

Opinnäytetyö (AMK)

Tietojenkäsittely

Tietoliikenne

2012

Otso Lahti

KÄYTTÖKOKEMUS JA WEB-SUUNNITTELU

– CASE: Zeeland Turku Oy:n esittelysivu



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

Otso Lahti

KÄYTTÖKOKEMUS JA WEB-SUUNNITTELU

Opinnäytetyössä pyritään selvittämään, mitä käyttökokemus tarkoittaa ja löytämään web-suunnittelulle ominaisia käyttökokemusratkaisuja sekä käydään läpi havainnollisten esimerkkien avulla, mistä käyttökokemus koostuu. Yksi web-suunnittelun ajattelua muuttava kehitysaskel tulevaisuudessa ja osittain jo vuonna 2012 on selauslaitteiden osuuksien jakautuminen tasaisemmin tietokoneiden, mobiili- ja viihdelaitteiden kesken. Sen lisäksi opinnäytetyö käsittelee suunnittelua ammattina ja millaista se käytännössä on.

Tutkimuksen rinnalla suunniteltiin uutta esittelysivua mainostoimisto Zeeland Turku Oy:lle. Zeeland Turku on osa suurempaa Zeeland Oyj -konsernia. Toimeksiantona oli viestiä Turun toimiston henkeä ja tietoa henkilöstöstä. Esittelysivu on yksittäinen pystysuunnassa vieritettävä kokonaisuus, josta puuttuu tarkoituksella alisivut. Suunnitelma sisälsi teknisen puolen lisäksi luovempaa sisällön suunnittelua.

Esittelysivu suunniteltiin toimimaan laitteesta riippumatta ja mahdollisimman hyvin myös puutteellisella modernin web-teknologian tuella. Rakenne on niin sanotusti responsiivinen ja ottaa siten huomioon sekä kapean että leveänkin selainäkymän. Sivun toteutus on toteutettu WordPress-julkaisujärjestelmän päälle. WordPress-teemakehitys perustuu standardiin HTML:n ja CSS:n yhdistelmään, mutta sivupohjatiedostot sisältävät myös PHP-kieltä.

Käyttökokemusta rikastettiin asteittain lisäämällä lopuksi sivua esitysmäisemmäksi muuttavan javascript-kerroksen ja CSS3-animointia. Sivun toimivuus ja pääsy sisältöön eivät kuitenkaan vaadi selaimelta tukea kummallekaan tekniikalle.

Opinnäytetyön tuloksena syntyi teknisesti valmis esittelysivu odottamaan sisällöntuottoa suunnitelman mukaan. Ihannetapauksessa kehitys jatkuu tulevaisuudessakin, kun sisältö ja käyttö nostaa esiin potentiaalisia ongelmia tai parannusideoita. Tutkimus paljasti web-suunnittelun muuttuvan entistä haasteellisemmaksi, kun laiteriippumattomuuden filosofia otetaan käyttöön laajemmin ja syvemmillä web-projekteissa. Responsiivisen suunnittelun toimintamallit ja teknisen toteutuksen standardisointi ei ole vielä valmis, joten siirtymävaihe vaatii paljon lisätutkimusta ja -työtä.

ASIASANAT:

käyttökokemus, suunnittelu, web-suunnittelu, responsiivinen suunnittelu, layout, HTML, CSS, PHP, sisällönhallinta

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Business Information Technology | Data Communications

2012 | Total number of pages 60

Instructor Esko Vainikka

Otso Lahti

USER EXPERIENCE AND WEB DESIGN

A recent advancement is reshaping the way we are going to approach web design in the future, and partly today too. It's the rising share of web usage among mobile and other media devices. This thesis tries to find out about user experience in web design and discover what problems lie in designing for any imaginable web capable device.

The material used for designing a new introductory site for Zeeland Turku consisted of the author's observations and studies. The company is a part of a bigger group called Zeeland. They needed a site that communicates the spirit of their local office and introduced their staff. The site consists of a long single page that's meant to be scrolled rather than navigated from subpage to subpage. The design work contained both visual and technical design concerning the implementation phase.

The site was designed to work well on any device and be backwards compatible in case the visitor's browser doesn't support modern web technologies. The structure of the site is responsive and thus takes into account even the narrowest of viewports and grows fluidly laying out elements on the page to suit different widths.

The site is implemented on a content management system called WordPress using Themble's custom theme "Bones". WordPress theme development is basic HTML markup and CSS style attributes, but it also includes a lot of PHP code in the theme template files. The purpose of WordPress is to make regular content updates easy with just a couple of quick lessons.

The site enhances progressively by adding CSS3 animations and javascript in the last stage. An own javascript file was developed to improve the site by making it feel more like a presentation. Because this was made honoring the philosophy of progressive enhancement, support for the modern web technology used in the last stage is not required for the site to work and for content to be visible.

Finalizing the technical phase was successful and it's ready for content to be produced according to the plan made during visual design. Ideally the development of the site is going to continue in the future when problems and new ideas come into play through production and actual use.

KEYWORDS:

user experience, design, web design, responsive web design, layout, HTML, CSS, PHP, content management

SISÄLTÖ

1 JOHDANTO	6
2 KÄYTTÖKOKEMUS	8
2.1 Päätoiminnot	10
2.2 Web-suunnittelun suuri käyttökokemusongelma	15
2.3 Web-suunnittelun haasteet	17
2.4 Lisää käyttökokemusesimerkkejä	20
3 SUUNNITTELU	24
3.1 Lähtötilanne	25
3.2 Tavoitteet	27
3.3 Toteutuslusta	29
3.4 Visuaalinen suunnitelma	33
4 TOTEUTUS	38
4.1 Toteutuslustan valmistelu	43
4.2 Perusrakenne ja tyylimääritys	44
4.2.1 Esiprosessoitu CSS	45
4.2.2 Sivupohjan rakenne	49
4.3 Animointi, skriptaus ja efektit	51
5 YHTEENVETO	56
LÄHTEET	58

KUVAT

Kuva 1. Iltalehden ahdas rakenne ja tunkeileva mainonta (Iltalehti 27.10.2012).	13
Kuva 2. Daring Fireball -blogin etusivu (Daring Fireball 2012).	14
Kuva 3. edenspiekermann.com (Eden Spiekermann 2012).	22
Kuva 4. "Living with Nest" -alasivu Nestin sivustolla (Nest 2012).	24
Kuva 5. YLE Uutiset -sivuston rakenne (YLE 2012).	34
Kuva 6. Ote rautalangasta.	36
Kuva 7. Flash-liitännäisen käytön osuuden lasku (W3Techs 2012).	38
Kuva 8. Artikkelin sisällön linkin valitsin.	45
Kuva 9. Toistuvat valitsimet.	46
Kuva 10. Sisäkkäiset valitsimet.	46

Kuva 11. Pyöreät kulmat määrittelevä mixin.	47
Kuva 12. Tiivis tapa määritellä pyöreät reunat.	47
Kuva 13. LESS mahdollistaa matemaattiset operaatiot.	48
Kuva 14. Tyylimääritelmän arvon laskeminen funktion avulla.	48
Kuva 15. CSS3-siirtymän määrittely.	53
Kuva 16. Sivun osioiden vähimmäiskorkeuden asetus.	54

1 JOHDANTO

Käyttökokemus on mielenkiintoisen määrittelemätön käsite. Se selittää itsensä hyvin, mutta on vaikeampi jakaa osiin. Erityisen hankalaa on arvostella käyttökokemuksen laatua. Internet tekee tästä yhä hankalampaa, koska lopullinen tuote riippuu käyttäjän selaimesta ja laitteesta. Kehityslaitteella mielenkiintoiset ratkaisut saattavat toimia huonosti tai hajota täysin suurella osalla käyttäjäkunnasta, jolloin käyttökokemus onkin yhtäkkiä huonompi.

Hyvä tasapaino mielenkiintoisen toteutuksen, selkeän ulkoasun ja laiteriippumattomuuden välillä on todella hankala saavuttaa. Parhaimmillaan hyvän web-suunnittelun lopputulos on juuri niin toimiva, ettei siihen kiinnitä huomiota. Suunnittelijana täytyy pitää katse tavoitteessa eikä reitissä siihen. Toteutustekniikat ja efektit eivät ole itseisarvoja, vaan sisältö ja sen muotoilu on tärkeintä. Ensimmäisilläkin on hetkensä, mutta sellaisen löytäminen vaatii kehittyntä arviointikykyä.

Asiakkaalle suunnittelu vaatii ylimääräisen kerroksen lopputuloksen ja suunnitelman väliin. Oikeat päätökset täytyy hyväksyttää asiakkaalla ja monesti tämä tarkoittaa myös tehdyn työn esittelyä. Käyttökokemuksen esittely on vaikeaa, jos se on vain visio päässä. Staattiset visuaaliset suunnitelmat helpottavat asiakasta ymmärtämään miltä sivu tulee näyttämään, mutta miten esitellä animaatioita ja monimutkaisia interaktiivisia tapahtumasarjoja?

Parhaankin suunnitelman edessä täytyy vielä varmistaa, että sen toteuttaminen on realistista. Web-teknologiat asettavat yhä vuonna 2012 yllättäviä rajoja. Tekstin tavutustuki ei ole yleistynyt selaimissa, ei ainakaan suomeksi. Responsiivinen rakenne tekee tarkasta elementtien sijoittelusta turhaa taistelua ja sisällön esittäminen älypuhelimien näytöllä tarkoittaa ylimääräisten elementtien karsimista.

Kuuluista kanadalainen entinen jääkiekon pelaaja Wayne Gretzky on sanonut: "I skate to where the puck is going to be, not where it has been". (Gretzky 2012)

Sitaatin voi kääntää siten, että Gretzky luistelee sinne, mihin kiekko on menossa eikä sinne, missä se oli ennen. Opinnäytetyöni yhtenä tavoitteena on ollut tutkia, miten tämä on mahdollista web-suunnittelussa. Kun suunnittelemme uutta, voimmeko tietää millainen internet on kahden vuoden päästä. Vai onko parempi suunnitella mukautuvasti ilman voimakkaita olettamuksia tulevaisuudesta ja siten hyvän käyttökokemuksen vaatimista olosuhteista?

Tein tutkielmani rinnalla projektityönä esittelysivua mainostoimisto Zeeland Turku Oy:lle. Olen ottanut sen huomioon kohdatessani käyttökokemusesimerkkejä ja kerron sen suunnittelusta lähtötilanteesta visuaalisen suunnitteluun. Lisäksi syvennyn tarkemmin sivun tekniseen toteutukseen esitellen joitain sekä sisälönhallinnan että rikkaan HTML5:n perusasioita. Havainnollistan myös esiprosessoidun CSS:n tarkoitusta vertaamalla tavanomaista CSS-määrittelyä esiprosessoitavaan LESS-kieleen.

2 KÄYTTÖKOKEMUS

Käyttökokemukselle ei ole olemassa yhtä selkeää virallista määritelmää. All about UX –sivusto on koonnut eri henkilöiden ja yhteisöjen omia määritelmiä yhteen. Jos yrittäisi yhdistää useamman eri käyttökokemuksen määritelmän, yksinkertaisimmillaan se tarkoittaa sitä, mitä käyttäjä tuntee käyttäessään tuotetta, palvelua tai järjestelmää. (All about UX 2012.) Käyttökokemus on määritelmien kirjosta päätellen todella laaja termi ja yhdistää myös kaikki erikseen arvioitavat osat, kuten toimivuuden, visuaalisen miellyttävyyden ja käytettävyyden. Näistäkin osa voi vaikuttaa toisiinsa subjektiivisesti. Olen itse havainnut, että käytön helppous saattaa johtua käyttöliittymän ulkonäöstäkin eikä pelkästään hyvästä käytettävyydestä. Toisaalta voimme mieltää helposti luettavat asiat myös visuaalisesti miellyttäviksi. Hyvin toteutetut, yhdessä toimivat osa-alueet luovat positiivisen käyttökokemuksen.

Jokainen käyttäjän aktiivisuutta vaativa asia voidaan arvioida käytettävyytensä suhteen. Käytettävyyys ja käyttökokemus eivät rajoitu pelkästään tietokoneohjelmiin, vaan useammankin määritelmän perusteella mistä tahansa voi jäädä huonouden tunne keskeneräisen käytettävyyden takia. Luit sitten sanomalehteä tai laitoit uutta tiskikonettasi päälle ensimmäistä kertaa, jonkun on täytynyt pohdita käyttäjää ja tehdä hankalia päätöksiä sen pohjalta, mitä hän luulee käyttäjän tekevän.

Perinteinen ongelma sanomalehdissä on se, että suuret lakanamalliset julkaisut, kuten Turun Sanomat, varaavat isommankin keittiön pöydän, jolloin sellaisen lukeminen erityisesti muidenkin ihmisten käyttämässä pöydässä on hankalaa. Suuren, niin kutsutun lakanamallisen sanomalehden formaatin valitsemiselle on toki syynsä: ison artikkelin taittaminen on helpompaa, kuvat voivat olla isompia ja täyden sivun mainoksiin mahtuu enemmän informaatiota tai tarjous tuotteita. Helsingin Sanomat on jo kuitenkin päättänyt siirtyä pois lakanasta ja ottaa käyttöön iltapäivälehdistä tutun tabloid-formaatin (Kauppalehti 1.8.2012).

Helsingin sanomilla on käytössä myös iPad-sovellus, joka on lakanaformaatin vastakohta. Siitä on etua myös tabloid-formaattiin nähden. Pieni kosketusnäytöllinen laite mahtuu yhä paremmin pienen yksion aamiaispöytään ja sitä kantaa helpommin mukanaan junassa puhumattakaan muista digitaalisuuden hyödyistä. Saat tuoreimman tilausnumerosi vaikket olisikaan kotona noutamassa postia, eikä vanhojen numeroiden säilöminen vie ollenkaan lisätilaa. Myös kierrätettävän paperin määrä vähenee ainakin teoriassa. Digitaalinen lehti mahdollistaa tietenkin myös videoiden käytön, interaktiiviset mainokset ja fonttikoon muuttamisen lukijan näön tai mieltymyksen mukaan.

Mietitäänpä hetki hyvin tuotettua kovakantista romaania kaupan hyllyllä. Varsinkin nykyään tätä ei voi korostaa tarpeeksi, sillä e-kirja on vakava kilpailija paperiversiolle. Näin voi päätellä Ison-Britannian tilastoista, joiden mukaan Amazon UK myy enemmän sähköisiä kuin painettuja kirjoja (BBC 2012). Hyllyllä tutkii kirjan kansikuvitusta, mikä on tietenkin toissijaista lukukokemuksen kannalta, mutta se luo mielikuvia sisällöstä. Avattaessa kirjan tuntee paperilaadun käsissä, ja lukijalla on toivon mukaan hyvin varaa pitää peukaloja marginaaleissa. Tämän lisäksi marginaalit antavat myös tekstille tilaa hengittää, mikä vähentää sivujen ahdistavuutta. Kyse ei ole niinkään visuaalisesta rikkaudesta vaan sen huomaamattomuudesta. Tämä on ristiriitaista, koska usein visuaalinen suunnittelu yhdistetään kuitenkin jonkinlaiseen luovaan tyyliin vaikka se ei sitä usein olekaan.

Kirjan suunnittelulle kenties tärkein asia on hyvä typografia, ja se on hyvä esimerkki työstä, jota tehdään siksi, ettei kenenkään tarvitsisi enää koskaan ajatella asiaa käyttäessään – tai tässä tapauksessa lukiessaan – tuotettasi. Yleensä rivit ovat vain sen verran harvoja, että niiden välit toimivat ikään kuin viivaimina ohjaten helposti rivin vasemmalta oikealle. Kirjasinkoko on valittu siten, että suurimman osan olisi helppo nähdä lukea tekstiä, mutta pitäen tekstin myös mahdollisimman tiheänä.

Sopivan paperilaadun ja musteen yhdistelmä luo juuri tietynlaiset reunat kirjaimiin, jolloin ne eivät ole liian suttuisia eivätkä myöskään liian kontrastisia. Fontti ja itse tekstikin häviää, eikä lukija enää keskity kirjaan vaan seuraa sisältöä ajat-

telematta asiaa. On tärkeää ymmärtää, ettei tämä arkielämän itsestäänselvyys olekaan itsestäänselvyys.

Yleensä edellä mainituista asioista joudutaan tinkimään, jos painetaan pitkiä teoksia. Täytyy kuitenkin ottaa huomioon myös, miten paksu ja painava kirjasta tulee, koska liian ison kirjan pitelevinen on hankalampaa kuin liian tiheän tekstin lukeminen.

Mutta hyvässä kirjassa parasta on tietenkin itse sisältö. Parhaalle sisällölle on silti loogista antaa myös paras mahdollinen muoto, jotta mikään ei pilaisi lukijan kokemusta.

2.1 Päätoiminnot

Tiskikone on kodinkoneena paljon käytetty, mutta sitä käytetään yleensä vain yhdellä tavalla ja sen käynnistäminen on monelle tuttu toistuva rituaali. Yleisimmin pesujauhesäiliöön kaadetaan oikeaan viivaan asti pesujauhetta, laitetaan säiliön kansi kiinni, suljetaan tiskikone, avataan sulkuhana ja asetetaan tiskikone päälle haluttuun ohjelmaan. Toistuvana prosessina tuossa on aika monta vaihetta ja niistäkin voisi teoriassa automatisoida osan.

Voisimme miettiä, minkä takia käynnistäminen sitten on niinkin monimutkainen prosessi ja yrittää yksinkertaistaa sitä ajattelemalla mitkä ovat pakollisia valintoja. Ohjelmavalinta on tietysti olemassa siksi, että voisit valita jonkin kymmenestä eri pesuohjelmasta ja käyttöliittymillä on luonnollisesti tapana monimutkaistaa vaihtoehtojen määrän kasvaessa.

Jotkut tiskikoneet osaavat peräti valita ohjelman pituuden mittaamalla tasojen painot. Minun mielestä tämä on todellista älyä, sillä alunperin koko ohjelman valinta jätetään käyttäjälle vain siksi, että kodinkone ei tiedä tarpeeksi kyseisen pesukerran muuttujista. Toisaalta esimerkiksi sulkuhanaa ei kannata automatisoida, koska se on turvallisuusominaisuus, jonka huolellista käyttöä jotkut vakuutusyhtiötkin pitävät ehtona tiskikoneesta aiheutuneiden vesivahinkojen korvaamiselle (Tapiola 2012). Yleisesti kaikkia toimintoja voi suunnitella vastaavas-

ti käänteisesti hankalammaksi. Esimerkiksi virtanapin voi upottaa, jottei siihen osu liian helposti tai caps lock -näppäin voi vaatia tarkoituksella virheellisen tuntuaisesti aavistusta pidemmän painalluksen, jottei se menisi vahingossa päälle (Apple 2012a).

Vaikka käytettävyyssuopissa edellä mainittuja ylimääräisiä vaiheita ei olisikaan, kaikkien tosimaailman asettamien rajojen sisällä ohjelman itsestään valitseva tiskikone on kuitenkin hyvin automatisoitu. Se säästää itsessään jo niin paljon ihmisten aikaa, että sen käynnistämisen monimutkaisuus on toissijaista. Tiskikoneen päätoiminto ja lupaus on astioiden puhdistaminen ihmisen puolesta. Miten se saavutetaan, on toissijaista.

Eräs toinen arjen mukavuuslaite on saavuttanut kenties yksinkertaisimman muotonsa. Nest on aidosti älykäs termostaatti, joka oppii asunnon aikataulut nopeasti vain seuraamalla lämpötilan säätöjä. Sitä ei kuulu ohjelmoida vaan se tekee jo viikon sisällä juuri niin kuin pitääkin. Se oppii siihen, ettei kukaan kaippaa lämpöä tai jäähdystystä työaikana eli kun kukaan ei ole säätämässä sitä, eikä tällöin tee mitään sisälämpötilalle säästären sähköä. Toisaalta se osaa myös ennakoida ja jäähdyttää asunnon valmiiksi, kun ensimmäinen asukas palaa kotiin. (Nest 2012.) On paljon tarkoituksenmukaisempaa, että lämpötila vain on melkein aina oikein kuin se, että saat ohjelmoida kiinteän aikataulun termostaatille.

Nestin käyttöliittymä on yksikössä oleva lämpötilansäätö ja värinäyttö, joka kertoo lämpötilan ja sen, lämmittääkö vai jäähdyttääkö järjestelmä sillä hetkellä. Yksikköä voi ohjata myös kirjautumalla siihen Nestin nettisivuilta tai iOS- ja Android-laitteisiin saatavalla sovelluksella. Etäkäyttöliittymä on tarkoitettu lähinnä poikkeustilanteisiin, kuten siihen, jos huomaat päivän olevan erityisen kylmä ja haluat lämmittää asuntoa tavallista lämpimämmäksi. Automaatiikan on hankala korvata ihmisen suhteellisuutta, eli vaikka sitä miten kylmänä päivänä jotkut haluavat takkatulen ääreen, vaikka asunto on yhtä lämmin kuin muulloinkin.

Älytermostaatti toimii mielestäni siksi, että se ratkaisee ihmisten jatkuvan mukavuusongelman lämpötilojen kanssa täysin samalla yksinkertaisuudella kuin kä-

sikäyttöinenkin termostaatti, eli käyttäjä ei joudu opettelemaan mitään uutta tai muuttamaan lähestymistään laitteeseen. Samalla se toimii myös kuin ajatus, koska käyttäjän mielikuva lähtiessään liian lämpimästä toimistosta, ulkoilmasta tai koulusta on mukavan viileä koti. Aluksi tämä saadaan aikaiseksi säätämällä termostaattia normaalisti, mutta muutaman päivän päästä käyttäjä saapuukin jo valmiiksi viileään kotiin. Samalla termostaatti on myös säästänyt sähköä sillä välin, kun kukaan ei ole paikalla. Käytettävyys on ottanut huomioon perimmäisen tarpeen ja auttaa vain ratkaisemaan ydinongelman, eikä kiertämään sitä ylimääräisellä kontrollilla.

Nettisivuja mietittäessä ei aina tule ajatelleeksi, että niissä on mukana käytettävyyttä, mutta tämä on toisaalta vain hyvä asia. Se kertoo siitä, että nettisivuja tekevät puhaltavat yhteen hiileen ja toistavat samoja käytettävyyskuvioita, mikä vuoksi netin selaaminen on suurimmaksi osaksi helppoa. Yleensä ainoa varsinainen käyttöliittymä nettisivuissa on navigaatio, mutta koska se on ainoa, ei ole mitään järkevää syytä jättää sitä keskinkertaiseksi.

Jos heijastellaan kirjaesimerkkiä, on netissä paljonkin luettavuusongelmia ja ahtaita marginaaleja. Nämä korostuvat mielestäni erityisen pahasti, jos olet ensin vierailut hyvin suunnitellulla sivulla ja silloin, kun sivun tarkoitus on pitkien artikkeleiden lukeminen. Esimerkiksi blogeja on paljon erilaisia ja niissä saattaa usein olla mainoksia sivun reunoilla tai peräti kappaleiden välissä. Niistä ei ole kuin harmia lukukokemukselle, pidit mainoksista tai et. Iltalehden verkkojulkaisu on hyvä esimerkki tällaisesta huonosta hengittävyydestä (kuva 1). Siinä yhdistyy mainosten aggressiivista sijoittelua ja pieniä tekstimarginaaleja.

Seiso muuttu
suhtautumisen: Yhä
useampi ulkomaalaisen
rikollinen yritetään
karkottaa 12:45

Lisää juttuja

ULKOMAAT

Vankilaan tuomittu
Berlusconi ei jätäkaan
politiikkaa 16:16

Lumituisku katkoi
sähköt tuhansilta
Virossa 15:25

Obama ja Romney
huippujuristi 10:03

Hurrikaani Sandy
lähestyy Yhdysvaltoja:
"Hyvin, hyvin iso
myrsky" 09:29

NY Times: Rahapula
vaarantaa USA:n
myrskynuhteet -
vuoden aukko uhkaa
08:39

Noottoriipärikerhon
tiloissa ammuskeltiin,
puukotettiin ja lyötiin
kirveellä Ruotsissa
08:19

Hirmumyrsky Sandy
tappanut lähes 40 -
katso kuvat tuhoista
07:39

Mitä juoksemista! Tästä
näyttää USA:n
presidentin viikko 05:20

Lisää juttuja

VIINDE

River Vilperistä tuli
Hatuntelija - katso
kuvat! 17:25

Lasse Pöysti päätti
uransa - Itku tuli 17:19

Riku Korhonen Sylville:
Olen ollut sängyssä 180
naisen kanssa 17:08

Ennakkosuosikki tuuppi
ja nimitteli törkeästi -
BB teki ylittävän
päätöksen 16:41

Tunnistatko tämän
pelottavan tähtiparin -
katso kuvat! 14:15

Idols-kakkonen jäi
ilman
levytysovimusta:
"Miksi mitään ei
tapaudu?" 13:12

BB-talossa raju riita -
tässä BB:n jatkotoimet
13:05

IL tapasi Bond-tytön:
"Suutelu Craigin kanssa
oli rankkaa" 12:08

Lisää juttuja

URHEILU

1. erät: Kärppien NHL-
vahvistus väkevä
17:57

Stoner ylitti yleisä -
kaatui ja jyräsi paalulle
17:30

Norwich tuotti lisää
harmaita hiuksia Aston
Villa -luotsille 17:28

Lasse Pöysti päätti uransa ITKU TULI



Lasse Pöysti, 85, päätti uransa täyden, seisoen taputtavan katsomon edessä. **Lue lisää...**

ILMOITUS

Katso sinkkuja lähialueeltasi!



Lity **e-kontakti.fi** jäseneksi ja löydä juuri sinulle sopiva kumppani.

Älä odota, vaan tartu tilaisuuteen. **Aloita sen oikean etsiminen jo tänään.**

► **Hae tuhansien sinkkujen joukosta, tästä!** **e-kontakti.fi**

D-vitamiinin valmistuttaja: "Ei ole mitään syytä huijata"



KATSO TAULUKKO TULOKSISTA!			
Seurava	1	2	3
1. Järjestyksen	100	100	100
2. Järjestyksen	100	100	100
3. Järjestyksen	100	100	100
4. Järjestyksen	100	100	100
5. Järjestyksen	100	100	100
6. Järjestyksen	100	100	100
7. Järjestyksen	100	100	100
8. Järjestyksen	100	100	100
9. Järjestyksen	100	100	100
10. Järjestyksen	100	100	100

Testi: Nämä D-vitamiinit pärjäsivät, nämä eivät

ULKOMAAT

URHEILU

Hän kauppaal neit-
sytensä: 600000 CI

Hihtäjättärien reaktio
kokuun: nakuku!

VIINDE

KATSO! Tiina Jylhä
tänään 50 vuotta -
nostalgiaa!

KATSO! Ohhoh, upea
Lola hurmaa pin up -
tyttöä!

IL-TV SUOMEN SUURIN NETTI-TV!

Jatkoon!

IL-TV

Teinipoikien uskomaton Rihanna-cover!

Iäkäs pariskunta
pelastui palavasta
autosta viime hetkellä

Onko tässä maailman
plenin koira?

Uskomaton museo -
Jokaisen Bond-fanin
toiveuni!

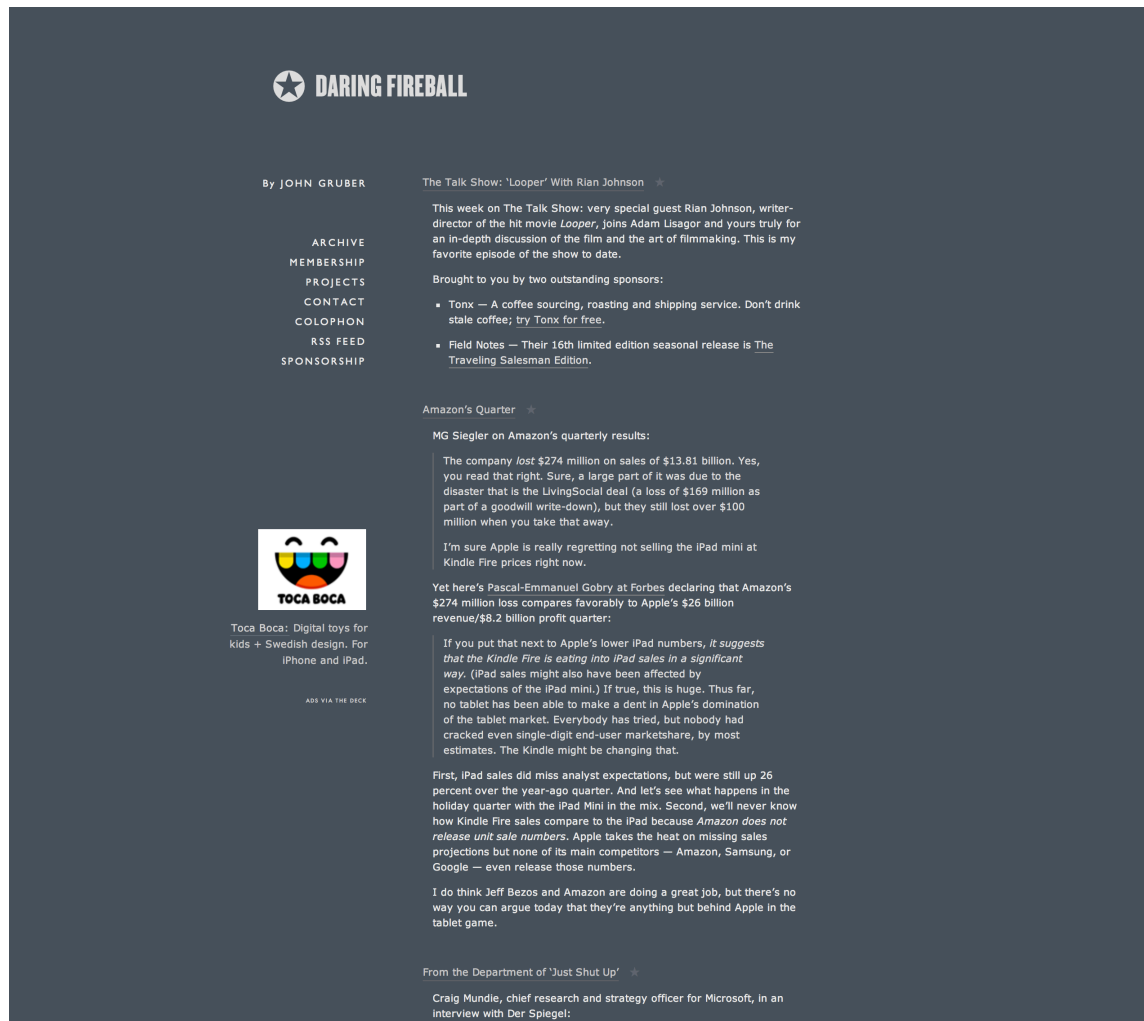
Mitä ihmettä -
Teinityttö omistaa
kaksi taloa!

LISÄÄ VIDEOITA

SAA

Kuva 1. Iltalehden ahdas rakenne ja tunkeileva mainonta (Iltalehti 27.10.2012).

Eräs esimerkki toimivaa mainontaa esittävästä ja tekstille tilaa varaavasta blogista on Daring Fireball (kuva 2), josta kerron myös erikseen lisää myöhemmin käytettävyyssesimerkeissä. Käytännössä sen mainonta on kirjoittajan itsensä suosittelemaa tai vähintään blogin sisältöön sopivaa, mikä nostaa myös sivun mainostilan arvoa. Joka viikon alussa ja lopussa blogaaja kertoo viikon sponso-rista ja yhdestä heidän tuotteestaan. Sen lisäksi sivunavigaation alla on vaihtu-va pienikokoinen mainoskuva (Daring Fireball 2012).



Kuva 2. Daring Fireball -blogin etusivu (Daring Fireball 2012).

Kuvagallerian toteutustapa on yksi ongelma, jossa pitää punnita kuvien määrää ja yksinkertaisuutta. Pienellä kuvamäärällä ei välttämättä kannata paljastaa kuvavuodukkoa, vaan laittaa ne näkymään yksi kerrallaan samaan näkymään nuolilla vaihdettavaksi. Tällaista toteutusta kutsutaan muun muassa kuvakaruselliksi ja se on kaikessa yksinkertaisuudessaan todella toimiva tapa esittää rajattu määrä kuvia. Toisaalta liian suurella kuvamäärällä sellaisen selaaminen on turhan hidasta ja ahdistavaa.

2.2 Web-suunnittelun suuri käyttökokemusongelma

Aikaisemmin esittelemäni kirjaesimerkki on vielä melko yksinkertainen siksi, että kyseessä on staattinen lopputuote. Siinä ei ole varsinaisia suuria muuttujia vaan kaikki päätökset tehdään kerran ja ne ovat pysyviä. Lisäksi kirjapaino on monta sataa vuotta vanha keksintö ja siksi alan kollektiivinen tietotaso on jo saavuttanut huippunsa. Digitaaliset mediat taas ovat dynaamisia sisältäen monia alustasta riippuvia muutoksia, mikä on tietenkin hyvä asia ja mahdollistaa paljon uudenlaista ajattelua, mutta ne vaativat myös erityistä käytettävyyssuunnittelua ollakseen yhtä itsestään selvän tuntuisia kuin staattisemmat mediat. Web-suunnittelu on vain noin 20 vuotta vanha ala, jos sen voi laskea alkaneen, kun ensimmäiset graafiset nettisivut tehtiin. (The Living Internet 2012.)

Nuorena alana web-suunnittelu on myös nopeasti kehittyvä, ja teknologiakin voi muuttua paljon pelkästään yhden vuoden aikana. Esimerkiksi vuonna 2012 julkaistiin iPad, jonka niin kutsuttu retina-näyttö perustuu jokaisen pikselin jakamiseen neljäksi eri pikseliksi. Kaikki elementit ovat yhä samankokoisia kuin edellisessäkin tavanomaisessa näytössä, mutta pikselimäärän nelinkertaistumisen tai resoluution kaksinkertaistumisen takia teksti ja kuvat piirtyvät huomattavasti tarkemmin. Tämän lisäksi retinan tarkoitus ja lopputulos on todistetustikin, ettei ihmissilmä enää kohtuulliselta käyttöetäisyydeltä erota pikselimatriisia (Jones, 2010). Tämä saa oman kokemukseni mukaan näytön muistuttamaan enemmän kiiltävää paperia kuin elektroniikkaa. Jos katseltavan kuvan koko ei riitä, se tietenkin toimii normaalisti, mutta ei vain täytä näytön potentiaalia ja näyttää mielestäni peräti huonommalta kuin tavanomaisella näytöllä. Kaksinkertaistetuille resursseille ei kuitenkaan ole mitään valmista mallia nykyisissä web-standardeissa, vaan lähinnä erilaisia javascript- ja web-palvelin-kiertoratkaisuja (Marquis 2012a). Tämän takia nettisivujen ylläpitäjälle kyse ei ole edes pelkästä pienestä päivityksestä, vaan retina-näyttö on hankala pulma täysin uusienkin sivujen kehityksessä.

Käyttökokemus web-suunnittelussa on kuitenkin syvemmässä kriisissä kuin pelkissä teknisissä yksityiskohdissa. Graafisen suunnittelun historian konteks-

tissa netti on tuore media. Ethan Marcotte pitää vanhojen termien siirtämistä uusiin medioihin luonnollisena ihmisen ominaisuutena. Mutta olemme ottaneet web-suunnitteluun mukaan myös konseptin, joka ei välttämättä enää siihen kuulu: kankaan. Marcotte toteaa tämän olevan taiteellisesti voimakas teko, koska se määrittää rajat tulevalle työlle ennen ensimmäistäkään pensselinvetoa. Ongelmalliseksi sen tekee web-suunnittelussa selain, joka on netin uloin kangas. Mutta joustavana ikkunana tai muuttuvana laitekokona selain ei ole mitenkään luotettava kangas. (Marcotte 2010a, 3.) Tämä on eräänlainen suunnittelijan painajainen, mutta ei pelkästään negatiivinen asia. John Allsopp kirjoitti jo vuonna 2000 seuraavasti:

"The control which designers know in the print medium, and often desire in the web medium, is simply a function of the limitation of the printed page. We should embrace the fact that the web doesn't have the same constraints, and design for this flexibility. But first, we must "accept the ebb and flow of things." (Allsopp 2000.)

Allsoppin lainaus tarkoittaa, että painomediassa tunnettu lopputuloksen kontrolli on vain painetun sivun luoma rajoite. Meidän tulisi hyväksyä ja sisäistää, ettei netissä ole samoja rajoitteita ja suunnitella pitäen tämä joustavuus mielessä. Toisin sanoen web-suunnittelussa ei ole mitään varmuutta vierailijan laitteesta ja selaimen muodosta eli kankaan rajoista. Tarkoitus on kuitenkin kyseenalaistaa kiinteän kankaan absoluuttinen paremmuus.

Mietittäessä internetiä kokonaisuutena kiinteän kankaan ongelma on hyvinkin ratkaiseva. Nykyään selaukseen käytetyt laitteet ovat sirpaloituneet niin moneen eri näyttökokoon ja -tarkkuuteen, että tiettyyn kokoon suunniteltu nettisivu on harvoin juuri sopivan kokoinen. Koen itse noin 1000 pikseliä leveän rakenteen selailun älypuhelimella pidemmän päälle rasittavaksi vaikka sivun zoomaus toimiikin hyvin iPhonella ja toisaalta samat sivut tuntuvat usein liian pieniltä isolla 27" näytöllä. Stat Counterin syyskuun 2012 maailmanlaajuisessa resoluutiotilastossa resoluutiot 1366x768 (21,73 %), 1024x768 (16,65 %) ja 1280x800 pikseliä (11,5 %) muodostavat noin 50 prosenttia kaikista resoluutioista (Stat Counter 2012a). Tämän lisäksi huomattavasti suurempi resoluutio 1920x1080 pikseliä eli myös niin kutsuttu full HD -resoluutio muodostaa 5.92 %:n osuuden. Kun tähän vertailuun otetaan mukaan mobiililaitteiden yleisin resoluutio 320x480 pikseliä

niin on selvää miksi yhteen kokoon suunnitteleminen ei ole järkevää. (Stat Counter 2012b.)

Jos vaikka alakoulun oppikirjan sisältö siirrettäisiin nettisivuksi, miten saataisiin varmistettua, että se pysyisi käytettävänä, eli ei ahdistaisi sisällön koon puolesta ja toimisi teknisesti mahdollisimman hyvin joka laitteella? Kyseessä ei ole mahdollon tehtävä, mutta se vaatii monimutkaisemman suunnittelutyön painettuun kirjaan verrattuna. Tämä onkin helppo jättää tekemättä, jolloin sivut tehdään käytännössä toimivaksi vain yhtä näyttökokoja ajatellen, mikä ei juuritasolla sisäistä selaimen muodostaman kankaan mahdollisuuksia vaan pysyy painetun median hengessä.

Pelkän dynaamisen rakenteen lisäksi erityisesti oppikirjan tapauksessa sisällönkin muotoa on hyvä kyseenalaistaa. Esimerkiksi fysiikan ilmiöitä voisi animoida ja harjoitustehtävät voisivat olla interaktiivisia. Allsopp tuntuu kannustavan nimenomaan myös painetun median esitystyylin haastamiseen web-suunnittelussa.

2.3 Web-suunnittelun haasteet

Web-suunnittelussa kenties tärkein, mutta hankalin vaihe on tarpeiden eli päätoiminnon selvittäminen. Nykyään yksi suosittu toimintamalli on suunnitella sivut niin sanotusti "mobile first" eli ensin mobiililaitteille. Tämä toimintamalli perustuu siihen, että mobiililaitte on kankaaltaan ahtain ja siksi yksinkertaisuutta vaativa alusta, joten sitä käytetään lähtökohtana käyttöliittymä- ja web-suunnittelulle (Wroblewski 2011, 19).

Mobile firstin käyttöä suositellaan, vaikka syyskuussa 2012 vasta noin 12% net-tisurffailusta tapahtuu mobiililaitteilla. Osasyynä on se, että kehityksen suunta on selvä. Vuosi sitten vastaava luku oli vain noin 6%. Mutta vaikka kasvava mobiilikäyttäjien määräkään ei kiinnostaisi, Luke Wroblewski väittää mobile firstillä olevan positiivisia vaikutuksia myös web-suunnitteluprosessiin. Mobiililaitteiden pieni koko pakottaa määrittelemään nettisivun tai -sovelluksen päätoiminnot (Wroblewski 2011, 19). Samaa pienelle näytölle suunniteltua rakennetta

ei ole tietenkään mielekästä skaalata suoraan isolle näytölle, vaan käyttöliittymää rikastetaan asteittain ja rakennetta levitetään, kun näyttökoon potentiaali kasvaa. Tällöin sivua voi jakaa palstoihin ja navigaation sekä sivupalkin voi siirtää sisällön vierelle, johon olemme tottuneet. Mobiililla tuoreesti mietittyä yksinkertaisuutta on sitten helppo verrata suunniteltaessa rikastamista. Jos lopputulos tuntuu huonommalta isolla näytöllä verrattuna mobiilikokoon, suunnittelu vaatii uusintaa.

Wroblewski demonstroi mobiiliin pakottaman yksinkertaisuuden ja ison näytön vapaamman suunnittelun ristiriitaa vertaamalla Southwest Airlines -lentoyhtiön nettisivuja PC:llä ja mobiililaitteella katsottuna. Mobiililaitteella korostuvat lentoyhtiön palvelut, kun taas PC:llä rakenne on sekoitus mainontaa ja oleellista toimintaa. Täten asiakkaiden tarpeiden näkökulmasta Southwest Airlinesin sivuilla on helpompi asioida mobiililaitteilla, mikä on hämmentävä realiteetti.

Aiemmin mainitsemani John Allsoppin lainaus on päässyt toteutumaan kunnolla vasta muutaman viime vuoden aikana uusien web-standardien avulla. Ethan Marcotte yhdistää näitä julkistaessaan responsiivisen web-suunnittelun käsitteen vuonna 2010 samannimisessä artikkelissa (responsive web design). Siitä on ollut paljon keskustelua web-suunnittelupiireissä siitä lähtien (Marcotte 2010b). On myös tärkeää ymmärtää miksi sitä tehdään, ja että se kuuluu mukautuvan suunnittelun piiriin. Pelkkä responsiivinen suunnittelu tekniikkana ei kuitenkaan vielä takaa laatua. Tällöin on siis täysin mahdollista saada aikaiseksi toimivia sivuja eri laitteilla, mutta niihin päädytään sattuman ja oletuksien kautta eikä perusteltujen päätösten avulla. Tai lopputulos on hyvä, mutta jotain erikoisen hyvää jää keksimättä.

Mukautuvan suunnittelun ideana on ottaa huomioon kaikki laitekoot ja unohtaa yksi ainut sivurakenne, eli aivan kuten Allsopp kirjoitti jo yli kymmenen vuotta sitten. Tämä joustavuus nostaa internetin yhä korkeammalle tasolle mediana. Mukautuva suunnittelu ei myöskään itsessään vaadi mitään tiettyä toteutustapaa tai juuri responsiivista suunnittelua. (Ponchot 2011.) Selaintekniikka on kuitenkin niin tarkkaan määriteltyä web-standardien kautta, ettei asioita voi tehdä kuin muutamalla tavalla. Pelkän responsiivisuuden lisäksi on myös otettava

huomioon käytettävyys hiiren ja kosketusnäytön kanssa sekä se, miten paljon näytölle voi käytännössä ahtaa elementtejä. Muun muassa tähän mobile first kannustaa. (Wroblewski 2011, 67-68.) Toisen ääripään mukautumista on tehdä tarvemäärittely uudestaan eri laitteille, mutta tällöin on tietysti otettava huomioon laitekannan ja käyttötapojen muuttuminen, mikä on Ethan Marcotten mukaan kestävä ratkaisu (Marcotte 2010a, 6).

Käyttöliittymässä pitää ottaa huomioon hiiren ja kosketusnäytön erot. Tällainen on muun muassa hover-tapahtuman puuttuminen (käyttäjä vie hiiren cursorin elementin päälle) kosketusnäytöstä, mikä hankaloittaa hoverista avautuvien pudotusvalikkojen käyttöä. Toisaalta kosketusnäytöllä käyttöliittymässä ei ole tarvetta nuolinapeille. Esimerkiksi kuvakaruselleissa on tavanomaista käyttää sivuille asetettuja nuolia kuvan vaihtamiseksi, mutta kosketusnäytöllä tutumpi ja luontevampi tapa on pyyhkäistä elementtiä, kuten edenspiekermann.comissa (Eden Spiekermann 2012).

Toimintamallien erojen lisäksi esimerkiksi Apple suosittelee omille tämän hetkille laitteilleen vähintään 44x44 pikselin levyisen kosketusalueen (Apple 2012b). Kosketettava elementti saa silti olla visuaalisesti pienempi, kunhan siihen on helppo osua. Luke Wroblewski on tutkinut useamman eri lähteen johtopäätöksiä kosketuselementin minimikoosta. Microsoft suosittelee vähintään 7x7 mm:n kokoista aluetta, mutta suosittelee suurempaa erityisesti usein käytetyille toiminnoille ja, jos elementti sijaitsee näytön reunoilla tai on osa kosketusten sarjaa, kuten puhelimen numeronäppäimissä (Wroblewski 2010.)

Koska responsiivisesta suunnittelusta on keskusteltu vasta muutaman vuoden ajan, siitä syntyneitä teknisiä tarpeita ei ole vielä täysin standardisoitu. Responsiivinen rakenne toimii hyvin useimmilla selaimilla CSS3:n media query -ominaisuuden avulla ja siitä on havaintoni mukaan saatavilla paljonkin tietoa eri lähteistä. (CSS-Tricks, 2010.) Toisaalta esimerkiksi sisältökuvien tarjoaminen on yhä ilman standardisoitua tapaa. Potentiaalisia ratkaisuja on kyllä monia, mutta yksikään niistä ei ole täydellinen tai kovinkaan helppo. Kehittäjien itse luomia ratkaisuja on useita, mutta sen lisäksi uudesta responsiivisesta HTML-tagista on tehty ehdotuksia W3C:lle ja WHATWG:lle ennenkin. (Marquis 2012b.)

Tämä ongelma on osittain sama kuin aikaisemmin mainitsemani Applen retina-näytön luomassa kaksinkertaisen kuvan tarpeessa, joten web-standardi responsiivisille sisältökuvilla ratkaisisi paljon. Staattisia pysyvään rakenteeseen kuuluvia kuvia on onneksi helppo tarjota käyttäen samaa niin kutsuttua media query -tekniikkaa kuin responsiivisessa suunnittelussa muutenkin.

2.4 Lisää käyttökokemusesimerkkejä

Kerron seuraavaksi opinnäytetyöni kirjoittamiseen käyttämästäni ohjelmasta. Arkisista asioista mikään ei ole minulle yhtä vaikeaa kuin kirjoittaminen, joten kiinnitän huomiota työskentelyyni. En käytä enää suoraan Microsoft Wordia, koska se on mielestäni tarpeettoman sekava useamman sivun tuottamiseen ja vie huomiota pois sisällöstä. Kun käytettävä ulkoasu ja dokumenttipohja tulee valmiina ratkaisuna, ei sitä ole mielekästä käyttää kirjoitusvaiheessa vaan vasta lopuksi. Tämä tekee tietenkin kuvien ja kuvioiden käsittelemisestä hieman haastavampaa, mutta ei mitenkään mahdotonta ja kirjoittamiseen verrattuna sekin on paljon helpompaa.

Writer on suunnittelutoimisto iA:n kehittämä kirjoitussovellus. Se on saatavilla Macin käyttämälle OS X -käyttöjärjestelmälle ja iPhonen, iPod Touchin sekä iPadin iOS:lle. Writerin konsepti on selkeä kirjoituskonetta muistuttava fontti, joka soveltuu erityisen hyvin tekstinkäsittelyyn. Se ei ole sinänsä maailman kaunein tai luettavin fontti, mutta kirjoittamisessa se toimii muutenkin kuin viittauksena menneeseen aikaan. Writerissä ei ole minkäänlaisia muotoilutyökaluja vaan muotoilu tapahtuu automaattisesti Markdown-kielen mukaan. Tarkoituksena ei ole antaa käyttäjän muotoilla valmista dokumenttia vaan luoda standardimallista tekstiä kirjoitusvaiheeseen. Tämä säästää paljon aikaa turhasta muotoilusta. (Information Architects 2012.)

Visuaalisesti Writer miellyttää, koska se on hyvin yksinkertainen, muttei kuitenkaan laiskan yksinkertainen. Tausta ei ole täysin valkoinen vaan lähempänä luonnonvalkoista ja muistuttaa paljon paperia. Siinä on myös lievä tekstuuri, jota ei huomaa kuin keskittymällä. Tämä pehmentää näkymää, eikä se näytä enää

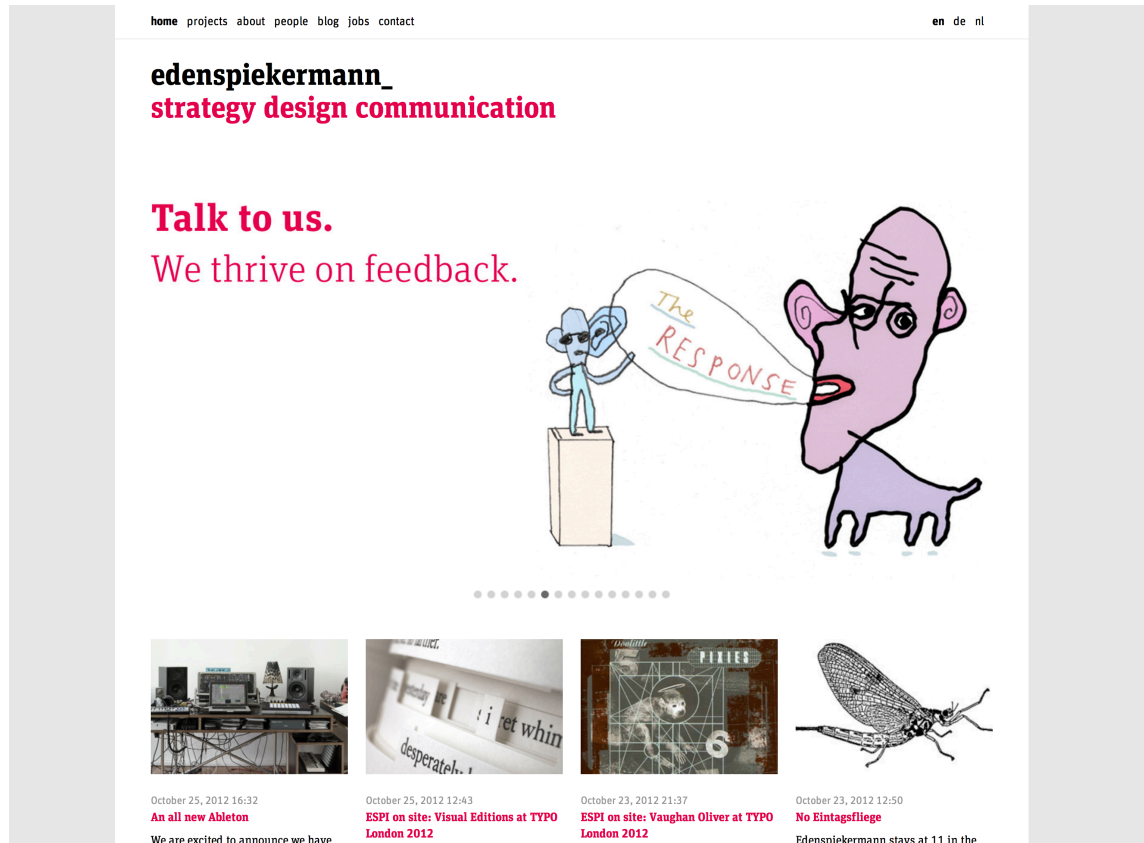
niin paljoa valkoiselta valolta. Teksti on noin 33% harmaata, mikä kuulostaa omituiselta, mutta koska näytöt perustuvat kuitenkin valoon, on harmaa suhteellinen väri ja teksti näyttääkin mustalta valkoista taustaa vasten. Harmaan käyttäminen kuitenkin pehmentää kontrastia. Itse olen havainnut kontrastilla olevan alarajan lisäksi yläraja. Tumman sininen teksti mustalla pohjalla on hankala lukea, mutta myös täysin musta teksti täysin valkoisella pohjalla on jokseenkin ahdistavaa.

Käytettävyyksesimerkkinä Writer toimii, koska se tarjoaa hyvin voimakkaasti rajatun käyttökokemuksen kirjoittamiseen ja mahdollistaa täydellisen keskittymisen tietokoneella. Se on hyvä esimerkki siitä miten rajua tarpeiden mukaan suunnitteleminen voi olla.

Daring Fireball on John Gruberin pitämä teknologiaan, tuotekehitykseen ja käyttökokemukseen keskittyvä blogi (kuva 2). Aihe eksyy välillä urheiluun ja suunnitteluunkin, mutta pääasiassa teemana on unohdetumpi teknologian kulttuurillinen laadukkuus. Kirjoitukset ovat usein lyhyitä ja perustuvat linkkiin, josta Gruber kirjoittaa oman mielipiteensä tai vastineen. Hän kirjoittaa myös säännöllisesti pitkiä blogikirjoituksia keskittyen johonkin tiettyyn aiheeseen.

Sivu eroaa blogista aiemmin mainittujen mainosten esittämistavan osalta, mutta myös yksinkertaisuudellaan. Pelkistetyllä etusivulla on navigaatio vasemmalla, logo sen päällä ja artikkelit oikealla puolella. Otsikko on linkki muualle internetiin ja oikealla puolella oleva tähti on linkki itse artikkeliin jakamista varten. Navigaatio ei ole korostettu, koska sitä käyttää todella harvoin ja pääpaino sivulla on blogin seuraamisessa.

Blogissa tällainen pelkistetty tekstisisältöä korostava visuaalinen tyyli on tarkoituksenmukainen ja kertoo tarvelähtöisyydestään. Moni amatöörin suunnittelema blogi sisältää paljon ylimääräistä visuaalisuutta, mikä ei ole minusta paras ratkaisu toistuvaan käyttöön, koska vahva tyyli vie alussa huomiota pois itse muuttuvasta sisällöstä ja alkaa tuntua vanhentuneelta ajan mittaan.



Kuva 3. edenspiekermann.com (Eden Spiekermann 2012).

Edenspiekermann on Saksassa, Hollannissa ja Yhdysvalloissa toimiva suunnittelutoimisto. Heidän nettisivunsa (kuva 3) ovat suhteellisen pelkistetyt ja korostavat sisältöä kontrolloidusti, joten jokainen alisivu tuntuu mietityltä kokonaisuudelta ja sivuja on mielenkiintoista selailla. Nämäkin sivut ovat tarkemmalla tarkastelulla pelkistetyt, mutta ne eivät välttämättä tunnu siltä.

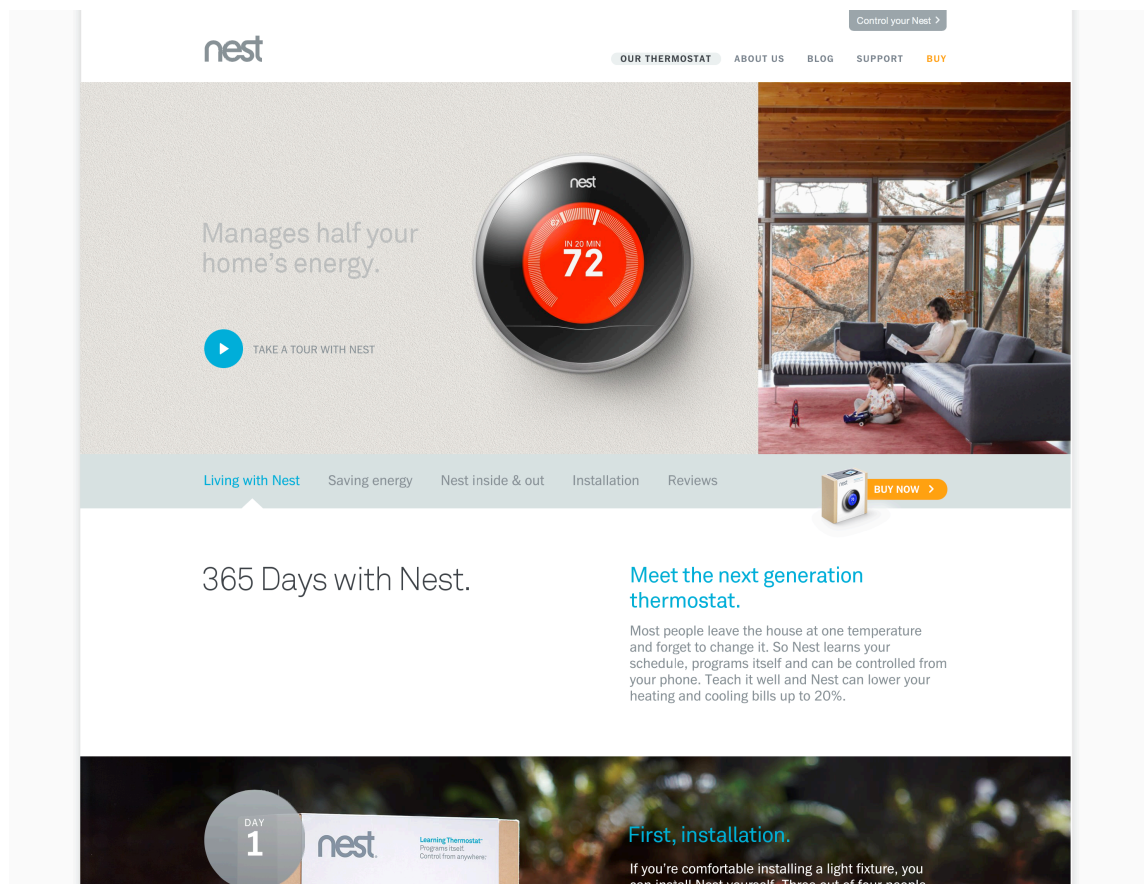
Esimerkkinä edenspiekermann.com on mielenkiintoinen siksi, että se toimii ja näkyy erityisen hyvin laitteesta riippumatta. Kuvakarusellien ohjaus toimii myös kuvaa raahaamalla. Henkilöstö- ja referenssisivuilla hyödynnetään selaimen epävarmaa leveyttä. Sivun elementit eli ihmisten tai referenssien kuvat ovat aina tietyn kokoisia, mutta niitä vain sijoitetaan useampaan palstaan selaimen leveyden mukaan. Blogisivusta huomaa myös, miten koko sivupohjaa ja selaimen leveyttä ei oteta käyttöön isommallakaan ruudulla. Sivua ei kannata täyttää edes pelkällä tekstillä, koska rivien leveydelle on hyödyllisempää asettaa jokin raja luettavuuden ylläpitämiseksi. (Holst 2010.) Tällöin ainoa isolla näytöllä saa-

vutettu hyöty on hengittävämpi rakenne, mutta se ei ole suinkaan mitenkään pieni etu. Suomen kielen sanojen pituus asettaa kuitenkin riveille hieman eri vaatimukset.

3 SUUNNITTELU

Opinnäytetyöni empiriana on toteuttaa mielenkiintoinen esityksen kaltainen esittelysivu Zeeland Turku Oy:lle. Se eroaa tavanomaisesta nettisivusta siten, ettei tarkoitus ole lähteä liikeideasta ja päätyä y-tunnukseen. Sen sijaan esittely keskittyy toimiston ihmisiin ja henkeen.

Esityksen kaltaisuus viittaa nettisivun rakenteeseen. Normaalisti nettisivut ovat epälineaariset ja navigointi jätetään vapaasti vierailijan valintojen varaan. Esityksen kaltaisessa rakenteessa kokemus on päinvastoin täysin lineaarinen ja navigaatio vain vierittää sivun otsikoiden välillä ilman sivulatauksia. Tämän tarkoitus on mahdollistaa kontrolli tarinankerronnalle, mitä kohtaa nykyään yhä



useammin netissä.

Kuva 4. "Living with Nest" -alasivu Nestin sivustolla (Nest 2012).

Eräs esimerkki tarinankerronnasta on käyttökokemusesimerkkinäkin mainittu Nest-älytermostaatin sivusto nest.com (kuva 4). Siellä on useampikin pystysuunnassa korkea alisivu, mutta selkein tarinankerronta on "Living with nest" -alisivulla, joka selventää Nestin tarkoituksen hyppien eteenpäin päivissä. Tarina päättyy hintaan ja saatavuuteen, mikä selventää, ettei kyseessä ole vain utopistinen konsepti.

Oman suunnittelutyöni hankala osuus on yrittää päätellä, mitä osia yhdistelen saamistani vaikutteista, sillä kaikki näkemäni ei missään nimessä ole oleellista Zeeland Turun tapauksessa. Vaikutteiden yhdisteleminen järkeväksi ja tarkoituksenmukaiseksi kokonaisuudeksi onkin hankala ja luovuutta vaativa prosessi.

Päädyin tähän suunnitelmaan käymällä läpi muilta oppimaani prosessia. Pyrin selvittämään tarkasti, mitä Zeeland Turku tarvitsee ja mihin tarkoitukseen. Pelkkä sana "nettisivu" tai "digitaalinen esittelymateriaali" ei kerro kaikkea. Puran nyt prosessin vaiheet ja esittelen sen jälkeen visuaalisen suunnitelman sekä koko konseptin.

3.1 Lähtötilanne

Suunnittelun ja taiteen eroa ei aina tunneta, mutta Erik Spiekermann tiivistää sen seuraavasti: suunnittelija hyödyntää samoja taiteellisia metodeja, mutta ratkaistaakseen asiakkaan ongelman oman maailmankuvansa ilmaisun sijaan. Spiekermann on kirjoittanut näin Mike Monteiron "Design is a job" -kirjan esipuheessa. (Spiekermann 2012.) Kyseinen kirja syventyy suunnittelualaan juuri tästä näkökulmasta, ja kertoo miten oikeaan ratkaisuun päätyminen on mahdollista vaikeankin oloisen asiakkaan kanssa. Yleensä asiakkaat haluavat toimia oikein, mutta he saattavat silti sortua erilaisiin huonoihin tapoihin ja vaatia tiettyjä suunnitteluratkaisuja, vaikka palkkasivatkin juuri oikean alan henkilön tai yrityksen prosessoimaan ongelmaa. Tärkeä osa suunnittelijan työtä on perustella päätökset hyvin ja saada asiakas luottamaan ratkaisuihin. (Monteiro 2012, 27.)

Asiakkaille suunnittelu on hankalaa monestakin syystä. Yleisesti varsinaisen ongelman etsimistä ja ratkaisua häiritsee asiakkaan vääristynyt kuva suunnitte-

lutyöstä. Liian monessa tapauksessa suunnittelijalta odotetaan välitöntä luovaa tulosta omiin ongelmiin ja ratkaisuehdotuksiin. Pahimmassa tapauksessa suunnittelija ei tunne vielä edes itse asiakasyrityksen toimintaa ja tuotteita kuin nopean nettihaun tuloksena, kun asiakas jo haluaa malleja. Rehellisen ammattilaisen onkin syytä vastustaa miellyttämisen kiusausta ja selittää miten oma suunnitteluprosessi toimii sen sijaan, että tarjoaisi heti ensimmäisessä tapaamisessa irtonaista "pikkukivaa" tilaajan iloksi, koska se harhauttaa pois ydinongelmasta. (Monteiro 2012, 57.) Tämä ei tietenkään sulje pois visuaalisia referenssejä, jotka ovat esimerkkejä siitä, millaisiin tuloksiin oma suunnitteluprosessi ja osaaminen on ennen johtanut. (Monteiro 2012, 15.)

Tunsin itse Zeeland Turku Oy:n sisältä melko hyvin jo ennen toimeksiantoa, koska suoritin kolme kuukautta harjoittelujaksostani siellä. Kerron kuitenkin Zeelandista ja sen historiasta selventääkseni lähtötilannetta sekä valintojeni syitä. Nykyään Zeeland Turku Oy on osa laajempaa Zeeland Oyj -yhtiötä, mutta toiminnan juuret ovat nimenomaan Turussa.

Zeeland Oyj on useasta tytäryhtiöstä koostuva konserni, joka kykenee tarjoamaan markkinoinnin ja mainonnan erikoisosaamista laidasta laitaan. Suurimmat tytäryhtiöt eli toimistot Zeelandilla on Helsingissä ja Turussa. Muita yhtiöitä ovat muun muassa Zeeland Idea Baltics, joka on Tallinnassa sijaitseva koko Itämeren alueelle palvelujaan myyvä digitoimisto ja Zeeland Maggie, joka on erikoistunut asiakaslehtien suunnitteluun sekä lopulta niiden tuottamiseen.

Zeeland alkoi Turusta alunperin Enala/Hyysalo –nimisenä mainostoimistona. Se perustettiin vuonna 1999 ja fuusioitui helsinkiläisen pienen Neljä tuumaa –mainostoimiston kanssa vuonna 2005 muuttaen myöhemmin nimensä Zeeland Oy:ksi. Nykyisin Helsingin toimisto työllistää eniten ihmisiä ja tätä kirjoittaessa koko Zeeland Oyj –konsernilla on yhteensä 120 työntekijää. Zeelandin Turun toimisto työllistää noin 20 ihmistä ja sijaitsee Turun keskustassa. Sillä on myös ollut tutkimuksissa Suomen paras palveluasenne ja kustannustehokkuus.

Läheisemmän tutustumisen tarve on perusteltavissa sillä, että mainonnansuunnittelu ja suunnittelutyö ylipäänsä vaatii luottamusta sekä tuntemusta molem-

milta osapuolilta. Suunnittelutyösuhteen ytimessä asiakas antaa liiketoimintansa ja ongelmansa suunnittelijoiden käsiin. Suunnittelutoimisto taas haluaa tuntea asiakkaansa yhteyshenkilöt syvemmin, jotta viestintä olisi mahdollisimman avointa ja sujuvaa. Jos suhteen vuorovaikutus on lähtökohtaisesti aukkoista, projektit sujuvat huonosti ja kaikki ongelmat näkyvät myös lopputuloksessa. (Monteiro 2012, 23.)

3.2 Tavoitteet

Lähtötilanne antaa viimeistään lisäselvitysten jälkeen hyvän kuvan asiakkaan tarpeista. Nämä eivät välttämättä tarkoita pelkästään hänen omia toiveita, vaan myös suunnittelijan havaitsemia ongelmia. Usein asiakas on liian lähellä ongelmaansa, eikä siten kykene keksimään oikeita tavoitteita ja ratkaisua. Tämän takia mainitsinkin aikaisemmin asiakkaan omien ratkaisuehdotuksien olevan ongelmallisia prosessin alkuvaiheessa. Järkevien tavoitteiden etsiminen onkin suunnittelijalta saatava tärkeä lisäarvo, mitä ei tule vähätellä projektin alkuvaiheessa. Tosin nämä löytyvät toki usein saman katon alta, mistä on apua jo suunnittelussa. Nämä kaksi asiaa eivät saisi kuitenkaan mennä sekaisin kummallakaan osapuolella. (Monteiro 2012, 57.)

Monteiron kokemuksien perusteella voisi väittää, että hyvän kaksisuuntaisen kommunikaation tuloksena on myös asiakkaan liiketoiminnan ja tuotteiden tai palvelujen ymmärrys. Omien kokemuksieni perusteella positiivisten tunteiden herättäminen itsessä on hankalaa tai peräti mahdotonta, jos käsiteltävä aihe on vieras. Toisaalta avoimesti innokas asiakas on herkkä tartuttamaan intohimoaan myös muihin ja suunnittelijaan. Asiakkaan työn kunnioittaminen on vaatimus hyvässä työsuhteessa (Monteiro 2012, 25.)

Suunnittelija kaipaa kunnioitusta itsekkin. Kenties paras mittari työn arvostamiselle on se, että suunnittelija saa haluamansa kohtuullisen korvauksen tekemästään työstä. Toisaalta tämä tarkoittaa myös sitä, että suunnittelijan täytyy tietää työnsä arvo. Jos hän ei itse kunnioita työtään, palkkiokaan tuskin on tarpeeksi korkea suhteessa elämisen muihin kustannuksiin. (Monteiro 2012, 33.)

Tarpeet ja tieto konkretisoidaan projektin tavoitteiksi. Tarpeet tulisi saada sopimaan keskenään siten, ettei mikään osa tunnu irtonaiselta, ja jotta ne palvelisivat oleellista asiaa. Asiakkaan ja suunnittelijankin haaveilemat tarpeet eivät aina ole realistisia asiakkaan asettamassa kontekstissa. Toisaalta tähän kuuluvat myös aika- ja rahatavoitteet. Parhaaseen ei aina ole varaa tai aikaa, joten lopputuloksen täytyy kunnioittaa myös niitä. Ajallisesti ei ole järkevää tavoitella sellaista lopputulosta, jolla on suuri riski jäädä keskeneräiseksi tai josta ei ehdi hiomaan pois keskinkertaisuutta.

Selvitin Turun toimiston tarpeita yhdessä kahden työntekijän kanssa, joista toinen on projektipäällikkö ja toinen art director. Nämä kaksi pitkään Zeelandissa työskennellyttä osasivat kertoa nopeasti, mitkä Turun toimiston edut ja ominaispiirteet ovat. Ihmiseen ja ympäristöön keskittyminen oli päätetty jo aiemmin, eli juuri niille oli tilausta. Minun ei ollut hankalaa sisäistää ajatusta, koska olin itse kokenut toimistohengen jo harjoitteluni aikana Zeelandilla kesällä 2011. Osittain osasin siis itsekin päätellä, mihin sivun tulisi keskittyä.

Zeeland Turun esittelysivulle koin tarpeelliseksi keskittyä toimiston uniikkeihin asioihin, eli ihmisiin ja ympäristöön. Sivun viestinnällisenä tavoitteena on kertoa Zeeland Turku Oy:n ihmisistä ja heidän muodostamastaan ympäristöstä. Tämä tarkoittaa myös perinteisen yrityssivun mielikuvan poistamista. Tämän takia päädyin yksittäiseen esittelysivuun, joka mahdollistaa ja toisaalta myös olettaa jonkinlaista tarinankerrontaa. Koko nettisivu luetaan kerralla ylhäältä alas, jolloin asiat voi esittää mielenkiintoisemmin, koska lukujärjestys on kontrolloitu ja osien välejä voi rikastaa miljöökuvilla.

Tämän lisäksi tavoitteena on pystyä päivittämään usein muuttuvia tietoja sisällönhallinnan kautta. Käytännössä se tarkoittaa henkilöstöä ja Zeeland Turun asiakkaita. Staattisten osien muuttaminen jälkeenpäin ei voi kuitenkaan olla sekään mahdotonta, ja niiden lisääminen sisällönhallinnan piiriin jää myöhemmin tarvittaessa toteutettavaksi optioksi.

Kolmantena tavoitteena on rikastaa sivun käyttökokemusta siten, että yritysesittelyn lisäksi mieleen jää kivoja kohtia. Tähän tavoitteeseen päästään muun mu-

assa animoinnilla, interaktiivisilla osilla ja mahdollisesti virkistävillä efekteillä. Selattavalla sivulla voi käyttää esimerkiksi parallaksia eli ilmiötä, jossa katsojan ja taustan välissä oleva elementti näyttää liikkuvan. Koska näyttö on kaksiulotteinen taso, parallaksi on pakotettu efekti ja omalla keksityllä kerroksellaan oleva elementti liikkuu tietenkin aivan oikeasti. Tällä tekniikalla ei haetakaan realismia tai itse parallaksia vaan illuusiota kolmiulotteisuudesta.

3.3 Toteutusalue

Nettisivut koostuvat modernin selaimen päässä kolmesta pääelementistä: HTML-rakenteesta, CSS-tyylimäärittelystä ja dynaamisuuden mahdollistavasta javascript-moottorista. Kaksi ensimmäistä ikään kuin määrittelevät sivun "taiton", kun taas javascript-skripteillä voi muokata molempia jälkikäteen. Lähtökohteisesti selain hakee palvelimelta staattisia sivuja, joita voi vielä muuttaa selaushetkellä javascriptillä. Tämä on kuitenkin vasta selaimen osuus, ja surffauksen suhteeseen kuuluu myös palvelin, josta sivut ladataan.

Web-palvelimen pääperiaate on tarjota HTML-tiedostoja ja niiden vaatimat resurssit, kuten kuva- ja javascript-tiedostot. On kuitenkin mahdollista muodostaa HTML-tiedostoja dynaamisesti lennosta palvelimella. Toisin sanoen selain ei tiedä eroa todellisen ja jokaisella sivulatauksella erikseen muodostetun HTML-tiedoston välillä. Vaikka standardi käytäntö onkin tarjota html-tiedosto, osoitteessa on silti usein jokin muu kuin .html-pääte, kun sisältö on muuttuvaa. Tähän on useampi tapa, mutta esimerkiksi PHP on erittäin yleinen siihen tarkoitettu kieli. Yksi yleisimmistä eduista on kierrättää samaa sivun ylä- ja alaosaa muuntaen vain tiettyjä osia sivusta esimerkiksi alisivun sisällön mukaan. Mahdollisuuksia tällaisessa ympäristössä on tietenkin rajattomasti.

Erillisten toisistaan riippumattomien HTML-tiedostojen ylläpito on aikaa vaativaa ja turhauttavaa työtä. Vaikka HTML:n oppimiskäyrä ei olekaan mahdollittoman jyrkkä eikä se ole varsinaisesti ohjelmointia, vaikka sitä tavataankin kutsua "koodaamiseksi", on se silti aluksi vierasta ja hankalaa kenelle tahansa sisällöntuottajalle. Muun muassa tähän tarkoitukseen on olemassa erilaisia sisällönhal-

lintajärjestelmiä. Niiden tarkoitus on luoda abstraktio web-teknologian ja sisällön välille. Esimerkiksi bloggaajan ei tarvitse ymmärtää HTML:ää kirjoittaakseen mediaakin sisältäviä artikkeleita. Abstraktiosta on hyötyä myös niille, jotka tuottavat HTML-rakennetta työkseen, sillä hyvät sisällönhallintatyökalut ovat tehokkaampia kuin HTML:n kirjoittaminen käsin.

Monet sisällönhallintajärjestelmät, kuten WordPress ja Drupal, ovat PHP-ympäristössä toimivia alustoja. Näistä kahdesta Drupal on vieläkin laajempi abstraktio, sillä se kattaa myös monia backend-kehityksen perustoja, kuten täysin omat sisältötyypit. WordPress on suunniteltu enemmän artikkeliajatteluun, eikä sitä ole siksi tarkoitettu useiden erilaisten sisältötyyppien käsittelyyn, vaikka se onkin mahdollista erilaisten lisäosien kautta, tai ohjelmoimalla käsin teemaan tuen uusille sisältötyypeille. Tämän takia Drupal onkin oma lisäapukerroksensa myös web-kehittäjille eikä pelkästään sisällöntuottajille. Drupalin kaltaisen järjestelmän käyttäminen jokaisessa sisällönhallintaa vaativassa projektissa on kuitenkin kyseenalaista, koska se vaatii melko paljon sekä kehittäjältä että palvelimelta. Drupaliin liittyy myös tietynlainen järjestelmän leima, joten pelkästään sen perinteisen "look and feelin" parantaminen on oma työnsä.

Sisällönhallintajärjestelmän ja ylipäättään koko toteutusalueen valintaan liittyy kuitenkin myös aikaisempi kokemus. Vaikka suunniteltaisiinkin blogia eli nimenomaan WordPressin vahvuusaluetta, se ei ole paras valinta, jos kukaan ei osaa toteuttaa sitä vaaditussa ajassa realististen visuaalisten suunnitelmien mukaan. Toteutusalueen on tarkoitus mahdollistaa monimutkaisetkin toteutukselliset tarpeet lyhyessä ajassa, eikä ohjata koko suunnittelua. Drupal saattaa olla ylimitoitettu ja väärä alusta blogille, mutta paras alusta on se, joka saadaan toimimaan parhaiten juuri kyseisessä tapauksessa.

Tämän projektin toteutusalueen valintaperusteet olivat vaadittavat sisältötyypit, alueen yksinkertaisuus ja toteutuksen ketteryys. Sisältötyypeissä täytyi miettiä tarvitseeko niiden olla helposti ja selkeästi päivitettävissä, vai ovatko ne osa kiinteää sivupohjaa. Tässä projektissa sivun sisältö mietittiin kerran, ja se luokitettiin suurimmaksi osaksi staattiseksi muuttuvan sisällön ympärille. Koska ihmiset ovat kuitenkin sivun keskeinen asia, ja ne valitettavasti vaihtuvat hiljalleen jokai-

sessä yrityksessä, jätettiin se osio helposti muokattavaksi. Kaiken muun sai asettaa staattisesti sivupohjaan. Henkilöstä eritellään vain nimi, kuva ja asiakaita, joten myöskään tietorakenne ei ole monimutkainen. Pohjan staattisiin osiin vaikuttavia muutoksia, kuten toimiston muuttoa, ei tapahdu niin usein, että olisi ollut perusteltavissa rakentaa toimivaa sisällönhallintaa niillekin. Tästä on myös se etu, että sisällönhallinnan suunnittelu vie paljon aikaa eikä lopputulos ole yleensä yhtä vapautuneen ja räätälöidyn näköinen. Tämän kaltaisessa sivussa se on tärkeää.

Tunsin itse parhaiten WordPressin ja koin sen olevan kulttuurillisesti huomattavasti hienompi kuin toinen osittain hallitsemani sisällönhallintajärjestelmä Drupal. Tarkoitan tällä sitä, millaisen kuvan WordPress ja sen yhteisö antavat itsestään. Drupal kannustaa monimutkaisuuteen, mikä on tämän projektin vastakohta. Lisäksi WordPress oli myös käytännöllisempi vaihtoehto, koska suunnittelin sivua pääasiassa selaimessa ja WordPress-teeman paljaamman HTML-pohjan ketteryys oli erittäin suuri etu. Pohjan rakenne oli oletuksena myös paljon kevyempi ja selkeämpi.

Sisällönhallintajärjestelmän tiedostot sijaitsevat yleensä web-palvelimella normaaliin tapaan, mutta pelkkien tiedostojen siirtäminen ei riitä. Perinteinen vaatimus on MySQL-tietokantaohjelmisto ja PHP-ympäristö. Usein puhutaan myös LAMP-kokonaisuudesta, joka tulee termeistä Linux Apache MySQL PHP. Linux on käyttöjärjestelmä ja Apache on HTTP-palvelin, joka hoitaa HTML-tiedostojen jaon URL-osoitteiden perusteella, eli sitä voisi kutsua myös itse web-palvelinohjelmaksi. WordPress vaatii ehdottomasti ainoastaan PHP:n. Sen lisäksi tietokantaohjelmiston vaihtaminen esimerkiksi SQLiteen vaatii pieniä muutoksia, mutta muuten WordPress ei periaatteessa vaadi tismalleen Linuxia ja/tai Apachea. WordPressin saa toimimaan esimerkiksi Windowsin IIS-web-palvelinohjelmalla, mutta tuki on huonompaa eikä Windowsista ole mitään lisäetua Linuxiin verrattuna. Tavallisesti suositellussa ympäristössä WordPressin asentaminen kestää noin viisi minuuttia, jonka jälkeen voi keskittyä heti pelkääseen teemakehitykseen.

WordPressin, kuten muidenkin sisällönhallintajärjestelmien, tärkein osa on käytettävä teema. Joskus tätä kutsutaan myös pohjaksi (template). WordPressissä teema käsittää sivupohjia, funktioita, tyylitiedostot sekä muut resurssit, kuten javascript-tiedostot ja ulkoasun kuvat. Osaava henkilö voi muokata tiedostoja mieleisekseen, ja oman teeman kehittäminen toisen pohjalta on yksi tapa saavuttaa lopputulos, koska uuden teeman kirjoittaminen tyhjästä vaatii paljon PHP-, HTML-, CSS- ja erityisesti WordPress-tietämystä. WordPressin mukana tulee tätä kirjoittaessa Twentyeleven-teema, joka on toimiva oletusteema, mutta ei kuitenkaan hyvä pohja kehitykselle. Koska se on suunniteltu osin helposti muokattavaksi teemaksi, pohjatiedostot ovat täynnä sitä mahdollistavaa koodia. Samalla rakennetta muutettaessa yleensä suurin osa tyylitiedostosta muuttuu turhaksi. Sekavaa valmista tyylitiedostoa on myös ikävä muokata edelleen itselleen sopivaksi ja se hidastaa vähintään kehitystahtia.

WordPress-teemoja on olemassa todella paljon erilaisia. On olemassa myös maksullisia teemoja, jotka jatkavat helpon muokattavuuden linjaa ja saattavat sisältää myös hienoja lisäominaisuuksia, kuten kuvakarusellin etusivulla. Sivut voi toki myös toteuttaa valitsemalla vain tarpeeksi hyvin sopivan teeman, mutta harvemmin tavoitteet osuvat yhteen minkään valmiin toteutuksen kanssa, aivan kuten asiakkaan ongelmallekaan ei ole olemassa valmista ratkaisua. Tällöin paras ratkaisu on teema, joka on tehty suoraan kehityspohjaksi eikä valmiiksi tyylitellyksi paketiksi. Tämä tarkoittaa esimerkiksi täysin tyhjää tyylitiedostoa ja nipullista hyödyllisiä funktioita, joita WordPressissä ei vakiona ole.

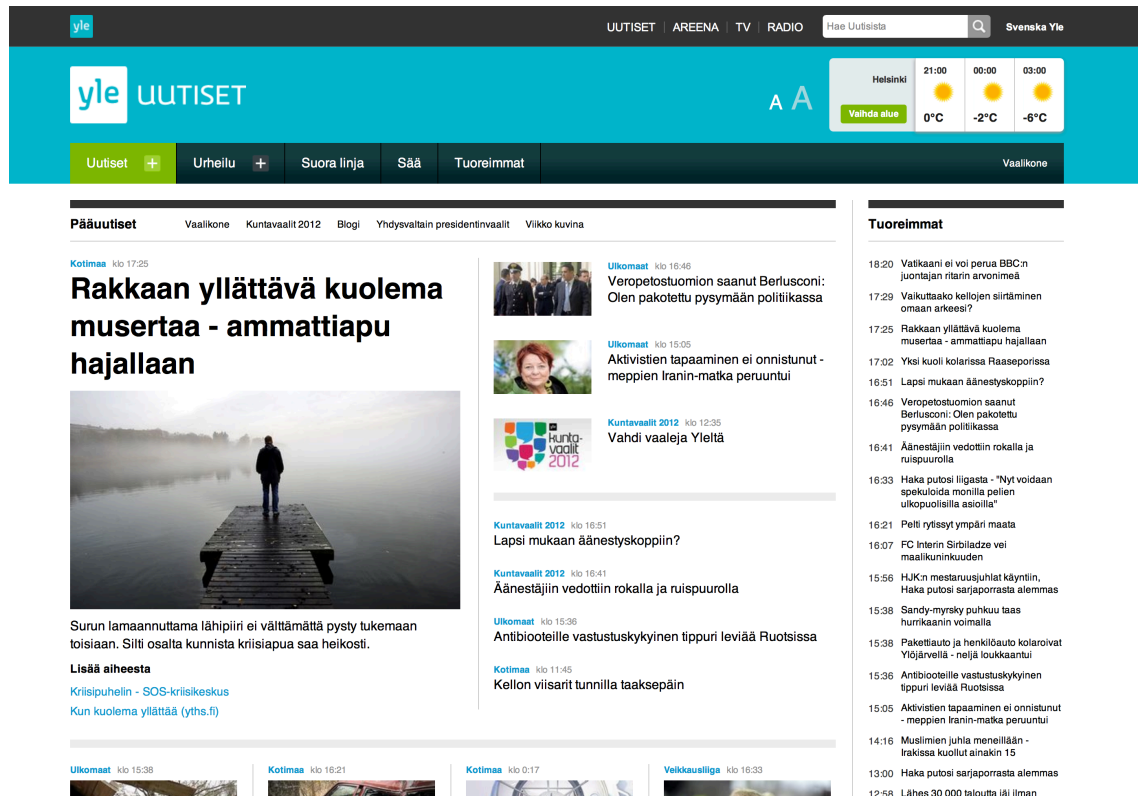
Valitsin itse Themblen Bones-teeman, joka on oletuksena melkein tyylittelemätön eli sitä ei tarvitse erikseen "siivota". Varsinainen hyöty on kuitenkin siinä, miten vähän tarvitsee siltikään tehdä itse toistuvaa työtä. Tyylitiedostoissa on valmiiksi usein käytettyjä rakenteita ilman määrittelyjä ja otsake sisältää kaikki tarvittavat määrittelyt responsiivista suunnittelua varten. Tyylitiedosto on sen lisäksi rakennettu tyylimäärittelyä helpottavista ja nopeuttavista LESS-tiedostoista, joista kerron lisää myöhemmin toteutuksen yhteydessä. Bones on moderni teema, joka on helppojen asetuksien sijaan suunniteltu olemaan mahdollisimman avoin lähtökohta täysin omanlaiselle teemalle.

3.4 Visuaalinen suunnitelma

Toteutuslupien valinta ja visuaalinen suunnittelu riippuvat toisistaan, joten järjestys ei ole niin yksiselitteinen, kuin otsikoillani annan ymmärtää. Omaan toteutuslupien valintaan vaikutti kuitenkin suuresti tavoitteista johdettu mielikuva rakenteesta ja sisällöstä. Myös lopullinen tieto tarvittavista sisältötyypeistä tulee käytännössä osittain vasta visuaalisen suunnittelun jälkeen, koska niitä saattaa tarvita lisää myös teknisessä toteutuksessa, eikä pelkän näkyvän ja itsestään selvän sisällön takia. Esimerkiksi blogiartikkeli on selkeästi oma sisältötyyppinsä, mutta etusivun vaihtuva päivän kuva tai satunnainen tviitti tietyllä Twitter-hashtagilla (kuten vaikka #kunnallisvaalit) saattavat tulla mukaan suunnitelmaan vasta rikastusvaiheessa.

Nettisivun visuaalinen suunnittelu on hankalampaa kuin esimerkiksi esitteen tai julisteen, eikä se sisällä samanlaista kontrollia kuin lehden suunnittelu: kuten totesin käyttökokemusosiossa, nettisivuille ei ole olemassa mitään yhtä tiettyä leveyttä ja korkeutta kuten printtimediassa. Erityisesti useita kertoja päivän aikana päivitettävät verkkojulkaisut elävät pienempinä kokonaisuuden muodostavina osina, ja niiden lähes jokainen rakenteen osa muuttuu jatkuvasti. Tällöin tiedon esittämisen suunnittelu ei ole enää edes täysin kontrolloitavissa, vaan se on ennakoivampaa. Jokaisessa muuttuvassa kohdassa täytyy miettiä sen sisällölle asetettavat rajat ja suunnitella rakenne joustavaksi, kun rajoja ei voi määrittää. Esimerkiksi päivän kuvan voi määrätä tiettyyn kuvakokoon ja -suhteeseen, mutta artikkelia ei. Samoin tviititkin voivat merkkirajoituksesta huolimatta kasvaa yllättävän pitkiksi riippuen rivien muotoilusta ja mahdollisista linkeistä. Tiheän päivitystahdin kannalta hallinnan pitäisi olla mahdollisimman automatisoitua. Tällaisia asioita ei ole hyvä jättää sisällöntuottajan ratkaistavaksi, koska se lisää turhaa toistuvaa työtä. Toimivassa julkaisussa sisällöntuottaja/päivittäjä voi keskittyä olennaiseen.

Hyvä esimerkki monimutkaisia palasia yhdistelevästä verkkojulkaisusta on YLE:n uutissivut (Kuva 5). Sivuilla on uutisjulkaisun tapaan päivittäin kymmeniä uutisia ja artikkeleita eri kategorioissa. Niiden esittäminen kompaktisti, selkeästi ja hierarkisesti on jatkuvaa muuttuvan sisällön muodosta riippumatta. Sivut ovat myös responsiiviset, eli koko sisältö ja rakenne sekä visuaalinen ilme säilyvät järkevästi selainikkunan tai laitteen muodosta riippumatta.



Kuva 5. YLE Uutiset -sivuston rakenne (YLE 2012).

Visuaalisen suunnitelman voi tehdä useammalla eri tavalla ja tärkeintä on käydä oma suunnitteluprosessi läpi, mutta tavanomaisesti lopullisen suunnitelman pitää olla myös esiteltävissä asiakkaalle. Yksi tapa on luoda niin kutsuttu nettileiska, josta näkee yksittäisen tai useamman sivupohjan rakenteen yhdessä mallisisällön ja visuaalisten elementtien kanssa. Lopputulos on myös helppo esitellä, koska se edustaa hyvin pitkälle varsinaista nettisivua. Tämän varjopuoli on kuitenkin se, että siinä suunnitellaan yhteen kokoon, mikä on huono tapa mukautuvan suunnittelun filosofiassa.

Sitoutumattomampi lähestymistapa suunnitellessa nettisivuja ikkunan muodosta riippumattomaksi on miettiä sisältöä ja sen muotoa, eikä pelkästään itse visuaalisen kokonaisuuden muotoa. Tällöin voi turvautua style tilesin kaltaiseen työkaluun. Style tiles tarkoittaa suoraan käännettynä tyylilaattoja eikä merkityskään ole siitä kaukana. Tarkoitus on tehdä sopivan kokoinen tyylejä esittelevä dokumentti halutulla ohjelmalla. CSS:ää osaava voi tehdä sellaisia selaimessakin, mistä on myös käytännön hyötyjä toteutuksen kannalta. Tyylilaattoja voi sitten tehdä useampia. Tarkoituksena on suunnitella muotoa sisällölle. Tarpeeksi rohkean asiakkaan voi saada hyväksymään pelkästään tyylilaatan tai kokoelman niitä ennen varsinaista nettilayoutia, mutta tärkeintä olisi prosessoida läpi eri vaihtoehtoja. (Warren 2012.)

Kun tilanne vaatii mallia kokonaisuudesta, eli niin kutsuttua nettileiskaa, se on tavallisesti tehty Adobe Photoshopilla ja esitellään sitten asiakkaalle. Rajoittavan kokonaisuusajattelun lisäksi nettileiskan ongelma on kaiken interaktiivisuuden ja animoinnin puute. Photoshop piirtää myös tekstiä huomattavasti karummin kuin selaimet. Lisäksi se ei ole kovinkaan täsmällinen tai järjestelmällinen, koska kaikki vertaukset liittyvät kuvankäsittelyyn eivätkä rakenteisiin. Jälkimmäinen on samantekevää Photoshopin tehokäyttäjälle, mutta lopputulos ei kuitenkaan edesauta itse sivun toteutusta mitenkään. Ohjelmalla suunnittelu on aina vähintään yhden askeleen irti lopputuloksesta. Osa web-suunnittelijoista kokeileekin asioita suoraan selaimessa, mikä vaatii tietenkin tarpeeksi ketteryyttä HTML- ja CSS-taidoissa. Lopputulos on kuitenkin valmista koodia, jota voi sitten upottaa sisällönhallinnan sivupohjiin tai vähintäänkin CSS-tiedostoa voi kierrättää. Lisäksi samalla saa kokeiltua interaktiivisuutta ja animointia. (Fisher 2009.)

Zeelandin esittelysivun lopullinen visuaalinen suunnitelma on hajautettu toteutuksen puolelle, mutta tein ensin niin kutsutun ”rautalangamallin” Photoshopilla (kuva 6). Tässä vaiheessa mielikuva sivun jaosta osioihin vahvistui ja sain idean pitää sivut aina koko näytön kokoisina. Rautalangan tekeminen helpottaa oman vision visualisoimista. Lisäksi se on mahdollisuus esitellä sivuston rakennetta puhtaasti ilman minkäänlaista graafista ilmettä.



Kuva 6. Ote rautalangasta.

Esitin tekemäni rautalangan kolmelle projektista vastaavalle henkilölle kahdessa eri leveydessä esitelläkseni, miten henkilöstökuvaruudukkoon tulee lisää palstoja ja navigaatio siirtyy sivuun.

Kun rautalanka hyväksyttiin, aloin toteuttaa sitä WordPress-teemaan, josta kerroin prosessina seuraavassa osiossa. Suoritin kuitenkin varsinaista graafista suunnittelua suoraan selaimessa toteutuksen aikana. Tämä kannatti erityisesti siksi, että sivun ilme koostuu suurelta osin taustakuvista, typografiasta ja animaatioista. Taustakuvien käyttäytyminen selainikkunan eri leveyksissä ja animointi kannattaa kokeilla muutenkin suoraan toteutustekniikoilla.

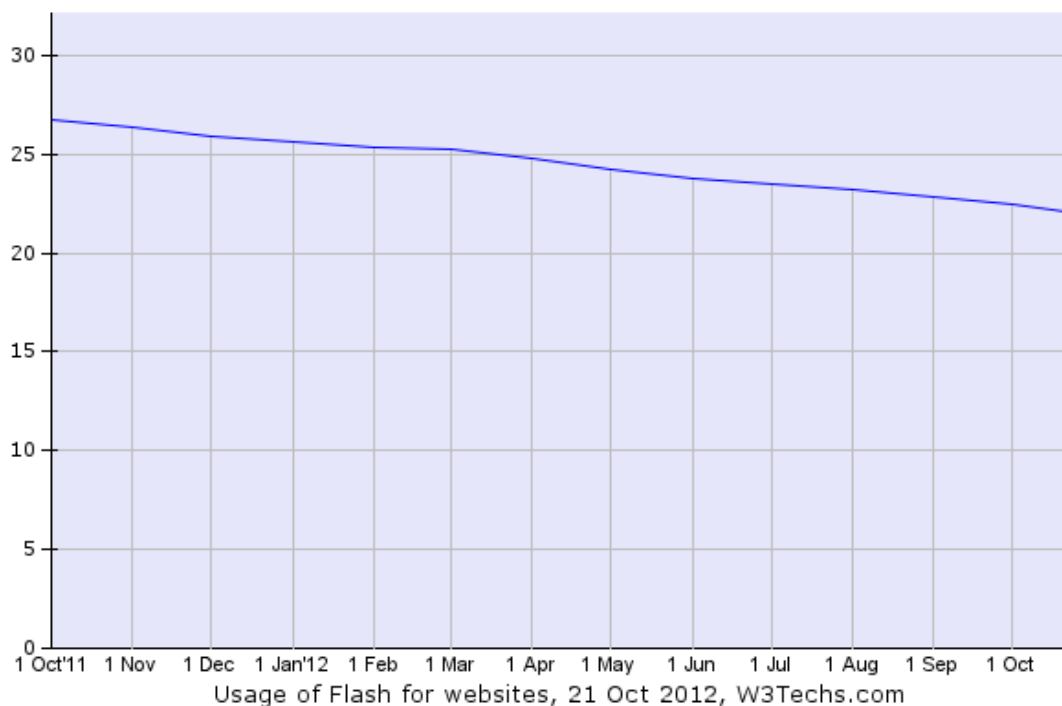
Kuvien määrä tarkoittaa kuitenkin myös sitä, että niitä piti kokeilla ja ottaa erilaisia. Henkilöstökuvien osalta taas tuli miettiä miten työntekijät esiintyvät kuvissa ja missä ne otetaan. Vääränlaiset kuvat olisivat ristiriidassa sivun muuten vapautuneemman tyylin kanssa. Esimerkkinä asialliset pukukuvat eivät missään nimessä kävisi niiden luoman tylsähkön imagoinsa takia, eivätkä ne olisi kovin rehellisiääkään.

Joidenkin osioiden taustalle tuleviin tilakuviin kokeilin erilaisia vaihtoehtoja. Olin kuvannut tätä ennen asuntoja, joten prosessi oli sikäli tuttu. Zeeland Turun tiloissa on paljon mielenkiintoisia yksityiskohtia, joten itsessään hyvien kuvien ottaminen ei olisi ollut hankalaa, mutta kuvan pitäisi toimia tummennettuna taustalla. Tämä hankaloittaa sikäli, että siinä ei saisi olla liikaa sekavia linjoja tai muita yksityiskohtia.

Taustakuvien lisäksi osioiden väleihin tulee tulevaisuudessa pienempiä polaroid-kuvien kaltaisia otteita yksityiskohdista, kuten esineistä ja ullakolle rakennetun tilan persoonallisista ominaisuuksista. Tämä onkin jo helpompaa ja pitää vain osata päättää, minkä esineiden kuvia käyttää, eikä sitäkään tarvitse tehdä vielä kuvausvaiheessa. Vauhtiin päästessä voi kasvattaa kuvapankkia myöhempää käyttöä varten. Yksittäisten henkilökuvien lisäksi tarkoitus on saada yksi hyvä ja pirteä ryhmäkuva työntekijöistä. Kaikkien ei välttämättä tarvitse näkyä siinä, kunhan se heijastaa Zeelandin ryhmähenkeä.

4 TOTEUTUS

Netti on aiemmin mainitun epävarman kankaansa takia mahdollisesti kaikkein monipuolisin media, mutta sen tuotantotyökalut ovat myös huomattavasti muita medioita jäljessä. Esimerkiksi printtimediassa on jo vuosia käytetty samoja, yleensä Adoben ohjelmia. (Trefis 2012.) Ne eivät aseta suuria rajoitteita halutulle lopputulokselle, tai haasta luovaa ihmistä liialla teknisyydellä. Web-teknologian perustoja ovat sivun rakenteen määrittelevä kuvauskieli HTML (HyperText Markup Language), tyylikieli CSS (Cascading Style Sheets) ja komentosarjakieli JavaScript, jolla voidaan ohjata selainta ja muunnella katseltavaa dokumenttia (sivua) latauksen jälkeenkin. Tämän lisäksi on yhä yleistä, että internetin lisämediat, kuten video ja monimutkaisemmat sovellukset, vaativat jonkinlaisen liitännäisen, kuten Adobe Flashin tai Microsoft Silverlightin. Näistä jälkimmäinen ei ole tosin koskaan saavuttanut hyvää asemaa ja Flashinkin osuus on laskussa (kuva 7).



Kuva 7. Flash-liitännäisen käytön osuuden lasku (W3Techs 2012).

Laskua selittävät muun muassa HTML5:n videon katseluun ja äänen kuunteleluun määrittämät standardit suoralla selaintuella ilman liitännäisiä.

HTML ja CSS toimivat tehokkaasti käytettynä yhdessä siten, että sivun rakenne ja sisältö, kuten otsikot, kappaleet ja kerrokset, määritellään HTML-tiedostossa. CSS-tiedostossa taas määritellään kaikki tyyli- ja ne pidetään kokonaan pois HTML-tiedostosta. Kyse on semantiikasta, sillä ajattelussa, jossa HTML on puhtaasti rakennetta ja CSS sivun esittämiseen käytettävä nippu sääntöjä, on selkeä sisällön ja ulkoasun erottelu toisistaan. Tällöin tarvitaan usein vain yksi tyyli-tiedosto kaikkia rakennetiedostoja varten, ja siten tyyli-tiedoston muutokset vaikuttavat suoraan selaimen tapaan esittää näitä eri tiedostoja. Ennen CSS1.0:a ja sen julkaisun jälkeenkin ainoa keino oli määritellä tyyliä suoraan HTML-tagisiin. Tämä teki yhtenäisistä muutoksista työlästä, koska määrittely piti tehdä uudelleen erikseen jokaiselle elementille ja sivulle. (Zeldman 2001.)

Selaimen ladataessa uutta sivua, se noutaa HTML:n ja lataa kaikki siinä määritellyt resurssit, kuten CSS-tiedoston, kuvat ja muut mediatiedostot. Tämän jälkeen sivu on staattinen animointia ja CSS:n pseudoluokkien määritelmiä lukuun ottamatta. Tällaisia ovat esimerkiksi hover-luokka, jolla voidaan määritellä tyyli-muutos, kun käyttäjä vie hiiren osoittimen linkin päälle. Kaikki muu vaatii selaimen erillistä ohjausta ja tätä varten on JavaScript-komentosarjakieli tai tavallisemmin ”skriptauskieli”. Sillä voidaan ohjata sivustoa latauksen jälkeenkin ja asettaa muun muassa linkille tapahtumakuuntelijan, joka ajetaan käyttäjän painaessa linkkiä.

HTML/CSS/JavaScript-yhdistelmän nykytila on mielenkiintoinen, sillä elämme murrostilassa, jossa web-kehittäjät ovat vähentämässä Flashin käyttöä korvaten sen HTML:n ja yhteisön uusilla tekniikoilla. Tarkemmin sanoen kyse ei ole niinkään pelkistä HTML:n uudistuksista, vaan kyse on enemmänkin kokonaisuudesta, jossa selaimet kykenevät paljon monipuolisempaan sisältöön kuin ennen, ja kehittäjät yhteisönä osaavat toteuttaa hyvinkin monimutkaisia kokonaisuuksia käyttäen ainoastaan HTML/CSS/JavaScript-yhdistelmää.

Kaikkia selainuudistuksia ja standardimuutoksia kutsutaan yleensä HTML5:ksi, mutta kyse on käytännössä myös CSS:n kolmosversion uudistuksista. Aivan kaikkea ei ole myöskään määritelty CSS3-standardiin, ja sellaisten ominaisuuksien käyttö ei ole kestävä. (Verou 2012.) JavaScript on lähtökohtaisesti jäykempi ja alkukantaisempi alusta kuin HTML ja CSS. Siihen on kuitenkin saatavilla paljon kehittäjää helpottavia ja siten lopputulosta parantavia kirjastoja. (Wikipedia 2012a.) Tämä laskee sen käytön kynnystä ja toisaalta vähentää huomattavasti työn määrää, kun pyörää ei tarvitse niin sanotusti keksiä uudelleen joka kerta.

HTML5:n keskeisimpiä uudistuksia ovat AV-tuki ja monipuolisempi rakenne, joka mahdollistaa yhden sivun jakamisen artikkeleihin ja sen osiin (header, section, footer). AV-tuella korvataan upotetun median liitännäisten tarve videosivustoilla, kuten Youtubessa ja Vimeossa, ja vaikka musiikkitiedostojen soittaminen Sound Cloudin kaltaisissa yhteisöissä. Tämä on erittäin huomattava parannus, koska kyse on yleensä Flash-liitännäisestä, jonka tavallinen PC-versio puuttuu mobiililaitteista teknisistä syistä ja Adobe on jo lopettanut sen mobiiliversion kehityksen (Adobe 2012). Yksi mielenkiintoinen liitännäisen kaltainen uudistus HTML5:ssä on canvas-elementti, joka mahdollistaa selaimen vektorigrafiikan piirron JavaScriptin avulla. Sillä voi teoriassa piirtää vaikka yrityksen logon reaaliajassa selaimen päässä. Käytännössä canvas-elementin käyttö tulee luultavasti olemaan paljon mielenkiintoisempaa ja luovempaa. Sillä on tehty esimerkiksi pelejä. (Canvas Demos 2012.)

CSS3:n uudistukset auttavat kehittäjiä animoimaan tyylimuutoksia siirtymien avulla. Jo pelkästään aiemmin mainitun hover-pseudoluokan aiheuttamat muutokset saadaan animoitua helposti ilman JavaScriptin käyttöä, mutta sen lisäksi CSS tukee myös pelkkää avainruutupohjaista animointia, jossa tyyliominaisuuksien arvot määritellään ajan funktiona. (Bradshaw 2012.)

Muita käytännöllisiä CSS3:n uudistuksia ovat varjot ja liukuvärit sekä 2D- ja 3D-muutokset. Jälkimmäisin mahdollistaa sivujen elementtien liikuttelun ja pyörittelyn 3D-tilassa, johon on asetettu perspektiivi. Tämä avaa paljon uusia mahdollisuuksia siirtymille ja informaation uudelleenlaiselle esittämiselle. Monia CSS3:n

kokeilevimpien ominaisuuksien käyttöönottoa hidastaa kuitenkin Microsoft Internet Explorerin Trident-moottorin hidas kehitystahti, sillä kyseinen selain on Windowsin oletusselain ja siksi hyvin käytetty. (Stat Counter 2012c.) Muun muassa siirtymät, animaatiot ja 3D-muutokset ovat tällä hetkellä vain Internet Explorer 10:n kokeiluversiossa (Wikipedia 2012b).

Siinä missä printtimediaa varten on olemassa selkeät työkalut ja helposti sisäistettävä työnkulku, on web-tekniologian tuottaminen hyvinkin paljon kehittäjän käsissä. Internet-sivustojen tuottoon on kyllä olemassa työkaluja, kuten Adobe Dreamweaver, joka kuuluu samaan Creative Suite -tuoteperheeseen Adoben printtimedian työkalujen kanssa. Kuitenkin jo jonkinlaisen sisällönhallintajärjestelmän käyttö sulkee pois Dreamweaverin kaltaisen työkalun käytön.

Adobe on kuitenkin yrittänyt tuoda web-suunnittelua vielä lähemmäs omaa taitto-ohjelmaansa InDesignia uudella Muse-ohjelmalla. Sen on tarkoitus tuntua taittamiselta, mikä on mielenkiintoinen ajatus ja omasta kokemuksesta voin kertoa sen pitävän paikkaansakin. Web-suunnittelija Elliot Jay Stocks kuitenkin tyrmää sen useista syistä. Ensinnäkin Musen vertauskuvallisuus taittamiseen on mukautuvan web-suunnittelun vastakohta. Ohjelma vaatii sivun leveyden määrittelyä uutta dokumenttia luotaessa. Toiseksi se käyttää web-fonttien sijaan vanhempaa tekniikkaa, jossa epästandardi fontti piirretään kuvaksi. Web-fontti tarkoittaa epästandardin fontin netissä jaettavaa laillista versiota. Menneisyydessä yhtenäistävän web-suunnittelijan on täytynyt valita standardi web-fontti, joiden valikoimalla ei ole juhlamista. Tällöin saatettiin kuitenkin korvata tekstiosikko kuvalla, jossa käytettiin haluttua fonttia. Muse palaa tähän unohtaen koko web-fonttien olemassaolon. Kolmanneksi Musen tuottama HTML ja CSS on täysin epäsemanttista, joten työn jatkaminen käsin jostakin kohtaa on erittäin turhauttavaa. Kaksi jälkimmäistä ja muutkin ongelmat olisi teoriassa korjattavissa ajan mittaan, kun ohjelma kehittyy, mutta taittovertaus tekee siitä lähtökohteisesti huonon työkalun tai vähintäänkin kasvattaa web-suunnittelusta kiinnostuneita väärään suuntaan. (Stocks 2011.)

Web-kehittäjän on käytännössä pakko osata tuottaa HTML- ja CSS-tiedostot käsin ilman erillistä ohjelmallista tukea. Tämä korostuu erityisesti HTML5:ttä

käytettäessä, koska sen tarjoamille uusille ominaisuuksille ei vielä ole yhtä hyviä ja yhtenäisiä ohjelmistoja. Animointitarkoituksiin on kylläkin olemassa esimerkiksi Tumult Hype, mutta sekään ei ole ratkaisu isommille kokonaisuuksille. Onneksi tiedostojen kirjoittamiseen on kuitenkin olemassa ohjelmia, jotka osaa- vat täydentää kieliä automaattisesti ja auttavat suuresti selkeyttämällä työnkul- kua. Esimerkki tällaisesta ohjelmasta on Panic Coda, joka hoitaa tarvittaessa tiedostojen versiointia (yksinkertaistettuna eräänlainen tapa pitää varmuuskopi- oita edellisistä tallennuksista) ja siirtää tiedoston automaattisesti nettisivun FTP- palvelimelle tallentaessa.

Stocks on myös todennut, että hyvän web-suunnittelijan olisi pakko osata to- teuttaa omia sivujaan. Kaikki alkoi tiivistä, jota hän kuitenkin joutui perustele- maan paremmin. Kyse on lopulta siitä, että suunnittelija osaisi tarvittaessa to- teuttaa yksinkertaisemman suunnitelman. Tämä johtuu siitä, että HTML ja CSS tai netin "koodi" on kaikkine puutteineen ja ongelmineen vahva osa suunnittelu- prosessia. Koodin asettamat rajat pitää hyväksyä ja ottaa osaksi suunnittelua eikä taistella tietämättä niitä vastaan. Lopputuloksena suunnittelijan ja frontend- kehittäjien osaamisen välillä tulisi olla sen verran päällekkäisyyttä, että molem- mat ymmärtävät toisiansa hyvin. Suunnittelija voi tällöin ajatella tietävänsä jon- kin olevan mahdollista ja järkevää, mutta hän tietää myös kollegansa toteutta- van sen paremmin ja kauniimmin kuin hän itse. (Stocks 2010.)

Työkalujen puutteen takia sisällön tuottaminen nettiin on teknistä, ja siksi perin- teisemmät graafisiin Adobein tai vaikka Corelin työkaluihin tottuneet luovat ihmi- set eivät enää kykene luomaan moderneja nettisivuja ilman kouluttautumista. Tämän lisäksi netin interaktiivinen luonto vaatii myös käytettävyyden ja käyttö- liittymäsuunnittelun ymmärrystä. Muilta osin netin visuaalisessa suunnittelussa pätee toki sama teoria kuin viimeiset sata vuotta. Hyvin mediansa tunteva ja siitä nauttiva art director voi jatkaa työskentelyä.

4.1 Toteutuslajan valmistelu

Ennen WordPressin asentamista pitää päättää, missä se tulee olemaan. Kehitystyön ja suunnittelun voi aloittaa paikallisestikin. Omassa tapauksessani Macilla tämä onnistuu MAMP:lla. Se on juuri tähän tarkoitukseen tehty ohjelma, joka luo pelkästään paikallisesti toimivan LAMP-ympäristön (korvaten lyhenteestä Linuxin Mac OS X:llä) yhdellä napin painalluksella. Paikallisesti toimiva tarkoittaa tässä tapauksessa sitä, ettei siihen oletuksena pääse kukaan netin yli. Käytin siis aluksi MAMP:ia, mutta koska työskentelen kahdelta eri koneelta, totesin tämän olevan hankalaa jo melko aikaisessa vaiheessa.

Otin seuraavaksi käyttööni Zeelandilla jo valmiiksi olleen web-hotellin. Tämä vaati oman kansion luomista FTP-palvelimella ja uuden MySQL-tietokannan luomista. Web-hotelleissa palvelinasetuksia on usein helppo säätää CPanel-käyttöliittymän kautta, ja sitä kautta pääsin myös kätevästi luomaan uuden tietokannan. Luotuani tietokannan siirsin WordPressin uusimman version FTP:lle luomaani kansioon.

Kun tiedostot ja tietokanta ovat valmiina, WordPress on helpoin asentaa velhon avulla, jonka se avaa automaattisesti, jos se ei löydä wp-config.php-tiedostoa. Velhossa määritellään tietokannan nimi, käyttäjän tunnukset ja osoite. Tietokanta sijaitsee välillä samassa osoitteessa kuin web-hotellikin, jolloin WordPressin oletus "localhost" riittää, mutta ei aina. Tietokanta-asetuksien jälkeen WordPress luo automaattisesti tarvittavat taulut tietokantaan ja kysyy lopuksi sivuston perustietoja, kuten nimen ja ylläpitäjän sähköpostin.

Viimeisenä vaiheena täytyy aktivoida käytettävä teema, ellei sitten tyydy WordPressin oletusteemaan. Valitsin jo aikaisemmin teemaksi Bonesin, joten seuraavaksi minun piti siirtää sen tiedostot WordPressin teemakansioon omaan alikansioon (wp-content/themes/bones/). Teeman asentaminen on oletusarvoisesti vain tiedostojen siirtämisen päässä, mutta tämä riippuu tietenkin täysin asennettavasta teemasta, sillä se voi muunnella WordPressin toiminnallisuutta paljonkin. Bonesin tapauksessa se vain aktivoidaan WordPressin teema-asetuksista.

4.2 Perusrakenne ja tyylimäärittely

WordPress, kuten sisällönhallintajärjestelmät yleensäkin, tekee perusrakenteen määrittelystä hieman hankalampaa kuin staattinen HTML yksinään. Oletuksena WordPress-sivu jakautuu kolmeen pääosaan: otsakepohjaan, sisältöpohjaan ja alatunnistepohjaan. Ensimmäinen ja viimeinen ovat teemakansiossa nimellä `header.php` ja `footer.php`, mutta sisältöpohja riippuu avattavan sivun luonteesta. WordPress etsii sivua vastaavaa tiedoston nimeä hierarkisesti tarkimmin määrittelystä nimestä yleisimpään nimeen. Toisin sanoen se etsii ensin sivun tunnistetta tai tynkänimeä (slug) vastaavaa (`page-[id].php/page-[slug].php`). Jos näitä ei löydy, se etsii sivulle määritellyn sivupohjan koneellista nimeä. Jos tätäkään ei löydy, WordPress valitsee laajimman pohjan eli pelkkän `page.php`:n. (WordPress Codex 2012a.)

Sivutyyppejä voi olla muitakin kuin pelkkä `page.php`, ja esimerkiksi blogiartikkelin tapauksessa kyse onkin `single.php`:stä. Tällöin sitä koskevat samat nimeämissäännöt, mutta blogin tai muun tiheämmin päivittyvän sisältötyypin tapauksessa pohja ei ehkä ole tapana tehdä yhtä täsmällisesti.

Tarinankerronnalliseen tyyliinsä liittyen nimesin esittelysivun `page-story.php`:ksi ja täsmensin tiedostossa pohjan nimeksi "Story Page Template". Tällöin WordPress löytää oikean sivupohjan, kun sisältötyyppi on sivu ja sille on asetettu "Story Page Template". Tämän jälkeen loin uuden sivun tällä sivupohjalla ja asetin sen WordPressin etusivuksi asetuksista.

WordPressin sivupohjatiedostot ovat sekoitus HTML:ää ja PHP:tä. Tämä tarkoittaa sitä, että sellaisen lukeminen on hankalampaa, koska rakenne on sekavampi kuin staattisen HTML-tiedoston. Sisältösivun perusrakenteena on vaihtuvaa sisältöä ympäröivät kerrokset, ja itse sisältö niin kutsutun "the loopin" sisällä. The loopin idea on tarkistaa, onko sisältöä olemassa, ja sitten ladata sisältö valmiiksi käyttöä varten. Funktiot, kuten `the_title()` ja `the_content()` perustuvat the loopin sisällön lataamiseen, eivätkä ne toimi ilman sitä. (WordPress Codex 2012b.) WordPressin oletusteema `twentyeleven` kutsuu vielä erikseen omaa sisältösivuaan the loopin sisällä, eli edes vaihtuvan sisällön rakennetta ei määri-

tellä samassa tiedostossa. Vaikka tämä ei olekaan pakollista, se helpottaa koodin kierrätystä. Kahdella eri sivupohjalla voisikin olla sama page.php-tiedosto, mutta eri rakenne vaihtuvalle sisällölle. (WordPress Codex 2012a.)

WordPressin tyylimäärittely perustuu tavanomaiseen CSS3:een, mutta Bones-teema käyttää esiprosessoitua CSS:ää. Vakiovaihtoehtoina siinä ovat LESS ja SASS. Nämä voi tietenkin korvata helposti jollakin muullakin kielellä tai käyttää vain CSS:ää, mutta Bonesissa molemmat vaihtoehdot sisältävät valmiiksi käytännöllisiä toistuvia kuvioita, kuten selaimien vakiotyylien nollaamisen, muuttaman mobile first -järjestyksessä aktivoituvan media queryn ja käteviä tyhjäksi jätettyjä CSS-luokkia.

4.2.1 Esiprosessoitu CSS

Valitsin itse LESSin, koska se oli minulle riittävän tuntuinen ratkaisu. Molemmat LESS ja SASS perustuvat samoihin CSS:n rakenneparannuksiin. CSS ei sisällä muuttujia ja sisäkkäitä luokkia. Tämä on suurin ongelma sen selkeydessä ja helppoudessa. Ongelmallista muuttujien puute on, jos esimerkiksi valitset yhtenäisen värin sävyn tietyille elementeille, määrittelet sen kerran CSS:ssä kaikille ja päätätkin sitten muuttaa sitä hieman jälkeinpäin. Tällöin se pitää muuttaa käsin jokaisesta määrittelystä, jolloin paraskin apu on vain tekstieditorin etsi ja korvaa -toiminto. Muuttujalla sävyn voisi määritellä yhdessä paikassa ja viitata siihen myöhemmin muissa tyylimääritelmissä. LESSin tapauksessa muuttuja on "@nimi". (LESS 2012.)

Sisäkkäiset luokat tarkoittaa rakennetta, joka tavallisesti CSS:ssä tarkoittaisi toistuvia valitsimia. Ensin saatettaisiin määritellä artikkelin linkin tyyli (Kuva 8.)

```
div#container article section.post-content a
{
    määrittelyä
}
```

Kuva 8. Artikkelin sisällön linkin valitsin.

Jos sitten haluttaisiin määritellä vielä erikseen hover-pseudoluokka a:lle, valitsin kirjoitettaiisiin uudelleen alusta alkaen omalle rivilleen (Kuva 9.)

```
div#container article section.post-content a
{
    määrittelyä
}
div#container article section.post-content a:hover
{
    lisää määrittelyä
}
```

Kuva 9. Toistuvat valitsimet.

Sisäkkäiset luokat pyrkivät selkeyttämään rakennetta ja vähentämään kirjoitettavaa syntaksia.

Sisäkkäisissä luokissa loput kohdat valitsimesta sijoitettaisiin luokan sisään. Normaalisti CSS:ssä valitsimen tyylimäärittelyt tulevat aaltosulkeiden sisään, mutta sisäkkäisessä rakenteessa niiden sisään voi laittaa myös uusia valitsimia (Kuva 10.)

```
div#container
{
    article
    {
        section.post-content
        {
            a
            {
                määrittelyä
                &:hover
                {
                    lisää määrittelyä
                }
            }
        }
    }
}
```

Kuva 10. Sisäkkäiset valitsimet.

Valitsin "&:hover" tarkoittaa nykyisen valitsimen hover-pseudoluokkaa. Toisin sanoen "&" merkitsee aina nykyistä valitsinta eli tässä tapauksessa a:ta. Jos se olisikin "a:hover", se tarkoittaisi a:n sisällä olevan a:n hoveria.

LESS:in ominaisuuksia ovat myös niin kutsutut mixinit, funktiot ja operaatiot. Mixinin tarkoitus on mahdollistaa tietyn luokan sisällön yhdistämisen toiseen. Mixinille voi myös antaa parametreja funktioiden tapaan. Tyypillisesti mixin on kätevin, kun tyyliä ei ole vielä kunnollista tukea eri selaimissa. Esimerkiksi pyöristetyt kulmat kannattaa määritellä käyttäen selainetuliitteitä, kuten "-moz" ja "-ms". (Kuva 11.)

```
.rounder-corners (@radius: 5px)
{
  border-radius: @radius;
  -webkit-border-radius: @radius;
  -moz-border-radius: @radius;
}
```

Kuva 11. Pyöreät kulmat määrittelevä mixin.

Kuvan 11 mixinille voi antaa parametrina kulman pyöreyyden, mutta se on oletuksena viisi pikseliä, jos sitä ei määritellä. Mixiniä käytetään kätevästi muun tyylimäärittelyn seassa (Kuva 12.)

```
div#container
{
  article
  {
    section.post-content
    {
      background-color: #ebebeb;
      .rounder-corners(3px);
    }
  }
}
```

Kuva 12. Tiivis tapa määritellä pyöreät reunat.

Funktiot ja operaatiot mahdollistavat arvojen määrittelyn suhteessa toisiin. Tällöin edellisessä esimerkissä kulmien pyöreys voisi olla riippuvainen toisesta

muuttujasta, kuten vaikka reunaviivojen paksuudesta. Kuvan 13 lopputulos on siis viisi pikseliä, mutta viivan paksuuden muuttaminen muuttaisi myös reunojen pyöreyyttä, mikä CSS:ssä tarkoittaisi paljon uudelleenkirjoitusta ja uusien arvojen laskentaa.

```
@general-border-width: 2px;
.rounded-corners(3px * @general-border-width);
```

Kuva 13. LESS mahdollistaa matemaattiset operaatiot.

Funktiot taas mahdollistavat monimutkaisempia laskuja ja tulostuksia. LESS:iin on sisäänrakennettuna muutamia värifunktioita, kuten vaalennusfunktio lighten(). Tälle annetaan haluttu värisävy ja se vaalentaa sitä halutulla prosentilla (Kuva 14.)

```
section.post-content
{
    background-color: lighten(#dddddd, 10%);
}
```

Kuva 14. Tyylimääritelmän arvon laskeminen funktion avulla.

Esiprosessoidun CSS:n etuja on hankala demonstroida. Sen sijaan ne huomaa paremmin käytössä. Muutaman projektin jälkeen puhtaaseen CSS:ään on turhauttavaa palata. Esiprosessoidusta CSS:stä on kuitenkin myös haittaa. Esiprosessoituna se tarvitsee nimittäin jonkinlaisen prosessoijan. LESS toimii joko selaimessa tai palvelimella, mutta tuotantovaiheessa paras ratkaisu on valmiiksi prosessoitu CSS-tiedosto, jolloin sitä ei enää tarvitse prosessoida sivua ladatessa.

CodeKit on esiprosessointiin luotu ohjelma, joka tukee LESS:iä, SASS:ia ja myös JavaScriptiä. Se seuraa projektikansiota ja prosessoi valmiin CSS-tiedoston heti, kun se havaitsee muutoksia LESS- tai SASS-tiedostoissa. CodeKit sisältää myös virheentunnistuksen ja se osaa minimoida JavaScript-tiedostoja.

CodeKit vaatii kuitenkin sivun kehittämisen paikallisesti, eli muutokset täytyy aina siirtää erikseen web-hotelliin, ellei sitten käytä paikallisesti toimivaa web-palvelinta, kuten MAMP:ia. Tämän takia selaimessa toimiva LESS-prosessori on ketterämpi ratkaisu kehitysvaiheessa, kun suorituskyky on vielä toissijaisella prioriteetilla. Silti CodeKit on kehitysvaiheessakin kätevä yhden pysähdyksen taktiikan taustalla pyörivä sovellus, jolloin monen tekniikan toteutusyksityiskohdista ei tarvitse miettiä.

4.2.2 Sivupohjan rakenne

Kuten kerroin suunnitteluosiossa, Zeelandin esittelysivu on kirjaimellisesti vain yksittäinen sivu. Tein kuitenkin selkeyden vuoksi sitä varten oman sivupohjansa, jotta tämän erikoissivun sisältö olisi omassa tiedostossaan. Koska kaikki ei ole muuttuvaa sisältöä, sivupohja on yhdistelmä staattista HTML:ää ja WordPressin sisältöhakua. Itse sivun sisällönhallinnassa olevaa sisältöä ei käytetä ollenkaan, koska tässä tapauksessa ei ollut helppoa määritellä mikä osa sivusta se olisi. Henkilöstö ja asiakkaat ovat molemmat muuttuvaa sisältöä, joten ne noudetaan erikseen WordPressin sisältöhakufunktiolla "get_posts()". Ilman tätä minun olisi pitänyt itse toteuttaa tietokantahaut ja sisällönhallinta molemmille sisältötyypeille. WordPress tai muu toteuttaa molemmat asiat tarpeeksi hyvin, eikä käsin tekeminen ole enää sen arvoista.

Zeeland Turun otsaketiedosto header.php koostuu pakollisen HTML5:n niin kutsutun boilerplate-koodin lisäksi vain Bones-teeman valmiiksi sisältävistä mobiililaitteita ohjaavista meta-tageista ja kevyestä sivulla näkyvästä konkreettisesta otsakkeesta. Tämän lisäksi PHP:ssa oleva wp_head()-funktio lisää WordPressin kautta vakiosisältöä, kuten päätyyli- ja javascript-tiedostot.

Boilerplatessa määritellään muun muassa dokumentin tyyppi (html) ja sivun otsikko. HTML5 Boilerplate -projekti pyrkii tarjoamaan parhaan mahdollisen lähtökohdan mukautuvalle web-suunnittelulle. Koska eri selaimet saattavat asettaa eri oletustyyliä, HTML5 Boilerplate sisältää muun muassa kaikki tyylimäärittelyt samaksi asettavan CSS-tiedoston. (HTML5 Boilerplate 2012.) Asteittaista rikas-

tusta varten mukana on myös Modernizr-kirjasto, joka abstrahoi monta monimutkaista selainominaisuustestiä helpoksi rajapinnaksi. Sillä voi esimerkiksi tarkistaa tukeeko selain animointia lyhyellä koodinpätkällä. (Modernizr 2012.)

Sisältösivupohja `page-story.php` kutsuu WordPressin perustoimintamallin mukaan aivan ensimmäisenä otsaketiedostoa `get_header()`-funktiolla. Tämä johtuu siitä, että WordPress avaa pelkästään sisältöpohjan, josta sitten lisätään otsake ja alatunniste sisällön lisäksi. Varsinaista sisältöä ympäröi div-elementti "content". Olen jakanut sivupohjan vielä erikseen sisältötiedostoksi Twentyeleven-oletusteeman mukaisesti, eli sivupohja tarkistaa vain onko sivua olemassa ja lisää sitten sisältötiedoston `get_template_part()`-funktiolla. Tämä on tavallisesti hyvä koodin kierrättämiseen, mutta tässä tapauksessa se korostaa ympäröivää rakennetta sisällön seasta (WordPress Codex 2012c).

Itse sisältö on jaettu HTML5-standardin mukaisesti `article`-elementin sisään osioihin eli `section`-tagien sisään. Navigoitava rakenne perustuu koko sivun korkuihin osioihin ja välikevennyksiin, jotka on merkitty poisluettavaksi eri luokalla navigointiskriptin takia. Sivulla pääotsakkeen ja sisällön välissä tai yli 1420 pikselin leveydessä sivulle tuleva navigointi perustuu näihin pääosioihin. Kerron navigoinnin toteutuksesta lisää seuraavassa osiossa. Koska `section`-tagia ei itsessään suositella käytettävän tyylimäärittelyä ja skriptausta varten, käytän koko sivun osiossa luokkaa "full-page", jolla on merkitystä sivun javascriptissä. Luokan lisäksi jokaisella koko sivun osiolla on myös id-tunnus, joista hyödytään sekä osioiden uniikissa tyylimäärittelyssä että navigointiskriptissä.

Henkilöstöosiossa käytetään ensimmäisen kerran sisällönhallinnassa olevaa sisältöä. WordPressin `wp_posts()`-funktiolla haetaan kaikki julkaistut henkilöartikkelit ja myös niiden vapaan sisällön kentät (custom fields). Jälkimmäinen on WordPressin artikkelien ominaisuus, joka mahdollistaa lisätekstikenttien luomisen. Tämä on melko yksinkertainen ja rajoittunut tapa ajatella artikkelia vapaampana sisältötyyppinä esimerkiksi Drupalin mahdollisuuksiin verrattuna, mutta sillä pärjää silti kevyissä tarpeissa. Henkilöartikkelissa vapaan sisällön kenttiä käyttäen saa listattua kätevästi työntekijän tärkeimmät asiakkaat.

Koska HTML:n rakenne-elementit eivät pääasiassa mene koskaan päällekkäin, henkilön kuvan päälle tuleva nimi vaati erikoisemman rakenteen ja CSS-tyylimäärittelyn. Kuva ja nimi ovat molemmat saman "image-frame"-luokan div-elementin sisällä, mutta nimelle on määritetty absoluuttinen sijainti. Tämän avulla sen sijainnin voi asettaa vapaasti sen sisältävän elementin reunojen perusteella.

Asiakasosiossa tapahtuu sivun toinen sisältöhaku koko sivun levyiseen Royal Sliderilla toteutettuun karuselliin. Royal Slider on Dmitry Semenovin kehittämä responsiivinen ja luonnollistakin käyttöliittymää (Natural User Interface) toteuttava jQuery-lisäosa. Siitä myydään myös erillistä WordPress-lisäosaa. (RoyalSlider 2012.) WordPress-lisäosan lisäarvo on sen helppokäyttöisyydessä, josta ei ollut varsinaisesti hyötyä minulle, koska olin jo valmiiksi tekemässä räätälöityä sivupohjaa. Huomasin tämän alun perin Eden Spiekermannin sivuilla, enkä ollut ennen nähnyt niin yksinkertaisen toimivaa kuvakarusellia. Asiakkaiden logot ovat WordPressissä omana sisältötyyppinä, vaikka ne ja työntekijät saisikin eroteltua toisistaan kategorioiden avulla. Näissä on loogisempaa käyttää eri sisältötyyppejä jo sisällönhallinnan selkeyden vuoksi. Bones-teema määrittelee vakiona uuden sisältötyypin WordPressiin, joten minun piti vain muuttaa määritelmästä kieltä henkilöstön mukaiseksi, jotta sisällönhallinnassa ei lukisi pelkästään "custom content type". WordPressissä artikkeleille ja omille sisältötyypeille on vakiona mahdollisuus asettaa pääkuva, ja se on tekstisisällön lisäksi ainut asia, jota esitän karusellissa. Molemmat näistä kuuluvat WordPressin perustoinnallisuuteen ja niiden haku tietokannasta on siksi helppo toteuttaa.

4.3 Animointi, skriptaus ja efektit

Käyttökokemusta netissä on muutettu mielenkiintoisemmaksi animoinnilla jo pitkään. GIF-kuvaformaatti mahdollisti yksinkertaiset 256-väriset minielokuvat, mutta aiemminkin mainittu Adobe Flash -liitännäinen (ennen Macromedia Flash) mahdollisti monimutkaisemmat ja interaktiiviset vektoripohjaiset animaatiot, jotka yleistyivät 2000-luvun vaihteessa. Macromedian 1990-luvulla kehittämä

Flash-teknologia kehittyi nopeasti ja ansaitsi ylivoimaisen enemmistön web-liitântöjen keskuudessa. Flash Player 5 toi ActionScript-skriptikielen, joka lisäsi Flashin potentiaalia sovellusalustana ja rohkaisti erityisesti pikkupelien tekijöitä.

Flash ei ole kuitenkaan nauttinut puhdasta suosiota, ja jo vuonna 2000 Jakob Nielsen kritisoi sitä pahoista ongelmista. Hän jakoi nämä kolmeen pääongelmaan: Flash tekee huonosta web-suunnittelusta helpompaa, rikkoo netin interaktion perustaa ja vie huomiota pois sivun ydinsisällöstä. Huonoa suunnittelua oli esimerkiksi liian pitkät ja käyttöä hidastaneet animaatiot sekä epästandardit käyttöliittymäelementit, kuten uniikit vierityspalkit. Interaktiota Flash-sivusto rikkoo irtoamalla sivuhistoriasta (ja siten selaimen edellinen-painikkeesta), eikä siitä pysty hakemaan mitään selaimen sivuhaulla. (Nielsen 2000.)

Myös Applen edesmennyt perustaja Steve Jobs on kirjoittanut kuuluisan avoimen kirjeen Flashin ongelmista kymmenen vuotta Nielsenin jälkeen. "Thoughts on Flash" keskittyy yksittäisen liitännäisen liian suureen valtaan ja tulevaisuuden mobiililaitteiden kanssa. Flash ei ole myöskään aidosti avoin sanan varsinaisessa merkityksessä, vaikka Adobe on niin antanutkin ymmärtää. Siihen liittyy myös vakavia tietoturvaongelmia. Flashia ei ole myöskään suunniteltu kosketusnäytöllisille laitteille, joten niille uudestaan tehtävät sivut voisi yhtä hyvin toteuttaa suoraan ilman liitännäisen tarvetta. (Jobs 2010.)

Jobsin kirjeessään tukema HTML5-standardi on tuonut Flashin keskeisiä ominaisuuksia suoraan selaimiin, ja osana suurempaa web-standardien kokonaisuutta se ei myöskään riko lähtökohtaisesti mitään nykyistä. Aikaisemmin mainitsemani multimedian lisäksi HTML5:n kanssa vahvasti kulkeva CSS3 on mahdollistanut natiivit siirtymät (transitions) ja animaatiot. CSS3-siirtymä tarkoittaa CSS-arvojen sulavaa muutosta ja animaatiot taas muuttavat arvoja sulavasti ajan funktiona. Näissä kahdessa on siis tärkeä ero käyttötarkoituksessa. Siirtymät säilyttävät myös tuen vanhempiin selaimiin erityisen hyvin käyttötarkoituksensa takia. Niillä ei saa muutettua sisältöä tai tyylimäärittelyä, joten väärinkäytettynäkään niillä ei saa hajotettua mitään vanhemman tai yksinkertaisemman selaimen tuesta ja asteittainen rikastus toteutuu.

Siirtymä pitää määritellä erikseen usealla eri selainetuliitteellä, mikä on hyvä todiste siitä, että CSS3 on yhä kehitysasteessa eikä selainmarkkinoiden tuki ole täydellinen. Siirtymässä annetaan sulautettava arvo (myös "all" eli kaikki käy arvoksi), siirtymän kesto ja animaation päätteiden sulautus. "Ease-in-out" tarkoittaa kiihtyvää liikettä alussa ja hidastuvaa liikettä lopussa, jolloin animaatio ei ala pongahtaen eikä pysähdy kuin seinään. (Kuva 15.)

```
-webkit-transition: background-color 0.25s ease-in-out;
-moz-transition: background-color 0.25s ease-in-out;
-ms-transition: background-color 0.25s ease-in-out;
-o-transition: background-color 0.25s ease-in-out;
transition: background-color 0.25s ease-in-out;
```

Kuva 15. CSS3-siirtymän määrittely.

CSS3-animaatioissa tilanne on hieman eri. Jos animaatiota käyttää esimerkiksi huomiota vaativassa käyttöliittymäelementissä, sille tulisi löytyä jokin korvike mikäli selain ei tue animointia. Jos animaatio aktivoidaan javascriptillä, tuen voi tarkistaa Modernizr:lla ja käyttää jQueryn animointia korvikkeena tai sitten tehdä vain yksinkertaisen muutoksen elementin luokkiin, jolloin CSS-tiedostossa voi olla animaation rinnalla "huonomman" vaihtoehdon määrittely. Olen itse huomannut CSS3:n animoinnin olevan yleensä huomattavasti sulavampi kuin jQueryn, ja CSS3 on muun muassa siksi ensisijainen vaihtoehto.

Zeeland Turun esittelysivulla siirtymiä käytetään laajan ikkunan sivunavigaatioissa ja henkilökuvissa. Navigaatioissa siirtymä on aktiivista osiota merkkeävässä valkoisessa värissä, joten kyse on pienestä yksityiskohdasta, joka kuitenkin tekee navigaatiosta hienomman. Välitön muutos navigaation aktiivisessa osiossa herättäisi vähän enemmän huomiota, mikä ei ole suinkaan aina tarkoitus. Lisäksi animaatio selventää paremmin tapahtunutta tilan muutosta. Henkilöstö-osiossa itse kuva tummenee hover-tapahtumassa ja nimi nousee kuvan yläreunaan. Näille sai myös kätevästi siirtymät, jotka lisäävät havainnollisuutta runsaasti, koska näin iso muutos saattaisi hämmentää ilman niitä. Siirtymän tarpeen takia tyylimäärittely on hieman erilainen, jos Modernizr:lla havaitaan siirtymätuen puute.

Lopulta esittelysivulla ei ole paljoa siirtymiä, mutta niitä onkin tarkoitus olla vain sen verran, että interaktion karkeat liikkeet sulautuvat. Erillistä animoitua sisältöä on kuitenkin myös olemassa. Esimerkiksi karttaosion sijaintia merkitsevä nasta tippuu kolmiulotteisen näköisesti kartalle katsojan perspektiivistä. Tämän kaltaisilla pienillä yksityiskohdilla saa mielestäni paljonkin viimeistelyn tunnetta positiivisen yllättymisen kautta.

Sivulla on kolme tärkeää javascript-ominaisuutta, jotka muuntavat käyttökoke-
musta presentaation kaltaiseksi. Koko sivun kokoisille osioille asetetaan minimi-
korkeus selainikkunan korkeuden mukaan, jolloin jokainen täyttää näkymän
vaikka sisältöä olisikin vähemmän. Käytän tässä ja muissakin dokumenttia lu-
kevissa ja muuttavissa kohdissa hyväksi erittäin suosittua jQuery-kirjastoa, joka
helpottaa huomattavasti javascriptin käyttöä. Se on nippu hyödyllisiä funktioita,
joista monen toteuttaminen olisi itsessään paljon suunnittelua ja ohjelmointiko-
kemusta vaativaa. Lisäksi jQuery tekee halutun elementin etsimisestä helppoa,
sillä se käyttää täysin samoja valitsimia kuin CSS:kin. Javascriptiin panostaessa
osaa todennäköisesti jo suurimman osan usein käytetyistä CSS-valitsimista,
mikä laskee kynnystä toteuttaa ensimmäisiä kokeiluja. Kuvan 16 lyhyt ote koo-
dista muuttaa jokaisen osion minimikorkeutta ikkunan koon mukaan.

```
$('.section.full-page').each(function() {  
    $(this).css('min-height', window.innerHeight)  
});
```

Kuva 16. Sivun osioiden vähimmäiskorkeuden asetus.

Toinen ominaisuus vastaa sivun navigaatiosta. Se toimii toki ilmeisesti käyttäen
id-tunnuksia standardiin tapaan, mutta navigaation liittyvä aktiivisen osion seu-
raaminen vaatii jo skriptauksia. Skripti perustuu vieritystapahtuman kuunteluun,
eli selain ajaa antamani funktion joka kerta, kun käyttäjä tai itse selain vierittää
sivua. Seuranta tapahtuu teoriassa pikselitasolla ja siksi funktio ei voi olla liian
raskas, jottei pikselikohtainen tarkkuus häviä. Liian raskas funktio saattaa näkyä
jopa selaimen toiminnan hidastumisena. Käytännössä muutokset täytyy tehdä

vain tarvittaessa, eikä aina sijaintiehdon täytyessä. Tällöin funktio tarkistaa vain onko selainikkunan näkymän yläreuna yhä osion vaihdoksessa asetettujen rajojen sisällä, joten dokumenttia luetaan minimaalisesti eikä sitä muuteta ollenkaan.

Kolmas ominaisuus tapahtuu osion vaihtumisen yhteydessä, eli se ei tarvitse omaa tapahtumakuuntelijaansa. Kun osio vaihtuu toiseen, uudesta etsitään aktivoitavia animaatioita sopivalla valitsimella. Olen jakanut nämä sellaisiin, jotka reagoivat joka kerta, kun osio aktivoituu ja niihin, jotka tapahtuvat vain kerran. Karttaan laskeutuva nasta on esimerkki jälkimmäisestä.

Pyrin tekemään presentaatiota parantavasta javascript-kokonaisuudesta ja sen käyttämistä HTML-luokista sen verran järjestelmällisiä, että uusien sivujen ja animoitavien osien lisääminen ei vaatisi niihin enää muutoksia. Sen lisäksi javascriptiä voisi kierrättää jatkossa vastaavissa projekteissa.

5 YHTEENVETO

Käyttökokemus web-suunnittelussa on vahvasti kiinni laiteriippumattomasta ajattelusta. Lisäksi ei voi luottaa siihen, että vierailijan selain tukisi hyvin HTML5:n paljon ja syystä mainostettuja tekniikoita. Internet Explorerin kaikkien versioiden erittäin huomattava noin 33%:n yhteenlaskettu osuus käytetyistä selaimista (noin 32%) yhdistettynä puutteelliseen HTML5- ja CSS3-tukeen osoittaa selkeän ongelman modernin web-tekniikan leviämisessä.

Rikastamalla nettisivua asteittain ja aloittamalla suunnittelu ensin mobiililaitteille, pidämme käyttökokemuksen näkyvällä jalustalla, ja ongelmat tulevat esiin selkeämmin. Korkeamman tason web-tekniikatuki isolla näyttöpinta-alalla ei saa huonontaa tai peittää nettisivun ydintoiminnallisuutta. Testaus eri laitteilla ja selaimilla sekä tekstipainotteisissa sivuissa Instapaperin kaltaisilla muotoilun riisuilla palveluilla, antaa hyvän kuvan siitä mikä on oikeasti toimivaa ja mikä ei.

Suunnittelutyön vaatima oma prosessi on jo itsessään vaikea saada hyväksi taidon ja kokemuksen kautta. Asiakas ei kuitenkaan välttämättä ole samaa mieltä päätöksistä, eikä näe asioita samasta näkökulmasta. Kukaan tuskin on ilkeyttään eri mieltä, vaan yhteistyö vaatii kaksisuuntaista kunnioitusta ja opettamista. Suunnittelijaan luottaminen vaatii rohkeutta.

Zeeland Turun esittelysivun tapauksessa olin onnekas, kun sain työskennellä tuttujen suunnittelualan ihmisten kanssa. Hyvät tulokset vaativat ennen kaikkea ryhmätyötä. Zeelandissa projektista vastaavat henkilöt auttoivat suuresti luovassa osuudessa, mutta minulla oli kuitenkin viimeinen sana web-suunnittelussa. Opin projektin aikana vastuuta, mutta toisaalta myös siitä milloin kannattaa ja täytyykin pyytää apua. Kaikilla on yleensä jokin erikoisosaamisalueensa, ja joku tietää aina paremmin. Ympäröivä osaaminen ja kokemus kannattaa siirtää lopputulokseen eikä omia työtä itselleen.

Esittelysivun toteutus oli selvästi lähempänä rutiinia, mutta sekin sisälsi uutta opittavaa. Käytin ensimmäistä kertaa esiprosessoitua CSS:ää ja yritin rikastaa

sivua asteittain, jotta se toimisi erikoisesta luonteestaan huolimatta vanhemmalakin alustalla.

Sivun vierittäminen on luonnollinen tapa levittää sisältöä laajemmalle alueelle nettisivulla. Se on saumattomampaa kuin sivun kääntäminen eikä aseta min-käänlaista kiinteää rajaa rakenteelle. Se on myös hyvä yksinkertainen laukaisin erilaisille animaatioille ja muille tapahtumille. Sen toteuttamiseen vaadittu javascriptin tarvitsi teknistä suunnittelua, jotta sivun kehitys ei vaatisi muutoksia kahdessa paikassa, eli HTML:ssä ja javascriptissä. Voisin myös kierrättää samaa pohjaa tulevaisuudessa.

Työn tuloksena on teknisesti toimiva WordPress-alusta ja suunnitelma sisällöstä, joka tuotetaan tulevaisuudessa opinnäytetyön ulkopuolella. Valokuvien ja mahdollisen kuvituksen tyyli on mietitty valmiiksi suuntaa-antavasti. Sisällön tuotto saattaa vielä muuttaa toteutuksen yksityiskohtia, koska väliaikaisella täytteellä ei löydä kaikkia epäkohtia eikä siten tiedä, miltä lopputulos näyttää ja tuntuu.

LÄHTEET

Adobe. 2011. Flash to Focus on PC Browsing and Mobile Apps; Adobe to More Aggressively Contribute to HTML5. 9.11.2011 Adobe Featured Blogs. Viitattu 21.10.2012. <http://blogs.adobe.com/conversations/2011/11/flash-focus.html>.

All about UX 2012. Viitattu 21.10.2012 <http://www.allaboutux.org/ux-definitions>.

Allsopp, J. 2000. A Dao of Web Design. 7.4.2000 A List Apart. Viitattu 21.10.2012. <http://www.alistapart.com/articles/dao/>.

Apple 2012a. Mac notebooks: Caps Lock modified to reduce accidental activation. Viitattu 21.10.2012. http://support.apple.com/kb/HT1192?viewlocale=en_US&locale=en_US.

Apple 2012b. iOS Human Interface Guidelines. Viitattu 21.10.2012. <http://developer.apple.com/library/ios/#documentation/userexperience/conceptual/mobilehig/Characteristics/Characteristics.html>.

BBC 2012. Amazon selling more Kindle ebooks than print books. 6.8.2012 BBC News Technology. Viitattu 21.10.2012. <http://www.bbc.co.uk/news/technology-19148146>.

Bradshaw, R. CSS3 Transitions, Transforms, Animation, Filters and More!. Viitattu 21.10.2012. <http://css3.bradshawenterprises.com/>.

Canvas Demos. Viitattu 21.10.2012. <http://www.canvasdemos.com/type/games/>.

Comparison of layout engines (CSS) 2012. Wikipedia. Viitattu 22.10.2012. [http://en.wikipedia.org/wiki/Comparison_of_layout_engines_\(Cascading_Style_Sheets\)#Properties](http://en.wikipedia.org/wiki/Comparison_of_layout_engines_(Cascading_Style_Sheets)#Properties).

CSS-Tricks 2010. Resolution specific stylesheets. Viitattu 21.10.2012. <http://css-tricks.com/resolution-specific-stylesheets/>.

Daring Fireball 2012. Viitattu 21.10.2012. <http://daringfireball.net/>.

Daring Fireball 2012. RSS Feed Sponsorship. Viitattu 21.10.2012. <http://daringfireball.net/feeds/sponsors/>.

Eden Spiekermann 2012. Viitattu 21.10.2012. <http://www.edenspiekermann.com>.

Fisher, M. 2009. Make Your Mockup in Markup. 24.12.2009 24 Ways. Viitattu 21.10.2012. <http://24ways.org/2009/make-your-mockup-in-markup>.

Gretzky, W. Brainy Quote. "I skate to where the puck...". Viitattu 24.10.2012. <http://www.brainyquote.com/quotes/quotes/w/waynegretz383282.html>.

Holst, C. 2010. Readability: the Optimal Line Length. 1.11.2010 Baymard Institute. Viitattu 21.10.2012. <http://baymard.com/blog/line-length-readability>.

HTML5 Boilerplate 2012. Viitattu 22.10.2012. <http://html5boilerplate.com>.

Iltalehti 27.10.2012. Viitattu 27.10.2012. <http://www.iltalehti.fi>.

Information Architects 2012. iA Writer. Viitattu 21.10.2012. <http://www.iawriter.com>.

Jobs, S. 2010. Thoughts on Flash. Apple huhtikuu 2010. Viitattu 21.10.2012. <http://www.apple.com/hotnews/thoughts-on-flash>.

- Jones B. 2010. Apple Retina Display. Jonesblog 24.6.2010. Viitattu 21.10.2012. <http://prometheus.med.utah.edu/~bwjones/2010/06/apple-retina-display/>.
- Kauppalehti 2012. Helsingin Sanomat vaihtaa tabloid-kokoon. Viitattu 21.10.2012. 1.8.2012 Kauppalehti. <http://www.kauppalehti.fi/etusivu/helsingin+sanomat+vaihtaa+tabloid-kokoon/201208225109?ext=rss>.
- LESS. Viitattu 22.10.2012. <http://lesscss.org>.
- List of JavaScript libraries 2012. Wikipedia. Viitattu 21.10.2012. http://en.wikipedia.org/wiki/List_of_JavaScript_libraries.
- Marcotte E. 2010a. Responsive Web Design. New York: A Book Apart
- Marcotte E. 2010b. Responsive Web Design. 25.5.2010 A List Apart. Viitattu 21.10.2012. <http://www.alistapart.com/articles/responsive-web-design/>.
- Marquis, M. 2012a. Responsive Images: How they Almost Worked and What We Need. 31.1.2012 A List Apart. Viitattu 21.10.2012. <http://www.alistapart.com/articles/responsive-images-how-they-almost-worked-and-what-we-need/>.
- Marquis M. 2012b. Responsive Images and Web Standards at the Turning Point. Viitattu 21.10.2012. <http://www.alistapart.com/articles/responsive-images-and-web-standards-at-the-turning-point/>.
- Modernizr 2012. Viitattu 22.10.2012. <http://modernizr.com>.
- Monteiro, M. 2012. Design is a Job. New York: A Book Apart
- Nest 2012. Living with Nest. Viitattu 21.10.2012. <http://www.nest.com/living-with-nest/>.
- Nielsen, J. 2000. Flash: 99% Bad. Useit.com 29.10.2000. Viitattu 21.10.2012. <http://www.useit.com/alertbox/20001029.html>.
- Ponchot, J. 2011. Responsive & Adaptive Web Design. 7.9.2011 Lullabot. Viitattu 21.10.2012. <http://www.lullabot.com/articles/responsive-adaptive-web-design>.
- RoyalSlider 2012. Viitattu 22.10.2012. <http://dimsemenov.com/plugins/royal-slider/>.
- Stat Counter 2012a. Top 10 Screen Resolutions Sep 2012. Viitattu 21.10.2012. <http://gs.statcounter.com/#resolution-ww-monthly-201109-201209>.
- Stat Counter 2012b. Top 14 Mobile Screen Resolutions Sep 2012. Viitattu 21.10.2012. http://gs.statcounter.com/#mobile_resolution-ww-monthly-201109-201209.
- Stat Counter 2012c. Browser Worldwide. Viitattu 22.10.2012. <http://gs.statcounter.com>.
- Stocks, E.J. 2010. Web designers who can't code. Viitattu 21.10.2012. <http://elliotjaystocks.com/blog/web-designers-who-cant-code/>.
- Stocks, E.J. 2011. Adobe Muse: a step in the wrong direction. Viitattu 21.10.2012. <http://elliotjaystocks.com/blog/adobe-muse-a-step-in-the-wrong-direction/>.
- Tapiola 2012. Suurasiakkaan esinevakuutus 2012. <http://www.tapiola.fi/NR/rdonlyres/A4C6C4B1-615E-4AB7-9A49-D8FB315AA02E/0/601Suurasiakkaanesinevakuutus.pdf>.
- The Living Internet 2012. Mosaic Web Browser History. Viitattu 24.10.2012 http://www.livinginternet.com/w/wi_mosaic.htm.

Trefis 2012. Viitattu 21.10.2012.

<https://www.trefis.com/forecastWidget?ticker=ADBE&driver=0057&comp=community&caWeekly=1&caDate=2010-08-22>.

Verou, L. 2012. Every Time You Call a Proprietary Feature "CSS3", a Kitten Dies. Viitattu 21.10.2012. <http://www.alistapart.com/articles/every-time-you-call-a-proprietary-feature-css3-a-kitten-dies/>.

Warren, S. Style Tiles. Viitattu 21.10.2012. <http://styletil.es>.

WordPress Codex 2012a. Template Hierarchy. Viitattu 21.10.2012. http://codex.wordpress.org/Template_Hierarchy.

WordPress Codex 2012b. The Loop. Viitattu 21.10.2012. http://codex.wordpress.org/The_Loop.

WordPress Codex 2012c. Function Reference/get_template_part. Viitattu 21.10.2012. http://codex.wordpress.org/Function_Reference/get_template_part.

Wroblewski, L. 2010. Touch Target Sizes. 4.5.2010 LukeW Ideation + Design. Viitattu 21.10.2012. <http://www.lukew.com/ff/entry.asp?1085>.

Wroblewski, L. 2011. Mobile First. New York: A Book Apart

YLE Uutiset 27.10.2012. Viitattu 27.10.2012. <http://www.yle.fi/uutiset/>.

Zeldman, J. 2001. From Table Hacks to CSS Layout: A Web Designer's Journey. 16.1.2001 A List Apart. Viitattu 21.10.2012. <http://www.alistapart.com/articles/journey/>.