

WYSIWYG-asettelu

Plone-sisällönhallintajärjestelmässä

Case: Tutkimushankkeen esittelysivu

Elli Manninen

Opinnäytetyö

Toukokuu 2017

Tekniikan ja liikenteen ala

Insinööri (AMK), Mediatekniikan koulutusohjelma

Tekijä(t) Manninen, Elli	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä 17.05.2017
	Sivumäärä 57	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi WYSIWYG asettelu Plone-sisällönhallintajärjestelmässä Case: Tutkimushankkeen esittelysivu		
Tutkinto-ohjelma Mediatekniikan koulutusohjelma		
Työn ohjaaja(t) Kari Niemi		
Toimeksiantaja(t) Jyväskylän yliopiston IT-palveluiden Kehittämispalvelut		
Tiivistelmä <p>Opinnäytetyön tavoitteena oli tutustua Plone-sisällönhallintajärjestelmään ja tutkia, miten Plone vastaa nykymailman sisällönhallintajärjestelmille asetettuihin haasteisiin sekä pe-rehtyä, kuinka Plonen taitto on kehittynyt vuosien aikana nykyiseen versioon. Tehtävänä oli myös toteuttaa uudistettu ulkoasu ja rakenne Jyväskylän yliopiston tutkimushankkeen esittelysivun pohjaksi Plonen Mosaic-editoria käyttäen. Työn toimeksiantajana toimi Jyväskylän yliopiston IT-palveluiden Kehittämispalvelut. Yliopistolla on vuosittain monta tutki-mushanketta käynnissä, mutta heillä ei ole yksittäistä pohjaa, minkä mukaan tutkimus-hankkeiden sivut pitäisi tehdä. Uudistuksesta tuli ajankohtainen, sillä yliopistolla oli par-haillaan käynnissä verkkosivujen päivittäminen, jonka osana on Mosaic-asettelun käyttöön ottaminen.</p> <p>Työssä tutkittiin WYSIWYGn merkitystä sekä lyhyesti sisällönhallintajärjestelmien historiaa. Työssä tarkasteltiin myös Plone-sisällönhallintajärjestelmää yleisellä tasolla sekä taiton kehityksen kannalta.</p> <p>Ulkoasupäivitys toteutettiin Plone 5 -sisällönhallintajärjestelmää varten. Tutkimushank-keen esittelysivupohjan toteuttaminen aloitettiin tutustumalla Mosaic-pohjaisen asettelun perusteisiin ja sen kautta suunniteltiin yliopiston antamat vaatimukset täyttävä asettelu.</p> <p>Opinnäytetyön tuloksena saatiin Mosaic-asettelulla tehty sivupohja yliopiston tutkimus-hankkeiden sivujen rakentamiseen Plonelle sekä käyttöohjeet sivupohjaan. Yliopiston verkkosivujen uudistuksen edetessä tutkimushankkeen esittelysivupohja otetaan jatkossa käyttöön uusia tutkimushankesivuja tehtäessä.</p>		
Avainsanat (asiasanat) Plone, sisällönhallintajärjestelmä, WYSIWYG		
Muut tiedot Liitteenä tutkimushanke esittelysivun käyttöohjeet, 14 sivua		

Author(s) Manninen, Elli	Type of publication Bachelor's thesis	Date 17.05.2017 Language of publication: Finnish
	Number of pages 57	Permission for web publication: x
Title of publication WYSIWYG layout in Plone content management system Case: Research project introduction page		
Degree programme Media Engineering		
Supervisor(s) Niemi, Kari		
Assigned by Development services of IT Services at the University of Jyväskylä		
Abstract <p>The aim of the bachelor's thesis was to get acquainted with the Plone content management system and to investigate how Plone responds to the challenges set for today's content management systems and to learn how Plone's layout designing has developed over the years in to its current version. The task was also to implement a revised layout and structure as a basis for the research project pages of the University of Jyväskylä using Plone's Mosaic -editor. The project was implemented in co-operation with the Development Services of IT Services at the University of Jyväskylä. The university has several ongoing research projects every year, however they do not have a general guide, according to which the pages of research projects should be made. Creating a guide layout came in to the picture because the university has been updating their websites continuously, which included the introduction of the Mosaic layout.</p> <p>The research studied the importance of WYSIWYG and briefly the history of content management systems. Plone content management system was also examined at a general level as well as at the layout development level.</p> <p>The layout was implemented for the Plone content management system version 5. The process to implement the project introduction page started by exploring the basics of Mosaic-based layout and by designing a layout that met the requirements of the university.</p> <p>The thesis resulted in a template for the construction of the University's research project pages using Mosaic layout for Plone. The instructions for the layout were also made. As the university's website updating is proceeding, the research project introduction page will be introduced in the future when the new research project pages are being made.</p>		
Keywords/tags (subjects) Plone, CMS, WYSIWYG		
Miscellaneous As an attachment the research project introduction page instructions, 14 pages		

Sisältö

1	Työn lähtökohdat	3
1.1	Tausta ja toimeksianto	3
1.2	Tavoitteet ja tehtävät	5
2	WYSIWYG ja sisällönhallintajärjestelmien lyhyt historia	5
2.1	WYSIWYG.....	5
2.2	Ennen sisällönhallintajärjestelmiä.....	8
2.3	Sisällönhallintajärjestelmien kehitys.....	9
2.4	Sisällönhallintajärjestelmät nykyään.....	10
3	Plone-sisällönhallintajärjestelmä	11
3.1	Plone.....	11
3.2	Historia	12
3.3	Tietoturva	13
3.4	Saavutettavuus ja käytettävyys.....	14
3.5	Hakukoneoptimointi.....	15
3.6	Responsiivisuus.....	16
3.7	Kehitysyhteisö	16
4	Plonen taiton kehitys.....	17
4.1	Yleistä	17
4.2	DTML	17
4.3	ZPT	21
4.4	Diazo	24
4.5	Mosaic	25
4.5.1	Mosaic-editori.....	26
4.5.2	Sisällön asettelun muuttaminen.....	27
4.5.3	Sisällön asettelun muokkaus	27
4.5.4	Tiilet	28

5	Tutkimushankkeen esittelysivu.....	28
5.1	Vanha tilanne.....	28
5.2	Vaatimukset.....	29
5.3	Uusi sivupohja	30
5.4	Ohjeistus jatkokäyttöön	34
6	Tulokset ja pohdinta	35
	Lähteet	37
	Liitteet	40
	Liite 1. Koodaamalla rakennetut tiilet	40
	Liite 2. Ohjeet tutkimushankesivun luomiseen ja sisällön täyttämiseen	44

Kuviot

	Kuvio 1. Plonen asettelutyylin valitseminen	26
	Kuvio 2. Mosaic-editorin näkymä.....	27
	Kuvio 3. Yliopiston Plonen testaussivu.....	30
	Kuvio 4. Yliopiston Mosaic-tiilet.....	31
	Kuvio 5. Testausteema ja yliopiston teemat.....	31
	Kuvio 6. Tiilien rakentaminen HTML-editorissa	32
	Kuvio 7. Erilliselle palvelimelle tehty asettelu.....	33
	Kuvio 8. Yliopiston ja opinnäytetyön tekijän kehittämät tiilet	33
	Kuvio 9. Tallennetut asettelutyylit	34

Taulukot

	Taulukko 1. Raportoidut haavoittuvaisuudet CVE-tietokannan mukaan	14
--	---	----

1 Työn lähtökohdat

1.1 Tausta ja toimeksianto

Pääsin työharjoittelussani tutkimaan ja kokeilemaan, miten Plonella onnistuu verkkosivujen toteuttaminen. Työharjoittelun aikana saamani perusosaamisen innoittamana halusin tutustua tarkemmin Plonen kehitykseen ja siihen, miten Plone vastaan nykypäivän verkkosivujen tekemisen haasteisiin. Halusin muun muassa saada vastaukset kysymyksiin, miten moderni asettelu tuodaan helposti sisällöntuottajan saataville ja miten Plone ratkaisee sen sekä miten ”mitä näet on mitä saat” toteutuu Plone-sisällönhallintajärjestelmässä.

Opinnäytetyön toimeksiantajana toimi Jyväskylän yliopiston IT-palveluiden Kehittämispalvelut. Kehittämispalveluiden tehtävänä on kehittää, ylläpitää ja tukea IT-palveluissa kehitettyjä sovelluksia yliopiston opiskelijoiden ja henkilökunnan käyttöön. Opinnäytetyön konkreettisena toimeksiantona oli Plone-sisällönhallintajärjestelmän Mosaic-editorilla tehtävä uudistettu ulkoasu ja rakenne Jyväskylän yliopiston tutkimushankkeen esittelysivun pohjaksi.

Jyväskylän yliopisto päätyi käyttämään Plonea sen takia, että talossa oli jo osaamista kyseisestä sisällönhallintajärjestelmästä sekä etuna oli selaimessa onnistuva muokkaaminen muihin vaihtoehtoihin verrattuna. Yliopistolla on yli 10 vuoden kokemus Plonen käytöstä ja kehityksestä. Yliopistolla alkoi vuonna 2001 erikseen rahoitettu hanke nimeltä Virtuaaliyliopisto, jonka tavoite oli kehittää verkkopohjaisia ratkaisuja opettamiseen ja opiskeluun. Vuonna 2003 yliopisto julkaisi Zopea käyttäen heille hankkeessa kehitetyt Moniviestin-videojulkaisualustan, muutaman verkkosivun ja Kirjoitusstudion. Zopen käyttö siihen aikaan perustui siihen, että Zope oli sovelluspalvelin, jonka päälle rakennettiin sovelluksia Python-kieltä käyttäen. Plone oli aluksi yksi sovelluksista, joita käytettiin Zopen päällä. Plone laitettiin viimeiseksi kerrokseksi palvelimen ja sovelluksen päälle käyttöliittymäkerrokseksi. Siihen aikaan yliopiston sivut olivat staattisia ja ne olivat hyvin hajautuneet, koska kaikilla oli omat sivunsa. Jyväskylän yliopiston varsinainen historia nykyisen sisällönhallintajärjestelmä Plonen kanssa alkoi vuonna 2004 yliopiston ensimmäisellä Plone sivustolla, joka oli henkilökönlön koulutusportaali. (Oksanen & Soukka 2017.)

Tietojärjestelmäarkkitehti Jussi Talaskivi tutustui vuonna 2002 Zopeen ja vuonna 2004 Ploneen, mikä mahdollisti yliopiston ensimmäisen Plone-sivuston kehittämisen. Hän teki töitä Rikupekka Oksasen kanssa Virtuaaliyliopisto-hankkeen parissa. Tammi-kuussa vuonna 2005 humanistinen tiedekunta, IT-tiedekunta ja kasvatustieteiden tiedekunta halusivat päivittää sivunsa Ploneen. Yliopiston viestintäyksikkö kehitti teknisenä apunaan Virtuaaliyliopistohanke ryhmän kanssa sivujen rakenteen alkuvaatimuksien ja muutaman tapaamisen pohjalta. Sivujen kehitys jatkui sisällön tuottamisena sekä henkilöstön kouluttamisena Plonen käyttöön, ja toukokuussa ensimmäinen tiedekuntien sivuista oli valmis. Syksyllä yliopistolle valmistui uusi etusivu, ja kuusi muuta tiedekuntaa saivat uudet sivut vuoden 2005 loppuun mennessä. (Oksanen 2014, 39-53.)

Kaikkien loppujen tiedekuntien sivut päivitettiin vuonna 2006. Vuonna 2007 yliopiston sivut hajautettiin ja kaikkiin sivuihin käytettiin samaa teemaa. Vuodesta 2007 eteenpäin yliopistolla alkoi omien Plone-lisäosien kehitys. Ensimmäisiä kehitettyjä lisäosia oli Dynapage, jota käytettiin etusivujen tekemiseen. Se sisälsi automaattisesti päivittyvät uutiset ja kuvakaruseelin. Ensimmäistä versiota käytettiin laitosetusivuna ja yliopiston etusivuun tehtiin räätälöidyn versio, jota ei pystynyt muokkaamaan. Vuonna 2009 teema päivitettiin, mutta sisältö pysyi samana. Ensimmäinen versio Portalview-lisäosasta otettiin käyttöön. Lisäosa mahdollisti portaalinäkymän, jolla pystyi tekemään laatikkoasetteluja. Portaalinäkymää käyttämällä lopputuloksesta saatiin aika lailla samanlainen kuin, mitä sivuilta oli haluttu. Portaalinäkymän ylläpito oli vaikeampaa, ja muokkaamisen osaaminen vaati koulutusta, mutta sivuun pystyi tällä tavalla muun muassa tuomaan muiden sivujen sisältöä ja tekemään listoja laatikoiden sisään. (Oksanen & Soukka 2017.)

Yliopisto on pysynyt Plonen parissa kaikki nämä vuodet, koska Plone itsessään on pysynyt markkinoilla ja kehittynyt tasaiseen tahtiin, mihin yliopiston IT-palveluiden Kehittämispalvelutkin ovat vaikuttaneet omalla kehittämistyöllään. Plone on ilmainen vaihtoehto avoimen lähdekoodin sisällönhallintajärjestelmänä, ja sitä ei pysty kukaan ostamaan pois markkinoilta. Näin se tulee jatkossakin olemaan käytettävissä. Yliopistolla on oltu tyytyväisiä mahdollisuuteen räätälöidä kaikki tarpeen mukaan, ja yliopistolla onkin hyvin monta räätälöityä ratkaisua tehtynä. Plone on hyvin monessa paikassa yliopistolla käytössä, muuallakin kuin vaan verkkosivuilla, kuten oppimateri-

aali-Koppa, intranet, maksupalvelut ja Moniviestin, joten loppujen lopuksi vaihtaminen Plonesta pois olisi hyvin hankalaa. (Mt.)

Muita yliopiston kehittämiä lisäosia ovat muun muassa vieläkin käytössä olevat Tutka sivu, jossa on lista julkaisuista, ja henkilökuntarosteri, jossa on lista henkilökunnasta. Viime vuodesta alkaen yliopistolla on ollut käynnissä verkkosivujen uudistus päätavoitteena uusi ilme sekä myös yhtenäinen ulkoasu ja helpompi käytettävyys, jonka osana on Mosaic-asettelun käyttöön ottaminen. Mosaic-asettelulla pystyy toteuttamaan saman ulkoasun kuin Dynapagella ja Portalviewillä, joten niiden käyttö ei enää ole pakollista. (Mt.)

1.2 Tavoitteet ja tehtävät

Opinnäytetyön tavoitteena oli tutustua Plone-sisällönhallintajärjestelmään ja tutkia, miten Plone vastaa nyky maailman sisällönhallintajärjestelmille asetettuihin haasteisiin sekä perehtyä, kuinka Plonen taitto on kehittynyt vuosien aikana nykyiseen versioonsa. Samalla oli tarkoitus toteuttaa uudistettu ulkoasu ja rakenne Jyväskylän yliopiston tutkimushankkeen esittelysivulle ja tällä tavoin todentaa tutkimuksen aikana havaitut asiat.

2 WYSIWYG ja sisällönhallintajärjestelmien lyhyt historia

2.1 WYSIWYG

”Mitä näet on mitä saat” what you see is what you get lyhennettynä WYSIWYG viittaa nykyään (HyperText Markup Language) HTML-editoriin, jossa muokattava sisältö näyttää samalta kuin sivu julkaistussa muodossa. Toisin kuin perinteiset editorit WYSIWYG keskittyy lopputulokseen, minkä avulla saa selkeämmän kuvan siitä, mitä luo samalla, kuin luo sitä. WYSIWYG-editorien kehityksen alusta alkaen niiden tarkoituksena on ollut yksinkertaistaa käyttäjän sisällön tuottamista. (Stec 2015.)

Nykyään termistä on tullut synonyymi verkkoeditointikokemukselle, ja muitakin samankaltaisia termejä on olemassa. ”Mitä näet on enemmän tai vähemmän mitä saat” what you see is more or less what you get lyhennettynä WYSIMOLWYG ja ”mitä

näet on melkein mitä saat” what you see is almost what you get lyhennettynä WYSIAWYG. (Stec 2015.) Näitä termejä käytetään kuvaamaan editorien todellista rajallisuutta, sillä hyvin harva editori oikeasti on täysin WYSIWYG toiminnoiltaan.

Ennen kuin WYSIWYG-termiä käytettiin kuvaamaan tekniikkaa, jonka ansiosta käyttäjät voivat visualisoida, miltä lopputuote voisi näyttää, termi ei liittynyt mitenkään tietotekniikkaan. Termi tuli tiettävästi ensimmäisen kerran ihmisten tietoisuuteen Flip Wilson-nimisen näyttelijäkoomikon drag-persoona Geraldinen iskulauseena. Hän esiintyi ensimmäisen kerran Geraldinena syyskuussa 1969 ja sen jälkeen säännöllisesti 1970-luvun alussa The Flip Wilson Show’ssa. Termiä ”mitä näet on mitä saat” käytettiin vaatimuksena hyväksyä Geraldinen koko persoonallisuus ja ulkonäkö. (Watkins 1998.)

Tekniikkaan termin tulosta on monta eri versiota, mutta aikaisin dokumentoitu WYSIWYG-tekniikoiden käyttö löytyy vuodelta 1968. Tietojenkäsittelytiedemies Douglas Engelbart piti tuote-esittelyn San Franciscon Brooks salissa 9.11.1968, jossa hän debytoi ensimmäisen version nykyäänkin jokaisella käytössä olevasta tietokonehiirestä. Hiiri toimi täsmälleen samalla tavalla kuin nykyajan hiiret, ja Engelbart käytti sitä esitellessään muut demonsa sisältämät asiat: "mitä näet on mitä saat" muokkaamisen, tekstin ja grafiikan näyttämisen yhdessä näytössä, jaetun näytön videoneuvottelun, ikkunat, versionhallinnan, tilannekohtaisen ohjeistuksen ja hyperlinkit. Demo oli todella edellä aikaansa maailmassa, jossa oli totuttu siihen, että tietokoneet olivat persoonattomia laatikoita, jotka lukivat reikäkortteja ja vähän ajan surinan jälkeen sylkivät ulos kasoittain teleprintteripaperia. Demo innoitti sukupolven teknologiaosaajia, ja demo tunnetaan nykyään nimellä The Mother of All Demos ”Kaikkien Demojen äiti”. (The Demo: 1968 n.d.)

Termin varsinaisesta ensimmäisestä käyttökerrasta teknologiassa on muutamia eri tarinoita, mutta WYSIWYG-ohjelmien kehityksen kannalta tärkein oli ensimmäisen WYSIWYG-asiakirjan valmisteluohjelma Bravon kehittäminen vuonna 1974. Bravo otettiin käyttöön ensimmäiseen täysin verkotettuun tietokoneeseen, Xerox Alto, joka kehitettiin Xerox PARCissa vuonna 1972. Xerox Altoa ei koskaan julkisesti markkinoitu, joten Bravo oli käytössä vain sisäisesti Xeroxin PARCissa. (Bravo (software) n.d.)

Charles Simonyi, Bravon toinen pääkehittäjästä, liittyi Microsoftiin vuonna 1981, ja hän aloitti kehittämään ohjelmistokehittäjä Richard Brodien kanssa WYSIWYG tekstinkäsittelyohjelmaa nimeltään Multi-Tool Word, joka myöhemmin nimettiin Microsoft Wordiksi vuonna 1983. Nykyisestä suosiostaan huolimatta Microsoft Word ei ollut ainoa laatuaan, kun se julkaistiin, useita muita työkaluja oli jo markkinoilla. (Microsoft Word n.d.)

WordStar oli ensimmäinen kypsä WYSIWYG-tekstinkäsittelyohjelma, jonka MicroPRO julkaisi vuonna 1979. MicroPro palkkasi John Robbins Barnabyn syyskuussa 1978, ja hän ohjelmoi yksinään WordStarin ensimmäisen version silloin nimeltään WordMaster. (WordStar n.d.)

WYSIWYG-ohjelmien omaksuminen 1970- ja 1980-luvun alussa oli hidasta epäkypsien grafiikkaominaisuuksia ja bittikarttanäyttöjen takia. Kuitenkin kyseisellä aikakaudella tapahtuneella kehityksellä oli iso merkitys WYSIWYG-ohjelmien tulevaisuuden näkymään. Kotitietokoneiden alkaessa yleistyä myös WYSIWYG-ohjelmia alkoi näkyä enemmän markkinoilla. (Stec 2015.)

Ensimmäinen verkkosivusto julkaistiin vuonna 1991 internettiin ja siitä lähti käyntiin HTML-kielen käyttö verkkosivujen koodauskielenä. HTML-kielen kehittyessä HTML-editorienkin piti alkaa monipuolistua. Tavoitteena oli luoda muokkauskäyttöliittymä, joka teki verkkosivujen rakentamisen helpoksi myös niille ihmisille, jotka eivät olleet perehtyneet hyvin HTML-kieleen. (Mt.)

Ensimmäisen WYSIWYG HTML-editorin nimeltä WebMagic julkaisi Silicon Graphics vuoden 1995 tammikuussa. WebMagicin rakensi pieni ryhmä osajia alle 80 päivässä käyttäen pohjana Amdahl yrityksellä kehityksen alla ollutta HTML-editoria. Saatuaan rahoituksen Silicon Graphicsin johtajalta ja operatiiviselta johtajalta John McCrea alkoi kehittää ideaansa tuotelinjasta nettipalveluita varten. John McCrea käynnisti projektin Way Tingin avulla, joka auttoi saamaan sopimuksen IBM-yhteensopivien keskustietokoneiden tekijän Amdahl kanssa. Amdahlilta he saivat toimivan koodin sekä kehittäjä David Koplasiin. (McCrea 2014.)

Vaikka WebMagic oli ensimmäinen WYSIWYG HTML-editori, saman vuoden lokakuussa Vermeer Technologies julkaisi FrontPage-nimisen editorin. Vuonna 1996 Mic-

rosoft osti Vermeer Technologiesin ja tämän takia FrontPage toimi ensimmäisenä WYSIWYG HTML-editorina Windowsissa. FrontPage oli arvokas etu Windowsille sen aikaisessa selainsodassa NetScapea vastaan, ja FrontPagen tarkoituksena oli luoda verkkosivuja Microsoftin omaan selaimeen Internet Exploreriin. (Forgaard 1996.)

Monia muitakin WYSIWYG HTML-editoreja alkoi tulla niihin aikoihin markkinoille. Nykyään hyvin tunnetun Dreamviewerin ensimmäisen version julkaisi yritys nimeltä Macromedia vuonna 1997, mutta osa sen juurista alkoi jo vuodesta 1995, jolloin Nick Bradbury kirjoitti HomeSite-nimisen editorin. HomeSite ei ollut WYSIWYG HTML-editori, se keskittyi verkkosivujen rakentamiseen suoraan koodaamalla. Maaliskuussa 1997 Allaire Corporation hankki HomeSiten ja Nick Bradbury liittyi Allaireen. Macromedia puolestaan hyvän yhteistyön jälkeen vuonna 2001 yhdisti Allaire Corporationin itseensä ja samalla yhdisti HomeSiten ominaisuuksia Dreamweaveriin. Vuoteen 2005 mennessä Macromedian osti Adobe ja yhdessä oikeudet Dreamweaver ohjelmaan. Sitten Adobe on säilyttänyt Dreamweaver-kehitysohjelmiston tuotteenaan, joka nykyään osa Adoben Creative Cloudia. (Bradbury 2009; Macromedia HomeSite n.d.)

Vuonna 1999 ilmiö blogien pitämisestä kasvatti suosiota. Sinä vuonna julkaistiin muutaman tunnetumpaa blogisivustoja, Blogger ja LiveJournal. Blogisivustoja voidaan pitää yhtenä sisällönhallintajärjestelmien alkumuodoista. LiveJournal ja Blogger ovat molemmat vielä käytössä, vaikkakin jälkimmäisen osti Google vuonna 2003. Vuonna 2003 Matt Mullenweg ja Mike Little julkaisivat ensimmäisen version WordPress-sisällönhallintajärjestelmästä, joka oli jatkokehitetty vuonna 2001 julkaistusta blogi sivustosta nimeltä b2 cafelog. (Blog n.d.)

2.2 Ennen sisällönhallintajärjestelmiä

Ennen sisällönhallintajärjestelmiä verkkosivut tehtiin koodaamalla sivut käyttäen HTML-merkkäuskieltä. Sivujen pääpaino oli saada tieto näkyviin ja toimia vaaditusti. Sivujen rakentaminen onnistui vain koodauskielten osaajilta, joten sivujen tekemisen yleensä hoiti joku asiaan palkattu osaaja. Tiedostot kirjoitettiin yksinkertaisella tekstieditorilla manuaalisesti ja sen jälkeen ne julkaistiin staattisena sivustona palvelimelle. Virheiden korjaus ja sivujen päivitys vaativat koodin muuttamisen editorissa sekä tiedostojen uudelleen julkaisemisen palvelimelle. (De Libero 2016.)

2.3 Sisällönhallintajärjestelmien kehitys

Ensimmäiset sisällönhallintajärjestelmät eli CMS:t (Content Management System) rakennettiin 90-luvulla tavoitteena helpottaa verkkosivujen rakentamista ja hallitsemista. Sisällönhallintajärjestelmien kehitys alkoi koodaajien toimesta, kun monet ihmiset halusivat, että sivujen omistajien olisi hyvä päästä muokkaamaan sivujensa sisältöä kyseisten sivujen sisältä. CMS:n alkuaikoina yleistä oli, että ammattilaisilla oli käytössä itse tehty CMSn, joten niiden määrä oli aluksi suuri.

Suuremmat sisällönhallintajärjestelmät saivat alkunsa 90-luvun lopussa ja 2000-luvun alkupuolella. Muiden kielten, kuten PHP vuonna 1995, ilmaantuessa käyttöön verkkosivujen ja sisällönhallintajärjestelmien kehittäminen helpottui. 2000-luvun alkuvuosina avoimen lähdekoodin sisällönhallintajärjestelmiä, kuten Drupal ja Joomla!, alkoi myös syntyä. Vaikka nämä sisällönhallintajärjestelmät usein tarvitsivat hieman räätälöintiä vastaamaan suunnitelmaa, sisältöä ja markkinointiorganisaatioiden käyttötarpeita, sivustot pystyttiin sisällönhallintajärjestelmää käyttämällä luomaan muutamissa tunneissa tavallisen editorin kanssa vaadittujen viikkojen tai kuukausien sijaan. CMS:n rakenteet ja ominaisuudet olivat aluksi aika yksinkertaisia, mutta kun internetteknologiat kehittyivät ja ihmisten tarpeet kasvoivat, kehittyivät myös sisällönhallintajärjestelmät tarpeiden mukaan. (De Libero 2016.)

Sisällönhallintajärjestelmien ominaisuudet lisääntyivät, ja seuraava vaihe CMS:n kehityksessä on piirtänyt olennaiset kriteerit sille, mitä CMS-alustan tulisi sisältää. Kehittäjät ottivat huomioon olennaisia tekijöitä kuten helppokäyttöisyys, joustavuus ja ominaisuuksien runsaus. Silloin perustettuihin CMS-alustoihin kuuluivat jo seuraavat asiat:

- tietokanta
- työnkulun automatisointi
- sisään/ulos kirjaudu toiminnallisuus
- WYSIWYG-editori (visuaalinen editor)
- aikataulutus
- linkkien hallinnointi
- metadata

(Nimchynska n.d.)

Vuonna 2003 CMS nimeltään WordPress tuli markkinoille ja sen plugin-arkkitehtuurin avulla se herätti nopeasti kehitys yhteisöjärjestöjen kiinnostuksen. WordPress oli hy-

vin yksinkertainen toiminnollisuuksiltaan niin kuin suurin osa sen ajan käytössä olevista CMS:tä. WordPress menestyi sen takia, että se mahdollisti rajattomat muokkaukset ja kolmannen osapuolen kehittämät laajennukset. Mutta silti näiden työkalujen käyttäjinä toimi vielä hyvin teknologiaa hyödyntävät osaajat. Enää ei tarvittu kovan linjan insinööriä perustamaan sivustoa, mutta tavallinen käyttäjä ei siihen yksin vielä pystynyt. WordPress käytti tyyppillisesti Web-suunnittelijat, jotka olivat oppineet koodaamaan tai olivat itseoppineita kehittäjiä. (Marohnić 2014.)

Nopean internet yhteyden levitessä useimpiin kehittyneisiin maihin, yritykset alkoivat tarvita verkkosivuja nopeasti. Tämä vaatimus johti useisiin eri tavalla kehitettyihin CMS:iin, joita pyrittiin kohdistamaan eri käyttäjäkunnille: bloggaajille, taiteilijoille ja oikeastaan kenelle tahansa, ketkä halusivat olla esillä verkossa. Nykyään se tarkoittaa lähes jokaista elävää ihmistä. Uusia tämän tapaisia sisällönhallintajärjestelmä työkaluja olivat Squarespace tai WIX. Ne rakennettiin henkilöille, joilla ei ollut koodaus taitoja ja/tai olivat juuri alkaneet käyttää tietokoneita. Kaikki sisällönhallintajärjestelmät alkoivat mainostaa kuinka yksinkertaisia, intuitiivisia ja helppokäyttöisiä heidän CMS:t olivat. Niihin aikoihin WordPress muutti myös markkinointiviestiään kotisivuillaan ja alkoi kohdistaa itseään enemmän suurelle yleisölle. (Mt.)

2.4 Sisällönhallintajärjestelmät nykyään

Nykyään periaatteessa kuka vain pystyy rakentamaan itselleen, vaikka kotisivut helposti ja vaivattomasti valitsemalla jonkin markkinoilla olevan sisällönhallintajärjestelmän. Markkinoilla on valittavissa monia ilmaisia, useammat jopa avoimen lähdekoodin sisältäviä, sisällönhallintajärjestelmiä sekä maksullisia lisäominaisuuksia ja tukea sisältäviä sisällönhallintajärjestelmiä. Käyttäjien ei tarvitse tietää ohjelmointikielistä ja koodaamisesta mitään, kun he voivat rakentaa ja räätälöidä helposti kotisivut itselleen käyttöliittymän kautta. Sisällönhallintajärjestelmien vaatimuksien määrä on kasvanut käyttäjämäärien kasvaessa, ja ominaisuuksissa on otettava huomioon erilaiset ihmiset.

Muutamia ominaisuuksia, joiden mukaan käyttäjät valitsevat nykyään sisällönhallintajärjestelmänsä:

- Haku – Millainen sisäinen hakuominaisuus on ja miten tieto noudetaan?

- Versiointi – Luoko järjestelmä varmuuskopioita sisällöstäsi? Puhdistetaanko sisältöä missään vaiheessa ja jos puhdistetaan niin, milloin?
- Sisällön muokkaaminen - Jos kaksi käyttäjää työskentelee samaan aikaan sisällön kimpussa, korvautuuko muutokset vai tuleeko sisällöstä olemaan kaksi versiota tallennettuna, joita verrataan, kun sisällön muokkaajat tajuavat, että sisältöä on muokattu useammasta paikkaa.
- Sisällön hyväksyminen – Onko olemassa työnkulun hallintaa, jossa sisältö täytyy hyväksyttäväksi toimittajalta?
- Skaalautuvuus ja laajennettavuus - Onko sisällönhallintajärjestelmän takana aktiivinen kehitys tiimi tai yhteisö kehittämässä ohjelmistoa niin, että se voi hyödyntää uusia kehitettyjä web-teknologioita? Mitkä muut yritykset tai organisaatiot käyttävät tätä tuotetta?
- Sisällön ajastus /vanhentuminen - Voidaanko sisältöä ajoittaa julkaisemaan automaattisesti ja poistumaan automaattisesti tietyssä päivämääränä ja kellonaikana?
- Monikielisyys - Onko sisällönhallintajärjestelmään tukea useammilla kielillä? Tukeeko se sitä hyvin?

Nykyään suurimmassa osassa on käytössä WYSIWYG-ominaisuuksia, mutta kaikki CMS:t eivät ole täysin WYSIWYG. WYSIWYG toteutuu eri sisällönhallintajärjestelmissä eri tasoilla: jotkut ovat täysin WYSIWYG-sisällönhallintajärjestelmiä ja osassa lisäosat vain toteuttavat WYSIWYGiä.

Wordpress on nykyään käytetyin sisällönhallintajärjestelmä ja blogisivusto. Drupal, Joomla! ja Blogger ovat myös todella suosittuja sisällönhallintajärjestelmiä nykyäänkin. (CMS Usage Statistics 2017.)

3 Plone-sisällönhallintajärjestelmä

3.1 Plone

Plone on ilmainen ja avoimen lähdekoodin sisällönhallintajärjestelmä, joka on rakennettu Zope-sovelluspalvelimen päälle käyttäen Pythonia. Plonea useimmiten käytetään isojen organisaatioiden intranettina ja verkkosivujen tekemiseen. Plonea käyttävät muun muassa organisaatiot kuten NASA, Amnesty International, eBay ja Nokia. (What is Plone? n.d.)

Plonen kehitys alkoi vuonna 2001, ja se on siitä lähtien ollut Plone yhteisön aktiivisessa kehityksessä melkein 900 koodin tuottajan voimin. Ploneen tuotettujen koodirivien määrä vastaa arviolta työtunneissa noin 350 vuotta. Plone on saatavilla yli 40 kielellä, ja sillä on hyvä turvallisuusluokitus verrattuna muihin merkittäviin CMS-

järjestelmiin. Plonen kehityksestä vastaa siis Plone-yhteisö, mutta vuonna 2004 perustettu voittoa tavoittelematon Plone-säätiö Plone Foundation omistaa Plonen immateriaalioikeudet ja tavaramerkit. Säätiö perustettiin turvaamaan Plonen säilyminen avoimen lähdekoodin sisällönhallintajärjestelmänä ja tukemaan Plone-yhteisön toimintaa. (About Plone n.d.)

Plonen avulla ei-tekniset ihmisetkin, jotka eivät tiedä koodaamisesta mitään, pystyvät luomaan ja ylläpitämään verkkosivuja. Plone on helppo ymmärtää ja käyttää, ja sen avulla käyttäjät voivat luoda verkkosivut muutamissa tunneissa, mutta silti se tarjoaa runsaan määrän aktiivisen yhteisön kehittämiä lisäosia ja laajennettavuutta tulevien vuosienkin käytön varalle. (What is Plone? n.d.)

Plone nimenä tulee saman nimiseltä bändiltä Plone, joka oli olemassa Warp nimisellä levy-yhtiöllä. Plone-sisällönhallintajärjestelmän kehittäjät Alan Runyan ja Alexander Limi kuuntelivat Plone bändiä, kun he tapasivat ja tekivät Plonen alkukoodausta sekä suunnittelua. (What does Plone mean? How is it pronounced? n.d.)

3.2 Historia

Plone projekti alkoi Alexander Limin, Alan Runyanin ja Vidar Andersen toimesta. Alan Runyan tapasi Alexander Limin Zopen keskustelupalstalla, ja he olivat molemmat sitä mieltä, että Zope ei ollut tarpeeksi hyvä käytettävyydeltään, joten he alkoivat kehittää parempaa käytettävyykokemusta Zope-sisällönhallintajärjestelmä ohjelmistonkehityksen (Zope Content Management Framework) päälle uudeksi käyttöliittymäksi. Vidar Andersen kehitti ensimmäisen suunnitelman käyttöliittymälle, Alexander Limi muutti suunnitelman HTML-koodiksi ja Alan Runyan integroi HTML-koodin Zopeen. Ensimmäinen versio julkaistiin vuonna 2001. Hanke kasvoi nopeasti yhteisöksi saaden runsaasti uusia kielikäännöksiä käyttäjien kehittäminä. Kasvu yhteisöksi johti vuotuisen Plone-konferenssin kehittämiseen vuonna 2003. (McKay 2004, ix-x.)

Plone-konferenssi pidetään edelleen vuosittain, mutta lisäksi pidetään sprinttejä, joissa eri ryhmät kehittäjiä ympäri maailmaa kokoontuvat työskentelemään Plonen kehityksen parissa. Sprinttien kesto vaihtelee muutamista päivistä viikkoon. Maaliskuussa 2004 Plone 2.0 julkaistiin. Sen julkaisu toi enemmän muokattavia ominaisuuksia Ploneen, ja se tehosti lisäosatoimintoja. Toukokuussa 2004 Plone-säätiö Plone

Foundation luotiin kehittämään, markkinoimaan ja suojelemaan Plonea. Säätiöllä on omistusvalta Plonen koodipohjaan, tavaramerkkeihin ja domain-nimiin. Vaikka säätiö perustettiin suojelemaan omistusoikeuksia, Plone on edelleen avoimen lähdekoodin sisällönhallintajärjestelmä. 12. maaliskuuta 2007 Plone 3 julkaistiin, ja sen julkaisu toi mukanaan samalla sivulla muokkaamisen, päivitetyn visuaalisen editorin ja vahvisti turvallisuutta, monien muiden parannuksien kanssa. Plone 4 julkaistiin syyskuussa 2010 ja nykyinen Plone 5 julkaistiin syyskuussa 2015. (Plone (software) n.d.)

3.3 Tietoturva

Plone on yleisesti tunnettu hyvästä turvallisuuden tasostaan, ja vain muutama Plonea käyttävään sivuun on hakkeroiduttu koskaan. Suurin osa Plonesta löydettyistä heikkouksista on aiheuttanut harmia vain sivuun itseensä. Arkaluontoisiin tietoihin tai palvelimeen ei koskaan ole päästy käsiksi. Plonesta on myös raportoitu jatkuvasti vähemmän heikkouksia kohdistuen tietokantoihin kuin muissa sisällönhallintajärjestelmissä. (Security track record n.d.)

Plonen turvallisuuden taso on yleisessä tiedossa, ja siihen luottavat isot organisaatiot kuten FBI, Brasilian hallitus ja Euroopan ympäristökeskus käyttämällä Plonea sisällönhallintajärjestelmänään. (Walsh 2011.)

Mitre on ei-voittoa tavoitteleva yhtiö, joka isännöi yleiset haavoittuvaisuudet ja alttiudet (Common Vulnerabilities and Exposures) CVE-tietokantaa. CVE-tietokanta tarjoaa maailmanlaajuisen raportointimekanismin kehittäjille ja teollisuudelle, Yhdysvaltain kansallinen haavoittuvaisuustietokanta (U.S. National Vulnerability Database) NVD perustuu CVE:hen. Mitren mukaan Plonella on alhaisin määrä raportoituja haavoittuvaisuuksia sen elinikänsä kuin muilla suosituilla sisällönhallintajärjestelmillä. (Ks. taulukko 1) (Plone (software) n.d.)

Taulukko 1. Raportoidut haavoittuvaisuudet CVE-tietokannan mukaan (CVE List Master Copy 2017)

CMS	Ensimmäinen julkaisu	Haavoittuvaisuudet	Lisäosiin liittyvät haavoittuvaisuudet
Plone	2001	73	ei raportoitu
Joomla!	2005	647	57
Wordpress	2003	1073	889
Drupal	2001	1009	6

Täytyy ottaa huomioon, että muut verratut sisällönhallintajärjestelmät ovat suosittumia, mikä tarkoittaa myös enemmän ihmisiä, jotka pystyvät havaitsemaan haavoittuvaisuuksia (Google Trends 2017). Suurien käyttäjämäärien takia kyseiset sisällönhallintajärjestelmät ovat myös todennäköisemmin suuremmissa riskissä joutua hakkerointiyrityksen kohteeksi. Muiden sisällönhallintajärjestelmien haavoittuvaisuuksien lukumäärää kasvattavat myös lisäosat ja liitännäiset, joita Plonessa on huomattavat paljon vähemmän kuin suosituimmissa sisällönhallintajärjestelmissä. (Ks. taulukko 1)

Plone on rakennettu Pythonilla ja Zopella, jotka ovat erittäin turvallisia alustoja, mikä edes auttaa Plonen turvallisuutta (CVE List Master Copy 2017). Plonen joustava käytönvalvontalista järjestelmän käyttöoikeuksille auttaa myös varmistamaan, että käyttäjät näkevät vain sisällön, mikä heidän on tarkoitus nähdä, ja näin lisää sisällön turvallisuutta.

3.4 Saavutettavuus ja käytettävyys

Plone tarjoaa erinomaisen käyttökokemuksen, joka tarjoaa korkean tason käytettävyyden ja saatavuuden. Tämä ei tule pelkästään esille Plonen helppolukuisesta koodista, vaan myös Plonen ydinasioista.

Plone täyttää WCAG v2.0 (Web Content Accessibility Guidelines) Web saavutettavuusohjeen tason "AA", joka lukee mukaan sokeat ja heikkonäköiset, kuurot ja huo-

nokuuloiset, oppimisvaikeudet, kognitiivisesti rajoittuneet, liikerajoittuneet, puhevammaiset, valoherkkyydet ja yhdistelmät näistä. (Accessibility n.d.)

Plone myös täyttää W3C:n WAI-AA ja Yhdysvaltain hallituksen § 508:n vaatimukset näkö heikentyneille ja motorisesti heikentyneille yksilöille. Näin sivustoa, joka on rakennettu Plonella, pystyvät käyttämään myös ihmiset, joilla on näkövammoja. (Features n.d.)

Plonella on käytettävyystiimi, joka arvioi käyttöliittymän Plonen parannusehdotuksissa, luo ja ylläpitää käyttöliittymän ohjeistusta sekä auttaa kehittäjiä luomaan hyvän ja yhtenäisen Plone-käyttäjäkokemuksen. Tiimin velvollisuuksia on myös käyttöttestata uudet käyttöliittymään liittyvät osat ja auttaa johtamaan pyrkimyksiä saada uusia parannuksia Plone käyttäjäkokemukseen. (Usability Team n.d.)

3.5 Hakukoneoptimointi

Plonen sisään on rakennettu monia hakukoneoptimointia helpottavia ominaisuuksia, joten se ei vaadi lisäosien asentamista. Sivun luomisen yhteydessä annettavasta otsikosta tulee oletuksena sivun verkkosivun URL-osoitteen loppu. Plonessa olevat URL-osoitteet ovat selkolukuisia ja avainsana rikkaita koska, jokainen osoitteessa -merkillä eroteltu sana on hakusana. Sivuissa käytettävän kuvauskentän sisältö näkyy hakukoneiden tuloksissa sekä Plonen sisällä hakutuloksissa, kansio listauksissa ja kokoelmisissa. Kuvia lisätessä kuvalle annetusta otsikosta generoituu automaattisesti alt-teksti, jonka pystyy myös vaihtamaan. (Wozniak 2010; Hendryx-Parker 2010.)

Plone luo automaattisesti (eXtensible Markup Language) XML-sivukarttatiedoston, joka kuvaa sivuston sisällön rakennetta. Sivukarttatiedosto auttaa hakukoneiden robotteja indeksoimaan sivustot tehokkaammin. Plone vastaa pyyntöihin, joiden seurauksena ei löydy sivua, oikealla tilakoodilla 404, ja se ehdottaa sivuston sisällöstä toisia sivuja haetun sivun URL-osoitteesta saatujen hakusanojen perusteella. Plones on myös sisään rakennettu tuki Google Analyticsia varten. (Hendryx-Parker 2010.)

3.6 Responsiivisuus

Plone 5:n oletusteema on nimeltään Barceloneta. Se on Diazo-teema, joka on tehty alusta alkaen käyttäen moderneja frontend teknologioita. Barceloneta on responsiivinen ja se kattaa kaikki Plonen käyttöliittymän osat, mukaan lukien sisällönhallintajärjestelmän taustalla käytettävän käyttöliittymän. Barceloneta perustuu Bootstrap 3:een, mutta se ei ole millään tavalla riippuvainen siitä. Siinä on käytetty osia Bootstrapin rakenteesta ja hyvistä käytänteistä. Siitä huolimatta se on täysin mukautettavissa Ploneen. (Barceloneta theme n.d.)

3.7 Kehitysyhteisö

Plonen kehityksen aloittivat perustajat Alexander Limi, Alan Runyan ja Vidar Andersen. Yhtiö Plone Solutions perustettiin vuonna 2003, jolloin Plone oli vielä projekti, jonka laillisesti omistivat Plonen perustajat. Vuotta myöhemmin Plone-yhteisö loi Plone-säätiön Plone Foundationin kehittämään, markkinoimaan ja suojelemaan Plonea. Plonen perustajat siirsivät Plonen immateriaalioikeudet Plone Solutionsilta Plone Foundationille. (Plone Solutions rebrands company as "Jarn" 2007)

Vuonna 2007 Plone Solutions vaihtoi nimekseen Jarn, jotta Plone-sana ei viittaisi heihin enää. Se helpotti Plone Foundationin tavaramerkkiongelmien valvomista sekä molempien yhtiöiden tasavertaista kilpailumahdollisuutta. Jarn on Plonen konsultointi- ja palveluntarjoajayritys Euroopassa. Jarn tarjoaa täyden valikoiman Plone palveluja kuten hosting, tuki, ylläpito, kehittäminen, konsultointi ja analyysi. (Mt.)

Plone-säätiöllä Plone Foundation on omistusvalta Plonen koodipohjaan, tavaramerkkeihin ja domain-nimiin. Vaikka säätiö perustettiin suojelemaan omistusoikeuksia, Plone on edelleen avoimen lähdekoodin sisällönhallintajärjestelmä. Tämä mahdollistaa sen, että Plonea ei voi ostaa pois markkinoilta, eli kehitys jatkuu aina niin kauan, kuin Plone yhteisö pysyy aktiivisena. (Mt.)

4 Plonen taiton kehitys

4.1 Yleistä

Plonen taiton kehityksen vaiheet on esitetty seuraavien lukujen avulla kronologisessa järjestyksessä.

Seuraavissa luvuissa tullaan viittaamaan paljon tageihin, jotka ovat tärkeä osa eri merkkauškieliä. Ne kertovat mihin kohtaan tietyn niminen elementti tulee laittaa. Tagi muodostuu <- ja >-merkistä sekä niiden väliin kirjoitettavasta nimestä. Tageja on kahdenlaisia: yksittäisiä ja laatikkotageja. Yksittäinen alkaa merkillä < ja loppuu merkkiin >. Laatikko alkaa tagilla ja loppuu samannimiseen tagiin, johon on lisätty /-merkki ennen tekstiä.

4.2 DTML

DTML (Document Template Markup Language) on Zopen tagipohjainenmerkkauškieli, joka dynaamisesti tuottaa, kontrolloi ja muotoilee sisältöä. Sitä käytetään yleisesti modulaaristen ja dynaamisten verkkokäyttöliittymien rakentamiseen. Se on palvelimen puolella toimiva kieli kuten PHP (PHP: Hypertext Preprocessor) ja JSP(JavaServer Pages). Zope suorittaa DTML-komennot palvelimella, ja tulos lähetetään nettiselaimen. DTML-koodausta voi käyttää kahdenlaisissa Zope-objekteissa: DTML-dokumenteissa ja DTML-metodeissa. DTML on tarkoitettu ulkoasun hallinnoinnista varten ja ei ole käytössä enää Zopessa. (Latteier & Pelletier 2002, 53-54.)

DTML:n hyviä puolia ovat, että se tukee sisällön ja asettelun uudelleen käyttöä, epäyhdenäisen tiedon muotoilua sekä ulkoasun erottamisen logiikasta ja datasta. Se mahdollistaa kaikenlaisen datan helpon ja tehokkaan muotoilun. Sitä voi käyttää metodien kutsumiseen, tietokanta tiedusteluun, Zopen sisällön tutkimiseen, lomakkeiden käsittelyyn ja jne. Tietokannan tiedustelussa DTML ei tarvitse kertoa, että on kyse tietokannan tiedustelusta ja mistä löytää kyseisen tietokannan. Kertomalla DTML:ille minkä nimistä objektia käsitellä se osaa itse hakea tiedot mistä on kyse ja missä sellainen objekti on. (Mts. 54-55)

Seuraava koodiesimerkki hakee ja uudelleen käyttää yleisen ylätunnisteen sekä alatunnisteen verkkosivuun.

```
<dtml-var standard_html_header>
<p>
  Hello world.
</p>
<dtml-var standard_html_footer>
```

DTML-komennot kirjoitetaan tageina, jotka alkavat sanalla dtml. Edellisessä esimerkissä muodostuva verkkosivu on HTML- ja DTML-kielen yhdistelmä, missä DTML-metodeiden avulla HTML-sivuun sijoitetaan standardi ylätunniste ja alatunniste. Lopputuloksena muodostuvan sivun koodi näyttää tältä:

```
<html>
<body bgcolor="#FFFFFF">
<p>
  Hello world.
</p>
<hr>
<p>
  Last modified 2000/10/16 by AmosL
</p>
</body>
</html>
```

Ylätunniste luokitteli sivun taustavärin valkoiseksi ja alatunniste lisäsi tiedon sivun viimeisestä muokkauskerrasta. (Mts. 54.)

DTML ei ole yleistarkoitushjelmointikieli, vaikka sillä onkin mahdollista toteuttaa monimutkaisia algoritmeja, mutta niiden toteuttaminen on hankalaa, joten sen takia se ei ole suositeltavaa. Esimerkiksi DTML:ää ei kannata käyttää laskutoimitusten tekemiseen, vaikka ne olisivatkin Pythonilla liian helppoja. DTML ei kannata myöskään käyttää jonojen käsittelyyn. (Mts. 55-56.)

DTML-syntaksi on samankaltainen kuin HTML-kielen, mutta sitä yleensä käytetään muiden merkkaukielien kanssa yhdistelemällä. DTML:ssä on kahdenlaisia tageja: yksittäisiä ja laatikkotageja. Yksittäinen alkaa merkillä < ja loppuu merkkiin >. Laatikko alkaa tagilla ja loppuu samanimiseen tagiin, johon on lisätty /-merkki ennen tekstiä eli DTML-tagit toimivat siis saman lailla kuin HTML:ssä ja XML:ssä. Esimerkki yksittäisestä DTML-tagista:

```
<dtml-var parrot>
```

Esimerkki laatikko DTML-tagista:

```
<dtml-in mySequence>
    <!-- this is an HTML comment inside the in
tag block -->
</dtml-in>
```

Kaikissa DTML-tageissa käytetään määritteitä, mutta jotkut määritteet ovat vapaaehtoisia, kuten var tagin missing määrite, jolla voi antaa oletusarvon muuttujalle siltä varalta, että muuttujan etsittyä arvoa ei löydy. Määritteet kertovat tagille miten sen pitää toimia, ja jollain määritteillä ei edes ole arvoa kuten määritteellä upper, jolla saa muutettua muuttujan kirjaimet isoiksi. (Mts. 56-57.)

Muuttujien lisääminen on DTML:ssä yksi perustehtävistä. DTML-muuttujien arvojen täyttäminen toimii aina samalla tavalla. Esimerkiksi täyttääkseen muuttujan arvon var tagi etsii ensiksi muuttujaa nykyisestä objektista, sen jälkeen samasta paikkaa missä kyseinen objekti sijaitsee mennessä ylemmäs root kansioon saakka ja viimeisenä verkkopyynnöstä eli lomakkeista ja kekseistä kunnes muuttuja löytyy. Zopen muuttujien etsintätyyli, selittää miten objektit voivat saada sisältöä ja käyttäytymistyyliä niiden vanhemmilta sekä miten sisältö voidaan määritellä yhdessä paikassa ja ottaa käyttöön monessa muussa objektissa. DTML etsii muuttujia myös lähetetyistä HTML-lomakkeista, mikä mahdollistaa sen, että erillisiä objekteja ei tarvita lomakkeiden käsittelyyn. (Mts. 57-59.)

Objektien paikoilla on todella suuri merkitys, jotta muuttujat (arvot) löytyvät oikeista paikoista. Globaalin ylätunnisteen pystyy esimerkiksi kumoamaan tekemällä alemmalle olevaan kansioon erillisen ylätunnisteen. Erillinen ylätunniste tulee käyttöön kaikkiin verkkosivuihin kansiossa ja sen alakansioissa. Saman niminen metodi kumoaa ylempänä olevan metodin, koska Zope löytää lähempänä olevan muuttujan etsintätyylillä. (Mts. 60-61.)

DTML:n tavallinen tagi automaattisesti löytää muuttujan ja kutsuu sitä ilman apua. Ilmaisut ovat tagimääritteitä, jotka sisältävät Python-koodin pätkiä, ja niiden avulla voi testata vaatimuksia helposti, sillä pelkällä DTML:llä tarvitsisi erillisen metodin siihen. Python-ilmaukset voivat tehdä DTML:stä vaikealukuista, mikä ei ole hyvä asia,

joten suositeltavampaa on käyttää ohjelmointia. Tyypillisin virhe, mitä ilmauksien kanssa voi tehdä, on sekoittaa ilmaisut tavallisen tagin syntaksiin. Tavallinen tagi ilman Python-ilmaisua.

```
<dtml-var objectValues>
```

Tagi, jossa on Python-ilmaisua.

```
<dtml-var expr="objectValues()">
```

Python-ilmauksia käyttäessä täytyy tietää, minkälaista muuttujaa on tekemässä, ja että käyttää oikeaa Python-syntaksia, jotta muuttuja voidaan asianmukaisesti näyttää. (Mts. 61-63.)

Yleisimmät DTML-tagit ovat var, if, else, elif ja in. Tagi var lisää muuttujia DTML-metodeihin ja dokumentteihin muuttujan etsintätyylillä, ja se pystyy käyttämään Python-ilmauksia. Tagi if voi testata ehtoja ja tehdä eri toimenpiteitä ehtojen tuloksen perusteella. Ehto on joko tosi tai epätosi. Tagi else mahdollistaa ehtoa testatessa toimenpiteen lisäämisen myös siinä tilanteessa, jos ehto onkin epätosi. Tagi elif lisää mahdollisuuden testata eri ehtoja samanaikaisesti, ja kyseisiä tajeja voi olla enemmän kuin yksi if-laatikotagissa. Tagi else tulee aina if-laatikotagin sisään viimeiseksi tagiksi.

```
<dtml-if expr="monkeys > monkey_limit">
  <p>
    There are too many monkeys!
  </p>
<dtml-elif expr="monkeys < minimum_monkeys">
  <p>
    There aren't enough monkeys!
  </p>
<dtml-else>
  <p>
    There are just enough monkeys.
  </p>
</dtml-if>
```

Kyseisessä esimerkissä if-tagin ehto täyttyy, jos apinoita on enemmän kuin apinoiden rajamäärä on arvoltaan ja silloin sivu näyttää ensimmäisen p-tagin sisällä olevan tekstin. Seuraavaksi olevan elif-tagin ehto täyttyy, jos apinoita on vähemmän kuin apinoiden rajamäärä on arvoltaan ja silloin sivu näyttää toisen p-tagin sisällä olevan tekstin. Lopuksi, jos kumpikaan aikaisemmista ehdoista ei täyty niin else-tagin ehto täyttyy ja

silloin sivu näyttää viimeisen p-tagin sisällä olevan tekstin. Tagi in on laatikkotagi niin kuin if-tagikin. Se käy läpi sarjan objekteja ja suorittaa pätkän koodia jokaista objektia kohden, kunnes kaikki sarjan objektit on käyty läpi. (Mts. 63-67.)

4.3 ZPT

ZPT (Zope Page Templates) on XML- ja HTML-kieleen perustuva teemoituskieli, joka sisältää (Template Attribute Language) TAL-, (TAL Expression Syntax) TALES-, ja (Macro Expansion TAL) METAL-kielet. TAL on määritteikieli, jota käytetään dynaamisten sivumallien luomiseen. TALES kuvaa ilmauksia, jotka toimittivat datan TAL:lle ja METAL:lle. METAL on toiminto HTML/XML-makron esikäsittelyyn. (27. Appendix C: Zope Page Templates Reference n.d.)

ZPT:llä tehty sivu on HTML- ja XML-kielellä kirjoitettu rakenne, joka sisältää komentoja. Komennot suoritetaan, kun sivu kutsutaan, joten tuotos on dynaaminen. Komentojen kirjoittamiseen käytetään TAL-kieltä ja komennot ovat elementin määritteitä, jotka kirjoitetaan elementtiin, jota halutaan muuttaa. Komennot löytyvät erillisestä nimiavaruudesta. Se mahdollistaa ZPT:ien pysymisen validoitavina XHTML-dokumentteina ja sen, että niitä pystyy muokkaamaan käyttäen WYSIWYG-editoria tai tarkastelemaan suoraan nettiselaimella. (Weitershausen 2008, 100)

TAL-tulkille ei ole väliä, missä järjestyksessä määritteet on kirjoitettu. Se toteuttaa kyseiset TAL-komennot ennalta määrättyssä järjestyksessä. Joitakin komentoja ei voi käyttää samassa elementissä yhtä aikaa kuten esimerkiksi content ja replace komennot. Seuraavana on käyty läpi kaikki TAL-komennot toteutusjärjestyksen mukaan. (Mts. 100)

Define määrittelee muuttujat tehden niistä paikallisia muuttujia. Niitä voi käyttää vain elementissä, jossa ne on luotu ja elementin sisällä olevissa elementeissä. Jos määritelmän perässä on global, muuttujat ovat käytettävissä kaikissa määritetyn elementin jälkeisissä elementeissä. Seuraavana oleva esimerkki määrittelee globaalin muuttujan.

```
tal:define="global company_name string:Zope Corp,
Inc."
```


Condition arvioi onko ilmaisu tosi vai epätosi. Jos ilmaisu on tosi, elementin käsittely jatkuu normaalisti. Muussa tapauksessa elementti poistetaan kokonaan sivusta. Seuraavana oleva esimerkki testaa onko muuttuja olemassa.

```
<p tal:condition="exists:request/message"
tal:content="request/message">message goes here
</p>
```

Repeat toistaa nykyisen elementin niin monta kertaa kuin ilmauksen palauttamassa jaksossa on kohtia. Jos jakso on tyhjä, elementti poistetaan kokonaan. Jakson kohdat kohta kerrallaan sitoutuvat paikalliseen muuttujaan toistuvan elementin muuttuja alueelle. (27. Appendix C: Zope Page Templates Reference n.d.)

Content täyttää elementin sisällön ilmauksen palauttamalla arvolla. Elementti ja sen ominaisuudet säilyvät, mutta elementin sisältö, elementin alla olevia elementtejä myöten korvautuu. (Weitershausen 2008, 100)

Seuraavassa esimerkissä lisätään jakso taulukon rivejä, ja käytetään contentia lisäämään sisällöt lisättyihin taulukkoriveihin. Repeat muuttujaa käytetään numeroimaan lisätyt rivit.

```
<table>
  <tr tal:repeat="item context/cart">
    <td tal:content="repeat/item/number">1</td>
    <td tal:content="item/description">Widget</td>
    <td tal:content="item/price">$1.50</td>
  </tr>
</table>
```

(27. Appendix C: Zope Page Templates Reference n.d.)

Replace korvaa elementin ilmauksen palauttamalla arvolla. Se korvaa elementin dynaamisella sisällöllä. (Weitershausen 2008, 101)

Esimerkissä on kaksi tapaa korvata sivun otsikko.

```
<span tal:replace="template/title">Title</span>
<span tal:replace="text template/title">Title</span>
```

(27. Appendix C: Zope Page Templates Reference n.d.)

Attributes määrittelee tai korvaa elementin ominaisuuksia vastaavan ilmauksen arvoilla. (Weitershausen 2008, 101)

Esimerkissä näkyy miten voi korvata kaksi ominaisuutta.

```
<textarea rows="80" cols="20"
  tal:attributes="rows request/rows;cols
  request/cols">
```

Omit-tag poistaa elementin alku ja loppu tagin jättäen elementin sisällön ja alla olevat elementit, jos annettu ilmaus on tosi. Jos ilmausta ei määritellä ilmauksen arvo oletetaan automaattisesti todeksi. Esimerkissä ei ole määritelty ilmausta, joten ainoa mikä näkyy, tuloksessa on p-tagin sisällä oleva teksti.

```
<div tal:omit-tag=""
  comment="This tag will be removed">
<p>...but this text will remain.</p>
</div>
```

On-error tarjoaa tavan virheidenkäsittelylle. Kun TAL-komento tuottaa virheen TAL-tulkki etsii error-komentoa kyseisestä elementistä, sitten ylemmästä elementistä ja niin edelleen kunnes ensimmäinen tulee vastaan. Kyseistä komentoa käsitellään kuin content-komentoa. Esimerkissä on kuvattu yksinkertainen komento, jota voi käyttää virheen sattuessa.

```
<b tal:on-error="string: Username is not defined!"
  tal:content="context/getUsername">Ishmael
</b>
```

Aikaisten esimerkkien mukaan TALilla pystyy siis korvaamaan, toistamaan tai jättämään pois elementin. (27. Appendix C: Zope Page Templates Reference n.d.)

TAL-komennoissa käsitellään paljon ilmauksia, ja kyseiset ilmaukset seuraavat TALES-syntaksi. TALES-ilmauksia on montaa tyyppiä. Path-ilmaukset sallivat pääsyn ominaisuuksiin, nimikkeisiin ja metodeihin tiedostojärjestelmän tai URL-polun kaltaisen merkinnän kautta. Polku sisältää yhden tai useamman ei tyhjän jonon, jotka on erotettu vinoviivoilla. Ensimmäisen jonon pitää olla sisään rakennetun tai käyttäjän tekemän muuttujan nimi ja jälkimmäiset jonot ovat polun osia. Jos kutsuttu objekti on metodi, funktio tai luokka se pystytään kutsumaan. Esimerkissä kutsutaan metodia values() polun my_recipes kohteeseen.

```
<!-- Call the values() method of the my_recipes folder -->
<p tal:repeat="recipe context/my_recipes/values">
Name of the dish: <span tal:replace="recipe/name" />
</p>
```

String-ilmaukset mahdollistavat suoraan tekstin lisäämisen sekä polkuilmauksen tuloksen lisäämisen ilmaukseen. Python-ilmaukset mahdollistavat Python-koodin osan arvioimisen TALES-ilmauksissa. (Weitershausen 2008, 101)

Ilmauksen jälkeen voi tulla myös määrite, joka muokkaa ilmauksen käytöstä tai tulosta. Not vaihtaa Boolean arvon epätodeksi. Nocall estää polkuilmauksen potentiaalisen objektin kutsumisen. Structure sallii ilmauksen tuloksen laittamisen ilman notee-
rausta. (Mts. 102)

METAL-kielellä määritellään makroja, jotka tarjoavat tavan määritellä palan asettelua yhdessä paikassa ja jakaa sen muualla. Makroon tehtävät muutokset näkyvät näin joka paikassa missä samaa makroa on käytetty. Makrojen tarkoitus on helpottaa tarvetta koodata samoja asioita monessa eri paikassa. (27. Appendix C: Zope Page Templates Reference n.d.)

4.4 Diazo

Diazo, joka tunnettiin aiemmin nimellä XDV, on avoimen lähdekoodin itsenäinen tekniikka. Sitä voidaan käyttää melkein minkä tahansa sivuston teemoittamiseen. Se on kirjoitettu Python-kielellä ja se luo XSLT:tä. Diazo on Plone-yhteisön kehittämä, ja Plone 4.3-versiosta lähtien se on ollut oletustapa teemoituksen tekemiseen Plone-sivustolle. (plone.app.theming 1.3.6 n.d.)

Suurin osa nykyaikaisista verkkopalvelimista tukee XSLT-muunnoksia, joten Diazon käyttöönotto WSGI:n, Ngnixin, Varnishin tai Apachen avulla on helppoa. Diazo luo erotuksen teematiedostojen ja muuntosääntöjen välille. Se mahdollistaa teematiedostojen käsittelyn tavallisessa HTML-muodossa, joten käyttäjien ei tarvitse osata XSLT:ää tai erityisiä teemoittamiseen liittyviä kieliä. Teematiedostot ovat staattista HTML-kieltä ja muuntosäännöt ovat Diazon omaa XML-syntaksiin pohjautuvaa kieltä. (Mt.)

Luodaan teema, joka koostuu yhdestä tai useammasta HTML-ja (Cascading Style Sheets) CSS-kieltä sisältävästä tiedostosta. Plonesta saadaan HTML-sivuja, joissa on teemoituksessa käytettävä sisältö. Diazolla yhdistellään Plonesta HTML-sisällön palasia luotuun HTML- ja CSS-teemamalliin. Diazon sulautumiskäyttäytymistä ohjaa

yksinkertainen tekstipohjainen sääntötiedosto, joka koostuu yksinkertaisista direktiiveistä, kuten <append>, <replace> ja <drop>. Se on yksinkertainen syntaksi, jonka kuka tahansa voi oppia käyttämään. Käyttämällä sitä, ei tarvitse ymmärtää mitään siitä, miten Plone toimii. Perus HTML ja CSS lukutaito ovat kaikki mitä tarvitset Diazon käyttöön. (Mt.)

Diazo toimii käyttämällä XSLT (eXtenstible Stylesheet Language Transformations) standardia, joka alun perin kehitettiin 1990-luvun lopulla, joka määrittelee kielen XML-asiakirjojen muuntamiseen muiksi XML-dokumenteiksi. Raaka XSLT on liian monimutkaista jokapäiväiseen käyttöön arjen verkkosivujen teemoittamistekniikaksi. Diazo tarjoaa yksinkertaisen ja käyttäjäystävällisen kokoelman perussääntöjä, jotka kootaan kulissien takana XSLT:ksi. XSLT-lausekkeiden täydellinen käyttäminen Diazo-sääntötiedostoissa on myös mahdollista. (Mt.)

Diazon teema koostuu kolmesta elementistä:

1. Yksi tai useampi HTML-tiedosto, jotka edustavat haluttua ulkoasua. Kyseisiä tiedostoja kutsutaan teematiedostoiksi. Nämä sisältävät paikkamerkkejä sisältöä varten, joka saadaan Plone-sisällönhallintajärjestelmästä. Yleisin tapa luoda teema on käyttää työpöytäohjelmistoja, kuten Dreamweaveria tai tekstieditoria. Niillä luodaan tarvittavat merkinnät, tyylit ja koodit teemaan sekä niiden avulla teemaa voi testata paikallisesti selaimessa.
2. Plonessa tuotettu sisältö, joka teemoitetaan.
3. Säännöt-tiedosto, mikä määrittelee mihin sisältö teemassa tulee.

Jos on tarvetta muuttaa olennaisesti Plonesta saatavaa sisältöä, se on tehtävä Plonen räätälöinnillä. Diazo ei korvaa täysin kaikkia räätälöinti tarpeita, joita Plonella voi olla tarvetta tehdä, mutta se vähentää huomattavasti Plonen sisällä tehtävien muutosten määrää. (Mt.)

4.5 Mosaic

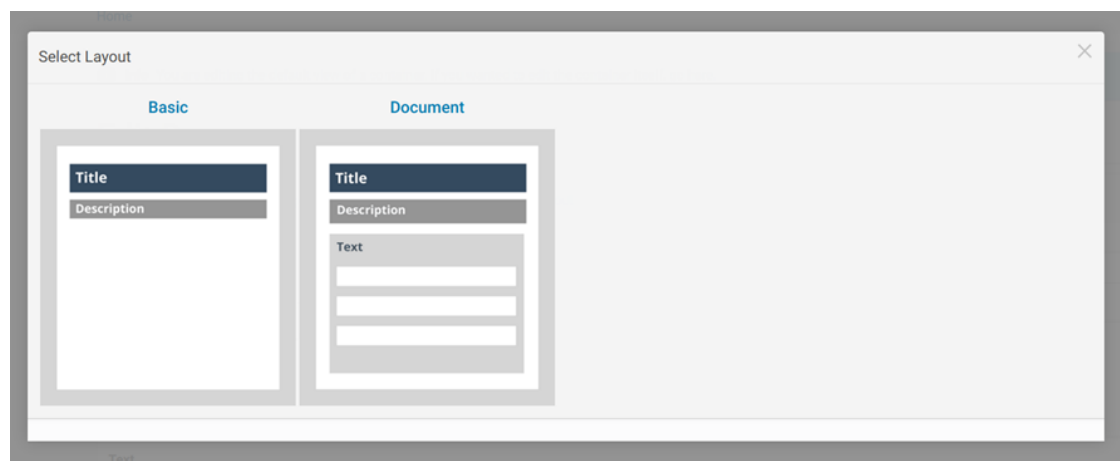
Mosaic on Plonen uusin asetteluratkaisu. Se on lisäosa Ploneen, joka mahdollistaa asettelujen hallitsemisen Plonen käyttöliittymästä. Mosaic on rakennettu Plone 5-versioon, mutta se toimii myös Plone 4.3-versiossa. Mosaic-pohjaisen asettelu perusosaa kutsutaan tiileksi. Asettelu muodostuu useiden tiilien yhdistelmästä. Tiili on dynaaminen osa verkkosivua, se voi olla tekstielementti, kuva, kenttä jne. Mosaic

tarjoaa editorin, jolla pystyy helposti sijoittelemaan tiiliä teeman rakenteeseen. (Plone Foundation 2017.)

4.5.1 Mosaic-editori

Mosaic-editorin hallinnoima sivu on tallennettu yksinkertaiseksi HTML-asiakirjaksi, joka edustaa kyseisen sivun varsinaista sisältöä erillisenä, julkaistavana resurssina, joka on vailla sivuston ulkoasua esimerkiksi globaaleja navigointielementtejä. Mosaic-editori on visuaalinen editori sivuille, jotka on tehty käyttämällä tiiliä. Se perustuu ruudukkojärjestelmään tiilien asettamiseksi sivulle WYSIWYG-tyyliin ja käyttämällä raahaa ja pudota tapaa. Mosaic-editorin avulla voi luoda monimutkaisiakin ulkoasuja unohtamatta tasapainoisuutta ja visuaalisuutta. (Plone Foundation 2017.)

Mosaic-editori otetaan käyttöön sivuun muuttamalla sivun oletusnäkyä Mosaic-asetteluun valitsemalla valikosta Esitystapa ja sen alta Mosaic layout. Mosaic-editori korvaa tällä tavalla oletus editorin/muokauslomakkeen. Seuraavaksi valitaan valikosta Muokkaa, jotta päästään muokkaamaan sivua. Jos tämä on ensimmäinen muokkauskerta nykyiseen kohteeseen, pyydetään valitsemaan asettelutyyli. (Ks. kuvio 1.)



Kuvio 1. Plonen asettelutyylin valitseminen

Mosaic-editorin avulla voidaan muuttaa sisältökenttien sisältöä (kuten oletus Plone-editorilla), mutta kentät näytetään asettelussa ja niiden sisältöä muokataan asettelussa. (Ks. kuvio 2.) Yläpalkissa on neljä eri painiketta:

- Save, kenttien sisältöihin ja asetteluun tehtyjen muutoksien tallentamiseen.
- Cancel, muutoksien peruuttamiseen.
- Properties, sisältöominaisuuksiin pääsemiseksi: se näyttää yleiset Plonen sivun muokkauslomake välilehdet, mutta ulkoasuun viittaavat kentät ovat piilotettuja.
- Layout, sisällön asettelun hallintaan.



Kuvio 2. Mosaic-editorin näkymä

4.5.2 Sisällön asettelun muuttaminen

Ylävalikossa olevan Layout-napin alta löytyy vaihtoehto Change, josta painamalla tulevat esiin kaikki asetteluvaihtoehdot. (Ks. kuvio 2.) Valinnat rajoitetaan nykyiseen sisältötyyppiin sovellettavaan ulkoasuun. Sivun sisältötyypille Mosaic ehdottaa oletuksena kahta asettelua: perus ja asiakirja.

4.5.3 Sisällön asettelun muokkaus

Ylävalikossa olevan Layout-napin alta löytyy myös vaihtoehto Customize, josta painamalla Mosaic-editori siirtyy tilaan, jossa voi edelleen muuttaa kentän arvoja, mutta muuttaa myös ulkoasua. (Ks. kuvio 2.) Leijumalla sivun sisällössä nykyiset tiilet korostuvat ja niitä voi vedellä eri järjestykseen. Napsauttamalla tiiltä voi muokata sen sisältöä ja napsauttamalla nykyisen muokatun tiilen ulkopuolella, se poistaa muokkaustilan käytöstä. Asettelun muokkaustilassa yläpalkissa on kaksi painiketta lisää:

- Format, joka tarjoaa erilaisia yksinkertaisia muotoiluvaihtoehtoja tiilille kuten keskitä sisältö, kello oikealle tai riville esimerkiksi muuta taustaväriä.
- Insert, jonka avulla voidaan lisätä uusia tiiliä asetteluun.

4.5.4 Tiilet

Tiilet edustavat sivun dynaamista osuutta. Alkeellisimmalla tasolla tiili on yksinkertaisesti HTML-asiakirja, jossa on julkaistava URL-osoite. Käytännössä tiilet yleensä toteutetaan selaimen näkyminä tiilen perusluokasta ja rekisteröidään <plone:tile /> (Zope Configuration Mark-up Language) ZCML-direktiiviin. Tämä mahdollistaa sen, että tiilillä on perus metatiedot ja automaattisesti luotavat muokkauslomakkeet mille tahansa muokattavalle ominaisuudelle, mitkä Mosaic näyttää käyttäjille. Tiiliä on kolme erilaista tyyppiä. (Plone Foundation 2017)

Tekstitiilet sisältävät staattista HTML-merkintää (WYSIWYG-muokattu teksti), joka on sijoitettu sisällön tai sivuston ulkoasuun. Tarkkaan ottaen tekstitiilet eivät ole tiiliä, koska ne eivät sisällä lainkaan tiilien hakemista tai sulauttamista. Ne tallennetaan osana sivua tai sivuston ulkoasua. Tekstitiiliä voidaan kuitenkin siirrellä ympäriinsä ja hallita kuin mikä tahansa muuta tiiliä. (Mt.)

Kenttätiilet esittävät metatietokentän arvon, kuten otsikon tai kuvauksen. Kenttätiilien arvoja voidaan muokata asettelussa, mutta arvo tallennetaan taustalla olevaan kenttään. Arvon pystyy indeksoimaan luettelossa, käyttämään navigointiin ja niin edelleen. (Mt.)

Sovellustiili on minkä tahansa muun tyyppinen dynaaminen tiili esimerkiksi kansion listaus, mediasoitin, kysely, some-upotus tai ihan mikä muu mikä tulee mieleen. (Mt.)

5 Tutkimushankkeen esittelysivu

5.1 Vanha tilanne

Yliopistolla on vuosittain monta tutkimushanketta käynnissä, ja ne ovat kaikki aina hieman erilaisia toisistaan. Yliopistolla ei ole yksittäistä pohjaa tai ohjetta, minkä mukaan tutkimushankkeiden kotisivut pitäisi tehdä. Tämä on johtanut siihen, että kaikilla on omanlaisensa sivut ja lopputulos on aika sekava. Kotisivuja on ollut hankala päi-

vittää ja hallita, koska ei ole ollut tiedossa yhteisiä vaatimuksia sisällölle tai rakenteelle, joten sivut eivät ole olleet yhtenäisiä.

Vanha tapa tutkimushanke kotisivujen luomiseen oli luoda oma kansio, johon jokainen rakensi kotisivut omien tarpeiden ja kykyjen mukaan. Kansioiden sisälle saattoi muodostua hyvinkin monimutkaisia rakenteita, jos tutkimushankkeissa oli paljon esimerkiksi tekijöitä tai paljon tuloksia, mitkä kaikki lisättiin kansion sisälle.

5.2 Vaatimukset

Uuden ratkaisun pitää pääasiassa helpottaa tutkimushanke kotisivujen luomista ja niiden ylläpitoa, antaen kotisivun luojille käsityksen siitä, miten kotisivut jatkossa pitää tehdä. Tämä helpottaa myös tulevaisuudessa sivujen käyttäjää tietojen löytämisessä ja sivujen käyttämisessä. Kun kaikki uudet tutkimushanke kotisivut luodaan käyttämällä tätä ratkaisua, ovat sivut yhtenäiset ja ulkoasu ajantasaisen näköinen.

Tutkimushankesivun asettelutyylin tavoite oli siis olla yksinkertainen pohjarakenne mihin ylläpitäjien on helppo täyttää tutkimushankkeen tiedot ja lisätä uusia kohtia tarvittaessa rikkomatta asettelua. Tarkoituksena oli myös kehittää ohje sivun sisällön täyttämiseen, mikä kertoo, mitkä ovat tarvittavia tietoja sivun käyttämiselle sekä ohjeistaa sisällön luomisessa niin, että sisältö toimii pohjassa ja sivu on rakenteeltaan hyvä saavuttamaan näkyvyyden hakukoneissa.

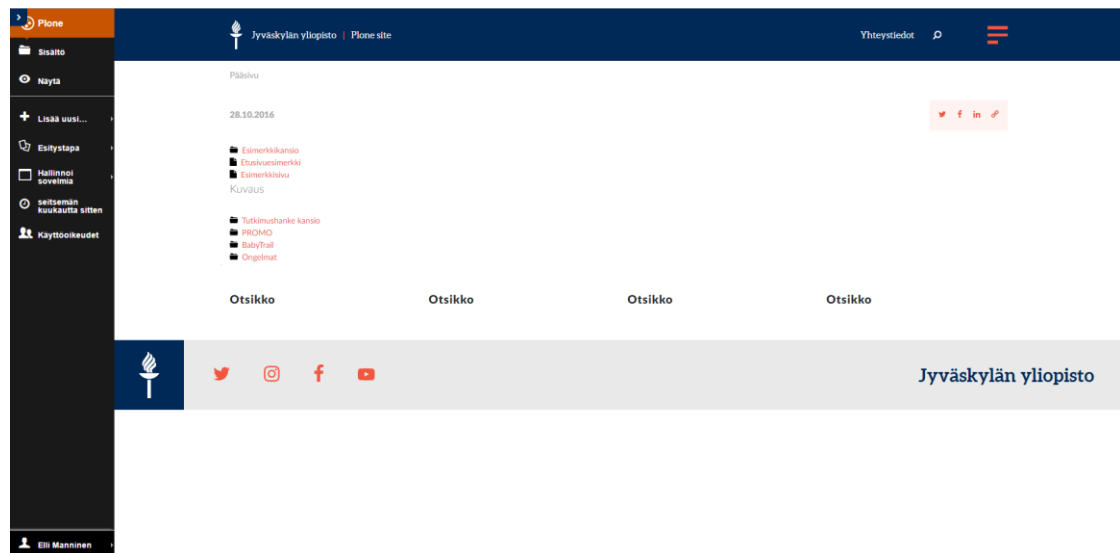
Opinnäytetyön aikana rakennettu pohja on Mosaic-editorilla luotu asettelun perusrakenne, jossa on tutkimushankkeen kotisivun luomiseen vaadittavat kohdat ja tarvittaessa kohtia voi lisätä. Pohjan rakenteeseen vaaditut kohdat olivat:

- Kuvakaruselli, johon tulee kuvia liittyen tutkimushankkeeseen, ja halutessa kuvien päälle voi laittaa kuvaustekstin tai linkin.
- Otsikko eli tutkimushankkeen nimi, mikä näkyy sivun alussa.
- Kuvaus, joka sisältää peruskuvauksen siitä minkälaisesta tutkimushankkeesta on kyse.
- Esittelyteksti
- Infolaatikko, jossa on kerrottuna tutkimushankkeen kesto, rahoittaja ja erillisen kotisivun osoite.
- Tekijät, missä on lista tutkimushankkeeseen osallistuvista ihmisistä.
- Lisätietoja/yhteystiedot kohtaan laitetaan tarkemmat yhteystiedot, joihin halutaan ihmisten ottavan yhteyttä.
- Ajankohtaista, mikä on joko RSS-syöte uutisista tai teksti, missä on kuvattu tutkimushankkeen ajankohtaiset asiat.

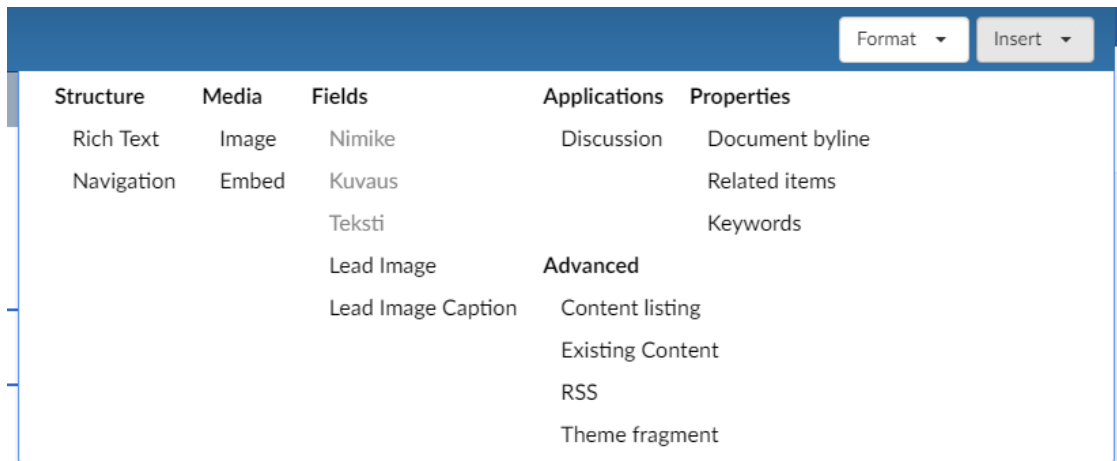
- Sosiaalinen media on kohta, johon tulee tutkimushankkeen some-syötteen. Some-kohtaan voi laittaa Facebook-, Twitter- tai Instagram-sovelluksen.
- Tulokset/julkaisut kohtaan tulee lista tutkimushankkeen aikana saaduista tuloksista.
- Yhteistyökumppanit kohtaan tulee lista kaikista tutkimushankkeen yhteistyökumppaneista ja heidän yhteistiedoista.

5.3 Uusi sivupohja

Tutkimushankkeen esittelysivun pohjan tekeminen alkoi Mosaic-asetteluun perheymisellä. Tutustuttiin testaussivuun, jolla esittelypohja tehtiin, ja sen tarjoamiin Mosaic-tiliin. (Ks. kuvio 3.) Tutkittiin, minkälaisia tiliä löytyy ja minkälaisen rakenteen pystyy tekemään kyseisillä tiilillä. (Ks. kuvio 4.) IT-palveluiden Kehittämispalveluiden kehittämät tiilet sai käyttöön Theme fragment tiilen kautta.

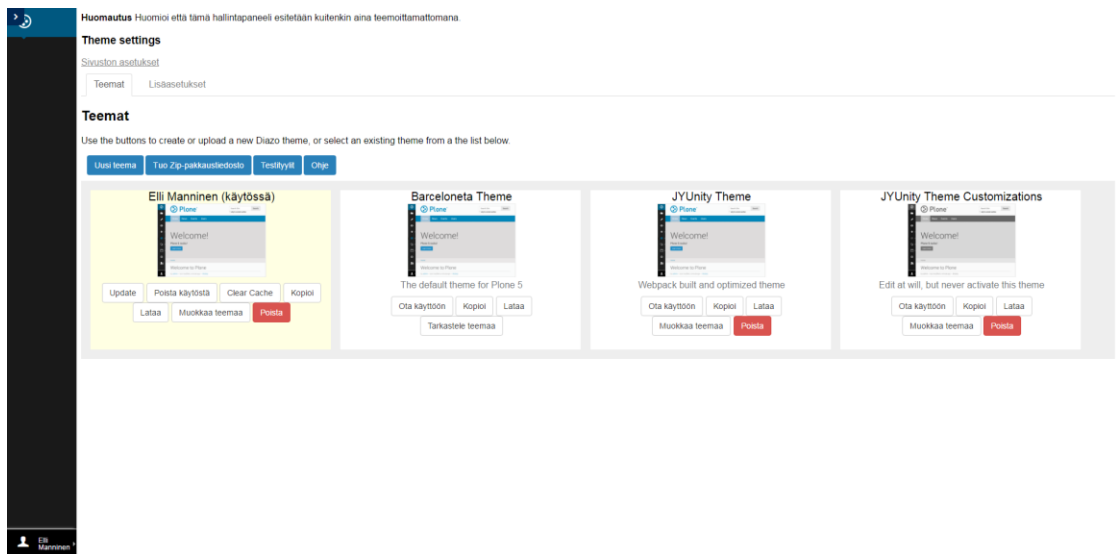


Kuvio 3. Yliopiston Plonen testaussivu



Kuvio 4. Yliopiston Mosaic-tiilet

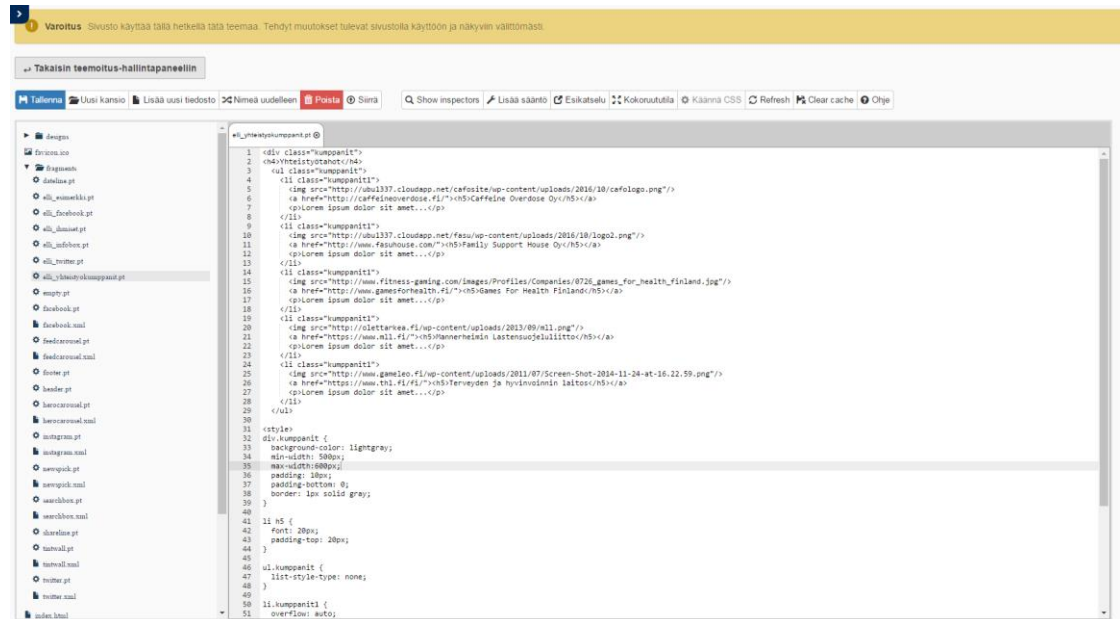
Yliopistolta saatuun esimerkki tutkimushankkeiden tiedoilla rakennettiin ensimmäinen esittelysivun versio, jonka avulla hahmoteltiin, mitä tiiliä esittelysivulle tarvittiin. Muutamia tarvittavia sisältöjä ei pystynyt tekemään asetteluun testisivuilla olevilla tiilillä, eikä tiilille pystynyt tekemään isompia tyyllittelyjä, joten näitä kokeiltiin rakentaa HTML- ja CSS-kieltä käyttäen. Teeman muokkaamiseen, missä pääsi rakentamaan tiiliä ja tekemään tiilille tyyllittelyitä, pääsi sivuston asetusten kautta.



Kuvio 5. Testausteema ja yliopiston teemat

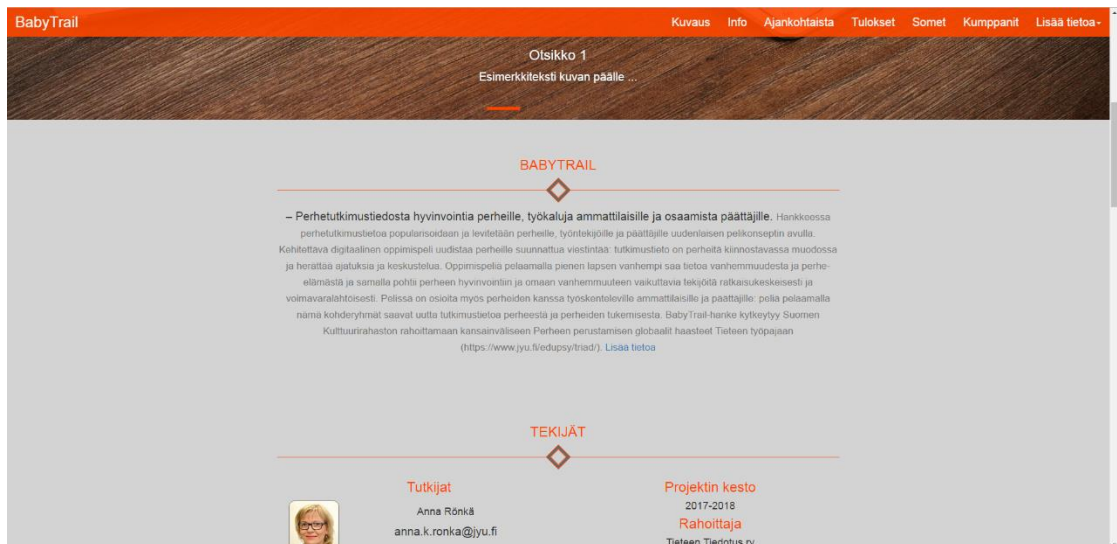
Editori, jolla tiiliä koodattiin, muistutti ihan tavallista HTML-editoria. (Ks. kuvio 6.)

Koodaamalla tehtiin infolaatikko-tiili, tekijät-tiili, yhteistyökumppanit-tiili ja sosiaalisen median tiiliä. (Ks. liite 1.)



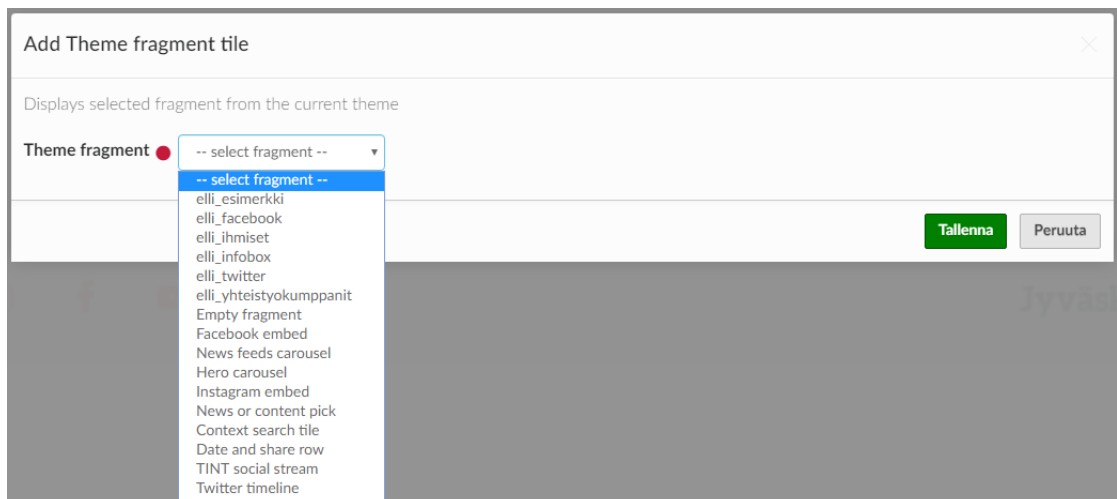
Kuvio 6. Tiilien rakentaminen HTML-editorissa

Joitain tiiliä ei saanut tehtyä pelkillä HTML- ja CSS-kielillä, joten asettelun kokonaisuus kokeiltiin tehdä erilliselle palvelimelle käyttäen HTML- ja CSS-kielen lisäksi Bootstrap frameworkia. (Ks. kuvio 7.) Kyseisellä kokonaisuudella saatiin tuotua esille ulkoasun ja rakenteen suunnittelu toimeksiantajalle.



Kuvio 7. Erilliselle palvelimelle tehty asettelu

Samalla aikaa, kun opinnäytetyö eteni, Mosaicia kehitettiin Jyväskylän yliopiston IT-palveluiden Kehittämispalveluiden toimesta. Kehittämispalveluiden ansiosta lopulta testaussivuun saatiin myös puuttuvat tiilet, joten suunniteltu asettelu oli mahdollista toteuttaa kokonaisuudessaan Mosaic-editorilla. (Ks. kuvio 8.)



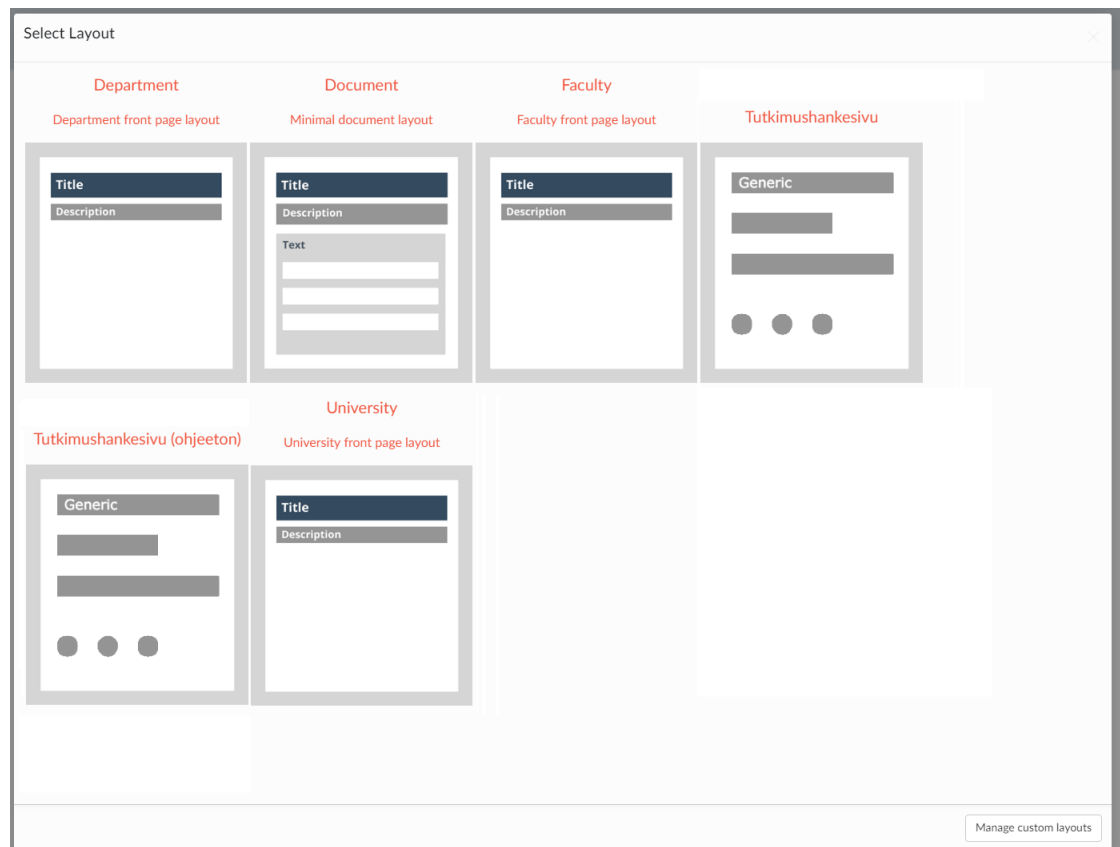
Kuvio 8. Yliopiston ja opinnäytetyön tekijän kehittämät tiilet

Lopullinen versio tutkimushankkeen esittelysivun pohjasta tehtiin käyttäen Mosaic-editorista löytyviä tiiliä. Kun pohja oli valmis, siitä tallennettiin asettelutyöli testaussi-

vuun mikä ajan myötä siirretään yliopiston varsinaiselle sivulle tutkimushankkeiden käyttöön.

5.4 Ohjeistus jatkokäyttöön

Tutkimushankkeen esittelysivun asettelupohjasta tehtiin loppujen lopuksi kaksi versiota ja molemmat tallennettiin omiksi asettelutyyleikseen. (Ks. kuvio 9.) Tavallisessa versiossa oli vain asetteluun vaaditut kohdat. Ohjeistetussa versiossa oli vaadittujen kohtien lisäksi myös tekstikenttiä, joihin oli kirjoitettu ohjeita. Ohjeissa kerrottiin, miten asettelutyylä käytetään ja miten sisällöt täytetään tiiliin. (Ks. liite 2.) Ohjeistuksesta tehtiin vielä erillinen sivurakenne, jotta ohjeita pystyi tarkastelemaan sivustolla ennen tavallisen sivupohja version käyttämistä.



Kuvio 9. Tallennetut asettelutyylit

6 Tulokset ja pohdinta

Opinnäytetyön tarkoituksena oli toteuttaa uudistettu ulkoasu ja rakenne Jyväskylän yliopiston tutkimushankkeen esittelysivun pohjaksi tutustuen samalla Plone-sisällönhallintajärjestelmään, WYSIWYG-asetteluun sekä Plonen Mosaic-editoriin. Lopputuotteena saatiin Plonessa käytettävä asettelutyöli. Opinnäytetyön kirjallisessa osassa käytiin läpi WYSIWYGn merkitystä, lyhyesti sisällönhallintajärjestelmien historiaa sekä tarkasteltiin myös Plone-sisällönhallintajärjestelmää yleisellä tasolla sekä taiton kehityksen kannalta.

Jyväskylän Yliopiston verkkosivujen uudistuksen edetessä tutkimushankkeen esittelysivun asettelupohja otetaan jatkossa käyttöön uusia tutkimushankesivuja tehtäessä. Opinnäytetyössä keskityttiin asettelupohjan kehittämiseen kurssin suorittajan näkökulmasta.

Asiakas oli tyytyväinen lopputuotteena saatuun sivupohjaan ja yhteydenpito pysyi mukavan ripeänä sähköpostin sekä tapaamisien muodossa. Varsinaisen käyttöönoton ollessa vielä edessä palaute on lähinnä rajoittunut Jyväskylän yliopiston IT-palveluiden Kehittämispalvelun kommentteihin. Hieman ahkerampi yhteydenpito toimeksiantajan kanssa olisi nopeuttanut ulkoasun kehitystä, tämä oli tärkeä huomio kehittäjälle itselleen. Jatkossa kannattaa olla itse aktiivisempi.

WYSIWYG-asetteluun sekä Plone-sisällönhallintajärjestelmän taiton kehitykseen perehtyminen syvensi ymmärrystä verkkokehittämisestä. Uusia tuttavuuksia olivat Plone-sisällönhallintajärjestelmän taiton kehityksen aikaisemmissa vaiheissa käytetyt kielet sekä WYSIWYG. Vanhastaan tuttujen tekniikoiden ymmärrystä päästiin syventämään HTML- ja CSS-formaattien muodossa.

Mosaic-tekniikka osoittautui helposti omaksuttavaksi tavaksi luoda teema Plone-sisällönhallintajärjestelmään. WYSIWYG-käyttöliittymän käyttäminen teki Mosaicin sisäistämisestä helppoa. Mosaicin avulla voitiin luoda vaivattomasti asettelutyöli Ploneseen. Testaussivulla tehtyä tutkimushankeasettelusivun rakentamista välillä vaikeutti vastaan tulleet virheet. IT-palveluiden Kehittämispalvelun avulla ne saatiin ratkottua ja ongelmat eivät lopulta hankaloittaneet liikaa tekemistä.

CSS hyödyntäminen tämän opinnäytetyön yhteydessä jäi pintaraapaisun tasolle. Jatkokehitys saattaa tuoda sille lisää käyttökohteita. CSS käytön määrä riippuu jatkossa siitä, kuinka paljon esittely sivun ulkoasua tullaan erkaannuttamaan Jyväskylän yliopiston sivulla jo olevasta ulkoasusta.

Lähteet

27. Appendix C: Zope Page Templates Reference. N.d. Viitattu 11.5.2017.
<http://docs.zope.org/zope2/zope2book/AppendixC.html#tales-overview>
- About Plone. N.d. Plonen verkkosivut. Viitattu 3.5.2017. <http://plone.com/about>
- Accessibility. N.d. Plonen verkkosivut Viitattu 3.5.2017.
<https://plone.org/accessibility-info>
- Barceloneta theme. N.d. Viitattu 15.5.2017. <https://docs.plone.org/adapt-and-extend/theming/barceloneta.html#barceloneta-theme>
- Blog. N.d. Wikipedia. Viitattu 3.5.2017. <https://en.wikipedia.org/wiki/Blog>
- Bradbury, N. 2009. HomeSite Discontinued. Viitattu 3.5.2017.
<http://nick.typepad.com/blog/2009/06/homesite-discontinued.html>
- Bravo (software). N.d. Wikipedia. Viitattu 3.5.2017.
[https://en.wikipedia.org/wiki/Bravo_\(software\)](https://en.wikipedia.org/wiki/Bravo_(software))
- CMS Usage Statistics. 2017. Viitattu 3.5.2017. <https://trends.builtwith.com/cms>
- CVE List Master Copy. 2017. Viitattu 3.5.2017. <http://cve.mitre.org/cve/cve.html>
- De Libero, G. 2016. The Content Management System Grows Up. Viitattu 3.5.2017.
<https://medium.com/@GeneDeLibero/the-content-management-system-grows-up-3779bfa8dad3>
- Features. N.d. Plonen verkkosivut. Viitattu 3.5.2017. <https://plone.com/features>
- Forgaard, R. 1996. The Early FrontPage History. Viitattu 3.5.2017.
<http://www.microsoftbob.com/post/The-Early-FrontPage-History>
- Google Trends. 2017. Viitattu 3.5.2017.
<https://trends.google.com/trends/explore?date=all&q=drupal,joomla,plone,wordpress&hl=en-US>
- Hendryx-Parker, C. 2010. 10 Reasons Why Plone Can Improve Your SEO. Viitattu 15.5.2017. <http://www.sixfeetup.com/blog/10-reasons-why-plone-can-improve-your-seo>
- Latteier, A. & Pelletier, M. 2002. The Zope Book. New Riders Publishing. 2. painos
- Macromedia HomeSite. N.d. Wikipedia. Viitattu 3.5.2017.
https://en.wikipedia.org/wiki/Macromedia_HomeSite
- Marohnić, V. 2014. What we can learn from the evolution of Content Management Systems. Viitattu 3.5.2017 https://thenextweb.com/dd/2014/02/20/can-learn-evolution-content-management-systems/#.tnw_kmnHRekL
- McCrea, J. 2014. THE UNTOLD (AND RATHER IMPROBABLE) STORY BEHIND THE FIRST REAL HTML EDITOR. Viitattu 3.5.2017 <https://therealmccrea.com/tag/webmagic/>
- McKay, A. 2004. The Definitive Guide to Plone. Apress

- Microsoft Word. N.d. Wikipedia. Viitattu 3.5.2017.
https://en.wikipedia.org/wiki/Microsoft_Word
- Nimchynska, J. N.d. The Evolution of CMS Platform: From Caveman to Homo Sapiens. Viitattu 3.5.2017. <https://cms2cms.com/blog/evolution-cms-platform-caveman-homo-sapiens/>
- Oksanen, R. 2014. Plone at University of Jyväskylä - 10 Years of Happiness. SlideShare 7.11.2014. Viitattu 20.4.2017. <https://www.slideshare.net/rioksane/ploneconf-oksanen2014>
- Oksanen, R. & Soukka, A. 2017. IT-palvelut. Jyväskylän Yliopisto. Haastattelu 9.3.2017.
- plone.app.theming 1.3.6. N.d. Viitattu 11.5.2017.
<https://pypi.python.org/pypi/plone.app.theming>
- Plone Foundation. 2017. Plone Mosaic. Viitattu 3.5.2017.
<https://github.com/plone/plone.app.mosaic>
- Plone (software). N.d. Wikipedia. Viitattu 3.5.2017.
[https://en.wikipedia.org/wiki/Plone_\(software\)](https://en.wikipedia.org/wiki/Plone_(software))
- Plone Solutions rebrands company as "Jarn". 2007. Plonen verkkosivut. Viitattu 3.5.2017. <https://plone.org/news/old-news/plone-solutions-rebrands-company-as-jarn>
- Security track record. N.d. Viitattu 3.5.2017. <https://plone.org/security/track-record>
- Stec, C. 2015. What's WYSIWYG? How Today's Online Editor Came to Be. Viitattu 3.5.2017. <https://blog.hubspot.com/marketing/history-of-the-wysiwyg-editor#sm.00012tqnraae2ecrs14220nph4d6f>
- The Demo: 1968. N.d. Viitattu 3.5.2017.
<http://www.doungengelbart.org/firsts/dougs-1968-demo.html>
- Usability Team. N.d. Plonen verkkosivut. Viitattu 3.5.2017.
<https://plone.org/community/usability>
- Walsh, M. 2011. Gov 2.0 guide to Plone. Viitattu 3.5.2017.
<http://govfresh.com/2011/03/gov-2-0-guide-to-plone/>
- Watkins, M. 1998. Flip Wilson, Outrageous Comic and TV Host, Dies at 64. Viitattu 3.5.2017. <http://www.nytimes.com/1998/11/27/arts/flip-wilson-outrageous-comic-and-tv-host-dies-at-64.html>
- What does Plone mean? How is it pronounced? N.d. Plonen dokumentaatio. Viitattu 3.5.2017. <https://docs.plone.org/intro/index.html#what-does-plone-mean-how-is-it-pronounced>
- What is Plone? N.d. Plonen dokumentaatio. Viitattu 3.5.2017.
<https://docs.plone.org/intro/index.html#what-is-plone>
- Weitershausen, P. 2008. Web Component Development with Zope 3. Springer. 3. painos

WordStar. N.d. Wikipedia. Viitattu 3.5.2017. <https://en.wikipedia.org/wiki/WordStar>

Wozniak, H. 2010. Leveraging Plone for Search Engine Optimization (SEO). SlideShare 21.11.2010. Viitattu 15.5.2017. <https://www.slideshare.net/heatherwoz/leveraging-plone-for-search-engine-optimization-seo>

Liitteet

Liite 1. Koodaamalla rakennetut tiilet

Facebook-tiilen koodi:

```
<iframe
src="https://www.facebook.com/plugins/page.php?href=https%3A%2F%2Fwww.facebook.com%2FJyvaskylaUniversity&tabs=timeline&width=340&height=500&small_header=false&adapt_container_width=true&hide_cover=false&show_facepile=true&appId"
width="340" height="500" style="border:none;overflow:hidden"
scrolling="no" frameborder="0" allowTransparency="true">
</iframe>
```

Twitter-tiilen koodi:

```
<a class="twitter-timeline"
href="https://twitter.com/uniofjyvaskyla" data-tweet-limit="3">Tweets by uniofjyvaskyla
</a>
<script async src="//platform.twitter.com/widgets.js" charset="utf-8">
</script>
```

Yhteistyökumppanit-tiilen koodi:

```
<div class="lista">
<h4>Yhteistyötahot</h4>
<ul class="lista">
<li class="lista1">

<a href="http://caffeineoverdose.fi/">
<h5>Caffeine Overdose Oy</h5></a>
<p>Lorem ipsum dolor sit amet...</p>
</li>
<li class="lista1">

<a href="http://www.fasuhouse.com/"><h5>Family Support House Oy</h5></a>
<p>Lorem ipsum dolor sit amet...</p>
</li>
<li class="lista1">

<a href="http://www.gamesforhealth.fi/"><h5>Games For Health Finland</h5></a>
<p>Lorem ipsum dolor sit amet...</p>
```

```

        </li>
        <li class="lista1">
            
            <a href="https://www.mll.fi/"><h5>Mannerheimin Lastensuo-
jeluliitto</h5></a>
            <p>Lorem ipsum dolor sit amet...</p>
        </li>
        <li class="lista1">
            
            <a href="https://www.thl.fi/fi/"><h5>Terveyden ja hyvin-
voinnin laitos</h5></a>
            <p>Lorem ipsum dolor sit amet...</p>
        </li>
    </ul>

<style>
div.lista {
    background-color: lightgray;
    min-width: 500px;
    max-width: 600px;
    padding: 10px;
    padding-bottom: 0;
    border: 1px solid gray;
}

li h5 {
    font: 20px;
    padding-top: 20px;
}

ul.lista {
    list-style-type: none;
}

li.lista1 {
    overflow: auto;
    min-height: 100px;
    margin-left: -20px;
}

li.lista1 img {
    float: left;
    margin-right: 15px;
    border: 1px solid gray;
    border-radius: 5px;
    height: 100px;
    width: 100px;
}
</style>
</div>

```

Tekijät-tiilen koodi:

```

<div class="ihmiset">
<h4>Tutkijat</h4>
<ul class="ihmiset">
  <li class="ihmiset1">
    
    <h5 class="red">Anna Rönkä</h5>
    <p>anna.k.ronka@jyu.fi</p>
  </li>
  <li class="ihmiset1">
    
    <h5 class="red">Eija Sevón</h5>
    <p>eija.sevon@jyu.fi</p>
  </li>
</ul>
<h4>Tohtorikoulutettavat</h4>
<ul class="ihmiset">
  <li class="ihmiset1">
    <img src=""/>
    <h5 class="red">Outi Alakärppä</h5>
    <p>outi.j.alakarppa@jyu.fi</p>
  </li>
</ul>

<style>
div.ihmiset {
  background-color: rgba(255, 125, 79, 0.7);
  padding: 10px;
  border: 1px solid gray;
}

ul.ihmiset {
  list-style-type: none;
}

li.ihmiset1 {
  overflow: auto;
  min-height: 100px;
  margin-left: -30px;
  margin-bottom: 10px;
}

li.ihmiset1 img {
  float: left;
  margin-right: 15px;
  border: 1px solid gray;
  border-radius: 10px;
}

```

```
    max-height: 100px;
    max-width: 100px;
}
</style>
</div>
```

Infolaatikko-tiilen koodi:

```
<div class="info">
<h4>Projektin kesto</h4>
<h5 class="red">2017-2018</h5>
<h4>Rahoittaja</h4>
<h5 class="red">Tieteen Tiedotus ry</h5>
<h4>Hankkeen kotisivut</h4>
<a
href="https://www.jyu.fi/edupsy/fi/laitokset/kas/babytrail"><h5
class="red">BabyTrail</h5></a>

<style>
div.info {
    background-color: rgba(255, 125, 79, 0.7);
    padding: 10px;
    border: 1px solid gray;
}
</style>
</div>
```

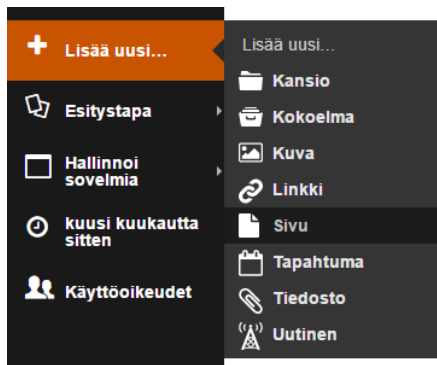
Liite 2. Ohjeet tutkimushankesivun luomiseen ja sisällön täyttämiseen

Ohjeet tutkimushankesivun luomiseen ja sisällön täyttämiseen

Tutkimushankesivun luominen

Ensimmäisenä pitää lisätä perussivu, vaihtaa sen esitystapa Mosaic-layoutiksi ja sitten sivuun pitää valita valmiista asettelutyyleistä oikea.

Kun olet sivustolla siellä paikassa, minne haluat tutkimushankesivun lisätä valitse vasemmalla olevasta valikosta Lisää uusi kohdasta Sivuu.



Eteen tulee sivu, jossa täytetään tutkimushankesivun perustiedot. Tässä kohtaa pakollisia kohtia täyttää on vaan tutkimushankkeen nimi Nimikekenttään. Kuvauskenttään voi lisätä hankkeen lyhyen kuvauksen ja tekstikenttään tulee laittaa hankkeen esittelyteksti. Molemmat pystytään lisäämään myös myöhemmin. Tutkimushankesivun URL-osoite muodostuu ylhäällä näkyvästä URL-osoitteesta ja Nimikekenttään kirjoitetusta nimestä. Esimerkiksi jos paikka mihin sivua lisätään, on osoitteessa <https://www.jyu.fi/hankkeet> ja Nimikekenttään kirjoitetaan nimeksi Hanke, on lisätyn tutkimushankkeen URL-osoite <https://www.jyu.fi/hankkeet/hanke>.

Välilehdessä Settings pystyy vaihtamaan url-osoitteeseen tuleva loppu Short name -kohdassa.

Sivu

Välilehdestä Kategorisointi pystyy lisäämään sivuun tageja.

Sivu

Perustiedot Settings Kategorisointi Päivät Omistaja Layout

Tagit Tägeja käytetään yleisesti sisällön ad-hoc järjestämiseen.

Kieli

Suomi x ▾

Samasta aiheesta

Current path: [🏠](#) / tutkimushanke-kansio

Tallenna Peruuta

Välilehdellä Päivät pystyy antamaan sivulle voimaantulopäivän ja viimeisen voimassaolopäivän kellon tarkasti.

Sivu

Perustiedot Settings Kategorisointi Päivät Omistaja Layout

Voimaantulopäivä Päivä, jolloin julkaistu sisältö ilmestyy näkyviin. Jos päivää ei valita, on sisältö näkyvä välittömästi.

Syötä päivämäärä... Syötä aika

Viimeinen voimassaolopäivä Sisällön vanhenemispäivä, jolloin sisältö piilotetaan automaattisesti. Jos vanhenemispäivää ei määritellä, sisältö pysyy julkaistuna.

Syötä päivämäärä... Syötä aika

Tallenna Peruuta

Välilehdellä Omistaja näkyy, kuka sivun on tehnyt ja sinne voi lisätä ihmisten nimiä, jotka ovat auttaneet sivun luomisessa. Välilehdessä pystyy kirjoittamaan sivulle erilliset tekijänoikeustiedot.

Sivu

Perustiedot Settings Kategorisointi Päivät Omistaja Layout

Tekijät Tämän sisältökohteen tekijät. Lisää tekijät kukin omalle rivilleen. Pääasiallisen tekijän tulisi olla ensimmäisenä.

✕ Elli Manninen

Sisällön luomiseen osallistuneet avustajat: Tämän sisällön tuottamiseen osallistuneet avustajat. Jokaisen nimen tulisi olla omalla rivillään.

Oikeudet Sisällön tekijänoikeustiedot.

Tallenna Peruuta

Välilehdellä Layout pystyy lisäämään sivuun erillisen asettelun koodilla. Tätä voi käyttää, jos sivun teemoitus on tarvetta tehdä koodilla. Välilehdeltä löytyy muitakin asettelun asetuksia erikoistarpeisiin.

Sivu

Perustiedot
Settings
Kategorisointi
Päivät
Omistaja
Layout

Content Layout Selected content layout. If selected, custom layout is ignored.

Custom layout Custom content and content layout of this page

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en" data-layout="*/@page-site-layout">
<body>
  <div data-panel="content"
    data-max-columns="4">
  <div class="mosaic-grid-row">
```

Site layout Site layout to apply to this page instead of the default site layout

Section site layout Site layout to apply to sub-pages of this page instead of the default site layout

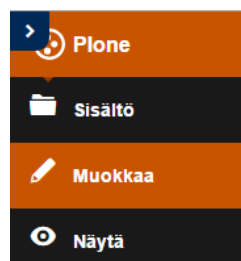
Tallenna
Peruuta

Kun tutkimushankesivun tietojen täyttäminen on valmista, painetaan Tallennappia ja sivu on luotu. Aikaisemmin esiteltyyn peruseditoriin pääsee muokkaamaan tietoja vielä myöhemmin Mosaic-editorin ylävalikosta Properties-napin kautta.

Luodun tutkimushankesivun esitystapa pitää muuttaa Mosaic-layoutiksi, jotta seuraavaksi pystytään valitsemaan asettelutyylisi sivulle. Esitystavan pystyy vaihtamaan vasemmalla olevasta valikosta Esitystapa kohdasta valitsemalla Mosaic-layout.

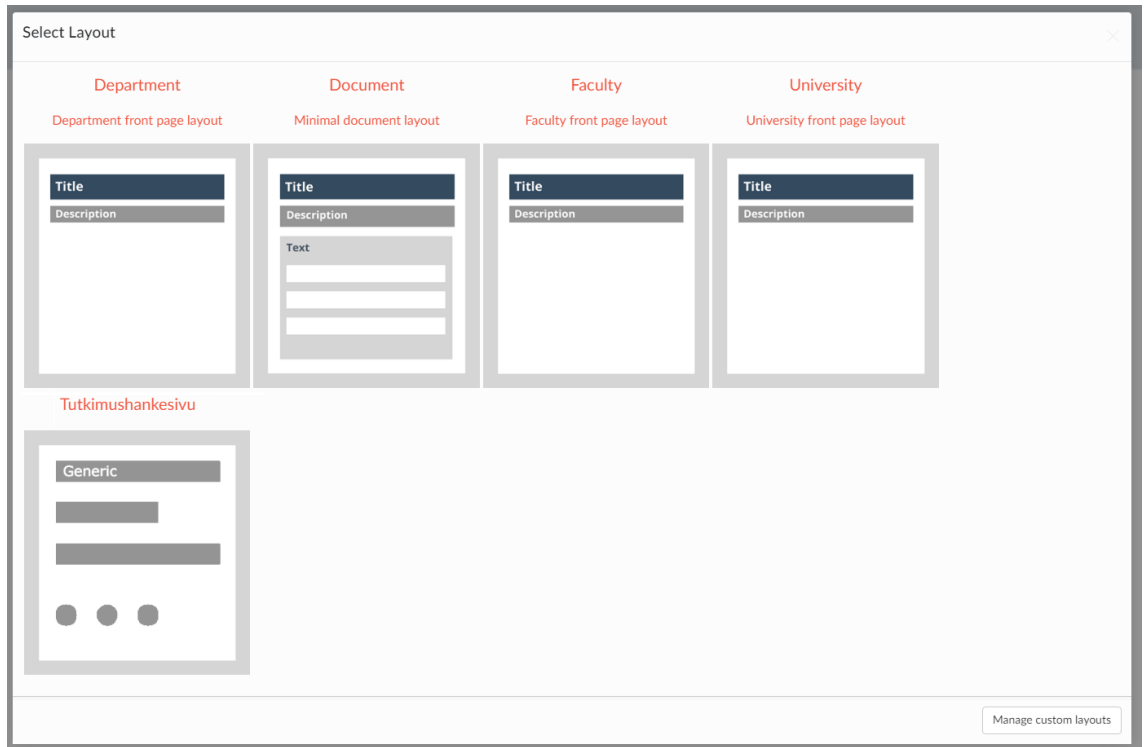


Tutkimushankesivun asettelutyylin valitseminen onnistuu klikkaamalla vasemmasta valikosta Muokkaa.



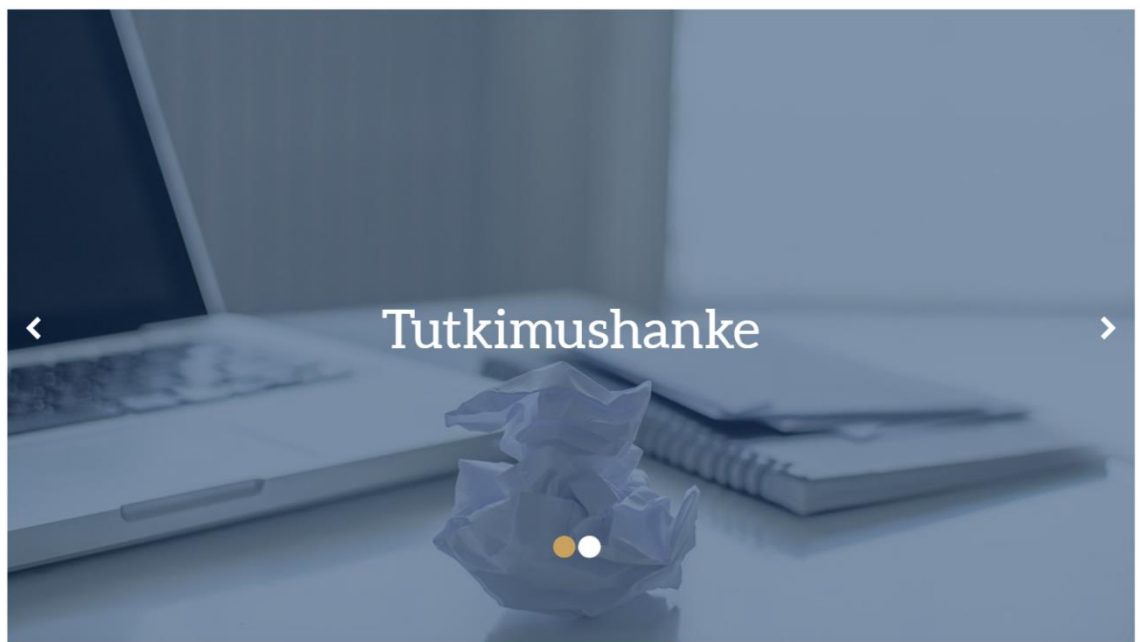
Esiin tulee rivi asettelutyylin vaihtoehtoja, joista pystyy valitsemaan yhden sivuun. Tutkimushankesivuun käytettävä asettelutyyli on nimeltään Tutkimushankesivu ja

valitsemalla sen sivuun luodaan tutkimushankkeissa käytettävä perusrakenne sisällölle.

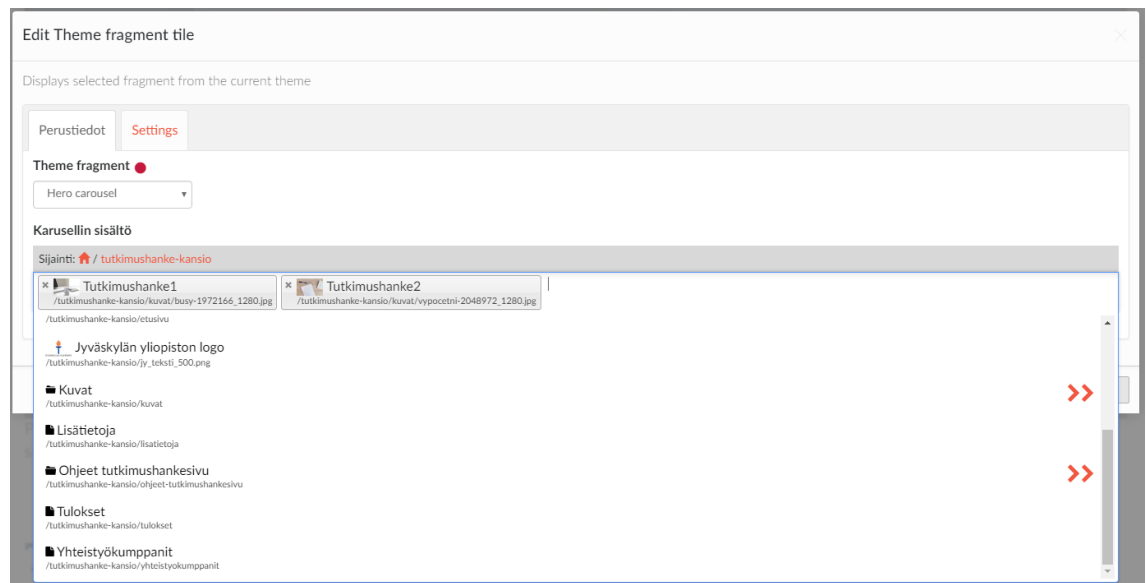


Tutkimushankesivun sisällön täyttäminen

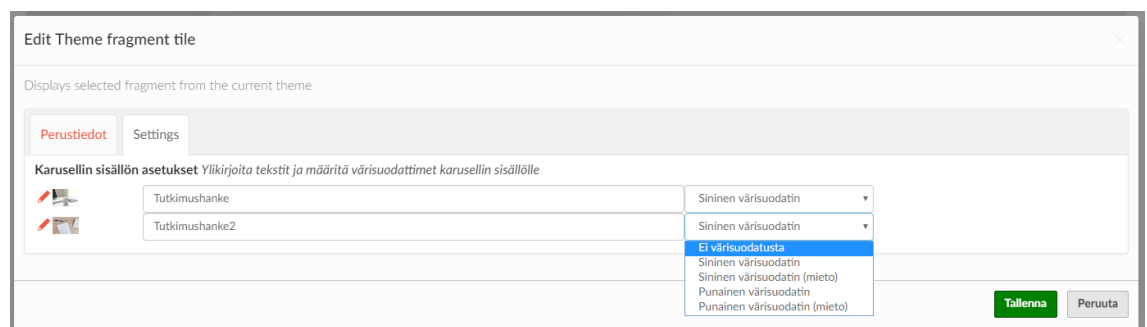
Kuvakaruselli:



Kuvakaruselliin tulee tutkimushankkeeseen liittyviä kuvia, joiden päällä voi olla tekstiä. Kuvakarusellin sisällön vaihtaminen onnistuu valitsemalla kentän ja painamalla ilmestyvää Edit-nappia. Karusellin kuvat pystytään lisäämään klikkaamalla valkoista aluetta sijainti kohdassa. Sijainti osoiterivillä lukee mistä parhaillaan kuvia etsitään ja kansioden sisälle pääsee nuolia painamalla. Takaisin päin pääsee klikkaamalla osoiterivillä edellisenä olevaa nimeä ja jos haluaa etsiä kuvia sivuston pääsivulta lähtien voi painaa osoiterivillä olevaa talo kuvaketta. Jokainen kuva pitää lisätä yksi kerrallaan samalla tyyllillä.



Kuvien tekstin pystyy määrittämään Settings-välilehdellä yksi kerrallaan jokaiseen kuvaan lisäämällä tekstin kuvan kohdalla olevaan tekstikenttään. Kuvien päälle voi valita myös värisuodattimen halutessaan.



Nimike:

Nimike: **Tutkimushanke**

Nimikekenttään, joka on ensimmäisenä, tulee tutkimushankkeen nimi. Kentässä näkyvä nimi täyttyy automaattisesti sivua luodessasi antamalla sivun nimellä, mutta voit muuttaa nimeä vielä halutessasi ylikirjoittamalla sen. Nimen muuttaminen täällä ei muuta selaimessa näkyvää URL-osoitteen loppua, jossa sivun nimi näkyy. Jos haluat muuttaa URL-osoitteessa näkyvää sivun nimeä, sinun täytyy mennä ylhäällä näkyvän Properties-napin kautta asetuksiin ja sieltä Settings-välilehdelle. Settings-välilehdellä on kohta Short name, ja siinä kohdassa olevan tekstin vaihtamalla voit muuttaa URL-osoitteen loppua.

Kuvaus:

Kuvaus

Kuvauskenttään tulee hankkeen kuvaus. Kuvaus voi esimerkiksi olla hankkeen nimen pidempi versio. Kuvauskentässä näkyy automaattisesti kuvausteksti, jos sivun luonnin yhteydessä sellainen kirjoitettiin.

Ensimmäinen tekstikenttä:

Teksti Tähän kenttään tulee tutkimushankkeen lyhyt esittelyteksti.

Ensimmäiseen tekstikenttään tulee tutkimushankkeen lyhyt esittelyteksti. Kentässä näkyy automaattisesti esittelyteksti, jos sivun luonnin yhteydessä sellainen kirjoitettiin.

Infolaatikko:

Yleistiedot

Projektin kesto
2017-2018

Rahoittaja
Jyväskylän yliopisto







Tutkimushankkeen kotisivut
[Jyväskylän yliopisto](#)

Tähän kenttää tulee tutkimushankkeen perustiedot. Otsikoihin ei tarvitse koskea, mutta niiden alla olevat kohdat korvataan tutkimushankkeen tiedoilla: tutkimushankkeen kesto, rahoittaja(t) ja erillisen kotisivun osoite. Otsikot ovat kokoa 3 ja tie-

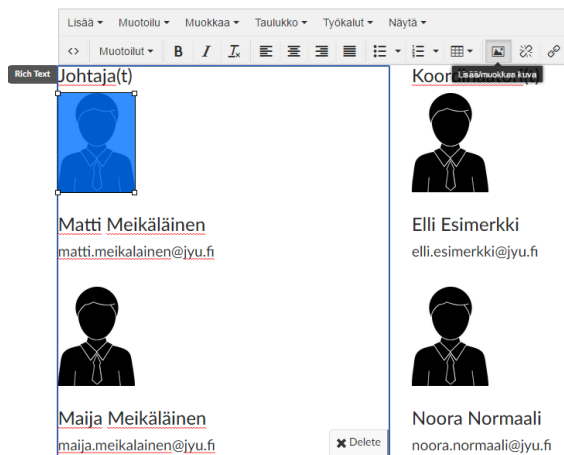
totekstien koko on 5. Viimeisenä oleva Tutkimushankkeen kotisivut kohdan voi poistaa kokonaan, jos tutkimushankkeella ei ole muita kotisivuja.

Tekijät:

Tutkimushankkeen tekijät

Rich Text Johtaja(t)	Koordinaatori(t)	Tohtorikoulutettava(t)
 <p data-bbox="389 622 576 680">Matti Meikäläinen matti.meikalainen@jyu.fi</p>	 <p data-bbox="815 622 967 680">Elli Esimerkki elli.esimerkki@jyu.fi</p>	 <p data-bbox="1241 622 1409 680">Tarja Tavallinen tarja.tavallinen@jyu.fi</p>
 <p data-bbox="389 855 576 913">Maija Meikäläinen majja.meikalainen@jyu.fi</p>	 <p data-bbox="815 855 983 913">Noora Normaali noora.normaali@jyu.fi</p>	 <p data-bbox="1241 855 1445 913">Seppo Suomalainen seppo.suomalainen@jyu.fi</p>

Tutkimushankkeen tekijät kohdassa listataan kaikki tutkimushankkeeseen osallistuvat ihmiset oikeiden titteleiden alle. Titteliotsikoita voi muuttaa ja lisätä tarvittaessa kuvaamaan paremmin osallistuvat ihmiset. Kuvien lisääminen ei ole pakollista, mutta jos niitä käytetään, kuvan kooksi asetuksista pitää valita näyte (128x128). Kuvan pysyy lisäämään joko etsimällä kuvan sivustolta, lataamalla kuvan koneelta tai lisäämällä kuvan ulkoisen linkin avulla.



Lisää kuva

Specify an image. It can be on this site already ("Internal Image"), an image you upload ("Upload"), or from an external site ("External Image").

Internal Image **Upload** External Image

Sijainti: / tutkimushanke-kansio

Kuvake Koko: Näyte (128 x 128)

Otsikko

Vaihtoehtoinen teksti

Tasaus: Inline

Cancel **Lisää**

Lisätietoja/yhteystiedot:

Rich Text: **Lisätietoja/yhteystiedot**

Tätä kenttää ei ole pakko sisällyttää tutkimushankesivun rakenteessa.

Tähän kenttään tulee tarvittaessa lisätietoja tutkimushankkeesta tai yhteystiedot, johon ihmisten halutaan ottavan yhteyttä.

Tätä kenttää ei ole pakko sisällyttää tutkimushankesivun rakenteessa. Tätä kenttää voi käyttää tutkimushankkeen lisätietojen näyttämiseen, jos sille on tarvetta tai yhteystietojen, johon ihmisten halutaan ottavan yhteyttä.

Ajankohtaista:

Theme fragment

AJANKOHTAISTA ITP

12.05.2017	09.05.2017	27.04.2017	25.04.2017
Seminaarinmäellä laaja, lyhyt tietoliikenneyhteyksien katkos 12.05.2017 klo 13:20-13:50	Converis-päivitys ke 10.5.2017 klo 15:30	Huomio opiskelijat: Uusi eTentti-tila 2.5.2017 alkaen	Converis-päivitys ke 26.4.2017 klo 15:30

Ajankohtaista

Tähän kenttään tulee tutkimushankkeen ajankohtaiset tiedot esimerkiksi uutiset. Voit listata ajankohtaiset asiat allekkain aikajärjestykseen, tekstillä tai vaikka ihan linkkejä käyttämällä. Voit käyttää hankkeen uutisten esittämiseen myös edellistä kenttää, jos hankkeen uutisista löytyy RSS-syöte. Vaihdat vain edellisen kentän lähteeksi oman RSS-syötteen, valitsemalla kentän ja painamalla ilmestyvää Edit-nappia.

Tähän kenttään tulee tutkimushankkeen ajankohtaiset tiedot esimerkiksi uutiset. Voit listata ajankohtaiset asiat allekkain aikajärjestykseen, tekstillä tai vaikka ihan linkkejä käyttämällä. Voit käyttää hankkeen uutisten esittämiseen myös edellistä kenttää, jos hankkeen uutisista löytyy RSS-syöte. Vaihdat vain edellisen kentän lähteeksi oman RSS-syötteen, valitsemalla kentän ja painamalla ilmestyvää Edit-nappia. Asetuksissa RSS-syötteen osoite kirjoitetaan sille varattuun tekstikenttään ja syötteitä voi olla useampi. Syöttelelle voi antaa nimen osoitekentän jälkeiseen kenttään ja syötteiden järjestystä voi muuttaa Up- ja Down-napilla.

Edit Theme fragment tile

Displays selected fragment from the current theme

Theme fragment ●

News feeds carousel

RSS-syötteet

https://www.jyu.fi/itp/ajankohtaista/ajankohtaista_etu/RSS	Ajankohtaista ITP	Up	Down	Remove
URL	Label	Up	Down	Remove

Tallenna Peruuta

Tulokset/Julkaisut:

Rich Text

Tulokset/Julkaisut

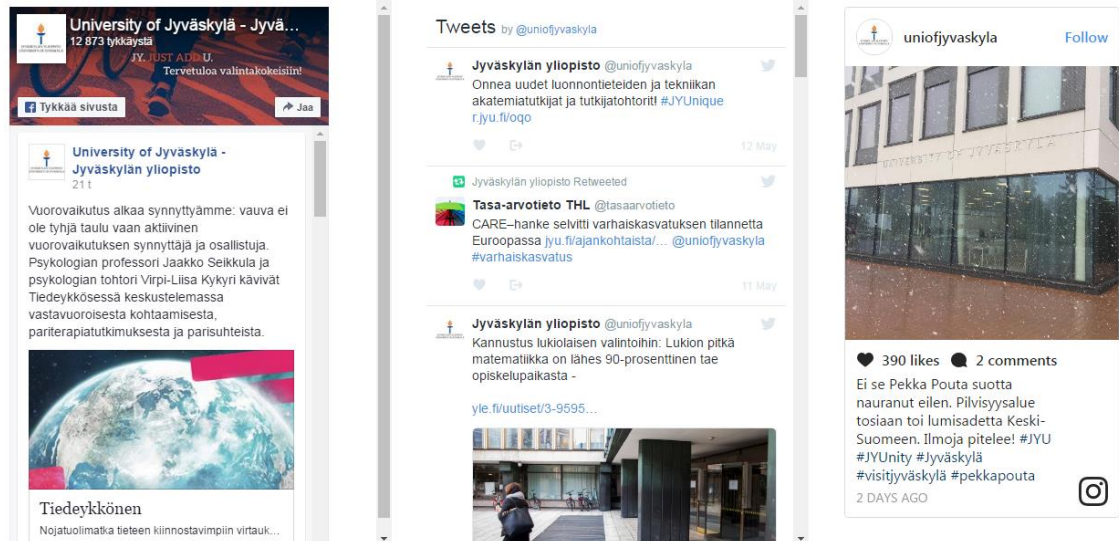
Tähän kenttään tulee tutkimushankkeen tulokset/julkaisut. Voit listata tulokset allekkain aikajärjestykseen ja käyttää linkkejä tai ihan vaan tekstiä. Linkit voivat viedä tiedostoihin tai ulkoisiin tiedonlähteisiin. Jos tuloksia/julkaisuja on paljon, on parasta listata uusimmat niistä tähän kenttään ja tehdä kokonaan eri sivu, jossa on listattu kaikki tutkimushankkeen tulokset/julkaisut. Viimeisenä kannattaa lisätä tämän kentän loppuun linkki, mikä vie tutkimushankkeen tulokset/julkaisut sivulle, jos sellainen oli tarvetta luoda.

Tähän kenttään tulee tutkimushankkeen tulokset/julkaisut. Voit listata tulokset allekkain aikajärjestykseen ja käyttää linkkejä tai ihan vaan tekstiä. Linkit voivat viedä tiedostoihin tai ulkoisiin tiedonlähteisiin. Jos tuloksia/julkaisuja on paljon, on parasta listata uusimmat niistä tähän kenttään ja tehdä kokonaan eri sivu, jossa on listattu kaikki tutkimushankkeen tulokset/julkaisut. Viimeisenä kannattaa lisätä tämän ken-

tän loppuun linkki, mikä vie tutkimushankeen tulokset/julkaisut sivulle, jos sellainen oli tarvetta luoda.

Sosiaalinen media:

Sosiaalinen media



Kaikkien kyseisten tiilien sisällön muuttamiseen pääsee valitsemalla haluttu tiili muokkaukseen klikkaamalla sitä ja painamalla Edit-nappia.

Facebook-tiilen asetusten ensimmäisessä kohdassa täytyy antaa halutun Facebook-sivun URL-osoite kokonaisuudessaan. Seuraavaksi pitää valita, mitkä välilehdet tiileen haluaa sisällyttää ja sekä minkä näköisenä haluaa Facebookin näkyvät tiilessä valitsemalla lisäasetuksista halutut kohdat. Käytä pientä otsikkoa, piilota kansikuva ja näytä kaverien naamat ovat kaikki vapaavalintaisia asetuksia eli niistä ei pakko valita yhtäkään, jos ei halua. Viimeisenä asetuksissa on Facebook-upotuksen leveys ja korkeus. Leveys on vapaaehtoinen arvo, mikä määrää Facebook-upotuksen leveyden tiilen sisällä, mutta ei varsinaisen tiilen leveyttä, Korkeus arvo määrää Facebook-upotuksen korkeuden tiilen sisällä, mutta koska tiilien korkeudella ei ole rajoitetta, annettu korkeus vaikuttaa myös tiilen korkeuteen. Korkeus on siis tämän takia pakollinen tieto. Leveyden ja korkeuden arvo kerrotaan pikseleiden määränä.

Edit Theme fragment tile ✕

Displays selected fragment from the current theme

Theme fragment ●

Facebook embed ▼

Facebook-sivun web-osoite ●

Välilehdet

Aikajana

Viestit

Tapahtumat

Käytä pientä otsikkoa

Piilota kansikuva

Näytä kaverien naamat

Leveys *Upotuksen leveys pikseleinä (voi jättää tyhjäksi)*

Korkeus ● *Upotuksen korkeus pikseleinä*

Tallenna Peruuta

Twitter-tiilen asetusten ensimmäisessä kohdassa täytyy antaa halutun Twitter-tilin käyttäjätunnus. Viimeisenä asetuksissa on Twitter-upotuksen leveys ja korkeus. Twitter-tiilen leveys ja korkeus asetukset toimivat ihan samalla tavalla kuin Facebook-tiilenkin samat asetukset.

Edit Theme fragment tile ✕

Displays selected fragment from the current theme

Theme fragment ●

Twitter timeline ▼

Twitter-käyttäjätunnus ●

Leveys *Upotuksen leveys pikseleinä (voi jättää tyhjäksi)*

Korkeus ● *Upotuksen korkeus pikseleinä*

Tallenna Peruuta

Instagram-tiilen asetusten ensimmäisessä kohdassa täytyy antaa halutun Instagram-kuvan URL-osoite kokonaisuudessaan. Viimeisenä asetuksissa on Instagram-upotuksen leveys ja korkeus. Instagram-tiilen leveys ja korkeus asetukset toimivat ihan samalla tavalla kuin Facebook-tiilenkin samat asetukset.

✕ Edit Theme fragment tile

Displays selected fragment from the current theme

Theme fragment ●

Instagram embed ▼

Instagram-valokuvan web-osoite ●

Leveys Upotuksen leveys pikseleinä (voi jättää tyhjäksi)

Korkeus ● Upotuksen korkeus pikseleinä

Yhteistyökumppanit:



Yhteistyökumppanit kohta muodostuu tekstikenttien asettelusta. Jokaisen tekstikentän rakenne sisältää yhteistyökumppanin logo-kuvan, nimen ja pienen esittelytekstin. Logo-kuva on lisätty lisää kuva painikkeella, ja jota painamalla esiin tulee kuvan lisäysvalikko. Kuvan pystyy lisäämään joko etsimällä kuvan sivustolta, lataamalla kuvan koneelta tai lisäämällä kuvan ulkoisen linkin avulla. Kuvan kooksi on suositeltavaa valita joko keskikokoinen (400x400) tai pieni (200x200). Kuvalla voi antaa halutessa otsikon ja vaihtoehdoisen tekstin, mitkä parantavat kuvan näkyvyyttä haussa. Yhteistyökumppanin nimi on linkki yhteistyökumppanin kotisivuille ja nimen fonttikokona on käytetty Otsikko 4. Yhteistyökumppanin esittelyteksti kannattaa pitää mahdollisimman lyhyenä, eikä sitä ole pakko sisällyttää olleenkaan, jos sille ei ole tarvetta. Jos yhteistyökumppaneista on tarvetta kertoa enemmän kuin muutaman lausetta kannattaa yhteistyökumppaneille tehdä erillinen sivu tutkimushankesivun alle. Tekstikenttien määrää voi vähentää klikkaamaan tekstikenttää ja painamalla Delete-nappia. Tekstikenttiä voi olla korkeintaan neljä rinnakkain ja niitä voi lisätä ylhäältä löytyvän Insert-valikon kautta.

Lisää kuva

Specify an image. It can be on this site already ("Internal Image"), an image you upload ("Upload"), or from an external site ("External Image").

Internal Image Upload External Image

Current path: [/ tutkimushanke-kansio](#)

Koko: Original

- Original
- Keskikokoinen (400 x 400)
- Näyte (128 x 128)
- Pieni (200 x 200)
- Suuri (768 x 768)
- banner-tall 1200x600

Otsikko

Vaihtoehtoinen teksti

Tasaus: Inline

Cancel Lisää

Lisää kuva

Specify an image. It can be on this site already ("Internal Image"), an image you upload ("Upload"), or from an external site ("External Image").

Internal Image Upload External Image

Drag and drop files from your computer onto the area below or click the "Browse" button.

Raahaa ja pudota tiedostot tähän...

Valitse tiedosto Browse

Otsikko

Vaihtoehtoinen teksti

Tasaus: Inline

Cancel Lisää

Lisää kuva

Specify an image. It can be on this site already ("Internal Image"), an image you upload ("Upload"), or from an external site ("External Image").

Internal Image Upload External Image

External Image URL (can be relative within this site or absolute if it starts with http:// or https://)

Otsikko

Vaihtoehtoinen teksti

Tasaus: Inline

Cancel Lisää