

Inleiding	3
Probleemstelling	3
Accessibility en Usability	4
Definitie accessibility	4
Definitie usability	4
Het internet en accessibility	5
Inleiding	5
Organisaties en wetgeving	5
W3C	5
WaSP	5
Section 508.....	6
ISO/TS 16071:2003.....	6
ISO/IEC 9126.....	6
Web development en accessibility.....	7
Scheiding van inhoud en opmaak	7
De vier lagen van web development	8
De eerste laag: server-side scripting (PHP, ASP, .Net)	8
De tweede laag: structuur en content (XHTML, XML)	8
De derde laag: client-side manipulatie van content en interactie (DOM)	8
De vierde laag: opmaak en positionering (CSS, XSL)	8
HTML zoals HTML bedoeld is	9
WAI-WCAG.....	11
Equivalenten alternatieven bieden voor audiovisuele content;	11
Leesbaar zijn zonder opmaak	12
Duidelijk zijn in welke taal er gesproken wordt	12
Mediumafhankelijk zijn.....	13
Duidelijke navigatie hebben.	13
Web development en usability	14
Waarom usability?	14
Usability in websites	14
Site navigatie.....	14
User Interface	15
Formulieren.....	15
Zoekfunctie	15
Ergernissen.....	16
Onduidelijkheid	16
Verwarring / Onverwachte dingen.....	16
Traagheid	16
Afhankelijkheid	16
De weg naar een vriendelijker internet	16
Accessibility, usability en Grafimatechtechnologie	17
Literatuurlijst.....	18

Inleiding

Usability en accessibility, twee mooie termen die vaak te pas en te onpas gebruikt worden. Wat betekenen die twee dingen eigenlijk precies, met name als het om internettoepassingen gaat? Wat zijn de ontwikkelingen op dit gebied?

“What’s up with web accessibility and usability these days?”

Maar goed, alle doodoeners daargelaten: usability en accessibility zijn de twee belangrijkste termen als het gaat om ‘klanttevredenheid’, en klanten zijn je brood. Het is dus belangrijk dat er aandacht besteed aan wordt.

Dit document beschrijft de huidige stand van zaken over deze twee onderwerpen en hun relatie met web development. Er zal ook met terugwerkende kracht gekeken worden naar de Grafimediatechnologie opleiding, waarna er aanbevelingen gedaan worden naar hoe er (meer) aandacht besteed kan worden aan deze onderwerpen binnen de opleiding.

Bewust als we zijn van het hypocriete ervan, willen we graag vermelden dat tijdens het schrijven van dit document, er van uit gegaan is dat de lezer enige kennis heeft op het gebied van web development.

Probleemstelling

Maar wat willen we precies hiermee? Wat is de drijfveer om hier onderzoek naar te doen? En hoe wordt dit uiteindelijk toegepast?

Dit rapport is geschreven in het kader van de workshop module van de Hogeschool Rotterdam. De bedoeling ervan is het verschaffen van informatie die betrekking heeft op het (globale) onderwerp: web accessibility & usability. Naast het geven van feitelijke informatie willen we er een persoonlijk tintje eraan geven door onze visie hierop te beschrijven. Dit wordt verder uitgelegd met modellen en toepassingsvoorbeelden. Op deze manier denken wij uiteindelijk antwoord te kunnen geven op de vraag:

“Wat voor oplossingen zijn er op het gebied van accessibility en usability die toepasbaar zijn binnen deze opleiding om kwaliteit van door studenten ontwikkelde websites te verbeteren?”

Accessibility en Usability

Allemaal leuk en aardig, maar wat houdt dat nou allemaal precies in? Laten we eerst definiëren waar we het precies over hebben, wel zo handig:

Definitie accessibility

“The degree to which web content is accessible to people, regardless of disability.”

Concreet gezegd houdt dit de mate in waarin content op het internet toegankelijk is voor wie dan ook. Net als bij openbare gebouwen moeten websites toegankelijk zijn voor mensen met en zonder handicap.

Definitie usability

“The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments.”

Usability is dus de mate van gebruiksvriendelijkheid van een product. In het geval van web usability is dit product de website.

Het internet en accessibility

Inleiding

Internet is een enorme bron van informatie, in sommige gevallen openbaar en in sommige gevallen afgeschermd. Deze informatie kan op meerdere manieren digitaal worden verspreid. De meest gebruikte toepassingen op dit moment zijn e-mail en websites.

Door de explosieve groei van het internet de afgelopen jaren is het een medium geworden dat op gelijke voet staat met andere informatiedragers als televisie, radio en krant, in sommige gevallen zelfs hoger.

Als internet zo belangrijk is geworden, waarom is het dan nog steeds zo slecht toegankelijk? Neem een simpel voorbeeld: een blind persoon kan de letters op het scherm uiteraard niet lezen; zou hij/zij daarom verstoten moeten blijven van dit medium? Als er aangepaste ondertiteling bestaat op televisie voor doven en slechthorenden, waarom zouden deze mensen dan audiovisuele content op websites niet ten volle kunnen beleven?

Dit geldt overigens niet alleen voor gehandicapten, ook ouderen die niet meer vast ter hand zijn en dus geen muis kunnen hanteren moeten zonder problemen een website kunnen bekijken, door middel van toetsenbordnavigatie bijvoorbeeld. Verder is er een opmars aan toepassingen aan de gang die ook het internet op kunnen. Denk aan telefoons, naast het inmiddels verouderde WAP zijn technieken als GPRS en UMTS steeds meer vertrouwd bij de consument. Maar ook handhelds (PDA's, palmtops) zijn steeds populairder aan het worden - met name in de zakenwereld.

Organisaties en wetgeving

W3C

De belangrijkste speler in deze ontwikkelingen is het World Wide Web Consortium (W3C), het internationale en overkoepelende internetorgaan. Binnen het W3C is er een speciale afdeling opgericht voor het benadrukken van accessibility op het internet. Het Web Accessibility Initiative (WAI) heeft onder andere richtlijnen opgesteld waar toegankelijke websites aan moeten voldoen, de Web Content Accessibility Guidelines, deze worden later nog besproken.

WaSP

Inmiddels zijn ook andere initiatieven genomen voor de bevordering van web accessibility; The Web Standards Project (WaSP), een in 1998 opgerichte organisatie voor de promotie en bevordering van web standaarden als (X)HTML, XML en CSS, is een andere belangrijke "coalitie", zoals ze zichzelf graag noemen.

Deze organisaties hebben samen flink aan de digitale snelweg getimmerd en niet zonder succes: de in 2001 gestarte campagne "Dreamweaver Task Force" heeft gezorgd voor een grote toename van accessibility in Macromedia's marktleidende visuele web editor. Sindsdien heeft de gehele productlijn van Macromedia, inclusief hun website grote vooruitgang geboekt in accessibility. Voorbeelden hiervan zijn verbeterde CSS ondersteuning, XHTML naleving en meer valide code generatie. Voor meer informatie, zie:

<http://webstandards.org/act/campaign/dwtf/>

Een andere campagne van WaSP, “Browse Happy”, wil benadrukken dat er (betere) alternatieven zijn in browserland dan Internet Explorer. Oppericht om andere browsers onder de aandacht te brengen met grotere veiligheid, betere navolging van de door W3C opgerichte standaarden en simpelweg betere functionaliteit biedt het WaSP alternatieven om - naar eigen zeggen - weer happy te browsen.

Section 508

“Inaccessible technology interferes with an individual's ability to obtain and use information quickly and easily. Section 508 was enacted to eliminate barriers in information technology, to make available new opportunities for people with disabilities, and to encourage development of technologies that will help achieve these goals.”

Aldus www.section508.gov, dat deze in 1998 in de Verenigde Staten aangenomen wet beschrijft. Volgens deze wet dienen overheidssites te voldoen aan gedefinieerde standaarden. Deze standaarden komen uit de door W3C opgestelde accessibility guidelines. Een belangrijk gegeven in de ontwikkeling van web accessibility, en ook in Nederland begint het belang hiervan te groeien: door een toename van zowel vergrijzing als gebruik van internet zal de groep met een functionele beperking steeds groter worden.

Aan de andere kant wordt informatie steeds vaker digitaal aangeboden. Dit onderstreept het toenemende belang van web accessibility en de verwachtingen zijn dat het niet lang zal duren voor een dergelijke wetgeving in Nederland ook van kracht zal zijn.

ISO/TS 16071:2003

“Ergonomics of human-system interaction -- Guidance on accessibility for human-computer interfaces”, is de titel van de ISO-richtlijnen voor accessibility.

ISO/IEC 9126

Dit zijn de ISO-richtlijnen voor usability.

Web development en accessibility

Scheiding van inhoud en opmaak

In het vorige hoofdstuk hebben we beschreven hoe de huidige stand van zaken is op het gebied van accessibility en usability. In theorie klinkt het ook allemaal aardig, maar hoe pas je dit nu toe op je websites? Hoe zorg je bijvoorbeeld dat mensen die niet kunnen zien jouw informatie krijgen via iets wat op het beeldscherm getoond wordt? Wat moet je doen als je image-map die je als navigatie gebruikt op een telefoon niet getoond wordt, omdat die niet genoeg geheugen heeft om grote plaatjes op te slaan?

Het antwoord hierop is simpel: scheid opmaak van inhoud. Plaatjes, bewegende gif-jes, flash animaties is allemaal heel leuk voor het oog, maar vergeet niet waar een website uiteindelijk om draait: informatie. Een bezoeker komt in de meeste gevallen voor informatie op een website, wil deze informatie snel hebben, zonder veel omwegen en afleiding. Zorg er dus voor dat te allen tijde de informatie zichtbaar is, op welke manier je dan ook surft.

“Maar het oog wil ook wat! Alleen maar tekst is ook zo saai, daar knappen bezoekers ook op af. Kan ik net zo goed een Word documentje schrijven.”

Klopt helemaal, gelukkig denken de mensen van W3C daar hetzelfde over: **enter CSS!** Cascading StyleSheets is ontworpen om die saaie HTML pagina's een beetje vormgeving te geven; tekst, positionering, kleurgebruik, achtergronden en zelfs enige vorm van interactie kan met behulp van een stylesheet.

“En wat heeft dat te maken met het verhaal hierboven? Wat heeft een blinde hieraan?”

Helemaal niets, een blinde kan immers niet zien. Echter door alle opmaak uit de HTML code van een pagina te houden kan een screenreader - een browser die tekst voorleest aan een blinde - gemakkelijk bij de tekst komen, zonder te hoeven worstelen door allerlei `` tags, opmaak attributen en tabellen die gebruikt worden voor de positionering van de layout. Op dezelfde wijze kan een telefoon gewoon dat plaatje negeren en het tekstmenu gebruiken om te navigeren.

Er zijn nog veel meer voorbeelden te noemen die het belang van het scheiden van opmaak en inhoud onderstrepen. Hier zullen we later nog op terug komen. We gaan eerst bespreken hoe het creëren van websites volgens bovenstaande manier tot uiting komt in de praktijk. Deze manier van werken vereist namelijk een bepaalde manier van denken, die je bezoekers uiteindelijk alleen maar ten goede komt. We willen dit graag illustreren door een model.

De vier lagen van web development

Technisch gezien doorloopt een webpagina 4 stappen voordat het uiteindelijk op het scherm getoond wordt. We bespreken al deze stappen even:

De eerste laag: server-side scripting (PHP, ASP, .Net)

Een pagina wordt door de browser opgevraagd aan een server. Deze server geeft antwoord in de vorm van een document. In het geval van een dynamische webpagina (bijvoorbeeld met een database-koppeling) wordt er een script uitgevoerd op de server (vandaar: server-side), waarna de output hiervan wordt terug gegeven. Voorbeelden van scripttalen zijn PHP en ASP.

De tweede laag: structuur en content (XHTML, XML)

Het document dat wordt terug gegeven is niets anders dan een bundel aan informatie, waar structuur en semantiek aan wordt gegeven door middel van een mark-up language, zoals HTML en XML. Op deze manier bestaat de informatie die het document bevat in een bepaalde structuur (bijvoorbeeld koppen, paragrafen en tabellen). Hierdoor is het door computers makkelijk te begrijpen (parsen), zonder dat daar allerlei andere uitleg voor benodigd is.

De derde laag: client-side manipulatie van content en interactie (DOM)

Dit document kan bij de bezoeker (client-side) gemanipuleerd worden door gebruik van een taal als Javascript. In principe kan alles aan de tweede laag gemanipuleerd worden door de gebruiker, mits de functionaliteit ervoor gegeven is (in de vorm van Javascript functies dus). De interface waarin dit gebeurt wordt het Document Object Model genoemd, en is ook in ontwikkeling door het W3C.



De vierde laag: opmaak en positionering (CSS, XSL)

De inhoud kan uiteindelijk vormgegeven worden door een 'opmaaktaal' als CSS of XSL. Waar het Document Object Model de inhoud en structuur van het document volledig kan beïnvloeden, kan CSS dit met de opmaak van het document. Op deze manier wordt inhoud dus van opmaak gescheiden.

"En hoe bevordert dit model mijn manier van denken?"

Door laag voor laag na te denken over hoe een website gemaakt moet worden zal de samenhang tussen deze lagen beter zijn, maar ook van de individuele lagen. Als je bij het schrijven van een pagina stil staat welke informatie belangrijker is, wat er eerst getoond moet worden en in welke vorm de informatie moet worden aangeboden krijg je een goed en semantisch (zichzelf omschrijvend) document. Een bijkomend voordeel hierbij is dat zoekmachines dit oppikken; belangrijke koppen worden eerder geïndexeerd dan een normale zin die door middel van een `` tag vergroot is om aandacht te trekken.

HTML zoals HTML bedoeld is

Bij deze manier van web development komt nog een ander aspect aan het licht: het ‘foutief’ gebruik van HTML. Nu is HTML alleen fout wanneer het niet goed gevalideerd wordt (syntax fouten, ontbreken van `<head>` tag etc.), maar je kan HTML ook foutief zien als er elementen gebruikt worden voor de verkeerde doelen. We maken een analogie naar een ‘normale taal’, Nederlands:

“Dees Sin Klopt Geen Meter Fan,,De Kohma”s Qlopt Ni, Elk Wort Begint Met Un Hooflettar En Ur Zitten Spelvauten Ihn.”

Dit is geen correct Nederlands, vergelijk dit met syntax fouten in een HTML pagina. Hierdoor valideert de pagina niet goed, we begrijpen niet goed wat er staat. We trekken de analogie even door:

- *“Met*
- *deze*
- *zin*
- *is*
- *niets*
- *mis,*

<i>we begrijpen</i>	<i>wat</i>	<i>er staat</i>
---------------------	------------	-----------------

en er staan geen spel- en grammaticafouten in.”

Dit is correct Nederlands, de zin loopt goed door en is vrij van fouten. Toch gebruiken we taal doorgaans niet op deze manier, logisch, het klopt voor geen meter. Die tabel en lijst slaan nergens op en dat laatste stuk staat rechts uitgelijnd.

Als het nergens op slaat, waarom wordt het dan bij HTML wel gedaan? Tabellen gebruiken om normale tekst in te plaatsen, `
` tags om tekst wat marge te geven, frames en iframes, je komt het vaak genoeg tegen op een niet/minder accessible site. Het is niet fout, maar HTML is niet ontworpen op deze manier.

Gebruik tabellen waarvoor ze bedoeld zijn: tabulaire data, niet voor layout, positionering van tekst of opsommingen. Gebruik CSS om marges en ruimtes aan te geven. Hier komt dat woord “semantisch” weer terug, een HTML document zou zichzelf moeten beschrijven. Dat betekent:

- kopjes tussen header tags (`<h1>` - `<h6>`);
- tekstparagrafen tussen paragraaf tags (`<p>`);
- opsommingen en lijsten in `` tags;
- `<legend>`, `<label>` en `<fieldset>` om formulieren toegankelijker te maken;
- `alt` attributen op `img` tags zodat afbeeldingen voor niet-visuele browsers beschreven worden;
- correcte opmaak van tabellen met `<thead>`, `<tbody>`, `<tfoot>`, `<caption>`, `<colgroup>` en de attributen `summary` en `abbr` zodat voor niet-visuele browsers tabellen duidelijker omschreven worden en beter navigeerbaar zijn.
- `Accesskey` en `tabindex` attributen om keyboardnavigatie te definiëren

Kortom, HTML zoals HTML bedoeld is. Je kan zo ver gaan als je zelf wilt:

- `<address>` voor adresgegevens;
- `` en `` in plaats van `` en `<i>`, zodat je semantische waarde geeft aan stukken tekst in plaats van visuele formattering eraan mee te geven;
- `` (verwijderde tekst - in de praktijk doorgestreept);
- `<ins>` (toegevoegde tekst);
- `<cite>` en `<blockquote>` voor citaties en `<q>` voor quotes;
- `<kbd>`, `<pre>` en `<code>` voor computercode en keyboardtoetsen;
- `<optgroup>` voor groepen opties in `<select>` tags

Gebruik van HTML zoals het bedoeld is maakt een document duidelijk voor de toepassingen die ze lezen; het mag duidelijk zijn dat het volgende stuk code...

```
<ul>
  <li>Eerste optie</li>
  <li>Tweede optie</li>
  <li>Derde optie</li>
</ul>
```

...een stuk beter te lezen is (voor zowel computerprogramma's als mensen) dan iets als:

```
<table width="100%" height="300" cellpadding="5" cellspacing="5">
<tr>
<td>&nbsp;</td><td>Eerste optie</td><td>&nbsp;</td>
</tr><tr>
<td>&nbsp;</td><td>Tweede optie</td><td>&nbsp;</td>
</tr><tr>
<td>&nbsp;</td><td>Derde optie</td><td>&nbsp;</td>
</tr>
</table>
```

Door HTML goed te gebruiken vergroot je de accessibility van een pagina ook. Door een lijst te gebruiken voor bijvoorbeeld navigatie zal een telefoon geen moeite hebben met je website. Wil je nog steeds plaatjes zien in het menu, dan kun je die plaatsen met CSS.

Om het goed gebruiken van HTML te bevorderen is het W3C een paar jaar geleden begonnen met het omvormen van HTML naar correct (well-formed) XML, wat resulteerde in XHTML (eXtensible Hypertext Markup Language). Doel hiervan was om de achterliggende code van een website schoner, strikter en beter gestandaardiseerd te maken. In XHTML zijn ten opzichte van HTML meer restricties opgelegd en een aantal tags verdwenen of niet meer ondersteund (zoals bekende opmaak tags als `` en `<u>`). XHTML wordt dan ook tegenwoordig meer en meer de standaard gezien voor accessible web content, doordat het meer ondersteuning biedt voor verschillende media, maar ook omdat het nu een vorm van XML is en daarom naadloos gekoppeld kan worden aan andere XML standaarden als Voice XML of XForms.

WAI-WCAG

De “Web Accessibility Initiative’s Web Content Accessibility Guidelines”, zijn richtlijnen die in 1999 zijn uitgegeven door het W3C om uit te leggen hoe websites accessible gemaakt kunnen worden voor mensen met een handicap. Handicaps die onder andere beschreven staan zijn:

- Zicht-, hoor- of beweegproblemen of problemen met het verwerken van (vormen van) informatie;
- Het niet in bezit zijn of niet kunnen gebruiken van invoerapparaten (toetsenbord, muis);
- Het gebruik van een tekstschermbord, een klein scherm of een trage internetverbinding;
- Het zich bevinden in een omgeving waar zintuigen beïnvloed worden of bezet zijn (bv: in een auto of een luide omgeving);
- Het gebruik van een niet-standaard browser, een voice browser, of een ander medium.

Door de guidelines te hanteren, afhankelijk van de mate van accessibility die je als developer wil hebben, zorg je ervoor dat je pagina’s onder andere:

Equivalenten bieden voor audiovisuele content;

Door middel van het gebruik van de `<object>` tag kun je ervoor zorgen dat browsers naadloos terugvallen op andere soorten media als de aangegeven media niet ondersteund wordt. Een voorbeeld hiervan is Macromedia Flash:

```
<object type="application/x-shockwave-flash" data="/menu.swf"
width="770" height="150">
  <param name="movie" value="/menu.swf" />
  <param name="wmode" value="transparent" />
  <param name="menu" value="false" />
  <!-- Alternative menu for no-flash browsers/media -->
  <a href="http://macromedia.com/"></a>
  <ul>
    <li><a href="/profiel/" accesskey="1" tabindex="1">Profiel</a></li>
    <li><a href="/cv/" accesskey="2" tabindex="2">CV</a></li>
    <li><a href="/pf/" accesskey="3" tabindex="3">Portfolio</a></li>
    <li><a href="/pop/" accesskey="4" tabindex="4">POP</a></li>
  </ul>
</object>
```

Hieruit blijkt dat Flash naadloos in een pagina getoond kan worden zonder niet-standaard HTML tags als `<embed>` te gebruiken. Bovendien kun je zorg dragen voor browsers die voor Flash geen ondersteuning bieden, maar die bijvoorbeeld wel `` tags (afbeeldingen) kunnen verwerken. Daarnaast kun je zelfs terugvallen op tekst.

Let bij het voorbeeld ook op de meegegeven accesskeys en tabindexen, die het mogelijk maken om met het toetsenbord deze links te selecteren. Accessibility binnen het Flash object is ook belangrijk, maar uiteraard een punt dat bij dit rapport buiten beschouwing wordt gelaten.

Leesbaar zijn zonder opmaak

Door het gebruik van CSS is het mogelijk om bij de structuur van een pagina (de HTML) volledig aandacht te besteden aan semantiek en geheel niet aan visuele vormgeving. Zonder CSS valt de browser dus terug op de logica van deze basisstructuur, met standaardvormgeving zoals die in de relevante HTML specificatie is aangegeven. Het gevolg is dat de informatie toonbaar is en door mens en computer leesbaar blijft, ook al ziet het er niet echt geweldig uit. Het doel hiervan is dat, mochten er uitzonderingen zijn (je kunt bij websites namelijk nooit zeker weten wat voor systeem een pagina aanspreekt), de informatie toch intact blijft, ook al valt de vormgeving weg.



Er is een grote hoeveelheid websites die hier niet aan voldoet. Zoals blijkt uit de website van de Hogeschool Rotterdam (rechtsboven) wordt de website gebrekkig leesbaar voor mens en machine door de CSS te verbergen. De tekst is niet leesbaar, afbeeldingen die onderdeel uitmaken van de vormgeving zijn nog zichtbaar, en (niet zichtbaar in de figuur) informatie als alt attributen is niet meegegeven.

Aan de andere kant zijn er steeds meer websites die geschreven zijn volgens de hierboven besproken manier. Een voorbeeld hiervan is de populaire Nederlandse IT-community Tweakers.net (rechts),



Duidelijk zijn in welke taal er gesproken wordt

Bij alle XML extensies is het mogelijk om een taal mee te geven aan een tag middels het attribuut `xml:lang`; daarnaast is het bij (X)HTML ook een optie om het attribuut `lang` aan te geven. Hierdoor bevestig je tegenover de browser dat de inhoud van de gespecificeerde tag van een bepaalde taal is. Daardoor hoeft de browser niet de inhoud van de tag zelf door te spitten en te analyseren welke taal het is, om vervolgens aan de gebruiker bepaalde opties weer te geven. Of, als het om een aurale browser gaat, kan die een bepaalde taal aannemen voor de sprekende stem.

Mediumonafhankelijk zijn

Mediumonafhankelijkheid is een belangrijk doel van het Internet en betekent in het kort dat informatie van waar dan ook aanspreekbaar moet zijn, zonder beperkingen. Uiteraard is dat ook een doel van web developers, dus aan de hand van een handig structureel ontwerp met XHTML en opmaak met CSS voor specifieke media, kan deze oplossing ook aangeboden worden. Een duidelijk voorbeeld is een menu of kaart van een restaurant: deze kan gepresenteerd worden aan standaard computerbrowsers in XHTML, waarbij een visuele stylesheet toegepast wordt om de huisstijl van het restaurant na te bootsen. Maar voor aurale browsers is het mogelijk om middels Voice XML en XSLT het menu interactief via stemfunctionaliteit te laten samenstellen. Kortom, alles wat je normaal via de telefoon kunt doen, moet ook met stem via het web kunnen.

Verder zijn er steeds meer toepassingen die op het internet kunnen, van mobiele telefoons tot game consoles of koelkasten. Deze hebben alle andere manieren van het aanspreken van webpagina's; mobiele telefoons kunnen over het algemeen veel minder informatie tegelijkertijd onthouden, terwijl game consoles veel meer gericht zijn op een multimedia ervaring.

Duidelijke navigatie hebben.

Door links zelfbeschrijvend te maken is het al gauw duidelijk waar ze heen zullen leiden. Dit zorgt voor een duidelijke navigatie. Ook het gebruik van afbeeldingen (pictogrammen) bevordert de duidelijkheid ervan, rekening houdend met het feit dat alleen een pictogram niet goed is (mocht de afbeelding niet getoond kunnen worden).

Richtlijnen van het W3C beschrijven nog meer toepassingsmogelijkheden: het groeperen van gerelateerde links, "next" en "previous" metalinks in het geval het een serie/reeks betreft en het implementeren van een globale zoekfunctie.

Web development en usability

Waarom usability?

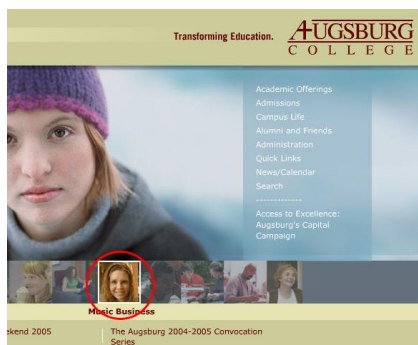
Naast dat een website accessible moet zijn voor zoveel mogelijk bezoekers binnen de doelgroep, moet er ook rekening gehouden worden met de ervaring die de bezoeker krijgt bij het gebruiken van de site. Usability is hier het kernwoord; gebruikersgemak bedraagt een groot deel van de totale ervaring. Usability en accessibility kunnen gezien worden als broer en zus: verwant aan elkaar, verschillend van aard, maar horen beide bij dezelfde ‘familie’. Het is daarom niet zomaar uit te vlakken, een site maak je immers voor je bezoekers en niet voor jezelf.

Usability in websites

Wat houdt usability in het geval van websites in? Alles. Van navigatie tot complete user interface, van formulieren tot clickstream (het aantal klikken dat de gebruiker verwijderd is van een pagina): usability komt voor door de hele website. Daarom is het belangrijk om tijdens het ontwerpen al in gedachten te houden wat handig/nuttig kan zijn voor de gebruiker, enkele voorbeelden:

Site navigatie

Zoals eerder gezegd komt een bezoeker doorgaans voor 1 ding op een site: informatie. Die informatie moet snel bemachtigd zijn, zonder veel omwegen. Ontwerp dus een duidelijke navigatie en laat duidelijk merken dat het navigatie is. Houd er ook rekening mee dat de navigatie simpel blijft: 4 geneste lagen zijn er eigenlijk al 2 teveel.



De Augsburg College website (links; http://www.augsburg.edu/main_nointro.html) laat niet duidelijk blijken dat er sprake is van navigatie. Er wordt een reeks foto's getoond waarvan je nooit zou verwachten dat deze kenmerkend zijn van bepaalde vakken die je op een school kunt volgen.



Daarentegen zijn de Amazon navigatietabs bijna synoniem voor duidelijke navigatie geworden: het is voor vrijwel iedereen duidelijk dat tabs navigatie aanduiden, omdat het doet denken aan bladwijzers.

User Interface

Dit is de alpha en omega; een site staat of valt bij user interface. Ga eens na hoe snel je beslist of je blijft op een site wanneer je zoekt naar iets, het is bevonden dat dit ongeveer 4 seconden is. Binnen deze seconden wil de gebruiker namelijk een aantal antwoorden: “Waar ben ik? Wat is dit? Wat zoek ik ook alweer? Heb ik vandaag wel een schone -- “ en weg!



Mozilla.org - duidelijke user interface

Denk na wat de verwachtingen bij gebruikers zijn; mensen verwachten een logo links bovenaan, dat klikbaar is en naar de startpagina leidt, in het eerste oogopslag moet de zoekfunctie duidelijk aanwezig zijn en zonder veel moeite gebruikt kunnen worden, ergens bovenaan de pagina ‘hoort’ een titel die duidelijk de inhoud van die pagina omschrijft, etc.

Formulieren

Formulieren kunnen net zo makkelijk in gebruik zijn, als dat ze gebruikers de haren uit het hoofd kunnen laten trekken. Hoe zijn ze bijvoorbeeld in te vullen, intuïtief of met een handleiding? Moet je je door 20 velden heen worstelen, op 7 verschillende pagina’s (voor het gemak vaak ‘stappen’ genoemd) of ben je in een paar seconden klaar? Is elk lullig veld verplicht en zo ja, moet je alles opnieuw invullen wanneer je een veld gemist hebt of krijg je direct netjes een melding?

Een formulier lijkt een vrij standaard iets: een paar velden, knop voor versturen (eventueel resetten) en klaar. Maar formulieren kunnen een grote bron van ergernis zijn, vooral omdat het elementen zijn waar je vaak niet omheen kan.

Zoekfunctie

Elke site met minstens een paar pagina’s zou een zoekfunctie moeten hebben. Er is niets vervelender dan je door pagina’s heen te worstelen, op zoek naar een paar woordjes van herkenning die wellicht, misschien, eventueel wel eens zouden kunnen zijn wat je zoekt. Een zoekfunctie op een globale plek op de site (eventueel ook een uitgebreide zoekfunctie) is dus erg gebruiksvriendelijk. Rest natuurlijk te zeggen dat de functie zelf makkelijk in gebruik moet zijn.



Ebay.com - duidelijke zoekfunctie

Ergernissen

Iets opgevallen in de vorige paragraaf?

“Ja, je bent een zeikerd.”

Als je in een winkel staat, verwacht je toch ook service? Niemand wil winkel na winkel afstruinen, overal moeten vragen waar iets staat (tenzij die dame achter de kassa zo leuk is) en uiteindelijk een uur in de rij staan om af te rekenen, toch? Waarom zou je dat bij een website wel doen dan?

Er is een ding wat al deze voorbeelden gemeen hebben: ergernissen. En dat is eigenlijk gelijk de kern van het hele verhaal over usability: zorg dat je je bezoekers zo min mogelijk irriteert. Een aantal bekende ergernissen op websites:

Onduidelijkheid

Onduidelijkheid over de positie in de site, de bedoeling van de site, bepaalde elementen in de site; allemaal reden genoeg om weg te gaan.

Verwarring / Onverwachte dingen

Pop-ups, banners die met javascript ineens onder je neus geplaatst worden, functionaliteit die asymmetrisch is met functionaliteit van het besturingssysteem; Windows gebruikers verwachten in een bevestigingsprompt 2 knoppen: “ok” en “cancel”, in die volgorde, van links naar rechts. Stel dat een website die knoppen omdraait en een gebruiker klikt “ok” in plaats van “cancel”, enig idee hoe frustrerend? Vast wel, we hebben het allemaal wel eens meegemaakt.

Traagheid

Traag ladende sites doen het niet goed op het internet; denk even terug aan die 4 seconden, als een site er al 11 seconden over doet om te laden had de gebruiker al 2,75 sites kunnen zien. Het is niet moeilijk om na te gaan wat efficiënter is.

Afhankelijkheid

Leuk, sites die gebruik maken van de nieuwste technieken, eerst een enorme hoeveelheid data moeten laden en hé, wacht eens even, deze plugins zijn ook nog nodig! Niemand houdt ervan om (voor hem) onbekende dingen te installeren, zeker met de hoeveelheid spyware, virussen en andere troep die rond zwerft tegenwoordig. Hou dat uit websites, tenzij het echt nodig is, of een duidelijke toegevoegde waarde heeft.

De weg naar een vriendelijker internet

Hielden alle websites maar rekening met dit soort dingen; helaas is dat niet zo. Vandaar dat usability veel aandacht verdient: een gebruiker die een prettige ervaring krijgt bij het bezoeken van een site, zal eerder geneigd zijn er terug te komen.

Er zijn slechts een paar voorbeelden besproken, want er zijn natuurlijk nog veel meer situaties denkbaar. Ze hebben echter allemaal een ding gemeen: gezond verstand. Een beetje logisch nadenken vanuit het perspectief van je bezoekers levert al een hoop verbeteringen op. En mocht je er echt zeker van willen zijn, dan kan je natuurlijk altijd testen met gebruikers. Hoe meer hoe beter!

Accessibility, usability en Grafimatechnologie

De opleiding Grafimatechnologie besteedt tegenwoordig veel aandacht aan usability. Er is zelfs een differentiatiemogelijkheid op dit gebied, dat zich onder andere manifesteert in uitvoerig usability testing. Een goede ontwikkeling, toch zien wij verbeteringen op dit punt. De opleiding zou meer aandacht kunnen besteden aan praktische vaardigheden voor het verbeteren van de usability van producten (waaronder websites). Het is vooralsnog veel theoretisch, en dat terwijl de opleiding zelf erg praktijkgericht van aard is.

De werkzaamheden op het gebied van usability zijn vooral in de post-productiefase gepland; dat wil zeggen dat er veel achteraf wordt gekeken wat er wel en niet goed is. Een gezonde doorontwikkeling zou zijn om de manier van werken proberen aan te passen hierop: zorg dat ontwikkelaars bewust usabilityaspecten meenemen in hun ontwerpen/producten. Dit zou in principe zo vroeg mogelijk moeten gebeuren, bij voorkeur in de onderbouw.

Op het gebied van accessibility zou de opleiding er goed aan doen om een hoop van de in dit document besproken theorie over te brengen op studenten. Simpele vragen als wat accessibility is en hoe dit toegepast kan worden leveren al vaak moeilijke blikken op, laat staan het eerder besproken vierlagenmodel. Door de student eerst nieuwsgierig te maken naar het onderwerp - web development is immers een populair gebied bij deze studenten - kan een vervolgmodule in de vorm van praktische vaardigheden genoeg voet aan de grond krijgen.

Voorbeelden van opdrachten zouden kunnen zijn: het opsporen van usability/accessibilityproblemen bij een vooraf gegeven website en het verbeteren hiervan en onderzoek uitvoeren naar huidige ontwikkelingen en technieken op dit gebied (eventueel webstandaarden). Op deze manier blijven studenten ook op de hoogte van nieuwe ontwikkelingen op een gebied wat toch al voortdurend in beweging is.

Om ook maar met een doodoener af te sluiten: er kan niet genoeg gezegd worden over accessibility en usability. Bent u na het lezen van dit document daar ook van mening van, dan raden wij aan om sites uit de literatuurlijst door te lezen, een simpele zoekopdracht op Google levert overigens ook genoeg leesvoer op.

Literatuurlijst

<http://www.netmechanic.com/accessibility/glossary.htm>

Een aantal duidelijke definities omtrent accessibility.

<http://www.projectauditors.com/Dictionary/U.html>

Een aantal duidelijke definities omtrent usability.

<http://www.webstandards.org>

De website van de Webstandards Campaign (WaSP), een door het W3C opgezette organisatie die wereldwijd het gebruik van web standards als XHTML en CSS moet promoten.

<http://webstandards.org/act/campaign/dwtf/>

De Dreamweaver Task Force van WaSP, die samen met Macromedia werkt aan het meer toegankelijk maken van Macromedia producten (als Dreamweaver).

<http://www.accessibility.nl/internet/>

Het internet onderdeel van de Stichting Bartiméus Accessibility website. Bevat veel informatie over toegankelijkheid van websites in het Nederlands, en hoe dat van toepassing is binnen Nederland.

<http://www.iso.ch/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=30858&ICS1=13>

De ISO website voor ISO standaard 16071 voor "ergonomics of human-system interaction - guidance on accessibility for human-computer interfaces".

<http://www.section508.gov/>

De Amerikaanse website voor Section 508, een wet die het verplicht maakt om voor overheidssites een dergelijk niveau van accessibility te hanteren.

<http://www.w3.org/MarkUp/Forms/>

De W3C website voor formulieren, met name de nieuwe standaard XForms.

<http://www.w3.org/2001/sw/>

De W3C website voor The Semantic Web, dat uitleg geeft over het doel van het World Wide Web, met name de correcte interpretatie van semantische documenten.

<http://www.w3.org/Voice/>

De W3C website voor stemgerelateerde standaarden als Voice XML.

<http://www.w3.org/TR/WCAG10/>

De W3C website voor de Web Content Accessibility Guidelines, zoals eerder besproken in dit rapport.

<http://www.webpagethatsuck.com/>

Een interessante website die slechte navigatie, user interfaces, e.d. deconstrueert en bekritiseert aan de hand van usability en accessibility argumenten.

<http://validator.w3.org/>

Website van W3C voor het valideren van (X)HTML code op syntax fouten, ontbrekende attributen e.d.

<http://jigsaw.w3.org/css-validator/>

Website van W3C voor het valideren van CSS code op syntax fouten, ontbrekende selectors e.d.

<http://pinkgreetings.com/peetr/portfolio/#documentatie>

Opgestelde checklist voor het controleren van verschillende usability aspecten op een website. Houdt rekening met de ISO 9126 norm.