

BACHELORARBEIT

im Studiengang Bachelor Informatik

Accessibility im Modern Web

Ausgeführt von: Bernhard Posselt

Personenkennzeichen: 1010257029

Begutachter: Dipl.-Ing. Mag. Dr. Michael Tesar

Wien, 8. Jänner 2013

Eidesstattliche Erklärung

„Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbständig angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher weder in gleicher noch in ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht. Ich versichere, dass die abgegebene Version jener im Uploadtool entspricht.“

Ort, Datum

Unterschrift

Kurzfassung

In dieser Arbeit wird ein Überblick über moderne Web-Technologien im Bereich Accessibility gegeben. Immer mehr öffentliche Einrichtungen stellen im Rahmen des *e-governments* heutzutage einen Zugang zu wichtigen Informationen und Services im Web bereit. Dadurch muss dieser Zugang universell erreichbar und nutzbar sein. Um eine Lösung für dieses Problem zu finden, wird ein Blick auf die Empfehlungen der W3C geworfen. Auch auf neuere Techniken wie ARIA, die erst in einer *Candidate Recommendation* vorhanden sind, wird eingegangen. Das Ziel dieser Arbeit ist, einen guten Überblick über den heutigen Stand der Accessibility im Web zu bieten und die Vorteile der Nutzung von Accessibility Methoden hervorstreichen.

Schlagwörter: Accessibility, Web, ARIA, HTML5

Abstract

This thesis will present an overview over Accessibility techniques used in the modern web. Over the last years, the usage of the web to present and access information and services of public institutions has increased dramatically. This requires services and information to be accessible for every citizen. To find a solution for this problem, this thesis will look at the W3C's recommendations and guidelines. Also newer techniques like ARIA and the Role Attribute, which are currently in the *Candidate Recommendation* phase, will be part of it. The goal of this thesis is to present the current status of Accessibility in the web and to show the advantages of using these techniques.

Keywords: Accessibility, Web, ARIA, HTML5

Inhaltsverzeichnis

1	Einführung	1
1.1	Problemstellung	1
1.2	Lösungsansatz	2
1.3	Aufbau	2
1.4	Begriffe	2
2	Arten von körperlichen Einschränkungen oder Lernschwierigkeiten und assistive Technologien	4
2.1	Personen mit Sehschwächen	4
2.2	Personen mit Hörschwächen	5
2.3	Personen mit motorischen Schwächen	5
2.4	Personen mit Lernschwierigkeiten	5
3	WCAG	6
3.1	WCAG Version 1.0	6
3.1.1	Success Criteria	6
3.1.2	Wahrnehmbarkeit	6
3.1.3	Bedienbarkeit	7
3.1.4	Verständlichkeit	8
3.1.5	Robustheit	9
3.2	WCAG Version 2.0	10
3.2.1	Success Criteria	10
3.2.2	Wahrnehmbarkeit	11
3.2.3	Bedienbarkeit	11
3.2.4	Verständlichkeit	12
3.2.5	Robustheit	13
3.3	Unterschiede zwischen WCAG Version 2.0 und 1.0	14
3.3.1	Success Criteria	14
3.3.2	Wahrnehmbarkeit	14
3.3.3	Bedienbarkeit	14
3.3.4	Verständlichkeit	14
3.3.5	Robustheit	14
4	ARIA	15
4.1	Roles	15
4.1.1	ARIA Roles Beispiele	16
4.2	States and Properties	17
4.2.1	Widget Attributes	17
4.2.2	Live Regions	17
4.2.3	Drag and Drop Attributes	18

4.2.4 Relationship Attributes	18
5 Zusammenfassung	19
6 Ausblick	20
Literaturverzeichnis	21
Abbildungsverzeichnis	23
Abkürzungsverzeichnis	24

1 Einführung

Das Web spielt im öffentlichen Bereich eine immer größere Rolle: Viele Services und Informationen werden bereits über eigens dafür erstellte Portale angeboten. Daraus folgt, dass das öffentliche Leben immer stärker mit dem Web verwoben wird. Wie auch bei anderen öffentlichen Einrichtungen muss ein universeller Zugang für alle Bürger ermöglicht werden, auch für jene, die mit körperlichen Einschränkungen oder Lernschwierigkeiten leben müssen, z. B. Blinde oder Personen mit eingeschränkten motorischen Fähigkeiten. Solange dies nicht vollständig möglich ist, muss als Ersatz eine zusätzliche Einrichtung bereitstehen, die äquivalente Möglichkeiten bietet [1, S. 8]. Es ist also auch im wirtschaftlichen Interesse öffentlicher Einrichtungen, ihre Services und Informationen barrierefrei anzubieten.

Dadurch, dass diese Gruppe eine Minorität in der Bevölkerung stellt und das Web auf oft mit speziellen Ein- oder Ausgabegeräten verwendet - z. B. blinde Personen verwenden Screen-Reader oder Brailles - muss oft ein zusätzlicher Aufwand bei der Erstellung von Webseiten betrieben werden. Ja öfters wird sogar ganz darauf vergessen[2, S. 7]. Roman Mauerhofer erfasst dieses Problem mit einem sehr treffenden Vergleich: „Es ist beispielsweise so, als ob ein Architekt bei der Planung eines Bahnhofes den Einbau von Aufzügen vergisst.“[2, S. 7]

Diese Gruppe macht jedoch keinen ganz so geringen Prozentanteil der Bevölkerung aus wie öfters angenommen. Laut einem Bericht des U.S. Census Bureau aus dem Jahre 2000 leben alleine in den USA ca. 49,7 Millionen Menschen mit körperlichen Einschränkungen und Lernschwierigkeiten (entspricht ca. 20% der amerikanischen Bevölkerung), davon haben 42,9 Millionen eine schwere Einschränkung und 6,8 Millionen haben eine so gravierende Einschränkung, dass sie Hilfe in ihrem alltäglichem Leben brauchen [3, S. 1]. Laut der „World Health Organization“ wird der weltweite Anteil an Personen mit körperlichen Einschränkungen oder Lernschwierigkeiten auf 500 bis 600 Millionen Menschen geschätzt [4].

Auch die UN erkennt eine immer größer werdende Wichtigkeit bezüglich einer angemessenen Bereitstellung von Informationen für körperlich eingeschränkten Personen und Personen mit Lernschwierigkeiten. Dies spiegelt sich in der „UN Convention of Rights for Persons with Disabilities“ wider, welche am 30. März 2007 unterzeichnet wurde. Sie erhielt die „meisten Unterschriften an einem Eröffnungstag in der Geschichte der UN Konventionen“[5].

Jedoch kann es nicht nur aus den oben genannten Gründen erforderlich sein, einen barrierefreien Zugang bereitzustellen, sondern es wird sogar oft gesetzlich vorgeschrieben. In den USA beispielsweise, gibt es dafür ein eigenes Gesetz, den „Americans with Disabilities Act“ aus dem Jahre 1990 (ADA) und den Abschnitt 508 aus dem „Rehabilitation Act“(1973) [6, S. 288-289]. Auch in Europa sind einige ähnliche Regelungen vorhanden [2, S. 7], doch eines haben die Meisten gemeinsam: Sie basieren nahezu alle zu einem großen Teil auf den Richtlinien der WAI (Web Accessibility Initiative), den WCAG (Web Content Accessibility Guidelines) [6, S. 289] [2, S. 7].

1.1 Problemstellung

Nicht nur in der EU wird Accessibility künftig eine größere Rolle spielen [7, Abschnitt EU], sondern auch in der Privatwirtschaft. Für viele ProgrammiererInnen stellt jedoch die Implementation und

das Testen von Accessibility Techniken einen nicht zu unterschätzenden Mehraufwand dar [1, S. 27] und dadurch werden ihnen, selbst wenn sie sie umsetzen wollen, oft Steine in den Weg gelegt. Dies erfolgt häufig in der Nennung nicht zu Ende gedachter Argumente wie „Barrierefreiheit bringt keine Vorteile“ [1, S. 28] oder „Behinderte gehören nicht zu unserer Zielgruppe“ [1, S. 31].

Ein weiteres Problem ist die geringe Verbreitung der Standards. Dies wurde in einer globalen Studie der *United Nations* aus dem Jahre 2006 festgestellt [8]: Nur drei von 100 getesteten Seiten erhielten die Bestnote [8, S. 7]. Dies ist unter anderem auf ein Fehlen folgender Punkte zurückzuführen [9, S. 13]:

- *Achtsamkeit*: Vergessen oder Übersehen von Fehlern und Problemen im Bereich Accessibility
- *Bildung*: Accessibility-Techniken sind unbekannt
- *Tool Support*: Techniken werden nicht von Programmierwerkzeugen unterstützt

1.2 Lösungsansatz

In dieser Bachelorarbeit soll ein genereller Überblick über die bestehenden und einsetzbaren Techniken gegeben werden, die hilfreich für Verbesserungen im Bereich Accessibility im Modern Web sind.

Dazu soll eine Übersicht über die vom W3C bereitgestellten Spezifikationen, den WCAG 1.0 [10], WCAG 2.0 [11] und ARIA [12] gegeben werden und außerdem soll auf die Unterschiede von WCAG 1.0 und WCAG 2.0 eingegangen werden.

Zusätzlich wird kurz auf die verschiedenen Arten von Einschränkungen eingegangen und die Probleme und verwendeten Hilfsgeräte werden aufgezeigt.

1.3 Aufbau

In Kapitel 2 wird ein kurzer Überblick über die häufigsten Einschränkungen gegeben und erklärt, welche assistiven Technologien verwendet werden können, um körperlich eingeschränkten Personen und Personen mit Lernschwierigkeiten beim Zugriff auf das Web zu helfen. Dies soll veranschaulichen, warum Webseiten für diese Geräte optimiert werden sollen.

In Kapitel 3 wird auf die Richtlinien des W3C, die WCAG, eingegangen und Entwicklung und Unterschiede zwischen der Version 1.0 und Version 2.0 betrachtet.

In Kapitel 4 wird ein Blick auf ARIA geworfen und Sinn und Einsatzzweck von *Roles*, *States* und *Properties* erläutert.

In Kapitel 5 wird eine Zusammenfassung der jetzigen Situation und ein Ausblick über kommende Richtlinien gegeben, welche sich derzeit noch im *Draft Status* befinden.

1.4 Begriffe

Da der Begriff „Menschen mit geistigen Behinderungen/Einschränkungen“ negativ behaftet ist, wird in dieser Arbeit stattdessen der Begriff „Menschen mit Lernschwierigkeiten“ verwendet. Dies ist gleich zu setzen mit den Abschnitten F70-F79 im ICD-10 [13]. Dazu zählen unter anderem Menschen mit Konzentrationsschwächen.

Der Begriff „Menschen mit körperliche Einschränkung“ wird synonym für Personen verwendet, welche durch eine Verletzung, Nichtfunktion oder Amputation eines Körperteiles Einschränkungen

im täglichen Leben erfahren müssen. Dazu zählen unter anderem Menschen mit Sehschwächen oder Blindheit und Menschen mit einer Fehlbildung der Finger oder Hände.

2 Arten von körperlichen Einschränkungen oder Lernschwierigkeiten und assistive Technologien

2.1 Personen mit Sehschwächen

Die Anzahl blinder Personen allein in den USA wird auf eine Million Personen und weltweit auf 38 Millionen Personen geschätzt[14, S. 1]. Als offiziell blind gilt eine Person, wenn sie unter 2% auf dem besseren der zwei Augen wahrnimmt [1, S. 12].

Jedoch umfasst diese Kategorie nicht nur blinde Personen, sondern auch Personen mit teilweisen Sehschwächen: Nimmt eine Person auf dem besten Auge weniger als 30% wahr, fällt sie bereits unter diese Kategorie[1, S. 12]. In Österreich beträgt die Anzahl der Personen mit Sehschwächen rund 318.000 Personen, das entspricht 3,9% der Bevölkerung [15, S. 13]. Es wird geschätzt, dass weltweit mindestens sechs Millionen Personen von so schweren Sehschwächen betroffen sind, dass für sie die Nutzung eines Computerbildschirmes unmöglich ist[16, S. 249].

Personen mit Sehschwächen haben die größten Probleme nicht mit den Eingabegeräten - die meisten Nutzer beherrschen das Tippen ohne auf die Tastatur zu schauen - sondern eher mit den Ausgabegeräten. Die am meisten genutzten assistiven Technologien in diesem Umfeld sind Screen-Reader und Brailles (siehe 2.1). Brailles und andere in Hardware umgesetzte Technologien sind jedoch meistens sehr teuer, da sie nur von einem geringen Prozentsatz der Bevölkerung benötigt werden und deren Erlernung, beispielsweise die der Brailleschrift, aufwändig und dementsprechend wenig verbreitet ist[16, S. 249-250]. Zusätzlich besteht die Braille aus vielen kleinen, mechanischen Teilen und ist deshalb besonders verschleißanfällig [17, S. 11].

Deshalb ziehen die Meisten auf Software basierende Lösungen vor, was dem Screen-Reader, etwa dem frei verfügbaren GNOME Orca [18], zu einer höheren Popularität verhilft[16, S. 249-250]. Ein

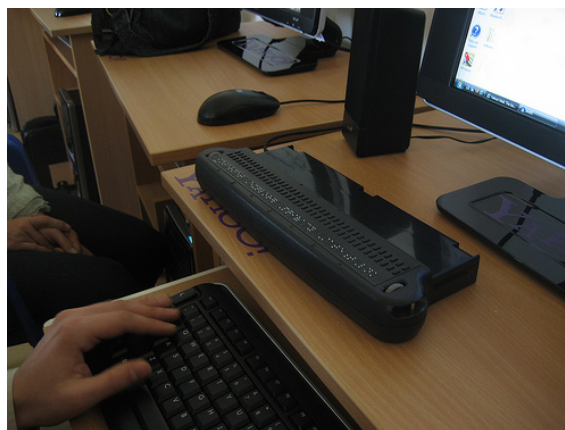


Abbildung 2.1: Beispiel einer Braille (©Philip Tellis, Creative Commons BY-SA)

Screen-Reader liest dem/der NutzerIn die Inhalte auf dem Bildschirm vor. Dies kann für sehende Personen als störend empfunden werden. Blinde Personen sind jedoch stärker auf auditive Reize ausgerichtet und kommen dementsprechend besser damit klar[17, S. 13].

Für diese Personen muss beim Erstellen von Webseiten besonders Folgendes beachtet werden: semantisch korrekter und sauberer HTML Code, der den Inhalt vom Layout trennt.[17, S. 13-15]

2.2 Personen mit Hörschwächen

Alleine in Österreich befinden sich ca. 202.000 Personen (2,5% der Bevölkerung) mit eingeschränkter Hörfähigkeit. Personen mit schweren Hörschwächen machen rund 2.000 Personen aus [15, S. 13-14]. Die größten Probleme entstehen für diese Gruppe, wenn eine Webseite eine bestimmte Information nur über akustische Signale wiedergibt. Dies kann z. B. ein einfaches Video oder ein Podcast sein, oder ein Warnsignalton. [17, S. 17][1, S. 20]

Für diese Personen muss vor allem Folgendes beachtet werden: Bereitstellung von Untertiteln bei Videos oder Transkriptionen von Podcasts und Verzicht auf „audio-only“-Lösungen. Das Anbieten von Videos mit Gebärdensprache kann nützlich sein, stellt jedoch einen hohen Aufwand dar, den jeder/jede AuftraggeberIn selbst für sich entscheiden muss. [17, S. 17][1, S. 20]

2.3 Personen mit motorischen Schwächen

Diese Gruppe macht laut dem U.S. Census Report 21,2 Millionen Personen aus. Das sind ca. 8,2 Prozent der Bevölkerung der USA [3, S. 1]. In Österreich beträgt deren Anzahl 1 Million bzw. 13% der Bevölkerung [15, S. 12]. Probleme mit der Computerbenutzung treten vor allem dann auf, wenn Personen ihre Hände nur eingeschränkt oder gar nicht verwenden können, beispielsweise Querschnittsgelähmte. Für diese muss eine alternative Steuerung des Computers angeboten werden. Darunter fallen z. B. Geräte, welche es den NutzerInnen erlauben, den Computer mit ihrer Zungenspitze zu bedienen.[17, S. 15-16]

Diese Art der Bedienung ist offensichtlich aufwändiger und schwieriger und profitiert deshalb am meisten von einer klaren Struktur der Seite und einer guten Bedienbarkeit mit der Tastatur.[17, S. 15-16][1, S. 18]

2.4 Personen mit Lernschwierigkeiten

Diese Gruppe macht laut dem U.S. Census Report 12,4 Millionen Personen aus. Das sind ca. 4,8 Prozent der Bevölkerung der USA [3, S. 1]. Zu dieser Gruppe gehören auch Analphabeten, welche etwa in Deutschland ca. vier Millionen Personen ausmachen (ca. 6,3% der deutschen Bevölkerung) [1, S. 19]. In Österreich sind 85.000 Personen bzw. 1% der Bevölkerung von Lernschwierigkeiten betroffen [15, S. 14].

Diese Personen haben vor allem Probleme mit zu komplizierten Sätzen oder einer fehlenden Struktur der Seite. Dadurch profitieren sie am meisten durch einfache und kurze Sätze, gute Strukturierung, den Einsatz von Bildern und einer Möglichkeit, sich Texte vorzulesen zu lassen, auch „Text-to-Speech“ genannt wird. [17, S. 18-19][1, S. 19]

3 WCAG

Die WCAG (Web Content Accessibility Guidelines) ist ein Dokument, welches vom W3C herausgegeben wurde um Webseiten auch für körperlich eingeschränkte Personen oder Personen mit Lernschwierigkeiten zugänglich zu machen.

Da sich viele Richtlinien mit der Optimierung des Inhaltes für weniger geläufige Plattformen beschäftigen, verbessert deren Anwendung auch die Plattformunabhängigkeit. Dies kommt vor allem Geräten zu Gute, die durch Formfaktor oder Einsatzgebiet in Größe oder Funktion eingeschränkt sind, beispielsweise Mobiltelefonen. [10, Abschnitt Abstract]

3.1 WCAG Version 1.0

Die Version 1.0 des Dokumentes wurde im Jahre 1999 finalisiert und beinhaltet 14 Richtlinien, welche jeweils Empfehlungen für WebentwicklerInnen enthalten. Diese Empfehlungen enthalten Unterpunkte, so genannte Checkpoints, die wiederum mit einer Priorität gekennzeichnet sind. [10, Abschnitt Abstract]

Für eine bessere Gegenüberstellung dieser Kriterien mit denen der WCAG 2.0 wird versucht, diese 14 Richtlinien der WCAG 1.0 in die gleiche Struktur wie die der WCAG 2.0 zu bringen.

3.1.1 Success Criteria

Die Kriterien, anhand derer die erfolgreiche Umsetzung der Richtlinien geprüft werden können, werden in drei verschiedene Kategorien aufgeteilt: *must*, *should* und *may* Kriterien, sprich Kriterien die erfüllt sein müssen, sollten und können. Anhand der Erfüllung dieser Kriterien lassen sich folgende Konformitätslevel definieren: [10, Abschnitt 4]

- A: Alle mit Priorität 1 angeführten Kriterien sind umgesetzt
- AA: Alle mit Priorität 1 und 2 angeführten Kriterien sind umgesetzt
- AAA: Alle mit Priorität 1, 2 und 3 angeführten Kriterien sind umgesetzt

Die deutsche BITV (Barrierefreie Informationstechnologie Verordnung), welche im „Bundesgesetz zur Gleichstellung behinderter Menschen“ festgeschrieben ist, baut z. B. auf diesen Kriterien auf. Sollen die BITV komplett erfüllt werden, müssen sowohl die Kriterien der Stufe 1, als auch die der Stufe 2 komplett erfüllt werden, was dem Konformitätslevel AA entspricht. [17, S. 38-39]

3.1.2 Wahrnehmbarkeit

Unter die Kategorie Wahrnehmbarkeit fallen folgende Richtlinien:

- Visuelle oder auditive Elemente sollten eine textbasierte Alternative besitzen [10, Abschnitt 6.1]
- Der Sinn des Inhalts sollte nicht nur von der Farbe abhängig sein [10, Abschnitt 6.2]

Visuelle oder auditive Elemente sollten eine textbasierte Alternative besitzen

Diese Richtlinie bezieht sich vor allem auf Personen mit Sehschwächen und besagt, dass für alle eingesetzten visuellen oder auditiven Elemente wie Bilder, Videos oder Klänge eine dementsprechende Alternative vorhanden sein sollte. Dies kann z. B. erreicht werden, indem `` Elemente immer ein `alt` Attribut mit der Beschreibung des Bildes besitzen, oder Videos/Klänge mit einem beschreibenden Text oder Transkription versehen sind. Von den Kriterien, die zur Erfüllung dieser Richtlinie aufgezählt sind, sind vier von fünf der Priorität 1 zuzuordnen. [10, Abschnitt 6.1]

Der Sinn des Inhalts sollte nicht nur von der Farbe abhängig sein

Diese Richtlinien beziehen sich vor allem auf Personen mit Rot-Grün Blindheit oder Personen, deren Monitor keine Farben ausgeben kann [17, S. 41]. Hier muss sicher gestellt werden, dass ein ausreichender Kontrast zwischen Hintergrund und Inhalt vorhanden ist (Priorität 2), und dass alle Informationen, die durch Farbe transportiert werden, auch ohne Farbe verstanden werden können (Priorität 1). [10, Abschnitt 6.1]

3.1.3 Bedienbarkeit

Unter die Kategorie Bedienbarkeit fallen folgende Richtlinien:

- Elemente, die von der Zeit abhängen, sollten stoppbar oder pausierbar sein [10, Abschnitt 6.7]
- Eingebettete Objekte sollten zugänglich sein [10, Abschnitt 6.8]
- Das Userinterface sollte auch auf anderen Plattformen bedienbar sein [10, Abschnitt 6.9]
- Eine gut strukturierte Navigation sollte verfügbar sein [10, Abschnitt 6.13]

Elemente, die von der Zeit abhängen, sollten stoppbar oder pausierbar sein

Es kann vorkommen, dass sich auf einer Webseite sich schnell bewegend oder blinkende Elemente befinden. Diese sind für Personen, welche z. B. eine Einschränkung in der Beweglichkeit ihrer Hände haben, besonders schwer zu bedienen. Für Personen mit Konzentrations- und Lernschwierigkeiten oder Sehschwächen können diese Texte sogar unlesbar sein (Screen-Reader können sich in Bewegung befindliche Texte nicht lesen). [10, Abschnitt 6.7]

Blinkende Elemente oder Flackern des Bildschirms, z. B. durch neu Laden der Seite, können außerdem bei Epileptikern Anfälle verursachen. Diese Elemente sollten daher vermieden werden, oder wenn verwendet, pausierbar oder abschaltbar sein. [17, S. 45]

Des Weiteren sollte sich die Seite auch nicht automatisch neu laden, da Screen-Reader sonst mitten im Lesen abbrechen können. Dadurch wird dann oft der Text von Neuem vorgelesen. Ein Verlust der Orientierung ist die Folge. [17, S. 45]

Eingebettete Objekte sollten zugänglich sein

Um die Zugänglichkeit der ganzen Seite zu gewährleisten, müssen auch einzelne, eingebettete Objekte, wie Java Applets oder Flash Container, zugänglich gestaltet sein. Ist dies nicht möglich, muss eine Alternative angeboten werden. [10, Abschnitt 6.8]

Das Userinterface sollte auch auf anderen Plattformen bedienbar sein

NutzerInnen verwenden die Seite auch oft auf anderen Geräten, wie z. B. auf Mobiltelefonen oder Terminals. Da diese Geräte oft verschiedene Eingabemethoden nutzen, muss die Seite auch dafür optimiert werden. Das kann z. B. durch einen funktionierenden Tabindex oder Shortcuts für oft verwendete Links erreicht werden. Des Weiteren sollten eingabespezifische *Event-Handler*, wie z. B. *onkeypress*, vermieden werden. [10, Abschnitt 6.9]

Eine gut strukturierte Navigation sollte verfügbar sein

Diese Richtlinie betrifft vor allem Menschen mit Lern- und Konzentrationsschwierigkeiten und Personen mit Sehschwächen. Eine einfache Navigation kann die Benutzung mit dem Screen-Reader angenehmer gestalten und die Orientierung verbessern. Dazu gehören unter anderem das Vorhandensein einer *Site-Map* und konsistenter Navigationselemente: z.B. müssen die Ziele eines Links klar ersichtlich und Überschriften müssen unterscheidbar von anderen Überschriften sein. [10, Abschnitt 6.13]

Von diesen Maßnahmen können auch andere NutzerInnen profitieren, da sie den gesuchten Inhalt auf der Seite besser finden können [17, S. 52].

3.1.4 Verständlichkeit

Unter die Kategorie Verständlichkeit fallen folgende Richtlinien:

- Verwendete Hauptsprache und Fremdsprachen sollten gekennzeichnet werden[10, Abschnitt 6.4]
- Information zu Kontext und Orientierung sollte verfügbar sein[10, Abschnitt 6.12]
- Der Inhalt der Seite sollte einfach und gut verständlich sein[10, Abschnitt 6.14]

Verwendete Hauptsprache und Fremdsprachen sollten gekennzeichnet werden

Ist Information über die eingesetzte Sprache vorhanden - das betrifft sowohl die Hauptsprache der Seite, als auch einzelne Textstellen und Abkürzungen - können z. B. Screen-Readern die Information in der korrekten Sprache ausgegeben. Um dies zu erreichen, sollte das *lang* Attribut verwendet werden (Priorität 1). Für Abkürzungen können das *<akronym>* und das *<abbr>* Element verwendet werden. [10, Abschnitt 6.4]

Information zu Kontext und Orientierung sollte verfügbar sein

Diese Richtlinie beschreibt vor allem den korrekten Umgang mit Frames und die korrekte Gruppierung von Elementen [17, S. 50]. Frames sollten identifizierbar sein und deren Zweck sollte für den/die BenutzerIn klar sein. Dafür werden die beiden Attribute *title* und *longdesc* vorgeschlagen. Des Weiteren sollten verwandte Informationen und Links mit den dafür vorgesehen HTML Elementen gruppiert werden, z. B. mit *<fieldset>* und *<optgroup>*. Die darin verwendeten Eingabeelementen sollten korrekt mit dem *<label>* Element beschriftet werden. [10, Abschnitt 6.12]

Der Inhalt der Seite sollte einfach und gut verständlich sein

Ein gute Verständlichkeit des Textes ist für sämtliche NutzerInnen wichtig. Dazu sollte das Seitenlayout konsistent sein und der Text mit Bildern oder Audio unterstützt werden. [10, Abschnitt 6.14]

Dies ist jedoch schwer erreich- und messbar. Außerdem erfordert der Einsatz von unterstützenden Bildern oder Audio selbst auch eine Alternative in Textform, was den Aufwand beträchtlich erhöht. [17, S. 52]

3.1.5 Robustheit

Unter die Kategorie Robustheit fallen folgende Richtlinien:

- Struktur und Präsentation sollten korrekt verwendet und getrennt werden[10, Abschnitt 6.3]
- Tabellen sollten korrekt verwendet werden[10, Abschnitt 6.5]
- Webseiten sollten abwärtskompatibel sein[10, Abschnitt 6.6]
- Webseiten sollten abwärtskompatibel zu älteren assistiven Technologien sein[10, Abschnitt 6.10]
- Standards und Richtlinien sollen verwendet werden[10, Abschnitt 6.11]

Struktur und Präsentation sollten korrekt verwendet und getrennt werden

Eine gute Trennung der verschiedenen *Layer* ist notwendig, damit auch Clients, die sich nur auf das *Markup* konzentrieren, den Inhalt korrekt darstellen können[17, S. 42-43]. Auch sollten die korrekten HTML Elemente für die jeweils dafür vorgesehenen Einsatzzwecke verwendet werden, um den Inhalt sinnvoll zu übermitteln. Als Beispiel dafür wird das Einbinden eines Zitates genannt: Dies sollte nicht mit *Stylesheets* oder durch Einrücken gekennzeichnet werden, sondern mit dem `<blockquote>` Element. [10, Abschnitt 6.3]

Tabellen sollten korrekt verwendet werden

Das Einhalten dieser Richtlinie hilft vor allem Personen mit Sehschwächen. Tabellen sollten nicht als Layoutwerkzeug missbraucht werden, sondern mit Informationen gefüllt werden, die auch am besten in eine Tabellenform passen. Wird das Layout z. B. mit Tabellen umgesetzt, haben unter anderem Screen-Reader massive Schwierigkeiten, auf der Webseite zu navigieren. Auch das inkorrekte Verwenden oder Weglassen von Tabellenüberschriften macht die Navigation für Screen-Reader unnötig kompliziert. [10, Abschnitt 6.5]

Webseiten sollten abwärtskompatibel sein

Beim Einsatz neuer Technologien muss auch die Abwärtskompatibilität zu älteren Browsern beachtet werden. Erwähnt werden unter anderem das Einsetzen von `<noscript>` oder `<noframes>` Elementen. Am Wichtigsten ist jedoch, dass die Seite auch ohne *JavaScript* und *Stylesheets* korrekt funktioniert, was vor allem für dynamische Seiten ein Problem darstellen kann. Sollte dies nicht möglich sein, muss eine Alternative bereitstehen. [10, Abschnitt 6.6]

Webseiten sollten abwärtskompatibel zu älteren assistiven Technologien sein

Diese Richtlinie schlägt in die gleiche Kerbe wie Richtlinie 6.6, aber bezieht sich mehr auf die Abwärtskompatibilität zu älteren assistiven Technologien. Da diese oft teuer sind, können sie nicht so oft durch aktuellere Geräte ersetzt werden [17, S. 48]. Ältere Geräte haben vor allem Probleme mit leeren Eingabeelementen und Popups. Dadurch entstehende Probleme müssen daher berücksichtigt werden. [10, Abschnitt 6.10]

Standards und Richtlinien sollten verwendet werden

Wann immer es möglich ist, sollten standardkonforme Techniken zum Einsatz kommen oder entsprechende Alternativen zur Verfügung stehen. Die Richtlinie listet unter den problematischen Technologien explizit PDF und Shockwave (das heutige Adobe Flash) auf und empfiehlt diese nur zu verwenden, wenn es unausweichlich ist. [10, Abschnitt 6.11]

3.2 WCAG Version 2.0

Version 2.0 wurde veröffentlicht, um die Anforderungen allgemeiner und testbarer zu gestalten [9, S. 20]. Daraus folgt jedoch auch, dass sie schwieriger zu verstehen und anzuwenden sind als die Richtlinien der vorigen Version [2, S. 24].

Im Gegensatz zu den WCAG der Version 1.0 ist die Version 2.0 in vier Kategorien gegliedert: Wahrnehmbarkeit, Bedienbarkeit, Verständlichkeit und Robustheit [2, S. 23-24]. Um beide Dokumente jedoch besser vergleichen zu können, wurde in dieser Arbeit versucht, die WCAG 1.0 in eine annähernd gleiche Struktur zu bringen.

3.2.1 Success Criteria

Wie auch in den der Version 1.0 gibt es drei verschiedene Konformitätslevel, die folgendermaßen erfüllt werden: [19, Abschnitt Understanding Requirement 1]

- A: Alle mit Konformitätslevel A angeführten Kriterien sind umgesetzt
- AA: Alle mit Konformitätslevel A und AA angeführten Kriterien sind umgesetzt
- AAA: Alle mit Konformitätslevel A, AA und AAA angeführten Kriterien sind umgesetzt

Die Unterteilung in diese Konformitätslevel wird anhand folgender Kriterien vorgenommen: [19, Abschnitt Understanding Levels of Conformance]

- Ist das Kriterium unabdingbar für körperlich eingeschränkte Personen oder Personen mit Lernschwierigkeiten, um die Seite bedienen und begreifen zu können?
- Ist das Kriterium umsetzbar?
- Ist das Kriterium ohne unverhältnismäßig großen Aufwand umsetzbar?
- Beeinträchtigt das Kriterium das Aussehen oder die Bedienung der Webseite?
- Gibt es *Workarounds*, wenn das Kriterium nicht erfüllt wird?

3.2.2 Wahrnehmbarkeit

Unter diese Richtlinie fallen Kriterien zur Präsentation und Wahrnehmbarkeit von Inhalten. Diese Inhalte sollten auch für körperlich eingeschränkte NutzerInnen und NutzerInnen mit Lernschwierigkeiten wahrnehmbar sein [11, Abschnitt 1] . Darunter fallen folgende Kriterien:

- Text Alternativen für Medien [11, Abschnitt 1.1 und 1.2]
- Adaptierfähigkeit des Layouts und der Struktur [11, Abschnitt 1.3]
- Unterscheidbarkeit des Inhaltes [11, Abschnitt 1.4]

Text Alternativen für Medien

Diese Richtlinie beinhaltet Hinweise zur Verwendung von Medien. Darunter fallen sowohl Bilder als auch Audio-, Video- oder Eingabeelemente. Eingabeelemente sollten immer durch ein *name* Attribut beschrieben sein und für Bilder, Videos, Audio sollte eine Textalternative bereitstehen (alle Level A). [11, Abschnitt 1.1 und 1.2]

Adaptierfähigkeit des Layouts und der Struktur

In dieser Richtlinie werden Kriterien zur Erstellung und Validierung einer flexiblen Struktur aufgelistet. Darunter fällt folgender Punkt: Die Beziehungen der präsentierten Inhalte und deren Reihung sollte nicht nur von der Formatierung und Form abhängig sein (Level A).[11, Abschnitt 1.3]

Unterscheidbarkeit des Inhaltes

Diese Richtlinie beschäftigt sich mit dem Kontrast und der Farbe des Inhaltes. Farbe sollte nicht als alleiniges Merkmal verwendet werden, um Information zu transportieren (Level A). Musik oder Töne sollten pausier- oder stoppbar sein und deren Lautstärke sollte anpassbar sein (Level A). Der Kontrast sollte sowohl für Bilder, Logos und Audio ausreichend sein (Level AA und AAA): Audio sollte keine Hintergrundgeräusche haben und lauter als 20dB sein, der Text sollte nicht mehr als 80 Zeichen in einer Zeile haben und nicht als Blocktext formatiert sein, der Zeilenabstand sollte mehr als 1,5 betragen und Hintergrundfarben sollten vom Inhalt unterschieden werden können. Des Weiteren sollte der Text ohne Probleme auf das Zweifache vergrößert werden können (Level AAA). Sollte Text in irgendeiner Weise in Bildern vorkommen, sollte dies nur einen dekorativen Zweck haben (Level AAA). [11, Abschnitt 1.4]

3.2.3 Bedienbarkeit

Diese Richtlinie beinhaltet Kriterien zur Bedienbarkeit der Webseite [11, Abschnitt 2]. Darunter fallen folgende Kriterien:

- Bedienbarkeit mittels Tastatur [11, Abschnitt 2.1]
- Bedienbarkeit von zeitkritischen Elementen [11, Abschnitt 2.2]
- Vermeidung von Ursachen von Epilepsieanfällen [11, Abschnitt 2.3]
- Navigierbarkeit [11, Abschnitt 2.4]

Bedienbarkeit mittels Tastatur

Diese Richtlinie besagt, dass die Webseite komplett und ohne großen Aufwand mit der Tastatur bedienbar sein sollte (Level AAA). Außerdem sollte es nie passieren, dass mit der Tastatur aus einem Block nicht mehr heraus navigiert werden kann (Level A). [11, Abschnitt 2.1]

Bedienbarkeit von zeitkritischen Elementen

Unter diese Richtlinie fallen Kriterien zum Umgang mit zeitkritischen Elementen. Unter diesen Begriff fallen Elemente, die abhängig vom jetzigen Zeitpunkt sind, beispielsweise sich bewegende, sich automatisch updatende oder blinkende Elemente. Für diese sollte eine Möglichkeit zur Pausierung, Abschaltung oder Anpassung des Zeitrahmens bestehen (Level A). Des Weiteren sollte der Inhalt der Webseite nicht von zeitkritischen Aktionen abhängen (Level AAA), Unterbrechungen, wie z. B. Updaten des Inhaltes, sollten abschaltbar oder aufschiebbar sein (Level AAA) und ein Verlust der *Session* sollte nicht zu Datenverlust beim erneuten Einloggen führen. [11, Abschnitt 2.2]

Vermeidung von Ursachen von Epilepsieanfällen

Diese Richtlinie beinhaltet Richtlinien, um Epilepsieanfälle vorzubeugen: Der Inhalt sollte im Rahmen einer Sekunde nicht mehr als drei mal Flackern (Level AAA) oder das Flackern sollte sich unter dem Grenzwert befinden (Level A). [11, Abschnitt 2.3]

Navigierbarkeit

Diese Richtlinie vereint Kriterien, welche sich mit der Navigierbarkeit der Seite auseinander setzen. Darunter fallen unter anderem: das Vorhandensein eines Titels der Webseite (Level A), mehr als eine Möglichkeit zum Auffinden von Informationen (Level AA), Informationen zum aktuellen Aufenthaltsort auf der Seite (beispielsweise durch den Einsatz von *Breadcrumbs*), Erkennen des Zweckes von Links anhand deren Linktexte oder Kontext (Level AAA und A) und der Einsatz von Überschriften, Labels (Level AA) und Abschnittsüberschriften (Level AAA). Außerdem sollte die Seite gut mit der Tastatur navigierbar sein: Dies inkludiert einen sichtbaren Fokus des aktuellen Elementes (Level AA), eine logische Fokusordnung (Level A) und eine Möglichkeit zum Überspringen von sich wiederholenden Inhalten, z. B. Werbeblöcken. [11, Abschnitt 2.4]

3.2.4 Verständlichkeit

Unter diese Richtlinie fallen Kriterien, die eine gute Verständlichkeit der Webseite fördern [11, Abschnitt 3]. Darunter fallen folgende Kriterien:

- Lesbarkeit und Verständlichkeit des Inhaltes [11, Abschnitt 3.1]
- Verhalten der Webseite [11, Abschnitt 3.2]
- Eingabehilfen [11, Abschnitt 3.3]

Lesbarkeit und Verständlichkeit des Inhaltes

Dieses Kriterium beschäftigt sich mit der generellen Verständlichkeit und Präsentation des Inhaltes. Ungebräuchliche Wörter und Abkürzungen sollten vermieden werden (Level AAA). Ist ein höheres Leselevel als das der ersten Sekundarschulstufe erforderlich, sollte eine Alternative angeboten werden (Level AAA). Fremdwörter, deren Kontext von der Aussprache abhängt, sollten als Audio ausgebaut sein (Level AAA). Des Weiteren sollten Informationen über die Standardsprache der Seite (Level A) und davon abweichende Absätze oder Wörter (Level AA) vorhanden sein. [11, Abschnitt 3.1]

Verhalten der Webseite

Dieses Kriterium beschreibt Regeln für ein vorhersehbares Verhalten der Webseite. Eingabeelemente sollten den Kontext nicht verändern, weder bei Eingabe noch bei Erhalten des Fokus (Level A). Sollte ein Kontextwechsel geschehen, ist dies explizit von dem/der NutzerIn herbeizuführen oder zumindest abschaltbar (Level AAA). Außerdem sollte die Konsistenz der Webseite gewährleistet sein: Navigationselemente sollten als Navigationselemente erkannt werden können, Eingabeelemente als Eingabeelemente (Level AA). [11, Abschnitt 3.2]

Eingabehilfen

Dieses Kriterium befasst sich vor allem mit der Behandlung von Fehlern, die durch Eingaben auftreten können. Tritt ein Fehler auf, sollte der/die NutzerIn mittels eines Textes verständigt werden (Level A); ist der Fehler bekannt, sollten Verbesserungsvorschläge angezeigt werden (Level AA). Außerdem sollte eine Möglichkeit bestehen, Eingaben rückgängig zu machen, zu korrigieren oder in einer zusätzlichen Übersicht zu bestätigen (Level AA und AAA). Des Weiteren sollten Eingabeelemente mit Labeln oder Anweisungen versehen werden (Level A) und eine kontextsensitive Hilfe angeboten werden (Level AAA). [11, Abschnitt 3.3]

3.2.5 Robustheit

Diese Richtlinie beinhaltet Kriterien für eine korrekte, geräte- und softwareübergreifende Darstellung und Interpretation der Webseite [11, Abschnitt 4.1]. Darunter fallen folgende 2 Level A Kriterien:

- Parsing [11, Abschnitt 4.1.1]
- Name, Role und Value [11, Abschnitt 4.1.2]

Parsing

Dieses Kriterium besagt im Wesentlichen, dass korrektes HTML Markup verwendet werden sollte, z. B. HTML Elemente müssen geschlossen werden oder es darf keine *ID* zweimal vorkommen. [11, Abschnitt 4.1.1]

Name, Role, Value

Dieses Kriterium besagt, dass alle Userinterface-Komponenten korrekt ausgelesen werden können müssen [11, Abschnitt 4.1.2]. Dies stellt kein Problem dar, wenn die dafür vorgesehenen HTML Elemente verwendet werden. Werden jedoch eigene Elemente für diese Zwecke erstellt, müssen

diese durch zusätzliche Techniken, wie etwa ARIA Roles, so beschrieben werden, dass eine korrekte Funktionsweise daraus abgeleitet werden kann [20, Abschnitt Intent of this Success Criterion]

3.3 Unterschiede zwischen WCAG Version 2.0 und 1.0

Die Spezifikation hat sich in den neun Jahren seit der Veröffentlichung der Version 1.0 stetig weiterentwickelt. Die WCAG 2.0 unterscheiden sich von der Version 1.0 nicht nur in der Struktur sondern auch im Umfang und der Testbarkeit. Wo in 1.0 noch eher allgemeine Hinweise gegeben wurden, sind in Version 2.0 klare und testbare Anleitungen und Richtlinien vorhanden. Auch die Umbenennung der Kriterien von Priorität zu Konformitätslevel zeigt den Wunsch nach einer besseren automatisierten Testbarkeit der Kriterien. [11, Abschnitt Introduction]

3.3.1 Success Criteria

Die *Success Criteria* wurden von ihrer Struktur praktisch übernommen und anders benannt. Anstatt Priorität wird jetzt der Begriff Konformitätslevel verwendet. Die Bezeichnungen der Stufen (A, AA, und AAA) und auch deren Erfüllung - jede Stufe schließt automatisch alle Kriterien der ihr untergeordneten Stufen mit ein - wurden jedoch nicht verändert.

3.3.2 Wahrnehmbarkeit

Die WCAG 2.0 geben im Gegensatz zu den WCAG 1.0 klare, testbare Hinweise, wie die Wahrnehmbarkeit des Inhaltes verbessert werden kann. Statt Hinweisen, die sich nur mit HTML Elementen und Attributen beschäftigen, werden jetzt auch konkrete Präsentationsrichtlinien genannt: Beispielsweise wird eine maximale Zeilenlänge und ein minimales Kontrastverhältnis erwähnt. Auch die Beschreibung der Audio und Video Alternativen ist wesentlich umfangreicher als die der WCAG 1.0 aus dem Jahre 1999.

3.3.3 Bedienbarkeit

Die Richtlinien zur Bedienbarkeit haben sich im Wesentlichen durch das Wegfallen von *Frames* und eingebetteten Objekten geändert. Eingebettete Objekte können mittlerweile weitgehend durch HTML5 Elemente ersetzt werden [21, Abschnitt audio und video element].

3.3.4 Verständlichkeit

Die Richtlinien im Bereich Verständlichkeit wurden vor allem im Bezug auf Eingabeelemente erweitert, was der stärkeren Verwendung von *JavaScript* geschuldet ist. Beispielsweise werden in der Version 2.0 mehr Richtlinien angegeben, wie und wann der Kontext der Seite verändert werden darf, z. B. sollte der Kontext nie durch bloßen Fokus eines Eingabeelementes verändert werden. Für die Lesbarkeit und Verständlichkeit der Webseite wird jetzt ein Kriterium, nämlich das Leselevel der ersten Sekundarschulstufe, angegeben.

3.3.5 Robustheit

Diese Sektion hat im Gegensatz zur Version 1.0 massiv an Inhalt eingebüßt. Es scheint, dass eine korrekte Verwendung von HTML vorausgesetzt wird und dass der Fokus nicht mehr so stark auf die Abwärtskompatibilität, sondern mehr auf die Aufwärtskompatibilität gelegt wird.

4 ARIA

Im Web werden heutzutage vermehrt *AJAX* und *JavaScript* eingesetzt, nicht nur um eine immersive Erfahrung zu bieten, sondern auch um ganze Applikationen umzusetzen. Diese Technik ermöglicht es, bestimmte Inhalte auf der Seite nachzuladen, ohne die ganze Seite neu laden zu müssen. Außerdem können eigene Eingabekomponenten erstellt werden. [2, S.26]

Viele vom Desktop bekannte Paradigmen können mittlerweile umgesetzt werden, z. B. „Drag and Drop“. Einige dieser Aktionen funktionieren jedoch nur mit speziellen Eingabegeräten. So muss für Drag and Drop zwingend eine Maus vorhanden sein. Für Nutzer eines Screen-Readers oder anderen assistiven Technologien wird es daher zunehmend schwieriger, mit der Webseite zu interagieren. [22]

ARIA, eine Abkürzung für Accessible Rich Internet Applications Suite, versucht dieses Problem mit zusätzlichen HTML Attributen zu lösen. Es erlaubt den ProgrammiererInnen zusätzliche Meta-Informationen über die Funktionsweise und den gerade aktiven Status der Seite bereitzustellen. Zu diesen Meta-Informationen gehören: *Roles*, *States* und *Properties* [22].

4.1 Roles

Die Aufgaben der ARIA Role Attributes ist es, die Struktur der Seite und deren Interaktionsmöglichkeiten zu beschreiben. Dies ist zum Teil auch schon mit den neuen HTML5 Elementen möglich, wie z. B. `<nav>` [21, Abschnitt 4.4.3] (vgl. *role="navigation"* [23, Abschnitt 3.1]), jedoch verwenden viele Personen noch ältere Browser, die diese neuen Funktionen noch nicht unterstützen. Daher werden oft neue Elemente mit *JavaScript* und *CSS* simuliert, was die Nutzung für körperlich eingeschränkte Personen deutlich erschwert. Auch bietet das Role Attribut zusätzliche Informationen, welche in der jetzigen HTML5 Version noch nicht vorhanden sind[21]. Das soll jedoch kein Freibrief für Entwickler sein, alle semantischen HTML Elemente durch nicht semantische Elemente mit Role Attributen zu ersetzen [24, Abschnitt 3].

Für das Role Attribut können ein oder mehrere, durch Leerzeichen getrennte Werte gesetzt werden. Der Wert des Role Attributes darf sich jedoch nicht ändern. Sollte dies jemals erforderlich sein, muss das HTML Element zuerst komplett entfernt und dann mit einem neuen Element mit der gewünschten Role ersetzt werden.[25, Abschnitt 5]

Die Werte, die ein Role Element annehmen kann, gliedern sich in vier verschiedene Kategorien: *Abstract Roles*, *Widget Roles*, *Document Structure* und *Landmark Roles*. [26, Abschnitt 5.2]

Abstract Roles

Abstract Roles sind für die Beschreibung der Aufgabe des Elementes gedacht und dürfen nicht im Inhalt vorkommen. Ein Beispiel für eine Abstract Role wäre z. B. *widget* oder *window*. [25, Abschnitt 5.3.1]

Widget Roles

Widget Roles sind für die Beschreibung von *Widgets* gedacht. Widgets sind Elemente, die den User mit der Seite interagieren lassen[25, Abschnitt 5.4, widget]. Ein Beispiel für eine Widget Role wäre z. B. *dialog* oder *progressbar*. [25, Abschnitt 5.3.2]

Document Structure

Die Document Structure beschreibt die Struktur der Seite. Ein Beispiel für eine Document Structure Role wäre z. B. *article* oder *list*. [25, Abschnitt 5.3.3]

Landmark Roles

Landmark Roles beschreiben wichtige Navigationselemente und Bereiche. Ein Beispiel für eine Landmark Role wäre z. B. *banner* oder *navigation*. [25, Abschnitt 5.3.4]

4.1.1 ARIA Roles Beispiele

Die nachfolgenden genannten Werte stellen nur eine kleine Auswahl aller verfügbaren Roles dar und sollen ein besseres Verständnis ihres Sinnes und Einsatzgebietes vermitteln:

- *banner*: Am Anfang einer Seite befindet sich oft der *Header*, welcher ein Bild, den Titel der Seite, oder zusätzliche Informationen zur Seite beinhaltet. Dieser sollte mit der *banner* Role versehen werden. [25, Abschnitt 5.4, banner]
- *dialog*: Viele *JavaScript Frameworks* wie z. B. *jQuery-UI* [27] bieten eine Möglichkeit an, Dialoge mit *JavaScript* und *CSS* zu simulieren. Diese Dialoge sind für NutzerInnen assistiver Technologien oft nicht als solche erkennbar und sollten deshalb mit der *dialog* Role gekennzeichnet werden. [25, Abschnitt 5.4, dialog]
- *main*: Ein durch die *main* Role gekennzeichnetes HTML Element sollte den eigentlich Inhalt der Seite beinhalten. Dadurch können assistive Technologien z. B. einen eigenen Shortcut anbieten, der bei der Aktivierung direkt zum Inhalt springt. [25, Abschnitt 5.4, main]
- *math*: Durch fehlende Möglichkeiten zur Darstellung mathematischer Funktionen wurden und werden oft Bilder oder *Markup Sprachen* wie *MathML* oder *LaTeX* verwendet. Die *math* Role hilft dem Browser die entsprechende Sektion in ein für NutzerInnen assistiver Technologien verständliches Format umzuwandeln und darzustellen. [25, Abschnitt 5.4, math]
- *menu* und *menuitem*: Wird verwendet, um Menüs und deren Einträge als solche zu kennzeichnen. Die Menüs sollten zudem mit der Tastatur bedient werden können. [25, Abschnitt 5.4, menu, menuitem]
- *navigation*: Wird verwendet, um Navigationselemente zu kennzeichnen. In älteren HTML Versionen wurde hierfür meist ein `<div>` Element mit der *ID* *nav* oder *navigation* verwendet. Dieses `<div>` Element sollte mit der *navigation* Role gekennzeichnet werden. [25, Abschnitt 5.4, navigation]
- *progressbar*: Durch die fehlende Unterstützung der neuen HTML5 `<input>` Elemente in einigen aktuell verwendeten Browsern, bieten Frameworks wie *jQuery-UI* [27] eine abwärts-kompatible, für assistive Geräte meist unbrauchbaren Darstellung eines Fortschrittsbalken

an. Diese sollte mit der *progressbar* Role gekennzeichnet werden. [25, Abschnitt 5.4, progressbar]

- *search*: Oft wird eine seitenweite Suche in Form eines normalen HTML `<input>` Elementes angeboten. Dieses Element kann mit dieser Role als Suche deklariert werden. [25, Abschnitt 5.4, search]
- *tooltip*: Diese Role sollte verwendet werden, um *Tooltips* zu kennzeichnen, die von manchen *JavaScript Frameworks* generiert werden. [25, Abschnitt 5.4, tooltip]

4.2 States and Properties

States und Properties werden verwendet, um Zustände von HTML Elementen zu beschreiben bzw. um Roles mit zusätzlichen Informationen zu versehen. Die Role *dialog* kann beispielsweise mit einem Label versehen werden, um den Dialog zu beschreiben (*aria-label* oder *aria-labelledby*) [25, Abschnitt 5.4, dialog]. Alle States und Properties Attribute werden mit einem vorangestellten *aria*-gekennzeichnet. [26, Abschnitt 6.1]

Im Gegensatz zu Roles können *States und Properties* im Laufe der Webseitenutzung ihre Werte ändern [2, S. 29]. Ein Beispiel dafür ist *aria-grabbed*, welches sich im Zustand *true* bzw. *false* befinden kann, je nachdem, ob gerade eine *Drag and Drop Operation* durchgeführt wird [26, Abschnitt 6.6, aria-grabbed].

State Attribute stellen jedoch nicht nur neue Funktionen bereit, sondern duplizieren zum Teil auch Information, die in HTML schon vorhanden sind und ausgelesen werden können, z. B. *aria-disabled* oder *aria-checked* [12, States and Properties] (vgl. `<input>` Element Attribute *checked* und *disabled* [28, Abschnitt Forms, Input Element]).

Dadurch können auch HTML Elemente als z. B. *checked* gesetzt werden, die ursprünglich nicht dafür vorgesehen waren, aber trotzdem so verwendet werden.

States und Properties lassen sich in folgende vier Kategorien einteilen: *Widget Attributes*, *Live Region Attributes*, *Drag and Drop Attributes* und *Relationship Attributes*. [26, Abschnitt 6.5]

4.2.1 Widget Attributes

Widget Attributes werden verwendet, um Widgets, sprich Interaktionselemente zu beschreiben. Ein Beispiel dafür wäre *aria-label*, welches für die Beschreibung eines Eingabefelds benutzt werden kann. Es gibt zwar das HTML `<label>` Element, welches dieser Aufgabe nachkommt; wird jedoch das `<input>` Element mitten im Beschreibungstext verwendet, kann dies für die NutzerInnen von Screen-Readern zu unklaren Situationen führen. [26, Abschnitt 6.6, aria-label]

4.2.2 Live Regions

Live Regions werden dazu verwendet, um Bereiche zu markieren, in denen sich der Inhalt dynamisch und ohne Zutun des/der BenutzerIn ändern kann, beispielsweise durch Nachladen von Informationen mittels *AJAX Request*. Ein Beispiel dafür wäre *aria-busy*, welches dem Client mitteilt, dass sich gerade ein Bereich updated. [26, Abschnitt 6.6, aria-busy]

4.2.3 Drag and Drop Attributes

Drag and Drop Attributes werden für die Kennzeichnung von Drag and Drop fähigen Elementen verwendet. Ein Beispiel dafür wäre das bereits erwähnte *aria-grabbed*. [26, Abschnitt 6.6, *aria-grabbed*]

4.2.4 Relationship Attributes

Relationship Attributes werden verwendet, um Beziehungen zwischen bestimmten HTML Elementen herzustellen. Dies kann zum Beispiel wichtig sein, wenn sich ein Label für ein Eingabeelement an einer weit entfernten Stelle auf der Webseite befindet. Dafür kann das *aria-labelledby* Attribut verwendet werden, welches mittels einer *ID* das Label auf der Seite referenzieren kann. [26, Abschnitt 6.6, *aria-labelledby*]

Eine Übersicht über alle möglichen States und Properties kann auf der Spezifikationsseite nachgeschlagen werden. [26, Abschnitt 6.6]

5 Zusammenfassung

Personen mit körperlichen Einschränkungen oder Lernschwierigkeiten sind nicht nur im alltäglichen, sondern auch im digitalen Leben in der Erstellung und Wahrnehmung von Inhalten beeinträchtigt. Beachtet der/die WebentwicklerIn jedoch die Empfehlungen und Richtlinien der W3C, die WCAG 2.0 und ARIA, kann diesen Personen eine hinreichend annehmbare Erfahrung im Netz geboten werden. Die dadurch durchgeführten Anpassungen bieten nicht nur eine bessere Erfahrung für Personen mit körperlichen Einschränkungen oder Lernschwierigkeiten, sondern auch für alle anderen NutzerInnen des Webs. Auch für die ProgrammiererInnen selbst kann eine Beachtung dieser Richtlinien einen positiven Effekt haben: durch die Trennung von Präsentation und Inhalt wird die Wartung und Weiterentwicklung der Webseite vereinfacht.

6 Ausblick

Weitere Empfehlungen des W3C, die sich im Moment in der Entwicklung befinden und nicht in dieser Arbeit erwähnt wurden, behandeln vor allem zusätzliche Dokumente zu ARIA und Empfehlungen für Werkzeuge zur Erstellung (z. B. Editoren) und Darstellung (z. B. Browsern) von Webinhalten. Aus Diesen wird ersichtlich, dass der Fokus von Accessibility zukünftig nicht nur auf den reinen Konsum von Inhalten beschränkt wird, sondern dass Personen mit körperlichen Einschränkungen oder Lernschwierigkeiten aktiv an der Weiterentwicklung des Webs und dessen Inhalten beteiligt und somit komplett in die digitale Gesellschaft integriert werden. [29]

Literaturverzeichnis

- [1] J. E. H. und Kerstin Probiesch, *Barrierefreiheit verstehen und umsetzen*. dpunkt.verlag, 2011.
- [2] R. Mauerhofer, "Auswirkungen von modernen Softwareentwicklungstechniken auf die Barrierefreiheit von Web-Anwendungen," Master's thesis, Technische Universität Wien, 2009.
- [3] J. Waldrop and S. M. Stern, "Disability Status: 2000," Website, 2000, <http://www.census.gov/prod/2003pubs/c2kbr-17.pdf> [Zugang am 8.12.2012].
- [4] WHO, "Disability, including prevention, management and rehabilitation," Website, WHO, <http://www.who.int/nmh/a5817/en/> [Zugang am 8.12.2012].
- [5] U. Nations, "Convention on the Rights of Persons with Disabilities," Website, UN, 2007, <http://www.un.org/disabilities/default.asp?navid=12&pid=150> [Zugang am 8.12.2012].
- [6] B. Sierkowski, "Achieving Web Accessibility," *SIGUCCS '02 Proceedings of the 30th annual ACM SIGUCCS conference on User services*, pp. 288–291, 2002.
- [7] "Policies Relating to Web Accessibility," Website, W3C, 2006, <http://www.w3.org/WAI/Policy/> [Zugang am 8.12.2012].
- [8] "UN Global Audit of Web Accessibility," Website, UN, 2006, <http://www.un.org/esa/socdev/enable/gawanomensa.htm> [Zugang am 8.12.2012].
- [9] S. Abou-Zahra, "Tool-Supported Web Accessibility Evaluation," Master's thesis, Technische Universität Wien, 2009.
- [10] "Web Content Accessibility Guidelines 1.0," Website, W3C, 1999, <http://www.w3.org/TR/WCAG10/> [Zugang am 8.12.2012].
- [11] "Web Content Accessibility Guidelines (WCAG) 2.0," Website, W3C, 2008, <http://www.w3.org/TR/WCAG/> [Zugang am 8.12.2012].
- [12] W3C, "Accessible Rich Internet Applications," Website, W3C, <http://www.w3.org/TR/2011/CR-wai-aria-20110118/> [Zugang am 8.12.2012].
- [13] "WHO ICD-10 F70-F79," Website, WHO, 2010, <http://apps.who.int/classifications/icd10/browse/2010/en#/F70-F79> [Zugang am 8.12.2012].
- [14] C. M. P. Jeffrey P. Bigham and R. E. Ladner, "WebAnywhere: A Screen Reader On-the-Go," p. 1, 2008.
- [15] M. B. L. und Dr. Erika Baldaszi, "Menschen mit Beeinträchtigungen: Ergebnisse der Mikrozensus-Zusatzfragen," PDF, Statistik Austria, 2007, http://www.bmask.gv.at/cms/site/attachments/4/4/1/CH2092/CMS1313493163518/mikrozensus_2007-bericht_statistik_austria_04-2008.pdf [Zugang am 1.1.2013].

-
- [16] J. K. Jonathan Lazar, Aaron Allen and C. Malarkey, "What Frustrates Screen Reader Users on the Web: A Study of 100 Blind Users," *International Journal of Human-Computer Interaction*, pp. 247–269, 2008.
- [17] A. R. und Dr. Michael Charlier, *Barrierefreies Webdesign*. Addison-Wesley, 2006.
- [18] "GNOME Orca," Website, GNOME, 2012, <http://projects.gnome.org/orca/> [Zugang am 8.12.2012].
- [19] "Understanding Conformance," Website, W3C, <http://www.w3.org/TR/UNDERSTANDING-WCAG20/conformance.html> [Zugang am 1.1.2013].
- [20] "Name, Role, Value: Understanding SC 4.1.2," Website, W3C, <http://www.w3.org/TR/UNDERSTANDING-WCAG20/ensure-compat-rsv.html> [Zugang am 1.1.2013].
- [21] W3C, "HTML5 Spezifikation," Website, W3C, <http://www.w3.org/TR/2011/WD-html5-20110525/> [Zugang am 8.12.2012].
- [22] —, "WAI-ARIA Overview," Website, W3C, <http://www.w3.org/WAI/intro/aria.php> [Zugang am 8.12.2012].
- [23] —, "XHTML Vocabulary," Website, W3C, <http://www.w3.org/1999/xhtml/vocab/> [Zugang am 8.12.2012].
- [24] —, "Role Attribute 1.0," Website, W3C, <http://www.w3.org/TR/2012/CR-role-attribute-20120712/> [Zugang am 8.12.2012].
- [25] —, "ARIA Roles," Website, W3C, <http://www.w3.org/TR/wai-aria/roles> [Zugang am 8.12.2012].
- [26] —, "ARIA States and Properties," Website, W3C, http://www.w3.org/TR/wai-aria/states_and_properties [Zugang am 8.12.2012].
- [27] T. jQuery Project jQuery.org, "jQuery UI," Website, The jQuery Project - jQuery.org, <http://jqueryui.com> [Zugang am 17.12.2012].
- [28] W3C, "HTML 4.01 Spezifikation," Website, W3C, <http://www.w3.org/TR/REC-html40> [Zugang am 8.12.2012].
- [29] —, "Accessibility Overview," Website, W3C, http://www.w3.org/standards/techs/accessibility#w3c_all [Zugang am 2.1.2013].

Abbildungsverzeichnis

2.1	Beispiel einer Braille	4
-----	----------------------------------	---

Abkürzungsverzeichnis

www	World Wide Web
W3C	World Wide Web Consortium
URL	Uniform Resource Locator
ARIA	Accessible Rich Internet Applications
WAI	Web Accessibility Initiative
AJAX	Asynchronous JavaScript and XML
BITV	Barrierefreie Informationstechnologie Verordnung
HTML	Hypertext Markup Language
XML	Extensible Markup Language