

Web-sisällönhallintajärjestelmien ominaisuudet

Antti Lopenen

Tampereen yliopisto
Informaatiotieteiden yksikkö / tko
Pro gradu -tutkielma
Ohjaaja: Timo Niemi
15.5.2012

Tampereen yliopisto

Informaatiotieteiden yksikkö / tko

Antti Loponen: Web-sisällönhallintajärjestelmien ominaisuudet

Pro gradu -tutkielma, 57 sivua

Toukokuu 2012

Tiivistelmä

Web-sisällönhallintajärjestelmä on järjestelmä, jolla hallinnoidaan www-sivuston sisältöä. Sen erona tavallisiin html-tiedostoihin perustuvaan www-sivustoon on, että sisältö erotetaan näyttötavasta ja tallennetaan tietokantaan. Jokaista url-soitetta kohden tietokannassa on sisältö ja sivupohja, joista rakennetaan käyttäjälle näytettävä www-sivu. Sivuston sisällön, ulkoasun ja rakenteen muokkaus tapahtuu toisistaan riippumattomasti erillisestä ylläpitokäyttöliittymästä. Täten web-sisällönhallintajärjestelmät nopeuttavat ja helpottavat www-sivujen ylläpitoa ja mahdollistavat sen ilman merkittäviä ohjelmointitaitoja.

Tässä tutkielmassa tarkastellaan kolmea yleisintä web-sisällönhallintajärjestelmää: Joomlaa, Drupalia ja WordPressiä. Ne kaikki ovat ilmaisia, avoimella lähdekoodilla toteutettuja ja omalle palvelimelle asennettavissa olevia järjestelmiä.

Kyseisten web-sisällönhallintajärjestelmien käyttöliittymiä ja tietokantakaavioita tarkastelemalla ja analysoimalla tutkielmassa kootaan niiden yhteiset ja hyödyllisimmät ominaisuudet. Niiden perusteella tutkielmassa kehitetään yleinen käsittekaavio web-sisällönhallintajärjestelmästä. Sen avulla voidaan ymmärtää paremmin web-sisällönhallintajärjestelmien toimintaa tai mahdollisesti kehittää oma web-sisällönhallintajärjestelmä.

Asiasanat: CMS, WCMS, Web Content Management System, Joomla, Drupal, WordPress

KIITOKSET

Kiitoksia gradun ohjauksesta Timo Niemelle ja Hannu Kangassalolle. Kiitokset työnantajalleni Symmetria Softwarelle mahdollisuudesta opintovapaaseen. Kiitos ystävälleni Antti Mattilalle opastuksesta LateXin käytössä. Lämpimät kiitokset rakkaalle avovaimolleni Emilia Tuurille tuesta tutkielman kirjoittamisen aikana sekä asiantuntijuudesta oikolukuvaiheessa.

Tampereen Pispalassa 24.4.2012

- Antti Lopenen

SISÄLLYS

1	Johdanto	1
2	Web-sisällönhallintajärjestelmien lähtökohdat	3
2.1	Käytännön toiminta	3
2.2	Tiedonhallinnallinen näkökulma	6
3	Yleisimmät web-sisällönhallintajärjestelmät	8
3.1	Joomla	8
3.2	Drupal	9
3.3	WordPress	10
3.4	Muut järjestelmät	11
3.5	Web-sisällönhallintajärjestelmiä muistuttavat ratkaisut	12
4	Web-sisällönhallintajärjestelmien keskeiset ominaisuudet	14
4.1	Näkymät ja niihin liittyvät toiminnot	14
4.2	Tunnisteet	16
4.3	Ylläpitokäyttöliittymä	17
4.4	Tietokantaratkaisut	20
4.5	Yhteenvedo vertailtavista järjestelmistä	26
4.6	Keskeisten ominaisuuksien ilmeneminen vertailtavissa järjestelmissä	27
5	Web-sisällönhallintajärjestelmän yleinen määrittely	29
5.1	Sivusto	29
5.2	Sivu	30
5.3	Sivuversio	32
5.4	Sivupohja	33
5.5	Sisältöelementti	35
5.6	Sisältöelementtityyppi	36
5.7	Kategoria	38
5.8	Tunniste	38
5.9	Kommentti	39
5.10	Käyttäjä	40
5.11	Käyttäjryhmä	41
5.12	Käyttöoikeustaso	42
5.13	Yksityisviesti	43
5.14	Koko järjestelmän käsitekaavio	44
6	Kritiikkiä web-sisällönhallintajärjestelmiä kohtaan	46
6.1	Käytettävyys	46

6.2	Suorituskyky	46
6.3	Tietoturva	47
6.4	Muita näkökohtia	47
7	Lopuksi	49
	Viiteluettelo	51
A	IDEF1X-visualisointitapa	55

1 JOHDANTO

World Wide Webin juuri synnyttyä 1990-luvun alussa internetin sisällöstä vastasivat suurimmaksi osaksi alan harrastajat ja organisaatiot sekä muutamat edistykselliset yritykset. Kaikki, joilla oli tarve saada oma kotisivu internetiin, osasivat tehdä tarvittavan html-koodin itse, tai sitten he saattoivat tilata sellaisen ammattilaiselta. Koska sivustoilla voitiin käydä vain harvoin, ei niitä tarvinnut päivittääkään usein.

Vuosituhanneen vaihteen aikoihin kuitenkin internetin sisällön määrä alkoi kasvaa räjähdysmäisesti, kun useammat yritykset, organisaatiot ja yksityiset henkilöt halusivat osallistua internetin suomiin mahdollisuuksiin. Sisältöä piti voida tuottaa verkkoon helposti ja nopeasti, ilman ohjelmointitaitoja. Internet-yhteyksien nopeudet kasvoivat ja hinnat halpenivat, jolloin sivustoilla voitiin käydä useammin ja niitä oli pystyttävä päivittämään useammin kuin ennen. Web-sisällönhallintajärjestelmien (*web content management system, WCMS*) kehityksen myötä näihin tarpeisiin pystyttiin vastaamaan.

Web-sisällönhallintajärjestelmiä käyttävissä sivustoissa sisältö ja esitystapa on erotettu toisistaan. Tietokantaan tallennetusta näyttöasusta riippumattomasta sisällöstä saadaan rakennettua www-sivu perustuen sivupohjiin, joihin sisältöä ladataan. Tietoa syötetään ja muokataan tietokantaan erillisen ylläpitokäyttöliittymän kautta. Toisin sanoen www-sivun muokkaus web-sisällönhallintajärjestelmällä on nopeaa, eikä se vaadi html-osaamista.

Web-sisällönhallintajärjestelmän käyttö mahdollistaa sivuston sisällön, ulkoasun ja rakenteen muokkaamisen toisistaan riippumatta. Lisäksi erilaisten haku- ja lajittelutoimintojen soveltaminen tai kertaalleen tallennetun tiedon käyttäminen uudelleen toisessa kontekstissa on mahdollista, koska sisältö on hyvin organisoidusti tallennettuna tietokantaan.

Markkinoilla on monia valmiita kaupallisia web-sisällönhallintajärjestelmiä, mutta silti suosituimmat web-sisällönhallintajärjestelmät ovat ilmaiseksi käyttöön otettavia ja avoimen lähdekoodin ansiosta myös omiin käyttötarpeisiin muokattavia. Jopa monet kaupalliset sivustot on rakennettu ilmaisilla avoimen lähdekoodin web-sisällönhallintajärjestelmillä.

Web-sisällönhallintajärjestelmiä on käytetty vasta noin vuosikymmenen verran, ja osa käytännön kehitystyöstä on jakautunut monille eri tahoille tuhansien riippumattomien kehittäjien toimesta. Tämän vuoksi web-sisällönhallintajärjestelmiä

tai niiden keskeisiä ominaisuuksia ei ole määritelty yleisesti ja tieteellisesti.

Tämän tutkielman eräänä tavoitteena on kartoittaa web-sisällönhallintajärjestelmien keskeiset ominaisuudet. Menetelmänä on tarkastella ja vertailla yleisimpien web-sisällönhallintajärjestelmien toimintoja, käyttöliittymiä ja tietokantaratkaisuja. Tavoitteena on löytää niistä samanlaisuuksia ja eroavaisuuksia. Tutkimuksessa keskitytään kolmeen suosituimpaan avoimen lähdekoodin web-sisällönhallintajärjestelmään: Joomlaan, Drupaliin ja WordPressiin. Niiden yhteisten ja hyödyllisimpien ominaisuuksien perusteella koostetaan lista web-sisällönhallintajärjestelmän keskeisistä ominaisuuksista sekä niiden välisistä suhteista. Tutkielmassa ne sisällytetään web-sisällönhallintajärjestelmälle kehitettyyn yleiseen käsittekaavioon.

Tämän käsittekaavion luominen on tarpeellista, sillä se tukee web-sisällönhallintajärjestelmää käyttävän ylläpitäjän ymmärrystä käyttämänsä järjestelmän toimintaperiaatteista. Lisäksi relevanttina voidaan pitää tarvetta ohjelmoida täysin uusi web-sisällönhallintajärjestelmä, jollaisen suunnittelu voidaan aloittaa tutkielmassa esitetyistä peruslähtökohdista. Ominaisuuksien teoreettinen listaaminen voi myös synnyttää jatkokehitysideoita web-sisällönhallintajärjestelmien käytön monipuolistamiseksi.

2 WEB-SISÄLLÖNHALLINTAJÄRJESTELMIEN LÄHTÖKOHDAT

The fundamental principle behind the Web was that once someone somewhere made available a document, database, graphic, sound, video or screen at some stage in an interactive dialogue, it should be accessible by anyone, with any type of computer, in any country.

Nämä mahtipontiset sanat ovat World Wide Webin luoja pidetyn Tim Berners-Leen kirjasta Weaving the Web [Berners-Lee & Fischetti, 1999]. Internet todellakin teki sisällön lukemisesta mahdollista lähes kenelle tahansa ja missä tahansa, mutta web-sisällönhallintajärjestelmät ovat mahdollistaneet myös sisällön tuottamisen, käsittelemisen ja julkaisemisen lähes kenelle tahansa. Tuskinpa Berners-Lee osasi edes haaveilla tästä 1990-luvun alussa rakennellessaan World Wide Webiä.

Jo puolivälissä World Wide Webin historiaa, vuonna 2001 järjestetyssä 14. Bledin e-Commerce-konferenssissa [eCe, 2001] todettiin internetin sisällön kasvavan räjähdysmäisesti ja sen seurauksena sisällön hallinnan vaikeutuvan. Hallinnoimattoman web-sisällön todettiin aiheuttavan informaatiokulun hidastumista, sisällön epäjohdonmukaisuutta, sivustojen navigaatioiden laadun heikkenemistä ja saman tiedon useaan kertaan tallentamista [eCe, 2001].

Samassa konferenssissa ennustettiin web-sisällönhallintajärjestelmien markkinoiden kymmenkertaistuvan viidessä vuodessa [eCe, 2001]. Ennuste web-sisällönhallintajärjestelmien tarpeellisuudesta oli osuva, mutta tuskin tuolloin osattiin aavistaa, miten suosittuja web-sisällönhallintajärjestelmät tulisivat olemaan jo muutamien vuosien kuluttua.

2.1 Käytännön toiminta

The days of writing HTML for large Web sites may be numbered as content management tools take the floor [Pon, 2009].

Web-sisällönhallintajärjestelmä mahdollistaa suurten web-sivustojen sisällön hallinnan [eCe, 2001] käsittäen sisällön luomisen, muokkaamisen, tallentamisen ja

julkaisemisen siten, että sen käyttäjältä ei edellytetä merkittäviä ohjelmointitaitoja [Pon, 2009]. Lisäksi web-sisällönhallintajärjestelmän avulla sisältöä voidaan luokitella ja indeksoida avustamaan tiedonhakua [Pon, 2009].

Shoara tutkijakollegoineen on esittänyt kahdeksan hyötynäkökantaa, jotka saadaan aikaan web-sisällönhallintajärjestelmällä [Shoara *et al.*, 2011]:

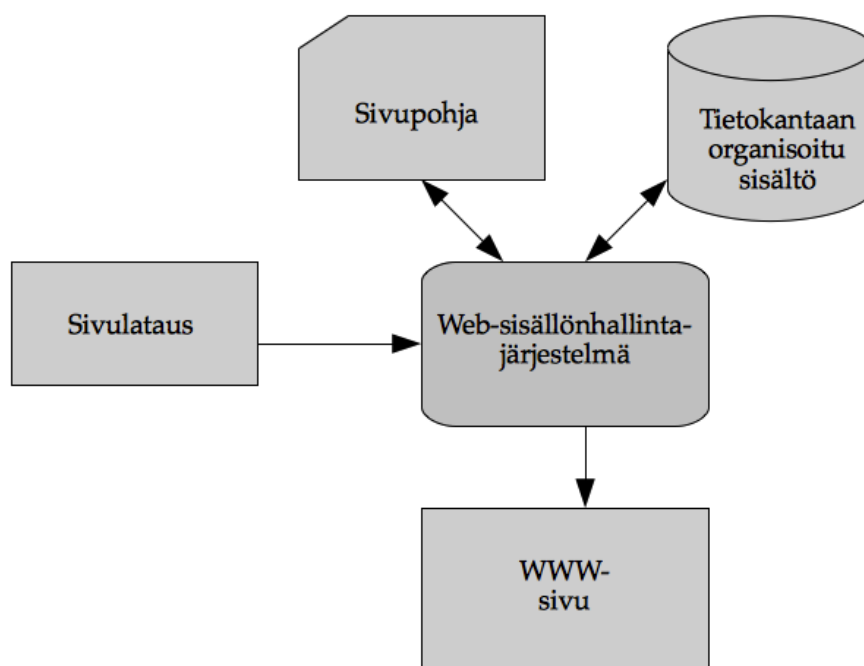
1. Ajan säästäminen sivuston ylläpidossa.
2. Talletettujen tietojen suurempi yhdenmukaisuus ja eheys.
3. Kehittyneemmän navigoinnin mahdollistaminen.
4. Monipuolisemmat mahdollisuudet sivuston laajentamiseen.
5. Hajautettu sivuston sisällön hallinta.
6. Paremman tietoturvan mahdollistaminen.
7. Samojen tietojen useaan kertaan tallentamisen välttäminen.
8. Huomattava säästö ylläpitokuluissa verrattuna perinteisiin staattisiin html-sivuihin.

Pelkästään nämä kahdeksan kohtaa puhuvat puolestaan, vaikka ne eivät edes sisällä mainintaa esimerkiksi helppokäyttöisyydestä ylläpitokäyttäjän näkökulmasta.

Yksinkertainen kärjistetty esimerkki web-sisällönhallintajärjestelmän käytöstä voisi olla seuraava. Käyttäjä kirjautuu sisään web-sisällönhallintajärjestelmään, luo uuden sivun, kirjoittaa tekstiä, muotoilee sitä, ja painaa lähetyspainiketta. Tämän jälkeen käyttäjän syöttämä teksti on välittömästi internetissä miljoonien ihmisten luettavissa. Ilman web-sisällönhallintajärjestelmää käyttäjän olisi pitänyt luoda uusi html-dokumentti ja tehdä sen esittämisessä tarvittava html-koodi, liittää sen sekaan sopivien tunnisteiden sisään, siirtää tiedosto palvelimelle ja linkittää se sivuston muihin osiin.

Tyypillisesti web-sisällönhallintajärjestelmä on itsessään web-sovellus, joka asennetaan ajettavaksi palvelimella [Pon, 2009]. Tämän jälkeen sivustolle ulkoapäin tehdyt sivulataukset aktivoivat web-sisällönhallintajärjestelmän toiminnon, joka

palauttaa tietokannasta tarkoituksenmukaisen sisällön ja siihen liittyvän sivupohjan. Sisällön ja sivupohjan perusteella piirretään selaimen kautta näytölle www-sivu (Kuva 2.1). Sivuston ylläpitäjät voivat sen sijaan kirjautua sisään erillisestä osoitteesta ylläpitokäyttöliittymään, josta varsinaista sivuston sisältöä hallitaan.



Kuva 2.1 Prosessikaavio web-sisällönhallintajärjestelmän toiminnasta

Joillakin kotisivupalveluntarjoajilla on myös valmiita paketteja, jotka sisältävät web-osoitteen lisäksi web-sisällönhallintajärjestelmän palvelimelle valmiiksi asennettuna [Frick, 2010]. Tällöin käyttäjä pääsee luomaan web-sivuston todella vähäisellä ohjelmointi- ja palvelinkonfigurointitaidolla. Tosin haittapuolina ovat sekä suuret kustannukset että riski siitä, että palveluntarjoajayritys lopettaa toimintansa, jolloin koko sivuston tietokanta lopettaa olemassaolonsa [Frick, 2010].

Web-sisällönhallintajärjestelmän asentaminen omalle palvelimelle ei kuitenkaan ole välttämättä työlästä, jos hyödynnetään palvelinkäyttöliittymäavustajia, kuten cPanelia tai Fantasticoa [Frick, 2010]. Niiden käyttö saattaa tosin rajoittaa asennettavan web-sisällönhallintajärjestelmän konfigurointia.

Tietenkin varminta on tehdä web-sisällönhallintajärjestelmän asennus ja konfigurointi manuaalisesti tai pyytää siihen asiantuntija-apua. Todennäköistä on, että asiantuntija-apua ei tarvita huolellisesti tehdyn asennuksen jälkeen web-sisällönhallintajärjestelmän käytön myöhemmissä vaiheissa [Pon, 2009].

2.2 Tiedonhallinnallinen näkökulma

Information quality can be no better than the supporting data resource quality [Brackett, 2000].

Web-sisällönhallintajärjestelmät mahdollistavat erityyppisten sisältöjen käsittelyn systemaattisesti. Web-sivustolla oleva tai sille haluttava sisältö voi olla huonosti organisoitua [eCe, 2001], mutta web-sisällönhallintajärjestelmät ohjaavat sisällön parempaan organisointiin.

Web-sisällönhallintajärjestelmän avulla www-sivulla oleva teksti jaetaan esimerkiksi otsikoihin ja leipätekstiin. Samoin sivulle voidaan luoda hypertekstilinkkejä sekä multimediaa, kuten kuvia, videota ja ääntä. Tämänkaltaiset erilaiset tietokiot voidaan dokumentoida, ja niiden paikat tietokannassa ovat järjestelmällisesti organisoituja.

Tietokantaan organisoituun sisältöön voidaan kohdistaa myös hakuja [Pon, 2009]. Tämä ei olisi läheskään yhtä suoraviivaista, jos sivun ulkoasun kuvaava html-koodi ja varsinainen sisältö olisivat samassa tiedostossa. Lisäksi tietokannassa olevaa sisältöä voidaan helposti käyttää uudelleen toisessa osassa sivustoa. Eri sivut voivat käyttää keskenään samoja komponentteja, kuten esimerkiksi navigointia. Samalla tiedon turha useampaan kertaan tallentaminen vähenee [Shoara *et al.*, 2011]. Myös tiedon siirtäminen kokonaan toiseen sivustoon helpottuu kopioidulla tietokannan sisältö toisen sivuston tietokantaan.

Edellä kuvattujen menetelmien tavoin web-sisällönhallintajärjestelmää käyttämällä saadaan tieto säännöllisesti organisoitua ja järjestettyä Michael H. Brackettin [2000] esittämien hyvien tiedonhallintaperusteiden mukaisesti. Näihin perusteisiin kuuluu muun muassa tietotyyppien käyttäminen, tiedon dokumentointi ja saman tiedon useaan kertaan tallentamisen välttäminen [Brackett, 2000]. Brackettin mukaan huonosti organisoitu tietosisältö maksaa sekä julkiselle sektorille että yksityisille yrityksille vuosittain miljoonia työtunteja [Brackett, 2000]. Koska

web-sisällönhallintajärjestelmän käyttäminen ohjaa tiedon hyvään organisointiin, voidaan sen käyttämistä siis perustella myös taloudellisilla syillä.

Useimmat web-sisällönhallintajärjestelmät ovat kytkettävissä suurempien tietojärjestelmien tietokantoihin. Esimerkiksi verkkokauppasovelluksessa kaupan sivuilla näkyvät tuotteet voidaan esittää siten, että ne vastaavat fyysisen varaston sisältöä sen sijaan, että kaupan sivujen tuotelistauksia pidettäisiin yllä manuaalisesti. Tämä mahdollistaa verkkokaupan tuotevalikoimien ja varastosaldojen reaaliaikaisen esittämisen. Sivustoa perustettaessa on kuitenkin syytä valita sellainen web-sisällönhallintajärjestelmä, joka mahdollistaa tulevaisuudessa mahdollisesti tarvittavat modulaariset laajennukset, vaikka niitä ei sivustoa perustettaessa vielä tarvittaisikaan [Frick, 2010].

Hackosin [2002] mukaan yrityksen siirtyminen sisällönhallintajärjestelmän käyttöön johtaa valtavaan käytännölliseen ja taloudelliseen hyötyyn. Sen seurauksena voidaan muuttaa yrityksen järjestämätön tieto hyödynnettävissä olevaksi informaatioksi ja siten yrityksen voimavaroiksi [Hackos, 2002]. Hackos [2002] pitää sisällönhallintajärjestelmään siirtymistä aikaa ja työtä vaativana toimenpiteenä, mutta hänen mukaansa se yleensä on lopulta kannattavaa. On ilmeistä, että nämä suurten yritysten sisäisiin sisällönhallintajärjestelmiin kohdistuvat väitteet pitävät paikkansa myös web-sisällönhallintajärjestelmiin.

3 YLEISIMMÄT WEB-SISÄLLÖNHALLINTAJÄRJESTELMÄT

Toukokuussa 2011 tehdyn tutkimuksen [Patel *et al.*, 2011] mukaan kolme suosituinta käytössä olevaa web-sisällönhallintajärjestelmää ovat Joomla, WordPress ja Drupal. Tutkimukseen vastasi 4000 web-sisällönhallintajärjestelmää käyttävää sivuston ylläpitäjää, joista yli 70 prosenttia oli valinnut jonkin näistä kolmesta järjestelmästä. Suosituin web-sisällönhallintajärjestelmä oli Joomla, toiseksi suosituin WordPress ja kolmanneksi suosituin Drupal.

Myös marraskuussa 2011 tehdyssä tutkimuksessa [Shoara *et al.*, 2011] Joomla ja WordPress olivat kaksi suosituinta web-sisällönhallintajärjestelmää, mutta Drupal vasta viidenneksi suosituin. Jälkimmäinen tutkimus on kuitenkin otannaltaan huomattavasti pienempi, joten ensimmäisen tutkielman tulosta voidaan pitää relevantimpana, vaikka se onkin puoli vuotta vanhempi. Jälkimmäinen tutkimus puoltaa silti Joomla ja WordPressin asemaa kahtena suosituimpana web-sisällönhallintajärjestelmänä ja Drupalin kuulumista suosituimpien joukkoon.

Joomla, WordPressin ja Drupalin suosiota selittää moni tekijä. Kaikki kolme ovat ilmaiseksi käyttöönotettavia ja vapaasti muokattavia avoimen lähdekoodin järjestelmiä. Lisäksi kaikkien ajo- ja tallennusalueena käytettävä PHP-ohjelmointikielen, MySQL-tietokannan ja Apache-palvelimen yhdistelmä on erittäin tyypillinen nykyaikaisen internet-sivuston toteutusympäristö. [Joomla!, 2012b; Drupal, 2010; WordPress, 2012c]

Matalista lähtötaaso-vaatimuksista huolimatta näitä kolmea järjestelmää voidaan käyttää erilaisien www-sivustojen perustamiseen ja ylläpitoon yksinkertaisista blogeista monimutkaisiin verkkoportaaleihin. Web-sisällönhallintajärjestelmien laajat konfigurointimahdollisuudet ja lisäosat mahdollistavat niiden räätälöinnin kuhunkin käyttötarkoitukseen sopivaksi.

3.1 Joomla

Joomla on tässä tutkimuksessa tarkasteltavista web-sisällönhallintajärjestelmistä nimellisesti nuorin. Joomla-nimi (kirjoitetaan joissain yhteyksissä Joomla!) otettiin käyttöön vanhemman Mambo-nimisen sisällönhallintajärjestelmän kehityshaaralle vuonna 2005 [Analogik Organisation, 2012]. Joomla on valinneet sivustonsa sisällönhallintajärjestelmäksi esimerkiksi maailman suurin verkkokauppa E-Bay ja huonekalujätti Ikea [Joomla!, 2012a].

Nuoresta iästään huolimatta Joomla oli vuoden 2011 toukokuussa [Patel *et al.*, 2011] ja marraskuussa [Shoara *et al.*, 2011] tehtyjen tutkimusten mukaan suosituin web-sisällönhallintajärjestelmä. Se on myös voittanut avoimen lähdekoodin web-sisällönhallintajärjestelmien palkinnon [Rahmel, 2009]. Joomlaa pidetään käyttäjäystävällisenä ja helposti asennettavana. Sen hitautta on moitittu [Frick, 2010], vaikka toistuvia sivulatauksia nopeuttamaan on kehitetty sivujen välimuistilataus (*page caching*) [Pon, 2009].

Avoimen lähdekoodin periaatteen mukaisesti Joomlaan on saatavilla tuhansia ilmaisia sivupohjamalleja sekä lisäosia, vaikka myös sivupohjamallien kaupallisia versioita on saatavilla [Patel *et al.*, 2011]. Joomlaan vakiovälineistöön kuuluvat myös muun muassa web-sisällönhallintajärjestelmille tyypilliset RSS-syötteet (*Really Simple Syndication*), mahdollisuus tehdä sivujen ulkoasusta tulostamiseen sopivat versiot, erilaiset hakutoiminnot ja mahdollisuus tallentaa sisältö eri kielillä [Pon, 2009].

3.2 Drupal

Drupalin perustivat antwerpeniläiset opiskelijat Dries Buytaert ja Hans Snijder vuonna 2000. Aluksi se oli vain muutaman opiskelijan välinen yhteydenpitoportaali, mutta lopulta sivusto kehittyi yleiseksi ja avoimeksi web-sisällönhallintajärjestelmäksi. Drupalia käyttävät nykyisin kotisivujensa ylläpitoon esimerkiksi tietoturvayritys Symantec Connect ja Yahoo!':n tutkimusorganisaatio Yahoo! Research [Drupal, 2012a].

Drupalin nimen taustalla on mielenkiintoinen tarina. Drupalin nimeksi piti nimittäin alun perin tulla *Dorp*, eli 'kylä' hollanniksi. Dries kuitenkin teki kirjoitusvirheen ja varasi osoitteen *www.drop.org*. Myöhemmin sana *drop* käännettiin hollanniksi, mistä syntyi sana *druppel*, joka englantilaisittain äännetään 'drupal'. [Drupal, 2012c]

Drupalia pidetään yhtenä monipuolisimmista web-sisällönhallintajärjestelmistä. Yhtäältä se antaa edistyneille käyttäjille paljon konfigurointimahdollisuuksia mutta toisaalta saattaa olla hankala aloitteleville käyttäjille [Patel *et al.*, 2011]. Frickin [2010] mukaan Drupalin konfigurointi vaatii melko vankkaa PHP- ja MySQL-osaamista, mikä viittaa siihen, että kaikkea konfigurointia ei voi tehdä ylläpito-käyttöliittymän kautta.

Drupalin erikoisuutena on sisäinen modulaarisuus. Drupalin toiminnot on jaettu moduleihin, joista järjestelmän ylläpitäjä voi valita tarvitsemansa moduulit järjestelmäänsä [Tomlinson *et al.*, 2010]. Drupalin moduleihin kuuluu esimerkiksi keskustelufoorumeita, kuvagallerioita ja ajanhallintajärjestelmiä [Tomlinson *et al.*, 2010]. Perusasennuksessa on mukana vain tärkeimmät moduulit, ja loput voi asentaa tarpeiden mukaan. Drupalia suositaankin, kun halutaan valita järjestelmän komponentit tarkasti ja optimoida järjestelmää jättämällä tarpeettomat osat asentamatta. Drupalin vahvuutena pidetään laajaa ja innokasta kehittäjäyhteisöä [Patel *et al.*, 2011], joka takaa myös modulien suuren määrän ja monipuolisuuden.

Drupal sai vuonna 2011 tehdyssä web-sisällönhallintajärjestelmien käytettävyydestä [López *et al.*, 2011] parhaat pisteet. Käytettävyydestä olivat mukana Drupalin lisäksi myös Joomla, OpenCMS, eZ Publish, Plone ja Typo3 [López *et al.*, 2011].

3.3 WordPress

WordPress sai alkunsa 2000-luvun alussa Cafelog- ja B2 -nimillä tunnetusta blogialustasta [Douglass *et al.*, 2006b]. Tämän alustan kehityksen loputtua yhdestä sen kehityshaaroista muodostui WordPress vuonna 2003 [Douglass *et al.*, 2006b]. Myöhemmin WordPressistä on tullut blogialustan lisäksi arvostettu ja laajalti käytetty web-sisällönhallintajärjestelmä [Leary, 2009]. WordPressin tunnetuimpia käyttäjiä ovat televisiokanava CNN ja sanomalehti New York Times [WordPress, 2012b].

Pitkän historiansa vuoksi WordPress on tullut hyvin dokumentoiduksi ja laajalti käytetyksi. Vuonna 2011 WordPressiin liittyviä kirjoja oli painossa yli 160, kun esimerkiksi Joomla ja Drupal jäivät kumpikin alle sadan kirjan rajan [Patel *et al.*, 2011]. Lisäksi Alexan miljoonan suosituimman sivuston listasta [Alexa, 2012] WordPress oli ylivoimaisesti käytetyin web-sisällönhallintajärjestelmä 12,5 prosentin osuudella listatuista sivustoista [Patel *et al.*, 2011].

WordPressin tarjontaan kuuluu omalle palvelimelle asennettavan avoimen lähdekoodin web-sisällönhallintajärjestelmän lisäksi myös ilmainen WordPressin palvelimella toimiva blogipalvelu. Blogipalvelun kotisivuksi on varattu osoite *www.wordpress.com*, kun taas web-sisällönhallintajärjestelmän kotisivut löytyvät osoitteesta *www.wordpress.org*.

3.4 Muut järjestelmät

Näiden kolmen yleisimmän web-sisällönhallintajärjestelmän lisäksi on olemassa myös lukuisia muita valmisratkaisuuina käyttöönotettavia web-sisällönhallintajärjestelmiä. Ajan ja tilan rajallisuuden vuoksi tässä tutkielmassa tarkastellaan vain kolmea yleisintä web-sisällönhallintajärjestelmää yksityiskohtaisemmin mutta kuitenkin seuraavaksi tehdään lyhyt katsaus muiden web-sisällönhallintajärjestelmien tarjontaan.

Suuri osa markkinoilla olevista web-sisällönhallintajärjestelmistä on ratkaisuiltaan hyvin samankaltaisia kuin Joomla, Drupal ja WordPress. Toisin sanoen ne ovat PHP:n, MySQL:n ja Apachen varaan rakennettuja avoimen lähdekoodin järjestelmiä. Näihin lukeutuvat esimerkiksi TYPO3 ja eZ Publish [Pon, 2009].

Plone on ilmainen ja avoimen lähdekoodin web-sisällönhallintajärjestelmä [Plone, 2012]. Se on rakennettu Python-ohjelmointikieltä käyttävän Zope-sovelluskehiksen päälle [Plone, 2012]. Zopen erikoisuutena on, ettei se käytä relaatiotietokantaa vaan omaa olio-orientoitunutta tietokantaratkaisuaan [Plone, 2012]. Marraskuussa 2011 tehdyssä tutkimuksessa [Shoara *et al.*, 2011] Plonella oli 7,1 prosentin osuus käytetyistä web-sisällönhallintajärjestelmistä.

Muista avoimen lähdekoodin web-sisällönhallintajärjestelmistä Microsoftin kehittämään ASP.NET-verkkosovelluskehikseen ja Microsoftin omien palvelinohjelmistojen tekniikkaan luottavat esimerkiksi Umbraco ja DotNetNuke [Pon, 2009]. Javaan puolestaan perustuu muun muassa OpenCms [Pon, 2009].

Kaupallisista järjestelmistä mainittakoon Javalla toteutettu Alfresco [Pon, 2009] ja Microsoftin web-sovelluskehikkoon perustuva Kentico [Kentico, 2012]. Esimerkiksi Kenticon lisenssien hinnat alkavat 1999 dollarista, mutta hinta sisältää kaikki kustannukset www-osoitteen varaamisesta palvelinylläpitoon [Kentico, 2012].

Monet organisaatiot ja yritykset ovat päätyneet lukuisista olemassaolevista ratkaisuksista huolimatta suunnittelemaan ja kehittämään oman web-sisällönhallintajärjestelmänsä [Pon, 2009]. Tämä melko kallis ja aikaavievä kehitystyö ainakin takaa sen, että järjestelmän toiminnot vastaavat käyttäjien tarpeita ja mahdollistaa saumattoman integroitumisen suurempien keskustietojärjestelmien kanssa. Toisaalta tämä lähestymistapa myös vaatii tuntuvasti enemmän ohjelmointitaitoja verrattuna valmiilla web-sisällönhallintajärjestelmällä suoritettuun sivuston perustamiseen ja konfigurointiin. Lisäksi valmiilla ratkaisuilla on takanaan joko

kaupallisen yrityksen testausprosessi tai avoimen lähdekoodin kehittäjien yhteisö, jonka vaikutusta järjestelmän toimivuuteen ei kannata aliarvioida.

3.5 Web-sisällönhallintajärjestelmiä muistuttavat ratkaisut

Monet web-sivustot on toteutettu web-sisällönhallintajärjestelmiä muistuttavilla ratkaisuilla, jotka eivät kuitenkaan ole web-sisällönhallintajärjestelmiä tässä tutkielmassa tarkoitettulla tavalla. Seuraavassa on lyhyt katsaus tyypillisimpiin esimerkkeihin web-sisällönhallintajärjestelmiä muistuttavista ratkaisuista.

3.5.1 Blogipalvelut

Blogien toimintaperiaate on sisältöä tuottavan käyttäjän näkökulmasta samantyyppinen kuin web-sisällönhallintajärjestelmissä. Toisin sanoen sisäänkirjautumisen jälkeen käyttäjän tarvitsee vain kirjoittaa sisältöä, ja lähettämisen painamisen jälkeen se on internetissä luettavissa kaikkialla. Tosin lähes kaikki muut web-sisällönhallintajärjestelmien keskeiset ominaisuudet puuttuvat blogipalveluista.

Blogit ovat eräällä tavalla web-sisällönhallintajärjestelmien edeltäjiä. Useimmat web-sisällönhallintajärjestelmät sisältävät edelleen blogin olennaisimmat toiminnot mutta lisäksi myös paljon muita ominaisuuksia. Voidaan sanoa, että blogipalvelu on tietyn tyyppinen sivusto, joka voidaan luoda web-sisällönhallintajärjestelmällä. Toisin sanoen blogipalveluja voidaan pitää web-sisällönhallintajärjestelmän tarjoamien palvelujen osajoukkona.

Noin kolmasosa maailman suosituimmista blogeista toimii maksuttomissa blogipalveluissa kuten Typepad, Blogsmith tai Blogger [Royal Pingdom, 2009]. Niissä käyttäjän ei tarvitse omistaa omaa palvelinta, vaan rekisteröityminen palveluun riittää. Palveluun saa käyttäjätunnuksen, jolla blogiin saa kirjoittaa sisältöä ja luokitella sitä asiansain. Blogikirjoituksia voi kommentoida ja etsiä hakusanojen perusteella. Blogin ulkoasua saa yleensä muokata esivalittavilla teemoilla, mutta sivun perusrakenne on tyypillisesti yhtenäinen kaikille blogipalvelussa kirjoitetuille blogeille.

3.5.2 Wikit

Wiki on toinen edustava esimerkki web-sisällönhallintasovelluksesta. Wikin periaatteena on siirtää sisällön hallinta ylläpidolta käyttäjille. Kuka tahansa saa tuottaa ja muokata tietoa, vaikkakin sivustojen ylläpitäjät valvovat usein käyttäjien wikiin syöttämää tietoa.

Epäilemättä kuuluisin esimerkki wikistä on Wikimedia Foundationin perustama avoin verkkotietosanakirja Wikipedia monine sisärsivustoineen, joihin lukeutuu esimerkiksi Wiktionary-sanakirja [Wikimedia Foundation, 2012]. Wikimedia Foundationin käyttämä ja kehittämä ilmainen avoimen lähdekoodin järjestelmä on nimeltään MediaWiki [Wikimedia Foundation, 2012]. Se on käytössä Wikimedia Foundationin omien sivustojen lisäksi monien muiden yhdistysten sivustoilla. Yhteisöllisten avoimien sivustojen lisäksi monet yritykset käyttävät wikiä sisällön hallintaan yrityksen sisäverkossa, jonka sisällä wikin sisältö on vapaasti muokattavissa.

Wikeistä puhuttaessa on myös oleellista mainita, että verkossa on olemassa sanakirjoja ja tietosanakirjoja, joiden sisältöä käyttäjät eivät kuitenkaan pääse muokkaamaan. Näiden palvelujen ylläpitoon mitä todennäköisimmin käytetään web-sisällönhallintajärjestelmää, mutta wikistä tällöin ei ole kyse.

4 WEB-SISÄLLÖNHALLINTAJÄRJESTELMIEN KESKEISET OMINAISUUDET

Tämän luvun tarkoituksena on kartoittaa web-sisällönhallintajärjestelmien keskeiset ominaisuudet ja vertailla niiden samanlaisuuksia ja eroavaisuuksia. Vertailun kohteena käytetään kolmea suosituinta web-sisällönhallintajärjestelmää eli Joomlaa, Drupalia ja WordPressia.

Frick [2010] jakaa web-sisällönhallintajärjestelmän toiminnot neljän eri pääkomponentin käsittelyyn: näkymien, tunnisteiden, ylläpitökäyttöliittymän ja tietokannan toimintoihin. Tämän luvun vertailussa käytetään samaa jaottelua täydennettynä muutamilla alaluokituksilla.

4.1 Näkymät ja niihin liittyvät toiminnot

Näkymillä tarkoitetaan sitä osaa sivustosta, jonka sivustolla vieraileva käyttäjä näkee ja jonka kautta hän sivustoa käyttää. Vaikka tavallinen käyttäjä ei välttämättä huomaa olevansa sivustolla, joka käyttää web-sisällönhallintajärjestelmää, reagoidaan palvelimella sivustolle tehtyihin sivulatauksiin eri tavalla kuin perinteisesti.

Perinteisessä web-sivustossa jokaista web-sivua vastaa oma html-tiedostonsa, jonka sisällä on määritelty sivun sisältö ja näyttötapa. Kun käyttäjä klikkaa linkkiä tai kirjoittaa osoitteen selaimen, palvelin hakee osoitetta vastaavan html-tiedoston ja palauttaa sen selaimelle, joka puolestaan näyttää sen käyttäjälle.

Web-sisällönhallintajärjestelmällä toteutetussa sivustossa ei sivulatauksen seurauksena haeta palvelimelta staattista html-tiedostoa vaan sellainen rakennetaan dynaamisesti. Jokaista yksikäsitteistä url-osoitetta vastaa tietokantaan tallennettu muotoilematon sisältö ja näyttötavan määrittelevä sivupohja, jotka yhdistämällä generoidaan näyttöruudulle piirrettävä www-sivu.

Näkymien käsittelyperiaatteissa ei esiinny eroja tässä tutkielmassa tarkasteltavien web-sisällönhallintajärjestelmien välillä.

4.1.1 Sivupohjat ja niiden käsittelyyn liittyvät toiminnot

Sivupohjat ovat eräänlaisia www-sivun sovitukseenkehikkoja, joista rakennetaan www-sivuja web-sisällönhallintajärjestelmän tietokannasta haetulla sisällöllä. Kaksi eri sivua saattavat käyttää samaa sivupohjaa mutta eri sisältöä, jolloin sivujen näyttökuvauksia ei tarvitse tehdä kahteen kertaan.

Web-sisällönhallintajärjestelmissä on tyypillisesti tarjolla valmiina useita erilaisia sivupohjia, mutta niitäkin on mahdollista muokata tai luoda jopa kokonaan uusia [Tomlinson et al., 2010; Rahmel, 2009]. Lisäksi internetissä on tarjolla tuhansia sivupohjia joko ilmaiseksi tai maksua vastaan [Frick, 2010].

Valmiiden sivupohjien käyttöönotto on nopeaa, mutta riskinä on se, että jokin toinen sivustokin käyttää samaa sivupohjamallia [Frick, 2010], minkä seurauksena molempien sivustojen sivut näyttävät samanlaisilta. Omien sivupohjien kehittäminen varmistaa ainakin sen, että tietojen näyttötavan saa tyyliä ja konfiguroida haluamallaan tavalla.

Web-sisällönhallintajärjestelmien valmiita sivupohjia on myös arvosteltu käytettävyyden puutteistaan [López et al., 2011], mutta toisaalta mitään takeita ei ole, että järjestelmän ulkopuolelta ladatuissa sivupohjissa tämä asia olisi toteutettu paremmin. Tästäkin asiasta voi varmistua vain tekemällä itse käytettävyyden normien mukaisia sivupohjia tai pyytämällä käytettävyyden arvioinnin asiantuntijalta.

Tarkasteltavissa web-sisällönhallintajärjestelmissä sivupohjien käytössä ei esiinny merkittäviä eroja.

4.1.2 Clean url -tekniikka

Kaikissa tarkastelluissa web-sisällönhallintajärjestelmissä on käytössä clean url -tekniikka. Se tarkoittaa, että osoitteiden perässä ei ole html-sivuille tyypillistä tiedostopäätettä vaan sivun osoite voi olla vapaavalintainen, esimerkiksi muotoa `http://www.sivusto.com/foo/bar` [Tomlinson et al., 2010]. Tämä osoite muutetaan palvelimella perinteisemmän näköisen osoitteen kutsuksi, esimerkiksi Drupalissa käytettäisiin osoitetta `http://www.sivusto.com/index.php?q=foo/bar`, missä `index.php` on Drupalin sisällönhallintajärjestelmän sivulatauksia käsittelevä kooditiedosto ja `q` on sivun yksilöivä `get`-parametri, jonka perusteella haetaan tietokannasta osoitteeseen liittyvä sisältö ja sivupohja [Tomlinson et al., 2010]. WordPressin tapa viitata sivuihin ilman clean url -tekniikkaa on hakea sivut nii-

den yksilöivien id-tunnusten perusteella. Tällöin osoite saattaa näyttää esimerkiksi seuraavalta: <http://www.sivusto.com/?p=10>, missä p on get-parametrina välitettävä sivuston id-tunnus.

Clean url -tekniikalla toteutetut osoitteet ovat usein miellyttävämpää luettavaa ihmissilmälle kuin tiedostopäätteet tai osoitteen perässä näkyvät get-parametrit. Se on myös hyödyllistä sivuston näkyvyydelle, koska sivut päätyvät todennäköisemmin hakukoneiden tuloslistauksiin, kun niiden osoitteissa ei näy get-parametria [Douglass *et al.*, 2006a]. Vaikka clean url -tekniikka sisältyykin lähtökohtaisesti web-sisällönhallintajärjestelmiin, se ei välttämättä ole oletusarvoisesti käytössä [Frick, 2010]. Sivuston ylläpitäjän tulisikin varmistaa clean url -tekniikan ja muun hakukoneoptimoinnin toiminta ennen sivuston liittämistä internetiin.

Clean url -tekniikan käytön mahdollisuus riippuu käytettävästä www-palvelimesta. Esimerkiksi Drupalin dokumentaatiossa [Drupal, 2012b] mainitaan, että jos käytössä on www-sivustoille tyypillinen Unixilla (tai Linuxilla) ajettava Apache-palvelin, jossa on mahdollisuus käyttää `mod_rewrite`-modulia, toimivat clean url -osoitteet moitteettomasti. Kyseinen `mod_rewrite`-moduli on ollut Apache-palvelinohjelmiston ominaisuus versiosta 1.3 lähtien [Apache, 2012]. Muiden palvelimien yhteydessä clean url -toiminnassa voi ilmetä poikkeavaa käyttäytymistä.

4.2 Tunnisteet

Tunnisteet (*tag*) ovat web-sisällönhallintajärjestelmän interaktiivinen rajapinta tietokannan ja näkymien välillä. Niiden avulla suoritetaan loogisia toimintoja näkymien esityksen yhteydessä käyttäjän syötteiden ja valintojen mukaisesti. [Frick, 2010].

Tunnisteet ovat sivupohjien ohjelmakoodin osia, jotka käsitellään tietokannasta ladatun sisällön ja käyttäjän antamien syötteiden perusteella. Tunnisteiden perusteella voidaan ilmaista yksinkertaisia loogisia operaatioita, kuten ehtolauseita tai silmukoita, ja täten luoda dynaamista html-koodia tietokannan sisällöstä.

Esimerkiksi jokaisessa blogin kommentista on samanlainen html-tunnisteilla merkitty kuvaus, mutta jokaisella kommentilla on oma yksilöllinen sisältönsä. Koodia ei kuitenkaan tarvitse kopioida tai kirjoittaa moneen kertaan. Tunniste saa aikaan silmukan toteuttamisen jokaiselle tietokannassa olevalle kommentille. Jokaisella silmukan kierroksella lisätään näkymään yksi kommentti kuvauksen mukaisella

html-koodilla.

Drupalissa on käytössä käsite blokki (*block*) [Tomlinson *et al.*, 2010]. Se on dynaamisesti tietokannan sisällöstä generoitu informaatiopalanen, joka on mahdollista näyttää tietyissä kohdissa tiettyjä sivuja [Tomlinson *et al.*, 2010]. Toisin sanoen voidaan ajatella kyseessä olevan pienikokoinen www-sivu suuremman www-sivun sisällä.

Esimerkiksi lista kuluvalle ajanhetkellä kirjautuneena olevista käyttäjistä voisi olla tällainen dynaamisesti muuttuva blokki [Tomlinson *et al.*, 2010]. Staattisella html-koodilla tällaista toimintoa ei voitaisi käytännössä mitenkään toteuttaa.

Drupalissa blokkeja voidaan asettaa tietyin kriteerein eri sivuille, esimerkiksi jollekin kategorialle tai pelkästään etusivulle. [Tomlinson *et al.*, 2010]. Blokeilla on myös oma hallintapaneeli, jossa ilmaistaan, mitä blokkeja missäkin sivuston osissa näytetään.

4.3 Ylläpitokäyttöliittymä

Web-sisällönhallintajärjestelmän käyttäjät, joilla on oikeus luoda ja/tai muokata sisältöä, pääsevät kirjautumaan ylläpitokäyttöliittymään. Se sisältää toiminnot sivuston sisällön ja ulkoasujen hallintaan. Osalle käyttäjistä voidaan antaa rajallisia oikeuksia, esimerkiksi oikeus vain sisällön muokkaamiseen [Frick, 2010].

Joomlan, Drupalin ja WordPressin ylläpitokäyttöliittymien analysointi on tehty niiden verkossa toimivien demoversioiden pohjalta [Cloud Hosting, 2012; OpenSourceCMS, 2012a; OpenSourceCMS, 2012b]. Joomlasta tutkittiin versiota 2.5, Drupalista versiota 7.12 ja WordPressistä versiota 3.3.1.

4.3.1 Päävalikko

Joomlan ylläpitokäyttöliittymän päävalikko on jaoteltu seitsemään pääosaan – sivuston yleiset asiat, käyttäjät, navigointivalikot, sisältö, komponentit, laajennukset ja ohje. Päävalikko on kuitenkin mahdollista muokata käyttäjäkohtaisesti, jotta käyttäjän näkökulmasta keskeisimmät toiminnot olisivat nopeasti saatavilla. Etusivulla on pikavalintoja yleisimpiin toimintoihin, kuten uuden sivun luominen tai tieto viimeksi sisäänkirjautuneista käyttäjistä.

Drupalin ylläpitokäyttöliittymä näyttää vaatimattomammalta kuin Joomlan. Etusi-

vu ei ole täynnä pikavalintoja, vaan käyttöliittymä on yksinkertainen ja pelkistetty. Drupalin päävalikon ylärivissä on valikot sisällön, navigoinnin, ulkoasun, käyttäjien ja modulien hallintaan sekä linkit asetuksiin, raportointitoimintoihin ja ohjeisiin.

WordPressin ylläpitokäyttöliittymän etusivu muistuttaa Joomlaa siinä, että pääsivun kojelaudassa (*dashboard*) on paljon toimintoja. Tärkeimmät toiminnot, kuten tilastot sisällön määrästä ja uusimmista kommentteista, täyttävät etusivun lähes kokonaan. Valikon päätasot ovat viestit (*posts*), media, linkit, sivut (*pages*), kommentit, ulkoasu, lisäosat (*plugins*), käyttäjät, työkalut ja asetukset. Lähes kaikkien valikkojen alta paljastuu lisää valikkotasoja.

4.3.2 Sisällön luominen

Sisällön luominen nopeasti ja vaivattomasti on eräs oleellisimmista ominaisuuksista web-sisällönhallintajärjestelmän käyttämisessä. Tämän vuoksi seuraavassa kuvataan, miten kyseinen piirre on toteutettu tarkasteltavissa web-sisällönhallintajärjestelmissä.

Joomlassa uusi sivu saadaan tuotettua yksinkertaisimmin valitsemalla sisältövalikon artikkelien hallinnasta uuden artikkelin luomiskomento. Drupalissa puolestaan sisällön nopeaa luomista varten on etusivulla oma linkkinsä, jonka klikkaamisen jälkeen käyttäjältä kysytään, haluaako hän luoda vapaamuotoisen sivun (*page*) vai tekstipohjaisen artikkelin, kuten blogitekstin (*post*). Wordpressissa on päävalikon etusivulla *Quickpress*-lomake, jonka avulla pystytään luomaan nopeasti uusi tekstisivu. Varsinaiset sisällönluontilomakkeet löytyvät kuitenkin tekstiartikkeleihin ja vapaamuotoisiin sivuihin viittaavien Posts- ja Pages-linkkien alta.

WordPressin ja Joomlaan sisällönluontilomakkeet ovat hyvin samankaltaisia. Molemmissa on *rich text* -editori, jossa on tekstinkäsittelyohjelmista tuttuja toimintoja, kuten kirjasimen valinta tai tekstin lihavointi. Molempien editorien käyttö tapahtuu *WYSIWYG*-periaatteella, eli tekstin muotoilut näkyvät tekstikentässä välittömästi. Kuten tekstinkäsittelyohjelmissä, voi myös Joomlaan *rich text* -editorissa tekstin tyylejä muuttaa kirjoittamisen jälkeenkin valitsemalla tekstiä ja asettamalla sitten tyylimäärittelyt valittuun tekstialueeseen.

WordPress ja Joomla mahdollistavat sisällön luomisen samanaikaisesti sekä *rich text* -editorilla että puhtaalla html-koodilla. Käyttäjän syöttämä sisältö päivittyy jatkuvasti molempiin lomakkeisiin, joista vain toinen on kerrallaan näkyvässä.

Täten käyttäjä voi hyödyntää html-koodin monipuolisuutta, mutta myös rich text -editorin nopeutta ja vaivattomuutta.

Sivun julkaisuvaihtoehdot ovat koko ajan näkyvissä riippumatta editointitavasta. Sivun on mahdollista tallentaa luonnoksena, asettaa sen näytölle tuleminen myöhempään ajankohtaan, tai asettaa se toisen sivun alasivuksi. Sivulla näytetään oletusarvoisesti sivun perustiedot, kuten kirjoittajan nimi ja luontiajankoh- ta, mutta Joomlaassa on mahdollista piilottaa niitä yksitellen.

Joomla tarjoaa sivun luomisen yhteydessä lomakkeen metadatalle, joka tarkoittaa lyhyttä asiasanalistaa sivun sisällöstä. Sen avulla hakukoneet lajittelevat sisältöä mahdollistamaan tehokkaampia hakuja. Jos sivulle on annettu metadata, löytyy sivu myös todennäköisemmin hakukoneella.

Joomlaassa on myös ainoana näistä kolmesta web-sisällönhallintajärjestelmästä mahdollista arkistoida sivuja. Arkistointia tarvitaan sivuille, joita ei haluta pois- ta kokonaan järjestelmästä mutta jotka halutaan poistaa sivuston näyttämien sivujen joukosta.

Drupalissa puolestaan ei ole käytössä rich text -editoria, vaan kirjoitusasun mää- rittelyt tehdään joidenkin käytettävissä olevien standardi-html-tunnisteiden avul- la. Tekstikentän alapuolelta löytyy lisävalintoja, kuten linkitys sivuston navigaa- tiovalikkoon, sivun osoitteen valinta ja julkaisuasetukset, jotka kuitenkin ovat paljon pelkistetyimmät kuin WordPressissa ja Joomlaassa. Esimerkiksi julkaisun ajastamista ei ole mahdollista tehdä.

Drupalin sisällönluontilomakkeen yksinkertaisuus johtuu siitä, että Drupalin mo- dulaarisuus antaa ylläpitäjän valita käyttämänsä tekstieditorimodulin. Eräs mah- dollinen ja hyvin tyyppillinen tekstieditorimoduli on TinyMCE, jolla saavutetaan WordPressin ja Joomlaan kaltainen rich text -editointikyky [Myers & French, 2008].

Kun verrataan näiden kolmen web-sisällönhallintajärjestelmän sisällönluontilo- makkeiden piirteitä, on ilmeistä, että Drupal tarjoaa käyttäjälle oletusarvoisesti vähiten vaihtoehtoja. Joomla ja WordPress antavat monipuoliset mahdollisuudet tekstin näyttöasun määrittelyyn ja julkaisemiseen, kun taas Drupal jättää paljon käyttäjän omien konfigurointien ja osaamisen varaan. Tämä tukee käsitystä Dru- palin profiloitumisesta monimutkaisempia toimintoja vaativalle käyttäjärühmälle, jolle tekstisivujen käsittely ei ole olennaisinta web-sisällönhallinnassa.

4.3.3 Raportointiominaisuudet

Drupalissa käyttäjän on mahdollista saada erilaisia raportteja sivustoon liittyvistä tilastoista. Esimerkiksi yleisimmät sivustolla käytetyt hakusanat tai sivustolla toistaiseksi tapahtuneet virhetilanteet voidaan tulostaa listauksina. Vastaavaa toiminnallisuutta ei löydy Joomlaista ja WordPressistä.

4.3.4 Sisällön kopiointi toiseen järjestelmään

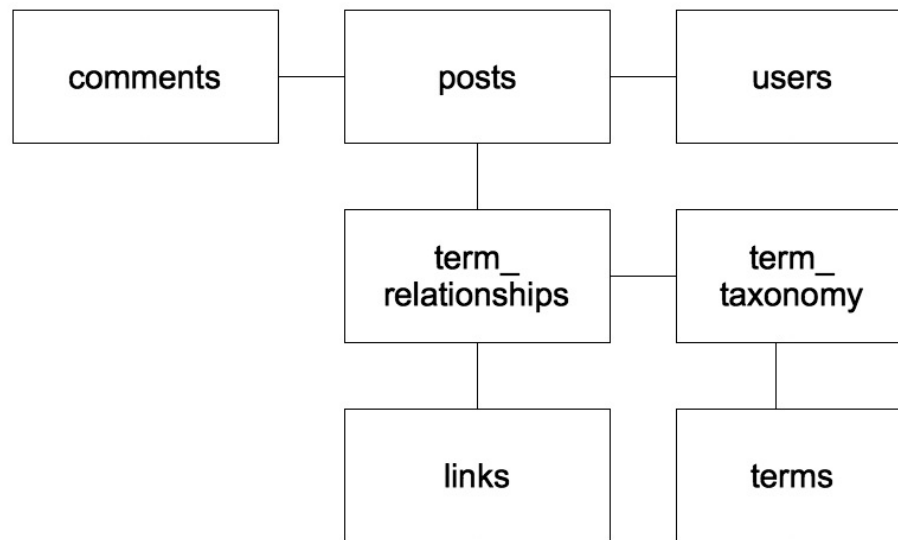
WordPressissä on mahdollista tuottaa XML-tiedosto kaikesta sivuston sisältämästä informaatiosta, kuten sivuista, kategorioista ja kommentteista. Tämä XML-tiedosto voidaan syöttää toiseen web-sisällönhallintajärjestelmään. Se luonnollisesti onnistuu helpoiten toiseen WordPress-sivustoon, mutta tietojen esitystapoja muuttamalla on mahdollista syöttää tiedot myös toisella alustalla toimivaan sivustoon. Vastaavaa toimintoa ei löydy Joomlaan ja Drupalin toiminnallisuuksista.

4.4 Tietokantaratkaisut

Koska Joomla, Drupal ja WordPress käyttävät avointa lähdekoodia, myös niiden tietokantakaaviot ovat saatavilla verkosta. WordPress tarjoaa tietokantakaavionsa nähtävillä sivustollaan [WordPress, 2012a], kun taas Joomlaan ja Drupalin tietokantakaaviot ovat aiheeseen perehtyneiden käyttäjien [Johnsen, 2010; Upsites Web Studio, 2011] *reverse engineering* -tekniikalla tuottamia ja käyttäjäyhteisön oikeiksi validoimia. Koska kaaviot ovat liian laajoja ja yksityiskohtaisia liitettäväksi osaksi tätä tutkielmaa, esitetään kaavioista alkuperäisten sijaan yksinkertaistetut ja rajatut versiot (Kuvat 4.1, 4.2 ja 4.3), jotka kattavat olennaisimpien toimintojen tietokantarakenteen.

4.4.1 Käyttäjän syöttämän sisällön rakenne

Web-sisällönhallintajärjestelmän olennaisin toiminto on tunnistaa käyttäjän antama url-osoite ja hakea siihen liittyvä sisältö tietokannasta. Tätä varten sivu on mallinnettava. Eri web-sisällönhallintajärjestelmissä sivut mallinnetaan eri tavoilla.

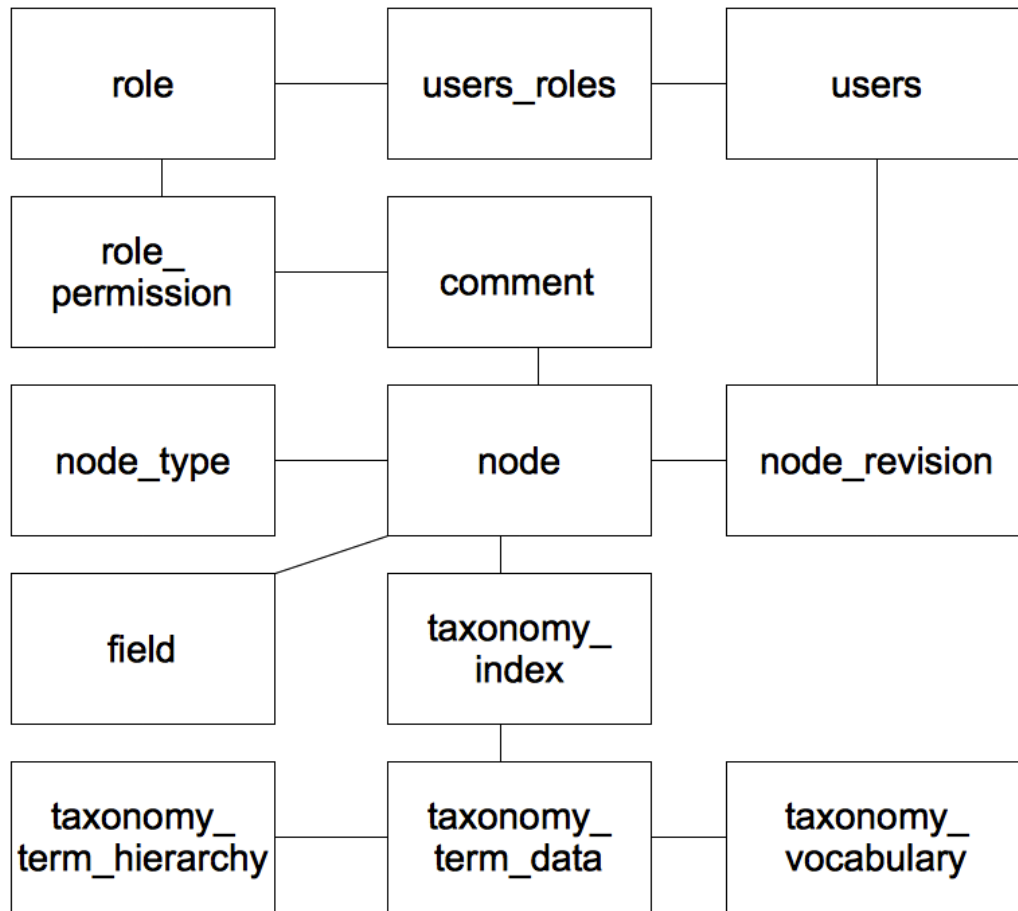


Kuva 4.1 Osa WordPressin tietokantakaaviosta. Yksi suorakulmio vastaa yhtä relaatioesitystä.

WordPressin tietokannassaan (Kuva 4.1) käyttämä käsitteistö liittyy ensisijaisesti blogeihin. Perussisältökäsite nimetään tietokannassa [WordPress, 2012a] englanninkielisellä termillä *post*, joka tarkoittaa yhtä blogikirjoitusta. Nämä post-tyyppiset sivut näytetään tietyssä www-osoitteessa uutisvirran tyyliä peräkkäin, kuten blogeissa. Lisäksi on mahdollista määritellä staattisia sivuja (*page*), joiden tapauksessa yhtä www-osoitetta vastaa yksi sivu. Molemmille sisältötyypeille on ominaista, että ne ovat monikäyttöisiä [Leary, 2009].

Drupalin tietokantarakenteessa (Kuva 4.2) [Upsites Web Studio, 2011] sivun tietokantataulun nimenä on *node* heijastaen sitä, että se on eräs solmu sivuston sivujen muodostamaa verkkoa vastaavassa graafesityksessä. Drupalin avulla luodut sivut organisoidaan sen tietokannassa aina samalla tavalla riippumatta siitä, onko kyseessä blogimerkintä vai staattinen tekstisivu [Frick, 2010].

Drupalissa on käytössä sivujen sisäinen versiohallinta. Siihen liittyvästä tietokantataulusta käytetään nimeä *node_revision*. Sen avulla sivua muokatessa edelliset versiot jäävät tietokantaan ja ovat uudelleen tarkasteltavissa.



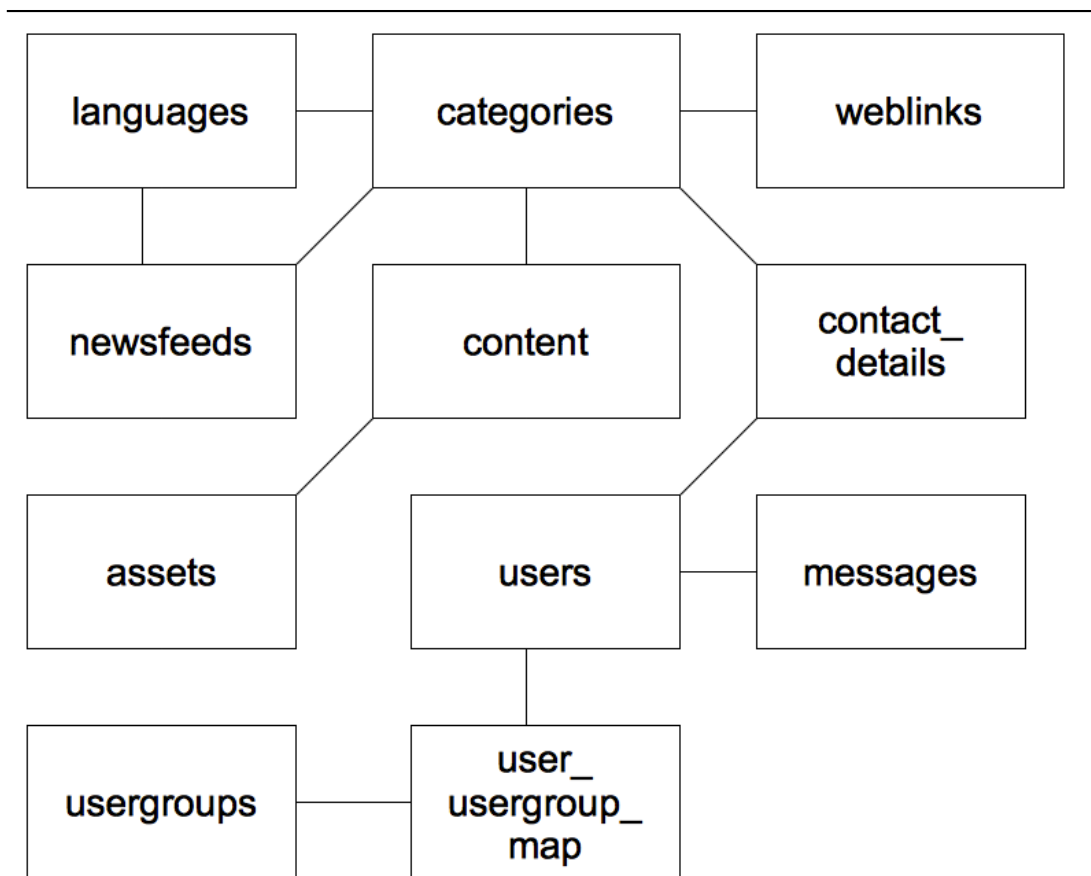
Kuva 4.2 Osa Drupalin tietokantakaaviosta. Yksi suorakulmio vastaa yhtä relaatioesitystä.

Joomla puolestaan on nimennyt sivun käyttämässään tietokantaratkaisussa (Kuva 4.3) [Johnsen, 2010] termillä *content* eli sisältö. Toisin sanoen kyseinen nimi viittaa parhaiten sen semanttiseen tulkintaan web-sisällönhallintajärjestelmässä.

Drupalissa sivun rakenne esitetään eri kenttien (*field*) avulla. Esimerkiksi sivun otsikolla, kuvilla ja leipätekstillä on omat kenttensä. Järjestelmässä on edellä mainittujen lisäksi lukuisia muita valmiita sisältötyyppejä. Samaten on helppo luoda omia sisältötyyppejä ilman ohjelmointiasiantuntemusta. [Tomlinson *et al.*, 2010]

Sekä Joomlaissa [Johnsen, 2010], Drupalissa [Upsites Web Studio, 2011], että WordPressissä [WordPress, 2012a] ilmaistaan sivun julkaisuajankohta, vaikka esi-

merkiksi Drupalissa ei ole mahdollista ajastaa sivun julkaisua oletusasetuksin.



Kuva 4.3 Osa Joomla tietokantakaaviosta. Yksi suorakulmio vastaa yhtä relaatioesitystä.

4.4.2 Url-osoitteiden käsittely

Drupalissa on oma relaatiotaulunsa clean url -tekniikkaa tukeville url-osoitteille, nimeltään *url_alias*. Se sisältää tiedon sivun id-numerosta ja url-osoitteen *alias*-osan. Lisäksi mukana on kieliparametri ilmaisemaan kieltä, jolla sisältö on kirjoitettu. [Upsites Web Studio, 2011]

Joomlaassa puolestaan on *weblinks*-niminen taulu, jossa on vastaavat perustiedot kuin Drupalissa mutta paljon enemmän tietoa sivusta, kuten sen luonti-, muokkaus- ja julkaisupäivät sekä tieto siitä, onko sivu julkinen. [Upsites Web Studio, 2011]

WordPressin tietokantakaaviossa [WordPress, 2012a] ei ole erillistä tietokanta-
taulukkoa clean url -tekniikan mukaiselle url-osoitteelle. Kyseessä voi tosin olla
puute tietokantakaavion mallinnuksessa.

4.4.3 Sisällön luokittelu

Yksi tärkeimmistä web-sisällönhallintajärjestelmien ominaisuuksista on kyky luok-
kitella sisältöä annettujen avainsanojen perusteella. Joomlaan [Johnsen, 2010] ja
WordPressin [WordPress, 2012a] tietokantakaavioissa puhutaan kategorioista (*ca-
tegory*), kun taas Drupalin [Upsites Web Studio, 2011] tietokannassa vastaava kä-
site on englanniksi *taxonomy* eli luokittelu. Vaikka käsitteiden englanninkieliset
nimet ovat erilaiset, niihin liittyvä toiminnallisuus on sama. Useimmiten sivus-
ton runko luodaan pääkategorioiden pohjalta, jotka sitten jaetaan alakategorioi-
hin tarpeen mukaan [Frick, 2010]. Myös sivuston navigaatio luodaan tyypillisesti
dynaamisesti kategorioiden perusteella.

Kaikissa kolmessa järjestelmässä sivuille voi merkitä tunnisteita (huom. tunnis-
teet tarkoittavat tässä eri asiaa kuin kappaleen 4.2 dynaamiset tunnisteet). Tun-
nisteet ovat asiasanoja, jotka kuvaavat sivua, kun taas kategoria on korkeammalla
abstraktiotasolla oleva luokitus. Tiettyjen tunnisteiden vaikutusalue ylittää kate-
goriat.

WordPressissa tunnisteita ylläpidetään useassa tietokantataulussa. Tunnisteet ja
kategoriat ilmaistaan *terms*-taulun kenttinä, jotka puolestaan yhdistetään sivui-
hin *term_relationships*-taululla. Taulu *term_taxonomy* puolestaan luokittelee
edellämainitut terms-taulun rivit. [WordPress, 2012a]

4.4.4 Käyttäjien oikeudet

Kaikissa vertailtavissa web-sisällönhallintajärjestelmissä on käytössä käyttäjätun-
nutukset [Upsites Web Studio, 2011; Johnsen, 2010; WordPress, 2012a]. Esimerkiksi
vain joillekin käyttäjille voidaan antaa oikeudet kirjautua sisään järjestelmään ja
hallita sivuston sisältöä [Frick, 2010]. Eri toimintoihin kohdistuvia käyttöoikeuk-
sia voidaan myöntää eri käyttäjille.

Kaikissa järjestelmissä käyttäjistä tallennetaan käyttäjätunnus ja salasana, joil-
la järjestelmään pääseminen tapahtuu. WordPress [WordPress, 2012a] tarjoaa li-
säksi mahdollisuuden tallentaa julkisesti esitettävä käyttäjätunnus (tallennettuna

tietokantaan nimellä *nickname*), joka näytetään sivustolla käyttäjän kirjoittamien tekstien ja kommenttien yhteydessä. Drupalissa puolestaan ei oletusarvoisesti tallenneta käyttäjän etunimeä ja sukunimeä, kuten Joomlaassa ja WordPressissä. Joomlaan erikoisuus on käyttäjien välinen yksityisviestijärjestelmä.

Itsestäänselvyytensä pidetään myös käyttäjän sähköpostiosoitteen tallentamista. Sähköpostiosoitetta tarvitaan esimerkiksi unohtuneen salasanan tilaamiseen tai joihinkin automaattisiin ilmoituksiin.

Käyttöoikeuksia ei tyypillisesti liitetä suoraan käyttäjille, vaan siihen käytetään esimääriteltäviä käyttöoikeustasoja, joissa on sallittu tai kielletty kukin järjestelmän ylläpitotoiminnoista. Lisäksi useimmiten käyttöoikeustasoja asetetaan käyttäjärühmäpohjaisesti yksittäisten käyttäjien sijasta.

Joomlaassa käyttäjiä voi ryhmitellä vapaasti ja uusia ryhmiä on mahdollista perustaa vapaasti. Ryhmät muodostavat myös hierarkkisen rakenteen. Kullakin ryhmällä on aina oma käyttöoikeustasonsa, mutta sama käyttöoikeustaso voi olla useilla ryhmillä. Myös uusia käyttöoikeustasoja voidaan perustaa. Käyttäjärühmät tallennetaan *usergroups*- ja *user_usergroup_maps*-tauluihin. Sisältökohtaiset hallintaoikeudet puolestaan tallennetaan *assets*-tauluun.

Drupalissa käyttäjällä on käyttöoikeustasoa vastaava rooli, johon liitetään käyttäjälle sallitut toiminnot. Käyttöoikeuksien liittäminen ja poistaminen on melko vapaata.

WordPressissä asia on hoidettu suoraviivaisemmin. Käyttäjälle voidaan valita rooli saatavilla olevasta listasta. Saatavilla olevia rooleja ovat ylläpitäjä (administrator), toimittaja (*editor*), kirjoittaja (*author*), osallistuja (*contributor*) ja tilaaja (*subscriber*). Ylläpitäjällä on oikeudet kaikkiin järjestelmän ylläpitotoimiin. Toimittajalla on oikeudet muokata kaikkea sivuston sisältöä, kun taas kirjoittajalla on muokkaus-oikeus vain omiin teksteihinsä. Osallistujalla on mahdollisuus lisätä teksteihin kommentteja, kun taas tilaaja voi vain seurata sivustoa. Roolien käyttöoikeuksia ei voi kuitenkaan muokata erikseen.

4.4.5 Kommentit

Kaikissa tarkasteltavissa web-sisällönhallintajärjestelmissä on mahdollista lisätä kommentteja sivuihin [Upsites Web Studio, 2011; WordPress, 2012a; Johnsen, 2010]. Tämä toiminto on käyttökelpoinen esimerkiksi blogeissa tai uutissivustois-

sa. Kuitenkin useimmilla sisältösiivuilla kommentointi yleensä piilotetaan näkyvistä.

Drupalissa sivujen sisältöä hallitaan versioittain, mutta kommentit on kuitenkin liitetty koko versiohallinnan ohi itse sivuun [Upsites Web Studio, 2011]. Tämä voi aiheuttaa sekä positiivisia että negatiivisia seuraamuksia. Itse sivun sisältö saattaa muuttua huomattavasti, jolloin vanhat kommentit eivät enää välttämättä liity muuttuneeseen sisältöön. Sen sijaan kommenttien liittyessä aina sivun tiettyyn versioon jouduttaisiin vanhat kommentit poistamaan pientenkin muutosten yhteydessä.

4.5 Yhteenveto vertailtavista järjestelmistä

Ominaisuuksien, käyttöliittymien ja tietokantakaavioiden analyysit antavat melko selkeät kokonaiskuvat kustakin web-sisällönhallintajärjestelmästä. Ne vastaavat ennalta esitettyjä käsityksiä [Patel *et al.*, 2011] eri web-sisällönhallintajärjestelmistä.

Tietokantaratkaisultaan monimutkaisin mutta perustoiminnoiltaan vähäisin Drupal vaatii käyttäjältään paljon konfigurointia mutta toisaalta tarjoaa käyttäjälle paljon vapauksia määrittellä toiminnallisuuksia. Siksi sitä suositellaankin harjaantuneemmille ja asiantunteville käyttäjille.

WordPress on ominaisuuksiltaan suppein mutta toisaalta kokonaisuudeltaan myös yksinkertaisin ja siten helpoin käyttää. Useimmat WordPressin vähäisistä ominaisuuksista tekevät siitä ennemmin blogialustan kuin varsinaisen web-sisällönhallintajärjestelmän.

Joomla asettuu ominaisuuksiltaan näiden kahden väliin. Se on nimittäin tarvittaessa monipuolinen ominaisuuksiltaan, mutta lähtökohtaisesti helppo ja yksinkertainen käyttää.

Vertailu ei ole tasapuolinen Drupalin osalta, koska siinä ei ole peruskokoonpanossa juurikaan niitä moduleita, jotka sisältävät WordPressin ja Joomlaan vakioominaisuuksia. Toisaalta jos Drupalin lisäosat olisi otettu mukaan tutkimukseen, olisi niitä pitänyt tutkia myös Joomlaan ja WordPressin osalta. Tällaisen selvityksen tekeminen kasvattaisi tutkimuksen määrää huomattavasti.

4.6 Keskeisten ominaisuuksien ilmeneminen vertailtavissa järjestelmissä

Kaikissa vertailtavissa järjestelmissä olennaisina toiminnallisuuksina esiintyneitä ominaisuuksia voidaan pitää keskeisinä web-sisällönhallintajärjestelmän yleiselle määrittelylle. Kuitenkin joitain ominaisuuksia voidaan pitää niin hyödyllisinä, että ne määritellään keskeisiksi ominaisuuksiksi, vaikka ne eivät esiintyisi kaikissa tarkastelluissa web-sisällönhallintajärjestelmissä.

Taulukkoon (Kuva 4.4) on koottu Joomlaan, Drupalin ja WordPressin keskeisimmät ominaisuudet, sekä ilmaistu, mitkä ominaisuudet kukin järjestelmä sisältää. Siitä nähdään, että ominaisuudet, jotka kaikki kolme tarkasteltua web-sisällönhallintajärjestelmää toteuttavat, ovat seuraavat: sisällön luominen sivulle, sisällön muokkaaminen sivulla, sivun poistaminen, sivun julkaiseminen, sivun luokittelu tunnisteilla ja kategorioilla, sivupohjien käyttäminen, sisällön kommentointi, käyttäjän luominen, käyttäjän muokkaaminen, käyttäjän poistaminen, käyttöoikeuksien käyttäminen ja sivun url-osoitteen nimeäminen.

Muut ominaisuudet ovat joko rakenteellisesti eri tavoin toteutettuja eri järjestelmissä tai puuttuvat kokonaan joistakin järjestelmistä. Kaikkia listattuja ominaisuuksia voidaan silti pitää tietyissä tilanteissa hyödyllisinä web-sisällönhallintajärjestelmien käytössä. Mikään tunnistettu ominaisuus ei vaikuta suoranaisesti haitalliselta tai käyttökelvottomalta toiminnolta. Yksityisviestien lähettäminen, sisällön julkaisun ajastaminen, hakukonemetadatan ja sisällön versiohistoria vaikuttavat monissa tilanteissa erityisen hyödyllisiltä, joten ne otetaan huomioon seuraavassa luvussa yleisen web-sisällönhallintajärjestelmän määrittelyssä.

<i>Ominaisuus</i>	<i>Joomla</i>	<i>Drupal</i>	<i>WordPress</i>
Sisällön luominen sivulle	x	x	x
Sisällön muokkaaminen sivulla	x	x	x
Sivun arkistointi	x		
Sivun poistaminen	x	x	x
Sivun julkaiseminen	x	x	x
Sivun julkaisun ajastaminen	x		x
Sivun julkaisun päättymisen ajastaminen	x		
Sivun hakukonemetadatat	x		
Sivun luokittelu tunnisteilla	x	x	x
Sivun luokittelu kategorioilla	x	x	x
Dynaaminen navigoinnin generointi	x	x	x
Sisällön muokkaaminen WYSIWYG-editorilla	x		x
Sisällön HTML-koodin muokkaaminen	x		x
HTML-tunnisteiden käyttäminen sisällössä		x	x
Sivupohjien käyttäminen	x	x	x
Sivun url-osoitteen nimeäminen (clean url)	x	x	x
Sisällön kommentointi	x	x	x
Sisällön versiohistoria		x	
Käyttäjän luominen	x	x	x
Käyttäjän muokkaaminen	x	x	x
Käyttäjän poistaminen	x	x	x
Erillinen julkinen käyttäjätunnus			x
Yksityisviestit käyttäjien välillä	x		
Käyttäjäryhmät	x		
Käyttäjäroolit		x	x
Käyttöoikeudet	x	x	x
Käyttöoikeuksien muokkaaminen	x	x	
Sivuston tilastojen raportointi		x	
Sivuston sisällön kopiointi XML-tiedostoon			x

Kuva 4.4 Taulukoesitys vertailtavien järjestelmien ominaisuuksista

5 WEB-SISÄLLÖNHALLINTAJÄRJESTELMÄN YLEINEN MÄÄRITTELY

Edellä on tarkasteltu ja analysoitu yleisimpiä käytössä olevia web-sisällönhallintajärjestelmiä, niiden ominaisuuksia ja niissä käytettyjä tietokantaratkaisuja. Seuraavassa esitetään ehdotus yleiseksi määrittelyksi web-sisällönhallintajärjestelmän toiminnallisuudelle perustuen edellä keskeisiksi todettuihin ominaisuuksiin.

Määrittely tehdään liiketoimintasääntöjä (*business rules*) [Business Rules Group, 2000] käyttäen. Jokaiselle web-sisällönhallintajärjestelmää määrittelevälle ominaisuudelle muodostetaan liiketoimintasääntö. Joukosta tällaisia yksikäsitteisiä ja atomisia liiketoimintasääntöjä voidaan muodostaa graafisesti koko järjestelmän kattava yhtenäinen käsitekaavio.

Painoteknisistä syistä käsitekaavio esitetään osittain käsitekohtaisesti ja lopuksi yksinkertaistettuna koko järjestelmän kattavana käsitekaaviona, joka ilmaisee, miten käsitekohtaiset kaaviot liittyvät toisiinsa. Käsitekaavioiden piirtämisessä käytetään IDEF1X-visualisointitapaa, joka on Business Rules Groupin valitsema visualisointitapa liiketoimintasäännöille [Business Rules Group, 2000]. IDEF1X-visualisointitapa on kuvattu liitteessä A.

Määrittely perustuu web-sisällönhallintajärjestelmien vertailussa havaittuihin näkökohtiin, mutta osa säännöistä esittää parannuksia olemassa oleviin järjestelmiin. Määrittely ei ilmaise kaikkea mahdollista, mitä web-sisällönhallintajärjestelmiin voidaan liittää tai on liitetty, vaan kyseessä on keskeisimpien ominaisuuksien esittäminen selkeästi jäsennellyllä tavalla.

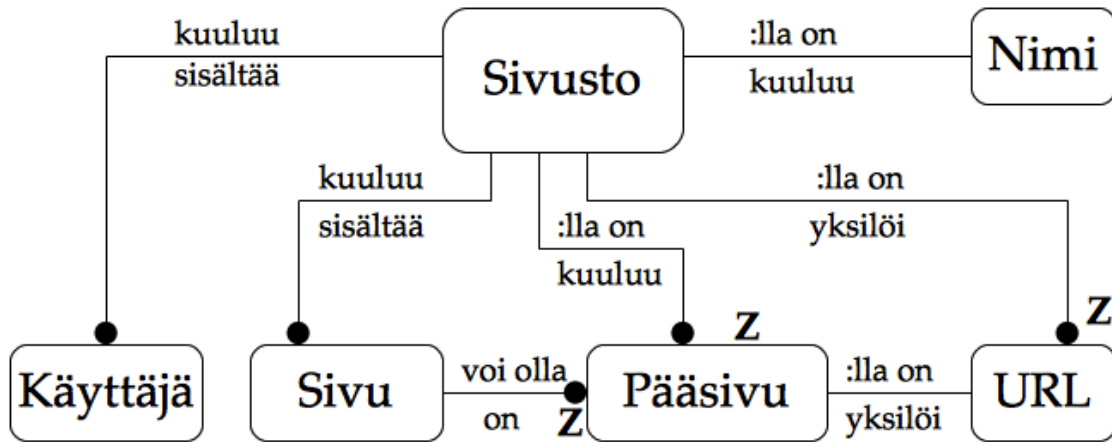
5.1 Sivusto

Sivustolla tarkoitetaan tässä määrittelyssä web-sisällönhallintajärjestelmällä tuotettua sivustoa ja sen hallitsemiseen tarvittavia toimintoja.

Sivusto määritellään seuraavien periaatteiden mukaisesti:

- Sivustolla voi olla useita käyttäjiä.
- Sivustolla voi olla useita sivuja.
- Yksi sivuston sivuista määritellään pääsivuksi.

- Sivustolla on oltava nimi.
- Sivuston pääsivun osoite on samalla koko sivuston url-osoite.
- Jos sivustolla ei ole yhtään sivua, sillä ei voi olla pääsivua eikä siten myöskään url-osoitetta, eikä sivusto täten näy internetissä.



Kuva 5.1 Sivuston käsittekaavio

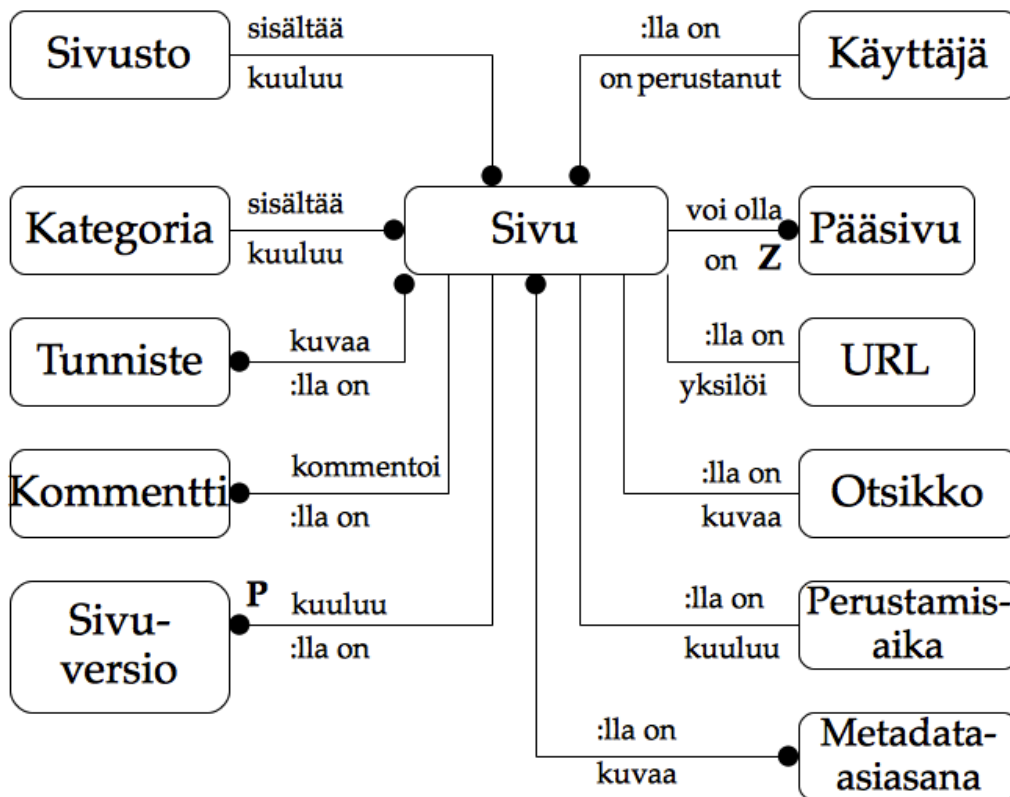
5.2 Sivu

Sivu on web-sisällönhallintajärjestelmän peruskäsite. Sivuuun liittyy aina yksikäsitteinen url-osoite. Itse sisältö kuitenkin tallennetaan sivuversioihin, joita yhdellä sivulla voi olla useita. Täten voidaan esimerkiksi luoda sivuversio, joka ajastetaan näkymään sivustolla tulevaisuudessa, tai asettaa sivulle uudestaan aiemmin käytöstä poistettu sivuversio. Yhtenä ajanhetkenä sivulla näytetään yksi sivuversio.

Jokaisen sivun tulee täyttää seuraavat ominaisuudet:

- Sivun pitää kuulua johonkin sivustoon.
- Sivulla on oltava otsikko.
- Sivulla on oltava yksikäsitteinen url-osoite.

- Sivulla on perustaja, joka on sivujen perustamiseen oikeutettu käyttäjä.
- Sivulla on perustamisaika.
- Sivulla on yksi tai useampia sivuversioita.
- Sivu kuuluu yhteen kategoriaan.
- Sivulla voi olla tunnisteita.
- Sivulla voi olla kommentteja.
- Sivulla voi olla useita metadata-asiasanoja hakukoneoptimointia varten.
- Sivu voi olla sivuston pääsivu.



Kuva 5.2 Sivun käsittekaavio

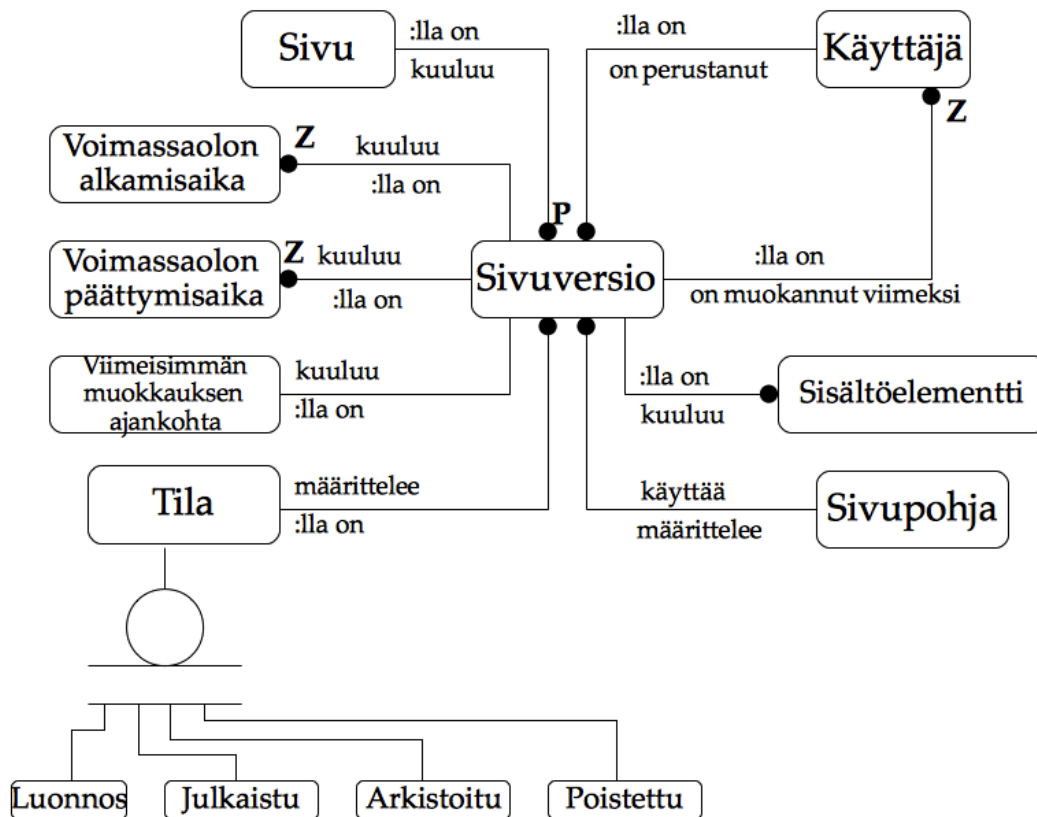
5.3 Sivoversio

Sivuersiossa määritellään edellä kuvatun sivun varsinainen sisältö. Sivuersioiden ajastuksen avulla määritellään, mikä niistä näytetään sivulla. Jos yksikään sivuersioista ei täytä julkaisun vaatimia kriteereitä, ei sivua näytetä sivustolla lainkaan.

Jokaisen sivuersioon liittyvät seuraavat säännöt:

- Sivuersion pitää liittyä yhteen ja vain yhteen sivuun.
- Sivuersio on tuotettu jonkin sivupohjan perusteella.
- Sivuersiolla on oltava sivupohjassa pakollisiksi määriteltyjen sisältöelementtityyppien mukaiset sisältöelementit.
- Sivuersiolla voi olla sivupohjassa vapaaehtoisiksi määriteltyjen sisältöelementtityyppien mukaisia sisältöelementtejä.
- Sivuersiolla voi olla voimassaolon alkamisaika.
- Jos sivuersion voimassaolon alkamisaikaa ei ole määritelty, sivuersio näytetään sivustolla välittömästi.
- Sivuersiolla voi olla voimassaolon päättymisaika.
- Jos sivuersion voimassaolon päättymisaikaa ei ole määritelty, sivuersio ei poistu koskaan automaattisesti näkyvistä sivustolta.
- Sivuersio on jossakin seuraavista tiloista: luonnos, julkaistu, arkistoitu tai poistettu.
- Jos sivuersion julkaisuaika on määrittelemätön tai aiempi kuin nykyhetki, julkaisun päättymispäivä on määrittelemätön tai myöhäisempi kuin nykyhetki ja sivuersio on tilassa julkaistu, näytetään tämä sivuersio sivustolla sivuun liittyvässä url-osoitteessa.
- Jos useat sivuersiot täyttävät edellä mainitut julkaisukriteerit, näytetään sivulla näistä se sivuersio, joka on viimeksi luotu tai jota on viimeksi muokattu.

- Sivuversion muokkaaminen ei merkitse automaattisesti uuden sivuversion luomista.
- Jos sivuversiota on muokattu, sivuversiolla on tieto viimeisimmästä muokkajasta, joka on tähän oikeutettu käyttäjä.
- Jos sivuversiota on muokattu, sivuversiolla on tieto viimeisimmän muokkauksen ajankohdasta.



Kuva 5.3 Sivuversion käsittekaavio

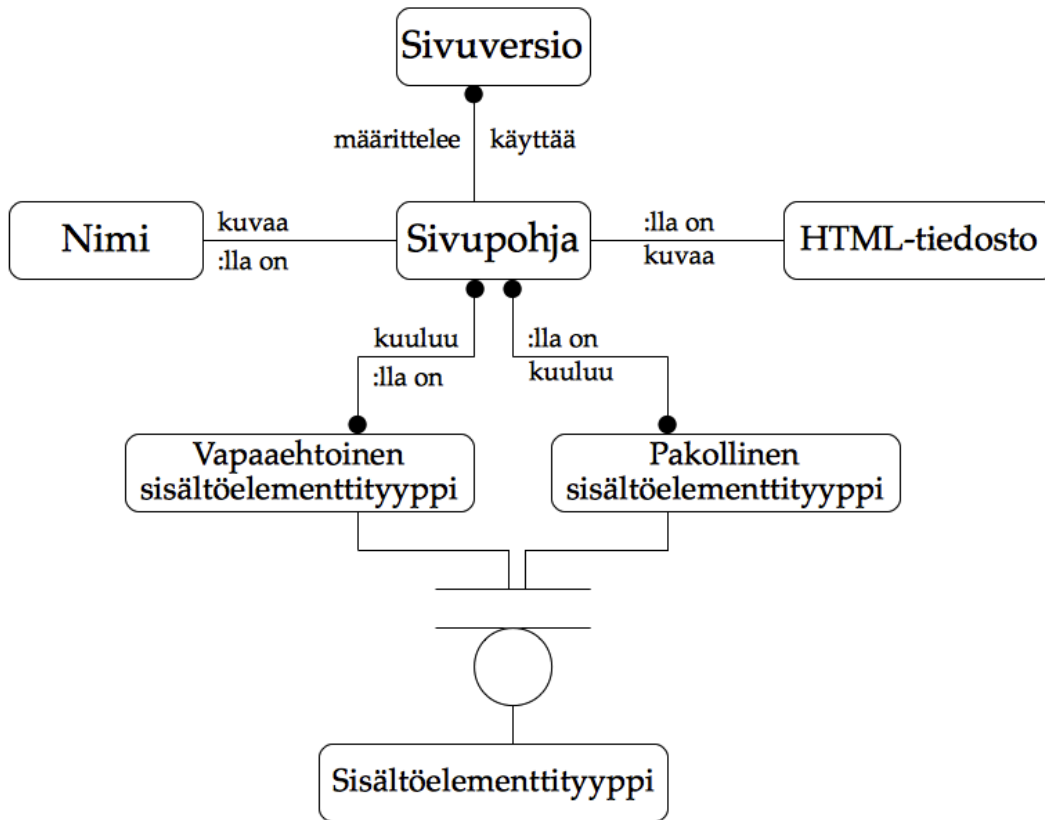
5.4 Sivupohja

Sivupohja määrittelee sivun näyttöasun organisoinnin näyttöruudulla. Sivupohja ilmaisee näytettävät sisältöelementit mutta ei sisältöelementteihin liittyvää teksti- tai multimediasisältöä.

Sivun ja sivupohjan välistä suhdetta voi verrata XML-dokumentin ja DTD-tiedoston väliseen suhteeseen. DTD-tiedosto nimittäin määrittelee sen toteuttavien XML-dokumenttien tietoalkiot (elementit ja attribuutit) ja niiden välillä vallitsevat rakenteet mutta ei tietoalkioiden arvoja.

Sivupohjiin liittyvät aina seuraavat ominaisuudet:

- Sama sivupohja voi olla käytössä monessa sivuversiossa.
- Sivupohjalla on oltava yksikäsitteinen nimi.
- Sivupohjalla on lista sen rakenteeseen liittyvistä pakollisista sisältöelementityypeistä.
- Sivupohjalla on lista sen rakenteeseen liittyvistä vapaaehtoisista sisältöelementtityypeistä.
- Sivupohjalla on html-tiedosto, johon on merkitty tunnisteilla sisältöelementtityyppien paikat.



Kuva 5.4 Sivupohjan käsittekaavio

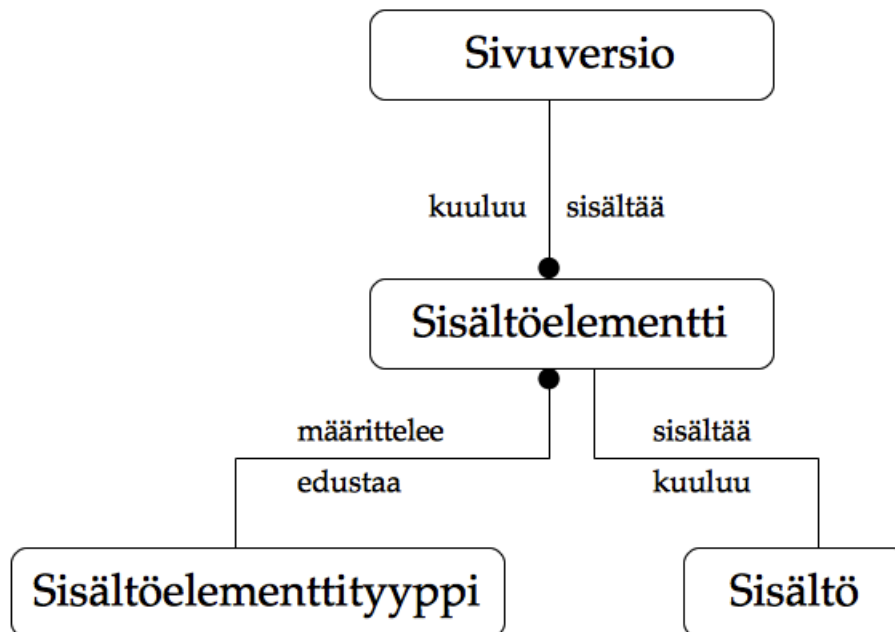
5.5 Sisältöelementti

Sivujen sisältöelementti on pienin sivuston määrittelyssä käytettävä tietoalkio. Se voi sisältää tekstiä tai multimediatiedoston. Sivupohjassa määritellään sivulla olevien sisältöelementtien määrä, mutta niitä voi olla mikä tahansa äärellinen määrä.

Sisältöelementtien pitää toteuttaa seuraavat ominaisuudet:

- Sisältöelementin on kuuluttava johonkin sivuversioon.
- Sisältöelementti edustaa yhtä sisältöelementtityyppiä.
- Sisältöelementillä on oltava sisältöä.

- Sisältöelementin sisältö tulkitaan sisältöelementin sisältöelementtityypin sisältötyypin mukaisesti joko tekstinä tai multimediana.



Kuva 5.5 Sisältöelementin käsitekaavio

5.6 Sisältöelementtityyppi

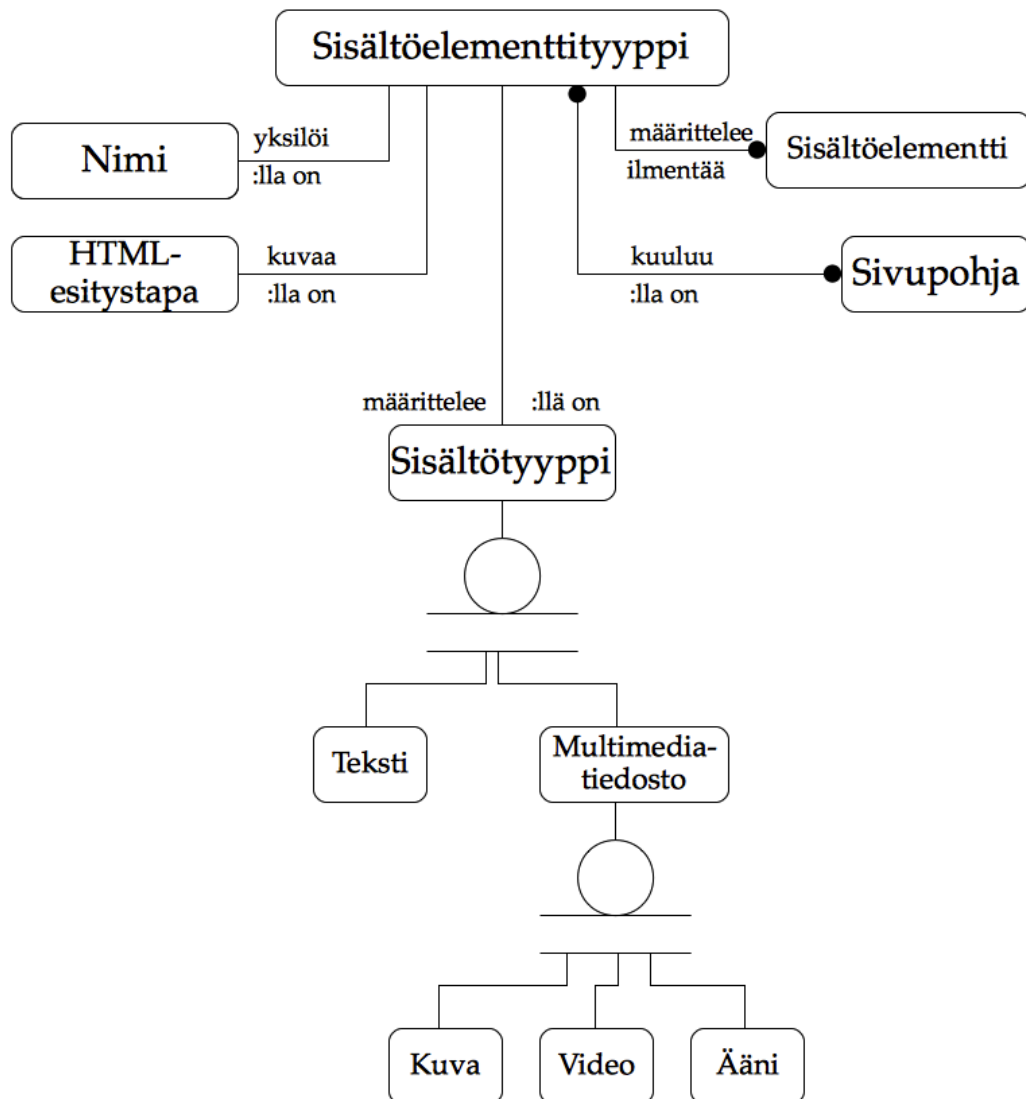
Sisältöelementtityypillä määritellään sisältöelementtien sisällön rakenne. Sivupohjat käyttävät sisältöelementtejä rakenteensa määrittelyyn.

Olio-ohjelmoinnin terminologialla sisältöelementtityyppi ymmärretään luokaksi, jonka sisältöelementti toteuttaa. Sisältöelementtityypejä voivat olla esimerkiksi otsikko, leipäteksti ja kuvituskuva.

Sisältöelementtityypit määritellään seuraavien sääntöjen mukaan:

- Sisältöelementtityypillä on sisältötyyppi.
- Mahdolliset sisältötyypit ovat teksti tai multimediatiedosto.

- Multimediatiedosto voi olla kuva-, ääni- tai videotiedosto.
- Sisältöelementtityypillä on yksikäsitteinen nimi.
- Sisältöelementtityypille on määritelty yksikäsitteinen html-esitystapa.
- Sisältöelementtityyppiä voidaan käyttää useissa sivupohjissa.
- Sisältöelementtityyppi voi määrittellä useita sisältöelementtejä.



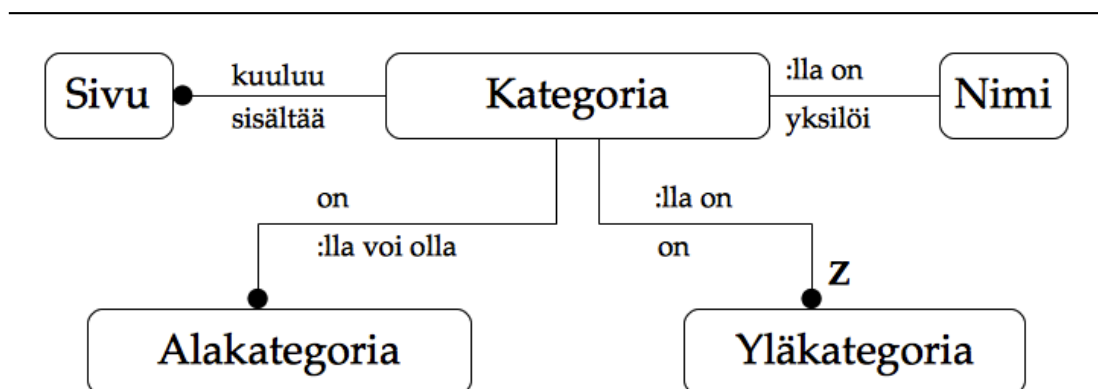
Kuva 5.6 Sisältöelementtityypin käsittekaavio

5.7 Kategoria

Kategorioilla ryhmitellään sivuston sivuja jonkin ryhmittelyperiaatteen perusteella. Kategoriat voivat sisältää alakategorioita, joten ne muodostavat hierarkkisen rakenteen, jonka avulla sivustolle voidaan määritellä navigaatiovalikko. Koska sivun pitää kuulua johonkin kategoriaan mutta kategorialla ei ole välttämättä yhtään sivua, on sivuston perustamisvaiheessa luotava ensin yksi kategoria.

Kategorian muodostamisessa noudatetaan seuraavia sääntöjä:

- Yhteen kategoriaan voi kuulua monta sivua.
- Jokaisella kategorialla on oltava nimi.
- Kategorialla voi olla alakategorioita.
- Kategorialla on yksi ja vain yksi yläkategoria, paitsi jos se on ylimmän tason kategoria, jolloin sillä ei ole yläkategoriaa.



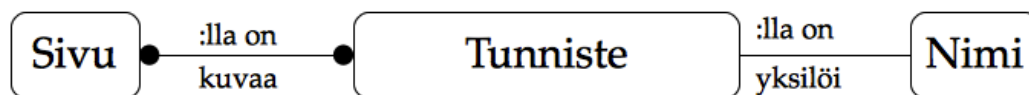
Kuva 5.7 Kategorian käsittekaavio

5.8 Tunniste

Tunnisteet ovat kategorioita vapaamuotoisempia sanojen tai asioiden perusteella tehtyjä luokitteluja. Ne voivat luokitella sisältöä yli kategoriarajojen ja sivujen. Kaikki tietyllä tunnisteella merkitty sisältö voidaan näyttää samassa näkymässä.

Tunnisteella on aina oltava seuraavat ominaisuudet:

- Tietty tunniste voi liittyä useisiin sivuihin.
- Jokaisella tunnisteella pitää olla nimi.



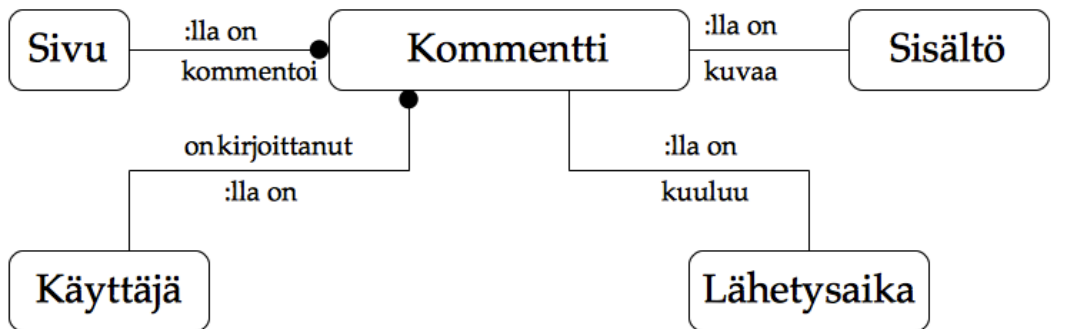
Kuva 5.8 Tunnisteen käsittekaavio

5.9 Kommentti

Sivulle voi kirjoittaa kommentteja, jos sivulla on sallittu kommentointi. Kommentit säilyvät sivulla riippumatta sivuversioiden lukumäärästä ja näkyvyydestä. Joissakin web-sisällönhallintajärjestelmissä on mahdollista kommentoida kommenttia muodostaen täten hierarkkisen rakenteen. Tämän tutkielman määrittelyssä ei sellaista kuitenkaan mahdollisteta, sillä kommenttien tulee ensisijaisesti koskea itse sivua – ei toisia kommentteja.

Kommentit esitetään seuraavien sääntöjen mukaisesti:

- Jokainen kommentti kuuluu tiettyyn sivuun.
- Jokaisella kommentilla on kirjoittaja.
- Jokaisella kommentilla on sisältö.
- Jokaiseen kommenttiin liitetään lähetysaika.



Kuva 5.9 Kommentin käsittekaavio

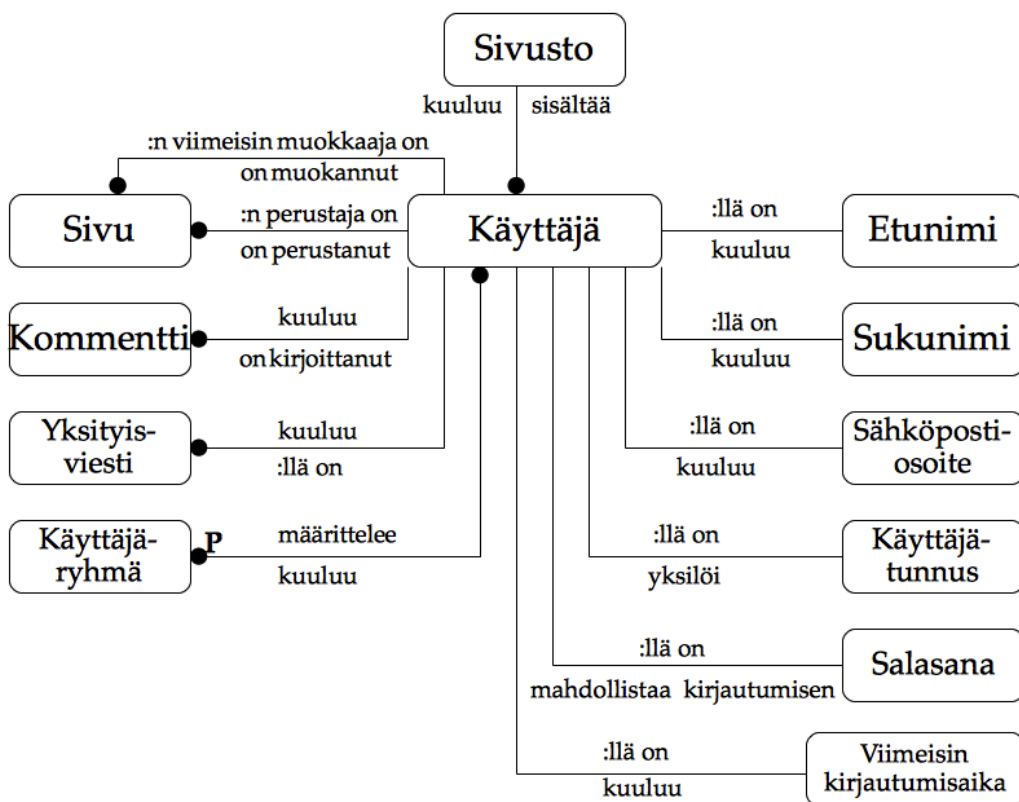
5.10 Käyttäjä

Käyttäjillä tarkoitetaan tässä määrittelyssä web-sisällönhallintajärjestelmän ylläpitoa suorittavia käyttäjiä eikä sivustolla vierailevia käyttäjiä.

Käyttäjien määrittelyssä noudatetaan seuraavia periaatteita:

- Käyttäjä kuuluu johonkin sivustoon.
- Käyttäjällä pitää olla etunimi ja sukunimi.
- Käyttäjällä on oltava sähköpostiosoite.
- Käyttäjällä on oltava yksikäsitteinen käyttäjätunnus.
- Käyttäjällä on oltava salasana.
- Jokainen käyttäjä kuuluu vähintään yhteen käyttäjäryhmään.
- Jokaisella käyttäjällä on käyttäjäryhmänsä kautta käyttöoikeustaso, joka määrittelee, mitä toiminnallisuuksia käyttäjä voi ilmaista järjestelmässä.
- Jos käyttäjä on kirjautunut sisään vähintään kerran, käyttäjälle on tallennettu viimeisin kirjautumisaika järjestelmään.
- Käyttäjä voi perustaa sivustolle sivuja.

- Käyttäjä voi muokata sivuston sivuja, jos hän on perustanut muokattavan sivun.
- Käyttäjä voi lähettää ja vastaanottaa yksityisviestejä.
- Käyttäjä voi kirjoittaa sivuille kommentteja.



Kuva 5.10 Käyttäjän käsittekaavio

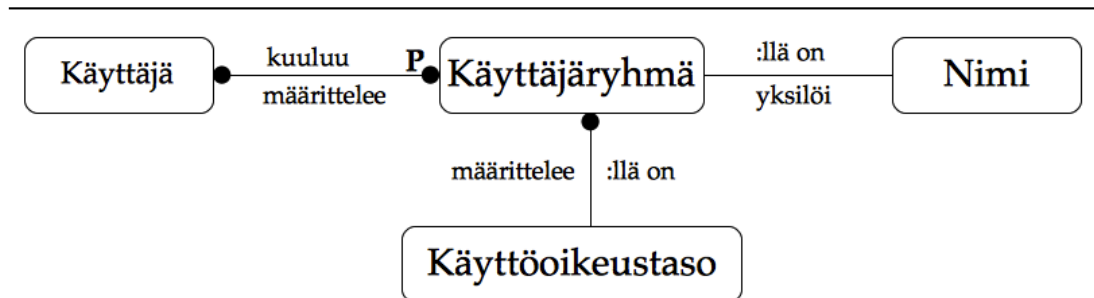
5.11 Käyttäjryhmä

Käyttäjien ryhmittely on hyödyllistä käyttäjien yleishallinnan ja käyttöoikeuksien hallinnan kannalta.

Käyttäjryhmän määrittely perustuu seuraaviin sääntöihin :

- Jokaisella käyttäjryhmällä pitää olla yksikäsitteinen nimi.

- Tiettyyn käyttäjäryhmään voi kuulua useita käyttäjiä.
- Jokaiselle käyttäjäryhmälle annetaan käyttöoikeustaso.



Kuva 5.11 Käyttäjärhmän käsitekaavio

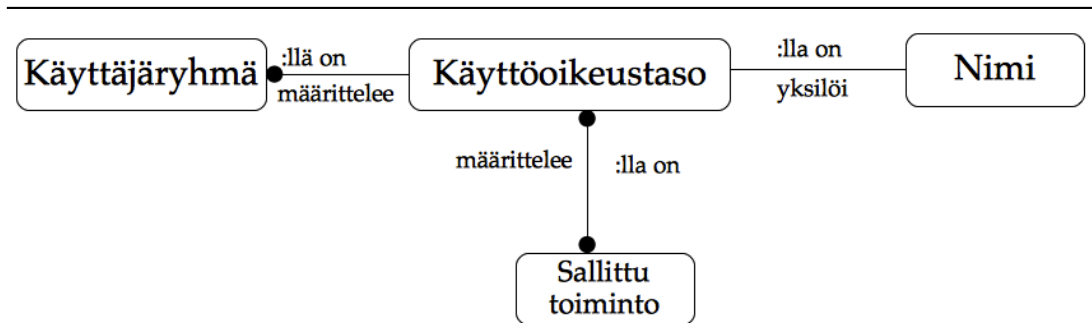
5.12 Käyttöoikeustaso

Käyttöoikeustasolla määritellään käyttäjälle sallitut toiminnot. Määrittely tehdään käyttäjärhmittain sen sijaan, että jokaiselle käyttäjälle määriteltäisiin sallitut toiminnot erikseen. Käyttöoikeuksia ei liitetä käyttäjärhmiin suoraan vaan käyttöoikeustasojen kautta. Tämä mahdollistaa käyttöoikeustasojen uudelleenkäytön. Semanttinen peruste käyttäjärhmien muodostamiselle voi olla erilainen, vaikka käyttöoikeustaso onkin sama.

Käyttöoikeustasot määritellään seuraavien periaatteiden mukaisesti:

- Käyttöoikeustasolla on oltava yksikäsitteinen nimi.
- Samalla käyttöoikeustasolla voi olla useita käyttäjärhmiä.
- Käyttöoikeustasolla voi olla useita sallittuja toimintoja.
- Jos käyttöoikeustasolla ei ole yhtään sallittua toimintoa, käyttöoikeustason käyttäjät eivät saa ilmaista mitään järjestelmän toimintoja.
- Sama sallittu toiminto voi liittyä useisiin käyttöoikeustasoihin.

- Käyttäjän sallitut toiminnot ovat kaikkien hänen kaikkien käyttäjäryhmien-
sä käyttöoikeustasojen sallitut toiminnot.



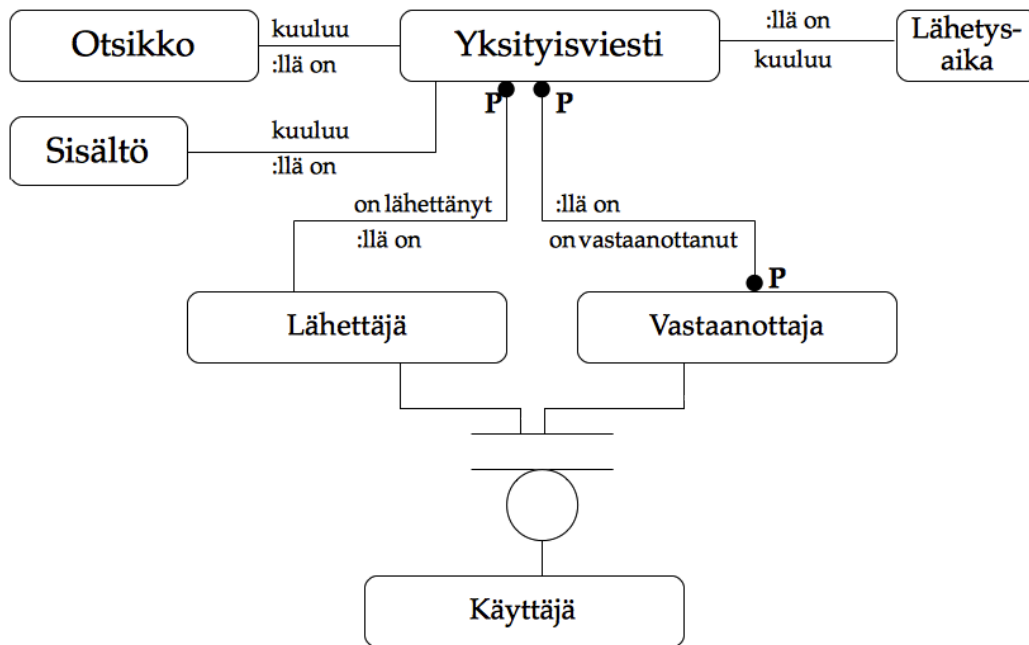
Kuva 5.12 Käyttöoikeustason käsittekaavio

5.13 Yksityisviesti

Käyttäjien välille määritellään yksityisviestijärjestelmä, jolla järjestelmän käyttäjät voivat lähettää toisilleen esimerkiksi järjestelmää koskevia viestejä. Viestit voivat olla myös järjestelmän automaattisesti generoimia ilmoituksia.

Yksityisviestien muodostamisperiaatteet:

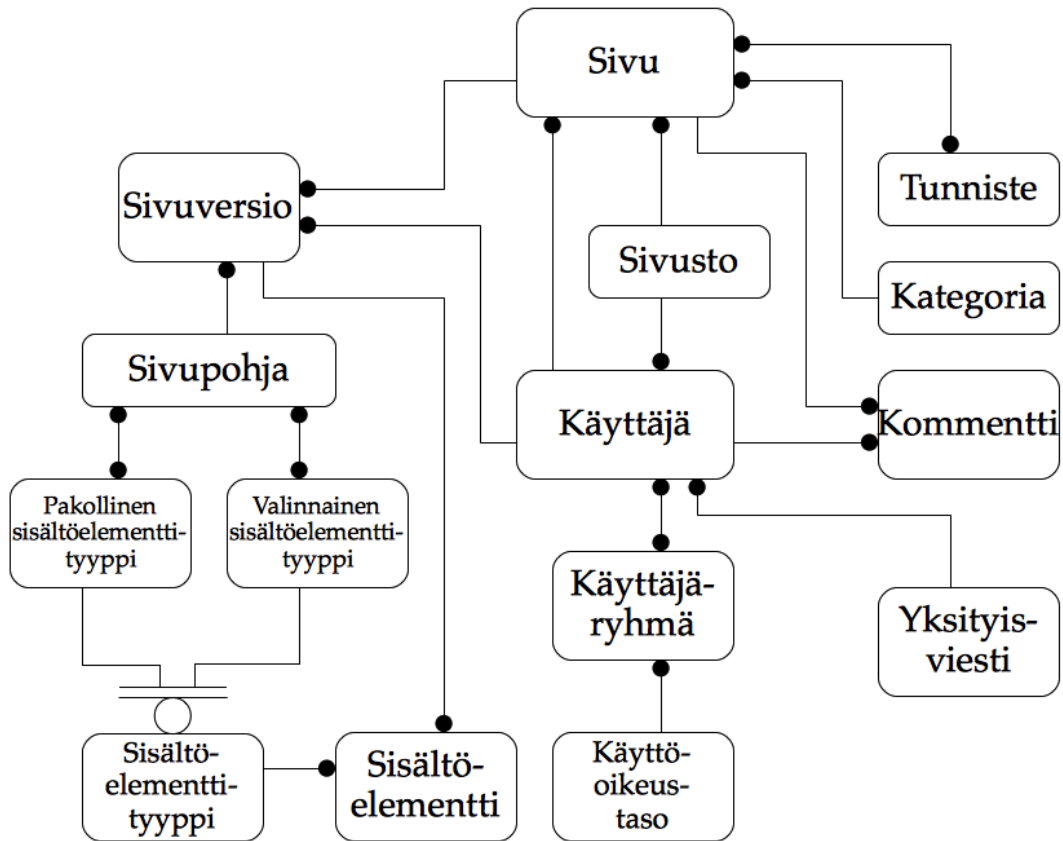
- Yksityisviestillä on oltava otsikko.
- Yksityisviestillä on oltava sisältö.
- Yksityisviestillä on oltava lähettäjä.
- Yksityisviestillä on oltava vähintään yksi vastaanottaja.
- Yksityisviestillä on täsmällinen lähetysaika.
- Yksityisviestin voivat lukea vain sen lähettäjä ja vastaanottajat.



Kuva 5.13 Yksityisviestin käsittekaavio

5.14 Koko järjestelmän käsittekaavio

Seuraavassa esitetään yksinkertaistettu versio koko järjestelmän käsittekaaviosta. Se kuvaa, mitkä ovat web-sisällönhallintajärjestelmän pääkomponentit ja niiden keskinäiset riippuvuussuhteet. Matalimpien tasojen käsitteet ja suhteiden sanalliset kuvaukset on jätetty tilan puutteen vuoksi pois. Ne voi kuitenkin lukea kunkin käsitteen omasta osakäsittekaaviosta.



Kuva 5.14 Koko järjestelmän yksinkertaistettu käsittekaavio

6 KRITIIKKIÄ WEB-SISÄLLÖNHALLINTAJÄRJESTELMIÄ KOHTAAN

Vaikka web-sisällönhallintajärjestelmissä on todettu olevan paljon hyviä ominaisuuksia, on syytä myös tarkastella niihin kohdistuvaa kritiikkiä. Kritiikkiä on esitetty web-sisällönhallintajärjestelmien käytettävyydestä, suorituskyvystä ja tietoturvasta.

6.1 Käytettävyys

Web-sisällönhallintajärjestelmillä luotujen sivustojen käytettävyyttä on arvosteltu. Vuonna 2011 tehdyn käytettävyysanalyysin [López *et al.*, 2011] mukaan web-sisällönhallintajärjestelmät eivät oletusarvoisesti johdata käyttäjiä luomaan käytettävyysnormien, kuten Nielsenin heuristiikkojen [Nielsen, 1994], mukaista sisältöä. Käytettävyyttä voidaan parantaa konfiguroimalla sivupohjia, mutta tämä vaatii käyttäjiltä asiantuntemusta web-ohjelmoinnista. Toisaalta käytettävyysosaaminen tai käytettävyysasiantuntijan konsultointi on muutenkin hyödyllistä suurta sivustoa perustettaessa.

6.2 Suorituskyky

Web-sisällönhallintajärjestelmällä luodut web-sivustot ovat dynaamisia, ja ne useimmiten latautuvat hitaammin ja kuormittavat palvelinta enemmän kuin staattiset html-sivut [Vaidyanathan & Mautone, 2009]. Tämä johtuu siitä, että staattinen html-sivu on välittömästi näytettävissä, kun taas web-sisällönhallintajärjestelmässä palvelimen pitää löytää ensin näytettävään sivuun liittyvä sisältö ja sovittaa se sivupohjaan ennen näyttämistä. Lisäksi mahdolliset näkymien tunnustekooodeissa tapahtuvat prosessoinnit, kuten silmukat ja ehtolauseet, saattavat hidastaa sivulatauksia.

Monissa web-sisällönhallintajärjestelmissä on mahdollisuus käyttää sivujen välimuistiin lataamista (*page caching*), mutta ilman sitä jokainen sivulataus aiheuttaa vähintään yhden tietokantakyselyn. Jos tietyllä web-sivulla on tiettyinä ajanhetkenä useita samanaikaisia kävijöitä, tietokantajärjestelmä ei välttämättä pysty vastaamaan tietokantaan kohdistuvien kyselyjen suureen määrään, ja koko sivuston käyttäminen saattaa jumiutua. Tämä puolestaan johtuu lopulta siitä, että

järjestelmässä tapahtunutta lukkiintumistilannetta ei saada purettua.

Suorituskyvyn säilyminen on kuitenkin yksi olennaisista vaatimuksista suurten sivustojen kehityksessä [Shoara *et al.*, 2011]. Se edellyttää samanaikaisuuden käsittelyn hallintaa. Jälleen kerran asiantuntija-apua tarvitaan sivuston suorituskyvyn optimoimiseksi.

6.3 Tietoturva

Vuonna 2009 tehdyssä tutkimuksessa [Vaidyanathan & Mautone, 2009] selvitettiin, että vaikka useimpien web-sisällönhallintajärjestelmien tietoturvan perusasiat ovat kunnossa, niissä voi piillä riskejä. Riskeihin luettiin esimerkiksi ns. *cross site scripting* -hyökkäykset, joissa käyttäjän näkymään ladataan tietokantaan tallennettu haitallinen komentosarja, tai SQL-injektiot, joissa tietokannan sisältöä päästään käsittelemään, jos lomakkeelta syötettävät tiedot luetaan SQL-komennoiksi tietokantaan sellaisenaan.

Vaidyanathan & Mautone esittivät tutkimuksessaan [2009] listan asioista, jotka uuden web-sisällönhallintajärjestelmän käyttöön ottavan ylläpitäjän pitäisi käydä läpi ollakseen varma, että sivuston tietoturva on riittävällä tasolla. Listassa mainitaan muun muassa tietokannan, evästeiden, istuntojen ja hakemistojen turvallisuusasetuksien konfigurointi. Sen ymmärtäminen ja käyttöönotto vaatii jo kohtalaista kokemusta www-järjestelmien ylläpidosta. Tutkimuksessa [Vaidyanathan & Mautone, 2009] ei ollut esimerkkijärjestelminä tässä tutkielmassa käsiteltyjä web-sisällönhallintajärjestelmiä Joomlaa, Drupalia ja WordPressia.

Shoara [et al., 2011] esittää päinvastaisen näkökulman tutkimuksessaan pitäen turvallisuutta yhtenä web-sisällönhallintajärjestelmien hyvistä puolista. Toisaalta myös Vaidyanathan ja Mautone [2009] pitävät omien tutkimustulostensa mukaisesti konfiguroitua web-sisällönhallintajärjestelmää parempana ratkaisuna kuin staattista html-sivustoa.

6.4 Muita näkökohtia

Valmiin web-sisällönhallintajärjestelmän käyttämiseen saattaa sisältyä joitakin hankaluuksia. Esimerkiksi saattaa olla sellaisia sivuston rakenteeseen tai ulkoasuun liittyviä tarpeita, joita ei voida toteuttaa edes monimutkaisimpien web-

sisällönhallintajärjestelmien suomilla mahdollisuuksilla. Välttämättä edes modulaariset lisäosat eivät tarjoa tarvittavia toiminnallisuuksia. Tällöin on mahdollista joko muokata avoimen lähdekoodin web-sisällönhallintajärjestelmää tai kehittää omiin tarpeisiin sopiva lisäosa. Poissuljettua ei ole myöskään kehittää oma web-sisällönhallintajärjestelmä sivustolle, jolloin tarvittavien resurssien määrä tosin nousee huomattavasti. Käytössä oleva web-sisällönhallintajärjestelmä voidaan myös tarvittaessa liittää hallitsemaan vain osaa sivuston sisällöstä. Viimeisenä vaihtoehtona on myös sisällönhallintaominaisuuksien pois jättäminen, jolloin menetetään web-sisällönhallintajärjestelmällä saavutettavat hyödyt.

Web-sisällönhallintajärjestelmä ei ole aina teknisesti mahdollinen ratkaisu. Halvimmissa webhotellipalveluissa ja palvelimissa ei ole edes mahdollista käyttää web-ohjelmointikieliä ja tietokantaa (kuten PHP:tä ja MySQL:ää), jotka web-sisällönhallintajärjestelmät lähes poikkeuksetta vaativat.

Lopuksi on syytä mainita, että web-sisällönhallintajärjestelmät on tarkoitettu suurten ja/tai usein päivittämistä tarvitsevien sivustojen ylläpitoon [eCe, 2001]. Ei ole todennäköisesti kannattavaa asentaa web-sisällönhallintajärjestelmää yksinkertaiselle ja/tai sisällöltään muuttumattomalle web-sivustolle, koska asennus- ja konfigurointityö saattaa olla suhteettoman työläs saavutettuun hyötyyn nähden.

Sivustoa perustettaessa on kartoitettava perusteellisesti tuleva sisällönpäivitystarve, käytettävissä oleva rahamäärä sivuston ylläpitoon ja sivuston käyttötarkoitus. Kartoituksen jälkeen voidaan tehdä johtopäätökset siitä, tarvitaanko sivustolla web-sisällönhallintajärjestelmää.

7 LOPUKSI

Tässä tutkielmassa on esitelty web-sisällönhallintajärjestelmien keskeiset käyttötarpeet ja toimintaperiaatteet. Niiden hyödyllisyys nopeassa sisällönhallinnassa ja monimutkaisten portaalien ylläpitämisessä on todettu kiistattomaksi.

Tutkimus on tehty tarkastelemalla ja analysoimalla kolmea yleisintä valmista web-sisällönhallintajärjestelmää – Joomlaa, Drupalia ja WordPressia. Niiden keskeisimmät ominaisuudet, samanlaisuudet ja eroavaisuudet on tunnistettu. Drupalia voidaan pitää käyttöominaisuuksiltaan vaativimpana mutta monipuolisimpana, kun taas WordPress on helpoin omaksua ja ottaa käyttöön mutta ominaisuuksiltaan suppein. Joomla taas asettuu näiden kahden väliin. Sen käyttäminen on yksinkertaista, vaikka se tarjoaa monipuoliset ominaisuudet ja on myös laajennettavissa lisäosilla.

Kolmen yleisimmän web-sisällönhallintajärjestelmän yhteisten ja hyödyllisimpien ominaisuuksien perusteella on tunnistettu web-sisällönhallintajärjestelmien keskeiset ominaisuudet ja luotu yleinen keskeiset ominaisuudet sisältävä käsittekaavio. Tämän käsittekaavion tarkoituksena on mahdollistaa oman web-sisällönhallintajärjestelmän kehittäminen.

Tutkielman aiheen valinta oli koko tutkielman tekemisen pisin prosessi mutta tapahtui lopulta luonnollisesti. Kiinnostus web-järjestelmiin on tullut työelämästä, tiedonhallinta taas maisteriohjelmasta. Näiden kahden yhdistelmänä löytynyt aihepiiri, web-sisällönhallinta, tuntui sekä kiinnostavalta että hyödylliseltä.

Joomlan, Drupalin ja WordPressin valinta vertailtaviksi järjestelmiksi oli mielekästä niiden yleisyyden vuoksi. Lisäksi ne ovat riittävän samankaltaisia vertailtavaksi mutta kuitenkin monin tavoin erilaisia. Kussakin on ainakin jokin sellainen ominaisuus, jota muissa ei ole. Lisäksi näistä kolmesta web-sisällönhallintajärjestelmästä löytyy hyvin lähdekirjallisuutta niiden suosion ja avoimen lähdekoodin vuoksi.

Keskeisenä tutkimusongelmana oli löytää web-sisällönhallintajärjestelmien keskeiset ominaisuudet. Siihen tutkielma antaa kattavan ja perustellun vastauksen. Tutkimusongelma on relevantti, koska tietääkseni vastaavaa vertailevaa tutkimusta web-sisällönhallintajärjestelmistä ei ole aiemmin tehty.

Tähän tutkielmaan olisi voitu luontevasti lisätä vielä oma käyttökokemustutkimus tai käytännön toteutus esitetystä käsittekaaviosta. Resurssit ja tutkielman

pituus asettivat kuitenkin rajoituksia, joiden puitteissa piti suorittaa karsimista. Lisäksi tarvetta omiin käyttökokemustutkimuksiin ei ollut, koska aiheesta oli tarjolla tuoreita ja monipuolisia tutkimustuloksia.

Tässä tutkielmassa ei esitetty juurikaan parannus- tai jatkokehitysideoita olemassaoleviin web-sisällönhallintajärjestelmiin. Jatkotutkimusta voisikin kohdistaa siihen, miten web-sisällönhallintajärjestelmien toimintaa voisi tehostaa ja parantaa entisestään esimerkiksi uusilla tietokantaratkaisuilla.

Kaikki tässä tutkielmassa tarkastellut web-sisällönhallintajärjestelmät toimivat suhteellisen perinteisillä relaatiotietokannoilla. Esimerkiksi oliopohjaisen tietokannan käyttäminen web-sisällönhallintajärjestelmässä voisi olla mahdollinen jatkotutkimuksen aihe. Vertailevaa tutkimusta voisi tehdä myös vertaamalla ilmaisia ja kaupallisia sovelluksia keskenään tai vertaamalla eri ohjelmointikielillä toteutettuja web-sisällönhallintajärjestelmiä.

VIITELUETTELO

- [Alexa, 2012] Alexa Top 500 Global Sites. Available at:
<http://www.alexa.com/topsites>. Last Checked: 10.1.2012.
- [Analogik Organisation, 2012] Analogik Organisation: History of Joomla!. Available at: <http://analogik.org/Historyofjoomla.html>. Last Checked: 5.1.2012.
- [Apache, 2012] Apache Module mod_rewrite Available at:
http://httpd.apache.org/docs/current/mod/mod_rewrite.html. Last Checked: 1.3.2012.
- [Berners-Lee & Fischetti, 1999] Tim Berners-Lee & Mark Fischetti. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. Harper San Francisco, 1st edition, 1999, 1–51.
- [Brackett, 2000] M.H. Brackett. *Data resource quality: turning bad habits into good practices*. Addison-Wesley information technology series. Addison-Wesley, 2000.
- [Business Rules Group, 2000] Business Rules Group: Defining Business Rules What Are They Really?. Available at:
http://www.businessrulesgroup.org/first_paper/br01c0.htm. Last Checked: 15.2.2012.
- [Cloud Hosting, 2012] Cloud Hosting: Joomla Demo. Available at:
<http://demo17.cloudaccess.net/administrator/>. Last Checked: 24.4.2012.
- [Douglass *et al.*, 2006a] Robert Douglass, Mike Little, & Jared Smith. Configuring Drupal. In *Building Online Communities with Drupal, phpBB, and WordPress*, pages 21–60. Apress, 2006.
- [Douglass *et al.*, 2006b] Robert Douglass, Mike Little, & Jared Smith. Introducing WordPress. In *Building Online Communities with Drupal, phpBB, and WordPress*, pages 369–378. Apress, 2006.

- [Drupal, 2010] Drupal Technology Stack. Available at:
<http://drupal.org/node/176052>. Last Checked: 29.12.2011.
- [Drupal, 2012a] Sites Made with Drupal. Available at:
<http://drupal.org/cases>. Last Checked: 29.12.2011.
- [Drupal, 2012b] Drupal Clean URLs. Available at:
<http://drupal.org/getting-started/clean-urls>.
Last Checked: 29.12.2011.
- [Drupal, 2012c] Drupal History. Available at:
<http://drupal.org/about/history>. Last Checked: 29.12.2011.
- [eCe, 2001] eCenter, Faculty of Organizational Sciences, University of Maribor.
Web Content Management, Bled, Slovenia, June 2001.
- [Frick, 2010] Tim Frick. *Return on Engagement: Content, Strategy and Design Techniques for Digital Marketing*. Elsevier, 1st edition, 2010.
- [Hackos, 2002] Joann T. Hackos. *Content Management for Dynamic Web Delivery*. John Wiley & Sons, Inc., New York, NY, USA, 2002, 7–49.
- [Johnsen, 2010] Torkil Johnsen: Joomla Database Schema. Available at:
<http://www.torkiljohnsen.com/2010/07/11/Joomla-1-6-database-schema/>.
Last Checked: 30.12.2011.
- [Joomla!, 2012a] Joomla. Available at: <http://www.joomla.org/>.
Last Checked: 30.12.2011.
- [Joomla!, 2012b] About Joomla. Available at:
<http://www.joomla.org/about-joomla.html>. Last Checked: 30.12.2011.
- [Kentico, 2012] Kentico. Available at: <http://www.kentico.com>.
Last Checked: 30.12.2011.
- [Leary, 2009] Stephanie Leary. About WordPress. In *Beginning WordPress 3*, pages 1–14. Apress, 2009.

- [López *et al.*, 2011] Juan López, Afra Pascual, Llucia Masip, Toni Granollers, & Xavier Cardet. Influence of web content management systems in web content accessibility. In Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno Nunes, Philippe Palanque, & Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2011*, volume 6949 of *Lecture Notes in Computer Science*, pages 548–551. Springer Berlin / Heidelberg, 2011.
- [Myers & French, 2008] Jim Myers & Larry French. Voice for all: adopting content management in frontline technical support. In *Proceedings of the 36th annual ACM SIGUCCS fall conference: moving mountains, blazing trails*, SIGUCCS '08, pages 113–120, New York, NY, USA, 2008. ACM.
- [Nielsen, 1994] Nielsen, Jakob. *Usability Engineering*. Academic Press, San Diego, 1994, 115–148.
- [OpenSourceCMS, 2012a] OpenSourceCMS: Drupal Demo. Available at: <http://www.opensourcecms.com/demo/2/183/Drupal>.
Last Checked: 24.4.2012.
- [OpenSourceCMS, 2012b] OpenSourceCMS: WordPress Demo. Available at: <http://www.opensourcecms.com/demo/wordpress>.
Last Checked: 24.4.2012.
- [Patel *et al.*, 2011] Savan K. Patel, Dr. V. R. Rathod, & Jigna B. Prajapati. Performance analysis of content management systems – Joomla, Drupal and WordPress. *International Journal of Computer Applications*, 21(4):39–43, May 2011. Published by Foundation of Computer Science.
- [Plone, 2012] Plone. Available at: <http://www.plone.org>.
Last Checked: 27.12.2011.
- [Pon, 2009] Pondicherry University, Puducherry. *Open Source Software for Content Management System*. INFLIBNET Centre, Ahmedabad, February 2009.
- [Rahmel, 2009] Dan Rahmel. Introduction to Joomla! In *Beginning Joomla!*, pages 1–9. Apress, 2009.
- [Royal Pingdom, 2009] Royal Pingdom: The blog platforms of choice among the top 100 blogs Available at: <http://royal.pingdom.com/2009/01/15/>

[the-blog-platforms-of-choice-among-the-top-100-blogs/](#).

Last Checked: 8.5.2012.

[Shoara *et al.*, 2011] Ahmad Shoara, Kambiz Badie, & Maryam Tayefeh Mahmoudi. Web design considerations for realizing functions in content management systems. In Azizah Abd Manaf, Akram Zeki, Mazdak Zamani, Suriyati Chuprat, & Eyas El-Qawasmeh, editors, *Informatics Engineering and Information Science*, volume 252 of *Communications in Computer and Information Science*, pages 463–478. Springer Berlin / Heidelberg, 2011.

[Tomlinson *et al.*, 2010] Todd Tomlinson, John K. VanDyk, Todd Tomlinson & John K. VanDyk. How Drupal works. In Lowman *et al.*, editor, *Pro Drupal 7 Development*, pages 1–11. Apress, 2010.

[Upsites Web Studio, 2011] Upsites Web Studio: Drupal Database Schema. Available at:

<http://upsitesweb.com/blog/2011/06/Drupal-7-database-schema>.

Last Checked: 30.12.2011.

[Vaidyanathan & Mautone, 2009] Ganesh Vaidyanathan & Steven Mautone. Security in dynamic web content management systems applications. *Commun. ACM*, 52:121–125, December 2009.

[Wikimedia Foundation, 2012] Wikimedia Foundation: Our Projects. Available at: http://wikimediafoundation.org/wiki/our_projects.

Last Checked: 19.1.2012.

[WordPress, 2012a] WordPress Database Description. Available at:

http://codex.wordpress.org/database_description.

Last Checked: 29.12.2011.

[WordPress, 2012b] WordPress. Available at: <http://wordpress.org>.

Last Checked: 29.12.2011.

[WordPress, 2012c] About WordPress. Available at:

<http://wordpress.org/about/>. Last Checked: 29.12.2011.

A IDEF1X-VISUALISOINTITAPA

Business Rules Group on käyttänyt liiketoimintasääntöjen käsittekaavioesityksissä IDEF1X-visualisointitapaa. Sitä on käytetty myös tämän tutkielman luvun 5 käsittekaavioiden esittämisessä. Tässä liitteessä esitellään lyhyesti IDEF1X-visualisointitapa.

IDEF1X-visualisointitavan käsittekaaviot koostuvat oliotyypeistä ja niiden välisistä suhteista. Oliotyyppien suhteet muodostavat yhtenäisen graafin, jossa voi olla syklejä ja joka on suuntaamaton. Jokainen kahden oliotyypin välinen suhde yhdistää oliotyypit toisiinsa kaksisuuntaisesti.

Suhteet muodostetaan riittävän atomiselle tasolle normalisoiduista liiketoimintasäännöistä. Ilmaisemalla oliotyyppi, suhde ja toinen oliotyyppi muodostuu luonnollisella kielellä tulkittava liiketoimintasääntö (tämä tosin on hankala toteuttaa yksikäsitteisesti suomen kielellä sijapäätteiden vuoksi).

Suhteiden molemmat suunnat tulkitaan myötöpäivään kulkevassa suunnassa. Suhteisiin liittyvät tulkinnat kirjoitetaan oliotyyppiä yhdistävän viivan läheisyyteen. Tulkinta, joka liittyy vasemmalla puolella olevan oliotyypin suhteeseen oikealla puolella olevaan oliotyyppiin, kirjoitetaan viivan yläpuolelle, kun taas viivan alapuolelle kirjoitetaan tulkinta, joka liittyy oikealla puolella olevan oliotyypin suhteeseen vasemmalla puolella olevaan oliotyyppiin. Vastaavasti alaspäin kuljettaessa tulkinta on ilmaistu oikealla ja ylöspäin kuljettaessa vasemmalla.

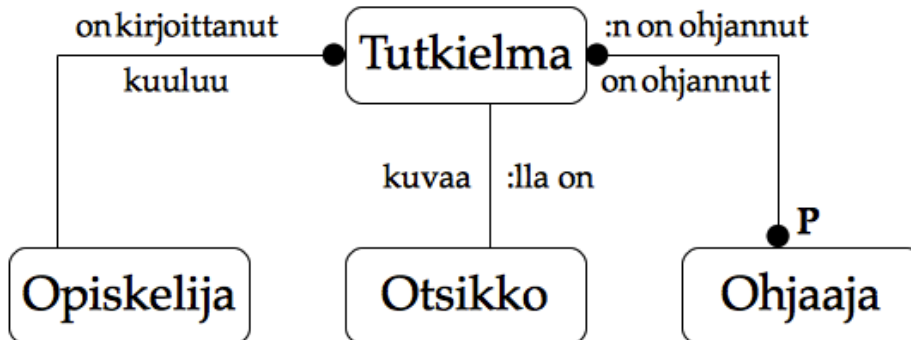
Oliotyyppien välisiin suhteisiin voi liittyä myös kardinaalisuus- ja pakollisuusrajoitteita. Nämä ilmaistaan Kuvassa A.1 esitetyillä erikoissymboleilla, jotka kiinnittyvät suhdetta ilmaisevan viivan päähän myötöpäivään kuljettaessa jälkimmäiseen oliotyyppiin kiinni. Suhteen eri suunnilla voi olla eri kardinaalisuus- ja pakollisuusrajoitteet, eivätkä ne vaikuta toisiinsa.

Mahdollinen yhden suhde yhteen ja vain yhteen	_____ Z ●
Pakollinen yhden suhde yhteen ja vain yhteen	_____
Mahdollinen yhden suhde yhteen tai useampaan	_____ ●
Pakollinen yhden suhde yhteen tai useampaan	_____ P ●

Kuva A.1 IDEF1X-visualisointitavan kardinaalisuus- ja pakollisuussäännöt.

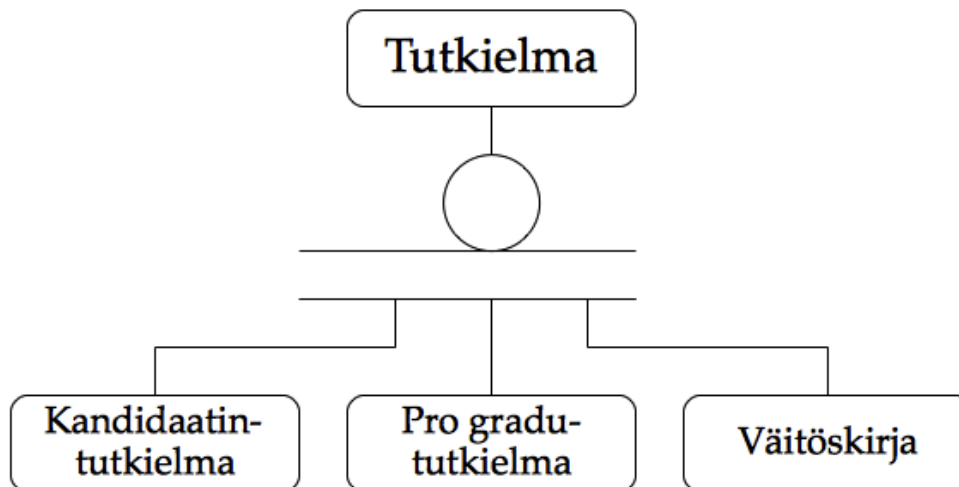
Seuraavassa esimerkissä muodostetaan muutamasta yksinkertaisesta liiketoimintasäännöstä IDEF1X-visualisointitavalla käsitelkaavio (kuva A.2), joka kattaa oliotyyppien ja suhteiden sekä kardinaalisuus- ja pakollisuusrajotteiden visuaaliset ilmaiset:

- Tutkielma kuuluu yhdelle ja vain yhdelle opiskelijalle.
- Opiskelija voi kirjoittaa useita tutkielmia, mutta opiskelijan ei ole tarvinnut välttämättä kirjoittaa vielä yhtään tutkielmaa.
- Tutkielmalla on oltava vähintään yksi ohjaaja, mutta ohjaajia voi olla myös useampia.
- Ohjaaja voi toimia ohjaajana useissa tutkielmissa.
- Tutkielmalla on oltava otsikko.



Kuva A.2 IDEF1X-visualisointitapaa havainnoillistava esimerkki.

Liiketoimintasäännöt ja IDEF1X mahdollistavat myös alaoliotyyppien käsittelyn. Jos oliotyyppi jakautuu alaoliotyyppihin siten, että jokainen oliotyypin ilmentymä edustaa jotain alaoliotyypeistään, olio ja sen alaoliotyyppit voidaan ilmaista toisiinsa yhteenkuuluvasti. Tästä käytetään esimerkkinä suljettua maailmaa, jossa kaikki tutkielmat ovat joko kandidaatin tutkielmia, pro gradu -tutkielmia tai väitöskirjoja. Esimerkki visualisoidaan kuvassa A.3.



Kuva A.3 Oliotyypin ja sen alaoliotyyppien visualisointi IDEF1X-visualisointitavassa.