächsten definieren. Ihre wichtigste Funktion ist es, Laufzeitanpassungen für den Nutzer transparent zu gestalten und dem Nutzer ein hohes Maß an Kontrolle zu gewährleisten.

Die ABS-Generierung und Adaptation umfasst drei wesentliche Aktivitäten:

- *Generierung der Benutzungsschnittstelle*: Zu Beginn einer neuen Interaktionssequenz wird eine Benutzungsschnittstelle auf Basis der ausgewählten Komponenten erzeugt.
- *Aktualisierungen der Profile*: Während einer Interaktionssequenz werden relevante Ereignisse der Interaktion aufgenommen und zur Aktualisierung des UP und DP genutzt. Bei signifikanten Veränderungen der beiden Profile wird ein erneutes Durchlaufen des dreistufigen Prozesses ausgelöst.
- *Adaptationen während der Nutzung*: Wird in einer wiederholten ABS-Vorbereitung eine neue Auswahl von Bedien- und Anzeigekomponenten erzeugt, die sich von der aktuell dargestellten Konfiguration der Benutzungsschnittstelle unterscheidet, werden Adaptationen der Benutzungsschnittstelle ausgelöst und durchgeführt.

7.1 Generierung der Benutzungsschnittstelle

Aus den ausgewählten Interaktionsmustern und Bedien- und Anzeigeelementen wird eine vollständige Benutzungsschnittstelle generiert. Dies erfolgt zu Beginn einer neuen Interaktionssequenz mit einer MyUI-Applikation und sobald Veränderungen in der Menge der ausgewählten Komponenten registriert werden. Damit wird jedes Mal, wenn ein neuer Interaktionszustand des AAIM aktiviert wird, ein ABS-Generierungsprozess gestartet, um eine geeignete Sicht für die neuen Interaktionsoptionen zu erstellen. Darüber hinaus kann die Benutzungsschnittstelle auch innerhalb eines Interaktionszustandes wiederholt generiert werden. Nämlich dann, wenn für eine andauernde Interaktionssituation eine neue Menge von Interaktionsmustern und Bedien- und Anzeigeelementen ausgewählt wird. Dieser Fall entspricht der in Abschnitt 7.3 beschriebenen Adaptation während der Nutzung.

Beim Erzeugen der ABS werden Applikationsdaten als Inhalte für die darzustellenden Komponenten und Bedien- und Anzeigeelemente benötigt. So stellen beispielsweise Menüelemente konkrete Applikationsinhalte und Optionen dar und Buttons erfordern textuelle oder ikonografische Beschriftungen, die sich aus dem Inhalt der Applikation ergeben. Hierfür stellt das AAIM die Verbindung zwischen den Bedien- und Anzeigekomponenten und den Datenquellen der Applikation her. Sogenannte „Datenakquisitionsfunktionen“ (data acquisition functions) verbinden die Interaktionssituationen des AAIM – und damit die Interaktionsmuster – mit den Applikationsdaten.

Der in dieser Arbeit vorgestellte Ansatz trifft keine Festlegung auf eine bestimmte Technologie der generierten Benutzungsschnittstelle bzw. der Generierungsmechanismen. Grundsätzlich können das MyUI-Rahmenmodell und seine wesentlichen Konzepte für barrierefreie ABS in jeder gängigen Oberflächentechnologie umgesetzt werden. Aufgrund der Vorteile bezüglich der

Plattformunabhängigkeit und der einfacheren Handhabung möglicher Erweiterungen des Entwurfsmusterrepositoriums empfehlen sich browserbasierte Implementierungen²⁴.

7.2 Rückkopplung zum Nutzer- und Kontextmanagement

Die ABS unterstützt die Funktionen der Afferenz und Inferenz des adaptiven Systems, indem sie während der Interaktion bestimmte Ereignisse, die Aufschlüsse über relevante Eigenschaften des Nutzers oder der Umgebung bieten können, an die MyUI-Kontextmanagementinfrastruktur zurückmeldet. Damit leistet die ABS einen wertvollen Beitrag zu einem validen Nutzer- und Umgebungsprofil (UP). Während der gesamten Interaktion registriert sie bestimmte Nutzereingaben und andere Interaktionsereignisse und teilt sie der Kontextmanagementinfrastruktur mit, damit die gegebenenfalls notwendigen UP-Änderungen ausgelöst werden können. So wird eine permanente Verfeinerung und Optimierung des UP auf der Grundlage von erkennbarem Nutzerverhalten unterstützt. Damit die hierfür eingesetzten Mechanismen auch bei der Überwindung von auftretenden Nutzungsbarrieren helfen können, ist es insbesondere interessant, solche Interaktionsereignisse zu erkennen, die auf Bedienprobleme hinweisen. Diese umfassen:

- *(Wiederholtes) Time-Out-Ereignis*
Wenn eine Nutzereingabe erwartet wird und der Nutzer in einem gewissen Zeitrahmen nicht reagiert, kann dies auf Probleme hinweisen. Mögliche Interpretationen umfassen u.a. Probleme beim Lesen oder beim Verstehen der dargestellten Text- oder Grafikelemente und Aufmerksamkeitsdefizite bzw. eine Überforderung durch eine zu komplexe grafische Darstellung.
- *Wiederholtes „Zurück“*
Wenn der Nutzer mehrmals hintereinander die „Zurück“-Funktion betätigt oder mehrmals hintereinander eine gerade getätigte Eingabe rückgängig macht („Undo“), kann das als ein Hinweis auf Bedienprobleme interpretiert werden. Möglicherweise kann sich der Nutzer nicht gut orientieren und findet nicht den richtigen Weg zum gewünschten Ziel bzw. findet nicht die richtige Funktion, um die gewünschte Aktion auszulösen.
- *Wiederholte unzulässige Nutzereingabe*
Wenn mehrmals hintereinander ein Interaktionsverhalten des Nutzers registriert wird, das nicht im Sinne einer zulässigen Nutzereingabe interpretiert werden kann, kann dies als ein Hinweis auf Bedienprobleme gewertet werden. Beispiele umfassen u.a. Mausclicks oder Bildschirmberührungen (Touch-Ereignisse), die kein interaktives Element treffen. In diesem Fall können kognitive oder motorische Einschränkungen die Problemursache darstellen.
- *Nutzerkonfigurationen in UP und UIP*
Eine weitere Kategorie von UP-relevanten Informationen, die von der ABS zur Kontextmanagementinfrastruktur zurück gemeldet werden, betreffen explizit vom Nutzer durchgeführte Änderungen im UP bzw. im UIP (vgl. Abschnitt 7.6).

²⁴ Die aktuelle Implementierung des MyUI-Systems basiert auf CakePHP (serverseitig) und Java Script (clientseitig) zur Generierung der ABS in HTML, CSS und AJAX/Java Script (vgl. Gacimartin et al., 2011 und Peissner et al., 2013).

- *Nutzerverhalten in Adaptationsdialogmustern*
Schließlich bieten die Adaptationsdialogmuster gewisse Einflussmöglichkeiten des Nutzers auf das Adaptationsverhalten des Systems. Wenn der Nutzer eine geplante Anpassung explizit akzeptiert oder ablehnt oder wenn er eine bereits durchgeführte Anpassung rückgängig macht, können daraus Rückschlüsse auf seine Anforderungen und Präferenzen gezogen werden (vgl. Abschnitt 7.5).

Darüber hinaus können externe (hardwaretechnische) Sensoren an die MyUI-Kontextmanagementinfrastruktur angeschlossen werden, die Nutzerverhalten erfassen können, das sich nicht an der ABS niederschlägt. Als Beispiele können Gesichtsausdrücke, Kopfausrichtung, Blickrichtung, etc. genannt werden. Bei Hernández et al., 2011 findet sich eine vollständigere Beschreibung der Quellen und Mechanismen zur Erfassung, Verarbeitung und Speicherung von Informationen für das UP. Die Interpretation der teilweise unspezifischen Interaktionsereignisse und der Umgang mit der damit verbundenen Ungewissheit stehen nicht im Fokus dieser Arbeit. Entscheidend für die efferente Komponente ist vielmehr die Fähigkeit, während einer Interaktionssequenz sofort auf Veränderungen im UP und im UIP durch eine entsprechende Anpassung der ABS reagieren zu können.

7.3 Adaptationen während der Nutzung

Die ABS-Anpassungen während einer Nutzungssequenz werden von der MyUI-Adaptationsengine durchgeführt. Wie in Abschnitt 3.5 erläutert, ist die Adaptationsengine eine wesentliche Komponente des Controllers der ABS. Sie führt permanent einen Abgleich der aktuell angezeigten mit den in der ABS-Vorbereitung zur Anzeige ausgewählten Komponenten und Elementen durch. Erkennt sie dabei eine Diskrepanz, startet sie eine Laufzeitanpassung, die den Übergang zu einer neuen Benutzungsschnittstelle, die sich aus den nun ausgewählten Komponenten und Elementen zusammensetzt, bewirkt. Laufzeitanpassungen werden von den *Adaptationsmustern* gesteuert. Adaptationsmuster beschreiben den Prozess des Übergangs von einer Instanz der ABS zur nächsten. Ein wesentliches Ziel der Adaptationsmuster ist es, den Anpassungsprozess für den Nutzer transparent und kontrollierbar zu gestalten. Der Nutzer soll bemerken, dass eine Anpassung erfolgt und erkennen, welche Aspekte der ABS wie angepasst werden. Darüber hinaus soll der Nutzer trotz eines hohen Maßes an Systeminitiative ausreichend viel Kontrolle über das Anpassungsverhalten erleben.

Das MyUI-Entwurfsmusterrepositorium besitzt zwei Typen von Adaptationsmustern, die jeweils unterschiedliche Ebenen einer Adaptation behandeln:

- *Adaptationsausführungsmuster* (Adaptation Rendering Patterns)
definieren die grafischen Übergänge bei der Ausführung von Anpassungen.
- *Adaptationsdialogmuster* (Adaptation Dialogue Patterns)
beschreiben den Dialog zwischen ABS und Nutzer vor, während und nach der eigentlichen Anpassung.

Laufzeitanpassungen werden immer durch eine Veränderung im UIP ausgelöst. Die Initiative hierfür kann sowohl beim System als auch beim Nutzer liegen. Eine automatische, systeminitiierte Anpassung liegt vor, wenn aufgrund erkannter Kontextveränderungen das UP oder DP angepasst wird und eine erneute ABS-Parametrierung ausgelöst wird. Darüber hinaus können auch Nutzer Anpassungen initiieren, indem sie Variablenwerte des UP oder UIP verändern. Durch die

Kombination von Nutzerinitiative und Systeminitiative sowie durch die Kontroll- und Einflussmöglichkeiten, die die Adaptationsdialogmuster den Nutzern bieten, wird MyUI zu einem System mit gemischter Initiative, das sowohl adaptive als auch adaptierbare Mechanismen vereint.

7.4 Adaptationsausführungsmuster (Adaptation Rendering Patterns)

Adaptationsausführungsmuster spezifizieren die detaillierte Ausführung von Laufzeitanpassungen während einer Interaktionssequenz. Sie zielen darauf ab, während einer Laufzeitadaptation die Nutzerorientierung zu unterstützen und eine möglichst hohe Kontinuität zwischen den Instanzen der ABS vor und nach der Adaptation herzustellen. Dazu verwenden sie Animationen, wie sie von Dessart und Kollegen (Dessart et al., 2011) vorgeschlagen werden. Diese animierten Übergänge sollen die Aufmerksamkeit des Nutzers auf die Bereiche der ABS lenken, die von den Anpassungen betroffen sind. Durch die Animation soll der Nutzer außerdem leichter verstehen können, dass und wie sich die neue Benutzungsschnittstelle aus einer Veränderung der vorherigen ABS-Instanz ergibt. Wenn beispielsweise eine Vergrößerung der Schriftelemente dazu führt, dass einige der bisher dargestellten Menüoptionen ausgeblendet werden müssen, kann eine Animation dem Nutzer verdeutlichen, dass die ausgeblendeten Optionen nun über einen „mehr“-Button zugänglich sind.

Adaptationsausführungsmuster betreffen verschiedene Ebenen und Aspekte der Benutzungsschnittstelle. Daher können Adaptationsausführungsmuster grundsätzlich zwei verschiedene Arten von Transitionen beschreiben:

- Wechsel von einem Interaktionsmuster zu einem anderen Interaktionsmuster desselben Bündels: Ein Beispiel wäre der Wechsel von einem einfachen textbasierten Menü in Listenform zu einem Kachelmenü mit überwiegend grafischer Beschriftung der Optionen. Adaptationsmuster dieses Typs werden *interaktionsspezifisch* genannt.
- Veränderung der Darstellungsparameter von Bedien- und Anzeigeelementen: Beispiele umfassen die Veränderung der Buttonbeschriftungen bezüglich Schriftgröße und Bild-Text-Verhältnis oder die Veränderung der Position oder Größe eines Buttons. Adaptationsmuster dieses Typs werden *generisch* genannt.

Für den ersten Fall ist jedem Bündel von Interaktionsmustern ein Adaptationsausführungsmuster zugeordnet, das „weiß“, welche Veränderungen es zwischen den Mustern eines Bündels geben kann und das die Transitionen der einzelnen Bedien- und Anzeigeelemente (zweiter Fall) in der richtigen Reihenfolge aufruft. Die generischen Adaptationsausführungsmuster des zweiten Falls können entweder durch interaktionsspezifische Adaptationsausführungsmuster aufgerufen werden oder direkt aufgrund von UIP-Veränderungen ausgelöst werden, die keinen Wechsel von Interaktionsmustern bedingen.

Adaptationsausführungsmuster können in den folgenden Relationen mit anderen Mustern stehen:

- *benötigt*
Diese Relation beschreibt die Beziehung von interaktionsspezifischen Adaptationsausführungsmustern zu generischen Adaptationsausführungsmustern. Die interaktionsspezifischen Adaptationsausführungsmuster begleiten den Wechsel zwischen Interaktionsmustern und verweisen dabei auf generische Adaptationsausführungsmuster,

die die Veränderung einzelner Darstellungsparameter von Bedien- und Anzeigeelementen beschreiben.

- `wird benötigt von`
Diese Relation ist die Umkehrrelation von `benötigt`.
- `benötigt <Variable> wie gesetzt von`
Diese Relation verbindet Adaptationsausführungsmuster mit Bündeln von Individualisierungsmustern. Zum Beispiel beziehen sich viele Adaptationsausführungsmuster in ihrer Lösungsbeschreibung auf die Variable `adaptationRenderingTime` des UIP, die die Geschwindigkeit der Adaptationsausführung bestimmt und deren Wert von Individualisierungsmustern des gleichnamigen Bündels gesetzt wird.

Darüber hinaus besteht ein weniger spezifischer Zusammenhang zwischen Adaptationsdialogmustern (s. Abschnitt 7.5) und Adaptationsausführungsmustern. Wird eine Laufzeitanpassung durchgeführt, so wird der Zeitpunkt der Adaptationsausführung vom jeweils aktiven Adaptationsdialogmuster bestimmt.

In Anlehnung an Dessart et al. (2011) decken die Adaptationsausführungsmuster die folgenden Animationsoperationen ab:

- Größenänderung (Resize)
Die Größe eines Bedien- und Anzeigeelements oder die Größe von Text- oder Bildelementen wird verändert.
- Positionsänderung (Relocate)
Die Position eines Bedien- und Anzeigeelements wird verändert.
- Einblenden/Ausblenden (Display/Undisplay)
Ein Bedien- und Anzeigeelement wird eingeblendet oder ausgeblendet.
- Ersetzen (Replace, widget transformation)
Eine Interaktionskomponente (Interaktionsmuster) wird durch eine andere Komponente desselben Bündels ersetzt. Dadurch werden andere Bedien- und Anzeigeelemente angezeigt.
- Aufteilen (Split/Distribute)
Ein Bedien- und Anzeigeelement oder eine Gruppe von Bedien- und Anzeigeelementen wird aufgeteilt und an getrennten Positionen dargestellt. Ein mögliches Beispiel ist die Auflösung eines „mehr“-Buttons in einzelne Menüeinträge, sobald alle angebotenen Optionen dargestellt werden können und der „mehr“-Button nicht mehr benötigt wird.

Tabelle 13 stellt eine Vorlage für die Beschreibung eines Adaptationsausführungsmusters dar. Für den Inhalt im Problemfeld wird zwischen interaktionsspezifischen und generischen Ausführungsmustern unterschieden. Da Adaptationsausführungsmuster keine Varianten besitzen, d.h. nicht in Bündel gruppiert sind, ist in dieser Vorlage keine Beschreibung des Kontexts vorgesehen.

| Adaptation rendering pattern | <Name des Adaptationsausführungsmusters> |
|-------------------------------|--|
| Problem | <p><Für interaktionsspezifische Adaptationsausführungsmuster: Beschreibung des von diesem Muster adressierten Gestaltungsproblems, das darin besteht, den Wechsel zwischen Interaktionsmustern eines bestimmten Bündels auszuführen></p> <p>'Interaction pattern E'</p> <p><Interaktionsmusterbündel, für das dieses Ausführungsmuster Adaptationen beschreibt></p> <p> <Für generische Adaptationsausführungsmuster: Beschreibung des von diesem Muster adressierten Gestaltungsproblems, das darin besteht, bestimmte Darstellungsparameter von Bedien- und Anzeigeelementen zu verändern></p> |
| Solution | <p>['requires'</p> <p><Auflistung der benötigten Adaptationsausführungsmuster in einer festgelegten Reihenfolge>]</p> <p><Beschreibung der Lösung des Musters; die Ausführung eines Anpassungsprozesses (interaktionsspezifisch) oder eines einzelnen Anpassungsschritts (generisch) wird in natürlicher Sprache spezifiziert. Handelt es sich um ein interaktionsspezifisches Adaptationsausführungsmuster, so werden die einzelnen Schritte der Anpassung durch Referenzierung der benötigten generischen Adaptationsausführungsmuster in der entsprechenden Reihenfolge aufgelistet.></p> |
| Code Reference | <p><Soweit möglich werden Teile der Lösung in einer programmierbaren Form angegeben. Hierfür kann z.B. ein technologieunabhängiger Pseudocode verwendet werden.></p> |
| Diagram | <p><Soweit möglich wird die Gestaltungslösung in einer schematischen Darstellung veranschaulicht.></p> |
| Rationale (references) | <p><Erläuterung bzw. Begründung der Lösung; falls vorhanden können Querverweise zu entsprechender Fachliteratur oder Normen und Standards angegeben werden.></p> |
| Ranking | <p><Bewertung der Reife des Musters aufgrund des Grads der Validierung auf einer dreistufigen Skala.> ::= '***' '**' '*'</p> |
| Required by | <p><Auflistung der Adaptationsausführungsmuster, die das Muster in ihrem Lösungsfeld referenzieren, um es in einem Adaptationsausführungsprozess zu verwenden, d.h. Auflistung der Muster, die in Relation benötigt mit dem beschriebenen Muster stehen; bei funktionsspezifischen Adaptationsanpassungsmustern bleibt dieses Feld leer.></p> |

Tabelle 13 Vorlage für die Beschreibung eines Adaptationsausführungsmusters

7.5 Adaptationsdialogmuster (Adaptation Dialogue Patterns)

Adaptationsdialogmuster beschreiben den Dialog zwischen Nutzer und ABS im Zuge einer Laufzeitanpassung. Dabei werden in der Regel drei Phasen unterschieden: (1) unmittelbar vor, (2) während und (3) nach der Anpassung. Die wesentlichen Module eines Adaptionsdialogmusters umfassen typischerweise eine Benachrichtigung des Nutzers über die Anpassung und Interaktionsoptionen, mit denen der Nutzer das Adaptionsverhalten des Systems beeinflussen kann.

Für Adaptionsdialogmuster ist nur eine Relation definiert:

- *Substituiert*
Diese Relation fasst Adaptionsdialogmuster zu Musterbündeln zusammen.

Darüber hinaus besteht eine enge, jedoch unspezifische Beziehung zu Adaptionsausführungsmustern. Alle Adaptionsdialogmuster referenzieren zur Durchführung der Adaptation auf der Darstellungsebene auf die Menge *AA* der Adaptionsausführungsmuster. Je nach Adaptionsdialogmuster erfolgt der Aufruf von *AA* zu unterschiedlichen Zeitpunkten im Adaptionsdialog. Die Auswahl der konkret zu verwendenden Ausführungsmuster hängt von den aktuell dargestellten Interaktionsmusterbündeln und damit von den auszuführenden ABS-Veränderungen ab.

Adaptionsdialogmuster zielen auf eine gesteigerte Usability und Akzeptanz von automatischen Laufzeitanpassungen, indem sie ...

- ... die *Transparenz* von Anpassungen erhöhen, d.h. sie machen Adaptationen für den Nutzer klar und verständlich; und
- ... die *Kontrollierbarkeit* von Anpassungen erhöhen, d.h. sie bieten dem Nutzer Einflussmöglichkeiten auf das Adaptionsverhalten des Systems.

Zur Steigerung der *Transparenz* können Adaptionsdialogmuster den Effekt der animierten Adaptionsausführungsmuster verstärken, indem sie durch textuelle oder ikonografische Hinweise explizit auf eine gerade ausgeführte Adaptation aufmerksam machen. Um zu vermeiden, den Bildschirm und die Aufmerksamkeit des Nutzers zu überladen und von der eigentlichen Interaktion abzulenken, beschränken sich die grafischen Adaptionshinweise auf einen definierten Bildschirmbereich, den „Adaptionsbereich“.

Für eine verbesserte *Kontrollierbarkeit* stehen grundsätzlich die folgenden Mechanismen zur Verfügung:

- Vor der Ausführung der Anpassung eine explizite Zustimmung des Nutzers einholen
- Nach der Ausführung der Anpassung eine explizite Zustimmung des Nutzers einholen
- Dem Nutzer eine Funktion zum Rückgängigmachen der bereits ausgeführten Anpassung anbieten (impliziter Kontrollmechanismus)

Damit ergibt sich für die MyUI-Adaptionsdialogmuster der in Abbildung 12 dargestellte Gestaltungsraum mit vier Entscheidungspunkten.

| Vor Anpassung | | Während Anpassung | | Nach Anpassung | | |
|-----------------------------------|---|-----------------------------------|---|------------------------------------|---|---|
| Explizite Zustimmung durch Nutzer | X | Grafische oder textuelle Hinweise | X | Explizite Bestätigung durch Nutzer | X | Möglichkeit, Anpassung rückgängig zu machen |
| keine explizite Zustimmung | | Keine Hinweise | | Keine explizite Bestätigung | | Keine „Rückgängig“-Funktion |

Abbildung 12 Gestaltungsraum der MyUI-Adaptationsdialogmuster

Es ist anzunehmen, dass explizite Bestätigungen ein höheres Maß an Nutzerkontrolle bewirken als der implizite Kontrollmechanismus über die Rückgängig-Funktion. Andererseits erfordern sie einen erhöhten Interaktionsaufwand durch einen zusätzlichen Interaktionsschritt, der für Nutzer mit Einschränkungen selbst wiederum Bedienprobleme mit sich bringen kann. Die Auswahl des am besten geeigneten Adaptionsdialogmusters kann damit als ein optimaler Kompromiss zwischen Transparenz und Beschränkung auf ein Minimum dargestellter Information sowie zwischen Kontrollierbarkeit und Interaktionseffizienz verstanden werden.

Die Steuerung der Laufzeitanpassungen erfolgt durch die MyUI-Adaptationsengine. Damit ist sie auch verantwortlich für die Auswahl des am besten geeigneten Adaptionsdialogmusters. Die MyUI-Adaptationsengine kann grundsätzlich die folgenden Kriterien in ihre Entscheidung einbeziehen:

- *Initiative der Anpassung*
Laufzeitanpassungen können vom System oder vom Nutzer initiiert werden. Dabei kommt es im Wesentlichen darauf an, ob die zugrundeliegenden Profiländerungen durch die automatische Kontexterkenkung der MyUI-Kontextmanagementinfrastruktur oder durch eine willentliche Konfiguration des Nutzers erfolgt sind (vgl. Abschnitt 7.6).
- *Schweregrad der Anpassung*
Für jede mögliche Adaptation wird ein Schweregrad ermittelt, der den angenommenen Effekt auf Nutzer und Interaktion reflektiert. Dazu wird jedem Paar von Interaktionsmustern eines Bündels und jedem adaptierbaren Parameter von Bedien- und Anzeigeelementen eine Schweregradschätzung auf einer dreistufigen Skala (niedrig, mittel, schwer) zugeordnet. Die resultierende Schweregradbewertung aller Änderungen einer Anpassung wird dann in die Entscheidung für ein Adaptionsdialogmuster herangezogen. So kann man sich beispielsweise vorstellen, dass geringfügige Anpassungen wie Leuchtdichteanpassungen aufgrund von veränderten Lichtverhältnissen keine speziellen Hinweise erfordern, während bei schwerwiegenden Anpassungen wie zum Beispiel dem Wechsel von manueller zu sprachgestützter Interaktion offensichtliche Hinweise und eine explizite Nutzerstimmung angemessen sind.
- *Individuelle Nutzereigenschaften*
Möglicherweise geben Nutzereigenschaften, wie sie im MyUI-Nutzerprofil gespeichert sind, Hinweise auf individuelle Präferenzen für einen bestimmten Anpassungsdialog. Interne Nutzerstudien des Fraunhofer IAO deuten beispielsweise daraufhin, dass Nutzer mit geringer Technikaffinität häufig einen Interaktionsstil mit mehr Bestätigungen und expliziten Rückmeldungen bevorzugen.

Im Gegensatz zu den anderen Entwurfsmustertypen, sind die Adaptionsdialogmuster generischer Bestandteil des MyUI-Rahmenmodells und damit nicht abhängig von den

implementierten Applikationen oder den jeweils adressierten besonderen Nutzerbedürfnissen. Daher werden die Adaptationsdialogmuster in den folgenden Abschnitten dargestellt. Auf die Darstellung der verwendeten Vorlage wird deshalb verzichtet.

7.5.1 Automatische Adaptation mit impliziter Nutzerzustimmung

| Vor Anpassung | Während Anpassung | Nach Anpassung | |
|-----------------------------------|-----------------------------------|------------------------------------|---|
| Explizite Zustimmung durch Nutzer | Grafische oder textuelle Hinweise | Explizite Bestätigung durch Nutzer | Möglichkeit, Anpassung rückgängig zu machen |
| keine explizite Zustimmung | Keine Hinweise | Keine explizite Bestätigung | Keine „Rückgängig“-Funktion |

Abbildung 13 Transparenz- und Kontrollmechanismen bei der automatischen Adaptation mit impliziter Nutzerzustimmung

Die Adaptation wird automatisch gestartet, ohne zuvor eine Nutzerzustimmung einzuholen. Während die Adaptation ausgeführt wird (Adaptationsausführungsmuster), wird in einem definierten Adaptationsbereich des Bildschirms ein animiertes Icon angezeigt, um den Nutzer darauf hinzuweisen, dass gerade eine Anpassung stattfindet. Die Geschwindigkeit und damit die Dauer der Adaptationsausführung wird über die UIP-Variable *adaptationRenderingTime* an die Anforderungen des Nutzers angepasst. Sobald die Adaptation vollständig ausgeführt worden ist, wird im Adaptationsbereich ein Button angeboten, mit dem der Nutzer die Adaptation rückgängig machen kann (s. Abbildung 13 und Abbildung 14). Der Rückgängig-Button bleibt danach erhalten, bis ein Übergang in einen anderen Interaktionszustand der Applikation ausgelöst wird. So kann der Nutzer zu jedem Zeitpunkt zur vorhergehenden ABS-Konfiguration zurückgehen. In Abbildung 15 ist eine Sequenz von Bildschirmansichten vor, während und nach der Adaptation dargestellt.

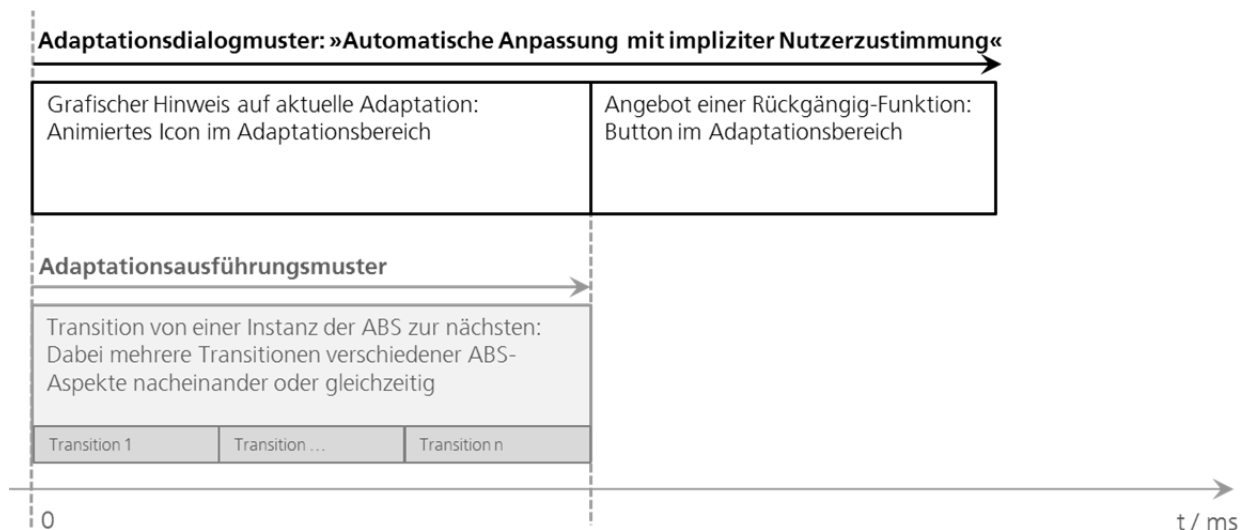


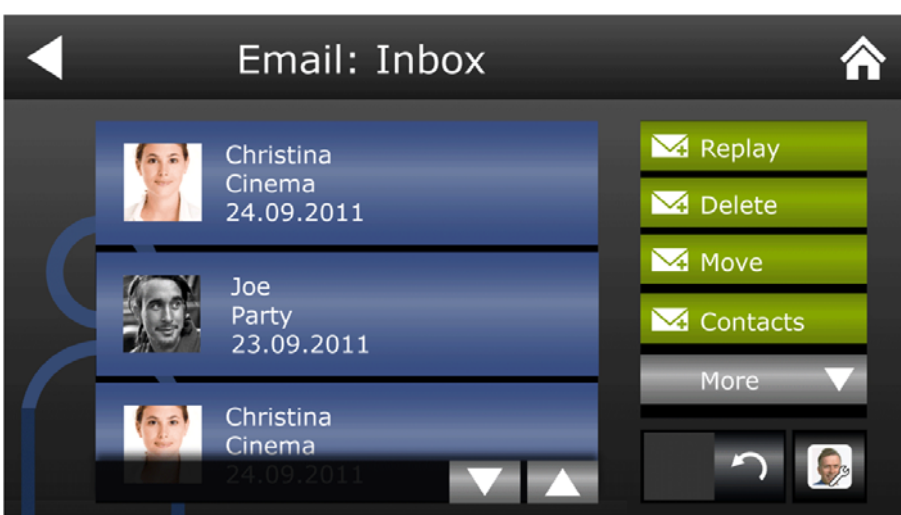
Abbildung 14 Ablauf einer automatischen Adaptation mit impliziter Nutzerzustimmung



(1) Vor der Adaptation:
 Rechts unten befindet sich der Adaptationsbereich mit einem immer verfügbaren Button zum Zugang zu UP und UIP.



(2) Während der Adaptation:
 Ein pulsierendes Icon (hier ein Chamäleon) weist darauf hin, dass gerade eine Adaptation ausgeführt wird.



(3) Nach der Adaptation:
 Der Nutzer kann die Adaptation über den Button mit dem geschwungenen Rückwärts Pfeil rückgängig machen.

Abbildung 15 Bildschirmsequenz während einer automatischen Adaptation mit impliziter Nutzerzustimmung

7.5.2 Explizite Nutzerbestätigung vor der Adaptation

| Vor Anpassung | Während Anpassung | Nach Anpassung | |
|-----------------------------------|-----------------------------------|------------------------------------|---|
| Explizite Zustimmung durch Nutzer | Grafische oder textuelle Hinweise | Explizite Bestätigung durch Nutzer | Möglichkeit, Anpassung rückgängig zu machen |
| keine explizite Zustimmung | Keine Hinweise | Keine explizite Bestätigung | Keine „Rückgängig“-Funktion |

Abbildung 16 Transparenz- und Kontrollmechanismen bei der expliziten Nutzerbestätigung vor der Adaptation

Bevor die Adaptation ausgeführt wird, wird der Nutzer in einem modalen Dialogfenster aufgefordert, der Anpassung zuzustimmen oder die Anpassung abzulehnen. Der Ausschnitt einer Vorschau der ABS nach der Adaptation illustriert den Effekt der intendierten Adaptation und erleichtert so die Nutzerentscheidung. Wenn der Nutzer die Adaptation ablehnt, wird das Dialogfenster wieder geschlossen und die ABS wird nicht verändert. Akzeptiert der Nutzer die Adaptation oder wird keine Nutzerreaktion registriert (Time-Out²⁵), so wird das Dialogfenster geschlossen und die Adaptation wird ausgeführt. Darüber hinaus wird die MyUI-Kontextmanagementinfrastruktur über die Nutzerentscheidung informiert (vgl. Abschnitt 7.2). Über einen Button im Adaptationsbereich kann der Nutzer die Adaptation rückgängig machen, solange er im selben Zustand der Applikation bleibt (Abbildung 16). Abbildung 17 stellt den Ablauf des Musters dar. In Abbildung 18 ist ein Beispiel eines Dialogfensters zur expliziten Nutzerbestätigung dargestellt.

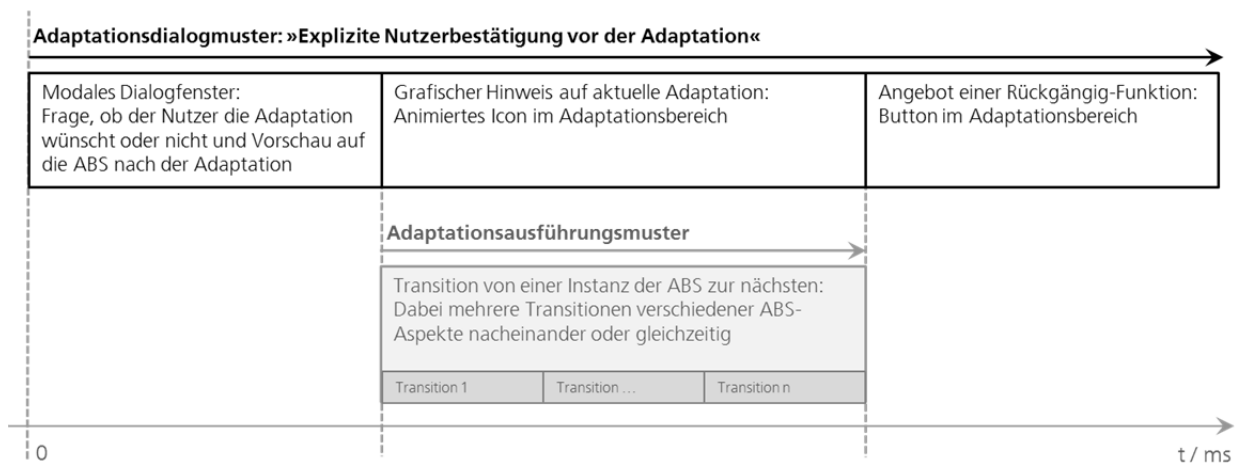


Abbildung 17 Ablauf des Adaptationsdialogmusters „Explizite Nutzerbestätigung vor der Adaptation“

²⁵ Das Time-Out-Intervall wird über die UIP-Variable *timeOut* an die individuellen Anforderungen angepasst.

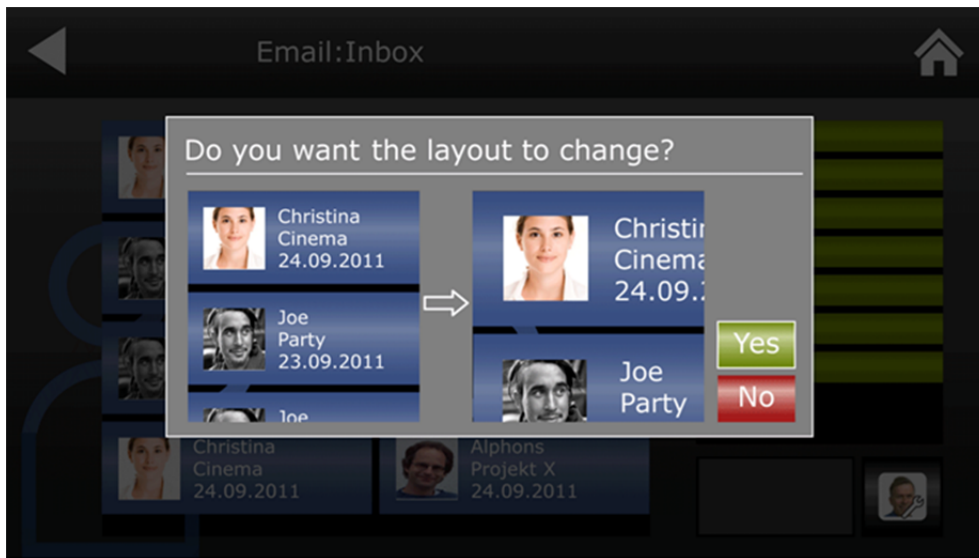


Abbildung 18 Adaptionsdialogmuster „Explizite Nutzerbestätigung vor der Adaptation“: Überblendung der aktuellen Benutzungsschnittstelle durch ein Dialogfenster mit Vorschau der geplanten Anpassung

7.5.3 Explizite Nutzerbestätigung nach der Adaptation

| Vor Anpassung | | Während Anpassung | | Nach Anpassung | |
|-----------------------------------|---|-----------------------------------|---|------------------------------------|---|
| Explizite Zustimmung durch Nutzer | X | Grafische oder textuelle Hinweise | X | Explizite Bestätigung durch Nutzer | Möglichkeit, Anpassung rückgängig zu machen |
| keine explizite Zustimmung | | Keine Hinweise | | Keine explizite Bestätigung | Keine „Rückgängig“-Funktion |

Abbildung 19 Transparenz- und Kontrollmechanismen bei der expliziten Nutzerbestätigung nach der Adaptation

Das Adaptionsdialogmuster „Explizite Bestätigung nach der Adaptation“ umfasst die in Abbildung 19 dargestellten Transparenz- und Kontrollmechanismen. Nach der selbstständig durchgeführten Adaptation wird die angepasste ABS zunächst für die Dauer eines Time-Out-Intervalls²⁶ dargestellt. In dieser Zeit hat der Nutzer bereits die Möglichkeit, über einen Button im Adaptionsbereich die Anpassung rückgängig zu machen. Tut er das nicht, wird nach Ablauf des Time-Outs ein modales Dialogfenster eingeblendet, das den Nutzer fragt, ob er die neuen Veränderungen beibehalten oder verwerfen möchte. Im Fall der Ablehnung wird die Adaptation rückgängig gemacht. Bei Akzeptanz oder bei Nichtreaktion innerhalb eines Time-Out-Intervalls wird die Adaptation beibehalten. In jedem Fall wird das Dialogfenster wieder geschlossen und die MyUI-Kontextmanagementinfrastruktur über die Nutzerentscheidung informiert (vgl. Abschnitt 7.2). Auch nach Schließen des Dialogfensters bleibt die Rückgängig-Option weiterhin erhalten, bis der Nutzer in einen anderen Zustand der Applikation wechselt. Der Ablauf des Musters ist in Abbildung 20 dargestellt.

²⁶ Das Time-Out-Intervall wird über die UIP-Variable *timeOut* an die individuellen Anforderungen angepasst.

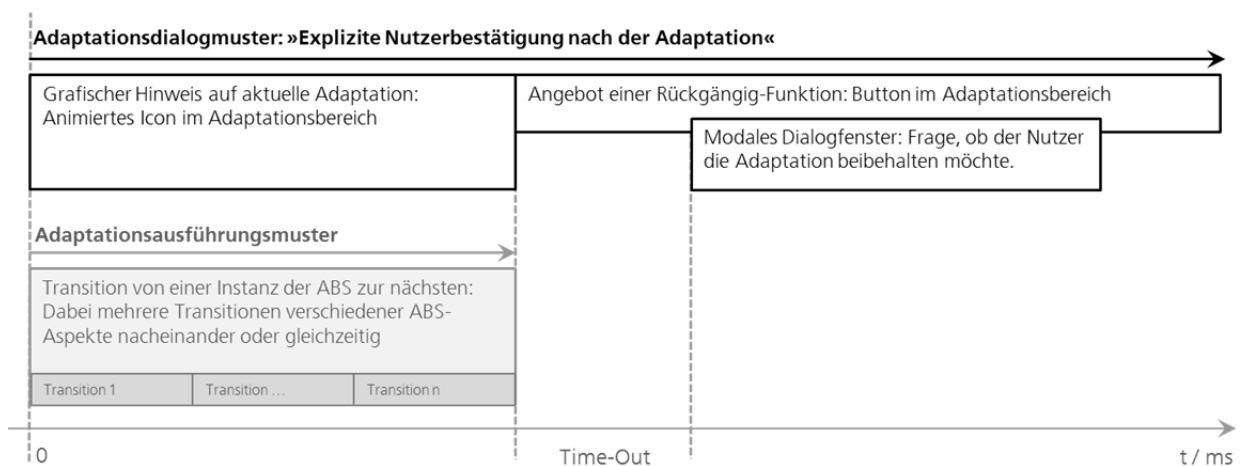


Abbildung 20 Ablauf des Adaptationsdialogmusters „Explizite Nutzerbestätigung nach der Adaptation“

7.6 Mechanismen zur nutzerinitiierten Anpassung der adaptiven Benutzungsschnittstelle

Vom System initiierte automatische Anpassungen der ABS sind ein wichtiges Merkmal des MyUI-Ansatzes. Daneben bietet das MyUI-System dem Nutzer auch die Möglichkeit, auf eigene Initiative Veränderungen der ABS zu bewirken. Das Rahmenmodell umfasst hierfür Konfigurationsdialoge zur Manipulation der UP-Variablen und zur Manipulation der UIP-Variablen. Dadurch wird das MyUI-System um das Prinzip der Adaptierbarkeit erweitert.

Der Zugang zu den beiden Profilen steht dem Nutzer zu jedem Zeitpunkt zur Verfügung. Über einen permanent dargestellten Button im Adaptationsbereich (vgl. Bildschirmdarstellungen in Abbildung 15) kann ein Konfigurationsmenü geöffnet werden, das den Zugang zum UP und zum UIP ermöglicht. Zunächst bearbeitet der Nutzer jeweils nur eine Repräsentation des UP bzw. des UIP. Erst nach Konsistenzprüfungen werden die vom Nutzer vorgenommenen Änderungen in den tatsächlichen Profilen der MyUI-Kontextmanagementinfrastruktur realisiert.

7.6.1 Nutzerinitiierte Anpassungen über das UP

Der permanent angebotene Zugang zum UP stärkt einerseits die Transparenz des MyUI-Systems. Der Nutzer kann alle über ihn gespeicherten Daten und Annahmen einsehen. Andererseits kann der Nutzer selbst die Variablenwerte des UP verändern. Dies kann dazu genutzt werden, um zu Beginn der MyUI-Nutzung eine gute Basis für die Generierung einer möglichst passgenauen ABS einzustellen und damit dem teilweise länger dauernden Prozess der allmählichen Anpassung vorzugreifen oder später, um Veränderungen der individuellen Präferenzen abzubilden und fehlerhafte Annahmen zu korrigieren.

Nutzerinitiierte Veränderungen des UP werden direkt der MyUI-Kontextmanagementinfrastruktur übermittelt. Genauso wie bei UP-Änderungen, die durch Sensorereignisse ausgelöst werden, wird dadurch der Prozess der ABS-Parametrierung neu gestartet und das UIP aktualisiert, was dann gegebenenfalls zu einer Adaptation der ABS führt. Der Prozess einer nutzerinitiierten Anpassung über das UP kann in die folgenden Schritte unterteilt werden:

1. Der Nutzer verändert einen oder mehrere Werte in der ABS-Repräsentation des UP.

2. Die MyUI-Kontextmanagementinfrastruktur wird über die vom Nutzer eingestellten Änderungen informiert und verändert das UP entsprechend.
3. Die UP-Änderung löst eine erneute ABS-Parametrierung aus, die zu einer Aktualisierung des UIP führt.
4. Die Veränderungen im UIP führen zu einer erneuten ABS-Vorbereitung und einer erneuten Generierung und Adaptation der ABS, die sich schließlich in einer veränderten Benutzungsschnittstelle niederschlagen.

7.6.2 Nutzerinitiierte Anpassungen über das UIP

Die zweite Möglichkeit für den Nutzer, Veränderungen an der ABS zu bewirken, besteht darin, die ABS-Einstellungen direkt über die Variablen des UIP zu bearbeiten. Hierfür steht in der ABS permanent eine Übersicht der UIP-Einstellungen zur Verfügung, die der Nutzer verändern kann.

Nutzerinitiierte UIP-Veränderungen setzen einen komplexen Prozess in Gang, der im Wesentlichen darauf abzielt, eine gegenseitige Konsistenz zwischen dem UP und dem UIP sicher zu stellen. Es wird angenommen, dass Nutzer das UIP so verändern, dass es ihren individuellen Bedürfnissen und Präferenzen besser entspricht. So können UIP-Veränderungen genutzt werden, um auch das UP anzupassen – und zwar so, dass es konsistent zum neuen UIP ist. Jedoch können bestimmte UIP-Veränderungen dazu führen, dass es unmöglich ist, das UP dazu in Einklang zu bringen. Diese Konflikte sind teilweise nicht eindeutig lösbar und können daher Klärungsdialoge erfordern, um den Nutzer einzubeziehen. Der Prozess einer nutzerinitiierten Anpassung über das UIP hat die folgenden Schritte:

1. Der Nutzer verändert einen oder mehrere Variablenwerte in der ABS-Repräsentation des UIP.
2. Durch eine umgekehrte Anwendung der Individualisierungsmuster wird ein UP identifiziert, das durch möglichst geringfügige Modifikationen des aktuellen UP eine Konsistenz zu der vom Nutzer geänderten UIP-Repräsentation herstellen kann.
3. Wenn dies nicht möglich ist, schlägt das System dem Nutzer weitere UIP-Änderungen vor. Am Ende dieses interaktiven Verhandlungsprozesses steht ein neues UIP, das vom Nutzer bestätigt wurde und zu dem ein konsistentes UP erstellt werden kann.
4. Die UP-Veränderungen werden an die Kontextmanagementinfrastruktur kommuniziert und das UP wird entsprechend aktualisiert.
5. Die Veränderungen im UP lösen erneut die Schritte der ABS-Vorbereitung und der ABS-Generierung und Adaptation aus, was schließlich in einer adaptierten Benutzungsschnittstelle resultiert.

8 Das Abstrakte Applikations-Interaktions-Modell (AAIM) zur Modellierung der adaptiven Benutzungsschnittstelle

Der dreistufige Prozess zur ABS-Generierung und Adaptation verläuft automatisch. Damit verändert sich die Rolle des Applikationsentwicklers signifikant. Sein Einfluss auf die spezifische Ausprägung und Gestaltung der ABS ist minimal. Dafür übernimmt ein automatisierter und regelbasierter Prozess die dynamische Erstellung von individualisierten Benutzungsschnittstellen für unterschiedliche Nutzerbedürfnisse und Kontextbedingungen. Der Entwickler definiert die Interaktion zwischen Nutzer und Applikation lediglich auf einer Ebene, die vom konkreten Aussehen und den konkreten Interaktionsmechanismen der Benutzungsschnittstelle unabhängig ist. Hierfür wurde im Rahmen dieser Arbeit ein eigenes Format entwickelt: das „Abstrakte Applikations-Interaktions-Modell“ (AAIM). Das AAIM dient als Basis der ABS-Generierung und Adaptation, in dem es die Gemeinsamkeiten aller möglichen ABS-Varianten definiert.

8.1 Zielsetzung und Rolle des AAIM

Das AAIM erfüllt die folgenden Funktionen im MyUI-Rahmenmodell (s. Abbildung 21):

- *Schnittstelle zwischen Entwickler und MyUI-Infrastruktur*
Das AAIM ist das einzige notwendige Artefakt, das der Entwickler einer neuen ABS erstellen muss. Damit ist das AAIM der zentrale Kontaktpunkt des Entwicklers zur MyUI-Infrastruktur mit ihrem Entwurfsmusterrepositorium. Um eine möglichst weite Verbreitung von MyUI-Technologien zu ermöglichen, muss insbesondere eine einfache und effiziente AAIM-Erstellung unterstützt werden.
- *Grundlage der Generierung und Adaptation der ABS*
Das AAIM ist der Ausgangspunkt der ABS Generierung. Für jeden Zustand einer MyUI-Applikation definiert das AAIM die Interaktionssituationen, die dem Nutzer präsentiert werden. Interaktionssituationen repräsentieren die Interaktionsoptionen des Nutzers zu einem bestimmten Zeitpunkt in der Applikation, z.B. eine Liste von Elementen ansehen, bestimmte Funktionen auslösen, zu einem anderen Applikationsbereich navigieren, etc. Jede Interaktionssituation ist mit einem Bündel von Interaktionsmustern verknüpft. Dadurch stellt das AAIM die zentrale Basis für die ABS-Vorbereitung dar, bei der die am besten geeigneten Interaktionskomponenten und Bedien- und Anzeigeelemente ausgewählt werden.
- *Schnittstelle zwischen ABS und Applikationsfunktionen*
Im AAIM wird spezifiziert, welche Applikationsfunktionen in Abhängigkeit von bestimmten Ereignissen und Zuständen der ABS aufgerufen werden. Applikationsfunktionen können Systemvariablen setzen, die Applikationsdatenbasis manipulieren oder Transaktionen ausführen, wie z.B. eine E-Mail versenden. Damit knüpft das AAIM Beziehungen zwischen der ABS (der View im MVC-Modell der Gesamtapplikation) und den Applikationsfunktionen, die im Backend des Systems ausgeführt werden (Controller).
- *Schnittstelle zwischen ABS und Applikationsdaten*
Die meisten Bedien- und Anzeigeelemente beziehen sich auf Applikationsinhalte. So

erfordert beispielsweise eine Auswahlliste bestimmte Inhaltselemente, aus denen der Nutzer wählen kann. Diese Inhaltselemente stammen üblicherweise aus dem Applikationsdatenbestand. Im AAIM werden die Beziehungen zwischen Interaktionssituationen und Applikationsdaten definiert, indem eine Interaktionssituation über eine Datenakquisitionsfunktion mit einer Datenbasis verbunden wird, aus der die darzustellende Interaktionskomponente ihren Applikationsinhalt bezieht. So verbindet das AAIM die ABS (View) über Datenakquisitionsfunktionen (Controller) mit der Applikationsdatenbasis (Model).

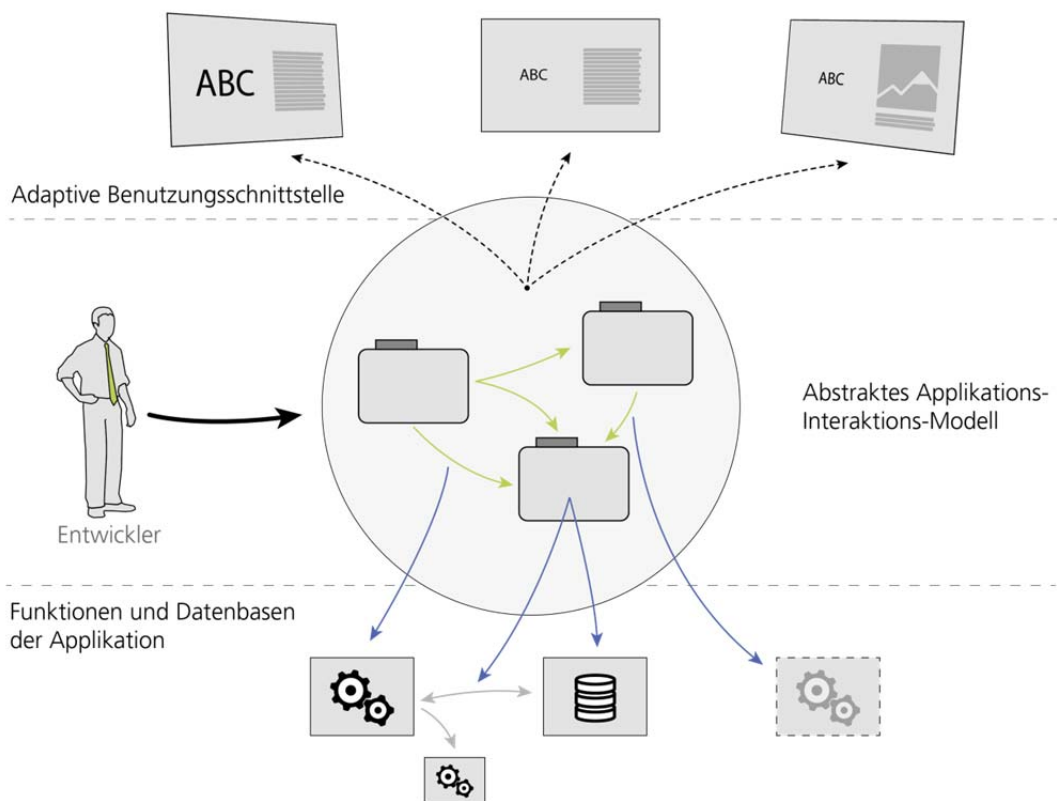


Abbildung 21 Rolle des AAIM im MyUI-Rahmenmodell

8.2 Zustandsdiagramme zur abstrakten Modellierung der Benutzungsschnittstelle

Das AAIM erweitert das Zustandsdiagramm der UML 2 (vgl. OMG, 2011 oder Jeckle et al., 2004). Mit Zustandsdiagrammen kann eine Benutzungsschnittstelle modelliert werden, ohne Aussagen über Darstellungsformate oder verwendete Bedienelemente zu treffen (vgl. Horrocks, 1999; Barnett et al., 2012). Diese Aspekte sind im Ansatz dieser Arbeit Gegenstand von Adaptationen und dürfen daher im AAIM nicht festgelegt sein.

Jeder einzelne Zustand des AAIM-Zustandsdiagramms repräsentiert einen bestimmten Status der ABS und die Interaktionsoptionen, die sich dem Nutzer bieten. Zu jedem Zeitpunkt ist die ABS in genau einem Zustand des AAIM. Des Weiteren gelten alle Eigenschaften der UML 2 Statecharts auch für das AAIM. Die folgende Darstellung konzentriert sich daher auf die Erweiterungen, die für diese Arbeit vorgenommen wurden, um den besonderen Anforderungen der ABS-

Modellierung gerecht zu werden. Dabei konzentrieren sich die vorgenommenen Erweiterungen im Wesentlichen auf das Konzept der *Interaktionssituation* und dessen Behandlung und Notation im AAIM.

8.3 Interaktionssituationen als Grundlage von Adaptationen

Das Konzept der Interaktionssituation (IS) ist ein Kernprinzip des MyUI-Adaptationsansatzes. Interaktionssituationen repräsentieren die Interaktionsmöglichkeiten, die der Nutzer an einem bestimmten Punkt in der Applikation hat. Dabei umfassen Interaktionsmöglichkeiten alle Aktivitäten, die der Nutzer an der Benutzungsschnittstelle ausführen kann wie zum Beispiel Informationen wahrnehmen, Eingaben tätigen, eine Option auswählen, usw. Eine bestimmte IS kann durch mehrere verschiedene Bedien- und Anzeigekomponenten realisiert werden. So kann zum Beispiel das Auswählen einer Option aus einer Menge von Optionen durch eine Auswahlliste oder durch ein Drop-Down-Menü oder durch ein Audiomenu mit Sprachbefehl realisiert werden. Diese Flexibilität stellt die wesentliche Grundlage für Adaptationen der Benutzungsschnittstelle dar. Je nach individuellen Nutzerbedürfnissen und Kontextbedingungen wird die am besten geeignete Bedien- und Anzeigekomponente für die aktuelle IS ausgewählt. Die verschiedenen Varianten von Bedien- und Anzeigekomponenten für ein und denselben Zweck (d.h. für ein und dieselbe IS) entsprechen einem Bündel von Interaktionsmustern (s. Abschnitt 6.1). So ist eine IS eine abstrakte Oberklasse von Bedien- und Anzeigekomponenten, die demselben Zweck dienen, und damit gleichbedeutend mit einem Bündel von Interaktionsmustern.

Trotz dieser 1:1-Beziehung zwischen Interaktionssituationen und Interaktionsmusterbündel ist eine Unterscheidung der beiden Konzepte sinnvoll, da sie unterschiedliche Betrachtungsperspektiven einnehmen. Dies drückt sich auch in den unterschiedlichen Benamungskonventionen aus. Während Interaktionsmuster aus der Systemperspektive benannt werden und typische Bezeichnungen von Komponenten einer Benutzungsschnittstelle erhalten, werden IS aus der Nutzerperspektive benannt. Namen von IS beginnen daher immer mit einem Verb, um die gerade möglichen Nutzeraktionen auszudrücken. Während ein Interaktionsmusterbündel die Menge der möglichen technischen Lösungen für ein bestimmtes „Interaktionsproblem“ (eine IS) zusammenfasst, beschreibt die IS dieses Problem aus der Perspektive der Nutzung bzw. der Interaktion, ohne bereits einen Bezug zu den möglichen Lösungen aufzunehmen. Bei der Modellierung einer Applikation in einem AAIM ist genau diese lösungsunabhängige Perspektive der IS gefragt, die den Nutzer und den Interaktionszweck in den Mittelpunkt stellt.

Um die wichtigsten Interaktionen, die in vielen Mensch-Technik Schnittstellen für verschiedene Applikationen vorkommen, abzudecken, umfasst das MyUI-Rahmenmodell einige generische Interaktionssituationen (s. Tabelle 14). Wie die Entwurfsmuster werden auch die Interaktionssituationen im MyUI-Repository vorgehalten. Die vom Rahmenmodell zur Verfügung gestellten Interaktionssituationen sind dabei lediglich ein Ausgangspunkt, der die Mechanismen des Zusammenspiels von IS und Interaktionsmuster veranschaulicht und einen schnellen Start mit der MyUI-Infrastruktur ermöglichen soll. Erweiterungen der Menge der Interaktionssituationen sind explizit vorgesehen.

| Interaktionssituation | Beschreibung |
|--|--|
| EditForm | Der Nutzer kann Informationen in einem Formular eingeben, das verschiedene Eingabe- und Auswahlelemente beinhalten kann. |
| MetaGoTo | Der Nutzer kann zu jeder Zeit (in jedem Zustand) in einen anderen Bereich bzw. einen anderen Zustand der Applikation wechseln. |
| NavigateTree | Der Nutzer erhält einen Überblick über eine hierarchisch strukturierte Menge von Elementen und kann aus dieser Menge ein Element auswählen. |
| ProvideFunctionAttributes | Der Nutzer kann für eine zuvor ausgewählte Funktion die zur Ausführung notwendigen Parameter angeben. |
| ProvideFunctionConfirmation | Der Nutzer kann eine zuvor ausgewählte Funktion bestätigen oder widerrufen, bevor die Funktion ausgeführt wird. |
| SelectFunction | Der Nutzer kann eine Funktion aus einer angebotenen Menge von Funktionen auswählen. |
| SelectGoTo | Der Nutzer kann in einen anderen Bereich bzw. einen anderen Zustand der Applikation wechseln. |
| SelectItem | Der Nutzer kann ein Element aus einer Menge von Elementen auswählen. |
| SelectItemWithAttributes | Der Nutzer kann ein Element aus einer Menge von Elementen auswählen. Um die Auswahl zu unterstützen, wird zu jedem Element neben einer ID oder Benennung eine Menge von Attributen dargestellt. |
| SelectMultipleItemsWithAttributes | Der Nutzer kann mehrere Elemente aus einer Menge von Elementen auswählen. Um die Auswahl zu unterstützen, wird zu jedem Element neben einer ID oder Benennung eine Menge von Attributen dargestellt. |
| SelectService | Der Nutzer kann verschiedene Applikationen auswählen und starten. Diese IS entspricht einem Hauptmenü, das mehreren Applikationen übergeordnet ist. |
| ViewGroupOfItemsWithAttributes | Der Nutzer kann eine Menge von Elementen ansehen. Zu jedem Element wird neben einer ID oder Benennung eine Menge von Attributen dargestellt. |
| ViewItemWithAttributes | Der Nutzer kann ein Element und dessen Attribute ansehen. |

Tabelle 14 Das ABS-Rahmenmodell stellt einige generische Interaktionssituationen zur Verfügung (alphabetische Reihenfolge)

Am Beispiel der IS EditForm wird deutlich, dass Interaktionssituationen zur konkreten Anwendung in einer Applikation eine gewisse Parametrierung benötigen. So muss für ein Formular beispielsweise angegeben werden, welche Eingabefelder angeboten werden und wie diese typisiert sind. Das bedeutet, dass zu jedem Feld eine ID und ein Wertebereich der zulässigen Eingaben angegeben werden müssen. Diese Parameter entsprechen den Eingangsparametern der Interaktionsmusterbündel (s. Abschnitt 6.1). Darüber hinaus repräsentieren die Ausgangsparameter eines Interaktionsmusterbündels die möglichen Ergebnisse der Interaktion in der entsprechenden IS. Für die IS EditForm sind diese beispielsweise die Nutzereingaben in den einzelnen Formularfeldern und die Angabe, ob die Eingaben vollständig sind und vom Nutzer bestätigt oder verworfen wurden. In der IS SelectFunction besteht das Interaktionsergebnis hingegen in der Auswahl einer bestimmten Applikationsfunktion, die dann ausgelöst wird. Das Interaktionsergebnis einer IS wird in Form eines Parametervektors geführt und kann so an folgende Interaktionssituationen und die Applikation weitergegeben werden. Die Eingangs- und Ausgangsparameter definieren damit die Schnittstellen für den Datenaustausch zwischen den konkreten Bedienelementen der ABS und der Applikationslogik.

8.4 Interaktionssituationen in Zuständen

In der UML 2 besitzt ein einfacher Zustand die Bestandteile Name, interne Aktivitäten und interne Transitionen (vgl. OMG, 2011). Zusätzlich zu diesen Eigenschaften enthält ein Zustand im AAIM eine Menge von Interaktionssituationen (vgl. Abbildung 22). Für Zustände im AAIM sind Name und Angabe der Interaktionssituationen obligatorisch. Darüber hinaus stehen alle Ausdrucksmöglichkeiten der UML 2 wie interne Aktivitäten oder Transitionen zur Verfügung. Es gibt vier verschiedene Typen von Interaktionssituationen, die in Zuständen enthalten sein können:

- *Primäre Interaktionssituation (PIS)*
Jeder einfache Zustand des AAIM hat genau eine PIS. Diese PIS repräsentiert den Hauptzweck des Zustands. Alle Interaktionssituationen des MyUI-Repositoriums können als PIS verwendet werden. Die einzigen Ausnahmen stellen die beiden unten beschriebenen Interaktionssituationen *SelectFunction* und *SelectGoTo* dar. Die PIS ist in der Regel mit einem spezifischen Objekt assoziiert, das auf der ABS dargestellt wird, um vom Nutzer betrachtet oder manipuliert zu werden. Dieses Objekt wird „Hauptobjekt“ des jeweiligen Zustands genannt.
- *Interaktionssituation SelectFunction*
Zusätzlich zur PIS kann ein Zustand optional die Interaktionssituation *SelectFunction* besitzen. Diese spezifiziert eine Menge von Applikationsfunktionen, die der Nutzer im aktuellen Zustand auslösen kann. Die Applikationsfunktionen, die in der *SelectFunction* referenziert werden, werden auf das Hauptobjekt des aktuellen Zustands angewandt und erweitern so die von der PIS angebotene Funktionalität.
- *Interaktionssituation SelectGoTo*
Zusätzlich zur PIS kann ein Zustand optional die Interaktionssituation *SelectGoTo* besitzen. Diese spezifiziert eine Menge von Zielzuständen, zu denen der Nutzer navigieren kann. Im Gegensatz zur *SelectFunction*, die häufig auch einen Zustandswechsel auslöst, bewirkt die *SelectGoTo* einen Übergang zu einem anderen Zustand, ohne den Datenbestand der Applikation zu verändern oder eine andere Transaktion auszulösen.
- *Interaktionssituation MetaGoTo*
Die Interaktionssituation *MetaGoTo* ist ein Spezialfall der *SelectGoTo*. Sie kann nur auf den Hauptzustand einer Applikation angewendet werden. Die *MetaGoTo* definiert die generischen Navigationsfunktionen einer Applikation, die in allen Zuständen konsistent in der gleichen Weise zur Verfügung stehen sollen. Beispiele umfassen „Zurück zum letzten Zustand“ oder „Navigation zum Hauptzustand (Home)“.

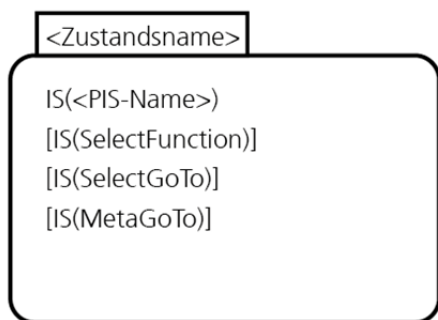


Abbildung 22 Einfacher Zustand im AAIM

Eine IS gilt jeweils für den Zustand, in dem sie enthalten ist. Neben Interaktionssituationen können Zustände alle in der UML 2 üblichen Argumente enthalten, um eine präzise Definition des Interaktionsverhaltens zu modellieren. So kann zum Beispiel eine interne Aktivität definiert werden, um Variablenwerte zu setzen, die im weiteren AAIM genutzt werden.

Zusammengesetzte Zustände enthalten keine PIS. Sie können aber alle drei optionalen Interaktionssituationen enthalten. Die Interaktionssituationen eines zusammengesetzten Zustands gelten für alle in ihm enthaltenen Zustände und bestehen, solange ein Unterzustand des zusammengesetzten Zustands aktiv ist. Dieses Prinzip kann zum Beispiel für die Definition übergeordneter Navigationsoptionen, die in der gesamten Applikation zur Verfügung stehen sollen, verwendet werden (IS MetaGoTo).

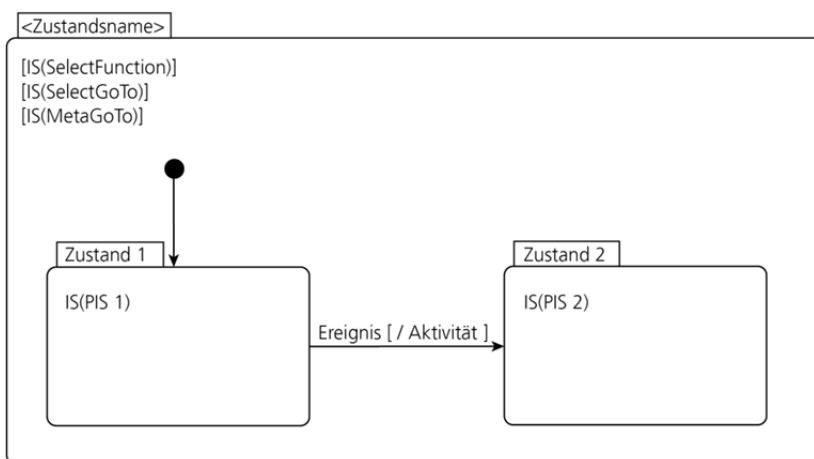


Abbildung 23 Zusammengesetzter Zustand im AAIM

Eine IS eines Zustands kann Übergänge zu anderen Zuständen auslösen. So wird beispielsweise eine IS, die dem Nutzer eine Menge von Funktionen zur Wahl stellt, in der Regel mit einer Menge von Transitionen zu anderen Zuständen, in dem die ausgewählten Funktionen ausgeführt werden, verbunden sein.

8.5 Interaktionssituationen an Zustandsübergängen

Zustandsübergänge sind im AAIM genau wie bei den UML Zustandsdiagrammen durch einen Ausgangs- und einen Zielzustand definiert. Übergänge werden durch ein Ereignis ausgelöst, das in der Regel einer Nutzerinteraktion entspricht. Gemeinsam mit dem Zustandsübergang kann eine zusätzliche Aktivität ausgelöst werden. Damit umfasst die Beschriftung eines Zustandsübergangs ein Ereignisargument und ein optionales Aktivitätsargument (s. Darstellung der Transition von Zustand 1 zu Zustand 2 in Abbildung 23).

Im AAIM werden Interaktionssituationen in Zuständen und als Argument von Transitionen zwischen Zuständen des AAIM verwendet. Eine IS eines Zustands wird üblicherweise auch an einer Transition referenziert, um anzugeben, welche Transition als Folge der Interaktion mit der IS genommen wird. Darüber hinaus können im AAIM bestimmte Interaktionssituationen auch an einer Transition dargestellt werden, ohne in einem Zustand verwendet zu werden. Diese Interaktionssituationen, die nur an einer Transition referenziert werden, werden genutzt, um Bestätigungen oder zusätzliche Informationen vom Nutzer einzuholen, bevor die Transition

ausgeführt werden kann und der nächste Zustand angesteuert werden kann. So erfordert beispielsweise die Funktion „verschiebe Datei in Ordner“ (als Ergebnis der IS SelectFunction des vorausgegangenen Zustands) die Angabe eines Zielordners. Oder die Ausführung einer irreversiblen Funktion erfordert eine explizite Nutzerbestätigung. Diese zusätzlich benötigten Argumente werden durch eine eigene IS vom Nutzer eingeholt, zum Beispiel durch ein Dialogfenster. Diese Klasse von IS kann neben den in Abschnitt 8.4 beschriebenen Interaktionssituationen als fünfter IS-Typ unterschieden werden und wird im Folgenden *Erweiterung einer IS* (ISE) genannt, da sie in der Regel der weiteren Klärung einer Interaktionssequenz mit einer anderen IS dient.

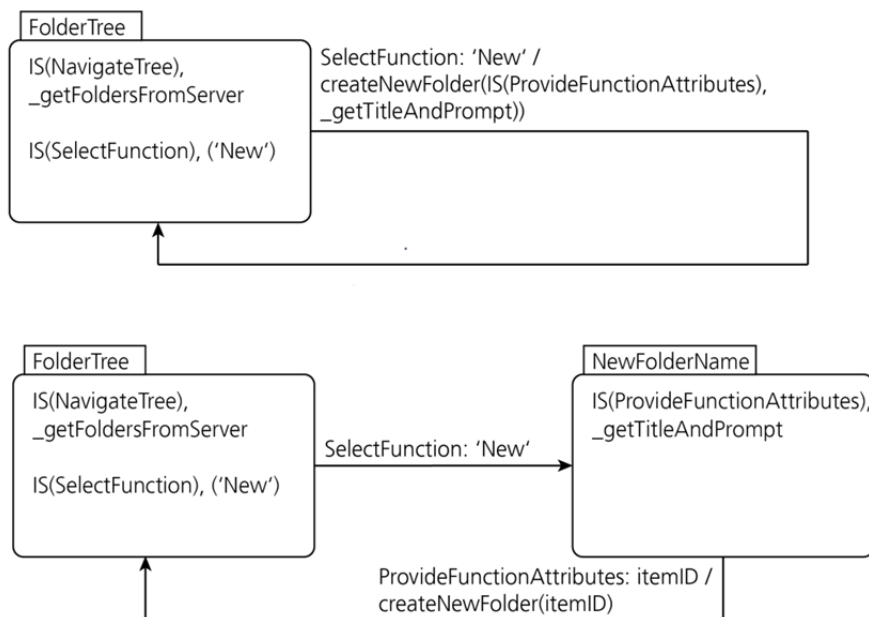


Abbildung 24 Transition mit einer IS-Erweiterung (ISE) in der verkürzten AAIM-Schreibweise (oben) und in der UML-konformen Schreibweise (unten)

Diese Form der Spezifikation von Interaktionssituationen an Transitionen dient der Übersichtlichkeit und besseren Handhabung des AAIM. Da sich Zustandsautomaten zu jedem Zeitpunkt genau in einem Zustand befinden müssen und Transitionen ohne zeitliche Verzögerung erfolgen müssen (vgl. Jeckle et al., 2004), wird diese Modellierungsform des AAIM lediglich als vereinfachende, alternative Notation eingeführt. Abbildung 24 illustriert ein Beispiel für eine Transition mit eigener ISE, einmal in der verkürzten AAIM-Schreibweise (oben) und einmal in der UML 2 – konformen Notation (unten).

8.6 Interaktionssituationen und ihre Syntax

Zur Definition der Ausdrucksmöglichkeiten von Interaktionssituationen wird die von der Object Management Group (OMG, 2011) verwendete Variante der Backus-Naur-Form (BNF) benutzt.

Die Syntax einer IS umfasst zwei Bestandteile: (1) Der „Eingangsterm“ der IS wird innerhalb eines Zustands dargestellt. Er bringt zum Ausdruck, dass eine bestimmte Option oder eine Menge von Optionen zu einem bestimmten Zeitpunkt zur Verfügung steht. (2) Der „Ausgangsterm“ der IS

dient als Beschriftung eines Zustandsübergangs, um zu definieren, welches Systemverhalten als Reaktion auf eine Benutzerinteraktion mit der IS folgt. Die einzige Ausnahme stellt hier die ISE dar, die in der vereinfachten AAIM-Form lediglich an einer Transition dargestellt wird und nur einen Ausgangsterm besitzt.

8.6.1 Eingangsterm der IS

Der Eingangsterm umfasst den Namen der IS und deren Eingangsparameter:

```

<IS-Eingangsterm> ::= 'IS(' <IS-Name> '), ' <Eingangsparameter>
<Eingangsparameter> ::= <direkt eingegebener Inhalt> | <Datenakquisitionsfunktionen>
<direkt eingegebener Inhalt> ::= [ (<Identifizier>',' [ <ID-Parameter>',' ])* ] <Identifizier> [ ',' <ID-Parameter> ]
<ID-Parameter> ::= [ (<Parameter> ',' )* ] <Parameter>
<Datenakquisitionsfunktionen> ::= [ ( '_'<Funktion> ',' )* ] ( '_'<Funktion> )
<Funktion> ::= <Funktionsname> '(' [ <Funktionsparameter> ] ')'
<Funktionsparameter> ::= [ (<Parameter> ',' )* ] <Parameter>

```

Die Eingangsparameter repräsentieren den Inhalt bzw. die Daten, die dem Nutzer über die Bedien- und Anzeigekomponente des aktivierten Interaktionsmusters dargestellt werden. Bei der Erstellung des AAIM bestehen zwei Möglichkeiten zur Spezifikation der Eingangsparameter: Entweder der Entwickler gibt die Parameter zu jeder IS direkt manuell an. Oder er referenziert eine Datenakquisitionsfunktion, die die Parameter in einem vordefinierten Format aus dem Datenbestand der Applikation besorgt.

Unabhängig davon, ob sie manuell oder über eine Datenakquisitionsfunktion referenziert werden, hängt das Format der Eingangsparameter von der jeweiligen IS und dem damit verbundenen Interaktionsmusterbündel ab. Während bei der IS EditForm zu jedem Formularfeld-Identifizier noch ID-Parameter zur Typisierung angegeben werden, werden sich die Eingangsparameter der IS SelectFunction auf die Identifizier der zur Auswahl gestellten Funktionen beschränken. Ähnlich verhält es sich mit den Interaktionssituationen SelectGoTo und MetaGoTo, bei denen die Eingangsparameter aus den Identifiern der anzusteuern Zustände bestehen. Darüber hinaus können optionale ID-Parameter angegeben werden, beispielsweise um die einzelnen Optionen zu priorisieren und darüber zu definieren, welche Inhalte bevorzugt dargestellt werden sollen und welche ausgeblendet werden können, falls die Anpassung an individuelle Nutzeranforderungen eine Reduktion der dargestellten Datenmenge erfordert.

Die Namen von Datenakquisitionsfunktionen werden in Kamelschreibweise mit einem kleinen Anfangsbuchstaben geschrieben. Sie beginnen mit einem Verb (in der Regel „get“) und einem dem Namen vorangestellten Unterstrich „_“, um sie von anderen Applikationsfunktionen zu unterscheiden (z.B. _getEmailsFromServer).

8.6.2 Ausgangsterm der IS

Der Ausgangsterm der IS beinhaltet den Namen der IS, das Interaktionsergebnis, und optional eine Referenz auf eine Applikationsfunktion, die zusammen mit der Transition als zusätzliche Aktivität ausgelöst wird:

```
<IS-Ausgangsterm> ::= <IS-Name> ':' <Interaktionsergebnis> [ '/' <Applikationsfunktion> ]  
<Interaktionsergebnis> ::= <Identifizier> [ ( ';' <ID-Parameter> )* ]  
<Applikationsfunktion> ::= <Funktionsname> '(' [ <Funktionsparameter> ] ')'  
<Funktionsparameter> ::= [ ( <Parameter> ';' )* ] <Parameter>
```

Das Interaktionsergebnis einer IS repräsentiert den Teil des Nutzerverhaltens während der IS, der bestimmt, welcher Zustand als nächstes angesteuert wird, d.h. welche Transition genommen wird. Damit ist das Interaktionsergebnis eine Teilmenge des Ausgangsparametervektors. Der Identifizier-Term des Interaktionsergebnisses repräsentiert die für den Zustandswechsel entscheidende Nutzeraktion, z.B. die ausgewählte Option der IS SelectService oder das ausgewählte Element der IS SelectItem. Die optionalen ID-Parameter werden für die Angabe weiterer Nutzereingaben verwendet, die einen Einfluss auf die Auswahl des nächsten Zustands besitzen. Bei einer IS EditForm zur Angabe von Kundendaten in einem Online-Bestellprozess beispielsweise stünde der Identifizier für die Nutzeraktion zur Bestätigung der eingegebenen Daten (z.B. über einen „Weiter“-Button), während die ID-Parameterfelder genutzt würden, um Bedingungen für die Auswahl des nächsten Zustands anzugeben. Zum Beispiel könnte ein ID-Parameter die gewünschte Zahlungsart repräsentieren und bei Auswahl der Option „Kreditkarte“ in diesem Formularfeld zu dem Zustand verweisen, in dem Kreditkartinformationen eingegeben werden können.

Die Namen von Applikationsfunktionen werden in Kamelschreibweise mit einem kleinen Anfangsbuchstaben geschrieben. Sie beginnen mit einem Verb, z.B. createNewEmail().

8.7 AAIM im Entwicklungsprozess einer adaptiven Benutzungsschnittstelle

In Bezug auf das AAIM beschränkt sich diese Arbeit auf die konzeptionellen Eigenschaften, die das AAIM zu einer geeigneten Grundlage für die Generierung einer ABS machen. Für die praktische Anwendung ergeben sich weitere Voraussetzungen, die nicht im Fokus dieser Arbeit stehen und in erster Linie durch ein geeignetes Entwicklungswerkzeug adressiert werden können. In einem Überblicksartikel zum MyUI-System (Peissner et al., 2012a) findet sich eine zusammenfassende Übersicht der MyUI-Entwicklungsumgebung:

- Die Entwicklungsumgebung unterstützt die einfache Erstellung eines AAIM durch einen grafischen Modelleditor. In diesem grafischen Editor kann der Entwickler neue Zustände erzeugen und den Zuständen per Drag-and-Drop Interaktionssituationen zuweisen. Dabei können die Eingangs- und Ausgangsparameter der Interaktionssituationen (bzw. der entsprechenden Interaktionsmusterbündel) direkt definiert werden. Je nach Definition und Parametrierung der Interaktionssituationen werden entsprechende Transitionen automatisch erzeugt. Sie können mit anderen Zuständen verbunden werden. So kann die gesamte Applikationsstruktur nach und nach modelliert werden.

- Neben der grafischen Repräsentation des AAIM bietet das Entwicklungswerkzeug eine Ansicht des Quelltexts der Dialogsteuerung der ABS, der aus dem AAIM erzeugt wird. Während das AAIM technologieunabhängig ist, ist auf Ebene des vom Entwicklungswerkzeug generierten Quelltexts der Dialogsteuerung bereits eine Technologieentscheidung getroffen²⁷. Der Entwickler kann jederzeit sowohl zur Ansicht als auch zur Bearbeitung des Modells zwischen beiden Repräsentationen wechseln, die stets zueinander konsistent gehalten werden.
- Eine weitere Komponente der Entwicklungsumgebung ist der MyUI-Entwurfsmuster-Browser. Er bietet einen direkten Zugang zum Entwurfsmusterrepositorium mit detaillierten Informationen zu den Interaktionssituationen und Entwurfsmustern und ihrer Rolle im Prozess der ABS-Generierung und Adaptation. Der Musterbrowser soll den Entwickler unterstützen, den MyUI-Ansatz zu verstehen und die richtigen Interaktionssituationen für das AAIM aufzufinden.
- Die Entwicklungsumgebung bietet eine Vorschau auf die verschiedenen möglichen Benutzungsschnittstellen, die auf der Basis des AAIM generiert werden können. Der Entwickler kann ein UP laden und verändern, um zu sehen, wie sich die generierte ABS für verschiedene Nutzer und Nutzungsbedingungen darstellen würde.
- Schließlich ermöglicht die Entwicklungsumgebung ein weitreichendes Customizing der generierten ABS über eine Veränderung des Customization-Profiles (vgl. Abschnitt 5.7). Um zu überprüfen, ob trotz Veränderungen der Schriftarten, Farben, Icons, etc. dennoch eine hohe Zugänglichkeit gewährleistet werden kann, steht eine Simulationsfunktion zur Verfügung, die durch die Anwendung verschiedener optischer Filter die ABS so darstellt, wie sie von Nutzern mit unterschiedlichen visuellen Einschränkungen wie z.B. grünem Star oder Farbfeldsichtigkeit wahrgenommen werden würde (vgl. Takacs et al., 2011).

²⁷ In der ersten Implementierung der MyUI-Entwicklungsumgebung wird hierfür CakePHP verwendet (vgl. Gacimartin et al., 2011).

9 Evaluierung im Anwendungsfeld barrierefreier Applikationen für interaktives Fernsehen

Diese Arbeit beschreibt einen generischen Ansatz, der die Realisierung von ABS zur Überwindung von Nutzungsbarrieren in verschiedenen Anwendungsbereichen ermöglicht. Die konkrete Ausführung des Ansatzes und dessen Validierung erfolgt am Beispiel von Informations- und Kommunikationsapplikationen für *interaktives Fernsehen (iTV)*.

Die *Relevanz* dieses Anwendungsfeldes ergibt sich unter anderem durch die große Bedeutung, die iTV in den letzten Jahren insbesondere für die Bereiche „Smart Living“ und „Ambient Assisted Living“ als zentrale Anzeige- und Bedienplattform erhalten hat. Einerseits ermöglicht es eine komfortable Bediensituation, die eher mit Freizeit als mit Computerarbeit assoziiert wird. Andererseits verspricht man sich eine deutlich weitere Verbreitung moderner Informations- und Kommunikationsanwendungen, wenn sie über eine Plattform angeboten werden, mit der auch weniger technikaffine Zielgruppen gut vertraut sind (vgl. BITKOM, 2013).

Die implementierten Anpassungsmechanismen zielen dabei auf die Überwindung von Nutzungsbarrieren, die typischerweise im Zusammenhang mit dem *Älterwerden* auftreten. Damit umfasst die Gruppe der adressierten Endnutzer alle Personen mit einer oder mehreren der folgenden Einschränkungen (klassifiziert nach WHO ICF²⁸):

Perzeptuelle Einschränkungen:

- Einschränkung des Sehvermögens (b210), insbesondere der Sehschärfe (b2100)
- Einschränkung des Hörvermögens (b230) und des Tastsinns (b265)

Motorische Einschränkungen:

- Einschränkungen der Muskelkraft (b730) und des Muskeltonus (b735)
- Einschränkungen der Kontrolle von Willkürbewegungen (b760) und unwillkürlichen Bewegungen (z.B. Tremor) (b765)
- Einschränkungen von Stimm- und Sprechfunktionen, insbesondere Artikulationsfunktionen (b320)

Kognitive Einschränkungen:

- Einschränkungen der Aufmerksamkeit (b140) und des Gedächtnisses (b144)
- Einschränkungen der mentalen Funktionen zur Erkennung und Interpretation sensorischer Reize (b156)
- Einschränkungen des Denkens (b160)
- Einschränkungen der kognitiv-sprachlichen Funktionen (b167)

²⁸ <http://www.who.int/classifications/icf/en/>

Mit der Gruppe der Älteren wird eine *hoch relevante Zielgruppe* adressiert, die in der Zukunft weiter an Umfang und gesellschaftlicher Bedeutung gewinnen wird. Im Zuge des demografischen Wandels wird gerade eine stärkere Einbindung der älteren Generation in Gesellschaft und Erwerbstätigkeit diskutiert. Nationale und internationale Forschungsprogramme²⁹ reflektieren die außerordentliche Bedeutung einer altersgerechten Technikgestaltung für die übergeordneten Ziele einer längeren aktiven Teilhabe am gesellschaftlichen und wirtschaftlichen Geschehen - aber auch einer hochwertigen und bezahlbaren Betreuung und Pflege im Alter.

Die gewählte Kombination aus Anwendungsfeld und Zielgruppe ist nicht nur hoch relevant, sondern auch besonders gut geeignet, um die potenziellen Vorteile von ABS nachzuvollziehen (vgl. Darstellung in der Einleitung (Kapitel 1) nach Kühme & Schneider-Hufschmidt, 1993):

- Die Zielgruppe der Älteren ist durch eine extreme Heterogenität bezüglich der perzeptuellen, kognitiven und motorischen Fähigkeiten und Einschränkungen gekennzeichnet. Daraus ergeben sich vielfältige individuelle Anforderungen an die Benutzungsschnittstellen, die sich kaum durch *ein* einheitliches Design für alle abdecken lassen.
- Gerade bei älteren Nutzern unterliegen die individuellen Anforderungen an Benutzungsschnittstellen dynamischen Veränderungen. Kurzfristige Veränderungen von Fähigkeiten und Einschränkungen ergeben sich häufig durch plötzliche gesundheitliche Einbrüche wie z.B. bei einem Schlaganfall oder durch Schwankungen der „Tagesform“, die insbesondere bei schlechterer Gesamtkonstitution stärker zur Wirkung kommen.
- Die Kontextbedingungen der Nutzung von iTV-Anwendungen sind relativ stabil. Tageszeitbedingte Veränderungen der Lichtverhältnisse und der Umgebungsgeräusche können dennoch Anpassungen der Benutzungsschnittstelle erforderlich machen. Darüber hinaus sind etliche iTV-Applikationen denkbar, die nicht ausschließlich über das Fernsehgerät bedient werden können, sondern zum Beispiel auch über ein Mobiltelefon.

Des Weiteren ist das gewählte Anwendungsfeld *repräsentativ* im Sinn der in Kapitel 2 herausgearbeiteten Anforderungen an den Lösungsansatz. Das heißt, die allgemein formulierten Anforderungen gelten auch für eine Lösung im konkreten Anwendungsfeld. Damit ist eine wesentliche Grundvoraussetzung für die Übertragbarkeit der spezifischen Validierungsergebnisse auf andere Anwendungsfelder erfüllt.

Zusätzlich ergibt sich aufgrund der hohen Heterogenität der älteren Nutzer eine besondere Komplexität der notwendigen Adaptationen. Wenn sich der entwickelte Ansatz für die Zielgruppe der Älteren erfolgreich anwenden lässt, sollte dies auch für weniger komplexe Anwendungsszenarien gelten.

²⁹ Z.B. der BMBF-Förderschwerpunkt "Mensch-Technik-Interaktion (MTI) für den demografischen Wandel" (<http://www.bmbf.de/foerderungen/21450.php>; abgerufen am 01. Juli 2013) und das Themenfeld „ICT for Health, Ageing Well, Inclusion and Governance“ des Arbeitsprogramms 2013 des 7. Rahmenprogramms der EU (http://ec.europa.eu/research/participants/portalplus/static/docs/calls/fp7/common/32767-annex_6_to_the_decision_ict_for_cap_en.pdf; abgerufen am 01. Juli 2013)

9.1 Evaluierungsplan

Die Evaluierung des Ansatzes dieser Arbeit erfolgt anhand der in Kapitel 2 hergeleiteten Anforderungen. Auf dieser Grundlage ergeben sich die in Tabelle 15 aufgeführten konkreten Fragestellungen für die Evaluierung.

| Anforderungen an Lösungsansatz | Fragestellungen für die Evaluierung |
|--|--|
| Adaptivität: automatische systeminitiierte Anpassungen (2.1) | (1) Erfolgt bei signifikanten Änderungen des Kontextwissens eine entsprechende automatische Anpassung der ABS? |
| Gemischte Initiative: Adaptivität und Adaptierbarkeit in einem System (2.1) | (2) Bietet das System neben automatischen Anpassungen auch die Möglichkeit für weitreichende nutzerinitiierte Anpassungen? (3) Bietet das System Mechanismen, die eine Kooperation zwischen Nutzer und intelligenten Automatismen des Systems ermöglichen, um das Anpassungsverhalten der ABS zu bestimmen? |
| Kontinuierliche Anpassung an ein selbstlernendes Profil während der Nutzung (2.2) | (4) Wird die ABS unmittelbar und ohne wesentliche Verzögerungen angepasst, wenn sich während der Nutzung das Kontextwissen signifikant ändert? |
| Transparenz: erkennbare, verstehbare und nachvollziehbare Adaptationen (2.3.1) | (5) Kann das System sicherstellen, dass die Benutzer die automatischen Anpassungen erkennen und verstehen und dass Verwirrung und Desorientierung vermieden werden? |
| Kontrollierbarkeit automatischer Adaptationen (2.3.2) | (6) Bietet das System Mechanismen, die den Benutzern eine bedarfsgerechte Kontrolle des Adaptationsverhaltens ermöglichen? |
| Anpassungen an Nutzer, Umgebung und technische Ausstattung (0) | (7) Ermöglicht die ABS-Infrastruktur eine Adaptivität an Veränderungen im Wissen über Nutzer, Umgebung und technische Ausstattung? |
| Anpassungen der Informationsdarstellung, Interaktion und Navigation (0) | (8) Ermöglicht die ABS-Infrastruktur eine Adaptivität der Informationsdarstellung, der Interaktion und der Navigation? |
| Modulare und erweiterbare Adaptationsmechanismen (2.5) | (9) Unterstützt die ABS-Infrastruktur eine Implementierung von ABS in Modulen, die eigenständig und in sich abgeschlossen sind? (10) Erlaubt die ABS-Infrastruktur Erweiterungen um zusätzliche Kontextbedingungen und Interaktionslösungen? |
| Adaptationsmechanismen öffentlich zugänglich und offen für externe Beiträge (2.5) | (11) Bietet die ABS-Infrastruktur Möglichkeiten, um die Adaptationsmechanismen für die Öffentlichkeit zugänglich zu machen und für externe Beiträge zu öffnen? |
| Effiziente Entwicklung: wiederverwendbare Komponenten und geringer Spezifikationsaufwand bei ausreichender Mächtigkeit (2.6.1) | (12) Ermöglicht die ABS-Infrastruktur, dass ABS bereits mit geringem Spezifikationsaufwand erstellt werden können? (13) Kann durch wiederverwendbare Komponenten der Aufwand zur Entwicklung neuer Applikationen bzw. neuer ABS sukzessive verringert werden? |
| Geringe Einstiegshürde und Kompetenzanforderungen auf Seiten der Entwickler (2.6.2) | (14) Unterstützt die ABS-Infrastruktur auch für weniger gut ausgebildete Entwickler einen leichten Einstieg in die Entwicklung von ABS? |

| | |
|---|--|
| Generierung der ABS für Entwickler nachvollziehbar und beherrschbar (2.6.3) | (15) Unterstützt die ABS-Infrastruktur die Entwickler dabei, den Zusammenhang zwischen einem AAIM und den generierten ABS zu verstehen und das Ergebnis der ABS-Generierung in ausreichendem Maße zu beeinflussen? |
|---|--|

Tabelle 15 Fragestellungen für die Evaluierung des Ansatzes auf Grundlage der hergeleiteten Anforderungen

Für die Erprobung und Evaluierung wird eine dreigliedrige Vorgehensweise gewählt:

- *Technische Evaluierung anhand einer Referenzimplementierung*
Anhand von ABS für iTV-Applikationen wird der Ansatz implementiert. Neben den Basisfunktionen des adaptiven iTV, die ein Hauptmenü zur Serviceauswahl und ein Konfigurationsmenü zu nutzerinitiierten Anpassung umfassen, dienen eine E-Mail-Applikation und eine Wettervorhersage als Beispielimplementierungen. Diese Form der Evaluierung zielt auf eine Validierung der generellen Funktionstüchtigkeit des Ansatzes und der Erprobung der objektiv feststellbaren technischen Eigenschaften der ABS-Infrastruktur. Es werden die Evaluierungsfragen 1, 2, 3, 4, 7, 8, 9, 10, 11 und 13 der Tabelle 15 adressiert.
- *Empirische Evaluierung mit potenziellen Endnutzern*
Zur gezielten Untersuchung der Eigenschaften des Rahmenmodells, die nur aus der Nutzerperspektive feststellbar sind, werden empirische Studien mit potenziellen Nutzern durchgeführt. Dabei geht es um die Fragestellungen der Transparenz und Kontrollierbarkeit der automatischen Adaptationen (vgl. Tabelle 15, Fragestellungen 5 und 6).
- *Empirische Evaluierung mit potenziellen ABS-Entwicklern*
Die praktische Eignung des Ansatzes für die kommerzielle Software-Entwicklung wird hauptsächlich in empirischen Studien mit Software-Entwicklern und Entscheidungsträgern aus Software-Unternehmen untersucht. Dabei stehen deren subjektive Bewertung und Akzeptanz der technischen Funktionen der ABS-Infrastruktur im Mittelpunkt des Interesses. Die Evaluierungsfragen 12, 14 und 15 der Tabelle 15 werden näher untersucht.

9.2 Technische Evaluierung anhand einer Referenzimplementierung

Für die Evaluierung des Ansatzes wird eine Beispielimplementierung benötigt, die sowohl eine programmiertechnische Umsetzung des ABS-Rahmenmodells als auch eine Entwicklung konkreter ABS für Beispiellapplikationen umfasst. Die Implementierung wurde im Rahmen des MyUI-Projekts³⁰ in Zusammenarbeit mehrerer Projektpartner durchgeführt. Die Darstellung der Implementierung beschränkt sich auf die Aspekte, die für die Evaluierung des Ansatzes relevant sind. Für eine ausführlichere Beschreibung der technischen Details wird auf Peissner et al., 2013 verwiesen.

Der in dieser Arbeit vorgestellte Ansatz für barrierefreie ABS ist generisch und nicht auf bestimmte Technologien der Benutzungsschnittstellen oder deren Generierung festgelegt. Für die Erprobung des Ansatzes wird eine Implementierung mit modernen Webtechnologien gewählt. Browserbasierte Implementierungen bieten eine hohe Plattformunabhängigkeit und laufen auch auf aktuellen iTV-Geräten. Die generierten ABS beruhen vorwiegend auf HTML mit CSS und AJAX/JavaScript für die dynamischen Elemente und Mechanismen. Die Module zur Generierung der ABS sind in CakePHP und JavaScript realisiert. Dabei bezieht sich CakePHP auf alle serverseitigen Module. Diese beinhalten die Mechanismen und Komponenten der ABS-Parametrierung sowie die Interaktionsmuster und Bedien- und Anzeigeelemente. Clientseitig werden alle dynamischen Prozesse der Adaptation gesteuert. Die wichtigsten hierbei beteiligten Komponenten umfassen die Adaptationsengine, die Adaptation-Handler der einzelnen Musterbündel und die Adaptationsmuster. Alle clientseitigen Mechanismen und Komponenten sind in JavaScript umgesetzt (vgl. Gacimartin et al., 2011 und Peissner et al., 2013). So können Laufzeitanpassungen innerhalb eines Zustands der ABS ausgeführt werden, ohne die komplette Seite neu zu laden.

Die Konkretisierung des Adaptationsprozesses beruht auf einer Aufteilung der Zuständigkeiten zwischen drei wesentlichen Komponenten:

- Jedes Bündel von Interaktionsmustern besitzt einen *Adaptation-Handler*, der die Adaptationslogik des jeweiligen Bündels beinhaltet und für die Auswahl des am besten geeigneten Musters zuständig ist. Alle in der aktuellen ABS verwendeten Interaktionsmuster und Bedien- und Anzeigeelemente werden durch Adaptation-Handler bei der Adaptationsengine registriert. So ist die Adaptationsengine stets informiert, welche Bedien- und Anzeigekomponenten aktuell verwendet werden. Darüber hinaus unterstützen die Adaptation-Handler die Auswahl des am besten geeigneten Adaptationsdialogmusters im Fall einer Laufzeitanpassung. Auf der Grundlage ihres „Wissens“, welches Interaktionsmuster ihres Bündels aktuell dargestellt wird und welches Muster nach der geplanten Adaptation dargestellt werden soll, geben sie eine lokale Schweregradabschätzung der Veränderungen innerhalb ihres Musterbündels ab (vgl. Abschnitt 7.5). Schließlich steuern die Adaptation-Handler die bündelspezifischen Anpassungsprozesse – insbesondere Veränderungen der Bedien- und Anzeigeelemente und Übergänge von einem zum nächsten Interaktionsmuster eines Bündels mitsamt den beteiligten Adaptationsausführungsmustern.

³⁰ <http://www.myui.eu/>

- Die *Adaptationsengine* steuert den gesamten Adaptationsprozess. Sie dient als Kommunikationsschnittstelle zur Kontextmanagementinfrastruktur und ruft in regelmäßigen Abständen von dort das aktuelle UP ab. Zugunsten einer einfachen Umsetzung nutzt die erste Implementierung einen Pull-Mechanismus zur Kommunikation des UP. Um längere Latenzen zwischen kritischen Interaktionsereignissen und entsprechenden Laufzeitanpassungen der ABS zu vermeiden, wird hierfür ein Pull-Intervall von einer Sekunde verwendet. Außerdem wendet die Adaptationsengine die Individualisierungsmuster an, um auf der Grundlage von UP und DP das UIP zu erstellen und zu aktualisieren (ABS-Parametrierung). Schließlich koordiniert sie die Adaptation-Handler aller aktuell aufgerufenen Interaktionsmusterbündel.
- Die *Adaptationssteuerung* sorgt dafür, dass die Laufzeitanpassungen für den Benutzer transparent und kontrollierbar sind. Sie führt die Adaptationsdialogmuster aus und holt entsprechende Nutzerreaktionen wie z.B. Zustimmung oder Ablehnung ein.

Abbildung 25 stellt das Konzept zur Implementierung von Laufzeitanpassungen dar. Wenn die Funktion `checkUIProfile` der Adaptationsengine Veränderungen im UIP feststellt, so werden die `prepareAdaptation`-Methoden der einzelnen Interaktionsmusterbündel (bzw. deren Adaptation-Handler) ausgeführt. Diese Methoden wählen das für das neue UIP am besten geeignete Interaktionsmuster aus und schlagen aufgrund des Vergleichs des neuen und des bisherigen Interaktionsmusters das aus ihrer Sicht am besten geeignete Adaptationsdialogmuster vor. Die Adaptationsengine sammelt die Ergebnisse dieses Prozesses aller aktiven Adaptation-Handler ein und übergibt sie ihrer Methode `beforeAdaptation`. Diese wertet die Ergebnisse aus und entscheidet, welches Adaptationsdialogmuster ausgeführt werden soll. Damit ruft sie die `beforeAdaptation`-Methode der Adaptationssteuerung auf, die den Teil des ausgewählten Adaptationsdialogmusters ausführt, der vor der Ausführung der Anpassung erfolgen soll. Für das Muster „Explizite Nutzerbestätigung vor der Adaptation“ wäre das die Abfrage einer expliziten Nutzerzustimmung. Bei den anderen beiden Mustern in Abschnitt 7.5 passiert nichts vor der Ausführung der Adaptation. Das Ergebnis der `beforeAdaptation` wird an die Adaptationsengine zurückgemeldet. Liegt keine explizite Ablehnung des Nutzers vor, so werden die `performAdaptation`-Methoden aller aktuell registrierten Adaptation-Handler gestartet. Diese sorgen für die Durchführung aller Anpassungen an der Benutzungsschnittstelle und setzen dabei die Vorgaben der Adaptationsausführungsmuster bezüglich der animierten Übergänge um. Sobald alle Anpassungen abgeschlossen sind, wird die `afterAdaptation`-Methode der Adaptationsengine aufgerufen, die – analog zur `beforeAdaptation` – an die Adaptationssteuerung übergibt, um den Teil des ausgewählten Adaptationsdialogmusters durchzuführen, der nach der Adaptation vorgesehen ist. Mögliche Ergebnisse dieser Methode umfassen eine implizite Zustimmung des Nutzers (keine Aktion), das Rückgängigmachen der Anpassung, sowie eine explizite Zustimmung oder Ablehnung der ausgeführten Anpassung im Dialogfenster des Musters „Explizite Nutzerbestätigung nach der Adaptation“ (vgl. Abschnitt 7.5.3). Das Ergebnis wird an die Adaptationsengine zurückgemeldet. Für die Fälle der Ablehnung und des Rückgängigmachens der Anpassung gibt die Adaptationsengine der Kontextmanagementinfrastruktur das Signal, die letzten UP-Änderungen rückgängig zu machen. Dies führt schließlich dazu, dass über den Prozess der ABS-Parametrierung auch die der Laufzeitanpassung zugrundeliegenden UIP-Änderungen rückgängig gemacht werden.

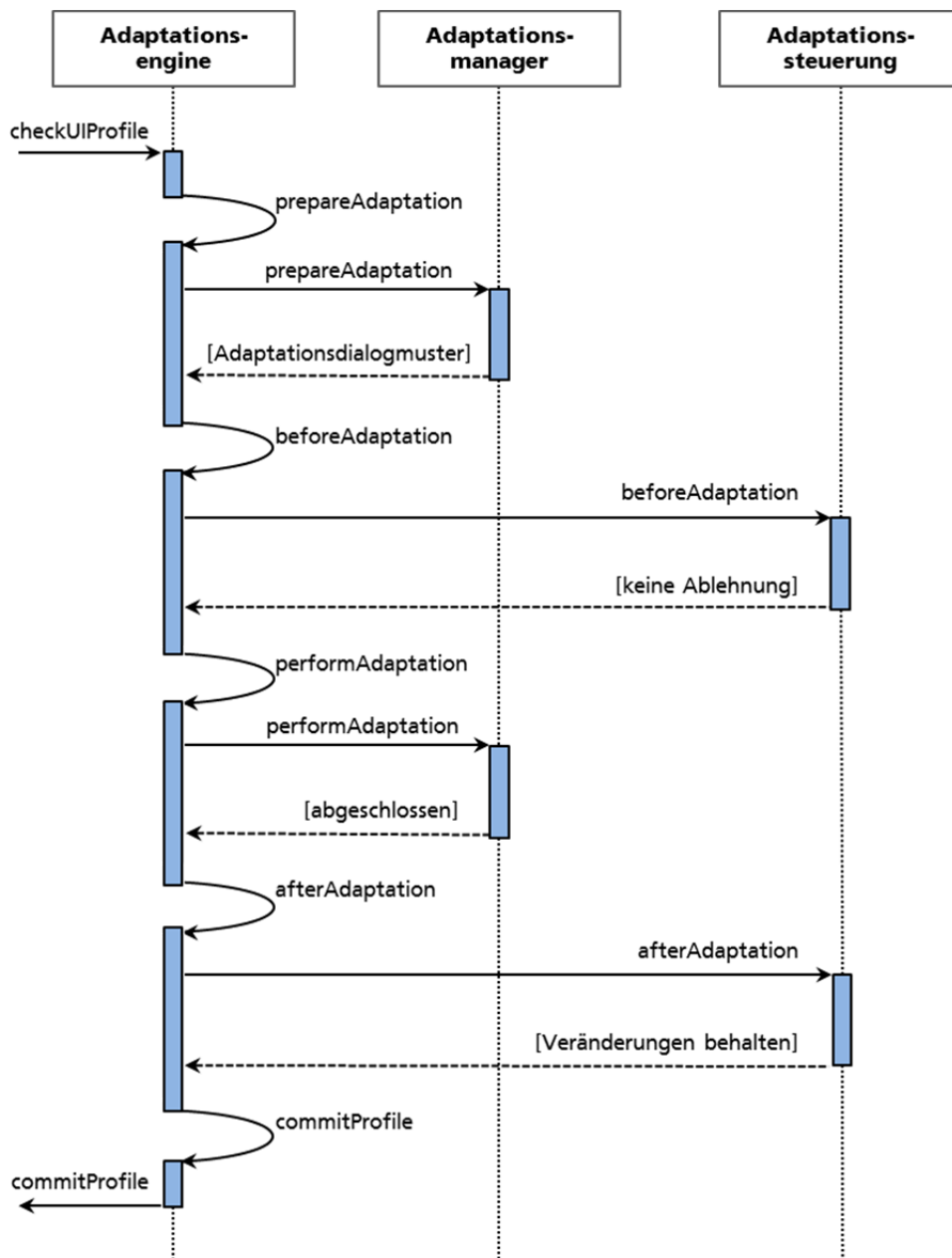


Abbildung 25 Implementierungskonzept für den Adaptationsprozess bei Laufzeitanpassungen (Sequenzdiagramm)

Die primäre Zielplattform der Referenzimplementierung ist ein browserbasiertes iTV-Gerät mit 32-Zoll-Bildschirmdiagonale des MyUI-Projektpartners Philips (Philips 32PFL7605H/12). Um die Anpassung an andere technische Geräte zu demonstrieren, wurde als weiteres Gerät ein Apple iPhone gewählt (s. Abbildung 26). Auf beiden Geräten laufen dieselben Applikationen ohne weitere Anpassungen des Quelltexts. Alle gerätespezifischen Anpassungen der ABS werden über unterschiedliche Ausprägungen des Geräteprofils (DP) gesteuert und über Veränderungen des UIP im Prozess der ABS-Parametrierung vermittelt (vgl. Bruikman et al., 2013).



Abbildung 26 MyUI-Referenzimplementierung auf iTV und iPhone

Die Beispielimplementierung umfasst drei adaptive Applikationen (Abbildung 32 im Anhang zeigt beispielhafte Bildschirmansichten):

- Die *Basisfunktionen des adaptiven iTV* umfassen ein Hauptmenü, das aus einer adaptiven Bildschirmansicht zur Auswahl einer Applikation besteht, und ein Konfigurationsmenü, in dem der Nutzer die Einstellungen des UIP und des UP verändern kann.
- Die *Wettervorhersage* umfasst zwei Bereiche: die Anzeige der Wetterprognose für die nächsten Tage und die Auswahl des gewünschten Orts.
- Die *E-Mail-Applikation* ist die komplexeste Beispielimplementierung. Sie bietet eine ABS für Googles Gmail. Sie nutzt die Gmail-Funktionalität und greift auf die Datenbasis eines Gmail-Accounts zurück. So kann der Nutzer über die generierte ABS die E-Mails seines Gmail-Accounts empfangen, lesen und bearbeiten sowie E-Mails von seinem Gmail-Account verschicken.

Wenn für eine bereits bestehende Applikation eine adaptive ABS mit der MyUI-Infrastruktur erstellt wird, sind im Wesentlichen zwei Schritte erforderlich:

- Erstellen eines Abstrakten Applikations-Interaktions-Modells (AAIM) (vgl. Kapitel 8).
- Sicherstellen, dass die im AAIM referenzierten Interaktionssituationen durch entsprechende Interaktionsmuster abgedeckt sind und gegebenenfalls zusätzlich benötigte Interaktionsmuster und Bedien- und Anzeigeelemente ergänzen.

Die Referenzimplementierungen dienen damit auch als Ausgangspunkt für die Erstellung zukünftiger ABS. Insbesondere gängige und generische Interaktionskomponenten, wie beispielsweise Elemente zur Auswahl, Texteingabe oder Funktionsanwahl bieten sich zur Wiederverwendung für eine Vielzahl von denkbaren Applikationen an. Je größer der Bestand bereits bestehender Interaktionsmuster ist, umso geringer wird der zu erwartende Aufwand für weitere Neuentwicklungen ausfallen.

Dennoch ist zu beachten, dass die im Rahmen der Referenzimplementierung erstellten Entwurfsmuster - mit Ausnahme der Adaptionsmuster - nicht Teil des in dieser Arbeit entwickelten MyUI-Rahmenmodells sind. Das Rahmenmodell bietet lediglich eine Infrastruktur und Mechanismen, auf deren Basis die konkreten Entwurfsmuster das in ihnen enthaltene Designwissen zur Gestaltung von barrierefreien ABS realisieren. Im Rahmen der Referenzimplementierung wurden die Entwurfsmuster nach bestem Wissen auf Grundlage des aktuellen Stands der Forschung und Technik erstellt. Die Frage, ob die damit generierten ABS tatsächlich in

der Lage sind, konkrete Nutzungsbarrieren zu überwinden, ist daher nicht Gegenstand dieser Arbeit, soweit sie von den konkreten Gestaltungslösungen abhängen.

Abbildung 33 im Anhang gibt einen Überblick der Entwurfsmuster, die nach Realisierung der drei Referenzapplikationen im MyUI-Entwurfsmusterrepositorium³¹ vorliegen. Insbesondere bei den Interaktionsmustern fällt auf, dass die Anzahl der Gestaltungsvarianten noch nicht einheitlich groß ist. Dadurch stehen für einige Elemente, wie beispielsweise die Titelleiste („Title Bar“) oder das Formularelement („Form“) neben den Standardelementen keine Varianten für eine Adaptation auf Interaktionsmusterebene zur Verfügung. Für die praktische Anwendung können derartige Lücken in der Implementierung eventuell bedeuten, dass auftretende Nutzungsbarrieren über Variationen in den anderen Mustertypen nicht ausreichend kompensiert werden können. Für die Evaluierung im Rahmen dieser Arbeit ergeben sich jedoch dadurch keine Einschränkungen, da sich die Evaluierung auf den Ansatz und nicht dessen vollständige Ausführung bezieht. Zur beispielhaften Illustration finden sich die Musterbeschreibungen von zwei Varianten des Interaktionsmusterbündels Main Menu im Anhang dieser Arbeit. Für alle weiteren Musterbeschreibungen gilt der Verweis auf das Musterrepositorium unter <http://myui-patterns.iao.fraunhofer.de>.

Als Beispiel für die Erstellung des AAIM dient die Darstellung eines AAIM-Ausschnitts der E-Mail-Applikation. Die E-Mail-Applikation ist in drei Hauptbereiche gegliedert: (1) In „Folders“ befinden sich der Posteingang und alle weiteren E-Mail-Ordner. Dort stehen dem Benutzer die wesentlichen Funktionen der Ansicht, Erstellung und Verwaltung von E-Mails zur Verfügung. (2) Der Bereich „Contacts“ dient der Verwaltung von E-Mail-Kontakten. (3) Unter „Settings“ können wesentliche Einstellungen angepasst werden. Jeder dieser Hauptbereiche wird im AAIM als zusammengesetzter Zustand modelliert. In Abbildung 27 ist der AAIM-Ausschnitt des „Folders“-Zustands dargestellt.

³¹ <http://myui-patterns.iao.fraunhofer.de>

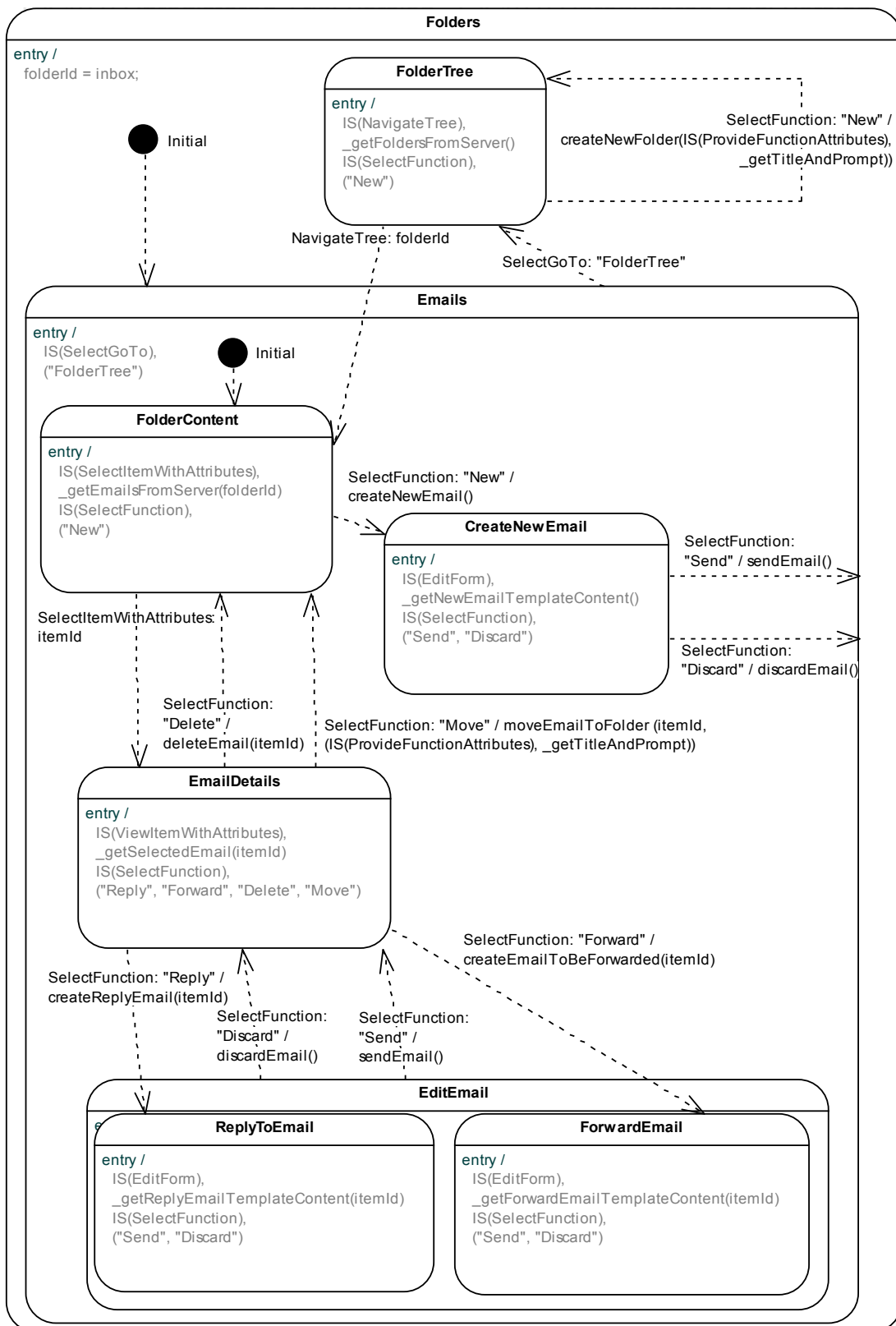


Abbildung 27 Ausschnitt des AAIM für die E-Mail-Applikation

9.2.1 Evaluierung der automatischen Adaptivität

9.2.1.1 Fragestellung

Erfolgt bei signifikanten Änderungen des Kontextwissens eine entsprechende automatische Anpassung der ABS? (Frage 1 der Tabelle 15; die untersuchte Anforderung ist in Abschnitt 2.1 beschrieben.)

9.2.1.2 Methode

Zur theoretischen Validierung werden gezielt die folgenden Eigenschaften der ABS-Infrastruktur untersucht:

- Werden Änderungen des Kontextwissens an die Mechanismen der ABS-Generierung und Adaptation kommuniziert?
- Bestehen Mechanismen, die bei bekannten Änderungen des Kontextwissens eine automatische Anpassung der ABS bewirken?

Zur empirischen Evaluierung dieser Fragestellung wurde eine Web-Applikation erstellt, mit der die Variablen des UP beliebig verändert werden können. So kann das Kontextwissen der Kontextmanagementinfrastruktur systematisch manipuliert werden, ohne auf die in dieser Arbeit nicht betrachteten Mechanismen der Kontexterfassung und Kontextinterpretation zurückgreifen zu müssen. Die Applikation ruft über das Internet das ausgewählte UP von der Kontextmanagementinfrastruktur ab und stellt die aktuellen UP-Werte dar. Über Schieberegler können alle Variablenwerte bearbeitet werden (s. Abbildung 28). Bestätigt der Nutzer die Veränderungen, wird das modifizierte UP an die Kontextmanagementinfrastruktur zurückgeschickt und ersetzt dort das bisherige Profil. So erhält die Adaptationsengine beim nächsten Abruf das neue UP und setzt den oben beschriebenen Adaptationsprozess in Gang.

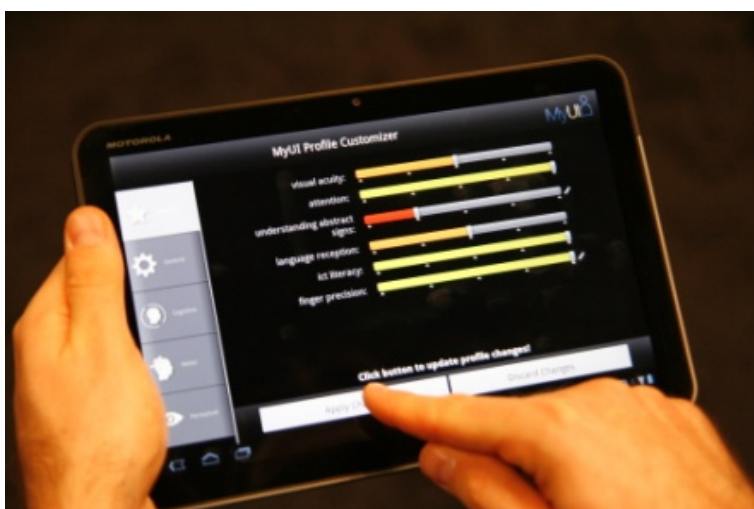


Abbildung 28 Web-Applikation zur gezielten Manipulation des Nutzer- und Umgebungsprofils (UP)

Die empirische Evaluierung erfolgt anhand eines Testszenarios, das verschiedene Veränderungen des UP in verschiedenen Situationen (Zuständen des AAIM) der Beispielimplementierung umfasst (s. Tabelle 16). Die Definition des Testszenarios orientiert sich am aktuellen Entwicklungsstand des Entwurfsmusterrepositoriums. Für jede getestete Situation und Anpassung stehen implementierte Entwurfsmuster bereit. Das Testszenario wird auf dem iTV-Gerät mit einer Fernbedienung durchlaufen. Es deckt drei verschiedene Prozesse zur Veränderung des UP ab:

- Ein Benutzer meldet sich an. Dadurch wird ein individuelles UP aktiviert.
- Afferenz und Inferenz der Kontextmanagementinfrastruktur erkennen und interpretieren Sensorereignisse.
- Über das UP bzw. UIP werden nutzerinitiierte Änderungen vorgenommen.

| # | ABS-Situation (Zustand) | UP | Kommentar/Interpretation |
|----|------------------------------------|---|--|
| 1 | Hauptmenü | Default-UP (keine Einschränkungen) | Das MyUI-Hauptmenü wird gestartet, kein Nutzer ist angemeldet. |
| 2 | | Erhöhung der UP-Variable <i>languageReception</i> von 0 auf 3 | Ein Nutzer wird angemeldet, dessen UP außer Probleme mit dem Sprachverstehen keine weiteren Einschränkungen reflektiert. |
| 3 | | Erhöhung der UP-Variable <i>fingerPrecision</i> von 0 auf 1 | Das UP verändert sich aufgrund eines Sensor-Events, z.B. erkennt die Kontextmanagementinfrastruktur ungewöhnliches Navigationsverhalten zwischen den Optionen des Hauptmenüs und schließt auf motorische Probleme mit der Fernbedienung. |
| 4 | E-Mail-Applikation: Posteingang | | Der Nutzer startet die E-Mail-Applikation. |
| 5 | | Erhöhung der UP-Variable <i>visualAcuity</i> von 0 auf 1 | Aufgrund eines Sensor-Events wird das UP verändert, z.B. erkennt die Kontextmanagementinfrastruktur, dass der Nutzer sich nach vorne beugt und schließt daraus, dass es sich um eine willentliche Geste zur Verringerung des Leseabstands handelt. |
| 6 | Hauptmenü | | Der Nutzer kehrt zum Hauptmenü zurück. |
| 7 | Wetter-Applikation: Wetterprognose | | Der Nutzer startet die Wetter-Applikation. |
| 8 | | Erhöhung der UP-Variable <i>visualAcuity</i> von 1 auf 2 | Erneut ergibt sich eine UP-Veränderung aufgrund eines Sensor-Events (wie oben). |
| 9 | Konfigurationsmenü | | Der Nutzer öffnet das Konfigurationsmenü und reduziert den UIP-Wert <i>screenComplexity</i> von „unconstrained“ auf „medium“, indem er die Option „fewer elements“ auswählt. |
| 10 | Wetter-Applikation: Wetterprognose | Erhöhung der UP-Variable <i>attention</i> von 0 auf 1 | Durch eine umgekehrte Anwendung der Individualisierungsmuster schlägt sich die UIP-Veränderung sofort im UP nieder (vgl. Abschnitt 7.6.2). Der Nutzer schließt das Konfigurationsmenü und kehrt zum letzten Applikationszustand zurück. |

Tabelle 16 Testszenario zur Evaluierung der automatischen Adaptivität

In den Schritten 2, 3, 5, 8 und 10 finden UP-Veränderungen statt. In diesen Schritten sollten aufgrund der in den relevanten Entwurfsmustern³² beschriebenen Adaptationsregeln automatische Anpassungen durchgeführt werden.

9.2.1.3 Ergebnisse

Die theoretische Evaluierung kommt zu dem Ergebnis, dass die wesentlichen Voraussetzungen für selbstständige Adaptationen vollständig erfüllt sind.

Die Kontextprofile UP und DP stellen die Eingangsgrößen für die Individualisierungsmuster dar (vgl. Abschnitt 5.6). Diese erzeugen und aktualisieren im Rahmen der ABS-Parametrierung das UIP (vgl. Kapitel 5), das als Grundlage für alle weiteren Schritte der ABS-Generierung und Adaptation dient (vgl. Kapitel 6 und 7). Damit können die Individualisierungsmuster als konzeptionelle Schnittstelle zur Übersetzung des Kontextwissens in Begriffe der ABS verstanden werden. Die technische Umsetzung der Profil-Kommunikation zwischen Kontextmanagement und ABS-Mechanismen wird in der technologieunabhängigen Beschreibung des Ansatzes (Kapitel 3 bis 8) nicht näher spezifiziert. Das in Abschnitt 9.2 beschriebene Implementierungskonzept für Laufzeitanpassungen bietet jedoch bereits eine Konkretisierung. Darüber hinaus ist der `profiles_controller` eine wesentliche Komponente des Controllers der CakePHP-Implementierung. Er ruft die Funktion `getUserProfileFromContextManager` der Adaptationsengine auf, die einen XML-RPC-Call³³ an die Kontextmanagementinfrastruktur sendet, um das aktuelle UP zu erhalten (vgl. Peissner et al., 2013).

Nach Aktualisierung des UIP auf Grundlage des UP und des DP folgen die Schritte der ABS-Vorbereitung und ABS-Generierung. Dabei werden zunächst die für die aktuell angenommenen Kontextbedingungen am besten geeigneten Bedien- und Anzeigekomponenten ausgewählt. Dann werden diese spezifisch ausgewählten Komponenten verwendet, um die ABS zu generieren und die notwendigen Adaptationen auszuführen. Die generischen Konzepte und Entwurfsmuster für diese beiden Schritte sind in den Kapiteln 6 und 7 detailliert beschrieben. In der Implementierung werden die dafür notwendigen Schritte insbesondere durch die Funktionen `checkUIProfile`, `prepareAdaptation` und `performAdaptation` der Adaptationsengine ausgeführt (s. Abschnitt 09.2).

Auch in der empirischen Evaluierung wird die automatische Adaptationsfähigkeit bestätigt. Alle erwarteten Anpassungen werden automatisch und in Übereinstimmung mit den Adaptationsregeln der Entwurfsmuster durchgeführt. Die Bildschirmansichten für jeden einzelnen Schritt des Szenarios finden sich im Anhang (Abschnitt 12.4). Einige der beschriebenen Adaptationen können auch anhand der MyUI-Demovideos unter <http://www.myui.eu/index.php?content=movies.html> nachvollzogen werden.

³² Individualisierungsmuster der Bündel *DISPLAY MODE*, *NAVIGATE AND SELECT RC*, *FONT SIZE* und *SCREEN COMPLEXITY*, Interaktionsmuster der Bündel *MAIN MENU*, *TITLE BAR*, *LIST WITH ATTRIBUTES*, *FUNCTIONS MENU*, *NAVIGATION MENU*, *FORM* und *SETTINGS MENU* und die dort aufgerufenen Bedien- und Anzeigeelemente (s. <http://myui-patterns.iao.fraunhofer.de>)

³³ XML-RPC steht für XML-„Remote Procedure Call“. XML wird zur Kodierung des Calls genutzt, HTTP als Kommunikationsmechanismus.

9.2.1.4 Interpretation und Diskussion

Zusammenfassend lässt sich feststellen, dass der beschriebene dreistufige Prozess der ABS-Generierung und Adaptation alle notwendigen Mechanismen zur Verfügung stellt, um automatische Anpassungen der ABS durchzuführen, wenn sich das Kontextwissen ändert. Das Konstrukt der verschiedenen Typen von Entwurfsmustern, die über eine Folge von Transformationen die Kontextprofile in eine konkrete ABS überführen, stellt sicher, dass mit den automatischen Anpassungen spezifisch auf das aktuelle Kontextwissen reagiert werden kann. In wie weit die resultierenden Anpassungen jedoch tatsächlich geeignet sind, um individuelle Nutzungsbarrieren zu überwinden, hängt in erster Linie von der inhaltlichen Qualität der Entwurfsmuster ab und liegt daher außerhalb des Fokus dieser Arbeit. Ebenso verhält es sich mit der Frage, ob überhaupt Entwurfsmuster für geeignete Anpassungen in jedem spezifischen Fall vorliegen. Hierin zeigt sich ein zentrales Problem des vorgestellten Ansatzes, das der generischen Natur des Rahmenmodells geschuldet ist. Der Ansatz liefert eine solide Infrastruktur für barrierefreie ABS. Die praktische Effektivität in der Umsetzung jedoch hängt schließlich maßgeblich von Umfang und Qualität des inhaltlichen Gestaltungswissens ab, das in den Entwurfsmustern repräsentiert ist.

9.2.2 Evaluierung der gemischten Initiative

9.2.2.1 Fragestellung

Die Evaluierung der gemischten Initiative erfolgt anhand von zwei Fragestellungen:

- Bietet das System neben automatischen Anpassungen auch die Möglichkeit für weitreichende nutzerinitiierte Anpassungen? (Frage 2 der Tabelle 15; die untersuchte Anforderung ist in Abschnitt 2.1 beschrieben.)
- Bietet das System Mechanismen, die eine Kooperation zwischen Nutzer und intelligenten Automatismen des Systems ermöglichen, um das Anpassungsverhalten der ABS zu bestimmen? (Frage 3 der Tabelle 15; die untersuchte Anforderung ist in Abschnitt 2.1 beschrieben.)

9.2.2.2 Methode

Wenn beide Fragestellungen positiv beantwortet werden können, gilt die Anforderung der gemischten Initiative als erfüllt. Unter Bezugnahme auf die Darstellung des Ansatzes (Kapitel 3 bis 8) wird aufgezeigt, wie und in welchem Umfang die Merkmale der gemischten Initiative umgesetzt sind.

9.2.2.3 Ergebnisse

Die Möglichkeiten, die der MyUI-Ansatz für eine nutzerinitiierte Anpassung bietet, sind in Abschnitt 7.6 detailliert beschrieben. Zu jedem Zeitpunkt der Interaktion mit einer MyUI-Applikation steht ein Konfigurationsmenü zur Verfügung. Über dieses Menü wird ein Zugang zum UP und zum UIP angeboten. Der Nutzer erhält eine Übersicht der Variablenwerte des UP bzw. UIP und kann diese selbst nach eigenen Wünschen verändern. Die nutzerinitiierten

Profiländerungen werden über die Mechanismen der ABS-Generierung und Adaptation sofort an der ABS wirksam. Abbildung 32 (3) im Anhang zeigt eine Bildschirmansicht des implementierten Konfigurationsmenüs. Über „My Settings“ gelangt der Nutzer in die Profilübersichten. Die drei Buttons rechts davon bieten einen Schnellzugriff auf wichtige UIP-Variablen (*fontSize*, *displayMode* und *screenComplexity*). Abbildung 32 (4) im Anhang stellt den Dialog zur Veränderung eines einzelnen Profilwerts dar (hier *visualAcuity* des UP).

Um darzustellen, welche Ausprägungen der gemischten Initiative im MyUI-System angelegt sind, wird in Tabelle 17 die Taxonomie von Dieterich et al. (1993) verwendet, in der zwischen Initiative, Vorschlag, Entscheidung und Ausführung der Anpassung unterschieden wird (vgl. Abschnitt 2.1).

| Kooperation zwischen Nutzer und System | | Beschreibung | | | | | | | | | | | | | | |
|---|--------|--------------|--------|------------|---|---|-----------|---|---|--------------|--|---|------------|---|--|--|
| <p>Nutzerkontrollierte Adaptivität</p> <table border="1"> <thead> <tr> <th></th> <th>System</th> <th>Nutzer</th> </tr> </thead> <tbody> <tr> <td>Initiative</td> <td>●</td> <td></td> </tr> <tr> <td>Vorschlag</td> <td>●</td> <td></td> </tr> <tr> <td>Entscheidung</td> <td></td> <td>●</td> </tr> <tr> <td>Ausführung</td> <td>●</td> <td></td> </tr> </tbody> </table> | | System | Nutzer | Initiative | ● | | Vorschlag | ● | | Entscheidung | | ● | Ausführung | ● | | <p>Im beschriebenen Ansatz stehen alle automatischen Adaptationen unter Kontrolle des Nutzers. Jede automatische Adaptation wird durch eines der in Abschnitt 7.5 beschriebenen Adaptationsdialogmuster begleitet. Jedes dieser Muster bietet dem Nutzer die Möglichkeit, automatische Anpassungen nach ihrer Durchführung wieder rückgängig zu machen. Damit trifft stets der Nutzer die letztendliche Entscheidung, ob die vom System vorgeschlagene und ausgeführte Adaptation bestehen bleiben soll oder nicht.</p> <p>Eine besonders starke Ausprägung der Nutzerkontrolle bieten die beiden Adaptationsdialogmuster mit expliziter Nutzerbestätigung vor bzw. nach der Adaptation (Abschnitte 7.5.2 und 7.5.3).</p> |
| | System | Nutzer | | | | | | | | | | | | | | |
| Initiative | ● | | | | | | | | | | | | | | | |
| Vorschlag | ● | | | | | | | | | | | | | | | |
| Entscheidung | | ● | | | | | | | | | | | | | | |
| Ausführung | ● | | | | | | | | | | | | | | | |
| <p>Systemunterstützte Adaptierbarkeit</p> <table border="1"> <thead> <tr> <th></th> <th>System</th> <th>Nutzer</th> </tr> </thead> <tbody> <tr> <td>Initiative</td> <td></td> <td>●</td> </tr> <tr> <td>Vorschlag</td> <td>●</td> <td></td> </tr> <tr> <td>Entscheidung</td> <td></td> <td>●</td> </tr> <tr> <td>Ausführung</td> <td>●</td> <td></td> </tr> </tbody> </table> | | System | Nutzer | Initiative | | ● | Vorschlag | ● | | Entscheidung | | ● | Ausführung | ● | | <p>Eine systemunterstützte Adaptierbarkeit liegt vor, wenn der Nutzer in einem Konfigurationsdialog Werte des UP verändert (vgl. Abschnitt 7.6.1). Die Initiative des Nutzers besteht darin, über die Veränderung des UP ein individuelles Anpassungsbedürfnis auszudrücken. Daraufhin generiert das System einen Vorschlag für die angepasste ABS und führt diesen unmittelbar aus. Über die Rückgängig-Funktion der Adaptationsdialogmuster bleibt die Entscheidung über die durchgeführte Adaptation beim Nutzer.</p> <p>Ein weiterer Fall dieses Modells liegt vor, wenn der Nutzer das UIP so verändert, dass das System nicht direkt ein dazu konsistentes UP ermitteln kann. Dann schlägt das System dem Nutzer in einem interaktiven Verhandlungsprozess weitere UIP-Änderungen vor, bis ein konsistentes UP existiert (vgl. Abschnitt 7.6.2).</p> |
| | System | Nutzer | | | | | | | | | | | | | | |
| Initiative | | ● | | | | | | | | | | | | | | |
| Vorschlag | ● | | | | | | | | | | | | | | | |
| Entscheidung | | ● | | | | | | | | | | | | | | |
| Ausführung | ● | | | | | | | | | | | | | | | |
| <p>Adaptierbarkeit</p> <table border="1"> <thead> <tr> <th></th> <th>System</th> <th>Nutzer</th> </tr> </thead> <tbody> <tr> <td>Initiative</td> <td></td> <td>●</td> </tr> <tr> <td>Vorschlag</td> <td></td> <td>●</td> </tr> <tr> <td>Entscheidung</td> <td></td> <td>●</td> </tr> <tr> <td>Ausführung</td> <td>●</td> <td></td> </tr> </tbody> </table> | | System | Nutzer | Initiative | | ● | Vorschlag | | ● | Entscheidung | | ● | Ausführung | ● | | <p>Über Änderungen der UIP-Werte kann der Nutzer direkten Einfluss auf die ABS ausüben. Er initiiert die Anpassung und gibt an, welcher Aspekt der ABS sich wie ändern soll. Lediglich die Ausführung der vom Nutzer spezifizierten ABS-Veränderung liegt dann beim System (vgl. Abschnitt 7.6.2). Solange die vom Nutzer vorgeschlagene UIP-Veränderung nicht zu einem Konflikt bei der Erstellung eines konsistenten UP führt, handelt es sich bei nutzerinitiierten UIP-Veränderungen immer um „reine“ Adaptierbarkeit. Sobald ein Klärungsdialog notwendig wird, würde man eher von „systemunterstützter Adaptierbarkeit“ sprechen (s. oben).</p> |
| | System | Nutzer | | | | | | | | | | | | | | |
| Initiative | | ● | | | | | | | | | | | | | | |
| Vorschlag | | ● | | | | | | | | | | | | | | |
| Entscheidung | | ● | | | | | | | | | | | | | | |
| Ausführung | ● | | | | | | | | | | | | | | | |

Tabelle 17 Verschiedene Formen der gemischten Initiative im MyUI-System

9.2.2.4 Interpretation und Diskussion

Der beschriebene Ansatz besitzt die geforderten Eigenschaften, um die Realisierung von ABS mit gemischter Initiative zu ermöglichen. Neben automatischen Anpassungen stehen weitreichende Möglichkeiten zur nutzerinitiierten Anpassung der ABS zur Verfügung. Darüber hinaus wird in Tabelle 17 gezeigt, dass der beschriebene Ansatz verschiedene Mechanismen bietet, die es ermöglichen, dass Nutzer und System kooperativ zusammenwirken, um das Adaptationsverhalten der ABS zu steuern.

Weitere Ausprägungen der Arbeitsteilung zwischen Nutzer und System wären grundsätzlich denkbar. Einige theoretisch mögliche Kombinationen klammern jedoch bereits Dieterich et al. (1993) als uninteressant aus. Vor dem Hintergrund der hier adressierten Zielgruppe erscheint insbesondere noch die „nutzerinitiierte Adaptivität“ (vgl. Dieterich et al., 1993) überlegenswert. Dabei geht vom Nutzer lediglich der Impuls für eine Adaptation aus. Vorschlag, Entscheidung und Ausführung liegen dann beim System. So könnte man sich vorstellen, dass der Nutzer jederzeit einen Adaptationsvorgang starten könnte, wenn die Interaktion ins Stocken gerät oder Nutzungsbarrieren vorliegen. Das System könnte dann eine alternative Variante der ABS generieren. Diese Ausprägung hätte den Vorteil, dass nur geringste Anforderungen an den Nutzer gestellt würden, der lediglich signalisieren muss, wann er eine Änderung wünscht oder benötigt. Allerdings würden sich auch Probleme ergeben: Das MyUI-System adaptiert die ABS stets so, dass sie vor dem Hintergrund des verfügbaren Kontextwissens und der verwendeten Adaptationsregeln als optimal gilt. Wünscht der Nutzer nun eine Veränderung, hätte das System zunächst keinen Anhaltspunkt, wie die Zugänglichkeit erhöht werden könnte. Nur durch einen auch für den Nutzer mühsamen Prozess von Versuch und Irrtum und mehrfachem Rückgängigmachen durch den Nutzer könnte unter Umständen eine Verbesserung erzielt werden. Anhand dieser Diskussion erscheinen die angebotenen Mechanismen der gemischten Initiative als sinnvoll gewählt und ausreichend.

9.2.3 Evaluierung der kontinuierlichen Anpassungsfähigkeit während der Nutzung

9.2.3.1 Fragestellung

Während die grundsätzliche Anpassungsfähigkeit der ABS in Abschnitt 9.2.1 festgestellt wird, stehen in diesem Teil der Evaluierung die Bedingungen für die Anpassungsfähigkeit während der Nutzung im Mittelpunkt. Insbesondere geht es darum, ob die ABS *unmittelbar* und *ohne wesentliche Verzögerungen* angepasst wird, wenn sich während der Nutzung das Kontextwissen ändert. (Frage 4 der Tabelle 15; die untersuchte Anforderung ist in Abschnitt 2.2 beschrieben.)

9.2.3.2 Methode

Zur Quantifizierung der Verzögerungen bei Laufzeitanpassungen werden empirische Zeitmessungen durchgeführt. Es wird die Zeit gemessen, die ausgehend vom Zeitpunkt des

Abrufs eines neuen UP benötigt wird, um zu einer vollständig angepassten, neuen Instanz der ABS zu gelangen. Der gemessene Adaptationsprozess umfasst die folgenden Teilschritte³⁴:

1. Client fordert von Server das aktuelle UIP an.
2. Server fordert von Kontextmanagementinfrastruktur das aktuelle UP an.
3. Kontextmanagementinfrastruktur kommuniziert das aktuelle UP an Server.
4. Server transformiert die aktuellen Kontextprofile UP und DP in ein UIP (ABS-Parametrierung).
5. Server kommuniziert das aktuelle UIP an Client.
6. Client-Funktion `checkUIProfile` vergleicht neues UIP mit bisherigem UIP.
7. Client-Funktion `prepareAdaptation` wählt geeignete Interaktionsmuster und Adaptationsdialogmuster aus.
8. Client-Funktion `beforeAdaptation` entscheidet sich für ein Adaptationsdialogmuster und führt dessen Teil aus, der vor der Anpassungsdurchführung liegt.
9. Client-Funktion `performAdaptation` führt die Anpassungen mit den animierten Übergängen der Adaptationsausführungsmuster durch.

Die Zerlegung in Teilschritte macht deutlich, dass die gemessene Gesamtprozesszeit nicht nur von der Performanz der Adaptationsmechanismen abhängt, sondern auch von

- der Geschwindigkeit der Netzwerk-/HTTP-Kommunikation zwischen Client und Server und zwischen Server und Kontextmanagementinfrastruktur,
- der Verarbeitungs- und Reaktionsgeschwindigkeit der Kontextmanagementstruktur und
- der Zeit, die zur Durchführung der Adaptationsmuster benötigt wird.

Da die Performanzmessung auf die Zeiten abzielt, die als adaptionsbedingte Verzögerungen wahrgenommen werden, werden die zur Durchführung der Adaptationsmuster benötigten Zeiten von der gemessenen Gesamtprozesszeit abgezogen. Denn die darin enthaltenen Teilschritte besitzen wesentliche Funktionen im Adaptationsprozess:

- Animationen beim Ein- und Ausblenden eines Dialogfensters im Fall des Adaptationsdialogmusters „Explizite Nutzerbestätigung vor der Adaptation“ (vgl. Abschnitt 7.5.2),
- die Wartezeit, bis der Nutzer in diesem Dialogfenster eine Entscheidung getroffen hat und
- animierte Transitionen der Adaptationsausführungsmuster zur Verdeutlichung der Anpassungen (vgl. Abschnitt 7.4).

Der Testaufbau umfasst einen Client-Rechner und einen Server-Rechner, auf dem zusätzlich der MyUI Context Manager lokal installiert ist. Damit besteht zwischen dem Server und der Kontextmanagementinfrastruktur lediglich eine lokale HTTP-Kommunikation. Die wesentlichen

³⁴ Die Client-Funktionen der Teilschritte 6 bis 9 werden in Abschnitt 09.2 und Abbildung 25 genauer erläutert.

technischen Daten des Testaufbaus sind in Tabelle 18 aufgeführt.

| Client | Netzwerk | Server |
|--|---|---|
| Prozessor: Intel Core i7-3520M 2,9 GHz | 100MBit/s oder 1GBit/s Ethernet LAN | Prozessor: Intel Core i5-2400 3,1 GHz |
| Arbeitsspeicher: 8 GB | | Arbeitsspeicher: 4 GB |
| Betriebssystem: Windows 8 | | Betriebssystem: Windows 7 |
| Browser: Google Chrome 28.0 | | http-Server: Apache/2.4.3 (Win32) PHP/5.4.7 |

Tabelle 18 Technische Daten des Testaufbaus zur Messung der Prozesszeiten für Laufzeitanpassungen

Die empirischen Messungen erfolgen anhand der in Tabelle 19 dargestellten Szenarien. Die Szenarien decken sowohl umfangreiche als auch geringe UP-Veränderungen und ABS-Anpassungen ab (s. Bildschirmansichten im Anhang, Abschnitt 12.5). Für jedes Szenario werden 15 Durchläufe und Zeitmessungen durchgeführt.

| # | Ausgangszustand | Zielzustand | Anpassungen |
|---|---|--|---|
| 1 | Hauptmenü Default-UP | Hauptmenü <i>visualAcuity=2</i> , ansonsten Default-UP | Die Größe der Schrift- und Bildzeichen wird erhöht. Dadurch wird eine Vergrößerung der Schaltflächen im Menü erforderlich. Vom 4x4-Menü wird zum 3x3-Menü gewechselt. |
| 2 | Hauptmenü <i>visualAcuity=2</i> , ansonsten Default-UP | Hauptmenü <i>visualAcuity=3</i> , <i>languageReception=3</i> , <i>attention=3</i> , <i>fingerPrecision=3</i> , ansonsten Default-UP | Die Zeichengröße wird erneut erhöht. Die Darstellung wechselt in den rein grafischen Modus ohne Textelemente. Die Anzahl der Menüoptionen wird deutlich reduziert, um Aufmerksamkeitsschwächen zu kompensieren. Alle verfügbaren Optionen werden auch über Zifferntasten ansteuerbar. Die entsprechenden Ziffern werden eingeblendet. |
| 3 | Wetterprognose <i>visualAcuity=3</i> , <i>languageReception=3</i> , <i>attention=3</i> , <i>fingerPrecision=3</i> , ansonsten Default-UP | Wetterprognose Default-UP | Die Größe der Schrift- und Bildzeichen wird wieder auf die Ausgangsgröße zurückgesetzt. Vom rein grafischen Stil wird zu einer vorwiegend textuellen Darstellung gewechselt. Die Funktionstaste „Choose City“ erhält wieder eine Textbeschriftung. Die Wettervorhersage wird von einem Tag auf fünf Tage erweitert. Die Zifferntasten-Beschriftungen werden ausgeblendet. |
| 4 | Wetterprognose Default-UP | Wetterprognose <i>ICTLiteracy=2</i> , ansonsten Default-UP | Die direkte Anwahl über Zifferntasten wird ermöglicht. Die interaktiven Elemente erhalten eine entsprechende Beschriftung mit Referenz zur Zifferntaste. |

Tabelle 19 Testszzenarien zur Evaluierung der kontinuierlichen Anpassungsfähigkeit während der Nutzung

9.2.3.3 Ergebnisse

In Tabelle 20 sind die Ergebnisse der Zeitmessungen dargestellt. Um die aufgeführten Prozesszeiten zu erhalten, werden die in den beteiligten Adaptationsausführungsmustern spezifizierten Zeiten für Animationen von den gemessenen Gesamtzeiten abgezogen (s. letzte Spalte in der Tabelle). Darüber hinaus beinhalten die Szenarien 1 und 2 Bestätigungsdialoge, die eine explizite Nutzerzustimmung zur geplanten Anpassung erforderten. In diesen Fällen werden auch die Zeiten zwischen Anzeige des Dialogfensters und Betätigung des Buttons zur Bestätigung von der gemessenen Gesamtzeit abgezogen. Insgesamt ergibt sich ein homogenes Bild. Der Mittelwert aller gemessenen Prozesszeiten liegt bei 3,49 Sekunden bei einer relativ kleinen Streuung innerhalb und zwischen den Testszenarien.

| Szenario | Gemessene Prozesszeiten (nach Korrektur) | | Dauer der Animationen (nicht berücksichtigt) |
|--------------------------|--|--------------------|--|
| | Mittelwert | Standardabweichung | |
| Szenario 1 (n=15) | 3,72 Sekunden | 0,74 Sekunden | 1,6 Sekunden |
| Szenario 2 (n=15) | 3,32 Sekunden | 0,33 Sekunden | 1,6 Sekunden |
| Szenario 3 (n=15) | 3,61 Sekunden | 0,17 Sekunden | 0,5 Sekunden |
| Szenario 4 (n=15) | 3,33 Sekunden | 0,50 Sekunden | 0,5 Sekunden |
| Gesamt (n=60) | 3,49 Sekunden | 0,50 Sekunden | Unterschiedlich (s.o.) |

Tabelle 20 Ergebnisse der Zeitmessungen zur Evaluierung der kontinuierlichen Anpassungsfähigkeit während der Nutzung. Die dargestellten Prozesszeiten ergeben sich nach Abzug der Zeiten für Animationen und Nutzerbestätigungen von den gemessenen Gesamtzeiten.

9.2.3.4 Interpretation und Diskussion

Die gemessenen Prozesszeiten für Laufzeitanpassungen von durchschnittlich 3,49 Sekunden wirken kurz. Die ABS-Infrastruktur scheint daher auch für kontinuierliche Laufzeitanpassungen während der Nutzung gut gerüstet zu sein. Ein wichtiger Maßstab zur Bewertung der Verzögerungen wird aber vor allem auch die subjektive Wahrnehmung und Akzeptanz der Nutzer der ABS sein. Eine empirische Untersuchung mit potenziellen Nutzern zur gezielten Behandlung dieser Fragestellung wurde nicht durchgeführt. In anderen Nutzerstudien, die im Rahmen des MyUI-Projekts andere Forschungsziele adressierten, wurde jedoch nie Kritik an zu langen Verzögerungen geäußert. Weder gab es explizite Kommentare der Probanden noch entsprechende Berichte der Testleiter. Es ist daher eher nicht zu erwarten, dass die gemessenen Verzögerungen von Nutzern der ABS als störend empfunden werden und sich daraus Akzeptanzprobleme ergeben könnten. Dies gilt insbesondere auch im Zusammenhang mit webbasierten Applikationen, bei denen die Benutzer daran gewöhnt sind, dass Datenabrufe vom Server oder aus dem Internet eine gewisse Zeit erfordern. Des Weiteren ist zu berücksichtigen, dass die gemessenen Verzögerungen nie als reine, durchgängige Wartezeiten erlebt werden, da sie immer wieder durch Animationen oder Bestätigungsdialoge unterbrochen werden.

Die relativ homogenen Verzögerungen lassen sich dadurch erklären, dass unabhängig vom Ausmaß der Adaptationen immer dieselben Prozessschritte der erneuten ABS-Generierung und

entsprechender Übergänge von einer zur nächsten ABS-Instanz durchgeführt werden. Dadurch ergeben sich auch für umfangreichere Veränderungen keine längeren Prozesszeiten.

9.2.4 Evaluierung der umfassenden Anpassungsfähigkeit

9.2.4.1 Fragestellung

Die umfassende Anpassungsfähigkeit wird anhand von zwei Fragestellungen bewertet:

- Ermöglicht die ABS-Infrastruktur eine Adaptivität an Veränderungen im Wissen über Nutzer (perzeptuelle, motorische und kognitive Eigenschaften), Umgebung und technische Ausstattung? (Frage 7 der Tabelle 15; die untersuchte Anforderung ist in Abschnitt 2.4 beschrieben.)
- Ermöglicht die ABS-Infrastruktur eine Adaptivität der Informationsdarstellung, der Interaktion und der Navigation? (Frage 8 der Tabelle 15; die untersuchte Anforderung ist in Abschnitt 2.4 beschrieben.)

9.2.4.2 Methode

Der Ansatz gilt als umfassend anpassungsfähig, wenn beide Fragestellungen positiv beantwortet werden. Unter Bezugnahme auf die Darstellung des Ansatzes (Kapitel 3 bis 8) wird aufgezeigt, wie und in welchem Umfang die geforderten Merkmale umgesetzt sind.

9.2.4.3 Ergebnisse

Die Anpassungsfähigkeit an Nutzer, Umgebung und technische Ausstattung ist insbesondere durch die Mechanismen der ABS-Parametrierung (Kapitel 5) gewährleistet. Eine wesentliche Voraussetzung ist die Existenz der Kontextprofile UP und DP. Das UP umfasst Variablen zu perzeptuellen (z.B. *visualAcuity* und *hearing*), motorischen (z.B. *speechArticulation* und *fingerPrecision*) und kognitiven Nutzereigenschaften (z.B. *languageReception* und *attention*) sowie zu bestimmten Eigenschaften der Umgebung (*ambientNoise* und *ambientLight*) (vgl. Abschnitt 5.4). Das DP beschreibt die Darstellungs- und Interaktionskapazitäten der technischen Ausstattung (vgl. Abschnitt 5.2). Gemeinsam stellen DP und UP sicher, dass das aktuell verfügbare Kontextwissen in der ABS-Infrastruktur repräsentiert ist. Die Individualisierungsmuster „übersetzen“ diese Profile dann in grundlegende Merkmale der Benutzungsschnittstelle und erzeugen bzw. aktualisieren das UIP (vgl. Abschnitte 5.5 und 5.6). Die Werte des UIP werden dann sowohl bei der Auswahl der geeigneten Interaktionsmuster als auch bei der Anpassung der verwendeten Bedien- und Anzeigeelemente herangezogen, um die ABS an die spezifischen Anforderungen anzupassen (s. Kapitel 6). Auch bei Laufzeitanpassungen werden die Einstellungen des UIP berücksichtigt, um die Geschwindigkeit der animierten Übergänge (Adaptationsausführungsmuster, Abschnitt 7.4) anzupassen.

Damit bietet der Ansatz die Voraussetzungen für Adaptivität an die spezifischen Eigenschaften des Nutzers, der Umgebung und der technischen Ausstattung. Wie und in welchem Umfang sich eine ABS dann tatsächlich anpassen kann, hängt jedoch nicht nur vom ABS-Rahmenmodell dieser Arbeit ab, sondern auch erheblich von den konkreten Interaktionsmustern und Bedien- und

Anzeigeelementen. Das AAIM modelliert die ABS unabhängig vom konkretem Erscheinungsbild und konkreten Interaktionsmechanismen (vgl. Kapitel 8). Dadurch ist eine Anpassungsfähigkeit der Informationsdarstellung und der Interaktion bereits angelegt. Variablen des UIP, wie zum Beispiel *displayMode*, *bodyTextFontSize* und *audioOutputVolume* bzw. *numericInput* und *voicelInput* bieten Ankerpunkte für die Auswahl geeigneter Interaktionsmuster, um verschiedene Varianten der Darstellung bzw. der Interaktionsmodalität zu erzeugen. Am Beispiel des Interaktionsmusterbündels Main Menu³⁵ der Referenzimplementierung wird gezeigt, wie eine Anpassung der Darstellung und der Interaktion mit den Mitteln des ABS-Rahmenmodells zu erreichen sind. Weniger ausgeprägt ist die Anpassungsfähigkeit der Navigation. Durch die Definition der Applikationszustände und der möglichen Transitionen zwischen den Zuständen im AAIM ist ein Skelett der Navigationsstruktur vorgegeben. Alle Variationen der Navigationspfade sind auf diesen Rahmen festgelegt. Die einzige Form der Variantenbildung, die sich direkt aus dem beschriebenen Ansatz ergibt, ist die Verteilung eines Applikationszustands auf unterschiedlich viele Bildschirmansichten. Ein Beispiel hierfür ist das Hauptmenü, das bei Standardschriftgröße und Bedienung per Cursortasten oder Zeigegerät auf einem einzigen Bildschirm dargestellt wird. Wenn jedoch die Schrift- und Zeichengröße erhöht wird (UIP-Variable *bodyTextFontSize*), die maximale Gesamtzahl der angezeigten Elemente reduziert wird (UIP-Variable *screenComplexity*) oder die Bedienung per Nummerntasten der Fernbedienung erfolgt (UIP-Variable *numericInput* = on), wird es auf mehrere sequenzielle Bildschirmansichten verteilt (s. Abbildung 32 (1) und (2) im Anhang). Über diese Mechanismen kann beispielsweise auch ein Formular mit umfangreichen Nutzereingaben zu einem sogenannten „Wizard“ angepasst werden, bei dem der Nutzer schrittweise von einer zur nächsten Eingabe geführt wird und pro Bildschirmansicht nur ein Eingabefeld dargestellt wird.

9.2.4.4 Interpretation und Diskussion

Zusammenfassend wird festgestellt, dass die ABS-Infrastruktur Adaptationen ausführen kann, die im Sinn dieser Arbeit als „umfassend“ bezeichnet werden (s. Abschnitt 2.4). Es stehen Mechanismen für Anpassungen in allen geforderten Bereichen zur Verfügung. Gewisse Einschränkungen bestehen jedoch in zwei Bereichen:

Einerseits liegt im aktuellen UP nur wenig Kontextwissen über die Nutzungsumgebung vor. Lediglich die beiden Variablen *ambientLight* und *ambientNoise* reflektieren Einschränkungen, die sich durch spezielle Umgebungsbedingungen ergeben. Die Vorgehensweise des beschriebenen Ansatzes, Umgebungsbedingungen direkt im Sinn von Beeinträchtigungen der Techniknutzung zu betrachten, ist sicher sinnvoll. Diese Perspektive schränkt die Menge der relevanten Umgebungsbedingungen deutlich ein. Eine weitere Einschränkung ergibt sich durch die Effizienz der Kontexterkenkung. Der Nutzen zusätzlicher Umgebungsinformation muss stets in einem vernünftigen Verhältnis zum erforderlichen technischen Aufwand und zur möglichen Beeinträchtigung der Privatsphäre des Nutzers stehen. Dennoch wäre es für zukünftige Arbeiten mit dem MyUI-Rahmenmodell interessant, weitere Umgebungsinformationen zu integrieren, um zum Beispiel besondere mobile und soziale Nutzungssituationen noch besser zu unterstützen. Dies wäre insbesondere auch dann interessant, wenn andere Anwendungsdomänen wie

³⁵ http://myui-patterns.iao.fraunhofer.de/index.php/Main_Menu

beispielweise ABS zur Steuerung und Überwachung industrieller Maschinen adressiert werden. Die Erweiterbarkeit der MyUI-Profile und des Entwurfsmusterrepositoriums bieten dafür gute Voraussetzungen.

Andererseits ermöglicht der Ansatz nur rudimentäre Anpassungen der möglichen Navigationspfade. Ein wesentlicher Grund hierfür liegt in der Interaktionsmodellierung über Zustandsdiagramme, die bereits eine gewisse Navigationsstruktur vorgibt. Zugunsten eines möglichst geringen Spezifikationsaufwands und einer möglichst geringen Einstiegshürde verzichtet der Ansatz auf eine ABS-Modellierung auf unterschiedlichen Ebenen, wie es bei einigen akademischen Ansätzen vorgesehen ist (vgl. Abschnitt 2.6.1). Eine zusätzliche Aufgaben- und Objektmodellierung (z.B. Mori et al., 2004; Paterno et al., 2009; Szekely, 1996 oder Sottet et al., 2007a) könnte vielfältigere Adaptationen der Navigationsstruktur ermöglichen. Allerdings scheint die Frage berechtigt, ob nicht die bereits umgesetzte Vereinfachung und Sequenzialisierung umfangreicher Bildschirmansichten durch die Verteilung einzelner Interaktionszustände auf mehrere Interaktionsschritte ausreicht, um wesentliche Nutzungsbarrieren zu überwinden. Es gilt abzuwägen, ob der erforderlichen Komplexitätszunahme in den Bereichen der Modellierung und der ABS-Generierung tatsächlich ein entsprechender Gegenwert für die Zugänglichkeit der ABS gegenüber steht.

Allerdings ist zu beachten, dass das in dieser Arbeit verwendete Konzept der „umfassenden Adaptivität“ eine wichtige Kategorie von Nutzungsbarrieren nicht ausreichend berücksichtigt: Nämlich die Barrieren, die sich bei der interkulturellen Nutzung von interaktiven Systemen ergeben können. Für eine Anpassung der auf der Benutzungsschnittstelle verwendeten Sprache finden sich grundlegende Voraussetzungen bereits in der MyUI-Infrastruktur. Wie in der zweiten Ansicht der Abbildung 32 im Anhang dargestellt, kann die kognitive UP-Variable *languageReception*

(s. Abschnitt 5.4) zu einer rein grafischen Ansicht ohne Textelemente führen. Eine gleichwertige Darstellung der ABS in einer anderen Sprache ist jedoch aktuell noch nicht umgesetzt. Die bereits angelegte UP-Variable *preferredLanguage* bietet lediglich einen Ausgangspunkt für mögliche Erweiterungen in Richtung Mehrsprachigkeit. Eine einfache Lösung könnte über das Customization-Profil (CP) erreicht werden (vgl. Abschnitt 5.7). Ähnlich wie für die Icons und Klänge könnten auch für die verwendeten Textelemente verschiedene Bibliotheken für die zu unterstützenden Sprachen angelegt werden, die dann über eine CP-Variable (entsprechend zu *graphicsLibrary* und *soundsLibrary* in Tabelle 9) aufgerufen werden. Damit würden sich die Rolle und der Einsatz des CP deutlich erweitern. Die Referenzierung eines spezifischen CP müsste dann über eine zweidimensionale UIP-Variable *skin* erfolgen, die dann neben der Bezeichnung des Markenbildes (Corporate Identity) auch die Sprachversion spezifiziert (z.B. {MyUI; German}). Damit wäre zunächst das Übersetzungsproblem im Rahmen der MyUI-Infrastruktur gelöst. Eine echte Anpassung an unterschiedliche kulturelle Standards würde noch deutlich umfangreichere Erweiterungen erfordern. Moderne Ansätze des interkulturellen Designs von Benutzungsschnittstellen gehen davon aus, dass nahezu in allen Gestaltungsbereichen Anpassungen an unterschiedliche Kulturen sinnvoll sind (vgl. Marcus, 2011). Unterschiedliche Farben und Icons könnten dabei noch leicht über das CP zur Verfügung gestellt werden. Andere Variationen wie zum Beispiel in den Navigationsmechanismen (komplexe vs. schlichte Strukturen, unterschiedliche Bestätigungsmechanismen, etc.) und in Darstellungsaspekten wie Textgröße auf Buttons, Verhältnis von Text zu leerem Hintergrund, Bild-Text-Verhältnis und Layoutausrichtung könnten nur über zusätzliche Entwurfsmuster in Kombination mit zusätzlichen UP- und UIP-Variablen erreicht werden. Grundsätzlich bietet die MyUI-Infrastruktur die dafür notwendigen

Voraussetzungen und Mechanismen und sowohl die Profile als auch das Entwurfsmusterrepositorium sind erweiterbar. Bei einer umfangreichen Abdeckung verschiedener Kontextbedingungen, die auch unterschiedliche Kulturen umfasst, würde jedoch die Anzahl der benötigten Entwurfsmuster schnell massiv zunehmen. Für jedes neue Individualisierungsmuster, das einen neuen UIP-Variablenwert setzt, müssten in mehreren Bündeln von Interaktionsmustern neue Varianten angelegt werden (vgl. die Beschreibung zu Erweiterungen bei Abdeckung zusätzlicher Nutzer- und Umgebungsbedingungen in Abschnitt 4.4.2).

9.2.5 Evaluierung der Modularität

9.2.5.1 Fragestellung

Unterstützt die ABS-Infrastruktur eine Implementierung von ABS in Modulen, die eigenständig und in sich abgeschlossen sind? (Frage 9 der Tabelle 15; die untersuchte Anforderung ist in Abschnitt 2.5 beschrieben.)

9.2.5.2 Methode

Zur Überprüfung der Modularität wird untersucht, inwieweit die definitorischen Merkmale einer modularen Software-Architektur im beschriebenen Ansatz realisiert sind (vgl. Balzert, 1996 und Pomberger, 2002):

- Die Operationen und Daten des Gesamtsystems werden auf mehrere Module verteilt. Dabei ist ein Modul „eine Sammlung von Algorithmen und Datenstrukturen zur Bearbeitung einer in sich abgeschlossenen Aufgabe“ (Pomberger, 2002).
- Die einzelnen Module sind in einen Rumpf und eine Schnittstelle gegliedert.
- Der Rumpf eines Moduls enthält die Implementierung der in der Schnittstelle definierten Dienstleistungen. Datenstruktur und Prozeduren der Implementierung sind für die anderen Module verborgen. Änderungen im Rumpf haben keine Auswirkungen auf die Benutzer eines Moduls (die anderen Module).
- Das Zusammenspiel der Module ist durch Schnittstellen festgelegt.
- Ein Modul besitzt eine Exportschnittstelle zur Definition, was es anderen Modulen zur Verfügung stellt bzw. welche Dienstleistung es exportiert und eine Importschnittstelle zur Angabe, was es von anderen Modulen benutzt.
- Die Beziehungen zwischen Modulen lassen sich durch die „Benutzbarkeit“ und das „Enthaltensein“, das die lokale Benutzbarkeit der enthaltenen Module ausdrückt, beschreiben.

9.2.5.3 Ergebnisse

Das MyUI-Rahmenmodell ist auf mehreren Ebenen durchgängig modular aufgebaut. Die drei Phasen der ABS-Parametrierung, ABS-Vorbereitung und ABS-Generierung und Adaptation können als modulare Teilsysteme (vgl. Balzert, 1996) des Gesamtprozesses verstanden werden. Jedes dieser Teilsysteme hat klar definierte Import- und Exportschnittstellen: Die ABS-

Parametrierung benutzt die von der Kontextmanagementinfrastruktur bzw. dem Gerätemanager zur Verfügung gestellten Profile UP und DP als Importschnittstelle. Die erbrachte Dienstleistung schlägt sich im UIP als Exportschnittstelle nieder (vgl. Kapitel 5). Die ABS-Vorbereitung importiert das UIP und stellt als Ergebnis eine Auswahl von geeigneten Interaktionsmustern und Bedien- und Anzeigeelementen bereit (vgl. Kapitel 6). Zusammen mit der aktuellen ABS stellt diese Auswahl die Importschnittstelle der ABS-Generierung und Adaptation dar. Die exportierte Leistung ist eine generierte bzw. aktualisierte ABS (vgl. Kapitel 7).

Die in den Rümpfen dieser Teilsysteme wirksamen Entwurfsmuster können selbst auch als Module betrachtet werden. Jede Entwurfsmusterbeschreibung enthält eine Problembeschreibung, die angibt, welche Leistung das Muster zur Verfügung stellt (Exportschnittstelle). Die Lösungsbeschreibung kann gemeinsam mit ihrer Implementierung (ausführbare Softwarekomponente, vgl. Abschnitt 4.1) als Rumpf des Musters betrachtet werden. Generell gilt, dass die im Rumpf deklarierten Prozeduren nach außen nicht sichtbar sind und dass Änderungen im Rumpf keine Auswirkungen auf die anderen Muster haben. Die vorhandenen Importschnittstellen sind in den Vorlagen zur Musterbeschreibung nicht durchgängig mit derselben Präzision dokumentiert. Zwar gibt es bei den gerätespezifischen Mustern (*D*), den Individualisierungsmustern (*I*) und den Interaktionsmustern (*K*) die Felder „Requires <variable>“ bzw. „Requires“ und „Uses <variable>“, die definieren, welche Ergebnisse anderer Muster in den Kontext- und Lösungsbeschreibungen benutzt werden (s. Tabelle 4, Tabelle 8 und Tabelle 11). Bei den Bedien- und Anzeigeelementen (*B*) finden sich explizite Verweise jedoch nur auf andere Muster, die erweitert werden („Extended by“). UIP-Variablen, die zu unterschiedlichen Darstellungen des Musters führen, werden lediglich implizit in der Lösungsbeschreibung genannt (Tabelle 12). Ähnlich verhält es sich mit den Adaptationsausführungsmustern (*AA*). Zugunsten einer Verdeutlichung der Trennung von Rumpf und Importschnittstelle, sollten die anderen Muster aus *AA*, mit denen das beschriebene Muster die Relation *requires* besitzt, nicht nur im Lösungsfeld referenziert werden (Tabelle 13).

Zwischen den einzelnen Entwurfsmustertypen besteht eine strenge funktionale Arbeitsteilung und Kapselung. Jeder Mustertyp erfüllt ganz spezifische Funktionen im Adaptationsprozess, die von den anderen Mustertypen nicht erfüllt werden (vgl. Abschnitt 4.2). Entsprechend der Enthaltensein-Beziehung modularer Architekturen (vgl. Balzert, 1996) sind die Entwurfsmuster je nach Typ nur lokal in dem Teilsystem (d.h. der Phase des Adaptationsprozesses) benutzbar, in dem sie enthalten sind. Ein gutes Beispiel hierfür sind die Bedien- und Anzeigeelemente, die einzelne generische ABS-Bausteine zur Verfügung stellen. Sie können nur von Interaktionsmustern benutzt werden. Sie entscheiden selbst über die Details ihres Aussehens, ohne dass die Interaktionsmuster, die sie benutzen, ihr konkretes Aussehen kennen oder beeinflussen können (vgl. Abschnitt 6.2).

Innerhalb der Mustertypen ergibt sich wiederum eine Funktionsverteilung über die unterschiedlichen Problemargumente der einzelnen Muster. Entwurfsmuster, die dasselbe Gestaltungsproblem adressieren, sind in Musterbündel zusammengefasst und dadurch von Entwurfsmustern zu anderen Problemen getrennt. Die Bündel von Entwurfsmustern aus *D*, *I*, *K* und *AD* besitzen selbst wieder Moduleigenschaften. In den Beschreibungen der Musterbündel von *D* und *I* werden die exportierten Dienstleistungen im Problemfeld beschrieben und im Feld

„Sets variables“ spezifiziert (s. Tabelle 3 und Tabelle 7). Genauso könnte man auch mit den Importschnittstellen verfahren. Während der Implementierung des Ansatzes wurde im Musterrepositorium³⁶ bereits für einige Individualisierungsmusterbündel angegeben, welche UP- und DP-Variablen als Eingangsgrößen verwendet werden, um das am besten geeignete Muster auszuwählen (z.B. Abbildung 36 im Anhang). In den Beschreibungen der Bündel von *K* werden Import- und Exportschnittstellen aus der Perspektive der Interaktions- und Applikationslogik beschrieben („IN“ und „OUT“ im Feld „Parameters“ in Tabelle 10). Eine zusätzliche Benennung der für die Auswahl des geeigneten Interaktionsmusters relevanten UIP-Variablen könnte leicht aus den Musterbeschreibungen abgeleitet werden, ist jedoch in der aktuellen Form nicht vorgesehen.

Schließlich sind die Beziehungen zwischen den verschiedenen Mustern und Mustertypen in Abschnitt 4.3 auch über die Grenzen der Teilsysteme hinweg (Phasen der ABS-Generierung und Adaptation) klar definiert. Die Relation *substituiert* gibt an, dass zwei Muster dasselbe Problem adressieren und damit im selben Bündel enthalten sind. Alle weiteren Relationen sind als Variationen der Benutzbarkeitsbeziehung zu verstehen. In den Relationen *setzt* *<Variable>* wie *benötigt von*, *setzt* *<Variable>* wie *genutzt von* und ihren Umkehrrelationen werden dabei die Profile DP und UIP als Schnittstellen verwendet.

9.2.5.4 Interpretation und Diskussion

Die ABS-Infrastruktur besitzt die wesentlichen Eigenschaften für modular aufgebaute ABS. Die identifizierten Teilsysteme ermöglichen eine effektive Arbeitsteilung zur ABS-Generierung und Adaptation. Die Entwurfsmuster und Musterbündel können als Module mit definierten Schnittstellen betrachtet werden. Dadurch ergeben sich beste Voraussetzungen für deren Wiederverwendung und Weiterentwicklung sowie für Erweiterungen des Entwurfsmusterbestands.

Durch eine exakte Spezifikation aller Schnittstellen könnte die Modularität des Ansatzes noch besser verdeutlicht werden. So sollte beispielsweise die Angabe der Importschnittstellen als fester Bestandteil in die Beschreibungsvorlagen der Musterbündel in *D*, *I* und *K* sowie der Entwurfsmuster *B* und *AA* integriert werden.

9.2.6 Evaluierung der Erweiterbarkeit

9.2.6.1 Fragestellung

Erlaubt die ABS-Infrastruktur Erweiterungen um zusätzliche Kontextbedingungen und Interaktionslösungen? (Frage 10 der Tabelle 15; die untersuchte Anforderung ist in Abschnitt 2.5 beschrieben.)

³⁶ <http://myui-patterns.iao.fraunhofer.de>

9.2.6.2 Methode

Anhand der Beschreibung des Ansatzes (Kapitel 3 bis 8) wird geprüft, in wie weit die ABS-Infrastruktur zukünftige Erweiterungen um zusätzliche Kontextbedingungen und Interaktionslösungen unterstützt.

9.2.6.3 Ergebnisse

In Abschnitt 4.4.2 (Erweiterbares Repositorium) wird ein generisches Modell für Erweiterungen des Entwurfsmusterrepositoriums eingeführt (s. Abbildung 6). Dort werden die Prozesse verschiedener Erweiterungen des Entwurfsmusterrepositoriums detailliert beschrieben. Es werden sowohl Erweiterungen um neue Kontextbedingungen als auch um neue Interaktionslösungen behandelt. Die beschriebenen Kontexterweiterungen decken alle relevanten Fälle zusätzlicher Nutzereigenschaften, Umgebungsbedingungen und technischer Geräte ab. Im Zusammenhang mit zusätzlichen Interaktionslösungen werden zwei Fälle beschrieben: einerseits die Erweiterung um neue Interaktionssituationen, andererseits die Erweiterung um zusätzliche Lösungen für bereits behandelte Interaktionssituationen. Damit kann der Ansatz als erweiterbar im oben beschriebenen Sinn betrachtet werden.

9.2.6.4 Interpretation und Diskussion

Es wird deutlich, dass die ABS-Infrastruktur gezielt auf Erweiterbarkeit ausgelegt ist. Mechanismen für die Durchführung verschiedener Erweiterungen sind fester Bestandteil des Ansatzes. Die Beschreibungen in Abschnitt 4.4.2 gehen jedoch auch auf Schwierigkeiten ein, die bei Erweiterungen auftreten können. Einerseits entstehen durch die Benutzbarkeitsbeziehungen zwischen den verschiedenen Entwurfsmustern (`benötigt`, `setzt <Variable> wie benötigt von` und `setzt <Variable> wie genutzt von` und deren Umkehrrelationen) Abhängigkeiten, die bei Erweiterungen berücksichtigt werden müssen, um eine maximale Effektivität der neu hinzugefügten Komponenten über alle drei Phasen des ABS-Generierungsprozesses zu gewährleisten. Andererseits erfordern Erweiterungen innerhalb bestehender Musterbündel eine genaue Berücksichtigung der If-Statements aller bereits bestehenden Muster des zu erweiternden Bündels, um dort eine überschneidungsfreie und vollständige Abdeckung der Kontextbedingungen zu sichern. Diese beiden Schwierigkeiten lassen sich jedoch aufgrund der Modularität des Ansatzes (vgl. Abschnitt 9.2.5) und der vorhandenen Schnittstellen gut eingrenzen und leicht bewältigen.

Eine weitere Schwierigkeit ergibt sich daraus, dass Erweiterungen nur auf Ebene der natürlichsprachlichen Entwurfsmusterbeschreibungen behandelt werden. Zwar lassen sich die dafür vorgesehenen Prinzipien leicht auf die ausführbaren Softwarekomponenten der Entwurfsmuster übertragen. Jedoch werden keine Mechanismen beschrieben, die bei Erweiterungen die Konsistenz zwischen natürlichsprachlicher und software-technischer Repräsentation der Entwurfsmuster sichern könnten. Für eine Unterstützung der praktischen Durchführung von Erweiterungen wäre ein definierter Prozess hilfreich, der u.a. klärt, welche der beiden Repräsentationen die führende ist, nach der die andere angepasst wird. Interessant könnten in diesem Zusammenhang auch automatische Parser und Code-Generatoren sein, die die natürlichsprachlichen Beschreibungen in ausführbaren Code überführen. Mit Generatoren zu unterschiedlichen Benutzungsschnittstellentechnologien könnte auch die Technologie-

unabhängigkeit des Entwurfsmusteransatzes gestärkt werden. Neben den technischen Herausforderungen, die eine automatische Code-Generierung mit sich bringen würde, liegt das größte Problem jedoch in der notwendigen Strukturierung und Regulierung der Musterbeschreibungen. Um ein automatisches Parsing zu unterstützen, müssten Regeln für die Syntax der Beschreibungen definiert und eingehalten werden. Dies würde vor allem zwei wesentliche Nachteile nach sich ziehen: Erstens könnten die Entwurfsmusterbeschreibungen nicht mehr frei in natürlicher Sprache formuliert werden. Die Entwurfsmusterautoren müssten die anzuwendenden Regeln erlernen. Es ist zu erwarten, dass über diese Hürde die Anzahl der beitragenden Musterautoren und der Gesamtumfang der externen Beiträge deutlich reduziert würden (vgl. Abschnitt 9.2.7). Zweitens wären zukünftige Erweiterungen auf die vordefinierten Ausdrucksmöglichkeiten beschränkt. Um dies zu umgehen, müssten auch die automatischen Transformatoren selbst erweiterbar sein. Dadurch würde wiederum eine Redundanz geschaffen werden. Erweiterungen des Entwurfsmusterrepositoriums würden unter Umständen zusätzlich aufwändige Erweiterungen der Parser und Generatoren erfordern.

9.2.7 Evaluierung der Möglichkeit, die Adaptationsmechanismen zu veröffentlichen und für externe Beiträge zu öffnen

9.2.7.1 Fragestellung

Bietet die ABS-Infrastruktur Möglichkeiten, um die Adaptationsmechanismen für die Öffentlichkeit zugänglich zu machen und für externe Beiträge zu öffnen? (Frage 11 der Tabelle 15; die untersuchte Anforderung ist in Abschnitt 2.5 beschrieben.)

9.2.7.2 Methode

Für die Evaluierung wird ein zweistufiges Vorgehen gewählt:

- Auf Grundlage der Beschreibung des Ansatzes (Kapitel 3 bis 8) wird untersucht, welche Voraussetzungen die ABS-Infrastruktur theoretisch zur Verfügung stellt.
- Anhand einer Beispielimplementierung eines öffentlichen Entwurfsmusterrepositoriums im Rahmen des MyUI-Forschungsprojekts und dessen aktive Nutzung durch Projektpartner wird evaluiert, wie sich die Öffnung des Repositoriums in der Praxis umsetzen lässt.

9.2.7.3 Ergebnisse

Abschnitt 4.4.3 (Öffentlich zugängliches Repositorium) widmet sich ausschließlich den Möglichkeiten zur Veröffentlichung der Adaptationsmechanismen und für eine Beteiligung externer Experten bei deren Weiterentwicklung und Erweiterung. Der Ansatz sieht eine internetbasierte Kollaborationsplattform vor, die sowohl die Veröffentlichung der Adaptationsmechanismen als auch die Integration externer Beiträge umfasst. Die Umsetzungsdetails sind jedoch nicht Bestandteil dieser Arbeit. Einzelheiten der Gestaltung und technischen Implementierung werden nicht behandelt. Die wichtigsten Eigenschaften der vorgesehenen Plattform werden lediglich in Form von Anforderungen benannt. Damit wird festgestellt, dass die

Öffnung der Adaptationsmechanismen im MyUI-Rahmenmodell einen hohen Stellenwert besitzt und dass die grundsätzlichen Voraussetzungen dafür in der ABS-Infrastruktur erfüllt sind.

Im Zuge der praktischen Evaluierung wurde ein öffentlich zugängliches MediaWiki³⁷ erstellt, das eine Veröffentlichung aller aktuellen Entwurfsmuster darstellt. Damit gilt der erste Teil der Fragestellung als empirisch validiert: Die Adaptationsmechanismen können der Öffentlichkeit zugänglich gemacht werden. Dank der klaren Strukturierung der Entwurfsmuster (vgl. Abschnitt 4.2) und ihrer natürlichsprachlichen Beschreibungen und Illustrationen kann davon ausgegangen werden, dass es sich hierbei nicht nur um die technische Möglichkeit des Datenabrufs handelt, sondern dass tatsächlich ein relativ breites Publikum die veröffentlichten Anpassungsmechanismen auch verstehen und nachvollziehen kann – auch wenn keine systematische Erhebung der Verständlichkeit durch Personen, die nicht aktiv in die Projektbearbeitung oder die Begutachtung des Projekts einbezogen waren, durchgeführt wurde. Eine Einschränkung der aktuellen Veröffentlichung ist jedoch darin zu sehen, dass nur die Adaptationsmechanismen veröffentlicht sind, die sich in den Entwurfsmustern niederschlagen. Damit ist im aktuellen MediaWiki zwar öffentlich beschrieben, *welche Anpassungen* durchgeführt werden. Jedoch ist aufgrund der technologieunabhängigen Entwurfsmusterbeschreibungen nicht genau dokumentiert, *wie* sie technisch durchgeführt werden. Dies könnte jedoch insbesondere für Entwickler von ABS interessant sein.

Der zweite Teil der Fragestellung bezieht sich auf die Einbeziehung externer Beiträge zur kooperativen Erweiterung des Entwurfsmusterrepositoriums. Zwar waren während der Erprobungsphase nur Mitglieder des Projektteams beteiligt. Dennoch lässt sich anhand dieser internen Erprobung die grundsätzliche Machbarkeit der Öffnung für externe Beiträge bestätigen.

9.2.7.4 Interpretation und Diskussion

Die MyUI-Infrastruktur sieht eine öffentliche Zugänglichkeit der Adaptationsmechanismen für lesenden und schreibenden Zugriff vor und bietet dafür die grundlegenden Voraussetzungen. Dies gilt jedoch nur in den Grenzen des in dieser Arbeit beschriebenen Ansatzes. Insbesondere bei der praktischen Evaluierung durch die Beispielimplementierung des Musterrepositoriums als MediaWiki wird deutlich, dass sich bei einer breiteren Öffnung der Adaptationsmechanismen Herausforderungen ergeben, zu denen das ABS-Rahmenmodell dieser Arbeit noch keine Lösungsansätze bietet:

- Aktive Beiträge erfordern eine genaue Kenntnis des Ansatzes. Dadurch ergibt sich eine gewisse *Einstiegshürde*. Insbesondere ist es wichtig, die funktionale Kapselung der einzelnen Mustertypen und die verwendeten Prinzipien der Variantenbildung genau zu kennen. Selbst innerhalb des Projektteams konnte dies nicht uneingeschränkt vorausgesetzt werden. Es zeigte sich, dass der Ansatz von einigen Kollegen als komplex empfunden wurde und dass der Einstieg in die aktive Mitarbeit am Repositorium teilweise schwer fiel. Es wurde deutlich, wie wichtig es sein wird, gutes Informationsmaterial in das MediaWiki zu integrieren. Insbesondere werden knappe und klare Beschreibungen der wichtigsten Grundlagen benötigt und instruktive Beispiele, die die Arbeit mit den

³⁷ <http://myui-patterns.iao.fraunhofer.de>

Entwurfsmustern illustrieren. Als Ergebnis dieser Erkenntnis wurden bereits erste Erläuterungen und Beispiele in das Wiki eingepflegt³⁸.

- Zwar kann die Sammlung der Entwurfsmusterbeschreibungen von jedem im MediaWiki angemeldeten Benutzer kommentiert, verändert und erweitert werden. Jedoch stehen noch keine erprobten Mechanismen zur Verfügung, um sicher zu stellen, dass zu einer neuen Entwurfsmusterbeschreibung auch eine *entsprechende und konsistente Softwarekomponente* vorliegt. Dieses Problem wird bereits in Abschnitt 9.2.6 (Evaluierung der Erweiterbarkeit) näher beschrieben.
- Die Mitarbeit verschiedener Entwurfsmusterautoren führt zu einem heterogenen Qualitätsniveau der Inhalte des Repositoriums. Unerwünschte Redundanzen und Inkonsistenzen können leicht entstehen. Für die erfolgreiche Umsetzung eines offenen Repositoriums wird es wichtig sein, geeignete Prinzipien der Organisation und Verwaltung des Repositoriums zu definieren. Insbesondere stellt sich die Frage, in wie weit man z.B. ähnlich wie bei Wikipedia³⁹ auf die Prinzipien der Selbstorganisation vertrauen kann, oder ob eine zentrale Redaktion zur Qualitätssicherung benötigt wird. Weisbecker (2002) schreibt beispielsweise in ihren Ausführungen zu Komponentenarchiven von einem „Wiederverwendungsteam“, das unter anderem Kommentare der Archivanwender auswertet, um Verbesserungen und Erweiterungen vorzunehmen und für die Verwaltung und Archivierung der wiederverwendbaren Komponenten zuständig ist.
- Die korrekte Angabe der Schnittstellen der Entwurfsmuster wird aufgrund der starken Vernetzung der Adaptationsmechanismen schnell zu einer komplexen und unübersichtlichen Aufgabe. Die aktuelle Implementierung als MediaWiki stellt keinerlei Mechanismen zur Verfügung, um sicher zu stellen, dass die Relationen zwischen den einzelnen Mustern stets widerspruchsfrei und vollständig aufgeführt sind. Wünschenswerte Funktionen zum besseren Management des Repositoriums betreffen insbesondere die *Konsistenzsicherung*. Sie umfassen u.a. die automatische Ergänzung von Relationen, die sich z.B. aus Umkehrrelationen ergeben, oder eine stärkere Strukturierung der Eingabeformulare zur Musterbeschreibung, um eine Beschränkung auf zulässige Terminologie und Syntax (z.B. in den Feldern zu Pseudocode und Relationen) zu erzwingen.
- Weitere Standardfunktionen bzw. „Basisdienste“ (Höb & Weisbecker, 2002) eines Komponentenarchivs, die für zukünftige Implementierungen interessant wären, betreffen vor allem das *Auffinden von Entwurfsmustern* über verschiedene Einstiege und Suchmechanismen.
- Schließlich wird die Effektivität des Ansatzes auch davon abhängen, wie viel Engagement externer Autoren für eine Mitarbeit an Erweiterungen der Adaptationsmechanismen gewonnen werden kann. Daher wird eine wesentliche Herausforderung darin bestehen,

³⁸ Siehe z.B. <http://myui-patterns.iao.fraunhofer.de/index.php/Introduction> und http://myui-patterns.iao.fraunhofer.de/index.php/Example_1:_Demo_Storyboard_for_End_User_Perspektive

³⁹ http://en.wikipedia.org/wiki/Reliability_of_Wikipedia

geeignete *Anreizmechanismen* zu schaffen, um Autoren zu Beiträgen und Kommentaren zu motivieren (vgl. HöB, 2005).

9.2.8 Evaluierung des sukzessiven Effizienzgewinns bei der Entwicklung weiterer adaptiver Benutzungsschnittstellen

9.2.8.1 Fragestellung

Kann durch wiederverwendbare Komponenten der Aufwand zur Entwicklung neuer Applikationen bzw. neuer ABS sukzessive verringert werden? (Frage 13 der Tabelle 15; die untersuchte Anforderung ist in Abschnitt 2.6.1 beschrieben.)

9.2.8.2 Methode

Es wird ein dreistufiges Evaluierungsverfahren gewählt, das prüft, in wie weit das Prinzip der Wiederverwendung auf das MyUI-Rahmenmodell anwendbar ist. Dabei wird auf das bei HöB (2005) beschriebene Verständnis von Wiederverwendbarkeit zurückgegriffen⁴⁰:

1. Wiederverwendbare Artefakte

Es wird untersucht, ob und auf welcher Ebene die für eine neue ABS *erforderlichen Komponenten* wiederverwendet werden können. Dabei wird insbesondere zwischen der Entwurfsebene (Entwurfsmuster und Architekturen) und der Ebene der Softwarekomponenten unterschieden. Die untersuchten Komponenten umfassen (vgl. Kapitel 3 bis 8)

- ein *AAIM* zur Definition der Applikationslogik und der Interaktionsmöglichkeiten,
- *Bündel von Interaktionsmustern* zur Abdeckung der im AAIM referenzierten Interaktionssituationen,
- die *Bedien- und Anzeigeelemente*, die zur Komposition der verwendeten Interaktionsmuster benötigt werden,
- die *Profile DP, UP, CP und UIP*, die als Schnittstellen der ABS-Parametrierung dienen,
- *gerätespezifische Muster*, die die Merkmale der technischen Ausstattung über das DP für die Anpassungsmechanismen der MyUI-Infrastruktur zugänglich machen,
- *Individualisierungsmuster* zur Anpassung übergreifender ABS-Eigenschaften (UIP) an Nutzer, Geräte und Umgebungen,
- die Adaptationsmechanismen der *Adaptationsengine*,

⁴⁰ HöB (2005) bezieht sich auf drei Ebenen der Wiederverwendung: (1) Wiederverwendetes Artefakt, (2) Domänenbezug und (3) Organisationsbezug. Da im Rahmen dieser Arbeit unerheblich ist, ob die Wiederverwendung organsiationsübergreifend oder innerhalb einer Organisation erfolgt, werden nur die ersten beiden Dimensionen der Wiederverwendung betrachtet.

- *Adaptationsausführungsmuster*, die Laufzeitanpassungen über animierte Transitionen verdeutlichen und
- *Adaptationsdialogmuster*, die dem Nutzer die Kontrolle von Laufzeitanpassungen ermöglichen.

Die Wiederverwendbarkeit dieser Komponenten wird unter der Voraussetzung untersucht, dass zukünftige Applikationen auf denselben ABS-Technologien basieren wie die Referenzimplementierung (s. Abschnitt 9.2).

2. Domänenbezug und Kontextbezug

Für die wiederverwendbaren Artefakte wird untersucht, ob es sich um eine *domänenübergreifende Wiederverwendung* handelt oder ob die Wiederverwendung auf eine spezifische Anwendungsdomäne beschränkt ist. Zusätzlich zu den bei Höß (2005) beschriebenen Dimensionen wird die Kontextspezifität der Wiederverwendung analysiert. Bei kontextübergreifender Wiederverwendung können die Komponenten wiederverwendet werden, unabhängig davon, welche Eigenschaften der technischen Ausstattung, des Nutzers und der Nutzungsumgebung bei der Anpassung der ABS berücksichtigt werden sollen.

3. Wiederverwendungsszenario

Anhand zweier Beispiele neu zu entwickelnder Applikationen aus dem Umfeld der Referenzimplementierung werden die Ergebnisse der ersten beiden Schritte validiert. Es wird insbesondere analysiert, in wie weit bestehende Entwurfsmuster wiederverwendet werden könnten und welche neuen Muster entwickelt werden müssten, um ABS für diese beiden Applikationen zu erstellen:

- a. Elektronischer Programmführer (EPG: Electronic Program Guide) mit den folgenden Funktionen:
 - Übersicht des Fernsehprogramms aller Sender auf einer Zeitleiste
 - Filter- und Suchmechanismen in der Programmübersicht
 - Anzeige von Informationen zur aktuellen Sendung
 - Funktionen zum Aufnehmen von Sendungen, Speichern, Durchsuchen und Abspielen von Aufnahmen, sowie weitere Standardfunktionen zur Wiedergabe von Video-Aufnahmen wie Pause, Zurückspulen, etc.
- b. Online Shop, dessen Funktionsumfang und Komplexität sich am Beispiel unter <http://www.lalaberlin-onlineshop.de>⁴¹ orientiert:
 - Startseite mit vier Einstiegen
 - Drei Hauptkategorien („Kollektion“, „Accessoires“ und „Sale“), jeweils mit einer Startseite und Unterkategorien mit Produktübersichten
 - Detaillierte Produktseiten mit jeweils drei Fotos und einem Bereich für eine vergrößerte Ansicht sowie einem Bereich für schriftliche Produktdetails und

⁴¹ Zugriff am 14. August 2013

Funktionen wie kaufen ("add to bag"), Größentabelle und Zugang zum Kontaktformular („Frage zum Produkt“)

- Eine Übersicht der zum Kauf ausgewählten Produkte ("Warenkorb")
- Kunden-Log-In mit Bereich für Neuansmeldung und ein Check-Out-Prozess mit Formularen für Name, Adresse, Zahlart, etc.
- Wechsel zwischen der deutschen und der englischen Sprachversion
- Redaktionelle und zusätzliche Informationen (z.B. Kontakt, Versandkosten, AGB, etc.)

9.2.8.3 Ergebnisse

Die Ergebnisse der ersten beiden Evaluierungsstufen sind in Tabelle 21 zusammengefasst. Das AAIM ist ein wesentliches Artefakt der Entwicklung einer neuen ABS. Für jede neue Applikation muss ein eigenes AAIM manuell erstellt werden. Lediglich die Funktionen und Interaktionsoptionen, die in allen Zuständen aller Applikationen in derselben Weise zur Verfügung gestellt werden sollen, können über einen zusammengesetzten Zustand als wiederverwendbares Artefakt modelliert werden. Alle anderen Komponenten der ABS-Infrastruktur sind sowohl auf der Entwurfsebene als auch als Softwarekomponenten voll wiederverwendbar. Die Wiederverwendung als Softwarekomponente gilt nur, solange stets dieselben ABS-Technologien verwendet werden. In diesem Fall können die Komponenten direkt zur Laufzeit wiederverwendet werden (vgl. Höß, 2005).

Einschränkungen der Wiederverwendbarkeit betreffen einerseits die Kontextprofile DP und UP und die damit verbundenen gerätespezifischen Muster und Individualisierungsmuster. Wenn neue Kontextbedingungen abgedeckt werden sollen, werden sowohl in den Profilen als auch bei den Entwurfsmustern Erweiterungen notwendig. Diese können auch Modifikationen an den bereits bestehenden Mustern erfordern und Erweiterungen am UIP bedingen. Dies kann schließlich auch dazu führen, dass neue Interaktionsmuster in bestehenden Bündeln erstellt werden müssen (vgl. Abschnitt 4.4.2).

Andererseits bestehen Einschränkungen bei der domänenübergreifenden Wiederverwendung der Interaktionsmuster. Diese beziehen sich insbesondere auf die Frage, in wie weit ein einziges generisches Bündel von Interaktionsmustern für alle Instanzen einer bestimmten Interaktionssituation genutzt werden sollte, ohne gewisse Anpassungen an die spezifischen Applikationsinhalte vorzunehmen. So ist beispielsweise für alle Instanzen der Interaktionssituation `SelectItemWithAttributes` vorgesehen, eine Variante derselben Listendarstellung aus dem Bündel „List with Attributes“ zu verwenden – unabhängig davon ob eine Liste von E-Mails, eine Liste von Fotos oder eine Liste von Arbeitsaufgaben dargestellt werden soll. Zwar ist dies grundsätzlich möglich, doch haben sich teilweise für die Darstellung unterschiedlicher Inhalte gewisse Quasistandards etabliert. Zur Optimierung der Gebrauchstauglichkeit und Erwartungskonformität könnte es daher sinnvoll sein, die Darstellung an diese inhaltspezifischen Formate anzupassen. Sobald Interaktionsmusterbündel um derartige Varianten ergänzt werden, müssten auch die interaktionsspezifischen Adaptationsausführungsmuster neu betrachtet werden. Dies gilt auch bei Erweiterungen im Zusammenhang mit neuen Kontextbedingungen.

Die Adaptationsdialogmuster sind in ihrer jetzigen Implementierung vollständig wiederverwendbar. Einschränkungen könnten sich nur bei den Auswahlregeln des geeigneten Musters ergeben, wenn neue Anwendungsdomänen oder Kontextbedingungen völlig andere Anforderungen mit sich bringen würden. Dies ist jedoch im Moment nicht abzusehen oder zu erwarten.

| Komponente | Artefakte und Dimensionen der Wiederverwendbarkeit | | | |
|-----------------------------------|--|-------------------------|--------------------------|--------------------------|
| | Teilentwürfe/ Architektur | Software- komponente | Domänen- übergreifend | Kontext- übergreifend |
| AAIM | ○ | ○ | | |
| Interaktionsmuster | ● | ● | ◐ | ◐ |
| Bedien- und Anzeigeelemente | ● | ● | ● | ● |
| DP | ● | ● | ● | ○ |
| UP | ● | ● | ● | ○ |
| CP | ● | ● | ● | ● |
| UIP | ● | ● | ● | ◐ |
| Gerätespezifische Muster | ● | ● | ● | ○ |
| Individualisierungs- muster | ● | ● | ● | ○ |
| Adaptationsengine | ● | ● | ● | ● |
| Adaptations- ausführungsmuster | ● | ● | ◐ | ◐ |
| Adaptations- dialogmuster | ● | ● | ● | ● |

Legende:

● Kriterium voll erfüllt

◐ Kriterium teilweise erfüllt

○ Kriterium nicht erfüllt

Tabelle 21 Bewertung der Wiederverwendbarkeit der Komponenten der ABS-Infrastruktur

Es wird deutlich, dass sich der Aufwand für die Entwicklung neuer adaptiver Applikationen insbesondere auf die Erstellung des AAIM und die Ergänzung von benötigten Interaktionsmustern, die sich noch nicht im Repositorium befinden, bezieht. Folgeaufwände entstehen, wenn neue Interaktionsmuster erstellt werden müssen, die selbst wieder zusätzliche Bedien- und Anzeigeelemente oder interaktionsspezifische Adaptationsausführungsmuster erfordern. Daher lassen sich Reduktionen des Entwicklungsaufwands neuer Applikationen bzw. ABS insbesondere durch die Wiederverwendung von bestehenden Interaktionsmustern erzielen. Aus diesem Grund beschränkt sich die folgende Darstellung der Ergebnisse des Wiederverwendungsszenarios auf die dafür benötigten Interaktionssituationen und Interaktionsmuster (s. Tabelle 22).

| Funktion | Interaktionssituation | Interaktionsmuster | Wiederverwendungspotenziale |
|---|---------------------------------|---------------------------------------|---|
| Elektronischer Programmführer | | | |
| Programmübersicht | ViewGroupsOfItemsIn TimeLine | Matrix – time line – unconstrained | Keine; ein neues Bündel ist erforderlich. |
| Suche in Programmübersicht | EditForm | Form | Hoch; die bestehenden Formularemuster können verwendet werden. Eigene Muster für Suchformulare wären jedoch sinnvoll. |
| Anzeige der Suchergebnisse | SelectItemWith Attributes | List with Attributes | Hoch; das bestehende Bündel kann verwendet werden. Spezifische Muster für Suchergebnisse wären jedoch sinnvoll. |
| Filter in Programmübersicht | FilterGroupOfItems | Filter Tabs | Keine; ein neues Bündel ist erforderlich. |
| Detaillierte Anzeige für ausgewählte Sendung | ViewItemWith Attributes | Item with Attributes | Hoch; das bestehende Bündel kann verwendet werden. Spezifische Muster für die Detailsicht einer Fernsehsendung wären sinnvoll. |
| Informationen zur aktuellen Sendung | ViewAdditional Information | Info Bar | Keine; ein neues Bündel ist erforderlich. |
| Medien-Funktionen (Aufnehmen, Ab- spielen, Pause, etc.) | ControlTime- DependentMedia | Media Control | Keine; ein neues Bündel ist erforderlich. |
| Online Shop | | | |
| Startseite mit vier Einstiegen | SelectEntry | Entry Menu – 1st Entry Dominant | Mittel; unter Umständen kann das bestehende Bündel Main Menu verwendet werden. |
| Hauptkategorien mit Unterkategorien | SelectGoTo | Navigation Menu | Hoch; eigene Muster für eine typische Web- Navigation in der Form eines umgedrehten L („Inverted L“) wären sinnvoll. |
| Produktübersicht in Unterkategorien | SelectItemWith Attributes | List with Attributes | Hoch; eigene Muster, die eine grafische Darstellung der Auswahloptionen besser unterstützen als in einer Liste, wären sinnvoll. |
| Detaillierte Produktseite | ViewItemWith Attributes | Item with Attributes | Hoch; mit neuen spezifischen Mustern könnten die beschriebenen Darstellungsfunktionen besser abgebildet werden. |
| Warenkorb | SelectItemWith Attributes | List with Attributes | Hoch; das bestehende Bündel kann verwendet werden. |
| Kundenanmeldung | EditForm | Form | Hoch; die bestehenden Formularemuster können verwendet werden. Eigene Muster für Formulare zur Nutzeranmeldung wären sinnvoll. |
| Sprachwahl | ChangeMode | Mode Selector | Keine; ein neues Bündel ist erforderlich. |

Tabelle 22 Interaktionssituationen und Interaktionsmuster, die zur Realisierung des Wiederverwendungsszenarios benötigt würden. Interaktionsmuster, die bereits bestehen und wiederverwendet werden können, sind in Fettschrift hervorgehoben.

9.2.8.4 Interpretation und Diskussion

Aufgrund der dargestellten Ergebnisse ist davon auszugehen, dass durch das Prinzip der Wiederverwendung der Aufwand zur Entwicklung neuer Applikationen und ABS sukzessive sinken wird. Wesentliche Komponenten der ABS-Infrastruktur sind so generisch aufgebaut, dass eine domänenübergreifende Wiederverwendung möglich ist.

Durch die Frage, wie stark das Erscheinungsbild von Bedien- und Anzeigekomponenten von den dargestellten Inhalten abhängen sollte, ergibt sich für die Interaktionsmuster nur eine bedingte Domänenunabhängigkeit. Insbesondere im Wiederverwendungsszenario wird deutlich, welche Probleme sich bei der domänenübergreifenden Wiederverwendung generischer Interaktionsmuster ergeben können. Je nach Anwendungsdomäne können Bildschirmszenen entstehen, die den dargestellten Inhalten kaum gerecht werden und Nutzererwartungen und Gewohnheiten verletzen. Durch die Verwendung der immer gleichen Bedien- und Darstellungskomponenten entstehen gleichförmige Ansichten über verschiedene Applikationen hinweg. Ohne visuelle Hinweise auf die dargestellten Inhalte wird die Orientierung des Nutzers deutlich erschwert.

Für eine Weiterentwicklung des MyUI-Rahmenmodells ist daher vorgesehen, in der Menge der Interaktionsmuster ein Stereotypen-Konzept einzuführen. So können neben den generischen Musterbündeln weitere Interaktionsmuster oder Interaktionsmusterbündel hinzugefügt werden, die wiederkehrende Inhaltstypen oder spezielle Anwendungsdomänen in besonderer Weise - d.h. stereotypisch - unterstützen. So könnten beispielsweise die Interaktionsmusterbündel List with Attributes und Item with Attributes durch Stereotype für die Übersicht und Detailansicht von Produkten ergänzt werden. Die Entscheidung, ob zur ABS-Generierung die generischen Muster oder die inhaltsspezifischen Stereotypen verwendet werden, müsste in die Hände der ABS-Entwickler gelegt werden. Dazu könnte die Parametrierung von Interaktionssituationen zusätzlich zu den Eingangs- und Ausgangsparametern um eine Stereotyp-Variable ergänzt werden. Dies würde dem Entwickler die Möglichkeit bieten, im AAIM zu definieren, welcher Stereotyp genutzt werden soll. Als Default der Stereotyp-Variable würden die generischen Interaktionsmuster verwendet werden. Durch die Erweiterbarkeit des Entwurfsmusterrepositoriums könnten nach und nach weitere Stereotypen hinzugefügt werden, um die Entwicklung neuer Applikationen in unterschiedlichen Anwendungsdomänen passgenau zu unterstützen.

Schließlich wird festgestellt, dass sich die dargestellte Evaluierung auf Neuentwicklungen im Rahmen derselben ABS-Technologien wie in der Referenzimplementierung bezieht. Eine technologieübergreifende Wiederverwendung besteht zumindest auf der Ebene der Konzepte, Entwurfsmuster und Profile des MyUI-Rahmenmodells.

9.3 Empirische Evaluierung der Transparenz und Kontrollierbarkeit mit potenziellen Endnutzern

In einer empirischen Studie mit potenziellen Endnutzern werden die Transparenz und die Kontrollierbarkeit der automatischen Adaptationen untersucht. Insbesondere steht dabei eine Bewertung der Effektivität und Akzeptanz der Adaptationsdialogmuster im Mittelpunkt. Das ABS-Rahmenmodell bietet drei verschiedene Adaptationsdialogmuster und kann bei der Auswahl des am besten geeigneten Musters verschiedene Kriterien einbeziehen (vgl. Abschnitt 7.5). Neben der Validierung der generellen Effektivität der Adaptationsmuster zielt die Studie daher auch darauf ab, festzustellen, welches Muster unter welchen Bedingungen zu bevorzugen ist. Auf der Grundlage der in Abschnitt 2.3.2 dargestellten Diskussion über die Kosten und Nutzen automatischer Anpassungen und deren Optimierung durch geeignete Kontrollmechanismen wird hierfür eine Kosten-Nutzen-Betrachtung von automatischen Anpassungen aus Nutzerperspektive eingeführt:

- Adaptationen haben einen *hohen Nutzen*, wenn sie helfen, individuelle Nutzungsbarrieren zu überwinden, z.B. Vergrößerung der Schriftzeichen, wenn Sehbeeinträchtigungen vorliegen.
- Adaptationen haben einen *geringen Nutzen*, wenn sie keinen positiven Effekt auf die individuelle Gebrauchstauglichkeit in der aktuellen Situation bieten. Dieser Fall liegt zum Beispiel vor, wenn vor der automatischen Adaptation keine Nutzungsbarriere besteht und aufgrund eines Fehlers in der Kontexterfassung dennoch eine Anpassung durchgeführt wird.
- Adaptationen haben *hohe Kosten*, wenn sie einen zusätzlichen Aufwand für den Nutzer mit sich bringen. Dabei kann es sich um mentalen Aufwand handeln, den das Zurechtfinden mit der veränderten Benutzungsschnittstelle erfordert, oder um zusätzlichen Interaktionsaufwand, der zum Beispiel entsteht, wenn der Nutzer eine unerwünschte Adaptation rückgängig machen muss oder wenn er aufgrund einer Vergrößerung der Schriftzeichen blättern oder scrollen muss, um die gewünschte Information zu erhalten.
- Adaptationen haben *geringe Kosten*, wenn sie keinen zusätzlichen Aufwand mit sich bringen.

Die Beziehung zwischen Kosten und Nutzen und den Adaptationsdialogmustern ist möglicherweise beidseitig. Einerseits könnten sich die verschiedenen Muster unter verschiedenen Kosten-Nutzen-Bedingungen als unterschiedlich effektiv erweisen, so dass je nach Kosten und Nutzen der Anpassung andere Muster bevorzugt werden. Andererseits ist anzunehmen, dass die gebotenen Kontrollmechanismen einen wesentlichen Beitrag zur Optimierung des subjektiven Kosten-Nutzen-Verhältnisses der automatischen Adaptationen leisten können. Insbesondere sollte der Nutzer die subjektiven Kosten von unerwünschten Anpassungen reduzieren können, wenn er die Möglichkeit erhält, Anpassungen abzulehnen oder rückgängig zu machen. Daher sollten sich die verschiedenen Adaptationsdialogmuster auch in diesem Bereich unterscheiden.

Eine ausführlichere Darstellung der durchgeführten Studie und ihrer Ergebnisse findet sich bei Peissner & Edlin-White, 2013.

9.3.1 Fragestellung

Die empirische Studie adressiert die Fragestellungen (5) und (6) des Evaluierungsplans (Abschnitt 9.1):

- *Transparenz*: Kann das System sicherstellen, dass die Benutzer die automatischen Anpassungen erkennen und verstehen und dass Verwirrung und Desorientierung vermieden werden? (Frage 5 der Tabelle 15; die untersuchte Anforderung ist in Abschnitt 2.3.1 beschrieben.)
- *Kontrollierbarkeit*: Bietet das System Mechanismen, die den Benutzern eine bedarfsgerechte Kontrolle des Adaptionsverhaltens ermöglichen? (Frage 6 der Tabelle 15; die untersuchte Anforderung ist in Abschnitt 2.3.2 beschrieben.)

Darüber hinaus wird untersucht, ob die gebotenen Kontrollmechanismen den Nutzer dabei unterstützen, das subjektive Kosten-Nutzenverhältnis der automatischen Anpassungen zu optimieren und ob sich hierin Unterschiede zwischen den verschiedenen Dialogmustern ergeben.

Schließlich soll die Studie darüber Aufschluss geben, ob in jeder Situation dasselbe Adaptionsdialogmuster verwendet werden kann oder in unterschiedlichen Kosten-Nutzen-Bedingungen unterschiedliche Muster zum Einsatz kommen sollten.

9.3.2 Methode

9.3.2.1 Versuchsbedingungen

Zur Untersuchung der Fragestellungen wird eine zusätzliche Vergleichsbedingung eingeführt, in der die automatische Adaptation ohne Adaptionsdialogmuster erfolgt. In dieser Bedingung wird die Anpassung automatisch ausgelöst und ausgeführt. Außer den animierten Übergängen der Adaptionsausführungsmuster gibt es keine zusätzlichen Hinweise. Der Nutzer hat keine Möglichkeit, die Anpassung zu beeinflussen, abubrechen oder rückgängig zu machen. Dadurch ergeben sich vier Versuchsbedingungen:

1. Die Vergleichsbedingung wird in der Folge „Baseline“ genannt.
2. Das Muster „Automatische Adaptation mit impliziter Nutzerzustimmung“ wird in der Folge mit „AI“ abgekürzt. Im Vergleich zur Baseline wird aufgrund der Rückgängig-Funktion und des animierten Icons (s. Abschnitt 7.5.1) eine etwas höhere Transparenz und Kontrollierbarkeit erwartet. Da bei Nutzerzustimmung keine zusätzlichen Interaktionsschritte anfallen, sollten keine Einschränkungen der Interaktionseffizienz auftreten.
3. Das Muster „Explizite Nutzerbestätigung vor der Adaptation“ (EV) sollte die Transparenz und Kontrollierbarkeit deutlich verstärken. Das Dialogfenster stellt einen deutlichen und auffälligen Hinweis auf die geplante Anpassung dar. Die Adaptation wird erst durchgeführt, wenn der Nutzer ausdrücklich zugestimmt hat. Jedoch unterbricht es den eigentlichen Interaktionsfluss und erzwingt einen zusätzlichen Interaktionsschritt. Das könnte auf einige Nutzer störend wirken und die Effizienz beeinträchtigen – insbesondere bei Adaptationen, die der Nutzer als nützlich empfindet.

4. Beim Muster „Explizite Nutzerbestätigung nach der Adaptation“ (EN) ist aufgrund des verzögerten Bestätigungsdialogs mit etwas geringeren Zuwächsen bei Transparenz und Kontrollierbarkeit zu rechnen als bei EV. Auch hier gibt es einen Extraschritt.

9.3.2.2 Hypothesen

Die Probanden der Studie erleben im Experiment verschiedene automatische Anpassungen mit jedem der drei Adaptationsdialogmuster und mit der Vergleichsbedingung in verschiedenen experimentell erzeugten Kosten-Nutzen-Situationen und bewerten die subjektiv wahrgenommene Transparenz und Kontrollierbarkeit sowie den Nutzungskomfort und die Akzeptanz. Darüber hinaus bewerten sie die subjektiven Kosten und Nutzen der Anpassungen. Drei Forschungshypothesen werden untersucht:

1. Die drei Adaptationsdialogmuster AI, EV und EN erhalten bezüglich Transparenz, Kontrollierbarkeit, Nutzungskomfort und Akzeptanz bessere Nutzerbewertungen als die Baseline und werden öfter bevorzugt.
2. Adaptationsmuster, die den Nutzern starke Kontrollmechanismen bieten, unterstützen sie besser, das subjektive Kosten-Nutzen-Verhältnis von Adaptationen zu optimieren. Insbesondere wird angenommen, dass die subjektiven Kosten von unerwünschten Anpassungen minimiert werden können.
3. Je nach Kosten-Nutzen-Situation wird die Akzeptanz der Adaptationsmuster unterschiedlich bewertet und unterschiedliche Muster werden bevorzugt. Es wird erwartet, dass bei Adaptationen mit hohen Kosten ein stärkerer Wunsch nach Kontrolle besteht, während bei Anpassungen mit hohem Nutzen effizientere Muster bevorzugt werden.

9.3.2.3 Probanden und Testaufbau

Insgesamt nehmen zwölf ältere Erwachsene an dem Experiment teil, sieben Frauen und fünf Männer. Das Alter der Teilnehmer liegt zwischen 49 und 73 Jahren bei einem Durchschnitt von 63 Jahren. Die Teilnahme ist freiwillig und wird mit EURO 50,-- für eine Testsitzung von etwa 90 Minuten vergütet. Die Hälfte der Probanden bezeichnet ihre IT-Kenntnisse als mittel, drei als gering und weitere drei als ziemlich gut. Bei der Auswahl der Probanden wird gezielt darauf verzichtet, Nutzer mit Einschränkungen zu beteiligen. Keiner der Teilnehmer erlebt in seinem Alltag nennenswerte Nutzungsbarrieren bei der Interaktion mit Technik. Der Grund für diese Stichprobenauswahl liegt in dem speziellen Experimentaldesign der Studie. Eine nähere Erläuterung findet sich in Abschnitt 9.3.2.4.

Das Experiment wird im Usability Labor des Fraunhofer IAO durchgeführt. Mit dem Probanden befindet sich noch ein Testmoderator im Raum, der den Probanden durch das Experiment führt, die Testaufgaben instruiert und die Interviews durchführt. Drei Windows-Computer werden eingesetzt. Auf dem ersten Computer läuft der interaktive Prototyp, der ein adaptives iTV-System mit einem Hauptmenü und einer E-Mail-Applikation zum Empfangen, Lesen, Erstellen und

Verschicken von E-Mails simuliert⁴². Dieser Computer ist mit einem 32-Zoll-Fernsehbildschirm des Typs Philips 32PFL7605H/12 verbunden und wird über eine typische Infrarot-TV-Fernbedienung gesteuert. Die anderen beiden Computer werden genutzt, um durch gezielte Veränderungen des UP im Stil eines Wizard-of-Oz-Experiments „automatische“ Adaptationen auszulösen. Der Proband sitzt in einem Abstand von etwa 2 Metern vom Fernsehbildschirm auf einem Sofa. Über die Fernbedienung interagiert er mit dem Testprototypen, um die Aufgaben des Experiments zu bearbeiten.

9.3.2.4 Aufgaben, Adaptationen und Kosten-Nutzen-Bedingungen

Drei Aufgaben werden bearbeitet. Während jeder Aufgabe wird eine systeminitiierte Laufzeit-anpassung durchgeführt:

1. *Aufgabe*: Starte die E-Mail-Applikation aus dem iTV-Hauptmenü
Adaptation: Zu den Beschriftungen der Menüoptionen werden Referenzen zu den Nummerntasten der Fernbedienung eingeblendet. So kann der Nutzer die angezeigten Optionen direkt mit einem Tastendruck erreichen, anstatt über die Cursor Tasten navigieren zu müssen. Es wird angenommen, dass diese Adaptation für Nutzer mit motorischen Einschränkungen nützlich ist, denen es schwer fällt, die kleinen Cursor-Tasten der Fernbedienung wiederholt und genau zu treffen.
2. *Aufgabe*: Wähle eine E-Mail von einem bestimmten Absender aus dem Posteingang aus.
Adaptation: Die Absendernamen werden durch Portraitfotos der Absender ersetzt. Diese Anpassung soll kognitiv eingeschränkten Nutzern helfen, die Probleme beim Lesen und Verstehen von geschriebener Sprache haben.
3. *Aufgabe*: Lies die ausgewählte E-Mail.
Adaptation: Die Schriftgröße wird erhöht. Diese Adaptation soll Nutzern mit Sehbeeinträchtigungen helfen.

Mit jedem dieser Paare von Aufgaben und Anpassungen werden die vier möglichen Kombinationen, die sich aus den beschriebenen hohen und geringen Kosten- und Nutzenniveaus ergeben, realisiert. Kosten und Nutzen werden dabei systematisch im Experiment variiert. Für die Variation der Kosten wird eine einfache Methode gewählt. Für die Bedingung „hohe Kosten“ werden die Gestaltung der Benutzungsschnittstelle und die konkrete Aufgabenstellung so angelegt, dass der Nutzer nach der Adaptation scrollen oder blättern muss, um die gestellte Aufgabe zu erledigen. Für die Bedingung „geringe Kosten“ wird die Aufgabe so gewählt, dass durch die Adaptation kein zusätzlicher Interaktionsaufwand entsteht.

Die experimentelle Variation des Nutzens erfordert ein aufwändigeres Verfahren. Hierfür ist es wichtig, dass nur Personen ohne spezielle Bedürfnisse am Experiment teilnehmen. Dadurch sind für alle Probanden die verwendeten Adaptationen ohne Nutzen. Die Teilnehmer erleben zunächst keine Nutzungsbarrieren. Durch künstliche Barrieren, die durch die verwendeten Adaptionen überwunden werden können, wird für jede Aufgabe eine Versuchsbedingung mit hohem subjektiven Nutzen geschaffen (s. Abbildung 29):

⁴² Dabei handelt es sich um eine speziell für das Experiment angepasste Variante der Referenzimplementierung (s. Abschnitt 09.2).

1. In der Versuchsbedingung „hoher Nutzen“ der ersten Aufgabe tragen die Nutzer Lederhandschuhe der Größe XL, die eine präzise Bedienung der Tasten der Fernbedienung erschweren sollen.
2. In der zweiten Aufgabe wird mit „Wingdings“ eine für die Nutzer unleserliche Schriftart verwendet.
3. In der dritten Aufgabe tragen die Probanden Brillen mit +1,5 oder +2,5 Dioptrien. In einem kurzen Vorversuch wird für jeden Probanden die besser passende Brillenstärke ausgewählt, so dass nur die angepasste, erhöhte Schriftgröße lesbar ist.



Abbildung 29 Hilfsmittel zur systematischen Variation des Nutzens von automatischen Adaptationen in der Nutzerstudie

An dieser Stelle ist es wichtig, darauf hinzuweisen, dass die auf diese Weise künstlich erzeugten Nutzungsbarrieren nicht darauf abzielen, irgendwelche motorischen, kognitiven oder perzeptuellen Einschränkungen nachzustellen. Es wäre naiv, zu denken, dass das Tragen von großen Lederhandschuhen bei der Bedienung einer TV-Fernbedienung ein ähnliches Gefühl hervorrufen könnte wie eine eingeschränkte Feinmotorik. Außerdem wäre es nicht möglich, mit diesen drei Versuchsbedingungen alle denkbaren Nutzungsbarrieren repräsentativ abzubilden. Vielmehr soll mit diesem Ansatz eine gezielte Variation des Nutzens von automatischen Laufzeitanpassungen erreicht werden. Hierfür wird das generelle Prinzip von Anpassungen mit einem hohen Nutzen angewendet: Der Nutzer erlebt eine Barriere und ist nicht in der Lage, sein Interaktionsziel zu erreichen. Dann erfolgt eine systeminitiierte automatische Adaptation und hilft, das Problem zu überwinden. Dieser Mechanismus konnte in dem durchgeführten Experiment erfolgreich nachgestellt werden. Die Nutzer der Studie wurden nach jeder Aufgabe gebeten, den subjektiven Nutzen der erlebten Anpassung zu bewerten. Die Nutzerbewertungen stimmten substantziell mit der experimentell beabsichtigten Nutzenbedingung überein (Cohen's Kappa $\kappa=0,729$, $n=96$). Für die experimentelle Variation der Kosten konnte eine mittelstarke

Übereinstimmung zwischen Experimentalbedingung und Nutzerbewertung ermittelt werden (Cohen's Kappa $\kappa=0,375$, $n=96$).

9.3.2.5 Experimentaldesign

Die Studie folgt einem 4 x 4 Within-Subjects-Design mit vier Kosten-Nutzen-Bedingungen und vier Adaptationsdialogmustern inklusive der Baseline-Bedingung. Jede der drei Aufgaben wird zweimal in allen 4 x 4 Testbedingungen bearbeitet. Die daraus resultierenden 96 Testfälle werden zufällig auf die zwölf Teilnehmer verteilt, so dass jeder Teilnehmer insgesamt acht Aufgaben erhält. Dabei durchläuft jeder Proband zwei der drei Aufgaben jeweils viermal - und zwar jede Aufgabe einmal in jeder Kosten-Nutzen-Bedingung und einmal mit jeder der vier Adaptationsmuster⁴³. Die ersten vier Durchläufe der ersten Aufgabe bearbeiten die Probanden naiv, d.h. ohne auf die verschiedenen Dialogmuster aufmerksam gemacht worden zu sein. Bevor sie die vier Testfälle mit der zweiten Aufgabe bearbeiten, erhalten die Probanden vom Testmoderator eine standardisierte Erklärung der verschiedenen Dialogmuster.

Als unabhängige Variablen der Studie werden subjektive Bewertungen der Adaptationsdialogmuster nach jeder der acht Testläufe erhoben. Hierfür werden vierstufige Likert-Skalen verwendet:

- *Transparenz* (zwei Items)
„Die Gestaltung verdeutlicht gut, dass sich eine Anpassung vollzieht.“
„Die Gestaltung vermittelt gut, was der Benutzer unternehmen kann, um die Änderung zu stoppen oder rückgängig zu machen.“
- *Kontrollierbarkeit* (zwei Items)
„Ich habe das Gefühl, dass ich die volle Kontrolle über das System und dessen Erscheinung/Anpassung habe.“
„Ich habe das Gefühl, dass es einfach ist, das System zu kontrollieren und dessen Erscheinung/Anpassung anzupassen.“
- *Nutzungskomfort* (zwei Items)
„Die Gestaltung unterstützt mich bei der komfortablen Systemnutzung.“
„Die Gestaltung hilft dabei, unnötige Nutzereingaben oder Interaktionsschritte zu vermeiden.“
- *Allgemeine Akzeptanz* (ein Item)
„Alles in allem gefällt mir die Gestaltung des eben gezeigten Anpassungsvorgangs.“

Darüber hinaus bewerten die Probanden auch die *subjektiven Kosten und Vorteile* der automatischen Adaptationen auf vierstelligen Likert-Skalen:

„Die gerade erlebte Anpassung bringt in meiner Situation Vorteile mit sich.“

⁴³ Die Beschränkung auf zwei der drei Aufgabentypen hat praktische Gründe: Eine Testsitzung soll nicht länger als 90 Minuten dauern, um die Aufmerksamkeitsspanne der Probanden nicht zu stark zu beanspruchen und um Verzerrungen, die sich aus Ermüdungseffekten ergeben können, auszuschließen. In einer Vorstudie zeigte sich, dass mehr als acht Testläufe innerhalb einer 90-Minuten-Sitzung nicht möglich waren.

„Die gerade erlebte Anpassung erschwert für diese Aufgabe die Interaktion mit dem System.“
„Die Vorteile der Anpassung überwiegen die Nachteile.“

Schließlich geben die Teilnehmer in den vier Testläufen, in denen sie bereits über die verschiedenen Muster aufgeklärt worden sind, an, welches der Muster sie im gerade erlebten Fall *bevorzugen*. In einem abschließenden Interview nach der Aufgabenbearbeitung wird mit dem Probanden diskutiert, welches Muster er *generell präferieren* würde.

9.3.2.6 Ablauf

Der Ablauf einer Testsitzung mit einem Probanden folgt dem folgenden Schema:

1. Begrüßung und Einführung in die Testsituation. Die Nutzer füllen einen Fragebogen aus, in dem Computerkenntnisse, Technikinteresse und Technikangst abgefragt werden.
2. Wiederholte Bearbeitung der ersten Aufgabe in vier Durchläufen. In jedem Durchlauf erfolgt eine systeminitiierte Adaptation mit einem anderen intendierten Kosten-Nutzen-Verhältnis und einem anderen Adaptationsdialogmuster bzw. der Baseline-Bedingung. In den Bedingungen mit hohem Nutzen wird die Adaptation 10 Sekunden nachdem der Nutzer aufgrund der künstlichen Barriere nicht mehr weiter kommt von einem Testassistenten manuell ausgelöst. Der Proband wird in dem Glauben gehalten, dass es sich um eine automatische, systeminitiierte Adaptation handelt (Wizard-of-Oz-Verfahren). In den Bedingungen mit geringem Nutzen wird die Adaptation 20 Sekunden nach Beginn der Aufgabenbearbeitung ausgelöst. Nach jedem Durchlauf erfolgt eine Bewertung anhand der oben beschriebenen Likert-Skalen.
3. Diskussion über die erlebten Adaptationen und Adaptationsdialoge. Die Probanden erhalten eine standardisierte Erläuterung der Adaptationsdialogmuster.
4. Wiederholte Bearbeitung der zweiten Aufgabe in vier Durchläufen genau wie im zweiten Schritt (oben). Zusätzlich gibt der Proband an, welches der vier Dialogmuster er im gerade erlebten Fall bevorzugt hätte.
5. Abschließendes Interview zur Diskussion, ob es eine generelle Präferenz eines der Dialogmuster gibt.

9.3.3 Ergebnisse

9.3.3.1 Allgemeine Validierung der Adaptationsdialogmuster

Die Transparenz, Kontrollierbarkeit und Akzeptanz der drei Adaptationsdialogmuster AI, EV und EN bewerten die Probanden signifikant besser als die der Baseline-Bedingung (s. Abbildung 30). Für die statistische Datenanalyse wird dazu jeweils die absolute Anzahl der positiven Bewertungen eines Kriteriums herangezogen. Ein Kriterium mit zwei Fragebogen-Items gilt dann als positiv bewertet, wenn beide Fragen mit einer 3 oder einer 4 auf der vierstufigen Likert-Skala bewertet worden sind. Ein χ^2 -Test (Bortz et al., 2000) gegen die Nullhypothese gleichverteilter Nutzerbewertungen für alle vier Dialogbedingungen ergibt signifikante Unterschiede für Transparenz ($\chi^2(3, N=96) = 19,80, p < 0,001^{**}$), Kontrollierbarkeit ($\chi^2(3, N=96) = 12,42, p = 0,006^{**}$) und Akzeptanz ($\chi^2(3, N=96) = 8,38, p = 0,039^*$). Wie bereits angenommen, erzielt

das Dialogmuster EV die besten Werte für Transparenz und Kontrollierbarkeit. In allen drei Kategorien schneidet die Baseline-Bedingung schlechter ab als jedes der drei Adaptationsdialogmuster. Die Bewertungen des Nutzungskomforts der vier Bedingungen unterscheiden sich nicht signifikant.

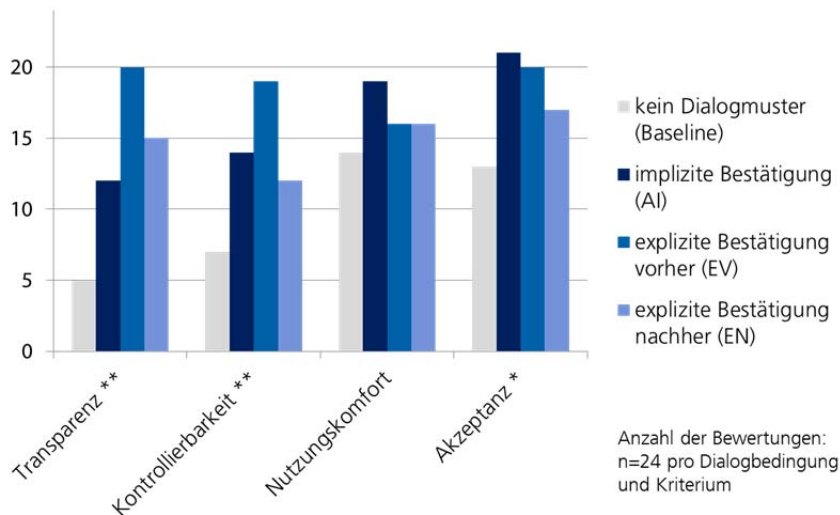


Abbildung 30 Anzahl positiver Nutzerbewertungen für Transparenz, Kontrollierbarkeit, Nutzungskomfort und Akzeptanz der verschiedenen Dialogbedingungen

Bei der Analyse der Nutzerpräferenzen ergibt sich ein ähnliches Bild (s. Tabelle 23). Die Präferenzurteile, die direkt nach der Aufgabebearbeitung erfasst worden sind, unterschieden sich signifikant für die vier Dialogbedingungen ($\chi^2(3, N=48) = 11,00, p=0,012^*$). In nur drei der insgesamt 48 Fälle (6,25%) wird die Baseline-Bedingung bevorzugt. Im Vergleich zu den Adaptationsdialogmustern ist sie damit mit deutlichem Abstand weniger beliebt. Die Muster AI und EV werden in den meisten Fällen bevorzugt. Derselbe Trend zeigt sich bei der allgemeinen Präferenz, die am Ende der Testsitzung abgefragt worden ist. Aufgrund der geringen Stichprobengröße (n=12) kann hierfür keine sinnvolle Statistik errechnet werden.

| | Baseline | AI | EV | EN |
|---------------------------------|----------|----|----|----|
| Präferenz pro Testfall (n=48) * | 3 | 17 | 17 | 11 |
| Allgemeine Präferenz (n=12) | 1 | 5 | 5 | 1 |

Tabelle 23 Nutzerpräferenzen für Adaptationsdialogmuster und Baseline (absolute Häufigkeiten)

9.3.3.2 Kontrolle der subjektiven Kosten von Adaptationen durch die Adaptationsdialogmuster

Betrachtet man alle 96 Testfälle, unterscheiden sich die verschiedenen Dialogbedingungen nicht in ihrer Fähigkeit, den Nutzer bei der Kontrolle der subjektiven Kosten von automatischen Adaptationen zu unterstützen. Das dafür erhobene Maß ist die Häufigkeit, mit der die Nutzer geringe subjektive Kosten ($Kosten_{subj.}$) berichten, obwohl sie gerade in einer Experimentalbedingung mit hohen intendierten Kosten ($Kosten_{int.}$) sind. Dieser Fall tritt für alle vier Dialogbedingungen in der gleichen Häufigkeit auf ($\chi^2(3, N=96) = 0,54, p=0,91$).

Betrachtet man jedoch nur die Testfälle nach der Einführung der Dialogmuster durch den Testmoderator, d.h. die Fälle, in denen sich die Nutzer der verschiedenen Adaptationsdialogmuster bewusst sind, findet man signifikante Unterschiede ($\chi^2(3, N=48) = 8,76, p=0,033^*$). Erwartungsgemäß erweisen sich die beiden Dialogmuster mit expliziten Bestätigungen als besonders effektiv, die Kosten von nachteiligen Anpassungen zu reduzieren (s. Tabelle 24).

| | Baseline | AI | EV | EN |
|--|----------|----|----|----|
| Alle Testfälle (n=96) | | | | |
| Kosten _{subj.} < Kosten _{int.} | 6 | 7 | 8 | 8 |
| Kosten _{subj.} ≥ Kosten _{int.} | 18 | 17 | 16 | 16 |
| Nach Einführung¹ (n=48) * | | | | |
| Kosten _{subj.} < Kosten _{int.} | 1 | 1 | 5 | 6 |
| Kosten _{subj.} ≥ Kosten _{int.} | 11 | 11 | 7 | 6 |

¹ „Nach Einführung“: Die Nutzer kennen die Adaptationsdialogmuster und die Unterschiede der Dialogbedingungen

Tabelle 24 Können Adaptationsdialogmuster den Nutzer unterstützen, die Kosten automatischer Adaptationen zu kontrollieren? Die Tabelle zeigt absolute Häufigkeiten von Fällen, in denen die Probanden geringe subjektive Kosten berichten, obwohl sie gerade in einer Testbedingung mit hohen intendierten Kosten sind (Kosten_{subj.} < Kosten_{int.})

9.3.3.3 Abhängigkeit von Kosten-Nutzen-Bedingungen

Die Analysen der Zusammenhänge zwischen Präferenz und Akzeptanz der Dialogbedingungen und den Kosten-Nutzen-Bedingungen lassen sich wie folgt zusammenfassen:

- *Kein Zusammenhang mit experimentellen Kosten (Kosten_{int.})*
Die experimentellen Kostenbedingungen zeigen keinen Einfluss auf die Nutzerpräferenzen (s. Tabelle 25). Eine Erklärung hierfür könnte sein, dass die Dialogmuster mit starken Kontrollmechanismen bereits in den Bedingungen mit geringen Kosten sehr häufig bevorzugt werden und ein weiterer Trend in die erwartete Richtung aufgrund eines Deckeneffekts kaum mehr möglich ist. Auch mit den Akzeptanzbewertungen besteht kein erkennbarer Zusammenhang.
- *Hoch signifikanter Zusammenhang zwischen subjektiven Kosten (Kosten_{subj.}) und Akzeptanz*
Die Akzeptanzbewertungen zeigen einen hoch signifikanten Zusammenhang mit dem Niveau der subjektiv berichteten Kosten ($\chi^2(7, N=96) = 22,21, p=0,002^{**}$). Insbesondere erhalten die Baseline und EN deutlich schlechtere Akzeptanzwerte in Situationen mit hohen empfundenen Kosten. Das deutet darauf hin, dass Adaptationsmechanismen mit geringen Kontrollmöglichkeiten für den Nutzer insbesondere dann schlecht bewertet werden, wenn durch die Anpassungen hohe subjektive Kosten entstehen.
- *Signifikanter Zusammenhang zwischen experimentellem Nutzen (Nutzen_{int.}) und Präferenz*
Je nach experimenteller Nutzenbedingung werden unterschiedliche Dialogmuster bevorzugt ($\chi^2(3, N=48) = 8,07, p=0,045^*$). Insbesondere wird die implizite Bestätigung (AI) bei großem Nutzen der Anpassung häufiger bevorzugt (s. Tabelle 25). Bezüglich der Akzeptanzbewertungen zeigen sich keine ähnlichen Effekte.

- *Signifikanter Zusammenhang zwischen subjektivem Nutzen ($Nutzen_{subj}$) und Präferenz sowie Akzeptanz*

Der oben beschriebene Effekt auf die Nutzerpräferenz zeigt sich noch deutlicher im Zusammenhang mit dem subjektiven Nutzen ($\chi^2=15,70$, $df=3$, $p=0,001^{**}$). Auch die Akzeptanzurteile verändern sich bei unterschiedlichen Niveaus des subjektiven Nutzens. Insbesondere wird die Baseline-Bedingung besser bewertet, wenn ein hoher Nutzen erlebt wird. Insgesamt spricht dieser Befund dafür, dass Kontrollmechanismen an Bedeutung verlieren, wenn der Nutzer einen großen Vorteil aus einer Anpassung ziehen kann.

| | Baseline | AI | EV | EN |
|--|----------|----|----|----|
| Experimentelle Kosten ($Kosten_{int.}$) | | | | |
| Geringe Kosten (n=24) | 2 | 7 | 8 | 7 |
| Hohe Kosten (n=24) | 1 | 10 | 9 | 4 |
| Experimenteller Nutzen ($Nutzen_{int.}$) | | | | |
| Geringer Nutzen (n=24) | 1 | 5 | 13 | 5 |
| Hoher Nutzen (n=24) | 2 | 12 | 4 | 6 |

Tabelle 25 Absolute Häufigkeiten der Nutzerpräferenzen für die Dialogbedingungen unter verschiedenen experimentellen Kosten- und Nutzenbedingungen

9.3.4 Interpretation und Diskussion

Die Ergebnisse bestätigen, dass das MyUI-System mit den Adaptionsdialogmustern effektive Mechanismen besitzt, um eine hohe Transparenz und Kontrollierbarkeit der automatischen Anpassungen sicher zu stellen. Insbesondere mit dem Muster EV werden die getesteten Adaptationen von den Probanden in etwa 80% der Fälle als transparent und gut kontrollierbar empfunden⁴⁴. Diese Werte sind vor allem vor dem Hintergrund des Versuchsaufbaus als sehr hoch zu bewerten, da in der Hälfte aller Fälle Adaptationen ausgeführt wurden, deren Ursachen für den Nutzer nicht nachvollziehbar sind.

Neben EV ist das Dialogmuster mit der impliziten Bestätigung (AI) als besonders effektiv hervorzuheben. AI erhält besonders gute Nutzerbewertungen für Nutzungskomfort und Akzeptanz und wird von den Probanden ebenso häufig als das bevorzugte Muster angegeben wie EV.

Die weiterführenden Analysen der Dialogmuster in verschiedenen Kosten-Nutzen-Bedingungen liefern interessante Aufschlüsse für deren praktische Anwendung in ABS. Die ursprüngliche Annahme, dass Kontrollmechanismen vor allem dann wichtig sind, wenn automatische Adaptationen auch Nachteile für den Nutzer bringen können, lässt sich bestätigen. Ebenso zeigt sich, dass Kontrollmechanismen bei besonders vorteilhaften Anpassungen weniger gefragt sind.

⁴⁴ Das Adaptionsdialogmuster EV erhält positive Nutzerbewertungen für Transparenz in 20 von 24 Fällen (83%) und für Kontrollierbarkeit in 19 von 24 Fällen (79%).

In diesen Fällen kommt es stärker auf den Nutzungskomfort und die Effizienz der Adaptationsdialoge an. Die Tatsache, dass sich nicht alle damit verbundenen Hypothesen durch die empirischen Daten bestätigen lassen, liegt unter Umständen an der relativ geringen statistischen Teststärke der nicht-parametrischen Analysen in Verbindung mit einer relativ kleinen Stichprobe. Umso zuverlässiger sind die signifikanten Ergebnisse der Studie zu interpretieren.

Insgesamt führen die Ergebnisse zum Schluss, dass beide Adaptationsdialogmuster AI und EV gemeinsam in einem System eingesetzt werden sollten: Ein schneller und unaufdringlicher Adaptationsprozess, der lediglich eine Rückgängig-Funktion für unerwünschte Adaptationen bietet (AI), und ein offensichtlicher und expliziter Dialog, in dem erst die Bestätigung des Nutzers abgewartet wird, bevor eine Anpassung vorgenommen wird (EV). AI eignet sich besonders gut für Situationen, in denen die Anpassungen große Vorteile für den Nutzer bringen können. Dieses Ergebnis bestätigt auch Trewins (Trewin, 2003) Behauptung, dass eine vollständige Nutzerkontrolle über ABS dann weniger wichtig ist, wenn die Adaptationen bestehende Nutzungsbarrieren überwinden können. EV hingegen sollte dann eingesetzt werden, wenn eine Anpassung mit großer Wahrscheinlichkeit auch Nachteile für den Nutzer bringen kann. Das Experiment zeigt, dass die explizite Nutzerkontrolle die negativen Effekte von automatischen Anpassungen reduzieren kann. Selbst wenn der Nutzer einer Anpassung zustimmen muss, um überhaupt einen gewissen Zugang zur gewünschten Funktionalität zu erhalten, scheint die Tatsache, dass er selbst entschieden hat, die subjektive Empfindung der mit der Anpassung verbundenen Kosten zu mildern.

Auf Grundlage dieser Ergebnisse wird die MyUI-Adaptationsengine so weiterentwickelt, dass die Entscheidung, welches Adaptationsdialogmuster bei einer systeminitiierten Laufzeitanpassung aufgerufen wird, einer einfachen Regel folgt:

- Bei Adaptationen, die ohne Substitution eines Interaktionsmusters erfolgen, wird das Dialogmuster AI verwendet. In diesen Fällen wird die ABS nur geringfügig verändert. Mit hoher Wahrscheinlichkeit ergeben sich daraus keine wesentlichen Nachteile für den Nutzer. Beispiele umfassen weniger starke Veränderungen der Schriftgrößen oder eine Veränderung des Text-Bild-Verhältnisses, ohne auf andere Bedien- und Anzeigekomponenten zurückzugreifen.
- Bei Adaptationen, die eine Substitution verwendeter Interaktionsmuster umfassen, wird das Dialogmuster EV verwendet. Über die Nutzung anderer Bedien- und Anzeigekomponenten werden derartige Anpassungen einen größeren und deutlicher wahrnehmbaren Effekt auf die ABS ausüben. Nachteile für den Nutzer, die unter anderem auch die Neuorientierung auf der veränderten ABS umfassen, sind in diesen Fällen wahrscheinlich.

Aufgrund des Experimentaldesigns, das durch die abstrakte Kosten-Nutzen-Betrachtung auf eine Generalisierbarkeit abzielt, sollten die Ergebnisse dieser Studie auf die generelle Frage von Kontrollmechanismen automatischer Adaptationen verallgemeinert werden können. Die beschriebenen Adaptationsdialoge und Kontrollmechanismen können auch in anderen ABS in ähnlicher Weise eingesetzt werden.

9.4 Empirische Evaluierung mit Software-Entwicklern und Entscheidungsträgern aus Software-Unternehmen

Die Validierung der praktischen Eignung des Ansatzes für die kommerzielle Software-Entwicklung basiert auf Studien mit potenziellen ABS-Entwicklern und Entscheidungsträgern aus Software-Unternehmen. Die Untersuchungen wurden im Rahmen des MyUI-Forschungsprojekts von Projektpartnern in Birmingham, England und Sevilla, Spanien durchgeführt. Die folgenden Ausführungen fassen relevante Ergebnisse der Studien zusammen und beruhen auf einem internen Projektbericht (Sánchez et al., 2012).

9.4.1 Fragestellung

Die folgenden Fragestellungen werden adressiert:

- *Effiziente Entwicklung*: Ermöglicht die ABS-Infrastruktur, dass ABS bereits mit geringem Spezifikationsaufwand erstellt werden können? (Frage 12 der Tabelle 15; die untersuchte Anforderung ist in Abschnitt 2.6.1 beschrieben.)
- *Geringe Einstiegshürde und Kompetenzanforderungen*: Unterstützt die ABS-Infrastruktur auch für weniger gut ausgebildete Entwickler einen leichten Einstieg in die Entwicklung von ABS? (Frage 14 der Tabelle 15; die untersuchte Anforderung ist in Abschnitt 2.6.2 beschrieben.)
- *Nachvollziehbarkeit und Beherrschbarkeit*: Unterstützt die ABS-Infrastruktur die Entwickler dabei, den Zusammenhang zwischen einem AAIM und den generierten ABS zu verstehen und das Ergebnis der ABS-Generierung in ausreichendem Maße zu beeinflussen? (Frage 15 der Tabelle 15; die untersuchte Anforderung ist in Abschnitt 2.6.3 beschrieben.)

9.4.2 Methode

Zur Beantwortung der Evaluationsfragen wird eine Fokusgruppen-Methode (vgl. Nielsen, 1997, Krueger & Casey, 2000 und Spath et al., 2012) eingesetzt. Insgesamt werden vier Sitzungen mit insgesamt 33 Probanden durchgeführt: Zwei Sitzungen in Birmingham, England und zwei in Sevilla, Spanien. Davon nehmen jeweils an einer Sitzung Entscheidungsträger von Software-Unternehmen, an der anderen Sitzung Software-Entwickler teil (vgl. Tabelle 26).

| | Entscheidungsträger | Software-Entwickler |
|------------|---------------------|---------------------|
| Birmingham | 10 | 4 |
| Sevilla | 11 | 8 |

Tabelle 26 Anzahl der Probanden (nach Rolle und Ort) zur Evaluierung der praktischen Eignung des Ansatz für die kommerzielle Software-Entwicklung

Je nach vorwiegendem Erfahrungshintergrund der beiden beteiligten Gruppen werden die drei adressierten Fragestellungen mit unterschiedlichem Gewicht behandelt. Während die Fragen der Wirtschaftlichkeit und Effizienz (Frage 12 der Tabelle 15) vor allem mit den Entscheidungsträgern diskutiert und bewertet werden, geht es in den Sitzungen mit den Entwicklern vorwiegend um

die Einstiegshürden und Kompetenzanforderungen (Frage 14 der Tabelle 15) sowie um die Nachvollziehbarkeit und Beherrschbarkeit der ABS-Generierung (Frage 15 der Tabelle 15).

Die Dauer der einzelnen Testsitzungen beträgt jeweils ca. drei Stunden für die Entscheidungsträger, ca. vier Stunden für die Entwickler. Der grobe Ablauf ist für beide Zielgruppen identisch. Nach einer Einführung in die Zielsetzung und die wesentlichen Konzepte der ABS-Generierung mit der MyUI-Infrastruktur folgt eine intensivere Beschäftigung mit der Vorgehensweise zur ABS-Entwicklung anhand eines konkreten Beispiels. Danach werden Verständnisfragen der Teilnehmer in einer Diskussionsrunde geklärt, bevor die Teilnehmer einen Fragebogen ausfüllen, in dem die interessierenden Eigenschaften des MyUI-Ansatzes bewertet werden. Schließlich ist Zeit für eine offene Diskussion, in der es vor allem um persönliche Meinungen, Vorbehalte, ungelöste Herausforderungen und Nutzenpotenziale geht. Der genaue Ablauf der Testsitzungen ist in Tabelle 27 dargestellt.

| Ablauf einer Testsitzung für ... | | |
|---|--|---|
| | Entscheidungsträger | Software-Entwickler |
| Phase 1: Einführung | <ol style="list-style-type: none"> 1. Präsentation der allgemeinen Zielsetzung und Grundzüge des Ansatzes durch den Moderator. 2. Präsentation der Rolle und der Aufgaben des Entwicklers im Prozess der ABS-Erstellung durch den Moderator: Insbesondere wird erläutert, dass der Entwickler im Wesentlichen nur das AAIM erstellt und die ABS dann automatisch generiert wird. Einflussmöglichkeiten auf die erzeugte ABS werden dargestellt. | |
| Phase 2: Vertiefung | <p>Vorführung der Erstellung einer ABS für die interaktive Wettervorhersage (vgl. Abschnitt 0) durch den Moderator:</p> <ol style="list-style-type: none"> 1. Der Moderator erläutert die Grundzüge der AAIM-Notation. 2. Der Moderator stellt die Funktionalität der Wetter-Applikation vor. 3. Der Moderator zeigt Schritt für Schritt, wie das AAIM der Wetter-Applikation mit Hilfe der MyUI-Infrastruktur erstellt wird und die Voraussetzungen für die ABS-Generierung geschaffen werden. 4. Der Moderator beantwortet Verständnisfragen der Teilnehmer. | <p>Erstellung einer ABS für die interaktive Wettervorhersage (vgl. Abschnitt 09.2) durch die Teilnehmer:</p> <ol style="list-style-type: none"> 1. Der Moderator erklärt die AAIM-Notation. 2. Der Moderator stellt die für die Aufgabe benötigten Werkzeuge und Funktionen der MyUI-Infrastruktur vor. 3. Der Moderator stellt die Funktionalität der Wetter-Applikation vor. 4. Jeder Teilnehmer erstellt mit der MyUI-Infrastruktur eine ABS für die Wetter-Applikation. 5. Währenddessen beantwortet der Moderator Verständnisfragen der Teilnehmer. |
| Phase 3: Fragebogen | Beantwortung eines Fragebogens durch die Teilnehmer. Schwerpunktthemen sind dabei: Marktpotenziale, Akzeptanz, Gebrauchstauglichkeit und Nützlichkeit des Ansatzes. | |
| Phase 4: Diskussion | Gruppendiskussion zu persönlichen Meinungen und allgemeinen Einschätzungen des Ansatzes, sowie zu erwarteten Herausforderungen und Potenzialen einer Einführung des Ansatzes in konkreten Projekten | |

Tabelle 27 Ablauf der Testsitzungen mit Entscheidungsträgern in Software-Unternehmen und Software-Entwicklern

Die in der Vertiefung verwendete Wetter-Applikation besteht aus zwei Bereichen: Ein Dialog zur Auswahl eines Ortes und eine Bildschirmansicht zur Darstellung der Wettervorhersage für den ausgewählten Ort (vgl. Abbildung 32 im Anhang, Bildschirmansicht 5 und 6).

Zur Erstellung der ABS für die Wetter-Applikation werden den Entwicklern alle in Abschnitt 8.7 beschriebenen Werkzeuge zur Verfügung gestellt. Damit hängen die Evaluationsergebnisse dieser Studie nicht nur von der Qualität des Ansatzes dieser Arbeit ab. Auch die Werkzeuge, die eine Implementierung und Weiterentwicklung der in dieser Arbeit beschriebenen Konzepte darstellen, fließen in die Betrachtung ein. Eine praktische und empirische Evaluierung mit den wesentlichen Zielgruppen der ABS-Infrastruktur – nämlich mit Entscheidungsträgern und Entwicklern in Software-Unternehmen – kann nur dann valide Aussagen über die zu erwartende Akzeptanz in der kommerziellen Software-Entwicklung erbringen, wenn die Teilnehmer der Studie einen realitätsnahen Eindruck möglicher Entwicklungsszenarien bekommen. Daher erscheint der Einsatz der Entwicklungswerkzeuge gerechtfertigt, auch wenn deren Realisierung nur in den Grundzügen auf dieser Arbeit beruht und sie zum Zeitpunkt der Evaluation eher als frühe Prototypen denn als marktreife Produkte vorliegen.

Aus Zeitgründen wurde mit der Modellierung der Wetter-Applikation bewusst eine einfache Testaufgabe ausgewählt. So kann ein Entwickler den gesamten Prozess innerhalb einer Testsitzung durchlaufen und erhält einen guten Eindruck vom Gesamtsystem.

Die in den Testsitzungen verwendeten Fragebögen bestehen aus 29 Items für die Entscheider und 27 Items für die Entwickler. Für die Evaluierung der hier untersuchten Eigenschaften des Ansatzes werden nur die dafür relevanten Items der Fragebögen betrachtet (s. Tabelle 28).

Anhand einer fünfstelligen Likert-Skala (1-5) geben die Teilnehmer das Ausmaß ihrer Zustimmung zu den Aussagen der einzelnen Items an. Die meisten Aussagen sind positiv formuliert, so dass eine hohe Zustimmung auf der Likert-Skala einer positiven Bewertung entspricht. Für die Auswertung der negativ formulierten Items (Fragen 4, 7 und 15 der Tabelle 28) werden die auf der Zustimmungsskala erfassten Werte transformiert, so dass auch hier hohe transformierte Werte positiven Bewertungen entsprechen⁴⁵. Im Folgenden werden nur die transformierten Werte berichtet. Generell können daher Werte über 3 als positive Bewertungen interpretiert werden, Werte unter 3 als negative Bewertungen. Mittelwerte zwischen 3 und 4 werden als „tendenziell positiv“ bezeichnet, Bewertungen über 4 als „deutlich positiv“.

⁴⁵ Transformationsregel für die auf der Likert-Skala erfassten Werte bei negativ formulierten Fragebogen-Items:
1→5, 2→4, 3→3, 4→2, 5→1.

| Entscheidungsträger | Software-Entwickler |
|--|--|
| Effiziente Entwicklung | |
| 1. „Die Arbeit mit den MyUI-Entwicklungswerkzeugen wird die effiziente Entwicklung von barrierefreien Anwendungen unterstützen.“ 2. „Im Vergleich zur aktuellen Praxis bei der Entwicklung von barrierefreien und personalisierten Benutzungsschnittstellen wird die MyUI-Entwicklungsumgebung die benötigten Ressourcen signifikant reduzieren.“ 3. „Die Benutzung der MyUI-Entwicklungswerkzeuge, um ABS zu generieren, wird die Produktentwicklungszeit (Time-to-Market) verbessern.“ | 4. „Die Komplexität des Programmierens mit MyUI ist höher als ich es von meinen üblichen Projekten zur Applikationsentwicklung gewohnt bin.“ 5. „Die Unterstützung von MyUI bei der Entwicklung von barrierefreien Benutzungsschnittstellen erlaubt es den Entwicklern, sich auf andere Aspekte der Applikationsentwicklung zu konzentrieren und diese zu verbessern.“ |
| Geringe Einstiegshürde und Kompetenzanforderungen | |
| 6. „Eine bestehende Applikation mit der MyUI-Entwicklungsumgebung anzupassen ist einfach und intuitiv.“ 7. „Die Entwicklung mit der MyUI-Entwicklungsumgebung erfordert zu viel technisches Wissen über die zugrundeliegenden Konzepte und Mechanismen der MyUI-Technologien.“ | 8. „Es war eine einfache Aufgabe für mich, die Wetter-Applikation mit der MyUI-Entwicklungsumgebung zu erstellen.“ 9. „Die MyUI-Entwicklungsumgebung ist ein handliches Werkzeug und bietet einen leichten Einstieg in das Programmieren.“ 10. „Schon nach kurzer Zeit werde ich mich kompetent fühlen, um auch komplexe Applikationen leicht mit der MyUI-Entwicklungsumgebung entwickeln zu können.“ |
| Nachvollziehbarkeit und Beherrschbarkeit | |
| 11. „Das AAIM ist gut geeignet, um Entwicklern eine klare und verständliche Visualisierung der Interaktionslogik einer Applikation zu bieten.“ 12. „Durch die Möglichkeit, eine Vorschau auf die Ergebnisse der adaptiven Applikation zu bekommen, können Fehler leicht behoben werden und Änderungen leicht vorgenommen werden.“ | 13. „Die Notation des AAIM ist leicht zu verstehen.“ 14. „Das AAIM vermittelt mir einen guten Eindruck, wie die Benutzungsschnittstelle schließlich aussehen wird.“ 15. „Die Arbeit mit der MyUI-Entwicklungsumgebung lässt mir keinerlei Kontrolle über die entstehende Benutzungsschnittstelle.“ |

Tabelle 28 Deutsche Übersetzung der Fragebogen-Items, anhand derer Entscheidungsträger von Software-Unternehmen und Entwickler die praktische Eignung des Ansatzes für die kommerzielle Software-Entwicklung bewerten (Original in Englisch)

9.4.3 Ergebnisse

In Abbildung 31 sind die Ergebnisse der Fragebogenerhebung dargestellt. Es fällt auf, dass die drei negativ formulierten Fragebogen-Items 4, 7 und 15 in allen drei Kategorien mit Abstand die schlechtesten Bewertungen erhalten. Möglicherweise haben einige Probanden die Ausrichtung dieser Skalen fehlinterpretiert. Diese Annahme wird darüber hinaus durch relativ hohe Varianzen der Bewertungen gestützt, die zum Ausdruck bringen, dass sich die Probanden bei diesen Items nicht einig waren. Die Aussagekraft der Ergebnisse dieser Fragebogen-Items muss daher in Frage gestellt werden.

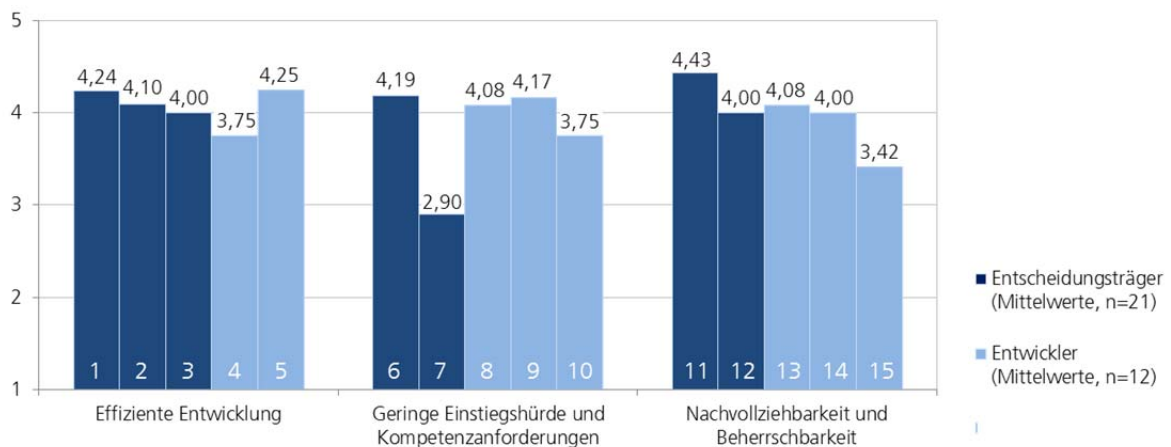


Abbildung 31 Ergebnisse der Befragung von Entscheidungsträgern von Software-Unternehmen und Entwicklern zur praktischen Eignung des Ansatzes für die kommerzielle Software-Entwicklung (Mittelwerte der Bewertungen auf fünfstelligen Likert-Skalen)

Für alle drei Kriterien liegen die Gesamtmittelwerte der Bewertungen von Entscheidungsträgern und Entwicklern in einem positiven Bereich (vgl. Tabelle 29). Schließt man die Bewertungen der negativ formulierten Likert-Items aus, um die oben beschriebenen Verzerrungen zu kontrollieren, verbessern sich die Ergebnisse und es ergeben sich deutlich positive Bewertungen für alle drei Kriterien (rechte Spalte der Tabelle 29). Allerdings leidet dann die Konstruktvalidität, da die ausgeschlossenen Items wesentliche Gesichtspunkte der evaluierten Kriterien abdecken.

| Kriterium | Mittelwert aller Bewertungen | Mittelwert ohne Negativ-Item |
|---|------------------------------|------------------------------|
| Effiziente Entwicklung | 4,08 | 4,13 |
| Geringe Einstiegshürde und Kompetenzanforderungen | 3,76 | 4,07 |
| Nachvollziehbarkeit und Beherrschbarkeit | 4,04 | 4,15 |

Tabelle 29 Mittelwerte aller Bewertungen von Entscheidungsträgern und Entwicklern zu den betrachteten Kriterien. Die rechte Spalte stellt die Mittelwerte dar, wenn die negativ formulierten Aussagen nicht in die Auswertung eingehen.

Die *Effizienz der Entwicklung* wird insgesamt sehr gut bewertet. Die Entscheidungsträger sind der Meinung, dass MyUI die Entwicklung barrierefreier Anwendungen effizient unterstützen wird (Mittelwert_{Item 1}: 4,24). Im Vergleich zur aktuellen Praxis erwarten sie, dass man mit MyUI barrierefreie Anwendungen mit weniger Ressourcen (Mittelwert_{Item 2}: 4,10) und geringerem Zeitaufwand (Mittelwert_{Item 3}: 4,00) erstellen kann. Auch die Entwickler erkennen Einsparpotenziale durch die Verwendung des vorgestellten Ansatzes. Im negativ formulierten Item 4 (Mittelwert_{Item 4}: 3,75) geben nur zwei der zwölf Befragten an, dass durch MyUI die Komplexität ihrer Arbeit steigen würde. Zehn der zwölf Probanden erhoffen sich durch die ABS-Generierung neue Freiräume, um sich auf andere Aspekte der Entwicklungsarbeit konzentrieren zu können (Mittelwert_{Item 5}: 4,25).

Einstiegshürde und Kompetenzanforderungen werden insgesamt tendenziell positiv bewertet. Die Entscheidungsträger sind der Meinung, dass der vorgestellte Ansatz eine einfache und intuitive ABS-Entwicklung ermöglicht (Mittelwert_{Item 6}: 4,19). Während die Entwickler die Bearbeitung der Testaufgabe als einfach bezeichnen (Mittelwert_{Item 8}: 4,08), bewerten sie den

Einarbeitungsaufwand, um ABS auch für komplexe Anwendungen mit Leichtigkeit entwickeln zu können, etwas zurückhaltender (Mittelwert_{Item10}: 3,75). Insgesamt sind die Entwickler dennoch überzeugt, dass die MyUI-Infrastruktur handlich ist und einen leichten Einstieg in die ABS-Entwicklung bietet (Mittelwert_{Item9}: 4,17). Der deutliche Ausreißer in diesem Themenkomplex ist die negativ formulierte Frage bezüglich der erforderlichen MyUI-spezifischen Kenntnisse (Mittelwert_{Item7}: 2,90). Vertraut man darauf, dass die Befragten dieses Item richtig interpretiert haben, so bleibt festzustellen, dass die Entscheidungsträger tendenziell der Auffassung sind, dass der Einarbeitungsaufwand in die MyUI-Technologien zu hoch ist.

Die *Nachvollziehbarkeit und Beherrschbarkeit* der ABS-Generierung werden insgesamt deutlich positiv bewertet. Besonders positiv ist das Votum der Entscheidungsträger bezüglich der klaren und verständlichen Visualisierung der Applikationslogik durch das AAIM (Mittelwert_{Item11}: 4,43). Diese Einschätzung teilen auch die Entwickler (Mittelwert_{Item13}: 4,08). Darüber hinaus sind sie der Meinung, dass das AAIM einen guten Eindruck der erzeugten ABS vermitteln kann (Mittelwert_{Item14}: 4,00) und damit leicht die Beziehung zwischen Modell und konkreter Benutzungsschnittstelle hergestellt werden kann. Die Ergebnisse zur Einflussnahme auf die generierte ABS fallen nicht so homogen aus. Während die Entscheidungsträger die Möglichkeiten der Korrektur und Änderung als gut bewerten (Mittelwert_{Item12}: 4,00), ergibt sich bei den Entwicklern nur eine tendenziell positive Bewertung der Kontrollmöglichkeiten im negativ formulierten Item 15 (Mittelwert_{Item15}: 3,75).

Die Validierungsergebnisse werden durch die folgenden Äußerungen aus den offen geführten Diskussionsrunden zum Ende jeder Sitzung ergänzt:

- *Effiziente Entwicklung*
Im Zusammenhang mit größeren und komplexeren Projekten bestehen Bedenken, dass der Entwicklungsansatz - und insbesondere die Erstellung des AAIM - schwierig werden könnte. Für kleinere und einfachere Applikationen liegt der Nutzen jedoch auf der Hand.
- *Geringe Einstiegshürde und Kompetenzanforderungen*
Die Entwickler wünschen sich mehr einführendes Material und Hilfestellungen, um einen leichten Einstieg in die Arbeit mit der MyUI-Infrastruktur zu finden. Insbesondere werden Beispielprojekte gewünscht, die als Orientierungshilfe und Ausgangspunkt für neue Entwicklungen dienen können.
- *Nachvollziehbarkeit und Beherrschbarkeit*
Die Verständlichkeit des AAIM könnte noch verbessert werden, wenn der Nutzer des Entwicklungswerkzeugs selbst bestimmen kann, welche Details des Modells angezeigt bzw. ausgeblendet werden. In einigen Situationen scheint es wichtig zu sein, eine vollständige Darstellung mit allen eingebundenen Applikationsfunktionen und allen Parametern zu bekommen, während in anderen Situationen eine einfachere, reduzierte Ansicht bevorzugt wird. Darüber hinaus wünschen sich die Entwickler nicht nur eine statische Vorschau der generierten ABS, sondern würden gerne auch einen besseren Eindruck davon erhalten, wie sich Adaptationen dem Nutzer darstellen werden. Bezüglich der Einflussnahme auf die generierten Benutzungsschnittstellen haben beide Teilnehmergruppen die Bedeutung der Differenzierung der eigenen Marke gegenüber potenziellen Mitbewerbern, die auch mit MyUI-Technologien arbeiten, hervorgehoben.

9.4.4 Interpretation und Diskussion

Die Evaluierung der in diesem Kapitel behandelten Kriterien zeigt sich methodisch schwierig. Zwar hängen die bewerteten Eigenschaften grundsätzlich von dem in dieser Arbeit vorgestellten konzeptionellen Rahmenmodell für ABS ab. Ihre empirische Überprüfung ist jedoch nur mittels einer konkreten Umsetzung in einer Entwicklungsumgebung möglich. Damit beruhen die Ergebnisse auf den Eindrücken und Erwartungen der beteiligten Entscheidungsträger und Entwickler, nachdem sie das MyUI-Rahmenmodell über die Entwicklungswerkzeuge genutzt haben. Eine zusätzliche Einschränkung ergibt sich daraus, dass es sich bei den genutzten Werkzeugen nicht um marktreife Produkte, sondern um Prototypen aus einem Forschungsprojekt handelt.

Aufgrund der durchweg positiven Evaluationsergebnisse kann jedoch darauf geschlossen werden, dass der vorgestellte Ansatz bezüglich der drei behandelten Kriterien wesentliche Fortschritte gegenüber dem Stand der Technik und Forschung erzielen kann. Im Vergleich zu den akademischen Ansätzen, die für die ABS-Generierung mehrere verschiedene, miteinander verbundene Modelle erfordern (vgl. Abschnitt 2.6.1), beschränkt sich MyUI auf einen pragmatischen Ansatz mit einem einzigen Modell: dem AAIM. Dieses basiert auf Zustandsdiagrammen, einem grafischen sowie gut etablierten und weithin bekannten Standard. Dieser Vorteil schlägt sich offensichtlich auch in den positiven Bewertungen der Probanden bezüglich Entwicklungseffizienz, Einstiegshürden und Kompetenzanforderungen nieder. Auch im Vergleich zur herkömmlichen Entwicklungspraxis trauen die Probanden dem dargestellten Ansatz der ABS-Generierung deutliche Effizienzvorteile zu.

Neuartige Ansätze, die die heutige Praxis der Software-Entwicklung radikal verändern würden, erfordern zunächst eine gewisse Einarbeitung in die zugrundeliegenden Konzepte. Dieses Problem ist nicht auf MyUI beschränkt. Dennoch ergibt sich daraus eine signifikante Hürde für die kommerzielle Nutzung. Entsprechend der Anregungen der Probanden werden gut verständliche und attraktive Einführungen und Anleitungen enorm wichtig sein, um Entwickler für den Ansatz zu gewinnen. Die Befragungsergebnisse deuten jedoch darauf hin, dass eine gewisse Einfachheit und Intuitivität der MyUI-Konzepte hierfür eine gute Grundlage geschaffen haben. Dies zeigt sich auch in der umfassend attestierten Nachvollziehbarkeit der Beziehungen zwischen ABS-Modell und den konkreten Benutzungsschnittstellen.

Das grundsätzliche Problem des wahrgenommenen Kontrollverlustes der Entwickler bei der Arbeit mit generierten Benutzungsschnittstellen kann durch den Ansatz anscheinend nicht vollständig ausgeräumt werden. Zwar werden einige der gebotenen Einflussmöglichkeiten, wie z.B. das Customization-Profil, erkannt und als gut bewertet. Diese scheinen den Befragten jedoch noch nicht weit genug zu gehen. Einerseits müssen die Potenziale, die die Erweiterbarkeit des Musterrepositoriums auch für eine Differenzierung durch ein markenspezifisches Erscheinungsbild bieten, in Einführungsmaterialien und Anleitungen klar kommuniziert werden. Andererseits sollten zukünftig noch weitere Einflussmöglichkeiten in Betracht gezogen werden, die es den Entwicklern ermöglichen, die resultierende ABS direkt auf Basis der generierten Vorschau zu bearbeiten, ohne Änderungen am Repository vornehmen zu müssen.

9.5 Zusammenfassung der Evaluierung

| Anforderungen an Lösungsansatz | Bewertung und Kommentar |
|--|---|
| Adaptivität: automatische systeminitiierte Anpassungen (2.1) | ● Der Ansatz bietet eine solide Infrastruktur für automatische Anpassungen an verschiedene Kontextbedingungen. Umfang und inhaltliche Eignung der Anpassungen hängen vom Gestaltungswissen der Entwurfsmuster ab. |
| Adaptivität und Adaptierbarkeit in einem System (2.1) | ● Neben automatischen Adaptationen stehen weitreichende Möglichkeiten der nutzerinitiierten Anpassung zur Verfügung. |
| Kontinuierliche Anpassung an ein selbstlernendes Profil während der Nutzung (2.2) | ● Anpassungen erfolgen kontinuierlich während der Nutzung und ohne wesentliche Verzögerungen. |
| Transparenz: erkennbare, verstehbare und nachvollziehbare Adaptationen (2.3.1) | ● Durch die Adaptationsdialogmuster werden automatische Anpassungen für die Nutzer gut erkennbar und verstehbar. |
| Kontrollierbarkeit automatischer Adaptationen (2.3.2) | ● Durch die Adaptationsdialogmuster werden automatische Anpassungen für die Nutzer gut kontrollierbar. |
| Anpassungen an Nutzer, Umgebung und technische Ausstattung (0) | ● Der Ansatz unterstützt Anpassungen an Nutzer, Umgebung und technische Ausstattung. Die Referenzimplementierung konzentriert sich vorwiegend auf perzeptuelle, motorische und kognitive Nutzereinschränkungen. |
| Anpassungen der Informationsdarstellung, Interaktion und Navigation (0) | ◐ Im Vergleich zu den umfangreichen Anpassungen der Informationsdarstellung und Interaktion bietet der Ansatz nur geringere Adaptationsmöglichkeiten der Navigation. |
| Modulare und erweiterbare Adaptationsmechanismen (2.5) | ● Die Modularität ist voll erfüllt. Die ABS-Infrastruktur ist auf Erweiterbarkeit ausgelegt. Mechanismen zur Konsistenzsicherung zwischen natürlichsprachlicher und software-technischer Repräsentation der Entwurfsmuster fehlen jedoch. |
| Adaptationsmechanismen öffentlich zugänglich und offen für externe Beiträge (2.5) | ◐ Die öffentliche Zugänglichkeit der Entwurfsmuster und deren Erweiterung durch externe Beiträge sind vorgesehen und angelegt. Ausgearbeitete Lösungen für deren effektive Umsetzung sind jedoch nicht Teil dieser Arbeit. |
| Effiziente Entwicklung: wiederverwendbare Komponenten und geringer Spezifikationsaufwand bei ausreichender Mächtigkeit (2.6.1) | ● Die befragten Entwickler und Entscheidungsträger bewerten die Effizienz des Ansatzes sehr positiv. Wesentliche ABS-Komponenten können wiederverwendet werden. Einschränkungen betreffen die domänenübergreifende Nutzung der Interaktionsmuster und die technologieübergreifende Wiederverwendung auf der Software-Ebene. |
| Geringe Einstiegshürde und Kompetenzanforderungen auf Seiten der Entwickler (2.6.2) | ● Die befragten Entwickler und Entscheidungsträger bewerten diese Aspekte positiv. |
| Generierung der ABS für Entwickler nachvollziehbar und beherrschbar (2.6.3) | ● Die befragten Entwickler und Entscheidungsträger bewerten diese Aspekte positiv. |

Legende:

● Breite Abdeckung, vertiefte Konzepte

◐ Teilweise Abdeckung, vertiefte Konzepte

◑ Breite Abdeckung, einfache Konzepte

○ Geringe Abdeckung, einfache Konzepte

Tabelle 30 Ergebnisse der Evaluierung des Ansatzes anhand der hergeleiteten Anforderungen

In Tabelle 30 sind die Ergebnisse der Validierung vor dem Hintergrund der in Kapitel 2 entwickelten Anforderungen zusammengefasst. Es zeigt sich, dass der in dieser Arbeit entwickelte Ansatz die Anforderungen an ABS zur Überwindung von Nutzungsbarrieren in deutlich höherem Maße erfüllt als die in Tabelle 1 bewerteten Lösungsansätze. Dieses Ergebnis untermauert den Beitrag dieser Arbeit zum Stand der Wissenschaft und Forschung.

10 Zusammenfassung und Ausblick

10.1 Zusammenfassung

Die vorliegende Arbeit beschreibt einen neuartigen Ansatz für Benutzungsschnittstellen, die sich selbstständig und dynamisch während der Nutzung an individuelle Nutzerbedürfnisse, unterschiedliche Umgebungsbedingungen und Interaktionsgeräte anpassen können. Dadurch kann eine hohe Zugänglichkeit (Accessibility) interaktiver Computersysteme erreicht werden: In Verbindung mit einer kontinuierlichen Erfassung des Nutzungskontexts können die in dieser Arbeit entwickelten Mechanismen zur Generierung und Adaptation der Benutzungsschnittstelle dafür sorgen, dass jeder Nutzer im Laufe der Interaktion eine Benutzungsschnittstelle erhält, die seinen individuellen Bedürfnissen zunehmend besser entspricht. Zwar besteht auch die Möglichkeit, dass die Nutzer die Einstellungen der Benutzungsschnittstelle selbst nach den eigenen Wünschen verändern. Aufgrund der automatischen Anpassungen erhalten aber auch die Nutzer, die aufgrund von Einschränkungen oder geringer Computer-Erfahrung Konfigurationsdialoge scheuen oder nicht bedienen können, eine individualisierte Benutzungsschnittstelle. Darüber hinaus können Nutzungsbarrieren und Bedienprobleme durch automatische Adaptationen direkt behoben werden, wenn sie auftreten. Veränderliche Nutzerbedürfnisse, die sich zum Beispiel häufig im Zuge des Älterwerdens ergeben, können durch dynamische Anpassungen selbstständig kompensiert werden.

Technisch beruht der Ansatz auf einer Modularisierung der Benutzungsschnittstelle. Alle anzupassenden Gestaltungsaspekte werden als Entwurfsmuster beschrieben. Für jeden Aspekt liegen mehrere alternative Entwurfsmuster vor, um verschiedene Lösungen für verschiedene Anforderungen anbieten zu können. Die Generierung der ABS zu Beginn der Interaktion basiert auf einer Komposition der für die aktuellen Kontextanforderungen am besten geeigneten Entwurfsmuster. Für Adaptationen während der Interaktion werden einzelne Entwurfsmuster durch andere ersetzt. Damit der Nutzer die intelligenten Anpassungen erkennen und kontrollieren kann, werden spezielle Anpassungsdialoge verwendet, die ebenfalls als Entwurfsmuster vorliegen.

Der Ansatz ist generisch und kann prinzipiell für verschiedene Anwendungen verwendet werden, um beliebige Nutzungsbarrieren durch entsprechende Anpassungen zu überwinden. Das dafür notwendige Gestaltungswissen wird aus den Entwurfsmustern gezogen. Im Rahmen einer ersten Implementierung wurde nur ein Grundstock von Entwurfsmustern entwickelt. Die Erweiterbarkeit des Entwurfsmusterrepositoriums ist daher eine wichtige Eigenschaft des Ansatzes, um sukzessive eine möglichst umfassende Abdeckung verschiedener Anforderungen, Einschränkungen und Interaktionssituationen zu ermöglichen. Das Entwurfsmusterrepositorium ist öffentlich zugänglich, um möglichst viele externe Experten an der Weiterentwicklung der Entwurfsmuster beteiligen zu können.

Für Entwickler von ABS bietet der Ansatz große Vorteile. Sie müssen nur ein abstraktes Modell der Interaktion erstellen, das als Grundlage für die automatische Generierung der ABS dient. Das eigens entwickelte Modellierungsformat setzt auf einen möglichst geringen Spezifikationsaufwand und den Einsatz von Zustandsdiagrammen als leicht verständlichen und weit verbreiteten Standard. Dadurch wird versucht, den Ansatz auch für die kommerzielle Software-Entwicklung attraktiv zu machen.

Die Evaluierung des Ansatzes erfolgt am Beispiel von webbasierten Applikationen für das interaktive Fernsehen, deren Benutzungsschnittstellen sich an typische Einschränkungen älterer Nutzer anpassen. Neben einer technischen Validierung anhand einer Referenzimplementierung werden empirische Studien mit potenziellen Endnutzern, potenziellen ABS-Entwicklern und Entscheidungsträgern aus Software-Unternehmen durchgeführt. Die Ergebnisse zeigen, dass der entwickelte Ansatz die relevanten Anforderungen an ABS zur Überwindung von Nutzungsbarrieren gut erfüllt und einen deutlichen Fortschritt gegenüber dem aktuellen Stand der Forschung und Technik erzielt.

10.2 Ausblick

Für einen gewinnbringenden Einsatz der vorgestellten Konzepte und Technologien sind viele weitere Anwendungsfelder denkbar, in denen die Barrierefreiheit eine zunehmend wichtige Rolle spielt. Ein Beispiel sind Anwendungen des E-Governments. Gerade für IT-gestützte Informations- und Transaktionsangebote öffentlicher Einrichtungen ist es wichtig, dass alle Bürger unabhängig von ihrer technischen Ausstattung, ihrer Nutzungskompetenz und eventueller körperlicher oder geistiger Einschränkungen vollen Zugang erhalten können. Auch für Mensch-Technik-Schnittstellen in der industriellen Fertigung kann sich ein großer Nutzen ergeben. Dynamisch adaptive Benutzungsschnittstellen können eine sinnvolle Erweiterung der Kontextadaptivität von „Smart Factories“ (Westkämper et al., 2005) darstellen. So könnten im Fall einer Maschinenstörung die Benutzungsschnittstellen für Instandhalter und Fertigungsleiter nicht nur an den aktuellen Zustand der Fertigung, sondern auch an individuelle Nutzereigenschaften wie zum Beispiel der Erfahrung mit der vorliegenden Störung angepasst werden (vgl. Lucke et al., 2008). Darüber hinaus steigt im Zusammenhang mit der Entwicklung zu zunehmend vernetzten und intelligenten Produktionsumgebungen die Komplexität und Diversität der Arbeitsaufgaben der einzelnen Mitarbeiter (vgl. Spath & Weisbecker, 2013). Gleichzeitig steigt die Heterogenität der Belegschaft durch den demografischen Wandel und die zunehmende Internationalisierung. ABS können hier zum Beispiel rollen- und situationsgerechte Visualisierungsformen ermöglichen und individualisierte Hilfestellungen bei der Erledigung anspruchsvoller Arbeiten bieten.

Motorisch bedingte Nutzungsbarrieren können häufig nicht allein durch Anpassungen in der Software der Benutzungsschnittstelle überwunden werden. Daher könnte eine Erweiterung des Ansatzes um „adaptiv variable Stellteile“ (Petrov & Maier, 2009) interessant sein. Veränderungen der physischen Eigenschaften von in Hardware realisierten Eingabeelementen könnten in diesem Zusammenhang nicht nur eingesetzt werden, um dem Nutzer Rückmeldungen über Systemzustände und verfügbare Bedienoptionen zu geben (vgl. Maier et al., 2008), sondern auch, um deren Funktionsweise an die motorisch-physischen Anforderungen eines spezifischen Nutzers anzupassen. In Kombination mit dem vorgestellten Ansatz können so Bediensysteme entstehen, die sich sowohl in den Bereichen der Software als auch der Hardware der Benutzungsschnittstelle auf individuelle und situative Anforderungen einstellen.

Der vorgestellte Ansatz würde die aktuelle Praxis der Gestaltung und Entwicklung von Benutzungsschnittstellen radikal verändern. Um dennoch Akzeptanz im Markt zu finden, wird es daher wichtig sein, dass der Ansatz einen deutlichen, offensichtlichen Mehrwert mit sich bringt und die Kosten seiner Einführung und Nutzung gering sind. Für die Weiterentwicklung der ABS-Infrastruktur ergeben sich insbesondere zwei Ziele: Der Einstieg muss auch für weniger technisch-versierte Entwickler noch einfacher gestaltet werden und die Vorbehalte gegen den oft befürchteten Kontrollverlust über generierte Benutzungsschnittstellen müssen ausgeräumt

werden. Der aktuelle Erfolg des „Responsive Web Designs“ (Marcotte, 2011), bei dem das Layout von Webseiten durch verschiedene CSS an das jeweils genutzte Gerät angepasst wird, zeigt, dass der Nutzen dynamischer Selbstanpassungen mehr und mehr geschätzt wird. Es besteht also die Hoffnung, dass sich angesichts des wachsenden Bewusstseins für die zunehmende Diversität der adressierten Nutzer auch eine höhere Akzeptanz für komplexere intelligente Anpassungsmechanismen einstellt.

Der Mehrwert des Ansatzes wird nicht nur von den Eigenschaften der vorgestellten Infrastruktur, sondern vor allem auch von Umfang und Qualität der in den Entwurfsmustern gebotenen Lösungsalternativen abhängen. Hierfür wird es wichtig sein, möglichst viele Fachleute mit spezifischem Gestaltungswissen zu Benutzungsschnittstellen für besondere Nutzerbedürfnisse für die aktive Mitarbeit an der Erweiterung und Verfeinerung des Entwurfsmusterrepositoriums zu gewinnen. Mit diesem Ziel werden sich zukünftige Arbeiten der Entwicklung entsprechender Strategien und einer geeigneten technischen und organisatorischen Infrastruktur für eine offene Kooperationsplattform widmen.

Eine weitere Voraussetzung für die Effektivität des entwickelten Ansatzes ist die Qualität des verfügbaren Kontextwissens, das als Grundlage für die Anpassungen der ABS dient. Die vorliegende Arbeit konzentriert sich auf die efferente Komponente des adaptiven Systems. Der Lösungsansatz beschreibt daher die Mechanismen zur Anpassung einer ABS an spezifische Kontextbedingungen. Wie das dafür benötigte Kontextwissen erfasst und gepflegt wird, ist jedoch Gegenstand der afferenten und inferenten Komponenten, die in dieser Arbeit nicht betrachtet werden. Die Erfahrungen aus dem MyUI-Forschungsprojekt⁴⁶ zeigen allerdings, dass es schwierig sein kann, verlässliche Informationen über den Nutzer und seinen Kontext direkt aus dem Interaktionsgeschehen zu ziehen (vgl. Hernández et al., 2011). Auch der Einsatz von Erkennungstechnologien zur Erfassung relevanter Ereignisse und Zustände, die sich nicht an der ABS niederschlagen, ist nicht unproblematisch. Neben den technischen Problemen mit der Erkennungsgenauigkeit und einer zutreffenden Interpretation ergeben sich auch Herausforderungen im Zusammenhang mit der Nutzerakzeptanz. Kameras und Entfernungssensoren verursachen nicht nur zusätzliche Kosten bei der Anschaffung und Installation eines adaptiven Systems. Sie riskieren auch die Ablehnung des Gesamtsystems durch Nutzer, die sich nicht „beobachtet“ oder „überwacht“ fühlen möchten. Unabhängig von der Frage, ob Kontextinformationen direkt aus der Interaktion oder über externe Sensoren erfasst werden, führt diese Diskussion auch zu ethischen und datenschutzrechtlichen Fragestellungen. Schließlich erfordert die Individualisierung der ABS die Sammlung und Speicherung personenbezogener Daten. Bei ABS zur Überwindung von Nutzungsbarrieren geht es um sensible Daten, die Aufschluss über gesundheitliche und arbeitsrelevante Einschränkungen oder Qualifizierungslücken geben können. Die Transparenz der gespeicherten Daten gegenüber dem Nutzer gewinnt daher besondere Bedeutung. Dieser Aspekt ist teilweise durch die Offenheit des Nutzerprofils (vgl. Abschnitt 7.6) abgedeckt. Darüber hinaus wird es wichtig sein, Datenmissbrauch durch Dritte auszuschließen.

Letztendlich wird jedoch die Akzeptanz der Endnutzer über den Erfolg des Ansatzes entscheiden. Die grundsätzliche Frage, ob sich eingeschränkte Nutzer überhaupt mit ABS, die sich automatisch und kontinuierlich anpassen, anfreunden können, wird in dieser Arbeit nicht systematisch

⁴⁶ <http://www.myui.eu/>

behandelt. Dies liegt vor allem daran, dass sich diese Arbeit nur auf einen Teilausschnitt konzentriert, während Nutzer immer das Gesamtsystem betrachten werden. Die im MyUI-Projekt durchgeführten Validierungsstudien (vgl. Edlin-White, 2012 - interner Projektbericht) deuten insgesamt darauf hin, dass ältere Nutzer durchaus aufgeschlossen sind und automatische Adaptationen positiv bewerten, wenn sie sie als hilfreich empfinden.

11 Summary

This work presents a novel approach to user interfaces that automatically adapt to individual user needs, different environments and different interaction devices. The approach aims at improving the accessibility of interactive computer systems: Combined with a constant process of context recognition the here developed mechanisms of user interface generation and adaptations during the interaction can help to provide an increasingly better fit between a user's needs and the individualized interface. Users can always adjust the settings of the user interface. But the automatic adaptations provide individualized interfaces also to those users who have problems with configuration dialogues due to impairments or lower levels of ICT literacy. Moreover, automatic adaptations during use can overcome barriers of use directly when they occur. This will also help to compensate for altering user needs and capabilities which often appear in the course of ageing.

From a technical point of view, a modular approach is taken. All adaptive design aspects of the user interface are described as design patterns. Every aspect is covered by several alternative patterns in order to provide different solutions for different requirements. In the beginning of an interaction, a user interface is generated by composing the design patterns that are best suited for the current context requirements. For adaptations during the interaction, particular design patterns are substituted by other patterns. For the user to be able to understand and control the intelligent adaptations specific adaptation patterns are introduced.

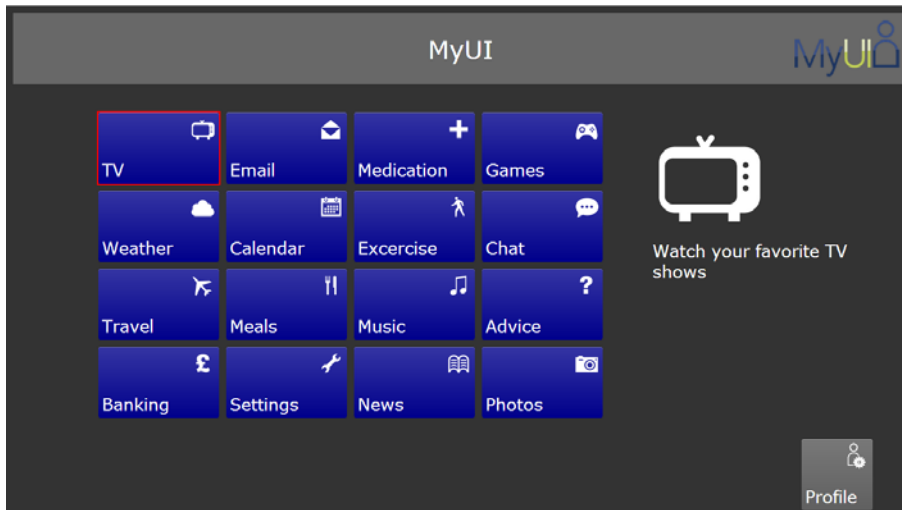
The approach is generic and can be used for any application to overcome diverse barriers of use by suitable adaptations. The required design knowledge is drawn from the design patterns. In a first implementation, only a basic set of patterns have been developed. With gradual extensions of the design patterns repository in the future, diverse requirements, impairments and interaction situations can be covered in a comprehensive manner. The design patterns repository is publicly accessible to engage as many external experts as possible in the further development of the design patterns.

The approach facilitates the development of adaptive user interfaces. Developers will create only an abstract model of the interaction which then serves as the basis for the automatic generation of the adaptive user interface. A new modelling format has been developed in order to support the industrial uptake of the approach. It aims at reducing the modelling effort and it builds on the well-established and comprehensible standard of state machine diagrams.

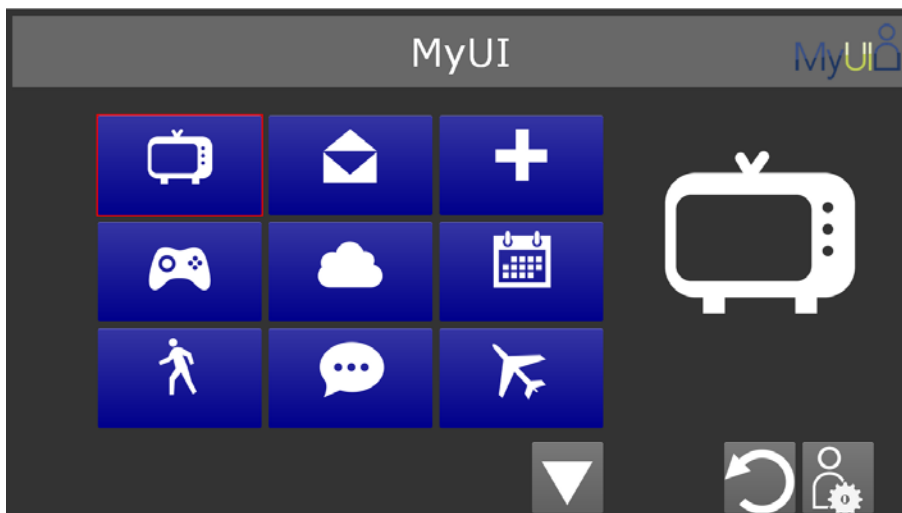
The evaluation of the approach is carried out for the example of web-based applications for interactive TV that adapt their user interfaces to impairments typically associated with ageing. The evaluation scheme includes a technical validation on the basis of a reference implementation and empirical studies with potential users and developers of adaptive user interfaces. The results indicate that the approach satisfies the requirements of adaptive user interfaces for improved accessibility and that the approach marks a significant progress beyond the current state of the art.

12 Anhang

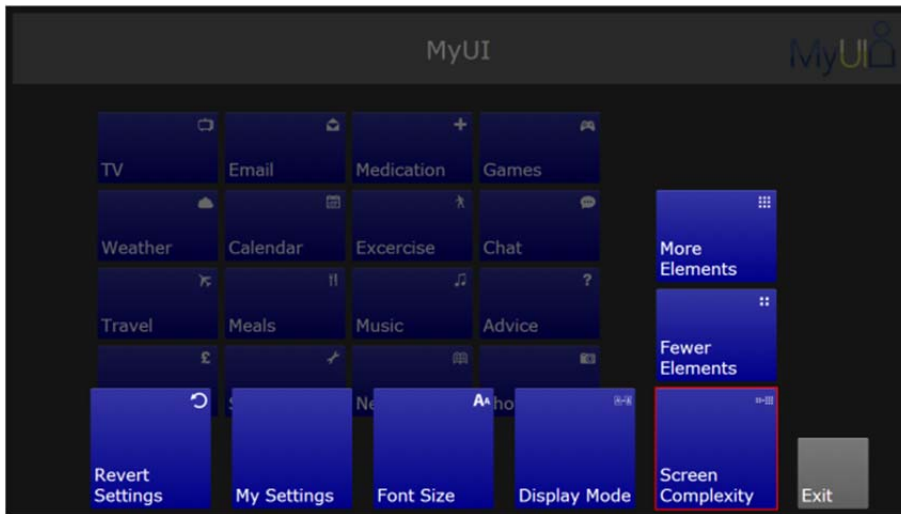
12.1 Bildschirmansichten der Referenzimplementierung



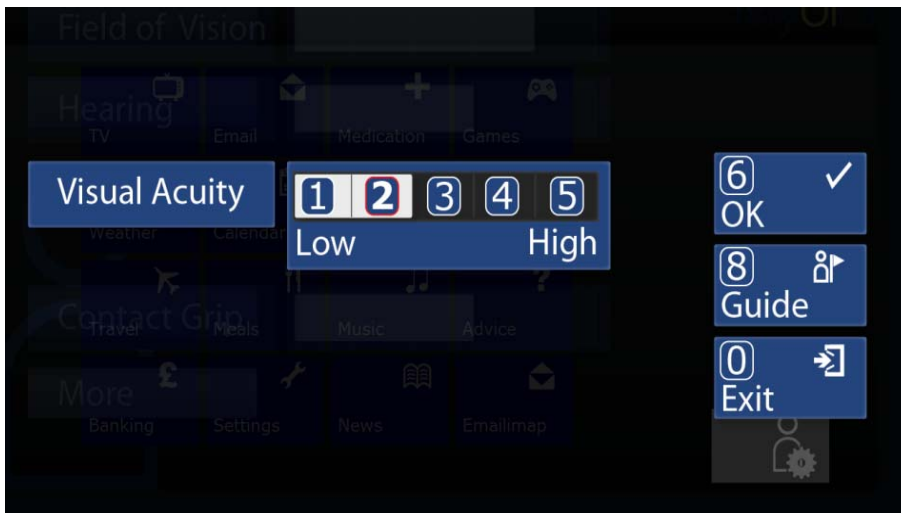
(1)
Hauptmenü:
Default-Version



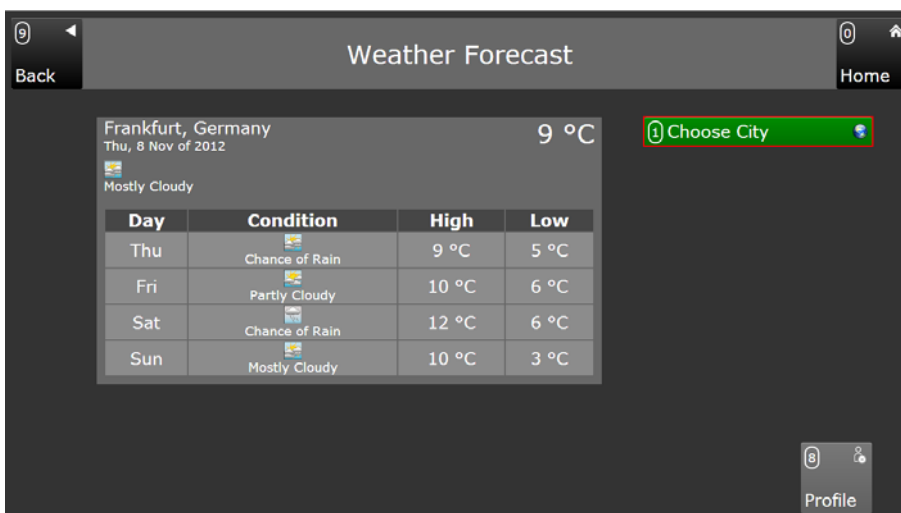
(2)
Hauptmenü:
reduziert, reiner Grafikstil



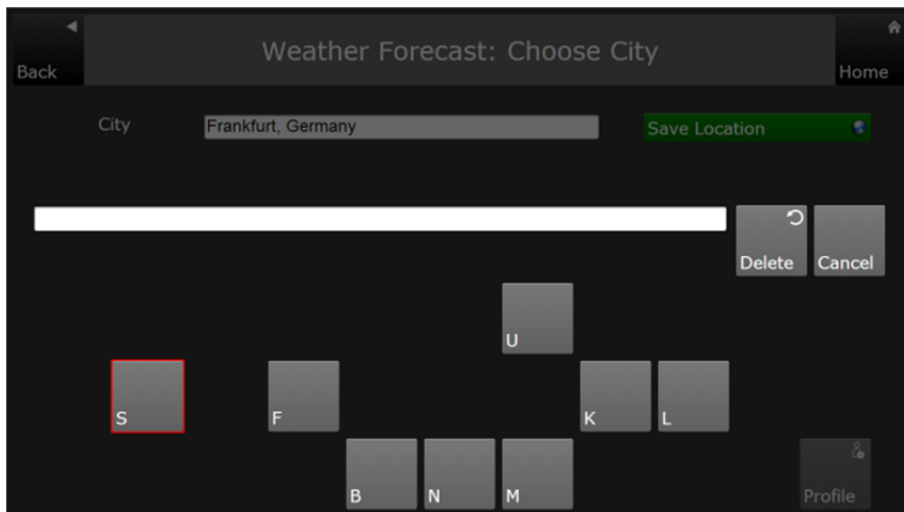
(3) Konfigurationsmenü



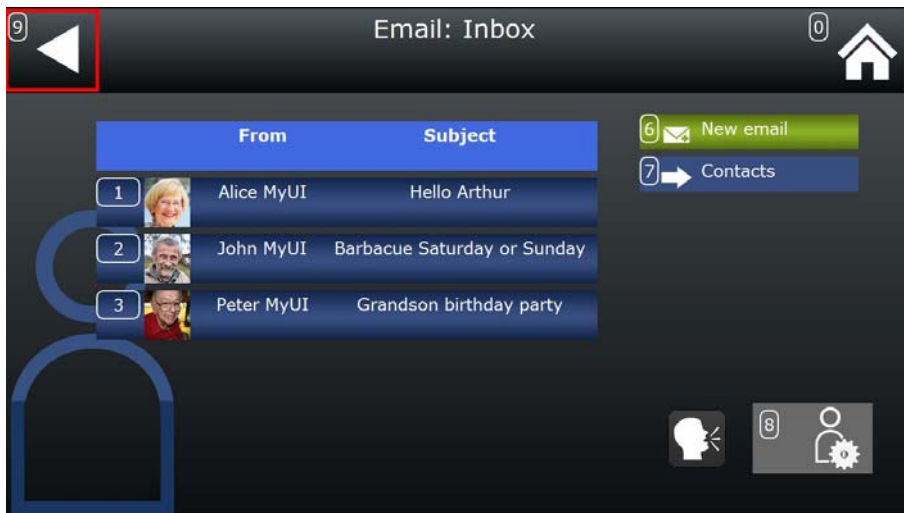
(4) Konfigurationsdialog: Einstellungsänderungen einer UP-Variable



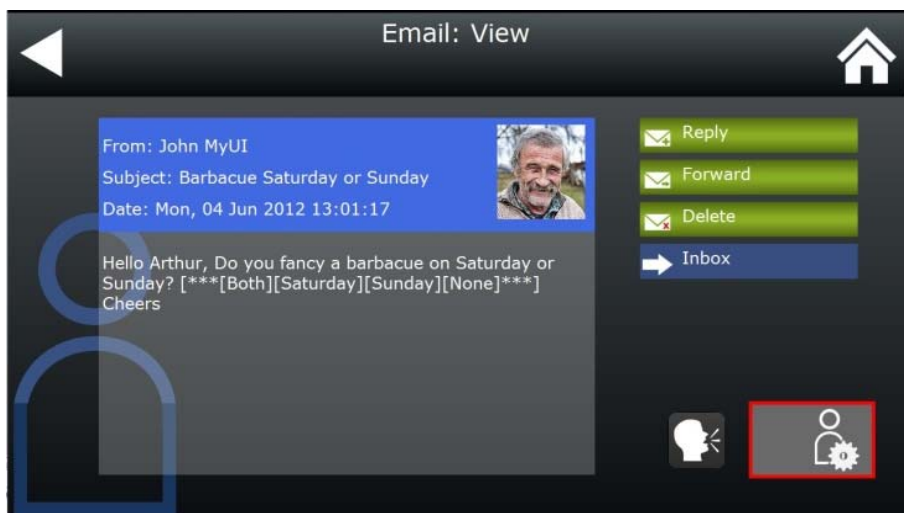
(5) Wetter-Applikation: Prognose



(6)
Wetter-Applikation:
Ortsauswahl



(7)
E-Mail-Applikation:
Posteingang



(8)
E-Mail-Applikation:
Detailansicht einer E-Mail

Abbildung 32 Bildschirmansichten der Referenzimplementierung

12.2 Überblick der im Rahmen der Referenzimplementierung erstellten Entwurfsmuster

| Gerätespezifische Muster | | | |
|--|---|--|---------------------------------------|
| Categorize iPhone | Available Microphone – FALSE | Available Keyboard - FALSE | Available Speaker – FALSE |
| Categorize iTV | Available Microphone - TRUE-internal | Available Keyboard - TRUE-QWERTY | Available Speaker - TRUE-headphones |
| Categorize unknown | Available Microphone - TRUE-external | Available Remote Control – FALSE | Available Speaker - TRUE-loudspeakers |
| Font Sizes iTV HD 720 px | Available Microphone - TRUE-headset | Available Remote Control – TRUE-standard | |
| Font Sizes iPhone | Available Touch – FALSE | Available Camera - FALSE | |
| Layout Grid – iTV | Available Touch – TRUE | Available Camera - TRUE | |
| Layout Grid - iPhone | Available Pointing Device - FALSE | Available Display - FALSE | |
| Interactive Element Size - iTV HD 720 px | Available Pointing Device - TRUE-mouse | Available Display - TRUE-42-1280:720 | |
| Interactive Element Size - iPhone | Available Pointing Device - TRUE-freehand | | |

| Individualisierungsmuster | | | |
|------------------------------|--|--------------------------------------|---------------------------------------|
| Font Size - Small (Default) | Audio Volume Low | Navigate and Select – Voice Input | Audio Speech Speed - medium (Default) |
| Font Size – Medium | Audio Volume Medium | Navigate and Select – no Voice Input | Audio Speech Speed – slow |
| Font Size – Large | Audio Volume High | Display Mode – mainly text (default) | Adaptation Rendering Time – Medium |
| Font Size - X Large | Navigate and Select– Touch Navigation | Display Mode – text and graphics | Adaptation Rendering Time - Slow |
| Layout Structure - 10-128-72 | Navigate and Select – Restricted Touch | Display Mode – mainly graphics | |
| Layout Structure - 10-64-96 | Navigate and Select – Highly Restricted Touch Navigation | Display Mode – graphics only | |
| LuminanceLow | Navigate and Select – No Touch Navigation | Display Mode – text only | |
| LuminanceMedium | Navigate and Select – Pointing Navigation | Screen Complexity – unconstrained | |
| LuminanceHigh | Navigate and Select – No Pointing Navigation | Screen Complexity – medium | |
| Confirmation Level – medium | Navigate and Select – RC Cursor Navigation | Screen Complexity – low | |
| Confirmation Level – high | Navigate and Select – RC Numeric Navigation | Screen Complexity – minimal | |

| Interaktionsmuster | | | |
|--|----------------------------|---|---|
| Title Bar – iTV | Main Menu – 4x4-Tile Style | Settings Menu - Vertically aligned – Grouped | List w/ Attributes - sequential - limited rows |
| Message Box for Confirmation - Default | Main Menu – 3x3-Tile Style | Settings Menu - Vertically aligned - Cover Flow | List w/ Attributes - sequential, 1st dominant, limited rows |
| Functions Menu - Group of Buttons | Main Menu – 2x2-Tile Style | Settings Menu - Horizontally aligned | List with Attributes - parallel style |
| Functions Menu - Numeric Navigation | Main Menu – List Style | Item with Attributes – Default | |
| Navigation Menu - Group of Buttons | Form – Default | Item with Attributes - Small Displays | |

| Bedien- und Anzeigeelemente | | | |
|-----------------------------|-------------------------|----------------|-------------------|
| Button | Action Button | DisplayLabel | Audio Control |
| Title Navigation Button | Drop-down Action Button | DisplayIcon | Cursor Navigation |
| Content List Button | Bar Button | DisplayPicture | |
| Content Menu Button | TextField | DisplayValue | |
| Content Control Button | Display Element | Preview | |

| Adaptations-dialogmuster | | |
|---|---|--|
| Automatic Adaptation with Implicit Confirmation | Explicit Confirmation before Adaptation | Explicit Confirmation after Adaptation |

| Adaptations-ausführungsmuster | | | |
|---------------------------------|-----------------------|---------------------|--------------------|
| Main Menu Adaptation | Adapt Widget Size | Split Content | Adapt Speech Input |
| Functions Menu Adaptation | Adapt Widget Position | Adapt New Content | |
| List with Attributes Adaptation | Adapt Widget Content | Adapt Tonal Output | |
| Adapt Font Size | Adapt Widget | Adapt Speech Output | |

Abbildung 33 Überblick der für die Referenzimplementierung dokumentierten Entwurfsmuster. Entwurfsmuster innerhalb eines Bündels sind nicht durch Linien getrennt. Die Entwurfsmusterbeschreibungen finden sich unter <http://myui-patterns.iao.fraunhofer.de>.

12.3 Beispiele für konkrete Entwurfsmusterbeschreibungen

| Interaction pattern | Main Menu – 4x4 Tile Style (iTV Default) |
|------------------------------|---|
| <p>Problem</p> | <p>Main Menu provides access to all available services/applications and might display important status information for some services (e.g. indicate new message for the email service). Note: The displayed items do not have a natural order. Order could be changed by the user or could be subject of adaptations (based on frequency of use).</p> <p>Some explanations to the Context:</p> <ul style="list-style-type: none"> • numeric navigation is relevant, because 16 keys are not navigable via numeric navigation, maximum there is 10. • we check the ratio of font-size and cell-size to find out, how much elements can be displayed <p>Interaction Situation = SelectService</p> |
| <p>Context</p> | <p>IF</p> <p>// check, whether the needed buttons size is small enough to be displayable: $\text{maxButtonSize}(\text{deviceCategory}, \text{cellSize}, \text{contentArea}, 4, 4) \geq \text{minButtonSize}(\text{bodyTextFontSize}, \text{complementaryTextFontSize}, \text{displayMode}, \text{interactiveElementMinimumSize})$</p> <p>// check, whether that number of elements may be displayed on screen AND $\text{remainingElementsPerScreen} \geq 16$ AND $\text{numericInput} = \text{off}$</p> <p>$\text{minButtonSize}$ calculates the minimal button size necessary for displaying all the information and enabling interaction, using bodyTextFontSize, $\text{complementaryTextFontSize}$, displayMode, $\text{interactiveElementMinimumSize}$.</p> <p>$\text{maxButtonSize}$ calculates the maximal possible button size using deviceCategory, cellSize, contentArea, and the number of buttons in a row and column.</p> <p>The pattern can be activated, only if the minimal necessary button size is equal or larger than the maximal possible button size.</p> |
| <p>Solution</p> | <p>THEN</p> <ul style="list-style-type: none"> • Fill the content area with 16 buttons in a 4x4 manner. Requires contentArea • Parametrize each button according to the information of the user profile. Requires Content Menu Button. • For displaying each button, requires displayMode, bodyTextFontSize, $\text{complementaryTextFontSize}$, titleFontSize. • For interaction with the buttons, requires cursorInput, pointingInput, touchInput. • Speech abilities are inherited from Audio Control • Use Content Control Button if there are more than 16 elements. • Show a preview of the currently highlighted option within the Preview. |
| <p>Code Reference</p> | <p>// calculate buttonSize using deviceCategory, cellSize, contentArea, bodyTextFontSize, $\text{complementaryTextFontSize}$, displayMode, $\text{interactiveElementMinimumSize}$</p> <pre> FOR (i=0; i<4; i++) FOR (j=0; j<4; j++) create new Content Menu Button coordinates of Menu Button are i * buttonSize.x and j * buttonSize.y tell menuButton bodyTextFontSize, $\text{complementaryTextFontSize}$, displayMode, $\text{interactiveElementMinimumSize}$ $\text{remainingElementsPerScreen} = \text{remainingElementsPerScreen} - 16$ </pre> |

| | |
|-------------------------------|---|
| | set <i>currentState</i> . |
| Diagram | <p>The diagram shows a 'Main Menu' interface with a 4x4 grid of blue tiles. Each tile contains a white icon and a text label. The tiles are: Row 1: TV (TV icon), Email (envelope icon), Medication (plus icon), News (book icon); Row 2: Weather (cloud icon), Calendar (calendar icon), Exercise (person icon), Games (game controller icon); Row 3: Chat (speech bubble icon), Travel (airplane icon), Meals (fork and knife icon), Photos (camera icon); Row 4: Music (musical note icon), Advice (question mark icon), Banking (pound sign icon), Settings (wrench icon). To the right of the grid is a video preview showing an elderly woman speaking, with the text 'AUW' and 'TEMA' visible in the background. The top right corner of the menu has the 'MyUI' logo.</p> |
| Rationale (references) | |
| Ranking | |
| Pattern Bundle | Main Menu |
| Substitutes | Main Menu – 3x3-Tile Style Main Menu – 2x2-Tile Style Main Menu – List Style |
| Requires | Content Menu Button Content Control Button Audio Control Preview |
| Uses <variable> | <i>deviceCategory</i> <i>cellSize</i> <i>contentArea</i> <i>minButtonSize</i> ?? <i>bodyTextFontSize</i> <i>complementaryTextFontSize</i> <i>displayMode</i> <i>interactiveElementMinimumSize</i> <i>remainingElementsPerScreen</i> <i>numericInput</i> |

Abbildung 34 Beschreibung des Interaktionsmusters „Main Menu – 4x4 Tile Style (iTV Default)“

| Interaction pattern | Main Menu – 3x3 Tile Style |
|-----------------------|---|
| Problem | <p>Main Menu provides access to all available services/applications and might display important status information for some services (e.g. indicate new message for the email service). Note: The displayed items do not have a natural order. Order could be changed by the user or could be subject of adaptations (based on frequency of use).</p> <p>A reduction to not more than 9 button will be necessary for certain users, e.g. if:</p> <ul style="list-style-type: none"> • the buttons have to be big, e.g. due to big font sizes or big sizes of interactive elements • numeric navigation is on, therefore only few elements can be displayed • the device is too small to display a larger amount of buttons <p>Interaction Situation = SelectService</p> |
| Context | <p>IF // check, whether the needed buttons size is small enough to be displayable:</p> <p>$\text{maxButtonSize}(\text{deviceCategory}, \text{cellSize}, \text{contentArea}, 3, 3) \geq \text{minButtonSize}(\text{bodyTextFontSize}, \text{complementaryTextFontSize}, \text{displayMode}, \text{interactiveElementMinimumSize})$</p> <p>AND // must allow at least 9 menu buttons</p> <p>$\text{remainingElementsPerScreen} \geq 9$</p> |
| Solution | <p>THEN</p> <ul style="list-style-type: none"> • Fill the content area with 9 buttons in a 3x3 manner. Requires <i>contentArea</i> • Parametrize each button according to the information of the user profile. Requires Content Menu Button • For displaying each button, requires <i>displayMode</i>, <i>bodyTextFontSize</i>, <i>complementaryTextFontSize</i>, <i>titleFontSize</i> • For interaction with the buttons, requires <i>cursorInput</i>, <i>numericInput</i>, <i>pointingInput</i>, <i>touchInput</i> • Audio abilities are inherited from Audio Control • Use Content Control Button if there are more than 9 elements • Show a preview of the currently highlighted option within the preview window |
| Code Reference | <pre>// Button calculations IF DisplayMode = "graphics only" THEN recalculate icon to a new, very much taller size using BodyTextFontSize tell menuButton: icon = parameter.icon ELSE IF DisplayMode = "mainly graphics" THEN recalculate icon to a new, taller size using BodyTextFontSize tell menuButton: <i>fontSize</i> = ComplementaryTextFontSize tell menuButton: <i>icon</i> = parameter.icon tell menuButton: <i>label</i> = parameter.label ELSE IF DisplayMode = "text and graphics" THEN recalculate icon to a new, slightly taller size using BodyTextFontSize tell menuButton: <i>fontSize</i> = BodyTextFontSize tell menuButton: <i>icon</i> = parameter.icon</pre> |

```

tell menuButton: label = parameter.label
ELSE IF
DisplayMode = "text only"
THEN
tell menuButton: fontSize = TitleFontSize
tell menuButton: label = parameter.label
ELSE // should be mainly text
tell menuButton: fontSize = BodyTextFontSize
tell menuButton: icon = parameter.icon
tell menuButton: label = parameter.label
IF
NumericInput = "on"
THEN
tell menuButton: number = currentNumber
.
tell menuButton: cursorInput = CursorInput
}

// Content Area: 4x4 Menu Button slots in the Content Area with the following grid
coordinates:

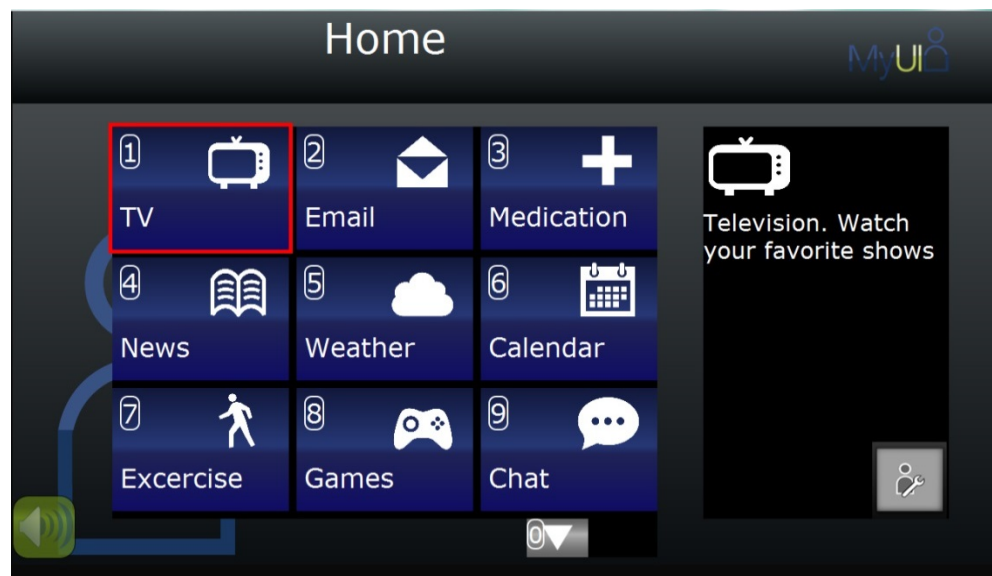
use ContentArea
(1) a to b
create Content Menu Button menuButton using (a to b);
do Button calculations
(2) ...
...
(9) ...

remainingElementsPerScreen = remainingElementsPerScreen - 9

set currentState.

```

Diagram



On the iTV

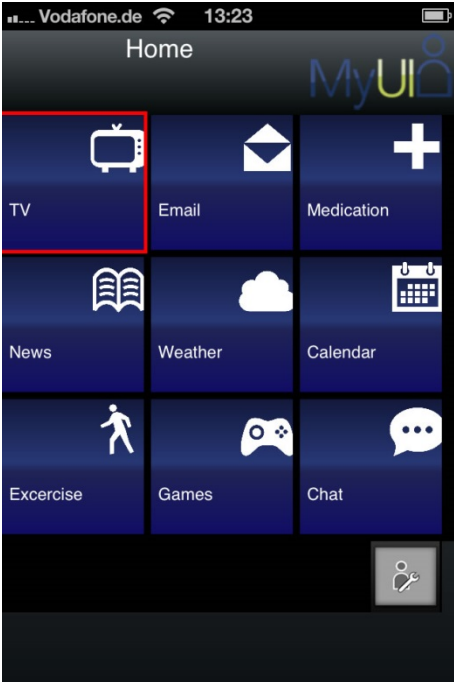
| | |
|-------------------------------|---|
| |  <p>On the iPhone</p> |
| Rationale (references) | |
| Ranking | |
| Pattern Bundle | Main Menu |
| Substitutes | Main Menu – 4x4-Tile Style Main Menu – 2x2-Tile Style Main Menu – List Style |
| Requires | Content Menu Button Content Control Button Audio Control Preview |
| Uses <variable> | <i>deviceCategory</i> <i>cellSize</i> <i>contentArea</i> <i>minButtonSize</i> <i>bodyTextFontSize</i> <i>complementaryTextFontSize</i> <i>displayMode</i> <i>interactiveElementMinimumSize</i> <i>remainingElementsPerScreen</i> <i>numericInput</i> |

Abbildung 35 Beschreibung des Interaktionsmusters „Main Menu – 3x3 Tile Style“

Display Mode

| Individualization Pattern Bundle | Display Mode |
|--|--|
| Problem | These patterns decide about the ratio between textual information and graphical information |
| State | 1 |
| sets <variable(s)> as used by | sets <i>displayMode</i> |
| Patterns | Display Mode – mainly text (default) Display Mode – text and graphics Display Mode – mainly graphics Display Mode – graphics only Display Mode – text only |

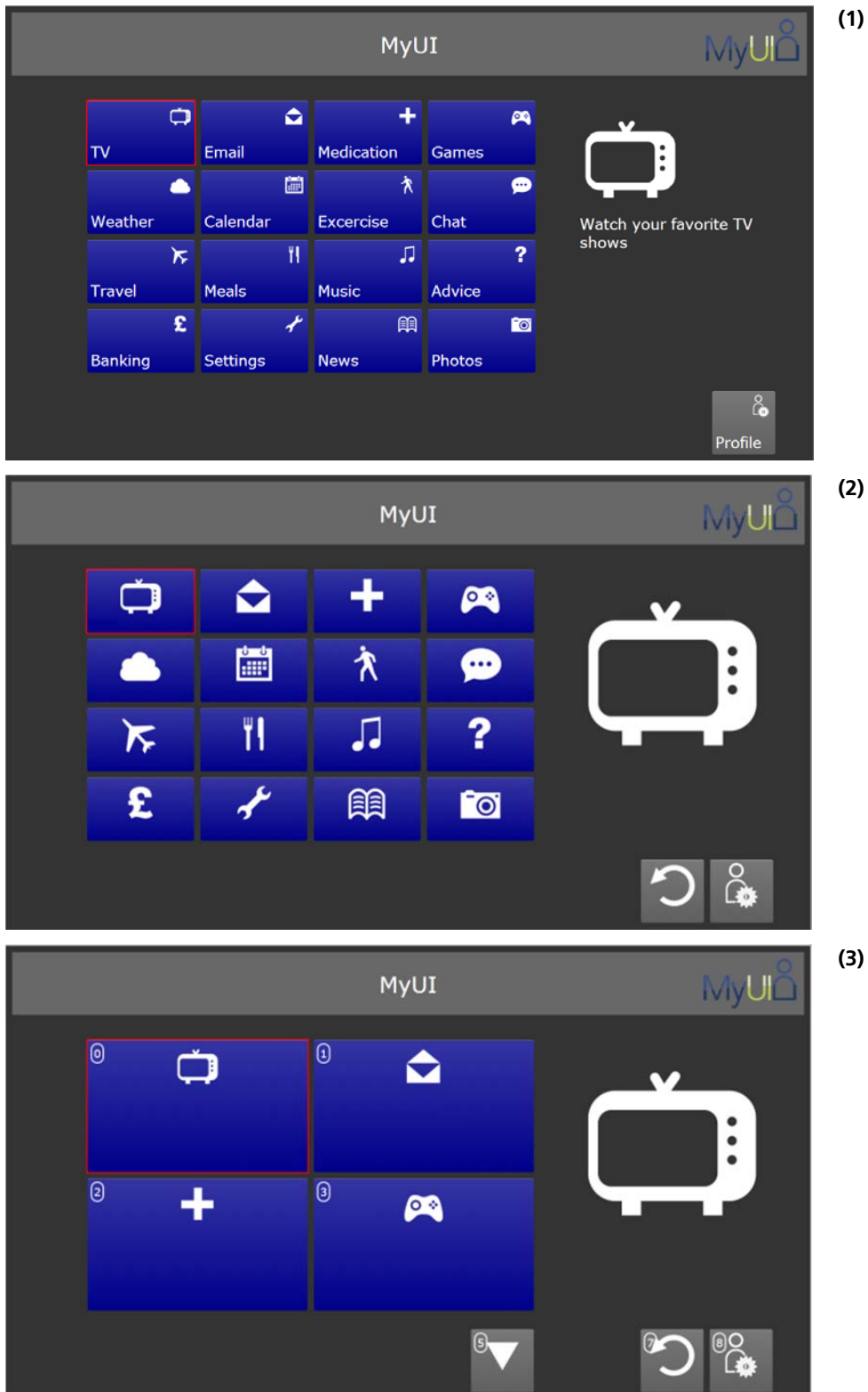
Context

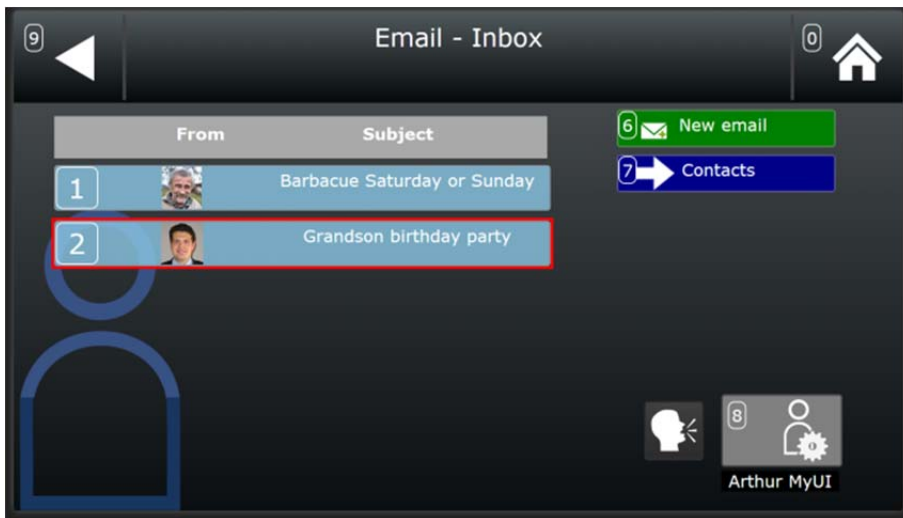
|understandingAbstractSigns | | | | | |
|---------------------------------|---|-----|-----|-----|-----|
| | 0 | 1 | 2 | 3 | 4 |
| languageReception | 0 | MT | MT | TO | TO |
| | 1 | TAG | TAG | MT | MT |
| | 2 | MG | MG | TAG | TAG |
| | 3 | GO | GO | TAG | TAG |
| | 4 | GO | GO | TAG | TAG |

| | tonly | Mtext | t&g | mgraphics | gonly |
|-----------------------------------|-------|-------|-----|-----------|-------|
| languageReception | 0 | + | | | |
| | 1 | | + | | |
| | 2 | | | + | |
| | 3 | | | | + |
| | 4 | | | | + |
| understandingAbstractSigns | 0 | + | | | |
| | 1 | + | | | |
| | 2 | + | | | |
| | 3 | + | | | |
| | 4 | + | | | |

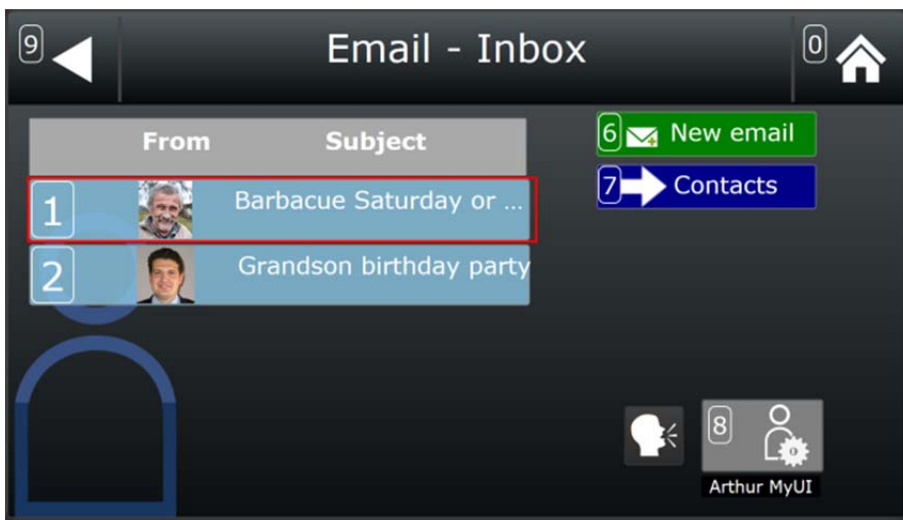
Abbildung 36 Beschreibung des Individualisierungsmusterbündels „DISPLAY MODE“ – kopiert aus dem MyUI-Entwurfsmusterrepositorium unter http://myui-patterns.iao.fraunhofer.de/index.php/Display_Mode. Das zusätzliche Feld „State“ wurde von den Entwicklern während der Implementierung eingefügt, um den Stand der inhaltlichen und technischen Realisierung zu kennzeichnen.

12.4 Bildschirmansichten des Szenarios zur Evaluierung der automatischen Adaptivität

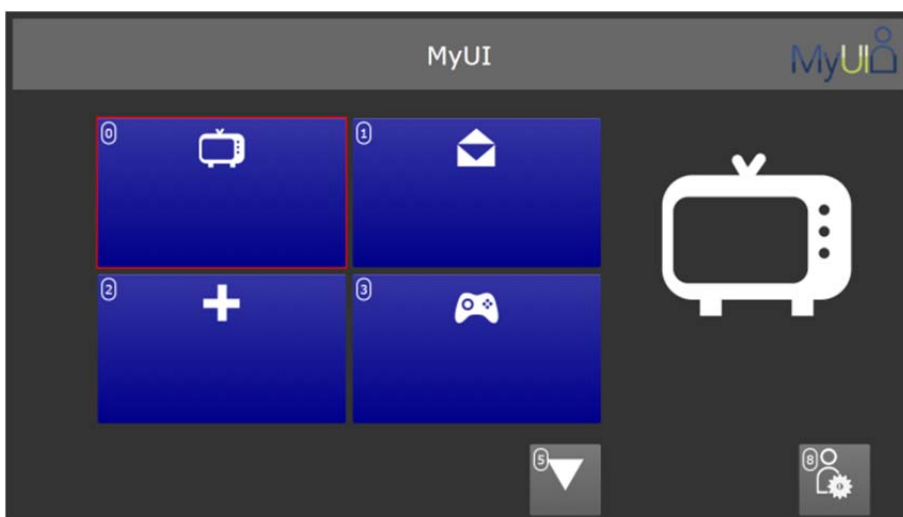




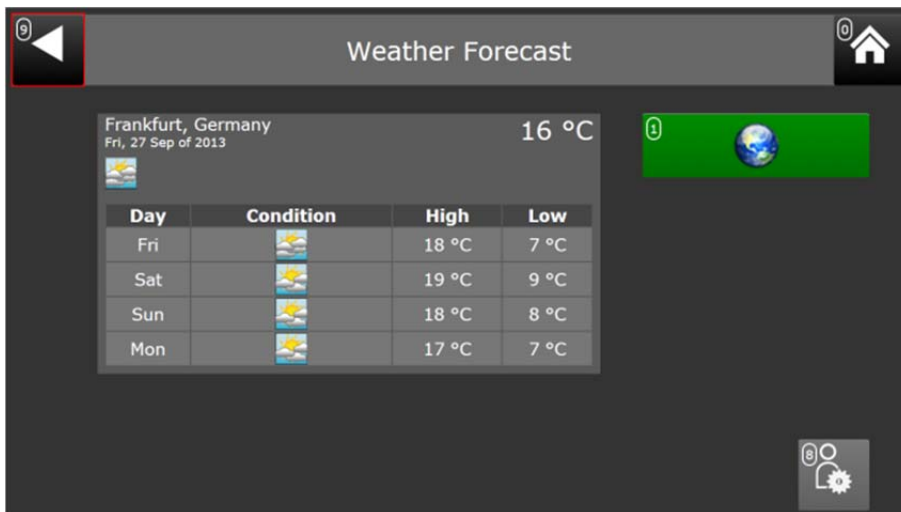
(4)



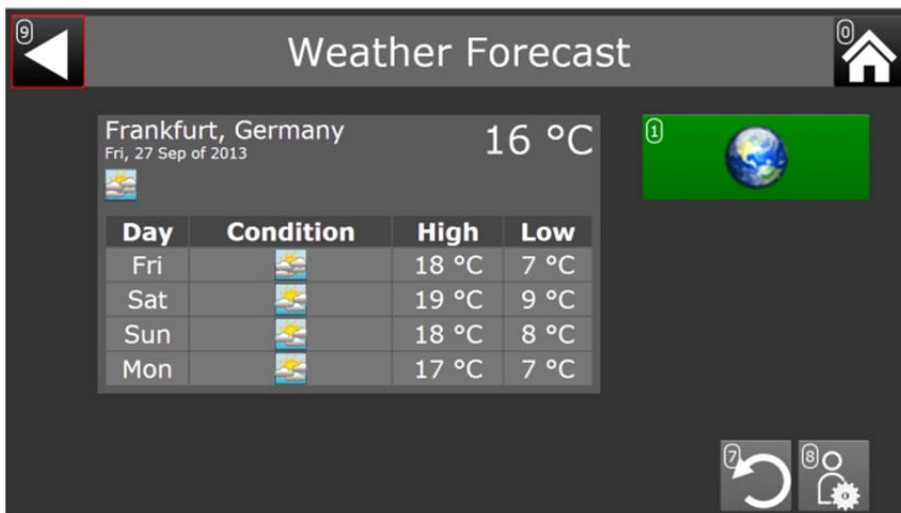
(5)



(6)



(7)



(8)



(9)

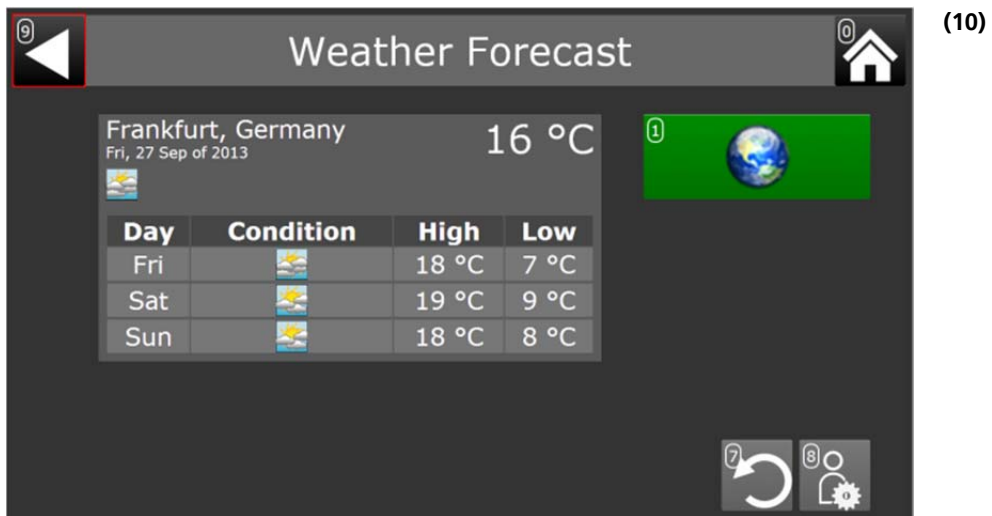
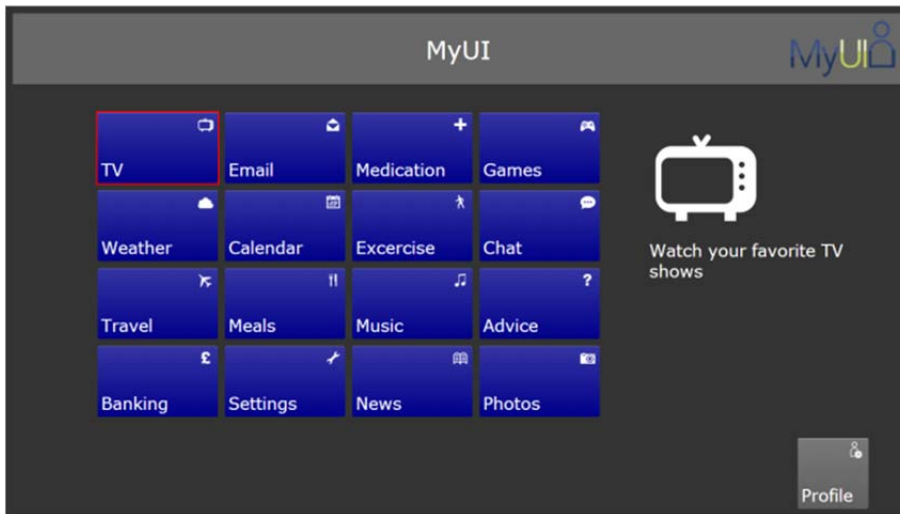
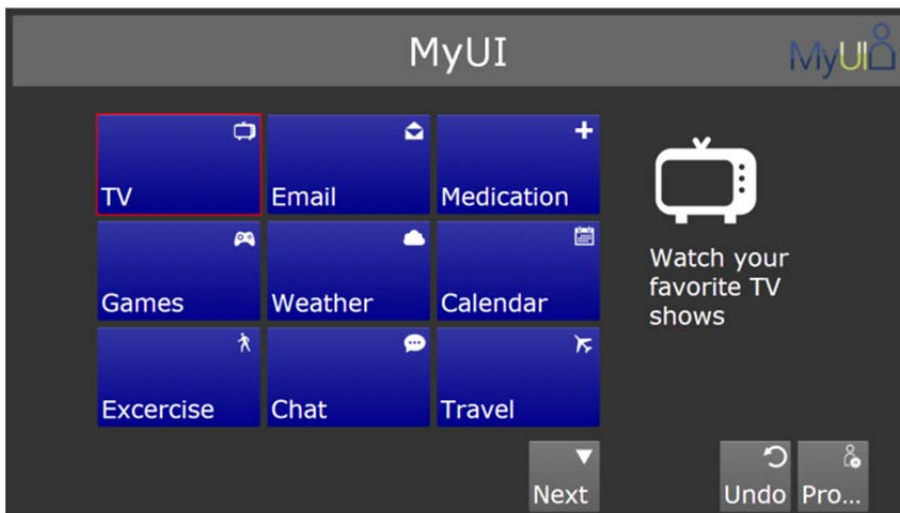


Abbildung 37 Bildschirmansichten des Testszenarios zur Evaluierung der automatischen Adaptivität (Beschreibung in Abschnitt 9.2.1.2)

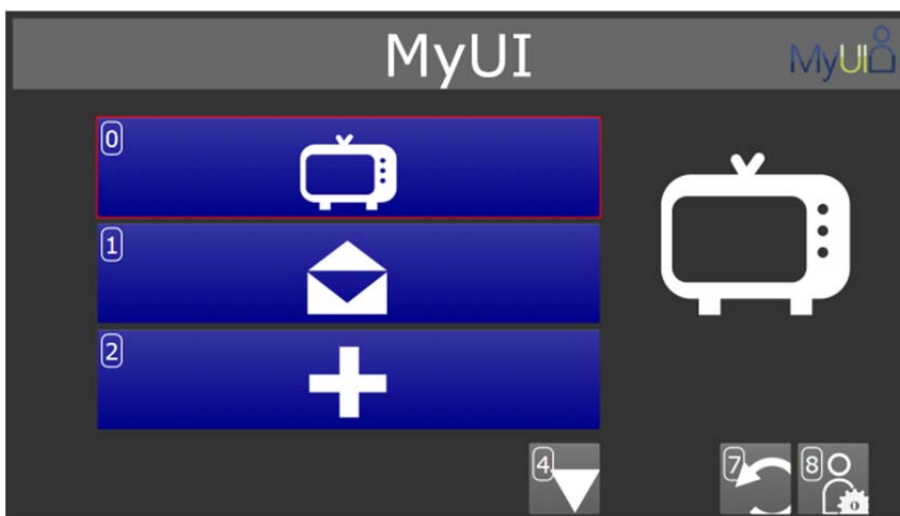
12.5 Bildschirmansichten der Szenarien zur Evaluierung der kontinuierlichen Anpassungsfähigkeit während der Nutzung



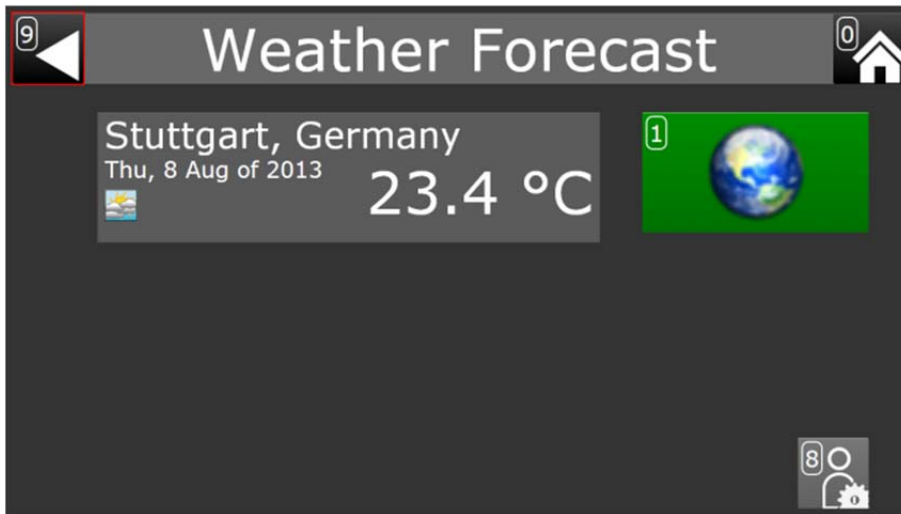
(1)
Ausgangszustand des Anpassungsszenarios 1



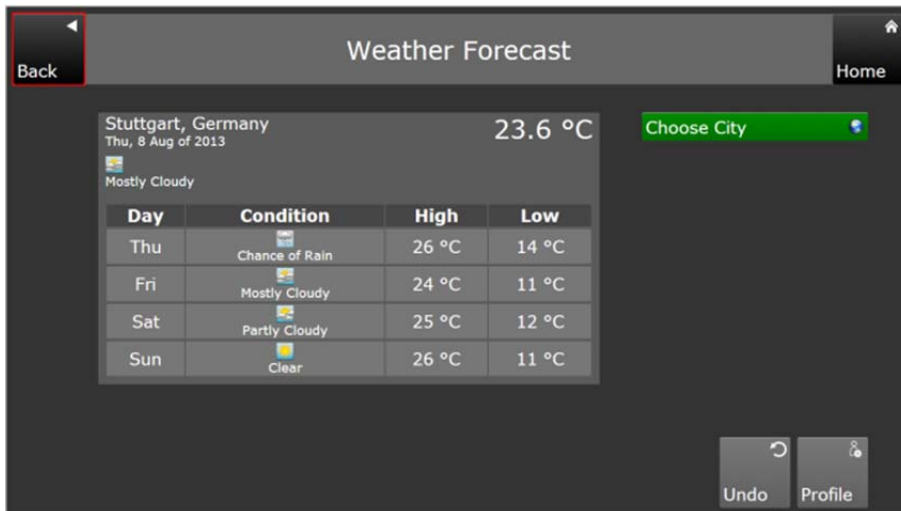
(2)
Zielzustand des Anpassungsszenarios 1 und Ausgangszustand des Anpassungsszenarios 2



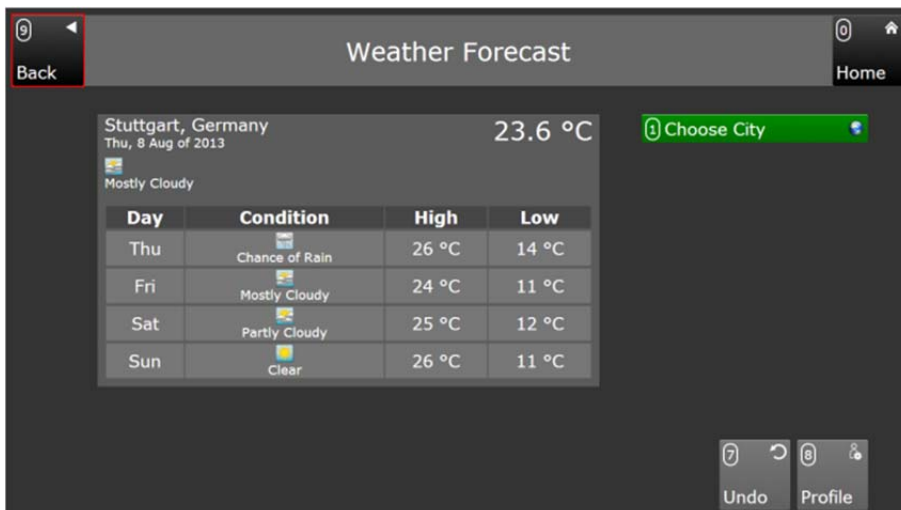
(3)
Zielzustand des Anpassungsszenarios 2



(4)
Ausgangszustand des
Anpassungsszenarios 3



(5)
Zielzustand des
Anpassungsszenarios 3 und
Ausgangszustand des
Anpassungsszenarios 4



(6)
Zielzustand des
Anpassungsszenarios 4

Abbildung 38 Bildschirmansichten der Testszenarien zur Evaluierung der kontinuierlichen Anpassungsfähigkeit (Beschreibung in Abschnitt 9.2.3.2)

13 Literatur

- Aarts, E., & Encarnação, J. (2006). Into Ambient Intelligence. In E. Aarts & J. Encarnacao (Hrsg.): True Visions, The Emergence of Ambient Intelligence. Berlin, Heidelberg: Springer-Verlag. S. 1-16.
- Abascal, J., Aizpurua, A., Cearreta, I., Gamecho, B., Garay, N., & Miñón, R. (2011). A modular approach to user interface adaptation for people with disabilities in ubiquitous environments. Internal Technical Report No. EHU-KAT-IK-01-11 (2011), <http://sipt07.si.ehu.es/icearreta/reports/AModularApproach.pdf>
- Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I. & Angel, S. (1977). A Pattern Language: Towns, Buildings, Construction. New York: Oxford University Press.
- Ali, M.F., Pérez-Quiñones, M.A., Abrams, M. & Shell, E. (2002). Building Multi-Platform User Interfaces with UIML. In Proceedings of the 2nd International Conference on Computer-Aided Design of User Interfaces (CADUI 2002). Berlin, Heidelberg: Springer-Verlag. S. 255-266.
- Balme, L., Demeure, A., Barralon, N., Coutaz, J. & Calvary, G. (2004). CAMELEON-RT: A Software Architecture Reference Model for Distributed, Migratable, and Plastic User Interfaces. In P. Markopoulos, B. Eggen & E. Aarts (Hrsg). Proceedings of 2nd European Symposium on Ambient Intelligent (EUSAI'2004). LNCS 3295. Heidelberg: Springer-Verlag. S. 291-302.
- Balzert, H. (1996). Lehrbuch der Software-Technik: Software-Entwicklung. Heidelberg, Berlin, Oxford: Spektrum Akademischer Verlag.
- Barnett, J., Akolkar, R., Auburn, R.J., Bodell, M., Burnett, D.C., Carter, J., McGlashan, S., Lager, T., Helbing, M., Hosn, R., Raman, T.V., Reifenrath, K., Rosenthal, N. & Roxendal, J. (Hrsg.) (2012). State Chart XML (SCXML): State Machine Notation for Control Abstraction. W3C Working Draft 6 December 2012. <http://www.w3.org/TR/scxml/>
- Biswas, P., Langdon, P., Duarte, C. & Coelho, J. (2011). Multimodal adaptation through simulation for digital TV interface. In Proceedings of the 9th international interactive conference on Interactive television (EuroITV '11). New York: ACM. S. 231-234.
- BITKOM (Hrsg.) (2013). Die Zukunft der Consumer Electronics – 2013. Berlin: BITKOM.
- Booch, G., Maksimchuk, R., Engle, M., Young, B., Conallen, J. & Houston, K. (2007). Object-Oriented Analysis and Design with Applications (3. Auflage). Reading, MA: Addison-Wesley Professional.
- Borchers, J.O. (2000). CHI meets PLoP: An interaction patterns workshop. SIGCHI Bulletin, 32(1). S. 9-12.
- Borchers, J. O. (2001). A pattern approach to interactive design. Chichester: John Wiley & Sons Ltd.
- Bortz, J., Lienert, G.A: & Boehnke, K. (2000). Verteilungsfreie Methoden in der Biostatistik (2. Auflage). Berlin, Heidelberg: Springer-Verlag.

- Bruikman, H., van de Korput, R., Hernández, J.A., Häbe, D. & Sellner, T. (2013). MyUI E-mail on iTV prototype. Public deliverable of the MyUI project.
http://www.myui.eu/deliverables/MyUI_D4-3_final.pdf
- Brusilovsky, P. & Maybury, M.T. (2002): From Adaptive Hypermedia To The Adaptive Web, In *Communications of the ACM*, 45 (5). S. 31–33.
- Bunt, A., Conati, C. & McGrenere, J. (2007). Supporting interface customization using a mixed-initiative approach. In *Proceedings of the 12th international conference on Intelligent user interfaces (IUI '07)*. New York: ACM. S. 92-101.
- Bunt, A., Conati, C. & McGrenere, J. (2009). Mixed-Initiative Interface Personalization as a Case Study in Usable AI. *AI Magazine*, 30 (4). S. 58-64.
- Caldwell, B., Cooper, M., Reid, L.G. & Vanderheiden, G. (Hrsg.) (2008). *Web Content Accessibility Guidelines (WCAG) 2.0*. W3C Recommendation 11 December 2008.
<http://www.w3.org/TR/WCAG20/>.
- Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L. & Vanderdonckt, J. (2003). A Unifying Reference Framework for Multi-Target User Interfaces. *Interacting with Computers*, 15 (3). S. 289–308.
- Calvary, G., Coutaz, J., Dâassi, O., Balme, L. & Demeure, A. (2004). Towards a new generation of widgets for supporting software plasticity: The “comet”. In R. Bastide, P. Palanque & J. Roth (Hrsg.): *Proceedings of the 2004 international conference on Engineering Human Computer Interaction and Interactive Systems (EHCI-DSVIS'04)*. Berlin, Heidelberg: Springer-Verlag. S. 306-324.
- Calvary, G., Serna, A., Coutaz, J., Scapin, D., Pontico, F., & Winckler, M. (2011). Envisioning Advanced User Interfaces for e-Government Applications: a Case Study. In S. Assar, I. Boughzala, & I. Boydens (Hrsg.): *Practical Studies in E-Government: Best Practices from Around the World*. Berlin, Heidelberg: Springer-Verlag. S. 205-228.
- Card, S.K., Newell, A. & Moran, T.P. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates Inc.
- Ceret, C. (2011). Toward a flexible design method sustaining UIs plasticity. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '11)*. New York: ACM. S. 307-310.
- Cockton, G. (1987). Some critical remarks on abstractions for adaptable dialogue managers. In D. Diaper & R. Winder (Hrsg.): *Proceedings of the 3rd Conference of the British Computer Society on People and Computers*. Cambridge: Cambridge University Press. S. 325-343.
- Coelho, J. & Duarte, C. (2011). The Contribution of Multimodal Adaptation Techniques to the GUIDE Interface. In C. Stephanidis (Hrsg.): *Proceedings of the 6th international conference on Universal access in human-computer interaction: Design for All and eInclusion (UAHCI'11)*. Berlin, Heidelberg: Springer-Verlag. S. 337-346.
- Coelho, J., Duarte, C., Biswas, P., & Langdon, P. (2011). Developing accessible TV applications. In *Proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility (ASSETS '11)*. New York: ACM. S. 131-138.

- Collignon, B. Vanderdonckt, J. & Calvary, G. (2008). Model-Driven Engineering of Multi-target Plastic User Interfaces. In Proceedings of the Fourth International Conference on Autonomic and Autonomous Systems (ICAS '08). Washington DC: IEEE Computer Society. S. 7-14.
- Coutaz, J. (2006). Meta-user interfaces for ambient spaces. In K. Coninx, K. Luyten & K.A. Schneider (Hrsg.): Proceedings of the 5th international conference on Task models and diagrams for users interface design (TAMODIA'06). Berlin, Heidelberg: Springer-Verlag. S. 1-15.
- Coutaz, J. (2010). User interface plasticity: model driven engineering to the limit! In Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '10). New York: ACM. S. 1-8.
- Coutaz, J., Balme, L., Alvaro, X., Calvary, G., Demeure, A. & Sottet, J.-S. (2007). An MDE-SOA approach to support plastic user interfaces in ambient spaces. In C. Stephanidis (Hrsg.): Proceedings of the 4th international conference on Universal access in human-computer interaction: ambient interaction (UAHCI'07). Berlin, Heidelberg: Springer-Verlag. S. 63-72.
- Demeure, A., Calvary, G., Coutaz, J. & Vanderdonckt, J. (2006). The Comets Inspector, Manipulating Multiple Interface Representations Simultaneously. In Proceedings of the 6th International Conference on Computer-Aided Design of User Interfaces (CADUI'06). Berlin, Heidelberg: Springer-Verlag. S. 167-174.
- Demeure, A., Calvary, G., Coutaz, J. & Vanderdonckt, J. (2007): The COMETs Inspector: Towards Run Time Plasticity Control Based on a Sematic Network. In K. Coninx, K. Luyten, K.A. Schneider (Hrsg.): Task Models and Diagrams for Users Interface Design, 5th International Workshop (TAMODIA 2006). LNCS 4385. Berlin, Heidelberg: Springer-Verlag. S. 324-338.
- Demeure, A. Calvary, G. & Coninx, K. (2008). COMET(s), A Software Architecture Style and an Interactors Toolkit for Plastic User Interfaces. In T.C. Graham & P. Palanque (Hrsg.). Interactive Systems. Design, Specification, and Verification. LNCS 5136. Berlin, Heidelberg: Springer-Verlag. S. 225-237.
- Dessart, C.-E., Motti, V. G. & Vanderdonckt, J. (2011). Showing user interface adaptivity by animated transitions. In Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '11). New York: ACM, S. 95-104.
- Deutsches Institut für Medizinische Dokumentation und Information, DIMDI (2004). Internationale Klassifikation der Funktionsfähigkeit, Behinderung und Gesundheit. Die englischsprachige Originalausgabe wurde 2001 von der Weltgesundheitsorganisation veröffentlicht als International Classification of Functioning, Disability and Health.
- Dieterich, H., Malinowski, U., Kühme, T. & Schneider-Hufschmidt, M. (1993). State of the Art in Adaptive User Interfaces. In M. Schneider-Hufschmidt, T. Kühme & U. Malinowski (Hrsg.): Adaptive User Interfaces: Principles and practice. Amsterdam: North-Holland. S. 13-48.
- Duarte, C. & Carriço, L (2006). A conceptual framework for developing adaptive multimodal applications. In Proceedings of the 11th international conference on Intelligent user interfaces (IUI '06). New York: ACM. S. 132-139.
- Edlin-White, R. (2012). Results of end-user participation and testing. Confidential deliverable of the MyUI project.

- Eugster, P.T., Felber, P.A., Guerraoui, R. & Kermarrec, A.-M. (2003). The many faces of publish/subscribe. *ACM Computing Surveys*, 35 (2). S. 114-131.
- Findlater, L. & McGrenere, J. (2004) A comparison of static, adaptive, and adaptable menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'04)*, New York: ACM. S. 89-96.
- Findlater, L. & Gajos, K.Z. (2009). Design Space and Evaluation Challenges of Adaptive Graphical User Interfaces. *AI Magazine*, 30 (4): S. 68-73.
- Fischer, G. (1993). Shared knowledge in Cooperative Problem-Solving Systems – Integrating Adaptive and Adaptable Components. In M. Schneider-Hufschmidt, T. Kühme & U. Malinowski (Hrsg.): *Adaptive User Interfaces: Principles and practice*. Amsterdam: North-Holland. S. 49–68.
- Frey, A.G., Céret, E., Dupuy-Chessa, S., Calvary, G. & Gabillon, Y. (2012). UsiComp: an extensible model-driven composer. In *Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems (EICS '12)*. New York: ACM. S. 263-268.
- Gacimartin, C., Hernandez, J.A. & Larrabeiti, D. (2011). A middleware architecture for designing TV-based adapted applications for the elderly. In J.A. Jacko (Hrsg.): *Human-Computer Interaction, Part I, HCI 2011, LNCS 6761*. Berlin, Heidelberg: Springer-Verlag. S. 443–449.
- Gajos, K.Z., Czerwinski, M., Tan, D.S. & Weld, D.S. (2006). Exploring the design space for adaptive graphical user interfaces. In *Proceedings of the working conference on Advanced visual interfaces (AVI '06)*. New York: ACM, S. 201-208.
- Gajos, K.Z. & Weld, D.S. (2004). SUPPLE: Automatically Generating User Interfaces. In *Proceedings of the 9th international conference on Intelligent user interfaces (IUI '04)*. New York: ACM. S. 93–100.
- Gajos, K.Z., Weld, D.S., & Wobbrock, J. O. (2010). Automatically generating personalized user interfaces with Supple. *Artificial Intelligence*, 174 (12-13). S. 910–950.
- Gajos, K.Z., Wobbrock, J.O. & Weld, D.S. (2008). Improving the Performance of Motor-Impaired Users with Automatically-Generated, Ability-Based Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'08)*. New York: ACM. S. 1257–1266.
- Gamma, E., Helm, R., Johnson R. & Vlissides J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley.
- Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8 (3). S. 231-274.
- Harper, R., Rodden, T., Rogers, Y. & Sellen, A. (Hrsg.) (2008). *Being Human: Human-Computer Interaction in 2020*. Cambridge: Microsoft Research Ltd.
- Hartmann, M., Schreiber, D. & Mühlhäuser, M. (2009). AUGUR: Providing Context-Aware Interaction Support. In *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems (EICS '09)*. New York: ACM. S. 123-132.
- Helms, J. & Abrams. M. (2008). Retrospective on UI description languages, based on eight years' experience with the User Interface Markup Language (UIML). *International Journal of Web Engineering and Technology*, 4 (2). S. 138-162.

- Hernández, J.A., Larrabeiti, D., Strnad, O. & Schmidt, A. (2011). Prototype for user context management infrastructure and user modelling. Public deliverable of the MyUI project. http://www.myui.eu/deliverables/MyUI_D1-2_final.pdf
- Höb, O. (2005). Ein System für das Wiederverwendungs-Management von Software-Komponenten. Dissertation Universität Stuttgart. Heimsheim: Jost-Jetter Verlag.
- Höb, O. & Weisbecker, A. (2002). Konzeption eines Repositories zur Unterstützung der Wiederverwendung von Software-Komponenten. In K. Turowski (Hrsg.): Tagungsband des 4. Workshops Komponentenorientierte betriebliche Anwendungssysteme (WKBA 4). S. 57-74.
- Horvitz, E. (1999). Principles of mixed-initiative user interfaces. In Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit (CHI '99). New York: ACM. S. 159-166.
- Horrocks, I. (1999). Constructing the User Interface with Statecharts. Boston, MA: Addison-Wesley.
- ISO/TC 159 „Ergonomics“ & CEN/TC 122 „Ergonomie“ (2008a). DIN EN ISO 9241-171:2008 Ergonomie der Mensch-System-Interaktion - Teil 171: Leitlinien für die Zugänglichkeit von Software.
- ISO/TC 159 “Ergonomics” (2008b). DIN EN ISO 9241-110:2006 Ergonomie der Mensch-System-Interaktion - Teil 110: Grundsätze der Dialoggestaltung.
- ISO/TC 159 “Ergonomics” (2010). ISO/TR 9241-100:2010 Ergonomics of human-system interaction - Part 100: Introduction to standards related to software ergonomics.
- Jameson, A. (2001). Systems That Adapt to Their Users: An Integrative Perspective. Saarbrücken: Universität des Saarlandes.
- Jameson, A. (2009). Understanding and Dealing with Usability Side Effects of Intelligent Processing. *AI Magazine*, 30 (4). S. 23-40.
- Jason, B., Calitz, A.P., & Greyling, J.H. (2010). The evaluation of an adaptive user interface model. In Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists (SAICSIT ,10). S.132-143.
- Jeckle, M., Rupp, C., Hahn, J., Zengler, B. & Queins, S. (2004). UML 2 glasklar. München, Wien: Carl Hanser Verlag.
- Kane, S.K., Wobbrock, J.O. & Smith, I.E. (2008). Getting off the treadmill: evaluating walking user interfaces for mobile devices in public spaces. In Proceedings of the 10th international conference on Human computer interaction with mobile devices and services (MobileHCI '08). New York: ACM. S. 109-118.
- Kobsa, A., Koenemann, J., & Pohl, W. (2001). Personalised hypermedia presentation techniques for improving online customer relationships. *The Knowledge Engineering Review*, 16 (2), S. 111–155.
- Krueger, R. A., & Casey, M. A. (2000). Focus groups: A practical guide for applied research (3. Auflage). Thousand Oaks, California: Sage Publications.

- Krzywicki, A., Wobcke, W. & Wong, A. (2010). An adaptive calendar assistant using pattern mining for user preference modelling. In Proceedings of the 15th international conference on Intelligent user interfaces (IUI '10). New York: ACM. S. 71-80.
- Kühme, T. (1993). A user-centered approach to adaptive interfaces. In W.D. Gray, W.E. Hefley, & D. Murray (Hrsg.): Proceedings of the 1st international conference on Intelligent user interfaces (IUI '93). New York: ACM. S. 243-245.
- Kühme, T. & Schneider-Hufschmidt, M. (1993). Introduction. In M. Schneider-Hufschmidt, T. Kühme & U. Malinowski (Hrsg.): Adaptive User Interfaces: Principles and Practice. Amsterdam: North-Holland. S. 1-9.
- Lavie, T. & Meyer, J. (2010). Benefits and costs of adaptive user interfaces. *International Journal of Human-Computer Studies*, 68 (8). S. 508-524.
- Lehmann, G., Blumendorf, M. & Albayrak, S. (2010). Development of context-adaptive applications on the basis of runtime user interface models. In Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '10). New York: ACM. S. 309-314.
- Leonidis, A., Antona, M., & Stephanidis, C. (2012). Rapid Prototyping of Adaptable User Interfaces. *International Journal of Human-Computer Interaction*, 28 (4). S. 213-235.
- Leuteritz, J.-P., Widloither, H., Mourouzis, A., Panou, M., Antona, M., & Leonidis, A. (2009). Development of Open Platform Based Adaptive HCI Concepts for Elderly Users. In C. Stephanidis (Hrsg.): *Universal Access in HCI, Part II, HCII 2009*. LNCS 5615. Berlin, Heidelberg: Springer-Verlag. S. 684–693.
- Lin, J. & Landay, J. A. (2008). Employing patterns and layers for early-stage design and prototyping of cross-device user interfaces. In Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems (CHI '08). New York: ACM. S. 1313-1322.
- Lucke, D., Constantinescu, C. & Westkämper, E. (2008). Kontextbezogene Anwendungen in der Produktion: Smart factory - Gestern, Heute und in der Zukunft. *wt Werkstattstechnik online*, 98 (3). Düsseldorf: Springer-VDI Verlag. S. 138-142.
- Mackay, W. E. (1991). Triggers and barriers to customizing software. In S. P. Robertson, G. M. Olson & J. S. Olson (Hrsg.): Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '91). New York: ACM. S. 153–160.
- Maier, T., Schmid, M. & Petrov, A. (2008). HMI with Adaptive Control Elements. *ATZautotechnology*, 8 (7). S. 50-55.
- Marcotte, E. (2011). *Responsive Web Design*. New York: A Book Apart.
- Marcus, A. (2011). Cross-cultural user-experience design. In Proceedings of the SIGGRAPH Asia 2011 Courses (SA '11). New York: ACM. Artikel 3, 201 Seiten.
- McGrenere, J., Baecker, R. M. & Booth, K. S. (2002). An evaluation of a multiple interface design solution for bloated software. In Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves (CHI '02). New York: ACM. S. 164-170.

- Meskens, J., Vermeulen, J., Luyten, K. & Coninx, K. (2008). Gummy for multi-platform user interface designs: shape me, multiply me, fix me, use me. In Proceedings of the working conference on Advanced visual interfaces (AVI '08). New York: ACM. S. 233-240.
- Mori, G., Paterno, F. & Santoro, C. (2004). Design and Development of Multidevice User Interfaces through Multiple Logical Descriptions. *IEEE Transactions on Software Engineering*, 30 (8). S. 507-520.
- Motti, V. G. (2011). A computational framework for multi-dimensional context-aware adaptation. In Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '11). New York: ACM, S. 315-318.
- Moustakas, K., Kaklanis, N., Tzovaras, D., Peissner, M., Lawo, M., Mohammad, Y., Biswas, P. & Hahn, V. (2011). User Model Interoperability Requirements. Public deliverable of the VUMS cluster. http://cluster-wiki.vr.iao.fhg.de/images/7/78/VERITAS_D1.6.4_resubmitted.pdf
- Myers, B., Hudson, S.E. & Pausch, R. (2000). Past, present, and future of user interface software tools. *ACM Transactions on Computer-Human Interaction (TOCHI) - Special issue on human-computer interaction in the new millennium, Part 1*, 7 (1). S. 3-28.
- Newman, M.W. & Landay, J.A. (2000). Sitemaps, storyboards, and specifications: a sketch of Web site design practice. In D. Boyarski & W.A. Kellogg (Hrsg.): Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques (DIS '00). New York: ACM. S. 263-274.
- Nichols, J., Chau, D.H. & Myers, B.A. (2007). Demonstrating the viability of automatically generated user interfaces. In Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '07). New York: ACM. S. 1283-1292.
- Nichols, J. & Myers, B. A. (2009). Creating a lightweight user interface description language: An overview and analysis of the personal universal controller project. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 16 (4). Artikel 17, 37 Seiten.
- Nichols, J., Myers, B. A. & Rothrock, B. (2006a). UNIFORM: automatically generating consistent remote control user interfaces. In R. Grinter, T. Rodden, P. Aoki, E. Cutrell, R. Jeffries & G. Olson (Hrsg.): Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06). New York: ACM. S. 611-620.
- Nichols, J., Rothrock, B., Chau, D. H. & Myers, B. A. (2006b). Huddle: automatically generating interfaces for systems of multiple connected appliances. In Proceedings of the 19th annual ACM symposium on User interface software and technology (UIST '06). New York: ACM. S. 279-288.
- Nielsen, J. (1993). *Usability Engineering*, Cambridge, MA: Academic Press.
- Nielsen, J. (1997). The use and misuse of focus groups. *IEEE Software*, 14 (1), S. 94-95.
- Norman, D. (1994). How might people interact with agents. *Communications of the ACM*, 37 (7). S. 68-71.
- OMG (2011). Unified Modeling Language Superstructure, V 2.4.1. <http://www.omg.org/spec/UML/2.4.1/Superstructure>
- Oppermann, R. (1994). Adaptively supported adaptability. *International Journal of Human Computer Studies*, 40 (3). S. 455-472.

- Oviatt, S. (2003). Multimodal interfaces: Fundamentals, evolving technologies and emerging applications. In J.A. Jacko & A. Sears (Hrsg.). *The human-computer interaction handbook*. Hillsdale, NJ: Lawrence Erlbaum Associates Inc. S. 286-304.
- Paramythis, A., Totter, A., & Stephanidis, C. (2001). A modular approach to the evaluation of Adaptive User Interfaces. In S. Weibelzahl, D. Chin & G. Weber (Hrsg.): *Proceedings of the Workshop on Empirical Evaluations of Adaptive Systems im Rahmen der 8th International Conference on User Modeling (UM'2001)*. S. 9-24.
- Paterno, F. (1999). *Model-Based Design and Evaluation of Interactive Applications*. London: Springer-Verlag.
- Paterno, F., Santoro, C., Mäntyjärvi, J., Mori, G. & Sansone, S. (2008). Authoring pervasive multimodal user interfaces. *International Journal of Web Engineering and Technology*, 4 (2). S. 235-261.
- Paterno, F., Santoro, C. & Spano, L. D. (2009). MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 16 (4). Artikel 19, 30 Seiten.
- Peissner, M., Dangelmaier, M., Biswas, P., Mohamad, Y., Jung, C. Wolf, P., González, M.F. & Kaklanis, N. (2011a). Interim Report on VUMS cluster standardization. Public deliverable of the VUMS cluster. <http://wiki.iao.fraunhofer.de/images/studien/vums-interim-report.pdf>
- Peissner, M. & Edlin-White, R. (2013). User Control in Adaptive User Interfaces for Accessibility. In P. Kotzé et al. (Hrsg.): *Proceedings of INTERACT 2013: 14th IFIP Conference on Human-Computer Interaction, Part I, LNCS 8117*. Berlin, Heidelberg: Springer-Verlag. S. 623-640.
- Peissner, M., Häbe, D., Janssen, D. & Sellner, T. (2012a). MyUI: generating accessible user interfaces from multimodal design patterns. In *Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems (EICS '12)*. New York: ACM. S. 81-90.
- Peissner, M., Häbe, D. & Schuller, A. (2011b). Adaptation concept and multimodal user interface patterns repository. Public deliverable of the MyUI project. http://www.myui.eu/deliverables/MyUI_D2-2_final.pdf
- Peissner, M., Janssen, D. & Sellner, T. (2012b). MyUI Individualization Patterns for Accessible and Adaptive User Interfaces. In *Proceedings of the First International Conference on Smart Systems, Devices and Technologies (SMART 2012)*. IARIA. S. 25-30.
- Peissner, M., Schuller, A. & Spath, D. (2011c). A design patterns approach to adaptive user interfaces for users with special needs. In J.A. Jacko (Hrsg.): *Human-Computer Interaction, Part I, HCII 2011, LNCS 6761*. Berlin, Heidelberg: Springer-Verlag. S. 268-277.
- Peissner, M., Ziegler, D., Häbe, D. & Schuller, A. (2013). User Interface Adaptation Engine Prototype. Public deliverable of the MyUI project. http://www.myui.eu/deliverables/MyUI_D2-3_final.pdf
- Peng, X., & Silver, D. L. (2007). Interface Adaptation Based on User Expectation. *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW '07)*. S.264-269.

- Petrov, A. (2012). Usability-Optimierung durch adaptive Bediensysteme. Dissertation Universität Stuttgart.
- Petrov, A. & Maier, T. (2009). Adaptiv variable Stellteile - Neueste Untersuchungen zur Merkmalerkennung und Zustandsdifferenzierung. Der Mensch im Mittelpunkt technischer Systeme, 8. Berliner Werkstatt Mensch-Maschine-Systeme. Berlin. S. 77-78
- Pomberger, G. (2002). Prozedurorientierte Programmierung. In P. Rechenberg & G. Pomberger (Hrsg.): Informatik-Handbuch (3. Auflage). München, Wien: Carl Hanser Verlag.
- Ringbauer, B., Peissner, M., & Gemou, M. (2007). From "design for all" towards "design for one" – A modular user interface approach. In C. Stephanidis (Hrsg.): Universal Access in HCI, Part I, HCII 2007, LNCS 4554. Berlin, Heidelberg: Springer-Verlag. S. 517–526.
- Sánchez, V., Sánchez, J. & Fergusson, M. (2012). Feedback from application developers. Confidential deliverable of the MyUI project.
- Savidis, A., Antona, M. & Stephanidis, C. (2005). A Decision-Making Specification Language for Verifiable User-Interface Adaptation Logic. International Journal of Software Engineering and Knowledge Engineering, 15 (6). S. 1063-1094.
- Savidis, A. & Stephanidis C. (2001). The Unified User Interface Software Architecture. In C. Stephanidis (Hrsg.): User Interfaces for All – Concepts, Methods and Tools. Mahwah, NJ: Lawrence Erlbaum Associates Inc. S. 389-415.
- Savidis, A. & Stephanidis, C. (2004). Unified user interface design: designing universally accessible interactions. Interacting with Computers, 16 (2). S. 243-270.
- Sloan, D., Atkinson, M.T., Machin, C. & Li, Y. (2010). The potential of adaptive interfaces as an accessibility aid for older web users. In Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A '10). New York: ACM. Artikel 35, 10 Seiten.
- Sottet, J.-S., Calvary, G., Coutaz, J., Favre, J.-M., Vanderdonckt, J., Stanculescu, A. & Lepreux, S. (2007a). A Language Perspective on the Development of Plastic Multimodal User Interfaces. Journal on Multimodal User Interfaces, 1 (2). S. 1–12.
- Sottet, J.-S., Ganneau, V., Calvary, G., Coutaz, J., Demeure, A., Favre, J.-M. & Demumieux, R. (2007b). Model-driven adaptation for plastic user interfaces. In C. Baranauskas, P. Palanque, J. Abascal & S. Junqueira Barbosa (Hrsg.): Proceedings of the 11th IFIP TC 13 international conference on Human-computer interaction (INTERACT'07). Berlin, Heidelberg: Springer-Verlag. S. 397-410.
- Spath, D. & Weisbecker, A. (Hrsg.) (2013). Potenziale der Mensch-Technik Interaktion für die effiziente und vernetzte Produktion von morgen. Stuttgart: Fraunhofer Verlag.
- Spath, D., Hermann, F., Peissner, M. & Sproll, S. (2012). User Requirements Collection and Analysis. In G. Salvendy (Hrsg.): Handbook of Human Factors and Ergonomics (4. Auflage). Chichester: John Wiley & Sons Inc. S. 1313-1322.
- Stephanidis, C., Paramythis, A., Akoumianakis, D. & Sfyraakis, M. (1998a). Self-adapting web-based systems: Towards universal accessibility. In C. Stephanidis and A. Waern (Hrsg.): Proceedings of the 4th ERCIM Workshop on 'User Interfaces for All'. 17 Seiten.

- Stephanidis, C., Paramythis, A., Sfyarakis, M., Stergiou, A., Maou, N., Leventis, A., Paparoulis, G. & Karagiannidis, C. (1998b). Adaptable and adaptive user interfaces for disabled users in the AVANTI project. In S. Triglia, A. Mullery, M. Campolargo, H. Vanderstraeten & M. Mampaey (Hrsg.): *Intelligence in Services and Networks: Technology for Ubiquitous Telecom Services*. Berlin, Heidelberg: Springer-Verlag. S. 153-166.
- Strnad, O., Felic, A., Schmidt, A. (2012a). Context Management for Self-adaptive User Interfaces in the Project MyUI. In R. Wichert & B. Eberhardt (Hrsg.): *Ambient Assisted Living, Advanced Technologies and Societal Change* (Vol. 6). Berlin Heidelberg: Springer-Verlag, S. 263-272
- Strnad, O., Felic, A. & Schmidt, A. (2012b). Context Management für selbst-adaptive Nutzerschnittstellen am Beispiel von MyUI. In *Tagungsband: 5. Deutscher AAL-Kongress 2012 - Technik für ein selbstbestimmtes Leben (AAL 2012)*. Berlin: VDE Verlag. 6 Seiten.
- Szekely, P. (1996). Retrospective and Challenges for Model-Based Interface Development. In F. Bodart & J. Vanderdonck (Hrsg.): *Proceedings of Design, Specification and Verification of Interactive Systems '96 (DSV-IS'96)*, Eurographics. Berlin, Heidelberg: Springer-Verlag. S. 1-27.
- Takacs, B., Simon, L. & Peissner, M. (2011). Sensing user needs: recognition technologies and user models for adaptive user interfaces. In J.A. Jacko (Hrsg.): *Human-Computer Interaction, Part I, HCII 2011, LNCS 6761*. Berlin, Heidelberg: Springer-Verlag. S. 498-506.
- Totterdell, P. & Rautenbach, P. (1990). Adaptation as a Problem of Design. In D. Browne, P. Totterdell & M. Norman (Hrsg.): *Adaptive User Interfaces*. London: Academic Press. S. 61-84.
- Trewin, S. (2003). Automating accessibility: the dynamic keyboard. *ACM SIGACCESS Accessibility and Computing*, Issue 77-78. S. 71-78.
- Trewin, S., Zimmermann, G. & Vanderheiden, G. (2002). Abstract user interface representations: how well do they support universal access? *ACM SIGCAPH Computers and the Physically Handicapped*, Issue 73-74. S. 77-84.
- Tsandilas, T. & Schraefel, M. C. (2005). An Empirical Assessment of Adaptation Techniques. In *Proceedings of CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05)*. New York: ACM, S. 2009–2012.
- Van den Bergh, J., Luyten, K. & Coninx, K. (2011). CAP3: context-sensitive abstract user interface specification. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '11)*. New York: ACM. S. 31-40.
- Vanderdonck, J., Coutaz, J., Calvary, G. & Stanculescu, A. (2008). Multimodality for Plastic User Interfaces: Models, Methods, and Principles. In D. Tzovaras (Hrsg.): *Multimodal user interfaces: signals and communication technology*. LNEE. Berlin, Heidelberg: Springer-Verlag. S. 61-84.
- Weibelzahl, S. (2002). *Evaluation of Adaptive Systems*. Dissertation Universität Trier.
- Weisbecker, A. (2002). *Software-Management für komponentenbasierte Software-Entwicklung*. Habilitationsschrift Universität Stuttgart. Heimsheim: Jost-Jetter Verlag.
- Weld, D., Anderson, C., Domingos, P., Etzioni, O., Lau, T., Gajos, K. & Wolfman, S. (2003). Automatically personalizing user interfaces. In *Proceedings of the 18th international joint*

conference on Artificial intelligence (IJCAI'03). San Francisco: Morgan Kaufmann Publishers Inc. S. 1613-1619.

Westkämper, E., Jendoubi, L., Eissele, M., Ertl, T. & Niemann, J. (2005). Smart Factories - Intelligent Manufacturing Environments. *Machine Engineering*, 5 (1-2). S. 114-122.

Wobbrock, J.O., Kane, S.K., Gajos, K.Z., Harada, S., & Froehlich, J. (2011). Ability-based design: Concept, principles and examples. *ACM Transactions on Accessible Computing*, 3 (3). Artikel 9, 27 Seiten.

In dieser »Schriftenreihe zu Arbeitswissenschaft und Technologiemanagement« werden die Dissertationen, die im Rahmen von Forschungs- und Entwicklungsarbeiten am Institut für Arbeitswissenschaft und Technologiemanagement IAT der Universität Stuttgart und am Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO entstanden sind, veröffentlicht.

Die beiden Institute verknüpfen universitäre Grundlagenforschung mit angewandter Auftragsforschung und setzen diese erfolgreich in zahlreichen Projekten praxisgerecht um.

Technologiemanagement umfasst dabei die integrierte Planung, Gestaltung, Optimierung, Bewertung und den Einsatz von technischen Produkten und Prozessen aus der Perspektive von Mensch, Organisation, Technik und Umwelt. Dabei werden neue anthropozentrische Konzepte für die Arbeitsorganisation und -gestaltung erforscht und erprobt. Die Arbeitswissenschaft mit ihrer Systematik der Analyse, Ordnung und Gestaltung der technischen, organisatorischen und sozialen Bedingungen von Arbeitsprozessen sowie ihren humanen und wirtschaftlichen Zielen ist dabei zentral in die Aufgabe des Technologiemanagements eingebunden.

ISBN 978-3-8396-0769-5



ISSN 2195-3414

Fraunhofer Verlag