

## 2 Definition der zentralen Begriffe

### 2.1 Barrierefreiheit

Der Begriff *Barrierefreiheit* ist typischerweise normativ bestimmt:

Barrierefrei sind bauliche und sonstige Anlagen, Verkehrsmittel, technische Gebrauchsgegenstände, Systeme der Informationsverarbeitung, akustische und visuelle Informationsquellen und Kommunikationseinrichtungen sowie andere gestaltete Lebensbereiche, wenn sie für behinderte Menschen in der allgemein üblichen Weise, ohne besondere Erschwernis und grundsätzlich ohne fremde Hilfe zugänglich und nutzbar sind. (BMJV 2002)

Barrierefreie Produkte können von möglichst allen Menschen in jedem Alter mit unterschiedlichen Fähigkeiten weitgehend gleichberechtigt und ohne Assistenz bestimmungsgemäß benutzt werden (DIN 2002: Nr. Pkt. 2.3), d.h. sie sind für alle Benutzer uneingeschränkt und ohne (menschliche) Assistenz bedienbar. *Alle Benutzer* schließt insbesondere und ausdrücklich Menschen mit Einschränkungen sensorischer, motorischer oder kognitiver Fähigkeiten mit ein. Der Kreis der Benutzer kann dabei variieren und ist nicht gleichzusetzen mit allen Menschen. Jedoch sind unter Benutzern auch jene eingeschlossen, die das System zwar derzeit nicht aktiv benutzen, jedoch potenziell Benutzer sein können, weil bspw. die Anwendungsdomäne für sie relevant ist. In dieser Forschungsarbeit werden „Barrierefreiheit“, „Zugänglichkeit“ bzw. „Accessibility“ synonym verwendet<sup>3</sup>. Barrierefreiheit im Internet (*Web Accessibility*) wendet diese Anforderungen auf das Web an.

Web Accessibility means that people with disabilities can use the Web. More specifically, Web accessibility means that people with disabilities can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web. (Henry 2005)

Barrierefreie Bedienung umfasst die umfassende Wahrnehmung und Kontrolle von Informationen durch den Benutzer. Für die Bedienung relevante Informationen sind wahrnehmbar, operabel und verständlich (vgl. Abbildung 2-1, W3C 2008b). Des Weiteren ist Robustheit gefordert, d.h. die Bedienung wird durch unterschiedliche Software bzw. Softwareversionen unterstützt.

Im Rahmen dieser Forschungsarbeit wird definiert: *Barrierefreiheit* bzw. *Zugänglichkeit* oder *Accessibility* ist die Eigenschaft eines webbasierten technischen Systems und insbesondere seiner Benutzungsschnittstelle, für alle Benutzer – Personen mit sensorischen, motorischen oder kognitiven Einschränkungen

---

3 Im Anglo-Amerikanischen ist für „Barrierefreiheit“ der Begriff „Accessibility“ üblich. Die Rückübersetzung von „Accessibility“ zu „Zugänglichkeit“ ist insbesondere in der Literatur zum World Wide Web verbreitet. Zum Beispiel wird in der autorisierten Übersetzung der *Web Content Accessibility Guidelines* (W3C 2008b) ins Deutsche (W3C 2009) „Accessibility“ durchgängig als „Zugänglichkeit“ übersetzt. Die BITV (Bundesministerium für Arbeit und Soziales 2011) ist in großen Teilen inhaltlich adäquat zu den WCAG und nutzt sowohl „barrierefrei“ als auch „zugänglich“.

eingeschlossen – die uneingeschränkte und autonome Bedienung zu unterstützen. Für die Bedienung relevante Informationen sind wahrnehmbar, manipulierbar und verständlich.

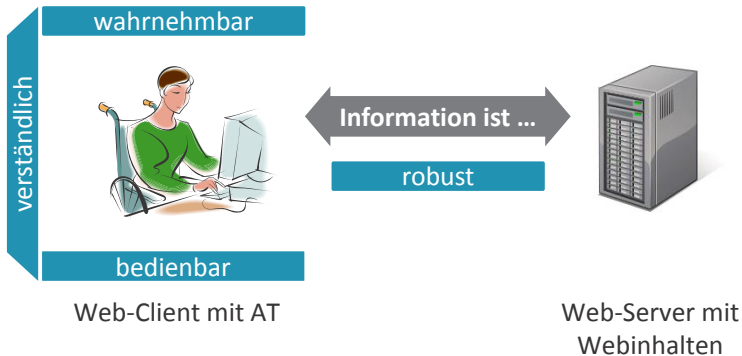


Abbildung 2-1: Prinzipien der WCAG 2.0 (W3C 2008b)

## 2.2 Benutzungsschnittstelle und Weboberfläche

Als *Benutzungsschnittstelle* oder auch *User Interface* (UI) wird in einem Mensch-Maschine-System allgemein das Subsystem verstanden, dass der Interaktion zwischen Mensch bzw. Benutzer und technischem System dient. Unter *Benutzungsschnittstelle* fallen „alle Bestandteile eines interaktiven Systems (Software oder Hardware), die Informationen und Steuerelemente zur Verfügung stellen, die für den Benutzer notwendig sind, um eine bestimmte Arbeitsaufgabe mit dem interaktiven System zu erledigen.“ (DIN 2008a: 7). Eine Benutzungsschnittstelle erlaubt dem Benutzer, das technische System zu bedienen, zu beobachten und damit verbundene Prozesse zu steuern. Benutzungsschnittstellen unterstützen die Eingabe (input) und Ausgabe (output) von Daten bzw. Information. Information ist Wissen in Bezug auf Objekte wie Fakten, Dinge, Prozesse oder Ideen einschließlich Konzepten, das in einem bestimmten Kontext eine spezifische Bedeutung besitzt:

„Knowledge concerning objects, such as facts, events, things, processes or ideas, including concepts, that within a certain context has a particular meaning“ (ISO 1993: 01.01.01)

Daten sind eine wiederholt interpretierbare formalisierte Repräsentation von Information, geeignet zur Kommunikation, Interpretation oder Verarbeitung:

„A reinterpretable representation of *information* in a formalized manner, suitable for communication, interpretation, or processing” (ISO 1993: 01.01.02, Hervorh. im O.)

Informationen sind Daten, die einen (dem Benutzer bekannten) Kontext besitzen. Daten sind eine Repräsentation von Information, die für Maschinen bearbeitbar und lesbar ist. Weiterhin erlaubt die Eingabe dem Benutzer die Manipulation des Systems. Die Ausgabe dient der Präsentation von Information. Insbesondere übermittelt sie dem Benutzer die Effekte der Manipulation. Benutzungsschnittstellen werden alternativ als *Benutzerschnittstelle*, *Bediensystem* oder *Bedienschnittstelle* bezeichnet. Da die Benutzungsschnittstelle typischerweise eine Einheit aus Hard- und Software ist, wird von der *Gesellschaft für Informatik* allgemein der Begriff *Ueware* vorgeschlagen:

„Ueware umfasst alle der Nutzung einer Maschine oder Anlage dienenden Hard- und Software-Komponenten.“ (Zühlke 2002)

Ueware grenzt das Bediensystem vom Hardwaresystem sowie dem Softwaresystem ab. Motivation für diese Abgrenzung ist auch der zunehmende Ressourceneinsatz (vgl. Abbildung 2-2) für die Realisierung der Benutzungsschnittstelle in der Produktentwicklung.

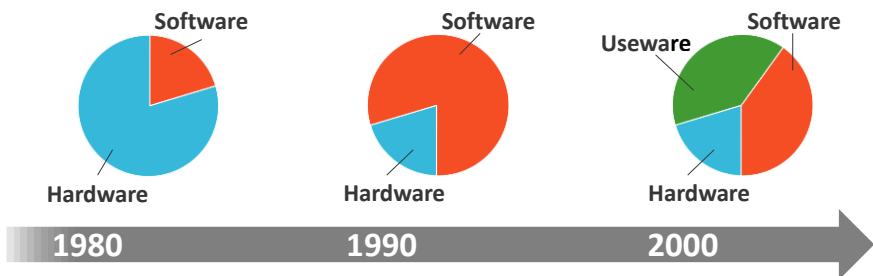


Abbildung 2-2: Entwicklungstendenz der Kostenanteile in der Produktgestaltung (Reuther 2003: Abb. 2-10, zit. Zühlke 2002, Layout d. Verfasser angepasst)

Myers und Rosson stellten 1992 eine Studie vor (Myers & Rosson 1992: 195–199), nach der 48 % des Sourcecode, 45 % der Entwicklungszeit, 50 % der Implementationszeit und 37 % des Wartungsaufwands für die Benutzungsschnittstelle aufgewendet werden. Siedersleben schätzt für die Realisierung der Benutzeroberfläche den Anteil am Gesamtaufwand bei interaktiven Systemen auf häufig 60 %, manchmal sogar 80 % (Siedersleben 2004: 235). Auch Petrasch schätzt diesen Anteil – bedingt durch steigende Anforderungen und die wachsende Zahl

heterogener Ein- und Ausgabegeräte – auf mindestens 50 % der Gesamtentwicklungszeit (Petrasch 2007: 5).

*Grafische Benutzungsoberflächen* (*Graphical User Interface* – GUI) sind Benutzungsschnittstellen für IKT-Systeme, die die Interaktion mittels grafischer Symbole unterstützen. Sie sind typisch für die Bedienung von Computern, Tablets und Smartphones. Alternative Bezeichnungen für *grafische Benutzungsoberfläche* sind *grafische Benutzeroberfläche* oder einfach *Bedienoberfläche*. Grafische Benutzungsoberflächen wurden zuerst als *lokale Oberflächen* verwendet. Lokale Oberflächen (auch *native Clients* genannt) sind die Benutzungsschnittstellen für Anwendungen, deren anwendungsspezifische Software lokal auf dem System vorhanden ist (Siedersleben 2004: 235). Im Rahmen dieser Forschungsarbeit werden Bedienoberflächen auf der Basis von Webtechnologien untersucht und *Weboberflächen* genannt. Im Gegensatz zu lokalen Oberflächen benötigen Weboberflächen keine Installation und können weltweit mit Hilfe eines Browsers bedient werden. Aus Sicht des Benutzers gelten lokale Oberflächen oft als angenehmer zu bedienen, da sie schneller reagieren und robuster sind. Auch werden Eingabefehler schneller abgefangen und die Bedienungsmöglichkeiten sind vielfältiger (Siedersleben 2004: 235).

### 2.3 Webanwendung

Eine *Webanwendung* bzw. *Webapplikation* (*Web Application*) ist ein Anwendungsprogramm, das grundlegend nach dem Client-Server-Prinzip gestaltet ist. Auf dem Client wird die Webanwendung typischerweise in einem Browser oder allgemeiner Benutzeragenten ausgeführt. Der Server ist für die Anwendungsfunktionalität und die Datenhaltung zuständig. Die Kommunikation zwischen Client und Server erfolgt auf Basis des *Hypertext Transfer Protocol* (HTTP) oder vergleichbarer bzw. typespezifischer Protokolle über das Internet oder ein Intranet.

Die technologische Plattform, auf die diese Forschungsarbeit sich bezieht, ist die Webanwendung. Webanwendungen unterscheiden sich von lokalen Anwendungen durch die Separation von Weboberfläche und Anwendungsserver. Der Browser trennt als Abstraktionsschicht die Anwendung vom lokalen System und unterstützt eine vom Betriebssystem unabhängige Entwicklung. Der technologischen Begrenzung der Webtechnologien steht der Vorteil ihrer Standardisierung gegenüber. Der Aufwand für die Softwaredistribution und -aktualisierung wird minimiert und eine lokale Installation entfällt. Inkompatibilitäten verschiedener Browser können jedoch die Anforderungen an die Entwicklung steigern. Weitere Nachteile stellen die Restriktion des Zugriffs auf native Ressourcen des lokalen Systems dar – bspw. Programmierschnittstellen – sowie Verzögerungen in der Bedienung durch das Nachladen von Webseiten – insbesondere bei Netzwerkverbindungen mit geringer Bandbreite und hohen Latenzzeiten. Der Wegfall

einer Installation der Weboberfläche unterstützt bei Webanwendungen kurze Zyklen der Auslieferung und die kontinuierliche Aktualisierung bzw. häufige Anpassung. Webanwendungen erweitern den für klassische Webseiten typischen Fokus auf der Darstellung von Information mit den Möglichkeiten einer komplexen Geschäftslogik. Um das Interaktionsverhalten durch das partielle Nachladen von Inhalten weiter zu verbessern, werden Techniken wie *JavaScript* (standardisiert als *ECMAScript*, ISO 2011) und *Ajax* (*Asynchronous JavaScript and XML*, Garrett 2005) genutzt (Wolffgang 2012: 41).

Die einschlägige Literatur unterscheidet häufig zwischen Webanwendungen und *Rich Internet Applications* (RIA, Allaire 2002: 2). Die Bezeichnung RIA bezieht sich auf Webanwendungen mit hochgradig clientseitig generierten Inhalten auf der Basis von Ajax- und JavaScript-Techniken. Ziel ist ein zu lokalen Anwendungen vergleichbares Verhalten der Weboberfläche, sodass auch klassische Softwareanwendungen wie Textverarbeitung bzw. Tabellenkalkulation browserbasiert realisiert werden können. Rich Internet Applications bieten damit die Möglichkeit, die Vorteile lokaler Anwendungen, wie schnelle Reaktion und flexible Interaktionsmöglichkeiten mit denen von webbasierten Anwendungen wie einfache Verfügbarkeit und Datensicherung zu verbinden. Sie verwenden oft eigene Plugins – bspw. *Silverlight* (Microsoft 2014b) – als Laufzeitumgebung im Browser oder auch entsprechende JavaScript-Frameworks. Die Grenze zwischen Webanwendungen und RIAs ist fließend, da Ajax und JavaScript auch durch klassische Plattformen der Webentwicklung wie *PHP* (The PHP Group 2014), *ASP.NET* (*Active Server Pages.NET*, Microsoft 2012) oder *JSF* (*Java Server Faces*, Oracle 2012) unterstützt werden.

## 2.4 Benutzungszentriertes Design

Benutzungszentriertes Design ist eine Methodologie zur Verbindung von HCI-Design mit Software Engineering. Sie wurde in den 1990er Jahren durch Constantine und Lockwood entwickelt (Constantine & Lockwood 1999). Benutzungszentrierte Entwicklung ist ein systematischer, modellgetriebener Prozess für Benutzungsschnittstellen, dessen Fokus auf der Benutzung des Systems liegt. Ziel ist es, das „kleinste“ und einfachste System zu entwickeln, das direkt und vollständig die Bedienungsaufgaben des Benutzers unterstützt (vgl. Constantine & Lockwood 2002: 43). Abstrakte Modelle beschreiben Benutzerrollen, Aufgaben und Inhalte der Benutzungsschnittstelle.

## 2.5 Modellgetriebene Entwicklung von Benutzungsschnittstellen

Die modellgetriebene Entwicklung von Benutzungsschnittstellen nutzt abstrakte Modelle zur Deklaration der Benutzungsschnittstelle, die anschließend durch Modellcompiler in ausführbaren Code übersetzt werden. Die Beschreibung von

Benutzungsschnittstellen umfasst zwei wesentliche Aspekte – den Interaktionsaspekt und den Präsentationsaspekt (Müller 2003: 18). Der Interaktionsaspekt beschreibt was dargestellt werden soll: die Ein- und Ausgaben der Interaktion mit dem Benutzer sowie ihre Logik. Der Präsentationsaspekt beschreibt wie etwas dargestellt werden soll: die Gestaltung der Benutzungsschnittstelle, ihr Layout, die Anordnung von UI-Elementen etc. Nach Schlungbaum erfüllt eine modellbasierte Entwicklungsumgebung für Benutzungsschnittstellen (*Model-based User Interface Development Environment*, MBUIDE) zwei notwendige Kriterien:

1. “MBUIDE must include a high-level, abstract and explicitly represented (declarative) model about the interactive system to be developed (either a task model or a domain model or both)”
2. “MBUIDE must exploit a clear and computer-supported relation from (1) to the desired and running UI. That means that there is some kind of automatic transformation like knowledge-based generation or simple compilation to implement the running UI.” (Schlungbaum 1996: 4)

Dieser Ansatz der modellbasierten Entwicklung hat sich als sehr anspruchsvoll erwiesen, da die vollständige Deklaration im Modell und die anschließende automatisierte Übersetzung sowohl an den Entwickler als auch den Compiler sehr hohe Ansprüche stellen, die durch die verfügbaren Werkzeuge bis heute nicht erfüllt werden. Die Folge war, dass in der Praxis die entworfenen Modelle meist nur für Zwecke der Dokumentation verwendet wurden. In den vergangenen Jahren wurden deshalb modellgetriebene (model-driven) Konzepte entwickelt, die den Entwurfsprozess begleiten und vorwärtstreiben. Dabei wird oft automatisch kompilierter und manuell geschriebener Code zusammen entwickelt, sodass die Modelle keine vollständige Deklaration leisten müssen und viele Standardfunktionalitäten auch ohne Modellierung implementiert werden. Die Modelle beschreiben in diesem Fall dann vorrangig die Objekt- und Beziehungsstruktur der zugrunde liegenden Logik – bspw. mit einem Klassenmodell. Dadurch können Modelle den Entwurf, die Verifikation und die Evaluation der Benutzungsschnittstelle unterstützen. Die Modelle der Benutzungsschnittstelle tragen auch dazu bei, Zweck, Zusammenhänge und -wirken der Entitäten zu erfassen und vorhersehbar zu machen (vgl. Charwat 1994).

Barrierefreiheit im virtuellen Raum  
Benutzungszentrierte und modellgetriebene  
Entwicklung von Weboberflächen

Vieritz, H.

2015, XX, 207 S. 86 Abb., Softcover

ISBN: 978-3-658-10703-1