

WWW-

# SISÄLLÖNHALLINTAJÄRJESTELMÄT

Case: DotNetNuke

LAHDEN AMMATTIKORKEAKOULU  
Tietojenkäsittelyn koulutusohjelma  
Yritysviestintäjärjestelmät  
Www-sisällönhallintajärjestelmät  
Case: DotNetNuke  
Kevät 2008  
Satu Mustonen

Lahden ammattikorkeakoulu  
Liiketaloudenlaitos

Satu Mustonen:

Www-Sisällönhallinta  
Case: DotNetNuke

Yritysviestintäjärjestelmien opinnäytetyö, 45 sivua, 2 liitesivua

Kevät 2008

## TIIVISTELMÄ

---

Tässä opinnäytetyössä oli tavoitteena selvittää ja perehtyä www-sisällönhallintaan ja sen järjestelmiin, niiden tärkeimpiin perusominaisuuksiin ja elementteihin. Sekä käydä läpi sisällönhallintaprosessia kokonaisuudessaan. Case-osiossa on keskitytty käsittelemään vapaan lähdekoodin DotNetNuke-järjestelmää ja verrattu sen ominaisuuksien vastaavuutta sisällönhallintajärjestelmiltä vaadittuihin peruselementteihin. Case osiossa käsitellään kyselytutkimuksen tuloksia, joka suoritettiin virallisen DotNetNuke-sivuston kautta maaliskuussa 2007.

Www-sisällönhallinnassa on keskitytty lähinnä monikanavajulkaisun ja moninaisten www-sisältöjen mukana tuomiin haasteisiin. Tunnusomaisimpia piirteitä www-sisällönhallinnalle on sisällön ja ulkoasun erottaminen toisistaan, monipuoliset sisältöyksiköt ja älykkäät sivupohjat, sekä sisältöjen personointi ja metatietojen korostunut rooli. Www-sisällönhallintajärjestelmien tärkeimmiksi ominaisuuksiksi nousivat sisältökokoelma, työvirranhallinta, yhteydet muihin järjestelmiin sekä hallintatyökalut ja administraatiojärjestelmä.

Case-osion DotNetNuke vastaa joiltakin osiltaan sisällönhallintajärjestelmiltä vaadittuja ominaisuuksia, kuten työvirranhallintaan pohjautuvilta moduuleiltaan sekä hallinta ja administraatiotyökaluiltaan, muttei DotNetNukea kuitenkaan voida kutsua sisällönhallintajärjestelmäksi. Kuvaavampi nimitys järjestelmälle on sovelluskehys tai verkkohallintajärjestelmä. DotNetNukeen vapaa lähdekoodi tarjoaa kuitenkin käyttäjilleen rajattomat mahdollisuudet muokata järjestelmää mihin suuntaan hyvänsä.

Www-sisällönhallinta ja sisällönhallinta kokonaisuudessaan kattavat monta osakokonaisuutta ja niiden rajat ovat suhteellisen häilyvät. Lisäksi ne ovat aihealueena jatkuvasti uusiutuvien menetelmien ja teknologioiden haasteena. Toisin sanoen sisällönhallintajärjestelmät ovat jatkuvasti kehittyviä järjestelmiä. Näin ollen sisällönhallinta tarjoaa monta hyvää ja monipuolista tutkimusaihetta niille, jotka ovat siitä kiinnostuneet.

Avainsanat: www-sisällönhallinta, www-sisällönhallintajärjestelmä, sisällönhallinta, DotNetNuke

Lahti University of Applied Sciences  
Faculty of Business Studies

SATU MUSTONEN:

Web Content Management Systems  
Case: DotNetNuke

Bachelor's Thesis in Business Information Systems, 45 pages, 2 appendices

Spring 2008

## ABSTRACT

---

This bachelor's thesis deals with web content management and its systems, their most important basic elements and the process of content management. The case study discusses the open source DotNetNuke system; how well do its elements correspond to the principles of the basic elements of the content management systems. The survey for the case study was performed through the official website of DotNetNuke in March and April 2007.

The web content management part focuses on the challenges of multichannel publishing and multiple contents of websites. The most typical features for web content management are the division between content and layout, complex content units and smart templates, as well as the impersonation of content and highlighted role of metadata. The most important features for web content management systems are content repository, workflow management, connections to other systems as well as the administration system and controlling tools.

DotNetNuke has some features required from content management systems, such as modules based on a workflow management and administration- and management tools. However, DotNetNuke cannot be called a representative of content management systems. A more suitable name for the system would be an application framework or a web management system. Nevertheless, for the users the open source code of DotNetNuke offers infinite possibilities to modify the system.

Web content management and content management overall tie many subassemblies together and it is hard to draw a line between them. Also they are a challenge for the developing methods and technologies. In other words they are continuously progressing systems. Under these circumstances content management offers many study subjects for those that are interested.

Keywords: web content management, web content management systems, content management, DotNetNuke

## SISÄLLYS

1	JOHDANTO	1
2	TUTKIMUSONGELMA JA RAJAUKSET	3
3	TUTKIMUSPROSESSI	3
4	SISÄLLÖNHALLINTAJÄRJESTELMÄT	4
4.1	Sisällönhallintajärjestelmien tarkoitus	6
4.2	Sisällönhallinta ja osa-alueet	8
4.2.1	Sisällön tallennus	9
4.2.2	Sisällönhallintajärjestelmiä	9
5	WWW-SISÄLLÖNHALLINTA	11
5.1	Www- sisällönhallinnan tavoitteet	12
5.2	Www- sisällönhallinnan prosessi	12
6	WWW – SISÄLLÖNHALLINTAJÄRJESTELMIEN OMINAISUUDET	15
6.1	Kokoelma	17
6.2	Työvirta	20
6.3	Yhteydet muihin järjestelmiin	21
6.4	Hallintatyökalut ja administraatiojärjestelmä	22
7	WWW-SISÄLLÖNHALLINTAJÄRJESTELMÄESIMERKKEJÄ	23
7.1	OpenCMS	23
7.2	Joomla	24
7.3	RedDot CMS	25
7.4	Hot Banana: Web Content Management Suite	25
8	CASE: DOTNETNUKE	27
8.1	DNN:n evoluutio	27
8.2	DNN:n filosofia	28
8.3	Avainominaisuuksia	29
8.4	DNN:n arkkitehtuuri	30
8.4.1	Presentaatiokerros	31
8.4.2	Bisneslogiikkakerros	32

8.4.3	Tiedonsaantikerros	32
8.4.4	Datakerros	32
8.5	Sovellusohjelmointirajapinta	33
8.6	Sisällönhallinta ja rakenne	34
8.7	Hallinnointi	35
9	TUTKIMUSTULOKSET	35
9.1	Puutteet	38
9.2	Laajennusmahdollisuudet	38
9.3	Tulosten yhteenveto	40
10	YHTEENVETO	41
	LÄHTEET	44
	LIITTEET	46
	LIITE 1/1	47
	LIITE 1/2	48

## 1 JOHDANTO

Www-sisällönhallinta juontaa juurensa jo 1990-luvun puoliväliin, jolloin suurin kasvu tapahtui, kun isot yritykset tarvitsivat kattavia www-sivustoja (Yo 2005, 2). Kuten kuvitella saattaa tiedonmäärä oli joissakin tapauksissa valtavaa ja välttämättä se ei ollut kaikissa helpoimmassa ja yksinkertaisimmassa muodossa. Tarvittiin sisällönhallintaa. Vaikkakin sisällönhallinta käsitteenä onkin vanha asia, www-sisällönhallinta on kuitenkin vielä suhteellisen nuori ja kukoistaa. Se kuinka www-yhteiskunta on jatkuvassa myllerryksessä ja uusia teknologioita kehitetään jatkuvasti, varmasti pitää www-sisällönhallinnan vielä pitkään kehittyvänä varantona.

Sisällönhallinta voidaan yksinkertaisuudessaan kuvata prosessina kerätä, organisoida, luokitella ja koostaa minkä tyyppisiä informaatioresursseja tahansa ja muokata niitä niin, että ne on mahdollista tallentaa, hakea, julkaista, päivittää ja uudelleenkäyttää jokaisen haluamalla tavalla (Yu 2005, 6).

Merkittävin tavoite sisällön julkaisemisessa www-sisällönhallintajärjestelmän kautta on automatisoida koko kirjoittamis- ja julkaisuprosessi yhdistämällä siihen www-hallintatyökalut laajoine tietovarastoineen ottaen huomioon kuitenkin, että käyttäjät tulevat vastaanottamaan hyvin järjesteltyä ja ajankohtaista tietoa joka näyttää ja tuntuu yhdenmukaiselta. Tällaiset tietovarastot on suunniteltu helpompaan resurssienhallintaan ja -takaisin saantiin, sekä niiden yhdistämiseen vaihtelevan valikoiman www-esityksiin. (Yu 2005, 6.)

Keskeisimpinä käsitteinä nostettakoon esiin mm. metatieto, dokumentit, sisältöyksiköt ja sisällönhallinta. Metatiedolla (engl. metadata) tarkoitetaan tietoa tiedosta. Metatiedot ovat tyypillisesti jokaisen itse määrittelemiä, mutta on myös olemassa metatietoja, joita tarvitaan lähes kaikissa tapauksissa. Tällaisia yleisiä metatietoja ovat esimerkiksi dokumentin luontipäiväys, laatijan nimi, otsikko ja asiasanat. Metatietoja tarvitaan esimerkiksi silloin kun sisältöä halutaan etsiä,

uudelleen järjestää tai vertailla eri versioita. Dokumentilla (engl. document) tarkoitetaan itsenäistä loogista sisältöyksikköä joka on tarkoitettu ensisijaisesti ihmisen tulkittavaksi (Honkaranta & Tyrväinen 2005, 550). Sisältöyksikkö (engl. content unit) tarkoittaa tietokokonaisuutta johon on liitetty hallinnointi-tarkoituksiin käytettävää metatietoa. Tämänkaltainen objekti voi olla esimerkiksi dokumentti, tiedosto, komponentti tai kappale asiakirjasta (Honkaranta & Tyrväinen 2005, 550-551). Sisällönhallinta (engl. content management) on laaja käsite joka liittyy moninaisten sisältöjen hallintaan, kuitenkin sähköisessä muodossa olevien esim. sähköpostiviesti, tietokanta, tekstimuotoinen dokumentti, kuva-, video- tai äänitiedosto.

Tarkoituksena siis tässä opinnäytetyössä on käsitellä sitä mitä sisällönhallinta on sekä mitä ovat sisällönhallintajärjestelmät. Esimerkkien kautta käydään läpi sisällönhallintaprosessia sekä niitä ominaisuuksia, jotka tekevät sisällönhallintajärjestelmästä sisällönhallintajärjestelmän. Kuitenkin koska koko sisällönhallinta terminä kattaa enemmän kuin yhden sisällönhallinnan osa-alueen, opinnäytetyö on rajattu koskemaan pääpiirteissään vain www- sisällönhallintaa sekä sen järjestelmiä.

Case-osiossa perehdytään vapaan lähdekoodin järjestelmään, DotNetNukeen, sekä käydään läpi tutkimustuloksia siitä, mitä mieltä aktiiviset DotNetNuken käyttäjät ovat, että järjestelmää verrataan www- sisällönhallintajärjestelmään. Tutkimusongelmana olikin selvittää vastaavatko DotNetNuken ominaisuudet sisällönhallinnan peruseriaatteisiin ja tältä pohjalta päätellä voidaanko sitä kutsua sisällönhallintajärjestelmäksi.

## 2 TUTKIMUSONGELMA JA RAJAUKSET

Tarkoituksena tässä opinnäytetyössä on selvittää mitä www-sisällönhallinta-järjestelmällä tarkoitetaan ja nostaa esille sisällönhallintajärjestelmien tärkeimpiä ominaisuuksia, joiden valossa Case-osion DotNetNukea tullaan tarkastelemaan. Tutkimusongelmana on siis selvittää vastaavatko DotNetNuken ominaisuudet niitä tärkeimpiä sisällönhallintajärjestelmiltä vaadittavia ominaisuuksia joita tässä opinnäytetyössä tullaan käsittelemään.

Opinnäytetyö on rajattu koskemaan vain www-sisällönhallintaa ja sen järjestelmiä. Koska sisällönhallinta jo pelkkänä käsitteenä on valtava ja kattaa sisäänsä niin monia eri osa-alueita ja pursuaa erinäisiä järjestelmiä moniin eri käyttötarkoituksiin, jolloin niiden kaikkien käsitteleminen ei ole mahdollista yhdessä työssä. Varsinkin jos halutaan paneutua enemmän niiden perimmäiseen käyttötarkoitukseen.

## 3 TUTKIMUSPROSESSI

Opinnäytetyöhön on valittu kvantitatiivinen Case-tutkimus. Tutkimus suoritettiin DotNetNuken virallisen www- sivuston kautta (<http://www.dotnetnuke.com>) maaliskuussa 2007 ja vastaajia koko tutkimuksessa oli yhteensä 10 kappaletta. Vaikka otanta olikin näin pieni, osa vastaajista kuului DotNetNuken kehitystiimiin ja toiset vastaajista olivat aktiivisia ja pitkäaikaisia DotNetNuken käyttäjiä. Näin ollen vastaukset perustuvat vankkaan ja täsmälliseen tietoon sekä kokemukseen aiheesta. Tutkimukseen tehtiin kysymyslomake (Liite 1), joka sijoitettiin DotNetNuke sivustolle. Lisäkysymyksiä esitettiin vastaajille aihealueesta riippuen.

Tietopohja opinnäytetyössä on sekä teoreettista että empiiristä. Opinnäytetyön laatija on työskennellyt aikoinaan DotNetNuke –järjestelmän kanssa, ja näin ollen siinä saavutettu tietotaito on ohjannut tässä työssä eteenpäin.

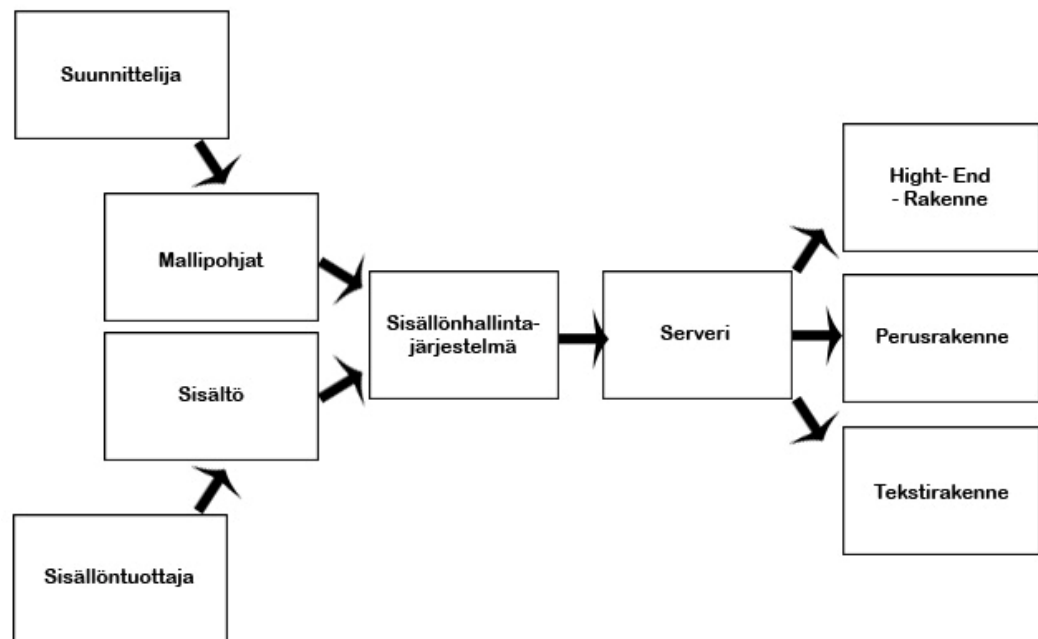


## 4 SISÄLLÖNHALLINTAJÄRJESTELMÄT

Sisällönhallintajärjestelmät (engl. Content Management System tai CMS), ovat tietokoneohjelmistojärjestelmiä, jotka edesauttavat käyttäjää prosessoimaan sisällönhallintaa. Toisin sanoen sisällönhallintajärjestelmät mahdollistavat helposti kontrolloida suurisisältöisiä dokumentteja ja muuta sisältöä, kuten kuvia ja multimediaresursseja. Sisällönhallintajärjestelmät mahdollistavat myös usein dokumenttien luonnin yhteistyössä monen henkilön kesken.

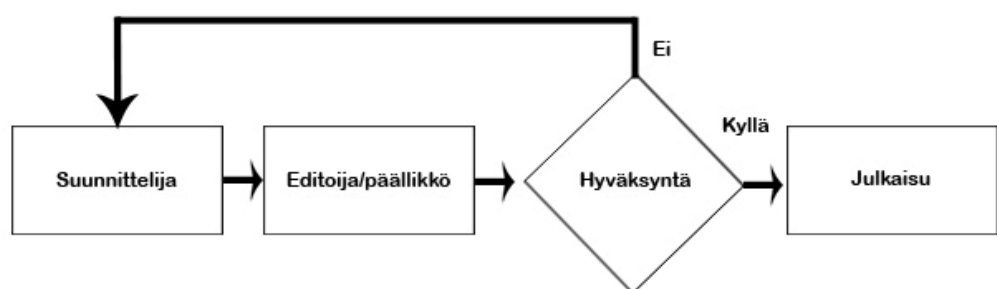
Tällä hetkellä sisällönhallintajärjestelmä käsite voidaan jakaa moneen eri osaan. Sisällönhallintajärjestelmätyyppejä on jo lukuisia, niistä voidaan erikseen ainakin mainita 1) dokumenttienhallintajärjestelmät, 2) www-sisällönhallintajärjestelmät, 3) verkkokauppajärjestelmät ja 4) aineistonhallintajärjestelmät.

Sisällönhallintajärjestelmän yhtenä päätehtävänä on siis erottaa rakenne ulkoasusta. Sisällönhallintajärjestelmiä on niin pieniä kuin isojakin, sekä yksinkertaisia kuin myös hyvin monimutkaisia. Mutta kaikki perustuvat kuitenkin samalle perusidealle. Järjestelmä antaa suunnittelijan keskittyä suunnitteluun valmiiden mallipohjien avulla. Toisin sanoen voitaisiin todeta, että vähemmänkin tietotekniikkaa tunteva henkilö voi sisällönhallintajärjestelmän avulla luoda hyvin monipuolisia sekä laaja-alaisia tietokokonaisuuksia. Seuraavan esimerkkikuvion 1, joka mukailee Ellisin (2002, 1) ideaa, mukaan tietosisällöt on tuotettu erillisessä ympäristössä ja sisällytetty sisällönhallintajärjestelmään. Serveri ohjaa tietosisällöt oikeaan sivumalliin ja lähettää ne eteenpäin siististi paketoituna loppukäyttäjille. (Ellis 2002, 1.)



Kuvio 1 Kuinka sisällönhallinta toimii (Ellis 2002, 1).

Kuitenkin sisällönhallintajärjestelmien tehtäviä on muitakin kuin vain hallita ja järjestellä sisältöä. Yksi erityisen tärkeä ominaisuus on työvirranhallinta. Myös seuraava Ellisin teoriaa mukaileva kuvio antaa esimerkin työvirran kulusta.



Kuvio 2 Sisällönhallinnan työvirta (Ellis 2002, 1).

Työvirranhallinta lähtee liikkeelle suunnittelijan tekemästä työstä, jonka sisällönhallintajärjestelmä ohjaa eteenpäin, seuraavaa vaihetta hoitavalle henkilölle. Jos dokumentti hyväksytään, ohjataan se suoraan serverille ja julkaistaan. Mutta jos sitä ei ole hyväksytty, ohjataan se takaisin suunnittelijalle tarvittavine korjausohjeineen tai ehdotuksineen. Kaikki tämä hoituu sisällönhallintajärjestelmän kautta. Ei siis ylimääräisiä sähköpostiviestejä, ei puheviestejä, ei tulostenippuja sekä punamustetta eikä huomiolappuja joka paikassa. Kaikki tuo edellä mainittu siis sisältyy sisällönhallintajärjestelmään itsessään. Lopputuloksena on yksinkertainen ja helppo tapa hoitaa sisältö ja julkaisu. (Ellis 2002, 1.)

Yksi tärkeimmistä sisällönhallintajärjestelmien ominaisuuksista, vielä edellämainittujen ominaisuuksien lisäksi, on yhteys muihin järjestelmiin ja ohjelmiin. Tämä ominaisuus mahdollistaa lähettää ja vastaanottaa tietosisältöä tallennuspaikasta. Myös hallinnollisella moduulilla on oma paikkansa sisällönhallintajärjestelmässä. Tämän avulla käyttäjä pystyy hallinnoimaan koko sisällönhallintajärjestelmää ja sen eri osia. (Boiko 2005, 88.)

#### 4.1 Sisällönhallintajärjestelmien tarkoitus

Kuten aiemmin jo todettiin, sisällönhallintajärjestelmiä on monia erilaisia ja jokainen paneutuu eri tehtäviin. Sisällönhallintajärjestelmien kohdealueet voidaan jakaa kolmeen osaan, 1) kohdemedia 2) kohdekoko ja 3) kohdemalli (Butcher 2006, 6). Kohdemedialla tarkoitetaan, että sisällönhallintajärjestelmä on yleensä suunniteltu jotain tiettyä käyttöä varten, joka antaa näin kohderyhmän sisällönhallintajärjestelmän käyttäjistä. Olkoon siis sitten kyse dokumenttienhallinnasta, www-sisällöstä tai muusta vastaavasta. Jokaisella sisällönhallintajärjestelmällä on oma yksityinen tavoite auttaa jossain tietyssä asiassa.

Toinen tapa määritellä sisällönhallintajärjestelmän kohdealueet on sen kohdekoko. Kohdekoolla viitataan käyttäjämääriin. Kyse on siis siitä kuinka monen käyttäjän tarpeisiin sisällönhallintajärjestelmä on kehitetty. On sisällönhallintajärjestelmiä,

jotka on kehitetty erityisesti pienille käyttäjämäärille, joiden dokumenttien tuotanto ei ylitä muutamaa tuhatta kappaletta . Tällaiset järjestelmät usein painottavat enemmän helppokäyttöisyyttä kuin työvirranhallintaa, edistynyttä muokkausta tai sitkeyttä. (Butcher 2006, 7-8.)

Sisällönhallintajärjestelmät, jotka ovat kehitetty vastaamaan keskisuuren yrityksen editoijien tarpeita, pyrkivät arkaluontoisesti tasapainottelemaan helppokäyttöisyyden ja monimutkaisten sisällönuontikiertojen välillä. Työvirranhallinta, versiointi, julkaisutuki ja eriytyminen eri käyttäjien välillä (administraattorit, sisällön toimittajat, päälliköt, ym.) on melkein elintärkeää tämän tyyppisille sisällönhallintajärjestelmille. Tällaiset sisällönhallintajärjestelmät pystytään usein laajentamaan käsittelemään suurempia toimittajaryhmiä, mutta suunnittelu- rajoitukset saattavat estää skaalaamasta suurempia määriä dokumentteja. (Butcher 2006, 7-8.)

Jos taas on kyseessä järjestelmä joka on suunnattu isolle määrälle dokumentteja ja käyttäjiä, sisällönhallintajärjestelmän on tarjottava dokumentinhallinnan lisäksi muokausprosessi, joka on tarpeeksi monimutkainen huolehtimaan sisällöstä edellyttäen mukautettua hyväksikäyttöä. Tämän tyyppiset sisällönhallintajärjestelmät voivat siis hallita miljoonia dokumentteja tai enemmän (terabittejä tai perabittejä dataa). Tällaiset suurille käyttäjäryhmille suunnatut sisällönhallintajärjestelmät vaativat merkittävän ohjelmistokehityksen luodakseen mukautettuja liittymäraajapintoja. (Butcher 2006, 7-8.)

Kolmantena luokittelu kriteerinä on kohdemalli, tällä tarkoitetaan sisällönhallintajärjestelmän luokittelua organisatorisen mallin perusteella. Organisatorinen malli kuvaa kaikki ne käyttäjät ja käyttäjäryhmät jotka ovat yhteydessä sisällönhallintajärjestelmään. Kaksi yleisintä organisatorista mallia ovat 1) yhteisömalli (engl. community model) ja 2) hankkija-käyttäjämalli (engl. provider-consumer model). Yhteisömallissa sisällönhallintajärjestelmä on yhteisöryhmittymien keskipisteenä. Oletetaan, että sisällönhallintajärjestelmän sisältö tulee yhteisöltä ja tällaiset sisällönhallintajärjestelmät painottavat jäsenyysasetuksia. Mahdollisuus edesauttaa ja ylläpitää sisältöä on tehty helpommaksi ja painopisteenä on tarjota

jokaiselle käyttäjälle henkilökohtainen tila tuottaa sisältöä. Kun taas hankkija-käyttäjämallissa sisällönhallintajärjestelmän paikka on yhden ryhmittymän sisällä, joka tuottaa sisältöä ja toinen ryhmittymä käyttää sisältöä. Tällä ensimmäisellä ryhmällä on etuoikeuttettu pääsy sisällönhallintajärjestelmään ja he voivat luoda ja ylläpitää dokumentteja sisällönhallintajärjestelmässä. Kun taas tämä toinen ryhmä on joko käyttäjiä tai vieraita, joilla on vain oikeus tarkastella sisältöä (joskus myös rajoitettu pääsy kehittämään sisältöä, kuten mahdollisuus antaa kommentteja), mutta he eivät kuitenkaan pääse nauttimaan kaikista sisällönhallintajärjestelmän muokkaukseen tarjoamista komponenteista. (Butcher 2006, 7-8.)

#### 4.2 Sisällönhallinta ja osa-alueet

Sisällönhallinnan päätarkoituksena on siis pyrkiä hallitsemaan digitaalisessa muodossa olevaa informaatioisisältöä mahdollisimman tarkoituksenmukaisesti. Sisältönä voidaan pitää niin tekstimuotoisia dokumentteja, www-sivuja, sähköpostiviestejä, niin kuva-, audio ja videotiedostojakin. Toisin sanoen kaikkea tallennettua sekä tallennettavissa olevaa informaatiota tai dataa. Sisällönhallinnan informaatioisisältö ei sinänsä kuitenkaan ole pelkkää dataa, vaan käsitettä voidaan laajentaa koskemaan kaikkea ihmisten ymmärrettävissä olevia tiedonmurusia. Sisällönhallinnassa keskeistä on myös yleisesti digitaalisen sisällön sisältämä metatieto eli liitännäistieto. Metatieto termillä tarkoitetaan tietoa tiedosta, eli määrittävää ja kuvailevaa tietoa jostakin tieto- tai sisältöyksiköstä. Metatietojen käytöllä voidaan helpottaa tietojärjestelmien välisiä tiedonsiirtoja, sekä myös eri paikoista haettavista tietojen yhdistämisestä. Metatietojen merkitys on erittäin suuri, varsinkin ajateltaessa tiedon automaattista järjestelyä, mutta toisaalta ongelmaksi saattaa muodostua eri järjestelmissä tuotettujen metatietojen yhteensopivuus.

#### 4.2.1 Sisällön tallennus

Sisällön tallennus on yksi sisällönhallintajärjestelmien tärkeimmistä osa-alueista ja sisällönhallintajärjestelmien ensisijainen funktio (Boiko 2005, 925). Se mihin sisältöä tallennetaan eri järjestelmissä, riippuu hyvin pitkälti järjestelmän valmistajasta. Yleistä kuitenkin on se, että suurin osa sisällönhallintajärjestelmistä tallentaa elementtinsä tietokantaan, toiset taas käyttävät tietokantaa metatietojen tallentamiseen ja pitävät elementit sisältötiedostoissa. Kuitenkin sisällön tallentamiseen tarvitaan jonkin sortin tietokanta. Yleisessä käytössä on joko relaatiotietokanta (engl. Relational database) tai vastaavasti objekti- tai XML-pohjainen tietokanta. Myös se mitä sitten tietokantaan tallennetaan riippuu järjestelmän valmistajasta. Uusia teknologioita kokeillaan jatkuvasti, yritettäessä saada paras lopputulos niin tietokanta kuin tiedosto maailmasta.

Riippumatta siitä mitä tallennusjärjestelmää käytetään, tiedon tallennuksessa on tärkeintä elementtien relaatioiden tallentaminen elementtien ja sisältömallien välillä (Boiko 2005, 969). Sisällönhallintajärjestelmien tiedontallentamisen yhteydessä käytetään tietokannan sijasta kokoelmaa. Kokoelma hoitaa tehtäviä suurin piirtein tietokannan tapaan, mutta sen ominaisuuksiin kuuluu paljon enemmän kuin pelkän tavallisen tietokannan.

#### 4.2.2 Sisällönhallintajärjestelmiä

Nykyään puhuttaessa sisällönhallinnasta siihen yhdistetään usein vain www-sisällönhallinta, on unohdettu tässä internetkeskeisessä maailmassa, mistä kaikki aikoinaan on lähtenyt kehittymään. Karkeasti sisällönhallinta ja sen järjestelmät voidaan jakaa kolmeen eri ryhmään: 1) dokumenttienhallintaan 2) julkaisunhallintaan ja 3) digitaalisten resurssien hallintaan. Digitaalisten resurssien paikalla voidaan tosin pitää myös rakenteisia dokumentteja.

Dokumentilla tarkoitetaan perinteisesti jonkinlaista asiakirjaa tai todistuskappaletta. Jotakin kirjoitettua informaatiota tai dataa jostakin asiasta.

Dokumentinhallinnalla pyritään helpottamaan sekä parantamaan dokumenttien käsittelyä, niiden arkistointia ja varastointia, jakamista sekä myös niiden kokonaisvaltaista hallinnointia.

Julkaisunhallinnalla taas pyritään helpottamaan koko verkkopalvelun sisällöntuotantoa sekä totta kai sen julkaisua verkossa. Julkaisunhallintatermiä ei yleisesti ottaen enää paljонkaan käytetä vaan puhutaan tuttavallisemmin www-sisällönhallinnasta. Kun opinnäytetyö keskittyy juuri tähän osa-alueeseen, ei siihen tässä vaiheessa sen enempää syvennyttä.

Digitaalisten resurssienhallinta (engl. Digital Asset Management tai DAM) taas pyörii lähinnä digitaalisten resurssien kommentoinnin, luetteloimisen, varastoinnin ja jäljityksen ympärillä. Digitaalisia resursseja voivat olla mm. valokuvat, videot, musiikki sekä myös animaatiot.

Sisällönhallinta on kokonaisuudessaan hyvin pitkälti yhteistyöprosessi, josta tärkeimpinä rooleina ja tehtävinä voidaan erotella seuraavat toimenkuvat. Ensimmäisen paikan saa editoija, joka on paljolti vastuussa tuotetusta sisällöstä, sen ulkonäöstä ja kieliastusta, toisena tulee julkaisija joka tietysti vastaa sisällön ja tuotetun dokumentin tai informaation sisällön julkaisusta. Administraattorilla on vastuullaan hallinnoida käyttäjäoikeuksia kansioissa ja tiedostoissa, mutta administraattorit saattavat avustaa myös käyttäjiä eri tavoilla. Viimeisenä muttei merkityksettömimpänä on käyttäjä, katselija tai vieras, eli persoona joka lukee tai muuten käyttää julkaistua tietoa ja informaatiota.

Kriittisimpänä kohtana koko sisällönhallinnassa on resurssien ja dokumenttien versioinnin merkitys ja tieto, joka niihin liittyy. Tekijän ja editoijien on pystyttävä palauttamaan vanhemmat versiot muokatuista tietosisällöistä prosessihäiriöiden ja epätoivottujen versiomuutosten jälkeenkin. Olipa kyse mistä tahansa sisällönhallintajärjestelmästä, kaikilla on niitä yhteisiä ominaisuuksia ja

käyttötarkoituksia, joiden pohjalta ne on luotu. Kaikki sisällönhallintajärjestelmät toimivat aivan samojen pääperiaatteiden mukaisesti.

## 5 WWW-SISÄLLÖNHALLINTA

Www- sisällönhallinta (engl. Web Content Management tai WCM) keskittyy toimialana ja tutkimusalueena erityisesti monikanavajulkaisun ja www- sisältöjen julkaisun haasteisiin (Honkaranta & Tyrväinen 2005, 550). Www- sisällönhallinta sisällönhallinnan osa-alueena on yhdistelmä järjestelmistä ja työvirranhallinnasta, jotka mahdollistavat verkkoadministraattorien hallinnoida isoja määriä dataa moninaisista lähteistä, sekä osatekijöiden helposti editoida tiedostoja ja julkaista sisältöä www- sivustoilla (Yu 2005, 2). Yksi www-sisällönhallintajärjestelmien päätehtävistä on antaa isolle epäsuhtaiselle joukolle verkkojulkaisijoita mahdollisuus julkaista heidän tuottamansa sisältö verkkosivustoilla ja saada julkaisuprosessi näyttämään näkymättömältä (Yu 2005, VI).

Sisällönhallintaan voidaan ottaa ainakin kolme erilaista näkökulmaa.

Ensimmäisenä tulee www -sisällönhallinta, toisena dokumenttien hallinta ja kolmantena rakenteiset dokumentit. Www – sisällönhallinnalle merkityksellisimpiä ja tunnusomaisimpia piirteitä ovat pienet sisältöyksiköt ja metatietojen hallinta. (Honkaranta & Tyrväinen 2005, 550.) Unohtamatta kuitenkin sivupohjiin perustuvaa julkaisua sekä sisältöjen ja ulkoa-asun erottamista toisistaan (Yu 2005, 2).

Www- sisällönhallinta on syntynyt pääasiallisesti kolmen eri osa-alueen yhteenliittymänä: 1) dokumenttienhallinta ja työvirran hallinta 2) ohjelmisto-kehityksen versionhallintaohjelmistot sekä 3) asiakkuudenhallintaohjelmistot ja verkkokauppasovellukset. (Goodwin & Vidgen 2002, 66-70.)

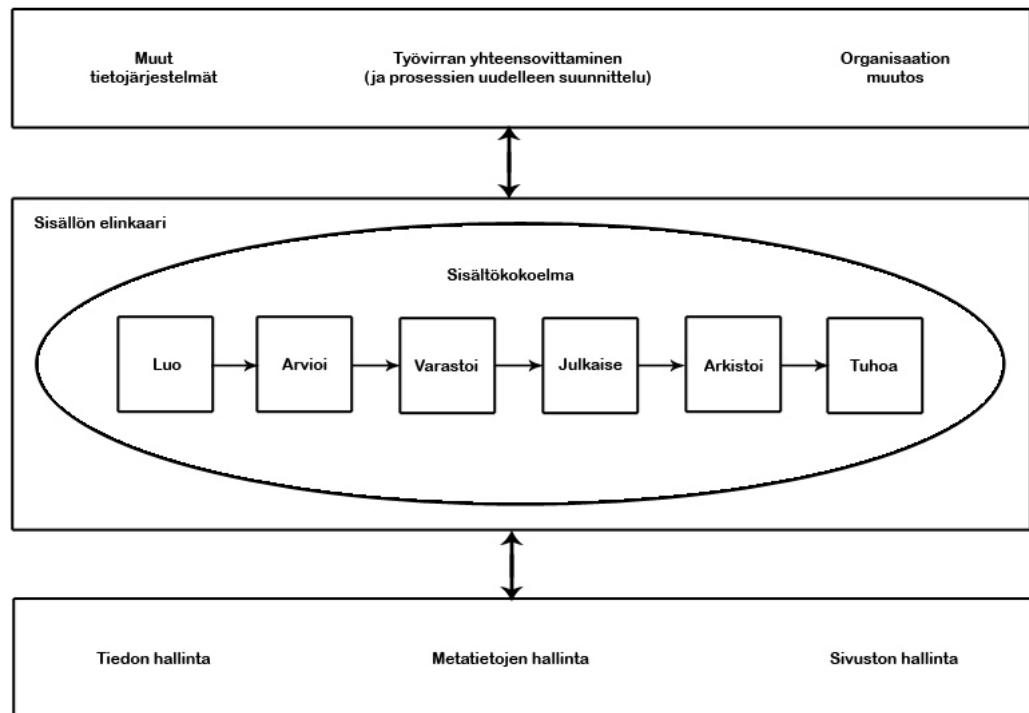


## 5.1 Www- sisällönhallinnan tavoitteet

Suurin osa yrityksistä ja organisaatioista hakee sisällönhallinnan kehittämisellä kustannusvähennyksiä sekä nopeampia tuotantoaikoja, parempaa resurssien hyödyntämistä sekä sisältöjen laatua, lisäksi myös parempaa asiakas- ja työtyytyväisyyttä. Yleisimpänä ongelmana organisaatioissa on se, että sisältöjä tuotetaan eri osastoilla, yksiköissä ja ryhmissä ja hyvin harvoin eri osastojen, yksikköjen tai ryhmien välillä käydään minkäänlaista viestintää tai keskustelua tietosisällöistä. Tuloksena tällaisesta monen tekijän tuottamasta materiaalista saattaa olla ristiriitaiset ja epäyhtenäiset tietosisällöt. Tästä ei yleensä kuitenkaan kerry kuin ylimääräisiä kustannuksia ja turhaa työtä. Edellä kuvatussa toimintamuodosta sisällön loppukäyttäjille saattaa aiheutua harmia, joskin tietosisällöt voivat olla epätarkkoja tai puutteellisia, niin myös vaikeasti löydettävissä sekä vanhentuneita. Sisällönhallinnan tarkoituksena on poistaa nuo edellä mainitut epäkohdat yrityksen tiedonhallinnassa ja käsittelyssä. Yhtenäinen järjestelmä tarjoaa näin ollen yrityksille ja organisaatioille organisoidumpaa prosessien ja tietosisältöjenhallintaa.

## 5.2 Www- sisällönhallinnan prosessi

Sisällönhallinta tulisi käsittää prosessina jota ohjelmistotyökalut ohjastavat ja avustavat. Lisäksi www-sisällönhallinnalle tyypillistä ovat monipuoliset sisällöt sekä tarve hallinnoida näitä sisältöjä jollain tavalla niiden luomisesta aina niiden lopulliseen tuhoamiseen asti. (Goodwin & Vidgen 2002, 66-70.)



Kuvio 3 Www –sisällönhallinnan viitekehys.

Yllä oleva kuvio 3 mukalee Goodwinin & Vidgenin (2002, 66-70) viitekehystä www-sisällönhallinnan prosessille. Kuvan nuolet tarkoittavat sitä mihin suuntaan tieto ja toiminta etenee, kun taas www -sisällönhallinnan eri osa-alueita kuvassa kuvastavat kolme suorakulmiota ja keskimmäisen suorakulmion sisässä olevat neliöt taas tarkoittavat eri vaiheita sisällön elinkaareissa. Ylin suorakulmio kuvaa viitekehyksessä organisaation muita tietojärjestelmiä, jo olemassa olevia työntekijöitä ja muutosta jota organisaatiolta edellytetään. Painopiste on eniten juuri ylimpänä olevan suorakulmion osa-alueilla, sillä nämä ovat kokonaisuuden kannalta tärkeitä tekijöitä, mutta ovat kuitenkin vaarassa jäädä mm. tietojärjestelmän käyttöönottoprojektien varjoon.

Kuvan keskimäinen suorakulmio sisältää koko viitekehysten ytimen, sisällön elinkaaren ja sisältökokoelman. Sisältökokoelmalla (engl. Content Repository) tarkoitetaan yhtenäistä kanavaa kaikkiin organisaation eri dokumenttiarkistoihin, tietokantoihin ja tiedostojärjestelmiin. Sisällön elinkaari kattaa sisällön kaikki vaiheet sisällön luomisesta aina sisällön tuhoamiseen järjestelmästä.

Sisällön luomisella ei viitekehysten mukaan ole merkitystä, se voidaan yhtä hyvin hoitaa erillisellä työkalulla, kuten esimerkiksi Microsoft Word tai vastaava kuin myös www-lomakkeella, riippuen tietenkin millaisesta sisällöstä on kysymys. Sisältö voidaan tuoda sen luomisen jälkeen sisällönhallintajärjestelmään ja liittää osaksi sisältökokoelmaa.

Sisällön arvioinnilla voidaan tarkoittaa monia eri asioita riippuen tilanteesta. Sisältöön voidaan vielä arviointivaiheessa lisätä muita tietoja, kuten metatietoja sekä siihen voidaan määrittellä sen varastointiin ja poistamiseen liittyviä asetuksia. Sisältö tallennetaan sisällönhallintajärjestelmään sen varastointivaiheessa ja se siirtyy siitä suoraan julkaisuvaiheeseen. Toiminnot julkaisuvaiheessa ovat riippuvaisia eri organisaatioiden omasta käytännöstä. Sisältöä voidaan prosessoida erilaisiksi eri käyttäjille ja/tai se voidaan julkaista useammassa eri julkaisukanavassa. Arkistointi voi olla joko automaattista tai käsinohjattua ja tuhoaminen tarkoittaa koko sisällön poistamista pysyvästi sisällönhallintajärjestelmästä. Tämä on yleensä ohjattu tapahtumaan automaattisesti, tietyn ajan kuluttua. (Goodwin & Vidgen 2002, 66-70.)

Kuvan alin suorakulmio sisältää www –sisällönhallintaprosessin osa-alueita jotka liittyvät mm. 1) tiedonhallintaan, 2) metatietojen hallintaan ja 3) sivustonhallintaan. Tiedonhallinnalla tarkoitetaan tallennettujen sisältöjen fyysistä tallentamista ja organisointia järjestelmään. Suurin osa aihetta käsittelevistä teoksista ja tekijöistä suosittelevat tähän erityisesti XML-pohjaisia teknologioita. Metatietojen tallentamisessa metatietojen avulla tulisi pystyä määrittelemään sisällön uudelleenkäytettävyys eikä ainoastaan ensijulkaisua. Sivuston hallinnalla tarkoitetaan monikanavajulkaisun hallintaa, jossa ulkoasun erottaminen sisällöstä on erittäin keskeisessä ja merkityksellisessä roolissa. (Goodwin & Vidgen 2002, 66-70.)

## 6 WWW – SISÄLLÖNHALLINTAJÄRJESTELMIEN OMINAISUUDET

Www-sisällönhallintajärjestelmät on yleisnimitys julkaisupainotteisille sisällönhallintajärjestelmille, mutta toisinaan se on myös synonyymi www-julkaisujärjestelmille ja julkaisujärjestelmille. Ne ovat sisällönhallintajärjestelmiä joilla voidaan niin julkaista kuin myös rakentaa yksittäisiä www-sivuja tai luoda kokonaisia www-sivustoportaaleja.

Mitä www-sisällönhallintajärjestelmiin yleisesti ottaen tulee, ne ovat järjestelmiä, jotka ovat kehitetty varta vasten isojen ja dynaamisten www-materiaalien hallinnointiin ja kontrollointiin, kuten HTML-dokumentteihin ja niihin liittyviin kuvamateriaaleihin. Www-sisällönhallintajärjestelmät on luotu helpottamaan dokumenttien ohjaamista, auditointia, editointia sekä reaaliaikaista hallinnointia. Muutamia niiden perus- tai avainominaisuuksia ovat mm. automaattiset sivumallit, helposti muotoiltavat ja editoitavat sisällöt sekä skaalattavat ominaisuudet. Automaattisilla sivumalleilla tarkoitetaan valmiiksi luotuja sivu- tai sivustopohjia, joihin yksinkertaisuudessaan voidaan vain lisätä haluttu sisältö, esimerkiksi pelkkää tekstiä tai kuvamateriaalia. Sisällön muokkaus on helppoa siinä suhteessa, koska varsinainen sisältömateriaali on erotettuna sivuston visuaalisesta asettelusta.

Useimmat www-sisällönhallintajärjestelmät sisältävät ns. WYSIWYG –editointi työkalun, joka edesauttaa ei-teknisen henkilön luoda ja muotoilla sisältöä haluamallaan tavalla. WYSIWYG on lyhenne englanninkielestä ja tarkoittaa ”What You See Is What You Get”, suomennettuna siis ”Sen mitä näet, sellaisen saat”. Tätä lyhennettä käytetään myös erinäisissä tekstinkäsittelyohjelmissa ja editoreissa, tutuin näistä ohjelmista lienee Microsoft Word-tekstinkäsittely-ohjelma, joka tukee kyseistä työkalua. Voidaan siis todeta, että useimmissa www-sisällönhallintajärjestelmissä on yhtä helppoa luoda ja käsitellä tekstipohjaista materiaalia kuin Microsoft Wordissä sekä muissa saman tyyppisissä tekstinkäsittelyohjelmissa. Skaalattavilla ominaisuuksilla taas tarkoitetaan yksinkertaisesti

asennettavia lisäosia ja moduuleita, joiden avulla voidaan laajentaa jo olemassa olevia ominaisuuksia.

Yleisimpiin ja aktiivisimpiin www-sisällönhallintajärjestelmien perusominaisuuksiin lukeutuvat myös www-standardien mukaiset päivitykset, jotka pitävät sisällönhallintajärjestelmän ajantasalla ja uusien lisäosien saatavuuden yksinkertaisena. Dokumenttien ja työvirran hallinnointi kuuluvat vielä niihin avainominaisuuksiin, joita useimmat www-sisällönhallintajärjestelmät tukevat. Dokumenttien hallinnoinnilla tarkoitetaan sitä, että sisällönhallintajärjestelmän avulla voidaan hallinnoida dokumentin koko elämänkaarta, ensimmäisestä luontihetkestä, tarkistuksen, julkaisun ja arkistoinnin kautta sen hävitykseen. Kun taas työvirran hallinnoinnilla tarkoitetaan erinäisten työtehtävien jaksottamista ja järjestämistä sisällönhallintajärjestelmän avulla.

Www- sisällönhallintajärjestelmät voidaan ryhmitellä neljälle eri tasolle: näennäiset www-sisällönhallintajärjestelmät, tietokantapohjaiset tai dynaamiset verkkosivustot, täydelliset www- sisällönhallintajärjestelmät ja yritys sisällönhallintajärjestelmät (Boiko 2005, 72-73).

Näennäiset www- sisällönhallintajärjestelmät ovat työkaluja kuten esimerkiksi Microsoftin muistilehtiö (engl. Notepad) ja FrontPage sekä Macromedian Dreamweaver. Nämä työkalut tarjoavat yksinkertaisen menetelmän www-sisällönhallinnalle, kuten standardiverkkosivujen luonnin sivupohjien avulla. Ne tarjoavat myös sivuluonnoksia ja linkitystyökaluja auttamaan tekijäänsä linkittämään sivut toimimaan keskenään. Julkaisunhallinta taas auttaa verkkosivuston ylläpitäjiä lataamaan uudet ja muokatut sivut www-servereille. Tällaiset työkalut ovat mainioita pienien ja yksittäisten www-sivujen/sivustojen hallintaan ja luontiin.

Tietokantapohjaiset tai dynaamiset www-sivustot eivät ole www-sisällönhallintajärjestelmiä teknisesti ottaen, vaan parhaiten niitä kuvaa termi verkkopohjaiset sovellukset. Yleisesti ottaen sovellus sisältää tietopohjan tallennettuna www-serverille. Tietopohja voidaan luoda käyttämällä

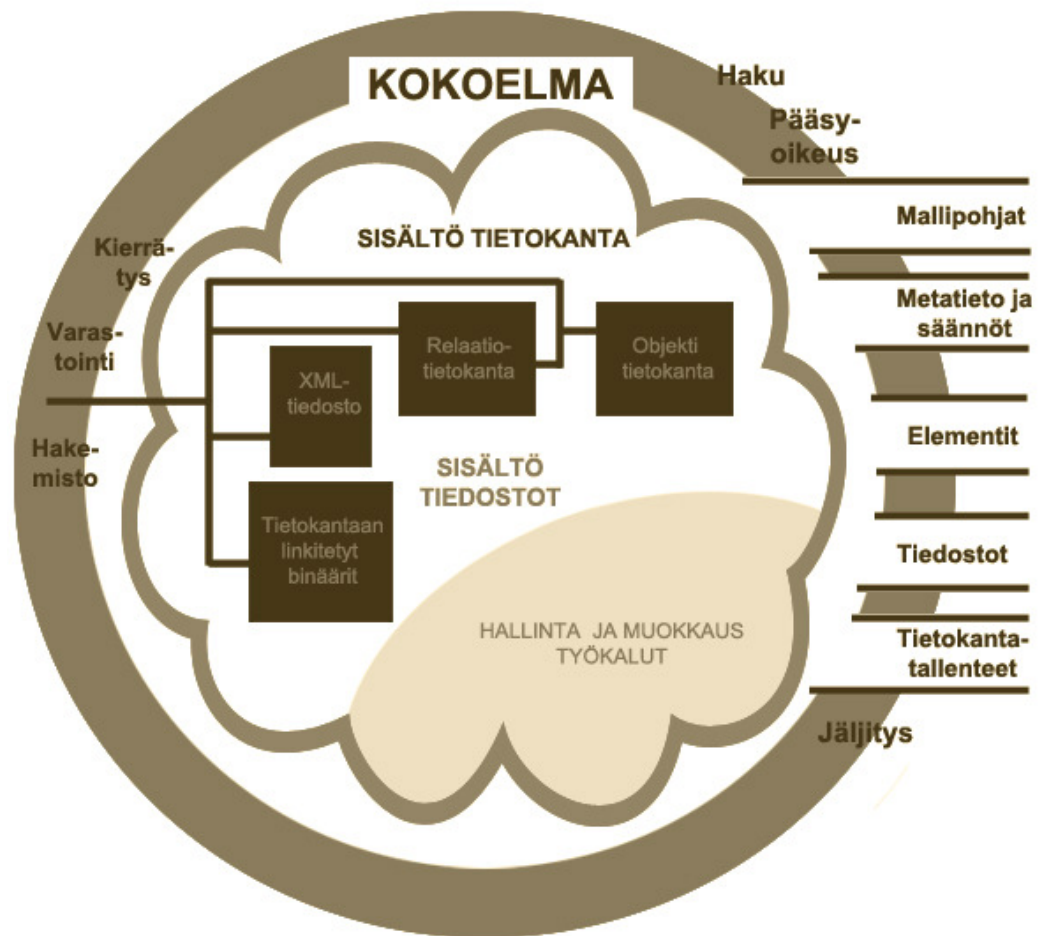
relaatiotietokantaa, XML-rakennetta tai muiden skriptikielten stukturoitua rakennetta, kuten esimerkiksi JavaScriptiä. Näillä sivustoilla on käytössä sivumalli joka yhdistää käyttäjän tietolähteeseen. Käyttäjän lähettäessä kutsun aktivoida tietopohja, sivumalli sisältää joko HTML-koodia, skriptejä tai muita objekteja jotka tulkitsevat nämä käyttäjän kutsut tietopohjalle, joka lähettää tarvittavan sisällön käyttäjän www-selaimeen HTML-sivuna. Dynaamiset verkkosivustot ja www-sisällönhallintajärjestelmät molemmat käyttävät tietokantaa prosessoimaan sisältöä. Nämä kaksi asiaa tulisi erotella toisistaan. Käyttäjällä voi olla käytössään dynaaminen sivusto joka ei tee sisällönhallintaa, mutta toisaalta taas sisällönhallintajärjestelmä voi olla yhtä helposti rakennettavissa kuin staattinen sivusto. (Boiko 2005, 75.)

Täydelliset www-sisällönhallintajärjestelmät toimivat läpi koko sisällöntuotantoprosessin (Boiko 2005, 79). Sisällön keräämisestä julkaisuun, sekä ne myös hallinnoivat sisällön tuottajia ja työvirtaa. Täydellinen www-sisällönhallintajärjestelmä sisältää relaatiotietokannan tarjoten kokoelman kaikenlaiselle sisällölle, aina tekstitiedostoista, HTML-tiedostojen, grafiikan kautta aina tyyllisivuihin asti. Yrityssisällönhallintajärjestelmät taas sisältävät kaiken sisällön tuotantoon.

Voidaankin todeta, että www-sisällönhallintajärjestelmä sisältää monimutkaisen yhdistelmän teknologioita, kokoelmia, prosesseja ja laadun tarkkailua ja kaikki vain sen takia, että niiden täytyy toimia yhdessä (Yu 2005, 2).

## 6.1 Kokoelma

Kokoelma (engl. Repository) on koko sisällönhallintajärjestelmän sydän. Tämän tallennuspaikan kautta kulkee kaikki raakamateriaali, joista julkaisut rakennetaan. Tallennuspaikan merkitys koko sisällönhallintaprosessissa on erittäin suuri. (Boiko 2005, 918.)



Kuvio 4 Korkeamman tason kokoelman rakennekuvauksesta, jossa näkyy kokoelman yksilöidyt osat ja sisällön varastointivaihtoehdot. (Boiko 2005, 919).

Kokoelma voidaan kuvitella tietokantana. Kuten tietokanta, niin kokoelmakin mahdollistaa käyttäjän tallentaa ja hakea sisältöä, mutta kokoelma on paljon muutakin. Ensimmäisenä merkittävänä erona tietokantaan, kokoelma voi hallinnoida useita eri tietokantoja. Kokoelmalla on liittymärajapinta toisiin järjestelmiin, joka tiedon haun ja käsittelyn suhteen on paljon monipuolisempi kuin yksittäiset tietokannat yleisesti ottaen. Jos katsotaan kokoelmaa yksittäisenä yksikkönä, kaikki se mitä tietokannoista tiedetään auttaa tässä ajattelussa ja helpottaa ymmärtämään kokoelman toiminnot. Useimpien kokoelmien kohdalla niiden ytimessä on tietokanta. Tietokannalla on näissä tapauksissa useimmiten kuitenkin paljon muokattua koodia ympärillä ja ne omaavat erityisen

käyttäjärajan, jolloin käyttäjät harvoin näkevät vilahdustakaan koko tietokannasta. (Boiko 2005, 918.)

Sisältökokoelmalla tulisi olla muutamia suositeltavia ja elintärkeitä ominaisuuksia. Ensimmäisenä sen tulee säilyttää sisältöä. Sisältökokoelmalla on oltava yksi keskeinen paikka johon sisältö sisällytetään (Boiko 2005, 925). Toisena ominaisuutena kokoelmaan tulee olla mahdollista syöttää dataa, mutta sen täytyy myös mahdollistaa käyttäjän etsiä näitä datatietoja. Kokoelman tulee myös antaa ulospäin sisältöä takaisin käyttäjälle sekä sen on mahdollistettava sisällön poisto. Lisäksi kaikkien hallintajärjestelmien tulee jollain tapaa segmentoida tietoa, jolloin kokoelman tulee tukea komponenttien konsepteja (Boiko 2005, 919-920). Hyvä järjestelmä helpottaa datan syöttöä, nimeämistä, paikantamista ja sisällön erottelua sen tyyppin mukaan, toisin sanoen rakenneosaluokan mukaan (Boiko 2005, 925). Sisältöä tulisi pystyä jäljittämään niin, että hallintajärjestelmä tuottaa tilastoja ja raportteja, jotka mahdollistavat arvioida sekä yksilöllisiä, että ryhmiteltyjen komponenttien tilaa. Kokoelman on tuettava myös työvirranhallinnan ajatusta. Näin ollen työvirranhallinta moduulit tulee olla hyvin tiukasti integroitu kiinni juuri kokoelmaan.

Toisaalta kokoelman tulisi myös tukea niin alkioden kuin kokonaisen tekstin etsintää, sekä massamenetelmiä. Komponenttien hallinnointi yksikerrallaan on liian hankalaa ja hidasta monissa tilanteissa. Hyvän hallintajärjestelmän tulisikin mahdollistaa määrittellä ja täsmentää operaatio ja sen olisi pystyttävä tekemään se yhä uudelleen ja uudelleen. Esimerkiksi dokumentteja tulisi pystyä hakemaan vanhenemispäivämäärien mukaan, ja niille tulisi pystyä määrittämään uusi vanhenemispäivämäärä yhdellä kertaa, ei siis suinkaan kymmelle dokumentille yksi kerrallaan.

Kaksi viimeisintä merkittävää sisältökokoelman ominaisuutta ovat organisaatiostandardien ja kenttätyyppien tukeminen. Sisältökokoelman tulisi toimia riippumatta siitä mitkä käyttäjäturva- ja verkkostandardit ovat käytössä yrityksessä. Esimerkiksi sisällönhallintajärjestelmien www-pohjaiset lomakkeet ja administraattori-työkalut eivät toimi lähiverkossa (engl. Local Area Network tai



LAN) ilman TCP/IP verkkoprotokollien käyttöä. Kenttätyyppien tukemisessa kokoelman tulisi metatietojen kohdalla mahdollistaa muidenkin kuin pelkän tekstitiedon syöttämisen. (Boiko 2005, 923.)

Sisällönhallintajärjestelmän tulisi pystyä ainakin metatietojen kohdalla erotella ja yhteen kasata metatietokentät. Toiseksi sisällönhallintajärjestelmässä tulisi pystyä seuraaman jonkinlaisen analyysityökalun avulla mitä kaikkea sisältöä tehdyt muutokset ja päivitykset koskevat. Kolmantena tärkeänä kohtana on tiedon tallennus elementeistä metadataan. Sisällönhallintajärjestelmästä riippuen saattaa olla vaikeaa tai työlästä selvittää mitä rakenneosia ja elementtejä kukakin käyttäjä missä ja milloin näkee. Tähän apuvälineeksi olisi hyvä ottaa pakolliset ja vapaaehtoiset elementit.

## 6.2 Työvirta

Aiemmissa kappaleissa on käsitelty työvirtaa (engl. workflow) ja sen hallintaa yhtenä merkityksellisempänä sisällönhallintajärjestelmän ominaisuutena. Työvirta on sarja askelia jotka toistuvat sisällönhallintajärjestelmässä tietyin väliajoin. Toisin sanoen työvirta on sykli. Työvirran takana oleva konsepti on sama kuin koko sisällönhallinnan konsepti: Tule organisoiduksi. (Boiko 2005, 757.) Vaikkakin työvirranhallinta on vain yksi osa koko sitä organisoitua järjestelmää jonka sisällönhallintajärjestelmät tarjoavat, on se yksi niistä tärkeimmistä ominaisuuksista jonka organisaatiot vaativat sisällönhallintajärjestelmältä, muttei kuitenkaan niin tärkeä kuin esimerkiksi sisältökokoelma. Työvirranhallinta ohjailee sisällönhallintajärjestelmän odotettuja operaatioita seuraavasti:

- *Prosessi: Työvirta tarjoaa hyvin rajoitetut prosessit sisällön liikkumiselle koko järjestelmän sisällä.*
- *Henkilöstö: Työvirta tarjoaa mahdollisuuden asettaa olettamuksia, joiden avulla voidaan luoda henkilöstö suunnitelmia ja kustannusarvioita.*

- *Askeleet: Työvirta paloittelee koko prosessin sarjaksi eri askeleita, joita ovat tehtävät joita joku tekee tietynä työnä sisällönhallintajärjestelmän esittämänä tehtävänä jossain sisällönhallintajärjestelmän osassa.*
- *Työvirran analyysit: Tämä analyysi sisältää kaikki ne tehtävät jotka sisällönhallintajärjestelmästä löytyy. Se organisoii ne prosesseiksi ja luo joukon määriteltyjä askelia jokaiseen prosessiin.*
- *Henkilöstö analyysit: Tämä analyysi sisältää ne määräävät tekijät työvirran tehtävien takana, ryhmittelee ne töiksi ja sen jälkeen tekee hienostuneita arvioita siitä kuinka paljon työaika kuluu tekijältä tämän tehtävän suorittamiseen. (Boiko 2005, 789.)*

Työvirta siis tarjoaa sisällönhallintajärjestelmän käyttäjille laadukkaan työkaluja organisoida tekijät ja tehtävät sekä niiden suoritusjärjestys. Työvirranhallintatyökalut antavat mahdollisuuden hallinnoida sisällön koko elinkaarta aina sen luonnista, muokkauksen ja julkaisun kautta sen poistoon.

### 6.3 Yhteydet muihin järjestelmiin

Sisällönhallinnan kokonaisuuden kannalta on tärkeää, että sisällönhallintajärjestelmä on mahdollisuus integroida erilaisiin infrastruktuureihin sekä tietojärjestelmiin. Tällaisia järjestelmiä ovat mm. seuraavat: organisaatioiden LAN (engl. Local Area Network) ja WAN (engl. Wide Area Network) ympäristöt, organisaatioiden käyttäjähallintajärjestelmät, yritysten metatieto- sekä tietojärjestelmät. Kytkemällä LAN- ja WAN-ympäristön sisällönhallintajärjestelmään voidaan organisaation sisällä vastaanottaa sekä lähettää sisältöä minne tahansa. Kytkemällä sisällönhallintajärjestelmän taas käyttäjähallintajärjestelmän piiriin, on sisällönhallintajärjestelmän mahdollisuus viitata suoraan yrityksen käyttäjerekisteriin ilman käyttäjätietojen uudelleen kirjoitusta. Esimerkiksi jos yrityksellä on globaali tuotekatalogijärjestelmä, joka ei ole osa sisällönhallintajärjestelmää. Tällöin liittämällä tällaisen metatietojärjestelmän sisällönhallintajärjestelmän piiriin, on mahdollista selata tuotetietoja niiden säilöntäpaikasta. Toisaalta paras ratkaisu tällaiseen metatietojärjestelmän

yhteyteen on kaksisuuntainen yhteys, jolloin myös tuotetietojen ylläpito onnistuu sisällönhallintajärjestelmän kautta. Yrityksien muiden tietojärjestelmien liittämiseen sisällönhallintajärjestelmän piiriin tarvitaan yleensä joku fyysinen yhteys, esimerkiksi LAN ympäristön avulla. Tällaisia tietojärjestelmiä ovat esimerkiksi ERP (engl. Enterprise Resource Planning) järjestelmät. (Boiko 2005, 105.)

#### 6.4 Hallintatyökalut ja administraatiojärjestelmä

Administraatiojärjestelmä on vastuussa parametrien ja rakenteen asettamisesta sisällönhallintajärjestelmälle ja se koskee kaikkia sisällönhallintajärjestelmän osia, seuraavaksi esitetyillä tavoilla. Kokoonpanojärjestelmässä hallinnointi sisältää henkilöstö konfiguraation, jossa roolit ja oikeudet jaetaan. Kokoonpanojärjestelmä hallinnoi myös järjestelmäkonfiguraatiota, johon sisällönhallinnan rakenne ja työvirta on sisällytetty, lisäksi kokoonpanojärjestelmä ylläpitää järjestelmän metatieto kenttiä ja listoja. Hallinnointijärjestelmässä administraattorit suorittavat tavallisia tietokannan hallinnointitehtäviä, kuten käyttäjän ylläpitoa ja käyttäjä-oikeuksia, sisällön varmistuksia ja arkistointia. Näiden kahden edellämainitun lisäksi hallinnointijärjestelmän administraattorit suorittavat sisältöön liittyviä erityistehtäviä, kuten luovat sisältötyyppejä, tekevät metatieto arvosteluja sekä kehittävät työvirtoja. Julkaisujärjestelmässä taas administraattorit varmentavat, että kaikki sisällön näyttämiseen tarvittavat laitteiston osat sekä sovellukset toimivat vaaditulla tavalla. Verkkosovellusten kohdalla administraattorin tehtäviin voi kuulua mm. www-serverien, sovellusserverien, sisällönhallinnan sovellusosien, tietokantojen ja muiden yhteen kuuluvien ohjelmien toimivat ja eivät koskaan ole ylikuormitettuja. (Boiko 2005, 103.)

## 7 WWW-SISÄLLÖNHALLINTAJÄRJESTELMÄESIMERKKEJÄ

Seuraavassa case-osiossa käsitellään vapaan lähdekoodin DotNetNuke-järjestelmää, mutta on hyvä ottaa huomioon muitakin markkinoilla olevia järjestelmiä, joista varmasti löytyy jokaisen käyttötarpeen huomioonottaen soveltuva paketti. Koska www-sisällönhallintaan on käytetty paljon aikaa ja rahaa viimeisien vuosikymmenien aikana, on sillä todennäköisesti kaikista kattavin valikoima sisällönhallintatuotteita.

### 7.1 OpenCMS

OpenCms on yksi vapaan lähdekoodin sisällönhallintajärjestelmien tuotteista, jonka juuret juontavat 2000 luvun vaihteeseen jolloin se lanseerattiin CeBIT expo 2000:ssa. OpenCms on pohjautunut sitä edeltäneeseen MhtCms:ään, joka ei kuitenkaan ollut vapaan lähdekoodin tuote. OpenCms perustuu puhtaasti Java- ja XML- teknologialle ja sen on tuottanut Alkacon Software LGPL-lisenssillä (engl. LGPL License tai GNU Lesser General Public License). LGPL-lisenssin on tarkoitus tehdä kompromissi kopiosuojan omaavan GPL:n (engl. GNU General Public License) ja muiden yksinkertaisien lisenssien (BSD, MIT, jne.) välillä. Pääsääntöisesti GPL ja LGPL erovat siinä, että LGPL-ohjelmistot voidaan linkittää yhteen ei-GPL-lisensoidun ohjelman osan kanssa. Tämä mahdollistaa esimerkiksi sen, ettei kaupallisten ohjelmistojen tarvitse julkaista lähdekoodia, vaikka ohjelma käyttäisikin LGPL-lisensoituja komponentteja. LGPL:ää on käytetty ensisijaisesti ohjelmisto kirjastoissa sekä yksittäisissä sovelluksissa, joista merkittävimpiä ovat Mozilla ja OpenOffice.org.

OpenCms on siis vapaan lähdekoodin järjestelmä, joka tarkoittaa sitä, että käyttäjä pääsee katsomaan sekä muokkaan lähdekoodia ja järjestelmä on myös maksuton. OpenCms toimii kaikissa vapaan lähteen ympäristöissä käyttäen Linuxia järjestelmälle, MySQL:ää tietokantana, Apachea www-serverinä sekä Tomcat:iä www-sovellus serverinä. Sisällönhallintajärjestelmänä sen perusominaisuuksiin

kuuluu mm. selainpohjainen työympäristö, projekteihin perustuva julkaisu, työvirran- ja tehtävähallinta, sisällön versiointi, dynaaminen ja staattinen sisällön julkaisu, sovellusserverin integraatiot ja paljon muuta. OpenCms:ssä merkittävintä todennäköisesti on se, että se käyttää tiedosto- ja kansiovertausta sisällönhallintaan kun taas linkkeihin perustuvaa vertausta. OpenCms luo virtuaalitiedostojärjestelmän (engl. Virtual File System tai VSF) tietokannan tiedoista, joka taas mahdollistaa helpon täysikokoisten sivujen selaamisen ja katselun.

## 7.2 Joomla

Toinen vapaan lähdekoodin sisällönhallintajärjestelmien esimerkeistä on Joomla. Joomla on tarkoitettu lähinnä intranettien ja www-sivustojen julkaisemiseen. Se on koodattu käyttäen PHP-ohjelmointikieltä ja MySQL-tietokantaa tietokantapohjana. Joomla'n historia juontaa vuoteen 2005 jolloin se haarautui Mambo-projektista erilliseksi vapaan lähdekoodin tuotteeksi ja sai oman brändinsä. Jo alkumetreillä Joomla'n kehittäjät halusivat uudelleen kirjoittaa koko koodipohjan ja tällä hetkellä (helmikuu 2008) siitä on saatavissa kolmas versio. Käyttäjät odottavat kokonaisuudessaan uuden version valmistuvan seuraavien vuosien kuluessa. Mainittavinta koko Joomla:ssa lienee sen vuonna 2006 ja 2007 voittamat Packt Publishingin myöntämät vapaan lähdekoodin sisällönhallintajärjestelmien palkinnot.

Joomla kuuluu siis osaksi kaikkia markkinoilla olevia vapaan lähdekoodin tuotteita. Se mikä saa sen jokseenkin erottumaan muista tuotteista on Joomla-paketin sisältämät monet eri osat, jotka on rakennettu mahdollisimman kokoonpantaviksi kuin on vaan pystytty. Kokoonpantavuus täten mahdollistaa lisäosien ja integraatioiden helpon toteutuksen. Joomla:ssa lisäosat ovat tavallaan taustalaajennuksia, jotka laajentavat Joomlaa uusilla toiminnoilla. Yhtenä esimerkkinä sanottakoon WikiBot, joka mahdollistaa tekijän käyttäen Wikitageja (engl. Wikitags) artikkeleissa, jotka luovat automaattisesti dynaamisia hyperlinkkejä Wikipedian artikkeleihin. Tietävästi Joomlaan on tällä hetkellä

saatavilla reilusti yli 2300 laajennusosaa (helmikuu 2008), joita OpenSourceMatters-yhteisö pitää yllä virallisessa laajennusosahakemistossa.

### 7.3 RedDot CMS

RedDot CMS on lähinnä suuremmille yrityksille tarkoitettu sisällönhallinta-järjestelmä. RedDot on Open Text Corporationin toimintayksikkö, jonka päätuote RedDot CMS on. RedDotilta löytyy tuotteita myös perinteiseen dokumentin-hallintaan, sekä heidän LiveServerinsä, joka on suunniteltu maksimoimaan sekä laajentamaan yrityksen sisällönhallintaresursseja.

RedDot CMS on Windows-pohjainen serverisovellus jolla on mahdollista hallita www-sisältöä useita käyttäjiä omaavassa ympäristössä. RedDot CMS:ssä käytetään MS SQL-serveriä ja Oraclen tietokantaa sisällönhallinnan apuna. RedDot CMS mahdollistaa myös LiveServerinsä avulla yhdistää erillisiä dokumentti-resursseja liittimien kautta ja tarjoilla ne www-sivuilla. Kyseinen LiveServer mahdollistaa moniportaaliset ratkaisut, sillä voidaan ylläpitää niin intra-, extra- kuin myös tavallisia yrityksen www-sivuja. RedDot CMS on yhdistelmä monipuolisia moduuleita ja resurssienhallintatyökaluja, mutta tarjoaa sinänsä kaikki sisällönhallintajärjestelmille ominaiset työkalut monikielisyystukineen.

### 7.4 Hot Banana: Web Content Management Suite

Hot Banana on itsenäisessä omistuksessa oleva Lyris Incin tytäryhtiö joka on perustettu vuonna 1999. HotBananan www-sisällönhallintatuote Web Content Management Suite (WCM Suite) on yrityksen käyttöön tarkoitettu tuote, jota on saatavilla joko ylläpidettynä (engl. Hosted CMS/SaaS tai Software-as-a-Service) tuotteena tai lisenssoituna ohjelmistona (engl. Licensed Software/ Coldfusion CMS). WCM Suite tarjoaa monipuoliset hakukoneoptimoinnit (engl. SEO tai Search Engine Optimization) ja SSL (engl. Secure Sockets Layer) sivulta sivulle salausmenetelmät (engl. SSL Page-by-Page management). HotBananan WCM Suite on myös muilta osiltaan hyvin monipuolinen järjestelmä, josta löytyy niin

kattavat administraatio-ominaisuudet kuin työvirranhallinta, se tukee myös RSS-syötteitä ja niiden hallintaa.

HotBananan WCM Suitella on mahdollista ylläpitää niin intra-, extra- kuin myös pienoismarkkinointisivustoja. WCM Suite on rakennettu niin sanotusti puoli-avoimella arkkitehtuurilla jonka sovellusohjelmointirajapinta (engl. Application Programming Interface tai API) mahdollistaa kolmannenosapuolen toimittajien ohjelmistojen ja moduulien helpon integraation ja mukauttamisen. HotBanana:n WCM Suiten voidaan sanoa olevan kypsä ja vakaa www-sovellus, joka on rakennettu Macromedian Cold Fusion MX Serverille. Se voi käyttää hyväkseen kaikki J2EE sovellusserverin ominaisuudet, mutta WCM Suite on mahdollista yhteensovittaa myös IBM:n WebSphere:n tai Macromedian JRUN:in kanssa. Microsoftin SQL Server on HotBanana:n pääasiallinen tietokantakoneisto, mutta kaiken kaikkiaan se on .NET-, XML- ja Web-palvelu valmis.

Edellä kuvattujen www-sisällönhallintajärjestelmien lisäksi markkinoilla on saatavilla tuotteita ja palveluntarjoajia laajasta hintahaarukasta. Tarjolla on vapaan lähdekoodintuotteista aina suuryritysten sisällönhallintaongelmien haasteisiin vastaavin tuotteisiin, joissa hinta nousee aina 75 000:een euroon asti. Vaikka markkinoilla on periaatteessa ilmaisia vapaan lähdekoodintuotteita sisällönhallintajärjestelmää tarvitsevalle ei välttämättä ole resursseja ylläpitää tai konfiguroida sitä omien tarpeidensa mukaiseksi. Tämän ovat monet pienet IT-yritykset huomanneet käyttää hyväkseen. He tarjoavat asiakkailleen vapaan lähdekoodintuotetta, jonka he valjastavat yrityksen tarpeen mukaiseksi ja velottavat siitä tehdyn työn. Aivan ilmaisista tuotteista ei siis voida puhua.

Samanlailla kuin hintahaarukasta puhuttaessa voidaan todeta käytetyistä ohjelmointiresursseista. Www-sisällönhallintajärjestelmissä on käytetty todennäköisesti kaikki mahdolliset resurssit ja ohjelmointikielet hyväksi mitä World Wide Webin kehittäjinä mukana olleet ovat aikojen saatossa tuottaneet,

tässä voidaan niistä mainita vielä ainakin Python- ja Perl-ohjelmointikielet sekä PostgreSQL tietokanta.

## 8 CASE: DOTNETNUKE

Tässä luvussa keskitytään tutkimaan järjestelmä DotNetNukea, myöhempänä DNN. DNN:sta käydään läpi hieman sen kehitystä sekä sen avain- ja perusominaisuuksia. Tutkimuksen tavoitteena oli selvittää DNN:n vastaavuus sisällönhallinnan yleisiin peruseräisiin, joita aiemmissa luvuissa on käsitelty. Eli voidaanko DNN:sta käyttää termiä www- sisällönhallintajärjestelmä, täyttääkö se ne ominaisuudet jotka sisällönhallintajärjestelmille on yleisesti asetettu. Tutkimus suoritettiin virallisen DNN yhteisön www –sivuston foorumilla, jossa aktiivisia ja rekisteröityneitä käyttäjiä on reilut 400 000 (maalis-huhtikuu 2007). Vastauksia tuli niukasti, mutta sitäkin osaavimmilta tahoilta. Ja näin ollen tiedon tarkkuus ja luotettavuus on hyvä. Kysymyksiin saatiin vastauksia niin DNN:n kehitystiimiltä kuin henkilöiltä joilla on vuosien kokemus työskentelystä DNN:n parissa.

### 8.1 DNN:n evoluutio

DNN on DotNetNuke Corporationin omistama projekti ja perustuu vapaaseen lähdekoodiin, tarkoittaen siis sitä, että se on ladattavana internetistä ilman käyttöoikeusmaksuja ja jokaisella käyttäjällä on oikeus muokata sen lähdekoodia haluamallaan tavalla. DNN on alun perin IBuySpy Workshop- projektin lopputulos. Sen kehittänyt Shaun Walker loi DNN:n IBuySpy portaalin laajennusosaksi. IBuySpy portaali on Microsoftin kehittämä esimerkki sovellus .NET –puiteohjelmalle. Marraskuussa 2004 Walkerin yritys, Perpetual Motion Interactive Systems Inc sai luvan lanseerata tuotemerkin nimeltä DotNetNuke, jolloin DNN:n tuotekehitys pääsi kunnolla käyntiin

VB.NET:illä ASP.NET –puiteohjelmalle koodattu DNN on laajennettavissa ja räätälöivissä jokaisen käyttäjän tarpeiden mukaan. DNN:een on saatavissa sekä



maksutta, että lisämaksulla erilaisia ulkoasuja ja moduuleja, joiden avulla voidaan helposti tuottaa ja ylläpitää erilaisia www -, intra- sekä extranet –sivustoja.

DNN on kehittynyt paljon muutaman vuoden sisällä. Kevästä 2004 lähtien, ohjelmasta on julkaistu yhteensä 23 versiota, lokakuuhun 2006 mennessä. Maaliskuussa 2007 DotNetNukesta oli saatavissa jo versio 4.4.1, joka vaatii käyttäjältä 2.0 ASP.NET –puiteohjelman toimiakseen. Mutta saatavissa on silti vielä myös 4.0 aikaisempia versioita, joiden käyttö hoituu vanhalla ASP.NET 1.1 –puiteohjelmalla.

DNN on saavuttanut merkittävää suosiota vapaan lähdekoodin järjestelmien keskuudessa muutaman vuoden sisällä ja voidaan sanoa, että se on yksi paljon käytetyistä järjestelmistä. DNN:n suosio perustuu lienee sen monikäyttöisyyteen ja käyttäjäystävällisyyteen sekä siihen, että DNN tukee moniportaalisia sekä -sivustoisia internetsivustoja, vain yhdellä järjestelmän asennuksella. Tätä koko olettamusta tukee se, että maaliskuussa 2007 DNN –yhteisön virallisen www-sivuston (<http://www.dotnetnuke.com>) rekisteröityneitä käyttäjiä on jo reilut 400 000 kappaletta, sekä myös se, että se on tunnetusti myös yksi jatkuvasti kehittyvistä järjestelmistä.

## 8.2 DNN:n filosofia

DNN projektin tarkoituksena oli luoda vapaan lähdekoodin tuote, joka edesauttaa ihmisten myötävaikutusta ja tiedon välitystä. DNN:n projektin omistajien mielestä on tärkeää antaa enemmän arvostusta käyttäjille ja kohdella heitä kuin kanssakehittäjiä, jolloin saavutetaan nopeammin kehittyvä koodin parannus ja virheiden poisto. Asiakkaiden kuuntelu on DNN projektin kannalta tärkeää. Painopiste pitää projektin omistajien mielestä kohdistaa ylemmän tason liike-elämän vaatimukseen ymmärtämällä ja toteuttamalla ne, ennemmin kuin luoda täysin täydellistä teknistä arkkitehtuuria. Lisäksi projektin omistajat ovat sitä mieltä, että jokaisella ohjelmistolla on mahdollisuus kehittyä mittaamattomasti jos resurssit on keskittetty oikeanlaisesti. Toiminnallisuus tulisi lisätä lisääviin

yksikköihin ennemmin kuin yrittää monoliittistä totetusta, sekä suurin osa käytetyistä voimavaroista tulisi sijoittaa ominaisuuksien toteuttamiseen, jolla on suurin hyöty ja laajin kulutus yhteisössä. Koko perusideana vapaan lähdekoodin DNN projektilla on se, että kun ohjelmoijat pystyvät lukemaan, uudelleenjakamaan ja muokkaamaan lähdekoodia, tällöin koko sovellus kehittyy. DNN on tarkoitettu käyttäjien kehitettäväksi, omaksuttavaksi ja se mahdollistaa myös virheiden korjauksen. (Walker, ym 2005, 14.)

### 8.3 Avainominaisuuksia

Virallinen DotNetNuke sivusto (<http://www.dotnetnuke.com>) pitää DNN:n avainominaisuuksina mm. sen asennuksen ja ylläpidon helpoutta, laajennettavuutta ja skaalattavuutta, sekä sen suorituskykyisyyttä ja hallinnoitavuutta. Sama sivusto kertoo DNN:n olevan täysin räätälöitävissä oleva sekä omaavan käyttäjäystävällisen rajapinnan.

DNN:n asennus siis hoituu lataamalla DNN:n viralliselta DNN –sivustolta (<http://www.dotnetnuke.com>). Tämä vaatii kuitenkin ensin rekisteröitymisen sivustolle. Yhden asennuksen jälkeen DNN:n avulla voidaan tarvittaessa luoda niin monta sivustoa/portaalia kun yrityksen tarve sitä vaatii. DNN:n ylläpito ei vaadi käyttäjältä juuri minkäänlaisia www-ohjelmointitaitoja, vaan periaattena on, että kuka tahansa voi halutessaan luoda omaperäisiä ja dynaamisia www-portaaleja.

DNN soveltuu monenlaisiin projekteihin, pienistä www-sivuista aina isojen organisaatioiden koko verkkopalveluiden hallintaan. Internet- ja intranetsivustot voidaan luoda pelkkien sisäänrakennettujen ominaisuuksien avulla tai laajentaa kolmansienosapuolien tarjoamilla lisäosilla. DNN tukee myös moniportaali-ratkaisuja. Eli yhdellä asennuksella voidaan hallinnoida useita eri www – portaaleja, koska DNN tarjoaa hallinnollisella tasolla liikkua niin host- kuin yksittäisen portaalin tasolla. Administraattoreilla on siis mahdollisuus hallinnoida

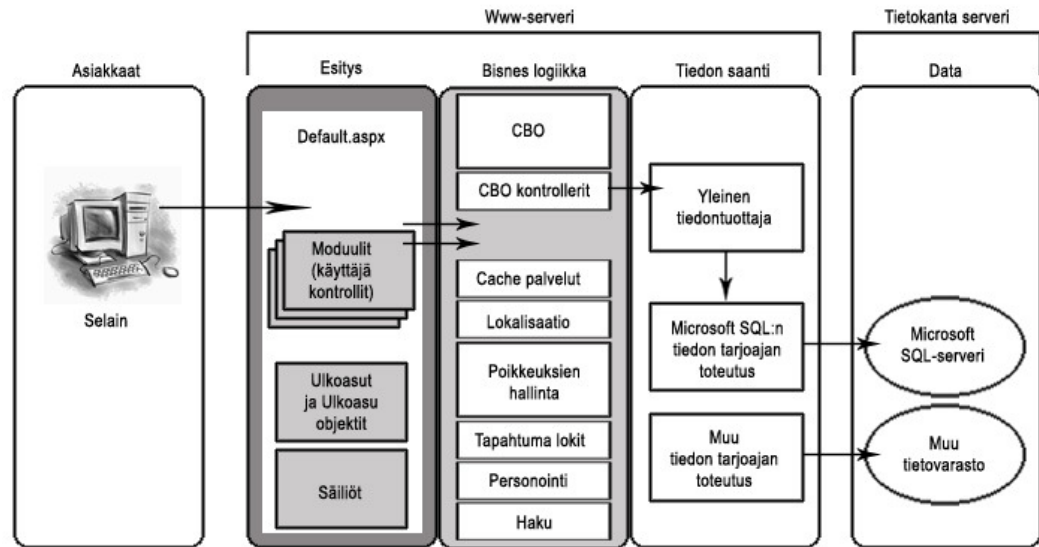
jokaista sivustoa erikseen, myös sen yksilöllistä ulkoasua ja identiteettiä, vain yhden ylläpitotilin avulla.

Käyttäjystävällisyyden perustana on tutkittu rajapinta, joka mahdollistaa käyttäjälleen hallinnoida kaikkia projektinsa osa-alueita. Sivustovelhot ja ohjeistus, sekä intuitiivinen rajapinta mahdollistavat DNN:lle yleisen käyttömukavuuden. DNN:n räätälöitävyys jokaisen omaan käyttöön lähtee ihan perusasioista. Mahdolliset muutokset voidaan tehdä portaalin jokaisella tasolla. Lähtien aina yksittäisistä ulkoasuaelementeistä, kuten kirjasimien tyylit ja värit, koko sivuston kattaviin tyylisivuihin. (Walker, ym 2005, 55-64.)

#### 8.4 DNN:n arkkitehtuuri

DNN:n arkkitehtuuri perustuu puhtaasti Microsoftin .NET arkkitehtuurille. Käytössä näistä on niin Visual Basic.NET kuin myös ASP.NET, jotka ovat Microsoftin kehittämiä webteknologioita. Serverien puolesta DNN käyttää niin Microsoftin Windows servereitä ja Microsoftin SQL serveriä. Käytössä ovat myös Microsoftin IIS palvelut (engl. Internet Information Services) ja www-lomakkeet (engl. Web Forms).

DNN:n arkkitehtuuri antaa sovelluksen jakautua kahden eri serverin välille: www-serverin ja tietokantaserverin. Www-serveri sisältää niin esitys- eli presentaatio ja bisneslogiikkakerrokset, kuin myös pääsyn käsiksi dataan eli tietoihin. Tietokantaserveri sisältää vain datakerroksen.



Kuvio 5 Mukailtu kuva DotNetNuken arkkitehtuurista, joka esittelee serverit kerroksineen (Walker, ym 2005, 195).

#### 8.4.1 Presentaatiokerros

Presentaatiokerros tuottaa liittymärajapinnan asiakkaiden pääsulle portaalin sovelluksiin. Tämä kerron sisältää mm. seuraavat elementit: 1) www-lomakkeet, joiden ensisijainen lomake on Default.aspx, jolta on pääsy kaikkialle portaalin uumeniin. Default.aspx:n sanotaan olevan juurihakemisto. 2) Sivumallit, jotka Default.aspx lataa jokaiselle sivulle portaalissa, perustuen jokaisen sivun yksittäisiin asetuksiin. 3) Säiliöt (engl. Container), jotka Default.aspx lataa, jokaiseen moduuliin perustuen jokaisen moduulin, sivun ja portaalin yksittäisiin asetuksiin. 4) Moduulien käyttäjäkontrollit (engl. Module User Controls), joita jokaisella moduulilla tulee olla ainakin yksi kappale, joka luo siis käyttäjärajapinnan kyseiselle moduulille. Default.aspx lataa myös käyttäjä-kontrollit jotka ovat säiliöiden ja sivumallien sisällä. Viimeisenä elementtinä presentaatiokerroksessa on 5) Asiakaspuolen skriptit (engl. Client-Side Scripts).

Asiakaspuolen JavaScript tiedostoja käytetään käyttäjärajapinnan puiteohjelman ytimessä. Esimerkiksi sivumallit saattavat käyttää tällaisia asiakaspuolen skriptejä. (Walker, ym 2005, 196.)

#### 8.4.2 Bisneslogiikkakerros

Bisneslogiikka kerros tuottaa logiikan koko ydinportaalin tapahtumille. Tämä kerros paljastaa monia ytimen palveluita ja kolmannenosapuolentoimittajien moduuleita, kuten lokalisaatiot, cache palvelut, poikkeuksien hallinta, tapahtumaloki, personointi, asennus ja päivitys, turvallisuus sekä myös haku-palvelut. Kyseinen kerros on tavallaan koti räätälöidyille bisnesobjekteille, jotka edustavat kokonaisuutta eli kollektiivisesti ovat koko portaali. Ensiarvoinen tarkoitus bisneslogiikkakerroksen räätälöidyillä objekteilla on tiedontallennus objekteista. (Walker, ym 2005, 198.)

#### 8.4.3 Tiedonsaantikerros

Tiedonsaantikerros tuottaa tietopalvelut bisneslogiikka kerrokselle. Tämä kerros siis mahdollistaa tiedon virtauksen sisään ja ulos tietojen tallennuspaikasta. Tiedonsaantikerros kokonaisuudessaan sisältää kaksi elementtiä. 1) Tiedontuottajanohjelmointirajapinta (engl Data Provider API), on siis abstrakti perusluokka joka asettaa sopimukset jotka API:n toteutuksen täytyy täyttää. 2) Tiedontuottajan ohjelmointirajapinnan toteutus (engl. Implementation of Data Provider API), on luokka joka periytyy tiedontuottajanohjelmointirajapinnalta ja täyttää asetetut sopimukset syrjäyttämällä tarvittavat jäsenet ja metodit. (Walker, ym 2005, 198.)

#### 8.4.4 Datakerros

Datakerros tuottaa tiedon tiedonsaantikerrokselle. Käytössä olevan tiedontallennuspaikan tulee tukea tiedontuottajan ohjelmointirajapinnan toteutusta täyttääkseen datakyselyt. Koska DNN:n tiedontuottajamalli on luotu hyvin joustavaksi, useita tiedontuottajia on saatavilla. Nämä sisältävät sekä alkuperäisen tiedontuottajat sekä kolmannenosapuolen toimittajien tuottamat tiedontuottajat, sisältäen niin ollen Microsoft SQL-serverin, Microsoft Accessin, mySQL:n ja Oraclen tuottajat. Alkuperäinen DNN julkaisu sisältää Microsoft SQL-serveri toteutuksen tiedontuottajanohjelmointirajapinnalle. (Walker, ym 2005, 199.)

## 8.5 Sovellusohjelmointirajapinta

Ydinsovellusohjelmointirajapinta on se todellinen voima DNN:n takana. Vaikuttamalla ja muuttamalla yleisiä ohjelmointirajapinnan sovelluksia DNN:a voidaan laajentaa mihin suuntaan hyvänsä. Toisinsanoen sovellusohjelmointirajapinta mahdollistaa laajentaa DNN:a hyvinkin monipuolisesti, lisäämällä esimerkiksi räätälöityjä moduuleita tai vaihtamalla ydin funktioita. Sovellusohjelmointirajapinnassa on valmiiksi monipuolisia ominaisuuksia joita laajentamalla tai muuttamalla voidaan koko DNN muokata jokaisen henkilökohtaistentarpeiden mukaisiksi. (Walker, ym 2005, 207.)

Kyseisiä ydinominaisuuksia ovat mm. 1) Tapahtumaloki (engl. Event Logging), joka DNN:ssa tuottaa laajan kategorian rekisteröinti palveluita, kuten poikkeus- ja turvallisuusloki sekä tapahtuma-auditointi (Walker, ym 2005, 208). 2) Poikkeustapausten (engl. Exception Handling) hallinta DNN:ssa käsitellään neljän eri metodin kautta, joiden avulla voidaan käsitellä poikkeuksia yhdenmukaisesti, kirjata poikkeustapausten jäljitys ja viitekehys, sekä välittää loppukäyttäjille käyttäjäystävällisiä viestejä (Walker, ym 2005, 216). 3) Paikallistamisessa (engl. Localization) DNN:n sovellusohjelmointirajapintat tarjoaa staattiset paikallistamisominaisuudet joilla käyttäjärajapinta voidaan muokata monikieliseksi räätälöidyissä portaalin moduuleissa. Staattisilla ominaisuuksilla tarkoitetaan sitä, että vain tekstiotsikot ja muut staattiset merkkijonot on paikallistettu ytimessä, joten dynaamisen sisällön paikallistaminen on moduulin luoja käsissä (Walker, ym 2005, 221). Paikallistamisen toetutuksessa DNN käyttää Windowsin resurssitiedostoja (RESX) muotoillakseen tallennetut käännökset (Walker, ym 2005, 222). 4) DNN:n oleva ajoitin (engl. Scheduler) on tarkoitettu kehittäjille tehtävien ajoittamiseen tietyn väliajoin. Ajoitin on myös vaihdettavissa tai muunneltavissa ilman että tarvitsee koskea ydinkoodiin. 5) DNN tarjoaa moninaisen kirjon HTTP-moduuleita laajentamaan tapahtumien peräkkäis-suoritusta asiakkaan ja serverin välillä. HTTP-moduuli mahdollistaa ennen ja jälkeen prosessoinnin jokaiseen tulevaan HTTP-pyyntöön (Walker, ym 2005, 232). 6) Moduulit (engl. Modules) näyttelevät DNN:ssa erillisessä asemassa toiminnallisuutta, joilla on mahdollisuus laajentaa portaalin runkoa (Walker, ym

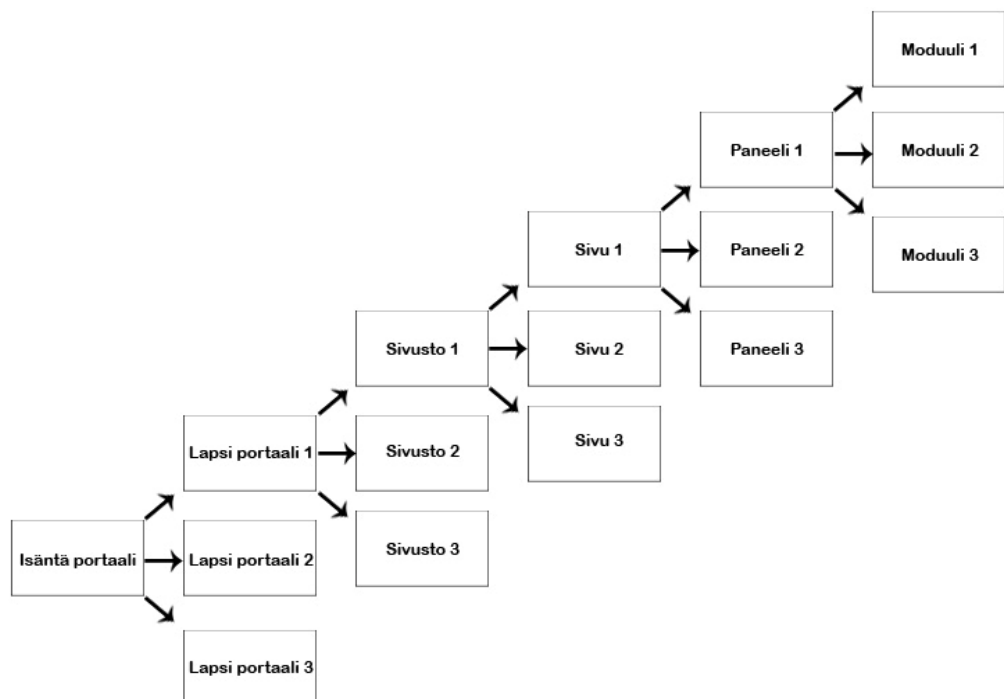
2005, 240). Kehityksen myötä DNN:n moduulit ovat enemmän ja enemmän toiminnallisuutta lisääviä osia, jotka toimivat yksittäisinä ilman että niiden tarvitseen kommunikoida ydinkoodin kanssa.

## 8.6 Sisällönhallinta ja rakenne

DNN:ssa sisällönhallinta perustuu hyvin pitkälti sen rakenteeseen. Järjestelmän rakenne luo pohjan toiminnallisuudelle ja sen perusteille sisällönhallintaan.

DNN:n kehittäjät haluavat käyttää termiä portaali, tehdäkseen eron jokaisen sen sisältämän sivuston kohdalla. Verkkosovellusportaali voidaan määritellä seuraavalla lauseella: ”Verkkosivusto, joka voidaan käsittää merkintäpisteenä muille verkkosivustoille sekä joka tarjoaa pääsyn hakukoneelle (engl. Search engine)”. DNN:ssa tämä on kuitenkin myös paljon muutakin kuin vain avoin ovi muille sovelluksille ja hakukoneille. (Walker, ym 2005, 55.)

Lähtökohtana on siis katsoa DNN:aa kokonaisuutena, joka lähtee yhdestä portaalista ja jakautuu yhdeksi tai useammaksi pienemmäksi portaaliksi. Portaalin administraattorin on mahdollisuus luoda yhteen portaaliiin useita satoja eri verkkosivustoja.



Kuvio 6 DNN rakenne

Kuvio 6 kuvaa yksinkertaisuudessaan DNN:n koko rakenteen, johon sen sisällönhallinta perustuu. Lähtökohta on siis yksi isäntäportaali, joka jakautuu useampiin pienempiin lapsiportaaleihin. Nämä lapsiportaalit taas jakautuvat useisiin sivustoihin, ja sivustot sivuihin. DNN- sivu kannattaa ajatella staattisena HTML-sivuna (Walker, ym 2005, 57). Erona tavalliseen HTML-sivuun on se, että sovellus lataa sisältö pohjan parametrien mukaan, jotka sille on asetettu lataavan ajonaikana (Walker, ym 2005, 57). Sivut sisältävät paneeleja, joihin tulevat moduulit liitetään. Paneelit mahdollistaa käyttäjän järjestellä sivun sisällön käyttäjäystävällisesti. Moduuleita voidaan kutsua DNN:n sydämeksi. Juuri moduulit ovat ne DNN:n komponentit, jolla DNN tarjoaa tarkoituksensa mukaisen ratkaisun näyttää relatiivinen ja helposti päivitettävä sisältö vierailijoille. (Walker, ym 2005, 60.) Perusmoduuleita tulee DNN:n asennuksen mukana useita, eri tarkoituksiin tarkoitettuja. Vaihtoehtoisia lisämoduuleita on saatavilla useilta toimittajilta. Vaikkakin yksi mahdollisuus on luoda niitä itse.

## 8.7 Hallinnointi

DNN tarjoaa kaksi erillistä tasoa hallinnoida. Yksittäistä portaalia hallinnoidaan 1) portaalin administraatiotyökalujen avulla ja 2) isäntätilin tai superkäyttäjän työkalujen avulla taas voidaan hallinnoida useita eri portaaleja kokonaisuudessaan tai yksitellen. Käytännössä katsoen portaalin administraattorina pääseen hallinnoimaan sivuston asetuksista ja käyttäjätileistä lähtien aina yksittäisiin tiedostoihin ja roskakoriin. Kun taas superkäyttäjä hallinnoi isäntäasetuksia ja portaaleja, SQL:ää, portaalin administraattoreita ja muita superkäyttäjätilejä. Superkäyttäjä kuitenkin pääsee myös yksittäisen portaalin tietoihin käsiksi ja pystyy muokkaamaan sivuston asetuksia ja malleja sekä lokitietoja.

## 9 TUTKIMUSTULOKSET

Tutkimuksen tarkoituksena oli siis selvittää, mitä DNN:n käyttäjät ovat mieltä sen sopivuudesta verrattaessa sitä sisällönhallintajärjestelmään. Vastaako DNN ominaisuuksiltaan asetettuja vaatimuksia sisällönhallintajärjestelmälle.



Tutkimuksen viideksi tärkeimmäksi osa-alueeksi valittiin sisältökokoelman liittymäraja-pinta (engl. repository interface), työvirranhallinta (engl. workflow management), yhteydet muihin järjestelmiin, hallintatyökalut (engl. administration tools) ja moduulit ja laajennettavuus. Tutkimus itsessään toteutettiin englannin kielellä, koska DNN-foorumi tätä edellytti, sekä siksi, että DNN-sivusto on kansainvälinen ja näin ollen kysymykset olisivat kaikkien ymmärrettävissä. Tutkimuksen kysymykset löytyvät liiteosiesta (Liite1).

Verrattaessa DNN:a sisällönhallintajärjestelmään kaikki vastaajat totesivat heti alkuun, ettei DNN:a voida kutsua aidoksi sisällönhallintajärjestelmäksi, vaikka sitä toki sellaisenakin on mahdollisuus käyttää, mutta parempiakin www-sisällönhallintajärjestelmiä on markkinoilla. Vastaajat kutsuivat DNN:a joko sovelluskehikseksi (engl. Application Framework) tai verkkohallintajärjestelmäksi (engl. Web Management System tai WMS). DNN:n ensimmäinen ongelma verrattaessa sitä sisällönhallintajärjestelmään on sen suunnittelun puutteellisuus, koska DNN on alun perin suunniteltu kehykseksi, jota jokainen käyttäjä voi muokata omien tarpeidensa mukaan. Koska DNN:n käyttö perustuu kutakuinkin juuri moduulien käyttöön, vastaajat totesivat, että DNN:lle on suunniteltu joitakin moduuleja sisällönhallinnanlaajennuksiksi.

Vastaajat totesivat ettei DNN:ssa ole suoranaista kehykseen rakennettua työvirranhallintaa. Totta on kuitenkin se, että DNN:n moduuleissa on mahdollisuus ajastaa sisältöön liittyviä tehtäviä, mutta DNN:ssa ei ole varsinaista sisältöön liittyvää raportointijärjestelmää, kuten todellisessa sisällönhallintajärjestelmässä tulisi olla. Koska todellisen sisällönhallintajärjestelmän tulisi antaa tietoa myös takaisin käyttäjälleen, tätä DNN ei siis tue.

Kun oikeassa sisällönhallintajärjestelmässä työvirranhallinta kulkisi sivun suunnittelijalta sivun hyväksymisen kautta julkaisuun automatisoidusti, DNN:ssa on tällä hetkellä vain yksi mahdollisuus: piilota sivu ennen hyväksyntää. Sisällönhallintajärjestelmän tulisi hallinnoida sivua sekä ilmoittaa sen statuksesta kaikille tämän sivun kanssa tekemisissä oleville henkilöille. Tällä tavalla työvirranhallinta resurssit mahdollistavat sivun laatijan olevan enemmän

yhteydessä sivun hallintaan, samalla ylläpitää tarkkaa kontrollia informaation laadusta ja tarkkuudesta sekä johdonmukaisuudesta.

DNN:ssa ei ole mitään tällaista sisällön tarkkailua. On kuitenkin olemassa joitakin moduuleja joihin on sisäänrakennettu tarkastuskontrollit, mutta nämäkään eivät tarkasta koko sivuston sisältöä vaan pelkästään sisällön moduulissa, näin ollen sitä paremmin kuvastaa sana dokumentinhallinta (engl. Document management).

Yhteyksistä muihin järjestelmiin vastaajat totesivat, että DNN-sovelluskehys on suunniteltu kommunikoidaan moduulien kanssa sekä hyväksymään kommunikointi moduuleilta takaisin päin, mutta kommunikointi muiden ei DNN-järjestelmien kanssa on mahdollista vain muokattaessa lähdekoodia. Mahdollista on, että esim. Microsoft Wordistä voit leikata ja liimata suoraan DNN-moduuleihin sisältöä, mutta suoraa kommunikaatiota Wordistä DNN:een ei ole sovelluskehukseen rakennettu.

DNN:ssa on mahdollisuus hallintatyökalujen avulla rajoittaa hyvinkin paljon sisällön näkyvyyttä selaajille, mutta DNN:ssa on kuitenkin vain kaksi eri hallinnointi tasoa administraattoreille. Hallintatyökalujakin on kuitenkin mahdollisuus laajentaa, eri moduulien kautta, mutta siis sovelluskehyksessä itsessään tällaista mahdollisuutta ei ole rakennettu. Vastaajat totesivatkin, että DNN saattaa näyttää ulkoapäin katsottuna jonkinlaiselta sisällönhallintajärjestelmältä, mutta administraattorien kannalta se ei sitä todellakaan ole.

DNN:a on paljon muokattu tulemaan lähemmäs sisällönhallintajärjestelmän perusideaa, mutta alkujaankaan se ei sisällönhallintajärjestelmäksi ole koskaan suunniteltu. Kuten aiemmin jo todettiin vastaajat kutsuivat DNN:a lähinnä verkkohallintajärjestelmäksi, joka toki sisältää kaikki toiminnot jotka kuuluvat sisällönhallinnan piiriin, ilman datan tarkastusta ja tallennusta.

## 9.1 Puutteet

Tutkimustuloksista kävi ilmi, että DNN:sta puuttuvat siis ne ensisijaiset sisällönhallintajärjestelmältä vaadittavat toiminnot, kuten sivujen muutosten ja versioiden rekisterinpito, käyttäjien varmenteet sekä muiden informaatiolähteiden ja järjestelmien integraatiot. Sekä tietysti koko sisältökokoelma, jota tällä hetkellä hoitavat DNN:ssa vain moduulit.

Vastaajat totesivat, että DNN:a voidaan kutsua sisällönhallintajärjestelmäksi parhaiten juuri silloin, kun käytössä ei ole muuta sisällönhallintajärjestelmää. Mutta parhaiten sitä kuvaa sana verkkohallintajärjestelmä tai sovelluskehys. Vaikka DNN on kehittymässä jokaisen uuden version myötä enemmän ja enemmän vastaamaan sisällönhallintajärjestelmää, niin sitä ei siksi kannata edes kutsua, ennen kuin sen sovelluskehys mahdollistaa käyttäjälleen täydellisen valvonnan erottelematta administraattoritasoja ja täten tarkkailutoimintojen manipuloinnin ilman, että koodia tarvitsee erikseen tätä varten muuttaa.

## 9.2 Laajennusmahdollisuudet

Koska vastaajat yksimielisesti eivät tunnustaneet DNN:n vastaavan sisällönhallintajärjestelmille annettuja peruseriaatteita, esitettiin DNN-foorumilla lisäkysymyksenä kuinka isoja muutoksia DNN tarvitsisi täyttääkseen sisällönhallintajärjestelmän perusvaatimukset, sekä mitä se vaatisi ohjelmoijalta.

Pääongelmaksi nostettiin käyttäjät ja roolit. Niiden avulla voidaan jo tällä hetkellä rajoittaa katselua aika hyvin sekä lisäksi DNN:ssa on lomakkeita, joilla sisällön iänmääritys ja päivämäärien avulla rajoittaminen onnistuu. Joten perusteet ovat jo olemassa valmiina. Kuitenkin administraattoritason laajentaminen on hieman monimutkaisempaa. Lähdekoodi on melko kokoonpantavaa sekä administraatio/hallinnointitoiminnot ovat muutettavissa vastaamaan todellisen sisällönhallintajärjestelmän vaatimuksia. Mutta silloin koodimuutokset tulee tehdä sovelluskehukseen, jos halutaan administraatiotoiminnot näyttää siellä missä

niitä tarvitaan. Tämä kuitenkin on mahdollista tehdä ilman sen suurempia vaatimuksia.

Työvirranhallinnan vaatimat muutokset, kuten hyväksynnit, voidaan tehdä sisäänrakennetuilla koottavilla toiminoilla, kuten viestien lähetys ja jne. Kaikista suurin muutos DNN:ssa tarvitaan tietotaulukoihin (engl. data tables). Nämä pitää laajentaa hyväksymään ylimääräiset tarkistukset sekä työvirran tiedot. Mutta vastaajien mielestä nekin ovat tehtävissä.

Pitää kuitenkin muistaa, että kun kerran tekee tällaisia muutoksia lähdekoodiin, menettää mahdollisuuden automaattisiin DNN-päivityksiin ja sovelluskehiksestä tulee täysin mukautettu paketti. Toisena mahdollisuutena on tehdä muutokset ja laajennukset DNN:n moduulien kautta ja säilyttää sovelluskehys, mutta tämä ei vastaajien mielestä ole paras lähestymistapa. Tätä mietittäessä on huomioitava lisämoduulit ja lisättävä työvirranhallinta niihin. Ongelmana tässä on kuitenkin se, että nykyään DNN:n lisämoduulit eivät tule lähdekoodin kanssa, vaan lähdekoodi joudutaan ostamaan vielä moduulin lisäksi, joten kustannukset saattavat nousta sen verran, että siinä vaiheessa olisi jo parempi mennä ja ostaa oikea sisällönhallintajärjestelmä.

Mietittäessä sitä voisiko DNN:n kehittäjät lisätä tarvittavat sisällönhallintajärjestelmältä vaadittavat ominaisuudet lähdekoodiin, vastaajat epäilevät ettei sitä tapahdu, koska alunperinkään DNN:a ei ole luotu varsinaiseksi sisällönhallintajärjestelmäksi. Ainoat jotka muuntavat DNN:n lähdekoodin vastaamaan sisällönhallintajärjestelmien vaatimuksia, ovat intranetin käyttäjät, jotka mukauttavat sovelluskehiksen tarvitsemakseen. Tavalliset verkkokäyttäjät eivät edes tarvitse tällaista oikeanlaista sisällönhallintajärjestelmää, vain yritys- ja julkisensektorinkäyttäjät tarvitsevat.

Eräs vastanneista DNN:n käyttäjistä kertoo muokanneensa sitä siitä lähtien kun aloitti DNN:n käytön. Hän toteaa, että hänen asiakkaansa, jotka käyttävät kahta tai kolmea eri sisällönhallintajärjestelmää tarvitsevat rajoituksia sisällön syötteisiin, yms. Tällaisissa tapauksissa hän lisää yksinkertaisia työvirranhallinta työkaluja,

joita hän nimeää sisällönhallitsijoiksi (engl. content admin) ja isäntähallitsijoiksi (engl. host admin). Vastaaja itse toimii isäntätasolla kaikille asiakkailleen, jolloin he eivät saa isännöintitoimintoja. Tällöin administraattorilla on kaikki tämän hetkiset olemassa olevat hallinnointityökalut, kun taas sisällön hallitsijoilla on rajoitetut sisällön muokkaus ja luonti mahdollisuudet.

Vastaajat totesivat, että ohjelmoijalle ei tulisi olla hankalaa tehdä tarvittavia muutoksia sovelluskehityksen lähdekoodiin. Koodi itsessään on mainiota sekä sen ydin on osista koottavissa. Mutta muokkaajan tulee pitää mielessä se, että lähdekoodi on käyttänyt jo kaikki resurssit mitä VB.NET –ohjelmointikieleen tulee. Näin ollen tämäkään ei uudelle ohjelmoijalle tulisi olla mitenkään vaikeaa huomata. Vastaajat kuitenkin haluavat muistuttaa, että tärkein asia muokattaessa DNN:a oikean sisällönhallintajärjestelmän suuntaan merkitsee sitä ettei DNN ole enää vain DNN.

### 9.3 Tulosten yhteenveto

Vaikka DNN:a ei voidakaan sanoa täydelliseksi sisällönhallintajärjestelmäksi, on siinä silti hivenen myös sitä. Yhteenvetona voidaan tutkimustuloksista päätellä, että vaikka DNN:sta löytyy tiettyjä sisällönhallintaan vaadittavia ominaisuuksia, DNN:a ei voida vain näiden ominaisuuksien perusteella kutsua www- sisällönhallintajärjestelmäksi. Sisältökokoelma puuttuu kokonaan DNN:sta. Työvirranhallinnan osalta moduuleista löytyy vain hieman ominaisuuksia näihin tehtäviin, joita työvirranhallinnalta odotetaan. Hallinnointityökalut DNN:ssa on jaettu isäntä- ja administraattoritasoille, nämä eivät tutkimukseen vastanneiden mielestä täyttäneet sisällönhallinnalle vaadittavia ominaisuuksia, ja niitä tulisi kokonaisuudessa laajentaa. Kuitenkin DNN:n lähdekoodiin olisi helppo toteuttaa sisällönhallintajärjestelmältä vaaditut ominaisuudet, sekä ominaisuuksia voi luoda lisää lisämoduulien avulla. Tämä mahdollistaisi siis DNN:n kehittymisen www- sisällönhallintajärjestelmäksi. Ongelmana tässä tulisi olemaan ettei DNN voitaisi enää päivittää, vaan siitä tulisi yksittäinen versio DNN:sta www- sisällönhallintajärjestelmänä, jota ei todellisuudessa enää voitaisiinkaan kutsua DNN:ksi.

## 10 YHTEENVETO

Sisällönhallinta voidaan siis jaotella karkeasti kolmeen eri ryhmään, joita ovat dokumentinhallinta, julkaisunhallinta eli www-sisällönhallinta sekä digitaalisten resurssienhallinta. Www-sisällönhallinta keskittyy näistä kaikista lähinnä monikanavajulkaisuun ja www-sisältöjen tuomiin haasteisiin. Www-sisällönhallinnasta voidaan erotella erikseen dokumenttien ja työvirranhallinta, version- ja asiakkuuksienhallintaohjelmistot, sekä julkaisunhallinta. Yhtenä tärkeänä osana vielä rakenteisten dokumenttien hallinta. Www-sisällönhallinnalle muita tunnusomaisia piirteitä ovat pienet sisältöyksiköt, älykkäisiin sivupohjiin perustuva julkaisu, metatietojen hallinnan korostunut rooli, sisältöjen personointi sekä sisältöjen, rakenteiden ja ulkoasun erottaminen toisistaan.

Www-sisällönhallinnassa keskitytään resurssien parempaan hyödyntämiseen sekä sisällön laaduntarkkailuun. Merkittävänä tekijänä näiden edellä mainittujen joukossa on myös työ- ja asiakastyytyväisyys. Eli kokonaisuudessaan parempaan tiedonhallintaan ja -käsittelyyn. Tärkeintä www-sisällönhallintajärjestelmille on pystyä hallitsemaan tietoresursseja aina niiden luomisesta tuhoamiseen asti, toisin sanoen koko tiedon tai resurssin elinkaaren ajan.

Tärkeimpinä komponentteina www-sisällönhallintajärjestelmistä nousi esille sisältökokoelma, työvirranhallinta, yhteydet muihin järjestelmiin sekä hallinta- ja administraatiotyökalut. Sisältökokoelman todettiin olevan sisällönhallintajärjestelmän sydän, koska sen kautta kulkee kaikki raakamateriaali. Sisältökokoelmalla on siis merkittävin rooli koko sisällönhallinnassa. Sisältökokoelma sisältää niin tietoresurssit kuin myös liittymärajapinnan muihin järjestelmiin, joka taas mahdollistaa tiedon käsittelyn ja haun suhteen enemmän kuin tavallinen tietokanta. Lisäksi sisältökokoelma tukee työvirranhallintaa, joka on toinen tärkeä komponentti. Työvirta kuvattiin olevan sarja askelia, jotka toistuvat sisällönhallintajärjestelmässä aina tietyin väliajoin. Työvirranhallinta ohjailee koko sisällöntuotantoprosessia aina tekijöistä tehtäviin asti. Yhteydet muihin järjestelmiin sisällönhallintajärjestelmästä mahdollistavat integroida sisällönhallintajärjestelmän muihin tietojärjestelmiin ja infrastruktuureihin. Organisaatioiden

LAN- ja WAN-ympäristöt, käyttäjähallintajärjestelmät, sekä tieto- ja metatieto-järjestelmät ovat tällaisia infrastruktuurikonaisuuksia, joiden avulla voidaan helpottaa mm. käyttäjärekisterin luontia ja ylläpitoa. Sisällönhallintajärjestelmät ovat riippuvaisia niiden admistraatiojärjestelmän asettamasta rakenteesta ja parametreista. Siihen sisältyy kokoonpanojärjestelmä, joka sisältää henkilöstön roolit ja menetelmät jolla niiden oikeudet jaetaan kuin myös ne ominaisuudet jotka työvirranhallinnassa ja sisällönhallinnan rakenteessa tarvitaan.

Www-sisällönhallintajärjestelmiä on markkinoilla lukuisia ja niitä on saatavilla niin maksullisia, joiden hinnat saattavat nousta aina 75 000:een euroon asti, kuin myös vapaan lähdekoodin tuotteita, joita on mahdollisuus ladata internetistä ja joiden lähdekoodi on vapaasti kaikkien muokattavissa. Case-osiossa käsiteltiin vapaan lähdekoodin tuotetta DotNetNukea. Tutkimusongelma oli; vastaako DotNetNuke-järjestelmän ominaisuudet niitä kaikkia jotka aikaisemmissa luvuissa nostettiin esille. Esitetty olettaus, että DotNetNuke on sisällönhallintajärjestelmä oli väärä, muttei aivan täysin. Tutkimustuloksista kävi ilmi, että DotNetNuke sisältää paljon jo sisällönhallintajärjestelmältä vaadittavia ominaisuuksia. Näitä olivat muun muassa työvirranhallintaan pohjautuvat moduulit, administraatio ja hallintatyökalut. Kuitenkin näitä molempia tulisi laajentaa, jotta ne täyttäisivät www-sisällönhallintajärjestelmältä vaadittavat automatisoidut tapahtumaketjut. Varsinaista sisältökokoelmaa DotNetNukesta ei löydy, mutta dokumenttien ja tietoresurssien hallinnointi onnistuu moduulien kautta. Vaikkakin DotNetNuke käyttää SQL-serveriä ja siinä on mahdollista luoda tietokantoja, ei voida olettaa, että DotNetNuke täyttäisi sisältökokoelmalta vaadittavat ominaisuudet. Näimpä todettiin, että DotNetNuke on hyvä työkalu esimerkiksi intra- ja extra-verkkosivustojen hallintaan, muttei välttämättä suurten tietomäärien organisoimiseen. Vaikkakin DotNetNuke on laajennettavissa suuntaan jos toiseenkin, on parempi kutsua sitä verkkohallintajärjestelmäksi tai sovelluskehikseksi.

Case-osion tutkimusta on vaikea lähteä yleistämään mihinkään suuntaan, mutta kaiken kaikkiaan se toi esille paljon sellaisia seikkoja, joita ei välttämättä kirjoista lukemalla voisi selvittää. Se on monipuolisena lisänä siihen mitä kirjallinen

teoreettinen materiaali tuo esille. Oikeiden ihmisten kokemukset antavat paljon enemmän ja helposti ymmärrettävämpää tietoa kuin se mitä muualta voi saada. Niinkuin DotNetNukessakin, www-sisällönhallinnassa on vielä paljon alueita tutkittavana ja jatkuvasti kehittyvänä aiheena, siitä riittäisi varmasti tarkasteltavaa monelta eri suunnalta. Tämän opinnäytetyön jatkaja voi helposti tarttua miettimään esimerkiksi lisäosien tai -ominaisuuksien merkittävyyttä www-sisällönhallintajärjestelmissä, tai ottaa saman näkökulman tutkittaessa jotain toista www-sisällönhallintajärjestelmää.



## LÄHTEET

Boiko B. 2005. Content management bible, Indianapolis: Wiley Publishing Inc.

Butcher M. 2006. Managing and Customizing OpenCMS 6, Birmingham: Packt Publishing.

Cranor C. D., Ethington R., Sehgal A., Shur D., Sreenan C., van der Merwe J.E. 2003. Design and Implementation of a Distributed Content Management System, Monterey: ACM.

Goodwin, S., Vidgen 2002. Content, content, everywhere...time to stop and think? The process of Web content management, Computing & Control Engineering Journal 13 (2) [verkkojulkaisu]. Institute of Electrical and Electronics Engineers, Inc [viitattu 30.5.2006]. Saatavissa: <http://www.ieee.org/>

Ellis J. 2002. CMS and the Single Web Designer [verkkojulkaisu]. A List Apart Magazine and the Authors [viitattu 30.5.2006]. Saatavissa: <http://www.alistapart.com/>

Honkaranta A. & Tyrväinen P. 2005. Content management in organizations, Teoksessa Khosrow-Pour M. (toim.) Encyclopedia of information science and technology, Hershey & London: Idea Group Publishing Inc, 550-555.

Robertson J. 2004. Open-source Content Management Systems, Step Two Designs Pty Ltd. Pdf-tiedosto. Saatavissa: <http://www.steptwo.com.au>

Walker S., Santry P.J., Brinkman J., Caron D., McCulloch S., Willhite S., Hopkins B. 2005. Professional DotNetNuke ASP.NET Portals, Indianapolis: Wiley Publishing Inc.

Yo H. 2005. Content and workflow Management for Library Websites: Case Studies, Hershey & London: Idea Group Publishing Inc.

## LIITTEET

LIITE 1/1

Hello to all!

I'm doing my bachelor's thesis about web content management and the case study is about DotNetNuke. I was wondering if people here could help me with it. I know only few people who had actually used Nuke, except myself, so they can't give me much help in this. You don't have to be professional, but I hope you have own opinions about it.

I'd really appreciate your help. Spend few minutes to answer easy basic questions related to web content management systems overall, and to Nuke. You can post your answers here in the forum under my topic.

The basic idea of the research is that, does DotNetNuke correspond to the basic idea and the elements needed in web content management.

#### REPOSITORY INTERFACE

1. How DotNetNuke's repository interface provides functions for input?
2. How DotNetNuke's repository interface provides functions for access to the database?
3. How DotNetNuke's repository interface provides functions for output?

#### WORKFLOW MANAGEMENT

4. Does the workflow management answer to your expectations of workflow management overall?
5. Are the workflow functions easy to use and manage?

#### CONNECTION TO OTHER SYSTEMS

6. How does the connection to other systems work in DotNetNuke?
7. Does DotNetNuke's connections also help you to work easily with other programs?

LIITE 1/2

## ADMINISTRATION TOOLS

are for administrators to configure and manage sites and portals and the whole CMS. In DotNetNuke administrative options are divided between host level and individual portal level. CMS's purpose is to make easier to handle large amount of documents and information.

8. Do administrative tools give you possibilities to manage your system as a one complex?

9. Are the given tools easy to use in DotNetNuke?

10. Is it easy to manage a large amount of information with the administration tools?

## MODULES AND EXPANDABILITY

DotNetNuke's basic idea is to work through the modules. It comes with its basic modules, but it's extendible with modules provided by third-party suppliers.

11. Do the basic modules give you possibility to manage your web content as a one complex?

12. Is it easy to extend your system with the third-party private assembly tools?

13. Is it possible to extend DotNetNuke with the third-party private assembly tools, so that it corresponds better to the content management challenge, or are they just for extending the easy-of-use?

I'll be very thankful to you all who have interest to participate in this, and spend your time to answer the questions. Hope this will be useful for the whole community too.