
Entwurfsprinzipien barrierefreier Web-Auftritte und ihre Umsetzung am Beispiel des Bildungsträgers “FAW gGmbH, Akademie Cottbus”

Bachelor-Arbeit zur Erlangung des Grades Bachelor of Science (B.Sc.) an der
Fakultät Informatik der Brandenburgischen Technischen Universität Cottbus

Jan Böschow

26.04.2005

Matrikel: 2029222

Betreuer: Prof. Dr. Gerd Wagner

Inhaltsverzeichnis

1. Aufgabenstellung und Lösungsansatz	4
1.1. Aufgabenstellung	4
1.2. Lösungsansatz	4
2. Einleitung	5
3. Barrierefreiheit im Internet	5
3.1. Begriffserklärung	5
3.2. Bedeutung barrierefreier Informationsangebote	6
3.3. Barrieren von Web-Auftritten	6
3.3.1. Sehbehinderte Benutzer	6
3.3.2. Hörgeschädigte Benutzer	8
3.3.3. Benutzer mit motorischen Einschränkungen	8
3.3.4. Sonstige Barrieren	9
3.4. Alternative Ein-/Ausgabesysteme	9
3.4.1. Hardware	9
3.4.2. Software	11
3.5. Fazit	12
4. Richtlinien und Empfehlungen	12
4.1. Überblick über die Richtlinien der WAI	12
4.2. Analyse und Beschreibung der Richtlinien der WAI	13
4.2.1. Alternativen für Audio- oder visuelle Inhalte	14
4.2.2. Einsatz von Farben	14
4.2.3. Markup und Stylesheets	15
4.2.4. Abkürzungen und fremdsprachige Inhalte	15
4.2.5. Verwendung von Tabellen	16
4.2.6. Neueste Webtechnologiestandards	16
4.2.7. Zeitsensitive Inhalte	16
4.2.8. Unabhängigkeit von Skripten und Applets	17
4.2.9. Unabhängigkeit von Ein-/Ausgabegeräten	17
4.2.10. Übergangslösungen	17
4.2.11. Nutzung von W3C-Technologien und -Richtlinien	18
4.2.12. Kontext und Orientierung	18
4.2.13. Navigationselemente	19
4.2.14. Natürliche Sprache und Formulierungen	19
4.3. Richtlinien anderer Einrichtungen	19
4.3.1. Barrierefreie Informationstechnik-Verordnung	20
4.3.2. Brandenburgisches Behindertengleichstellungsgesetz	21
4.4. Umsetzung der Richtlinien	21
4.4.1. XML	21
4.4.2. XHTML	22
4.4.3. CSS	24
4.5. Fazit	27
5. Evaluierung von Web-Auftritten	27
5.1. Validator "Cynthia Says"	28
5.2. Validator "Bobby"	29
5.3. Manuelle Evaluierungsmethode	30

5.3.1. Grundlegende Kriterien	31
5.3.2. Optische Gestaltung	31
5.3.3. Alternativen für Audio- und visuelle Inhalte	33
5.3.4. Navigation und funktionale Elemente	34
5.3.5. Seitenstruktur und Markup	35
5.3.6. Natürliche Sprache	37
5.4. Einsatz der Evaluierungsmethode am konkreten Beispiel	37
5.5. Einsatz der verschiedenen Evaluierungsmethoden	42
6. Auftritt der Akademie Cottbus	42
6.1. Anforderungen	43
6.1.1. Barrierefreiheit	43
6.1.2. Optische Gestaltung	44
6.1.3. Inhalt	45
7. Modellierung und Implementierung	46
7.1. Datenebene	47
7.2. Funktionsebene	49
7.3. Präsentationsebene	50
7.4. Evaluierung des Auftrittes	52
8. Zusammenfassung	55
Literaturverzeichnis	57
A. Eidesstattliche Erklärung	58
B. Abbildungsverzeichnis	59
C. Abkürzungsverzeichnis	60
D. Glossar	61
E. XHTML-Elemente und -Attribute	62
F. Dokumente zur Modellierung und Implementierung	65

1. Aufgabenstellung und Lösungsansatz

1.1. Aufgabenstellung

In dieser Arbeit werden die Aspekte der barrierefreien Gestaltung von Internetauftritten diskutiert und Entwurfsprinzipien in Verbindung mit den neuesten Webtechnologiestandards erläutert. Darauf aufbauend wird eine Evaluierungsmethode entwickelt, die es ermöglicht, Informationsangebote im Internet auf Barrierefreiheit zu überprüfen. Anschließend wird eine konkrete Implementierung eines barrierefreien Web-Auftrittes für den Bildungsträger "FAW gGmbH, Akademie Cottbus" vorgestellt, an welcher die entwickelte Evaluierungsmethode angewandt wird.

1.2. Lösungsansatz

Nach der Einleitung, die im zweiten Kapitel zu finden ist, gibt das dritte Kapitel zunächst einen Überblick über das Thema der „Barrierefreiheit im Internet“. Auch die Bedeutung barrierefreier Informationsangebote kommt in diesem Abschnitt zum Ausdruck. Darauf aufbauend erfolgt eine Erläuterung verschiedener Probleme, mit denen behinderte Menschen bei der Arbeit mit dem Internet konfrontiert werden.

Bei der Planung und dem Entwurf von barrierefreien Web-Auftritten sind eine Vielzahl von Regeln zu beachten, die aus den Richtlinien der WAI des W3C hervorgehen und auch in anderen Empfehlungen zu finden sind. Das vierte Kapitel enthält Beschreibungen und Analysen dieser Richtlinien. Zudem werden die empfohlenen Markup-Sprachen und ihre Vor- und Nachteile im Hinblick auf Barrierefreiheit untersucht.

Das fünfte Kapitel beschäftigt sich mit einer Methode zur Evaluierung bestehender Web-Auftritte, welche es ermöglicht, Informationsangebote im Internet auf Aspekte der Barrierefreiheit zu überprüfen und eine Bewertung dieser Angebote vorzunehmen. Der Einsatz der entwickelten Methode wird anschließend anhand eines konkreten Beispiels verdeutlicht.

Vor der Umsetzung der Entwurfsprinzipien auf den zu entwickelnden Web-Auftritt des Bildungsträgers "FAW gGmbH, Akademie Cottbus" muss eine Anforderungsdefinition für diesen konkreten Anwendungsfall erstellt werden, die im sechsten Kapitel zu finden ist.

Danach wird mit Hilfe geeigneter Sprachen der Auftritt entwickelt. Begleitend dazu erfolgt eine ständige Überprüfung des Auftrittes auf die Einhaltung der Entwurfsprinzipien und gegebenenfalls eine Korrektur. Das siebte Kapitel enthält die Beschreibung der Implementierung und zeigt eventuelle Probleme auf, die bei der Entwicklung barrierefreier Informationsangebote entstehen. Zusätzlich wird anhand der Evaluierungsmethode aus dem fünften Kapitel eine Überprüfung des entwickelten Web-Auftrittes durchgeführt.

Eine Zusammenfassung der wichtigsten Aspekte dieser Arbeit findet sich im abschließenden achten Kapitel.

2. Einleitung

In den letzten Jahren hat das Internet stetig an Bedeutung gewonnen. Inzwischen stellt es ein Informations- und Kommunikationsmedium dar, welches aus dem Leben vieler Menschen nicht mehr wegzudenken ist. Vor allem die Vielzahl der Möglichkeiten, die das Internet eröffnet, macht es unentbehrlich. Dabei sollte es selbstverständlich sein, dass Informationen, die im Internet bereitgestellt werden, unkompliziert zu nutzen sind. Jedoch ist dies oftmals nicht der Fall. In Deutschland leben ungefähr 6,7 Millionen schwerbehinderte Menschen. Dazu zählen etwa 500.000 Blinde und 155.000 Sehbehinderte. Hinzu kommen 300.000 Hörgeschädigte und ungefähr eine Million Menschen mit körperlichen Einschränkungen. Dabei sind nur wenige dieser Menschen von Geburt an schwerbehindert. Die meisten haben auf Grund von Unfällen, Krankheiten oder Alterungserscheinungen mit Beeinträchtigungen zu kämpfen. Vor allem für Menschen mit Behinderungen stellen viele Angebote im Internet große Hürden dar, die es ihnen zum Teil unmöglich machen, Zugang zu Informationen zu erhalten. Doch gerade diesen Benutzergruppen eröffnet ein einfacher Zugang zu Informationsangeboten Möglichkeiten, die im bisherigen Leben nicht oder nur eingeschränkt gegeben waren.

Im Jahr 2002 wurde das Behindertengleichstellungsgesetz (BGG) verabschiedet, welches öffentliche Einrichtungen unter anderem dazu verpflichtet und privaten Einrichtungen zumindest empfiehlt, Informationsverarbeitungssysteme barrierefrei zu gestalten. Trotzdem bleibt vielen Menschen der Zugang zu diesen Informationssystemen noch immer erschwert, ja teilweise sogar gänzlich verwehrt. Dies bedeutet für sie oft den Ausschluss aus einem wichtigen Teil des gesellschaftlichen Lebens. Sehbehinderte, Hörgeschädigte und motorisch Behinderte stehen bei vielen Web-Auftritten vor unüberwindbaren Problemen, obwohl diese Benutzergruppen das Internet sehr intensiv nutzen. Von der Beachtung verschiedener Entwicklungs-Prinzipien für Web-Auftritte und der damit verbundenen Verbesserung der Zugänglichkeit würden jedoch alle Benutzer profitieren.

Diese Arbeit soll sich mit den Aspekten der Entwicklung barrierefreier Web-Auftritte befassen, welche anhand eines Auftrittes für den Bildungsträger "FAW gGmbH, Akademie Cottbus" umgesetzt werden. Zusätzlich wird eine Methode vorgestellt, die es ermöglicht, bestehende Informationsangebote auf Zugänglichkeit zu prüfen. Diese Methode wird an dem zu entwickelnden Auftritt angewandt.

3. Barrierefreiheit im Internet

3.1. Begriffserklärung

Der Begriff der Barrierefreiheit bzw. Accessibility bezeichnet die Möglichkeit, Gebrauchsgegenstände oder Objekte ohne Einschränkungen nutzen zu können. Diese uneingeschränkte Zugänglichkeit gilt für alle Benutzer, es soll also keine Alternative für behinderte oder beeinträchtigte Menschen geschaffen werden. Vielmehr sollen alle Benutzergruppen den gleichen Zugang nutzen können.

In Verbindung mit Web-Auftritten versteht man unter dem Begriff der Barrierefreiheit die

uneingeschränkte Nutzbarkeit von Informationsangeboten im WWW. Dies beinhaltet nicht nur die behindertengerechte Gestaltung von Web-Seiten. Vielmehr soll beim Entwurf barrierefreier Informationsangebote die Möglichkeit geschaffen werden, jedem Benutzer mit jeder Browsertechnologie den Zugang zu allen Informationen eines Web-Auftrittes sowie die uneingeschränkte Interaktion mit der Web-Seite zu gewährleisten. Dieses Merkmal ist auch unter dem Begriff Design-for-all bekannt.

3.2. Bedeutung barrierefreier Informationsangebote

Der Begriff der Barrierefreiheit gewinnt immer mehr an Bedeutung. Vor allem behinderte und beeinträchtigte Menschen profitieren stark von modernen Informationstechnologien. Einschränkungen der Mobilität können zum Teil kompensiert werden, sei es durch die Verbindung mit dem Arbeitgeber über Internet und Intranet oder durch das Einkaufen in Online-Shops und das Erledigen von Bankgeschäften per Online-Banking. Durch die Verabschiedung des Behindertengleichstellungsgesetzes soll zudem die Kommunikation mit öffentlichen Behörden ohne Barrieren im Internet möglich sein. Aber auch für alle anderen Benutzer sind barrierefreie Informationsangebote von großer Bedeutung. Vor allem in Zeiten, in denen das Internet ständigen Änderungen unterliegt, ist es vorteilhaft, wenn Web-Seiten leicht zu bedienen sind und ein schneller Zugriff auf die gesuchten Informationen ermöglicht wird. Mobile Geräte und Systeme mit kleinen Bildschirmen erfordern dynamisch anpassbare Präsentationen von Web-Seiten, um Inhalte und Informationen darstellen zu können. Um diesen Forderungen gerecht zu werden, erarbeitete das W3C in seiner WAI spezielle Richtlinien für die Entwicklung barrierefreier Web-Seiten. Die Einhaltung dieser Richtlinien ist nicht nur für beeinträchtigte und behinderte Menschen von Vorteil, sondern für alle Benutzer. Sie ermöglichen es, Web-Seiten zu entwickeln, die intuitiv bedienbar sowie leicht verständlich sind und die Informationsfindung im Internet stark vereinfachen.

3.3. Barrieren von Web-Auftritten

Bei der Entwicklung von Web-Auftritten sollte auf Einhaltung der formalen Spezifikationen der verwendeten Markup-Sprache geachtet und eine strikte Trennung von Inhalt und Präsentation eingehalten werden. Dies ist nicht nur für behinderte und beeinträchtigte Benutzergruppen von Vorteil, sondern für alle Benutzer und vereinfacht zudem die Pflege von Web-Auftritten erheblich. Trotzdem stehen behinderte und beeinträchtigte Menschen vor speziellen Problemen im Umgang mit Informationsangeboten im WWW. Oftmals werden beim Entwurf und bei der Entwicklung von Web-Auftritten vor allem Aspekte der optischen Präsentation in den Vordergrund gestellt. Dies bedeutet aber meist, dass diese Benutzergruppen keinen Zugang zu Informationen und Funktionalitäten eines solchen Auftrittes erhalten, dieser jedoch zumindest stark eingeschränkt ist. Im Folgenden sollen die typischen Probleme und Lösungsmöglichkeiten aufgezeigt werden.

3.3.1. Sehbehinderte Benutzer

Zur Gruppe der sehbehinderten Benutzer gehören blinde, farbenblinde und fehlsichtige Anwender. Für farbenblinde und fehlsichtige Menschen ist es vor allem wichtig, bei der Gestaltung von Web-Auftritten auf eine geeignete Farbwahl und ausreichenden Kontrast von

Vorder- und Hintergrundfarben zu achten. Es sollte darauf Wert gelegt werden, Rot-/Grün-Kombinationen zu vermeiden, da die meisten der farbenblinden Benutzer Probleme haben, diese Farben zu unterscheiden. Auch Kombinationen von Blau und Grün können einigen Menschen Probleme bereiten. Es soll allerdings nicht grundsätzlich auf solche Farben verzichtet werden. Schwierigkeiten sind nur dann zu erwarten, wenn diese Farbkombinationen oder Abstufungen dieser Farben in direkter Nähe zueinander eingesetzt werden.

Auf blinkende Inhalte oder Laufschriften ist gänzlich zu verzichten, zumal die entsprechenden Elemente `<blink>` und `<marquee>` nicht in den HTML-Spezifikationen des W3C vorgesehen sind. Für fehlsichtige Anwender empfiehlt es sich, einen möglichst hohen Kontrast zu wählen. Auch die Möglichkeit, Farben zu invertieren, zum Beispiel eine Präsentation mit der Vordergrundfarbe Weiß und Schwarz als Hintergrundfarbe, sollte bei der Entwicklung barrierefreier Web-Auftritte in Erwägung gezogen werden.

Realisieren lassen sich solche Anforderungen durch den Einsatz eines Style-Switchers, welcher dem Benutzer die Möglichkeit gibt, verschiedene Präsentationen zu wählen. Durch die Trennung von Inhalt und Gestaltung ist es mit Hilfe von Skript-Sprachen kein Problem, dies umzusetzen. Ein wichtiger Punkt ist die Wahl der Schriftgrößen. Anhand von Stylesheets lassen sich Schriften mit relativen Einheiten bemaßen. Das ermöglicht dem Benutzer, entweder mit der entsprechenden Browserfunktion oder durch einen, auf dem Web-Auftritt vorhandenen, Schriftgrößendialog die gewünschte Schriftgröße zu wählen. Auf eine statische Bemaßung von Schriften sollte also verzichtet werden. Blinden Benutzern kommt eine sinnvolle und logische Strukturierung des Inhaltes zu Gute. Sie nutzen meist alternative Ausgabegeräte wie Braillezeilen und Screenreader, welche den Inhalt eines Web-Auftrittes linearisieren, also Zeile für Zeile ausgeben. Diese Linearisierung kann sehr problematisch sein, wenn für die Gestaltung eines Auftrittes Frames oder Tabellen genutzt werden. Da Frames eine Kombination von einzelnen autarken Seiten darstellen, stehen alternative Ausgabegeräte vor großen Problemen. Suchmaschinen und Screenreader sind meist nicht in der Lage, Frames zu verarbeiten und geben diese als leere Dokumente aus oder aber können nicht erkennen, in welchem Zusammenhang die einzelnen Frames stehen. Auf den Einsatz dieser Technik sollte also nach Möglichkeit verzichtet werden. Wird sie dennoch eingesetzt, müssen die einzelnen Seitenteile eindeutig benannt sein. Auch sollte ein **NOFRAMES**-Bereich existieren, damit Browsingtechnologien die Frames nicht unterstützen, entsprechend reagieren können.

Tabellen sollten nur dann eingesetzt werden, wenn sie auch Daten beinhalten. Oftmals werden diese Elemente aber zur Präsentation genutzt, nicht selten mit mehreren, in sich verschachtelten Ebenen. Sehbehinderte Benutzer haben in diesen Fällen kaum eine Möglichkeit, die Seitenstruktur zu erkennen, zumal durch Screenreader und Textbrowser die Inhalte von Tabellen zeilenweise ausgegeben werden.

Wenn Datentabellen zum Einsatz kommen, müssen Spalten und Zeilen einander zuzuordnen sein. Dies lässt sich durch die Verwendung von Achsenbeschriftungen und die Zuordnung der entsprechenden Datenzellen erreichen. Bilder und Grafiken sind für blinde Benutzer unzugänglich. Da jedoch nicht auf den Einsatz solcher Gestaltungselemente verzichtet werden soll, ist dafür Sorge zu tragen, für jedes grafische Element, welches unabdingbar für das Verständnis des Inhaltes ist, eine alternative textuelle Beschreibung vorzusehen.

Das gilt auch für multimediale Inhalte, wie zum Beispiel Animationen und Filme. Hier muss

zudem darauf geachtet werden, die textuellen Alternativbeschreibungen synchron zum Ablauf ausgeben zu lassen. Eines der wichtigsten Elemente eines Web-Auftrittes ist die Navigation. Oftmals werden Navigationselemente ausschließlich mit Hilfe von Grafik dargestellt. In solchen Fällen haben blinde Benutzer keine Möglichkeit, mit dem Auftritt zu interagieren und die Informationen zu nutzen. Navigationselemente sollten daher immer als Text dargestellt werden. Zudem ist darauf zu achten, dass Änderungen von solchen Elementen, beispielsweise die Anzeige des aktiven Links, nicht allein durch Farben hervorgehoben werden. Hilfreich sind auch Breadcrumb-Navigationen, welche dem Benutzer immer eine Orientierung auf dem gesamten Informationsangebot ermöglichen und Sprungmarken, um Navigationen oder Kopfelemente überspringen zu können und sofort zum eigentlichen Inhalt zu gelangen.

3.3.2. Hörgeschädigte Benutzer

Für hörgeschädigte Benutzer stellen vor allem Audio-Inhalte auf Web-Seiten erhebliche Barrieren dar. In absehbarer Zukunft werden multimediale Inhalte auf Web-Auftritten eine immer größere Rolle spielen. Deshalb muss, ähnlich wie bei Menschen mit Sehbehinderungen, beim Einsatz solcher Elemente auch immer eine textuelle Alternativbeschreibung vorhanden sein.

Auch Animationen und Filme sollten mit entsprechenden Alternativen versehen sein, da ansonsten der Zugang zu den Informationen solcher Inhalte für hörgeschädigte Menschen nicht möglich ist. Außerdem ist beim Einsatz von multimedialen Elementen darauf zu achten, dass Text und Bild synchron zueinander ablaufen.

3.3.3. Benutzer mit motorischen Einschränkungen

Benutzer mit motorischen Einschränkungen sind auf Eingabegeräte wie Sensoren, Großfeldtastaturen oder Sticks angewiesen.

Daher ist es erforderlich, Navigationselemente durch diese Eingabegeräte nutzbar zu machen. Es sollten alle Verknüpfungen anhand einer logischen Reihenfolge nummeriert werden. Das **tabindex**-Attribut, das in HTML spezifiziert ist, bietet genau diese Möglichkeit und ermöglicht den schnellen Wechsel zwischen Navigationselementen und Links über die Tabulator-Tasten.

Weiterhin ist es empfehlenswert, wichtige Navigationselemente, beispielsweise die Hauptnavigation, mit Kurztasten zu belegen, so dass diese einfach per Tastendruck aufgerufen werden können.

Probleme stellen dynamische Navigationsstrukturen dar, wenn sie zum Beispiel durch JavaScript erzeugt werden können. Diese lassen sich erst bei genauer Positionierung des Mauszeigers ausklappen, was für Menschen mit motorischen Beeinträchtigungen problematisch sein kann. Auf solche Elemente sollte daher grundsätzlich verzichtet werden.

3.3.4. Sonstige Barrieren

Neben den oben genannten Problemen gibt es noch einige weitere Hindernisse, die bei der Entwicklung von Web-Auftritten entstehen können. Ein wichtiger Punkt ist dabei die Wahl der natürlichen Sprache. Viele Benutzer haben Probleme, komplexe und tief verschachtelte Formulierungen zu erfassen. Aus diesem Grund ist bei der Gestaltung von barrierefreien Informationsangeboten auf eine kurze und prägnante Ausdrucksweise zu achten. Die Sprachwahl sollte über den gesamten Web-Auftritt hinweg konsistent sein.

Wie bereits erwähnt, muss auf blinkende Elemente und Laufschriften verzichtet werden. Bei Menschen mit fotosensitiver Epilepsie können bestimmte Kombinationen von Farbe und Blinkfrequenz Anfälle hervorrufen. Doch auch sonst ist ein Flimmern von Bildschirminhalten höchst unangenehm. Über animierte Inhalte, Filme und sich selbständig bewegende Objekte muss der Benutzer immer die Kontrolle haben, sie beispielsweise anhalten oder abbrechen können. Dabei darf es jedoch zu keinem Informationsverlust kommen.

Der Einsatz von JavaScript zur Darstellung von Navigations- oder Inhaltselementen sollte vermieden werden. Screenreader und Textbrowser verarbeiten solche Skripts nicht. Hinzu kommt, dass die Verwendung von JavaScript in den meisten Browsern abgeschaltet werden kann. In diesem Fall sind solche Web-Auftritte nicht mehr nutzbar.

3.4. Alternative Ein-/Ausgabesysteme

Für Menschen mit Behinderungen oder Beeinträchtigungen gibt es eine Vielzahl von Geräten und Programmen, die ihnen den Umgang mit Informationssystem erleichtern. Während viele davon einen hohen finanziellen Aufwand bedeuten, sind andere, vor allem Software, frei erhältlich oder sogar in bestehenden Betriebssystemen integriert.

Im Folgenden werden einige dieser Geräte und Programme vorgestellt.

3.4.1. Hardware

Für sehbehinderte Menschen und Blinde kommen vor allem Braillezeilen und Vergrößerungswerkzeuge, die auch als Hardware, also unabhängig von der eingesetzten Software, verfügbar sind, in Betracht. Braillezeilen können in Verbindung mit Screenreadern Bildschirminhalte darstellen. Auch gibt es Braille-Tastaturen, welche anstelle von Standard-Tastaturen die Eingabe von Daten erleichtern sollen.



Abbildung 1. Braillezeile [1]

Braillesysteme arbeiten mit beweglichen Stiften. Jeweils acht dieser Stifte stellen ein Zeichen in Brailleschrift dar, wobei abhängig von der Größe des Systems bis zu 80 Zeichen gleichzeitig gelesen und ausgegeben werden können. Inzwischen sind selbst Braillesysteme für Organizer, Handys und Notebooks erhältlich. Auch für grafische Darstellungen sind Braillezeilen entwickelt worden, bei denen der Benutzer über ein Feld mit Taststiften die wichtigsten Elemente der Grafik ertasten kann.

Motorisch behinderte Nutzer sind vor allem auf alternative Eingabegeräte angewiesen. Auch hier gibt es eine Vielzahl verschiedener Systeme. Als Maus-Ersatz eignen sich Tastenmäuse, bei denen der Mauszeiger über Richtungstasten bewegt wird. Auch Joysticks erfüllen diese Aufgabe.



Abbildung 2. Großfeldtastatur [2]

Weitere Eingabegeräte sind Großfeldtastaturen, bei denen die Tastendurchmesser bis zu 2,6cm betragen. Sie eignen sich für Benutzer, die Probleme haben, normale oder kleine Tastaturen zu bedienen. Bei schweren motorischen Einschränkungen kommen Sensoren zum Einsatz. Diese können, je nach Grad und Art der Behinderung, zum Beispiel mit Füßen, Knien, Handflächen oder Ellenbogen, aber auch mit Hilfe des Mundes bedient werden und so vollständig die Maus ersetzen.

3.4.2. Software

Auch Softwaresysteme, welche die Zugänglichkeit von Informationssystemen unterstützen sollen, gibt es in vielfältigen Ausführungen. Vor allem für Screenreader in Verbindung mit Web-Auftritten ist es hilfreich, die darzustellenden Informationen in Textform vorliegen zu haben. Für diesen Zweck können Textbrowser eingesetzt werden, welche die gewünschten Web-Seiten als reinen Text anzeigen und grafische Elemente einzig anhand ihrer Alternativbeschreibungen darstellen. Auch Stylesheets werden von Textbrowsern ignoriert.

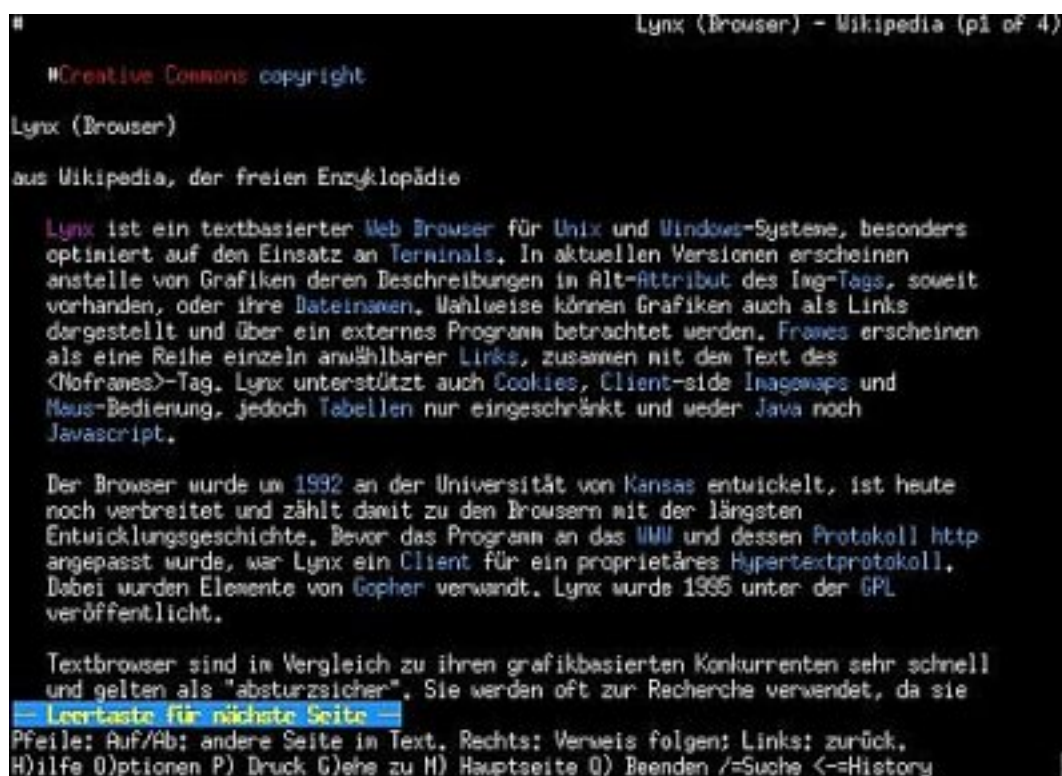


Abbildung 3. Textbrowser Lynx [3]

Zudem sind diese Browser allein durch Tastaturen zu bedienen, was auch motorisch eingeschränkten Benutzergruppen zu Gute kommt.

Screenreader geben Bildschirminhalte wahlweise auf Braillezeilen oder als Sprachausgabe wieder. Sie sind damit ein universell einsetzbares Mittel für sehbehinderte und hörgeschädigte Benutzer. Weitere Software-Lösungen zur Förderung der Barrierefreiheit sind beispielsweise Bildschirmlupen oder akustische und optische Rückmeldungen bei Systemereignissen oder Benutzereingaben. Diese Funktionen sind in gängigen Betriebssystemen meist vorhanden.

Für verbesserte Zugänglichkeit in Web-Auftritten bieten zudem die meisten grafischen Browser entsprechende Optionen an, mit denen der Benutzer zum Beispiel Farben und Schriftgrößen festlegen kann.

3.5. Fazit

Barrierefreie Web-Seiten stellen höhere Ansprüche an den Entwickler und bedeuten vor allem einen größeren Aufwand als die Entwicklung von Seiten, die nicht nach den Prinzipien der Barrierefreiheit entworfen werden.

Dies resultiert daraus, dass die Einhaltung der Richtlinien die Verwendung der neuesten Webtechnologien erfordert, was oftmals eine höhere Einarbeitungszeit zur Folge hat. Außerdem muss trotz des Einsatzes neuester Techniken die Kompatibilität und damit die Nutzbarkeit von Web-Auftritten auch mit älteren oder alternativen Browsingtechnologien gewährleistet sein.

Für Entwickler barrierefreier Informationsangebote stellt sich zudem das Problem, Ansprüche und Bedürfnisse behinderter und beeinträchtigter Menschen zu erkennen und daraus Lösungen zu finden, die diesen Ansprüchen gerecht werden. Trotzdem bieten Web-Auftritte, die im Hinblick auf Barrierefreiheit entwickelt wurden, eine Vielzahl von Vorteilen. Dies sind vor allem die uneingeschränkte Zugänglichkeit durch jede Benutzergruppe und die vereinfachte Pflege und Wartung.

Das Hauptaugenmerk sollte demnach nicht nur auf Benutzergruppen gerichtet sein, die aufgrund von Behinderungen beeinträchtigt sind, sondern auf alle Benutzer. Insbesondere gilt das für ältere Menschen und Menschen, die durch Krankheit oder andere Umstände temporär beeinträchtigt sind.

Es ist auch nicht immer möglich und oftmals auch nicht notwendig, allen Forderungen von Richtlinien nachzukommen. Vielmehr sollte dafür Sorge getragen werden, ein Verständnis für die Belange der verschiedenen Benutzergruppen zu erlangen und durch Einhaltung von Spezifikationen und Empfehlungen eine möglichst große Anzahl von Barrieren zu vermeiden.

4. Richtlinien und Empfehlungen

4.1. Überblick über die Richtlinien der WAI

Die WAI stellt eine Teilorganisation des W3C dar und hat es sich zur Aufgabe gemacht, Richtlinien zur Entwicklung und Gestaltung barrierefreier Web-Auftritte zu erarbeiten. Diese Richtlinien (WCAG) existieren inzwischen in zwei Versionen.

Während sich die erste Version (WCAG1.0) vor allem an Entwickler und Gestalter von Web-Auftritten wendet, findet man in der zweiten Ausführung (WCAG2.0) auch Empfehlungen für Unternehmen und Gesetzgeber. Inzwischen ist Version 1 der Richtlinien in die Gesetzgebungen einer Vielzahl von Ländern eingeflossen und bildet die Basis vieler Validatoren für Barrierefreiheit, wie zum Beispiel "*Cynthia Says*" und "*Bobby*". Version 2 ist momentan noch nicht als offizieller Standard verabschiedet, enthält aber im Vergleich zur ersten Version weniger technische Fakten und ist somit leichter verständlich.

Beide Richtlinien bestehen aus drei verschiedenen Prioritätsstufen, welche wiederum in verschiedene Checkpunkte aufgeschlüsselt sind. Dabei stellt die Einhaltung aller, in einer

Stufe enthaltenen, Checkpunkte die Erfüllung der jeweiligen Stufe dar. Wird einer der Checkpunkte fehlerhaft umgesetzt, so kann die entsprechende Prioritätsstufe nicht erreicht werden. Die Inhalte der Richtlinien lassen sich folgendermaßen grob beschreiben:

- *Prioritätsstufe 1* - Die Richtlinien dieser Stufe müssen vom Entwickler eingehalten werden. Andernfalls ist es bestimmten Benutzergruppen unmöglich, auf alle Inhalte des Web-Auftrittes zuzugreifen. Sie enthält damit die Muss-Kriterien.
- *Prioritätsstufe 2* - Diese Stufe sollte vom Entwickler erfüllt werden, ansonsten ist es für eine oder mehrere Benutzergruppe(n) schwierig, auf alle Inhalte des Web-Auftrittes zuzugreifen. Die Punkte dieser Stufe werden im Allgemeinen als Soll-Kriterien bezeichnet.
- *Prioritätsstufe 3* - Die Richtlinien dieser Stufe können vom Entwickler umgesetzt werden. Sollte dies nicht der Fall sein, so kann es unter Umständen für bestimmte Benutzergruppen etwas schwieriger sein, an alle Informationen eines Web-Auftrittes zu gelangen. Somit stellen die Checkpunkte dieser Stufe die Kann-Kriterien dar.

Um die Einhaltung der verschiedenen Stufen zu signalisieren, entsprechen diese so genannten Konformitätsstufen:

- *Konformitätsstufe "A"* - wenn alle Checkpunkte der Prioritätsstufe 1 erfüllt wurden,
- *Konformitätsstufe "AA" oder "Double A"* - wenn alle Punkte der Prioritätsstufen 1 und 2 erfüllt wurden, sowie
- *Konformitätsstufe "AAA" oder "Triple A"*, falls alle Checkpunkte aus allen Prioritätsstufen erfüllt wurden.

Die Einhaltung aller Stufen ist zwar generell erwünschenswert, lässt sich aber oftmals nicht garantieren, da die erste Version der Richtlinien bereits im Jahr 1999 verabschiedet wurde und somit zum Teil nicht mehr dem Stand der Technik entspricht sowie möglicherweise Widersprüche zur gewünschten Funktionalität des Web-Auftrittes auftreten.

4.2. Analyse und Beschreibung der Richtlinien der WAI

In dieser Arbeit soll von der ersten Version der Richtlinien (WCAG1.0) der WAI ausgegangen werden, da diese momentan die Grundlage für die gängigsten abgeleiteten Richtlinien bildet.

Die WCAG1.0 besteht aus zwei Teilen, wobei der erste Teil die grundlegende Funktionalität von Web-Auftritten definiert. Insbesondere betrifft dies die Unabhängigkeit von verwendeten Browsingtechnologien und Ein-/Ausgabegeräten.

Der zweite Teil bezieht sich auf den eigentlichen Inhalt der Auftritte, also auf Formulierungen und Navigationselemente.

Insgesamt besteht die WCAG1.0 aus 14 Teil-Richtlinien, welche wiederum verschiedene Checkpunkte enthalten. Der strukturelle Aufbau lässt sich folgendermaßen definieren:

- Richtlinie mit Titel und Nummer,
- Beschreibung der Richtlinie
mit Ziel und betroffener Benutzergruppe,
- Checkpunkte zur Erfüllung der Richtlinie
jeweils mit Nummer, Beschreibung des Punktes, Prioritätsstufe, Referenzen zu verwandten Checkpunkten und Hinweisen zu technischen Lösungsmöglichkeiten.

Allgemein setzt die WAI den Einsatz von W3C-konformen Markup-Sprachen voraus, wobei Inhalt und Präsentation strikt getrennt und eine Unabhängigkeit der eingesetzten Browsingtechnologie garantiert werden soll.

Aus diesem Grund wird vom W3C der Einsatz von XHTML/XML in Verbindung mit "Cascading Stylesheets" (CSS) empfohlen.

4.2.1. Alternativen für Audio- oder visuelle Inhalte

Die erste Richtlinie der WCAG1.0 definiert die Verwendung von Audio- oder visuellen Inhalten. Dabei soll nicht grundsätzlich auf Bilder, Audio-Dokumente oder Animationen verzichtet, sondern vielmehr für solche Elemente ein bedeutungsäquivalenter Alternativtext durch die HTML-Attribute **alt** oder **longdesc** vorgesehen werden. Allerdings wird das **longdesc**-Attribut von den meisten Browsern nicht unterstützt. Bei langen Alternativtexten, die auf andere Seiten ausgelagert werden müssen, empfiehlt es sich daher, entsprechende Verknüpfungen neben den grafischen Elementen anzubieten.

Besonders wichtig ist bei zeitabhängigen Inhalten, wie zum Beispiel Animationen, Filmen und Audio-Dokumenten, der synchrone Ablauf der Alternativtexte. Bildbereiche, die Verknüpfungen enthalten, so genannte Image-Maps, sind immer mit alternativen Textverknüpfungen zu versehen. Anderenfalls ist es Benutzern, die keinen grafischen Browser verwenden können, nicht möglich, im Bild vorhandene Verknüpfungen zu nutzen.

Diese Forderungen entsprechen der höchsten Priorität, müssen also erfüllt werden, um die minimale Stufe der Barrierefreiheit zu garantieren.

4.2.2. Einsatz von Farben

Die zweite Richtlinie befasst sich mit dem Einsatz von Farben für die Darstellung von Informationen.

Zunächst müssen alle Elemente, die farblich präsentiert werden, auch ohne den Einsatz von Farben verfügbar sein. Dies betrifft vor allem die Gestaltung von Navigationselementen. Auch diese Forderung entspricht der höchsten Prioritätsstufe.

Des Weiteren sollten Vorder- und Hintergrundfarben so gewählt sein, dass auch farbenblinden Benutzern genügend Kontrast zur Unterscheidung von Vorder- und Hintergrund geboten wird. Dabei muss auch darauf geachtet werden, dass sich möglicherweise eingebettete Bilder gut vom Hintergrund abheben.

4.2.3. Markup und Stylesheets

Auch die Verwendung der Markup-Sprachen und Stylesheets, in denen Web-Auftritte entwickelt werden sollten, wird vom W3C definiert. Bei entsprechender Verfügbarkeit ist der Einsatz von speziellen Markup-Sprachen der Verwendung von Grafik vorzuziehen. Ein Beispiel ist die Verwendung von MathML für mathematische Inhalte.

Außerdem sollen die entwickelten Seiten den Vokabularen der verwendeten Dokument-Typen genügen. Zu überprüfen ist dies zum Beispiel mit Hilfe von Validatoren wie dem Markup-Validator des W3C.

Weitere Checkpunkte besagen, dass in Stylesheets relative Angaben, wie **em** oder prozentuale Maße, für Schriftgrößen und Layouts bevorzugt eingesetzt werden sollen, um die Lesbarkeit auf kleinen Bildschirmen zu gewährleisten. Manche Browser lassen zudem bei Verwendung absoluter Maßangaben keine Skalierung zu.

Die Checkpunkte dieser Richtlinie entsprechen den Prioritätsstufen 2 und 3, sollten jedoch trotzdem erfüllt werden, da der Einsatz valider Dokumente sowie die Trennung von Inhalt und Präsentation für alle Benutzer von Vorteil sind und auch die Wartung und Pflege von Informationsangeboten erheblich vereinfachen.

4.2.4. Abkürzungen und fremdsprachige Inhalte

Die im Internet-Auftritt verwendete natürliche Sprache sollte im Markup deklariert sein, damit alternative Ausgabegeräte die entsprechenden Vokabulare nutzen können. Für diesen Zweck empfiehlt sich Attribut **xml:lang**, welches für XHTML/XML verfügbar ist.

Außerdem verfügen diese Ausgabegeräte oftmals über die Möglichkeit, auf gekennzeichneten Wechsel der natürlichen Sprache zu reagieren, zum Beispiel, wenn sich ein fremdsprachiger Begriff im Text befindet. Auch hierfür bietet sich oben genanntes Attribut an.

Eine Kennzeichnung der Sprache des Web-Auftrittes ist auch sinnvoll, damit Suchmaschinen aus dem Informationsangebot des Internet nach Web-Auftritten in einer bestimmten Sprache suchen können. Fehlen solche Angaben, ist dies meist nicht möglich.

Daher fordert diese Richtlinie, Sprachwechsel im Dokument über entsprechende Attribute wie **xml:lang** zu kennzeichnen, Abkürzungen mit zugehörigen Markup-Elementen, wie zum Beispiel **<abbr>** und **<acronym>**, zu deklarieren und für den gesamten Web-Auftritt die natürliche Sprache zu definieren.

4.2.5. Verwendung von Tabellen

Die Verwendung von Tabellen für die Festlegung der Präsentation ist eine weit verbreitete Technik bei der Entwicklung von Web-Auftritten. Da aber alternative Ausgabegeräte, wie zum Beispiel Screenreader und Braillezeilen, Tabellen linearisieren, also zeilenweise ausgeben, ist die Verwendung von Layout-Tabellen nicht erwünscht.

Vielmehr sollten hier die Möglichkeiten von Stylesheets genutzt werden. Für tabellarische Inhalte fordert diese Richtlinie, dass Zeilen und/oder Spalten stets in Verbindung zur Tabellenüberschrift gebracht werden können. Dies lässt sich beispielsweise durch Verwendung von **axis**-, **scope**- und **headers**-Attributen erreichen, die in HTML zur Verfügung stehen.

Diese Forderung entspricht der höchsten Prioritätsstufe und kommt vor allem Benutzern zu Gute, die nur einen Teil der Tabelle zeitgleich erfassen können, was sowohl bei Braillezeilen und Screenreadern, als auch bei kleinen Bildschirmen der Fall ist.

4.2.6. Neueste Webtechnologiestandards

Um Problemen vorzubeugen, die mit der Verwendung älterer Technologien verbunden sind, werden vom W3C im Allgemeinen die neuesten Standards empfohlen.

Die Verwendung veralteter Elemente und Attribute sollte möglichst ausgeschlossen werden. Dadurch steigt aber die Gefahr, dass Web-Auftritte das Kriterium der Unabhängigkeit von verwendeten Browsingtechnologien nicht mehr erfüllen. Spezielle Ein-/Ausgabegeräte für beeinträchtigte Menschen sind mit hohen Anschaffungskosten verbunden, weshalb sie oft mehrere Jahre im Einsatz bleiben. Solche Systeme sind dann meist nicht in vollem Maße mit aktuellen Webtechnologien kompatibel. Daher muss ungeachtet des Einsatzes neuester Technologien und der Vermeidung vergangener Standards garantiert werden, dass der entwickelte Web-Auftritt auch auf diesen Geräten oder älteren Browsern zugänglich ist.

Entwickler sind durch diese Richtlinie daher dazu angehalten, die Lesbarkeit ihrer Webauftritte auch ohne Stylesheets, eingebundene Skripts oder Applets und dynamische Inhalte zu gewährleisten. Diese Forderungen liegen in der höchsten Prioritätsstufe.

Des Weiteren sollte dafür Sorge getragen werden, Skripts und Applets unabhängig vom verwendeten Eingabegerät zu entwerfen, dynamische Funktionen also auch ohne Maus bedienbar zu machen.

4.2.7. Zeitsensitive Inhalte

Die siebte Richtlinie definiert die Verwendung von beweglichen und zeitsensitiven Inhalten, wie zum Beispiel blinkende oder automatisch scrollende Objekte und auch Weiterleitungen von Dokumenten. Nach Möglichkeit sollten blinkende Inhalte vermieden werden, sie müssen zumindest aber kontrolliert werden können. Bewegliche Objekte wie Laufschrift sollte man anhalten können.

Der Benutzer muss also jederzeit in der Lage sein, das Verhalten solcher Objekte zu kontrollieren.

Hinzu kommt, dass die Verwendung der Elemente **<blink>** und **<marquee>** für blinkende Inhalte und Laufschriften zwar oft auf Web-Auftritten zu finden, in den W3C-HTML-Spezifikationen jedoch nicht definiert ist.

4.2.8. Unabhängigkeit von Skripts und Applets

In den Web-Auftritt eingebettete Skripts und Applets sollten unabhängig von verwendeten Ein-/Ausgabegeräten funktionieren. Oft werden mit Hilfe von Skripts dynamische Navigationsstrukturen entwickelt, welche jedoch meist nur mit der Maus bedient werden können. Auch findet man auf vielen Web-Auftritten eingebettete multimediale Inhalte, die entsprechende Plugins voraussetzen. Alternative Ausgabegeräte haben dann keine Möglichkeit, den Inhalt solcher Applets zu verarbeiten.

Der Einsatz solcher Elemente sollte demnach möglichst vermieden, zumindest jedoch die Zugänglichkeit zu den enthaltenen Informationen gewährleistet werden.

Elemente, die für den Web-Auftritt unabdingbar sind, wie zum Beispiel Navigationsstrukturen, müssen so gestaltet sein, dass ihre Funktionsweise unabhängig vom verwendeten Ein-/Ausgabegerät garantiert werden kann.

4.2.9. Unabhängigkeit von Ein-/Ausgabegeräten

Diese Richtlinie fordert die Unabhängigkeit des Web-Auftrittes und seiner Funktionalität von den verwendeten Ein-/Ausgabegeräten. Benutzer, die über kein Zeigergerät wie Maus oder Stift verfügen, können bei Formularen oder Navigationselementen Schwierigkeiten haben, wenn diese keine Aktivierung durch Eingabegeräte wie Tastaturen oder Spracherkennungssysteme zulassen.

Ein wichtiger Punkt dieser Richtlinie besagt, dass Image-Maps, also Bilder, in denen Verknüpfungen existieren, immer clientseitig ausgeführt werden müssen. Ansonsten ist die Unabhängigkeit von verwendeten Ein-/Ausgabegeräten nicht in jedem Fall garantiert, da das Bild möglicherweise mit anderen Koordinaten dargestellt wird.

In weiteren Checkpunkten dieser Richtlinie wird empfohlen, Verknüpfungen, Formulare und Objekte in einer logischen Reihenfolge über Tab-Indizes zu nummerieren. Das entsprechende HTML-Attribut ist **tabindex**. Wichtige Verknüpfungen, wie zum Beispiel die Hauptnavigation, sollten über Kurztasten (Accesskeys) erreichbar sein, was durch die Nutzung des **accesskey**-Attributes in HTML ermöglicht wird.

4.2.10. Übergangslösungen

Die WAI des W3C geht bei der zehnten Richtlinie nicht davon aus, dass sie in Zukunft notwendig sein wird, da sie sich vor allem auf ältere Browsingtechnologien bezieht.

Es wird empfohlen Pop-Ups zu vermeiden, da diese nicht von allen Benutzern wahrgenommen werden können. Außerdem sollten Eingabeelemente von Formularen immer mit einer Kennzeichnung assoziiert sein, um die Zugehörigkeit zu einem Formular zu

identifizieren.

Ein weiterer Checkpunkt empfiehlt, eine lineare Textvariante von Inhalten zur Verfügung zu stellen, falls mehrspaltiger Text verwendet wird. Außerdem sollte es möglich sein, Listen von Verknüpfungen direkt zu überspringen, da diese von alternativen Ausgabegeräten linearisiert, also zeilenweise ausgegeben werden. Bei komplexen Navigationsstrukturen kann das sehr lange dauern. Der Benutzer ist ohne Sprungmarken nicht in der Lage, sofort an die gewünschten Informationen zu gelangen.

4.2.11. Nutzung von W3C-Technologien und -Richtlinien

Mit den Checkpunkten dieser Richtlinie verweist das W3C auf eigene Standards und Technologien. Sie fasst alle anderen Richtlinien des W3C, nicht nur die der WAI, allgemein zusammen.

Die Einhaltung dieser Kriterien ist nicht nur empfehlenswert, da das W3C internationale Richtlinien geschaffen hat, welche weitestgehend von den gängigen Browsingtechnologien unterstützt werden. Anders als bei Fremdformaten sind für W3C-konforme Web-Auftritte keine Erweiterungen oder spezielle Programme für Browser notwendig.

Zu nutzen sind Technologien des W3C, insofern vorhanden, und anzuwenden sind die Empfehlungen der WAI.

Hingewiesen sei darauf, dass es den Richtlinien der WAI zuwider läuft, eine barrierefreie Alternative zu einem nicht-barrierefreien Web-Auftritt zu schaffen. Dies wird damit begründet, dass Alternativ-Auftritte in unregelmäßigeren Abständen gepflegt werden. In diesem Fall sind die aktuelleren Informationen für bestimmte Benutzergruppen nicht erreichbar. Außerdem stellt ein getrennter Zugang für beeinträchtigte Benutzer eine Diskriminierung dar, die im Sinn der Barrierefreiheit ausgeschlossen sein muss.

4.2.12. Kontext und Orientierung

Vor allem auf komplexen Web-Auftritten ist es hilfreich, Inhalte in logische Blöcke zu unterteilen. Diese Blöcke sollten über entsprechende Attribute bezeichnet werden, um die Orientierung im Dokument zu vereinfachen.

Vor allem die Verwendung von Frames ist im Zusammenhang mit barrierefreier Entwicklung von Web-Auftritten problematisch. Textbrowser und Screenreader stellen Frames oftmals als leere Dokumente dar. Andere alternative Ausgabegeräte erfassen Frames immer als getrennte Seiten, so dass kein logischer Zusammenhang hergestellt werden kann. Das kann beispielsweise dann der Fall sein, wenn ein Frame die Navigation enthält und ein anderes den eigentlichen Seiteninhalt. Umgehen lässt sich dieser Umstand nur über die Beschreibung des kontextuellen Zusammenhangs der Frames mit Hilfe der entsprechenden Markup-Attribute wie **title**, **longdesc** und **alt** in HTML.

Auch Formulare und ihre Eingabelemente sollten mit den Möglichkeiten der Markup-Sprachen eindeutig als Block deklariert werden, wofür in HTML entsprechende **optgroup**-, **option**-, **field**- und **fieldset**-Elemente verfügbar sind.

HTML bietet außerdem die Elemente `<h1>` bis `<h6>` an, die für Überschriften und Seiten-Titel zu nutzen sind, da diese auch von Screenreadern und Textbrowsern erkannt und entsprechend hervorgehoben werden können.

4.2.13. Navigationselemente

Einer der wichtigsten Aspekte eines Web-Auftrittes ist die Navigation. Eine einfache, klare und logisch zusammenhängende Navigation erleichtert die Interaktion mit dem Informationssystem enorm. Auch dafür hat die WAI entsprechende Richtlinien verabschiedet.

Essentiell ist die eindeutige Identifizierung von Navigationselementen. Der Name einer Verknüpfung sollte sofort Klarheit über den zu erwartenden Inhalt der verknüpften Seite schaffen. Zudem sollten zusammengehörige Navigationselemente als solche identifiziert werden, beispielsweise über eine Überschrift.

Hilfreich können auch so genannte Breadcrumb- oder Trail-Navigtionen sein, die dem Benutzer immer die Möglichkeit geben, die genaue Position im Web-Auftritt zu ermitteln.

Auch Site-Maps, die einen Überblick über alle Seiten eines Web-Auftrittes geben, werden von der WAI empfohlen. Nach Möglichkeit sollten dabei Informationen über vorhandene Funktionen zur Unterstützung der Barrierefreiheit nicht fehlen, so zum Beispiel eine Auflistung vergebener Kurztasten. Der letzte Punkt betrifft Suchfunktionen in Web-Auftritten. Diese sind mit der Möglichkeit auszustatten, nach verschiedenen Kriterien suchen und somit auf unterschiedliche Fähigkeiten der Benutzer eingehen zu können.

4.2.14. Natürliche Sprache und Formulierungen

Die letzte Richtlinie der WAI definiert die Verwendung von Formulierungen und natürlicher Sprache. Es muss darauf geachtet werden, einfache und prägnante Formulierungen zu verwenden. Viele Benutzer haben Schwierigkeiten, komplexe und lange Sätze im Zusammenhang zu verstehen.

Wie bereits in der vierten Richtlinie gefordert, sollten Abkürzungen, nicht selbsterklärende Formulierungen und Wechsel der natürlichen Sprache mit den entsprechenden Elementen und Attributen der jeweiligen Markup-Sprache deklariert sein.

Zudem sollte dafür Sorge getragen werden, Sprachwahl, Satzbau und Formulierungen über den gesamten Web-Auftritt konsistent zu halten.

4.3. Richtlinien anderer Einrichtungen

Während die Richtlinien der WAI international anerkannt sind, gibt es auch in einzelnen Ländern Empfehlungen von verschiedenen Institutionen zur Entwicklung barrierefreier Informationsangebote. Diese beziehen sich oftmals nicht nur auf Internet-Auftritte, sondern auf alle öffentlich zugänglichen Angebote, die mit Hilfe der Informationsverarbeitung erstellt wurden. Im Folgenden sollen die Richtlinien der Bundesrepublik Deutschland und des

Bundeslandes Brandenburg vorgestellt werden.

4.3.1. Barrierefreie Informationstechnik-Verordnung

In der Bundesrepublik Deutschland wurde durch das bundesweite Behindertengleichstellungsgesetz eine Verordnung für barrierefreie Informationstechnik (BITV) erlassen, die sich auf zwei Prioritätsstufen beschränkt. Diese sind aus den Konformitätsstufen der WAI abgeleitet. Dabei entspricht Prioritätsstufe 1 den ersten beiden Stufen der WCAG1.0 und Prioritätsstufe 2 der dritten Stufe der WCAG1.0.

Alle Bundesbehörden müssen ihre Informationsangebote so gestalten, dass sie der Prioritätsstufe 1 der BITV entsprechen. Informationsangebote sind dabei Web-Auftritte, öffentlich zugängliche Intranet-Auftritte und grafische Programmoberflächen. Desweiteren sind "zentrale Navigations- und Einstiegsangebote" ([4], §3 Abs. 2) so zu gestalten, dass sie die zweite Prioritätsstufe erfüllen. Spätestens zum Ende des Jahres 2005 haben alle Bundesbehörden die Bedingungen der BITV umzusetzen.

Beide Prioritätsstufen enthalten zunächst die gleichen Anforderungen, unterscheiden sich jedoch in den Bedingungen. Die Stufen sind in folgende Anforderungspunkte gegliedert.

1. Es sind textuelle Alternativen für Audio- und visuelle Inhalte anzugeben,
2. Texte und Grafiken müssen auch ohne Farben zugänglich sein,
3. Markup und Stylesheets haben den formalen Spezifikationen zu entsprechen,
4. Wechsel der natürlichen Sprache und Abkürzungen sind zu kennzeichnen,
5. Tabellen sind nur für die Darstellung tabellarischer Daten einzusetzen und mit entsprechenden Elementen zu beschreiben,
6. die Nutzbarkeit von Web-Auftritten muss auch dann möglich sein, wenn aktuelle Technologien nicht vom Browser unterstützt werden,
7. zeitsensitive Inhalte müssen durch den Nutzer kontrolliert werden können,
8. in Web-Auftritte eingebettete Benutzerschnittstellen müssen direkt zugänglich sein,
9. die Unabhängigkeit von Ein-/Ausgabegeräten ist zu gewährleisten,
10. auch ältere Browsingtechnologien müssen Zugang zum Auftritt haben,
11. die verwendeten Technologien sollen öffentlich gemacht werden und vollständig dokumentiert sein,
12. es sind Informationen zur Orientierung bereitzustellen,
13. Navigationselemente müssen übersichtlich und logisch aufgebaut sein und

14. sofern es dem Verständnis dient, sind weitere Maßnahmen zu ergreifen, zum Beispiel Ergänzung von Texten durch Audio-Dokumente.

Die Bedingungen der ersten Prioritätsstufe entsprechen insgesamt der WCAG1.0 der WAI. In der zweiten Prioritätsstufe werden einige Ergänzungen gemacht, die sicherstellen sollen, dass auch Hauptnavigationen und Einstiegsangebote zugänglich sind.

4.3.2. Brandenburgisches Behindertengleichstellungsgesetz

Das Land Brandenburg hat, wie einige andere Bundesländer auch, ein eigenes Behindertengleichstellungsgesetz (BbgBGG) erlassen. Der §9 des Gesetzes beschäftigt sich mit der Schaffung barrierefreier Informationssysteme. Auch hier werden öffentliche Einrichtungen des Landes verpflichtet, Web-Auftritte sowie öffentlich zugängliche grafische Programmoberflächen barrierefrei zu gestalten. Anzuwendende Richtlinien und Regeln sind hierbei die Anforderungen und Bedingungen der BITV.

4.4. Umsetzung der Richtlinien

Die Umsetzung der Richtlinien der WAI und auch der BITV stellt für Entwickler deutlich mehr Aufwand bei der Implementierung von Web-Auftritten dar. Nicht nur, dass neueste Webtechnologiestandards erlernt und eingesetzt werden sollen, auch muss ein gewisses Verständnis für die Belange und Interessen der verschiedenen Benutzergruppen vorhanden sein.

Bei der Entwicklung ist dann darauf zu achten, dass alle syntaktischen Regeln eingehalten, Richtlinien beachtet und umgesetzt sowie Inhalte logisch und prägnant aufgebaut werden. Oftmals resultiert daraus eine höhere Komplexität des Informationsangebotes, was dem Entwickler den Überblick über Quelltexte erschweren kann. Man bedenke beispielsweise die sinnvolle Verwendung von Tab-Indizes bei komplexen Auftritten. Hinzu kommt, dass bei vielen Informationsangeboten größter Wert auf eine ausgefallene Präsentation gelegt wird, welche oftmals mit älteren Technologien, wie zum Beispiel Tabellen und Frames, einfacher zu entwickeln ist. Die Richtlinien fordern jedoch die Auslagerung in Stylesheets, was bei komplexen Gestaltungselementen nicht immer einfach umzusetzen ist.

Im Folgenden werden die empfohlenen Webtechnologiestandards auf ihre Tauglichkeit für bestimmte Anforderungen in Verbindung mit Barrierefreiheit untersucht.

4.4.1. XML

Als ältester Standard der Webtechnologien präsentiert sich HTML. Ein großes Problem dieser Markup-Sprache ist jedoch der feste und begrenzte Satz von Elementen. Sie enthält außerdem Elemente und Attribute, welche ausschließlich für die Präsentation von Dokumenten im WWW vorgesehen sind. Während HTML auf der Metasprache SGML basiert, ist XML eine Teilmenge dieser Sprache. XML bietet die Möglichkeit, beliebige Daten bzw. Elemente zu definieren und zu nutzen. Beschrieben werden diese Elemente in DTDs oder auch XML-Schema. Sie lassen sich mit den Sprachen von XSL in andere

Strukturen, beispielsweise HTML oder in andere Datenformate, wie zum Beispiel PDF, wandeln.

Zu XSL gehören verschiedene Sprachen, mit denen XML-Dokumente transformiert werden können. XSL als solches beschreibt mit Hilfe von Stylesheets und Formatierungsanweisungen das jeweilige Dokument als Baum und basiert, wie auch XSLT, auf der Markup-Sprache XML.

XSLT bietet die Möglichkeit, XML-Dokumente in beliebige andere Baumstrukturen zu transformieren, beispielsweise in HTML. Dabei werden die Regeln, anhand derer die Umwandlung erfolgt, in XSLT-Stylesheets festgelegt. In diesen Stylesheets lassen sich eine Reihe von Anweisungen unterbringen, um das Dokument in das gewünschte Zielformat zu konvertieren. Dabei muss das Ziel nicht zwingend die optische Gestaltung sein, auch andere Datenstrukturen lassen sich erzeugen. Somit ist XSLT ein mächtiges Werkzeug, um XML-Dokumente zu verarbeiten.

Ein weiteres Mitglied der XSL-Familie ist XPath, welches direkt Teile der Baumstruktur von XML-Dokumenten adressieren kann. XPath ist eines der Mittel, in XSLT-Stylesheets bestimmte Dokument-Bereiche zu selektieren und zu formatieren.

Alle drei Teilsprachen lassen sich gemeinsam oder unabhängig voneinander verwenden. Für Web-Auftritte ist aber meist eine Kombination der Sprachen nötig.

Durch den Einsatz von XML und XSL lassen sich sehr komplexe und flexible barrierefreie Auftritte entwickeln. Alle Forderungen an Markup-Sprachen, die in den WCAG gestellt werden, lassen sich mit Hilfe dieser Sprachen vollständig umsetzen. Auch kann CSS mit XML und XSL in Verbindung gebracht werden, so dass einer ansprechenden Präsentation der Informationen nichts im Wege steht.

Ein großes Problem ist jedoch, dass XML und XSL noch nicht von allen Browsingtechnologien unterstützt werden, vor allem ältere Browser und Textbrowser können Dokumente dieser Art oft nicht verarbeiten. Dies widerspricht aber der Forderung, dass auf einem barrierefreien Web-Auftritt allen Benutzergruppen der Zugang zu allen Informationen mit allen möglichen Ein- und Ausgabegeräten sowie Browsingtechnologien gewährleistet sein muss.

Als Lösung bieten sich dann entweder der Einsatz von serverseitigen Skript-Sprachen wie PHP oder ASP an, die mit entsprechenden Erweiterungen die Transformation der XML/XSL-Dokumente in HTML vornehmen oder aber die Entwicklung des Web-Auftrittes in XHTML.

4.4.2. XHTML

HTML Version 4.01 wurde inzwischen durch XHTML, die Extensible Hypertext Markup Language, ersetzt. Diese Markup-Sprache basiert auf XML. Dadurch müssen bei der Entwicklung von Web-Auftritten mit XHTML die formalen Regeln der Markup-Sprache eingehalten werden, was beim HTML-Standard nicht unbedingt nötig ist, da viele Browser tolerant auf Syntaxfehler reagieren.

Die Einhaltung dieser Regeln, also das Entwickeln valider Web-Auftritte, ist vor allem im Hinblick auf die Barrierefreiheit von großem Vorteil. Syntaktisch korrekte Auftritte werden ohne Probleme von modernen Browsingtechnologien verarbeitet, während Fehler in der Syntax auf unterschiedlichen Ausgabegeräten zu nicht vorhersehbaren Problemen führen können.

Als weitere Neuerung hat das W3C Elemente und Attribute, die ausschließlich für die Präsentation von Web-Auftritten vorgesehen waren, aus XHTML ausgeschlossen. Dies wurde in XHTML1.0 vor allem in der STRICT-Variante durchgesetzt, die Varianten TRANSITIONAL und BASIC bieten hingegen Kompatibilität zu HTML. XHTML1.1 entspricht weitgehend der STRICT Variante aus Version 1.0 und enthält keine Kompatibilitäts-Modi.

Durch den Ausschluss der Präsentationselemente und -attribute ist eine Trennung von Inhalt und Gestaltung erforderlich. Dies ermöglicht es, Informationen unabhängig vom Ausgabegerät zu präsentieren. Für Benutzer, die Braille-Zeilen, Textbrowser oder Screenreader benutzen, sind Gestaltungselemente meist nicht relevant. Oftmals irritieren diese sogar, da sie, als Text gelesen, keinen logischen Zusammenhang mit den eigentlichen Informationen eines Web-Auftrittes bilden.

Da XHTML eine Weiterentwicklung des HTML-Standards darstellt, ist eine Vielzahl der Elemente, die bereits in HTML definiert sind, auch in der Beschreibung von XHTML zu finden. Dadurch haben Browsingtechnologien, die XHTML nicht verarbeiten können, die Möglichkeit, eine Analyse der Dokumentstruktur vorzunehmen und den entsprechenden Web-Auftritt als HTML zu verarbeiten. Zwar fehlen dann die Gestaltungselemente, die Informationen des Auftrittes sind aber trotzdem nutzbar. Für barrierefreie Web-Auftritte ist es erforderlich, Spezifikationen der verwendeten Markup-Sprache einzuhalten, also valide Dokumente zu erstellen. Da XHTML auf XML basiert, gilt dafür eine strengere Syntax als bei HTML. Die wichtigsten Forderungen sind folgende:

- Element- und Attributnamen müssen klein geschrieben sein,

Beispiel: `<option selected="selected">`

- Attributwerte müssen in Anführungszeichen stehen,

Beispiel: ``

- leere Attribute sind nicht erlaubt,

Beispiel: `<option selected>` ist nicht erlaubt. Statt dessen: `<option selected="selected">`

- alle Elemente müssen geschlossen werden, wobei leere Elemente mit einem Schrägstrich geschlossen werden können. Um die Kompatibilität mit älteren Browsern zu garantieren, sollte bei leeren Elementen darauf geachtet werden, ein Leerzeichen vor dem schließenden Schrägstrich anzugeben.

Beispiel: `<p>Absatz</p>
`

- In der Wurzel **<html>** muss immer der XML-Namensraum angegeben werden,
Beispiel: **<html xmlns="http://www.w3.org/1999/xhtml">**
- das Attribut **lang** zur Auszeichnung der natürlichen Sprache muss durch **xml:lang** ersetzt werden.

XHTML beruht auf dem Namensraumkonzept von XML. Somit existieren für verschiedene Anforderungsbereiche verschiedene Dokumenttypen bzw. Schemata, wie zum Beispiel MathML für mathematische Inhalte. Durch die Kombination unterschiedlicher Namensräume lassen sich individuelle Markup-Sprachen definieren. XHTML/XML sind daher mit abstrakten Datentypen in objektorientierten Programmiersprachen vergleichbar.

Nicht alle Browser sind allerdings in der Lage, solche Erweiterungen zu verarbeiten. In diesen Fällen haben sie die Möglichkeit, unbekannte Elemente als Text darzustellen, sie zu ignorieren oder die Installation eines Plug-Ins zu veranlassen. Dies stellt zwar eine Einschränkung der geforderten "Zugänglichkeit für alle Benutzergruppen" dar, jedoch sind selbst einige moderne Browser wie der Microsoft Internet Explorer zur Zeit nicht fähig, spezielle Dokumentenformate zu verarbeiten. Aus diesem Grund sollte momentan im Hinblick auf Barrierefreiheit auf Formate wie MathML oder SVG verzichtet, zumindest aber eine textuelle Alternative angeboten werden.

Alle Anforderungen der WCAG lassen sich mit XHTML erfüllen. Für jede Richtlinie, welche Checkpunkte beinhaltet, die sich auf verwendete Markup-Sprachen beziehen, bietet XHTML entsprechende Elemente oder Attribute.

Für die Verwendung von XHTML in der Entwicklung barrierefreier Web-Auftritte sprechen außerdem die Kompatibilität zu HTML und damit die Nutzbarkeit auf alternativen und älteren Browsingtechnologien, sowie die strengen Anforderungen an die Validität der Dokumente. Anhang E enthält eine Übersicht über zum Teil optionale, aber für die Zugänglichkeit von Web-Auftritten sinnvolle XHTML-Attribute und -Elemente.

4.4.3. CSS

Oftmals wurde die Gestaltung von Web-Auftritten durch die Verwendung von HTML-Elementen und -Attributen vorgenommen. Für alternative Browsingtechnologien stellen jedoch Elemente, die für die Präsentation zweckentfremdet wurden, erhebliche Probleme dar oder irritieren den Benutzer. Solche Elemente sind in der Spezifikation von XHTML nicht definiert, außerdem wird im Sinne der Barrierefreiheit eine strikte Trennung von Inhalt und Präsentation verlangt.

Für die inhaltliche Struktur bieten sich Markup-Sprachen wie XHTML oder XML an. Die optische Präsentation sollte mit Hilfe von CSS vorgenommen werden.

Diese Stylesheet-Sprache wurde in Version 1 im Jahr 1996 vom W3C spezifiziert. Die meisten gängigen Browser unterstützen sie inzwischen vollständig. Im Jahr 1998 wurde dann Version 2 vorgestellt, die aber noch immer nur zum Teil unterstützt wird, was vor allem an Lücken und Fehlern in der Spezifikation liegt.

CSS bietet alle Möglichkeiten zur Gestaltung, wie sie auch in HTML durch spezielle Elemente und Attribute definiert waren. Darüber hinaus stellt CSS weitere Funktionen zur Verfügung, die es erlauben, die Präsentation auf alternativen Ausgabegeräten, wie zum Beispiel Screenreadern oder Druckern, zu beeinflussen. Die Unabhängigkeit von der inhaltlichen Struktur hat zusätzlich den Vorteil, dass Stylesheet-Dokumente bei Bedarf ausgetauscht werden können, beispielsweise, wenn ein Style-Switcher zum Einsatz kommt und der Benutzer einen kontrastreichen Stil wählt. Die wichtigsten Merkmale von CSS sind folgende:

- Es lassen sich verschiedene Ausgabemedientypen definieren, beispielsweise für Drucker, Screenreader, Projektoren oder Kleingeräte,
- jedes Element lässt sich frei positionieren und ausrichten,
- Abstände und Bemaßungen können für jedes Element unabhängig von anderen Elementen gewählt werden, wobei sowohl feste als auch relative Werte erlaubt sind, was beispielsweise die Darstellung auf kleinen Bildschirmen ermöglicht,
- für Tabellen und Listen existieren spezielle Formatierungsmöglichkeiten,
- Schriften können für jedes Element frei gewählt werden, beispielsweise über die Angabe der Schriftfamilie (serifenbehaftete oder serifenlose Schriftarten) oder über direkte Wahl des Schriftnamens,
- die Bemaßung der Schriften kann fest, durch Angabe der Pixel, oder relativ erfolgen, wobei eine relative Maßangabe die Skalierbarkeit von Schriften fördert,
- für jedes Element lassen sich unabhängig von anderen Elementen die Vordergrund- und Hintergrundfarbe festlegen,
- für alternative Ausgabegeräte, insbesondere Drucker, lassen sich Seitengrößen, -ränder und -umbrüche definieren,
- für Screenreader lassen sich beispielsweise Sprechgeschwindigkeit, Stimmlage und Lautstärke festlegen.

Die Verwendung von CSS erfordert eine eindeutige Klassifizierung von Elementen und wohlstrukturierte, syntaktisch korrekte Markup-Dokumente. So sollten zum Beispiel Überschriften mit den entsprechenden HTML-Elementen **<h1>** bis **<h6>** deklariert werden. Wenn sich Navigationsstrukturen als Listen präsentieren, so liegt die Verwendung der Elemente **** und **** nahe. Die Formatierung wird durch CSS vorgenommen. Die gleichen Möglichkeiten bieten zwar auch **<div>**-Bereiche, allerdings sind diese eigenschaftslos. Es lässt sich also nicht erkennen, ob es sich beispielsweise um Überschriften oder Listen handelt.

Stylesheets lassen sich auf unterschiedliche Weise in Markup-Dokumente einbinden. Zunächst besteht die Möglichkeit, Stil-Definitionen direkt als Attribute in den HTML-Elementen anzugeben. Als Beispiel ein Absatz mit zentriertem Text:

<p style="text-align: center;">Zentrierter Text.</p>

Nachteilig an dieser Methode ist jedoch, dass bei gleichen Elementen, beispielsweise mehreren Absätzen mit zentrierten Inhalten, die Stil-Definitionen für jedes Element erneut angegeben werden müssen. Dies kann bei komplexen Dokumenten äußerst unübersichtlich werden und erschwert Änderungen und Pflege.

Aus diesem Grund lassen sich Klassen und Pseudo-Klassen in CSS definieren. HTML-Elemente können über Attribute an diese Klassen gebunden werden. Als Beispiel erneut ein Absatz mit zentriertem Text. Zunächst die Stil-Definition:

<style type="text/css"> p.center { text-align: center; } </style>

Das entsprechende Markup gestaltet sich wie folgt:

<p class="center">Zentrierter Text</p>

Auf diese Weise ist es möglich, die Stil-Definitionen in einen Block auszulagern. Sie gelten dann für das gesamte Dokument. Diese Methode hat aber den Nachteil, dass die Stil-Definitionen in jedem Markup-Dokument erneut vorgenommen werden müssen, was bei komplexen Definitionen die Pflege und Wartung erheblich erschwert.

Die letzte Möglichkeit besteht darin, die Definitionen in ein externes Dokument, das so genannte Stylesheet, auszulagern. Dieses kann dann im Markup-Dokument importiert werden. Der Vorteil ist ein globales Stil-Dokument für alle Seiten eines Web-Auftrittes, was Änderungen und Pflege deutlich vereinfachen. Ausserdem bietet sich diese Methode beim Einsatz eines Style-Switchers an, der je nach Wahl das entsprechende Stylesheet laden kann. Da ältere Browser, wie zum Beispiel der Netscape 4.7, nur einen Bruchteil der möglichen CSS-Definitionen unterstützen, besteht die Möglichkeit, dass Auftritte, die von komplexen Stylesheets Gebrauch machen, nicht korrekt verarbeitet werden können. Dieses Problem beginnt bei Darstellungsfehlern und endet bei nicht angezeigten Web-Seiten. Im Normalfall werden externe Stil-Definitionen über folgenden Befehl geladen:

<link rel="stylesheet" type="text/css" href="style.css" />

In diesem Fall laden auch ältere Browser das CSS-Dokument und versuchen es zu verarbeiten, was oft nicht gelingt. Abhilfe schafft hier ein spezieller Befehl:

<style type="text/css">

@import url("style.css");

</style>

Die Verwendung dieses Befehls bewirkt, dass Browser wie der Netscape 4.7 das externe Stylesheet nicht laden und die Web-Seite unformatiert anzeigen. Es sollte jedoch darauf geachtet werden, keine CSS-Medientypen anzugeben. Dies ist zwar möglich und auch vorgesehen, allerdings überspringt der Internet Explorer in diesem Fall den Import-Befehl.

4.5. Fazit

Die Trennung von Inhalt und Präsentation bedeutet zunächst einen höheren Entwicklungsaufwand. Allerdings kommt sie der Vereinfachung von Pflege und Wartbarkeit des gesamten Auftrittes zu Gute.

Abhängig vom zu erwartenden Umfang des Web-Auftrittes muss die Wahl der Markup-Sprache erfolgen. XML bietet sich für Auftritte an, bei denen ein einfacher und übersichtlicher Weg gesucht wird, Seiten mit ähnlichen Strukturen zu erzeugen. Die Möglichkeit, beliebige Daten in Strukturen zu definieren, zeichnet diese Markup-Sprache aus. Die Transformation in HTML-Dokumente erfolgt durch XSL. Problematisch ist dabei jedoch die Kompatibilität mit älteren Browsingtechnologien, da diese XML und XSL nicht verarbeiten können. Deswegen empfiehlt sich der Einsatz von serverseitigen Skript-Sprachen, die das Erzeugen der HTML-Dateien übernehmen und diese an den Browser senden. Doch auch bei der Verwendung serverseitiger Skripts kann es zu Problemen kommen, die den Einsatz von XML/XSL erschweren. Durch die Flexibilität von XML, die es zulässt, beliebige Elemente zu definieren, werden auch die Skripts entsprechend umfangreich. Jedes Element muss im Skript definiert sein, damit es verarbeitet werden kann. Hinzu kommt, dass beispielsweise globale Variablen nach Änderung in den XSL-Stylesheets der Skript-Sprache nicht mehr zur Verfügung stehen, der aktualisierte Wert also nicht mehr in den Skripts vorhanden ist.

Aus diesen Gründen empfehlen sich XML und XSL nur eingeschränkt für die Entwicklung barrierefreier Web-Auftritte. Soll der Auftritt jedoch viele Seiten mit der gleichen Inhaltsstruktur enthalten, die sich nur anhand der eingesetzten Daten unterscheiden, so kann es durchaus sinnvoll sein, die Kombination dieser Sprachen und einer serverseitigen Skript-Sprache einzusetzen. In diesem Fall wird auch die Verwendung einer Datenbank vermieden.

Für die Verwendung von XHTML spricht vor allem die Kompatibilität zum Vorgänger HTML4.01. Dadurch kann gewährleistet werden, dass der Web-Auftritt auch in Verbindung mit älteren Browsingtechnologien zugänglich ist. Ob der Auftritt allein durch statische XHTML-Dokumente oder dynamisch, mit Hilfe serverseitiger Skript-Sprachen, aufgebaut wird, hängt von der Komplexität ab. Bei Einsatz von universellen Inhalten, die beispielsweise über eine Datenbank gefüllt werden, lässt sich eine dynamische Entwicklung nicht umgehen. Ist der zu erwartende Umfang weniger groß und sind keine variablen Inhalte vorgesehen, empfiehlt sich eher ein statischer Auftritt.

Ob nun XML oder XHTML verwendet wird, es muss immer Augenmerk auf die Trennung von Präsentation und Inhalt gelegt werden. In beiden Fällen wird also der Einsatz von validem CSS vorausgesetzt.

5. Evaluierung von Web-Auftritten

In den Richtlinien der WAI wird darauf hingewiesen, dass bei der Entwicklung barrierefreier Web-Auftritte auf valides Markup und den formalen Spezifikationen entsprechende Stylesheets geachtet werden muss. Während das W3C zur Evaluierung dieser Forderungen entsprechende Validatoren anbietet, die Markup und CSS sowohl getrennt als auch in Verbindung miteinander verarbeiten und bewerten können, gibt es seitens der WAI keine Möglichkeiten der automatischen Überprüfung auf Barrierefreiheit.

Im WWW sind allerdings einige unabhängige Validatoren zu finden, die angegebene Web-Auftritte anhand bestimmter Kriterien untersuchen. Trotz allem wird meist nach der Untersuchung der Dokumente ein manueller Durchlauf empfohlen, da beispielsweise Farb- und Kontrastwerte nicht automatisiert zu überprüfen sind. Im Folgenden sollen zwei dieser Evaluierungs-Angebote und ihre Vor- und Nachteile näher beschrieben werden.

5.1. Validator "Cynthia Says"

Das Web-Portal *Cynthia Says* [<http://www.cynthiasays.com>] bietet die Möglichkeit, einzelne Seiten bestehender Web-Auftritte zu analysieren. Dabei kann die Untersuchung nach verschiedenen Richtlinien erfolgen, unter anderem nach den drei Konformitätsstufen der WCAG1.0. Zudem lassen sich verschiedene Browser emulieren, darunter der Microsoft Internet Explorer in den Versionen 4 bis 6, der Netscape-Browser in den Versionen 2 bis 6, aber auch der Textbrowser Lynx und der Browser Opera in der dritten Generation. *Cynthia Says* lässt außerdem optional die Suche nach beweglichen Inhalten und animierten Grafiken zu, was vor allem für die Vermeidung von flimmernden Bildschirminhalten hilfreich ist.

Gestartet wird die Analyse durch die Eingabe der Adresse (URL) des zu untersuchenden Auftritts und die Auswahl der anzuwendenden Richtlinien. Als Beispiel sei die Überprüfung nach den drei Konformitätsstufen der WCAG1.0 dargestellt. Das Ergebnis präsentiert das Portal in drei Tabellen, entsprechend den Stufen der WCAG1.0. Diese enthalten die jeweiligen Richtlinien und deren Checkpunkte, wobei für jeden Checkpunkt der Ausgang der Analyse angegeben wird. Folgende Untersuchungsergebnisse sind dabei möglich:

- *Yes* - Die automatische Evaluierung wurde bestanden,
- *No* - die automatische Überprüfung wurde nicht bestanden,
- *Warning* - die Analyse wurde bestanden, allerdings entsprechen einige Teile nicht in vollem Umfang den Richtlinien, was beispielsweise bei Alternativtexten der Fall ist, wenn diese die minimale Länge nicht erreichen oder die maximale Länge überschreiten,
- *N/V* - es fand keine Überprüfung statt,
- *N/A* - es wurden keine Elemente gefunden, auf die dieser Checkpunkt zutrifft und
- *No Value* - die Untersuchung dieses Checkpunkts muss manuell durchgeführt werden.

Das Portal kann somit nur Checkpunkte analysieren, deren Anwendung sich im Markup widerspiegelt. Um die Erfüllung der anderen Punkte muss sich der Entwickler selbst kümmern. Es wird keine Überprüfung der Validität der Stylesheets oder der Markup-Dokumente durchgeführt, weshalb diese Form der Überprüfung auf Barrierefreiheit nur als Hilfe bei der Entwicklung, jedoch nicht als vollständige Garantie für einen barrierefreien Auftritt angesehen werden sollte.

Auf eine manuelle Überprüfung der einzelnen Inhalte der Richtlinien sollte trotz der Hilfe von *Cynthia Says* nicht verzichtet werden.

Entwurfsprinzipien barrierefreier Web-Auftritte und ihre Umsetzung am Beispiel des Bildungsträgers "FAW gGmbH, Akademie Cottbus"

Checkpoints	Passed		
	Yes	No	Other
Priority 3 - Basic			
4.2 Specify the expansion of each abbreviation or acronym in a document where it first occurs. <ul style="list-style-type: none"> o Rule: 4.2.1 - Identify the use of ABBR elements within the document. <ul style="list-style-type: none"> o Note: Document does not use any ABBR elements. o Rule: 4.2.2 - Identify the use of ACRONYM elements within the document. <ul style="list-style-type: none"> o Note: Document does not use any ACRONYM elements. 			N/A
4.3 Identify the primary natural language of a document. <ul style="list-style-type: none"> o Rule: 4.3.1 - Documents are required to use the META element with the 'name' attribute value 'language' in the Head section. <ul style="list-style-type: none"> o Note: This rule has not been selected to be verified for this checkpoint. o Rule: 4.3.2 - The HTML (Root) element must use the 'lang' attribute. <ul style="list-style-type: none"> o Failure - The HTML (Root) element does not use the 'lang' attribute. 		No	
9.4 Create a logical tab order through links, form controls, and objects. <ul style="list-style-type: none"> o Rule: 9.4.1 - All Anchor, AREA, BUTTON, INPUT, OBJECT, SELECT and TEXTAREA elements are required to use the 'tabindex' attribute. <ul style="list-style-type: none"> o Warning - One or more Anchor, AREA, BUTTON, INPUT, OBJECT, SELECT and TEXTAREA elements do not use the 'tabindex' attribute. 			WARNING
9.5 Provide keyboard shortcuts to important links (including those in client-side image maps), form controls, and groups of form controls. <ul style="list-style-type: none"> o Rule: 9.5.1 - All Anchor, AREA, BUTTON, INPUT, LABEL, LEGEND, and TEXTAREA elements are required to use the 'accesskey' attribute. <ul style="list-style-type: none"> o Warning - One or more Anchor, AREA, BUTTON, INPUT, LABEL, LEGEND, and TEXTAREA elements do not use the 'accesskey' attribute. 			WARNING
10.5 Until user agents (including assistive technologies) render adjacent links distinctly, include non-link, printable characters (surrounded by spaces) between adjacent links. <ul style="list-style-type: none"> o Rule: 10.5.1 - All Anchor elements not surrounding images cannot be directly adjacent. <ul style="list-style-type: none"> o Failure - Anchor Element found at Line: 13, Column: 223 is directly adjacent to the Anchor element that precedes it. o Failure - Anchor Element found at Line: 13, Column: 310 is directly adjacent to the Anchor element that precedes it. o Failure - Anchor Element found at Line: 13, Column: 402 is directly adjacent to the Anchor element that precedes it. o Failure - Anchor Element found at Line: 13, Column: 487 is directly adjacent to the Anchor element that precedes it. o Failure - Anchor Element found at Line: 13, Column: 652 is directly adjacent to the Anchor element that precedes it. 		No	
11.3 Provide information so that users may receive documents according to their preferences (e.g., language, content type, etc.)			
13.5 Provide navigation bars to highlight and give access to the navigation mechanism.			

Abbildung 4. "Cynthia Says:" Auszug aus den Analyseergebnissen von www.google.de

Anhand der Grafik ist zu erkennen, dass der automatische Test von www.google.de nach Konformitätstufe 3 nicht erfolgreich war, da einige Richtlinien nicht erfüllt werden.

5.2. Validator "Bobby"

Auch der Validator *Bobby* [<http://webxact.watchfire.com>] testet einzelne Seiten von Web-Auftritten auf Aspekte der Barrierefreiheit. Eine Besonderheit im Vergleich zu anderen Validatoren ist, dass Kriterien überprüft werden, welche die Qualität, die Zugänglichkeit und den Datenschutz betreffen. Die Qualität einer Seite wird beispielsweise an fehlerhaften Verknüpfungen und fehlenden textuellen Alternativen gemessen.

Die Zugänglichkeit richtet sich, wie bei anderen Validatoren auch, nach den Richtlinien und Checkpunkten der WCAG1.0. Auch hier werden die Ergebnisse entsprechend den Konformitätsstufen in Tabellen ausgegeben. *Bobby* unterteilt diese in automatische und manuelle Checkpunkte. Ein Vorteil für Entwickler ist die Ausgabe der Zeilennummern, an denen manuelle Überprüfungen durchgeführt werden sollen. Grundsätzlich gibt der Validator Warnungen aus, wenn es einer manuellen Analyse bedarf.

Der Datenschutzbericht enthält zum einen Hinweise zur Verschlüsselung der Seite, beispielsweise per HTTPS, zum anderen zu Cookies und Formularabfragen. Auch wird getestet, ob Inhalte von externen Seiten geladen werden oder Verknüpfungen zu externen Auftritten vorhanden sind. Allerdings sollte dieser Bericht eher als Information gewertet werden.

Entwurfsprinzipien barrierefreier Web-Auftritte und ihre Umsetzung am Beispiel des Bildungsträgers "FAW gGmbH, Akademie Cottbus"

Page last checked on Thu 09/08/2005 at 12:05am.

General **Quality** **Accessibility** **Privacy** Expand All | Collapse All

⊗ This page **does not comply** with all of the automatic and manual checkpoints of the W3C Web Content Accessibility Guidelines, and **requires repairs and manual verification.**

	Automatic Checkpoints			Manual Checkpoints		
	Status	Errors	Instances	Status	Warnings	Instances
Priority 1	✓	0	0	⚠	11	17
Priority 2	⊗	4	9	⚠	18	30
Priority 3	⊗	4	10	⚠	10	10

Priority 1 Checkpoints Collapse Section | Top of Page

⚠ **Warnings**
11 tests, 17 instances on page Expand Code Fragments

	Guideline	Instances	Line Numbers
1.1	If an image conveys important information beyond what is in its alternative text, provide an extended description.	1	13
2.1	If you use color to convey information, make sure the information is also represented another way.	2	13, 17
4.1	Identify any changes in the document's language.		
5.1	If this is a data table (not used for layout only), identify headers for the table rows and columns.	1	17
5.2	If a table has two or more rows or columns that serve as headers, use structural markup to identify their hierarchy and relationship.	2	17, 17
6.1	If style sheets are ignored or unsupported, ensure that pages are still readable and usable.		
6.3	Provide alternative content for each SCRIPT that conveys information or functionality.		
6.3	Make sure pages are still usable if programmatic objects do not function.	3	7, 14, 18
7.1	Make sure that the page does not cause the screen to flicker rapidly.		
8.1	Provide accessible alternatives to the information in scripts, applets, or objects.	3	7, 14, 18
14.1	Use the simplest and most straightforward language that is possible.		

Abbildung 5. "Bobby:" Auszug aus den Analyseergebnissen von www.google.de

An dieser Grafik lässt sich ablesen, dass die Überprüfung von www.google.de eine Konformität zur Stufe 1 der WCAG1.0 ergab. Allerdings sind auch hier Warnungen zu finden, weshalb eine manuelle Analyse durchgeführt werden sollte.

Wie auch *Cynthia Says* lässt sich der Validator *Bobby* eher als Hilfe während der Entwicklung ansehen. Automatische Analysen sind im Allgemeinen grundsätzlich nur Hilfsmittel und dürfen nicht als Maßstab dienen. Manuelle Überprüfungen bleiben in allen Fällen unabdingbar.

5.3. Manuelle Evaluierungsmethode

Bei der Entwicklung barrierefreier Auftritte können automatische Validatoren nur bedingt helfen. Eine manuelle Analyse der entwickelten Seiten sollte zusätzlich immer durchgeführt werden. Auch bestehende Auftritte lassen sich manuell im Hinblick auf Barrierefreiheit untersuchen. Dabei bedarf es jedoch auch einiger Hilfsmittel, wie zum Beispiel verschiedener Browser. Auch HTML-Editoren können hilfreich sein.

Im Folgenden soll eine Methode vorgestellt werden, die es anhand von Prüfpunkten erlaubt, bereits fertig entwickelte Web-Auftritte auf Barrierefreiheit zu überprüfen. Die Punkte lehnen sich dabei an die Checkpunkte der WCAG1.0 an, wobei allerdings nur zwei Prioritätsstufen zum Einsatz kommen. Die Umsetzung der ersten Stufe ist für barrierefreie Web-Auftritte unabdingbar. Die Kriterien der zweiten Stufe stellen Empfehlungen dar, deren Einhaltung einer verbesserten Zugänglichkeit zu Gute kommt.

5.3.1. Grundlegende Kriterien

Um einen gewissen Grad an Barrierefreiheit garantieren zu können, müssen bei der Entwicklung von Web-Auftritten einige grundlegende Regeln eingehalten werden. Viele Informationsangebote beinhalten Zertifikate des W3C, welche belegen sollen, dass valides XHTML oder auch valides CSS eingesetzt wurde. Diese Fälle, also die Verwendung von validem Markup und valider Stylesheets, lassen sich meist auf recht einfache Art und Weise überprüfen.

Prüfkriterien (Priorität 1)	Vorgehensweise
1. Trennung von Inhalt und optischer Präsentation	Analyse des Markups mit entsprechendem Editor. Alle optischen Gestaltungstechniken müssen durch CSS umgesetzt sein. (zum Beispiel an Stelle von <code><body bgcolor="#ffffff"></code> => <code><body style="background-color: #fff;></code>)
2. Validität des Markups	Validierung des Dokumentes durch einen Markup-Validator, zum Beispiel den des W3C: http://validator.w3.org
3. Validität der Stylesheets	Validierung der Stylesheets durch den Validator des W3C: http://jigsaw.w3.org/css-validator
4. Verzicht auf Layout-Tabellen und Frames	Analyse des Markups mit entsprechendem Editor. Tabellen dürfen nur für die Repräsentierung von Daten eingesetzt, auf Frames muss vollständig verzichtet werden
5. Lesbarkeit des Auftrittes ohne CSS	Überprüfung mit Hilfe entsprechender Browser, beispielsweise Netscape 4.7 oder Opera im Benutzer-Stil-Modus.

Wie bereits erwähnt, sollten externe CSS-Definitionen im Zweifelsfall über die **@import**-Regel in das Markup geladen werden. Nur so übergehen ältere Browser das Stylesheet und geben das unformatierte Dokument aus.

Die Umsetzung der obigen Punkte ist nicht nur im Sinne der Barrierefreiheit sinnvoll, sondern sollten bei jeder Entwicklung eines Web-Auftrittes eingehalten werden. Den formalen Spezifikationen entsprechende Markup-Dokumente und Stylesheets sind die Grundlagen eines jeden Auftrittes.

5.3.2. Optische Gestaltung

Die optische Gestaltung spielt bei der Entwicklung von Web-Auftritten eine immer größere Rolle. Barrierefreie Auftritte müssen nicht auf ansprechende Gestaltungselemente verzichten.

Vielmehr bietet die Trennung von Inhalt (XHTML oder XML) und Präsentation (CSS) viele Möglichkeiten, um zugängliche, aber trotzdem einladende Web-Seiten zu erstellen.

Allerdings müssen einige Punkte beachtet werden, die auch bei der manuellen Evaluierung eine Rolle spielen. Die folgende Übersicht soll die wichtigsten Prüfkriterien aufzeigen, anhand derer entschieden werden kann, ob die grafische Gestaltung die Zugänglichkeit des Auftrittes unterstützt.

Prüfkriterien (Priorität 1)	Vorgehensweise
1. Kontrastreiche Präsentation	Analyse der Vordergrund- und Hintergrundfarben. Sie müssen sich deutlich voneinander abheben. Dies betrifft Schriftfarben, aber auch Grafiken und Bilder.
2. Verzicht auf bestimmte Farbkombinationen	Überprüfung von Farbkombinationen. Rot und Grün beziehungsweise Blau und Grün dürfen nicht in direkter Nähe zueinander auftreten.
3. Hervorhebung von Elementen nicht nur durch Farbwahl	Analyse aller Navigationselemente und Verknüpfungen. Aktive und besuchte Elemente dürfen nicht allein durch Wahl verschiedener Farben gekennzeichnet sein.
4. Veränderbare Vorder- und Hintergrundfarben	Es muss entweder direkt auf dem Web-Auftritt oder mit Hilfe von Browserfunktionen möglich sein, individuelle Farbkombinationen zu wählen.
5. Skalierbare Schriftgrößen	Überprüfung, ob sich Schriftgrößen mit Hilfe der entsprechenden Browserfunktionen oder direkt auf dem Web-Auftritt verändern lassen. Das Layout des Auftrittes sollte bei Änderung der Schriftgrößen erhalten bleiben.
6. Verzicht auf blinkende Inhalte oder Laufschriften	Es dürfen keine blinkenden Inhalte oder Laufschriften vorhanden sein.
7. (optional) Verzicht von weißer Schrift bei invertiertem Kontrast	Wird die Möglichkeit geboten, eine Präsentation mit invertiertem Kontrast zu wählen (schwarzer Hintergrund), sollte keine weiße Schriftfarbe genutzt werden. Vielmehr ist eine Wahl von gedämpften Weiß (Abstufung zu Gelb) sinnvoll.

Grafische Gestaltungsmöglichkeiten stehen im Sinne der Barrierefreiheit eher im Hintergrund. Da es aber für viele Einrichtungen wichtig ist, eine ansprechende Präsentation

ihres Informationsangebotes zu bieten, werden die grafischen Möglichkeiten oftmals ausgeschöpft. In diesem Fall ist die Umsetzung dieser Prüfkriterien außerordentlich wichtig für sehbehinderte Benutzer, wobei auch Menschen ohne Beeinträchtigung davon profitieren. Ohne die Einhaltung dieser Punkte ist eine Zugänglichkeit des entsprechenden Auftrittes nicht oder nur teilweise gegeben. Der Web-Auftritt ist in diesem Fall nicht barrierefrei. Auch in den Richtlinien der WAI entsprechen diese Prüfkriterien der höchsten Prioritätsstufe.

5.3.3. Alternativen für Audio- und visuelle Inhalte

Während Grafiken, Bilder, Animationen und Filme für blinde und stark sehbehinderte Benutzer nicht zugänglich sind, stehen hörgeschädigte Menschen bei Filmen und Audio-Inhalten vor einer Barriere. Entgegen der Auffassung, dass nur Audio- und visuelle Inhalte mit alternativen Texten versehen werden sollten, wenn sie für das Verständnis des Gesamtinhaltes notwendig sind, sollten solche Elemente vielmehr immer mit Alternativtexten ausgestattet sein. Einzig in der Ausführung der Beschreibung lässt sich zwischen wichtigen und eher unwichtigen Elementen unterscheiden.

Prüfkriterien (Priorität 1)	Vorgehensweise
1. Textuelle Alternativen für Bildelemente	Evaluierung mit Textbrowsern und mit Browsern ohne automatisches Laden von Grafiken. Alle grafischen Elemente müssen mit Hilfe des alt - oder longdesc -Attributes eine textuelle Alternative zur Verfügung stellen. Möglich ist auch ein Verweis zu einer entsprechenden Unterseite. Für das Gesamtverständnis wichtige Grafiken sollten einen aussagekräftigen Text aufweisen.
2. Textalternativen für Audio-Inhalte	Audio-Dokumente müssen auch in reiner Textform vorhanden sein.
3. Alternativtexte für Filme und Animationen	Für Filme und Animationen müssen entsprechende, synchron ablaufende Textversionen angeboten werden.
4. Beschriftung von Image-Maps	Überprüfung mit Hilfe von Textbrowsern. In Image-Maps enthaltene Bild-Verknüpfungen müssen mit dem alt -Attribut beschriftet sein. Zusätzlich muss das Element <map> benutzt werden, um die gesamte Karte zu bezeichnen.

Wie auch in den WCAG1.0-Richtlinien, sind diese Prüfkriterien in der höchsten Prioritätsstufe zu finden. Es ist außerordentlich wichtig, diese Punkte einzuhalten, da andernfalls die betroffenen Inhalte für alternative Ausgabegeräte nicht zur Verfügung stehen.

5.3.4. Navigation und funktionale Elemente

Einer der wichtigsten Bereiche eines Web-Auftrittes ist die Navigation. Sie kann die Interaktion mit dem Benutzer erheblich erleichtern oder aber, falls sie ungünstig umgesetzt ist, erheblich erschweren. Der Entwicklung dieser Bereiche sollte demnach besondere Aufmerksamkeit geschenkt werden.

Auch funktionale Elemente, wie zum Beispiel Skripts, können zur Verwirklichung von Navigationsstrukturen eingesetzt werden. Hier müssen jedoch einige Punkte beachtet werden, die in den WCAG1.0 unter die Richtlinien "Unabhängigkeit von Skripts und Applets" sowie "Unabhängigkeit von Ein-/Ausgabegeräten" fallen.

Prüfkriterien (Priorität 1 und 2)	Vorgehensweise
1. Aussagekräftige Bezeichnung für Navigationselemente (Priorität 1)	Sind die eingesetzten Navigationselemente ausreichend und aussagekräftig beschriftet? Vermieden werden müssen Verknüpfungen, wie zum Beispiel "Klicken Sie hier".
2. Konsistenz der Hauptnavigation (Priorität 1)	Die Hauptnavigation muss auf allen Seiten, einschließlich Unterseiten, konsistent sein.
3. Markierung aktiver Seitenbereiche und Breadcrumb-Navigation (Priorität 1)	Die aktive Seite muss in der Navigationsstruktur markiert sein. Da dies nur durch CSS möglich ist, muss eine zusätzliche Navigation ("Sie befinden sich hier") verfügbar sein, welche es erlaubt, die aktuelle Position im Auftritt zu bestimmen (Breadcrumb-Navigation).
4. Verzicht auf Skripts und Applets in wichtigen Bereichen (Priorität 1)	Auf Skripts und Applets muss verzichtet werden, wenn diese unbedingt für die Funktion des Auftritts notwendig sind.
5. Verwendung von Skripts, Applets und zeitsensitiver Inhalte in sonstigen Bereichen (Priorität 1)	Analyse mit Textbrowser. Werden Skripts und Applets eingesetzt, so müssen enthaltene Informationen auch ohne diese verfügbar sein. Zeitsensitive Inhalte müssen sich kontrollieren, gegebenenfalls stoppen und neustarten lassen.
6. Sinnvolle und logische Nummerierung von Navigationselementen für die Benutzung von Tastaturen (Priorität 1)	Analyse mit Hilfe gängiger Browser und der Tabulator-Taste. Werden Verknüpfungen in einer logischen Reihenfolge angesprungen, wenn als Eingabegerät die Tastatur verwendet wird? Sind alle Verknüpfungen, auch die im Inhaltsbereich, auf diese Art erreichbar?
7. Verzicht auf dynamische Navigationselemente (Priorität 2)	Evaluierung mit Hilfe gängiger Browser. Auf dynamische Navigationselemente, wie zum Beispiel ausklappbare Menüstrukturen, sollte

Prüfkriterien (Priorität 1 und 2)	Vorgehensweise
	nach Möglichkeit verzichtet werden.
8. Erreichbarkeit der Hauptnavigation mit Hilfe von Kurztasten (Priorität 2)	Überprüfung mit Hilfe gängiger Browser. Hilfreich ist eine Erreichbarkeit der Elemente der Hauptnavigation über Kurztasten, beispielsweise STRG+ALT+1 für die Startseite. Tasten, die bereits für Browserfunktionen vergeben sind, dürfen nicht verwendet werden.
9. Anzahl der Hauptnavigationseinträge und der Seitenebenen (Priorität 2)	Es sollten nicht mehr als 9 Navigationseinträge in der Hauptnavigation zur Verfügung stehen. Die Anzahl der Seitenebenen sollte sich auf maximal 3 beschränken.

Die Punkte der höchsten Priorität müssen umgesetzt werden, da es ansonsten für viele Benutzergruppen nicht möglich ist, mit dem Auftritt zu interagieren. Dies betrifft vor allem die Möglichkeit, Navigationselemente mit Hilfe der Tastatur zu erreichen sowie den Einsatz von Breadcrumb-Navigationen. Die Kriterien der mittleren Prioritätsstufe stellen Empfehlungen dar, deren Einhaltung angestrebt werden sollte.

5.3.5. Seitenstruktur und Markup

Der strukturelle Aufbau von Web-Seiten kommt vor allem dann zum Tragen, wenn Browsingtechnologien eingesetzt werden, die nicht CSS-fähig sind. Trotz des Verzichts auf Layout-Tabellen und Frames, können bei falscher Verwendung von CSS-Technologien Fehler entstehen. Des Weiteren sollten bestimmte Strukturelemente durch die entsprechenden Möglichkeiten der Markup-Sprache repräsentiert werden.

Prüfkriterien (Priorität 2)	Vorgehensweise
1. Seitenstruktur: <ol style="list-style-type: none"> 1. Hauptnavigation gefolgt vom 2. Inhaltsbereich, danach die 3. (optionale) Unterseiten-Navigation 	Überprüfung der Seitenstruktur in einem Textbrowser (beispielsweise Lynx) oder mit ausgeschaltetem CSS (zum Beispiel im Benutzer-Stil-Modus von Opera).
2. Sprungmarken für Seitenbereiche	Bei unsichtbaren Sprungmarken: Analyse mit Hilfe von Textbrowsern oder mit ausgeschaltetem CSS. Lassen sich die einzelnen Bereiche überspringen? Beispiel:

Prüfkriterien (Priorität 2)	Vorgehensweise
	über der Hauptnavigation eine Verknüpfung "Springe zum Inhalt".
3. Beschriftung der Seitenbereiche	Bei unsichtbaren Beschriftungen: Analyse mit Hilfe von Textbrowsern oder mit ausgeschaltetem CSS. Ist jeder einzelne Bereich mit einer Überschrift versehen? Beispiel: "Dies ist die Hauptnavigation"
4. Übersicht über die Seitenstruktur und die Barrierefreiheit unterstützende Funktionen	Web-Auftritte sollten eine, der Seitenstruktur entsprechende, Übersicht (Sitemap) aufweisen. Sinnvoll ist zudem ein Überblick über Funktionen, beispielsweise Kurztastenbelegungen, die die Zugänglichkeit fördern.
5. Markierung von Verknüpfungen	Verknüpfungen sollten markiert sein. Zum Beispiel: "Link: [Startseite]". Die Markierungen können durch CSS unsichtbar sein. Überprüfung mit Hilfe von Textbrowsern oder ausgeschaltetem CSS.
6. Verwendung entsprechender Markup-Elemente für Verknüpfungen, Überschriften, usw.	Evaluierung mit Hilfe eines Editors. Für Überschriften sollten die Elemente <h1> bis <h6> eingesetzt werden. Navigationsstrukturen sind als Listen mit zu kennzeichnen.
7. Beschriftung und Kennzeichnung von Datentabellen	Überprüfung mit Hilfe eines Editors. Die Inhalte von Datentabellen sollten Bezug auf die entsprechenden Achsen bzw. Spalten-/Zeilenüberschriften haben. Für die Beschriftung von Tabellen und die Kenntlichmachung von Bezügen zu Spalten/Zeilen sollten die entsprechenden Attribute axis , scope , headers oder summary genutzt werden.
8. Kennzeichnung von Formularen	Evaluierung mit Hilfe eines Editors. Formulare (Listen, Eingabefelder, usw.) sollten mit Hilfe der Attribute label und for gekennzeichnet werden.

Oben genannte Kriterien entsprechen der Prioritätsstufe 2, sollten also eingehalten werden. Sie sind hilfreich, um die Übersichtlichkeit der Seite zu wahren und kommen vor allem in Textbrowsern und Screenreadern zur Geltung. In grafischen Browsern, welche CSS

beherrschen, spielen diese Punkte eher eine untergeordnete Rolle.

5.3.6. Natürliche Sprache

Die letzten Prüfkriterien betreffen die Wahl der natürlichen Sprache. Dazu gehören sowohl Satzbau und Formulierungen, als auch die Auszeichnung der verwendeten Sprache und die Kennzeichnung von Abkürzungen und Akronymen.

Prüfkriterien (Priorität 2)	Vorgehensweise
1. Verwendung kurzer, prägnanter Sätze und Formulierungen	Sätze und Formulierungen sollten so kurz und aussagekräftig wie möglich sein. Eine Satzlänge von mehr als 9 Worten ist nach Möglichkeit zu vermeiden. Die Art des Satzbaus und der Formulierungen soll über den gesamten Web-Auftritt konsistent eingehalten werden.
2. Kennzeichnung verwendeter Sprache	Überprüfung mit Hilfe eines Editors. Die verwendete Sprache ist am Beginn des Dokumentes über das xml:lang -Attribut anzugeben.
3. Kennzeichnung von Sprachwechseln	Überprüfung mit Hilfe eines Editors. Sprachwechsel im Inhalt sind über das xml:lang -Attribut zu kennzeichnen.
4. Definition von Abkürzungen und Akronymen	Überprüfung mit Hilfe eines Editors oder durch gängige Browser. Für Abkürzungen und Akronyme sollten die entsprechenden Elemente <abbr> und <acronym> verwendet werden. Gängige Browser zeigen dann die volle Bedeutung an.

Nicht nur für Menschen mit psychischen Beeinträchtigungen sind die oben aufgeführten Kriterien von Vorteil. Auch anderen Benutzern erleichtern sie die Erfassung des Inhaltes. Leider werden die Attribute zur Auszeichnung der Sprache (**xml:lang**) nicht von allen Browsern unterstützt. Da sie jedoch in den formalen Spezifikationen des W3C zu finden sind und in Zukunft sicher mehr Bedeutung erhalten, sind sie in der Prioritätsstufe 2 aufgeführt.

5.4. Einsatz der Evaluierungsmethode am konkreten Beispiel

Die Analyse von bestehenden Web-Auftritten mit Hilfe der oben vorgestellten Methode soll anhand des Auftritts der Stadt Cottbus (<http://www.cottbus.de>) verdeutlicht werden. Als Markup-Sprache für den Web-Auftritt wird HTML4.01 in der Transitional-Variante

verwendet.

Kriterien aus 5.3.1 (grundlegende Kriterien):

1. Inhalt und Präsentation sind nicht vollständig voneinander getrennt. Mit Hilfe eines Editors ist erkennbar, dass zwar CSS-Stildefinitionen geladen, jedoch bei einigen Elementen spezielle Attribute zur Anpassung der optischen Präsentation eingesetzt werden. So enthält beispielsweise das Element `<body>` das Attribut `bgcolor` zur Festlegung der Hintergrundfarbe.
2. Die Markup-Überprüfung durch den Validator des W3C schlägt fehl. Einige Elemente enthalten Attribute, die nicht in den Spezifikationen von HTML4.01 vorgesehen sind. Außerdem werden stellenweise invalide Attributwerte verwendet. Als Beispiel sei ein Auszug aus den Analyseergebnissen des Validators gegeben:

```
Line 10, column 34: there is no attribute "TOPMARGIN" <body
bgcolor="#cccccc" topmargin="0" leftmargin="0" marginheight="0"
marginwidth
```

3. Eine Überprüfung der Stildefinitionen ist auf Grund des invaliden Markups nicht möglich.
4. Die optische Einteilung der Seiten erfolgt mit Hilfe von Layout-Tabellen, was durch die Linearisierung in alternativen Ausgabegeräten für viele Benutzer Probleme aufwerfen kann.
5. Auch ohne den Einsatz von CSS sind alle Informationen zugänglich, was jedoch an der Verwendung von HTML-Attributen zur optischen Gestaltung liegt.

Ergebnis: Die grundlegenden Kriterien, wie zum Beispiel die Verwendung von validem Markup und die Trennung von Inhalt und Präsentation, sind nicht erfüllt. Hinzu kommt, dass die Präsentation mit Hilfe von Layout-Tabellen erfolgt. Die Barrierefreiheit ist auch mit der erfolgreichen Umsetzung des fünften Punktes nicht gegeben. Der Auftritt ist demnach für bestimmte Benutzergruppen nicht oder nur unter stark erschwerten Bedingungen zugänglich.

Kriterien nach 5.3.1 wurden insgesamt nicht bestanden.

Kriterien nach 5.3.2 (optische Gestaltung):

1. Der Auftritt der Stadt Cottbus präsentiert sich in verschiedenen Farben, im Allgemeinen auch kontrastreich. Allerdings finden sich auf einigen Unterseiten ungünstige Farbkombinationen. So enthält zum Beispiel die Unterseite *Tourismus* einen Bereich mit braunem Hintergrund und schwarzer Schriftfarbe, außerdem werden Navigationsstrukturen mit weißer Schrift auf hellgrauer Hintergrundfarbe dargestellt.
2. Wie bereits im ersten Punkt erläutert, werden verschiedene Farbkombinationen eingesetzt, deren Verwendung die Zugänglichkeit erschwert.

3. Aktive und besuchte Elemente in Navigationsstrukturen lassen sich auch ohne den Einsatz von CSS erkennen, sind also nicht allein durch die Wahl von Farben markiert.
4. Mit Hilfe entsprechender Browserfunktionen lassen sich Vorder- und Hintergrundfarben anpassen. Allerdings enthält der Auftritt sehr viele Grafiken, auch Image-Maps, die sich dadurch nicht verändern lassen.
5. Eine Änderung der Schriftgröße ist mit Skalierungsfunktionen von Browsern möglich.
6. Auf Laufschriften und blinkende Inhalte wurde verzichtet.
7. Das Kriterium trifft nicht auf den Auftritt der Stadt Cottbus zu.

Ergebnis: Durch die schlecht gewählten Farbkombinationen ist der Zugang zum Web-Auftritt nicht problemlos möglich. Verschiedene Präsentationsstile sind nicht vorgesehen, Schriftgrößen und Farben lassen sich jedoch mit Hilfe entsprechender Browserfunktionen verändern.

Kriterien nach 5.3.2 wurden insgesamt bestanden, die Umsetzung ließe sich jedoch verbessern.

Kriterien nach 5.3.3 (Alternativen für Audio- und visuelle Inhalte):

1. Bildelemente enthalten zwar das **alt**-Attribut, allerdings wurde dieses leer gelassen. Für Grafiken und Bilder existieren somit keine textuellen Alternativen. Textbrowser und Screenreader geben in diesem Fall nur den Dateinamen des entsprechenden grafischen Elementes aus.
2. Auf den Seiten sind keine Audio-Inhalte zu finden.
3. Zu den angebotenen Kurzfilmen unter *Trailer* sind keine Alternativen verfügbar.
4. Es werden Image-Maps eingesetzt. Diese sind auch mit Hilfe von Textbrowsern nutzbar, allerdings sollten textuelle Beschreibungen der Karten angeboten werden.

Ergebnis: Das Fehlen von Alternativen für grafische und multimediale Inhalte stellt eine enorme Barriere für viele Benutzergruppen dar. Die Zugänglichkeit des Auftritts ist auch nach diesen Kriterien nicht gegeben.

Kriterien nach 5.3.3 wurden insgesamt nicht bestanden.

Kriterien nach 5.3.4 (Navigation und funktionale Elemente):

1. Die Beschriftungen der Navigationselemente sind aussagekräftig und eindeutig.
2. Elemente der Hauptnavigation sind auf allen Unterseiten konsistent.

3. Die aktuelle Position im Web-Auftritt lässt sich durch die vorhandene Breadcrumb-Navigation immer erkennen.
4. Problematisch stellt sich der Einsatz von Java-Skripts dar, die zur Realisierung der Navigationsstrukturen verwendet werden. Bei eingeschaltetem JavaScript lassen sich auch die zur Verknüpfung gehörenden und farblich hervorgehobenen Felder auswählen. Ohne JavaScript müssen die Verknüpfungen in der Hauptnavigation immer genau angesteuert werden, für Menschen mit motorischen Einschränkungen ein nicht zu verachtendes Hindernis. Zudem sind Effekte, die sich leicht mit CSS umsetzen lassen, durch Skripts realisiert worden. Das betrifft zum Beispiel Farbänderungen bei Mausberührung u.Ä..
5. Auf zeitsensitive Inhalte wurde, bis auf die oben erwähnten Kurzfilme, verzichtet.
6. Eine Nummerierung der Verknüpfungen ist nicht vorhanden. Für die Verwendung der Tastatur muss auf die entsprechende Funktionalität des verwendeten Browsers vertraut werden.
7. Navigationsstrukturen wurden mit Hilfe von JavaScript realisiert. Ohne den Einsatz von Skripts sind sie nur erschwert nutzbar.
8. Kein Navigationselement ist über Kurztasten erreichbar.
9. Die Hauptseite enthält eine Vielzahl von Verknüpfungen, die sich bei Verwendung von Screenreadern und Textbrowsern nur schwer überschauen lassen. Die Anzahl der Seitenebenen ist meist größer als drei.

Ergebnis: Erneut sind wichtige, der Prioritätsstufe 1 entsprechende, Kriterien nicht erfüllt worden. Vor allem auf die Verwendung von Skripts, deren Abschalten die Navigation auf den Seiten erschwert, muss im Sinne der Barrierefreiheit verzichtet werden.

Kriterien nach 5.3.4 wurden insgesamt nicht bestanden.

Kriterien nach 5.3.5 (Seitenstruktur und Markup):

1. Die Forderungen nach einer klaren Seitenstruktur sind nicht erfüllt. Mit Hilfe des Textbrowsers Lynx ist erkennbar, dass auf allen Unterseiten zunächst die Unterseiten-Navigation, gefolgt von Haupt- und Breadcrumb-Navigation, erscheint. Dies beeinflusst die Übersichtlichkeit bei linearisierenden Ausgabegeräten erheblich.
2. Sprungmarken sind auf keiner der einzelnen Seiten zu finden.
3. Die einzelnen Seitenbereiche enthalten keine Beschriftung. Für sehbehinderte Menschen ist nicht erkennbar, in welchem Teil der Seite sie sich befinden.
4. Eine Inhaltsangabe existiert, eine Auflistung von Funktionen, die die Barrierefreiheit unterstützen, wird nicht angeboten.

5. Als äußerst problematisch stellt sich hier der Einsatz von Layout-Tabellen dar. Verknüpfungen sind nur durch die einzelnen Tabellen-Zellen voneinander abgegrenzt, so dass Textbrowser und Screenreader diese als fortlaufenden Text ausgeben.
6. Überschriften sind teilweise durch eigenschaftslose **<div>**-Elemente erzeugt worden, die entsprechenden CSS-Formatierungen unterliegen und somit durch alternative Ausgabegeräte nicht als Überschrift, sondern als normaler Text ausgegeben werden.
7. Es sind keine Datentabellen vorhanden.
8. Eine Bezugnahme auf Formulare mit Hilfe von **label** und **for** erfolgt nicht. In alternativen Ausgabegeräten kann ein Formular nicht mit seiner Bezeichnung in Beziehung gesetzt werden.

Ergebnis: Anhand der Ergebnisse der Analyse dieser Kriterien lässt sich erkennen, welche Auswirkungen die Verwendung von Layout-Tabellen und der Verzicht auf sinnvoll gewählte Markup-Elemente haben können. Die Zugänglichkeit des Web-Auftrittes wird erheblich erschwert, möglicherweise einigen Benutzern sogar gänzlich verwehrt.

Kriterien nach 5.3.5 wurden insgesamt nicht bestanden.

Kriterien nach 5.3.6 (Natürliche Sprache):

1. Auf kurze und prägnante Sätze ist im Allgemeinen geachtet worden. Da eine Umsetzung dieser Forderung nicht immer möglich ist, wird die empfohlene Länge von 9 Worten teilweise überschritten.
2. Eine Verwendung des **xml:lang**-Attributes lässt sich nicht erkennen.
3. Sprachwechsel treten auf (zum Beispiel auf der Unterseite *Wirtschaft*), werden aber nicht genauer spezifiziert, was die Aussprache von Audio-Programmen, die durch Screenreader angesteuert werden, beeinflussen kann.
4. Abkürzungen sind nicht erläutert bzw. mit Hilfe der entsprechenden Elemente definiert. Als Beispiel sollen die Akronyme (*ÖPNV*, *ARGE*, usw.) im *Sitemap* dienen, denen keine Erklärung zugeordnet ist.

Ergebnis: Auch diese Checkpunkte konnten nicht zufriedenstellend umgesetzt werden. Allerdings ist davon auszugehen, dass die Einhaltung dieser Kriterien eher einen ergänzenden Charakter hat.

Kriterien nach 5.3.6 wurden insgesamt nicht bestanden.

Die Überprüfung des Web-Auftrittes der Stadt Cottbus schlug fehl. Da die Prioritätsstufe 2 eine Verfeinerung der ersten Stufe darstellt, lässt sich bereits mit dem Nichtbestehen der grundlegenden Punkte (5.3.1 - 5.3.3, teilweise 5.3.4) von einem, für beeinträchtigte Benutzergruppen, äußerst schwer zugänglichen Auftritt sprechen. Selbst grundlegende, nicht

unbedingt die Barrierefreiheit betreffende Kriterien, beispielsweise die Verwendung von validem Markup, werden nicht eingehalten. Sie sind jedoch Voraussetzung für die Beseitigung verschiedener Hindernisse in Web-Auftritten.

5.5. Einsatz der verschiedenen Evaluierungsmethoden

Evaluierungsmethoden sollten immer dann eingesetzt werden, wenn es darum geht, barrierefreie Web-Auftritte auf ihre Zugänglichkeit zu überprüfen. Automatisierte Validatoren sind dabei hilfreiche Werkzeuge, können aber nur Probleme bzw. Barrieren erkennen, die in den Markup-Dokumenten vorliegen. Es lässt sich jedoch davon ausgehen, dass Web-Seiten, die automatisierte Tests nicht bestehen, keinesfalls barrierefrei sind. Der Sinn einer manuellen Methode zur Analyse von Web-Auftritten ist es demnach, eine verfeinerte, auf automatischen Analysen aufbauende Überprüfung zu ermöglichen und dem Entwickler zu helfen, Probleme zu erkennen, die von Software-Werkzeugen nicht gefunden werden können.

Die oben vorgestellte manuelle Evaluierungsmethode geht gezielt von nur zwei Prioritätsstufen aus. Diese basieren auf den Richtlinien der WCAG1.0, fassen aber einige Checkpunkte zusammen oder trennen andere auf. Zudem wurden die Prioritäten für verschiedene Punkte neu vergeben. Werden alle Prüfkriterien bestanden, so ist der entsprechende Web-Auftritt barrierefrei nach der Konformitätsstufe 2 der WCAG1.0. Zu bemerken ist jedoch, dass das Nichtbestehen nur eines Punktes die Zugänglichkeit zum Auftritt stark erschwert, bei Punkten der Prioritätsstufe 1 möglicherweise gänzlich verhindert. Eine Umsetzung aller Prüfkriterien ist im Sinne der Barrierefreiheit empfehlenswert.

6. Auftritt der Akademie Cottbus

Während öffentliche Einrichtungen gesetzlich verpflichtet sind, barrierefreie Web-Auftritte anzubieten, steht es privaten Unternehmen frei, ihre Informationsangebote zugänglich zu gestalten. Dennoch ist es generell zu empfehlen, bei der Planung und Entwicklung von Internet-Auftritten auf einen gewissen Grad der Barrierefreiheit zu achten. Für Unternehmen, die in Rehabilitationsbereichen tätig sind, sollte es hingegen selbstverständlich sein, barrierefreie Auftritte anzubieten.

Die Fortbildungsakademie der Wirtschaft (FAW) gemeinnützige Gesellschaft mbH ist im Bereich der Bildung tätig, wobei das Angebot Jugendmaßnahmen, Maßnahmen zur Wiedereingliederung in den Arbeitsmarkt und auch berufliche Rehabilitation umfasst. Zur Zeit ist die FAW bundesweit mit 29 Akademien und etwa 80 Außenstellen vertreten. Der praxisorientierte Teil dieser Arbeit beinhaltet den Entwurf und die Entwicklung eines barrierefreien Web-Auftrittes für die Akademie Cottbus.

Für die Modellierung des Web-Auftrittes der Akademie Cottbus kommt die Beschreibungssprache WebML zum Einsatz. Diese erlaubt es, unterschiedliche Abstraktionsebenen zu definieren. Dabei wird der Entwicklungsprozess in verschiedene Teile untergliedert, die aufeinander aufbauen. Zunächst erfolgt die Anforderungsdefinition. Sie bestimmt, welche Daten im System verarbeitet werden sollen, welche Funktionalitäten erfüllt werden müssen und wie sich das System dem Benutzer optisch präsentiert. Darauf aufbauend

wird ein Struktur- bzw. Datenmodell entwickelt, das die Beziehungen der verschiedenen Daten verdeutlicht. Ausgehend vom Datenmodell werden im Kompositions- bzw. Navigationsmodell die eigentlichen Teile eines Web-Auftrittes definiert, wobei es verschiedene Sichten auf die Seiten des Auftrittes geben kann. In dieser Stufe sind die vorher definierten Daten als so genannte Units untergebracht und befinden sich in Pages, die konkrete Seiten des Gesamtauftrittes darstellen. Dadurch werden die Beziehungen von Daten und Seiten zum Ausdruck gebracht. Das Ergebnis dieses Schrittes ist eine abstrakte Beschreibung des Gesamtauftrittes, die im nächsten Schritt mit Stildefinitionen verbunden, also mit einer optischen Gestaltung versehen wird.

6.1. Anforderungen

Die Anforderungen, die an den Auftritt der Akademie Cottbus gestellt werden, beziehen sich in erster Linie auf Punkte der Barrierefreiheit. Da das Teilnehmerprofil der Rehabilitationsmaßnahmen nicht fest definiert ist, können Interessenten, die den Auftritt besuchen, potentiell jeder Benutzergruppe angehören. Aus diesem Grund müssen alle Checkpunkte der Richtlinien zur Barrierefreiheit beachtet und nach Möglichkeit erfüllt werden. Als Evaluierung kommen sowohl automatische Werkzeuge als auch die im fünften Kapitel vorgestellte manuelle Evaluierungsmethode zum Einsatz.

Aufgrund veränderlicher Daten zur Vereinfachung von Wartung und Seitenpflege sowie der eventuellen Übernahme des Auftrittes in andere Akademien, sollen alle Seiten dynamisch generiert werden. Einzig Bereiche, die nur selten einer Änderung unterliegen, werden fest in die Markup-Dokumente eingebunden.

6.1.1. Barrierefreiheit

Um die Zugänglichkeit zum Auftritt für alle Benutzergruppen zu garantieren, werden verschiedene, die Barrierefreiheit unterstützende, Funktionalitäten in den Auftritt integriert. Zunächst muss auf eine kontrastreiche Farbwahl geachtet werden, wobei unterschiedliche Kombinationen zur Auswahl stehen, die sich durch Hinzufügen neuer Stylesheets erweitern oder durch Anpassung der bestehenden Definitionen leicht ändern lassen. Zusätzlich lassen sich Schriftgrößen stufenlos skalieren.

Auf den Einsatz von Frames und Tabellen, die ausschließlich für die optische Gestaltung zuständig sind, wird gänzlich verzichtet. Sämtliche die Präsentation betreffende Funktionen werden mit Hilfe von CSS implementiert. Alle Strukturen, für die entsprechende Markup-Elemente zur Verfügung stehen, werden mit diesen umgesetzt. Überschriften beispielsweise sind durch `<h1>` bis `<h6>`, Navigationsstrukturen als Listen `` dargestellt.

Alle Seitenbereiche müssen sich überspringen lassen, wobei diese Sprungmarken nur in Browsern oder für Ausgabegeräte sichtbar sein sollen, die CSS nicht unterstützen. Auf die gleiche Weise werden Seitenbereiche beschriftet und Verknüpfungen markiert. Die Reihenfolge der Bereiche soll den oben aufgeführten Kriterien entsprechen, also zunächst die Hauptnavigation, gefolgt vom Inhalt und danach die optionale Unterseitennavigation. Der Auftritt der Akademie Cottbus wird vor der Hauptnavigation eine Breadcrumb-Navigation enthalten.

Alle Verknüpfungen sind über Tab-Indizes erreichbar, der gesamte Auftritt lässt sich also auch ohne den Einsatz der Maus bedienen. Zusätzlich sind für Elemente der Hauptnavigation Kurztasten vergeben, deren Belegung auf einer separaten Hilfe-Seite aufgeführt werden soll.

Grafiken und Bilder werden mit entsprechenden Alternativtexten versehen. Da im Web-Auftritt keine für das Verständnis des Inhaltes wichtige grafische Elemente vorgesehen sind, erfolgt mit Hilfe des **alt**-Attributes eine kurze und prägnante Ausführung der alternativen Beschreibungen. Abkürzungen und Akronyme sind durch die entsprechenden Möglichkeiten des Markups definiert.

6.1.2. Optische Gestaltung

Für die Gewinnung neuer Teilnehmer ist es außerordentlich wichtig, eine ansprechende optische Gestaltung zu bieten. Diese soll sich, was Farbwahl und Einsatz von Logos oder Banner betrifft, am zentralen Auftritt der FAW gGmbH orientieren.

Umgesetzt wird ein dreispaltiges Layout mit Kopf- und Fußzeile. Dabei sollen sich in der Kopfzeile Logo und Banner der FAW, in der Fußzeile eine Sprungmarke zum Seitenanfang befinden. Die linke Spalte enthält Hauptnavigation und Style-Switcher. Die rechte Spalte ist optional und beinhaltet die Unterseitennavigation, falls es weiterführende Seiten zum aktuellen Inhalt gibt. Zwischen beiden Spalten befindet sich der Inhaltsbereich, welcher in etwa 65% der Größe des Browserfensters verwenden soll. Die Seitenstruktur kann allerdings von Browsingstechnologien, die kein CSS unterstützen, nicht gehalten werden. Die Inhalte werden hingegen linearisiert, also nicht in Spalten, sondern zeilenweise ausgegeben. Aus diesem Grund sind die unter 6.1.1 angesprochenen Sprungmarken und die Reihenfolge der Seitenbereiche wichtig.

Schriftgrößen sind durch einen Style-Switcher anpassbar. Da der Web-Auftritt eher mit Hilfe elektronischer Ausgabegeräte als in gedruckter Form betrachtet werden wird, kommt durchgehend eine serifenlose Schriftart zum Einsatz, was die Lesbarkeit erhöhen soll. Die Skalierbarkeit der Schriftgrößen erfordert relative Bemaßungen.

Auch Farbschemata lassen sich mit Hilfe eines Style-Switchers auswählen. Vorgesehen sind folgende Kombinationen:

- Standard-Stil mit schwarzer Schriftfarbe auf weißem oder hellgrauem Hintergrund. Für Präsentationen der FAW gGmbH wird ein spezieller Wert der Farbe Grün verwendet. Abstufungen dieses Grüntons markieren unter anderem aktive Navigationselemente.
- Stil mit hohem Kontrast, wobei schwarze Schrift auf weißem Hintergrund verwendet wird. Dieser wird bei aktiven Navigationselementen invertiert.
- Stil mit invertiertem Kontrast. Hier kommen eine Abstufung von Weiß als Schriftfarbe und schwarzer Hintergrund zum Einsatz. Dies wird ebenfalls bei aktiven Navigationselementen invertiert.
- Darstellung ohne Stylesheet. Diese Möglichkeit schaltet die Verwendung von CSS vollständig ab. Die Seiten werden linear ausgegeben und alle, auch versteckte, Elemente

werden sichtbar.

Aktive Navigationselemente werden jedoch nicht allein durch Farben markiert, sondern enthalten spezielle Zeichen, so genannte Bullets. Da es keine einheitlichen Aussagen darüber gibt, welches Zeichen als Markierung aktiver Bereiche genutzt werden sollte, wird am Anfang des Auftrittes ein entsprechender Hinweis eingebunden, der beispielsweise von Screenreadern ausgegeben werden kann.

6.1.3. Inhalt

Der Besucher soll durch den Web-Auftritt der Akademie Cottbus einen Überblick über den Standort selbst, über die Außenstellen und aktuell angebotene Maßnahmen erhalten. Als zusätzliche Informationen stehen Anfahrtsbeschreibungen sowie eine kurze Beschreibung der Region und ihrer Geschichte zur Verfügung. Dabei soll sich die Anzahl der Seitenebenen auf drei beschränken, um die Navigation zu erleichtern. Außerdem enthält die Hauptnavigation höchstens neun Einträge, im konkreten Fall nur acht. Damit werden die Forderungen des letzten Prüfkriteriums aus Abschnitt 5.3.4 erfüllt. Durch die dynamische Erzeugung der Navigationsstrukturen ist es jedoch möglich, beliebig viele Einträge einzubinden.

Das folgende Diagramm zeigt die Inhaltsstruktur des Auftrittes der Akademie Cottbus.

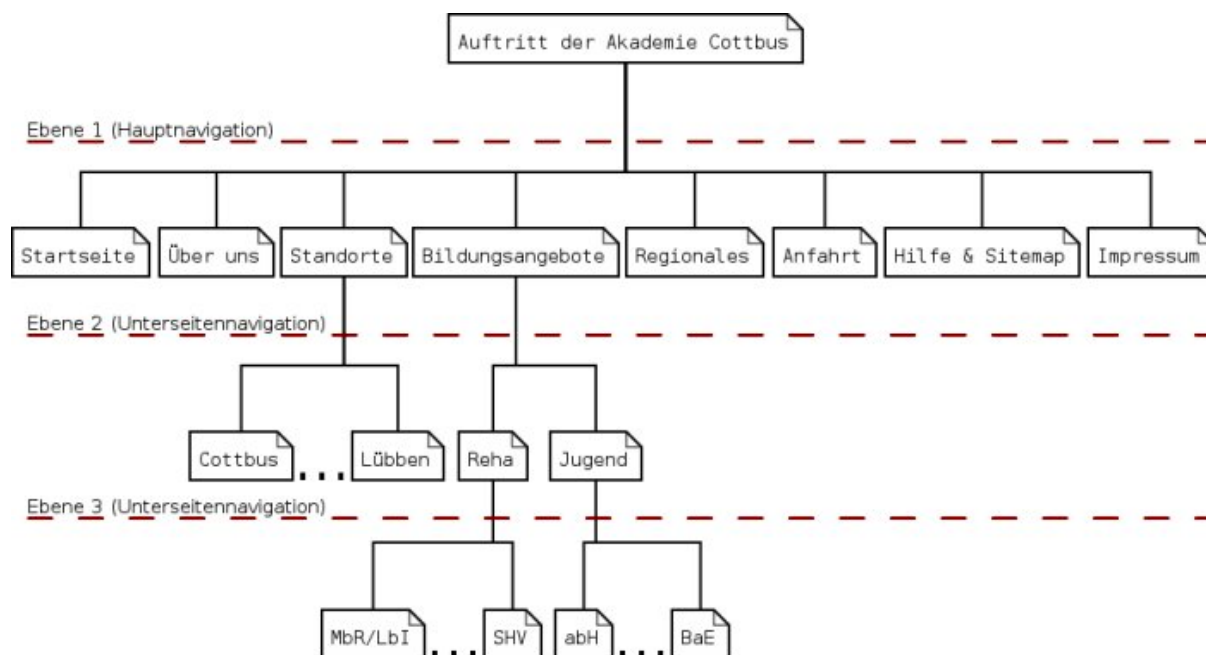


Abbildung 6. Inhaltsstruktur des Web-Auftrittes

Da sich einige Informationen von Zeit zu Zeit ändern können, sollen diese nicht fest in das Markup eingebunden werden. Anstelle statischer XHTML-Seiten werden die Markup-Dokumente zur Laufzeit mit Daten gefüllt. Dadurch soll es möglich sein, Änderungen schnell

und ohne Modifikation des Markups vorzunehmen, was den Einsatz serverseitiger Skript-Sprachen erfordert. Auf die Verwendung von Datenbanken soll dennoch zunächst bewußt verzichtet werden, da die zu verwaltenden Daten diesen Aufwand nicht rechtfertigen. Trotzdem ist die Möglichkeit vorzusehen, für veränderliche Daten entsprechende Datenbanken einzusetzen. Die dynamisch zu verwaltenden Informationen umfassen die gesamten Navigationsstrukturen, die Daten der verschiedenen Standorte bzw. Außenstellen und Details zu Maßnahmen in den Bildungsangeboten. Diese werden zunächst in so genannten Flat-File-Datenbanken, also in einfachen Dateien, abgelegt. Nach Bedarf können die Daten in XML-Dateien oder relationale Datenbanken übernommen werden, was jedoch einen Eingriff in die Implementierung erforderlich macht.

7. Modellierung und Implementierung

Das für den Auftritt genutzte Modellierungsmodell lässt sich auch auf die Implementierung anwenden. WebML nutzt ein Schichtenmodell, das Daten, Funktionen und optische Präsentation während des Entwurfs voneinander trennt. Die Umsetzung des Auftritts erfolgt ebenfalls mit Hilfe einer 3-Schichten-Architektur, dem so genannten 3Tier-Layout.

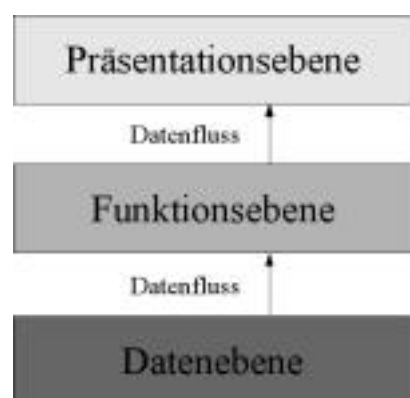


Abbildung 7. 3-Schichten-Modell (3Tier-Layout)

Wie das WebML-Modell für den Entwurf, erlaubt der Einsatz dieser Architektur eine strikte Trennung von Daten, Funktionen und Darstellung in der Implementierung. Ein Zugriff von einer oberen auf eine untere Schicht ist nicht erlaubt. So dürfen Daten nur von der Datenebene an die Funktionsebene gereicht werden, aber nicht umgekehrt. Ebenso ist es nicht vorgesehen, aus der Präsentationsebene auf Funktionen der darunterliegenden Schicht zuzugreifen.

Für den Web-Auftritt der Akademie Cottbus kam die Skript-Sprache PHP in Verbindung mit XHTML-Templates zum Einsatz. Alle Skripts sind objekt-orientiert entworfen, so dass ein Zugriff auf Seitenteile über die entsprechenden Objekte von Klassen möglich ist. Diese Vorgehensweise erlaubt das Laden von Bereichen, die auf dem gesamten Auftritt zur Verfügung stehen. Bei Anforderung einer Seite wird zunächst der Rahmen erzeugt, welcher Kopf- und Fußzeile, Breadcrumb-, Hauptnavigation und Style-Switcher sowie gegebenenfalls die Unterseiten-Navigation enthält. Eine weitere Klasse ist für das Laden des entsprechenden Inhaltes verantwortlich. Diese Architektur ist als Rahmenarchitektur bzw. als Entwurfsmuster

Framework bekannt.

Vor allem die Pflege und die Wartung der Seiten werden durch die Verwendung dieser Architektur erleichtert, da keine Änderungen an den Klassen nötig sind, die den Rahmen erzeugen. Bei Modifikation von Inhalten werden entweder direkt die entsprechenden XHTML-Templates, also die Markup-Dokumente, oder aber die Daten im Database-Layer geändert.

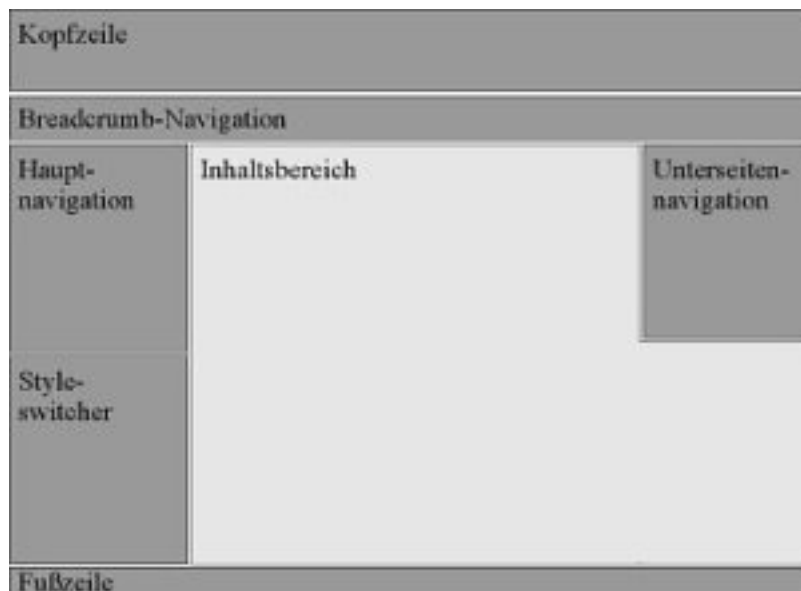


Abbildung 8. Rahmenarchitektur (Entwurfsmuster Framework) des Auftritts

Die dunkleren Bereiche repräsentieren den Rahmen, der über den gesamten Auftritt hinweg konsistent bleibt. Einzig die Unterseiten-Navigation ist nicht immer verfügbar. Im hellen Bereich ist der Inhalt zu finden, der über die entsprechende Klasse je nach Anforderung erzeugt wird.

7.1. Datenebene

Wie bereits angedeutet, werden veränderliche Daten in einer Flat-File-Datenbank abgelegt. Da dies nur Navigationseinträge, Standortinformationen und Details zu Bildungsangeboten umfasst, ist der Einsatz einer relationalen Datenbank nicht erforderlich. Das folgende Schema soll die Beziehung der einzelnen Daten untereinander verdeutlichen. Es wurde mit Hilfe der Modellierungssprache WebML erstellt.

Wie am folgenden Beispiel zu erkennen ist, können Einträge der Navigationsstrukturen verschieden aufgebaut sein. Für Navigationseinträge ist nur die Vergabe von Namen per **name** zwingend erforderlich. Optional kann jeder Eintrag durch **tpl** einen Verweis auf ein XHTML-Template enthalten, in dem statisches Markup definiert ist oder welches Platzhalter enthält, die im Business-Logic-Layer durch Daten ersetzt werden. Zu diesem Zweck lässt sich per **data** jedem Navigationseintrag die entsprechende Datendatei zuordnen. Eine weitere

Möglichkeit betrifft die Ausführung von Skripten. So kann jedem Eintrag an Stelle eines Templates auch ein Skript zugeordnet sein. Der zugehörige Typ ist dann **script**. Zum Erzeugen dynamischer Navigationsstrukturen, die auch Unterseiten enthalten dürfen, lässt sich **sub** verwenden. Zu bemerken ist dabei, dass die Einträge der Unterseiten den gleichen hier aufgeführten Regeln folgen müssen.

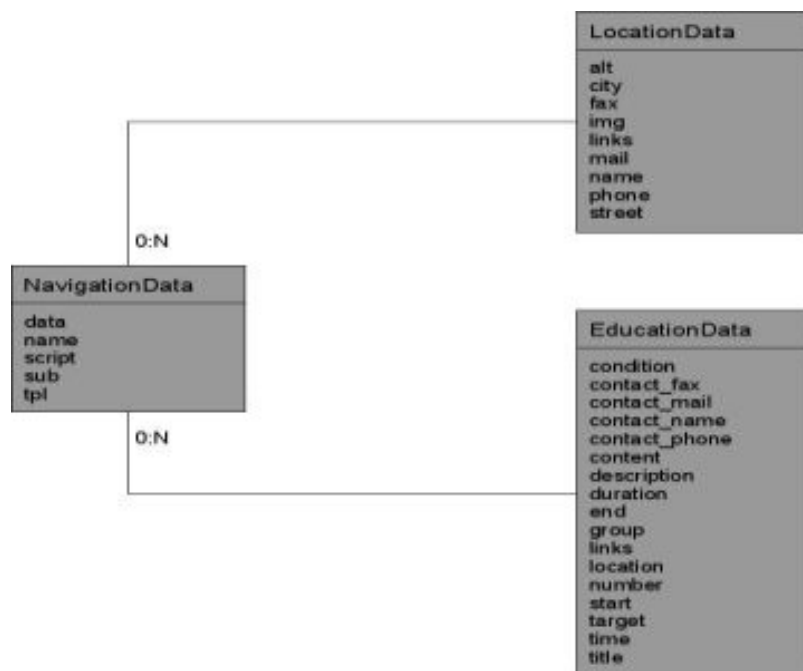


Abbildung 9. Datenstruktur

Das Beispiel des Quelltextes verdeutlicht den Aufbau der Navigationseinträge. Über die Identifikationsnummern, die jedem Array zugeordnet sind, werden die entsprechenden Templates geladen, gegebenenfalls mit Daten gefüllt oder auch Skripts ausgeführt. Die erste Ebene in der Struktur enthält alle Einträge der Hauptnavigation, jede weitere Ebene wird auf dem Auftritt als Unterseiten-Navigation dargestellt.

```
$navConfig = array (
    1 => array ("name" => "Startseite",
              "tpl" => "templates/home.tpl",
              "data" => "data/cb.dat"),
    2 => array ("name" => "Über uns",
              "tpl" => "templates/about.tpl"),
    3 => array ("name" => "Standorte",
              "tpl" => "templates/location.tpl",
              "sub" => array (
                31 => array ("name" => "Cottbus",
                          "tpl" => "templates/location_sub.tpl",
                          "data" => "data/cb.dat"),
                32 => array ("name" => "Senftenberg",
                          "tpl" => "templates/location_sub.tpl",
                          "data" => "data/sfb.dat"))));
```

Daten für Standorte und Informationen zu Bildungsangeboten werden in jeweils separaten Dateien abgelegt, in denen wiederum je ein PHP-Array enthalten ist. In diesem befinden sich

dann beispielsweise Adressen oder Kontaktinformationen. Um die Tab-Index-Folge zu erhalten, muss darauf geachtet werden, bei eventuell vorhandenen Verknüpfungen die Anzahl dieser über eine Variable namens **\$links** festzulegen.

7.2. Funktionsebene

Die zweite Ebene enthält Funktionen, welche die Aufgaben haben, das Framework zu erstellen, Seiteninhalte zu laden und XHTML-Templates mit Daten zu füllen, sowie Style-Switcher-Funktionalitäten anzubieten. Die Beziehungen von Daten und einzelnen Seiten sind anhand des WebML-Kompositionsmodells im Anhang F (Abb. 11) definiert.

Zum Framework gehört die Generierung der Kopf-/Fußzeilen und Navigationsstrukturen. Alle einzelnen Seitenbereiche sind durch eigenständige Klassen definiert, so dass beispielsweise zum Aufbau der Hauptnavigation nur die Erzeugung eines entsprechenden Navigationsobjektes notwendig ist. Jede Klasse bedient nur ihr zugeordnetes XHTML-Template. Daher ist eine Änderung der Reihenfolge der Erzeugung von Objekten nicht ohne Weiteres möglich und würde die Validität des Markups gefährden oder zerstören. Als Template-Klasse kommt *FastTemplate* zur Anwendung, die unter der GPL steht und somit frei genutzt und auch verändert werden darf. Ein Klassendiagramm befindet sich im Anhang F (Abb. 12).

Seiteninhalte werden anhand der IDs aus den Daten für die Navigationsstrukturen geladen. Die Übergabe der entsprechenden Seiten-ID erfolgt mit Hilfe des **GET**-Parameters, wobei vor dem Importieren der angeforderten Seite überprüft wird, ob diese einen Eintrag in der Navigationsstruktur besitzt. Sollte dies nicht der Fall sein, erfolgt das Laden der Startseite, was Missbrauch der **GET**-Funktionalität verhindern soll.

Für die Barrierefreiheit ist es erforderlich, den Auftritt per Tastatur bzw. alternativer Eingabegeräte bedienen zu können. Da eine Indizierung aller Verknüpfungen in der Präsentationsebene durch dynamische Daten verhindert wird, muss im Business-Logic-Layer ein globaler Zähler für Tabulator-Indizes geführt werden. Realisiert wurde dieser mit Hilfe der **SESSION**-Funktionen in PHP. Jeder Link bekommt somit eine eindeutige Nummer, die entsprechend immer weiter erhöht wird. Ein dabei auftretendes Problem ist die Verwendung statischer Seiten, da hier kein Zähler geführt werden kann. Als Konsequenz ergibt sich, dass jede Seite, die eine Verknüpfung enthält, zunächst in der Funktionsebene bearbeitet bzw. ein Platzhalter im entsprechenden Template ersetzt werden muss.

Entsprechend den Richtlinien wurden für alle Elemente der Hauptnavigation Kurztasten-Zugriffe ermöglicht. Die verwendeten Tasten entsprechen den einstelligen IDs in der ersten Ebene der Navigationsstruktur. An dieser Stelle ergibt sich ein weiteres Problem. Da die IDs numerisch sind, lassen sich maximal zehn Kurztasten, von 0 bis 9, zuordnen und somit auch nur entsprechend viele Elemente in der Hauptnavigation unterbringen. Es muss also darauf geachtet werden, diese Grenze einzuhalten, was jedoch auch von den Richtlinien angesprochen wird, welche höchstens neun Hauptseiten empfehlen.

Für die Abarbeitung der Navigationseinträge kommen rekursive Funktionen zum Einsatz, die anhand der Ebene der Navigationsstrukturen bestimmen, ob es sich um Haupt- oder Unterseiten-Navigation handelt. Auch zur Erzeugung des Sitemaps werden entsprechende Funktionen ausgeführt. Anhand der aktuellen Seiten-ID wird bestimmt, welche Seite zur Zeit

geladen ist. Dies ist vor allem für die Generierung der Breadcrumb-Verknüpfungen wichtig, wird jedoch auch genutzt, um die aktive Seite in den Menüs zu markieren.

Der Einsatz einer Breadcrumb-Navigation macht es erforderlich, den vom Benutzer gewählten Pfad durch die Seitenstruktur zu speichern. Dafür kommt eine Funktion zum Tragen, die je nach ID der angeforderten Seite den entsprechenden Weg in der Navigationsstruktur durchläuft und dabei die einzelnen Schritte speichert. Nur so lässt sich der Pfad korrekt anzeigen, wenn beispielsweise ein direkter Sprung zu einer Unterseite der dritten Ebene erfolgt, was durch manuelle Änderung des **GET**-Parameters durchaus möglich ist. Als Beispiel befindet sich der Quelltext der Breadcrumb-Navigation im Anhang F (Quelltext 2).

Die Style-Switcher-Funktionen ermöglichen die Änderung von Schriftgröße und Präsentationsstil. Dargestellt werden sie durch Auswahllisten in XHTML. Die Übertragung der gewählten Einstellung erfolgt mit Hilfe von **POST**-Parametern, die durch die entsprechenden Funktionen ausgewertet werden. Bei einer Änderung von Schriftgröße oder Präsentationsstil über den Style-Switcher wird die Seite neu aufgebaut und es erfolgt mittels **POST** eine Übergabe der Änderungen an die zuständige Funktion. Die Schriftgrößen-Änderung erfolgt in 20%-Schritten. Der ermittelte Wert wird dann im entsprechenden XHTML-Template in eine Stildefinition eingesetzt. Ähnlich arbeitet auch die Funktion zur Änderung des Stylesheets, wobei hier der Name der zu ladenden Definitionsdatei getauscht und an das Template übergeben wird. Problematisch werden solche Änderungen beim Seitenwechsel, da sie dann verloren gehen. Aus diesem Grund speichern die Style-Switcher-Funktionen den gewählten Zustand in einer **SESSION**-Variablen. So ist es möglich, die gewünschten Einstellungen auch beim Seitenwechsel beizubehalten.

7.3. Präsentationsebene

Für die optische Gestaltung wurde ausschließlich valides XHTML1.0 in der Strict-Variante in Verbindung mit CSS2.0 eingesetzt. Alle Markup-Bestandteile, außer den dynamisch erzeugten, sind dabei in Templates abgelegt. Da durch die Verwendung des Frameworks alle Seitenbereiche in unterschiedliche Templates ausgelagert sind und erst beim Seitenaufruf zusammengesetzt werden, ergeben sich während der Entwicklung einige Probleme. Es muss immer sichergestellt sein, dass Markup-Dokumente valides XHTML verwenden. Vor allem bei Seitenbereichen, die sich über mehrere Templates erstrecken, kann die Einhaltung dieser Forderung jedoch schwierig sein. Wie in der Beschreibung der Funktionsebene bereits angesprochen, resultiert daraus auch, dass die Erzeugung der unterschiedlichen Bereiche in einer festen Reihenfolge geschehen muss.

Problematisch stellt sich auch die Verwendung von CSS dar. Nach den Richtlinien für barrierefreie Informationsangebote muss garantiert werden können, dass der Auftritt mit jeder Browsingtechnologie genutzt werden kann. Die Schwierigkeit besteht darin, valides CSS zu implementieren und trotzdem ein einheitliches Erscheinungsbild zu gewährleisten. Der entwickelte Auftritt ist sowohl mit gängigen Browsern wie dem Internet Explorer 6.0 und Firefox 1.0 als auch mit alternativen und älteren Browsingtechnologien, wie zum Beispiel dem Textbrowser Lynx und dem veralteten Browser Netscape 4.7, getestet worden. Da Letzterer nur mangelnde CSS-Unterstützung bietet, musste der bereits vorgestellte **@import**-Befehl zum Laden der Stildefinitionen genutzt werden, die der Browser übergeht.

Probleme mit Stylesheets haben aber nicht nur ältere Browser. Auch gängige Programme, wie der Internet Explorer, verhalten sich bei einer Vielzahl von CSS-Regeln nicht konform. Dies liegt vor allem an schlechter Umsetzung der CSS-Spezifikationen durch die Software-Hersteller. Die Fehler reichen von falscher Berechnung von Innen- und Außenabständen bis hin zu Elementen, die nicht angezeigt werden. Für alle Probleme gibt es Möglichkeiten, sie zu umgehen, allerdings oftmals nur unter Verwendung von invaliden Stildefinitionen.



Abbildung 10. Seite mit Unterseiten-Navigation und Standard-Stil

Farben und Kontraste wurden so gewählt, dass es keine Kombinationen gibt, die beispielsweise für Benutzer mit Farbblindheit problematisch sind. Auch Hintergrundbilder werden nicht eingesetzt, da diese oftmals die Lesbarkeit des Textes erschweren. Per Style-Switcher lassen sich, entsprechend der Anforderungsdefinition, zusätzlich verschiedene Präsentationsstile auswählen. Um Kontrastproblemen vorzubeugen, wird in den kontrastreichen Darstellungsmodi keine Kopfzeile dargestellt, da diese allein durch grafische Elemente umgesetzt und somit kein Farbwechsel möglich ist.

Das Bildschirmfoto zeigt eine Seite des Web-Auftrittes in der Standarddarstellung. Trotz der Verwendung von Grafiken sind alle Informationen auch textuell verfügbar, das betrifft auch die oben abgebildete Karte. Im konkreten Fall wurde auf den Einsatz einer Image-Map verzichtet und die Standortinformationen über ein Unterseiten-Menü zugänglich gemacht.

7.4. Evaluierung des Auftrittes

Der entwickelte Web-Auftritt trägt das Logo zur Doppel-A-Konformität nach WCAG1.0-Richtlinien. Die Überprüfung auf Zugänglichkeit wurde zunächst mit Hilfe der automatischen Validatoren *Bobby* und *Cynthia Says* durchgeführt. Die manuelle Analyse erfolgte anhand der Evaluierungsmethode, die in Kapitel 5.3 vorgestellt wurde. Dabei sind alle Anforderungen eingehalten worden. Die folgende Auflistung enthält die Merkmale des Auftrittes, geordnet nach den Prüfkriterien der Evaluierungsmethode.

Kriterien aus 5.3.1 (grundlegende Kriterien):

1. Inhalt und Präsentation sind durch Verwendung von XHTML1.0 und CSS2.0 voneinander getrennt.
2. Die Überprüfung des Markups wurde mit Hilfe des W3C-Markup-Validators vorgenommen und erfolgreich abgeschlossen.
3. Auch die Stildefinitionen sind nach Analyse mit Hilfe des W3C-Validators konform zu den Spezifikationen.
4. Es wurden keine Layout-Tabellen oder Frames verwendet, die Seitenstruktur wird durch ein Box-Modell in CSS definiert.
5. Ohne den Einsatz von CSS sind alle Informationen zugänglich. Der Auftritt wird dann linearisiert dargestellt.

Ergebnis: Kriterien nach 5.3.1 wurden insgesamt bestanden.

Kriterien nach 5.3.2 (optische Gestaltung):

1. Die Präsentation ist in verschiedenen Kontraststufen möglich. Grafiken und Schriften heben sich gut vom Hintergrund ab.
2. Auf Farbkombinationen (Rot/Grün sowie Blau/Grün in unmittelbarer Nähe zueinander) wurde verzichtet.
3. Elemente werden nicht nur durch die Wahl von Farben hervorgehoben. Aktive Navigationselemente enthalten ein entsprechendes Zeichen (»).
4. Vordergrund- und Hintergrundfarben lassen sich entweder per Style-Switcher oder über entsprechende Browserfunktionen verändern.
5. Schriftgrößen wurden relativ bemaßt und lassen sich durch den Style-Switcher oder im Browser stufenlos verändern.
6. Der Auftritt enthält keine Laufschriften oder blinkende Inhalte.

7. Das Darstellungsschema *Kontrast invertiert* enthält keine weißen, sondern von Weiß abgestufte Schriftfarben.

Ergebnis: Kriterien nach 5.3.2 wurden insgesamt bestanden.

Kriterien nach 5.3.3 (Alternativen für Audio- und visuelle Inhalte):

1. Jedes Bildelement ist mit Hilfe des **alt**-Attributes mit einem Alternativtext versehen worden.
2. Audio-Inhalte wurden nicht eingesetzt.
3. Es existieren keine Filme oder Animationen.
4. Auf die Verwendung von Image-Maps ist verzichtet worden.

Ergebnis: Kriterien nach 5.3.3 wurden insgesamt bestanden.

Kriterien nach 5.3.4 (Navigation und funktionale Elemente):

1. Alle Navigationselemente tragen eindeutige, aussagekräftige Beschriftungen.
2. Die Hauptnavigation ist auf allen Seiten, einschließlich den Unterseiten, konsistent.
3. Aktive Navigationselemente sind durch ein entsprechendes Zeichen (») markiert. Zusätzlich steht eine Breadcrumb-Navigation zur Verfügung.
4. Auf eingebettete Skripts und Applets wurde vollständig verzichtet.
5. Der Auftritt enthält auch keine Skripts und Applets für Navigationsstrukturen. Auch zeitsensitive Inhalte werden nicht eingesetzt.
6. Alle Verknüpfungen sind durch entsprechende Funktionen nummeriert. Die Reihenfolge entspricht ihrer Position auf der Gesamtseite.
7. Es sind keine dynamischen Navigationsstrukturen eingesetzt worden.
8. Alle Elemente der Hauptnavigation sind durch Kurztasten erreichbar.
9. Insgesamt enthält die Seite 8 Hauptnavigationseinträge und ist auf 3 Ebenen begrenzt.

Ergebnis: Kriterien nach 5.3.4 wurden insgesamt bestanden.

Kriterien nach 5.3.5 (Seitenstruktur und Markup):

1. Die Seitenstruktur entspricht den Forderungen, da zunächst die Breadcrumb-

Navigation, gefolgt von der Hauptnavigation und dem Inhaltsbereich sowie gegebenenfalls der Unterseiten-Navigation, dargestellt wird.

2. Alle Seitenbereiche enthalten Sprungmarken, um direkt zum nächsten Bereich zu gelangen. Diese sind nur ohne die Verwendung von CSS sichtbar.
3. Jeder Seitenbereich trägt eine Beschriftung, die nur ohne CSS ausgegeben wird.
4. Es existiert eine Seite *Hilfe & Sitemap*, die Hinweise zu Funktionen enthält, welche die Zugänglichkeit unterstützen. Zudem gibt es einen Überblick über die Seitenstruktur.
5. Verknüpfungen sind durch oben genanntes Zeichen markiert und enden mit einem Punkt. Diese Markierungen sind ohne CSS immer und bei jedem Link sichtbar.
6. Überschriften sind durch die vorgesehenen Markup-Elemente definiert. Verknüpfungen bzw. Navigationsstrukturen wurden durch Listen umgesetzt.
7. Es sind keine Datentabellen vorhanden.
8. Formulare (Style-Switcher) sind mit Hilfe von **label** und **for** gekennzeichnet. Die Auswahllisten lassen sich so dem entsprechenden Formular zuordnen.

Ergebnis: Kriterien nach 5.3.5 wurden insgesamt bestanden.

Kriterien nach 5.3.6 (Natürliche Sprache):

1. Es wurde darauf geachtet, kurze und aussagekräftige Sätze sowie Formulierungen zu verwenden. Einige Sätze überschreiten die empfohlene Länge von 9 Worten, sind dennoch so prägnant wie möglich formuliert worden.
2. Die verwendete Sprache ist durch das **xml:lang**-Attribut spezifiziert.
3. Sprachwechsel treten nicht auf.
4. Abkürzungen, beispielsweise *FAW*, werden mit entsprechenden XHTML-Elementen eingesetzt.

Ergebnis: Kriterien nach 5.3.6 wurden insgesamt bestanden.

Insgesamt wurden alle Kriterien der höchsten Prioritätsstufen eingehalten. Da die manuelle Evaluierungsmethode eine Zusammenfassung der Richtlinien der WAI und des Bundes darstellt, beinhaltet dies auch die Barrierefreiheit des Auftrittes im Sinne der Konformitätsstufe A der WCAG1.0 und der ersten Stufe der BITV. Auch die optionalen Richtlinien der zweiten Stufe wurden umgesetzt. Der Web-Auftritt der Akademie Cottbus wird somit allen Forderungen der BITV gerecht und erfüllt auch die Anforderungen der zweiten Konformitätsstufe der WCAG1.0.

8. Zusammenfassung

Barrierefreie Informationsangebote gewinnen mehr und mehr an Bedeutung. Für Menschen mit physischen oder psychischen Einschränkungen bietet die Verwendung des Internet Möglichkeiten, zu denen sie im sonstigen Leben keinen oder nur eingeschränkten Zugang haben. Durch das Internet können soziale und gesellschaftliche Kontakte geknüpft, Einkäufe getätigt oder Bankgeschäfte erledigt werden. Bei der Erstellung von Web-Auftritten sollten daher grundsätzlich für Barrierefreiheit Sorge getragen werden, obwohl der Entwurf und die Entwicklung mit erheblich höherem Aufwand verbunden sind als es bei herkömmlichen Systemen der Fall ist. Dies betrifft jedoch nicht nur Entwickler, sondern erfordert ein Umdenken aller Personen, die an Entwurf, Gestaltung und Implementierung eines Web-Auftrittes beteiligt sind. Auch andere Verantwortliche, beispielsweise Unternehmensleitungen, müssen bereits während der Planung von Informationsangeboten eine Reihe von Aspekten der Barrierefreiheit beachten. Die schrittweise Modellierung eines solchen Systems hilft, die Zugänglichkeit der verschiedenen Ebenen zu garantieren und letztendlich ein barrierefreies Informationsangebot zu entwickeln.

Es kann jedoch nie Ziel sein, einen vollständig barrierefreien Web-Auftritt zu erstellen, vielmehr müssen so viele Hindernisse wie möglich ausgeschlossen werden. Diese Zugänglichkeit betrifft alle Benutzer, nicht nur Menschen mit Beeinträchtigungen. Soll ein entsprechendes Informationsangebot erstellt werden, ist es zunächst notwendig, die Bedürfnisse der verschiedenen Benutzergruppen zu erkennen und zu verstehen. Vor allem das stellt viele Entwickler vor große Probleme. Hilfreich kann es sein, während der Entwicklung verschiedene Ein- und Ausgabegeräte bzw. Browsingtechnologien zu nutzen, um ein Gefühl für die Probleme zu bekommen, vor denen Menschen mit Beeinträchtigungen stehen. Doch auch ohne den Einsatz verschiedener Technologien lassen sich durch die Einhaltung einiger Regeln zugängliche Angebote schaffen.

Beginnend bei der Trennung von Inhalt und optischer Präsentation, über die Verwendung von Markup-Dokumenten, die den Spezifikationen entsprechen, bis hin zu der Einhaltung von Richtlinien, die sich mit dem Thema der Barrierefreiheit beschäftigen, stehen den Entwicklern viele Möglichkeiten offen, die Zugänglichkeit zu verbessern. Die in dieser Arbeit entwickelte Evaluierungsmethode basiert auf den Richtlinien der WAI, stuft aber die verschiedenen Kriterien in andere Prioritäten ein. Sie ermöglicht die Analyse und Bewertung bereits bestehender Web-Auftritte und ist für die Entwicklung neuer Angebote hilfreich. Die Einhaltung aller Regeln sollte immer angestrebt werden und stellt eine Barrierefreiheit nach der zweiten Konformitätsstufe der WCAG1.0 dar.

Während die Wahl der verwendeten Markup-Sprache abhängig vom Umfang ist, den der entsprechende Auftritt annehmen wird, stehen für die Umsetzung der optischen Gestaltung Cascading Stylesheets zur Verfügung. Dadurch ist eine Trennung von Inhalt und Präsentation möglich. CSS bietet eine Vielzahl von Hilfsmitteln, die für die Erstellung komplexer Präsentationen zum Einsatz kommen können. Problematisch ist dabei nur die Interpretation der Stildefinitionen durch die unterschiedlichen Browser, da zum Teil nicht alle Inhalte der CSS-Spezifikationen unterstützt oder aber einige falsch interpretiert werden. Zudem sollen barrierefreie Angebote auch auf älteren oder alternativen Browsern lesbar sein, die aber meist nicht oder nur eingeschränkt mit CSS umgehen können. Letztendlich erschweren diese Aspekte die Umsetzung ansprechender Gestaltungen und erhöhen den Einarbeitungsaufwand für Entwickler. Trotzdem ist die vielfach vertretene Meinung, barrierefreie Web-Auftritte müssen sich mit reinen textuellen Darstellungen begnügen, nicht haltbar.

Der Web-Auftritt der Fortbildungsakademie der Wirtschaft gGmbH, Akademie Cottbus wurde entsprechend der Doppel-A-Konformität als praktischer Teil dieser Arbeit entwickelt. Er setzt alle Richtlinien der vorgestellten manuellen Evaluierungsmethode sowie die ersten beiden Richtlinien der WCAG1.0 um und verwendet Markup und CSS, welche konform zu den Spezifikationen des W3C sind. Als zusätzliche Funktionen stehen Style-Switcher zur Verfügung, über die Schriftgrößen und verschiedene Stile auszuwählen sind. Möglich ist dies durch die Verwendung der Skript-Sprache PHP, wobei die gesamte Funktionsebene auf Objektorientierung beruht. Dies vereinfacht die Pflege und die Austauschbarkeit der Skripts. Die Evaluierung des Web-Auftrittes erfolgte durch automatische Validatoren und zusätzlich durch die in dieser Arbeit beschriebene Methode.

Auch wenn noch keine Verpflichtung durch den Gesetzgeber für private Einrichtungen besteht, sollte es für alle Anbieter und Entwickler von öffentlich zugänglichen Informationssystemen selbstverständlich sein, barrierefreie Seiten zu erstellen oder bestehende Auftritte zu überprüfen und gegebenenfalls zu modifizieren. Für die Zukunft ist eine weitere Steigerung der Anzahl barrierefreier Web-Auftritte zu erwarten, zumal es mit kommenden Browsingtechnologien bessere Unterstützung für bestehende Standards und Spezifikationen geben wird. Die Verabschiedung neuer Richtlinien zur Barrierefreiheit, wie zum Beispiel den WCAG2.0, wird neue Möglichkeiten eröffnen. Entwickler und auch Verantwortliche in den Unternehmen können dann ihre Informationsangebote gezielter und mit größerem Verständnis für die Belange beeinträchtigter Menschen entwerfen und implementieren, wobei letztendlich jeder Benutzer von leicht zugänglichen Informationen im Internet profitiert.

Literaturverzeichnis

Henning Behme und Stefan Mintert. Copyright © 2000. 3-8273-1636-7. Addison-Wesley-Verlag. *XML in der Praxis*.

Matthias Kannengiesser und Damir Enseleit. Copyright © 2004. 3-7723-6559-0. Franzis Verlag GmbH. *PHP5 - Praxisbuch und Referenz*.

Stefan Münz und Wolfgang Netzger. Copyright © 2002. 3-8273-1636-7. Franzis Verlag GmbH. *HTML - Die Profireferenz*.

Technischer Jugendfreizeit- und Bildungsverein (tjfbv) e.V.. Copyright © 2003. *Broschüre Barrierefrei kommunizieren*. <http://www.barrierefrei-kommunizieren.de>.

World Wide Web Consortium. Copyright © 1999. *Web Content Accessibility Guidelines 1.0*. <http://www.w3.org/TR/WCAG10/>.

World Wide Web Consortium. Copyright © 2004. *Web Content Accessibility Guidelines 2.0 Working Draft*. <http://www.w3.org/TR/WCAG20/>.

[bib1] Wikimedia Foundation Inc.. *Braille-Schrift*. <http://de.wikipedia.org/wiki/Brailleschrift>.

[bib2] Wikimedia Foundation Inc.. *Großfeldtastatur*. <http://de.wikipedia.org/wiki/Großfeldtastatur>.

[bib3] Wikimedia Foundation Inc.. *Lynx (Browser)*. [http://de.wikipedia.org/wiki/Lynx_\(Browser\)](http://de.wikipedia.org/wiki/Lynx_(Browser)).

[bib4] Bundesministerium des Innern. Copyright © 2002. *Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz*. <http://bundesrecht.juris.de/bundesrecht/bitv/>.

Bundesministerium für Gesundheit und Soziale Sicherung. Copyright © 2002. *Gesetz zur Gleichstellung behinderter Menschen*. <http://bundesrecht.juris.de/bundesrecht/bgg/>.

Ministerium für Arbeit, Soziales, Gesundheit und Familie des Landes Brandenburg. Copyright © 2003. *Gesetz zur Gleichstellung behinderter Menschen im Land Brandenburg*. http://www.brandenburg.de/media/1336/bgg_brandenburg.pdf.

Maxdesign. Copyright © 2003, 2004. *A web standards checklist*. <http://www.maxdesign.com.au/presentation/checklist.cfm>.

Jan Eric Hellbusch. *Das Web und die Zugänglichkeit*. <http://www.barrierefreies-webdesign.de>.

Aktion Mensch. *Einfach für alle*. <http://www.einfach-fuer-alle.de>.

Bundesministerium für Gesundheit und soziale Sicherung. Copyright © 2005. *DVD Barrierefreies Internet*.

A. Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Bachelor-Arbeit selbständig und ohne unerlaubte Hilfe angefertigt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Cottbus, 26.07.2006

Jan Böschow

B. Abbildungsverzeichnis

Abbildung 1	Braillezeile für die Ausgabe von Bildschirmhalten in Blindenschrift
Abbildung 2	Großfeldtastatur der Firma Igel für motorisch behinderte Benutzer
Abbildung 3	Textbrowser Lynx
Abbildung 4	Validator "Cynthia Says": Auszug aus den Analyseergebnissen von www.google.de nach Konformitätsstufe 3 WACG1.0
Abbildung 5	Validator "Bobby": Auszug aus den Analyseergebnissen von www.google.de nach Konformitätsstufe 3 WACG1.0
Abbildung 6	Inhaltsstrukturdiagramm des Web-Auftrittes der FAW gGmbH, Akademie Cottbus
Abbildung 7	3-Schichten-Modell (3Tier-Layout)
Abbildung 8	Rahmenarchitektur (Entwurfsmuster Framework) des Web-Auftrittes der Akademie Cottbus
Abbildung 9	Datenstruktur der veränderlichen Daten des Web-Auftrittes
Abbildung 10	Bildschirmfoto: Seite mit Unterseiten- Navigation und Standard-Stil
Abbildung 11 (Anhang F)	Kompositionsmodell in WebML
Abbildung 12 (Anhang F)	Klassendiagramm

C. Abkürzungsverzeichnis

BbgBGG	Brandenburgisches Behindertengleichstellungsgesetz
BGG	Behindertengleichstellungsgesetz
BITV	Barrierefreie Informationstechnik- Verordnung
CSS	Cascading Style Sheets
DTD	Document Type Definition
GPL	GNU General Public License
HTML	Hypertext Markup Language
HTTPS	Hypertext Transfer Protocol Secure
PHP	Rekursives Akronym: PHP Hypertext Preprocessor
SGML	Standard Generalized Markup Language
SVG	Scalable Vector Graphics
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WAI	Web Accessibility Initiative
WCAG	Web Content Accessibility Guidelines
WebML	Web Modelling Language
WWW	World Wide Web
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language
XPath	XML Path Language
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language Transformation

D. Glossar

Accessibility	Englischer Begriff für Zugänglichkeit bzw. Barrierefreiheit.
Braillezeile	Ausgabegerät für sehbehinderte Menschen, das an den Rechner angeschlossen werden kann und Bildschirminhalte in Blindenschrift ausgibt.
Browser	Software zur Anzeige von Web-Seiten
Browsingtechnologie	Sammlung aller Möglichkeiten (Software/Hardware) zur Anzeige bzw. Ausgabe von Web-Seiten
Evaluierung	Überprüfung/Analyse auf Einhaltung bestimmter Regeln.
Image-Map	Grafik, die mit mehreren Verknüpfungen hinterlegt ist. Oft genutzt für die Gestaltung von Landkarten.
Markup	Quelltext, der in einer Sprache, wie zum Beispiel HTML oder XML erstellt wurden.
Screenreader	Software, die Bildschirminhalte als Sprache über angeschlossene Audiogeräte und/oder auf Braillezeilen ausgibt.
Skript-Sprachen	Programmiersprachen, die für kleinere Aufgaben geeignet sind und oft für Web-Anwendungen genutzt werden.
Stylesheets	Dokumente oder Bereiche in einer Web-Seite, die Definitionen zur optischen Präsentation der Seite enthalten.
Validator	Software, die andere Software oder Web-Seiten auf die Einhaltung bestimmter Regeln hin überprüft.
Validität	Den spezifischen Regeln entsprechende Dokumente

E. XHTML-Elemente und -Attribute

Attribut	Attribut zulässig in Element	Beschreibung
abbr="TEXT"	td, th	Kurzbeschreibung für (Kopf-)Zellen von Tabellen
accesskey="TASTE"	a, input, label, legend, textarea	Tastenangabe für Kurztasten-Zugriffe auf Link-Elemente, Formulare
axis="NAME"	td, th	Kategorienamen für (Kopf-)Zellen von Tabellen
alt="TEXT"	area, img	Kurzer Alternativtext für Bilder und Grafiken (auch Image-Maps)
for="ID"	label	Bezug einer Beschriftung zu einem Element mit ID
headers="ID"	td, th	Bezug einer Zeile zu einer anderen Zelle mit ID
id="ID"	allen XHTML-Elementen	Eindeutige Identifizierung eines Elementes
longdesc="URL"	img	Adresse für lange Alternativtexte für Bilder und Grafiken
scope="BEREICH"	td, th	Geltungsbereich einer Zelle (mögliche Werte: col, row, colgroup, rowgroup)
style="CSS-DEFINITION"	allen XHTML-Elementen	Erlaubt die Festlegung von Stilen für das entsprechende Element
summary="TEXT"	table	Einführende Zusammenfassung über den Inhalt einer Tabelle
tabindex="NR"	a, area, input, object, select, textarea	Nummerierung von Elementen. Gibt die Reihenfolge an, in der sie durch die Tabulatortaste

Attribut	Attribut zulässig in Element	Beschreibung
		angesprochen werden
title="TEXT"	allen XHTML-Elementen	Element-Titel. Wird meist angezeigt, wenn das Element mit dem Mauszeiger berührt wird
xml:lang="CODE"	allen XHTML-Elementen	Legt die verwendete Sprache fest und kennzeichnet Wechsel der natürlichen Sprache

Element	Art	Beschreibung
abbr	Inline	Auszeichnung einer Abkürzung
acronym	Inline	Zeichnet Text als Akronym aus
blockquote	Block	Definiert einen Text als Zitat
cite	Inline	Definiert Quelle und Autor für ein Zitat (Zitat mit q oder blockquote)
dfn	Inline	Zeichnet Text als Definition aus
div	Block	Eigenschaftsloser Block (kann mit CSS formatiert werden)
dl	Block	Liste von Definitionen
dt	darf nur innerhalb von dl vorkommen	Listeneintrag in Definitionsliste
em	Inline	Markiert Text als besonders wichtig (wird von Screenreadern stärker betont)
fieldset	Block	Fasst mehrere logisch zusammengehörige Formularfelder in einer Struktur zusammen

Element	Art	Beschreibung
h1 bis h6	Block	Definiert Text als Überschrift
li	darf nur innerhalb von ol oder ul vorkommen	Listeneintrag in (un-)geordneter Liste
ol	Block	Geordnete (nummerierte) Liste
span	Inline	Eigenschaftsloser Bereich (kann mit CSS formatiert werden)
strong	Inline	Zeichnet Text als stärker betont aus (wird von Screenreadern beachtet)
ul	Block	Ungeordnete Liste

Anmerkung

Blockelemente erzeugen immer einen Absatz und dürfen Text, Inline-Elemente und teilweise auch andere Block-Elemente enthalten. Inline-Elemente erzeugen keinen Absatz, werden innerhalb von Block-Elementen eingesetzt und dürfen meist weitere Inline-Elemente, aber keine Block-Elemente enthalten.

F. Dokumente zur Modellierung und Implementierung

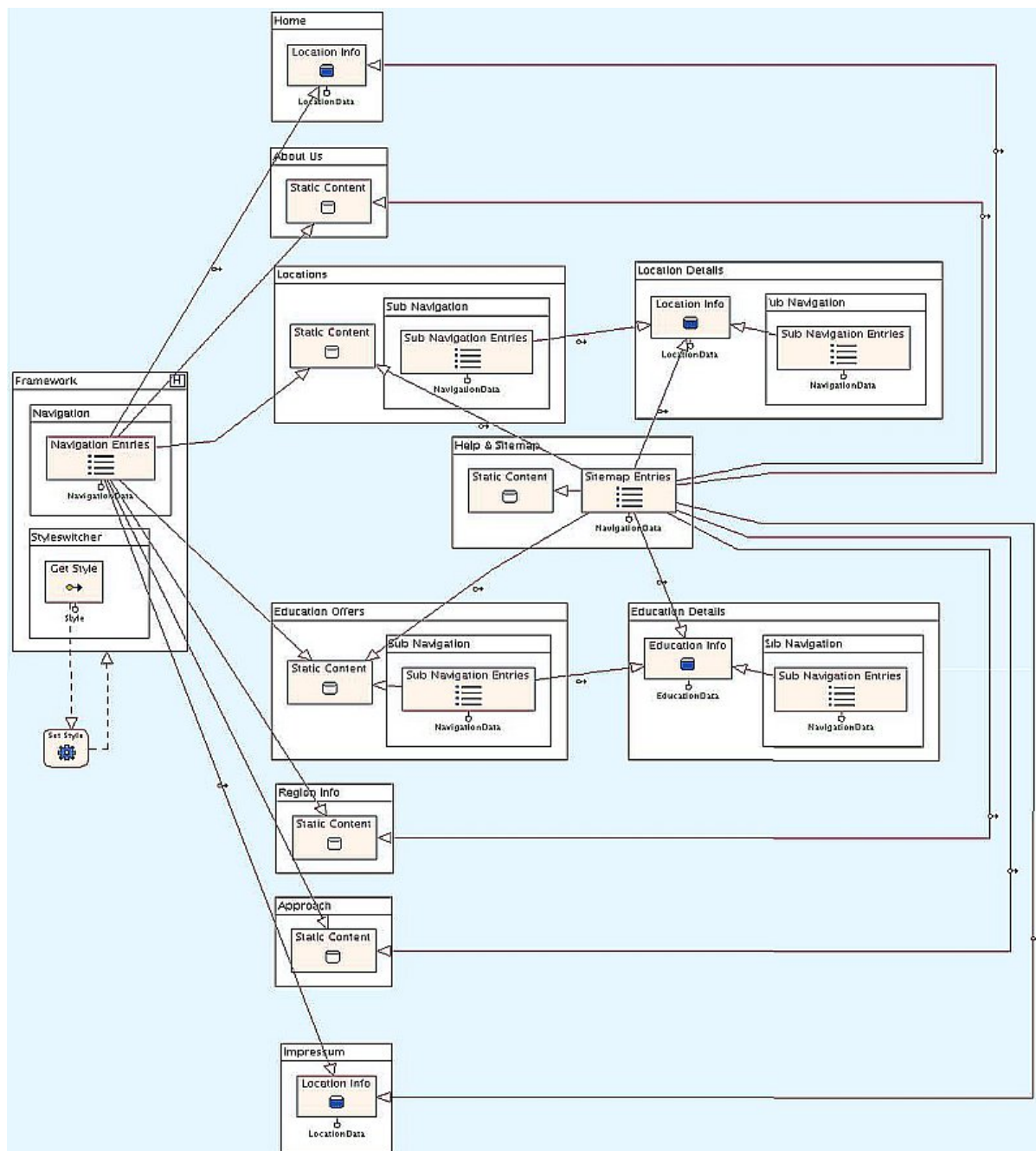


Abbildung F.1. Kompositionsmodell in WebML

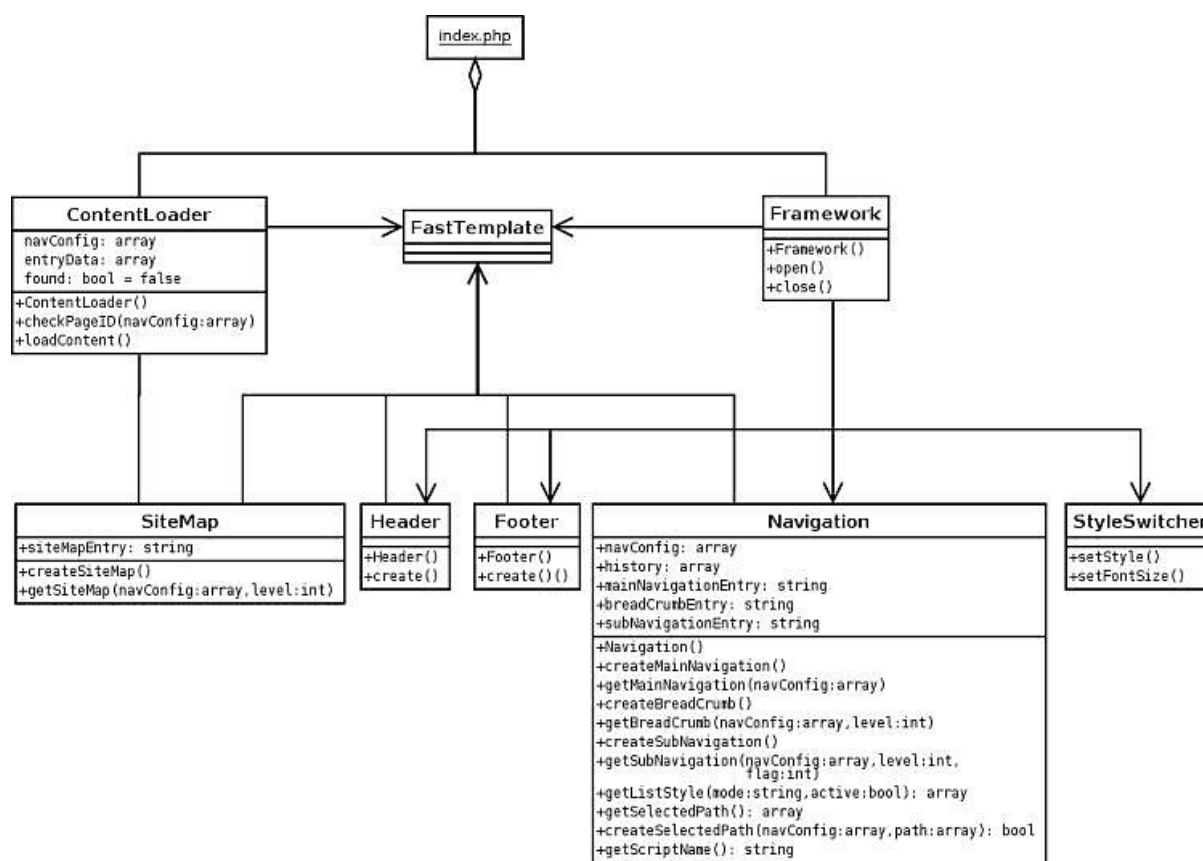


Abbildung F.2. Klassendiagramm

```

/*
 * Breadcrumb-Navigation.
 * Durchsucht rekursiv alle Ebenen des Navigations-Arrays und erzeugt
 * den zugehoerigen XHTML-Link, falls Eintrag=aktive Seite.
 *
 * @param: $navConfig = array mit Navigationskonfiguration
 * @param: $level = integer Aktuelle Ebene im Array
 */
function createBreadcrumb ($navConfig, $level) {
    foreach ($navConfig as $id => $entry) {
        $path = $this->getSelectedPath();
        $active = ($id == $_SESSION['id']) || ((isset ($path[$level])
            && ($path[$level] == $id)) ? true : false;
        if ($active) {
            $this->breadcrumbEntry .= '<li class="breadcrumb"><strong>';
            $this->breadcrumbEntry .= '<a href="?id='.$id.'" ';
            $this->breadcrumbEntry .= 'tabindex="'. $_SESSION['tabindex'].'">';
            $this->breadcrumbEntry .= '<span class="bullet_visible">&#187; </span>';
            $this->breadcrumbEntry .= htmlspecialchars ($entry['name']);
            $this->breadcrumbEntry .= '</a><dfn>.</dfn></strong></li>';
            $_SESSION['tabindex']++;

            if (isset($entry['sub']))
                $this->createBreadcrumb ($entry['sub'], $level + 1);
        } // if
    } // foreach
} // function createBreadcrumb()

```