

Nachvollziehbare Adaption grafischer Benutzerschnittstellen

Alexander Kröner

Technische Hochschule Nürnberg Georg Simon Ohm

alexander.kroener@th-nuernberg.de

Abstract. Die Adaption von Benutzerschnittstellen an Nutzungskontexte ist heute eine weit verbreitete Praxis, die durch Wiedererkennbarkeit das Nutzungserlebnis befördern und durch Wiederverwendbarkeit den Entwicklungsaufwand in einer heterogener werdenden Bandbreite von Endgeräten beherrschbar machen kann. Allerdings kann eine Adaption auch das Nutzungserlebnis beeinträchtigen. Dieser Bericht soll Forschern und Entwicklern im Bereich grafischer Benutzerschnittstellen einen Einblick in dieses Thema bieten und als Hilfestellung bei den damit verbundenen Herausforderungen dienen. Dazu bündelt er die Ergebnisse einer Recherche zu den Hintergründen negativer Effekte von Adaption und beschreibt konzeptionelle und technische Ansätze zu möglichen Gegenmaßnahmen.

1 Hintergrund

Kontextsensitive Anwendungen passen Inhalt, Präsentation und Interaktion an den Kontext¹ ihres Benutzers² an (Dey, 2001). Eine solche Adaption von Benutzerschnittstellen kann das Benutzererlebnis befördern, wenn beispielsweise mittels unterschiedlicher Endgeräte eine Anwendung bedient oder die aus großen Informationsmengen resultierende „Information Overload“ (Weiser, 1999) bewältigt werden soll.

Technisch kann diese Adaption sehr unterschiedlich realisiert werden, wie das Beispiel des World Wide Web zeigt. So findet im Falle von Adaptive Web Design (Gustafson, 2011) die Anpassung vor der Auslieferung an den (Web-) Client statt. Responsive Web Design (Marcotte, 2011) dagegen strebt eine Anpassung auf dem Client selbst an. Auch wenn beide Herangehensweisen sich technisch stark unterscheiden, ist die gemeinsame Zielsetzung eine Verbesserung der Gebrauchstauglichkeit³ unter Berücksichtigung des Kontextes des Benutzers.

Aber auch wenn die Adaption einer Benutzerschnittstelle den Benutzer bei der Erreichung seiner Ziele unterstützen soll, so kann sie doch negativ auf das Benutzererlebnis wirken. So muss der Benutzer nach einer Veränderung einer Benutzerschnittstelle zunächst erkennen, dass eine Änderung vorliegt. Gelingt dies nicht, so kann ein Verlust an Orientierung oder ein Verwechseln von Steuerelementen folgen. Im Falle einer responsiven Webseite beispielsweise kann das Ersetzen einer Navigationsleiste durch eine

¹ Der Nutzungskontext beschreibt eine Kombination von Benutzern, Zielen, Aufgaben, Ressourcen und technischer, physikalischer, sozialer, kultureller und organisationsbezogener Umgebung (DIN EN ISO 9241-11).

² Zur stilistischen Vereinfachung wird in diesem Bericht o.B.d.A. die maskuline Form für Personenbezeichnungen und personenbezogene Substantive verwendet.

³ Die Gebrauchstauglichkeit beschreibt das „Ausmaß, in dem ein System, ein Produkt oder eine Dienstleistung durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen“ (DIN EN ISO 9241-11).

kompaktere Darstellung verwirrend wirken (Lek, 2014). Besonders problematisch sind solche Effekte, wenn die Aufmerksamkeit des Benutzers nicht auf die Benutzerschnittstelle gerichtet ist und deren Änderung autonom vom System durchgeführt wird. Dies kann eine Abweichung des vom Benutzer angenommenen Zustandes („Modus“) von Benutzerschnittstelle und Anwendung befördern („Mode Confusion“, vgl. Lee & Eom, 2015). Deren Konsequenz – für den Benutzer unerwartete Reaktionen auf Eingaben – wird auch als „Automation Surprise“ bezeichnet (Sarter et al., 1997).

Ist die Änderung erkannt, so muss der neue Aufbau der Benutzerschnittstelle vom Benutzer untersucht und verstanden werden. Der damit verbundene Zeitaufwand senkt temporär die Effizienz im Umgang mit einer Benutzerschnittstelle, was den Zielen der Gebrauchstauglichkeit widerspricht. Zudem muss der Benutzer das Zusammenspiel von Ursache und Auswirkung einer Anpassung erlernen. Gelingt dies nicht, kann dies einen Verlust an wahrgenommener Kontrolle und Vertrauen in das System nach sich ziehen (Muir, 1994).

Bestehende Normen gehen nur eingeschränkt auf negative Auswirkungen einer Adaption ein, was durch die enorme Vielfalt möglicher Nutzungskontexte und entsprechender Anpassungen begründet sein kann. Speziell für die Anwendung im World Wide Web fordert die Norm DIN EN ISO 9241-151 die Unabhängigkeit von bestimmten Eingabegeräten. Die Darstellung des Inhalts wird diskutiert, betrachtet aber eine statische Situation. Allerdings wird im Falle einer Individualisierung (hier vorrangig der Wahl der Inhalte) ein Kenntlichmachen der individuellen Gestaltung und Anpassung gefordert. Liegt der Anpassung ein Benutzerprofil zugrunde, dann sollte dieses im Sinne der Nachvollziehbarkeit angezeigt werden können. Zudem wird betont, dass eine automatische Individualisierung den Benutzer verwirren kann – und deswegen auch vom diesem abgeschaltet werden können sollte.

Im Zentrum der folgenden Ausführungen steht die Frage, wie man in Konzept und Umsetzung adaptiver Benutzerschnittstellen negativen Einflüssen insbesondere auf die Gebrauchstauglichkeit entgegen kann. Besondere Aufmerksamkeit gilt dabei der Nachvollziehbarkeit der Auswirkungen einer Adaption visueller Merkmale.

2 Intelligibility und Scrutability

Das Verstehen und Nachvollziehen des Erscheinungsbildes und der Bedienung der Benutzerschnittstelle einer Anwendungssoftware (im Folgenden kurz Anwendung) ist in der Literatur eng gekoppelt mit den Begriffen Intelligibility und Scrutability.

In der Kommunikation mittels gesprochener Sprache beschreibt die Intelligibility, wie verständlich Sprache in einer bestimmten Umgebung ist (Wikipedia, 2020b). Diese Definition thematisiert den Einfluss der Umgebung (z.B. dort vorhandener Störgeräusche) darauf, „*wie viel Anstrengung der Zuhörende aufbringen muss, um das gesprochene Wort innerhalb der Mischung zu verstehen*“ (FhG, 2020). Erkennbarkeit gesprochener Wörter in den Vordergrund. Für grafische Benutzerschnittstellen fokussiert folgende, an Vermeulen (2013) angelehnte, Definition das Verständnis der Bedienung.

Def. 1 (Intelligibility): Die Intelligibility einer grafischen Benutzerschnittstelle bezeichnet die Verständlichkeit von Präsentation und Bedienung in einem gegebenen Nutzungskontext.

Teilweise synonym ist in der Literatur der Begriff Scrutability gebräuchlich. Je nach Definition des Begriffs unterstreicht dieser eine (mögliche) aktive Rolle des Menschen im Verständnisprozess eines

Sachverhalts⁴. Dies zeigt sich in Definitionen von „scrutable“ als „*ergründbar, nachvollziehbar*“ (Beolinguus, 2021) und „*capable of being understood by careful study or investigation*“ (Dictionary.com, 2021), die Aspekte der Intelligibility mit der Untersuchung des Sachverhalts durch den Menschen mischen. Um die Trennung vom Begriff der Intelligibility zu stärken, folgt dieser Bericht letzterer Lesart.

Def. 2 (**Scrutability**): Die Scrutability einer Anwendung beschreibt deren Fähigkeit, dem Benutzer durch Beobachtung und Inspektion ein Nachvollziehen des Verhaltens der Anwendung in einem Nutzungskontext zu ermöglichen.

3 Einordnung

Die Auswirkung von Änderungen in einer Benutzerschnittstelle auf deren Bedienung berührt verschiedene Themen im Umfeld der Gebrauchstauglichkeit. Dieser Bericht betrachtet nur einen Ausschnitt dieses Themenkomplexes; entsprechende Anknüpfungspunkte und Abgrenzungen werden im Folgenden näher erläutert.

3.1 Benutzerschnittstellen

Dieser Bericht diskutiert die Auswirkung von Änderungen in der Präsentation einer gegebenen Benutzerschnittstelle. Änderungen im Funktionsumfang auf Basis der Logik der Anwendung werden hier vernachlässigt. Ebenfalls nicht näher betrachtet werden Veränderungen aufgrund eines geänderten Ausgabegerätes. Beispielsweise kann die Interaktion mit einer Anwendung über mehrere Geräte hinweg erfolgen.⁵ Deren individuelle Ausgabemöglichkeiten können eine gerätespezifische Anpassung der Benutzerschnittstelle erfordern. Der Wechsel des Gerätes wird im Folgenden als bewusste Handlung des Benutzers gesehen, was die Wahrnehmung von Änderungen begünstigt. Allerdings können solche Änderungen sehr umfassend ausfallen und auch die Auswahl der von der Anwendung über die Benutzerschnittstelle angebotenen Funktionen beeinflussen. Dieses Zusammenspiel wird als spezifisch für die Zielsetzung der jeweiligen Anwendung angesehen und nicht weiter untersucht.

Eine weitere Schwerpunktsetzung gilt den Merkmalen der betrachteten Benutzerschnittstellen. Der Schwerpunkt wird im Folgenden auf grafische Benutzerschnittstellen gesetzt, die dem WIMP-Paradigma⁶ folgen. Die Modalität der Ausgabe wird dementsprechend als grafisch angenommen. Die Modalität der Eingabe und ein Wechsel der Modalitäten werden nicht näher betrachtet.

3.2 Individualisierbarkeit

Im Sinne der Gebrauchstauglichkeit sollte ein interaktives System es den Benutzern erlauben, die Benutzerschnittstelle an die individuellen Erfordernisse und Vorlieben zu adaptieren (DIN EN ISO 9241-110). Dies kann auch der Barrierefreiheit dienen; beispielweise kann man einer beeinträchtigten Sehkraft mit einer Veränderung von Schriftgröße und Farbwahl entgegenen.

Eine solche Anpassung als Ergebnis eines willentlichen Akts des Benutzers kann aus Sicht der Intelligibility insofern vorteilhaft sein, dass der Benutzer eine Veränderung der Benutzerschnittstelle als Ergebnis seiner Handlung erwartet. Dies begünstigt das Wahrnehmen einer Änderung. Darüber hinaus können aber auch solche willentlichen Akte unerwartete Änderungen in der Benutzerschnittstelle nach sich ziehen. Es wird daher im Folgenden angenommen, dass die Auswirkungen einer Individualisierung

⁴ Eine solche aktive Rolle kann das Vertrauen des Benutzers in das System stärken – und auch als Grundlage einer Korrektur adaptiven Verhaltens dienen (Kay, 2006).

⁵ Dies stellt eine Variante der Cross-Device Interaction dar, in der mehrere Interaktionsgeräte – auch potentiell zeitgleich – zur Bedienung einer Anwendung verwendet werden (vgl. Houben et al., 2017).

⁶ WIMP steht für Windows, Icons, Menus, Pointer, vgl. Gentner & Nielsen (1996).

keiner gesonderten Diskussion bedürfen und die in diesem Bericht getroffenen Überlegungen auch in solchen Fällen anwendbar sind.

3.3 Lokalisierung

Ein Spezialfall ist die Anpassung der Benutzerschnittstelle an die „[...] *in einem bestimmten geographisch oder ethnisch umschriebenen Absatz- oder Nutzungsgebiet (Land, Region oder ethnische Gruppe) vorherrschenden, lokalen sprachlichen und kulturellen Gegebenheiten*“ (Wikipedia, 2020a).

Eine solche „Lokalisierung“ kann zahlreiche Aspekte einer Benutzerschnittstelle betreffen. Als Beispiel können die Webseiten der BBC dienen, die in zahlreichen Lokalisierungen vorliegen. Diese unterscheiden sich teilweise stark in Layout und Typographie, aber auch die Auswahl der Inhalte kann betroffen sein (BBC, 2021). Bei Änderungen innerhalb einer lokalisierten Benutzerschnittstelle ist zudem zu erwarten, dass auch der Prozess der Änderung selbst einer Lokalisierung unterliegen kann – z.B., ob ein Element in Folge einer Änderung links oder rechts eingeblendet wird.

Im Folgenden wird die Behandlung negativer Effekte für den Benutzer aufgrund einer durch Lokalisierung motivierten Änderung der Benutzerschnittstelle weitgehend ausgeblendet. Dem liegen die Annahmen zugrunde, dass ein Benutzer eine Anwendung nicht in mehreren Lokalisierungen bedient und eine Lokalisierung vom System vor Beginn der Interaktion gewählt wird. Sollte der Benutzer die Lokalisierung während der Interaktion verändern, können die folgenden Anpassungen ähnlich dem Fall einer Individualisierung gehandhabt werden.

Sollte dennoch eine Reduzierung negativer Effekte in Folge einer Lokalisierung erforderlich sein, lassen sich die in den folgenden Abschnitten beschriebenen Überlegungen übertragen. Dabei ist zu beachten, dass resultierende Maßnahmen zur Stärkung der Intelligibility ebenfalls einer Lokalisierung unterliegen sollten – ein Aspekt, der in diesem Bericht nicht weiter untersucht wird.

4 Einflussfaktoren

Die Nachvollziehbarkeit von Änderungen in einer Benutzerschnittstelle wird von verschiedenen Faktoren positiv wie negativ beeinflusst. Vor einer Diskussion von Ansätzen zur Stärkung der Nachvollziehbarkeit werden zunächst ausgewählte Faktoren näher betrachtet.

4.1 Komplexität einer Änderung

Die Effizienz der Zielerreichung in einem Nutzungskontext lässt sich nach Bevan & Macleod (1994) bestimmen anhand des zur Zielerreichung nötigen Einsatzes von Ressourcen. Um die Auswirkung einer Adaption auf die Gebrauchstauglichkeit zu untersuchen, kann die zur Erledigung einer gegebenen Aufgabe nötige Zeit vor und nach einer Änderung verglichen werden. Von besonderem Interesse aus Sicht der Intelligibility ist die Zeitspanne T zwischen dem Abschluss der Änderung durch das System und dem Beginn der Bearbeitung der Aufgabe durch den Benutzer. Sie kann als Indikator genutzt werden, um zu identifizieren, wann unabhängig von der eigentlichen Bearbeitung der (geänderte) Aufbau der Benutzerschnittstelle grundsätzlich verstanden ist.

Im Falle einer hohen Intelligibility sollte sich T analog Hick (1952) am Informationsgehalt der Benutzerschnittstelle (hier insbesondere Zahl und visuelle Merkmale von Steuerelementen) orientieren. Voraussetzung ist, dass der Benutzer mit der Benutzerschnittstelle vor und nach der Änderung bereits vertraut ist. Mit Blick auf die Vielfalt möglicher Änderungen scheint dies nur eingeschränkt erreichbar; es ist also zu erwarten, dass der Benutzer sich zunächst mit den Änderungen in der Benutzerschnittstelle vertraut macht. In Anlehnung an Erkenntnisse von Cockburn & Gutwin (2008) zum Umgang trainierter und untrainierter Benutzer mit hierarchischen Listen wird im Folgenden angenommen, dass nach einer

Adaption vor allem die Anzahl der Änderungen für den Zeitaufwand bestimmend ist. Diese Überlegung beruht darauf, dass ein mit der ursprünglichen Benutzerschnittstelle vertrauter Benutzer sich in erster Linie mit den Änderungen und nicht mit der vollständigen Benutzerschnittstelle vertraut machen muss. Dies wiederum legt nahe, dass eine niedrige Anzahl von Änderungen günstig für die Intelligibility einer Benutzerschnittstelle ist.

4.2 Mentales Modell einer Änderung

Um Ereignisse vorherzusagen oder zu erklären, konstruiert der menschliche Verstand Modelle, die das Verhalten der Wirklichkeit imitieren (Craik, 1943, S. 51). Nielsen (2010) führt am Beispiel von Webseiten aus, dass ein solches mentales Modell Wissen oder Annahmen des Benutzers zur Bedienung umfasst. Seiner Diskussion folgend besteht eine wesentliche Aufgabe des Designs der Benutzerschnittstelle darin, die Bildung eines (zum Umgang mit der Anwendung) ausreichend präzisen mentalen Modells grundlegender Aspekte der Webseite zu befördern – was der Zielsetzung der Intelligibility entspricht.

Ist das mentale Modell des Benutzers bekannt, kann eine dem Modell konforme Benutzerschnittstelle die vom Benutzer benötigte Zeit zur Orientierung senken, wie Roth et al. (2013) am Beispiel von Webseiten ausgewählter Anwendungsbereiche zeigen.

Diese Betrachtungen lassen sich auf Änderungen in adaptiven Benutzerschnittstellen übertragen. Wenn der Benutzer aufgrund eines mentalen Modells eine Änderung in der Benutzerschnittstelle grundsätzlich erwartet, dann kann dies seine Aufmerksamkeit erhöhen und somit den initialen Zeitaufwand zum Entdecken von und Reagieren auf Änderungen verkürzen. In gleicher Weise kann es positiv wirken, wenn die Umsetzung der Änderung den Annahmen des Benutzers entspricht, da die Merkmale der Umsetzung dann nicht erst erlernt werden müssen.

Für die Stärkung der Nachvollziehbarkeit von Änderungen ist die Kenntnis des diesbezüglichen mentalen Modells des Benutzers somit von großem Interesse. Diese Kenntnis zu erlangen, kann sich aber als fordernd erweisen. So sind die einem Modell zugrundeliegenden Glaubensstrukturen einer Inspektion nur eingeschränkt zugänglich; Befragungen können daher zu einem unvollständigen oder fehlerhaften Ergebnis führen (Norman, 1983, S. 11). Zudem basiert ein Modell auf individuellen Erfahrungen, was eine Übertragung auf andere Benutzer behindert.

5 Nachvollziehbarkeit stärken

Negativen Effekten von Adaption auf die Bedienbarkeit kann man entgegenwirken. Im Folgenden werden zwei grundlegende Ansätze zur Erreichung dieses Ziels betrachtet. Implizite Maßnahmen sollen negativen Effekten auf der Ebene der Präsentation vorbeugen. Explizite Maßnahmen ergänzen eine Benutzerschnittstelle um Erläuterungen anstehender oder bereits durchgeführter Änderungen.

5.1 Implizite Maßnahmen

Folgt man den Überlegungen des vorigen Kapitels, dann sollten Änderungen einer Benutzerschnittstelle so durchgeführt werden, dass sie leicht erkannt werden können und in ihrer Form dem mentalen Modell des Benutzers entsprechen.

5.1.1 Wahrnehmung von Änderungen unterstützen

Nach einer Adaption muss der Benutzer zunächst erkennen, dass eine Änderung stattfand. Mit solchen optischen Ereignissen sind (in Anlehnung an Gibson, 1966) die Veränderung der Flächenanordnung, der Textur oder auch das Erscheinen oder Verschwinden von Teilen des sichtbaren Bildes verbunden.

Beeinträchtigt werden kann die Wahrnehmung solcher Änderungen durch konzentrierte Aufmerksamkeit auf einen von der Änderung nicht betroffenen Aspekt der Benutzerschnittstelle oder auch durch

Unaufmerksamkeit.⁷ Potentiell befördert wird diese Problematik dadurch, dass Aktionen des Benutzers, die eine Adaption nach sich ziehen, unter Umständen nur lose an deren Wirkung gekoppelt sind. Als Beispiel sei die Änderung der Größe eines Fensters durch Manipulation des Fensterrahmens genannt, die als direkte Wirkung den Rahmen ändert, aber indirekt auf die enthaltenen Steuerelemente wirken kann.

Zur Unterstützung der Wahrnehmung von Änderungen kann man die Aufmerksamkeit des Benutzers auf (anstehende oder durchgeführte) Veränderungen von Elementen lenken. Dies kann beispielsweise durch das Prinzip des „Highlightings“ erfolgen (Butler et al., 2010). Im Falle grafischer Benutzerschnittstellen basiert dieses Prinzip auf einer Änderung visueller Eigenschaften. Analog Statusmeldungen und Systemmeldungen spielt bei der konkreten Ausprägung des Highlightings die Unterscheidbarkeit von anderen (nicht von einer Änderung betroffenen) Elementen der Benutzerschnittstelle eine wesentliche Rolle (vgl. DIN EN ISO 9241-161). Das Highlighting von Änderungen bietet sich an, wenn Änderungen sich auf wenige Bereiche der Darstellung beschränken; beispielsweise empfehlen Butler et al. (2010) eine Beschränkung auf 10% der sichtbaren Anzeige. Dabei ist zu beachten, dass die Betonung von Änderungen die Aufmerksamkeit des Benutzers von aktuell verfolgten Zielen ablenken kann, was dem Ziel eines effizienten Arbeitens entgegenstehen kann.

Je nach Art der Änderung (etwa der Anordnung) muss der Benutzer zudem Elemente der Benutzerschnittstelle wiederfinden. In diesem Zusammenhang ist die Wahrnehmungskonstanz relevant, z.B. die Konstanz von Formen und Farben (Myers, 2014, S. 260), die dazu beiträgt, ob ein Objekt unabhängig von Änderungen erkannt wird. Verändert sich also eine grafische Benutzerschnittstelle (auch aufgrund eines Highlightings), so sollten visuellen Eigenschaften der dargestellten Elemente zumindest in Teilen invariant bleiben.

5.1.2 Erwartungen durch Gestaltung erfüllen

Die Normierung grafischer Benutzerschnittstellen bietet Leitlinien, die aufgrund einer häufigen Umsetzung Erwartungen befördern können. So empfiehlt die Norm DIN EN ISO 9241-125 zur visuellen Informationsdarstellung eine konsistente Platzierung von Bereichen, was im Fall einer Adaption senkend auf deren Komplexität wirken kann. Vorschläge zur Auswirkung der Größenanpassung von Fenstern auf deren Inhalte umfassen Beibehalten, Skalieren und Umordnen. Dies kann dazu führen, dass Teile der Benutzerschnittstelle nicht länger sichtbar sind. Die Norm spricht allerdings keine Empfehlung dazu aus, welche Vorgehensweise wann zu empfehlen ist. Zudem wird das Ausblenden oder Abstrahieren von Komponenten, beides Ansätze adaptiven Designs, nicht weiter angesprochen. Plattformspezifische Empfehlungen können aber durchaus weitergehende Hinweise liefern. So empfehlen die Apple User Interface Guidelines in Bezug auf ein adaptives Layout eine Minimierung der Änderungen sowie eine Beibehaltung des im Fokus befindlichen Contents (Apple, 2021a).

Die konkrete Umsetzung solcher Normen wird beeinflusst durch kulturelle Aspekte (wie am Beispiel der Webseiten der BBC sichtbar, vgl. BBC, 2021) und den Anwendungskontext. Im Falle von Webseiten beispielsweise ist eine (inhaltliche und das Layout bestimmende) Aufteilung der Benutzerschnittstelle in Header, Content, Sidebar und Footer durchaus gängig, die auch durch Werkzeuge zur Erstellung von Webseiten befördert wird.⁸ Solche Werkzeuge unterstützen Autoren wiederum in der Umsetzung adaptiven Verhaltens in einer durch das Werkzeug für den Anwendungskontext vorgegebenen Weise.

⁷ In der Literatur wird dies mit dem Begriff der „change blindness“ beziehungsweise spezieller mit der „inattentional blindness“ verbunden (Hagendorf et al., 2011, S. 16).

⁸ Beispielsweise bildet das Web-Content-Management-System WordPress die Struktur einer Webseite auf hierarchisch organisierte Vorlagen ab. Diese einheitliche Struktur befördert das Schaffen einheitlicher Präsentationsvorlagen, sogenannter Themes (WordPress, 2021).

Durch die hohe Verbreitung von Werkzeugen⁹ stärkt dies die Verbreitung bestimmter Adaptionenformen, was letztlich die Erwartungen von Benutzern zum Verhalten einer Webseite beeinflusst.

Diese Erwartungen können sich durch die Verbreitung neuer Interaktionstechnologien verändern. Als Beispiel kann hier die Verbreitung responsiver Webseiten dienen. So beobachteten Maurer et al. (2010) in der mobilen Nutzung von Webseiten noch eine Präferenz für das Nutzen der unveränderten Webseite gegenüber mobilen Fassungen. Dem gegenüber steht die heutige Bedeutung responsiver Webseiten, exemplarisch für das Feld der Suchmaschinenoptimierung diskutiert von Hodgson (2018). Diese Entwicklung legt nahe, dass zumindest responsives Verhalten heute Bestandteil des mentalen Modells der Benutzer von Webseiten ist. In Konsequenz sollte daher für eine gegebene Anwendung und Benutzergruppe zunächst ermittelt werden, welche Erwartungen zu adaptivem Verhalten vorliegen.

5.2 Explizite Maßnahmen

Je nach Art der Benutzerschnittstelle und der Komplexität ihrer Änderung reichen implizite Maßnahmen möglicherweise nicht aus, um aus der Änderung resultierende negative Effekte ausreichend zu schwächen. Dann bietet eine explizite Erklärung einen Ansatzpunkt, um das Verständnis des Verhalten einer Benutzerschnittstelle zu befördern.

Solche Erklärungen lassen sich unter anderem klassifizieren nach Zeitpunkt, Abhängigkeit von der Anwendungsdomäne, Ort, Initiative und Modalität (Vermeulen, 2014, S. 36). Im Kontext des vorliegenden Berichts wird diese Klassifikation eingegrenzt. Als Ort wird die zu bedienende Benutzerschnittstelle angenommen im Gegensatz beispielsweise zu einem komplementären Hilfesystem. Mit dem Ziel einer Fokussierung auf grafische Benutzerschnittstellen wird zudem die Modalität auf „textuell“ beziehungsweise „visuell“ beschränkt. Eine textuelle Erklärung kann beispielsweise genutzt werden, um „Warum“- oder „Warum nicht“-Fragen des Benutzers zu Änderungen der Benutzerschnittstelle anzusprechen (Vermeulen, 2014, S. 32). Visuelle Techniken können verwendet werden, um eine bestehende Benutzerschnittstelle im Sinne der Erklärung zu ergänzen, etwa um eine grafische Vorschau möglicher Auswirkungen von (noch nicht durchgeführten) Handlungen des Benutzers (vgl. das Designprinzip „Feedforward“ nach Bellotti & Edwards, 2001).

Erklärungen zum Verhalten einer Benutzerschnittstelle, die während des Dialogs von Mensch und Maschine kommuniziert werden, sind letztlich eine spezialisierte Form einer Rückmeldung des Systems mit Informationen zur Benutzerschnittstelle. Somit sind die Vorgaben der DIN EN ISO 9241-161 zur Umsetzung von Statusinformationen anwendbar. Neben der Unterscheidbarkeit einer Erklärung von anderen visualisierten Daten sollte im Zuge einer Erklärung demnach der Nutzungskontext angegeben werden, in dem die Änderung erfolgte.

Dabei ist zu beachten, dass Erklärungen die Benutzerschnittstelle potentiell ihrerseits ändern und somit deren (visuelle) Komplexität steigern können. Zudem können Faktoren wie etwa Änderungsfrequenz und Umfang der Mitteilung die Kommunikation der eigentlich identifizierten Änderungen zu einer kognitiven Belastung für den Benutzer machen. Weitere flankierende Maßnahmen, etwa zur Abstraktion und Kategorisierung von Änderungshinweisen, scheinen daher empfehlenswert.

6 Umsetzungsaspekte

Aufgrund der Vielfalt möglicher Anpassungen im Zuge einer Adaption stellt sich die Frage, wie die Umsetzung der zuvor diskutierten Maßnahmen im Rahmen eines Entwicklungsprozesses systematisch

⁹ Im März 2021 wurden nur 37,1% der von W3Techs regelmäßig untersuchten Webseiten ohne Zuhilfenahme eines (W3Techs bekannten) Web-Content-Management-Systems erstellt (W3Techs, 2021).

unterstützt werden kann. Im Sinne einer Orientierungshilfe folgt daher ein nicht abschließender Einblick in Vorgehensweisen und Werkzeuge, die zu diesem Zweck herangezogen werden können.

6.1 Entwurf

Ein nutzerzentrierter Entwicklungsprozess sieht vor, Informationen zum Nutzungskontext zu erheben. Dies kann Hinweise zu Unterschieden in Kontexten liefern, die letztlich eine Adaption der Benutzerschnittstelle an diese Kontexte motivieren. Die folgende Diskussion betrachtet eine (mögliche) Herangehensweise an diesen Fall näher.

Als Ausgangspunkt für die folgenden Schritte kann eine Benutzerschnittstelle für einen ausgewählten Nutzungskontext entworfen werden. Wird beispielsweise eine mehrheitlich mobile Nutzung erwartet, bietet „Mobile First“ einen Ansatz, der mobile Nutzung berücksichtigt und durch Einsatz responsiven Designs auf Varianten dieses Kontextes ausgedehnt werden kann (Wroblewski, 2011). Stößt diese Vorgehensweise an ihre Grenzen, z.B. aufgrund stark abweichender Ein- und Ausgabegeräte, können zusätzliche Entwürfe von Bedienschnittstellen geschaffen werden.

Der mit dieser Vorgehensweise verbundene Aufwand ist stark an die Zahl der benötigten Varianten gekoppelt. Erschwerend wirkt die Notwendigkeit, dass Varianten aufgrund der in Kapitel 5 ausgeführten Überlegungen zum mentalen Modell von Änderungen nicht als unabhängig betrachtet werden können. Ebenfalls fordernd ist der Entwurf von Übergängen zwischen Varianten einer Benutzerschnittstelle. Um diese für spätere Benutzer erlebbar zu machen (und mit deren mentalem Modell abzustimmen), ist letztlich bereits in der Entwurfsphase ein interaktives System empfehlenswert.

Entwicklungswerkzeuge können diese Anforderungen explizit ansprechen und so helfen, den mit dem Entwurf adaptiver Systeme verbundenen Aufwand beherrschbar zu machen. Als Beispiel soll eine Auswahl des Funktionsumfangs von Adobe XD 38 (Adobe, 2021) zum Zeitpunkt des Verfassens dieses Artikels dienen:

- Oberflächen-Entwürfe können miteinander verknüpft werden; deren Zusammenspiel kann in Form von Klick-Prototypen¹⁰ interaktiv erlebt werden. Dies kann auch verwendet werden, um das Zusammenspiel von Varianten der Benutzerschnittstelle zu entwerfen und zu illustrieren.
- Ein „Responsive Resize“ als Reaktion auf eine Größenanpassung der Zeichenfläche passt zuvor entsprechend markierte Elemente der Benutzerschnittstelle in Größe und Position an. Dies unterstützt den Entwurf von Varianten und befördert die Vorhersage einer auf Größe und Anordnung beschränkten Adaption.
- Werkzeuge wie z.B. „Komponenten“ ermöglichen die Schaffung wiederverwendbarer Elemente der Benutzerschnittstelle und deren Änderung an zentraler Stelle. Dies stützt die Bildung von Varianten.

Eine solche Funktionalität kann auch Bestandteil von Programmierumgebungen sein. So bietet Apple Interface Builder Möglichkeiten, eine Benutzerschnittstelle für mehrere Geräteklassen (Smartphone, Tablet, Desktop) zu definieren. Damit verbunden sind die Konfiguration der automatischen Adaption des Layouts (Größe, Abstände) zur Laufzeit, die individuelle Verwaltung dieser Konfigurationen für die jeweiligen Geräteklassen sowie die Möglichkeit einer Vorschau. (Apple, 2021b)

¹⁰ Ein Klick-Prototyp basiert auf einer (möglicherweise vereinfachten) visuellen Präsentation mehrerer Zustände einer Benutzerschnittstelle. Durch Klicken kann zwischen diesen Zuständen gewechselt werden (DX Heroes, 2021).

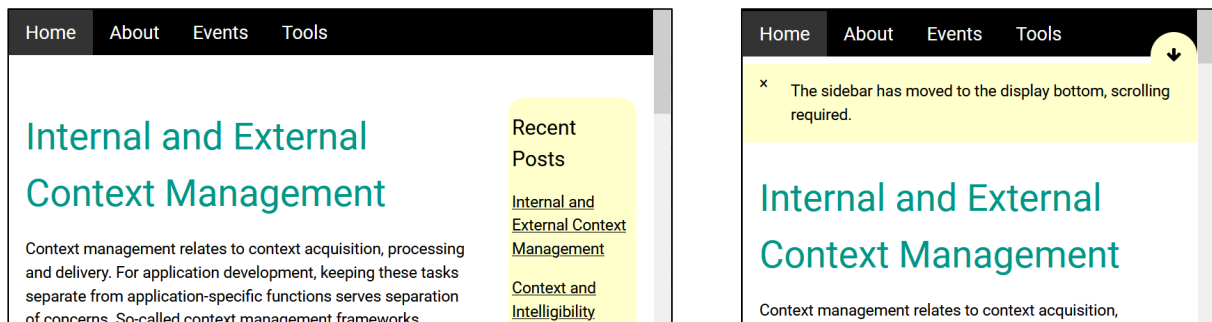


Abbildung 1: Links die ursprüngliche Darstellung einer Webseite. Nach einer Größenänderung (rechts) ist die Seitenleiste nicht länger sichtbar; Änderungen in der Darstellung können die Nachvollziehbarkeit dieses Verhaltens stärken.

6.2 Modellbasierte Adaption

Benutzerschnittstellen können anhand von Modellen automatisch generiert werden, beispielsweise um bei stark variierenden und zahlreichen Nutzungskontexten den Aufwand für Entwicklung und Pflege der zugrundeliegenden Software beherrschbar zu machen. Dem Ansatz folgend wird die Anpassung einer Benutzerschnittstelle durch ein Adaptionsmodell beschrieben, das Art und Ablauf der Anpassung vor oder während der Laufzeit beschreibt (Calvary et al., 2003). Die Verfügbarkeit eines solchen Modells eröffnet einen Weg, um – im Sinne der Scrutability – Vorgehensweise und Gründe einer Anpassung dem Benutzer zu vermitteln und so die Intelligibility der Benutzerschnittstelle aktiv zu stärken.

Ein modellbasierter Ansatz kann auch zur Umsetzung passiver Maßnahmen verwendet werden. Wie dies gelingen kann, lässt sich beispielsweise dem Ansatz von Lee & Eom (2015) im Feld der Automatisierung entnehmen. Dessen Grundlage bildet ein Modell, das Zustände der Anwendung sowie Zustandswechsel der Benutzerschnittstelle und Zustandswechsel der Automation in sich vereinigt. Eine systematische Analyse bietet einen Weg, um Abweichungen zwischen dem vom Benutzer auf Basis der Benutzerschnittstelle erwarteten Zustand der Anwendung und deren tatsächlichem Zustand zu identifizieren. Diese können dann im Sinne einer passiven Maßnahme angewendet werden, um das Verhalten des Systems dem mentalen Modell des Benutzers anzupassen.

6.3 Erklärungen integrieren

Die Integration der Erläuterung einer (noch ausstehenden, gerade laufenden oder bereits durchgeführten) Anpassung in eine grafische Benutzerschnittstelle kann technisch auf verschiedene Weise gelöst werden. Mögliche Lösungen hängen dabei ab von der angestrebten Form der Erläuterung sowie den technischen Merkmalen des Frameworks, das der Benutzerschnittstelle einer Anwendung zugrunde liegt. Die im Folgenden aufgeführten Beispiele sind dementsprechend nicht als Empfehlung für einen Anwendungsfall zu verstehen, sondern sollen Ansatzpunkte zur technischen Umsetzung am Beispiel der Entwicklung von Webseiten vermitteln.

6.3.1 Beispiel: Einsatz Framework-spezifischer Techniken

Adaption kann bewirken, dass Steuerelemente ein- oder ausgeblendet werden. Aus Sicht der Nachvollziehbarkeit ist insbesondere letzterer Fall von Interesse, da sich Maßnahmen zur Stärkung der Nachvollziehbarkeit auf ein nicht mehr sichtbares Element beziehen.

Abbildung 1 zeigt einen solchen Fall. Wird die Anzeige verkleinert, verschwindet aufgrund einer Umgestaltung des Layouts die Seitenleiste aus dem sichtbaren Bereich. Zur Stärkung der Nachvollziehbarkeit wird ein dieses Verhalten erläuterndes Panel eingeblendet (explizite Maßnahme). Das Panel kann vom Benutzer geschlossen werden, um die Komplexität der Darstellung nicht permanent zu erhöhen. Ergänzend wird ein Steuerelement eingeblendet, das dem Benutzer permanent direkten Zugang zu den Inhalten der Seitenleiste gewährt. Zur Stärkung der Wahrnehmungskonsistenz sind diese Elemente in der Farbe der Seitenleiste gehalten (passive Maßnahme).

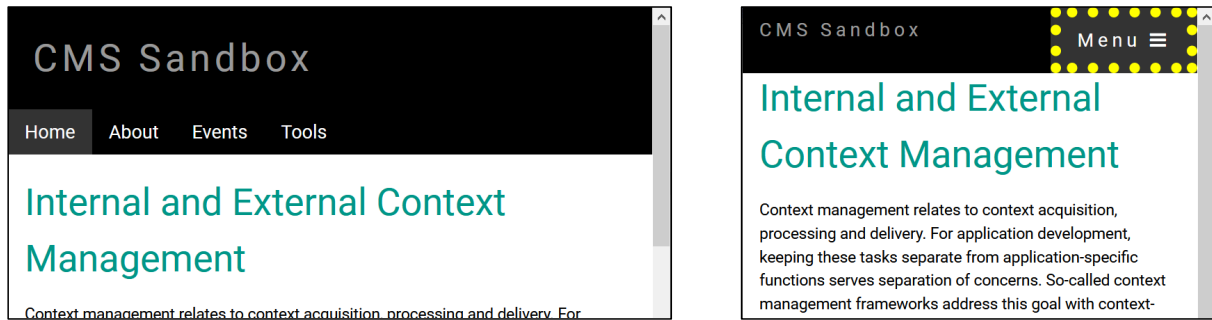


Abbildung 2: Links die ursprüngliche Darstellung, rechts die Reaktion auf eine Verkleinerung der Anzeige. Neu eingeblendete Steuerelemente (hier ein „Hamburger“-Menü) werden automatisch identifiziert und hervorgehoben.

Im Beispiel dient eine mit W3.CSS (W3Schools, 2021) entwickelte Webseite als Grundlage. Zur Steuerung der Sichtbarkeit in Abhängigkeit von der Anzeigegröße definiert das Framework CSS-Klassen, die das Verhalten der CSS-Eigenschaft `display` festlegen. Die Anwendung dieser Klassen basiert auf Medienabfragen (Media Queries). Verweist ein HTML-Element beispielsweise auf die Klasse `w3-hide-large`, wird das Element ab einer durch die mit dieser Klasse definierte Bildschirmgröße ausgeblendet.

Für die Implementierung einer Webseite ist die Kenntnis dieser Klassen erforderlich. Diese Kenntnis lässt sich auch zum Erklären des responsiven Verhaltens einsetzen, auch wenn die genauen Parameter der Media Query unbekannt sind. Durch Analyse der Klassenverweise lässt sich zumindest in Teilen systematisch ermitteln, welche Elemente auf einer Webseite auf eine Änderung der Displaygröße reagieren werden. Diese Information kann man verwenden, um mittels der gleichen Klassen komplementäre erklärende Elemente zu schaffen, die dann auftauchen, wenn andere verschwinden.

Der hier verwendete Ansatz nutzt Verweise auf CSS-Klassen eines gegebenen CSS-Frameworks. Diese sind nicht Framework-übergreifend normiert, was die Übertragung des Beispiels auf andere Frameworks erschwert. Eine Analyse der mit eben diesen Klassen verbundenen Media Query durch Inspektion des CSS Object Models zur Laufzeit könnte helfen, dieses Problem zu lösen. Die entsprechende Technologie war zum Zeitpunkt der Erstellung dieses Berichts aber nur experimentell nutzbar (Mozilla, 2021).

Die Aussagekraft der Analyse von Verweisen auf CSS-Klassen ist beschränkt. Im Falle von Webseiten können Änderungen der Benutzerschnittstelle beispielsweise auch durch JavaScript (mithin Anwendungslogik) oder den Webbrowser im Zuge der Darstellung erfolgen (z.B. Zeilenumbrüche).

6.3.2 Beispiel: Framework-unabhängiges Vorgehen

Besitzt man keine oder nur eingeschränkte Kenntnis des einer Anpassung zugrundeliegenden Regelwerks, dann lassen sich doch deren Auswirkungen auf die Benutzerschnittstelle an deren visuellen Repräsentationen (sofern abrufbar) beobachten. So kann im Fall von Webseiten die Überprüfung von Änderungen der durch CSS berechneten Stileigenschaften von HTML-Elementen Hinweise auf Aspekte einer Anpassung liefern.

Im in Abbildung 2 gezeigten Falle werden (durch Zugriff auf den „Computed Style“) nach einer Änderung der Media Query die Werte der CSS-Eigenschaft `display` an ausgewählten Elementen überprüft. Liegt eine Änderung vor, so werden, dem Prinzip des Highlightings folgend, die Elemente optisch hervorgehoben – hier durch eine temporäre Änderung des Rahmens.

Der im Beispiel verwendete Ansatz berücksichtigt nicht Änderungen, die keine Wertänderung an der visuellen Repräsentation eines Elements nach sich ziehen. Solche Änderungen können beispielsweise

implizit durch Änderungen übergeordneter Komponenten erfolgen und werden erst im Zuge der Berechnung der Darstellung (hier durch den Webbrowser) wirksam.

Durch das ausschließliche Beobachten presentationsrelevanter Merkmale (wie im Beispiel der Sichtbarkeit) ist eine automatische Unterscheidung bedienrelevanter und nicht bedienrelevanter Änderungen schwer leistbar. Zur Folge hat dies, dass beispielsweise Änderungen an dekorativen Elementen (die durchaus auch der Benutzerführung dienen können) in gleicher Weise behandelt werden wie das Erscheinen eines Steuerelements.

Weiterhin kann eine einzige Änderung der Benutzerschnittstelle unter Umständen zahlreiche Änderungen in der Darstellung nach sich ziehen. Sind diese Änderungen kleinteilig, kann das beschriebene Vorgehen die Komplexität der Darstellung erhöhen und die Wahrnehmungskonstanz beeinträchtigen.

7 Zusammenfassung

Die Adaption grafischer Benutzerschnittstellen bietet einen Weg, um veränderliche Aspekte des Nutzungskontexts in die Bedienung einer Anwendung einfließen zu lassen und so deren Gebrauchstauglichkeit zu stärken.

In diesem Bericht wurde thematisiert, dass eine Adaption den Benutzer aber auch verwirren und so die Gebrauchstauglichkeit beeinträchtigen kann. Untersuchungsgegenstand waren Faktoren, welche eine Veränderung der Verständlichkeit der Benutzerschnittstelle aufgrund einer Adaption bewirken können.

Einschlägige Literatur betont die besondere Bedeutung der Komplexität einer Benutzerschnittstelle und des mentalen Modells, das Benutzer zu deren adaptivem Verhalten entwickeln. Dessen individuelle Prägung sowie die Vielfalt möglicher Nutzungskontexte behindern eine Normierung adaptiven Verhaltens, was sich in eher allgemein gehaltenen Empfehlungen gängiger Normen ausdrückt. Die Literatur kann aber dazu herangezogen werden, um Empfehlungen zur Gestaltung nachvollziehbaren adaptiven Verhaltens zu formulieren. Implizite Maßnahmen sollen negative Effekte einer Adaption eingrenzen durch Berücksichtigung der Erwartungen des Benutzers bezüglich einer Adaption. Hinweise zu eben diesen Erwartungen lassen sich beispielsweise ableiten aus Erfahrungswerten des Benutzers zu adaptivem Verhalten in Verbindung mit Aspekten der menschlichen Wahrnehmung von Änderungen, etwa von Aufmerksamkeit und Wahrnehmungskonstanz. Explizite Maßnahmen können ergänzend wirken, um beispielsweise in Form von visuellen oder textuellen Hinweisen die Adaption zu erklären.

Die Umsetzung adaptiven Verhaltens wird technisch heute gängig unterstützt. Um dieses Verhalten – beispielsweise durch Erklärungen – nachvollziehbar zu realisieren, sind aber weitere Schritte seitens der Entwickler erforderlich. Ausgewählte Beispiele demonstrierten Framework-abhängiges und -unabhängiges Vorgehen. Vielversprechend erscheint ein modellbasiertes Vorgehen, wie Beispiele aus der Avionik illustrieren.

Die Ausführungen in diesem Bericht gewähren nur einen Einblick in die Thematik und sind nicht als abschließend zu betrachten. Insbesondere wurden im Sinne der Konzentration verschiedene Eingrenzungen – etwa in den Bereichen geräteübergreifender Interaktion und Lokalisierung – vorgenommen, die weitergehende Untersuchungen nahelegen. Gleiches gilt für die Umsetzung passiver und aktiver Maßnahmen; dieser nur angerissene Aspekt bietet weitere offene Fragestellungen in Entwurf und Implementierung adaptiven Verhaltens.

8 Danksagung

Die durch diesen Bericht dokumentierte Untersuchung profitierte stark vom interdisziplinären Austausch mit anderen Forschern. Besonders danken möchte ich an dieser Stelle Prof. Walter Mehl und Prof. Tilman Zitzmann (Fakultät Design, TH Nürnberg Georg Simon Ohm) und M. Sc. Vanessa Hönig, deren Hinweise insbesondere die hier erfolgte Diskussion menschlicher Wahrnehmung von Änderungen in einer grafischen Benutzerschnittstelle beeinflusste.

9 Literatur

- Adobe (2021). Adobe XD | UI/UX-Design und Zusammenarbeit. Adobe. Abgerufen am 25.03.2021 von <https://www.adobe.com/de/products/xd.html>
- Apple (2021a). Adaptivity and Layout - Visual Design - iOS - Human Interface Guidelines - Apple Developer. Apple Inc. Abgerufen am 26.03.2021 von <https://developer.apple.com/design/human-interface-guidelines/ios/visual-design/adaptivity-and-layout/>
- Apple (2021b). Xcode - Interface Builder - Apple Developer. Apple Inc. Abgerufen am 25.03.2021 von <https://developer.apple.com/xcode/interface-builder/>
- BBC (2021). Why are the BBC websites different outside the UK? BBC. Abgerufen am 12.03.2021 von <https://www.bbc.co.uk/contact/questions/help-using-bbc-services/international-differences>
- Bellotti, V., & Edwards, K. (2001). Intelligibility and Accountability: Human Considerations in Context-Aware Systems. *Human-Computer Interaction* 16(2), S. 193–212. L. Erlbaum Associates Inc., USA. DOI: http://dx.doi.org/10.1207/S15327051HCI16234_05
- Beolinguus (2021). scrutable : Wörterbuch / Dictionary (BEOLINGUS, TU Chemnitz). Abgerufen am 11.03.2021 von <https://dict.tu-chemnitz.de/dings.cgi?query=scrutable>
- Bevan, N., & Macleod, M. (1994). Usability measurement in context. *Behaviour & Information Technology*, 13(1–2), S. 132–145. Taylor & Francis. DOI: [10.1080/01449299408914592](https://doi.org/10.1080/01449299408914592)
- Butler, J., Holden, K., & Lidwell, W. (2010). *Universal Principles of Design*. Revised, Updated Edition. Rockport Publishers. ISBN: 978-1-61058-065-6
- 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. DOI: [10.1016/S0953-5438\(03\)00010-9](https://doi.org/10.1016/S0953-5438(03)00010-9)
- Cockburn, A., & Gutwin, C. (2008). A Predictive Model of Human Performance with Scrolling and Hierarchical Lists. *Human Computer Interaction* 24(3), S. 273–314.
- Craik, K. (1943). *The nature of explanation*. Cambridge University Press. ISBN 978-0521094450
- DX Heroes (2021). Practice Clickable Prototype | Developer Experience Knowledge Base. DX Heroes. Abgerufen am 25.03.2021 von <https://developerexperience.io/practices/clickable-prototype>
- Dey, A.K. (2001). Understanding and Using Context. *Personal and Ubiquitous Computing* 5(1), S. 4–7. Springer: London. DOI: [10.1007/s007790170019](https://doi.org/10.1007/s007790170019)
- Dictionary.com (2021). Definition of Scrutable at Dictionary.com. Stand Abgerufen am 11.03.2021 von <https://www.dictionary.com/browse/scrutable>
- DIN EN ISO 9241-11, Ergonomie der Mensch-System-Interaktion - Teil 11: Gebrauchstauglichkeit: Begriffe und Konzepte (ISO 9241-11:2018); Deutsche Fassung EN ISO 9241-11:2018
- DIN EN ISO 9241-110, Ergonomie der Mensch-System-Interaktion - Teil 110: Interaktionsprinzipien (ISO 9241-110:2020); Deutsche Fassung EN ISO 9241-110:2020
- DIN EN ISO 9241-125, Ergonomie der Mensch-System-Interaktion - Teil 125: Empfehlungen zur visuellen Informationsdarstellung (ISO 9241-125:2017); Deutsche Fassung EN ISO 9241-125:2017
- DIN EN ISO 9241-151, Ergonomie der Mensch-System-Interaktion - Teil 151: Leitlinien zur Gestaltung von Benutzungsschnittstellen für das World Wide Web (ISO 9241-151:2008); Deutsche Fassung EN ISO 9241-151:2008

- DIN EN ISO 9241-161, Ergonomie der Mensch-System-Interaktion - Teil 161: Leitfaden zu visuellen User-Interface-Elementen (ISO 9241-161:2016); Deutsche Fassung EN ISO 9241-161:2016
- FhG (2020). Intelligibility Meter ermöglicht objektive Messung und Anzeige von Sprachverständlichkeit bei der Medienproduktion. Pressemitteilung, Fraunhofer-Institut für Digitale Medientechnologie. Stand 10.12.2020. Abgerufen am 29.04.2021 von https://www.idmt.fraunhofer.de/de/Press_and_Media/press_releases/2020/Intelligibility_Meter_Nuendo11.html
- Gentner, D., & Nielsen, J. (1996). The Anti-Mac interface. *Communications of the ACM* 39(8), S. 70–82, ACM Press. DOI: <http://dx.doi.org/10.1145/232014.232032>
- Gibson, J.J. (1966). *The senses considered as perceptual systems*. Houghton Mifflin: Boston.
- Gustafson, A. (2011). *Adaptive Web Design: Crafting Rich Experiences with Progressive Enhancement*. Easy Readers, LLC; 1st edition. ISBN 978-0134216140. Online: <https://adaptivewebdesign.info/1st-edition/>
- Hagendorf, H., Krummenacher, J., Müller, H.J., & Schubert, T. (2011). *Wahrnehmung und Aufmerksamkeit*. Springer: Berlin Heidelberg. DOI: [10.1007/978-3-642-12710-6](https://doi.org/10.1007/978-3-642-12710-6)
- Hick, W.E. (1952). On the rate of gain of information. *Quarterly Journal of Experimental Psychology* 4, S. 11–26.
- Hodgdon, M. (2018). 8 Statistics that Prove Responsive Web Design is Essential to SEO. infront webworks, 14.05.2018. Abgerufen am 05.05.2020 von <https://www.infront.com/blog/8-statistics-that-prove-responsive-web-design-is-essential-to-seo/>
- Houben, S., Marquardt, N., Vermeulen, J., Klokmose, C., Schöning, J., Reiterer, H., & Holz, C. (2017). Opportunities and Challenges for Cross-Device Interactions in the Wild. *Interactions* 24(5), S. 58–63. ACM: New York, NY, USA. DOI: [10.1145/3121348](https://doi.org/10.1145/3121348)
- Kay, J. (2006). Scrutable Adaptation: Because We Can and Must. In Wade, V.P., Ashman, H., & Smyth, B. (Eds.), *Adaptive Hypermedia and Adaptive Web-Based Systems*, S. 11–19. Springer: Berlin Heidelberg. DOI: [10.1007/11768012_2](https://doi.org/10.1007/11768012_2)
- Lee, S.H., & Eom, H. (2015). Design of Driver-Vehicle Interface to Reduce Mode Confusion for Adaptive Cruise Control Systems. In *Adjunct Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (AutomotiveUI '15), S. 67–71. ACM: New York, NY, USA. DOI: [10.1145/2809730.2809757](https://doi.org/10.1145/2809730.2809757)
- Lek, R. (2014). User Experience Report: Is Responsive Design the Answer to the Growth in Mobile Devices? Usabilla B.V. Stand 20.03.2014. Abgerufen am 10.03.2021 von <https://usabilla.com/blog/user-experience-report-responsive-design-answer-growth-mobile-devices/>
- Marcotte, E. (2011). *Responsive Webdesign*. 1. Auflage. A Book Apart: New York. ISBN: 978-0-9844425-7-7
- Maurer, M.E., Hausen, D., De Luca, A., & Hussmann, H. (2010). In Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries (NordiCHI '10), S. 739–742. DOI: [10.1145/1868914.1869018](https://doi.org/10.1145/1868914.1869018)
- Mozilla (2021). CSS Object Model (CSSOM). Mozilla. Abgerufen am 19.04.2021 von https://developer.mozilla.org/en-US/docs/Web/API/CSS_Object_Model
- Muir, B.M. (1994) Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in automated systems, *ERGONOMICS*, 37(11), S. 1905–1922. DOI: [10.1080/00140139408964957](https://doi.org/10.1080/00140139408964957)
- Myers, D.G. (2014). *Psychologie*. 3. Auflage. Springer: Berlin Heidelberg. DOI: [10.1007/978-3-642-40782-6](https://doi.org/10.1007/978-3-642-40782-6)
- Nielsen, J. (2010). Mental Models. Nielsen Norman Group. Stand: 17.10.2010. Abgerufen am 16.03.2021 von <https://www.nngroup.com/articles/mental-models/>
- Norman, D.A. (1983). Some Observations on Mental Models. In *Mental Models*, S. 7–14. Psychology Press. DOI: [10.4324/9781315802725-5](https://doi.org/10.4324/9781315802725-5)
- Roth, S.P., Tuch, A.N., Mekler, E.D., Bargas, J.A., & Opwis, A.K. (2013). Location matters, especially for non-salient features—An eye-tracking study on the effects of web object placement on different types of websites. *International Journal of Human-Computer Studies* 71(3), S. 228–235. Elsevier. DOI: [10.1016/j.ijhcs.2012.09.001](https://doi.org/10.1016/j.ijhcs.2012.09.001)

- Sarter, N.B., Woods, D.D., & Billings, C.E. (1997). Automation Surprises. In Savendy, G. (Ed.), *Handbook of Human Factors and Ergonomics* (2nd Ed.), S. 1926-1943. Wiley, NY, USA. ISBN 978-0471116905
- Vermeulen, J. (2014). *Designing for Intelligibility and Control in Ubiquitous Computing Environments*. Doctoral Dissertation. Maastricht University, Faculty of Sciences. Abgerufen am 20.03.2021 von <http://jovermeulen.com/Research/PhDDissertation2014>
- Vermeulen, J., Luyten, K., & Coninx, K. (2013). Intelligibility Required: How to Make Us Look Smart Again. In *Proceedings of Romanian National Conference of Human-Computer Interaction (RoCHI)*, S. 1–8. MatrixRom Publishing House. Online: <http://hcibib.org/bibtoc.cgi?file=bibdata/RoCHI13.bib>
- W3Schools (2021). W3.CSS Home. Refsnes Data. Abgerufen am 29.04.2021 von <https://www.w3schools.com/w3css/>
- W3Techs (2021). Usage Statistics and Market Share of Content Management Systems, March 2021. Q-Success. Stand: 17.03.2021. Abgerufen am 17.03.2021 von https://w3techs.com/technologies/overview/content_management
- Weiser, M. (1999). The computer for the 21st century. *ACM SIGMOBILE Mobile Computing and Communications Review* 3(3), S. 3–11. ACM: New York, NY, USA. DOI: [10.1145/329124.329126](https://doi.org/10.1145/329124.329126)
- Wikipedia (2020a). Lokalisierung (Softwareentwicklung). In Wikipedia, Die freie Enzyklopädie. Stand: 18.06.2020. Abgerufen am 12.03.2021 von [https://de.wikipedia.org/w/index.php?title=Lokalisierung_\(Softwareentwicklung\)&oldid=201081332](https://de.wikipedia.org/w/index.php?title=Lokalisierung_(Softwareentwicklung)&oldid=201081332)
- Wikipedia (2020b). Intelligibility (communication). In *Wikipedia, The Free Encyclopedia*. Stand 21.20.2021. Abgerufen am 29.04.2021 von [https://en.wikipedia.org/w/index.php?title=Intelligibility_\(communication\)&oldid=1008061969](https://en.wikipedia.org/w/index.php?title=Intelligibility_(communication)&oldid=1008061969)
- WordPress (2021). Template Hierarchy | Theme Developer Handbook | WordPress Developer Resources. WordPress.org. Abgerufen am 17.03.2021 von <https://developer.wordpress.org/themes/basics/template-hierarchy/>
- Wroblewski, L. (2011). *Mobile First. A Book Apart*. Online: <http://mobile-first.abookapart.com/>