

Henry Joutsijoki (toim.)

**Tietojenkäsittelytieteellisiä tutkielmia
Kevät 2018**



TAMPEREEN YLIOPISTO

INFORMAATIOTIETEIDEN YKSIKÖN RAPORTTEJA 61/2018

TAMPERE 2018

TAMPEREEN YLIOPISTO
INFORMAATIOTIETEIDEN YKSIKÖN RAPORTTEJA 61/2018
KESÄKUU 2018

Henry Joutsijoki (toim.)

**Tietojenkäsittelytieteellisiä tutkielmia
Kevät 2018**

ISBN 978-952-03-0811-7 (pdf)

ISSN-L 1799-8158

ISSN 1799-8158

Professori Erkki Mäkisen muistolle

Sisällysluettelo

Tapani Landén: <i>Sijaintitietojen hyödyntäminen käyttäjän todentamisessa</i>	1
Henna Lehto: <i>Laskennallinen luovuus ja luovat ohjelmat</i>	17
Kia Lieke: <i>Search Engine Optimization</i>	32
Sari Lindström: <i>Ohjelmistoprojektin hallinta monitoimittajaympäristössä</i>	47
Toni Linnusmäki: <i>Välimuistin hyödyntäminen hakukoneissa</i>	62
Jukka Lipsanen: <i>Pääkomponenttianalyysi kuvanpakkauksessa</i>	77
John Mäkelä: <i>Agenttipohjainen mallinnus filosofian tutkimusmenetelmänä</i>	93
Jonna Paksunen: <i>Palveluarkkitehtuuri liiketoimintatiedon hallinnan järjestelmissä</i>	114
Johanna Sipinen: <i>Tietojärjestelmien eettiset ongelmat</i>	129
Anna Solantausta: <i>Automatic Humor Recognition in Verbal Contexts</i>	151
Miklos Strömberg: <i>Nudge ja retoriikka – käyttäjän toiminnan ohjailu verkkopalveluissa</i>	166
Tuisku Taivainen: <i>Kulttuurierot suhtautumisessa informaatiotiheyteen ja asetteluun käyttöliittymäsuunnittelussa</i>	187
Tuomas Tammela: <i>Maaston generoiminen fraktaalien avulla</i>	210
Väinö Teisko: <i>Katsaus sumeaan klusterointiin</i>	226
Joonas Tontti: <i>Co-Teacher verkkopalvelun käyttöliittymän esittely ja evaluointi</i>	248
Sami Tuomaala: <i>Teknologinen singulariteetti – tulevaisuuden uhka tai mahdollisuus</i>	280
Saija Uusimäki: <i>Verkkokauppaa Suomessa</i>	301
Toni Weckroth: <i>Mielipideanalyysi ja Twitter: Big data – aikakauden kristallipallo?</i>	324
Yizhou Ye: <i>Puolustautuminen sanakirjoihin perustuvilta salasanahyökkäyksiltä</i>	349

Sijaintitietojen hyödyntäminen käyttäjän todentamisessa

Tapani Landén

Tiivistelmä.

Tarkastelen tutkielmassani sitä, kuinka sijaintitietoja voidaan hyödyntää käyttäjän todentamisessa ja kuinka tällaisia keinoja tällä hetkellä käytetään. Esitän myös näkemyksiä siitä, miksi sijaintitietoja kannattaa hyödyntää käyttäjän todentamisessa porautumalla yleisimmän todennusmenetelmän, käyttäjätunnus – salasana-parin, ongelmallisuuteen. Mobiililaitteiden yleistymisen myötä käyttäjien sijaintitietoja on saatavilla yhä enenevässä määrin ja tästä syystä onkin tärkeää ymmärtää, miten niitä voi hyödyntää. Toisaalta taas useat palvelut vaativat jonkinlaisen todentamismenetelmän käyttämistä käyttäjän tunnistamiseksi, joten on varsin loogista, että painetta näiden kahden asian yhdistämiseen on olemassa. Kun tähän vielä lisätään se tosiasia, että käyttäjät eivät pidä salasanojen ja tunnusten syöttämisestä, ja toisaalta taas se, että sijaintitiedot saadaan kerättyä ilman käyttäjän henkistä tai fyysistä kuormittamista, voidaan ymmärtää sijaintitietojen käyttämisen tuomat edut osana erilaisten palveluiden todentamismenetelmiä. Toisaalta taas sensitiivisten henkilötietojen, mitä sijaintitiedot ovat, keräämiseen liittyy paljon ongelmia ja lainsäädännöllisiä vaatimuksia, jotka sijaintitietoja käyttäviä palveluita kehittävien tahojen on syytä pitää mielessään.

Avainsanat ja -sanonnat: Sijaintitiedot, todentaminen, MLC, GNSS, salasanat.

1. Johdanto

Salasanojen yli 50 vuoden valta-asema käyttäjien todentamismenetelmänä on säilynyt kaiken muun tekniikan kehityksestä huolimatta, vaikka ne sisältävät tietoturvaongelmia eivätkä ole lainkaan pidettyjä käyttäjien keskuudessa [Bonneau *et al.* 2015]. Todennusmenetelmät vaikuttavatkin olevan ominaisuuksiltaan Pareto-tasapainossa – siis yhden toivotun ominaisuuden saaminen vaatii toisesta ominaisuudesta tinkimistä [Bonneau *et al.* 2012]. Tästä syystä on tärkeää, että todennusmenetelmän valinta tehdään arvioimalla halutun päämäärän kannalta tärkeimpiä ominaisuuksia ja valitsemalla sitten näiden arvioiden perusteella halutut ominaisuudet parhaiten toteuttava menetelmä.

Yhä useamman Internetiin kytketyn laitteen ollessa mobiililaite ja yleisen laitteistojen kehittymisen myötä paikkatietojen käyttö ja saatavuus on kasvanut viime vuosien aikana. Esimerkiksi Ericssonin "Mobile traffic" -raportti vuoden 2017 kolmannelta kvartaalilta osoittaa, että mobiilidataliikenne kasvoi maailmanlaajuisesti 65% vuoden aikana. Kasvun myös odotetaan pysyvän samansuuntaisena [Ericsson 2018].

Paikkatietojen keräämiseen on olemassa useita eri vaihtoehtoja, joiden hyödyntämistä käyttäjän todentamisessa voi tapauksesta riippuen käyttää eri tavoilla. Koska kaikilla menetelmillä hankittuja paikkatietoja ei voi pitää täysin luotettavina, on ohjelmistojen kehittäjien otettava tämä huomioon. Useasti paikkatietojen käyttöä käyttäjien todentamisessa tuleekin käyttää tästä syystä vain yhtenä osana sitä. Erityisesti asiakaspuolen antamiin tietoihin tulee suhtautua varauksella, sillä käyttäjän on mahdollista väärentää tällä tavoin kerättyjä tietoja.

Koska paikkatiedot voivat kertoa niihin liittyvästä käyttäjästä arkaluontoisia asioita, liittyy niiden käyttöön eettisiä ja lainsäädännöllisiä kysymyksiä. Esimerkiksi Suomessa työnantaja ei saa kerätä työntekijään yhdistettävää paikkatietoa edes työntekijän suostumuksella, ellei työn erityispiirteiden takia tämä ole perusteltavissa [Tietosuojala 2018]. Toisaalta taas lainsäädäntö voi joissakin tapauksissa myös lisätä tarvetta paikkatietoihin perustuvien palveluiden kehittämiseksi. Esimerkiksi kuluttaja-asiamiehen vuonna 2016 tekemän linjauksen mukaan operaattoreilla on vastuu selvittää mobiilimaksuilla tehtyihin ostoihin liittyviä epäselvyyksiä [Kuluttaja-asiamies 2016]. Tällaisissa tapauksissa käyttäjän sijaintitiedoista voi olla hyötyä.

Käyn tutkielmani luvussa 2 läpi joitakin käyttäjän todentamismenetelmiä sekä niistä yleisimmän, eli salasanojen, käyttöön liittyviä ongelmia. Luvussa 3 esittelen erilaisia tapoja saada käyttäjän sijaintitietoja. Luvussa 4 käyn läpi tapoja hyödyntää sijaintitietoja käyttäjän todentamisessa ja kerron tarkemmin Visan Mobile Location Confirmationista. Luku 5 on yhteenveto.

2. Todentamismenetelmät

Tällä hetkellä tekstipohjaiset salasanat ovat edelleenkin yleisin tapa implementoida käyttäjän todentaminen web-palveluun. Keskeisimpänä syynä tälle on se, että se on palveluntarjoajalle helpoin keino todentamisen käyttöönottoon. Lisäksi sen suurena etuna on se, että tämä todennusmenetelmä on yleisyytensä takia käyttäjälle tuttu, minkä johdosta käyttäjän ei tarvitse opetella sen käyttöä [Bonneau *et al.* 2012].

Sen sijaan salasanojen käyttö todentamismenetelmänä sisältää useita erilaisia käytettävyyttä ja tietoturvaongelmia, joista monet juontuvat siitä, että yleensä salasanan valinta ulkoistetaan käyttäjälle, jolloin suojauksen vahvuus on riippuvainen käyttäjästä. Vaikka salasanoille on olemassa ohjeita, eivät ne ole kaikkialla samanlaisia eikä käyttäjää voida vaatia noudattamaan niitä kaikkia: esimerkiksi ohjeistus siitä, että käyttäjä ei käyttäisi samaa salasanaa monessa palvelussa, jää usein toteutumatta [Bonneau *et al.* 2015] ja on mahdotonta valvoa. Toisaalta näillä vaatimuksilla ei ole todistettu olevan vaikutusta epätoivottujen seuraamusten ehkäisyssä [Bonneau *et al.* 2015].

Käyttäjillä on usein tunnukset moneen eri paikkaan ja useiden käyttäjätunnus – salasana -parien muistaminen on henkisesti kuormittavaa, mikä johtaa helposti samojen tunnusten käyttöön eri sivustoilla [Dhamija and Dusseault 2008]. Lisäongelman salasanojen käytössä aiheuttaa se, että palveluiden kehittäjät eivät aina noudata suositeltuja käytäntöjä: monissa palveluissa salasanojen yhteydessä ei käytetä tiivistettä, vaan ne tallennetaan sellaisinaan tietokantaan. Jos taas tiivistettä käytetään, ei välttämättä käytetä suolaa, mikä tekee salasanan arvaamisen sateenkaaritauluja hyödyntävien raakavoimahyökkäysten avulla mahdolliseksi [Bonneau *et al.* 2015]. Tässä yhteydessä tiiviste tarkoittaa salausfunktioita ja suola siihen lisättävää ylimääräistä käyttäjään liitettyä uniikkia merkkijonoa, joiden avulla salasanat muutetaan sellaiseen muotoon, että niiden muuntaminen takaisin selväkielisiksi käy nykyaikaisilla välineillä lähes mahdottomaksi.

Koska salasanat perustuvat ”mitä käyttäjä tietää?” –tekijään, niiden käyttöön liittyy paljon tietoturvariskejä, joita vastaan ei voida suojautua täydellisesti pelkästään tekniikkaa parantamalla. Salasanoja voidaan myös kerätä niin kutsutuilla ”kalastelusivuilla”. Kalastelusivu rakennetaan näyttämään käyttäjän tuntemalta palvelulta, mutta jonka tarkoitus on saada hänet luovuttamaan käyttäjätunnuksensa ja salasansansa sivustoa todellisuudessa ylläpitävälle, mahdollisesti pahantahtoiselle taholle [Bonneau *et al.* 2015].

Internetissä olevien kalastelusivujen lisäksi myös reaali maailman ihmiset voivat esittää olevansa jotain sellaista, mitä he eivät ole. Ihmiset saattavat luovuttaa salansansa pahantahtoiselle taholle, joka esiintyy työpaikan tietojärjestelmien ylläpitäjänä [Lal *et al.* 2015]. Ihmisen alttius totella auktoriteetteja – tai sellaisena esiintyvää tahoja – etenkin asioissa, jotka tuntuvat heistä vaikeilta, tekee heistä erityisen haavoittuvaisia sosiaaliselle hakkeroinnille. Tämä on vaarana erityisesti isoissa organisaatioissa, jonka henkilöstö ei välttämättä edes tiedä, kuka heidän tietojärjestelmien ylläpitäjä on.

Salasanat eivät kuitenkaan ole enää nykyaikana ainoa keino todentaa käyttäjä. Teknologian kehittyessä uusia tapoja tunnistaa käyttäjä ilmaantuu ja esimerkiksi matkapuhelimien ympäristöstään dataa keräävien sensoreiden määrän kasvaessa näitä keinoja voidaan myös ottaa enenevässä määrin käyttöön.

Biometrinen todentaminen perustuu ”mitä käyttäjä on?” – tekijään. Tämä tarkoittaa sitä, että mikä tahansa käyttäjään liittyvä uniikki ja mitattava ominaisuus voi olla biometrisen todentamisen perusta. Fyysisiä ominaisuuksia ovat esimerkiksi kasvopiirteet ja sormenjäljet kun taas käyttäytymiseen liittyviä käyttäjän ääni ja allekirjoitus [Lal *et al.* 2015]. Salasanoihin verrattuna biometristä todentamista käytettäessä ei tarvitse muistaa mitään, joten tällä tavoin todentaminen on käyttäjälle erittäin vähätöistä. Lisäksi biometrisiä tunnisteita on äärimmäisen vaikea väärentää [Lal *et al.* 2015].

”Mitä käyttäjä tietää?” – ja ”mitä käyttäjä on?” – tekijöiden lisäksi on olemassa myös kolmas tekijä, johon Smart Card -todentaminen perustuu: ”mitä käyttäjällä on?” – tekijä. Smart Cardia käytettäessä tosin usein käytetään myös PIN-koodia, jolla tietosuojaan saadaan uusi turvallisuuskerros. Kortti luetaan tarkoitukseen tehdyllä lukulaitteella, johon myös syötetään mahdollinen koodi. Koska kortti on fyysinen ja suhteellisen pieni objekti, sen heikkoutena on että se voidaan varastaa tai se voi hukkua [Lal *et al.* 2015].

3. Paikannustekniikoita

Koska käytössä oleva teknologia määrittää niin mahdollisuudet kuin rajoituksetkin, käyn seuraavaksi läpi joitakin paikannustekniikoita. Monet paikkatietoja keräävät järjestelmät, kuten esimerkiksi matkapuhelimet, eivät ole täysin riippuvaisia vain yhdestä paikannusmenetelmästä vaan käyttävät useampaa me-netelmää. Esimerkiksi satelliittipaikannusta käyttävät päätelaitteet voivat käyttää niin GPS:ää kuin Galileotakin.

3.1 Satelliittijärjestelmät

Navstar Global Positioning System (GPS) on Yhdysvaltojen puolustusministeriön alunperin sotilaskäyttöön kehittämä yksisuuntaiseen satelliittipaikannukseen perustuva järjestelmä, jossa yli 20 000 kilometrin korkeudessa oleva satelliitti lähettää 27 watin voimalla signaalia, johon on koodattu satelliitin atomikellosta saatu aikaleima ja satelliitin senhetkinen sijainti kiertoradalla [van Diggelen 2009]. Tämän lisäksi satelliitti lähettää myös näennäissatunnaista kohinaa (Pseudorandom noise, PRN). GPS-signaalin vastaanottajalaitteiden kellojen epätarkkuuden takia niiden tulee saada signaali

vähintään neljältä satelliitilta, jotta epätarkkuudet voidaan eliminoida matemaattisesti [van Diggelen 2009].

Koska perinteinen GPS suunniteltiin ulkokäyttöön, siitä kehitettiin paranneltu versio A-GPS (Assisted GPS), jota käytetään nykyään yleisesti esimerkiksi matkapuhelimien paikannuksessa. Siinä missä perinteinen GPS joutuu etsimään satelliitista lähetetyn signaalin ilman tietoa sen taajuudesta ja tämän jälkeen dekoddaamaan sen, A-GPS tarjoaa paikannuspalvelimen laskeman tiedon odotetusta taajuudesta ja satelliittien sijainnin yleensä jonkin langattoman verkon, kuten matkapuhelinverkon, kautta.

Käytännössä GPS-vastaanotin vastaanottaa satelliitin lähettämän näennäissatunnaisen kohinan, mikä luonteensa vuoksi ei kärsi esteistä samalla tavalla kuin signaalin data-osuus. Tämän näennäissatunnaisen kohinan avulla paikannuspalvelimelta saadaan oikean satelliitin lähettämä data. Tämä pudottaa ensimmäisen paikannuksen ajan noin minuutista sekuntiin. A-GPS on myös kooltaan, tarkkuudeltaan ja virrankulutukseltaan ylivoimainen tekniikka perinteiseen GPS:ään verrattuna. [Diggelen 2009].

Muita merkittäviä satelliittipaikannusjärjestelmiä (Global Navigation Satellite System, GNSS) ovat venäläinen, myöskin alunperin sotilaskäyttöön rakennettu yksisuuntainen Glonass ja EU:n ja ESA:n yhdessä siviilikäyttöön kehittänyt kaksisuuntainen Galileo [Li *et al.* 2015]. Myös Kiina on lähtenyt kehittämään omaa järjestelmäänsä, BeiDouta, joka on lähettänyt vuodesta 2012 sijainti- ja paikannustietoja Aasian – Tyynenmeren -alueelle.

Glonass-järjestelmän ensimmäinen satelliitti on laukaistu kiertoradalle vuonna 1982 ja järjestelmää ylläpitää Venäjän avaruushallinnon alainen IAC. Toisin kuin GPS, Glonass on täysin avoin. Glonass on GPS:n tavoin täysin toiminnassa, joskin sitä päivitetään jotta se voisi vastata paremmin nykyaikaisiin tarpeisiin [Li *et al.* 2015].

Galileon on tarkoitus tulla täyteen toimintavalmiuteen vuonna 2020, jolloin se koostuu yhteensä 30:stä eri tasoilla kulkevasta satelliitista. Verrattuna GPS:ään ja Glonassiin Galileon on tarkoitus olla tarkempi sekä myös kaksisuuntainen, mikä tarkoittaa sitä, että järjestelmää voi käyttää myös esimerkiksi hätäviestien lähettämiseen. Lisäksi Galileo on täysin siviilien ylläpitämä, mikä erottaa sen GPS:stä ja Glonassista [Prasad 2005].

Kuten Galileon, niin samoin myös BeiDoun on tarkoitus olla täydessä toimintavalmiudessa vuoden 2020 loppuun mennessä. Lopullinen järjestelmä tulee sisältämään viisi Geostationary Earth Orbit -satelliittia, kolme Inclined Geo-Synchronous Orbit -satelliittia ja 27 Medium Earth Orbit -satelliittia. Kun järjestelmä on täysin toimintavalmiudessa, kaikki neljä GNSS:a muodostavat yli 100 satelliitin kokonaisuuden [Li *et al.* 2015].

3.2 Matkapuhelinverkko

Matkapuhelinverkkoa ei alunperin olla kehitetty paikantamismielessä, mutta koska siihen liitettyjen laitteiden määrä on kasvanut räjähdysmäisesti, on sijaintitietojen kerääminen sen avulla yksi, yleisesti hätävararataisuksi tarkoitettu vaihtoehto. Matkapuhelinverkon sekä siihen liitettyjen päätelaitteiden jatkuvan kehityksen myötä paremmat paikannustekniikat ovat mahdollistuneet, joskin paikannus on siitakin huolimatta epätarkkaa verrattuna satelliittipaikannusjärjestelmiin.

Puhelun yhdistäminen vaatii tiedon matkapuhelimen sijainnista matkapuhelinverkon solutasolla ja tämän paikan määrittämiseen on kolme eri tapaa: sijainnin päivittäminen joka kerta kun solusta siirrytään toiseen, jokaisen verkossa olevan solun päätelaitteiden sijainnin paikantaminen ja verkon jakaminen osa-alueisiin, joiden välillä siirryessään päätelaite ilmoittaa millä osa-alueella on [Mukherjee *et al.* 2016].

Päätelaite voidaan paikantaa, mikäli se sijaitsee kolmen tukiaseman kantavuusalueella. Tämä paikantaminen tehdään mittaamalla signaalien kulkemiseen kulunutta aikaa ja laskemalla niistä puhelimen etäisyys kustakin tukiasemasta [Mukherjee *et al.* 2016].

3.3 Internetiin perustuvat järjestelmät

Suoraviivaisen signaaleihin perustuvan mittaamisen lisäksi on myös muita tapoja saada paikkatietoja. Internet perustuu tietokoneiden välillä lähetettäviin Internet-protokollan mukaisiin TCP/IP-paketteihin, jotka koostuvat metatietoa sisältävistä *otsakkeista* (header) ja varsinaisesta *sisältöosasta* (body). Otsaketietoihin sisältyy IP-osoite, jota voidaan hyödyntää laitteen sijainnin määrittelyssä. IP-osoitteiden kartoitus on mahdollista niiden sisältämien operaattoriin liittyvien tietojen avulla: IP-osoite muodostuu Internetin runkoverkossa operaattoriin liitetystä numerosarjasta sekä sen jälkeisestä liittymään yksilöidystä numerosarjasta.

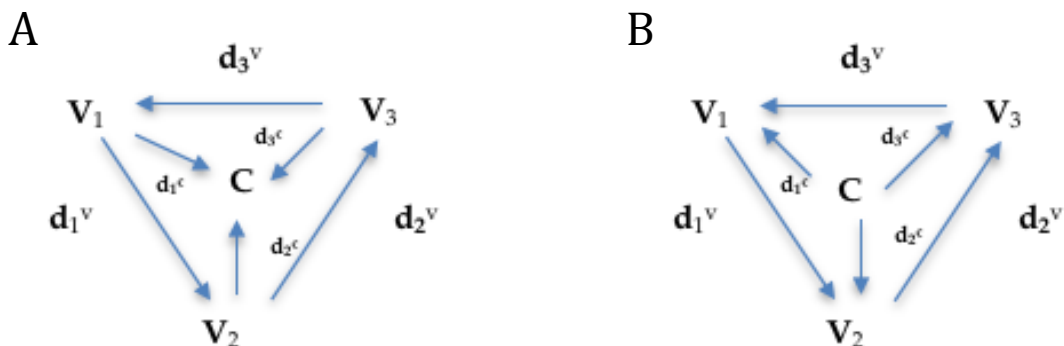
Laitteen fyysisen sijainnin selvittäminen IP-osoitteen perusteella ei ole yksinkertainen tehtävä johtuen siitä, että IP-osoite ei itsessään ole sidottu paikkaan. Tämän takia IP-osoitteiden fyysisiä sijainteja kartoitetaan tietokantoihin, joista voi saada tietokannasta riippuen sijaintitietoja eri tarkkuuksilla. Lisäksi IPv6:n käyttöönotto tulevaisuudessa lisää osoitteiden määrää tekijällä 2^{96} , jolloin jo valmiiksi sekava Internetin topologia mutkistuu entisestään [Koch *et al.* 2013]. Asiaa mutkistaa myös se, että isoilla organisaatioilla kuten Applella ja IBM:llä on käytössä omat IP-osoitevaruutensa, joten lähellä toisiaan olevilla laitteilla voi olla täysin erilaiset

IP-osoitteet. Kenelläkään ei ole myöskään velvollisuutta ilmoittaa IP-osoitteiden fyysistä sijaintia. [Koch *et al.* 2013].

Asioita mutkistaa myös se, että käyttäjä voi halutessaan salata IP-osoitteensa. Tähän tarkoitukseen esimerkiksi F-secure tarjoaa virtuaalista erillisverkkopalvelua (Virtual Private Network, VPN), jossa käyttäjän tietoliikenne ohjataan tunneloituna välityspalvelimen kautta. Tässä järjestelmässä välityspalvelin salaa ja purkaa datapaketteja. Tällöin käyttäjän laitetta ei voida paikantaa IP-osoitteen avulla ja käyttäjä voi myös esittää olevansa esimerkiksi eri maassa päästäkseen käsiksi maarajoitteita sisältäviin palveluihin.

Abdoun ja muiden [2017] esittelemä CPV-tekniikka (Client Precense Verification) perustuu Internetin välityksellä lähetettävien pakettien matka-aikoihin. Käytännössä vähintään kolmelta palvelimelta, joiden sijainti on tunnettu, lähetetään paketti pyynnön lähettäneelle päätelaitteelle, jonka sijainti halutaan varmistaa. Pyyntöjen ja vastausten viiveestä voidaan laskea Internetin nopeuden (joka on noin kahdeksasosa valonnopeudesta) ja trigonometristen funktioiden avulla todennäköisyys sille, että kyseinen asiakaspääte sijaitsee väitetyllä alueella [Abdou *et al.* 2017]. Kuvassa 1 havainnollistetaan yllä kuvailtu tilanne

Tätä tekniikkaa voidaan käyttää esimerkiksi silloin, kun halutaan varmistaa että uhkapelisivuston käyttäjä sijaitsee osavaltiossa, jossa uhkapelit ovat sallittuja. Etuna CPV:llä GPS:n tai muihin GNNS-järjestelmiin nähden on se, että GPS:n antamat koordinaatit annetaan asiakaslaitteelta palvelimelle, jolloin käyttäjällä on mahdollisuus päästä manipuloimaan asiakaslaitteelta lähetettäviä tietoja, kun taas CPV:llä palvelimet laskevat käyttäjän sijainnin. IP-osoitteeseen perustuvia paikkatietoja voidaan taas väärentää tai ne voidaan salata (ks. IP-osoitteista kertova kohta) [Abdou *et al.* 2017].



Kuva 1: Kuvaus tilanteessa, jossa kolme tunnetulla sijainnilla olevaa todentajaa (v) mittaavat etäisyyden (d) asiakaspääteeseen (c) lähettämällä sille paketin (A)

ja sen jälkeen vastaanottamalla asiakkaan lähettämän paketin (B). Tiedossa on myös todentajien välinen etäisyys. [Abdou *et al.* 2017]

3.4 Lyhyen etäisyyden tekniikat

Aiemmissä kohdissa esitetyt tekniikat perustuvat kaikki varsin kaukana toisistaan olevien laitteiden sijainnin määrittämiseen esilaisten laskelmien avulla. Tällaisten tekniikoiden lisäksi on olemassa myös tekniikoita, joita käytetään paljon lyhyempien matkojen välillä, mutta jotka kuitenkin perustuvat laitteiden keskinäiseen sijaintiin. Tällaisia tekniikoita ovat esimerkiksi Wifi ja Bluetooth.

Näiden lisäksi myös pankkikorteissa käytettävä lähimaksuominaisuus ja matkakorttien matkan maksaminen perustuvat kortin ja lukulaitteen väliseen lyhyeen etäisyyteen, jolloin lukulaitteen sijainnin ollessa tiedossa myös kortin käyttäjän sijainti tiedetään lukutapahtuman hetkellä. Sijaintitietojen saaminen näiden mainittujen tekniikoiden avulla vaatii kuitenkin sen, että signaalin lähettävän laitteen sijainti on tiedossa, mikä taas vaatii jonkun aiemmin mainitun, isomman mittakaavan tekniikan hyödyntämistä.

Luonnollisesti käyttäjä voi myös itse ilmoittaa oman sijaintinsa: tällaiseen tietoon tulee kuitenkin aina suhtautua varauksella, joten minkäänlaiseen sensitiiviseen dataan tällä tavoin saadulla tiedolla ei tulisi päästä käsiksi. Toisaalta taas esimerkiksi mainonnan kohdentaminen jonkin kaupungin säätietoja katsovalle henkilölle voi olla joissain tapauksissa mielekästä.

4. Paikkatietojen hyödyntäminen käyttäjän todentamisessa

Kuten edellisessä luvusta käy ilmi, paikkatietoja voidaan kerätä usealla erilaisella menetelmällä, joista jokaisella on hyvät ja huonot puolensa. Se, minkälaista teknologiaa tai niiden yhdistelmää ohjelmistokehittäjä aikoo käyttää, riippuu aina siitä, minkälaiseen palveluun niitä on käyttämässä.

Mikäli kehitettävässä palvelussa ei ole sensitiivistä tietoa tai vaihtoehtoisesti palveluntarjoajalla ei ole syytä olettaa palvelun väärinkäytön olevan käyttäjän intresseissä, palvelun käytettävyyttä nousee tärkeämmäksi kriteeriksi kuin mahdollisiin väärinkäyttöihin varautuminen, jolloin myös käyttäjän todentamisessa voidaan käyttää menetelmiä, jotka menestyvät parhaiten näiden ominaisuuksien vertailussa. Erityisesti tällaisissa palveluissa pelkkä sijainti on niin käyttäjälle kuin palvelun tarjoajallekin helppo ja riittävä tapa toteuttaa todentaminen.

4.1 Beaconit

Beaconien (suora käänös ”merkkitulo”) aika alkoi vuonna 2013 Applen lanseeratessa iBeacon-protokollan. Google taas julkisti oman Eddystone-Beaconin vuonna 2015 ja nykyisin nämä kaksi yhtiötä hallitsevat Beacon-markkinoita.

Beacon-lähetin on pieni, Bluetoothin avulla signaalia lähettävä laite. Tätä signaalia BLE (Bluetooth Low Energy) Beacon-lähetin pystyy lähettämään 70 metrin päähän. Jos tällainen Beacon-lähetin lähettää signaalin 300 millisekunnin välein, sen käyttöikä on pienen litiumpariston kanssa noin kaksi vuotta [Akinsiku and Jadav 2016].

Beacon-protokollissa BLE Beacon-laitteille määritellään neljä eri roolia: *ohjeislaite* (Peripheral device), joka voi olla esimerkiksi älykäs lämmönsäätelyjärjestelmä; *päälaite* (Central device), joka voi olla esimerkiksi älypuhelin; *lähetin* (Broadcaster), joita esimerkiksi iBeacon ja Eddystone ovat; sekä *tarkkailija* (Observer), joka on dataa joltain lähettimeltä vastaanottava laite. Kaksi ensinmainittua ovat kaksisuuntaisia kun taas kaksi viimeisintä ovat yksisuuntaisia [Akinsiku and Jadav 2016].

Sekä Applen iBeacon että Googlen Eddystone toimivat yksisuuntaisina lähettiminä. Näiden tarkkailijana toimii iBeaconin tapauksessa iOS-käyttöjärjestelmällä toimiva laite kun taas Eddystone toimii sekä iOS-käyttöjärjestelmällä että Androidin Observerin kanssa [Akinsiku and Jadav 2016].

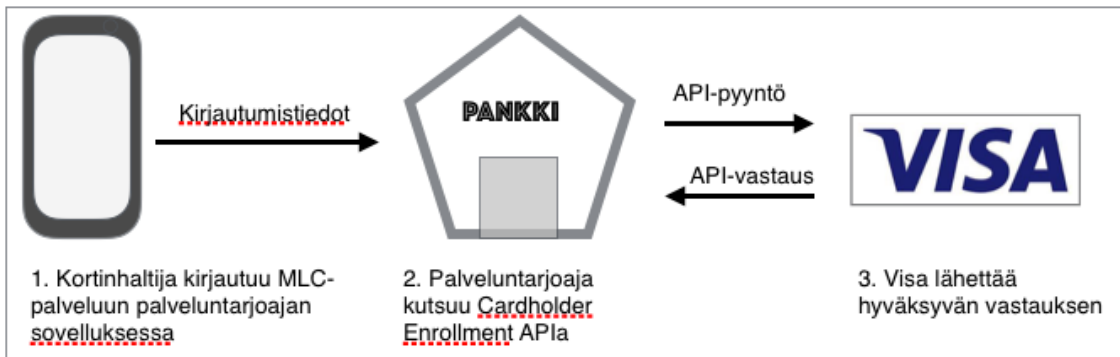
Beaconien lyhyehkö toimintasäde mahdollistaa sen käyttämisen käyttäjän todentamisessa esimerkiksi kohdennustarkoituksissa. Käyttäjä voidaan todentaa potentiaalisesti asiakkaaksi silloin kun hän on liikkeen sisällä ja näin hänelle voidaan kohdentaa Beacon-lähettimen avulla esimerkiksi virtuaalisia tarjouskuponkeja [Akinsiku and Jadav 2016]. Toisaalta taas suurissa yleisötapahtumissa Beacon-lähettimen lähettämä ID-numero voisi toimia tunnisteena käyttäjän antaman nimimerkin kanssa tapahtumaan liittyvällä yksityisellä keskustelualueella.

Esimerkki pidemmällä välimatkoilla sijaintitietojen hyödyntämisestä käyttäjän todentamisessa on luottokortin käyttäminen ulkomailla, mikä näyttäytyy luottoyhtiöille epäilyttävänä toimintana ja saattaa aiheuttaa transaktion hyväksymättä jättämisen. Tähän ongelmaan Visa on kehittänyt palvelun, jonka avulla pankki pystyy määrittämään mobiilipankkisovellusta käyttävän käyttäjän sijainnin ja sen avulla todentamaan epäilyttävältä vaikuttavan tapahtuman.

4.2 Mobile Location Confirmation

MLC (*Mobile location confirmation*) on luottoyhtiö Visan kehittämä palvelu, jonka tarkoituksena on vähentää luottokorttitapahtumien väriä hylkäysten määrää parantamalla Visan palveluita tarjoavan yrityksen mahdollisuuksia saada varmennettua käyttäjän sijainti [Visa 2018]. Ensimmäisenä tämän palvelun otti käyttöönsä Arabiemiraattien Emirates NBD vuonna 2015 [PR Newswire 2015].

MLC implementoidaan osaksi Visan palveluita tarjoavan yrityksen mobiilisovellusta niin että kun käyttäjä, jonka haluaa käyttää MLC-palvelua, kirjautuu sisään sovellukseen ja asettaa MLC-palvelun päälle, asiakassovellus lähettää kirjautumistietoja palveluntarjoajalle, mistä ne taas lähetetään Visalle, joka lähettää palveluntarjoajalle hyväksymisviestin. Tämä prosessi esitetään kuvassa 2.



Kuva 2: MLC:n kirjautumisvaihe kohta kohdalta [Visa 2018].

Kirjautumistiedot sisältävät PAN-numeron (*Primary account number*), palveluntarjoajan luoman laitteeseen liitetyn 32 merkkiä pitkän uniikin ID-numeron, joka voidaan tehdä esimerkiksi MAC-osoitteen, päivämäärän ja ajan perusteella, sekä Visan palveluntarjoajalle antaman kuusimerkkisen ID-numeron [Visa 2018].

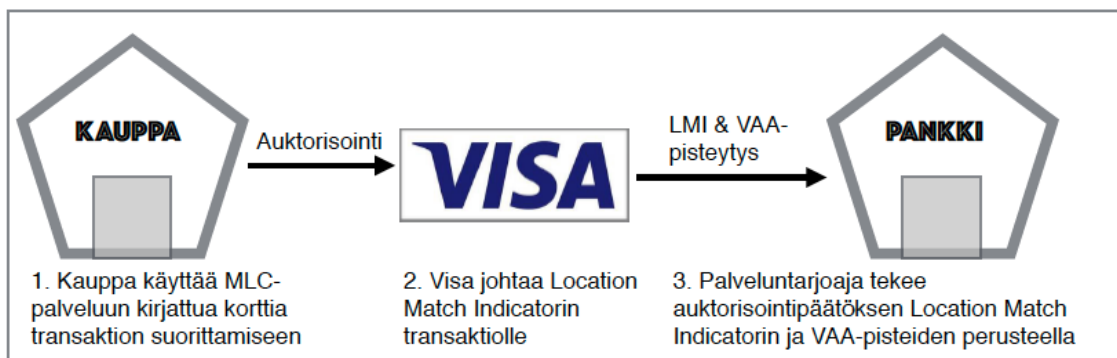
Kun kirjautuminen on hyväksytty, palveluntarjoajan sovellukseen sisällytetty Mobile Location Agent aloittaa lähettämään säännöllisesti paikkatietoja Visalle. Mobile Location Agent on Visan tarjoama lähdekoodimuodossa oleva ohjelmointirajapinta, jonka päälle ja pois päältä asettaminen on palveluntarjoajan vastuulla. Palvelun ollessa päällä se lähettää käyttäjän sijaintiin liittyvää dataa Visalle tallennettavaksi säännöllisesti, joskin se sisältää rajoituksia, joiden avulla käyttäjän yksityisyys on paremmin suojassa. Esimerkiksi käyttäjän ollessa kotialueeksi määritellyllä alueella, palvelu ei lähetä tietoja Visalle yhtä usein. Tietyn etäisyyden matkustaminen taas saa palvelun lähettämään sijaintitietoja, vaikka normaali tietojenlähetyskertojen välinen aika ei olisikaan kulunut. Tämä vaihe on esitetty kuvassa 3



Kuva 3: Sijainnin päivitys [Visa 2018].

Palveluntarjoaja voi myös päättää, että sijaintitiedot kulkevat palveluntarjoajan kautta ennen Visalle päätymistä, joskin tällöin palveluntarjoaja ottaa vastuun niiden lähettämisestä Visalle Location Update API:n avulla.

Lopulta kun seuraava transaktio suoritetaan, Visa johtaa transaktion yritysosapuolen sijaintitiedoista sekä MLC:n kautta saaduista käyttäjän sijaintitiedoista Location Match Indicatorin ja lähettää sen Visa Advanced Authorization -pisteiden välityksellä palveluntarjoajalle. Visa Advanced Authorization on analytiikkaan perustuva, käyttäjien toimia ennakoiva pisteytysjärjestelmä, jonka avulla palveluntarjoaja pystyy arvioimaan transaktioiden luotettavuutta [Visa 2018]. Tämä vaihe esitetään kuvassa 4.



Kuva 4: Transaktion auktorisointiprosessi [Visa 2018].

MLC-palvelu toimii vain iOS- tai Android-järjestelmää käyttävissä matkapuhelimissa, joten puhelimen mallin lisäksi käyttöjärjestelmä määrittelee tavan, jolla MLC saa paikkatiedot. Tämä tarkoittaa siis sitä, että sijaintitietojen kerääminen on asiakaspuolen vastuulla, mikä mahdollistaa kyseisiin tietoihin käsikäsittämisen ja muokkaamisen. Toisaalta taas käyttäjän intresseissä ei suurimmassa osassa tapauksista olisi väärentää omaa sijaintiaan, sillä paikka-

tietoja käytetään maksun vahvistamiseen, mikä on yleisesti toivottua vain silloin, kun on oikeasti maksamassa jotain.

Koska palvelu hyödyntää matkapuhelinta, voi esimerkiksi käsilaukun varastanut henkilö saada käsiinsä sekä luottokortin että puhelimen, josta MLC saa käyttämänsä sijaintitiedot, jolloin varkaan tekemä MLC:llä vahvistettu transaktio todennäköisesti hyväksytään, mikäli kortin oikea omistaja ei ole ehtinyt ilmoittamaan varkaudesta.

4.3 Eri todennusmenetelmiä paikkatietoihin yhdistävät tekniikat

Koska yksikään todennusmenetelmä ei ole kaikilta osiltaan ylivertainen muihin nähden, korostetaan tutkimuksessa useamman todennusmenetelmän yhtäaikaista käytöstä syntyviä hyötyjä [Bonneau *et al.* 2012]. Käyttäjätunnuksella ja salasanalla toteutettuun todennusmenetelmään saadaan yksi suojakerros lisää implementoimalla niiden rinnalle paikkatietojen hyödyntäminen.

Älypuhelimien yleistymisen on avannut uusia mahdollisuuksia hyödyntää käyttäjän sijaintia tunnistautumisessa. Käyttäjän ollessa tietyssä paikassa hänelle voidaan esittää kyseiseen paikkaan liittyviä kysymyksiä. Sijaintitietoja voidaan hyödyntää myös sellaisissa tapauksissa, joissa tietyn esineen, kuten autonavaimien, vieminen laitteen lähistölle aktivoi halutun laitteen.

Käyttäjän todentamisessa voidaan käyttää hyödyksi yhdistelmää, jossa käyttäjään yhdistetty laite on tietyllä määritellyllä alueella. Näin esimerkiksi käyttäjän puhelimella kotona käytetty pankkisovellus voi toimia vähemmän käyttäjää kuormittavalla todennusmenetelmällä kun taas saman palvelun käyttö muualla vaatii ylimääräisen suojakerroksen, esimerkiksi vaihtuvan salasanan.

4.4 Sijaintitiedoista riippuvaiset palvelut (Location-based Services, LBS)

Sijaintitiedot voivat olla myös palvelun bisnesmallin keskiössä, jolloin niiden luotettavuus on ensiarvoisen tärkeää. Tällaisia palveluita kutsutaan *sijaintitiedoista riippuvaisiksi palveluiksi* (Location-based services). Tähän palveluiden ryhmään kuuluvat muun muassa sellaiset palvelut, jotka perustuvat tekijänoikeuksilla suojattuun materiaaliin ja joiden sisältö vaihtelee sijainnista riippuen (esimerkiksi Spotify ja Netflix); uhkapelejä tarjoavat sivustot, jotka voivat rajoittaa osaa palveluistaan aluekohtaisesti; nettiäänestykset sekä paikallisuuteen perustuvat sosiaalisen median palvelut (esimerkiksi Tinder) [Abdou *et al.* 2015].

Etenkin Amerikassa sijaintitietoja hyödyntäviä palveluita käyttää alati kasvava määrä ihmisistä: jo vuonna 2011 28 prosenttia aikuisista käytti

jonkinlaista mobiilipohjaista LBS:a ja näiden palveluiden seuraavan vuosikymmenen arvon laskettiin olevan 700 miljardia dollaria. [Masters 2014].

Sijaintitietoihin perustuvia palveluita voi käyttää myös hyväntekeväisyyteen. Hyvänä esimerkkinä tällaisesta on Grand Visual -nimisen yrityksen The Microloan Foundationille tekemä kampanja, jossa interaktiivinen mainostaulu sisälsi keskeneräisen, kolikoista muodostetun kuvan hymyilevästä afrikkalaisesta naisesta. Lähettämällä tekstiviestin taululla olevaan numeroon käyttäjä lahjoitti tietyn rahasumman, mikä käynnisti mainostauluun animaation, joka sisälsi kiitosviestin, jossa lahjoituksen antanutta henkilöä kiitettiin nimeltä, minkä jälkeen kourallinen kolikoita lentää kuvan päälle ja piirtää naisen kuvan valmiimmaksi. Viikonlopun aikana tämä mainostaulu tavoitti 321 000 henkilöä ja sen avulla 21 afrikkalaista naista sai perustettua oman yrityksen. [Grand visual 2012].

4.5 Sijaintitietojen käyttöön liittyvät haasteet

Vaikka palveluntarjoajat pystyvät tuottamaan parempia palveluita ja helpompia tapoja tunnistaa käyttäjä käyttäessään sijaintitietoja, liittyy niiden käyttöön myös haasteita.

Sijaintitietojen ollessa luonteeltaan sensitiivisiä, moni käyttäjä miettii niitä keräävien palveluiden luotettavuutta. Lähiaikoina paljastuneet, massiiviset tietojen väärinkäytökset esimerkiksi Cambridge Analytican tapauksessa, jossa yhtiö käytti Facebookista saatuja henkilötietoja hyödyksi kampanjoissa, joihin sillä ei ollut lupaa niitä käyttää, on nakertanut ihmisten luottamusta yritysten kykyyn toimia vastuullisesti sensitiivisen tiedon kanssa.

Myös tämän vuoden toukokuun lopulla voimaan astuva EU:n tietosuojauudistus kertoo siitä, että yhteiskunnallisella tasolla ollaan herätty uudenlaiseen tarpeeseen suojata käyttäjien yksityisyyttä Internetissä olevilla palveluilla. Tässä uudistuksessa linjataan suoraviivaisesti rekisterinpitäjän velvollisuuksia, joita EU:n alueella tarjottavissa palveluissa on noudatettava ja jotka liittyvät nimenomaan henkilötietoihin.

Tietosuojavaltuutettu määrittää henkilötietojen käsittelyn seuraavilla kuudella kohdalla:

1. *Henkilötietoja on käsiteltävä lainmukaisesti, asianmukaisesti ja rekisteröidyn kannalta läpinäkyvästi*
2. *Henkilötietoja on käsiteltävä luottamuksellisesti ja turvallisesti*
3. *Henkilötietoja on kerättävä ja käsiteltävä tiettyä, nimenomaista ja laillista tarkoitusta varten*

4. *Henkilötietoja on kerättävää vain tarpeellinen määrä henkilötietojen käsittelyn tarkoitukseen nähden*
5. *Henkilötietoja on päivitettävää aina tarvittaessa – epätarkat ja virheelliset henkilötiedot on poistettava tai oikaistava viipymättä*
6. *Henkilötietoja on säilytettävä muodossa, josta rekisteröity on tunnistettavissa ainoastaan niin kauan kuin on tarpeen tietojenkäsittelyn tarkoitusten toteuttamista varten. [Tietosuojavaltuutettu 2018]*

Palveluntarjoajien on siis otettava nämä kohdat huomioon myös käsitellessään käyttäjään liittyviä sijaintitietoja, sillä ne lukeutuvat henkilötiedoiksi.

5. Yhteenveto

Tästä tutkielmastani käy ilmi, että pelkästään salasanoihin tukeutuminen ainoana todennusmenetelmänä ei enää nykypäivänä ole järkevää saati kannattavaa. Olen nostanut yhden tekniikan, käyttäjän sijaintitietojen hyödyntämisen, tässä etualalle ja käsitellyt siihen liittyviä mahdollisuuksia ja heikkouksia sekä erästä jo olemassaolevaa palvelua.

Salasanojen ongelmallisuuden takia erityisesti useampaa erilaista todennusmenetelmää yhdistävät palvelut ovat avainasemassa ja niitä tullaan todennäköisesti näkemään tulevaisuudessa yhä enemmän. Useampaa erilaista todennusmenetelmää hyödyntämällä todentamisesta saadaan turvallisempaa ja usein myös käyttäjälle helpompaa ja vähemmän kuormittavaa.

Sijaintitietojen käyttäminen onkin juuri tällä tavalla käytettynä parhaimmillaan. Pelkkään sijaintiin perustuva todentaminen ei useassa tapauksessa tule kysymykseenkään, mutta tilanteissa, joissa sijainti on keskeisessä osassa palvelua, kuten tässä tekstissä esittelemässäni Visan Mobile Location Confirmation -palvelussa, sijainnin käyttäminen maksun varmistusvaiheessa helpottaa luotonantajien tapahtuman hyväksymisessä ja tekee asioimisesta käyttäjän näkökulmasta aiempaa helpompaa.

Uskon, että ongelmistaan huolimatta sijaintitietoihin perustuvia todennuspalveluita tullaan tulevaisuudessa kehittämään lisää. Niiden ollessa käyttäjien näkökulmasta erittäin helppokäyttöisiä ja vähän vaativia, ja toisaalta taas sijaintitietojen helpon saatavuuden takia ne ovat myös palveluntarjoajille helppoja kohteita, on niiden käytön lisääntymiselle erittäin ymmärrettävät perusteet.

Viiteluettelo

- Abdel Rahman Abdou, Ashraf Matrawy and Paul van Oorschot. 2015. CPV: Delay-based location verification for the Internet. *IEEE Transactions on Dependable and Secure Computing* 14, 2, 130-144.
- Adegboyega Akinsiku and Divyesh Jadav. 2016. BeaSmart: A beacon enabled smarter workplace. In: *Proc. of the NOMS 2016 - 2016 IEEE/IFIP*, 1269-1272 .
- Joseph Tonneau and Cormac Herley, Paul van Oorschot and Frank Stajano. 2015. Passwords and the evolution of imperfect authentication. *Communications of the ACM* 58, 7, 78-87.
- Joseph Bonneau, Cormac Herley, Paul van Oorschot and Frank Stajano. 2012. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In: *Proc. of the 2012 IEEE Symposium on Security and Privacy*, 553-567
- Frank van Diggelen. 2009. A-GPS: Assisted GPS, GNSS, and SBAS. Artech House.
- Racha Dhamija and Lisa Dusseault. 2008. The seven flaws of identity management: Usability and security challenges. *IEEE Security & Privacy* 6, 2, 24-29.
- Ericssonin www-sivusto. *Mobile traffic report Q3 2017*. < www.ericsson.com >. Checked 2.4.2018.
- Grand Visual. 2012. *Live Donation Digital Poster Campaign*.
- Robert Koch, Mario Golfing and Gabi Dreo Rodosek. 2013. Geolocation and verification of IP-addresses with specific focus on IPv6. In: *Proc of the 5th International Symposium of Cyberspace Safety and Security*, 150-170.
- Kuluttaja-asiamies. 2016. *Operaattorilla selkeä vastuu liittymälaskulla veloitettavista ostoksista*. Uutiskirje 2/2016.
- Nilesh A. Lal, Salendra Prasad and Mohammed Farik. 2015. A review of authentication methods. *International Journal of Scientific & Technology Research* 4, 8, 246-249.
- Xingxing Li, Maorong Ge, Xiaolei Dai, Xiaodong Ren, Mathias Fritsche, Jens Wickert and Harald Schuh. 2015. Accuracy and reliability of multi-GNSS real-time precise positioning: GPS, GLONASS, BeiDou, and Galileo. *Journal of Geodesy* 89, 6, 607-635.
- Harvey P Masters. 2014. *Location-Based Services : Developments and Privacy Issues*. Nova Science Publishers, Inc.
- Anwasha Mukherjee and Debashis De. 2016. Location management in mobile network: A survey. *Computer Science Review* 19, 1-14.

PR Newswire. 2015. *Leading Middle Eastern Bank First to Adopt Finsphere's Mobile Identity Solution ; Emirates NBD to offer Visa's Mobile Location Confirmation (MLC) Service.*

Ramjee Prasad and Marina Ruggieri. 2005. *Applied satellite navigation using GPS, GALILEO and augmentation systems.* Artech House.

Tietosuojavaltuutetun [www-sivusto](http://www.tietosuoja.fi). *Paikantaminen työelämässä.* < www.tietosuoja.fi >. Katsottu 2.4.2018.

Laskennallinen luovuus ja luovat ohjelmat

Henna Lehto

Tiivistelmä.

Laskennallinen luovuus on uusi ja vähän tutkittu tietojenkäsittelytieteen ala ja yksi tekoälyn alalaji. Tekoäly on tietokoneen toimintaa, jota voidaan luokitella inhimilliseksi. Luovat ohjelmat liittyvät vahvasti tekoälyyn, sillä ne osaavat esimerkiksi kirjoittaa tarinoita, maalata maalauksia tai säveltää musiikkia, mitkä kaikki ovat ihmisille ominaisia taitoja. Tässä tutkielmassa tutkitaan ja vertaillaan luovuuden ja laskennallisen luovuuden käsitteiden eri määritelmiä sekä esitellään erilaisia lähestymistapoja luovien ohjelmien tekemiseen.

Avainsanat ja -sanonnat: Laskennallinen luovuus, luovat ohjelmat, luovuus, assosiaatio, Markovin prosessi.

1. Johdanto

Coltonin ja Wigginsin [2012] mukaan luovia ihmisiä sekä heidän panostustaan kulttuuriseen edistymiseen arvostetaan korkeasti. Luovuutta pidetään yhtenä asiana, joka tekee ihmisestä ihmisen, ja siksi sitä pidetään suuressa arvossa ja yritetään suojella huolella. Toivomme, että meidän luova käyttäytymisemme olisi enemmän opittua kuin ohjelmoitua [Wiggins *et al.* 2009], joten luovuuden opettaminen tietokoneohjelmille tulisi näin ollen tapahtua mekaanisen ohjelmoinnin sijaan toteuttamalla tietokoneille kyky oppia luovuutta.

Luovat ohjelmat voivat olla hyödyllisiä erilaisissa laboratorioissa tai kaupallisella alalla [Boden 1998], sillä esimerkiksi luovuuden testaamisen on ehdotettu laajentavan tietokoneen kykyä esiintyä ihmisenä mittaavaa Turingin testiä [Turing 1950] tai jopa korvaavan sen [Varshney *et al.* 2016]. Lisäksi jo nyt on kehitetty muusikoita auttavia ohjelmia kappaleiden nimeämiseen tai inspiraation etsimiseen [Settles 2010].

Laskennallisen luovuuden tutkimuksen tavoitteena on mallintaa, simuloida tai kasvattaa luovuutta käyttämällä laskennallisia menetelmiä. Ennalta ohjelmoituja järjestelmiä kritisoidaan usein siitä, että ne tekevät vain sitä, mitä niitä on käsketty tekemään, joten yksi laskennallisen luovuuden kritiikin aiheista on ohjelmien oma-aloitteisuuden puute. Tässä kohtaa kuitenkin tiedonlouhinnan ja koneoppimisen metodit voivat auttaa. [Toivonen and Gross 2015.] Tiedonlouhinta on tiedon analysointia, jonka tarkoitus on löytää entuudestaan tuntemattomia suhteita sekä tiivistää tietoa tavoin, jotka ovat sekä ymmärrettäviä että hyödyllisiä tiedon omistajalle [Hand *et al.* 2001]. Koneoppimisen tarkoitus

on puolestaan saada ohjelma tekemään uusia asioita sille syötetyn datan perusteella ilman, että sitä ohjelmoidaan tekemään kyseistä toimintaa. Koneoppimisessa siis ohjelmaa yritetään opettaa tekemään uutta. Luovien ohjelmien toimintaa saadaan laajennettua merkittävästi tiedonlouhintaa ja koneoppimista hyödyntämällä, sillä näin tietokoneiden kyky tehdä ohjelmoijastaan riippumattomia itsenäisiä valintoja kehittyisi. Lisäksi tietokoneet ovat usein tarkempia kuin ihmiset, joten tiedonlouhintaa ja koneoppimista käyttämällä järjestelmät voivat oppia sellaisia yksityiskohtia, joita yksittäiset ihmiset eivät olisi välttämättä osanneet niille opettaa.

Laskennallinen luovuus liittyy vahvasti tekoälyyn ja sen tutkimukseen. Colton ja Wiggins [2012] kuitenkin huomauttavat, että tekoälyn harjoittaminen painottuu valtaosin ongelmanratkaisuun – älykäs tehtävä, jonka haluaisimme ohjelman tekevän, muodostetaan tietynlaiseksi ratkaistavaksi ongelmaksi. He kuitenkin painottavat, että olisi sopimatonta kuvailla sonaatin tuottamista tai valokuvasta maalaamista ongelmaksi, joka pitäisi ratkaista. Ongelmanratkaisupohjainen näkökulma voisi olla kuitenkin hyödyllinen tietyissä kohtaa artefaktien, eli luovien tuotosten kuten esimerkiksi maalausten ja sinfonioiden, tuottoa, sillä esimerkiksi runoa kirjoittaessa ongelma voisi olla kahden keskenään sointuvan sanan löytäminen.

Tässä tutkielmassa esitellään ensin luovuuden määritelmiä, jonka jälkeen tutkitaan laskennallista luovuutta tutkimusalana. Lopuksi esitellään yksityiskohtaisemmin muutamien luovien ohjelmien toimintaa.

2. Luovuus

Jotta laskennallista luovuutta voidaan tutkia käsitteenä sekä tieteenalana, tulee ensin ymmärtää luovuuden käsite. Luovuus on inhimillinen taito, ja sen määrittäminen tai tutkiminen on haastavaa, sillä ihmisen mieltä ja aivoja on erittäin hankala tutkia [Wiggins *et al.* 2009]. Bodenin [1998] mukaan arvot, ja näiden myötä myös luoviksi määrittelemämme tuotokset, muuttuvat kulttuurin ja ajan mukana. Esimerkiksi 1400-luvun todenmukaisina ihmiskuvina ilmenevän renessanssin ajan mielipide taiteesta on hyvinkin poikkeava verrattuna 1900-luvun alun surrealismiin, joka on abstraktia taidetta. Lisäksi tavoitteelliset ja tunteelliset tekijät vaikuttavat usein siihen, miksi pidämme tai emme pidä joistain asioista, eikä nykyinen tekoäly osaa käsitellä näitä tekijöitä [Boden 1998].

2.1 Luovuuden määritelmiä

Luovuudelle ei ole olemassa yhtä tarkkaa määritelmää, sillä kyseessä on hankalasti mitattava taito ja henkilökohtainen ominaisuus. Gervás [2009] huomauttaakin, että jos yksittäinen puhuja mainitsisi luovuuden käsitteen yleisön edes-

sä, todennäköisesti jokaiselle kuulijalle syntyisi erilainen mielikuva asiasta. Tämä ei tosin tarkoita, etteikö määritelmiä luovuudelle olisi silti olemassa.

Bodenin [1998] mukaan luova ajatus on uudenlainen, yllättävä sekä sen mielenkiintoisuus, hyödyllisyys, kauneus tai muu vastaava piirre tekee siitä arvokkaan. Settles [2010] puolestaan määrittelee luovuuden kyvyksi tehdä johdopäätöksiä jo olemassa olevien ideoiden, sääntöjen ja mallien asettamien rajojen yli sekä luoda uusia, yhtä merkityksellisiä ajatusmalleja. Varshney ja muut [2016] ovat määritelleet luovuuden samankaltaisesti, sillä heidän mukaansa luovuus on kyky poikkeavaan ajattelutapaan uusien ideoiden luomiseksi. Luovuus on ajatus tai artefakti, joka on tunnustettu uudenaikaiseksi ja sen lisäksi se on sovelias, tarpeellinen tai arvokas kyseisen luovuuden alan sosiaaliselle ryhmälle.

Näistä kolmesta määritelmästä voidaan todeta, että luovuus on sekä hyödyllistä ja arvokasta että yllättävää ja ennennäkemätöntä. Kyseisten piirteiden samanaikainen toteuttaminen on haastavaa, ja Colton ja Wiggins [2012] toteavatkin luovuuden vaativan kaikkien älykkyyden osa-alueiden yhtäaikaista toimintaa. Lisäksi uutuutta, yllätyksellisyyttä ja arvoa on hankala määrittää objektiivisesti sekä yllättävyys ja uutuus ovat myös päällekkäisiä vaatimuksia. Idea on harvoin yllättävä ja arvokas, ellei se ole uudenlainen. [Toivonen and Gross 2015.]

2.1 Luovuuden luokittelut

Kuten todettu, tarkan ja yksiselitteisen määritelmän löytäminen luovuudelle on hankalaa, joten luovuutta pyritään ymmärtämään paremmin kategorisoimalla sitä alaluokkiin.

Yksi yleinen keino on jakaa luovuus kahteen osaan Bodenin [1998] määrittelemien P-luovuuden (psykologinen luovuus) ja H-luovuuden (historiallinen luovuus) mukaan. H-luovuudeksi määritellään sellaiset luovat ajatukset, joita ei ole koskaan ennen ilmennyt kenellekään muulle ihmishistorian aikana. P-luovuudeksi lasketaan sellaiset luovat ajatukset, jotka eivät ole aikaisemmin ilmenneet kyseisellä henkilöllä. Bodenin [1995] mukaan luovuutta ovat myös siis sellaiset ajatukset, jotka muut ihmiset ovat aikaisemmin jo keksineet, sillä ehdolla, että kyseinen ajatus on sellainen, jota kyseinen yksilö ei ole itse aikaisemmin voinut saada. Toisaalta tämä rikkoo yhtä luovuuden useissa määritelmässä esiintynyttä ominaisuutta eli ajatuksen uutuutta. Jos yksilö ryhtyy toteuttamaan samaansa luovaa ideaa, jonka muut ovat jo toteuttaneet, imitaation ja matkimisen riski on suuri. Imitaatio ei ole omintakeista tai luovaa toimintaa [Toivonen and Gross 2015]. Voidaan kuitenkin ajatella, että jos henkilö ei ole voinut saada kyseistä ajatusta koskaan ennen, hän ei ole tietoinen muiden jo

saamista samanlaisista ajatuksista, jolloin mahdollinen matkiminen on tahaton-ta.

Boden [1998] on luokitellut P-luovuuden kolmeen alaluokkaan: *yhdistelevä* (combinational), *tutkiva* (exploratory) ja *muuttuva* (transformational) luovuus. Yhdistelevä luovuus vaatii uudenlaisen, epätodennäköisen yhdistelmän jo aiemmin tunnettuja ideoita. Tutkiva luovuus edellyttää, että uudet ideat syntyvät käsitteellisen avaruuden tutkimisen seurauksena. Käsitteellinen avaruus on joukko käsitteitä, joiden on todettu olevan hyväksyttäviä esimerkkejä kaikesta, mitä luodaan [Wiggins *et al.* 2009]. Muuttuva luovuus on luovuutta, joka syntyy yhdestä tai useammasta ulottuvuuden muutoksesta, jolloin ennennäkemättömät rakenteet voivat muodostua [Boden 1998]. Wiggins ja muut [2009] kuvailevat tutkivan luovuuden olevan tutkimuksesta seuraavaa luovaa prosessia, kun taas muuttuva luovuus on olemassa olevien sääntöjen muokkaamisprosessi. Toisin sanoen, tutkiva luovuus syntyy olemassa olevien tietojen tutkimisesta, ja muuttuvassa luovuudessa näiden tietojen muutos johtaa luovuuteen.

Laskennallinen luovuus on lähes aina P-luovuutta. Lisäksi tietokoneet ovat tuottaneet esimerkkejä kaikista kolmesta P-luovuuden alalajista, mutta kaikkein onnistuneimmat lopputulokset ovat yleensä tutkivaa luovuutta. [Boden 1998.] Laskennallista luovuutta ja sen ominaisuuksia esitellään tarkemmin seuraavassa luvussa.

2.2 Luovuus ja satunnaisuus

Tieteellisten alojen ihmiset yrittävät usein määritellä luovuuden uudenlaiseksi yhdistelmäksi vanhoja ideoita. Siinä tapauksessa luovan asian yllätyksellisyys johtuisi vain siinä käytetyn satunnaisen yhdistelmän epätodennäköisyydestä ja tässä näkökulmassa luovuus olisi vain arvaamattomuutta. [Boden 1995.] Näin ollen myös yhdistelevää luovuutta voitaisiin pitää vain satunnaisuuteen perustuvana ominaisuutena, vaikka Toivonen ja Gross [2015] huomauttavat, ettei satunnainen käytös ole luovaa.

Satunnaisuudella ja arvaamattomuudella on kuitenkin merkittävä osa luovuudessa, kuten luovuuden määritelmistä kävi ilmi. Bodenin [1995] mukaan luovuudessa on silti myös paljon rajoitteita ja sääntöjä eli satunnaisuuden vastakohtia. Luovuudessa yhdistyy rajoitteet ja arvaamattomuus sekä tuttuus ja yllätyksellisyys. Voidaankin siis todeta, että satunnaisuus on erittäin tärkeä piirre luovuudessa, mutta se ei yksinään riitä täyttämään luovuuden määritelmää.

Ihmisen harjoittellessa tiettyä taitoa tämä taito kehittyy ja yksilöstä tulee taitava taidon harjoittamisessa käytetyssä ajattelutavassa. Harjoitteluvaiheessa on vielä hankala ennustaa, kuinka henkilö tulee tekemään tiettyjä valintoja, mutta

vuosien harjoittelun jälkeen hänen ”satunnaiset” valintansa voivat olla helposti ennustettavia. [Boden 1995.] Vaikka kokeneempien tekijöiden valinnat ovat vähemmän satunnaisia, se ei kuitenkaan tarkoita sitä, etteivätkö he voisi silti olla luovia. Kokeneet ovat todennäköisesti enemmän kykeneviä esimerkiksi yhdistelevän luovuuden harjoittamiseen kuin noviisit, sillä voidaan olettaa, että kokeneen henkilön tuntemus vanhoista yhdistettävistä ideoista on suurempi kuin aloittelijan. Tässä tulee huomioida, että tieto omista aikaisemmista ajatuksista lasketaan myös vanhoihin ideoihin, jolloin kokeneella on väistämättä enemmän tietoa kuin vasta-alkajalla.

Kokemus ja kokemattomuus eivät kuitenkaan aina määritä luovuuden laatua. Boden [1995] vertaa luovien valintojen tekoa kaaokseen, jossa täysin lainalaiset järjestelmät (kuten kokeneen ihmisen mieli) voivatkin olla täysin arvaamattomia. Luovia ajatuksia ei voi ennustaa yksityiskohtaisesti, eikä siinä koskaan tulla onnistumaankaan, koska ihmisen mieli on liian omalaatuinen siihen. [Boden 1995.] Luovuus on siis osittain satunnaisuutta, mutta satunnaisuus ei ole yksinään luovuutta.

3. Laskennallinen luovuus

Artefaktien tuottaminen mekaanisesti ei ole uusi keksintö, sillä esimerkiksi Mozart kehitti aikanaan noppapelin, jonka tarkoitus oli säveltää valsseja. Pitkästä historiastaan huolimatta laskennallinen luovuus on vasta vähän aikaa sitten kehittynyt itsenäiseksi, kansainvälisesti tunnustetuksi tutkimusalaksi, jolla on oma identiteetti ja tutkimuskysymykset. [Toivonen and Gross 2015.]

3.1 Laskennallisen luovuuden ja sen tutkimuksen määritelmä

Laskennallisen luovuuden määritelmät vaihtelevat siinä missä luovuuden määritelmätkin. Varshney ja muut [2016] määrittelevät laskennallisen luovuuden olevan tietokoneiden käyttöä sellaisten ideoiden tai artefaktien luomiseen ja valitsemiseen, jotka voitaisiin arvioida luoviksi, jos ne olisivat ihmisen tekemiä. Määritelmässä siis korostetaan sitä, että tietokoneiden tuottamien artefaktien tai uusien ajatusten tulee olla samalla tavalla luovia kuin ihmisten tuotokset, jotta ne voitaisiin määritellä luovaksi. Myös Colton ja muut [2009] määrittelevät laskennallisen luovuuden sellaisiksi ohjelmiksi, jotka ilmentävät käytöstä, jonka voisi laskea luovaksi ihmisten tekemänä. Tietokoneen tuottamien artefaktien luovuuden arviointi ihmisten luovuuteen vertaamalla voi kuitenkin olla väärin. Laskennalliset järjestelmät eivät ole ihmisiä, eikä niiden tuottama luovuus tule koskaan olemaan täysin samaa kuin ihmisen. Järjestelmien saavuttama luovuus voi olla luovuutta, mutta ei samalla tavalla kuin me miellämme sen. [Colton and Wiggins 2012.]

Tutkimusalana laskennallisen luovuuden tavoite on mallintaa tai simuloida luovuutta ja kehittää ihmisen luovuutta käyttämällä laskennallisia menetelmiä [Toivonen and Gross 2015]. Colton ja Wiggins [2012] puolestaan kuvaavat laskennallisen luovuuden tutkimuksen perustuvan sellaisten vastuullisten tietokonejärjestelmien tutkimukseen, joiden käytös on puolueettomien arvioijien mielestä luovaa. Vastuullisuudella tarkoitetaan esimerkiksi järjestelmän kykyä kehittyä tai kykyä käyttää esteettisiä keinoja itsetuottamien artefaktien arvioimiseksi. Colton ja Wiggins [2012] korostavat myös arvioijien puolueettomuutta oikeudenmukaisessa arvioinnissa. Arvioinnilla ei tarkoiteta ideoiden tai artefaktien arvojen arvioimista, vaan arvioijat tarkastelevat luovien toimien ja niiden tulosten vaikutusten arviointia. Boden [2004] puolestaan määrittelee laskennallisen luovuuden neljällä tutkimuskysymyksellä, joihin sen tutkimus pyrkii vastaamaan: (1) Voivatko laskennallisesti tuotetut ideat auttaa meitä ymmärtämään ihmisen luovuutta? (2) Voiko tietokone tehdä asioita, jotka edes vaikuttavat olevan luovia? (3) Voiko tietokone tunnistaa luovuutta? (4) Voivatko tietokoneet olla itsenäisesti luovia?

On huomattavaa, että mikään edellä esitetyistä tutkimuskysymyksistä ei pohdi alan eettisiä kysymyksiä, kuten tietokonejärjestelmien tai niiden arvioijien vastuuta, jota Colton ja Wiggins [2012] määritelmässään painottavat. Voidaan siis todeta, että laskennallisen luovuuden tutkimuksenkin määritelmät eroavat toisistaan merkittäväillä osa-alueilla, mutta tämä voi johtua tutkimusalan tuoreudesta.

3.2 Kritiikkiä

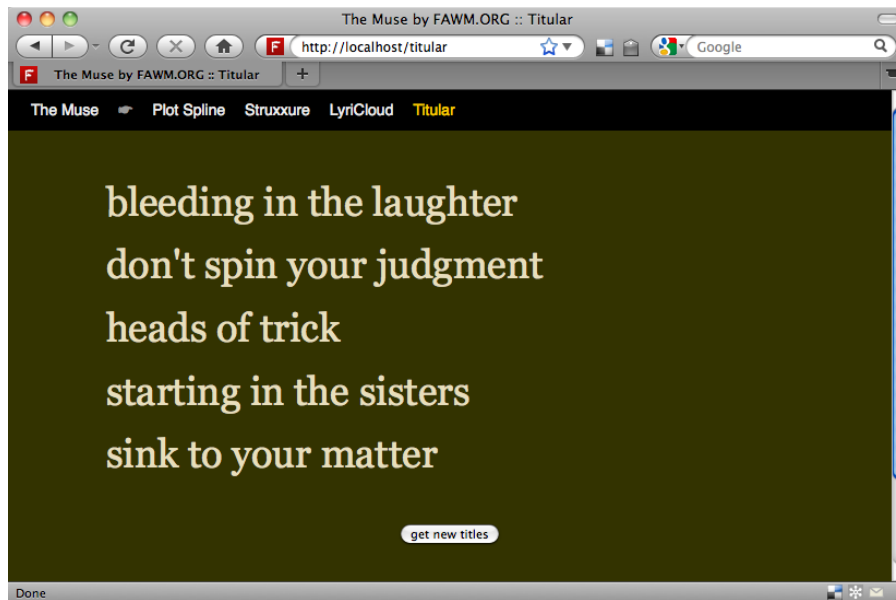
Suuri laskennallisen luovuuden aiheuttama kysymys on se, voiko tietokoneelle antaa kunniaa ajatusten aidosta keksimisestä. Ennalta ohjelmoituja järjestelmiä kritisoidaan tekevän vain sitä, mitä sitä käskettiin tekemään. Näin ohjelmilla olisi erittäin vähän, jos yhtään, luovuutta. [Toivonen and Gross 2015.] Argumentti ei kuitenkaan pidä paikkaansa, sillä koneoppimisen ja tiedonlouhinnan myötä tietokone ei ole enää rajoitettu tekemään vain, mitä sen ohjelmoija sitä käskää [Wiggins *et al.* 2009]. Cardoso ja muut [2009] ehdottavat, että ratkaisu luovien järjestelmien omintakeisten tuotosten syntyyn olisi opettaa järjestelmiä oppimaan, mitä ne tekevät, ennen kuin yritetään tehdä se luovasti. Toisaalta luovien alojen ihmisetkin ovat (yleensä) ensin joutuneet opettelemaan taitonsa ennen kuin ovat yrittäneet olla siinä luovia. Esimerkiksi jazzpianisti on todennäköisesti ensin opetellut soittamaan pianoa ennen kuin on yrittänyt haastavaksi mutta luovaksi ajatellun jazzimprovisaation soittamista. On kuitenkin huomioitava, että tietokone ei ole ihminen, joten järjestelmän inhimillinen opettaminen ei ole välttämättä mahdollista.

Tietokoneita voidaan syyttää myös imitoinnista tai matkimisesta, jotka eivät ole aikaisemmin esitettyjen luovuuden määritelmien mukaan luovuutta. Jos tietokoneelle opetetaan luovuutta vain näyttämällä olemassa olevia ihmisten tekemiä artefakteja, se ryhtyy helposti matkimaan niitä [Toivonen and Gross 2015]. Lisäksi, jos ihmisen luomat artefaktit pärjäävät parhaiten vertailutesteissä, kone voi oppia matkimaan parhaiten pärjänneiden teosten tyylejä tuottaakseen yhtä laadukkaan tuloksen [Pease and Colton 2011]. Ohjelmalle on kuitenkin pakko antaa dataa oppimista varten, jolloin ohjelmalla tuotetun artefaktin arvioiminen joko luovaksi koneen omaksi ratkaisuksi tai kopioiduksi pastissiksi on vaikeaa. Eron voi kuitenkin löytää palaamalla Bodenin [1998] yhdistelevän luovuuden määritelmään, jonka mukaan yhdistelevä luovuus on uudenlainen, epätodennäköinen yhdistelmä jo aiemmin tunnettuja ideoita. Eli järjestelmä saa hyödyntää olemassa olevien artefaktien ominaisuuksia omassa tuotoksessaan, mutta sen tulee yhdistää oppimansa tieto yhdeksi uudeksi, ennennäkemättömäksi kokonaisuudeksi. Tällöin tuotosta ei voida laskea matkituksi.

Palataan myös aikaisemmin esiteltyyn satunnaisuuden ja luovuuden suhteeseen. Suosittu tapa toteuttaa laskennallista luovuutta on laskemalla mahdollisten valintojen todennäköisyyksiä suhteessa muihin valintoihin. Todennäköisyyksiin luotettaessa on kuitenkin suuri riski tuottaa mitäänsanomaton, yllätyksetön ja tylsä tulos. Settles [2010] mainitsee, kuinka laskettaessa englanninkielisten sanojen todennäköisiä yhdistelmiä lauluissa ohjelman yleisimmät valinnat olivat muun muassa sanat *love*, *heart* ja *baby*, jotka ovatkin usein käytettyjä ja yllätyksettömiä sanavalintoja. Wiggins ja muut [2009] toteavat vastaavasti erittäin todennäköisten teosten olevan yleensä syntaktisesti oikein, mutta (muusiikillisesti) tylsiä.

3.3 Esimerkkejä luovista ohjelmista

Vaikka laskennallinen luovuus on käsitteenä ja tutkimusalana tuore, on silti olemassa lukuisia ohjelmia, jotka yrittävät täyttää luovuuden määritelmää tuloksellaan. Yksi tällainen ohjelma on laulun nimiä ehdottava Settlesin vuonna 2010 julkaisema Títular. Títular opetettiin muodostamaan nimiä antamalla sille yli satatuhatta laulua, jonka jälkeen sille opetettiin sanojen tyyppin tunnistamista sekä niiden mahdollisia yhdistelmiä. [Settles 2010.] Kuvassa 1 on kuvankaappaus Títularin käyttöliittymästä, joka näyttää käyttäjälleen aina viisi nimiehdokasta kerrallaan.



Kuva 1. Titularin käyttöliittymä [Settles 2010]

Toinen esimerkki luovuuteen pyrkivästä ohjelmasta on dekkareita kirjoittava Kleinin ja muiden [1973] tekemä Novel Writer. Novel Writer luo murhatarinoita, joiden miljöö sijoittuu viikonloppuna järjestettyihin juhliin. Sen toiminta pohjautui simulaatioon, jossa hahmojen käyttäytyminen ja mahdolliset tapahtumat määrättiin todennäköisyyslaskennan avulla. [Gervás 2009.] Alla on esimerkki tämän varhaisen tarinankertojaohjelman tuotoksesta:

“The day was Monday. The pleasant weather was sunny. Lady Buxley was in the park. James ran into Lady Buxley. James talked with Lady Buxley. Lady Buxley irted with James. James invited Lady Buxley. James liked Lady Buxley. Lady Buxley liked James. Lady Buxley was with James in a hotel. James caressed Lady Buxley with passion. James was Lady Buxley’s lover. Marion following saw the affair. Marion saw the affair. Marion was jealous.” [Gervás 2009.]

Kolmas esimerkki on maalauksia tuottava Simon Coltonin Painting Fool [Painting Fool 2018]. Painting Foolin toiminta perustuu muun muassa tiettyjen elementtien asetelun sääntöihin, kuvien samanlaisuuden tunnistamiseen sekä erilaisiin metodeihin, joilla luodaan abstrakteja taideteoksia sekä kuvien suodattimia. Kuvassa 2 on Paintin Foolin tuottama teos, jonka se teki erään Afganistanin sotaa käsittelevän sanomalehtiartikkelin perusteella. [Colton and Wiggins 2012.]



Kuva 2. Painting Foolin tuottama teos [Painting Fool 2018]

Kuvassa on kollaasi, jossa on pommikone, räjähdys, äiti ja lapsi, tyttö uskonnollisessa päähineessä sekä sotahautoja [Colton and Wiggins 2012].

Jokainen kolmesta esitellystä ohjelmasta on onnistunut tuottamaan jonkun ihmisen luovuutta imitoivan lopputuloksen. Näiden tuotosten luovuutta ei kuitenkaan arvostella tässä tutkielmassa tarkemmin, sillä arvioijan subjektiivisuus saattaisi vaikuttaa lopputulokseen.

4. Luovien ohjelmien toiminta

Kuten aikaisemmin on mainittu, yleinen tapa toteuttaa luovuutta on laskea todennäköisyyksiä eri perustein. Se, minkä perusteella todennäköisyyksiä laskeaan, vaihtelee. Ohjelma voi esimerkiksi laskea, kuinka todennäköisesti ihminen tekisi asian, mutta kuten aikaisemmassa luvussa on mainittu, seuraamalla vain kaikkein todennäköisimpiä valintoja tulos on helposti tylsä ja muistuttaa aikaisemmin tuotettuja teoksia.

Todennäköisyyslaskennasta voi kuitenkin olla hyötyä laskennallisessa luovuudessa, etenkin osana algoritmia, ei yksinään. Seuraavaksi esitellään kaksi potentiaalista laskennallisen luovuuden toteutustapaa, assosiaatio ja Markovin prosessin sovelluksia, joista jälkimmäinen pohjautuu selkeästi todennäköisyyslaskentaan. Assosiaatiosta puolestaan on olemassa erilaisia variaatioita siitä, kuinka lasketaan kahden asian välinen yhteys, ja todennäköisyyksien laskentaa on luontevaa hyödyntää jollain tapaa näiden yhteyksien laskennassa.

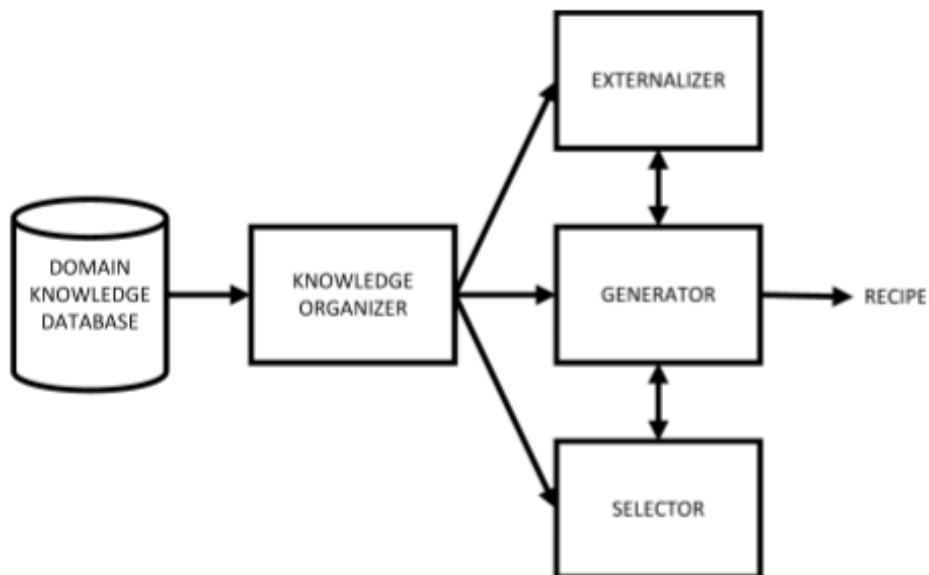
4.1 Assosiaatio

Assosiaatio on alkuperäisen teeman tai ajatuksen laajentamista niin paljon kuin mahdollista [Settles 2010]. Assosiaatio tarkoittaa yhteyttä kahden tai useamman

asian välillä ja se on tyypillistä ihmisten luovuudessa ja siinä, kuinka ihmisen muistin haun ajatellaan toimivan. Tutkimukset ihmisen luovuudesta ovat tehneet läpimurtoja huomaamalla, kuinka ihmisen oivalluskyky ja uudet näkemykset ovat usein assosiaatioiden tulosta. [Varshney *et al.* 2016.] Esimerkiksi taulua maalatessa taivaan väriksi voi helposti assosoida sinisen värin, sillä maalaaja on todennäköisesti usein nähnyt sinisiä taivaita. Assosiaatio on siis mahdollinen keino yrittää toteuttaa luovuutta tietokoneilla.

Varshney ja muut [2016] kehittivät ruokareseptejä luovan ohjelman hyödyntäen assosiaatiota. Uuden reseptin ainesosien etsiminen etenee seuraavasti: Aloitetaan yhdestä pienestä asiasta ja suunnitellaan päämäärää, kuten ruuan uudenlaisuutta, makua, ainesosien tasapainoisuutta tai hintaa. Seuraavaksi rakennetaan raaka-aineista lista lisäämällä iteratiivisesti komponentteja, jotka assosioituvat jo aikaisemmin valittujen kanssa ja sopivat suunniteltuun päämäärään. Kulinaarisessa luovuudessa assosiativiset suhteet voivat syntyä tuotteiden yleisestä käytöstä yhdessä, samankaltaisesta mausta, yhteisestä sadonkorjuuajankohdasta tai jopa samankaltaisesta nimestä. Lopuksi harkitaan uudenlaisten artefaktien lisäämistä jo olemassa oleviin artefakteihin. [Varshney *et al.* 2016.] Esimerkkinä, valitaan ensin yksi ainesosa, olkoon se peruna. Valitaan päämäärä, joka tässä tapauksessa voisi olla halpa pääruuan lisuke. Ryhdytään etsimään komponentteja yhteensopivuuden perusteella ja saadaan ensimmäiseksi tulokseksi voi. Jatketaan komponenttien etsimistä ja saadaan muun muassa kerma ja valkosipuli uusiksi ehdotuksiksi. Kun ei löydetä enää kriteereitä täyttäviä komponentteja, etsintä lopetetaan.

Kuvassa 3 havainnollistetaan ohjelman toimintaa kokonaisuutena.



Kuva 3. Lohkokaavio reseptejä tuottavasta ohjelmasta [Varshney *et al.* 2016]

Tietokanta koostuu kulinartistisen alan tiedosta, sisältäen tiedon tyyleistä, mausta, johdonmukaisuuksista, yhdistelmistä, kehityksestä, alueellisuudesta, kulttuurista ja ruuanlaittotavoista. Kaikki algoritmin kolme pääelementtiä, *ilmentäjä* (externalizer), *generaattori* (generator) ja *valitsija* (selector), saavat syötteensä kategorisoidusta tietokannasta. Generaattori tuottaa uusia ehdotuksia ainesosalistaan etsien inspiraatiota kannasta, ilmentäjä määrää ainesosien määrän ja reseptin työvaiheet tietokannassa jo olemassa olevien reseptien perusteella ja valitsija arvioi suunnitellun idean luovuutta vertaamalla uutta reseptiä kannassa oleviin sekä valittujen raaka-aineiden ominaisuuksiin. Valitsija mallintaa ihmisen havaintokykyä, makuaistia ja kulttuuria sekä arvioi uusien ideoiden uutuutta ja laatua. Näiden piirteiden mittaaminen on määritelty tietojoukkojen perustassa olevilla tiedoilla kulinartistisista lajeista, kulinaariseen tieteseen liittyvistä tiedoista ja kokeilullisen datan hedonisesta psykofysiikasta. [Varshney *et al.* 2016.] Aikaisemmin esitetyn perunareseptin tapauksessa generaattori etsii ainesosat (voi, kerma, valkosipuli), jonka jälkeen valitsija arvioi ainesosien laadun ja lopuksi ilmentäjä kokoaa reseptin työvaiheet. Tässä tapauksessa tulee tosin huomioda, että mainittujen raaka-aineiden yhdistelmä ei ole välttämättä kovin uudennlainen tai luova, joten algoritmi ei todennäköisesti tuottaisi vastaavaa tulosta.

Reseptiä tuottavan ohjelman toimintatapaa voisi hyödyntää reseptien lisäksi myös muiden luovuutta vaativien artefaktien luomisessa, esimerkiksi taulun tekemisessä. Tällöin generaattori etsisi taulun aiheen, jonka laadun valitsija arvioisi ja ilmentäjä määräisi taulun tekovaiheet ja -tavat. Tässä esimerkissä kyläkin ilmentäjällä on suurin osa teoksen tuottamisessa ja sen toimintaa tulisi muokata ja tarkentaa juuri taulun tekemisen mukaiseksi.

Assosiaatiota hyödyntää myös Settlesin [2010] muusikoiden inspiraation avuksi tarkoitettu LyriCloud, joka ehdottaa käyttäjän syöttämään, niin sanottuun *siemensanaan* (seed word), liittyviä muita sanoja. Siemensanaan liittyvät sanat perustuvat metaforiin, synonyymeihin ja sanojen väliseen sointuvuuteen. Esimerkiksi jos ohjelmalle syöttää sanan *dream*, se näyttää muun muassa sanat *american, bad, smithereen* ja *nightmare*. [Settles 2010.] Ohjelma näyttää joka kerta erilaisen joukon siemensanaan sopivia sanoja, vaikka syötetty sana olisi sama.

Kuten aikaisemmin esitellyssä reseptien kokoajassa, LyriCloudin toiminta perustuu toimintaan, jossa yhdestä komponentista saadaan laajempi, yhteensopiva kokonaisuus erilaisia komponentteja. Sanojen välinen assosiaatio laskeaan LyriCloudissa kaavan avulla, joka määrittää kahden sanan keskeisen yhteensopivuuden. LyriCloud laskee sille syötetylle siemensanalle *s* yhteensopivuuden joka toiselle sanalle *w*, joka on ohjelman sanastossa, käyttäen seuraavaa kaavaa:

$$\text{sim}(s, w) = (1 + \log c(s, w)) * \log \frac{N}{u(w)},$$

jossa $c(s, w)$ on numero, joka kertoo, kuinka useasti s ja w esiintyvät samalla rivillä tietokannassa olevassa lyriikassa, $u(w)$ on uniikkien sanojen lukumäärä, joiden kanssa w ilmenee millä tahansa rivillä ja N on tietokannassa olevan sanaston koko. [Settles 2010.]

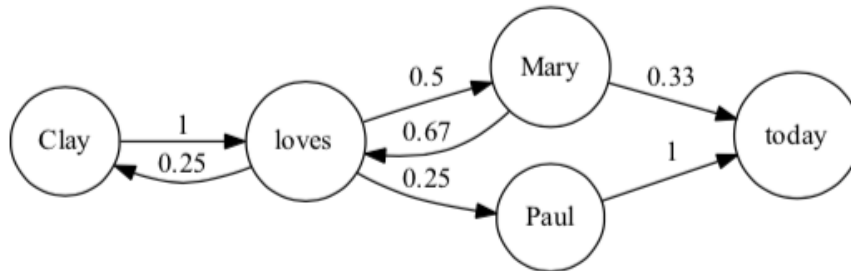
Settlesin [2010] mukaan käyttäjät eivät kuitenkaan pitäneet LyriCloudista kahdesta syystä: ohjelman eniten suosittelat sanat olivat paljon käytettyjä ja tavallisia (esimerkiksi *love, baby, give, true, sweet, song, girl*) tai käyttäjät eivät nähneet yhteyttä kaikkien sanojen välillä. Ensimmäinen ongelma on yksi yleinen laskennallisen luovuuden jo tässäkin tutkielmassa esitelty haaste, mutta jälkimmäinen ongelma voi olla enemmän subjektiivinen, sillä siinä missä toiset kokevat ehdotettujen sanojen olevan epäolennaisia syötesanaan nähden, toiset voivat nähdä sanojen yhteyden ja inspiroitua siitä.

4.2 Markovin prosessin sovelluksia

Barbierin ja muiden [2012] mukaan Markovin prosessi on satunnainen prosessi, joka luo vaiheita, joiden todennäköisyys riippuu äärellisestä määrästä aikaisempia vaiheita. Markovin prosessia voidaan hyödyntää esimerkiksi lauseen seuraavan sanan etsimisessä, jolloin seuraavan vaiheen, eli sanan, valinta perustuu edellä esiintyviin sanoihin. Näin Markovin prosessia voi hyödyntää laskennallisessa luovuudessa muun muassa tarinankerrontaan, runouteen tai laulun sanoittamiseen. Markovin prosessissa käytetään usein satunnaiskulkua ja n -grammia, joka sanojen etsimisen tapauksessa kuvaa yhtenäistä sanojen sarjaa, jossa n on sanojen lukumäärä [Barbieri *et al.* 2012]. *Satunnaiskulku* (random walk) puolestaan perustuu seuraaville säännöille: (1) On olemassa alkupiste. (2) Etäisyys pisteestä toiseen on vakio. (3) Kaikki suunnat ovat yhtä todennäköisiä. Siirtymä on täsmälleen yhden askeleen verran, paikallaan pysyminen tai pidemmät askeleet eivät ole sallittuja. [Wikipedia 2018.] Sanoja etsittäessä säännöt siis tarkoittavat sitä, että lause alkaa aina jostain ja kaikki sanat ovat yhtä todennäköisiä, sillä niiden välinen etäisyys, eli esimerkiksi yhteensopivuuden todennäköisyys, on sama. Lisäksi lauseeseen ei voi lisätä kuin yhden sanan kerrallaan, sillä useammat sanat tai paikallaanpysyminen eivät ole luvallisia.

Markovin prosessin yksi sovellus on *Markovin ketjut* (Markov chains). Settles [2010] testasi Markovin ketjuja Titularin, aikaisemmin esitellyn laulujen otsikkoja tuottavan ohjelman toiminnassa. Otsikon jokainen sana w_i luotiin ehdollisella todennäköisyydellä $P(w_i | w_{i-n}, \dots, w_{i-1})$, jossa n on aikaisemmin esiintyneiden sanojen lukumäärä [Settles 2010]. Tässä siis seuraavaksi valittavan sanan todennäköisyys lasketaan sillä ehdolla, kuinka hyvin se sopisi aikaisemmin esiintyneiden sanojen jatkeeksi. Kuvassa 4 kuvataan Markovin ketjujen

toimintaa viiden sanan välillä. Settles [2012] totesi kuitenkin, että tämä lähestymistapa tuottaa yleensä samanlaisia otsikoita kuin opetusdatassa oli. Lisäksi niissä tapauksissa, kun n -grammilla tehdyt otsikot tuottivat kokonaan uuden tuotoksen, olivat ne yleensä pitkiä ja käsittämättömiä, kuten ”*Too Many A Woman First the Dark Clouds Will It Up*”. Markovin ketjujen käyttäminen vaatii siis lisärajoitteita, jotta lopputulokset voisivat täyttää millään tavalla luovuuden kriteerejä.



Kuva 4. Esimerkki Markovin ketjuista. Numerot kertovat valinnan todennäköisyyden sanasta nuolen osoittamaan toiseen sanaan. [Barbieri *et al.* 2012.]

Barbieri ja muut [2012] kehittivät lyriikan tuottoa varten Markovin prosessiin perustuvat Markovin rajoitteet. Markovin prosessi M voidaan muuttaa rajoitetuksi Markovin prosessiksi \tilde{M} seuraavin toimenpitein:

1. \tilde{M} luo vain säkeistöt, jotka toteuttavat kontrolloidut rajoitteet.
2. Kelvolliset säkeistöt tehdään samoilla todennäköisyyksillä niin prosessissa M kuin prosessissa \tilde{M} riippuen rajoittavista tekijöistä.

Lyriikan tuoton rajoittavia tekijöitä ovat riimi, tahti ja syntaktinen oikeellisuus. [Barbieri *et al.* 2012.] Markovin rajoitteita käyttäen tuotettiin seuraava runo:

*There is a note in his eyes
He backs the bat of the key
Down the song in my eyes
You back the beat of the sea*
[Barbieri *et al.* 2012].

Markovin rajoitteisiin perustuvan toiminnan voidaan huomata tuottavan selkeästi johdonmukaisempia tuloksia kuin pelkkä Markovin ketjujen käyttö. Barbieri ja muut [2012] toteavat, että menetelmää saisi vielä kehitettyä entisestään laajentamalla ja tarkentamalla rajoitteiden määritelmiä. Tällöin Markovin rajoitteiden käyttöä voisi laajentaa ja sillä tehdyt artefaktit voisivat olla hyvinkin lähellä ihmisen luovaksi luokittelemia teoksia.

5. Yhteenveto

Tutkielman alussa käsiteltiin luovuuden eri määritelmiä sekä näiden perusteella todettiin luovuuden olevan hyödyllistä ja arvokasta sekä yllättävää ja ennen-

näkemätöntä. Todettiin myös, että nämä neljä adjektiivia ovat helposti päällekkäisiä. Huomattiin myös, että satunnaisuus on hyvin oleellinen osa luovuutta, mutta pelkkä satunnaisuus ei riitä luovuuteen. Luovuuden määritelmien yhteydessä tutkittiin myös Bodenin [1998] historiallisen ja yhdistelevän luovuuden käsitteitä ja todettiin tähän asti toteutettujen laskennallisesti luovien ohjelmien olevan enemmän yhdistelevää luovuutta.

Laskennallisen luovuuden määritelmiä tutkiessa huomattiin erilaisia näkemyksiä alasta, sillä osa määritelmistä vertasi tietokoneen luovuutta ihmisen luovuuteen ja Coltonin ja Wigginsin [2012] määritelmässä pohdittiin käsitettä enemmän eettisyyden kannalta. Lisäksi esiteltiin laskennallista luovuutta koskevaa kritiikkiä, joita ovat muun muassa matkimisen vaara, yllätyksettömyys ja alkuperäisen idean keksimisen puute. Luvun lopuksi annettiin esimerkkejä kolmesta luovasta ohjelmasta.

Lopuksi esiteltiin tarkemmin kahta toimintatapaa laskennallisen luovuuden toteuttamiseksi. Assosiaatiota on käytetty onnistuneesti reseptien valmistuksessa, mutta laulun sanojen tekoa inspiroivan ohjelman onnistumisesta oli eriäviä mielipiteitä. Assosiaation lisäksi esiteltiin Markovin prosessin eri sovelluksia, Markovin ketjuja ja Markovin rajoitteita. Voitiin todeta, että Markovin rajoitteet toimivat Markovin ketjuja paremmin, sillä rajoitteita hyödyntäen lopputulos oli ymmärrettävämpi ja muistutti tyyliltään ihmisen tuottamaa runoa.

Laskennallinen luovuus on tutkimusalana alkutekijöissään, mutta tulee todennäköisesti kasvamaan tekoälyn kehittymisen rinnalla. Toisaalta luovat ohjelmat voivat jakaa vahvasti mielipiteitä, sillä luovuus on niin inhimillinen piirre, että sen toteutus tietokoneella on suuri harppaus kohti ihmismäisiä koneita, jotka voivat olla tulevaisuudessa joko uhka tai hyöty.

Viiteluettelo

- Gabriele Barbieri, François Pachet, Pierre Roy and Mirko Degli Esposti. 2012. Markov constraints for generating lyrics with style. In: *Proceedings of the 20th European Conference on Artificial Intelligence*, 115–120.
- Margaret Boden. 1995. Creativity and unpredictability. *Stanford Humanities Review* 4, 2, 123–139.
- Margaret Boden. 1998. Creativity and artificial intelligence. *Artificial Intelligence* 103, 1–2, 347–356.
- Margaret Boden. 2004. *The Creative Mind: Myths and Mechanisms*. Psychology Press.
- Amílcar Cardoso, Tony Veale and Geraint A. Wiggins. 2009. Converging on the divergent: The history (and future) of the international joint workshops in computational creativity. *AI Magazine* 30, 3, 15–20.

- Simon Colton, Ramon López de Mántaras and Oliviero Stock. 2009. Computational creativity: coming of age. *AI Magazine* 30, 3, 11–14.
- Simon Colton and Geraint A. Wiggins. 2012. Computational creativity: the final frontier? In: *Proceedings of the 20th European Conference on Artificial Intelligence*, 21–26.
- Pablo Gervás. 2009. Computational approaches to storytelling and creativity. *AI Magazine* 30, 3, 49–62.
- David J. Hand, Heikki Mannila and Padhraic Smyth. 2001. *Principles of data mining*. The MIT Press.
- Sheldon Klein, John F. Aeschlimann, David F. Balsiger, Steven L. Converse, Claudine Court, Mark Foster, Robin Lao, John D. Oakley and Joel Smith. 1973. Automatic novel writing: A status report. Computer Science Department, The University of Wisconsin.
- Painting Fool. 2018. <http://www.thepaintingfool.com>. Read 27.2.2018
- Alison Pease and Simon Colton. 2011. On impact and evaluation in computational creativity: A discussion of the Turing test and an alternative proposal. In: *Proceedings of the AISB Symposium on AI and Philosophy*, 39–47.
- Burr Settles. 2010. Computational creativity tools for songwriters. In: *Proceedings of the NAACL HLT 2010 Second Workshop on Computational Approaches to Linguistic Creativity*, 49–57.
- Hannu Toivonen and Oskar Gross. 2015. Data mining and machine learning in computational creativity. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5, 6, 265–275.
- Alan Turing. 1950. Computing machinery and intelligence. *Mind* LIX, 236, 433–460.
- Lav R. Varshney, Jun Wang and Kush R. Varshney. 2016. Associative algorithms for computational creativity. *The Journal of Creative Behavior* 50, 3, 211–223.
- Geraint A. Wiggins, Marcus T. Pearce and Daniel Müllensiefen. 2009. Computational Modelling of Music Cognition and Musical Creativity. In: Roger T. Dean (ed.), *The Oxford Handbook of Computer Music*. Oxford University Press, 383–420.
- Wikipedia. 2018. Random walk. https://en.wikipedia.org/wiki/Random_walk. Read 13.4.2018

Search Engine Optimization

Kia Lieke

Abstract.

The number of websites in the internet grows continuously. That is when making them accessible and easily findable comes to the purpose. With the techniques of search engine optimization, website owners and developers can make their websites more usable and easier to find. The contest between websites and search engines is constant and new methods are implemented and monitored to make the website even more visible than before. This thesis is about search engine optimization and its modes of operation. In this thesis I examine what search engine optimization is, how it can be implemented and measured and how marketing can be combined with all of that. In addition, I am going to clarify the usage of search engine optimization methods with some online sneaker store use cases.

Keywords: Search Engine Optimization, Search Engine Marketing, Social Media Optimization, Pay-per-click, Paid Search Marketing

1. Introduction

Search engines have become a major part of people's lives since the amount of new data is growing daily. People use search engines to find products and websites, but a great part of the internet users use search engines to navigate to pages they already know. Websites modified with the methods of search engine optimization are more accessible, findable and visible.

Search engine optimization is nowadays also a way to improve business and get customers. However, it is important to remember that search engine optimization's focus is on the users, not the search engines. [Gudivada *et al.* 2015.] The main goal of search engine optimization is to rank higher in the search engine results page to get more visitors [Lemos and Joshi 2017].

Search engine optimization can be divided into different kinds of methods, which are white hat methods and black hat methods. White hat methods are the allowable ones while using black hat methods is punishable. White hat methods can be divided further into on-page and off-page optimization. On-page optimization techniques are the ones that are implemented during the development in contrast to off-page optimization methods which are implemented after the development [Khan and Mahmood 2018].

When enhancing search engine optimization with marketing techniques such as paid search marketing and pay-per-click, search engine marketing is established. Search engine marketing combines the best components of search engine optimization and marketing to get visitors to a website [Fiorini and Lipsky 2012].

The purpose of this thesis is to find out the methods and usages of search engine optimization, as well as the possible challenges. The theory and practices in this thesis are mainly focused on Google and its algorithm. This thesis is an overview of the usage and methods of search engine optimization, how it differs from search engine marketing and how to measure the impacts.

2. What is search engine optimization?

Search Engine Optimization, also known as SEO, is the process of adjusting and editing a website in order to achieve the highest ranking on a search engine. The main goal of search engine optimization is to make websites accessible and visible in Search Engine Result Pages (SERPs). [Zhang and Dimitroff 2005.] In the era of technology, search engine optimization has become more critical due to growing number of users who use search engines to discover new websites and brands [Zhang and Cabage 2013]. Search engines have become an easy way to navigate to a website even when you know the address [Egri and Bayrak 2014].

Search engine optimization should make websites more understandable for both users and search engine robots by redesigning the structure and content [Moz 2018]. The process of optimizing a website consists of techniques that are used mainly to increase the visibility and ranking of a website [Duk *et al.* 2013]. Moreover, SEO enables websites to get more visitors on their sites by improving the website's rank in search results [Lemos and Joshi 2017].

Search engine optimization methods centralize in end-users' behavior [Fiorini and Lipsky 2012]. Improving the visibility of a website by search engine optimization increases the number of visitors at the website and makes the website appear more competitive and usable. In addition to these factors, a higher ranked website can help with social promotion and a larger visibility can increase the sales of a brand. [Krrabaj *et al.* 2017.]

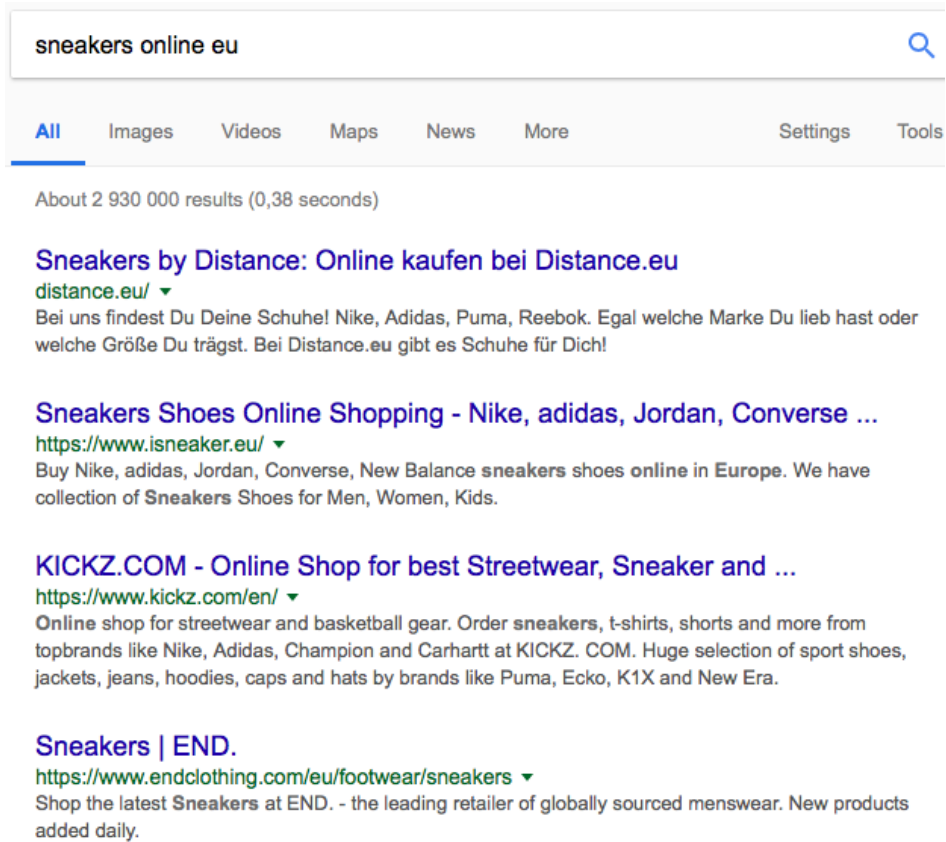


Figure 1. Search Engine Result Page [Google Search 2018a].

2.1. Terminology

To understand the process and components of search engine optimization, it is useful to know a few key terms that occur several times in this thesis. Crawler is a software to index pages from the web and to crawl is the process of finding those pages. Index is the place where the crawler stores the pages found. [Google 2018b.] Search query is the text the user wrote to the search engine. Search engines try to match the given search query to some keywords on the websites. Keywords are the connecting pieces between the website and the search engine. Crawlers index pages by the keywords to save some space. [Moz 2018.] Organic search results are the results that are not sponsored or paid [Gudivada *et al.* 2015]. The higher ranked pages are the ones on the top of the results list [Khan and Mahmood 2018].

2.2. Why to do search engine optimization?

When trying to find information, Internet users usually tend to read only the first page of search engine results, which emphasizes the significance of search engine optimization [Duk *et al.* 2013]. If a website is on the first page of search

results, the better opportunities it has to be found by consumers [Ho *et al.* 2010]. Search engine optimization permits websites to compete with each other [Lemos and Joshi 2017]. Search engine results page consists of organic and paid results, but the organic search results have been noticed to be more appealing to searchers and consumers than the sponsored results [Kai *et al.* 2014]. Implementing the methods of search engine optimization on a website is not just for search engines any more. It has become beneficial for both users and search engines since it makes the websites structure more understandable. Search engine optimization can be considered as a free way of marketing and it is less expensive than paid placement methods. [Duk *et al.* 2013.] The longer the visitor stays on a website, the more satisfied he or she is with it [Egri and Bayrak 2014]. Ranking high in the search engine result page boosts organization's visibility, increases sales and helps to build the user's trust, which again can lead to a better brand development. [Gudivada *et al.* 2015.]

Search engine optimization can increase the traffic to a website from a search engine [Fiorini and Lipsky 2012], since search engines presume that if a website has a lot of traffic, the worthier its contents must be [Moz 2018]. Ranking higher in a search engine can get the website more visitors with little or no costs [Knezevic and Vidas-Bubanja 2010], which means that search engine optimization can assist companies to stay competitive and in sight on the web [Khan and Mahmood 2018].

2.3. Social Media Optimization

Bhargava [2014] describes Social Media Optimization as “a process of optimizing your site/blog to be more visible in social media searches and sites, more easily linked by other sites and more frequently discussed online in blog posts and other social media”. Social media optimization can be executed by simply following the five rules of social media optimization. The first two rules are to create shareable content and to make sharing easy. Hence sharing content takes only minimal effort. The third rule is to reward engagement, because comments and discussion are what matters the most in the social web. The fourth rule of proactively sharing content aims at sharing content in all kinds of ways, for instance, tweeting about the new slides on Slideshare. The last rule is about encouraging the mashup. The mashup allows people to add their own voice to content published in social web. [Bhargava 2014.]

The five rules of social media optimization can be pursued in different ways to achieve the optimal outcome. For content publishers it is beneficial to have content shared by someone who has influence in social networks and a lot of followers [Moz 2018]. Social networks have become more important to search

engines since even more people try to find the answer from different social medias [Gudivada *et al.* 2015].

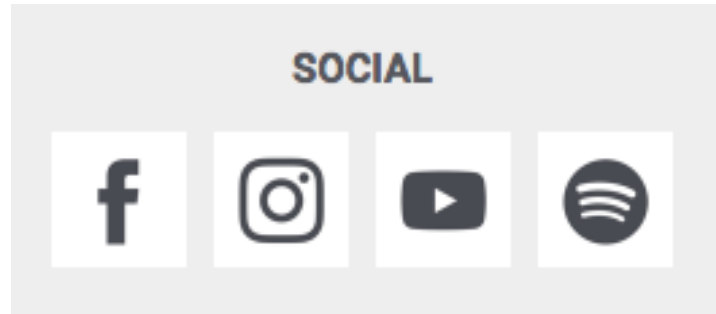


Figure 2. Making sharing content easy [Snipes 2018].

2.4. Paid Search Marketing, Pay-per-click

Pay Per Click (PPC) and Paid Search Marketing (PSM) are alternative practices to achieve visibility. Pay per click can be summarized as a way of paying keyword advertisement. The more popular a keyword is, the higher its costs per click are. [Duk *et al.* 2013.] In PPC only the clicks are paid by the advertiser [Fiorini and Lipsky 2012]. Pay per click is a way of advertising that is provided by search engines [Ho *et al.* 2010].

Paid Search Marketing is done by search engines displaying paid or sponsored search results along with other search results [Kai *et al.* 2014]. In paid search marketing companies can pay to have their website placed at a fixed position in the results page [Yang 2010].

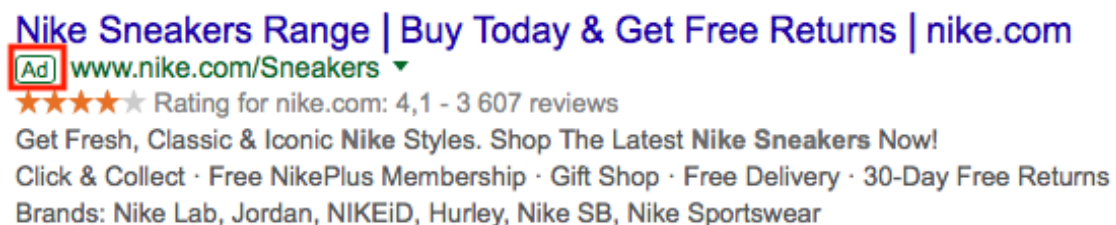


Figure 3. Paid ad in search results [Google Search 2018b].

3. How to do search engine optimization?

Implementing search engine optimization requires multiple different procedures. The methods and techniques of search engine optimization can be divided to white hat methods and black hat methods [Duk *et al.* 2013]. To gain success in search engine optimization demands knowledge in both usability and user experience [Moz 2018]. The visibility and ranking of a website is contributed by various factors, such as website structure and content [Zhang and Dimitroff 2005]. However, implementing the techniques once is not enough. Doing search engine optimization can require a lot of testing and constant developing [Moz 2018].

3.1. White hat methods

White hat methods consist of on-page and off-page optimization [Gudivada *et al.* 2015]. The on-page optimization factors can be controlled and performed by the website designers and developers, but the off-page optimization factors cannot [Zhang and Dimitroff 2005]. For this reason, on-page optimization methods are implemented when developing the website and off-page optimization methods when the website and its content is complete [Khan and Mahmood 2018]. To succeed in ranking high in the search result page, both on-page and off-page optimization methods should be used [Duk *et al.* 2013]. Search engine companies provide starter guides and webmaster tools to help websites and developers to follow and use the white hat practices and methods [Gudivada *et al.* 2015].

3.1.1. On-page optimization

On-page optimization focuses more on the website content and structure [Gudivada *et al.* 2015]. Therefore, on-page optimization should cover the page's content, HTML tags and the architecture. In addition, optimizing the content of the website should make the website pleasant and friendly for the user, not for the search engine. [Gupta *et al.* 2016.] Creating new content frequently keeps the crawlers coming back to the website which can help to increase the ranking of the website [Zhang and Cabage 2013].

Keywords should be used in the titles, text and metadata of a website [Moz 2018]. The page title is the first thing a crawler notices, so it should be descriptive and brief [Gupta *et al.* 2016]. Each page should have their own title for search engines to know the pages are separate [Google 2018b]. Generally, the choice of words should be similar to the vocabularies that the consumers use, and the content should use those keywords naturally [Gudivada *et al.* 2015]. The earlier the keyword occurs in a paragraph, the more weight that keyword

has in increasing the ranking of a website [Lemos and Joshi 2017]. If the keywords are included in the website URL, the more effect they have [Ho *et al.* 2010]. When the specificity of a keyword increases, the number of search results containing the specified keyword will decrease [Lemos and Joshi 2017].

The meta description tag is primarily the origin of the small text snippet displayed in a search engine beneath the title and the link [Moz 2018]. Hence, the description should be a short summary of the content of a page [Google 2018b]. Rich snippets are a type of structured data, which helps the webmasters to markup content, so a search engine can easily identify the type of that content [Moz 2018]. The structured data can contain for example the opening hours, prices, customer reviews or the location of the business [Google 2018b]. Including the location of the business somewhere in the website can help the website be shown high on the result page since search engines show results that are accurate to a specific location [Duk *et al.* 2013].

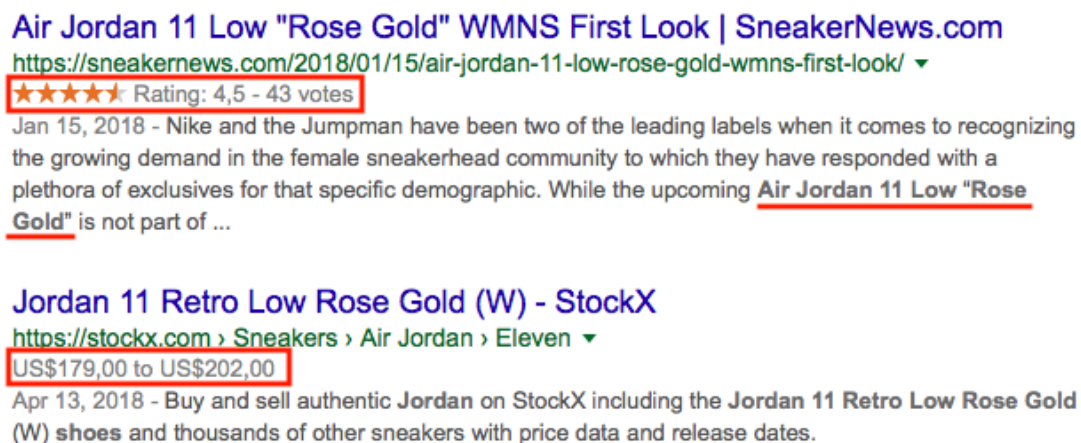


Figure 4. Rich snippets [Google Search 2018c].

On top of that, the different types of HTML headings make the page more structured and understandable [Gupta *et al.* 2016]. The anchor text of a link should tell the user where the link points to [Gudivada *et al.* 2015]. Moreover, images should have the "alt" attribute as a description for search engines [Moz 2018].

Lastly, having a privacy policy site can be considered as a measurement of trustworthiness by search engines but it also helps to make the website to look more professionally managed [Gudivada *et al.* 2015]. Creating a custom 404 page makes the user experience better and adding a link to the front page makes it more likely to users to stay on the website [Google 2018b].

3.1.2. Off-page optimization

Off-page optimization mainly consists of inbound and external links [Gudivada *et al.* 2015]. Websites with backlinks from external websites rank higher [Knezevic and Vidas-Bubanja 2010]. External links from other websites affect the ranking when visitors navigate to a website through those links. Creating backlinks can be difficult if the one website has no relevant content or the other website will not benefit at all from creating those backlinks. [Krrabaj *et al.* 2017.] Having a blog to promote the website and to increase its visibility is an effective way of doing off-page optimization [Gupta *et al.* 2016]. Links can be built on social networks such as Facebook, Twitter and LinkedIn [Zhang and Cabage 2013].

Breadcrumbs and sitemaps are considered as inbound link practices on a website [Gudivada *et al.* 2015]. Including breadcrumbs in a website helps the users navigate back to a previous page on the website [Google 2018b]. Furthermore, they improve the usability of the website. Creating a directory structure for a website allows crawlers to navigate through the website and index its content but it also helps with maintenance. [Gudivada *et al.* 2015.]

```

▼<url>
  <loc>https://www.afew-store.com/en/newsletter/</loc>
  <lastmod>2017-06-02</lastmod>
  <changefreq>always</changefreq>
  <priority>0.7</priority>
</url>
▼<url>
  <loc>https://www.afew-store.com/en/brands/</loc>
  <lastmod>2017-06-02</lastmod>
  <changefreq>always</changefreq>
  <priority>0.7</priority>
</url>
▼<url>
  <loc>https://www.afew-store.com/en/sneaker-archiv/</loc>
  <lastmod>2017-06-02</lastmod>
  <changefreq>always</changefreq>
  <priority>0.7</priority>
</url>
▼<url>
  <loc>https://www.afew-store.com/en/accessoires/</loc>
  <lastmod>2017-06-02</lastmod>
  <changefreq>always</changefreq>
  <priority>0.7</priority>
</url>
▼<url>
  <loc>https://www.afew-store.com/en/nike/</loc>
  <lastmod>2018-04-13</lastmod>
  <changefreq>always</changefreq>
  <priority>0.7</priority>
</url>

```

Figure 5. Part of a websites sitemap [Afew Store 2018].

Having a sitemap helps search engines to find more content on your site [Moz 2018]. Sitemap can also help the user to find the desired content and it might come in handy if the navigational structure of a website is created using dropdown menus [Gudivada *et al.* 2015]. A site owner can block crawlers from specific pages by adding a Meta Robots tag or a robots.txt file [Moz 2018]. Additionally, adding a rel="nofollow" to a link restricts the crawler from following that link. Website owners can utilize the rel="nofollow" to avoid link injection spam. [Moz 2018.]

3.2. Black hat methods

Black hat methods are not approved by search engines. Using black hat methods on a website is punished by lowering the rank of a website in the search results or even by deletion of the website [Duk *et al.* 2013].

Keyword stuffing is a black hat method that involves adding keywords and key phrases to a website repeatedly to make the website appear more relevant to a search engine [Moz 2018]. Ergo, keyword stuffing is mostly about trying to rank higher and there is no attempt of giving useful information for users when doing keyword stuffing [Duk *et al.* 2013]. Changing the color of the text same as the background to make the text seem invisible to users so the hidden keywords are shown only to crawlers is also considered as keyword stuffing [Duk *et al.* 2013].

When trying to rank higher in every possible way, websites create link exchange programs to make the website look more popular by having links back and forth from other websites [Moz 2018]. Further, payment can be offered in exchange of placement of links to get users to a website through links from different locations. Another black hat method is to use doorway pages which redirect users elsewhere and they are simply created for specifically ranking higher in search engines. [Duk *et al.* 2013.]

4. What is search engine marketing?

Search engine marketing, also known as SEM, is a technique for promoting websites by ranking them higher in search engines. SEM attempts to increase the visibility of a website in the result page by using the methods of search engine optimization, paid search marketing, pay-per-click and advertising. [Duk *et al.* 2013.] The primary purpose of search engine marketing is to draw more visitors to the website and to increase sales by ranking high in the search engine results page [Wang and Hao 2011]. Utilizing the marketing techniques to get traffic and visitors to a website is what search engine marketing is mainly about [Fiorini and Lipsky 2012].

Overall, search engine marketing is promoting a website in a search engine in order to get targeted customers and visitors [Knezevic and Vidas-Bubanja 2010]. Search engine marketing can be seen as a way of search engine optimization but included with various advertising options [Killoran 2013]. When pondering the methods of search engine marketing, it can be noticed that they can be divided in to two, the methods of search engine optimization and paid advertising [Ho *et al.* 2010]. Search engine marketing is a form of marketing in order to increase the visibility of a website in search engine result pages [Duk *et al.* 2013]. Compared to search engine optimization, in search engine marketing the websites can also pay the search engines in to get traffic and visitors to the website [Fiorini and Lipsky 2012]. The methods of search engine marketing are used to create traffic and get more visitors to a website [Knezevic and Vidas-Bubanja 2010].

5. Measuring the impacts of search engine optimization

The methods and practices of search engine optimization are changing continuously to match the changing techniques and developing rankings. Monitoring the websites performance in search engines should be an ongoing task in order to reform the website to fit the current guidelines and practices. Keeping up with the changes can be hard and expensive. [Gudivada *et al.* 2015.] Unfortunately, implementing the methods of search engine optimization will not guarantee the higher ranking but it can improve the chances [Kai *et al.* 2014]. Different search engines have different strategies and ranking algorithms which makes search engine optimization more difficult [Zhang and Dimitroff 2005]. The results from optimizing a website cannot be predicted and even seeing the first results can take a lot of time [Knezevic and Vidas-Bubanja 2010].

The results of search engine optimization can be measured and monitored with multiple tools. Google offers tools such as Analytics, Adwords, Webmaster and Trends and a one other than by Google is SeoQuake. [Khan and Mahmood 2018.] With SeoQuake the website owner can monitor different kinds of parameters about the website and its visitors. Parameters about a particular page are in example the number of social shares. Parameters about the domain could be the Google Index, the amount of monthly traffic and the compete rank. [SeoQuake 2018].

A great tool to analyze the whole performance of a website is Google Search Console [Google 2018b]. With Search Console the website owner can receive messages about occurring issues, reports about the performance, tools to modify the website better and even resources to learn more. Search console can come in handy when testing the content and accessibility of the website. [Google

2018c.] A common examinee, a bounce is a single-page session which happens when a user enters a website and exits it immediately. Bounce rate is calculated by dividing single-page sessions by all sessions on a website. [Google 2018a.] Even more precise information about the website can be obtained when comparing the bounce rate and the length of the session spent on the website [Egri and Bayrak 2014]. Bounce rate and many other rates can be monitored in Google Analytics [Google 2018a].

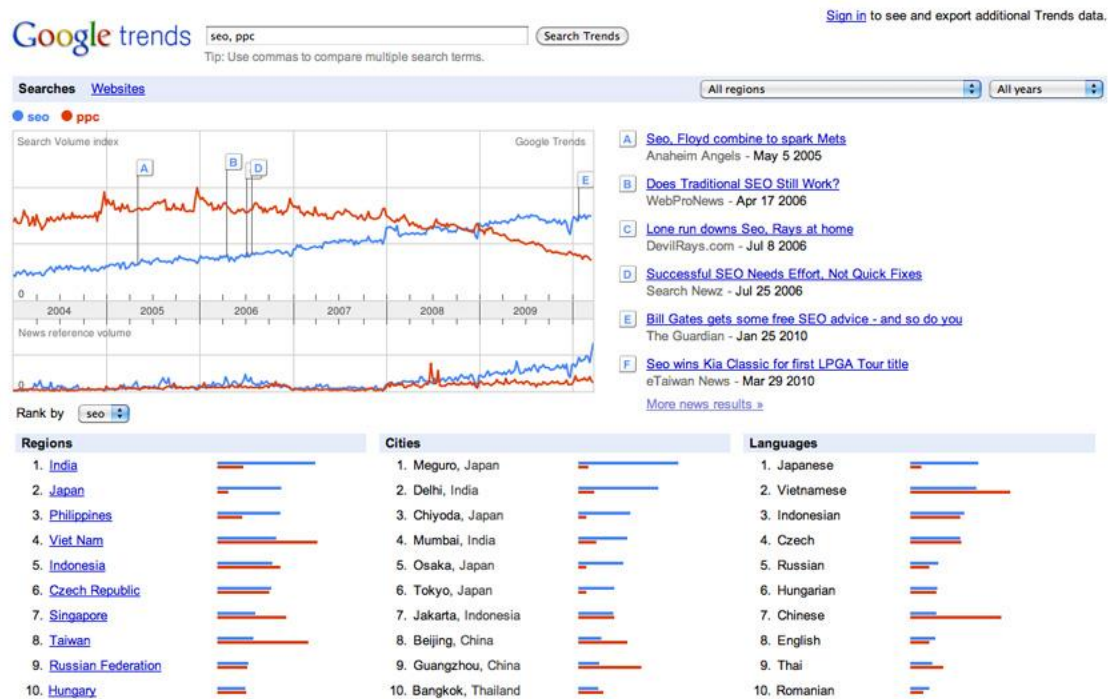


Figure 6. Keywords in Google Trends [Moz 2018]

6. Search engine optimization in online sneaker stores

Knowing the advantages of SEO and SEM can help to build a successful online business model and have impacts on online company growth and creating new innovations [Knezevic and Vidas-Bubanja 2010]. Ranking higher in the first search engine results page can boost the visibility of the store [Gudivada *et al.* 2015]. Having a search engine optimized website is important in the e-commerce to help potential customers to find what they are looking for [Duk *et al.* 2013]. In the case of online sneaker stores, higher rank in the search engine result pages may lead to more customers [Khan and Mahmood 2018]. In addition, the more customers online store receives, the more possibilities there will be in increasing the sales. Having a stable amount of sales can get stores noticed

by larger sneaker manufacturers and it can spark off the possibilities to sell limited edition collections.

Revealing the incoming sneaker releases in social media can help a store rank higher, since Google crawls social media sites as well as other websites. More links can attract more visitors and affect the visibility [Zhang and Cabage 2013]. Bhargava's [2014] first three rules of social media optimization can be executed in online sneaker stores by creating a social media account or a site to communicate with the customers and to share information about the newest releases. Online sneaker stores can get their website promoted by social media influencers by sending early releases to the influencers in exchange for visibility in social media. There are also a whole lot of social media profiles dedicated just to share pictures and videos about new releases. Having external links from all kinds of social media services will help to rank higher [Knezevic and Vidas-Bubanja 2010].

In online sneaker stores specific keywords can include the name and brand of the shoe, as well as the colorways and available sizes. The website users may favor different vocabularies and phrases than the creators of the content [Gudivada *et al.* 2015]. One certain key phrase for sneaker buyers to search for is "limited edition", while the people accountable for the sales use phrase "limited stock". In, addition, different user groups can use different synonyms of a word, "sneakers" vs "shoes". When the key phrase is highly specific, the chances are that the user is trying to make a purchase [Moz 2018].

It is beneficial to have the website shared by someone who has a lot of influence in the social networks [Moz 2018]. The sharing party could for example be a blog made to review new sneakers or an individual with a lot of followers. However, visibility in the social networks does not always reflect to authenticity and validity. Black hat methods such as keyword stuffing and link exchange can be used by sellers and stores selling fake sneakers. Those kinds of websites try to get more visibility to get visitors to increase the sales of counterfeit products.

7. Conclusion

Search engine optimization makes websites easily findable in search engines. This thesis described the usage, methods and impacts of search engine optimization as well as search engine marketing. My purpose was to introduce the basic features and approaches which occur when optimizing a website. I divided search engine optimization into its definition, methods of implementation, impacts and how to enhance SEO with marketing. My main goal was to make search engine optimization more understandable and to understand how all the

various techniques can be implemented in online sneaker stores. The introduction of the techniques helped to notice how search engine optimization is not rocket science. It is a compilation of minor alteration to a website, which however play a big role in the ranking of the website.

There are no shortcuts when trying to rank high in a search engine results page. Search engine optimization offers tools and methods to endeavor it yourself, though. To succeed in all search engines requires a lot of investigation and development but to succeed at least in one of them is a great start. This thesis can help with the beginning of optimizing a website to rank higher.

The constant changes in the algorithms of a search engine makes search engine optimization a bit troublesome. Hence, search engines do publish guides to understand the basics of search engine optimization and its methods at that very moment. This thesis introduced the essential methods and modes of operation that apply even when the algorithms undergo changes. Despite everything, the main functionality of the search engines stays the same. However, this thesis only introduces the methods and techniques superficially. All modes of operation do not work for everyone in the same way. The examples and instructions given are common usages and every website owner should learn and embrace methods that are specified to work with that certain website.

8. References

- Afew Store. 2018. Sitemap. <https://www.afew-store.com/en/sitemap.xml>
Checked 29.4.2018.
- Rohit Bhargava. 2014. Social media optimization: An easy guide to marketing and distributing your content online. https://www.slideshare.net/rohitbhargava/social-media-optimization-an-easy-guide-to-marketing-and-promoting-your-blog/22-UPDATED_SLIDE_2014Final_Authors_NoteDespite Checked 9.3.2018.
- Sanja Duk, Dunja Bjelobrck and Mia Carapina. 2013. SEO in e-commerce: balancing between white and black hat methods. In: *Proc. of the 36th International Convention on Information and Communication Technology, Electronics and Microelectronics*, 390-395.
- Gokhan Egri and Coskun Bayrak. 2014. The role of search engine optimization on keeping the user on the site. *Procedia Computer Science* 36, 335-342.
- Pierre M. Fiorini and Lester R. Lipsky. 2012. Search marketing traffic and performance models. *Computer Standards & Interfaces* 34, 6, 517-526.
- Google. 2018a. Google analytics help - Bounce rate. <https://support.google.com/analytics/answer/1009409?hl=en&topic=1120718&ctx=topic> Checked 22.4.2018.

- Google Search. 2018a. Search: sneakers online eu. Checked 29.4.2018.
- Google Search. 2018b. Search: nike. Checked 29.4.2018.
- Google Search. 2018c. Search: air jordan 11 low rose gold. Checked 29.4.2018.
- Google. 2018b. Google search console Help - Search engine optimization (SEO) starter guide. https://support.google.com/webmasters/answer/7451184?hl=en&ref_to_pic=3309469 Checked 5.2.2018.
- Google. 2018c. Google search console Help - Using search console with your website. https://support.google.com/webmasters/answer/6258314?hl=en&ref_to_pic=3309469 Checked 22.4.2018.
- Venkat N. Gudivada, Dhana Rao and Jordan Paris. 2015. Understanding search-engine optimization. *Computer* 48, 10, 43-52.
- Swati Gupta, Nitin Rakesh, Abha Thakral and Dev Kumar Chaudary. 2016. Search engine optimization: Success factors. In: *Proc. of the 4th International Conference on Parallel, Distributed and Grid Computing (PDGC)*, 17-21.
- Li-Hsing Ho, Meng-Huang Lu, Jui-Chen Huang and Hui-Yi Ho. 2010. The application of search engine optimization for internet marketing: An example of the motel websites. In: *Proc. of the 2nd International Conference on Computer and Automation Engineering (ICCAE)*, 380-383.
- Kai Li, Mei Lin, Zhangxi Lin and Bo Xing. 2014. Running and chasing - The competition between paid search marketing and search engine optimization. In: *Proc. of the 47th Hawaii International Conference on System Science*, 3110-3119.
- M. N. A. Khan and A. Mahmood. 2018. A distinctive approach to obtain higher page rank through search engine optimization. *Sadhana* 43, 43, 1-12.
- John B. Killoran. 2013. How to use search engine optimization techniques to increase website visibility. *IEEE Transactions on Professional Communication* 56, 1, 50-66.
- Boris Knezevic and Marijana Vidas-Bubanja. 2010. Search engine marketing as key factor for generating quality online visitors. In: *Proc. of the 33rd International Convention MIPRO*, 1193-1196.
- Samedin Krrabaj, Fesal Baxhaku and Dukagjin Sadrijaj. 2017. Investigating search engine optimization techniques for effective ranking: A case study of an educational site. In: *Proc. of the 6th Mediterranean Conference on Embedded Computing (MECO)*, 1-4.
- Snipes. 2018. Snipes. <https://www.snipes.com> Checked 29.4.2018.

- Joyce Yoseph Lemos and Abhijit R. Joshi. 2017. Search engine optimization to enhance user interaction. In: *Proc of International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, 398-402.
- Moz. 2018. The beginners guide to SEO. <https://moz.com/beginners-guide-to-seo> Checked 5.2.2018.
- SeoQuake. 2018. Introduction. <https://www.seoquake.com/guide/index.html> Checked 22.4.2018.
- Caili Wang and Xiaotong Hao. 2011. Comprehensive model for evaluating search engine marketing effect. In: *Proc. of the 4th International Conference on Business Intelligence and Financial Engineering*, 231-234.
- Nan Yang. 2010. Strategy of the search engine marketing for high-tech enterprises. In: *Proc. of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, 318-321.
- Jin Zhang and Alexandra Dimitroff. 2005. The impact of webpage content characteristics on webpage visibility in search engine results (Part I). *Information Processing and Management* 41, 3, 665-690.
- Sonya Zhang Sonya and Neal Cabage. 2013. Does SEO matter? Increasing classroom blog visibility through search engine optimization. In: *Proc. of the 46th Hawaii International Conference on System Sciences*, 1610-1619.

Ohjelmistoprojektin hallinta monitoimittajaympäristössä

Sari Lindström

Tiivistelmä.

Tässä tutkielmassa kuvaan monitoimittajaympäristössä toteutetun ohjelmistoprojektin hallinnassa tutkimuksen valossa huomioitavia seikkoja. Erityisesti tarkastelen hajautettuun tiimirakenteeseen liittyviä erityispiirteitä. Käsittelen aihetta ohjelmistokehittäjän näkökulmasta. Tämä tutkielma on kirjallisuuskatsaus. Etsin tietojenkäsittelytieteen sekä muiden soveltuvien alojen tieteellisestä kirjallisuudesta kuvauksia projektinhallinnan erityispiirteistä monitoimittajaympäristössä. Tarkoituksena on myös tutkia, tarjoaako tieteellinen kirjallisuus ratkaisuja esiin tuleviin ongelmiin. Monitoimittajaympäristön tyypilliset ongelmat näyttäisivät liittyvän toisaalta projektin teknisen hallinnan monimutkaisuuteen, mutta vielä enemmän projektiin osallistujien välisen viestinnän ja kommunikaation ongelmiin. Kommunikaation parantaminen edellyttää asiantuntijoiden viestintätaitoihin panostamista, mutta pohjimmiltaan sekä teknisissä että vuorovaikutukseen liittyvissä ongelmissa on kyse johtamisen ongelmista, jotka ovat ratkaistavissa oikeanlaisella johtamisella. Projektinjohdon tulee johtaa sekä asioita että ihmisiä ja lisäksi ymmärtää aina konteksti, jossa johtaminen tapahtuu.

Avainsanat ja -sanonnat: Hajautettu ohjelmistotyö, monitoimittajaympäristö, projektinhallinta, johtaminen, merkityksellistäminen, ketterä ohjelmistokehitys

1. Johdanto

Organisaatiot hankkivat IT-palvelunsa yhä useammin hajautetulla mallilla useamman toimittajan monitoimittajaympäristössä. Hajautetulla mallilla tavoitellaan alhaisempia kuluja, riskien hajautusta sekä parempaa kilpailukykyä. [Strakh 2015.] Tyypillisesti organisaatioiden järjestelmäkehitystyö on ulkoistettu joko osittain tai kokonaan ja toimittajat valitaan kilpailutuksin. Edelleen, ylläpito ja järjestelmien pienkehitystyön toimittajat kilpailutetaan myös säännöllisin väliajoin.

Sen lisäksi, että ohjelmistotuotanto tapahtuu usein ympäristössä, jossa yhden projektin tuotantoon osallistuu useampi toimittaja ja tiimirakenne on hajautettu, ohjelmistoprojektit ovat myös yhä monimutkaisempia sisällöltään. Sisällön lisäksi voidaankin usein todeta myös ohjelmistoprojektien toimintaympäristön olevan käytännössä varsin monimutkainen.

Tässä tutkielmassa tarkastellaan kirjallisuuslähteiden avulla monitoimittajaympäristössä tapahtuvaan ohjelmistokehitykseen liittyviä haasteita. Tarkoituksena on tutkia, millaisia ratkaisuja tieteellinen kirjallisuus tarjoaa kuvattuihin ongelmiin. Strakhin [2015] mukaan monitoimittajamalliin liittyvä yritysrajat ylittävä yhteistyö asettaa johtamiselle korkeat vaatimukset. Davidekova ja Hvorecky [2017] puolestaan esittävät, että yhteistyö fyysisesti erillään työskentelevien tiimin jäsenten välillä on perinteistä samassa tilassa tapahtuvaa työskentelyä alttiimpaa häiriötekijöille ja vaatii siksi tiiminvetäjältä erityistä tukea toimivan vuorovaikutuksen saavuttamiseksi. Tarkastelun esioletuksena onkin, että monitoimittajamalli ei tarjoa organisaatiolle keinoa johtamisen ongelmien välttämiseen, vaan projektijohdon ammattitaito on se, jolla saadaan käytettävissä olevat resurssit tekemään oikeita asioita oikeaan aikaan.

Tarkastelun toisena esioletuksena on, että tarjolla olevat projektinhallinnan opit ja työkalut sekä tieteellisestä kirjallisuudesta popularisoidut toimintamallit, kuten esimerkiksi Lean-menetelmä, eivät nekään tarjoa organisaatiolle yksioikoista toimintamallia ongelmien välttämiseen. Lean-menetelmä on toimintastrategia, jossa painopiste on resurssitehokkuuden sijaan virtaustehokkuudessa [Modig ja Åhlström 2016]. Ohjelmistotuotannossa viime vuosien aikana yleistyneestä Lean-ajattelusta on Fitzgeraldin ja Stolin [2017] mukaan useissa tapauksissa omaksuttu hyvin kapea-alainen käsitys, joka liittyy sen ainoastaan ketteriin menetelmiin.

Myöskään tämä tutkielma ei tarjoa yksioikoista toimintamallia monitoimittajaympäristöön liittyvien ongelmien välttämiseen. Toimintamallin tarjoamisen sijaan tämä tutkielma auttaa ymmärtämään ongelmien syitä ja etsimään niihin ratkaisuja tieteellisistä lähtökohdista sekä mahdollisesti myös välttämään sellaiset sudenkuopat, jotka tiede on jo selkeästi osoittanut ongelmiksi.

2. Hajautetun tiimin etuja

Nykyisessä toimintaympäristössä säännölliset kilpailutukset toimittajanvaihdoksineen ja ulkoistukset johtavat käytännössä usein siihen, että ohjelmistokehitystiimi harvoin on rakenteeltaan pysyvä [Zhuge 2001] ja fyysisesti koko ajan samassa paikassa. Kehittynyt viestintäteknologia puolestaan mahdollistaa osaltaan hajautetun tiimi- ja projektirakenteen [Kotlarsky and Oshri 2005]. Näin ollen tarkasteltaessa ohjelmistoprojektin tuotantoa monitoimittajaympäristössä, onkin syytä tarkastella samassa yhteydessä hajautettuun tiimirakenteeseen liittyviä erityispiirteitä.

Gorton ja Motwani [1996] tarkastelivat tutkimuksessaan ohjelmistotuotantoprosessia globaalisti hajautetussa tiimissä, projektin johtamista sekä tiimin

kommunikaatiota. Gortonin ja Motwanin [1996] tutkimuksen tarkoituksena ei ollut tuottaa uusia työkaluja hyödynnettäviksi projektissa. Sen sijaan tutkimus pyrki löytämään hajautetun ohjelmistotyön kehittämisen yleisiä periaatteita, joita voidaan hyödyntää viitekehyksenä tulevaisuudessa luotaville hajautetun ohjelmistotuotannon työvälineille. Gortonin ja Motwanin [1996] tutkimuksen jälkeen on kehitetty ohjelmistoprojektien hallintaan tehokkaita ja käyttökelpoisia työvälineitä. Niistä voidaan kuitenkin saada hyötyä ainoastaan, jos ne ottavat huomioon haasteiden taustalla olevat juurisyyt. Sen vuoksi tämä vuonna 1996 tehty tutkimus on edelleen ajankohtainen ja sen tuotokset suurelta osin sovellettavissa hajautetun ohjelmistotyön haasteisiin.

Gortonin ja Motwanin [1996] tutkimuksessa käsiteltiin hajautetun tiimiraakenteen käyttämisellä mahdollisesti saavutettavia etuja ohjelmistoprojektissa sekä siihen liittyviä haasteita. Ohjelmistotuotannon hajauttamisen etuna he näkivät muun muassa aikaeron hyödyntämisen suoman mahdollisuuden tuotteen tehokkaampaan valmistusprosessiin ja sitä kautta nopeampaan markkinoille saantiin. Aikaeron hyödyntäminen hajautetussa ohjelmistotyössä perustuu ohjelmointityön pilkkomiseen yhtä aikaa työstettävissä oleviin osiin. Aikaeroa hyödyntävä työskentely saattaa esimerkiksi toimia niin, että työstetään projektin osakokonaisuuksia vuorotellen hoitaen ohjelmointi toisessa maayksikössä ja laadunvarmistus aina toisessa. Koodin katselmointi ja testaus tapahtuisi näin ollen ohjelmoijien lepovuoron aikana ja he voisivat jatkaa seuraavana päivänä ohjelmointityötä suoraan seuraavaan vaiheeseen.

Gortonin ja Motwanin [1996] mukaan aikaero saattaa kuitenkin hidastaa projektin etenemistä, jos prosessi ei ole hallittu. Jos toinen yksikkö ei jostain syystä ole hoitanut omaa osuuttaan, joudutaan sen tuotosta odottelemaan. Aikaerosta johtuen ei myöskään voida reaaliajassa käsitellä esiin tulleita kysymyksiä eri yksiköiden välillä. Papadopouloksen [2015] mukaan perinteinen vesiputousmalli sopii hyvin hajautettuun ohjelmistotyöhön selkeän, ennalta määrätyn ja tarkasti kontrolloidun toteutusstrategian sekä perusteellisen dokumentoinnin vuoksi. Myös ketteriä menetelmiä voidaan soveltaa, ja käytännössä usein sovelletaankin, hajautetussa ohjelmistotyössä. Papadopouloksen [2015] mukaan ketterien menetelmien soveltaminen hajautetussa ohjelmistotyössä lisää kuitenkin projektinhallinnan monimutkaisuutta.

Ohjelmointityön osittamisen lisäksi aikaeron hyödyntämisen edellytyksenä on Gortonin ja Motwanin [1996] mukaan avoin ja toimiva kommunikaatio hajautetun tiimin jäsenten välillä. Tämä vaatii hajautetussa tiimissä teknologisten viestintävälineiden ja tekniikoiden lisäksi viestintäosaamista. Myös projektin johtamisen näkökulmasta viestintäosaaminen on olennainen taito tiimin ohjaa-

misessa. Kuten Gorton ja Motwani, myös Kotlarsky ja Oshri [2005] tutkivat kommunikaatiota hajautetussa ohjelmistokehitystyössä ja esittävät, että tehokas viestintäteknologia mahdollistaa viestinnän maantieteellisesti erillään työskentelevien projektin jäsenten välillä, mutta viestintäteknologia ei yksistään riitä takaamaan onnistunutta kommunikaatiota. Kotlarskyn ja Oshrin [2005] mukaan onnistuneeseen kommunikaatioon hajautetussa projektirakenteessa tarvitaan viestintäteknologiaa täydentäväksi mekanismiksi ihmisten välisiä tekijöitä, kuten sosiaaliset siteet ja tiedon jakaminen.

Davidekova ja Hvorecky [2017] puolestaan esittävät, että nykyaikainen teknologia mahdollistaa työskentelyn virtuaalisessa ympäristössä, jossa fyysinen läsnäolo ei enää ole välttämätöntä tiimin yhteistyön kannalta, mutta onnistuneen kommunikaation kannalta olennaista on valita tilanteeseen sopivat viestintäteknikat ja välineet tarjolla olevista vaihtoehdoista. Davidekovan ja Hvoreckyn [2017] mukaan ei ole yhtä, joka tilanteeseen sopivaa viestintäteknikkaa, vaan tilanteeseen sopivimmat tekniikat tulee harkita tapauskohtaisesti. Esimerkkejä Davidekovan ja Hvoreckyn [2017] mainitsemista viestintä- ja tiedonvälitystekniikoista ovat sähköposti, tiedonhallintajärjestelmät, intranetit, pikaviestimet sekä digitaaliset keskustelualustat. Kontekstiin soveltuvan viestintäteknologia lisäksi Davidekovan ja Hvoreckyn [2017] mukaan virtuaalisen tiimin onnistuneen viestinnän edellytyksenä on tiimin jäsenten motivaatio sekä vahva tiimin vetäjältä saatu tuki.

3. Viestintä hajautetussa tiimissä

Varhelahti ja Mikkilä-Erdmann [2016] tarkastelivat työelämän viestintää sekä asiantuntijoilta vaadittavaa teknologiavälitteistä viestintäosaamista. Tutkimuksen tuloksissa nousi esille asiantuntijoiden lisääntynyt tarve viestintäosaamiselle. Vielä vahvemmin korostui teknologian välityksellä tapahtuvaan viestintään liittyvä eettisen osaamisen tarve. Viestintäosaamisen käsitteellä tarkoitettiin Varhelahten ja Mikkilä-Erdmannin [2016] tutkimuksessa asiantuntijaverkostossa tapahtuvaa vuorovaikutussuhdetta, joka sisältää käsitteet viestintäosaaminen ja vuorovaikutusosaaminen. Eettisellä osaamisella puolestaan tarkoitettiin tutkimuksessa viestinnän tarkoituksenmukaisuutta ja toisten kunnioittamista viestintätilanteissa.

Varhelahten ja Mikkilä-Erdmannin [2016] mukaan etenkin sellaisissa asiantuntijatyöissä, joissa vuorovaikutus eri kulttuurien välillä ja verkostomaisuus lisääntyvät, on uhkana, että osaamiskuilu teknologisen viestinnän koetun osaamisen ja vaadittavan osaamisen välillä saattaa kasvaa tulevaisuudessa. Varhelahten ja Mikkilä-Erdmannin [2016] kyselytutkimukseen vastanneista vain

osa piti kulttuuristen erojen ymmärtämistä tärkeänä viestintäosaamisessa. Heidän näkemyksensä mukaan kuitenkin kuilu koetun osaamisen ja vaaditun osaamisen välillä oli suuri. Osa vastanneista ei puolestaan Varhelahden ja Mikkilä-Erdmanin [2016] mukaan pitänyt kulttuuristen erojen ymmärtämistä kovinkaan tärkeänä, vaikka näin olisi voinut olettaa asiantuntijan toimiessa kansainvälisissä verkostoissa. Ne Varhelahden ja Mikkilä-Erdmanin [2016] tutkimukseen vastanneet, jotka eivät pitäneet kulttuuristen erojen ymmärtämistä tärkeänä, eivät myöskään kokeneet, että kuilu koetun osaamisen ja vaaditun osaamisen välillä olisi erityisen suuri.

Kotlarskyn ja Oshrin [2005] mukaan kehittyneiden teknologisten viestintävälineiden käytöstä huolimatta monikulttuurisessa, globaalisti hajautuneessa tiimissä viestintä ja siihen liittyvä tiedon jakaminen välillä epäonnistuu. Onnistunut tiedon jakaminen on Kotlarskyn ja Oshrin [2005] mukaan kuitenkin olennainen tekijä tiimin menestykselliseen yhteistyöhön. Viestinnän tärkeys nousee esille ainakin jollain tasolla useissa tämän tutkielman lähteistä tarkastelunäkökulmasta, painopistealueista ja tutkimuskysymyksistä riippumatta. Varhelahden ja Mikkilä-Erdmanin [2016] sekä Kotlarskyn ja Oshrin [2005] lisäksi vuorovaikutusta ja viestintää tarkastelivat esimerkiksi Davidekova ja Hvorecky [2017], Fellows ja Liu [2016], Gorton ja Motwani [1996], Lee ja muut [2006] sekä Papadopoulos [2015]. Mielestäni kaikkien kansainvälisissä verkostoissa toimivien asiantuntijoiden olisikin hyödyllistä pohtia, onko heidän globaaliin ympäristöön liittyvä viestintäosaamisensa riittävällä tasolla, vaikka eivät pitäisikään kulttuuristen erojen ymmärtämistä viestinnässä kovinkaan tärkeänä. Näin siitäkin huolimatta, että epäonnistumisen juurisyy ei välttämättä liity puutteelliseen viestintäosaamiseen.

Varhelahden ja Mikkilä-Erdmannin [2016] tutkimuksessa kävi ilmi, että yleisin teknologinen viestintäväline oli tutkimuksen tekohetkellä sähköposti. Mahdollisia syitä sille, että esimerkiksi videoviestintävälineitä ei käytetty erityisen laajasti olivat Varhelahden ja Mikkilä-Erdmannin mukaan ongelmat tekniikassa, epävarmuus uuden teknologia käytössä tai yksinkertaisesti se, että organisaatio ei tarjoa niitä tiimien käyttöön. Parhaana viestintätapana Varhelahden ja Mikkilä-Erdmanin tutkimuksen vastauksissa pidettiin kasvokkain tapahtuvaa viestintää. Gortonin ja Motwanin [1996] esittämän aikaeroedun hyödyntäminen puolestaan edellyttäisi ajasta riippumattoman viestintävälineen, kuten sähköpostin, hyödyntämistä. Gortonin ja Motwanin [1996] tavoiteltavana pidettävä aikaeroedun hyödyntäminen sulkeekin käytännössä ulkopuolelle tiimin viestintäkeinoista Varhelahden ja Mikkilä-Erdmanin [2016] parhaaksi esittämän viestintätavan, eli kasvokkain tapahtuvan viestinnän.

Papadopouloksen [2015] mukaan sen lisäksi, että hajautettu tiimi hyödyn­ tää tehokkaasti teknisiä viestintävälineitä, voidaan tiimin vuorovaikutussuhteita parantaa järjestämällä tiimille yhteinen tapaaminen kasvokkain ainakin ker­ ran projektin toteutuksen aikana. Papadopouloksen [2015] tarkoittama tiimin yhteinen tapaaminen olisi hänen mukaansa mieluiten järjestettävä mahdolli­ simman varhaisessa vaiheessa projektia tiimin yhteisten tavoitteiden varmis­ tamiseksi. Kuten Papadopoulos [2015], myös Kotlarsky ja Oshri [2005] esittävät, että teknologian välityksellä viestimisen lisäksi globaalisti hajautetun tiimin on tärkeää tavata toisensa kasvokkain jossain vaiheessa. Tehokas yhteistyö tiimissä edellyttää Kotlarskyn ja Oshrin [2005] mukaan luottamuksellisia välejä tiimin jäsenten kesken. Luottamuksen muodostumista edistää stressittömäksi koettu, säännöllinen vuorovaikutus tiimin jäsenten välillä, ja kasvokkain tapahtuva viestintä ja vuorovaikutus ovat tässä Kotlarskyn ja Oshrin [2005] mukaan tär­ keä luottamusta edistävä mekanismi.

Varhelahti ja Mikkilä-Erdmann [2016] esittävät, että teknologiavälitteinen viestintäosaaminen, verkostoitumisosaaminen, kulttuurien ymmärtäminen ja kielitaito ovat asiantuntijan osaamisalueita, joita ei tule erottaa toisistaan. Edel­ leen, he esittävät näiden taitojen kokonaisvaltaista huomioimista osana asian­ tuntijuuden kehittämistä. Riittävän osaamisen varmistamiseksi tulevaisuudessa Varhelahten ja Mikkilä-Erdmanin [2016] mukaan myös nämä osaamisen alueet tulisikin huomioida korkeakoulutuksessa sekä asiantuntijaorganisaatioiden osaamisen kehittämisen prosesseissa.

4. Projektin johtamisen opit ja improvisointi

Projektin johtamisen metodit määritellään yleensä kokoelmaksi sääntöjä, ohjeita tai parhaita käytäntöjä, joita projektin toteuttamisessa käytetään. Valittu projek­ tin johtamisen lähestymistapa puolestaan rajaa projektin hallinnassa käytet­ tävät menetelmät. Projektien johtamisessa käytetään yhä enemmän perinteisten menetelmien rinnalla ketteriä menetelmiä. Sekä perinteisillä että ketterillä me­ netelmillä on havaitut etunsa ja haittansa, vertailtavasta ominaisuudesta riip­ puen. Yksittäisessä projektissa on myös mahdollista yhdistää perinteisiä ja ket­ teriä toimintatapoja. Tärkeintä on kuitenkin, että valittu metodi mukautetaan projektiin, eikä päinvastoin. [Spundak 2014.]

Klein ja muut [2014] esittävät, että projektin johtaminen ei käytännössä koskaan noudata täysin jotain tiettyä johtamisoppia, valittua menetelmää tai projektinjohtamisen työkalua, vaan projektin johto improvisoi toimintaa ohja­ tessaan ainakin jossain määrin. Improvisoinnin aste voi Kleinin ja muiden [2014] mukaan vaihdella vähäiseksi luokiteltavasta sovittujen toimintatapojen yksilöllisestä tulkinnasta tilanteeseen, jossa toimitaan kokonaan ilman ennalta

suunniteltua toimintamallia. Kleinin ja muiden [2014] mukaan projektit ovat monimutkaisia sosiaalisia rakennelmia, joissa kaikkia tapahtumia ei voida ennakoita ja improvisointi on väline yllättävän tapahtuman käsittelemiseksi.

Improvisointi ei Kleinin ja muiden [2014] mukaan ole osoitus tietämyksen tai osaamisen puutteesta. Osaamattomuuden sijaan tiukoista toimintatavoista poikkeaminen osoittaa Kleinin ja muiden [2014] mukaan, että projektin johdolla on riittävä osaaminen tunnistaakseen milloin on tarkoituksen mukaista irrottautua kaavasta ja valita sopivampi etenemistapa. Onnistunut improvisointi edellyttää Kleinin ja muiden [2014] mukaan kontekstin ymmärtämisen lisäksi laajaa tietämystä. Laajasta ja syvällisestä johtamisen teorioiden osaamisesta onkin heidän mukaansa projektin johdolle hyötyä sopivien toimintatapojen valinnassa sekä niiden joustavassa soveltamisessa.

Monimutkaisuus on nykyään projekteille yleisesti tunnistettu ominaisuus. Toimintaympäristö ja sen mukana projekteihin liittyvät ongelmat muodostuvat yhä monimutkaisemmiksi. Tällöin projektin hallintaan suunnitellut työvälineet ja toimintatavat saattavat osoittautua epätarkoituksenmukaisiksi käsillä olevaan monimutkaiseen ongelmaan ja sen seurauksena suunnitellaan uusia, entistä monimutkaisempia työvälineitä. Uusien, entistä monimutkaisempien työvälineiden suunnittelun ja käyttöönoton sijaan olisi kannattavampaa pysähtyä miettimään, mitä projektinjohtajat todellisuudessa tekevät. Näin voitaisiin nykyisistä johtamisteorioista, johtamisen käytäntöjen teoreettisen ymmärtämisen kautta, edetä kohti joustavampia johtamiskäytäntöjä. [Klein *et al.* 2014.]

5. Merkityksellistäminen

Tietojärjestelmäprojektien hallinta on pitkälti johtamiseen liittyvä kysymys. Vaikka yleisesti puhutaan projektin johtamisesta, johdettavana on käytännössä kuitenkin projektissa työskentelevät ihmiset sekä heidän työpanoksensa. Fellows ja Liu [2015] tarkastelivat projektityöskentelyn monikulttuurisiin ja kompleksisiin toimintaympäristöihin liittyviä haasteita sekä teknisestä että ihmisenäkökulmasta.

Painopisteenä Fellowsin ja Liun [2015] tutkimuksessa oli kulttuurieroista ja ihmisten toimintatavoista johtuvat haasteet sekä niiden ratkaisut. Heidän mukaansa tiimin ollessa globaalisti hajautunut ja monikulttuurinen saattavat tilanteiden ja asioiden erilaiset tulkinnat aiheuttaa ristiriitoja ja näin alentaa tiimin suorituskykyä. Avaintuloksenaan tutkimus esitteli käsitteen *merkityksellistäminen* (sensemaking).

Fellows ja Liu [2015] esittävät tutkimuksessaan, että yleisen näkemyksen mukaan sekä organisaation kulttuurilla että kansallisella kulttuurilla on vaikutusta tilanteiden tulkintoihin ja merkitysten luomiseen, vaikkakaan tarkkaa

mekanismia tähän ei tunneta. Koska tämän päivän globaalissa maailmassa työvoima saattaa liikkua maasta toiseen ja ihmiset saavat vaikutteita monista kulttuureista, voi kansallisen kulttuurin vaikutusta käsitteiden ja asioiden merkityksellistämiseen pitää mielestäni korkeintaan suuntaa antavana.

Fellowsin ja Liun [2015] mukaan projektissa työskentelevien yksilöiden kokemus siitä, että he kykenevät projektissa toimiessaan luomaan merkityksiä tilanteille, muokkautuu vähitellen yhteistyön jatkuessa. Merkityksellistäminen on iteratiivinen prosessi, johon vaikuttaa muun muassa yksilöiden aikaisemmat kokemukset ja heidän tulkintansa tilanteista, tapauskohtaiset seikat sekä se, miten muut tiimin jäsenet vastaavat tulkintoihin.

Fellows ja Liu [2015] toteavat merkityksellistämisen syntymisen tapahtumaketjun eroavan selkeästi niistä projektinjohtamisen malleista, joita yleensä käytetään projektinhallinnassa. Projektinjohtamisessa yleensä käytettyjen mallien tarkoitus on saavuttaa projektissa nopeasti tuloksia ja jatkaa sitten tuottavaa toimintaa tasaisesti ja ennustettavasti koko projektin ajan. Fellowsin ja Liun [2015] tutkimuksessa esitetään kuitenkin myös, että projektit harvoin toteutuvat täsmällisesti kuten aluksi suunniteltiin. Alkuperäisestä täsmällisestä suunnitelmasta poikkeamista ei Fellowsin ja Liun [2015] tutkimuksessa pidetä perustavaa laatua olevana haittana, vaan sen sijaan esitetään toimimista joustavasti, ketterien mallien mukaan.

Sekä tiimin tehokkuuden ja sujuvan yhteistyön että yksilön itsensä kannalta on tärkeää, että hän tiimin jäsenenä kykenee luomaan kontekstiin sopivia merkityksiä tilanteille. Merkityksellistäminen kuitenkin vaatii oman prosessinsa, eikä ilmesty yksilön ja tiimin voimavaraksi sillä silmänräpäyksellä kun jokin teknisen projektinhallinnan metodi otetaan käyttöön. Kun projektitiimit ovat tyypillisesti harvoin kovin pitkäikäisiä ja niiden sisälläkin henkilöiden vaihtuvuus on suurta, on merkitykselliseksi tekemisen prosessin toteutuminen niissä usein haasteellista. [Fellows and Liu 2015.]

Fellowsin ja Luin [2015] näkemyksen mukaan moninaisuuteen ja dynaamisuuteen liittyvät haasteet tulisi hyväksyä ja ottaa huomioon jo projektin perustamisvaiheessa. Asiantuntijoiden kommunikointitaidot ovat myös tässä avainasemassa ja asiantuntijoiden viestintäosaamista olisi syytä kehittää kohti vaadittua osaamistasoa, kuten Varhelahti ja Mikkilä-Erdmann [2016] esittävät.

6. Ketterä hajautettu ohjelmistokehitys

Papadopoulos [2015] tarkasteli tutkimuksessaan *”Moving from traditional to agile software development methodologies also on large, distributed projects”* ketterien ohjelmistokehitysmenetelmien etuja perinteisiin menetelmiin verrattuna. Erityisesti Papadopoulos [2015] tutki ketterien menetelmien soveltamista laajoissa,

hajautetuissa projekteissa. Etujen tarkastelunäkökulma oli arvon tuottamiseen asiakkaalle perustuva kilpailukyky nopeasti kehittyvässä ja muuttuvassa liiketoiminnan ympäristössä. Papadopouloksen [2015] tutkimuksessa vertailtiin toisiinsa kahta eri projektia, joista toinen toteutettiin perinteisillä menetelmillä ja toinen ketterillä. Projektitiimit eivät olleet täysin identtisiä, kuin ei myöskään valmistettava lopputuote, jolloin täysin eksaktia vertailua ei ollut mahdollista tehdä. Vertailu perinteisen ja ketterän toimintamallin välillä pystyttiin kuitenkin Papadopouloksen [2015] tutkimuksessa tekemään riittävän tarkasti luotettavan lopputuloksen saamiseksi.

Ketterät toimintamallit ovat jo vakiinnuttaneet asemansa pienemmissä projekteissa, mutta Papadopouloksen [2015] tutkimus antaa selkeää tieteellistä evidenssiä ketterien toimintamallien soveltuvuudesta tämän päivän ohjelmistokehitystyöhön myös laajoissa, hajautetuissa projekteissa. Papadopouloksen [2015] tutkimuksen tuloksissa esitetään, että ketterät ohjelmistokehityksen menetelmät ovat perinteisiä tehokkaampia siinäkin tapauksessa, että kyseessä on laaja projekti, joka toteutetaan hajautetulla projektirakenteella. Papadopoulos [2015] esitti tutkimuksensa pohjalta asiakkaan tyytyväisyyden tuotteeseen lisääntyneen ketterässä toimintamallissa verrattuna perinteiseen toimintamalliin. Ketterien toimintamallien tehokkuus näkyi Papadopouloksen [2015] tutkimuksen tuloksissa ensinnäkin niin, että pystyttiin tuottamaan paremmin lisäarvoa asiakkaalle. Toiseksi, ketterien menetelmien tehokkuus näkyi tutkimuksen tuloksissa kohonneena työtyytyväisyytenä projektitiimissä.

Ketterien menetelmien toimintamallissa asiakkaalla on mahdollisuus tehdä muutoksia tuotteeseen kehitystyön jo alettua, mikä selittää osaltaan asiakkaan lisääntyneen tyytyväisyyden tuotteeseen. Papadopouloksen [2015] tutkimus osoitti kuitenkin ketterän toimintamallin myös parantaneen tuotteen laatua. Papadopouloksen [2015] tutkimuksessa esiin tullut tuotteen laadun parantuminen selittyy paljolti testausstrategialla. Perinteisessä toimintamallissa testaus ajoitettiin tiettyyn hetkeen kehitystyön loppupuolelle, jolloin virheitä löytyi paljon kerralla ja kehitystiimi joutui tekemään niiden korjaamiseksi huomattavan paljon töitä saadakseen tuotteen valmiiksi määräaikaan mennessä. Ketterässä toimintamallissa sen sijaan testausta tehtiin jo aikaisemmassa vaiheessa ja virheitä päästiin myös korjaamaan varhaisemmassa vaiheessa. Virheiden kokonaisuus oli ketterässä mallissa merkittävästi pienempi kuin perinteisessä ja tuote oli valmis määräaikaan mennessä ilman kehitystiimin kohtuuttomia loppumetrioiden ponnisteluja.

Papadopouloksen [2015] tutkimuksesta ilmenee, että ketteriä menetelmiä noudattaneissa projektitiimissä työtyytyväisyys oli selkeästi korkeampi kuin perinteisiä menetelmiä noudattaneissa tiimissä. Tämä selittyy sillä, että työ-

kuorma jakautui tasaisemmin koko projektin ajalle ja loppupuolen työläs virheidenkorjausurakka vältettiin kokonaan. Ketteriä menetelmiä noudattaneen projektitiimin jäsenet saivat myös konkreettisesti tuottaa asiakkaalle lisäarvoa koko projektin ajalla saadessaan valmiiksi osakokonaisuuksia sprintteihin jakotetulla työskentelyajalla. Tämän lisäksi läheisempi yhteistyö tiimin jäsenten välillä ja työskentely yhteisen päämäärän saavuttamiseksi paransi Papadopouloksen [2015] mukaan tiimin kommunikaatiota ja yhteenkuuluvuutta.

Perinteiset ohjelmistokehitysmenetelmät ovat tyypillisesti perustuneet vahvaan hierarkiaan ja perusteelliseen, ajantasaiseen dokumentointiin. Ketterissä toimintamalleissa taas painopiste on yhteistyön ja toimintojen välisen koordinaation lisäämisessä. Perinteisessä vesiputousmallin mukaisessa ohjelmistokehityksessä on tyypillisesti projektin laajentuessa ja monimutkaisuuden lisääntyessä vahvistettu projektin hallintaa lisäämällä johtotason valvontaa ja uusia menettelytapoja sekä tarkistuspisteitä. Projektin toteuttamisen vaiheet on erotettu omiin tiimeihinsä ja suoritettu erikseen, mallin mukaan etenevässä järjestyksessä. Ketterissä toimintamalleissa kuitenkin hierarkia tulisi pitää mahdollisimman matalana. Tiimejä saa olla useita ja projektin toteutus hajautettua, mutta eri tiimien, kuten esimerkiksi arkkitehtuurista, määrittelyistä ja toteutuksesta vastaavien tiimien tulisi työskennellä koko ajan limittäin jokaisessa projektin vaiheessa. Tällöin jokaisesta tiimistä tulisi olla tapaamisissa ja kokouksissa mukana vähintään yksi avainhenkilö. [Papadopoulos 2015.]

Nähdäkseni yleisin toimintamalli nykyaikaisessa ohjelmistokehityksessä on ketterät menetelmät. Käytännössä useasti toimintamalli on kuitenkin jotain perinteisen vesiputousmallin ja ketterän kehityksen väliltä; tietoisesti valitaan ketterä toimintatapa, joka kuitenkin saattaa sisältää vesiputousmallin piirteitä. Papadopouloksen [2015] mukaan ketterän toimintamallin onnistuneen käyttöönoton kannalta on ensiarvoisen tärkeää panostaa siihen, että ohjelmistokehitystiimille viestitään selkeästi käytäntöjen taustalla olevat periaatteet.

Ketterä toimintamalli, jossa testausta tehdään jatkuvasti projektin edetessä, vaatii resursseja myös asiakkaan puolelta. Kokemukseni mukaan asiakasorganisaatiossa ei välttämättä ole varattu resursseja erityisesti testaukseen, vaan testausta tekevät asiantuntijat oman työnsä ohella, jolloin jatkuva kehitysprosessin aikana tapahtuva testaus on ajallisesti pois asiantuntijoiden varsinaisten perustehtävien hoitamisesta. Asiantuntija, jonka olisi tarkoitus testausta tehdä, saattaa joutua aikataulupaineiden vuoksi keskittymään ainoastaan varsinaisiin perustehtäviinsä testauksen jäädessä toissijaiseksi. Jos yhteisistä menettelytavoista ei ole sovittu riittävän selkeästi ja avoimesti, asiantuntija saattaa myös kokea, että hän joutuu tekemään ohjelmistokehittäjille kuuluvia tehtäviä, mikä tuskin lisää tyytyväisyyttä ohjelmistotuotteeseen. Ketterästä ohjelmistokehi-

tysmallista voidaankin mielestäni saada asiakkaalle lisäarvoa ainoastaan silloin, kun prosessi on hallittu ja toimintatavat on viestitty kaikille osapuolille riittävän selkeästi.

7. Tehokkuus ja työelämän laatu sekä muita näkökulmia

Markkula [2011] tarkasteli väitöskirjassaan johtamisajattelun merkitystä organisaatioiden tehokkuuden lisäämisessä sekä työelämän laadun parantamisessa. Hän esitteli käsitteet tehokas johtaja, hyvä johtaja sekä viisas johtaja ja kuvasi näihin käsitteisiin liittyviä johtamistapoja. Tehokas johtaja tarkoittaa Markkulan [2011] mukaan tuotannollisesta näkökulmasta taitavaa, aikaansaavaa johtajaa, eli hän johtaa asioita. Hyvä johtaja puolestaan on Markkulan [2011] mukaan hyvä ihmisten johtaja. Viisas johtaja on molempia edellä mainittuja yhdesä ja hänessä yhdistyy sekä asioiden että ihmisten johtamisen taito.

Viisaan johtajan toimintatapaan kuuluu Markkulan [2011] mukaan muun muassa selkeät tavoitteet, oikeudenmukaisuus sekä mielenkiintoisten työtehtävien tekemisen mahdollistaminen. Markkula [2011] esittää, että tulevaisuudessa johtaminen tulee olemaan yhä haastavampaa ja johtajalta vaaditaan entistä monipuolisempia johtamistaitoja. Oikeanlaisia johtajia kuitenkin tarvitaan organisaatioissa, jotta ne todella voisivat hyödyntää ihmisten asiantuntemusta. Tarkasteltaessa tehokkuutta ja työelämän laatua Markkula [2011] esittää, että työelämän laadun parantaminen lisää tehokkuutta. Tämän vuoksi ihmislähtöinen johtamistapa ei ole ainoastaan eettisesti kestävä valinta [Markkula 2011], vaan myös organisaation edun mukaista.

Lee ja muut [2006] esittävät tutkimuksessaan *Aligning IT components to achieve agility in globally distributed system development*, että ketterä IT-strategia, ketterä IT-infrastrukturi ja ketterä projektinhallinta ovat avaintekijöitä, jotka mahdollistavat ketterän toimintatavan globaalisti hajautetun systeemikehityksen ympäristössä. Lee ja muut [2006] kuvaavat tutkimuksessaan ketterien toimintamallien roolin mainituissa ympäristöissä ja selittävät, miten tällaiset toimintamallit voidaan heidän näkemyksensä mukaan siinä kontekstissa saavuttaa.

Ketterän IT-strategian saavuttaminen edellyttää kehitystyön nopeaa muutumista liiketoiminnan tarpeisiin paikallisella tasolla. Ketterä strategia mahdollistaa myös paikallisten resurssien hyödyntämisen nopealla aikajänteellä. Ketterän IT-infrastruktuurin keinoja puolestaan ovat esimerkiksi standardoitu sovellusala, modulaarinen ohjelmistoarkkitehtuuri sekä internet-pohjaiset viestintävälineet. Johtaminen on avainasemassa ketterässä, globaalisti hajautetussa ohjelmistotyössä. Projektin johdon tulee kyetä huolehtimaan lokalisoimista, tehtävien jaosta ja viestinnästä kaikkien mukana olevien toimijoiden

kanssa. Lisäksi hänellä tulee olla hyvä kokonaiskuva projektista. [Lee *et al.* 2006.]

Su ja Levina [2011] tutkivat valmistavan teollisuuden toimintamalleja kansainvälisessä monitoimittajaympäristössä ja tutkivat niiden soveltamista informaatioteknologian (IT) alalla. Sun ja Levinan mukaan kansainvälisissä IT-alan palveluntuotannon ulkoistuksissa on yleistynyt toimintamalli, jossa palvelut ostetaan hajautetusti usealta toimittajalta. Tutkimustietoa ilmiöstä ei kuitenkaan ole juurikaan saatavilla. Sen sijaan valmistavan teollisuuden alalla vastaavaa ilmiötä, sekä siihen liittyvää toimittajahallintaa, on tutkittu runsaasti. Su ja Levina [2011] ovatkin tutkimuksessaan integroineet nämä teollisuuden toimintamallit IT-alan vastaavaan ilmiöön ja täydentäneet tutkimustaan kahdella esimerkkitapauksella.

Su ja Levina [2011] tunnistivat tutkimuksessaan neljä erilaista kumppanuuden tyyppiä monitoimittajamallissa. He muodostivat näiden neljän kumppanuuden ja kumppanuuden laatua mittaavan kahden ulottuvuusmittarin avulla viitekehysten kumppanuustyypeille. Toisaalta jaotteluperusteena oli se, kuinka syvää kumppanuus oli ja toisaalta miten paljon kumppaneita käytettiin tietyn toiminnon tuottamiseen. Syvässä kumppanuudessa Sun ja Levinan [2011] mukaan yhteistoiminta on pitkäaikaista, eikä lyhyen tähtäimen oman edun tavoittelua juurikaan esiinny. Usean toimittajan mallissa kustannus asiakkaalle on tyyppillisesti ainakin aluksi alhaisempi. Sitoutumisen taso on kuitenkin alhaisempi molemmin puolin, mikä johtaa suureen vaihtuvuuteen. Sun ja Levinan [2011] mukaan useammasta toimittajasta saattaa aiheutua suuremmat hallinnolliset kustannukset.

8. Yhteenveto

Globaalisti hajautettu tiimirakenne voi haasteistaan huolimatta olla ohjelmistotuotannossa hyvinkin tehokas. Tämä edellyttää kuitenkin, että tietyt työskentelyyn liittyvät perusasiat ovat kunnossa. Ensiksikin, työskentelyn pitää olla hyvin organisoitua. Toiseksi, kommunikaation tulee olla sujuvaa ja tehokasta. On projektin johdon tehtävä huolehtia, että nämä työn tekemisen puitteet ovat kunnossa. Toteutettaessa ohjelmistoprojektia monitoimittajaympäristössä johtamisen merkitys korostuu vielä entisestään.

Strakhin [2015] mukaan monitoimittajaympäristössä odotetaan hyvää johtamista. Usean toimittajan ympäristö asettaa johtamiselle korkeat vaatimukset Strakhin [2015] mukaan esimerkiksi sen vuoksi, että luottamuksen rakentaminen toimijoiden välille yli organisaatorajojen on haastavaa. Toimijoiden välinen luottamus on kuitenkin ensiarvoisen tärkeää sujuvan yhteistyön kannalta. Monitoimittajaympäristössä on ensisijaisen tärkeää myös motivoida toimittajia

yhteistyöhön toistensa kanssa. [Strakh 2015.] Lee ja muut [2009] puolestaan tarkastelivat monitoimittajaympäristöön liittyvää toimittajanhallintaa ja oppimisprosesseja tutkimuksessaan *Multi-vendor outsourcing: relational structures and organizational learning from a social relation perspective* ja kuten Strakh, esittävät monitoimittajaympäristön olevan yhden toimittajan ympäristöä haastavampi johtamisen kannalta. Leen ja muiden [2009] mukaan tilanteeseen sopivalla monitoimittajamallilla sekä avoimella osaamisen ja tiedon jakamisella osapuolten kesken voidaan monitoimittajaympäristössä kuitenkin saavuttaa tilanne, jossa sekä asiakas että toimittajat hyötyvät monitoimittajamallista.

Käytännössä nykyaikainen viestintätekniikka, ohjelmistotuotannon työvälineet sekä projektin johtamisen opit tarjoavat käyttökelpoiset keinot ohjelmistoprojektin työtehtävien organisointiin. Ketterät toimintamallit sopivat hyvin monitoimittajaympäristössä ja hajautetuissa tiimeissä tapahtuvaan ohjelmistokehitystyöhön [Lee *et al.* 2006]. Projektin johdon tulee valita tarkoituksenmukaisimmat työvälineet tiimille ja huolehtia käyttöönotosta. Tehtävän voi toki delegoida, mutta vastuu on nähdäkseni silti projektin johdolla.

Hajautettu tiimirakenne ja usean toimittajan toimintaympäristö asettavat kommunikaatiolle ja viestintäosaamiselle haasteita. Asiantuntijoiden teknisen ja eettisen viestintäosaamisen tuleekin olla riittävällä tasolla [Varhelahti ja Mikkilä-Erdmann 2006]. Yleensä toimintaympäristö asettaa vaatimuksia myös kieli- taidolle, koska yhä useampi ohjelmistoprojektissa työskentelevä asiantuntija käyttää työkielenään ainakin osittain jotain muuta kieltä kuin äidinkieltään. Monikulttuurisuus ja tiimien moninaisuus tuovat myös omat ongelmansa kommunikaatiolle. Monikulttuurisuus ja tiimien moninaisuus tulisi ottaa huomioon jo projektin perustamisvaiheessa [Fellows and Liu 2015].

Teknisistä asiasisällöistä ja projektinhallinnan tehokkaista työvälineistä huolimatta projektia toteuttavat kuitenkin ihmiset ja sen vuoksi projektissa täytyy asioiden johtamisen lisäksi nähdäkseni huolehtia myös ihmisten johtamisesta. Ihmisten ja asioiden johtaminen eivät ole toistensa poissulkevia asioita ja Markkulan [2011] mukaan viisas johtaja osaakin johtamistyössään huomioida nämä molemmat osa-alueet ja edesauttaa siten työntekijöiden työssä jaksamista sekä organisaatioon sitoutumista. Työnsä mielekkääksi kokevat ihmiset ovat Markkulan [2011] mukaan yleensä myös tehokkaampia työntekijöitä. Kun työn tekemisen puitteet ovat kunnossa ja työntekijät voivat kokea onnistumisia työssään, he kokevat parempaa työtyytyväisyyttä, joka nostaa tuottavuutta, ja niin edelleen. En näe syytä, miksi tämä ei pitäisi myös monitoimittajaympäristössä toteutettavassa ohjelmistokehitystyössä.

Viiteluettelo

- Monika Davidekova and Jozef Hvorecky. 2017. ICT collaboration tools for virtual teams in terms of the SECI model. *International Journal of Engineering Pedagogy* 7, 1, 95-116.
- Richard Fellows and Anita Liu. 2016. Sensemaking in the cross-cultural contexts of projects. *International Journal of Project Management* 34, 2, 246-257.
- Brian Fitzgerald and Klaas-Jan Stol. 2017. Continuous software engineering: A roadmap and agenda. *The Journal of Systems and Software* 123, 176-189.
- Ian Gorton and Sanjeev Motwani. 1996. Issues in co-operative software engineering using globally distributed teams. *Information and Software Technology* 38, 10, 647-655.
- Louis Klein, Christopher Biesenthal and Erlend Dehlin. 2014. Improvisation in project management: a praxeology. *International Journal of Project Management* 33, 2, 267-277.
- Julia Kotlarsky and Ilan Oshri. 2005. Social ties, knowledge sharing and successful collaboration in globally distributed system development projects. *European Journal of Information Systems* 14, 1, 37-48.
- Jae-Nam Lee, Cheng Suang Heng and Jung Lee. 2009. Multi-vendor outsourcing: relational structures and organizational learning from a social relation perspective. In: *Proceedings of ICIS 2009*, 71.
- One-Ki (Daniel) Lee, Probir Banerjee, Kai H. Lim, Kuldeep Kumar, Jos van Hillegersberg and Kwok Kee Wei. 2006. Aligning IT components to achieve agility in globally distributed system development. *Communications of the ACM* 49, 10, 48-54.
- Marja Markkula. 2011. *Johtaminen, tehokkuus ja työelämän laatu – organisaatioiden toiminnan kulmakivet*. Väitöskirja, Filosofinen tiedekunta, Vaasan yliopisto.
- Niklas Modig ja Pär Åhlström. 2016. *Tätä on Lean: ratkaisu tehokkuusparadoksiin*. Rheologica Publishing.
- Georgios Papadopoulos. 2015. Moving from traditional to agile software development methodologies also on large, distributed projects. *Procedia - Social and Behavioral Sciences* 175, 12, 455-463.
- Mario Spundak. 2014. Mixed agile/traditional project management methodology – reality or illusion? *Procedia - Social and Behavioral Sciences* 119, 939-948.
- Tero Strakh. 2015. *Palvelun saatavuuden varmistaminen monitoimittajaympäristössä*. Pro gradu –tutkielma. Informaatiotieteiden yksikkö, Tampereen yliopisto.
- Ning Su and Natalia Levina. 2011. Global multisourcing strategy: integrating learning from manufacturing into IT service outsourcing. *IEEE Transactions on Engineering Management* 58, 4, 717-729.

- Mervi Varhelahti ja Mirjamaija Mikkilä-Erdmann. 2016. Viestintää teknologian välityksellä asiantuntijaverkostoissa. *Media & viestintä* 2016, 39.
- Hai Zhuge. 2001. Knowledge flow management for distributed team software development. *Knowledge-Based Systems* 15, 465-471.

Välimuistin hyödyntäminen hakukoneissa

Toni Linnusmäki

Tiivistelmä

Tutkielman tarkoituksena on esitellä nykyaikaisen hakukoneen toimintaa ja tutustua, kuinka välimuistia hyödynnetään sen toiminnassa. Tutkielmassa vertaillaan erilaisia tapoja varastoida tietoa välimuistiin ja käyttää sitä hakutulosten näyttämässä käyttäjälle. Jotta välimuistiin voidaan varastoida tietoa ja se on relevanttia hakutulosten kannalta, tutkielmassa esitellään erilaisia tekniikoita, joita hakukoneet hyödyntävät pitämään käyttämänsä välimuistit ajan tasalla. Tutkielmassa esitellään myös, miten hakukoneiden välimuisteissa voidaan hyödyntää koneoppimista.

Avainsanat: Koneoppiminen, välimuisti, hakukone, hakukyselyt, hakutulokset.

1 Johdanto

Hakukoneet ovat nykypäivänä jokaisen saatavilla ja käytettävissä. Suurimmat hakukoneet ovat olleet käydyimpien sivujen joukossa jo vuosikymmenien ajan [2]. Hakukoneet ovat meille arkipäivää ja tässä tutkielmassa tarkastellaan niiden toimintaa tarkemmin.

Tässä tutkielmassa keskitytään hakukyselyiden prosessointiin ja tarkemmin siihen, kuinka hakukoneet käsittelevät kyselyitä ja käyttävät apunaan välimuistia tuottaessaan käyttäjälle hakutuloksia. Välimuisti on tärkeä osa hakukonetta, jonka avulla saadaan näytettyä kyselyiden tulokset tehokkaammin käyttäjälle. Tutkielmassa esitellään, kuinka hakukoneen välimuisti toimii ja mitä tietoa sinne voidaan säilöä. Nykypäivänä hakukoneet hyödyntävät myös koneoppimista tietojen varastoimisessa, mikä on hakukoneen tehokkuuden kannalta erittäin tärkeää.

Ymmärtääksemme, miten välimuistia hyödynnetään hakukoneissa, on ensin ymmärrettävä hakukoneen toimintaa. Luvussa 2 esitellään hakukoneen toimintaa kokonaisuutena. Luku 3 tarkastelee välimuistin hyödyntämistä hakutulosten näyttämässä ja tallentamisessa, erilaisia välimuistimalleja sekä hakutulosten esihakemista. Luvussa 4 esitellään erilaisia tilanteita, joissa voidaan hyödyntää koneoppimista tietojen varastoimisessa hakukoneiden välimuistiin ja hakukyselyiden tunnistamisessa.

2 Hakukoneiden toiminta

Yleisenä terminä hakukone on ohjelmisto, joka hakee tietoa ja esittää löytämänsä tiedon käyttäjälle. Nykyaikana hakukoneella kuitenkin tarkoitetaan

yleensä internetissä toimivia hakukoneita, joita tässä tutkielmassa käsitellään. Suosituimpia hakukoneita ovat muun muassa Google, Baidu ja Yahoo, jotka kaikki sijaitsevat kymmenen eniten käydyimmän sivujen joukossa tällä hetkellä [2]. Näistä länsimaiselle ihmiselle tuntemattomin on Baidu, joka on suosituin kiinankielinen hakukone. Tässä tutkielmassa keskitytään pääasiassa Google-hakukoneen toimintaan, koska se toimii hyvänä esimerkkinä nykyaikaisesta internet-hakukoneesta.

Nykyaikaisen hakukoneen standardit ovat asettaneet Sergei Brin ja Larry Page [4] julkaisemalla Googlen, josta on tullut kaikista suosituin hakukone ja eniten vierailtu sivusto maailmassa [2]. Ymmärtääksemme, miten hakukoneet hyödyntävät välimuistia ja koneoppimista nykypäivänä, on tutustuttava tarkemmin, miten hakukone käsittelee ja hakee tietoa.

Gan ja Suel [7] ovat jaotelleet hakukoneiden toiminnan yleisesti neljään vaiheeseen:

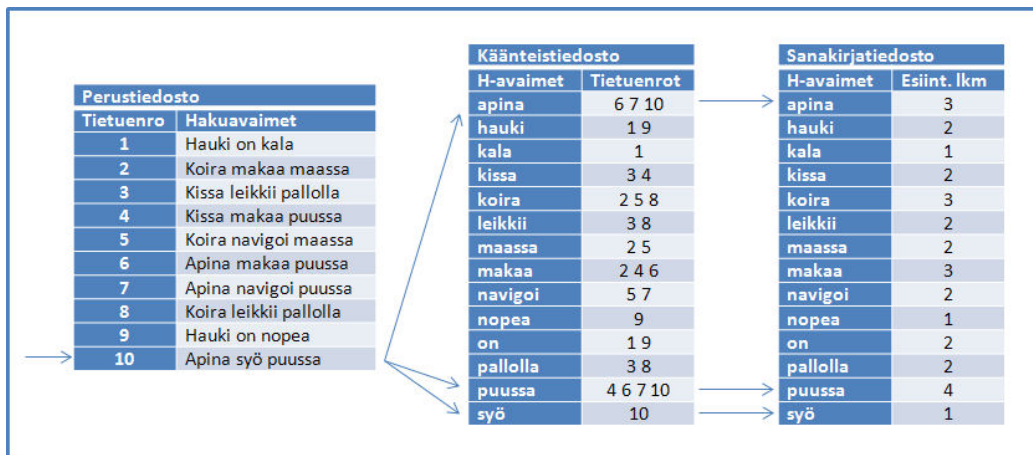
1. datan hankinta
2. datan läpikäynti ja esikäsittely
3. hakemistojen luominen
4. hakukyselyjen prosessointi.

Ensimmäisestä vaiheesta huolehtivat suurilta osin *hakurobotit* (web crawler), jotka käyvät läpi internetin sivustoja erittäin suurella laskentateholla ja tallentavat keräämänsä datan. Sivustojen hankinnan jälkeen toisessa vaiheessa niistä kerätään tietoa ja suoritetaan erilaisia toimintoja, joissa esimerkiksi poistetaan mahdolliset kaksoiskappaleet ja roskapostisivustot. Jokaiselle sivustolle määrätään oma uniikki numero, jota kutsutaan *dokumentin tunnistenumeroiksi* (docID). Jokainen dokumenttia kuvaava tunnistenumero vastaa yhtä URL-osoitetta eli yhtä dokumenttia. [4, 7]

Kolmannessa vaiheessa indeksointifunktiot lukevat ja käyvät lävitse dokumentteja, joista jokaisesta muodostetaan joukko sanaesiintymiä. Näistä joukoista käytetään nimitystä *osuma* (hit). Jokainen osuma kertoo sitä vastaavan sanan, sijainnin, fonttikoon ja sanan tyylin. Osumat tallennetaan osittain järjestettyihin peräkkäistiedostohakemistoihin niitä vastaavien dokumenttien tunnistenumeron perusteella. [4]

Toinen tärkeä funktio indeksoinnissa on kaikkien sivustolla olevien linkkien läpikäynti ja niiden tallentaminen linkkitiedostoon. Tämä tiedosto sisältää lopulta tarpeeksi tietoa jokaisesta linkistä, esimerkiksi mistä ja mihin se osoittaa. Linkkitiedostosta muodostetaan kaikista relatiivisista eli suhteellisista URL-osoitteista absoluuttinen URL, josta saadaan tiedostoa vastaava dokumentti tunnistenumero ja tiedosto voidaan asettaa peräkkäistiedostohakemistoon oikealle paikalle. Absoluuttinen osoite on viittaussijainnista riippumaton osoite, joka alkaa aina protokollalla (esim. "http://"). Ab-

soluuttinen osoite voi olla esimerkiksi "https://www.google.fi/". Suhteellinen osoite koostuu absoluuttisesta osoitteesta ja sen jälkeen tulevista palvelimen alikansioista. Lopulta peräkkäistiedostoista voidaan muodostaa tietokanta sivustoista, jotka koostuvat dokumenttien tunnistenumeropareista ja niiden perusteella voidaan laskea sivustolle PageRank-arvo. Google käyttää PageRank-järjestelmää sivustojen järjestämiseen niiden luotettavuuden ja laadukkuuden perusteella. [4, 8]



Kuva 1. Tallentaminen käänteistiedostoon [16].

Peräkkäistiedostoista haku on erittäin raskasta ja vie paljon aikaa. Tämän vuoksi hakukoneen on muodostettava peräkkäistiedostoista *käänteistiedostot* (inverted index), joita varten peräkkäistiedostot on uudelleenjärjestettävä käänteistiedostoksi jokaisen niissä esiintyvän sanan mukaan. Jokaiselle sanalle määrätään *tunnistenumero* (wordID). Kuva 1 havainnollistaa, kuinka hakukone luo itselleen käänteistiedoston ja sanakirjatiedoston. Kuvassa 1 esitelty perustiedosto vastaa yhtä hakukoneen peräkkäistiedostoa. Hakukoneiden toiminnan jaottelun neljännessä kohdassa, hakukyselyiden prosessoinnissa, hakukone käyttää luomaansa sanakirjaa ja PageRank-tietokantaan apunaan prosessoidessaan käyttäjän syöttämiä hakukyselyitä. [4]

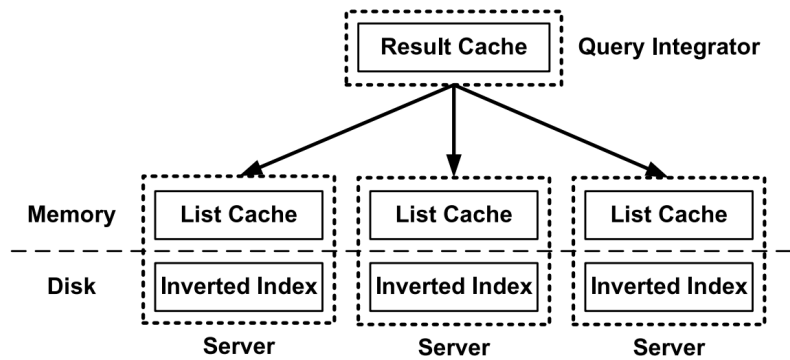
Hakukoneet toimivat nykyään erittäin pitkälti samalla tavalla kuin Brin ja Page [4] ideoivat artikkelissaan. Nykypäivänä tietotekniikan kehittyessä yksittäiset tekniikat parantuvat, mutta yleisesti ottaen hakukoneen toiminta on pysynyt samankaltaisena. Artikkelissa esiteltiin myös jatkokehitysideoita, joista yksi oli välimuistin hyödyntäminen hakutietokantojen rakentamisessa [4].

3 Välimuistin hyödyntäminen hakukoneissa

Välimuistilla (cache) tarkoitetaan pientä ja erittäin nopeaa muistia, jota tietokoneet tai ohjelmistot käyttävät säilömään väliaikaista tietoa tietyksi ajan-

jaksoksi. Väliaikainen tieto, jota säilötään välimuistissa, on yleensä osa päämuistin tiedosta. Tiedon hakeminen välimuistista on hyödyllistä, sillä tietojen hakemiseen välimuistista kuluu yleensä 10 - 25 prosenttia ajasta, joka kuluu päämuistista hakemiseen. [14]

Hakukoneiden kannalta välimuistin käyttämisestä on paljon hyötyä ja sitä on tutkittu paljon. Internetin kasvaessa jatkuvasti hakukoneet joutuvat luomaan hakemistoja miljardeista sivustoista ja yhden hakutuloksen arvioimiseen menee täten enemmän aikaa. Näin ollen välimuistien käyttö nykyaikaisissa hakukoneissa on elintärkeää, jotta hakutulokset saadaan palautettua käyttäjälle mahdollisimman nopeasti. [3]



Kuva 2. Kaksitasoinen arkkitehtuuri, jossa tulosten varastointi tapahtuu kyselyintegraattorilla ja hakemiston varastointi yksittäisten palvelinten muistissa [17].

Kuva 2 havainnollistaa, kuinka hakukone hyödyntää välimuistia kahdella päätasolla. Käyttäjän hakukyselyn vastaanottava ja eteenpäin välittävä kyselyintegraattori tarkastaa, löytyykö sama hakukysely ja sen tulokset integraattorin omasta välimuistista. Jos kysely on lähetetty aiemmin välimuistin aikajakson sisällä, voidaan kyselyn tulos palauttaa käyttäjälle uudelleenlaskematta hakutuloksia. Tätä tapaa hyödyntää välimuistia kutsutaan *tulosten varastoinniseksi* (result cache). Toinen tapa hyödyntää välimuistia on *hakemiston varastointi* (list cache), joka on käytössä alemmalla tasolla kuin tulosten varastointi. Hakemiston varastoinnissa hakukoneen yksittäinen palvelin muodostaa hakukyselyistä käänteistiedostoja useimmiten käytetyistä hakutermeistä. Jokaista hakutermiä ei voi säilöä välimuistiin, sillä sen tila ei riittäisi siihen. Tästä johtuen hakukoneen on myös päätettävä, mitä hakutermejä se pitää tärkeänä ja säilöttävä ne. [3, 7]

Välimuistia on kahdenlaista: *staattista* (static cache) ja *dynaamista* (dynamic cache). Staattinen välimuisti on yhteydetöntä ja vain luettavaa muistia, johon ei voi kirjoittaa jatkuvasti. Se säilöo tietoa edellisistä ja yleensä ensisijaisesti yleisimmistä hakukyselyistä. Hakulokeista saatavan tiedon avulla välimuistissa olevaa tietoa päivitetään säännöllisesti. Dynaaminen

välimuisti säilöö taas tietoa tiettyjen asetettujen sääntöjen perusteella. Jokaisen uuden hakukyselyn tullessa hakukone arvioi, onko kyselyllä enemmän arvoa kuin jollakin vanhalla kyselyllä, joka on säilötty välimuistiin. Jokaisen kyselyn kohdalla päätetään, voidaanko jokin vanha tieto poistaa välimuistista, jotta saadaan uutta tietoa tallennettua välimuistiin. Tähän käytetään erilaisia välimuistin hylkäämisalgoritmeja, jotka esitellään tarkemmin luvussa 3.1. Hakukoneella on tarkoituksena saada mahdollisimman relevanttia tietoa tallennettua välimuistiin, jotta hakutulosten näyttäminen käyttäjälle vie mahdollisimman vähän resursseja. [3, 7, 10]

Yleisimpiä tapoja vertailla välimuistiin varastoimisen tehokkuutta ovat *osumat* (cache hit) ja *ohitukset* (cache miss) sekä *kustannussäästö* (cost saving). Välimuistiin tallennetaan yleensä kyselyitä ja hakutermejä, joista varastoidaan hakukyselyiden tuloksia. Toinen tapa on tallentaa kokonaisia hakemistoja alemmalla tasolla. Näitä tapoja tutkitaan tarkemmin seuraavaksi.

3.1 Välimuistin hylkäämisalgoritmit

Voidaksemme tarkentaa erilaisia tekniikoita välimuistiin säilömiseen on ensin ymmärrettävä, mitä tarkoittavat *välimuistin hylkäämisalgoritmit* (cache eviction algorithm), jotka toimivat dynaamisessa välimuistissa ja ylläpitävät olennaista tietoa välimuistissa. Hylkäämisalgoritmi päättää, mitä tietoa säilytetään ja poistetaan välimuistista sen ollessa täynnä. [7]

Kaksi yleisintä algoritmia tähän tarkoitukseen ovat *LFU* (Least Frequently Used) ja *LRU* (Least Recently Used). LRU-algoritmissa välimuistissa oleviin tiedostoihin on lisätty aikaleima siitä, koska sitä on viimeksi käytetty ja sen perusteella voidaan poistaa vanhin tieto välimuistissa. Tämä algoritmi on kaikista käytetyin välimuistin hylkäämisalgoritmeista tietotekniikassa. LFU-algoritmi taas poistaa tiedon, jota on käytetty kaikista vähiten. Jokaiselle välimuistiin tallennetulle asialle on lisätty tieto siitä, kuinka monta kertaa tietoa on haettu välimuistista. Tämän perusteella voidaan poistaa vähiten käytetty tieto. [3, 5, 7]

Hylkäämisalgoritmien suorituskykyjen välillä on eroja ja niiden käyttöön vaikuttaa suuresti välimuistin rakenne. Näiden tekniikoiden ymmärtäminen on tärkeää, koska seuraavissa kohdissa käsitellään erilaisia välimuistirakenteita, joissa käytetään erilaisia välimuistin hylkäämisalgoritmeja ja vertaillaan niiden tuloksia. Näitä hylkäämisalgoritmeja on erilaisia ja tutkijat ovat kehittäneet omia versioita LRU- ja LFU-algoritmeista. Kehitetyt tekniikat yleensä perustuvat esiteltyihin yleisimmin käytettyihin hylkäämisalgoritmeihin.

3.2 Staattinen välimuistin käyttö

Baeza-Yates ja muut [3] tutkivat, kuinka hakemistojen varastoimisella voidaan saavuttaa parempia tuloksia kuin kyselyiden varastoimisella. Heidän

mukaan kyselyiden varastoisissa suurin ongelma on kyselyiden yksilöllisyys. Samoja kyselyitä ei välttämättä haeta usein, joten tästä muodostuu ongelma, jos käytössä on vain dynaaminen välimuisti. Tällöin hylkäämisalgoritmit saattavat poistaa kyselyn välimuistista, koska välimuistin tila on rajallinen ja kyselyä ei haeta usein. Toinen ongelma tässä tilanteessa on yksittäiset kyselyt, koska lähes puolet heidän tutkimuksessa käyttämästään kyselydatasta oli yksittäisiä kyselyitä, jotka eivät toistuneet uudelleen.

3.2.1 Hakukyselyjen vs. hakutulosten varastointi

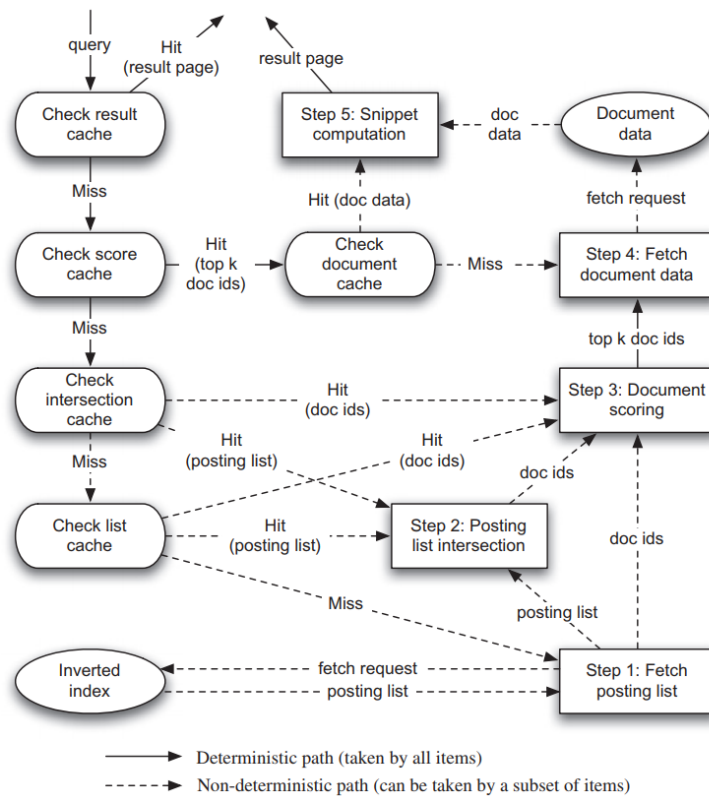
Baeza-Yates ja muut [3] osoittavat tutkimuksessaan, että hakemistojen varastoisella saavutetaan parempi välimuistin osumatarkkuus kuin varastoisella kyselyjen tuloksia. He vertailivat dynaamisen ja staattisen välimuistin eroa käyttämällä samaa algoritmia päättämään, mitä hakemistoja pitäisi varastoida. Staattinen välimuisti tuotti parempia osumatarkkuuksia pienemmällä välimuistin koolla, koska se ei vaadi avukseen hylkäämisalgoritmeja ja on täten tehokkaampi arvioimaan kyselyitä. Kuitenkin koon kasvaessa sekä dynaaminen että staattinen välimuisti toimivat yhtä hyvin.

Baeza-Yates ja muut [3] tutkivat, kuinka suuri osa välimuistista kannattaa hyödyntää hakemistojen ja kuinka suuri osa kyselyiden tulosten varastoisuuteen. He käyttivät kehittämänsä algoritmia, jolla he tutkivat dynaamisten ja staattisten välimuistien eroja. He mittasivat erikokoisten staattisten välimuistien keskimääräistä vasteaikaa ja totesivat saavansa parhaan tuloksen, kun 70 prosenttia välimuistin koosta käytettiin varastoimaan kyselyiden vastauksia ja 30 prosenttia käytettiin hakemistojen varastointiin.

Baeza-Yates ja muut [3] käyttivät tutkimuksessaan kaksitasoisia välimuistia, kuten aiemmin esiteltiin kuvassa 2. Ozcan ja muut [10] esittelivät tutkimuksessaan uuden viisitasoisen tavan hyödyntää staattista välimuistia hakukoneen käsitellessä kyselyitä. Viisitasoinen välimuisti sisältää samat toiminnot kuin kaksitasoinen, mutta siihen on lisätty tulosten ja hakemistojen väliin uusia tasoja, joihin tallennetaan erityyppisiä tietoja.

3.2.2 Viisitasoinen välimuistiarkkitehtuuri

Yleensä välimuistien tehokkuuteen liittyvissä tutkimuksissa käytetään kaksitai kolmetasoista arkkitehtuuria. Kuva 3 esittää, kuinka viisitasoinen välimuistiarkkitehtuuri toimii. Kaksi- ja kolmitasoiset arkkitehtuurit toimivat samalla tavalla, mutta niissä jätetään viisitasoisesta arkkitehtuurista tasoja pois. Kaksitasoiseen arkkitehtuuriin kuuluu *tulosten* (result cache) ja *hakemistojen* (list cache) varastoisuus. Kolmitasoiseen arkkitehtuuriin kuuluu kaksitasoisen arkkitehtuurin lisäksi esilaskettujen *hakemistojen risteymien* (intersection cache) varastoisuus. Viisitasoiseen arkkitehtuuriin on kuitenkin lisätty *esilasketut arvot* (score cache) suosituimmille dokumenteille ja niiden perusteella *varastoidut dokumentit* (document cache). [10]



Kuva 3. Kyselyn kulku viisitasoisessa staattisessa välimuistiarkkitehtuurissa [10].

Kuvassa 3 on kuvattu viisitasoisen välimuistiarkkitehtuurin työn kulkua. Jos kyselyllä ei löydetä tasolta *osumia* (hit), tapahtuu *välimuistin ohitus* (miss). Tässä tapauksessa siirrytään alaspäin seuraavalle tasolle, kunnes löydetään osuma. Osuman löytyessä välimuistista voidaan dokumentti palauttaa paljon tehokkaammin kuin hakemalla se suoraan levyllä tallennetusta käänteisindeksistä. Tämänkaltaisen välimuistiin varastoiminen tapahtuu yhdellä solmulla ja eri tasojen koot voivat muuttua koko välimuistin koon pysyessä kuitenkin samana riippuen palvelimen välimuistin koosta. [10]

Ozcan ja muut [10] saivat parhaan tuloksen viisitasoisesta arkkitehtuurista käyttämällä ahnetta valitsemisalgoritmia, jonka tehtävänä oli valita jokaisen välimuistitason koko ja niihin varastoitavat esilasketut arvot. Valitsemisalgoritmi optimoi välimuistin koon prosenttiosuudet välimuistin kokonaiskapasiteetista jokaiselle eri tasolle. Algoritmin avulla rakennetut staattiset välimuistit koostuivat tuloksista (18 %), dokumenttien arvosanoista (2 %), hakemistojen risteymistä (15 %), hakemistoista (60 %) ja dokumenteista (5 %). Näin jaetulla välimuistilla saatiin 18,4 ja 9,1 prosentin parannukset kaksi- ja kolmitasoisiiin välimuisteihin, kun mitattiin koko kyselyn prosessointiaikaa. Heidän mukaansa parannukseen vaikuttivat suuresti käyttämänsä tiukasti rakennettu arkkitehtuuri, joka toimii yhdellä hakukoneen solmulla. Suurem-

missä hakukoneissa välimuistit on hajautettu käyttämään useampia solmuja yhdessä, joka voi tuoda lisää viivettä hakukyselyiden suorittamiseen.

Molemmista tutkimuksista on erityisen tärkeää kuitenkin huomata samankaltaiset prosenttiosuudet hakemistojen ja tulosten varastoimiselle, joita pidetään monissa tutkimuksissa tärkeimpinä osina välimuistiin varastoimisessa. [3, 10]

3.3 Dynaaminen välimuisti

Dynaaminen välimuisti, kuten aiemmin on esitelty, säilöö tietoa ennalta määrättyjen sääntöjen perusteella. Yleensä sitä käytetään säilömään suosittujen kyselyiden tuloksia. Dynaamisen välimuistin tarkoituksena on selvittää tehokkaasti suurista sarjoista kyselyitä. Kyselyitä voi tulla yhtäkkiä sarjoissa samasta aiheesta jostakin syystä, esimerkiksi sosiaalisen median ilmiöistä tai vaikkapa luonnonilmiöistä. Todellisuudessa hakukone yleensä jakaa tulosten tallentamisen staattisten ja dynaamisten välimuistien välille hyödyntäen molempien vahvuuksia. Toinen tapa käyttää dynaamista välimuistia tehokkaasti on käyttää niin suurta välimuistia, että se tulee harvoin täyteen ja välimuistien hylkäämisalgoritmeja ei tarvitse käyttää usein. [1]

Altingovde ja muut [1] esittelivät tutkielmassaan uuden tavan käyttää hyödyksi dynaamista välimuistia. He kutsuivat tätä tapaa nimellä *Second Chance*. Normaalin tulosten säilömisestä lisäksi välimuistiin säilötään toiselle tasolle dokumenttien tunnisteita eli docID:ta. Ne kertovat hakukoneelle, mitä dokumentteja hakukysely vastaa, jotta hakukone voi hakea dokumentit ilman koko kyselyn läpikäymistä. Dokumenttien tunnisteiden tallentamisesta on hyötyä, sillä niitä voidaan tallentaa samaan tilaan paljon enemmän kuin hakukyselyiden tuloksia.

Second Chance -tallennusideassa on tarkoituksena tarkastella ensimmäisenä edellisten hakukyselyiden tuloksien osuutta välimuistista. Jos kysely on jo haettu entuudestaan ja sen tulokset löytyvät välimuistista, voidaan hakukyselyn tulokset palauttaa käyttäjälle heti. Kyselyn saapuessa välimuistin toiselle tasolle voidaan tarkistaa, onko kysely haettu aikaisemmin ja löytyvätkö kyseisen kyselyn tulosten dokumenttien tunnisteet välimuistista. Dokumenttien tunnisteiden löytyessä hakukone voi hakea suoraan tulokset tunnisteiden perusteella. Tällaisella tallennusmallilla kyselyille annetaan ikään kuin toinen mahdollisuus ennen niiden käsittelyä. [1]

Altingovde ja muut [1] saivat artikkelissaan parhaan tuloksen, kun noin 20 prosenttia välimuistin koosta käytettiin dokumenttien tunnisteiden tallentamiseen. He osoittivat, että isommilla osuuksilla hakukone hyödynsi liikaa dokumenttien tunnisteita eikä välimuistissa ollut tarpeeksi tilaa tallentaa hakukyselyiden tuloksia, mikä johti tehokkuuden heikkenemiseen. Heidän mukaansa dokumenttien tunnisteiden tallentaminen on vähemmän tehokasta, sillä niissä voidaan tallentaa enemmän suoraan kyselyiden tuloksia välimuistiin, joten tunnisteille ei ole tarvetta.

3.4 Dokumenttien esihakeminen välimuistiin

Esihakeminen (prefetching) tarkoittaa dokumenttien tai tietojen hakemista välimuistiin ennen kuin kysely niitä vaatii. Esihakeminen on hyödyllistä, sillä sen avulla voidaan hakea taustalla tietoa välimuistiin samalla, kun käyttäjälle näytetään alkuperäinen haettu sivu. Esihakemisen avulla saadaan tehostetua hakukoneen toimintaa, koska kyselyitä vastaavat sivustot löytyvät yleensä esihaettuina välimuistista. Second Chance -tallennustavan tekijät jatkoivat alkuperäistä tutkimustaan tutkimalla, kuinka dokumenttien esihakeminen vaikuttaa välimuistin tehokkuuteen. He käyttivät välimuistista avukseen dokumenttien tunnisteille varattua osuutta, jonne säilöivät esihaettua tietoa käyttäjän syöttämien kyselyiden perusteella. [11]

Ozcan ja muut [11] vertailivat tutkimuksessaan kahta erilaista esihakemisalgoritmia. Ensimmäinen algoritmi haki käyttäjän kyselyn perusteella kiinteään *esihakemiskertoimen* (prefetching factor) määrän verran seuraavia hakutuloksia ja tallensi niiden docID:t tunnisteita sisältävään välimuistin osaan. Jos kyselyä vastaava dokumentti löytyi jo välimuistista, ei ollut tarpeellista esihakea seuraavia sivuja. Kuitenkin kyselyn tuottaessa välimuistin ohituksen, haettiin tulospöytä välimuistiin kyselyä vastaava sivu ja docID-välimuistiin tuloksia vastaavista sivuista seuraavat aina esihakemiskertoimeen asti.

Toinen Ozcan ja muut [11] käyttämistä esihakemisalgoritmeista oli mukautuva algoritmi, jonka Fagni ja muut [6] esittelivät tutkimuksessaan. Mukautuva algoritmi käyttää apunaan dokumenttien ennalta laskettua arvoa ja tietoa siitä, löytyykö kyselyä vastaava dokumentti jo välimuistissa olevista tuloksista. Koska on yleisesti tiedossa, että harva käy muilla kuin hakukoneiden ensimmäisellä hakutulossivulla, mukautuva algoritmi käyttää tätä tietoa hyödykseen hakiessaan dokumentteja välimuistiin. Käyttäjän hakiessa ensimmäistä hakutulossivua, jota ei löydy välimuistista, esihakemisalgoritmi hakee tulossivun lisäksi myös seuraavan tulossivun ja tallentaa sen docID-välimuistiin. Jos käyttäjä pyytää toista hakutulossivua, hakee algoritmi seuraavat tulossivut välimuistiin aina esihakemiskertoimeen asti välittämättä siitä, onko toinen hakutulossivu jo välimuistissa. Käyttäjän haluaman sivuston ollessa jokin muu kuin ensimmäinen tai toinen hakutulossivu algoritmi esihakee seuraavia sivustoja esihakemiskertoimeen asti vain, jos haettavaa sivua ei löydy jo välimuistista. [6, 11]

Esihakemalla sivustoja Second Chance -välimuistiin saavutettiin 13,2 prosentin parannus kyselyiden läpikäyntiin käyttämällä kiinteää algoritmia ja 25,2 prosentin parannus käyttämällä mukautuvaa algoritmia verrattuna ilman esihakemista toimivaan välimuistiin, kun mukaan laskettiin esihakemisesta koituvat kustannukset. Parhaimmat tulokset saatiin myös käyttämällä 50 - 70 prosenttia välimuistin koosta dokumenttien tunnisteiden varastointiin. Tätä Ozcan ja muut [11] perustelivat siten, että esihaettu data varastoitettiin dokumenttien tunniste välimuistiin ja vain tärkeimmät varastoitettiin tulospöytä välimuistiin, joten tunnisteille tarvittava osuus välimuistista oli paljon

suurempi kuin ilman esihakemista. [11]

4 Koneoppiminen hakukoneiden välimuistissa

Koneoppiminen (machine learning) on tietojenkäsittelytieteen ala, joka mahdollistaa ohjelmiston oppimaan itse tekemisistään ja parantamaan näin ollen itseään jatkuvasti. Hakukoneissa koneoppimista käytetään tuottamaan käyttäjille osuvia hakutuloksia mahdollisimman nopeasti ja tehokkaasti. Hakukoneet käyttävät koneoppimista esimerkiksi tekstintunnistukseen, jossa ohjelmisto tunnistaa ja korjaa kirjoitusvirheitä sekä tunnistaa eri kirjoitusasuja. [12, 13]

Tässä luvussa tutustumme erilaisiin osa-alueisiin, joissa hakukoneet käyttävät koneoppimista hyväkseen käyttäjän kyselyiden tunnistamiseen ja erilaisten tietojen varastoimisessa välimuistiin.

4.1 Yksittäisten hakukyselyiden tunnistaminen

Hakukyselyiden ja niiden tulosten varastoiminen välimuistiin on erityisen tärkeää hakukoneen tehokkuuden kannalta. Useasti kuitenkin ongelmana välimuistiin varastoimisen kannalta on kyselyiden yksilöllisyys. Baeza-Yates ja muiden [3] mukaan heidän käyttämästään testidatasta lähes puolet oli yksittäisiä kyselyitä, jotka eivät toistuneet uudelleen. Kyselyt muodostuvat ongelmaksi, kun niiden tulokset sijaitsevat välimuistissa ja välimuistin hylkäämisalgoritmi poistaa jonkin uudelleen toistuvan kyselyn tiedot yksittäisen sijaan. [3, 11]

Ozcan ja muut [11] käyttivät Second Chance -tutkimuksessaan apunaan erilaisia koneoppimisalgoritmeja tunnistamaan yksittäisiä hakukyselyitä ja säilömään ohjelmiston yksittäisenä pitämät hakukyselyt vain dokumenttien tunnisteille varattuun osuuteen välimuistista eli docID-välimuistiin. Tällä tavalla heidän tarkoituksena on säilöä välimuistiin jokin tieto yksittäisistäkin kyselyistä, koska koneoppimisalgoritmi on saattanut luokitella joitakin kyselyitä väärin. Kyselyiden tulosten tai dokumenttien tunnisteiden ollessa välimuistissa yksittäisen kyselyn tulosten näyttämiseen kuluu paljon vähemmän aikaa kuin tulosten uudelleen hakuun indeksistä. [11]

Koneoppimista käyttämällä kyselyiden yksilöllisyyden tunnistamiseen saatiin jopa 9,3 prosentin parannus Second Chance -välimuistin hakutulosvälimuistin osumatarkkuuteen. Tunnisteiden osuus välimuistista kärsi osumatarkkuudessa samalla, koska tunnisteissa oli enemmän yksittäisiä ja näin ollen hyödyttömiä tietoja kyselyiden kannalta. Toinen asia, mikä vaikutti osumatarkkuuteen, oli dokumenttien siirtyminen ja jääminen hakutulosvälimuistiin, sillä toistuvat kyselyiden tunnisteet docID-välimuistista varastoi- tiin hakutulosten välimuistiin. Jos dokumentti oli vain tunnistevälimuistissa ja kysely vaati tuloksiinsa sen, hakukone osasi tallentaa sen tulosvälimuis-

tiin ja poistaa sen tunnistevälimuistista. Ottamalla huomioon algoritmin kokonaiskustannus saatiin myös 2,4 - 6,2 prosentin parannus hakukyselyiden käsittelyaikaan, joka voi luoda isossa skaalassa huomattavia säästöjä hakukoneiden toimintaan. [11]

4.2 Koneoppiminen hakutulosten varastoinnissa

Koneoppimista voi käyttää hyödyksi myös hakutulosten varastoimisessa. Ohjelmisto voi itse muodostaa tuloksista haluamiansa tietoja ja varastoida ne niiden perusteella välimuistiin. Kucukyilmaz ja muut [9] käyttivät erilaisia koneoppimisalgoritmeja yrittäessään parantaa tulosten varastoimista staattiseen ja dynaamiseen välimuistiin tarkoituksenaan parantaa välimuistien osu-matarkkuutta.

Type	Feature	Description
Query	QUERY_LENGTH	Number of characters in the query string
	TERM_COUNT	Number of terms in the query string
	PROTOCOL_PRESENT	Presence of a protocol string in the query string
	DOMAIN_PRESENT	Presence of a domain name in the query string
	MISSPELLED	Presence of misspelling
	AVG_TERM_LENGTH	Average number of characters in query terms
	PAGE_NUMBER	Requested result page number
	QUERY_TIME	Hour of the day the query was submitted
Session	USER_LOGGED_IN	Whether the user is logged in or not
	CTR	Clickthrough rate
	CTR_TOP_ONE	Clickthrough rate for the top result
	HIT_COUNT	Number of matching results
	DAYTIME_COUNT	Daytime query frequency
	TIME_COMPATIBLILTY	Daytime/nighttime compatibility
Index	MIN_POSTING_COUNT	Number of postings for the rarest term
	MAX_POSTING_COUNT	Number of postings for the most common term
	AVG_POSTING_COUNT	Average posting list size of query terms
Term freq.	MIN_TERM_FREQ_MINUTE	Min. query term freq. in the last one minute
	MAX_TERM_FREQ_MINUTE	Max. query term freq. in the last one minute
	AVG_TERM_FREQ_MINUTE	Avg. query term freq. in the last one minute
	MIN_TERM_FREQ_HOUR	Min. query term freq. in the last one hour
	MAX_TERM_FREQ_HOUR	Max. query term freq. in the last one hour
	AVG_TERM_FREQ_HOUR	Avg. query term freq. in the last one hour
	MIN_TERM_FREQ_DAY	Min. query term freq. in the last one day
	MAX_TERM_FREQ_DAY	Max. query term freq. in the last one day
	AVG_TERM_FREQ_DAY	Avg. query term freq. in the last one day
Query freq.	QUERY_FREQ	Query frequency
	QUERY_FREQ_MINUTE	Query frequency in the last one minute
	QUERY_FREQ_HOUR	Query frequency in the last one hour
	QUERY_FREQ_DAY	Query frequency in the last one day

Kuva 4. Koneoppimisalgoritmin hyödykseen käyttämiä ominaisuuksia välimuistiin varastoimisessa [9].

Kuvassa 4 esitellään ominaisuuksia, jotka Kucukyilmaz ja muut [9] määrittivät tutkimuksessaan. He käyttivät tutkimuksessaan hyväkseen myös yksittäisten kyselyiden tunnistamiseen suunnattua algoritmia, joka auttoi mää-

rittelemään kyselyiden toistuvuuden. Käyttämällä määriteltyjä ominaisuuksia koneoppimisalgoritmit pystyivät helpommin vertailemaan kyselyitä toisiinsa ja niiden perusteella päättämään, mitkä olivat relevantteja kyselyitä sekä minkä kyselyiden tuloksia kannattaa varastoida välimuistiin tulevaisuutta varten. [9]

Kucukyilmaz ja muut [9] saavuttivat tutkimuksessaan vain noin yhden prosentin parannuksen välimuistin osumatarkkuuteen. Heidän mielestään parannus on kuitenkin merkittävä, sillä yhden prosentin parannuksella voidaan saavuttaa merkittäviä rahallisia säästöjä erittäin suurissa hakukoneissa. He kuitenkin totesivat tutkimuksessaan, että heidän esittelemällään koneoppimistavalla varastoida tietoja välimuistiin voidaan saavuttaa jopa 9,6 prosentin parannus osumatarkkuuteen, jos tapaa optimoidaan vastaamaan tarkemmin hakukoneen tarpeita. [9]

4.3 Tietojen varastoiminen tai hylkääminen

Suurin kysymys välimuistiin varastoimisessa on, onko varastoitava tieto relevanttia tulevaisuuden hakukyselyiden kannalta. Sulaiman ja muut [15] pyrkivät tutkimuksessaan yhdistämään erilaisia koneoppimismalleja välimuistipalvelimelle avuksi tietojen luokitteluun. Heidän tarkoituksenaan on tutkia, miten erilaiset koneoppimismallit auttavat palvelinta luokittelemaan varastoitavia tietoja ja päättämään, onko jokin tieto varastoinnin arvoista.

Sulaiman ja muut [15] eivät käyttäneet tutkimuksessaan alustana hakukonetta vaan tiedonsiirtoa palvelimelta käyttäjille. Tämä tilanne vastaa kuitenkin hakukoneen toimintaa suurilta osin ja koneoppimista voidaan käyttää samalla tavalla luokittelemaan tietoa hakukoneen palvelimella päättämään, mitä tietoa on relevanttia varastoida välimuistiin. Tutkimuksessa huomattiin, että käyttämällä koneoppimismalleja päättämään, tallennetaanko jokin tiedosto välimuistiin, saavutettiin huomattavia parannuksia palvelimen toimintaan. Heidän tutkimuksessaan parhaimmat koneoppimismallit saavuttivat lähes 100 prosentin tarkkuuden tiedostojen varastoimisessa välimuistiin. [15]

Koneoppimismallit käyttivät hyväkseen samankaltaisia ominaisuuksia ja tietoja päättäessään tiedostojen varastoimisesta kuin hakukoneet, joten erilaisia koneoppimistekniikoita voidaan käyttää myös hakukoneiden välimuistipalvelimilla päättämään, mitkä hakutulokset ovat relevantteja lähitulevaisuudessa. Sulaiman ja muut [15] tutkimuksen tuloksia ei voi verrata suoraan hakukoneisiin, koska testiympäristö ja -data olivat erilaisia, mutta tutkimuksesta voidaan nähdä kuinka paljon koneoppimisesta voi olla hyötyä kompleksisten päätösten tekemisessä.

5 Yhteenveto

Välimuistin käyttö hakukoneissa on erityisen tärkeää niiden toimivuuden kannalta. Nykyaikana tietoa on ympärillämme erittäin paljon ja sen kaiken läpikäymiseen ja varastointiin vaadittava laskentateho on erittäin suuri. Välimuistin hyödyntäminen auttaa hakukoneita säilömään jo haettua tai muuten relevanttia tietoa nopeasti saatavilla olevaan muistiin.

Tutkimuksessani esittelin erilaisia tapoja säilöä tietoja hakutuloksista ja -kyselyistä välimuistiin sekä osoitin välimuistin hyödyllisyyden ja tarpeen hakukoneille. Lisäksi esittelin erilaisia tekniikoita, joilla hakukone voi varastoida tietoa ja käyttää sitä tehokkaammin näyttäessään käyttäjälle hakukyselyä vastaavan tuloksen. Tutkielmassani kävi ilmi, että varastoidakseen tietoja välimuistiin, hakukone tarvitsee avukseen erilaisia apuvälineitä pitääkseen välimuistissa olevan tiedon relevanttina.

Hakukoneen toiminnan kannalta on tärkeää, että välimuistissa olevat tiedot ovat relevantteja. Tähän tarkoitukseen kävin tutkielmassani läpi tietojen esihakemista välimuistiin, jolla voidaan ennakoida käyttäjän seuraavaksi haluamia kyselyitä. Esittelin myös tutkielmassani erilaisia mahdollisuuksia, joilla koneoppimista voidaan käyttää apuna tietojen varastoisessa välimuistiin.

Tulevaisuudessa tiedon lisääntyessä jatkuvasti tutkielmassani esiteltyiden tekniikoiden kehittäminen on tärkeää hakukoneiden toiminnan kannalta. Jatkotutkimuksessa tulisi kehittää koneoppimista, sillä sen avulla saavutetaan todennäköisesti suurin hyöty tulevaisuudessa. Tiedon kehittyessä jatkuvasti hakukoneita on kehitettävä koko ajan ja koneoppiminen on osoittautunut erittäin hyödylliseksi työkaluksi hakukoneiden tulevaisuuden kannalta.

Viitteet

- [1] Ismail Sengör Altingovde, Rifat Ozcan, Berkant Barla Cambazoglu, and Özgür Ulusoy. 2011. Second chance: A hybrid approach for dynamic result caching in search engines. In: Clough P. *et al.* (eds.), *Advances in Information Retrieval (ECIR 2011), Lecture Notes in Computer Science* 6611, Springer, 510-516.
- [2] Amazon Alexa. www.alex.com/topsites. Checked 03.03.2018.
- [3] Ricardo Baeza-Yates, Aristides Gionis, Flavio Junqueira, Vanessa Murdock, Vassilis Plachouras, and Fabrizio Silvestri. 2007. The impact of caching on search engines. In: *Proc. of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 183-190

- [4] Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* 30, 1-7, 107-117.
- [5] Ehcache. www.ehcache.org/documentation/2.7/apis/cache-eviction-algorithms.html. Checked 04.03.2018.
- [6] Tiziano Fagni, Raffaele Perego, Fabrizio Silvestri, and Salvatore Orlando. 2006. Boosting the performance of Web search engines: Caching and prefetching query results by exploiting historical usage data. *ACM Transactions on Information Systems* 24, 1, 51-78.
- [7] Qingqing Gan and Torsten Suel. 2009. Improved techniques for result caching in web search engines. In: *Proc. of the 18th International Conference on World Wide Web (WWW '09)*, 431-440
- [8] Internetopas. <http://www.internetopas.com/sanasto/>. Checked 29.04.2018.
- [9] Tayfun Kucukyilmaz, Berkant Barla Cambazoglu, Cevdet Aykanat, and Ricardo Baeza-Yates. 2017. A machine learning approach for result caching in web search engines. *Information Processing & Management* 53, 5, 834-834.
- [10] Rifat Ozcan, Ismail Sengör Altingovde, Berkant Barla Cambazoglu, Flavio P. Junqueira, and Özgür Ulusoy. 2012. A five-level static cache architecture for web search engines. *Information Processing & Management* 48, 5, 828-840.
- [11] Rifat Ozcan, Ismail Sengör Altingovde, Berkant Barla Cambazoglu, and Özgür Ulusoy. 2013. Second chance: A hybrid approach for dynamic result caching and prefetching in search engines. *ACM Transactions on the Web* 8, 1, Article 3, 22 pages.
- [12] Kevin Rowe. 2018. How search engines use machine learning: 9 things we know for sure. *Search Engine Journal*. www.searchenginejournal.com/how-search-engines-use-machine-learning/224451/. Checked 29.04.2018.
- [13] Neha Sharma, Rashi Agarwal, and Narendra Kohli. 2016. Review of features and machine learning techniques for web searching. In: *Proc. of the 11th International Conference on Industrial and Information Systems (ICIIS)*, 312-317.
- [14] Alan Jay Smith. 1982. Cache memories. *ACM Computing Surveys* 14, 3, 473-530.

- [15] Sarina Sulaiman, Siti Mariyam Shamsuddin, Ajith Abraham, and Shahida Sulaiman. 2011. Intelligent web caching using machine learning methods. *Neural Network World* 21, 5, 429-452.
- [16] Wikipedia. <https://fi.wikipedia.org/wiki/Käänteistiedosto>. Checked 29.04.2018.
- [17] Jiangong Zhang, Xiaohui Long, and Torsten Suel. 2008. Performance of compressed inverted list caching in search engines. In: *Proc. of the 17th International Conference on World Wide Web (WWW '08)*, 387-396.

Pääkomponenttianalyysi kuvanpakkauksessa

Jukka Lipsanen

Tutkielmassa tarkastellaan pääkomponenttianalyysin käyttöä RGB-kuvien pakkaamiseen. Tutkimuksen osana toteutettiin kaksi pääkomponenttianalyysin perustuvaa kuvanpakkausmenetelmää. Tulosten perusteella pääkomponenttianalyysi sopii ainakin välttävästi kuvanpakkaukseen. Pakkaustehokkuuden parantamiseksi havaittiin hyviksi esiprosessointivaiheiksi jakaa kuva ruutuihin ja muuntaa se YCbCr-väriavaruuteen. DEFLATE-algoritmi osoitautui tehokkaaksi häviöttömäksi pakkausmenetelmäksi pääkomponenttianalyysin yhteyteen.

Avainsanat: Pääkomponenttianalyysi, kuvanpakkaus, SSIM, RGB

1 Johdanto

Digitaalisten kuvien määrä on kasvanut räjähdysmäisesti digitaalisten kameroiden ja tietokoneiden yleistyessä. Tallennustilan rajallisuuden takia on tullut tarpeelliseksi pakata kuvia tilan säästämiseksi. Pääkomponenttianalyysi on Pearsonin 1900-luvun alussa kehittämä lineaarialgebran menetelmä, mikä yleistyi tietokoneiden myötä 1900-luvun loppupuolella [4]. Pääkomponenttianalyysin avulla voidaan löytää datan tärkeimmät komponentit eli akselit millä suurin osa variaatiosta sijaitsee [4].

Pääkomponenttianalyysillä on monia sovelluksia esimerkiksi datan pakkaamisessa ja moniulotteisen datan visualisoinnissa [4]. Pääkomponenttianalyysiä ei kuitenkaan käytetä yleisesti kuvanpakkaukseen. Suositut menetelmät kuten JPEG ja PNG eivät käytä pääkomponenttianalyysiä [6, 14]. Tässä tutkielmassa tutkitaan pääkomponenttianalyysin käyttöä RGB-kuvien pakkaamiseen ja mahdollisia syitä menetelmän epäsuosioon.

Santo [11] tutki pääkomponenttianalyysin käyttöä lääketieteellisten kuvien pakkaamiseen. Artikkelissa toteutettiin yksinkertainen metodi, jossa harmaasävykuvan rivit muodostivat datamatriisin rivit. Metodin tehokkuutta arvioitiin neliöidyn keskivirheen avulla. Artikkelin testiaineisto oli kuitenkin hyvin puutteellinen, sillä siinä käytettiin vain yhtä testikuvaa.

Qian Dun artikkelissa [10] pääkomponenttianalyysiä käytettiin hyperspektraalisten kuvien pakkaamiseen. Hyperspektraaliset kuvat sisältävät monia eri kerroksia, jotka kuvaavat eri aallonpituuksia. Näin kuvan voidaan ajatella olevan kolmiulotteinen. Artikkelissa pääkomponenttianalyysiä käytettiin JPEG2000-metodin yhteydessä. Kehitetyllä menetelmällä oli diskreettiin aallonmuunnokseen verrattuna parempi kuvanlaatu ja se suoriutui paremmin poikkeamien tunnistuksessa.

Nakouri [2] hyödynsi pääkomponenttianalyysiä QR-hajotelman yhteydessä kasvokuvien pakkaamiseen. Artikkelissa kehitettiin pääakselihajotelmaan

perustuva menetelmä ja iteratiivinen menetelmä. Uusia menetelmiä verrattiin muihin pääkomponenttianalyysiin perustuviin menetelmiin. Testeissä arvioitiin muodostuvaa keskivirhettä ja menetelmien sopivuutta kasvun tunnistukseen. Toteutetut menetelmät suoriutuivat testeissä hyvin.

Tutkitussa kirjallisuudessa 2D-pakkausmenetelmien tarkastelu on jäänyt hieman teoreettiselle tasolle. Esimerkiksi Santo [11] tarkasteli pakkaussuhdetta yksinkertaisesti säilytettyjen pääkomponenttien määrän avulla. Tämä jättää huomioimatta datan liukulukuesityksen, sillä liukulukuja tyypillisesti säilytetään 32-bittisessä muodossa. Sen sijaan kokonaislukumuodossa pikselin intensiteetti on usein esitetty 8-bittisenä lukuna, jolloin vaadittava tila on neljä kertaa pienempi yhtä lukua kohden. Tarkastelluissa 2D-kuvien pakkausmenetelmissä ei lisäksi menetelmiä verrattu suosittuihin kuvanpakkausmenetelmiin kuten JPEG- tai PNG-menetelmiin. Näin ollen jää hieman epäselväksi, mikä on kehitettyjen menetelmien käytännön hyödyllisyys.

Luvussa 2 tarkastellaan ensin pääkomponenttianalyysin ymmärtämiseen vaadittavat esitiedot ja sitten itse pääkomponenttianalyysi. Tämän jälkeen käydään digitaalisten kuvien teoriaa hieman läpi ja sitten tarkastellaan erilaisia kuvanlaadun mittareita. Luvun 2 lopuksi käsitellään pääkomponenttianalyysiä hyödyntäviä kuvanpakkausmenetelmiä. Luvussa 3 verrataan kuvanpakkausmenetelmien suorituskykyä keskenään ja suosittuihin JPEG- ja PNG-menetelmiin. Luvussa 4 tuloksia arvioidaan ja tarkastellaan pääkomponenttianalyysin sopivuutta kuvanpakkaukseen. Luvussa 5 tutkielman tulokset kerrataan nopeasti ja tehdään yhteenveto.

2 Menetelmät

2.1 Lineaarialgebra

Määritelmä 2.1. Vektorin b projektio vektorille a on

$$\frac{a^T b}{a^T a} [1].$$

Jos a on yksikkövektori eli $a^T a = 1$, projektio on $a^T b$.

Määritelmä 2.2. Datamatriisi A on $n \times m$ -matriisi, missä rivit edustavat mittauksia ja sarakkeet edustavat mitattavia suureita. Alkio A_{ij} on siis suureen j arvo mittaukselle i .

Määritelmä 2.3. Olkoon A $n \times n$ -matriisi. Matriisin A ominaisarvoyhtälö määritellään kaavalla $Ax = \lambda x$, kun $x \in \mathbb{R}^n \neq 0$ ja λ on matriisin A ominaisarvo. [1]

Määritelmä 2.4. Matriisi C on $m \times m$ kovarianssimatriisi $n \times m$ datamatriisille A , mikäli $C_{ij} = \sigma_{ij}$, $i \neq j$ ja $C_{ii} = \sigma_i^2$, kun σ_{ij} on suureiden i ja j välinen kovarianssi ja σ_i^2 on suureen i varianssi. Otskovarianssimatriisi voidaan

laskea matriisilaskulla

$$C = \frac{1}{m-1} \bar{A}^T \bar{A},$$

missä $\bar{A}_{ij} = A_{ij} - \bar{a}_j$ ja \bar{a}_j on sarakkeen j keskiarvo. [4]

Määritelmä 2.5. Joukko $S = \{s_1, s_2, \dots, s_n\}$ on lineaarisesti riippumaton, mikäli sen vektoreille pätee ekvivalenssi

$$a_1 s_1 + a_2 s_2 + \dots + a_n s_n = 0 \leftrightarrow a_1 = a_2 = \dots = a_n = 0. [1]$$

Määritelmä 2.6. Joukon S lineaarikombinaatioiden joukko $\text{lin}(S)$ on

$$\text{lin}(S) = \{\lambda_1 s_1 + \lambda_2 s_2 + \dots + \lambda_n s_n \mid \lambda_i \in \mathbb{R}, s_i \in S\}. [1]$$

Määritelmä 2.7. Joukko S virittää vektoriavaruuden A , jos $A \subseteq \text{lin}(S)$. [1]

Määritelmä 2.8. Joukko S on vektoriavaruuden A kanta, mikäli se virittää joukon A ja sen vektorit ovat lineaarisesti riippumattomat. [1]

Määritelmä 2.9. Joukko S on vektoriavaruuden A ortonormaali kanta, mikäli se on joukon kanta ja kaikille vektoreille x_i on voimassa

$$\begin{aligned} x_i \cdot x_i &= 1, \\ x_i \cdot x_j &= 0, \text{ kun } i \neq j. [1] \end{aligned}$$

Määritelmä 2.10. Pääkomponenttiavaruus matriisille A on sen pääkomponenttien lineaarikombinaatioiden joukko.

2.2 Pääkomponenttianalyysi

Pääkomponenttianalyysi on matemaattinen keino esittää moniulotteinen data matalammassa ulottuvuudessa. Muunnos suoritetaan siten, että mahdollisimman paljon alkuperäisestä informaatiosta säilytetään. Korkeaulotteiselle datalle on tyypillistä, että eri suureiden välillä on keskinäisiä riippuvuuksia. Näin suurin osa datan vaihtelusta voi sijaita muutamassa pääkomponentissa. [4]

Pääkomponenteille on monenlaisia erilaisia määritelmiä. Usein ne on määriteltä joko kovarianssimatriisin tai korrelaatiomatriisin ominaisvektoreina. [4] Tässä tutkielmassa käytetään kovarianssimatriisimääritelmää. Koska kovarianssimatriisi on symmetrinen matriisi, tiedetään että sen ominaisvektorit ovat toistensa suhteen kohtisuoria. [1]

Näin ollen pääkomponenteista voidaan muodostaa ortonormaali joukko ja selvästi pääkomponenttien joukko on kanta matriisin A rivivektoreille. Näin ollen voidaan $n \times m$ -matriisille A määrittää vaihtoehtoinen m -ulotteinen esitys matriisina B siten, että

$$A = BD,$$

missä B on $n \times m$ -matriisi ja D on $m \times m$ pääkomponenttimatriisi.

Tiedon pakkaamisen näkökulmasta on hyödyllisempää määrittää matala-ulotteinen esitys, mikä säilyttää mahdollisimman paljon matriisin variaatiota. Tämä suoritetaan valitsemalla k ominaisarvoltaan suurinta pääkomponenttia. Näin voidaan määrittää matriisille A k -ulotteinen approksimaatio E

$$A \approx \tilde{A} = EF,$$

missä E on $n \times k$ -matriisi ja F on $k \times m$ pääkomponenttimatriisi. Matriisi \tilde{A} sisältää p % matriisin A variaatiosta ja p voidaan laskea yhtälöstä

$$p = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^m \lambda_i},$$

kun ominaisarvot λ_i on järjestetty suurimmasta pienempään. [4]

Määritelmä 2.11. Matriisin A pääakselihajotelma on

$$A = ULX^T,$$

missä U ja X ovat $n \times r$ - ja $p \times r$ -matriiseja, joilla on ortonormaalit sarakevektorit ja L on diagonaalimatriisi [4].

Pääkomponenttianalyysin kannalta pääakselihajotelma on merkittävä siten, että matriisin U sarakevektorit ovat myös matriisin A pääkomponentteja. Lisäksi matriisin L diagonaalialkiot ovat näitä pääkomponentteja vastaavia ominaisarvoja. Koska pääakselihajotelmaan on kehitetty tehokkaita algoritmeja, se tarjoaa käytännöllisen tavan suorittaa pääkomponenttianalyysi. [4]

2.3 Digitaalinen kuva

Harmaasävyinen digitaalinen kuva voidaan esittää $n \times m$ -matriisina A , missä A_{ij} on kuvan (i, j) pikselin intensiteetti. Värikuvien tapauksessa on digitaalisella kuvalla useampi intensiteettimatriisi, yksi jokaista eri värikomponenttia varten. Tässä tutkielmassa tarkastellaan RGB-kuvia. Kuvien informaatio on siis ilmaistu punaisen, vihreän ja sinisen värin intensiteetin avulla.

Lisäksi tutkielmassa tarkastellaan myös JPEG-menetelmän YCbCr-väriavaruuden käyttöä pakkausvaiheessa. YCbCr-avaruudessa väri ilmaistaan luminanssilla (Y), krominanssilla siniseen (Cb) ja punaiseen (Cr). [5] Krominanssilla tarkoitetaan värin eroa siniseen/punaiseen väriin, jos luminanssi on sama [8].

2.3.1 RGB - YCbCr muunnos

RGB-väriavaruuden pikselin (R, G, B) muunnos YCbCr-väriavaruuteen saadaan kaavoilla

$$\begin{aligned} Y &= 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B, \\ C_b &= -0.1687 \cdot R - 0.3313 \cdot G + 0.5 \cdot B + 128, \\ C_r &= 0.5 \cdot R - 0.4187 \cdot G - 0.0813 \cdot B + 128[5]. \end{aligned}$$

Muunnos toiseen suuntaan tapahtuu seuraavilla kaavoilla:

$$\begin{aligned} R &= Y + 1.402 \cdot (C_r - 128), \\ G &= Y - 0.3441 \cdot (C_b - 128) - 0.7141 \cdot (C_r - 128), \\ B &= Y + 1.772 \cdot (C_b - 128)[5]. \end{aligned}$$

2.4 Kuvanlaadun mittaus

Objektiiviset kuvanlaadun mittarit voidaan jakaa kahteen eri luokkaan alkuperäiskuvan saatavuuden perusteella. Täyden viitekuvan menetelmissä alkuperäinen virheetön kuva on saatavilla, mikä kattaa suurimman osan nykyisistä menetelmistä. Tämän sijaan kuvanlaatua voidaan mitata myös sokeasti ilman viitekuvaa. [13] Kuvanpakkaamisen kannalta mielenkiintoisia ovat lähinnä täyden viitekuvan menetelmät, sillä alkuperäiset kuvat ovat usein saatavilla. Tässä kohdassa esitellään useita kuvanlaadun mittareita, joita myöhemmin käytetään pakkausmenetelmien arvioimiseen.

Määritelmä 2.12. Neliöity kesquivirhe (mean squared error) voidaan laskea kahdelle harmaasävykuvalle kaavalla

$$MSE = \frac{\sum_{m,n} (I_1(m,n) - I_2(m,n))^2}{M \cdot N},$$

kun kuva on M pikseliä leveä ja N pikseliä korkea ja I_i on kuvan i intensiteetti pikselissä (m, n) [7].

Määritelmä 2.13. Maksimi signaali-kohina suhde (peak-signal-to-noise ratio) voidaan laskea kaavalla

$$PSNR = 10 \log_{10} \left(\frac{R^2}{MSE} \right),$$

kun R on maksimiero intensiteettien välillä [7].

MSE ja PSNR ovat perinteisiä kuvanlaadun mittareita, koska ne on helppo toteuttaa ja niillä on selvät fyysiset merkitykset [13].

Määritelmä 2.14. Rakenteellisen samankaltaisuuden indeksin (structural similarity index) määritelmä on

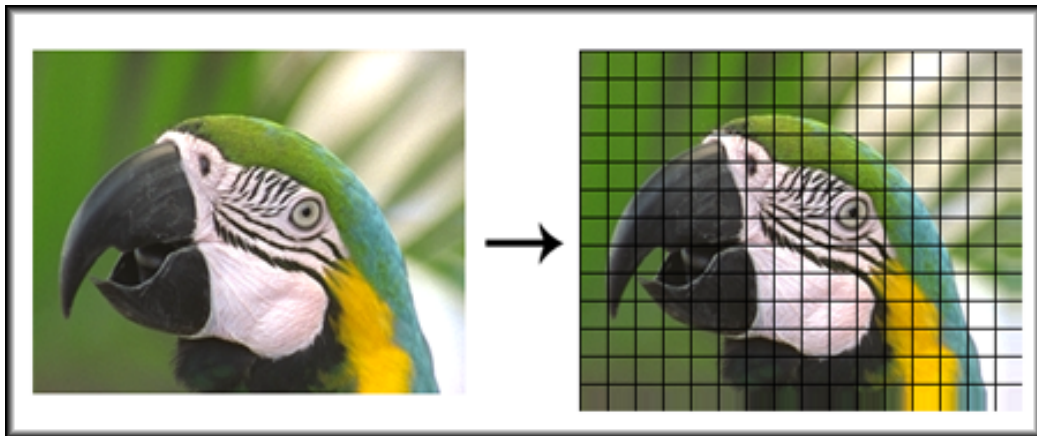
$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma,$$

missä $\alpha, \beta, \gamma > 0$ ovat vakioita, $l(x, y)$ on virhe luminanssissa, $c(x, y)$ on virhe kontrastissa ja $s(x, y)$ mittaa rakenteellisia virheitä.

Yksityiskohtainen määritelmä löytyy Wangin artikkelista [13]. Kuvalle voidaan määrittää mittari MSSIM (mean structural similarity index), mikä on yksinkertaisesti SSIM-arvon keskiarvo kuvassa. Tutkijoiden mukaan se vastaa ihmisen subjektiivista käsitystä kuvanlaadusta paremmin kuin muut yleiset mittarit, kuten MSE ja PSNR. Syyksi he arvioivat ihmisen visuaalisen systeemin taipuvaisuutta hahmottaa erilaisia rakenteita. [13]

2.5 Pakkausmenetelmät

2.5.1 Esiprosessointi

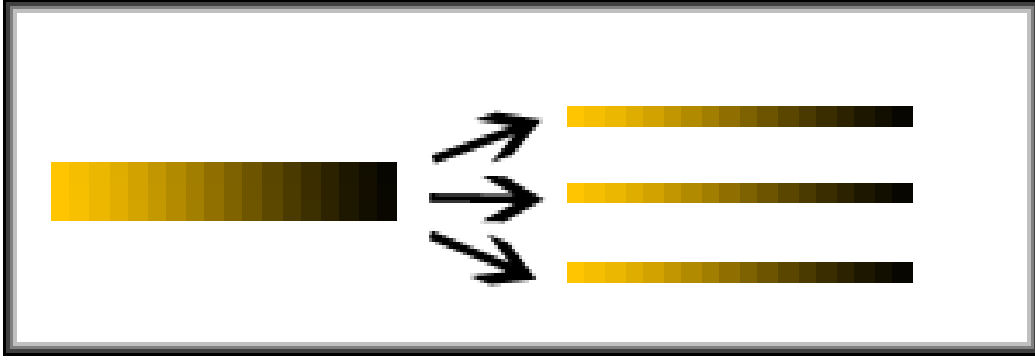


Kuva 1: Kuvan jakaminen ruutuihin. Alkuperäinen kuva lähteestä [12].

Ruutumenetelmä jakaa ensin kuvan 8×8 ruutuihin. Samanlainen esiprosessointivaihe on myös JPEG-menetelmässä [6]. Tämä muunnos on havainnollistettu kuvassa 1. Ruuduissa, jotka sijaitsevat kuvan reunalla, voi osa ruudun pikseleistä joutua kuvan ulkopuolelle. Tässä tapauksessa kuvan ulkopuoliset arvot korvataan lähimmän pikselin arvolla.

8×8 ruudusta jokaisen pikselin R , G ja B intensiteetti muodostaa yhden sarakkeen. Näiden sarakkeiden järjestys on mielivaltainen, eikä sillä ole pääkomponenttianalyysin lopputuloksen kannalta merkitystä. Näin 8×8 pikselin ruudusta muodostuu 192-elementtinen rivivektori. Vaihtoehtoisesti voidaan käyttää YCbCr-väriavaruutta, missä muunnos suoritetaan vastaavasti.

Pikselirivimenetelmä on Santon [11] kehittämä menetelmä yleistettynä värikuville. Menetelmässä kuvan pikselirivit muutetaan vektoreiksi. Tämä



Kuva 2: Kuvan muuntaminen pikseliriveiksi.

muunnos on havainnollistettu kuvassa 2. Vektoreille suoritetaan samankaltainen värimuunnos kuin ruutumenetelmässä. Datamatriisin sarakkeina ovat siis pikselien intensiteetit eri värien suhteen. Pakkausmenetelmä toteutetaan muuten vastaavasti kuin ruutumenetelmä, jotta eri lähestymistapojen hyötyjä voidaan verrata keskenään.

Määritelmä 2.15. Pakkaussuhde r on

$$r = \frac{\bar{s}}{s},$$

missä \bar{s} on pakattu koko ja s on pakkaamaton koko.

Jos säilytetään p % pääkomponenteista, matriisin pakkaussuhde r on

$$r = \frac{A + B}{U} = \frac{n \cdot (m \cdot p) + (m \cdot p) \cdot m}{n \cdot m}.$$

Pakkaussuhteen määrittelyssä:

1. A on matriisin datan määrä pääkomponenttiavaruudessa.
2. B on pääkomponenttien datan määrä.
3. U on pakkaamattoman datan määrä.

Erityisesti jos datamatriisi on muodostettu pikseliriveistä voidaan sanoa, että $n \approx m$. Näin ollen

$$(2.1) \quad r = \frac{n \cdot (m \cdot p) + (m \cdot p) \cdot m}{n \cdot m} \approx \frac{2n^2 p}{n^2} = 2p.$$

Oletetaan, että leveys ja korkeus ovat huomattavasti suurempia kuin 8 pikseliä. Jos kuva jaetaan 8×8 ruutuihin, voidaan sanoa, että $m \ll n$. Näin ollen $B \ll A$, jolloin

$$(2.2) \quad r \approx \frac{A}{U} = \frac{n \cdot (m \cdot p)}{n \cdot m} = p.$$

Tästä voidaan päätellä, että tilavaatimus ruutumethodissa on noin puolet pikseliriveihin perustuvasta versiosta, mikäli halutaan säilyttää sama murto-osa pääkomponentteja.

2.5.2 Datamatriisin pääkomponenttianalyysi

Datamatriisille suoritetaan esiprosessoinnin jälkeen pääkomponenttianalyysi. Algoritmi saa kaksi parametria m ja p , joiden perusteella määritetään valittujen pääkomponenttien määrä. Parametri m määrittää kuinka monta prosenttia datan variaatiosta täytyy pääkomponenttien kuvata. Jos tarvittaisiin enemmän kuin p % pääkomponenteista tämän variaation kuvaamiseen, valitaan p %. Muussa tapauksessa valitaan pienin mahdollinen määrä pääkomponentteja siten, että variaatioehto toteutuu. Pääkomponenttianalyysin laskemiseen käytettiin pääakselihajotelmaan perustuvaa algoritmia.

Datalle lasketaan tämän jälkeen uusi esitys pääkomponenttiavaruuden avulla. Pääkomponenttien joukko muodostaa ortonormaalin kannan, jolloin vektorien ortogonaalisuuden myötä koordinaatit voidaan määrittää laskemalla projektio kullekin kannan vektorille erikseen. Tämä projektio voidaan laskea määritelmää 2.1 käyttäen.

2.5.3 Liukulukujen pakkaaminen

Toisin kuin pikselien intensiteetit, pääkomponenttien elementit ja pääkomponenttiavaruuden koordinaatit voivat olla mielivaltaisia reaalilukuja, erityisesti myös negatiivisia. Käytännössä ne on siis tallennettu muistiin joko 32-bittisinä tai 64-bittisinä liukulukuina. Jos pikselin intensiteetti on esitetty 8-bittisenä lukuna, tämä heikentää huomattavasti pakkaussuhdetta.

Yksinkertainen keino tallentaa liukuluku x tiiviimpään kokonaislukumuotoon \tilde{x} ja muuntaminen takaisin approksimaatioksi $\bar{x} \approx x$ voidaan toteuttaa kaavoilla

$$\tilde{x} = \text{round}(x \cdot n)$$

ja

$$\bar{x} = \frac{\tilde{x}}{n},$$

missä n määrittää muunnoksen tarkkuuden. Muunnoksessa tapahtuva virhe on selvästi korkeintaan $\frac{1}{2n}$. Suuremmilla muuttujan n arvoilla muunnos on tarkempi, mutta luvun säilyttäminen vaatii enemmän tilaa. Tämä muunnos käytännössä vastaa luvun kiintopiste-esitystä [3]. Esimerkiksi kun $x = 62,36$ ja $n = 8$ muunnokset suoritetaan seuraavasti:

$$\begin{aligned} \tilde{x} &= \text{round}(62,36 \cdot 8) = \text{round}(498,88) = 499 \\ \bar{x} &= \frac{499}{8} = 62,375, \end{aligned}$$

jolloin absoluuttinen virhe on

$$s_{\bar{x}} = |62,375 - 62,36| = 0,015 < \frac{1}{2 \cdot 8} = 0,0625.$$

Minimi- ja maksimiarvot kyseisessä menetelmässä ovat

$$min = \frac{INT_MIN}{n}$$

ja

$$max = \frac{INT_MAX}{n},$$

missä INT_MIN ja INT_MAX ovat valitun kokonaislukutyyppin minimi- ja maksimiarvot. Näiden rajojen ulkopuolella olevia arvoja ei voida tarkasti esittää. Tutkielmassa valittiin tyyppiä 16-bittinen kokonaisluku eli short int.

2.5.4 Tallennus

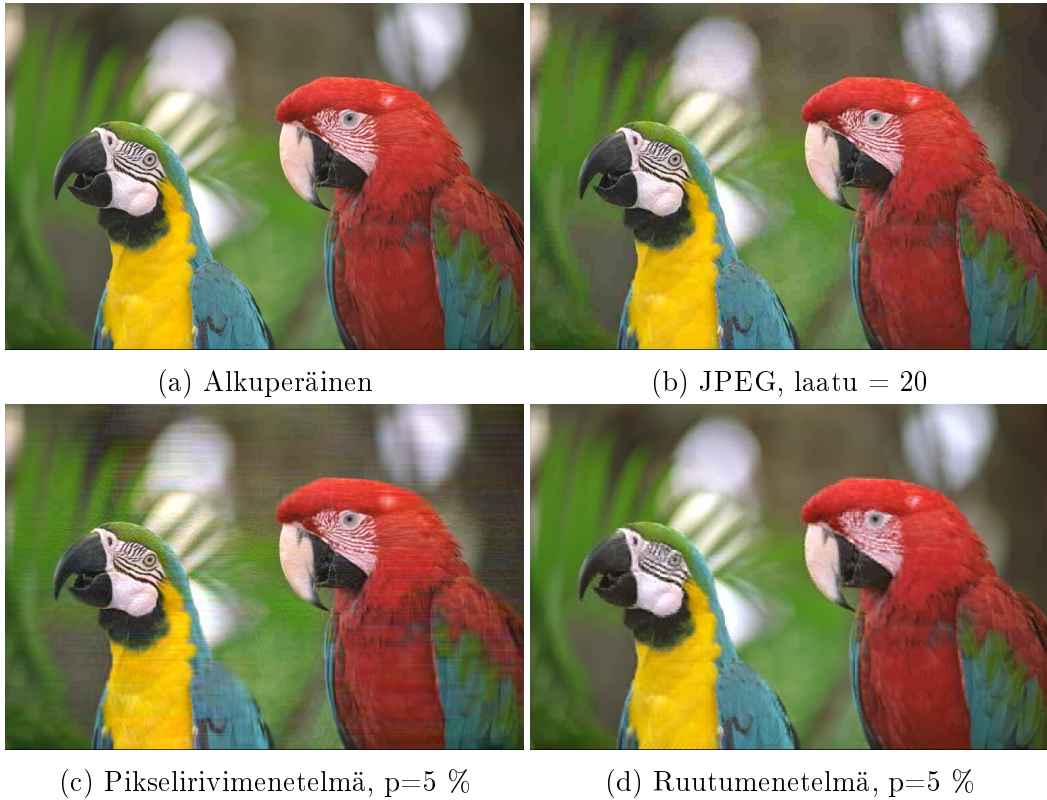
Tallentaessa data levyille tallennetaan seuraavat tiedot:

- kuvan leveys
- kuvan korkeus
- datamatriisin sarakkeiden keskiarvot
- pääkomponentit
- vektorien koordinaatit pääkomponenttiavaruudessa.

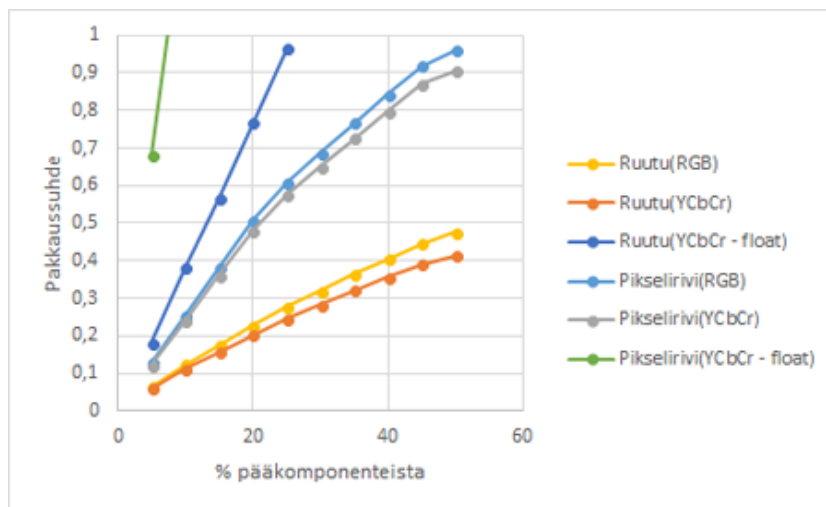
Tutkielmassa määritellyssä menetelmässä data pakataan tämän jälkeen DEFLATE-algoritmeilla ja kirjoitetaan tiedostoon. DEFLATE-algoritmi on yhdistelmä L77 algoritmista ja Huffmanin koodista ja se pakkaa datan häviöttömästi. [9]

2.6 PNG- ja JPEG-menetelmät

Edellä esitettyjen pakkausmenetelmien suorituskykyä verrataan suosittuihin PNG ja JPEG menetelmiin. JPEG on häviöllinen pakkausmenetelmä, joka perustuu diskreettiin kosinimuunnokseen [6]. Se muodostaa testeissä hyvän vertailukohdan, koska esitellyt menetelmät ovat häviöllisiä. PNG on häviötön pakkausmenetelmä, joka perustuu yksinkertaisiin suodatusmenetelmiin ja DEFLATE-algoritmiin [14]. PNG häviöttömänä menetelmänä määrittää häviöllisen menetelmän pakkaussuhteelle ylärajan. Syy on selvä, sillä häviöllisen menetelmän käyttäminen ei ole järkevää, jos menetelmällä pakkaussuhde on suurempi ja kuvanlaatu huonompi.



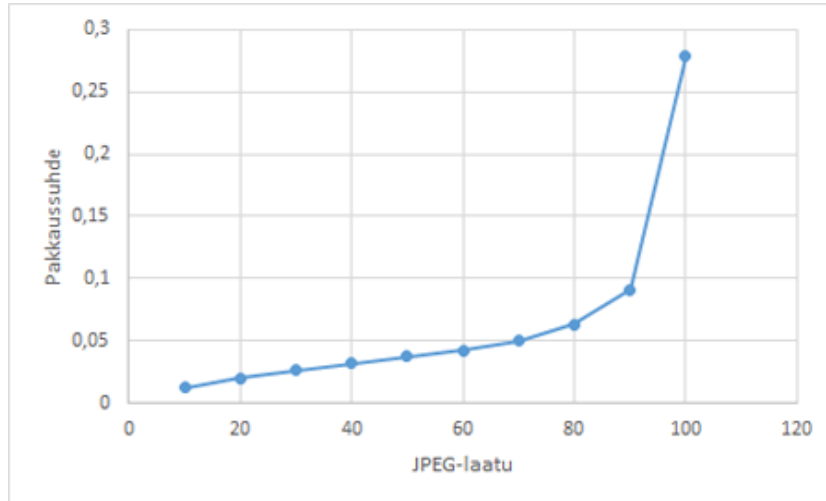
Kuva 3: Kuva pakattuna eri algoritmeilla. Alkuperäinen kuva lähteestä [12].



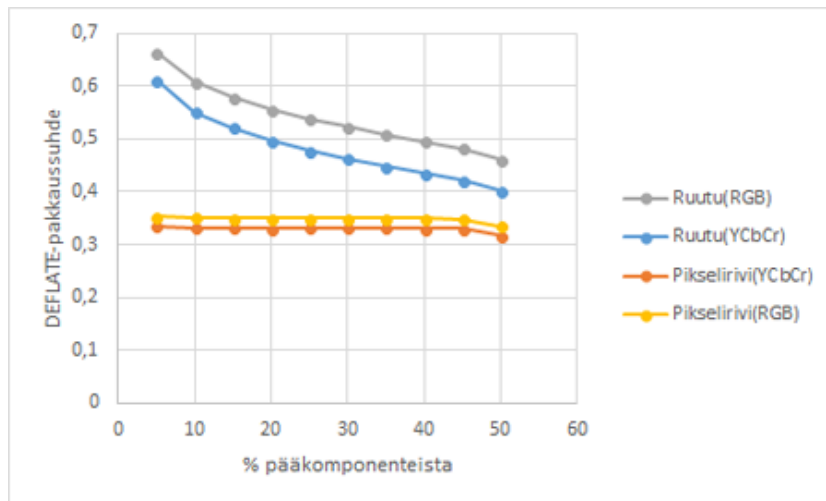
Kuva 4: Pääkomponenttimenetelmien pakkaussuhde säilytettyjen pääkomponenttien prosentin funktiona.

3 Tulokset

Eri menetelmien suorituskykyä mitattiin lähteestä [12] noudetuilla 24 kuvalla. JPEG-menetelmän suorituskykyä mitattiin laatuasetuksilla 10, 20, ...,



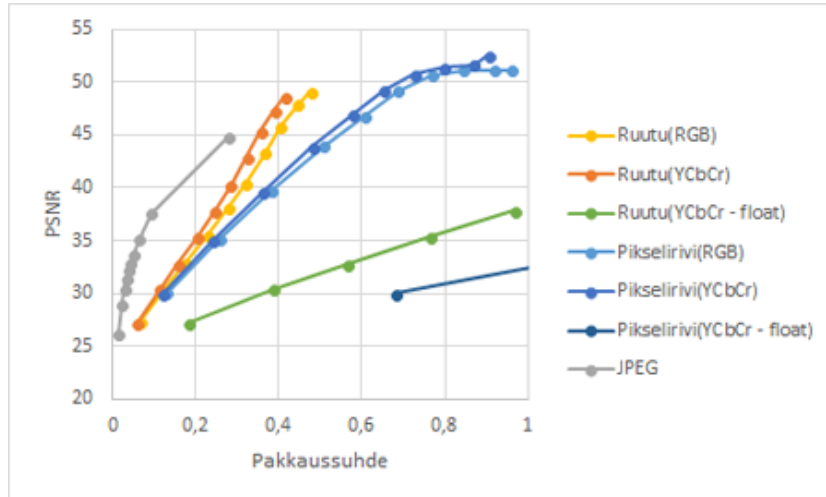
Kuva 5: JPEG-menetelmän pakkaussuhde JPEG-laadun funktiona.



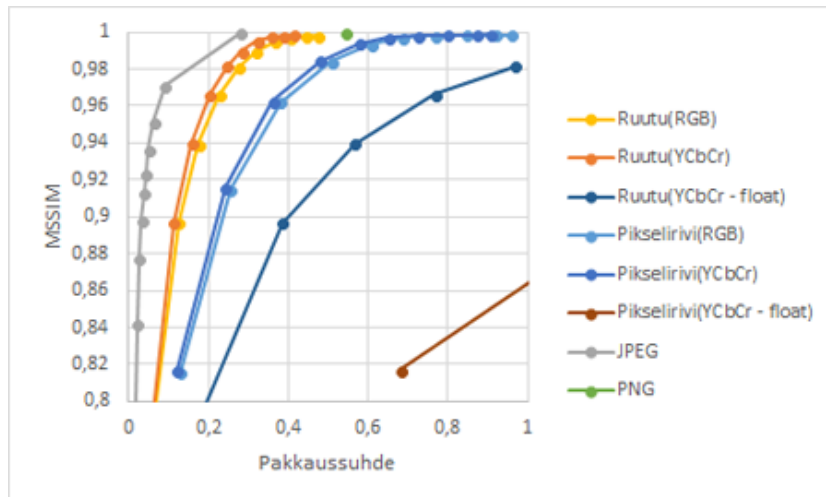
Kuva 6: DEFLATE-algoritmin pakkaussuhde säilytettyjen pääkomponenttien prosentin funktiona.

100. Pääkomponentteihin perustuvilla menetelmillä suorituskykyä mitattiin parametreilla $p = 5\%$, 10% , ..., 50% , $m = 100\%$, $n = 128$ pääkomponenteille ja $n = 1$ muuten. Lisäksi pääkomponenttimenetelmille käytettiin myös 32-bittiseen liukulukuun(float) perustuvaa tallennusmuotoa. Kaikille menetelmille laskettiin pakkaussuhde, MSSIM ja PSNR kaikilla parametrien arvoilla. Kuvassa 3 on esimerkki eri algoritmien toiminnasta.

Pääkomponenttimenetelmien pakkaussuhteet säilyttyjen pääkomponenttien prosentin funktiona on esitetty kuvassa 4. Pakkaussuhteiden väliset erot vastaavat likimäärin kaavoilla (2.1) ja (2.2) ennustettua $1 : 2$ suhdetta. Molemmilla menetelmillä YCbCr-väriavaruudella pakkaussuhde oli hieman parempi. Käyttäessä float-esitystä ruutumenetelmän pakkaussuhde suureni kes-



Kuva 7: Pakkausmenetelmien PSNR pakkaussuhteen funktiona.



Kuva 8: Pakkausmenetelmien MSSIM pakkaussuhteen funktiona.

kimäärin 3,56 kertaiseksi ja pikselirivimenetelmällä 5,62 kertaiseksi.

JPEG-menetelmän pakkaussuhde JPEG-laadun funktiona on esitetty kuvassa 5. JPEG-menetelmässä pienillä laatuarvoilla saavutettiin erinomaisia pakkaussuhteita. Laadulla 100 pakkaussuhde on likimain sama kuin ruutumenetelmässä arvolla $p = 25 \%$ ja pikselirivimenetelmässä $p = 10 \%$. PNG-menetelmän pakkaussuhde oli 0,544. Koska PNG-menetelmä on häviötön pakkausalgoritmi, pakkaussuhteen ollessa yli tämän arvon on algoritmi käytännössä hyödytön.

Kuvassa 6 on esitetty kuinka paljon DEFLATE-algoritmi pystyi pakkaamaan pääkomponenttimenetelmien tuottamaa dataa. DEFLATE pakkasi pikselirivimenetelmän dataa huomattavasti tehokkaammin. YCbCr-väriavaruus paransi hieman pakkaussuhdetta molemmissa menetelmissä. Kun menetel-

missä säilytetään enemmän pääkomponentteja huomataan, että pakkaussuhde paranee.

Kuvassa 7 on esitetty menetelmien PSNR pakkaussuhteen funktiona. PNG-menetelmälle PSNR on ∞ , sillä häviöttömässä pakkauksessa $MSE = 0$. Näin ollen sitä ei voida piirtää kaavioon. JPEG-menetelmä on tällä mittarilla paras pienillä pakkaussuhteilla. Jos vaaditaan suurempia PSNR-arvoja, ruutumenetelmä ja PNG pärjäävät paremmin. Pikselirivimenetelmä oli menetelmistä huonoin. YCbCr-väriavaruus suoriutui RGB-väriavaruutta paremmin molemmilla pääkomponenttimenetelmillä.

Kuvassa 8 on esitetty menetelmien MSSIM pakkaussuhteen funktiona. JPEG-menetelmä on tällä mittarilla selvästi paras häviöllisistä menetelmistä. Ruutumenetelmä oli häviöllisistä menetelmistä toiseksi paras ja pikselirivimenetelmä oli tälläkin mittarilla huonoin. YCbCr-väriavaruus suoriutui RGB-väriavaruutta paremmin tälläkin mittarilla.

4 Pohdinta

Pääkomponenttimenetelmät suoriutuivat kokonaisuudessaan JPEG-menetelmää huonommin. Erityisesti matalilla pakkaussuhteilla JPEG-menetelmä oli selkeästi parempi varsinkin MSSIM-mittarilla. Ruutumenetelmä oli pääkomponenttimenetelmistä parempi ja menetelmä oli melko lähellä JPEG-menetelmää PSNR-mittarilla. Selkeästi suurempien vektorien käyttö pikselirivimenetelmässä ei merkittävästi parantanut kuvanlaatua, ja suurten pääkomponenttien vaatima ylimääräinen tila heikensi pakkaussuhdetta.

Liukulukujen tallennus kiintopiste-esityksen avulla toimi erinomaisesti molemmilla menetelmillä. Ero tallennustilassa oli huomattavasti suurempi kuin kaksinkertainen, mistä voidaan päätellä että DEFLATE-algoritmi toimii tehokkaammin kiintopistelukuja tiivistäessä. Molemmilla mittareilla ero kuvanlaadussa samalla pääkomponenttimäärällä oli äärimmäisen pieni. Tästä voidaan päätellä, että koordinaattien tiivistämisestä koituvat virheet eivät vaikuttaneet merkittävästi kuvanlaatuun. Tämä johtunee siitä, että intensiteetit kuvissa on esitetty kokonaislukujen avulla, jolloin pienellä virheellä ei ole merkitystä intensiteetin pyöristyessä samaan kokonaislukuun.

DEFLATE-algoritmi pystyi selvästi pienentämään pakattujen kuvien koosta. Pikselirivimenetelmässä DEFLATE toimi tehokkaammin ja algoritmi pakkasi tehokkaammin kuvien dataa YCbCr-väriavaruudessa. Lisäksi DEFLATE-algoritmi toimii tehokkaammin, kun säilytetään enemmän pääkomponentteja. Kaikissa näissä tilanteissa voidaan arvella, että datassa on enemmän toistuvia arvoja, mikä mahdollistaa datan esittämisen tiiviimmässä muodossa.

5 Yhteenveto

Pääkomponenttianalyysiä on tässä tutkielmassa onnistuneesti sovellettu kuvanpakkaamiseen. Erityisesti se on hyödyllinen häviöllisessä pakkauksessa, jossa vähemmän tärkeitä pääkomponentteja ei tarvitse säilyttää. Häviötön DEFLATE-algoritmi paransi huomattavasti menetelmien pakkaussuhdetta. YCbCr-väriavaruuden käyttö paransi kuvien pakkaussuhdetta verrattuna RGB-väriavaruuteen. Kiintopiste-esityksen käyttäminen tallennusmuotona osoitautui hyväksi menetelmäksi, sillä molemmilla pääkomponenttimenetelmillä pakkaussuhde parani moninkertaisesti käyttäessä kiintopiste-esitystä.

Verrattuna JPEG-menetelmään toteutetuilla menetelmillä kuvanlaatu oli heikompi, erityisesti pienemmillä pakkaussuhteilla. Näin ollen niitä ei voi suositella kuvanpakkaukseen. On kuitenkin mahdollista, että pääkomponenttianalyysi voisi toimia osana jotain muuta kuvanpakkausalgoritmia.

Viitteet

- [1] Gilbert Strang. 1980. *Linear Algebra and Its Applications*, Second edition. Academic Press.
- [2] Haïfa Nakouri. 2016. An incremental two-dimensional principal component analysis for image compression and recognition. *In: Proc. of the 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, 725 - 731
- [3] Hayden So. 2006. *Introduction to Fixed Point Number Representation*. <http://www-inst.eecs.berkeley.edu/~cs61c/sp06/handout/fixdpt.html>. Checked 11.5.2018.
- [4] Ian Jolliffe. 2002. *Principal Component Analysis*, Second Edition. Springer.
- [5] International Telecommunication Union. 2011. *Information technology – Digital compression and coding of continuous-tone still images: JPEG File Interchange Format (JFIF)* <http://www.itu.int/rec/T-REC-T.871-201105-I/en>. Checked 24.3.2018.
- [6] International Telecommunication Union. 1993. *ISO/IEC 10918-1*
- [7] Mathworks. Compute peak signal-to-noise ratio (PSNR) between images. <https://se.mathworks.com/help/vision/ref/psnr.htm>. Checked 25.3.2018.
- [8] Oxford Dictionaries. <https://en.oxforddictionaries.com/definition/chrominance>. Checked 24.3.2018.
- [9] Peter Deutsch. 1996. *DEFLATE Compressed Data Format Specification version 1.3*. <https://tools.ietf.org/html/rfc1951>. Checked 26.3.2018.
- [10] Qian Du and James E. Fowler. 2007. Hyperspectral image compression using JPEG2000 and principal component analysis for image compression and recognition. *IEEE Geoscience and Remote Sensing Letters* 10, 2, 201-205.
- [11] Rafael do Espírito Santo. 2012. Principal component analysis applied to digital image compression. *Einstein (São Paulo, Brazil)* 10, 2, 135-139.
- [12] True Color Kodak Images. <http://r0k.us/graphics/kodak/>. Checked 26.4.2018.

- [13] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4, 600-612.
- [14] W3C. 2003. *Portable Network Graphics (PNG) Specification (Second Edition)*. <https://www.w3.org/TR/PNG/>. Checked 29.4.2018.

Agenttipohjainen mallinnus filosofian tutkimusmenetelmänä

John Mäkelä

Tiivistelmä.

Agenttipohjainen mallinnus on menetelmä, jossa monimutkaisten järjestelmien toimintaa kuvataan simuloimalla yksittäisten osien, agenttien, vuorovaikutusta keskenään ja ympäristön kanssa. Agenttipohjaista mallinnusta on käytetty esimerkiksi taloustieteissä, yhteiskuntatutkimuksessa ja biologiassa, mutta käytännön sovelluksia akateemisen filosofian tutkimuksessa on ollut verrattain vähän. Tämän tutkielman tavoite on esitellä agenttipohjaisen mallinnuksen peruskäsitteitä ja tutkia agenttipohjaisen mallintamisen soveltuvuutta erityisesti filosofian tutkimukseen.

Avainsanat ja -sanonnat: Agenttipohjainen mallinnus, agentti, ajatuskoe, metodologia, simulaatiomalli, tieteenfilosofia, VirtuousEmotivist.

1. Johdanto

Monet reaali maailman järjestelmät koostuvat suuresta joukosta vuorovaikuttavia osia, joiden sisäinen toiminta eroaa toisistaan. Tällaisten *kompleksisten systeemien* tutkiminen kokeellisesti on usein kallista tai mahdotonta. Esimerkiksi monet sosiaaliset, biologiset tai taloudelliset järjestelmät ovat tässä mielessä kompleksisia. Mallinnus on tapa tutkia reaali maailman ongelmia silloin kun ei ole mahdollista järjestää aitoa koetta.

Erilaiset mallit voi jakaa analyyttisiin malleihin ja simulaatioihin. Analyytisessä mallissa lopputulos riippuu kokonaan annetuista parametreista. Eri muuttujien riippuvuus voidaan esittää esim. differentiaaliyhtälöryhmänä. Yksi analyyttisten mallien ongelma on, että ratkaisua ei välttämättä ole olemassa kaikissa tilanteissa tai se on vaikea löytää. [Borshchev and Filippov 2004.]

Simulaatiomalleissa yhtälöryhmän sijasta määritellään joukko sääntöjä, jotka kertovat miten systeemin tila seuraavalla ajanhetkellä seuraa sen tilasta nykyisellä ajanhetkellä. Simulaatiomallin ajaminen tai suorittaminen tarkoittaa sitä, että lähdetään alkutilasta, lasketaan välitilat ja päädytään lopputilaan. [Borshchev and Filippov 2004.] Eräs tieteenfilosofinen tulkinta tästä prosessista on, että jos määritellyn mallin suoritus tuottaa empiirisesti oikealta vaikuttavan lopputilan, tämä muodostaa yksinään riittävän *selityksen* (explanans) tutkittavalle ilmiölle [Epstein 2006]. Kriitikot ovat kuitenkin esittäneet, että mallin ajaminen osoittaa pelkästään sen, että määrittelyistä seuraa kyseinen lopputila, ei sitä, *miten* määritellyt säännöt aiheuttavat lopputilan [León-Medina 2017; Di Paolo *et al.* 2000].

Agenttipohjaiset mallit ovat eräs simulaatiomallien alalaji. Tämän tutkielman ensimmäinen tavoite on esitellä agenttipohjaisen mallinnuksen peruskäsitteet (luku 2) ja vertailla mallintamiseen käytettyjä ohjelmistokehyksiä keskenään, osittain filosofian tutkimukseen soveltuvuuden näkökulmasta (luvut 3, 4 ja 6). Tutkielman toinen tavoite on määritellä ja toteuttaa esimerkkimalli, joka auttaisi yhteiskuntafilosofisen teorian analysoinnissa, ymmärtämisessä ja havainnollistamisessa (luku 5).

2. Agenttipohjainen mallinnus

2.1. Määritelmä

Agenttipohjainen mallinnus (lyh. ABM) on simulaatiomallintamisen alalaji, jossa *selitettävä ilmiö* (explanandum) mallinnetaan agenttien välisenä vuorovaikutuksena sekä agenttien ja ympäristön välisenä vuorovaikutuksena. Agentti on autonominen yksilö tai muu olio, jolla on tiettyjä attribuutteja, toimintoja ja mahdollisesti myös päämääriä. Agentit toimivat ympäristössä, joka voi olla esim. geometrinen tai perustua empiirisiin havaintoihin.

Agenttipohjaisessa mallintamisessa agentin käyttäytyminen määritellään säännöillä, ja tarkkaillaan agenttien vuorovaikutuksen lopputulosta. Agenttien käyttäytymissääntöjä voi aluksi luonnostella sanallisesti, jonka jälkeen ne voidaan kirjoittaa ohjelmakoodiksi, usein samalla hyödyntäen ohjelmistokehyksiä, jotka tarjoavat ohjelmalle valmiin ja yleisemmin testatun rungon. Agenttien käyttäytyminen, strategiat ja sisäinen tila voivat muuttua ajassa tai vuorovaikutuksen seurauksena. [Wilensky and Rand 2015.]

2.2. Ohjelmistokehykset

Ohjelmistokehys muodostaa ohjelman rungon, ja siinä on tarkoituksenmukaisesti aukkoja ennalta odotettuja täydennyksiä varten [Koskimies ja Mikkonen 2005]. Agenttipohjaisen mallintamisen ohjelmistokehyksissä voi olla valmiina esimerkiksi komponentteja agenttien ympäristön luomista ja agenttien toiminnan ajoittamista varten. [Masad and Kazil 2015.]

Kravari ja Bassiliades [2015] vertailivat 24 agenttipohjaisen mallintamisen ohjelmistokehystä viidellä eri kriteerillä. Kriteerien löytämiseksi kirjoittajat tutustuivat olemassa olevaan kirjallisuuteen löytääkseen yleisimmin käytetyt tai standardeihin perustuvat kriteerit. Tutkielmani kannalta näistä oleellimmat kriteerit ovat *käytettävyys* (usability), *toimintakyky* (operating ability) ja *pragmaattisuus* (pragmatics). Kravari ja Bassiliades jakavat nämä kriteerit edelleen alakategorioihin. Esimerkiksi toimintakyky jakautuu viiteen osa-alueeseen: *suorituskykyyn* (performance), *vakauteen* (stability), *kestävyyteen* (robustness), *ohjelmointikieleen* ja *käyttäjärjestelmään*.

Taloudellisten, sosiaalisten ja biologisten järjestelmien suuren agenttimäärän vuoksi erityisesti suorituskyky ja skaalautuvuus ovat näille olennaisia kriteerejä. Esimerkiksi EU:n talousaluetta kuvaava Eurace@Unibi-malli käyttää FLAME-ohjelmistokehystä, koska se pystyy skaalautumaan rinnakkaislaskennan avulla suuriin agenttimääriin. [Dawid *et al.* 2012.] Varsinaisesta yhteiskuntatutkimuksesta ja biologiasta poikkeavissa sovellusalueissa suorituskyvyllä voi olla vähemmän merkitystä, mutta näillä alueilla agenttipohjaisten mallien käyttöä oli vaikeampi tutkia, koska julkaisujen määrä on pieni. Eri sovellusalueilla korostuvat myös eri arviointikriteerit. Esimerkiksi sotilaallisissa sovelluksissa turvallisuus voi olla tärkeää, kun taas joissakin akateemisissa sovelluksissa se voi olla lähes merkityksetöntä.

Viiden kriteerin avulla tehdyn vertailun sisällä Kravari ja Bassiliades [2015] luokittelivat ohjelmistokehystä esimerkiksi sovellusalueen ja FIPA-yhteensopivuuden mukaan. FIPA-järjestön (IEEE Foundation for Intelligent Physical Agents) yhteensopivuusstandardit tukevat ABM-ohjelmistokehysten keskinäistä yhteensopivuutta ja tiedonvaihtoa. Vain pieni osa (4/24) noudatti arviointihetkellä täysin FIPA-spesifikaatioita. Eniten käytetyistä ohjelmistokehystistä JADE, Jadex ja Jack ovat täysin FIPA-yhteensopivia, Jason on osittain yhteensopiva, ja NetLogo ei ole ollenkaan FIPA-yhteensopiva.

Ohjelmistokehukset näyttävät soveltuvan melko tasaisesti eri sovellusalueisiin, ja useimmat soveltuvat moneen tai jopa enemmistöön eri sovellusalueista. Ohjelmointikielistä Java oli kaikkein suosituin, ja monet ohjelmistokehystistä tukevat useampaa kuin yhtä ohjelmointikieltä. [Kravari and Bassiliades 2015.]

2.3. Ohjelmistokehysten yksityiskohtainen vertailu

Tämän tutkielman puitteissa ei ole mahdollista vertailla kaikkia saatavilla olevia ohjelmistokehystä. Niinpä keskityn myöhemmin luvussa 4 vertailemaan vain Kravarin ja Bassiliadeksen [2015] vertailun mukaan suosituimpia ohjelmistokehystä.

3. Agenttipohjainen mallinnus ja akateeminen filosofia

3.1. Aikaisemmat toteutukset

Agenttipohjaisesta mallinnuksesta on käytännössä sovellettu melko vähän akateemisessa filosofiassa. Sen sijaan kirjallisuushakuni perusteella suurin osa keskustelusta keskittyy ABM:n ja yleisemminkin simulaatioiden tieteenfilosofiin merkityksiin ja ongelmiin (esim. Di Paolo ja muut [2000], Epstein [2011] ja León-Medina [2017]). Kuitenkin esim. Vallier [2017] esittelee agenttipohjaisen mallin,

joka kuvaa poliittista vakautta yhteiskunnassa, tavoitteenaan erityisesti verrata sitä rawlsilaiseen vakauden käsitykseen. Jopa uskontotutkimuksessa on harjoitettu agenttipohjaista mallintamista buddhalaisuuden ja altruismin tutkimuksessa [Yang *et al.* 2015].

On väitetty, että simulaatiomallit tuovat uuden työkalun filosofian tutkimukseen, jota voidaan käyttää kiistanalaisten ajatuskokeiden parempaan havainnollistamiseen. Tässä näkemyksessä simulaatiomalli on eräänlainen apuväline, jota voidaan käyttää silloin kun ihmisen luontainen päättelykyky ei riitä ajatuskokeiden viemiseen loppuun asti. [Dennett 1994.] Tämä näkemys on saanut kritiikkiä osin sen takia, että simulaatiot eroavat ajatuskokeista tietyillä olennaisilla tavoilla, joita käsitellään myöhemmin [Di Paolo *et al.* 2000].

3.2. Tieteenfilosofiset kannat ja simulaatiomalli ajatuskokeena

Tieteenfilosofiset kannat koskien simulaatiomalleja voidaan jakaa karkeasti kahteen ääripäähän, jotka eroavat siinä, mitä tieteellistä arvoa mallilla ajatellaan olevan. Ensimmäinen ääripää on ”vahva” kanta, jonka mukaan simulaatiomalli ei ole pelkästään malli, vaan *ilmentymä* tutkittavasta reaalimaailman ilmiöstä. Tämän tulkinnan mukaan simulaatioista saaduilla havainnoilla on täysin sama arvo kuin reaalimaailmassa tehdyillä havainnoilla. Toisen ääripään mukaan simulaatioilla ei ole mitään tieteellistä arvoa, sillä ne ovat vain tietokoneohjelmia, joiden toiminta riippuu täysin premisseistä tai syötteestä. Tämän näkemyksen mukaan, jos premissit tai syöte ovat jo valmiiksi tiedossa, niin simulaation synnyttämät loogiset johtopäätökset eivät voi muodostaa uutta *havaintoa*. [Di Paolo *et al.* 2000.]

Tavoitteeni tässä tutkielmassa ei ole esittää kantaa simulaatiomallien tieteellisestä arvosta, suuntaan tai toiseen. Merkityksellistä oman tutkielmani kannalta on se, että riippumatta siitä minkä tulkinnan omaksuu, simulaatio voi olla heikoimmillaankin vähintään harjoitus, jonka tarkoitus on havainnollistaa teoriaa ja auttaa tutkimuksen ensimmäisessä, kartoittavassa vaiheessa. Tässä mielessä simulaatiomallit ovat *läpinäkymättömiä ajatuskokeita*, joissa filosofisen ajatuskokeen *selityksellinen läpinäkyvyys* (explanatory opacity) ja lopputuloksen itsestänselvyys jää pois, mutta toisaalta mahdollistetaan monimutkaisempi laskenta tai päättely kuin mitä yksittäisen ihmisen olisi mahdollista harjoittaa [Di Paolo *et al.* 2000]. Nimitän tällaista asennetta tietokonesimulaatioiden kehittämiseen myöhemmin *ajatuskoeomaiseksi ymmärtämiseksi*.

4. Ohjelmistokehysten soveltuvuus filosofisiin ajatuskokeisiin

4.1. Olennaiset kriteerit

Jos tavoitteena on agenttipohjaisen mallintamisen käyttö metodina filosofian tutkimuksessa, suuren agenttimäärän ja suorituskyvyn sijasta tärkeämpää voisi olla esimerkiksi yksinkertaisuus ja nopea opittavuus, koska nämä kaksi mahdollistaisivat ajatuskoemaisen ja kartoittavan etenemistavan. Mitä enemmän simulaatiosta muodostuu ajattelun *jatke*, joka edesauttaa ymmärtämistä, sitä enemmän se alkaa muistuttaa oikeaa ajatuskoetta. Vastaavasti mitä enemmän ohjelmointikielen syntaksi ja ohjelmistokehysten monimutkaisuus luovat muuria käyttäjän ja tietokoneen välille, sitä läpinäkymättömämpi simulaatiosta muodostuu.

Kravari ja Bassiliades [2015] luokittelevat eri ohjelmistokehysten käytettävyyden kahteen tutkielmani kannalta merkitykselliseen kategoriaan: *yksinkertaisuuteen* (simplicity) ja *opittavuuteen* (learnability). Molemmat ovat tärkeitä ajatuskoemaisen ymmärtämisen kannalta. Ne myös edesauttavat nopeaa prototyyppointia kartoittamisvaiheessa. Tämä kartoittaminen on luonteeltaan nopeaa uusien ideoiden testaamista, jossa on tavoiteltavaa, että epäonnistuneet ideat kuluttavat mahdollisimman vähän turhaa aikaa.

Yksinkertaisuuden ja opittavuuden lisäksi kolmas valitsemani kriteeri on ohjelmointikieli. Tämä kriteeri on mukana siksi, että mallin ajatuskoemaista ymmärtämistä edesauttaa, jos käytössä oleva ohjelmointikieli on jo entuudestaan tuttu ja syntaksiltaan helppo. Ohjelmointikieli on kriteerinä kuitenkin subjektiivinen, ja lopullinen valinta riippuu siis tällöin siitä, mikä edistää mallin kehittäjässä parhaiten ajatuskoemaista ymmärtämistä.

4.2. Vertailu

Valikoin ohjelmistokehyksistä mukaan vertailuun suosion ja kehitysaktiivisuuden perusteella NetLogon, Jasonin, ja Jadexin. Lisäksi sisällytän vertailuun myös Mesa-ohjelmistokehysten, vaikka se ei ehtinytkään todennäköisesti uutuudestaan johtuen Kravarin ja Bassiliadeksen [2015] vertailuun. Syy Mesa-projektin sisällyttämiseen on, että se on kehitykseltään aktiivisin Python 3-kielellä toteutettu ohjelmistokehys, ja tukee valmiiksi www-selainpohjaista visualisointia.

4.2.1. NetLogo

NetLogo on Lisp-sukuinen ohjelmointikieli, ohjelmistokehys ja kehitysympäristö. NetLogon kehitysympäristö on vertailussa kaikkein monipuolisin, ja sitä käyttääkseen ei tarvitse olla ammattiohjelmoija. NetLogo onkin suunniteltu Logo-kielen hengessä siten, että se soveltuisi mahdollisimman hyvin myös ope-

tuskäyttöön. NetLogo sisältää tuen sekä 2D- että 3D-visualisoinneille ja sisäänrakennettuja mekanismeja rinnakkaisuuden toteuttamiseksi. [Tisue and Wilensky 2004.] NetLogo-kehitysympäristön mukana tulee paljon esimerkkimalleja, ja lisäksi siitä on tehty oppikirja [Wilensky and Rand 2015]. Löysin kirjallisuushaun aikana eniten NetLogoa käsitteleviä julkaisuja.

4.2.2. Jason

Jason on AgentSpeak-ohjelmointikielen toteutus. AgentSpeak eroaa merkittävästi useimmista ohjelmointikielistä, sillä siinä ei ole lainkaan muuttujakäsitettä, vaan agenteilla on alussa *uskomuksia* (initial beliefs) ja *suunnitelmia* (plans). [Bordini *et al.* 2007.]

```
cogitoErgoSum.  
+cogitoErgoSum <- .print("Hello World!").
```

Koodikatkelma 1. AgentSpeak-ohjelmointikielen syntaksi [Bordini *et al.* 2007].

Esimerkissä agentilla on aluksi uskomus ("uskon että *cogito ergo sum*"), ja suunnitelma (aina kun päädyt uskomaan, että *cogito ergo sum*, tulosta "Hello World"). AgentSpeakin paradigma on luonteeltaan agenttiorientoitunutta tai *agenttiohjelmointia* (agent-oriented programming), yleisemmän *olio-ohjelmoinnin* (object-oriented programming) sijaan. Bordini ja muut [2007] kuvailevat samassa yhteydessä *BDI-mallia* (beliefs, desires, intentions), joka on agenttiorientoitunut tapa käsitteellistää tietokoneohjelmien toimintaa ikään kuin niillä olisi sisäinen mentaalinen tila.

4.2.3. Jadex

Jadex on BDI-malliin perustuva ohjelmistokehys, joka yhdistää Java-ohjelmointikielen ja JSON- tai XML-kuvauskielet. Jadex-kehyksessä agenttien suunnitelmat ohjelmoidaan Java-kielellä, ja uskomukset määritellään kuvauskielellä. Uskomukset voivat olla mitä tahansa Java-objekteja. [Braubach *et al.* 2005.]

```
<belief name="system_time"  
class="long" updatarate="1000">  
<fact>System.currentTimeMillis()</fact>  
</belief>
```

Koodikatkelma 1. Jadexin XML-syntaksi [Braubach *et al.* 2005].

4.2.4. Mesa

Mesa on neljästä vertailuun valitusta ohjelmistokehyksestä nuorin. Mesa-ohjelmistokehyksestä on kirjallisuushaun perusteella tehty vähemmän julkaisu- ja kuin muista vertailun ohjelmistokehyksistä. Mesan erityispiirteitä ovat Python 3-kieli ja Jupyter Notebook-kehitysympäristö, joka soveltuu erityisesti mallin tulosten analysointiin. Mesa oli myös vertailussa ainoa ohjelmistokehys, jossa on valmis tuki www-selainpohjaiselle visualisoinnille. [Masad and Kazil 2015.]

4.3. Vertailun tulokset

Kokosin vertailun tulokset yksinkertaiseen taulukkoon, jossa relevantit kriteerit on arvioitu asteikolla Matala-Keskinkertainen-Korkea.

	NetLogo	Jason	Jadex	Mesa
Yksinkertaisuus	Keskink.	Keskink.	Matala	Korkea
Opittavuus	Korkea	Keskink.	Matala	Keskink.
Ohjelmointikieli	NetLogo/Lisp	AgentSpeak	Java, JSON	Python

Taulukko 1. Vertailun tulokset

NetLogo sai keskinkertaisen yksinkertaisuusluokituksen, koska sen käyttäminen edellyttää monien NetLogo-erityisten käsitteiden hahmottamista. Se saa kuitenkin opittavuudessa korkeimman luokituksen, koska siitä on julkaistu kirja ja kehitysympäristön mukana tulee vertailun laajin esimerkkikokoelma.

Jason sai keskinkertaisen luokituksen yksinkertaisuudessa, sillä sen syntaksi pyrkii olemaan intuitiivinen BDI-mallin toteutus, mutta käyttää vaikeaselkoisia symboleita varattujen sanojen sijaan. Toisaalta voidaan nähdä, että Jasonin AgentSpeak-syntaksi voisi olla vasta-alkajalle jopa yksinkertaisempaa kuin perinteiset ohjelmointikielet. Opittavuus on keskinkertaista, sillä siitä on julkaistu kirja, mutta agenttiorientoitunut paradigma eroaa radikaalisti muunlaisesta ohjelmoinnista, ja niinpä on epätodennäköistä, että vasta-alkajalla olisi valmiiksi tietämystä siitä tai muita resursseja kuin kyseinen kirja.

Jadex sai huonoimman luokituksen yksinkertaisuudessa ja opittavuudessa, koska siitä ei ole julkaistu kirjaa, ja mallin toteutus on jaettu ohjelmointi- ja kuvauskielten välille. Tämä tarkoittaa myös, että Jadexin käyttämiseksi on osattava kahta kieltä, kun muut vertailun ohjelmistokehykset edellyttävät vain yhden oppimista.

Mesa sai korkean luokituksen yksinkertaisuudessa, sillä IPython-kehitysympäristö on vertailun yksinkertaisin ja helppokäyttöisin, mutta opitta-

vuus kärsii, koska siitä ei ole julkaistu kirjaa. Mesa on muihin vertailun ohjelmistokehyksiin verrattuna pienikokoinen, joten sen toimintaa on mahdollista ymmärtää myös lukemalla lähdekoodia suoraan. Mukana tulevien esimerkkien määrä on kohtalainen, mutta NetLogon mukana niitä tulee enemmän.

5. VirtuousEmotivist-malli

5.1. Johdanto

Tässä luvussa esitelly agenttipohjainen malli on löyhästi Alasdair MacIntyren modernia moraalifilosofiaa kohtaan esittämän kritiikin inspiroima. MacIntyre esittää kirjassaan *After Virtue* [1981, suom. Hyveiden jäljillä 2004], että modernissa etiikan diskurssissa on mahdotonta ratkaista erimielisyyksiä, sillä erilaiset kilpailevat argumentit ovat samaan aikaan näennäisesti objektiivisia ja toisaalta yhteismitattomia, koska ne nojaavat toistensa kanssa epäyhteensopiviin lähtökohtiin, joiden väliltä ei voi valita millään yhteiskunnassa vakiintuneella tavalla. MacIntyre kutsuu tästä asiaintilasta seuraavaa asennetta *emotivismiksi*. Emotivismin puutteiden korjaamiseksi MacIntyre muodostaa oman aristoteelisen hyve-etiikan näkemyksensä, jossa keskeistä on *tradition* harjoittaminen. Määrittelemäni malli pyrkii tyylytelten jäljittelemään joitain MacIntyren teorian aspekteja, mutta toimii lähinnä yleisenä esimerkkinä tai harjoituksena eräästä mahdollisesta agenttipohjaisen mallin toteutuksesta. Mallin tulokset seuraavat täysin suunnitteluprosessin aikana tehdyistä oletuksista, joiden ei voida katsoa tämän tutkielman rajoitteet huomioon ottaen olevan täsmällinen tai edes eräs tulkinta MacIntyren näkemyksestä.

5.2. Mallin määrittely

VirtuousEmotivist-malli koostuu kolmesta agenttiluokasta ja visualisointi- sekä malliluokista. Mallin (*VirtuousEmotivistModel*) ympäristö on 25x25 solun ruudukko, jossa jokainen solu voi olla joko tyhjä, sisältää *hyveellisen agentin* (*VirtuousAgent*) tai *emotivistisen agentin* (*EmotivistAgent*). Nämä kaksi agenttiluokkaa periytyvät "*uskova agentti*"-luokasta (*BelievingAgent*) joka sisältää joitain yleisiä ominaisuuksia kuten uniikin id:n, uskomustaulukon ja päättävyyden sekä vallan määrän. Ruudukossa läsnä olevat agentit liikkuvat käyttäytymissääntöjen mukaan joko satunnaisesti valittuun tyhjäan ruutuun tai jäävät paikoilleen, jos niiden Moore-naapurustossa (yhden ruudun päässä) on tarpeeksi samankaltaisia agentteja. Paikoilleen jäädessään agentin sanotaan olevan "onnellinen". Mallia ajetaan askeleittain siten, että jokaisella askeleella kaikki agentit aktivoituvat satunnaisessa järjestyksessä yksi kerrallaan, ja suorittavat askel-metodin, joka toteuttaa kaiken agentin käyttäytymisen. Mallille ei ole määritetty tarkkaa lopputilaa, joten tuloksia analysoidessa täytyy valita mieli-

valtaisesti jokin askelten lukumäärä, johon simulaatio päätetään. Eräs loppupisteen määritelmä voisi olla myös se, kun tarpeeksi suuri määrä agenteja on saavuttanut sekä onnellisuuden että vakuuttumisen tilan, ja pysynyt siinä riittävän kauan.

5.2.1. Uskova agentti

Uskova agentti-luokka sisältää parametrin ja toiminnot, jotka ovat sekä hyveellisillä että emotivistisilla agenteilla. Uskovan agentin uniikki id on hyödyllinen analyysivaiheessa, jos halutaan seurata yksittäisen agentin kehitystä ajan kuluessa. Vallan ja päättäväisyyden määrä koskevat nykyisessä malliversiossa ainoastaan emotivistisia agenteja, mutta ne on sijoitettu uskovaan agenttiin, koska myös hyveelliselle agentille olisi mahdollista toteuttaa kyseiset mekaniikat.

Attribuutit:

uniikki id, sijainti,

uskomukset sanakirja, jossa avain on uskomus (esim. 'A') ja elementti on uskon vahvuus reaalilukuna välillä 0...1, siten että kaikkien uskomusten vahvuuksien summa on 1. Uskomukset ovat toisiansa poissulkevia, joten jos yksi uskomus vahvistuu, täytyy muiden vastaavasti laskea yhteensä saman verran.

vallan määrä, päättäväisyyden määrä (reaalilukuja)

onnellinen, vakuuttunut, elävä (totuusarvoja)

Toiminnot:

hae vahvin uskomus, muodosta uskomusten merkkijono, kuole

5.2.2. Emotivistinen agentti

Emotivistinen agentti liikkuu jokaisella askeleella satunnaiseen tyhjään ruutuun tai vaihtoehtoisesti jää paikoilleen, jos naapurustossa on tarpeeksi muita emotivisteja. Paikoilleen jäämisessä naapureiden spesifillä uskomuksella ei ole väliä. Emotivistinen agentti levittää jokaisella askeleella omaa uskomustaan naapureille ("väittelee" tai "käännyttää"). Uskomuksen välittymiseen vaikuttaa uskomusta levittävän agentin valta, vastaanottavan agentin päättäväisyys ja ennakkoluuko uskomusta kohtaan. Valinnanvaraisesti voidaan säätää myös, että emotivistinen agentti liikkuu lisäksi tietyllä todennäköisyydellä satunnaisesti toiseen tyhjään ruutuun, riippumatta onnellisuudesta.

Lopuksi emotivistinen agentti tarkistaa onnellisuutensa (oliko naapurustossa tarpeeksi emotivisteja), ja vakuuttumisensa (ylittääkö vahvin uskomus kynnysarvoparametrin). Emotivistinen agentti ei voi kuolla, paitsi kääntyessään hyveelliseksi agentiksi.

Attribuutit:

bias-taulukko: jokaista uskomusta vastaava ennakkoluuloisuus kyseistä uskomusta kohtaan

Metodit:

väittele naapurin kanssa, askel

5.2.3. Hyveellinen agentti

Hyveellinen agentti liikkuu jokaisella askeleella satunnaiseen tyhjään ruutuun tai vaihtoehtoisesti jää paikoilleen, jos naapurustossa on tarpeeksi muita saman uskomuksen jakavia hyveellisiä agenteja. Hyveellinen agentti ei pyri kääntymään muita hyveellisiä agenteja omaan uskomukseensa, vaan tyytyy vahvistamaan naapurustossa olevien samanmielisten uskomuksia ja kääntymään naapurustossa olevia emotivistisia agenteja hyveellisiksi. Jos kääntymys onnistuu, emotivistinen agentti kuolee ja sen tilalle tulee kääntäjän uskomuksen omaava hyveellinen agentti. Valinnanvaraisesti voidaan säätää myös, että hyveellinen agentti liikkuu lisäksi tietyllä todennäköisyydellä satunnaisesti toiseen tyhjään ruutuun, riippumatta onnellisuudesta.

Lopuksi hyveellinen agentti tarkistaa onnellisuutensa ja vakuuttumisensa. Jos hyveellinen agentti on onneton askelen lopussa, se menettää elinvoimaa. Mitä vähemmän samanmielisiä naapureita hyveellisellä agentilla on, sitä enemmän elinvoimaa se menettää. Jos elinvoima laskee alle nollan, hyveellinen agentti kuolee ja poistuu ruudukosta. Jos hyveellinen agentti on onnellinen askeleen lopussa, se saa uutta elinvoimaa

Attribuutit:

elinvoima

Metodit:

käännyttä emotivistisesti hyveelliseksi, vahvista samanmielisten uskomuksia, askel

5.3. Alustus

Ympäristö ja agentit alustetaan mallin parametrien mukaan. Parametreja ovat ruudukon koko, agenttien tiheys, hyveellisten agenttien prosentuaalinen määrä, onnellisuuteen vaadittava määrä samankaltaisia naapureita, uskomuksen muuttumisen määrä, kääntettävien/väiteltävien naapurien määrä, kääntämisen onnistumisen todennäköisyys, vakuuttumisen kynnyksarvo, ylimääräisen satunnaisliikkumisen todennäköisyys, elinvoiman vähentymisen määrä, uskomusten jakauma ja vahvimman uskomuksen alkuarvo, ennakkoluulot us-

komuksia kohtaan, ja ylimääräistä valtaa tai päättäväisyyttä omaavien emotivististen agenttien määrä ja uskomus.

Alustuksen aikana ruudukkoon lisätään uskomusjakaumaparametrien mukainen suhteellinen määrä eri uskomuksiin uskovia agenteja, joiden uskon vahvuus alustetaan alkuarvoksi. Parametreista riippuen osalla emotivistisista agenteista voi olla lisättyä valtaa tai päättäväisyyttä. Valta vahvistaa agentin kykyä levittää omaa uskomustaan, ja päättäväisyys vastaavasti suojaa muiden agenttien vaikuttamisyrityksiltä.

Parametrien vaikutukset voivat olla moninaisia, ja sopivien alkuarvojen löytäminen ei ole itsestään selvää. Esimerkkinä yhteydessä alkuarvot tai parametrivälit on pyritty valitsemaan kokeilemalla siten, että ne tuottavat mielenkiintoista käyttäytymistä eivätkä tukahduta muiden parametrien vaikutusta. Reaalielämän järjestelmää mallinnettaessa alkuarvot voidaan alustaa empiirisen tutkimuksen perusteella. Parametrien vaikutusten havainnollistamiseksi on tehty kaksi herkkyysanalyysiä alakohdassa 5.6.2.

5.4. Mahdolliset tutkimuskysymykset

Eräs kysymys, jota mallilla voi tutkia, on miten ja millä ehdoilla (parametreilla) valtaosa *hyveellisistä agenteista* löytävät tradition ja kykenevät kääntymään valtaosan *emotivistisista agenteista* hyveellisiksi. Toinen mahdollinen tutkimuskysymys on se, miten emotivististen agenttien valta, päättäväisyys ja ennakkoluulot vaikuttavat uskomusten leviämiseen.

5.5. Toteutus

Malli toteutettiin Mesa-ohjelmistokehyksellä hyödyntäen valmiita visualisointi- ja ajastinkomponentteja. Agenttien paikoilleenjämissääntö perustuu Mesa-projektin Schelling-esimerkkiin, joka taas perustuu alun perin Schellingin [1971] esittämään malliin. VirtuousEmotivist-mallin lähdekoodi on liitteessä 1.

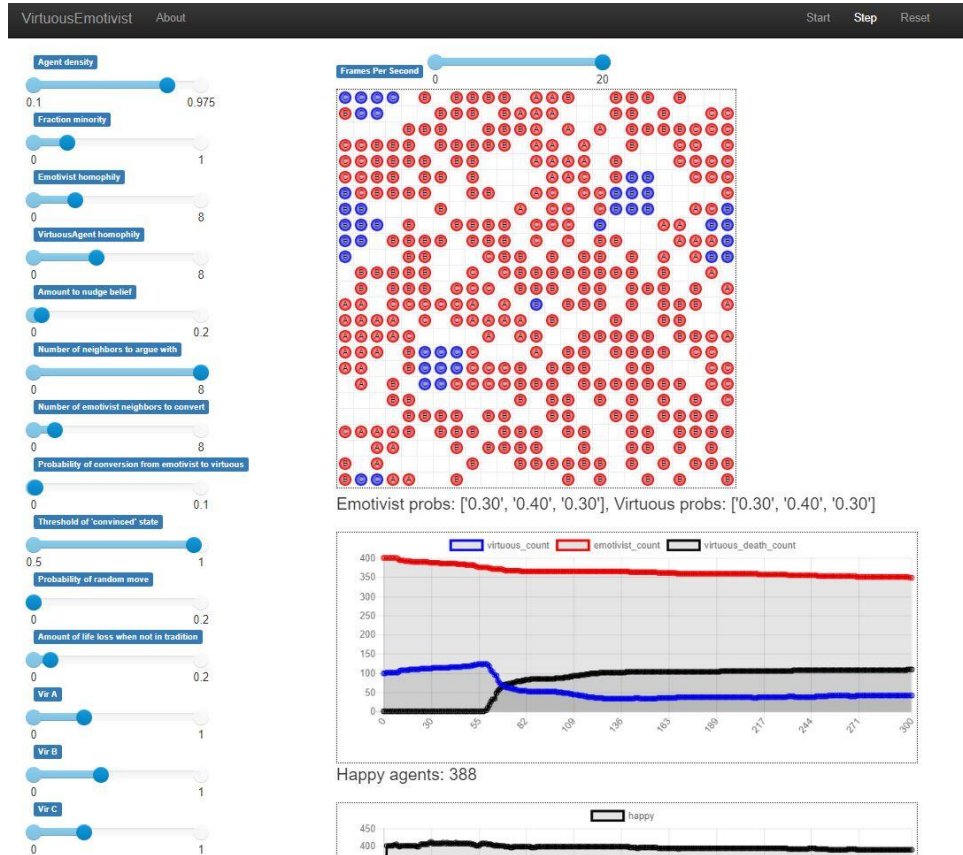
5.5.1. Visualisoinnit

Kuvan 1 demonstraatio ajettiin muutamien poikkeuksien oletusparametreilla. (poikkeukset: `virtuous_homophily=3`, `traditionless_life_decrease=0.02`).

Kuvasta 1 näkyy, että hyveelliset agentit ovat päässeet muodostamaan joitakin kestäviä saarekkeita ("traditioita"), mutta suuri osa hyveellisistä agenteista on ehtinyt kuolemaan löytämättä traditiota (ks. `virtuous_death_count`).

Päävisualisointina käytettiin ruudukkoa, joka havainnollistaa ympäristön, agentit ja niiden vahvimman uskomuksen, sekä ylimääräistä valtaa tai päättäväisyyttä omaavien agenttien määrän (pallon reunus vaihtaa väriä). Ruudukon ohella käyttöliittymä sisältää viivadiagrammit hyveellisten, emotivististen ja

kuolleiden hyveellisten agenttien määrästä, sekä onnellisten ja vakuuttuneiden agenttien määrästä. Viivadiagrammien ja ruudukon lisäksi käyttöliittymässä on vielä palkkidiagrammit hyveellisten ja emotivististen agenttien uskomusjakaumista.



Kuva 1. Interaktiivisen version käyttöliittymä (askel nro. 300)

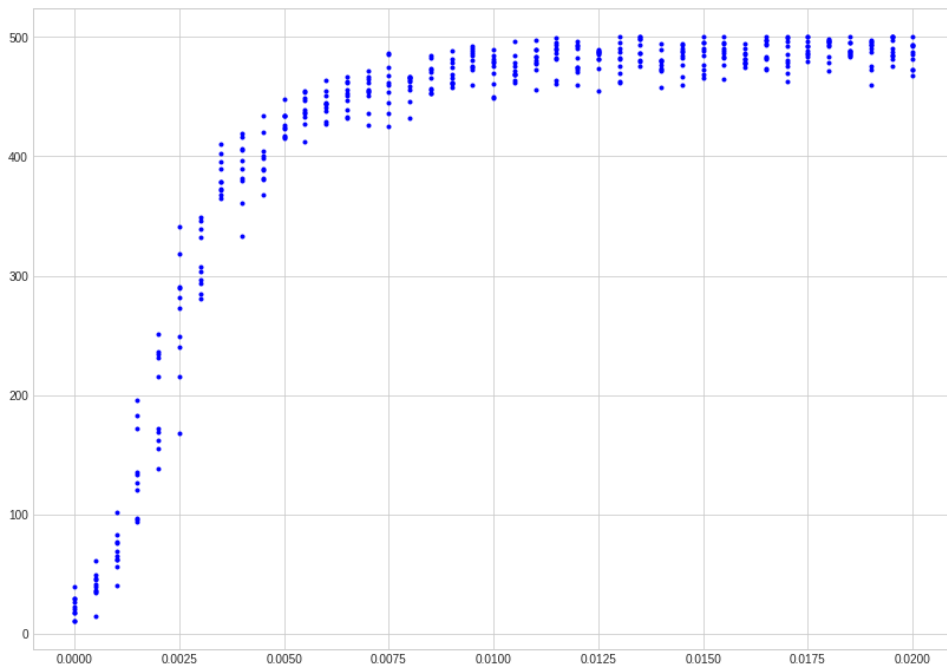
5.6. Tulokset

Tuloksia esitellään tässä luvussa ensin hajontakuviolla, jotka antavat silmämääräisen käsityksen kahden muuttujan välisestä riippuvuudesta. Joidenkin hajontakuvioiden yhteydessä on piirretty myös histogrammi, joka kuvaa muuttujan frekvenssijakaumaa. Jokaista hajontakuvioita kohden mallia ajettiin 100 tai 300 askelta, jokaiselle x-akselin arvolle 10-kertaa vaihtaen satunnaisluku-generaattorin siemenlukua. Ei ole arvioitu, onko 10 kertaa riittävä määrä, eikä mallia olisi ollut edes mahdollista ajaa laskentaresurssien puutteen vuoksi merkittävästi pidempään tai useampia kertoja. Kuvissa 3-7 ylimääräistä valtaa tai päättäväisyyttä omaavien agenttien määrä ei vaikuta sen uskomuksen agenttien kokonaismäärään alussa, vaan päättäväiset ja voimakkaat agentit korvaavat uskomuksensa tavallisia agenteja.

Tulokset ovat toistettavissa ajamalla liitteen 3 Jupyter Notebook-lehtiä. Kiinnostavat parametrit ja parametrivälit valikoitiin ensin ajamalla mallin in-

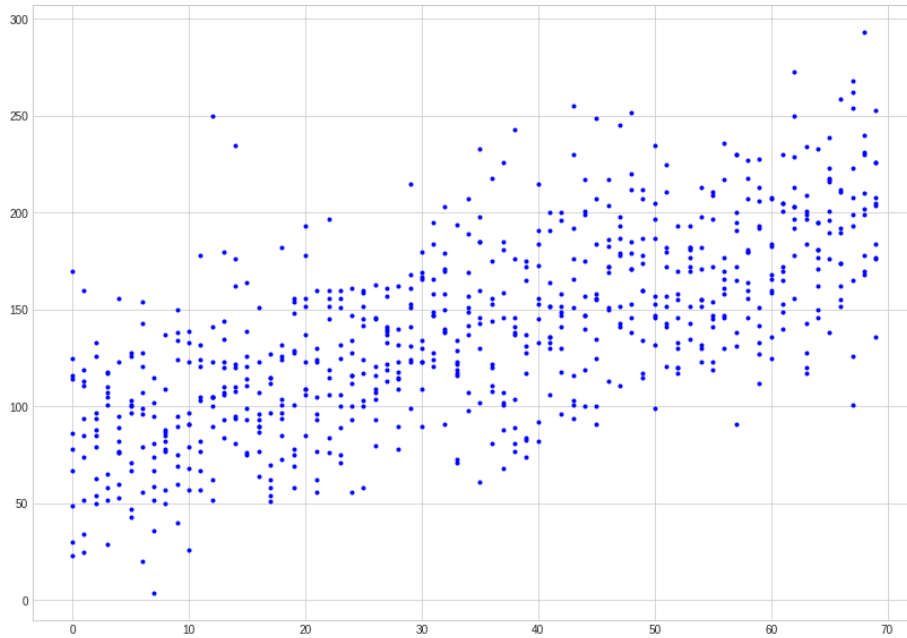
teraktiivista versiota, jonka jälkeen valikoiduilla parametreilla ajettiin mallia rinnakkaisesti IPython-klusterissa, tallentaen valitun ulostulomuuttujan arvot.

5.6.1. Hajontakuviot ja histogrammit



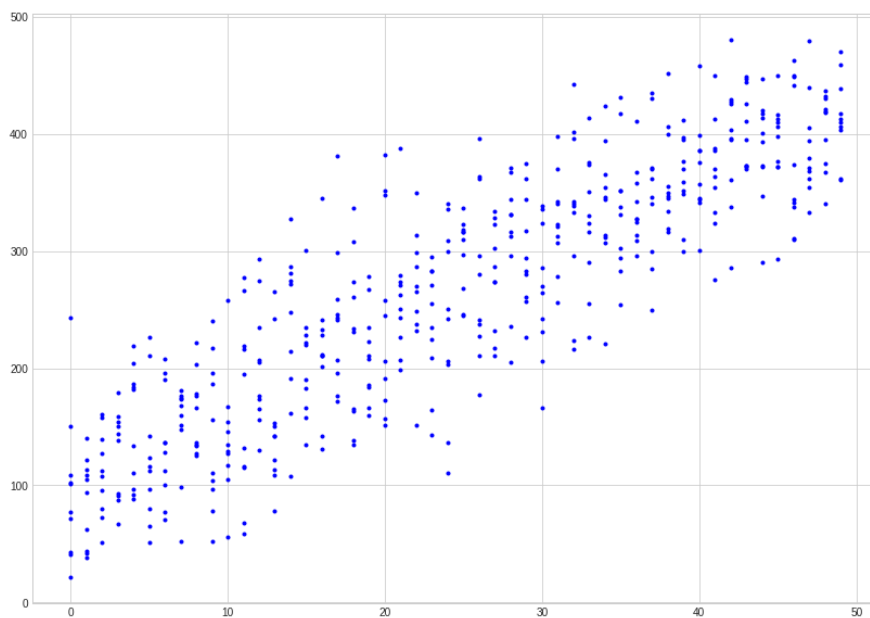
Kuva 2. X-akselilla hyveelliseksi käännyttämisen todennäköisyys. Y-akselilla hyveellisten agenttien määrä 300 askeleen jälkeen. `virtuous_homophily=3`

Kuvasta 2 näkyy, että kun mallia ajetaan 300 askelta, käännyttämistodennäköisyyden vaikutus on melko jyrkkä. Useimmiten yli yhden prosentin käännyttämistodennäköisyyksillä lähes koko ruudukko täyttyy hyveellisistä agenteista.



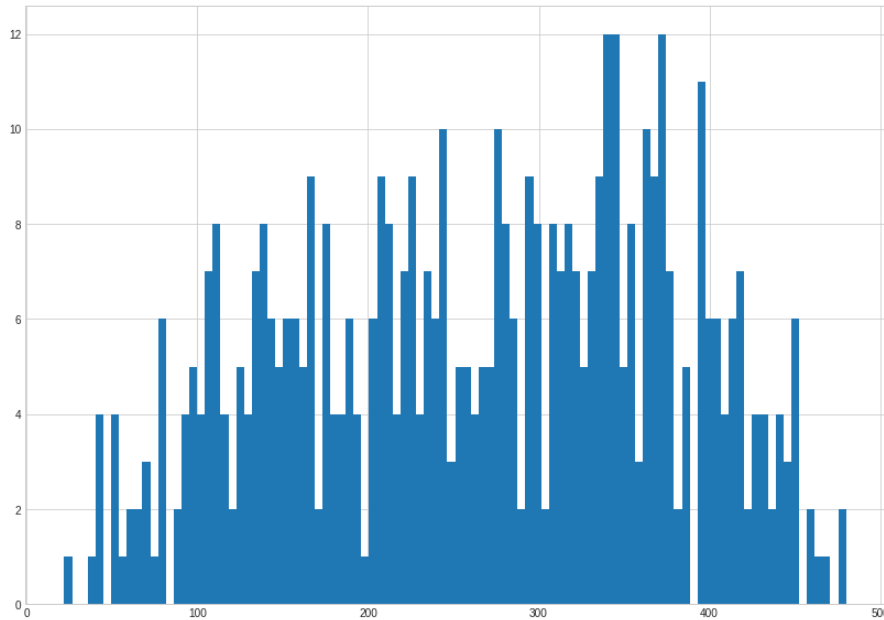
Kuva 3. Mallissa pelkästään emotivistisia agenteja. X-akselilla extra-päätäväisten ($extra_det=0.75$) A-uskovien agenttien lukumäärä 0-askeleella. Y-akselilla kaikkien A-uskovien agenttien lukumäärä 300 askeleen jälkeen. $homophily=3$

Kuvasta 3 näkyy, että päätäväisten A-uskovien agenttien määrällä on vaikutus A-uskovien agenttien kokonaismäärään. Agentin päätäväisyys merkitsee sitä, että muiden käännytysyritykset tehoavat heikommin kyseiseen agenttiin. Päätäväisyys näyttää korreloivan käännytettyjen määrän kanssa, vaikkakin vaihtelu on suurta.



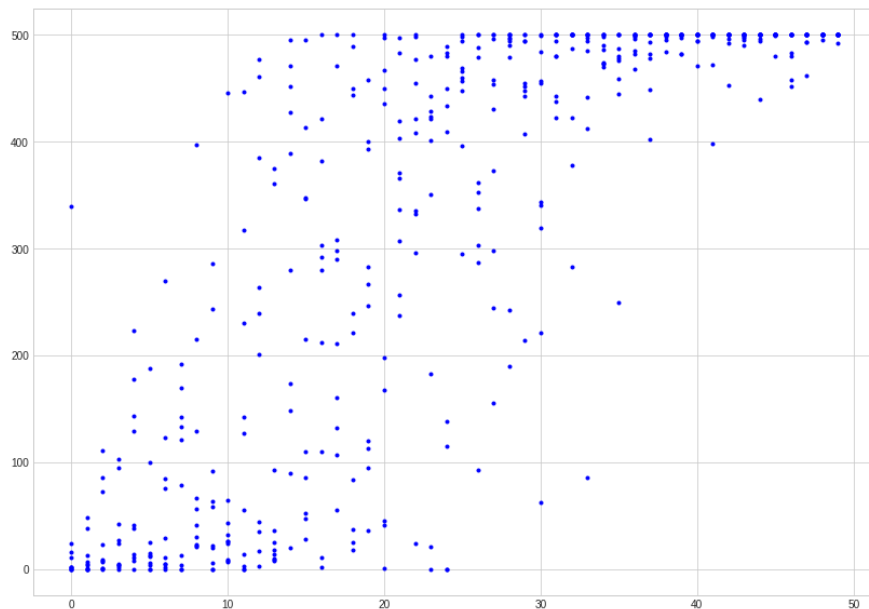
Kuva 4. Mallissa pelkästään emotivistisia agenteja. X-akselilla extra-päätäväisvoimakkaiden ($extra_detpow$) A-uskovien agenttien lukumäärä askeleella 0. Y-

akselilla kaikkien A-uskovien agenttien lukumäärä. Mallia ajettiin 100 askelta. $\text{random_move_prob}=0.0125$

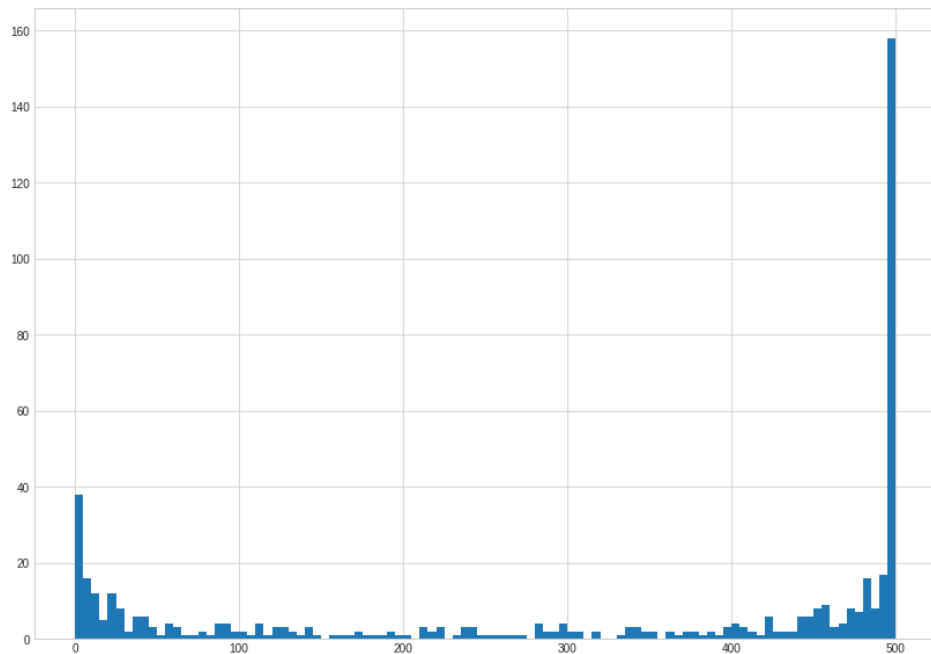


Kuva 5. Vastaava histogrammi kuvalle 4. X-akselilla A-uskovien lukumäärä askeleella 300. Y-akselilla lukumäärän frekvenssi.

Kuvasta 4 näkee, että lyhyellä aikavälillä (100 askelta) A-uskovien detpow-agenttien määrällä on melko lineaarinen vaikutus A-uskovien lukumäärään.



Kuva 6. Vastaavanlainen tilanne kuin kuvassa 4, mutta mallia ajettiin 300 askelta sadan askeleen sijaan. Mallissa pelkästään emotivistisia agenteja. X-akselilla extra-päätäväis-voimakkaiden (detpow) A-uskovien agenttien lukumäärä askeleella 0. Y-akselilla kaikkien A-uskovien agenttien lukumäärä.



Kuva 7. Vastaava histogrammi kuvalle 6. X-akselilla A-uskovien lukumäärä askeleella 300. Y-akselilla lukumäärän frekvenssi.

Kuvista 6 ja 7 näkyy, että 300 askeleen jälkeen mallilla on kaksi yleistä lopputulosta (0 tai 500). Tämän voi tulkita siten, että useimmiten joko A-uskovat "häviävät" ja joutuvat enemmistön (B-uskovat) käännyttämäksi, tai sitten A-uskovat "voittavat" ja onnistuvat käännyttämään kaikki muut agentit puolelleen.

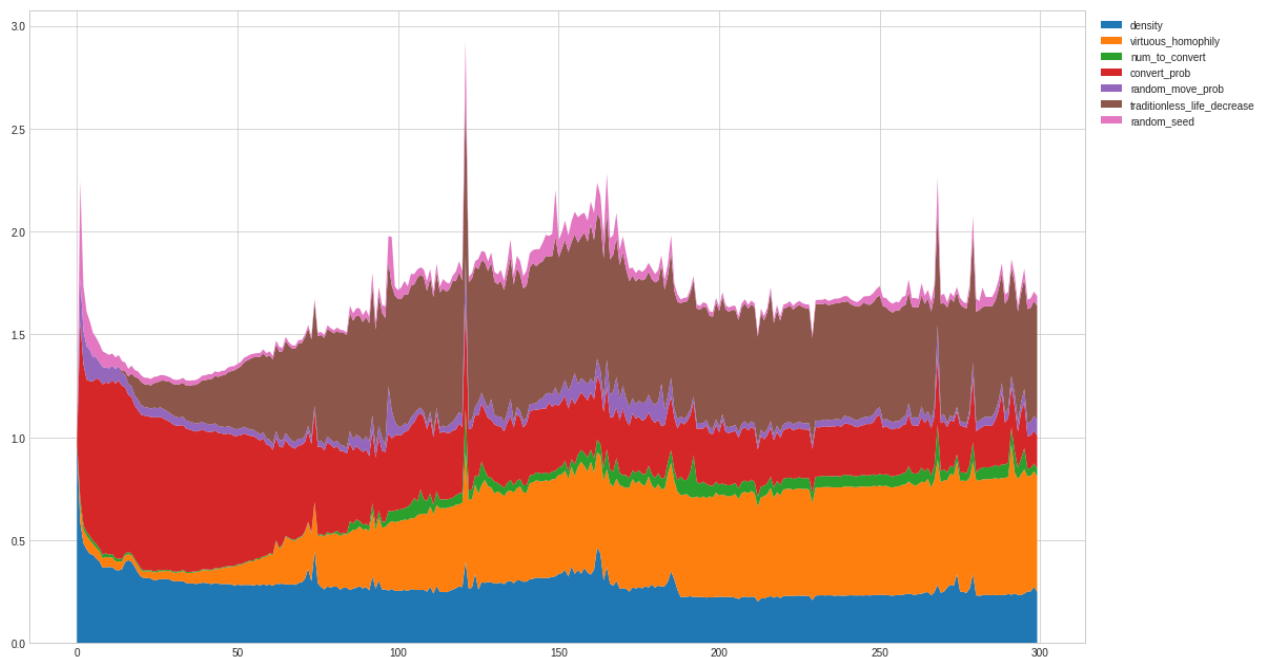
5.6.2. Herkkyysanalyysit

Herkkyysanalyysit ovat joukko menetelmiä, joilla voidaan tarkastella mitkä mallin parametrit aiheuttavat lopputuloksen vaihtelun, ts. miten herkkä mallin lopputulos on eri parametrien muuttamiselle. Tässä luvussa esitellyt diagrammit on muodostettu laskemalla *Sobol-kokonaisvaikutusindeksit* osajoukolle kaikista parametreista; muut parametrit on alustettu oletusarvoiksi. Herkkyysindeksien tarkemmat määritelmät sivuutetaan, ja tyydytään tämän tutkielman puitteissa pelkästään kokeilemaan menetelmän valmista toteutusta (SALib-kirjasto). Koska menetelmän rajoitteita tai vaatimuksia ei ole tarkistettu, tulokset eivät välttämättä ole luotettavia.

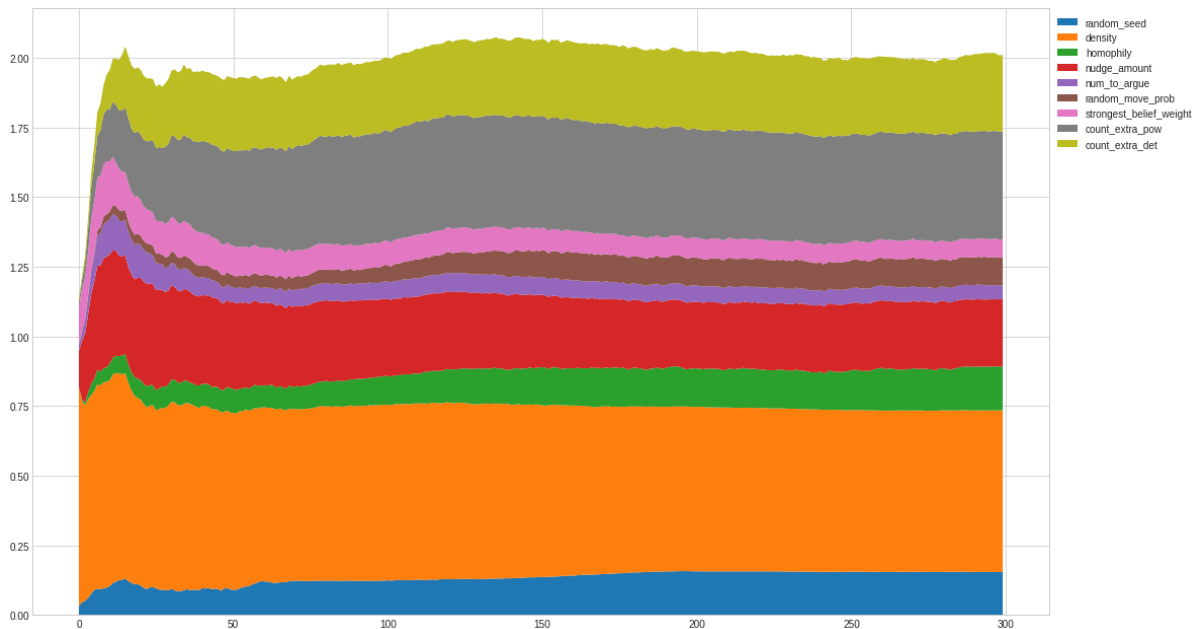
Määritelmällisesti *ensimmäisen kertaluvun herkkyysindeksi* kuvaa tietyn parametrin (syötteen) vaikutusta mitatun muuttujan (lopputulos/ulostulo) varianssiin. *Toisen ja korkeamman kertaluvun herkkyysindeksit* kuvaavat parametrien interaktioiden vaikutusta mitatun muuttujan varianssiin. *Kokonaisvaikutusindeksi* (total-effect index) kuvaa yhteen lasketusti sekä ensimmäisen kertaluvun vaikutusta, että parametrin kaikkien interaktioiden vaikutusta. Interaktiolla tarkoitetaan tilannetta, kun kahden tai useamman parametrin vaikutus mallin

lopputulokseen ei ole pelkästään parametrien erillisvaikutusten summa. [Saltelli *et al.* 2008.]

Mallia ajettiin noin 2000 kertaa, jokainen kerta 300 askeleen verran. Ajokertojen parametrit valittiin satunnaisotoksena määritellyiltä parametriväleiltä käyttämällä SALib-kirjaston Saltelli-otosfunktiota. Jokaisella askeleella tallennettiin tarkasteltavan muuttujan arvo herkkyysanalyysia varten. Ei ole arvioitu, onko 2000 kertaa riittävä otoskoko. Lasketuista herkkyysindekseistä muodostuu aikasarja (yksikkönä askel), josta selviää, mitkä parametrit olivat ratkaisevassa asemassa tietyllä askeleella. [Ligmann-Zielinska and Sun 2010.]



Kuva 8. Tilanne: Hyveelliset agentit pyrkivät kääntymään emotivistisia agentteja. Y-akselilla kokonaisvaikutusindeksit: väritetyt alueet kuvaavat seitsemän eri parametrin vaikutusta (mukaan lukien interaktiot kyseisen parametrin kanssa) hyveellisten agenttien määrään. X-akselilla askelnumero ("aika").



Kuva 9. Tilanne: A-uskovat emotivistiset agentit pyrkivät kääntämään muita emotivistisia agenteja A-uskoviksi. Y-akselilla kokonaisvaikutusindeksit: väritetyt alueet kuvaavat yhdeksän eri parametrin vaikutusta (mukaan lukien interaktiot kyseisen parametrin kanssa) A-uskovien agenttien määrään. X-akselilla askelnumero ("aika"). Count_extra-parametrit koskevat A-uskovia agenteja. Mallissa pelkästään emotivistisia agenteja

5.6.3. Tulosten luotettavuus

Tulosten luotettavuutta ei ole arvioitu, koska tutkielman tavoitteena ei ole esittää päteviä tuloksia malliesimerkin osalta. Aika- ja laajuusrajoitteiden vuoksi on mahdollista, että tuloksissa on puutteita tai epävarmuuksia, koska kaikkia asiaankuuluvia tekijöitä tai menetelmien ennakoitua ei ole otettu huomioon.

6. Tietokonesimulaatioiden läpinäkymättömyys

Mitä enemmän malliin lisätään parametreja ja käyttäytymistä, sitä enemmän ajatuskoemainen ymmärtäminen vaikeutuu. Yksittäisten parametrien vaikutus ei enää havainnollistu intuitiivisesti, ja voi riippua monesta muusta parametrista, joiden yhteisvaikutus on intuitiivisesti epäselvä. Tulosten tarkempaan analysointiin tarvitaan tilastollisia menetelmiä. Voidaan nähdä, että simulaatioajatuskokeiden hyvä puoli on juuri se, että niistä voidaan systemaattisesti ja automatisoidusti kerätä ja analysoida dataa, toisin kuin perinteisistä ajatuskokeista. Tämä hyöty voi kuitenkin seurata ymmärryksen kustannuksella, jolloin simulaatiomalli alkaa muistuttaa "mustaa laatikkoa". Tulosten kommunikointi muille tiedeyhteisön jäsenille voi olla vaikeaa, esimerkiksi simulaatioiden tutkimusraportoinnin standardien puutteen vuoksi. [Lorscheid *et al.* 2012.]

Jatkotutkimuksen kannalta mielenkiintoista olisi tutkia esimerkiksi millaiset visualisoinnit ja suunnitteluparadigmat edistävät ajatuskoemaista ymmärtämistä, ja miten tulosten analysoinnista saisi tehtyä käyttäjäystävällisempää. Jos agenttipohjaista mallintamista sovelletaan filosofian tutkimuksen piirissä, on kenties epätavallisempaa, että simulaation lopputuloksia raportoitaisiin, ainakin jos odotettavissa ei ole suurta muutosta akateemisen filosofian metodologiassa. Tällöin on erityisen tärkeää, että mallintamisen tuloksena syntyy ymmärrys, jonka voi ilmaista sanallisesti ilman viittausta alkuperäiseen malliin.

7. Yhteenveto

Tässä tutkielmassa esiteltiin agenttipohjainen mallinnus ja tutkittiin sen sovelluksia akateemiseen filosofiaan, sekä teoriassa että käytännössä. Teorian tasolla todettiin, että agenttipohjaisella mallintamisella toteutetut ”ajatuskokeet” kohtaavat tieteenfilosofisia ongelmia. Kysymys on pitkälti siitä, voiko mallin kehitysprosessi ja lopputulos tuottaa ymmärryksen, selityksen tai jopa havainnon jostakin ilmiöstä, ja jos voi, minkä luonteisia nämä ovat.

Käytännön tasolla tutkielman tekeminen tuotti esimerkkimallin, joka ilmentää paikoitellen mielenkiintoista käyttäytymistä, mutta ei sovellu ainakaan sellaisenaan mallin pohjalla olevan filosofisen argumentin (MacIntyren modernin moraalifilosofian kritiikki) erittelyyn kovinkaan helposti. Yksityiskohtaisempi soveltaminen edellyttäisi syvempää käyttäytymisen mallintamista kuin mitä tämän tutkielman puitteissa oli aiheellista lähteä toteuttamaan.

Esimerkkimallin ja kirjallisuuskatsauksen perusteella näyttää mahdolliselta, että agenttipohjaista mallintamista voisi soveltaa myös filosofian tutkimuksessa. Analyysimenetelmien, mallin havainnollistamisen ja toiminnan ymmärtämisen kohdalla esille nousseiden haasteiden vuoksi on kuitenkin ymmärrettävää, miksi agenttipohjainen mallinnus ei ole saavuttanut laajempaa suosiota filosofian tutkimuksessa.

Viiteluettelo

- Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldridge. 2007. *Programming Multi-Agent Systems in AgentSpeak Using Jason*. Wiley.
- Andrei Borshchev and Alexei Filippov. 2004. From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools. In: *Proc. of the 22nd International Conference of the System Dynamics Society*, 959-966.
- Lars Braubach, Alexander Pokahr and Winfried Lamersdorf. 2005. Jadex: A BDI-agent system combining middleware and reasoning. In: Rainer Un-

- land, Monique Calisti, Matthias Klusch (eds.). *Software Agent-Based Applications, Platforms and Development Kits*. Birkhäuser Basel. 143-168.
- Herbert Dawid, Simon Gemkow, Philipp Harting, Sander van der Hoog, and Michael Neugart. 2012. The Eurace@Unibi model: An agent-based macro-economic model for economic policy analysis. Bielefeld Working Papers in Economics and Management No. 05-2012. Bielefeld University, Faculty of Business Administration and Economics.
- Daniel Dennett. 1994. Artificial life as philosophy. *Artificial Life* 1, 291-292.
- Ezequiel A. Di Paolo, Jason Noble, and Seth Bullock. 2000. Simulation models as opaque thought experiments. *Artificial Life* 7, 497-506.
- Brian Epstein. 2011. Agent-based modeling and the fallacies of individualism. In: Paul Humphreys and Cyrille Imbert (eds.), *Models, Simulations, and Representations*. Routledge, 115-144.
- Joshua M. Epstein. 2006. *Generative Social Science: Studies in Agent-Based Computational Modeling*. Princeton University Press.
- Kai Koskimies ja Tommi Mikkonen. 2005. *Ohjelmistoarkkitehtuurit*.
- Kalliopi Kravari and Nick Bassiliades. 2015. A survey of agent platforms. *Journal of Artificial Societies and Social Simulation* 18, 1, Article 11.
- Francisco J. León-Medina. 2017. Analytical sociology and agent-based modeling: Is generative sufficiency sufficient? *Sociological Theory* 35, 3, 157-178.
- Arika Ligmann-Zielinska and Libo Sun. 2010. Applying time-dependent variance-based global sensitivity analysis to represent the dynamics of an agent-based model of land use change. *International Journal of Geographical Information Science* 24, 12, 1829-1850.
- Iris Lorscheid, Bernd-Oliver Heine, and Matthias Meyer. 2012. Opening the 'black box' of simulations: increased transparency and effective communication through the systematic design of experiments. *Computational and Mathematical Organization Theory* 18, 22-62.
- Alasdair C. MacIntyre. 1981. *After Virtue*. University of Notre Dame Press.
- David Masad and Jacqueline Kazil. 2015. Mesa: an agent-based modeling framework. In: *Proc. of the 14th Python in Science Conference*, 53-60.
- Andrea Saltelli, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, and Stefano Tarantola. 2008. *Global Sensitivity Analysis: The Primer*.
- Thomas C. Schelling. 1971. Dynamic models of segregation. *Journal of Mathematical Sociology* 1, 488-493.
- Seth Tisue and Uri Wilensky. 2004. NetLogo: A simple environment for modeling complexity. In: *Proc. of the International Conference on Complex Systems*, 16-21.

- Kevin Vallier. 2017. Three concepts of political stability: an agent based model. *Social Philosophy and Policy* 34, 1, 232-259.
- Uri Wilensky and William Rand. 2015. *An Introduction to Agent-Based Modeling : Modeling Natural, Social, and Engineered Complex Systems with NetLogo*. MIT Press.
- Yu-Hsiang Yang, Huimin Bhikshu and Rua-Huan Tsaih. 2015. The power of one sentient being: The computer simulation of a bodhisattva's altruism using agent-based modelling. *Contemporary Buddhism* 16, 2, 330-354.

Liite 1. model.py**Liite 2. server.py****Liite 3. VirtuousEmotivistNotebook.ipynb**

Koodiliitteet ovat niin pitkiä, ettei niitä ole sisällytetty tähän dokumenttiin. Tutkielman ohella palautetaan lähdekoodi, jonka uusin versio on myös saatavilla osoitteesta <https://github.com/woroko/VirtuousEmotivist>

Palveluarkkitehtuuri liiketoimintatiedon hallinnan järjestelmissä

Jonna Paksunen

Tiivistelmä.

Tutkielmassani esittelen palveluarkkitehtuuria ja liiketoimintatiedon hallinnan järjestelmiä, ja tarkemmin toiminnanohjausjärjestelmiä. Tutkielmassa tavoitteenä oli selvittää, miten palveluarkkitehtuurin soveltaminen liiketoimintatiedon hallinnan järjestelmissä parantaisi järjestelmien laatua ja niiden kehitysprosessia. Muiden palveluarkkitehtuurin sovelluskohteiden ja palveluarkkitehtuurin luonteen perusteella palveluarkkitehtuurista on hyötyä liiketoimintatiedon hallinnan järjestelmien kehityksessä, sillä se alentaa järjestelmän kehityskustannuksia ja yhtenäistää toimintoja.

Avainsanat ja -sanonnat: Palveluarkkitehtuuri, liiketoimintatiedon hallinta, liiketoimintatiedon hallinnan järjestelmät, toiminnanohjausjärjestelmä

1. Johdanto

Liiketoimintatiedon hallinta on keskeinen osa yritysten toimintaa ja päätöksentekoa. Nykäsen ja muiden [2016] mukaan 95 prosentissa suomalaisista huippuyrityksistä käytetään päätöksenteon tukena liiketoimintatiedon hallinnan järjestelmiä, jotka parantavat ja nopeuttavat päätöksentekoa. Järjestelmien ollessa merkittävässä roolissa yritysten päätöksenteossa niihin käytetään väistämättä paljon työtunteja. Esimerkiksi rahoitusalan ammattilaiset käyttävät yli puolet työajastaan järjestelmien käyttöönottoon, valintaan ja koulutukseen liittyvissä tehtävissä [Nykänen ym. 2016].

Palveluarkkitehtuurilla tai palvelukeskeisellä arkkitehtuurilla on hieman eri merkityksiä eri yhteyksissä. Laskey ja Laskey [2009] määrittelevät palveluarkkitehtuurin ajatusmalliksi, jolla organisoidaan ja pakataan toiminnallisia yksiköitä erillisinä palveluina, joita voidaan yhdistellä ratkaisuksi liiketoiminnan ongelmiin. Palveluarkkitehtuuri on myös määritelty tietojärjestelmien arkkitehtuurimalliksi, joka tukee palvelukeskeistä liiketoiminnan kehitystä [opengroup 2018].

Koska yritykset käyttävät suuren osan sekä rahallisista että henkilöresursseistaan liiketoimintatiedon hallinnan järjestelmiin, on oleellista, että järjestelmät toimivat, ne ovat tehokkaita ja helppoja käyttää. Palveluarkkitehtuuri tarjoaa uuden mallin tietojärjestelmien suunnittelulle. Se siirtää palvelut suunnittelun keskiöön aiemman järjestelmäkeskeisyyden sijaan, mikä saattaisi mahdollistaa tehokkaampien ja toimivampien järjestelmien toteuttamisen.

Tutkielmassani pyrin selvittämään palveluarkkitehtuurin hyötyjä yritys-elämälle liiketoimintatiedon hallinnan järjestelmien näkökulmasta. Aloitan katsauksella liiketoimintatietoon ja miten yritykset hyödyntävät sitä, minkä jälkeen esittelen muutamia käytössä olevia liiketoimintatiedon hallinnan järjestelmiä. Seuraavaksi avaan tarkemmin palveluarkkitehtuuria käsitteenä ja pohdin, miten sitä voidaan hyödyntää liiketoimintatiedon hallinnan järjestelmissä ja onko sitä jo mahdollisesti hyödynnetty. Vertailen myös palveluarkkitehtuurin hyötyjä ja haittoja muihin arkkitehtuurimalleihin verrattuna. Lopuksi esittelen tutkielmani johtopäätökset.

2. Liiketoimintatiedon hallinta

Liiketoimintatieto on tietoa, jonka varassa yrityksen johto tekee liiketoimintapäätöksiä. Tätä tietoa voidaan hankkia sekä yrityksen sisäisesti että sen ulkopuolelta. Sisäistä tietoa saadaan esimerkiksi käytössä olevista toiminnanohjausjärjestelmistä tai IoT-ratkaisujen (Internet of Things, asioiden Internet) myötä Internetiin liitetyistä asioista tai esineistä. Näitä esineitä ja asioita voivat olla esimerkiksi työkoneet. Ulkoista tietoa yritys voi kerätä muun muassa mediasta, markkinatutkimuksista tai kilpailijoiden vuosikertomuksista. Tässä tutkielmassa keskityn sisäiseen tietoon ja sen keruussa käytettäviin järjestelmiin.

Elbashirin ja muiden [2008] mukaan liiketoimintatiedon hallinta tarkoittaa keinoja analysoida liiketoimintatietoa johdon päätöksenteon tukena erilaisissa liiketoimintatehtävissä. Wu ja muut [2007] puolestaan määrittivät liiketoimintatiedon hallinnalle kolme keskeistä tavoitetta: 1) tarjota yksi versio totuudesta koko organisaatiolle, 2) tarjota yksinkertainen järjestelmän toteutus, käyttöönotto ja ylläpito, 3) toimittaa strategista, taktista ja operationaalista tietoa. Liiketoimintatiedon hallinta voidaankin tiivistää tarkoitukseksi parantaa organisaatioiden suorituskykyä paremman päätöksenteon myötä [Nykänen ym. 2016].

Liiketoimintatiedon hallinta otetaan yrityksissä usein käyttöön, koska yrityksen johto haluaa parantaa yrityksen toimintaa [Nykänen ym. 2016].

2.1 Liiketoimintatiedon hallinnan järjestelmät

Tässä kohdassa esittelen yleisesti liiketoimintatiedon hallinnan järjestelmiä sekä järjestelmien luokittelua. Teen myös lyhyen katsauksen näiden järjestelmien historiaan.

Liiketoimintatiedon hallinta (business intelligence) terminä syntyi jo 1800-luvun lopulla, mutta erilliseksi tieteenalaksi se muotoutui vasta 1950-luvulla. IBM ja Siebel, nykyisin Oracle, alkoivat 1970-luvulla kehittää ensimmäisiä liiketoimintatiedon hallinnan järjestelmiä, jatkaen aina 1990-luvulle asti. 2000-

luvulla markkinoilla oli jo useita liiketoimintatiedon hallinnan järjestelmiä. Pian palveluntuottajat alkoivat tuottaa pilvipalveluissa toimivia järjestelmiä, jotka toimivat helposti useilla eri alustoilla. Vuoden 2016 alussa liiketoimintatiedon hallinnan järjestelmät tuottivat jo noin 9 miljardia dollaria liikevoittoa.

Markkinoilla on useita valmiita järjestelmäratkaisuja liiketoimintatiedon hallinnalle, kuten SAP, Microsoft SharePoint ja DundasBI. Useimmat järjestelmät ovat räätälöitävissä yrityksen koon ja tarpeiden mukaan. Joidenkin yritysten liiketoiminta voi kuitenkin olla niin erityistä, että ne päätyvät teettämään juuri omiin tarpeisiinsa sopivan järjestelmän.

Liiketoimintatiedon hallinnassa yleisiä tietojärjestelmäratkaisuja ovat eriliset sovellukset, jotka voidaan luokitella yhdeksään eri tyyppiin. Taulukkolaskentaohjelmilla, kuten Microsoftin Excelillä, käsitellään numeerista dataa. Raportointiohjelmistot keräävät dataa eri lähteistä ja koostavat ne luettavaan muotoon. Verkossa toimivat *analyttiset prosessointijärjestelmät* (OLAP, online analytical processing) kehitettiin vastaamaan nopeasti *moniulotteisiin analyttisiin kyselyihin* (MDA, multidimensional analysis), esimerkiksi myynnin raportointiin. Erilaiset *hallintapaneelit* (business dashboard) koostavat yhdelle näytölle reaaliaikaista tietoa yrityksen toiminnasta, esimerkiksi kaavioiden ja taulukoiden muodossa. Hallintapaneelien kaltaisia sovelluksia ovat myös liiketoiminnan toimien seurantasovellukset (BAM, business activity monitoring), jotka toisin kuin hallintapaneelit, näyttävät laajemmin tietoa yrityksen toiminnasta, esimerkiksi riskienhallinnasta. Tiedonlouhinnalla pyritään löytämään oleellista tietoa suurista tietomassoista. Datavarastot (data warehouse) ovat useasta eri lähteestä integroidun datan keskussäilöjä ja niitä käytetään raportointiin sekä data-analyysiin. Lokaalit informaatiojärjestelmät (LIS, local information systems) rakennetaan muita liiketoimintatiedon hallinnan työkaluja käyttäen pääosin maantieteellisen raportoinnin tueksi. Datan puhdistus (data cleansing, data cleaning) on prosessi, jossa etsitään ja poistetaan tai korjataan korruptoitunutta tai väärää tietoa esimerkiksi tietokannoista.

Monet näistä edellä mainituista sovelluksista sisältyvät usein suurempiin järjestelmäkokonaisuuksiin, kuten toiminnanohjausjärjestelmiin. Niitä voidaan myös koota komponenteiksi jonkin tietyn teollisuuden alan tarpeisiin räätälöityyn ohjelmistoon. Joitakin sovelluksia voidaan lisäksi tarjota ohjelmointirajapintoina.

Suurin osa tarjolla olevista sovelluksista ja järjestelmistä ovat maksullisia ja yksityisomisteisia, mutta tarjolla on muutamia avoimen lähdekoodin sekä ilmaisia että maksullisia järjestelmiä.

2.2 Toiminnanohjausjärjestelmät

Ehkä suosituimpia liiketoimintatiedon hallinnan järjestelmiä ovat *toiminnanohjausjärjestelmät* (ERP, enterprise resource planning), esimerkiksi SAP. Tyypillisesti toiminnanohjausjärjestelmät ovat integroituja järjestelmiä, sillä ne toimivat lähestulkoon reaaliaikaisesti. Niillä on yksi tietokanta, jota kaikki sovellukset hyödyntävät ja niillä on yhtenäinen ulkonäkö sovellusten välillä ja niiden käyttöönotto vaatii huolellista sovelluksien integraatiota. Koska toiminnanohjausjärjestelmät koostuvat näistä aiemmin mainituista sovelluksista, ne tarjoavat työkaluja yrityksen keskeisimpien prosessien hallintaan. Informaatio yrityksen toiminnasta ja prosesseista päivittyy jatkuvasti, jolloin yrityksen johto voi saada reaaliaikaista tietoa päätöksenteon tueksi. [Wikipedia 2018]

Toiminnanohjausjärjestelmien kehityksen voidaan katsoa alkaneen 1960-luvulla varastohallinnasta, joka laajeni varasto- ja tilausjärjestelmäksi 1980-luvulle saavuttaessa. Ensimmäisiä varsinaisia toiminnanohjausjärjestelmiä tuli markkinoille 1990-luvulla. Tällä hetkellä uusimmat toiminnanohjausjärjestelmät luokitellaan uuden sukupolven elinkaarettomiksi toiminnanohjausjärjestelmiksi (cERP). [Aamuvuori ja Valtee 2017]

Toiminnanohjausjärjestelmät koostuvat moduuleista eli alijärjestelmistä, jotka sisältävät aiemmin mainittuja sovelluksia. Kirjanpitojärjestelmään kuuluvat esimerkiksi pääkirja ja ostovelat, ja laskentatoimen järjestelmään muun muassa budjetointi. Henkilöstöhallinnan järjestelmässä käsitellään yrityksen henkilöstöön liittyviä toimia, esimerkiksi rekrytoinnin tietoja ja eläköitymissuunnitelmia. Tuotannon järjestelmä sisältää tietoa muun muassa laadunhallinnasta ja tuotteista, ja tuotannon järjestelmään liittyvä tilausten käsittelyjärjestelmä käsittelee tietoa esimerkiksi tilauksista, varastotilanteesta ja toimituksista. Toimitusketjujen ja logistiikan hallintajärjestelmässä käsitellään osittain samaa tietoa kuin tilausten käsittelyjärjestelmässä (esimerkiksi varaston tilanteesta), minkä lisäksi järjestelmä sisältää tietoa toimituksista ja palveluketjusta. Projektinhallintajärjestelmän työkaluja käytetään projektin suunnitteluun, aikataulutukseen, kustannusarvioiden tekemiseen ja muihin projektille oleellisiin tehtäviin. Toiminnanohjausjärjestelmiin voi sisällyttää moduulin myös asiakassuhteiden hallinnalle (customer relationship management, CRM), joskin nämä järjestelmät luokitellaan joissain tapauksissa liiketoiminnan tuen järjestelmiin (business support system, BSS). Lisäksi toiminnanohjausjärjestelmissä on datapalveluita, jotka ovat erilaisia niin sanottuja itsepalvelurajapintoja esimerkiksi asiakkaille tai työntekijöille. [Wikipedia 2018]

Toiminnanohjausjärjestelmiä tuottava ja toimittava yritys Oscar Software on määritellyt seitsemän ”ERP-syntiä” eli tavallisimpia vanhan järjestelmän

ongelmia teoksessaan *Toiminnanohjauksen onnena ja tuonela: ERP:stä liiketoiminta-alustaan – kokeneet konkarit kertovat* [Aamuvuori ja Valtee 2017].

1. Manuaalinen ja vajaakäytetty ERP. Järjestelmää hyödynnetään vain laskukoneena, ja kaikki muu tehdään manuaalisesti. Tällöin työtunteja kuluu turhaan manuaaliseen tietojen käsittelyyn, eli yritys menettää rahaa.

2. Päällekkäiset ohjelmistot. Yrityksellä voi olla jokaiselle toiminnan osaluokalle, esimerkiksi tuotannolle ja varastolle omat erilliset järjestelmät yhden kaiken kattavan järjestelmän sijaan.

3. Liian iso ohjelmisto tai liian iso toimittaja. Pienen yrityksen ei kannata ostaa suuren monikansallisen yrityksen toimittamaa suurille yrityksille suunniteltua järjestelmää.

4. Vanha järjestelmä, joka on usein muuttumaton ja jäykkä. Järjestelmä ei tällöin mahdollista yrityksen toimintatapojen kehittämistä.

5. Ohjelmiston elinkaari on päättymässä tai päivitys on mahdotonta räätälöinnin vuoksi. Ohjelmiston tuki on loppumassa, eikä sitä enää kehitetä. Liian pitkälle kustomoitua ohjelmistoa on vaikeaa ja kallista päivittää. Mukauttamisen sijaan voisi harkita valmiita ratkaisuja tarjoavaa ohjelmistoa.

6. Järjestelmä rakentuu liian monista integroiduista järjestelmistä, ettei toimintoja uskalleta kehittää liiketoiminnan vuoksi, vaan ohjelmaa kehittäessä joudutaan aina turvautumaan uusiin integraatioihin. Nämä integraatiot voivat rikkoutua helposti päivitysten yhteydessä, ja on vaikea osoittaa, kenen vastuulla on korjata ilmenevät ongelmat. Integraatioiden päivitys on myös kallista.

7. Järjestelmän puute. Ilman kunnollista yritysten tarpeet kohtaavaa järjestelmää kasvu on hankalaa sekä automatisointi on mahdotonta. Yrityksen liiketoiminnasta on myös hankala saada kokonaiskuvaa.

Aamuvuori ja Valtee [2017] esittelevät nykyaikaisen liiketoiminta-alustan, joka ratkaisee aiemmin esiteltyt perinteisen toiminnanohjausjärjestelmän sudenkuopat. Heidän mukaan tämän päivän piensovellusten teko nopealla syklillä voi johtaa siihen, että päivitettävyyden ja ylläpidettävyyden vaarantuvat. Nykyaikainen liiketoiminta-alusta kuitenkin korjaisi tämän ongelman erottamalla käyttöliittymän ja liiketoiminta-alustan kehityksen toisistaan erillisiksi kehityshaaroiksi, jolloin voidaan toteuttaa kohdennettuja näkymiä ja toimintoja ERP-moduulien sijaan. Aamuvuori ja Valtee [2017] argumentoivat, että näin toimimalla ”kehityssykli ja uusimmat teknologiat pysyvät jatkuvasti ajan tasalla, mutta perus businesslogiikan elinkaari ei vaarannu”. Lisäksi he esittävät, että modernin liiketoiminta-alustan etuja ovat myös helppokäyttöisyys, joustavuus ja mahdollisuus saada yhdistettyä kaikki tarvittavat henkilöt saman toiminnanohjauksen äärelle. Teoksessa ei kuitenkaan tuoda esiin, millaisia strategioita, arkkitehtuurimalleja ja ohjelmistokehityksen malleja käyttämällä nykyaikai-

nen liiketoiminta-alusta toteutetaan. Aamuvuori ja Valtee eivät myöskään kerro enempää mainitsemistaan kohdennetuista näkymistä ja toiminnoista, joten jää epäselväksi, miten ne eroavat perinteisistä ERP-järjestelmien moduuleista.

2.3 Järjestelmien merkitys yrityksissä

Erilaiset liiketoimintatiedon hallinnan järjestelmät tarjoavat yrityksen johdolle arvokasta dataa yrityksen toiminnasta. Yrityksen johto voi tarvita ajankohtaista tietoa esimerkiksi yksittäiseen projektiin käytetyistä työtunneista, sen hetkisesestä myynnistä määrissä tai rahassa mitattuna tai henkilöstökuluista. Nykäsen ja muiden [2016] mukaan yritysten käytössä olevan informaation määrä ja informaatiotyypit kasvavat nopeasti, mihin yritysten on kyettävä vastaamaan. Kasvavan informaatiomäärän käsittely vaatii tehokkaita ja toimivia järjestelmiä.

Päätöksenteko yrityksissä mielletään usein rationaalisen valinnan teorian mukaan, jossa eri vaihtoehtoja arvioidaan saatavilla olevan informaation perusteella, ja paras vaihtoehto valitaan tämän arvioinnin perusteella [Shollo 2013]. Yritykset myös olettavat, että kaikkea tarjolla olevaa informaatiota käytetään päätöksenteossa [Nykänen ym. 2016]. Liiketoimintatiedon hallinnan järjestelmien yrityksen johdolle tarjoama reaaliaikainen ja tarkka informaatio tukee tällaista informaatiopohjaista päätöksentekoa. Shollo [2013] kuitenkin tarkentaa, että päätöksentekijät käyttävät liiketoimintatiedon hallinnan tuottamaa lopputulosta päätöksenteossa, eikä analyttistä prosessia, joka tuottaa lopputuloksen.

Shollo [2013] tunnisti myös neljä muuta tapaa käyttää liiketoimintatiedon hallinnan tuottamaa informaatiota: 1) Informaatiota saatetaan täydentää, jos se itsessään ei riittänyt vakuuttamaan muita päätöksentekijöitä. 2) Liiketoimintatiedon hallinnan tuottamaa tietoa saatettiin myös korvata täysin, jos sen ajateltiin sisältävän väärää tietoa tai jos se generoitiin poliittisia tarkoituksia varten. 3) Tieto saattoi olla monitulkintaista, jolloin päätöksentekijä saattoi tulkita tietoa omaa näkökantaansa tukevasti. 4) Tiedon tuottamisessa ja keruussa mukana olevat henkilöt saattoivat myös muokata tuotettavaa tietoa muuttamalla olettamuksia ja lähtökohtia, jolloin he pystyivät tuottamaan omia tarkoituksiaan tukevaa tietoa. Liiketoimintatiedon hallinnan järjestelmien tuottama tieto ei siis ole täysin riippumatonta, vaan siihen vaikuttavat sen parissa työskentelevät ihmiset, myös päätöksentekijät itse.

Elbashir ja muut [2008] tutkivat liiketoimintatiedon hallinnan järjestelmien vaikutuksia yrityksiin määrittämällä liiketoiminnan osa-alueet, asiakastietous, toimittajasuhteet ja sisäinen tehokkuus, joita parantamalla järjestelmien voi ajatella vaikuttavan positiivisesti yrityksen toimintaan. He huomasivat, että kaikki määritellyt osa-alueet korreloivat positiivisesti liiketoimintaan ja sitä kautta positiivisesti yrityksen toimintaan. Hyvällä liiketoimintatiedon hallinnan järjes-

telmällä voi siis olla hyvinkin positiivinen vaikutus yrityksen toimintaan, jos se kykenee parantamaan yrityksen toimintaa määritellyillä kolmella liiketoiminnan osa-alueella.

Järvinen ja muut [2016] huomasivat, että liiketoimintatiedon järjestelmiä otettiin käyttöön yrityksissä, koska niiden ajateltiin parantavan liiketoimintatietoutta, tehokkuutta ja päätöksentekoa. Lähes kaikissa heidän tutkimissaan yrityksissä yrityksen johto käytti säännöllisesti staattisia raportteja apunaan, mutta alle puolet loivat näitä raportteja itse järjestelmien avulla. Noin 60 prosenttia käyttivät muiden henkilöiden koostamia analyyseja. Järvisen ja muiden [2016] mukaan liiketoimintatiedon hallinnan järjestelmien tuottamat erilaiset raportit ja analyysit tukevat päätöksentekoa yrityksissä. Aamuvuoren ja Valteen [2017] mukaan järjestelmien suurin hyöty on kuitenkin se, että liiketoimintatieto on kaikkien saatavilla.

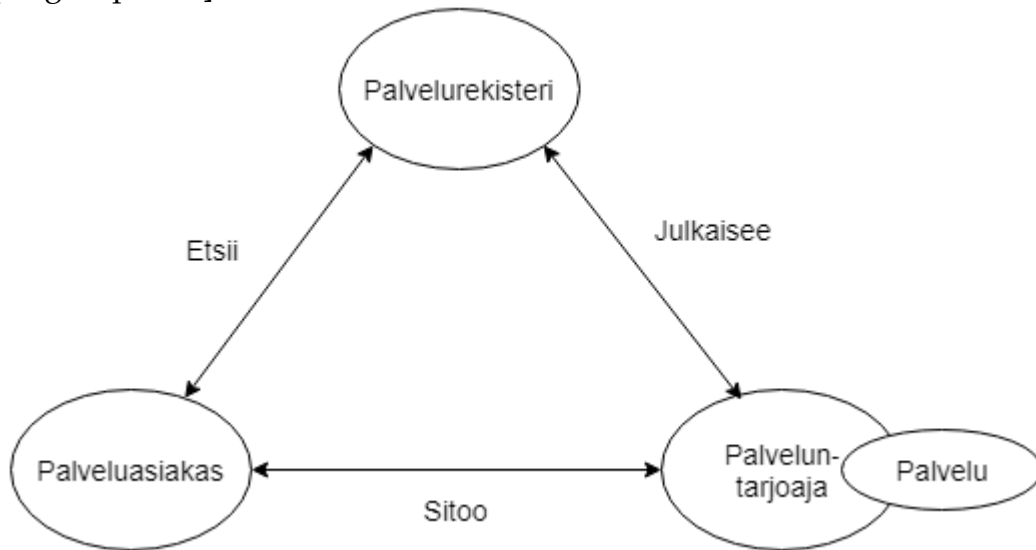
Valtee ja Aamuvuori [2017] esittelevät tulevaisuuden mahdollisuuksia ja kehityssuuntia toiminnanohjausjärjestelmille. Teoksen asiantuntijahaastatteluiden perusteella Aamuvuori ja Valtee toteavat, että toiminnanohjausjärjestelmät ovat nykyään ”liiketoiminnan selkäranka, jota täydennetään erilaisilla pilvipalveluilla ja ketterillä moduuleilla”. Tämän kehityksen perusteella he ennustavat, että olemme siirtymässä kohti hyvin integroituja liiketoiminta-alustoja ja pitkäaikaisia kehityskumppanuuksia ostajan ja toimittajan välillä. Lisäksi he odottavat pilvipalveluiden roolin kasvavan tulevaisuudessa myös liiketoiminta-alustojen toimitusprojekteissa. He uskovat myös, että järjestelmät, joilla voidaan simuloida ja mallintaa erilaisia tilanteita ja skenaarioita tulevat laajemmin yritysten käyttöön tulevaisuudessa.

Liiketoimintatiedon hallinnan järjestelmät nähdään siis yrityksissä tärkeässä asemassa. Niiden pohjalta tehdään päätöksiä ja seurataan yrityksen liiketoiminnan suuntaa ja kehittymistä. Yritykset voivat myös suunnitella toimintaansa niiden tarjoaman informaation perusteella. Voidaan olettaa, että tulevaisuudessa kaikki yritykset käyttävät jotain liiketoimintatiedon hallinnan järjestelmiä liiketoimintansa tukena. Enenevässä määrin uusilla tietojärjestelmillä pyritään myös muokkaamaan liiketoiminnan taustalla olevia ajattelu- ja toimintamalleja.

3. Palveluarkkitehtuuri

Palveluarkkitehtuuri, tai palvelukeskeinen arkkitehtuuri, on herättänyt mielenkiintoa viime vuosina niin työelämässä kuin akateemisessa maailmassa. Yhtenäistä määritelmää sille ei ole kuitenkaan löydetty. Esimerkiksi Laskey ja Laskey [2016] määrittelevät palveluarkkitehtuurin ajatusmalliksi, jolla voidaan ratkaista liiketoiminnan ongelmia, kun taas IBM:n mukaan palveluarkkitehtuuris-

sa otetaan käyttöön palveluita, eli tietoverkossa toimivia loogisia yksiköitä [IBM 2018]. Palveluarkkitehtuuria on luonnehdittu myös yksinkertaisesti tietojärjestelmien arkkitehtuurimalliksi ja tavaksi ajatella palvelukeskeisesti [Opengroup 2018].



Kaavio 1 Palveluarkkitehtuurin toimintaperiaate. Sommerville [2011]

Sommervillen [2011] määritelmässä (kaavio 1) palveluntarjoajat suunnittelevat ja ottavat käyttöön palveluita ja määrittelevät näiden palveluiden rajapinnat. He myös julkaisevat tietoa näistä palveluista saatavilla olevaan rekisteriin. Palveluasiakkaat (service clients), jotka haluavat käyttää palvelua löytävät kyseisen palvelun määritelmät ja palveluntarjoajan. He voivat sitten yhdistää sovelluksensa kyseisen palvelun kanssa ja kommunikoida sen kanssa käyttäen standardisoituja palveluprotokollia. [Sommerville 2011] Sommervillen määritelmässä palvelu tarkoittaa "uudelleenkäytettävää ohjelmistokomponenttia, joka kapseloi erillisiä toiminnallisuuksia, joita voidaan jaella eteenpäin ja joihin voidaan päästä käsiksi ohjelmoinnillisesti" ja web-palvelu tarkoittaa palvelua, johon päästään käsiksi käyttämällä standardisoituja Internet- ja XML-pohjaisia protokollia. Nämä määritelmät palveluarkkitehtuurille ja palvelulle pohjautuvat vahvasti teknologiselle ajattelulle, joka on laajempien määritelmien mukaan vain yksi osa palveluarkkitehtuuria.

Kun palveluarkkitehtuurista puhutaan kokonaisuutena, on vaikea rajoittaa näkökulmaa vain arkkitehtuuriin, sillä on väistämättä otettava huomioon myös muitakin tekijöitä, kuten yrityssuunnittelu (business design) ja toimitusprosessit (delivery process) [Microsoft 2018]. Palveluarkkitehtuuri on siis enemmän kuin pelkkä arkkitehtuurimalli – se kuvaa koko prosessia viitekehysineen aina suunnittelusta toteutukseen ja toimitukseen asti. Voidaankin todeta, että palveluarkkitehtuuri on arkkitehtuurimalli, joka toimii liiketoiminnan kontekstissa tietojärjestelmien suunnittelussa, toteutuksessa ja toimituksessa.

sa. Tutkielmassani käytän palveluarkkitehtuurista edellä olevaa laajaa määritelmää.

Laskey ja Laskey [2009] tähdentävät, että palveluarkkitehtuuri itsessään ei tarjoa ratkaisuja liiketoimintaongelmiin, jos niitä ongelmia ei voi ratkaista ilman palveluarkkitehtuuria. Heidän mukaan ainoa liiketoimintaongelma, johon palveluarkkitehtuuri voi suoraan tarjota ratkaisua on tietoteknisten palvelujen helpottaminen kuluttajien ja palveluntarjoajien välisen tehokkaan vuorovaikutuksen mahdollistamiseksi. Palveluarkkitehtuurin ei myöskään ole tarkoitus vastata omistuksen, laillisten ja sopimuksellisten velvollisuuksien, turvallisuuden ja luotettavuuden, tai muiden liiketoimintaan liittyvien aiheiden kysymyksiin. Laskey ja Laskey [2009] huomauttavat myös, että palveluarkkitehtuuri tarjoaa tapoja käyttää standardoituja mekanismeja saavuttaa piilossa olevaa kapasiteettia.

Ensimmäisen kerran palveluarkkitehtuuria terminä käyttivät Shulte ja Natis vuonna 1996 kuvaamaan monikerroksista tietojenkäsittelyä, joka auttaa yrityksiä jakamaan logiikkaa ja dataa useiden sovellusten ja käyttötapojen välillä. Ajatus IT-palveluiden tarjoamasta pääsystä IT-pohjaisiin yritystoimintoihin esitettiin kuitenkin jo muutamaa vuotta aiemmin vuonna 1991. Vuosikymmenen loppupuolella, vuonna 1998, The Network Applications Consortium laajensi käsitettä kattamaan myös yleisiä liiketoimintahuolia. He kehottivat yrityksiä siirtymään palvelukeskeiseen arkkitehtuurimalliin, jossa strategiset sovellukset on jaettu käyttäjäpalveluihin, liiketoimintapalveluihin, datapalveluihin ja perinteisiin palveluihin (legacy services). [Laskey and Laskey 2009].

Palveluarkkitehtuuri on todella alkanut kasvamaan ja leviämään 2000-luvun alussa palvelukeskeisen ajattelun myötä. 2010-luvulla on tehty tutkimuksia muun muassa *palveluarkkitehtuurihallinnon* (service-oriented architecture governance, SOAG) vaikutuksista IT-palveluiden joustamiseen ja palveluiden uudelleenkäyttöön [Joachim *et al.* 2013], palveluarkkitehtuurihallinnon arviointivälinekehityksestä [Hassanzadeh *et al.* 2011] sekä palveluarkkitehtuuripohjaisen yritysarkkitehtuurimallin (enterprise architecture, EA) kehittämisestä [Alwaidain *et al.* 2016].

Suomessa toteutettiin vuosina 2014-2017 valtiovarainministeriön toimesta kansallinen palveluarkkitehtuurihanke (KaPA), jonka tarkoituksena oli luoda yhteentoimiva digitaalisten palveluiden infrastruktuuri [Valtiovarainministeriö 2018]. Ohjelmassa luotiin ”kansallinen palveluväylä (tiedon välityskerros), kansalaisten, yritysten ja viranomaisten tarvitsemat yhteiset palvelunäkymät, uusi kansallinen sähköinen tunnistusratkaisu sekä kansalliset ratkaisut organisaatioiden ja luonnollisten henkilöiden roolien ja valtuutusten hallintaan” [Valtiovarainministeriö 2018]. Ohjelman tuloksena syntyi suomi.fi-verkkopalvelu, joka

tarjoaa sähköisen asioinnin tukipalveluita. Kansallisen palveluarkkitehtuurihankkeen yhden ulkopuolisen arvioijatahon mukaan ”ohjelma luo edellytyksiä hallinnon ja palvelutuotannon läpäisevälle digitalisoimiselle ja uudelle yhteistyön toimintakulttuurille”, mutta kritisoi hanketta siitä, että hanke ei ole tarjonnut tarpeeksi tukea palveluita käyttöönottaneille organisaatioille [Owalgroupp 2017]. Arviointiraportista käy myös ilmi, että projektin aikana yli puolet kunnista arvioivat Kansallisen palveluarkkitehtuuriprojektin hyödylliseksi niin yrityksille, yhteisöille ja kunnille kuin koko julkishallinnollekin, mutta projektin loputtua alle puolet kunnista pitivät sitä hyödyllisenä. Ainoastaan hyöty kansalaisille nähtiin suurena. Osaltaan tähän hyödyllisyyden arvioinnin laskuun saattaa vaikuttaa raportissa esiin tuotu tavoitteiden ja päämäärien selkeytyminen projektin edetessä. Kuntien oli helpompaa arvioida konkreettisia hyötyjä, kun päämäärät olivat realistisempia ja selkeämpiä. Projektin lähtökohta ja tavoitteet nähtiin kuitenkin positiivisina ja tavoiteltavina. Kansallisen palveluarkkitehtuurihankkeen pohjalta voidaan todeta, että kuten missä tahansa muussakin projektissa, selkeät ja realistiset tavoitteet jo projektin alussa sujuvoittavat projektin kulkua ja antavat realistisia odotuksia sidosryhmille. Projekti osoitti myös palveluarkkitehtuurin toimivuuden laajassa palveluiden ja tietojärjestelmien integroinnissa.

4. Palveluarkkitehtuurin soveltaminen liiketoimintatiedon hallinnan järjestelmissä

Tässä luvussa tutkin ensin, miten palveluarkkitehtuuria voisi teorian tasolla hyödyntää liiketoimintatiedon hallinnan järjestelmien suunnittelussa, minkä jälkeen esittelen esimerkin järjestelmästä, joka toteutettiin palveluarkkitehtuuria hyödyntäen. Luvun lopussa vertailen palveluarkkitehtuuria muihin yleisesti käytettyihin tietojärjestelmien arkkitehtuurimalleihin.

Palveluarkkitehtuurin pohjalle rakennetun järjestelmän tulee tarjota tarpeille ja kapasiteetille näkyvyyttä, sen on tarjottava kuluttajille ja palveluntarjoajille tavan vuorovaikuttaa, ja sen on tuotettava konkreettisia ratkaisuja kuluttajan liiketoimintaongelmiin [MacKenzie *et al.* 2006]. Kuten kaikkien muidenkin tietojärjestelmien, myös palveluarkkitehtuuripohjalta toteutettujen järjestelmien on hyvä noudattaa yleisiä tietojärjestelmien laatustandardeja.

Palveluarkkitehtuurin mahdollisiin hyötyihin lukeutuvat suurien liiketoimintajärjestelmien hallitun kasvun helpottaminen, suurien järjestelmien verkon yhteentoimivuuden mahdollistaminen ja organisaatioiden välisten aktiiviteettien kustannuksien pienentäminen. Palveluarkkitehtuurin kasvaminen todennäköisesti johtaa myös rakenteellisiin muutoksiin organisaatioissa, ja palveluarkkitehtuurin puolestapuhujien mukaan nämä muutokset johtavat kette-

rimpiin, skaalautuvampiin, kehittyviin ja hallittavampiin järjestelmiin ja organisaatioihin. [Laskey and Laskey 20009] Voidaan siis olettaa, että palveluarkkitehtuurin soveltaminen liiketoimintatiedon hallinnan järjestelmissä johtaisi positiivisiin muutoksiin yrityksissä sekä itse liiketoiminnan että liiketoimintatapojen kannalta.

4.1. Palveluarkkitehtuuria käyttäen toteutettu liiketoimintatiedon hallinnan järjestelmä

Hewlett-Packard esitteli vuonna 2007 kehittämänsä liiketoimintatiedon hallinnan järjestelmän, jonka toteuttamisessa he käyttivät palveluarkkitehtuuria [Wu *et al.* 2007]. Seuraavaksi esittelen heidän ratkaisunsa ja järjestelmän kehitysprosessin.

Wun ja muiden [2007] mukaan liiketoimintatiedon hallinnan järjestelmiä on perinteisesti kehitetty palvelinpuoli edellä, mutta liiketoimintatiedon hallinnan järjestelmien kehityksessä käytetyt arkkitehtuurimallit ovat alkanet muuttua kohti *hajautettuja monikerroksisia analyttisiä yritysjärjestelmiä* (distributed multi-tier enterprise analytic applications). Julkaisussaan he esittelevät palveluarkkitehtuurimallin liiketoimintatiedon hallinnan järjestelmille, joka mahdollistaa näiden hajautettujen järjestelmien saumattoman integraation yhdeksi liiketoimintatietoympäristöksi.

Tutkijoiden mukaan suurimmat haasteet järjestelmälle olisivat tehokas informaation toimitus ja teknologioiden integraatio. Liiketoimintatiedon hallinnan järjestelmien tulee toimittaa oikeaa tietoa oikeille käyttäjille oikeaan aikaan. Tiedon lähde voi tulla mistä tahansa integroimattomasta lähteestä, joten se pitää käsitellä ennen kuin se voidaan tehokkaasti toimittaa käyttäjälle. Teknologioiden integraatio -ongelmalle tutkijat esittivät ratkaisuksi palveluarkkitehtuuria. Heidän mukaan useat erilliset teknologiset ratkaisut liiketoimintatiedon hallinnalle johtavat muun muassa päällekkäisyyksiin, epä johdonmukaiseen informaatioon ja tietojärjestelmäratkaisujen kulujen kasvuun.

Wu ja muut [2007] esittelivät IT-palvelun johdon suorituskyvyn seurannan järjestelmän, jonka he aikoivat toteuttaa uudelleen palveluarkkitehtuuria hyödyntäen. Vanhasta järjestelmästä he tunnistivat viisi ongelmaa: 1) yksittäiset datavarastot (data marts) eri data lähteille, 2) ei yhtenäistä tapaa jakaa tietoa eri alustojen välillä, 3) joustamattomat ja suljetut omat datan keruu- ja käsittelyprosessit (ETL, extract, transform and load), joita olisi vaikea uudelleenkäyttää jaettavina palvelukeskeisinä komponentteina, 4) käyttäjäpuolen (front-end) toiminnallisuudet on toteutettu tiukasti yhden liiketoimintatiedon hallinnan järjestelmän kanssa toimiviksi, ja 5) yksinkertaisen työnkulun puute. Palvelu-

arkkitehtuurin soveltamisen uudessa järjestelmässä oli tarkoitus korjata edellä mainitut ongelmat.

Tutkijat aloittivat jakamalla ja muuttamalla vanhan järjestelmän komponentit uudelleenkäytettäviksi palvelukeskeisiksi komponenteiksi, jotka kykenisivät kommunikoidaan keskenään standardoitujen viestintäprotokollien avulla. Tämän ratkaisun hyödyiksi he luettelivat muun muassa integroidun ja yhdenmukaisen data-arkkitehtuurin, skaalautuvat ja joustavat datan keruu- ja käsittelyprosessit sekä uudelleenkäytettävät ja laajennettavat palvelut.

Uutta järjestelmää kehitettiin vaiheittain ketteriä kehitysmenetelmiä käyttäen. Kehitys aloitettiin keskitetystä datavarastosta (RDS, reporting data store), minkä jälkeen he toteuttivat datan keruu- ja käsittelyprosessit. Kolmannessa vaiheessa toteutettiin datavaraston julkaisupalvelut, jotka mahdollistavat datan käytön ulkoisissa järjestelmissä. Neljännen vaiheen tavoitteena oli IT-palveluiden suorituskyvyn indikaattoreiden arvioiminen datan perusteella ja niiden esittäminen helposti luettavalla tavalla. Viimeinen vaihe oli järjestelmän käyttöönotto.

Järjestelmän testauksen ja validoinnin perusteella Wu ja muut [2007] toteivat, että heidän palvelukeskeinen ratkaisunsa oli merkittävästi yksinkertaisempi ja halvempi kuin vastaavat muilla tavoin toteutetut järjestelmät. Järjestelmän saavuttama joustavuus teki kuitenkin siitä monimutkaisemman. Kaiken kaikkiaan heidän mukaan palveluarkkitehtuurin hyödyntäminen on paras tapa vähentää järjestelmän kehityksen ja ylläpidon kokonaiskustannuksia sekä minimoida riskit. Tämän esimerkkitapauksen pohjalta voidaan todeta, että palveluarkkitehtuurin soveltaminen liiketoimintatiedon hallinnan järjestelmien kehityksessä voi tulla kokonaiskustannuksiltaan halvemmaksi ja lopputuloksena syntyy pitkäikäinen, yksilöitävä ja käytettävyydeltään hyvä järjestelmä liiketoiminnan tarpeisiin.

4.2. Palveluarkkitehtuuri muihin arkkitehtuurimalleihin verrattuna

Vaikka palveluarkkitehtuurista on todistetusti hyötyä järjestelmien suunnittelussa ja toteutuksessa, se ei kuitenkaan ole ainoa arkkitehtuurimalli järjestelmäsuunnittelussa. Esittelen muutamia muita yleisesti käytettyjä arkkitehtuurimalleja ja vertailen niitä palveluarkkitehtuuriin. Sommervillen [2011] mukaan tietojärjestelmien arkkitehtuurisen suunnittelun voi jakaa kahteen tasoon: yksittäisten ohjelmien suunnitteluun ja monimutkaisten yritysjärjestelmien suunnitteluun. Sommerville [2011] korostaa myös, että tietojärjestelmien arkkitehtuurissa suunnittelussa on otettava huomioon tietojärjestelmälle asetetut vaatimukset ja niiden tärkeys, jotta voi löytää tietojärjestelmälle sopivimman arkkitehtuurimallin. Keskityn vertailemaan arkkitehtuurimalleja liiketoimintatiedon hallinnan järjestelmien näkökulmasta.

Kerrostettu arkkitehtuurimalli (layered architecture) tukee asteittaista ohjelmistokehitystä. Mallissa järjestelmän toiminnallisuus on jaettu erillisiin kerroksiin (esimerkiksi tietokannat ja sovellukset), ja jokainen kerros tarvitsee vain suoraan sen alla olevan kerroksen palveluita. Jos järjestelmän käyttöliittymää ei muuteta, kerroksia voi vaihtaa vastaaviin uusiin kerroksiin. Yksittäisen kerroksen rajapinnan muutos vaikuttaa ainoastaan suoraan sen ylä- tai alapuolella oleviin kerroksiin. Kerrostetun arkkitehtuurimallin etuja ovat kokonaisten kerrosten korvaaminen ja redundanttien toimintojen (esimerkiksi autentikaation) lisääminen jokaiseen kerrokseen, mikä parantaa järjestelmän luotettavuutta. Haittoja puolestaan ovat kerrosten selkeään rajaamisen vaikeus ja useiden kerrosten aiheuttamat järjestelmän suorituskykyongelmat. [Sommerville 2011] Tämä arkkitehtuurimalli sopii palveluarkkitehtuuria paremmin pienempien yksittäisten järjestelmien suunnitteluun. Toisin kuin palveluarkkitehtuurissa, sen painopiste on toiminnallisuudessa ja toteutettavuudessa, jolloin käytettävyys jää taka-alalle.

Säilöarkkitehtuuri (repository architecture) perustuu koko järjestelmän yhteiseen datan keskussäilöön. Kaikki järjestelmän data säilötään yhteiseen säilöön, joka on kaikkien järjestelmän komponenttien saatavilla. Komponentit kommunikoivat vain keskussäilön kautta. Arkkitehtuurimallin etuna on, että komponentit voivat olla täysin itsenäisiä. Lisäksi dataa voi hallinnoida yhdenmukaisesti, sillä se säilötään yhdessä paikassa. Datan säilöminen yhdessä paikassa on myös mallin heikkous, sillä jokin ongelma säilössä vaikuttaa koko järjestelmän toimivuuteen. [Sommerville 2011] Yrityksissä, joissa liiketoimintatiedon hallinnan järjestelmät ovat keskeisessä asemassa päätöksenteossa ja yrityksen toiminnassa, koko järjestelmän kaatuminen ja mahdollisesti arvokkaan datan menetys keskussäilön ongelman vuoksi voi tulla yrityksille kalliiksi.

Asiakas-palvelin -arkkitehtuurimalli (client-server architecture) on perusperiaatteeltaan hyvin lähellä palveluarkkitehtuuria. Tässä arkkitehtuurimallissa järjestelmän toiminnallisuus on jaettu palveluihin, jotka kaikki toimivat eri palvelimilla. Asiakkaat käyttävät näitä palveluja palvelinten kautta. Arkkitehtuurimallin huomattavin etu on palveluiden hajauttaminen tietoverkossa, jolloin eri toiminnot ovat kaikkien käyttäjien käytössä ja kaikkien palveluiden ei tarvitse ottaa kyseisiä toimintoja käyttöön. Palveluita ja palvelimia voidaan poistaa tai ottaa käyttöön ilman, että se vaikuttaa muihin palveluihin ja palvelimiin. Haittana ovat yksittäisten palvelimien alttius esimerkiksi palvelunestohyökkäyksille, mikä tekee järjestelmän suorituskyvystä ennalta-arvaamatonta, sekä mahdolliset ongelmat eri tahojen omistamien palvelinten vuoksi. [Sommerville 2011] Haitoistaan huolimatta tämä arkkitehtuurimalli voisi sopia hyvin palve-

luarkkitehtuurin pariksi itse järjestelmän arkkitehtuuriseen suunnitteluun palveluarkkitehtuurin ollessa koko tietojärjestelmäprojektin mallina.

Palveluarkkitehtuurille on vaihtoehtoja, mutta kuten Sommerville huomautti, tietojärjestelmäprojekteissa on valittava järjestelmän keskeisimpien vaatimusten mukainen arkkitehtuurimalli. Palveluarkkitehtuuri sopii usein liiketoimintatiedon hallinnan järjestelmien suunnitteluun ja toteutukseen, mutta muista arkkitehtuurimalleista ja -näköyksistä voi myös olla hyötyä. Palveluarkkitehtuurin suurin etu muihin arkkitehtuurimalleihin verrattuna on kuitenkin sen ulottuvuudet pelkän tietojärjestelmän arkkitehtuurin suunnittelun ulkopuolella, sillä se ottaa huomioon myös liiketoiminnan kontekstin ja koko tietojärjestelmäprojektin viitekehukset.

5. Yhteenveto

Liiketoimintatiedon hallinnan järjestelmien merkityksen kasvaessa yrityksissä niiden on kyettävä vastaamaan alati muuttuviin tarpeisiin ja mahdollistamaan yritysten kasvun. Järjestelmien on oltava helposti muokattavissa yritysten tarpeisiin, helppokäyttöisiä ja luotettavia, ja niiden on tuotettava relevanttia dataa johtoportaan lisäksi myös koko organisaatiolle. Palveluntuottajilla on paine tuottaa nopeasti ja tehokkaasti liiketoimintatiedon hallinnan järjestelmiä erilaisen ja erikokoisten yritysten käyttöön.

Palveluarkkitehtuuri on ottamassa jalansijaa tietojärjestelmien tuotannossa tehokkaana arkkitehtuurimallina. Yhtenäisen määritelmän puutteesta huolimatta se on levinnyt tietojärjestelmien tuottamisesta organisaatorakenteisiin ja johtamismalleihin. Kansallisen palveluarkkitehtuurihankkeen ja sen onnistumisen myötä se on saanut näkyvyyttä Suomessakin, mutta tietojärjestelmiä tuottavat yritykset ovat kuitenkin Suomessa hitaita ottamaan palveluarkkitehtuuria käyttöön järjestelmätuotannossa.

Palveluarkkitehtuuri vastaa yritysmaailman järjestelmätarpeisiin. Sen avulla voidaan tuottaa ketterästi ja kustannustehokkaasti eri yritysten tarpeisiin räätälöitävissä olevia liiketoimintatiedon hallinnan järjestelmiä, jotka kattavat yritysten kaikki järjestelmätarpeet monien irrallisten eri järjestelmien sijaan. Palveluarkkitehtuurin pohjalta toteutetut liiketoimintatiedon hallinnan järjestelmät voivat muuttaa yrityksen toimintatapoja, mutta joustavat silti yrityksen muuttuessa. Ne mahdollistavat ja tukevat yritysten kasvua.

Palveluarkkitehtuurin hyödyistä huolimatta nähtäväksi jää onko se tullut pysyäkseen, vai onko se vain yleisen palvelukeskeisen ajattelumallin sivutuote, joka syrjäytetään uuden ajattelumallin tullessa pinnalle. Sen tuomat hyödyt liiketoimintatiedon hallinnan järjestelmille voivat kuitenkin parhaassa tapauk-

sessä kantaa pitkälle uusien järjestelmien mahdollistaessa yritysten kasvun ja muuntautumisen nopeasti muuttuvassa maailmassa.

Viiteluettelo

- Riina Aamuvuori ja Minna Valtee. 2017. *Toiminnanohjauksen onnena ja tuonena. ERP:stä liiketoiminta-alustaan – kokeneet konkarit kertovat*. Oscar Software Oy.
- Ayed Alwadain, Erwin Fieft, Axel Korthis and Michael Rosemann. 2016. Empirical insights into the development of a service-oriented enterprise architecture. *Data Knowledge Engineering* 105, 39-52.
- Alireza Hassanzadeh, Leila Namdarian and Sha'ban Elahi. 2011. Developing a framework for evaluating service oriented architecture governance (SOAG). *Knowledge-based Systems* 24, 716-730.
- IBM. 2018. Service-oriented architecture (SOA) https://www.ibm.com/support/knowledgecenter/en/SSMQ79_9.5.1/comm.ibm.egl.pg.doc/topics/pegl_serv_overview.html. Checked 29.3.2018
- Nils Joachim, Daniel Beimborn and Tim Weitzel. 2013. The influence of SOA governance mechanisms on IT flexibility and service reuse. *Journal of Strategic Information Systems* 22, 86-101.
- Kathryn B. Laskey and Kenneth Laskey. 2009. Service oriented architecture. *Wires Computational Statistics* 1, 101-105.
- Microsoft. 2018. Understanding service-oriented architecture. https://msdn.microsoft.com/en-us/library/aa480021.aspx#aj1soa_topic2. Checked 29.3.2018
- Erno Nykänen, Marko Järvenpää and Henri Teittinen. 2016. Business intelligence in decision making in Finnish enterprises. *Nordic Journal of Business*, 65, 2, 24-44.
- Opengroup. 2018. Service-oriented architecture - What is SOA? http://www.opengroup.org/soa/source-book/soa/p1.htm#soa_definition. Checked 22.3.2018
- Ian Sommerville. 2011. *Software Engineering*. Addison-Wesley.
- Valtiovarainministeriö. 2018. Kansallinen palveluarkkitehtuuri. <http://vm.fi/palveluarkkitehtuuri>. Luettu 4.4.2018
- Wikipedia. 2018. Enterprise resource planning. https://en.wikipedia.org/wiki/Enterprise_resource_planning. Checked 2.5.2018
- Liyua Wu, Gilad Barash, and Claudio Bartolini. 2007. A service-oriented architecture for business intelligence. In: *Proc. of the International Conference on Service-Oriented Computing and Applications(SOCA'07)*.

Tietojärjestelmien eettiset ongelmat

Johanna Sipinen

Tiivistelmä.

Tämän kirjallisuuskatsauksen aiheena on tietojärjestelmiin liittyvät eettiset haasteet. Aihetta lähestytään ensisijaisesti vastuun ja yksityisyyden näkökulmista. Tietojärjestelmien etiikka on melko vähän tutkittu aihe, vaikka erilaiset järjestelmät ovat osa melkein jokaisen arkipäivää tänä päivänä. Tietojärjestelmien suunnittelussa eettisyys ei välttämättä ole korkealle priorisoitu kriteeri onnistunutta suunnittelua arvioitaessa. Tietojärjestelmien eettisyyttä pohdittaessa vastuun kysymykset nousevat helposti esiin, koska yhden yksittäisen vastuutahon määrittäminen on usein vaikeaa. Selkeästi rajatun vastuutahon puuttuminen tekee myös tietojärjestelmien kehittämisestä eettisessä mielessä haasteellisen tehtävän. Yksityisyyden ja tietosuojan huomioiminen asettavat omat haasteensa myös, koska eettiset ongelmat järjestelmissä liittyvät usein arkaluontoiseen ja salassapidettävään tietoon. Mitä enemmän vastuun ja yksityisyyden eettisiä kysymyksiä huomioidaan järjestelmän kehittämisessä, sitä parempia järjestelmiä on mahdollista myös rakentaa. Tämän kirjallisuuskatsauksen tärkein löydös on tietojärjestelmien keskeisten eettisten ongelmien tunnistaminen. Tutkielman rajauksen sisällä keskeisimpiä ongelmia ovat vastuuseen ja yksityisyyteen liittyvät haasteet. Koska eettiset teoriat tarjoavat harvoin valmiita vastauksia, vaativat ne pohdintaa ja oman toiminnan vaikutusten tunnistamista. Etiikkaa ja eettisyyttä on mahdollista käyttää oman toiminnan tukena, antamaan vahvistusta sille mitä itse pitää oikeana. Toisaalta myös enemmistön demokratia voi olla toimiva ratkaisu monissa eettisesti ongelmallisissa tilanteissa. Toinen keskeinen löydös on lainsäädännön tarkempi huomioiminen kehittämistyössä. EU:n yleinen tietosuoja-asetus voi olla askel kohti parempaa yksityisyydensuojaa tulevaisuudessa. Asetus itsessään ei välttämättä tuo suuria muutoksia jo olemassa oleviin henkilötietoja sääteleviin lakeihin, mutta se voi tuoda muutoksia yksityisyydensuojan lainsäädännön toteutumisen seurantaan.

Asiasanat: Etiikka, tietojärjestelmä, vastuu, yksityisyys

1. Johdanto

Tämän tutkielman tarkoituksena on selvittää, millaisia eettisiä ongelmia tietojärjestelmiin liittyy. Tutkielma on toteutettu kirjallisuuskatsauksena. Aihetta lähestytään käsittelemällä eettisyyttä ja tietojärjestelmiä ensin omina kokonaisuuksinaan, jonka jälkeen aihetta käsitellään molemmista näkökulmista yhdessä.

Etiikka on filosofian alue, joka käsittää eettisen toiminnan periaatteita oikeasta ja väärästä sekä näkemyksiä siitä, mikä on moraalisesti tai eettisesti oikein [Thiroux and Krasemann 2012.] Eettisyys on laajentunut useille alueille, kuten liiketoiminnan ja teknologian etiikkaan, terveydenhuollon etiikkaan sekä kirjanpidon etiikkaan. Erilaiset kansalliset ja maailmanlaajuiset kriisit ovat lisänneet tarvetta yritysten eettisen toiminnan tarkistamiseen sekä toimintamenetelmien muuttamiseen. Yritysten toimintaan liittyvät kiinteästi myös niissä käytettävät tietojärjestelmät sekä niitä käyttävät ihmiset. Yrityksillä alkaa olla käytössä eettisiä ohjeita henkilöstölle, koskien järjestelmissä olevien tietojen käyttämistä. [Weiss 2017.] Monissa sosiaalialan ja terveydenhuollon järjestelmissä käsitellään arkaluontoisia tietoja, joiden käsittely turvallisesti on ensiarvoisen tärkeää. Enenevässä määrin organisaatioissa joudutaan kiinnittämään huomiota henkilötietojen yksityisyydensuojaan ja tietosuojaan. Yksityisyyteen liittyviä kysymyksiä käsitellään enemmän tutkielman viimeisissä luvuissa.

Tietojärjestelmien eettisyyteen liittyy myös tietoturvallisuuden kuuluvia kysymyksiä, mutta ne eivät ole tässä tutkielmassa varsinaisessa pääosassa. Tutkielman tavoitteena on saada tietoa siitä, millaisia erilaisia eettisiä asioita tietojärjestelmiin liittyy. Tavoitteiden saavuttamiseksi tarkastelen lähemmin seuraavia kysymyksiä:

- 1) Mitä eettisiä kysymyksiä tietojärjestelmiin liittyy?
- 2) Miksi tietojärjestelmien eettisyys on merkityksellistä?

Tutkielma jakautuu kuuteen lukuun. Johdannon jälkeen luvussa 2 esitän lyhyesti tietojärjestelmän käsitettä ja siihen liittyviä määritelmiä. Luvussa 3 käsittelen etiikan alaan liittyviä perusteita. Luvussa 4 tarkastelen tietojärjestelmiin liittyviä eettistä vastuuta käsitteleviä kysymyksiä. Luvussa 5 syvennän aihetta ja pohdin yksityisyyden ongelmaa eri näkökulmista. Viimeisessä luvussa teen yhteenvetoa tutkielman aiheesta ja käsittelen aiheen esiin nostamia ajatuksia ja pohdintoja.

2. Tietojärjestelmä

Tässä luvussa on tarkoitus pohtia lyhyesti tietojärjestelmää eri näkökulmista. Näkökulman ei ole tarkoitus olla erityisen tekninen, vaan enemmän yhteiskunnallinen. Asiaa lähestytään siitä näkökulmasta, että tietojärjestelmän on tarkoitus tuottaa jotakin hyötyä käyttäjälleen, eikä olla pelkästään syöte-tuloste -systemi tiedon prosessointia varten. Käytännöllisintä on aloittaa tietojärjestelmän määrittelystä.

Tietojärjestelmälle löytyy useita eri määritelmiä. Arduinin ja muiden [2015, 3] mukaan tietojärjestelmä voidaan määritellä toiminnoiksi, joiden avulla etsitään, lisätään, tallennetaan ja käsitellään tietoa. Tietojärjestelmän pääasiallisena tarkoituksena on tuottaa tietoa yrityksen henkilöstölle, mikä helpottaa yrityksessä tehtävää päätöksentekoa. Järjestelmän tarkoitus on siis tuottaa prosessoitua tietoa, joka helpottaa päätöksentekoa ja toimintojen koordinoitua. Tietojärjestelmissä tietty tehtävä suoritetaan tietystä ympäristössä, minkä vuoksi ympäristön vaikutukset on myös hyvä ottaa huomioon. [Arduin *et al.* 2015, 3.] Tärkeä asia tietojärjestelmää määriteltäessä on miettiä, sisällytetäänkö käyttäjät osaksi tietojärjestelmää. Esimerkiksi Alterin [1999, 2] mukaan tämä on tilannesidonnaista ja pitää päättää analysoitavasta ongelmasta riippuen joka kerta erikseen. Tyypillisesti esimerkiksi kassajärjestelmää käyttävät asiakaspalvelijat kannattaa sijoittaa osaksi järjestelmää, koska he käyttävät järjestelmää tiedon prosessoimiseen ennalta määrättyllä tavalla. Sen sijaan esimiehet, jotka saavat tietonsa hallinnollisista järjestelmistä, kuuluvat tietojärjestelmän ulkopuolelle, koska he käyttävät niitä esimiestyöhön liittyviin tehtäviin. [Alter 1999, 2.] Toisaalta tässä kohtaa Alterin näkemystä voi hieman kyseenalaistaa, koska esimiehet voivat tehdä hyvinkin ratkaisevia päätöksiä perustuen tietojärjestelmän tuottamiin tietoihin.

Tietojärjestelmä voi olla manuaalinen tai tietokoneistettu [Stair and Reynolds 2014]. Tietojärjestelmällä on myös aina jokin tarkoitus [Arduin *et al.* 2015]. Stairin ja Reynoldsin [2014] mukaan tietojärjestelmän tarkoituksena on helpottaa yrityksen pyrkimyksiä parantaa liiketoimintaansa. Tietojärjestelmän ja sen tuottaman tiedon arvo on suoraan yhteydessä siihen, miten järjestelmä auttaa päätöksentekijöitä saavuttamaan organisaation asettamat tavoitteet. Tietojärjestelmä on käyttöönotettava huolellisesti, jotta yhteiskunta ja liiketoiminta voivat saavuttaa järjestelmien tarjoamat parhaat mahdolliset hyödyt.

Tietojärjestelmän tuottama tieto on olennaista menestyvien ja kilpailukykyisten yritysten toiminnassa. Tietojärjestelmän tieto lisää liiketoiminnallista arvoa sekä tuottaa asiakkaille heidän tarvitsemiaan tuotteita

ja palveluja. Itse asiassa, monilla aloilla yrityksen olemassaolo ja selviytyminen voi olla mahdotonta ilman kattavaa IT-järjestelmää. [Laudon and Laudon 2007.]

3. Etiikka pähkinänkuoressa

Tässä luvussa pohditaan lyhyesti mitä on etiikka ja millaista on eettinen toiminta. Aihetta lähestytään ensisijaisesti Spinozan näkemysten pohjalta. Spinoza on sanonut teoksessaan *Etiikka*, että järjen mukaan elävä ihminen on vapaa ihminen [Spinoza 1996]. Tällä Spinoza tarkoittaa, että ihminen on enemmän aktiivinen kuin passiivinen kaikessa sellaisessa toiminnassa, joka häntä kiinnostaa. Vapaus on siis tietoisuutta välttämättömyydestä, jonka voimme saavuttaa nähdessämme maailman *sub specie aeternitatis* (universaali perspektiivi), sekä itsemme sidottuna maailman muuttumattomiin lakeihin. Spinozan mukaan vapaa ihminen pyrkii hyvään sekä välttämään pahaa, vastustamaan vaaroja ja toimimaan aina rehdisti. Spinoza ei kuitenkaan tarkoita vapaudella vapaata tahtoa, vaan ihmisen kykyä tehdä työtä rauhassa sekä ilmaista mielipiteitä ja pyrkiä päämääriin, joita järki heille esittää. [Scruton 1998.]

Filosofisessa mielessä etiikkaa voidaan pitää yksilön tapana etsiä oikeellisuutta elämäänsä. Oikeellisuuden näkökulmasta jokainen eettisin perustein tehty päätös voi viedä yksilöä lähemmäksi oikeellista ja autenttista elämää. Eettisyyttä voidaankin pitää olemisen konkreettisimpana ja käytännöllisimpänä toimintana yksilön ammatillisessa ja henkilökohtaisessa todellisuudessa. Jokainen teko tai tekemättä jättäminen on siten eettinen valinta. Pelkkä tietoisuus muista ihmisistä ja ympäröivästä maailmasta on eettinen asema, johon ihmisen käyttäytyminen ja tietoisuus rakentuvat. [Boldizar and Korhonen 1999, 288.]

Eettisyys ei kuitenkaan ole synonyymi sanalle 'hyveellisyys'. Eettisen päätöksen tekeminen ei automaattisesti tarkoita oikean päätöksen tekemistä, vaan se tarkoittaa sitoutumista yksilön eettiseen pystyvyyteen. Esimerkiksi, jos yksilö päättää heittää myrkyllistä jätettä Itämereen, se on eettinen valinta. Yksilö saattaa kokea teon oikeaksi tai vääräksi, mutta kokee silti tekevänsä jotakin jollekin muulle. Hyveet sen sijaan ovat oikeamielisiä katsottuja ratkaisuja, ja hyveitä käsittelee hyve-etiikka. [Boldizar and Korhonen 1999.]

Hyve-etiikassa käsitellään kysymyksiä siitä, mikä on hyveellistä ja moraalisesti hyvää, eli millaista on hyveellinen toiminta. Voidakseen olla hyveellinen, tulee yksilön tähdätä kohti hyveellistä toimintaa. Tämä määrittää yksilön moraalisuuden. [Kulczak 1999.] Airaksisen ja Frimanin [2008] mukaan perushyveitä ovat viisaus, rohkeus, kohtuullisuus ja oikeamielisyys. Viisaus on

kykyä ymmärtää tekojensa seuraukset, rohkeus taas on uskallusta ilmaista mielipiteensä suoraan ja reilusti. Kohtuullisuudella tarkoitetaan keskitien kulkemista elämässä. Oikeamielinen henkilö haluaa antaa toiselle sen, mikä hänelle kuuluu. [Airaksinen ja Friman 2008.]

Hyve-etiikkaan liittyy läheisesti deontologia, joka on etiikan alan tutkimusta oikeasta ja väärästä, pakollisuudesta ja vapaaehtoisuudesta sekä kielletyistä asioista. Voidakseen olla hyvä tai ihailtava, on omattava ominaisuuksia joita voi vertailla, kuten hyvä tai paras sekä enemmän tai vähemmän ihailtava. Tämä skaala on olemassa hyve-etiikassa, mutta deontologiassa skaalaa ei ole – ei voida sanoa jonkin olevan oikea, oikeampi, oikein. Tämän vuoksi deontologia vertautuu pikemminkin lakiin ja lainopillisiin päätöksiin. [Carr and Steutel 2005.]

4. Tietojärjestelmiin liittyvät eettiset vastuukysymykset

Ihmiset eivät ole aina eettisiä. Jos olisimme aina eettisiä, ei eettisistä aiheista tarvitsisi edes keskustella. Me aina välillä teemme asioita, jotka eivät ole oikein tai edes oikeudenmukaisia. Emme aina toimi ihmiskunnan tai edes itsemme parhaaksi. Voiko siis eettisyyttä ja eettisyyden opiskelusta pitää merkityksellisenä? Sivistyneet yhteiskunnat tarvitsevat tapoja arvioida jokapäiväisiä toimiaan. Etiikka tarjoaa tavan pohtia sitä, mitä ihmisten pitäisi tehdä ja kuinka heidän tulisi käyttäytyä. Informaatioteknologia ja tietojärjestelmät vain lisäävät osaltaan pohdinnan monimutkaisuutta. Informaatioteknologiaan ja tietojärjestelmiin liittyvä ongelma on kaksijakoinen. Ensinnäkin informaatioteknologialla on ominaisuuksia, jotka sallivat sen käyttäjiä toimimaan tavalla, joka vaikuttaa muihin henkilöihin. Joskus tämä on tahatonta, kuten käyttäjien tietojen tallentamista ja niiden käyttämistä, mutta joskus kehitetään myös tietojärjestelmiä jotka voivat olla vahingollisia käyttäjilleen. [Loch *et al.* 1988, 653.] Tietojärjestelmiin voidaankin katsoa liittyvän paljon erilaisia eettisiä kysymyksiä. Seuraavaksi käsitellään eettistä vastuuta tietojärjestelmien toiminnasta sekä yksilön omaa vastuuta tietojärjestelmän käyttäjänä.

4.1. Kenellä on eettinen vastuu tietojärjestelmästä

Tietojärjestelmien kehittämiseen liittyy usein kysymys vastuusta. Kuka on vastuussa virheellisesti toimivista järjestelmistä? Heimo ja muut [2013] mainitsevat esimerkkinä tapauksen, jossa toimimaton sädehoitoa antava kone tappoi kuusi ihmistä. He toteavat artikkelissaan, että vastuullinen taho tulisi

olla joko tilaaja tai toimittaja. Tämä näkemys perustetaan filosofi John Rawlsin [1997] näkemykseen heikompien puolustamisesta. Toisaalta artikkelissa mainitaan myös Immanuel Kantin näkemys, jonka mukaan vastuuta on mahdotonta laittaa yhdelle ihmiselle - tai edes ryhmälle ihmisiä - joka voisi 'maksaa takaisin' tapahtuneen virheen. [Heimo *ym.* 2013.]

Voimme siis alustavasti todeta, että vastuullisia voivat olla joko järjestelmän tilaaja tai toimittaja. Toisaalta on myös hyvä miettiä, missä määrin järjestelmien kehittäjät ovat eettisessä vastuussa järjestelmän toiminnasta. Ajattelevatko kehittäjät tätä asiaa tai ovatko he tietoisia tietojärjestelmiin liittyvistä eettisistä kysymyksistä esimerkiksi koulutuksen kautta. Vartiainen [2011] mukaan tietojärjestelmien kehitystyössä jätetään usein huomiotta sosiaaliset, poliittiset ja psykologiset kysymykset. Tietoteknisellä alalla työskentelevien ihmisten ongelmana pidetään usein inhimillisten tekijöiden ohittamista taloudellisten ja teknisten tekijöiden kustannuksella tietojärjestelmien kehitystyössä. [Vartiainen 2011.] Itse asiassa Stoodleyn [2009, 1-2] mukaan IT-alan ammattilaiset havahtuivat aikanaan itse ensimmäisenä eettisen toiminnan merkityksellisyyteen. Esimerkkinä Stoodley mainitsee kansainvälisen kyselyn, jonka mukaan kyselyyn osallistuneista järjestelmäammattilaisista 90 prosenttia oli sitä mieltä, että organisaatioiden tulisi asettaa ammatillisen eettisyyden standardit. 1960-luvulla puhuttiin myös Stoodleyn [2009, 1-2] mukaan yleisesti, että kun työntekijät astuivat tietokonekeskukseen sisään, niin "eettisyys lensi samaan aikaan ikkunasta ulos". Tämän vuoksi IT-ympäristössä toimiminen vaatiikin monenlaisia taitoja: ammattilaisen on oltava toimintatavoiltaan vastuullinen sekä kykenevä ratkaisemaan ei-teknisiä eettisiä kysymyksiä ja kehittämään proaktiivisesti sellaisia taitoja, joiden avulla on mahdollista vähentää eettisten ongelmien ilmenemistä.

Kolmas eettisessä mielessä pohdintaa aiheuttava ryhmä ovat järjestelmien käyttäjät. Missä määrin eettisiä haasteita aiheuttavat järjestelmien tarjoamat toiminnallisuudet sekä käyttäjät, jotka käyttävät järjestelmän toiminnallisuuksia. Heimo ja muut [2015] pohtivat artikkelissaan peruskouluoppilaiden Wilma-tietojärjestelmää eettisestä näkökulmasta. Artikkelissa kirjoitetaan muun muassa opettajan mahdollisuudesta käyttää Wilma-järjestelmää välineenä oppilaan rankaisemiseen, mikäli ei tule toimeen oppilaan kanssa. Wilmasta puhutaankin lasten "rikosrekisterinä". Jotkut vanhemmat kieltäytyvät käyttämästä Wilmaa, koska kokevat sen olevan kuin vahtikoira. Vanhemmat toivovat palautteen antamisen tapahtuvan henkilökohtaisesti eikä persoonattoman tietojärjestelmän kautta. Edellä mainittujen ongelmien lisäksi myös järjestelmän yksityisyyden ongelmat

huolettavat vanhempia. [Heimo *ym.* 2015.] Kenellä on vastuu tässä tapauksessa tietojärjestelmän eettisestä toiminnasta? Järjestelmän toimittajalla, tilaajalla vai käyttäjillä? Järjestelmän tilaaja on halunnut järjestelmään tietynlaisia toiminnallisuuksia, mutta käyttäjät valitsevat ainakin joissain tilanteissa itse tapansa hyödyntää niitä. Voiko tietojärjestelmä ajaa käyttäjänsä toimimaan oman etiikkansa vastaisesti? Voiko järjestelmä esimerkiksi pakottaa tekemään jollakin lomakkeella valinnan, jonka vaihtoehdot käyttäjä kokee oman etiikkansa vastaiseksi? Ihminen ei ehkä aina pysty olemaan Spinozan [1998] näkemyksen mukainen eettinen ja vapaa ihminen käyttäessään tietojärjestelmää.

Tietojärjestelmän tarjoaman tiedon oikeellisuuteen liittyy myös eettisiä vastuukysymyksiä. Todennäköisesti kenelläkään ei ole tyhjentävää näkemystä Wilmaan kirjatun tiedon oikeellisuudesta. Yleensä ajatellaan kuitenkin, että enemmistö on oikeassa. Tällaiseen 'enemmistön demokratia' -ajatteluun perustuvat esimerkiksi Wikipedian artikkelit: tieto hyväksytään oikeellisena, jos suurin osa ihmisistä on samaa mieltä. Toisaalta suurin osa ihmisistä voi olla myös pahasti väärässä. Joukkoistaminen ei takaa tiedon oikeellisuutta eikä tiedon verifiointikaan ole aina helppoa. [McBride 2014, 27.] Vastuu tiedon oikeellisuudesta voi olla vielä isompi ongelma sellaisissa paikoissa ja järjestelmissä, joissa virheen tekijä on mahdollista paikantaa, ja virheen seuraukset voivat olla vakavia. Esimerkkinä tällaisista voidaan mainita virheet pankin järjestelmissä tai virheet sään ennustamiseen käytettävissä järjestelmissä. [McBride 2014, 27.] Pankin tietojärjestelmissä virhe tiedon oikeellisuudessa voi olla erityisen vakavaa, koska pankeissa käsitellään suuria rahamääriä. Mason [1986] mainitsee tekstissään Lous Marchin tapauksen, jossa maahanmuuttajataustainen Louis oli tunnollisesti vuosien ajan maksanut asuntolainansa lyhennyksiä, mutta kerran pankin tietojärjestelmässä tapahtui virhe. Louis oli maksanut kuukausittaisen lyhennyksensä, mutta tietojärjestelmän mukaan lyhennystä ei ollut maksettu. Sama tapahtui myös seuraavana kuukautena. Lopulta tilanne johti monien mutkien jälkeen siihen, että Louis löysi lakimiehen, joka suostui haastamaan pankin oikeuteen. Ainoa asia, mitä pankki pystyi sanomaan todistenaan, oli, että "tietokoneet tekevät virheitä. Pankit tekevät virheitä myös." [Mason 1986.]

Vastuukysymyksiin liittyy usein myös vastuun välttely. Weckert [2013] esittää kaksi suurinta syytä vastuun välttelylle. Ensimmäinen syy on se, että vastuuta katsotaan liian kapeasta näkökulmasta. Vastuu nähdään vain tiettyjen määrittelyjen noudattamisena, eikä vastuuta katsota laajemmasta näkökulmasta - eli mitä seurauksia kehitetyn järjestelmän toiminnalla voi olla. Toinen syy on se, että vastuuta harvoin pystytään osoittamaan vain yhdelle

henkilölle kuuluvaksi, joten silloin kukaan ei käytännössä ole vastuussa tietojärjestelmän ongelmista. Tämän vuoksi tietojärjestelmien kehittämistä tuleekin katsoa laajemmasta näkökulmasta, jotta järjestelmäkehittäjät voivat kutsua itseään todellisiksi ammattilaisiksi. Tietojärjestelmäalan ammattilaisilla ei ole varsinaisia kirjoitettuja vaatimuksia toimia hyveellisesti, mutta yksilöinä heillä voidaan katsoa olevan sama vastuu kuin muillakin yhteiskunnan jäsenillä. [Weckert 2013.]

4.2 Yksilön ja organisaation vastuu

Edellisessä kohdassa sivuttiin yksilön ja organisaation vastuuta. Tässä kohdassa on tarkoitus syventää yksilön ja organisaation vastuuseen liittyviä erilaisia näkökohtia sekä pohtia, mitkä tekijät voivat vaikuttaa yksilön toiminnan eettisyyteen.

Akman ja Mishran [2009, 1251] mukaan tietoisuus on yksi viidestä merkittävimmästä eettisestä kysymyksestä informaatioyhteiskunnassa. Tietoisuudella on suuri merkitys silloin, kun puhutaan eettisistä ja moraalisisista asenteista tietojärjestelmien käyttämisessä. Akhman ja Mishran [2009, 1251] mukaan organisaatiot kohtaavat usein tietojärjestelmiin kohdistuvia turvallisuusrikkomuksia, jotka on tehty tarkoituksella tai tahattomasti. Tämän vuoksi organisaatioiden tulisikin tarjota työntekijöilleen hyvin organisoituja koulutusohjelmia eettisissä kysymyksissä. IT-etiikassa ja siihen liittyvässä lainsäädännössä koulutus usein vähentää ongelmien vakavuutta. Ainakin Tomein [2010] mukaan opiskelijoille suunnatun eettisen koulutuksen tulokset olivat pääosin hyviä: koulutuksella nähtiin olevan positiivinen vaikutus opiskelijoiden eettiseen päätöksentekoon ja käytökseen. Tutkimuksen mukaan koulutuksen suorittamisen jälkeen todennäköisyys työnantajan yksityisyyden loukkaamisesta pieneni hieman, ja yksilöiden toimintaa ohjasivat erilaiset kriteerit kuin aikaisemmin. Vaikka tiettyjä toimenpiteitä ei edes pidetty "laittomina", oli koulutuksen jälkeen vähemmän todennäköistä, että työntekijät tekivät mitään sellaista, mikä mahdollisesti loukkaisi työnantajan yksityisyyttä. Ozin [1992, 425] mukaan tätä kutsutaan työetiikaksi eli työntekijä suojelee työnantajan intressejä. Saman näkemyksen jakaa myös Grant [2010], jonka mukaan koulutuksella on positiivinen vaikutus yksilön moraaliseen kehittymiseen. Valitettavasti Ozin ja Grantin näkemystä kohti on olemassa myös ainakin yksi vastakkainen näkemys, josta Jewels ja Evans [2005, 386] kirjoittavat artikkelissaan. Heidän mukaansa moraalialia ei voi opettaa ihmiselle, koska arvot ovat niin syvälle juurtunut asia ihmisessä. Totuus lienee jossain näiden kahden näkemyksen välissä.

Ozin [1992, 425] mukaan ammattilaisen tulee aina ajatella yhteisön hyvinvointia toimiessaan työroolissaan. Tietojärjestelmillä on suuri vaikutus yhteisön turvallisuuteen, yksityisyyteen ja taloudellisiin intresseihin. Tiettyjen eettisten teorioiden mukaan voidaan sanoa, että jos etujen välillä on ristiriitaa, valitaan se joka tuottaa hyvää mahdollisimman suurelle joukolle ihmisiä. Tällä tarkoitetaan käytännössä sitä, että suuren joukon edun tulee mennä pienen joukon edun edelle. Dhariwalin ja muiden [2010] artikkelissa samaa asiaa kutsutaan termillä *prima facie* -moraalisesti hyvää on sellainen, joka täyttää hyväksyttävyyden kriteerit rationaalisten toimijoiden edustaman suuren yhteisön mielestä. IT-ammattilaisen moraaliseen vastuuseen kuuluu myös Ozin [1992, 425] mukaan tiedon yksityisyyden ja turvallisuuden suojeleminen, lakien noudattaminen, muiden henkilöiden hämäyksen välttäminen sekä tietojärjestelmiin liittyvien tietojen kouluttaminen muulle yhteisölle.

Bellin ja Adamin [2004, 165] ajatuksena on, että suuri osa IT-ammattilaisten eettisestä toiminnasta perustuu hiljaiseen tietoon, ei niinkään kirjoitettuihin ohjeisiin tai tiukkaan valvottuun toimintaan. Hiljaisella tiedolla tarkoitetaan tietoa, jota on vaikea muotoilla tai kuvata sanoilla. Hiljainen tieto perustuu toimintaan tietyssä kontekstissa. [Nonaka 2000.] Bellin ja Adamin [2004] mukaan IT-ammattilaisten eettisessä toiminnassa ei ole kyse tiukasti seurattavista käytösohjeista, vaan pikemminkin siitä, että toiminta perustuu ajan kuluessa kehittyneisiin sosiaalisiin sopimuksiin. Nämä sosiaaliset sopimukset taas perustuvat sellaisiin eettisiin ohjeisiin, jotka ammattilaiset ovat itse kehittäneet ajan kuluessa. [Bell ja Adam 2006.]

Nonakan [2000] mukaan hiljaisessa tiedossa voidaan erottaa kahdenlaisia elementtejä: kognitiivisia ja teknisiä. Kognitiivinen elementti perustuu mentaalimalleihin, joissa ihminen luo ja manipuloi analogioita mielessään. Tekninen elementti taas käsittää konkreettisen tietotaidon, kuten tietynlaiset kädentaidot tai vaikka pyörällä ajamisen. IT-ammattilaisen hiljainen tieto saattaa perustua tällaisille kognitiivisille elementeille. Tärkeää on kuitenkin huomioida, että hiljaisen tiedon kognitiivinen elementti viittaa yksilön mielikuviin todellisuudesta sekä visioihin tulevaisuudesta, eli mikä on ja miten asioiden tulisi olla. [Nonaka 2000.]

4.3 Yksilölliset ominaisuudet ja eettinen toiminta

Paradicen ja Dejoien [1991, 4] mukaan teoriapohjainen empiirinen tutkimus IT-ammattilaisten eettisestä päätöksenteosta on vähäistä. Tämä saattaa johtua siitä, että eettisen toiminnan empiirinen tutkiminen on melko vaikeaa. Toinen syy voi olla myös se, että työntekijän eettiseen toimintaan vaikuttavat erilaiset

yksilölliset piirteet, kuten ikä ja sukupuoli. Ikä voi olla vaikuttava tekijä siinä mielessä, että eettiset näkemykset muuttuvat iän ja kokemuksen karttuessa. [Leonard *et al.* 2003, 147.] Yleensä vanhemmat ihmiset ovat huolestuneempia moraalisisista kysymyksistä ja muiden hyvinvoinnista kuin nuoremmat ihmiset [Gattiker and Kelley 1999]. Työkokemuksen määrä voi olla myös moraaliseen toimintaan vaikuttava tekijä. Esimerkiksi Grant [2010] mainitsee tutkimuksessaan, että henkilöiden joilla ei ole ollenkaan työkokemusta, on helpompi muuttaa toimintaansa eettisempään suuntaan. Syy tähän voi löytyä siitä, että työkokemusta omaamattomat tulevat helpommin tietoiseksi kontekstista, jossa moraaliset tekijät nousevat esiin.

Sukupuoli voi olla myös eettiseen toimintaan vaikuttava tekijä, mutta sen vaikutusta on tutkittu vähän eikä teorian tietoa ole olemassa paljoa. [Adam and Ofori-Amando 2000, 46]. Gattiker ja Kelleyn [1999, 236] näkemyksen mukaan naiset ovat yleisesti ottaen huolestuneempia tekniikan erilaisista vaikutuksista kuin miehet. Naiset ovat herkempiä tunnistamaan epäeettistä toimintaa ja naiset ovat myös huolestuneempia yksityisyyteen sekä tekijänoikeuksiin liittyvistä kysymyksistä. Samantapaiseen tulokseen on tullut myös Khazanchi [1995, 744] tutkimuksessaan, jossa todetaan, että sukupuolella on vahva merkitys (0.05 merkityksellisyys) kyvyssä erottaa epäeettistä toimintaa. Naiset päihittävät siis miehet selvästi epäeettisen toiminnan huomaamisessa. Naiset seuraavat myös enemmän vallitsevaa kulttuuria ja sosiaalisia normeja, kun taas miehet seuraavat enemmän henkilökohtaisia asenteita ja uskomuksia. Edellä mainittujen ominaisuuksien ja persoonallisuuden piirteiden lisäksi ihmisen eettiseen toimintaan voi vaikuttaa myös ympäristö. Tämän vuoksi yksilön toiminta saattaa olla hyvin tilanneriippuvaista. [Leonard *et al.* 2003, 147.]

Grant [2010] erottaa neljä erilaista tekijää, jotka vaikuttavat ihmisen eettiseen käytökseen. Jokaisen tekijän tulee olla läsnä, jotta eettinen toiminta ylipäätään on mahdollista. Ensimmäinen tekijä on moraalinen herkkyyys, eli kyky tulkita tilanteita ja huomioida asioita. Toinen tekijä taas on moraalinen käsitys, eli kyky toimia moraalisesti ja päättää oikeista toimenpiteistä vaihtelevissa tilanteissa. Kolmantena tekijänä Grant [2010] mainitsee moraalisen motivaation, millä tarkoitetaan kykyä priorisoida toistensa kanssa ristiriidassa olevia arvoja. Viimeinen tekijä on moraalin luonne, jolla tarkoitetaan erilaisten kykyjen, kuten rohkeuden ja periksiantamattomuuden kyvyn omaamista sekä kykyä ohittaa häiriötekijöitä ja asioiden tekemisen kykyä.

4.4 Mitä on hyvä ja eettinen tietojärjestelmän käyttö

Todennäköisesti monikaan ei tule miettineeksi sitä, käyttääkö tietojärjestelmiä eettisesti tai hyvin. Asiaa ehkä kannattaisi miettiä, koska Dhariwalin ja muiden [2010] mukaan eettinen tietojärjestelmän käyttö lisää yleistä hyvinvointia ja onnellisuutta. Hyvä teknologia lisää parhaimmillaan koko yhteisön onnellisuutta: hyvän tekniikan voidaan katsoa olevan onnellisuutta lisäävä tekijä ja huonon tekniikan taas onnellisuutta vähentävä tekijä. Moraalisen tietotekniikan on hyvä seurata yleisiä periaatteita liittyen ihmisen vapautteen ja oikeudenmukaiseen kohteluun sekä arvokkuuden suojelemiseen ja hyväntahtoisuuden periaatteiden noudattamiseen. Nämä periaatteet voivat myös parhaimmillaan ohjata ihmisiä tietojärjestelmien käytössä. [Dhariwal *et al.* 2010.]

Hyvässä tietojärjestelmän käytössä otetaan huomioon käytön konteksti. Tietojärjestelmä ei koskaan ole olemassa tyhjiössä, vaan järjestelmän käyttö tapahtuu aina jossakin sosiaalisessa tai organisationaalisessa kontekstissa. Kontekstin ymmärtäminen voi olla tärkeää, koska yksi järjestelmä ei ole sama järjestelmä eri konteksteissa. Yleensä näitä kontekstiin liittyviä sosiaalisia kysymyksiä ei osata ennalta miettiä. [Kaplan and Maxwell 2005.] Tämä voikin olla yksi syy siihen, miksi tietojärjestelmiin liittyy niin paljon erilaisia eettisen vastuun ongelmia.

5. Tietojärjestelmien yksityisyyden ongelmat

Aikaisemmissa kohdissa on käsitelty pääasiassa eettiseen vastuuseen liittyviä kysymyksiä. Tässä tutkielman viimeisessä luvussa painotetaan yksityisyydensuojan eettisiä kysymyksiä. Erityisesti henkilötietojen yksityisyyttä käsitellään tietosuojan näkökulmasta. Yksityisyys tietojärjestelmissä on ajankohtainen aihe uuden tietosuojalainsäädännön myötä, koska laissa säädetään virheellisen tiedon korjaamisesta ja tarpeettoman tiedon poistamisesta [Tietosuojavaltuutettu 2018].

5.1 Yksityisyys ja tietosuoja

Wahlströmin ja muiden [2011, 473] mukaan ihmiset pyrkivät luomaan ja ylläpitämään vapautta, identiteettiä ja autonomiaa. Yksityisyys voidaan saavuttaa esimerkiksi kieltäytymällä osallistumisesta tai antamalla virheellistä tietoa. Mikäli ihmiset antavat virheellistä tietoa, vaarantaa se minkä tahansa sellaisen teknologian tehokkuutta, joka on riippuvainen virheettömästä ja tarkasta tiedosta. [Wahlström *et al.* 2011.]

Yksityisyyden voidaan katsoa muodostuvan eri tavoin. Ensimmäinen tapaus nousee yhteiskunnan viestinnän muodoista. Esimerkiksi joissakin yhteisöissä on sallittua vetäytyä pois yhteisön vaikutuksen piiristä, kun taas toisissa yhteisöissä se ei ole mahdollista. Se, missä määrin ihminen odottaa yksityisyyttä tietyssä kontekstissa, riippuu siis yleensä kulttuuriympäristöstä. Toiseksi, yksityisyyden odotukset muodostuvat yksilön henkilökohtaisten preferenssien mukaan. Ihmisten yksityisyyden tarve on erilainen: toiset kaipaavat enemmän yksityisyyttä kuin toiset. Koska kulttuurin normit ovat erilaisia ja vaikutuksille alttita, ovat siten myös yksityisyyden odotukset erilaisia. Yksityisyyden odotukset ovat riippuvaisia muutoksille yksilön välittömässä kontekstissa. [Wahlström *et al.* 2011, 473.]

Yksityisyyden problematiikka liittyy läheisesti tietojärjestelmiin: esimerkiksi julkishallinnon järjestelmiin kerätään paljon yksityisyyden kannalta kriittisiä henkilötietoja. Tietojen kerääminen perustuu julkiseen turvallisuuteen, mutta ongelmaksi voi muodostua asiakkaiden edun sekä julkisen edun välinen konflikti. Miten voidaan tasapainoilla sen välillä, mikä on tietoja keräävän tahon etu ja asiakkaan etu. [Wester *et al.* 2010.] Tässä palataan jälleen ristiriitaan yhteisön edun ja kollektiivisen edun välillä. Westerin ja muiden [2010] artikkelissa tämä arvojen välinen ongelma ei ratkea pelkästään kollektiivin etua

priorisoimalla, vaan kirjoittajien mukaan asiaa tulisi pohtia tarkemmin. Luottamus ja turvallisuus ovat modernin yhteiskunnan teknisen infrastruktuurin kulmakiviä. Sosiaalinen luottamus on merkityksellistä silloin, kun pohditaan miten riskejä on tarkoitus huomioida ja johtaa. Sosiaalinen luottamus on erityisen tärkeää silloin, kun puhutaan yksityisyyden kysymyksistä, jotka koskevat yhteiskunnallisia ja demokraattisia prosesseja. [Wester *et al.* 2010.]

Sosiaalisen luottamuksen näkökulmasta on mielenkiintoista, miten kahtia jakautuneita näkemyksiä yksityisyyden kysymyksissä löytyy. Esimerkiksi Westerin ja muiden [2010] mukaan suhtautuminen yksityisyyden kysymyksiin vaihtelee yksilöllisten ominaisuuksien, kuten iän ja sukupuolen perusteella. Tutkimustuloksissa ihmiset jakautuvat niihin, jotka ovat erityisen huolissaan yksityisyyden kysymyksistä sekä niihin, jotka eivät ole ollenkaan huolissaan siitä mihin heidän henkilökohtaisia tietojiaan voidaan käyttää. [Wester *et al.* 2010.] Tämä laittaakin pohtimaan, miksi ihmiset ovat huolissaan yksityisyyden kysymyksistä. Voiko huoli johtua siitä, että aina ei ole mahdollista vaikuttaa omien tietojen käyttämiseen, eli ei ole mahdollisuutta kontrolloida itseen liittyvää tietoa. Huoli voi olla ihan miettimisen arvoinen, esimerkiksi Ylioppilastutkintolautakunnan verkkosivujen palveluntarjoajalle tapahtui tietovuoto, jossa 6000 ihmisen tiedot olivat ulkopuolisten saatavilla verkossa yli kahden kuukauden ajan. Näkyvillä oli henkilötietoja puhelinnumerosta aina henkilötunnukseen asti. [Grün 2018.] Tähän tilanteeseen saattaa mahdollisesti tuoda muutosta Euroopan Unionin yleinen tietosuojasetus (GDPR), joka asettaa uusia vaatimuksia henkilötietojen ja erityisesti arkaluontoisten henkilötietojen käsittelylle ja henkilötietoja sisältävien rekisterien pitämiseksi. Tietosuojasetusta aletaan soveltaa 25.5.2018 alkaen kaikissa EU:n jäsenmaissa ja sitä sovelletaan lähtökohtaisesti kaikkeen henkilötietojen käsittelyyn. [Tietosuojavaalututettu 2018.]

Uuden tietosuojasetuksen mukaan henkilötietoja on voitava anonymisoida ja pseudonymisoida, mikä voi olla merkityksellistä tietojen yksityisyyden kannalta. Anonymisoinnilla tarkoitetaan "Henkilötiedon tunnistettavuuden poistamista siten, että yhdistäminen rekisteröitävänsä ei ole enää mahdollista" [Valtiovarainministeriö 2016]. Pseudonymisointi taas määritellään tiedon "salanimellä julkaisemiseksi" [Valtiovarainministeriö 2016]. Pseudonymisoinnilla siis tarkoitetaan henkilötietojen sellaista käsittelemistä, jossa tietoja ei ole mahdollista suoraan yhdistää henkilöön käyttämättä lisätietoja. Lisätiedot on säilytettävä erillään, jotta yhdistämistä tunnettuun tai tunnistettavissa olevaan henkilöön ei pääse tapahtumaan. [Valtiovarainministeriö 2016.]

Tietosuojaan liittyvällä lainsäädännöllä on merkitystä, koska esimerkiksi Phelps ja muiden [2000, 27-28] tutkimuksen mukaan ihmiset haluavat enemmän kontrollia tietojensa käyttämisen suhteen. Puolet vastaajista oli sitä mieltä, että haluaisi enemmän tietoa siitä, mihin heidän tietojensa käytetään. Mielenkiintoinen yksityiskohta tutkimuksessa oli, että yksityisyydestä olivat eniten huolissaan ammatillisia opintoja tai vähän lukio-opintoja suorittaneet. Sen sijaan lukiosta tai ammatillisesta koulutuksesta valmistuneet eivät olleet yhtä huolestuneita yksityisyydestä. [Phelps *et al.* 2000, 27-28.] Tätä tutkimustulosta joutuu ehkä hieman kyseenalaistamaan, mutta todennäköisesti muita vastaavia tutkimuksia ei ole olemassa, joihin verrata.

5.2 Yksityisyys tietojärjestelmien kehittämisen näkökulmasta

Tietojärjestelmien kehittämisessä ja rekisterin pidossa on tärkeää huomioida yksityisyydensuoja, koska esimerkiksi terveydenhuollon tietojärjestelmiin ja rekistereihin kerätään asiakkaista arkaluontoista tietoa. Tiedot sairauksista, kuten riippuvuuksista, mielenterveyden ongelmista ja HIVista/AIDSista ovat erityisen arkaluontoisia. Järjestelmissä on tietoa terveydenhuollon ammattilaisista ja julkisuuden henkilöistä, jotka saavat hoitoa terveydenhuollossa. [Reid *et al.* 2003.] Sosiaalialan tietojärjestelmissä on myös yksityisyyden kannalta arkaluontoista tietoa. Tällaisen tiedon käsittely edellyttää henkilöstöltä osaamista tietosuojaan liittyvissä kysymyksissä. EU:n yleinen tietosuojasetus opastaakin koulutuksiin ja ohjeistuksiin liittyvissä asioissa. Asetuksessa sanotaan, että rekisterinpitäjän tulee huolehtia henkilötietoja käsittelevien työntekijöiden asianmukaisesta koulutuksesta. Koulutukseen kuuluu ohjeistuksen laatiminen, jossa kuvataan millainen henkilötietojen käsittely liittyy tehtävänkuvaaan. Rekisterinpitäjän vastuulle kuuluu varmistaa, että tietojen käsittelijällä ja henkilötietojen käsittelyyn osallistuvilla henkilöillä on riittävä tietosuojasaaminen. [Valtiovarainministeriö 2016.]

EU:n yleinen tietosuojasetus vaikuttaa myös uusien tietojärjestelmien kehittämiseen yksityisyyden näkökulmasta. Tietojärjestelmää rakennettaessa järjestelmän kehittäjiä on huomioitava yksityisyyteen liittyvät lait ja säännökset suunnittelutyössä. Monissa järjestelmissä implisiittiset yksityisyyden kysymykset ovat yhtä tärkeitä kuin muut toiminnalliset ja turvallisuuden vaatimukset, mutta niitä harvoin analysoidaan saati suunnitellaan huolellisesti kehittämisprosessin alussa. Paljon yleisempää on se, että järjestelmiin pyritään istuttamaan yksityisyyden suoja jälkikäteen, jolloin se kasvattaa kustannuksia ja vaarantaa kokonaissuunnittelun integriteetin. Yksi

suurimmista ongelmista on saada lainsäädäntö ja tietotekniikka toimimaan hyvin yhteen. Lakikieltä on usein vaikea saada taipumaan tietojärjestelmän vaatimusmäärittelyihin. Tässä kohtaa on hyvä huomioda, että yleensä yksityisyyden ongelmat järjestelmissä eivät johdu tahallisesta laiminlyönnistä suunnittelussa, vaan pikemminkin epähuomiossa tehtyihin virheisiin. Esimerkiksi yhdysvaltalainen terveydenhuollon toimija Kaiser Permanente paljasti vahingossa yksityistä terveystietoa (ajanvarauksia, vastauksia potilaiden kysymyksiin ja lääketieteellisiä neuvoja) yli 800 potilaalle heidän terveydenhuollon web-portaalinsa kautta. Tällaisilla yksityisyyden loukkauksilla voi olla vakavia seuraamuksia yrityksille ei pelkästään rahallisessa mielessä, vaan myös asiakkaiden menetettynä luottamuksena yritystä kohtaan. [Guarda and Zannone 2008.]

Tämä herättääkin pohtimaan, onko meidät pakotettu luottamaan yrityksiin. Miten voimme käydä tänä päivänä lääkärissä ilman että meistä jää erilaisia merkintöjä tietojärjestelmiin. Miten voimme luottaa, että tietoa käsitellään luottamuksella. Kyse ei ole pelkästään siitä, että tietojärjestelmät itse ovat turvallisia, koska tietojärjestelmiä käyttävät aina ihmiset. Tiedot voivat vuotaa järjestelmistä pelkästään inhimillisen virheen vuoksi ja seuraamukset voivat olla vakavia. Tiedot ovat niin hyvin turvassa kuin mikä on yrityksen tai organisaation heikoin lenkki. Meidät on käytännössä pakotettu luottamaan siihen, että yksityisyydensuojamme on kunnossa. Kyseessä ei siis välttämättä ole mikään yksilön tekemä oma valinta.

Pohdintaa herättää myös se, millainen vastuu järjestelmän kehittäjällä on yksityisyyden suhteen. Walsham [1996, 77] mainitsee artikkelissaan esimerkin järjestelmäkehittäjästä, joka on huolissaan työpaikkansa järjestelmän yksityisyydensuojasta. Järjestelmäkehittäjä on maininnut huolestaan yrityksen johdolle, mutta nämä eivät ole reagoineet positiivisesti hänen huoliinsa. Mikäli työntekijä ei pysty muuttamaan esimiesten näkemyksiä, niin tuleeko hänen kunnioittaa enemmän työsopimustaan yrityksen kanssa vai omia arvojaan ja näkemyksiään yksityisyydestä ja luottamuksellisuudesta? Monet tutkijat ovat käsitelleet aihetta hyvän järjestelmäkehittämisen näkökulmasta, joka painottaa järjestelmän suunnittelun ja kehittämisen prosessien eettisyyttä. Esimerkiksi *voimaantumisen* (Empowerment) käsitteestä on kirjoitettu viime aikoina melko paljon. Rappaportin [1987, 121] mukaan voimaantumisella tarkoitetaan yksilön päätäntävaltaa omasta elämästään. Tässä yhteydessä voimaantumisella tarkoitetaan lähinnä organisaation alemman hierarkian työntekijöiden vallan lisäämistä työtehtävien suorittamisessa. Tällä voidaan nähdä olevan yhteyksiä eettisen toiminnan lisääntymiseen organisaatiossa. Voimaantumisen ajatuksen pohjalta on syntynyt erilaisia projektejakin, kuten Utopia -projekti, jossa

suunniteltiin tietokonepohjaisia välineitä graafisen alan ammattilaisille. [Walsham 1996, 77.]

Voimaantumiseen liittyy myös kyvykkyys, joka representoi yksilön pystyvyyttä sekä itsenäisenä toimijana että osana ryhmää, jonka tarkoituksena on välttää vahinkoa ja etsiä arvostettuja toiminnan muotoja. Arvossa pidettyä voi olla esimerkiksi kyky tehdä järkeviä ratkaisuja eri tilanteissa. [Johnstone 2007, 76-77.] Voimaantumiseen liittyy myös mahdollisuus ja vapaus toimia ympäristössä, jossa yksilön oma potentiaali pääsee parhaiten esiin. Kun katsotaan tietotekniikan ja tietojärjestelmien myötävaikutusta siitä perspektiivistä, miten ne edesauttavat ihmisten mahdollisuuksia elää arvojensa mukaista elämää, saadaan laajempi näkemys nykyhetken huoliin, joita ei pysty tällä hetkellä riittävästi erottamaan. Kyvykkyyteen keskittyminen helpottaa eettistä keskustelua ja intuitiota siitä, mikä on todella merkityksellistä sen sijaan, että keskityttäisiin pelkästään jonkin eettisen teorian metodologiaan. [Johnstone 2007, 86.]

Voiko edellä mainittu voimaantumisen näkemys olla kuitenkin yksi tapa siirtää vastuuta liikaa organisaatiolta työntekijälle? Kuinka pahaan välikäteen työntekijä voi joutua pyrkiessään noudattamaan yksityisyyteen liittyviä lakeja ja säännöksiä, joita työnantaja ei noudata. Kenellä oikeastaan on viime kädessä vastuu siitä, miten eettisesti organisaatio toimii yksityisyyteen liittyvissä kysymyksissä. Organisaatiossa voidaan laatia eettisiä ohjeita, mutta niiden noudattaminen voi silti olla tehtävä, jota kaikki johtavassa asemassa olevat eivät välttämättä tue riittävästi. Viime kädessä on luotettava lainsäädäntöön ja ihmisten omaan kontrolliin, joka sanelee miten pahimmilta ylilyönneiltä välttyttäisiin. Ihmisten sosiaalinen kontrolli on myös melko toimiva väline pahimpien ylilyöntien ja laiminlyöntien välttämiseen – aina jossain vaiheessa joku pistää pisteen liiallisille laiminlyönneille.

Lopuksi on hyvä todeta, että *“moraali on yksilön ongelma, ei tekninen ongelma”* [Introna 2007, 12.] Yleisin näkemys tekniikasta on, että se on artefakti tai väline tai systeemi, joka on saatavilla ihmisiä varten, jotta he voivat saavuttaa tavoitteensa – kirjoittaa sähköpostia, tallentaa tietoa, kommunikoida toistensa kanssa ja niin edelleen. Tämä näkemys on melko syvään juurtunut intuitio siitä, miten järjestelmät nähdään meistä erillään olevana asiana. Kuitenkin kun käytämme erilaisia järjestelmiä, niillä on yleensä jokin vaikutus meihin. Järjestelmiin liittyvät asiat nähdään myös enemmän tai vähemmän teknisinä kysymyksinä, jotka ovat pääasiassa suunnittelijoiden ja insinöörien toimintaluetta. Tärkeää on kuitenkin muistaa, että yhteiskunnan tapa käyttää järjestelmiä sekä tämän käyttämisen vaikutus meihin, ovat myös omia relevantteja kysymyksiänsä. [Introna 2007, 12.]

6. Yhteenveto

Tässä tutkielmassa lähdettiin liikkeelle pohtimalla, mitä eettisiä ongelmia tietojärjestelmiin liittyy sekä miksi eettisten kysymysten pohdinta on merkityksellistä. Tietojärjestelmiin liittyy useitakin eettisiä kysymyksiä, mutta tärkeimmiksi kysymyksiksi nousivat erilaiset vastuuseen ja luottamukseen liittyvät kysymykset. Toinen keskeinen ongelma on tietojärjestelmien yksityisyyden ongelmat. Vastuuseen ja myös yksityisyyteen liittyvissä asioissa tärkeäksi osoittautui ammattilaisten etiikka. Tietojärjestelmien kehittämisessä ammattilaisten etiikka ei näytä aina olevan se tärkeimpänä pidetty asia, vaan muut kriteerit johtavat kehittämistyötä. Kuitenkin tietojärjestelmissä säilytetään paljon arkaluontoista tai muuten suojeltavaa tietoa, jonka ei toivoisi joutuvan väärin käsiin tai joutuvan muuten väärinkäytetyksi. Vaikka ammattilaisilla olisikin kyky ja halu toimia eettisesti, asettaa yrityksen johto ja tulostavoitteet helposti erilaisia vaatimuksia toiminnalle. Monet eettisesti asioita ajattelevat tuntevat ongelman painon omassatunnossaan ja joutuvat tekemään mahdollisesti vaikeitakin valintoja omaan arvopohjaansa tukeutuen. Vastuun kysymys on siten monellakin tapaa ongelmallinen, koska se ei ole erityisen selvärajaista – milloin vastuu kuuluu työntekijälle ja milloin se kuuluu yrityksen johdolle tai käyttäjälle. Filosofiset teoriat tarjoavat tukea ongelman ratkaisemiseen, kuten suurten joukkojen etujen laittamisen pienen joukon etujen edelle. Tähän samaan ajatukseen pohjautuu myös ”enemmistön demokratia” ajattelu, jota käytetään myös eettisten ongelmien hahmottamisessa.

Toisaalta myös lainsäädäntö asettaa omat vaatimuksensa yrityksen ja työntekijöiden toiminnalle, mutta seurataanko lakien noudattamista riittävästi? Onko lainsäädäntö aina eettistä? Luultavasti ei, koska lakeja laativat ihmiset, jotka eivät välttämättä pysty aina huomioimaan kaikkia asiaan liittyviä vivahteita. Kysymykset ovat usein monimutkaisempia kuin miltä lakeja laatiessa aluksi näyttää. Lainsäädäntö ei myöskään aina pysy mukana muutoksien vauhdissa. Toisaalta lakeja voidaan aina muuttaa eettisemmiksi, jos riittävästi löytyy siihen suuntaan ohjaavaa tahtotilaa. Nämä ovatkin sitten enemmän poliittisen päätöksenteon tulosta.

Eettisyyttä pohdittaessa usein vaikeinta on löytää se ”oikea” totuus, jota ei todennäköisesti ole edes olemassa. Miksi tietojärjestelmien eettisyyden pohtiminen on tärkeää? Se voi olla tärkeää siksi, että parhaimmillaan ajattelu voi ohjata toimintaa kohti kokonaisuuden kannalta parasta ratkaisua. Pahimmillaan se voi aiheuttaa päättämätöntä toimintaa, koska ei päästä minkäänlaiseen lopputulokseen siitä, miten tulisi toimia eettisesti. Eettisyys on

kuitenkin aina jossain määrin subjektiivista ja siksi kokonaisuuden havainnointi on merkityksellistä. Etenkin yksityisyyden kysymyksissä eettisyyden pohdinta on tärkeää, koska sitä ei ole tutkittu erityisen paljon ja sen merkitys on koko ajan tärkeämpi yhteiskunnan nopean muutoksen vuoksi.

Viiteluettelo

- Alison Adam and Jacqueline Ofori-Amanfo. 2000. Does gender matter in computer ethics? *Ethics and Information Technology* 2, 1, 37-47.
- Timo Airaksinen ja Mervi Friman. 2008. *Asiantuntija-ammattin etiikka*. HAMK.
- Ibrahim Akman and Alok Mishra. 2009. Ethical behavior issues in software use: An analysis of public and private sectors. *Computers in Human Behavior* 25, 6, 1251-1257.
- Steven Alter. 1999. A general, yet useful theory of information systems. *Communications of the Association for Information Systems* 1, 3, Article 13.
- Pierre-Emmanuel Arduin, Michel Grundstein, and Camille Rosenthal-Sabroux. 2015. *Information and Knowledge Systems*. John Wiley & Sons.
- Frances Bell and Alison Adam. 2004. Whatever happened to information Systems ethics. Caught between the devil and the deep blue sea. *Information Systems Research*, 169-174.
- Alexander Boldizar and Outi Korhonen. 1999. Ethics, morals and international law. *Ejil* 10, 2, 279-211.
- David Carr and Jan Steutel. 2005. *Virtue Ethics and Moral Education*. Routledge.
- Manju Dhariwal, Ramesh C Pradhan, and Raghurib Sharan. 2010. Engaging the student of technology in an ethical discourse in the information age: thoughts of Wiener and Gandhi. In: *Proceedings of the Eleventh International Conference on the Possibilities of Ethical ICT*, 62-71.
- Urse E. Gattiker and Helen Kelley. 1999. Morality and computers: Attitudes And differences in moral judgements. *Information Systems Research* 10, 3, 233-254.
- Candace T. Grant. 2010. Influencing the ethical awareness of young ICT professionals. In: *Proceedings of the Eleventh International Conference on the Possibilities of Ethical ICT*, 248-256.
- Emma Grün. 2018. Tuhansille kolahtelee nyt postiluukusta kirje tietovuodosta – jos tilasit kopion ylioppilastodistuksesta, saatat olla yksi heistä. *Helsingin Sanomat*.
- Paola Guarda and Nicola Zannone. 2008. Towards the development of privacy-aware systems. *Information and Software Technology* 51, 2, 337-350.
- Olli I. Heimo, Minna M. Rantala, and Kai K. Kimppa. 2015. Wilma ruined my life: how an educational system became the criminal record for the adolescents. *SIGCAS Computers & Society*, 45, 3, 138-146.

- Olli I. Heimo, Jani S. Koskinen, and Kai K. Kimppa. 2013. Responsibility in acquiring critical governmental information systems: Whose fault is failure? In: *Proceedings of the Thirteenth International Conference on the Possibilities of Ethical ICT*, 213-217.
- Lucas D. Introna. 2007. Maintaining the reversibility of foldings: Making the ethics (politics) of information technology visible. *Ethics and Information Technology* 9, 11-25.
- Tony Jewels and Nina Evans. 2005. Ethical IT behavior as a function of environment. *Issues in Informing Science and Information Technology* 2, 383-393.
- Justine Johnstone. 2007. Technology as empowerment: a capability approach to computer ethics. *Ethics and Information Technology* 9, 73-87.
- Bonnie Kaplan and Joseph A. Maxwell. 2005. Qualitative research methods for evaluating computer information systems. In: *Proceedings of the Evaluating the Organizational Impact of Health Care Information Systems*, 30-55.
- Deepak Khazanchi. 1995. Unethical behavior in information systems: The gender factor. *Journal of Business Ethics* 14, 9, 741.
- Richar P. Kulczak. 1999. *Virtue ethics and Aristotle*. University of London, Department of Philosophy.
- Kenneth C. Laudon, and Jane P. Laudon. 2007. *Management of Information Systems: Managing the Digital Firm*. Pearson Prentice Hall.
- Lori N.K. Leonard, Timothy Paul Cronanb, and Jennifer Kreie. 2003. What Influences IT ethical behavior intentions-planned behavior, reasoned action, perceived importance, or individual characteristics? *Information & Management* 42, 1, 143-150.
- Karen D. Loch, Sue Conger, and Effy Oz. 1998. Ownership, privacy and monitoring in the workplace: A debate on technology and ethics. *Journal of Business Ethics* 17, 6, 653-663.
- Richard O. Mason. 1986. Four ethical issues of information age. *MIS Quarterly* 10, 1, 5-12.
- Neil Kenneth McBride. 2014. Active ethics: an information systems ethics for the internet. *Journal of Information, Communication and Ethics*. 12, 1, 27-44.
- Ikujiro Nonaka. 1994. A Dynamic Theory of Organizational Knowledge Creation. *Organization Science* 5, 1, 14-37.
- Effy Oz. 1992. Ethical standards for information systems professionals: A case for a unified code. *MIS Quarterly* 16, 4, 423-233.
- David B. Paradice and Roy M. Dejoie. 1991. The ethical decision-making

- process of information systems workers. *Journal of Business Ethics* 10, 1, 1-21.
- Joseph Phelps, Glen Nowak and Elizabeth Ferrell. 2000. Privacy concerns and consumer willingness to provide personal information. *Journal of Public Policy and Marketing* 9, 27-41.
- Julian Rappaport. 1987. Terms of empowerment/exemplars of prevention: toward a theory for community psychology. *American Journal of Community Psychology* 15, 2, 121-148.
- Jason Reid, Ian Cheong, Matthew Henricksen, and Jason Smith. 2003. A novel use of RBAC to protect privacy in distributed health care information systems. In: *ACISP'03 Proceedings of the 8th Australasian Conference on Information Security and Privacy*, 403-415.
- Roger Scruton. 1988. *Spinoza*. Otava.
- Benedictus de Spinoza. 1994. *Etiikka*. Gaudeamus.
- Ralph Stair and George Reynolds. 2014. *Principals of Information Systems*. Course Technology.
- Ian Stoodley. 2009. *It-professionals' experience of ethics and its implications for IT education*. Ph. D. Dissertation, Faculty of Science and Technology, Queensland University of Technology.
- Tietosuojavaltuutettu. 2018. Tietosuojasetus. www.tietosuoja.fi. Luettu 1.5.2018.
- Jacques P. Thiroux and Keith W. Krasemann. 2012. *Ethics: Theory and practice*. Pearson Prentice Hall.
- Lawrence A. Tomei. 2010. *ICTs for Modern Educational and Instructional Advancement: New Approaches to Teaching*. Information Science Reference.
- Tero Vartiainen. 2011. ICT-etiikka: moraalisen käyttäytymisen näkökulma. In: Mika Laakkonen, Suvi Lamminpää ja Jarno Malprade (toim.) *Informaatioteknologian filosofia*. Lapin yliopistokustannus, 37-48.
- Kirsten Wahlström, N. Ben Fairweather and Helen Ashman. 2011. Brain-computer interfaces: A technical approach to supporting privacy. In: *Proceedings of the Twelfth International Conference on the Possibilities of Ethical ICT*, 313-321.
- Geoff Walsham. 1996. Ethical theory, codes of ethics and IS practice. *Information Systems Journal* 6, 69-81.
- John Weckert. 2013. *Computer Ethics*. Routledge.
- Misse Wester and Per Sandin. 2010. Privacy and the public – Perception and

acceptance of various applications of ICT. In: *Proceedings of the Eleventh International Conference on the Possibilities of Ethical ICT*, 580-586.

Valtiovarainministeriö. 2016. VAHTI-raportti 1/2016. www.vahtiohje.fi. Luettu 1.5.2018.

Joseph W. Weiss. 2017. Ethics and CSR research in top-ranked IS journals 1980-2013: a developing trend or anomaly? In: *Proceedings of the 50th Hawaii International Conference on System Sciences*, 5758-5762.

Automatic Humor Recognition in Verbal Contexts

Anna Solantausta

Abstract.

Automatic humor recognition in verbal contexts refers to the computational task of determining if an instance of language can be classified as being humorous. Because humor is a high-level communication feature essential for people, computers need to be equipped with the ability to identify and respond to humor appropriately for the interaction between people and machines to be natural. However, humor is a difficult concept to define and identifying humor-eliciting features occurring in text or speech can sometimes be a difficult task even for human beings. In recent years, attempts at automatic humor recognition have become more prevalent with the increase in the deployment of different types of conversational agents. This paper describes some discernible qualities of verbal humor as well as some of the different definitions offered for humor and offers an introduction to automatic humor recognition and its implementations.

Keywords: Computational humor, humor recognition, verbal humor

1. Introduction

All people, regardless of their respective culture and upbringing, are familiar with humor. Raskin [2012] points out that humor is by no means a rare phenomenon, and that remarkably many situations have some innate quality to them that allows most people to find them funny. In the domain of social interaction, humorous exchanges are commonplace, and it is typical for us to try to elicit laughter from our conversation partners [Wyer and Collins 1992]. Yet, the specific nature of this phenomenon continues to puzzle and elude us [Mihalcea and Strapparava 2005]. Humor has been theorized about for millennia, but it has been proven to be very difficult to define [Taylor and Mazlack 2004]. Despite the lack of work involving the comprehensive description of the underlying mechanisms of humor [Ritchie 2001], numerous definitions and theories of humor exist today [Taylor and Mazlack 2004].

We might be tempted to define humor as something that elicits laughter in the hearer, but it could be argued that it would oversimplify the matter. A witty remark might elicit a roaring cloud of laughter upon the first hearing, but one might merely crack a smile hearing it again later, even though one might still find it very humorous. [Hurley *et al.* 2011.] In addition, although all people

are capable of recognizing humor, not everyone finds mirth in the same utterances or situations [Raskin 2012]. Taste in humor greatly varies from person to person, and it can be affected by a variety of things, such as the person's current mood and their recent experiences [Taylor and Mazlack 2004]. There are also jokes, such as inside jokes, that are only funny to a restricted group of people [Raskin 2012]. Humor is contextual and convoluted, but it undoubtedly plays an inherent part in people's lives [Hurley *et al.* 2011]. I will attempt to give a brief overview of humor and its different definitions and theories in Section 2. The theory discussion will be limited to the modern theories of humor as the classical theories (Greek and Latin) are outside the scope of this paper.

As a subdiscipline of computational linguistics, research in computational humor is tasked with developing the necessary resources and algorithms for computers to be able to not only produce, but also understand humor [Hempelmann *et al.* 2006]. Much of the research has been focused on generating humor with significantly less attention given to the task of recognizing it. Automatic humor recognition seems to comprise of qualities that make it significantly more difficult to model for computers than humor generation. [Mihalcea and Strapparava 2006.] I will cover these characteristics in Section 3 as well as introduce some implementations of automatic humor recognition in Section 4.

One necessary resource for further advances in computational humor is a complete model of humor. Although much discussed in multiple different fields, there has been an absence to some degree of studies delving into detailed description and precise analysis of the underlying mechanisms of humor [Ritchie 2001]. The creation of such a model seems far from a simple task considering the complex nature of humor, but as is pointed out by Ritchie [2001], it is by no means impossible. According to Ritchie [2001], the complexities merely signal that there is great variety in the factors that the model must address, and that the model must be able to accurately portray them. This topic, in addition to other problems faced in automatic humor recognition, will be discussed in greater detail later in Section 3.

2. Humor

2.1. Defining Humor

As mentioned earlier in the introduction, a typical inclination is to define humor in relation to laughter. Hurley *et al.* [2011] identify a common circular pattern in dictionaries where *humor* is defined as something that makes people

laugh, and then *laughter* is defined as the reaction humor elicits from the hearer. Laughter does not, however, always follow humor nor does humor always precede laughter. One factor that shows the two concepts to be asymmetrical is the existence of the so called *non-Duchenne laughter*. It is used to describe situations in which we laugh nonspontaneously, usually with some other purpose in mind besides expressing true mirth. For example, one might force out a laugh in an attempt to preserve the conversation partner's feelings after they have told a joke that is not found to be funny. [Hurley *et al.* 2011.]

We should not, then, define humor simply as something that is capable of making people laugh. In the context of verbal humor we can, however, identify several common features that could indicate the presence of humor. One common feature that many humorous verbal exchanges seem to share is their tendency to violate a set of communication principles called *Grice's maxims* [Attardo 1990]. These principles are based on the idea that communication is a cooperative activity where all parties involved behave in a rational manner. Conversation participants are generally expected to provide enough information (*maxim of quantity*), convey relevant information (*maxim of relevance*), be clear and concise (*maxim of manner*), and only provide details that they themselves believe to be true (*maxim of quality*). As we normally expect our conversational partners to follow these maxims, flouting or violating the principles signal to us that the speaker might be implying something beyond the literal meaning of what was explicitly said. [Plag *et al.* 2015.]

In the case of a principle violation, the implied meaning is often meant to be humorous. Take, for instance, the following example:

"Do you believe in clubs for young men?"

"Only when kindness fails."

– W.C. Fields

This example of verbal humor violates the maxim of manner by being ambiguous. [Attardo 1990.] The source for the ambiguity is the word *club* and its two meanings that significantly differ from each other: a group dedicated to some activity (1) and a blunt-force weapon (2). The sentence structure of the question will lead most of us to choose the first meaning for the word *club*, but the response forces us to look for the second meaning because it shifts our perception of the question. This shift in the interpretation of *club* can potentially lead to a feeling of mirth. It is important to note, however, that we do not consciously choose a meaning for the polysemous word (i.e. a word with several meanings)

in the beginning, but instead unconsciously apply a suitable reading (usually the one that comes to mind most quickly) [Wyer and Collins 1992].

In an effort to identify qualities that separate humorous and serious texts from another, Mihalcea and Pulman [2007] found humorous texts to frequently include two salient features: human-centeredness and negative polarity. The human-centeredness of humor refers to the fact that jokes often seem to include human-centric vocabulary, and they frequently use personal pronouns, such as *you* and *I*. Many jokes also contain references to different professions, such as lawyers and programmers, and to human “weaknesses”, such as alcoholic beverages and stupidity. The negative polarity of humor refers to the negative word forms, and words with negative connotations that are also often employed in jokes. [Mihalcea and Pulman 2007.]

Humorous texts also frequently include specific stylistic features and rhetoric devices, such as *alliteration*, *rhyme*, and *word antonymy* [Mihalcea and Strapparava 2006]. Alliteration and rhyme refer to two different types of sound repetition patterns. Instances of alliteration are found at the beginning of a syllable (usually in word-initial position), while types of rhyme involve the end of a syllable (usually in word-final position). [Montgomery *et al.* 2007.] An example of a joke with an alliteration chain (underlined) would be the following: “Infants don’t enjoy infancy like adults do adultery” [Mihalcea and Strapparava 2006]. Although the spelling matches the alliteration pattern in this example, it is important to note that the written form of a word does not indicate alliteration or rhyme as both are strictly concerned with sounds and not spellings [Montgomery *et al.* 2007]. Word antonymy, then, concerns the use of word pairs with opposite meanings for comedic effect. Examples of antonymy would be, for example, the word pairs *quiet* - *loud* and *modest* - *proud*. [Mihalcea and Strapparava 2006.]

2.2. Humor Theories

Although theoreticians have yet to produce a single universally accepted theory of humor [Hurley *et al.* 2011], a three-group classification is commonly accepted for the modern theories of humor. These groups are: incongruity theories, hostility/disparagement theories, and release theories [Attardo 2010]. According to release theories, humor may facilitate the release of psychological and physical tensions [Attardo 2014] or it may, for example, release people from different laws or conventions. Linguistically, release theories account for situations where the rules of language or communication are in some way bro-

ken or contorted. Typical examples would be puns or situations that involve infractions to Grice's maxims. [Attardo 2010.] Hostility/disparagement theories conclude that people derive amusement from other people's misfortunes [Wyer and Collins 1992]. Since incongruity theories are widely used in the domain of computational humor, I will next provide a slightly more in-depth description of incongruity theories.

Incongruity is one of the oldest explanations offered for humor. According to the incongruity theories, humor rises when we first recognize an incongruity between two concepts and subsequently resolve that discrepancy in our minds. Incongruity theories are sometimes referred to as incongruity resolution theories since the resolution of the incongruity can be seen by some scholars to be equally as important as the incongruity perceived between the concepts. [Attardo 2014.] Puns, like the club joke presented earlier in this section, are classic examples of incongruity since they juxtapose two different semantic meanings. However, the incongruity in humorous exchanges need not be limited to semantic juxtaposition. Let us look at a joke that can be explained by applying incongruity theories in its analysis, and that includes an incongruity in the punchline that is not restricted to the semantic domain:

A blind man enters a department store, picks up his dog by its tail and begins swinging it over his head. A clerk hurries over and says, "Can I help you, sir?" "No thanks," the man replies, "I'm just looking around."

The punchline is at first glance a typical example of a pun since the joke requires us to think about the idiomatic meaning of *looking around* as well as its literal meaning. Yet, relying on our knowledge of the semantic meanings alone will not allow us to appreciate the joke. That is because the joke also requires us to tap into our general knowledge, namely that seeing-eye dogs can be used by blind individuals to "look around". [Wyer and Collins 1992.]

3. Automatic Humor Recognition

Automatic humor recognition refers to the computational task of determining whether a given utterance or sentence is expressing a humorous sentiment either explicitly or implicitly. Traditionally, the task of humor recognition has been formulated as a binary classification task where humorous instances of language are distinguished from nonhumorous ones based on a predetermined set of linguistic features. [Yang *et al.* 2015.] For example, Mihalcea and Strapparava [2006] applied machine learning techniques to distinguish humorous

one-liners from nonhumorous sentences based on a few stylistic elements. The experiment will be discussed in more detail in Section 4.1.

Automatic humor recognition is also concerned with identifying the specific triggers that cause an utterance or sentence to be humorous through the analysis of linguistic as well as extra-linguistic information. These triggers can then be used to discriminate between humorous and nonhumorous instances of language. [Pérez 2012.] An example of this would be the experiment formulated by Yang *et al.* [2015] where they sought to automatically detect humor triggers found in instances of written language. The experiment will be discussed later in Section 4.4.

In a binary classification task in humor recognition, negative (nonhumorous examples of language) and positive (humorous examples of language) datasets are used to automatically train computational models and evaluate their performance. The datasets are first input into a classifier as training data. Commonly used classifiers in automatic humor recognition include, for example, Naïve Bayes and Support Vector Machines (SVM). A Naïve Bayes text classifier uses joint probabilities of words and documents to estimate the probability of a given category. SVMs are binary classifiers that aim to discover the hyperplane that best separates a set of positive and negative examples with maximum allowed margin. The accuracy of the learning process is affected by the quantity of the training data; large amounts of training data have the potential of improving the overall accuracy. [Mihalcea and Strapparava 2005.]

The type of data used in experiments relating to humor recognition varies, but different types of *corpora* are useful sources for the datasets. A corpus is a collection of texts presented in a machine-readable format. These types of collections can include samples from both written and spoken language and they can be specialized in different ways. For example, a corpus might include texts written only in a particular dialect of a specific language (e.g. the British National Corpus (BNC) includes samples of British English while the Corpus of Contemporary American English (COCA) includes samples of American English). [Plag *et al.* 2015.] Datasets used in automatic humor recognition can also be collected, for example, with the aid of web crawlers [Ravi and Ravi 2017]. For example, Mihalcea and Strapparava [2005] collected their positive dataset (jokes consisting of a single line) from the internet using an automatic bootstrapping algorithm.

3.1. Advantages of Automatic Humor Recognition

As the design and deployment of conversational agents of various types is becoming more and more commonplace, it is important to acknowledge the increasing need for more natural interaction between the agents and the people conversing with them [Fung *et al.* 2016]. Since recognizing and responding to humor can be seen as a universal human trait [Raskin 2012], it is important that the conversational agents are equipped to deal with humorous exchanges of different varieties [Mihalcea and Pulman 2007]. Without the ability to detect and appropriately respond to humorous exchanges, such as witticisms, the interaction between the agent and the person interacting with it is bound to be unnatural [Fung *et al.* 2016].

People's apprehension towards physical robots has been a long-standing issue in robotics. Users prefer interacting with anthropomorphic machines but, although often extremely humanlike, anthropomorphic robots appear as cold and distant entities to humans. This distrust is partly because of the inability of robots to recognize and respond to human emotions and intent. Robots' ability to understand and react to high level communication features, such as humor, could help alleviate the sense of suspicion people feel towards humanlike robots. [Fung *et al.* 2016.]

Through accurate humor recognition, enterprises conducting business on the internet could potentially provide entertainment value for customers visiting their web page by recognizing and retrieving funny reviews given to their products [Pérez 2012]. However, not all funny reviews written on product pages are written with a positive meaning in mind. That is where automatic humor recognition could also provide value to companies through opinion mining. Successfully detecting irony in humorous reviews could provide valuable information for the business that could then be used, for example, to mend the company's possible negative image [Pérez 2012].

3.2. Problems Faced in Automatic Humor Recognition

As was pointed out by Ritchie [2001], computational humor is a relatively young and undeveloped field of study, and it is unclear whether it even has "any well-defined methodologies". Also, many implementations of automatic humor recognition are based on the incongruity theory and its different variations, but these theories are not without their problems. Hurley *et al.* [2011] criticize incongruity theories because they do not offer any explanation as to *why* an incongruity is found to be funny. The theories merely offer us a way to clas-

sify utterances as either humorous or nonhumorous. In addition, incongruity theoreticians have decidedly different ideas about what kinds of incongruities lead to humor [Hurley *et al.* 2011]. The fact that much of the research in computational humor is based on these theories alone could potentially yield problems in the future.

Another issue facing automatic humor recognition has to do with the nature of the research in the field. According to Ritchie [2001], much of the research has largely ignored important parts of human activity, such as emotion. People's emotions and motivations play a big role in the way they react to and understand humor, but many of the implementations of automatic humor recognition have neglected this fact in order to simplify the task of automatically recognizing humor [Hurley *et al.* 2011]. This propensity to simplify the task is criticized by both Ritchie [2001] and Hurley *et al.* [2011], because it artificially modularizes people's behavior and mental models. This may subsequently compromise the naturalness of the resulting model for humor recognition [Hurley *et al.* 2011].

In addition, much of the data used in automatic humor recognition research has used distinctly prototypical, formal examples of humorous language [Pérez 2012; Zhang 2014]. These types of examples falsely provide a very static view of verbal humor [Zhang 2014] even though one of the main characteristics of natural language is its tendency to continuously evolve [Pérez 2012].

Another aspect of humor that makes humor recognition challenging is that the use of humor is often built on a vast foundation of world knowledge [Ritchie 2001]. Take, for example, the seeing-eye dog joke in Section 2.1. Understanding the punchline requires us to know something specific about the world around us—outside the domain of language. In order to understand the joke, we must have prior knowledge about how blind individuals interact with the world around them. We cannot simply infer this crucial piece of information from the textual context of the punchline. The fact that understanding even a very short joke can sometimes require extensive amounts of external knowledge makes automatic humor recognition a difficult task to model for computers [Yang *et al.* 2015].

Furthermore, humorous utterances occurring in communicative situations are especially dependent on the context of the dialogue itself in addition to the background knowledge the participants have accumulated in past conversations. This makes automatically recognizing humorous exchanges in dialogue especially difficult when compared to, for example, distinguishing one-liners

from nonhumorous sentences. A humorous dialogue utterance is often preceded by a setup for the humor, whereas one-liners are not preceded by any preparation of the audience. In regular dialogue, the setup itself might not be humorous in nature and it could occur anywhere in the dialogue, not necessarily adjacent to the punchline, which can make automatically recognizing it very difficult. Without recognizing the setup of the punchline, however, many utterances lose their humorous meaning and are instead classified as nonhumorous instances of language. [Bertero and Fung 2016.]

Semantics also notoriously cause problems for humor recognition. In their research, Yang *et al.* [2015] found that phrase meaning was often the cause behind the classifier misclassifying a humorous sentence as nonhumorous, and vice versa. For example, the sentence “How does the earth get clean? It takes a meteor shower” was classified as nonhumorous, because *meteor shower* was not classified as one of the humor inducing elements of that sentence. Since its intended meaning was not accurately recognized, the combined comical effect of *earth*, *clean*, and *meteor shower* was completely lost upon the classifier. [Yang *et al.* 2015.]

4. Implementations of Automatic Humor Recognition

4.1. Detecting Humor Based on Stylistic Features

Mihalcea and Strapparava [2006] compared nonhumorous texts with humorous ones by looking for instances of alliteration, antonymy, and adult slang in the texts. The investigation was restricted to humor found in *one-liners* only, in order to make the task more feasible. A one-liner is a joke comprising of a single humorous sentence. The humorous effect in one-liners can be achieved through the use of creative language and different linguistic structures (e.g. the use of rhetoric devices such as alliteration or antonymy). One-liners are often employed in humor recognition tasks because of their relatively simple syntax in comparison with longer jokes. This makes them ideal for machine learning since the humorous features are “guaranteed to be present in the first (and only) sentence”. [Mihalcea and Strapparava 2006.]

The different stylistic features were identified by using different lexical resources. Instances of alliteration chains were automatically extracted by use of an index created on top of a pronunciation dictionary. Antonyms and instances of adult slang in the one-liners were identified using WordNet and WordNet Domains. [Mihalcea and Strapparava 2005.] WordNet is a lexical database of

English where words are organized into sets of cognitive synonyms (i.e. words that share at least one sense in common), also called *synsets*, each expressing a distinct, lexicalized concept [Miller 1995].

The experiment was formulated as a traditional text classification problem where humorous and nonhumorous examples were input into an automatic classifier. Mihalcea and Strapparava compared results obtained with two different classifiers: the multinomial model of Naïve Bayes and SVM. [Mihalcea and Strapparava 2006.]

The nonhumorous data set contained Reuters news titles, proverbs, sentences extracted from the BNC, and sentences extracted from the Open Mind Common Sense (OMCS) database. Both the humorous and nonhumorous examples had the same length restrictions so all sentences included in the datasets had an average length of 10–15 words, and they were similar in overall composition. This effectively ensures that the automatic classifier does not learn to distinguish between the different types of texts based on the length of the texts or their obvious differences in vocabulary. [Mihalcea and Strapparava 2006.]

The classification results had an accuracy ranging from 73.22% (comparing one-liners to BNC sentences) to 96.67% (comparing one-liners to Reuters news titles) using Naïve Bayes and 77.51% (comparing one-liners to BNC sentences) to 96.09% (comparing one-liners to Reuters news titles) using SVM. The high accuracy in comparing one-liners to Reuters news titles aligns itself with the finding that Reuters titles, when compared with one-liners, are significantly different in respect to their stylistic features. In all the datasets examined, alliteration as a stylistic feature was found to be the most useful indicator of humor. [Mihalcea and Strapparava 2005.]

4.2. Detecting Humor Based on Acoustic-prosodic Cues

Purandare *et al.* [2006] focused on analyzing and recognizing acoustic-prosodic cues (e.g. pitch, intensity, and tempo) instead of the more traditional stylistic features. The data used in the analysis (75 dialog excerpts) was extracted from the television show called Friends [Purandare *et al.* 2006]. Sitcoms offer an interesting realm for the task of automatically recognizing humor as they include a full dialogue setting paired with a laugh track that signals the location of the humorous utterances [Bertero and Fung 2016]. Purandare *et al.* [2006] were able to automatically label utterances in the dataset as either humorous or nonhumorous based on whether or not the utterance was followed by laughter.

The humor recognition task was formulated as a classical supervised learning problem by using a decision tree algorithm ADTree and running a 10-fold cross validation experiment on the 1629 speaker turns featured in the dataset. [Purandare *et al.* 2006.]

Purandare *et al.* [2006] found that, during the first iterations, the automatic classifier recognized humorous spoken turns in the dialogue more often based on the acoustic-prosodic features rather than the lexical features. This suggests that speech features, rather than lexical content, are a stronger indicator when it comes to discerning between humorous and nonhumorous utterances in speech. However, the automatic classifier did achieve its best results when acoustic-prosodic features were used in combination with lexical features. [Purandare *et al.* 2006.]

4.3. Detecting Humor Based on Linguistic and Affective Features

Zhang and Liu [2014] recognized humor found on Twitter by performing affect and sentimental analysis on tweets. Since the data was retrieved from social media, it contains more natural examples of informal humorous language instead of the more formal literary examples often used in previously conducted humor recognition tasks (e.g. Mihalcea and Strapparava 2005). [Zhang and Liu 2014.]

Zhang and Liu [2014] recognized several different humor-related features which could then be used in the automatic classification of humorous and non-humorous tweets. The humor-related features identified in the study included phonetic, morpho-syntactic, lexico-semantic, pragmatic features, and affective features. Affective features refer to words that are sentimentally loaded, such as *idiot*. [Zhang and Liu 2014.]

Altogether Zhang and Liu [2014] input fifty humor-related features into the Gradient Boosted Regression Tree (GBRT) model for humor recognition. GBRT is a boosting method with decision tree base learners and it was chosen for this task as it can handle heterogeneous features naturally and it outperformed both the SVM and Naïve Bayes classifiers in preliminary experiments conducted by Zhang and Liu. The classifier achieved, at best, an accuracy of 85% in automatically recognizing between humorous and nonhumorous tweets. [Zhang and Liu 2014.]

Zhang and Liu [2014] found that lexico-semantic and morpho-syntactic features are the strongest indicators for the presence of humor found on Twitter. Zhang and Liu also discovered that humor on Twitter has noticeable stylistic

tic features, such as sound patterns, that are not found in humorous non-tweets; these features are unique to humor found on Twitter. [Zhang and Liu 2014.]

4.4. Detecting Humor Based on Semantic Features

Yang *et al.* [2015] formulated a text classification task in which humorous sentences were distinguished from nonhumorous ones in terms of meaning incongruity, ambiguity, phonetic style, and personal affect. The ensemble learning method, Random Forest, was used to perform 10-fold cross-validation on the humorous and nonhumorous datasets. Unlike single decision trees that are prone to produce results with high variance and bias, random forests offer a way to find a more balanced result between two extremes using averaging. [Yang *et al.* 2015.]

Yang *et al.* [2015] also proposed a method for automatically extracting semantic units called *humor anchors* from humorous datasets. A humor anchor is a set of meaningful word spans that prompt humor in a sentence. [Yang *et al.* 2015.] The detection of humor anchors differentiates the study from other research in the field, as the experiment was not simply formulated as a binary classification task where sentences are classified as either humorous or nonhumorous. The anchor extraction worked well in identifying the salient semantic features that form the source of humor in the given data [Yang *et al.* 2015].

Both the task of recognizing humor and extracting humor anchors was found to be hindered by common issues in humor recognition—some of which were discussed earlier in Section 3.3—such as phrase meaning, external knowledge, and humor categorization. For example, the need for external knowledge caused problems with classifying jokes involving idioms and social phenomena. [Yang *et al.* 2015.]

5. Conclusion

Humor is, without a doubt, an essential part of people’s interpersonal communication everywhere in the world. As such, all people are familiar with it as a concept, but the task of creating a comprehensive, formal definition of humor has been proven difficult to accomplish. In our daily lives, humor is often quickly connected with laughter, but in a more formal context it is important to resist the temptation to define humor simply in relation to laughter. Humor is much more than something preceding a burst of laughter, and in the case of verbal humor we are able to identify some common features that can indicate

the presence of humor in a sentence or an utterance, such as human-centeredness of the vocabulary and alliteration.

As the interaction between people and different types of conversational agents becomes more prevalent, it is important for the agents to be able to recognize when the human users resort to humorous retorts. Automatic humor recognition is concerned with the creation and evaluation of such computational models to the recognition of humor. Humor expressed in verbal contexts can be difficult to automatically recognize for various reasons. The sometimes incredibly vast amounts of background knowledge needed for understanding a joke and the complex jungle of word meanings can significantly complicate the task for computers.

Although research in automatic humor recognition has not been completely void of problems, experiments in automatic humor recognition have seen success in distinguishing between humorous and nonhumorous instances of language as well as automatically detecting humor triggers by analysing the linguistic and extra-linguistic information surrounding utterances and sentences. Overall, the results have been highly encouraging and hold great promise for future work in automatic humor recognition as the amount of research in the field is continuously increasing.

References

- Salvatore Attardo. 1990. The violation of Grice's maxims in jokes. In: *Proc. of the Annual Meeting of the Berkeley Linguistics Society*, 355–362.
- Salvatore Attardo. 2010. *Linguistic Theories of Humor. Vol. 1*. Walter de Gruyter.
- Salvatore Attardo. 2014. *Encyclopedia of Humor Studies*. Sage Publications.
- Dario Bertero and Pascale Fung. 2016. A long short-term memory framework for predicting humor in dialogues. In: *Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 130–135.
- Pascale Fung, Dario Bertero, Yan Wan, Anik Dey, Ricky Ho Yin Chan, Farhad Bin Siddique, Yang Yang, Chien-Sheng Wu, and Ruixi Lin. 2016. Towards empathetic human-robot interactions. arXiv:1605.04072.
- Christian Hempelmann, Victor Raskin, and Katrina E. Triezenberg. 2006. Computer, tell me a joke... but please make it funny: Computational humor with ontological semantics. In: *Proc. of the FLAIRS Conference*, 746–751.
- Matthew M. Hurley, Daniel C. Dennett, and Reginald B. Adams. 2011. *Inside Jokes: Using Humor to Reverse-Engineer the Mind*. MIT Press.

- Rada Mihalcea and Stephen Pulman. 2007. Characterizing humour: An exploration of features in humorous texts. In: *Proc. of the International Conference on Intelligent Text Processing and Computational Linguistics*, 337–347.
- Rada Mihalcea and Carlo Strapparava. 2005. Making computers laugh: Investigations in automatic humor recognition. In: *Proc. of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 531–538.
- Rada Mihalcea and Carlo Strapparava. 2006. Learning to laugh (automatically): Computational models for humor recognition. *Computational Intelligence* 22, 2, 126–142.
- George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM* 38, 11, 39–41.
- Martin Montgomery, Alan Durant, Nigel Fabb, Tom Furniss, and Sara Mills. 2007. *Ways of Reading: Advanced Reading Skills for Students of English Literature*. Routledge.
- Ingo Plag, Sabine Arndt-Lappe, Maria Braun, and Mareile Schramm. 2015. *Introduction to English Linguistics*. Walter de Gruyter.
- Amruta Purandare and Diane Litman. 2006. Humor: Prosody analysis and automatic recognition for F* R* I* E* N* D* S. In: *Proc. of the 2006 Conference on Empirical Methods in Natural Language Processing*, 208–215.
- Antonio R. Pérez. 2012. *Linguistic-based patterns for figurative language processing: the case of humor recognition and irony detection*. Ph. D. Dissertation, Department of Information Systems and Computation, Polytechnic University of Valencia.
- Victor Raskin. 2012. *Semantic mechanisms of Humor*. Vol. 24. Springer.
- Kumar Ravi and Vadlamani Ravi. 2017. A novel automatic satire and irony detection using ensembled feature selection and data mining. *Knowledge-Based Systems* 120, 15–33.
- Graeme Ritchie. 2001. Current directions in computational humour. *Artificial Intelligence Review* 16, 2, 119–135.
- Julia M. Taylor and Lawrence J. Mazlack. 2004. Computationally recognizing wordplay in jokes. In: *Proc. of the Annual Meeting of the Cognitive Science Society* 26, 26, 1315–1320.
- Robert S. Wyer, and James E. Collins. 1992. A theory of humor elicitation. *Psychological Review* 99, 4, 663–688.

- Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. 2015. Humor recognition and humor anchor extraction. In: *Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2367–2376.
- Renxian Zhang and Naishi Liu. 2014. Recognizing humor on Twitter. In: *Proc. of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 889–898.

Nudge ja retoriikka - käyttäjän toiminnan ohjailu verkkopalveluissa

Miklos Strömberg

Tiivistelmä.

Verkkopalvelut yrittävät jatkuvasti vaikuttaa käyttäjän toimintaan. Tässä tutkielmassa tarkastellaan aikaisemman kirjallisuuden pohjalta sitä, kuinka käyttäjiä ohjailaan verkkopalveluissa nudgen ja retoriikan avulla. Käyttäytymistieteistä, poliittisesta filosofiasta ja taloustieteistä peräisin oleva nudge-konsepti on mikä tahansa valinta-arkkitehtuurin osa, jonka avulla käyttäjän toimintaan voidaan vaikuttaa ennustettavalla tavalla. Nudgen toiminta perustuu vapaaehtoisuuteen, käyttäjää ei ohjailta kielloilla tai rajoitteilla. Digitaalinen nudge toteutetaan verkkopalveluissa käyttöliittymäelementtien avulla. Tutkielmassa perehdytään digitaalisen nudgen toteutuksen lisäksi myös niiden suunnitteluun. Verkkopalvelut käyttävät retoriikan vaikuttamisen keinoja saavuttaakseen tavoitteensa. Retoriikka ilmenee verkkopalveluissa niiden informaationvälityksen, ulkoasun tai käytettävyyden kautta – toisaalta myös nudgen taustalla voi vaikuttaa (epäsuora) retorinen argumentointi. Retoriikan avulla vahvistetaan koko palvelun uskottavuutta tai käyttäjän tarvetta toimia palvelussa, kun taas nudge (tai useamman nudgen samanaikainen käyttö) liittyy aina yksittäiseen valintatilanteeseen.

Avainsanat ja -sanonnat: Vuorovaikutteinen teknologia, nudge, retoriikka, käyttäjän toiminnan ohjailu, suostutteleva teknologia.

1. Johdanto

Verkkopalvelut yrittävät vaikuttaa käyttäjän toimintaan monin tavoin. Tarve käyttäjän ohjailulle voi muodostua verkkopalvelun kaupallisista päämääristä tai halusta sitouttaa käyttäjä palveluun. Toisinaan päämääränä voi olla palvelun käyttökokemuksen parantaminen – käyttäjää voidaan opastaa palvelun käyttöönotossa tai hänelle voidaan esitellä palvelun uusia ominaisuuksia. Tässä tutkielmassa keskitytään tarkastelemaan verkkopalveluiden käyttämää *retoriikkaa ja nudgeja*, joilla verkkopalvelu voi ohjata käyttäjää kohti haluttua toimintaa. Nudge ja retoriikka kuuluvat tutkielman kontekstissa *suostuttelevaan teknologiaan* (persuasive technology), jossa käyttäjien asenteita ja käyttäytymistä pyritään ohjaamaan suostuttelulla ja *sosiaalisella vaikuttamisella* (social influence), mutta heitä ei pakoteta tietynlaisiin toimintamalleihin.

Englanninkielinen sana nudge tarkoittaa tönäisyä, tökkäystä tai varovais- ta suostuttelua. Arkikielessä sillä viitataan useimmiten toimintaan, jossa henki-

lö tönäisee hellästi (esimerkiksi kyynärpäällä) toista henkilöä kiinnittääkseen hänen huomionsa johonkin ympäristössä tapahtuvaan asiaan. Nudgella pyritään ohjailemaan ihmisiä epäsuorasti kohti toivottua käyttäytymistä, ilman että muita vaihtoehtoja kielletään tai rajoitetaan.

Käyttäytymistaloustieteilijät Thaler ja Sunstein [2008] popularisoivat nudge-termin kirjassaan *Nudge: improving decisions about health, wealth, and happiness*. He määrittelevät nudgen tarkoittavan mitä tahansa valinta-arkkitehtuurin osaa, joka vaikuttaa ennustettavalla tavalla ihmisten käyttäytymiseen, ilman muiden vaihtoehtojen kieltämistä tai taloudellisten kannustimien merkittävää muuttamista. Kouluruokala voi ohjata oppilaita kohti terveellisempiä ruokailutottumuksia esimerkiksi asettamalla linjastoon hedelmiä silmien korkeudelle [Thaler and Sunstein 2008].

Nudgen käsitteen kuvauksen jälkeen luvussa 2 esitellään yleisimpiä digitaalisissa toimintaympäristöissä (verkkosivuilla) esiintyviä nudgeja. Esittelyjen yhteydessä kuvataan lyhyesti nudgen toiminnan taustalla vaikuttavia psykologisia ilmiöitä ja pyritään antamaan konkreettinen esimerkki kyseisen nudgen digitaalisesta toteutuksesta.

Luvussa 3 tarkastellaan verkkopalveluiden retoriikkaa. Tutkielmassa esitellään Aristoteleen retoriikan kolme vaikuttamisen keinoa (*logos, paatos ja eetos*) ja pohditaan miten ne ilmenevät verkkopalveluissa. Vaikuttamisen keinojen lisäksi esitellään myös niihin liittyvä ajallinen komponentti *kairos*, joka toimii vuorovaikutuksessa muiden suostuttelun välineiden kanssa.

Retoriikan suostuttelun välineiden tunnistaminen verkkopalveluissa ei aina ole yksioikoista. Kun eetos tai paatoksellinen retoriikka on usein hyvinkin suoraviivaista ja tunnistettavaa, voi logos ilmetä abstraktimmalla tasolla, palvelun käytettävyyteen tai sen tarjoamaan informaatioisisältöön sidottuna.

Tässä tutkielmassa ei pohdita syvemmin käyttäjien motivaatiota. On kuitenkin syytä olettaa, että verkkopalveluiden käyttöön vaikuttavat palvelusta riippumattomat tekijät, kuten käyttäjän tarpeet tai mielihalut. Verkkopalveluiden retoriikan vaikutusta käsitellään olettaen, että käyttäjillä on jokin motivaatio tai kannustin kyseisen palvelun käyttöön. Voidaan myös olettaa, ettei kaupallinen verkkopalvelu pysty retoriikallaan suostuttelemaan siihen täysin kielteisesti suhtautuvaa käyttäjää. Kaupallisten verkkopalveluiden (kuten verkkokauppojen) retoriikalla ja nudgeilla pyritäänkin siihen, että käyttäjän motivaatio muuttuu konkreettiseksi toiminnaksi.

Luvussa 4 käsitellään sitä, kuinka nudge toteutetaan digitaalisessa ympäristössä. Nelivaiheinen suunnitteluprosessi koostuu tavoitteen ja kohderyhmän määrittelystä, sekä kontekstiin sopivan nudgen valinnasta ja testauksesta. Tut-

kielman loppuksi luvussa 5 pohditaan nudgen ja retoriikan suhdetta ja luvussa 6 tehdään yhteenveto tutkielman keskeisimmästä sisällöstä.

2. Nudge

Englanninkieliselle termille nudge ei ole vielä vakiintunutta suomennosta. Käytänkin tutkielmassani alkukielistä termiä, sillä sen kääntämisestä ei olisi merkittävää hyötyä. Suomenkieliset käänösvaihtoehdot eivät onnistu kuvaamaan käsitettä ja ne kuulostavat kontekstiin nähden epäsopivilta. Toisaalta on jokseenkin yleistä jättää käänös tekemättä, varsinkin jos alkukielinen termi on lyhyt ja iskevä.

Nudge on alun perin käyttäytymistieteistä, poliittisesta filosofiasta ja taloustieteistä peräisin oleva konsepti, jolla kuvataan epäsuoraa vaikuttamista yksilön tai ryhmän päätöksentekoon. Konseptille on olennaista, että vaikutukselle alistuminen perustuu vapaaehtoisuuteen. Yksilön tai ryhmän motiiveihin, kannustimiin ja päätöksentekoon yritetään vaikuttaa *positiivisen vahvistamisen* (positive reinforcement) ja epäsuoran viittaamisen keinoin. Jotta vaikuttamista voidaan laskea nudgeksi, sen täytyy olla helposti vältettävissä [Thaler and Sunstein 2008].

Valinta-arkkitehtuuri (choice architecture) [Thaler and Sunstein 2008] kuvaa keinoja, joilla voidaan esittää erilaisia vaihtoehtoja, sekä näiden esittämistapojen vaikutusta päätöksentekoon. Esimerkiksi vaihtoehtojen lukumäärä, järjestys, vaihtoehtojen ominaisuuksien esittämistapa ja *oletusvaihtoehdon* (default) esivalinta ovat tekijöitä, jotka vaikuttavat päätöstä tehtäessä [Johnson *et al.* 2012].

Valinta-arkkitehtuureja voidaan suunnitella käyttämällä tarkoituksenmukaisesti siihen soveltuvia nudgeja (nudging) [Mirsch *et al.* 2017]. Nudgen vaikutuksesta ihminen tekee tietyn valinnan tai käyttäytyy tietyllä tavalla. Vaikutus saavutetaan muokkaamalla ympäristöä siten, että ihmisen automaattiset kognitiiviset prosessit virittyvät puoltamaan haluttua lopputulosta. [Saghai 2013].

Vaikka valinta-arkkitehtuuriin voidaan vaikuttaa nudgeja käyttämällä, on tärkeää ymmärtää, ettei valintavaihtoehtoja voi esittää neutraalisti. Kaikki valintavaihtoehtojen esitystavat vaikuttavat valinnan tekijän päätökseen, sillä ne sisältävät vähintään implisiittisen oletusvaihtoehdon. Joskus oletusvaihtoehto on se, ettei mitään valintaa tehdä, vaan pysytään *vallitsevassa tilassa* (status quo). Tämä implisiittinen oletusvaihtoehto valitaan useammin, vaikka jokin toinen vaihtoehto esiteltäisiin oletuksena. [Johnson *et al.* 2012]

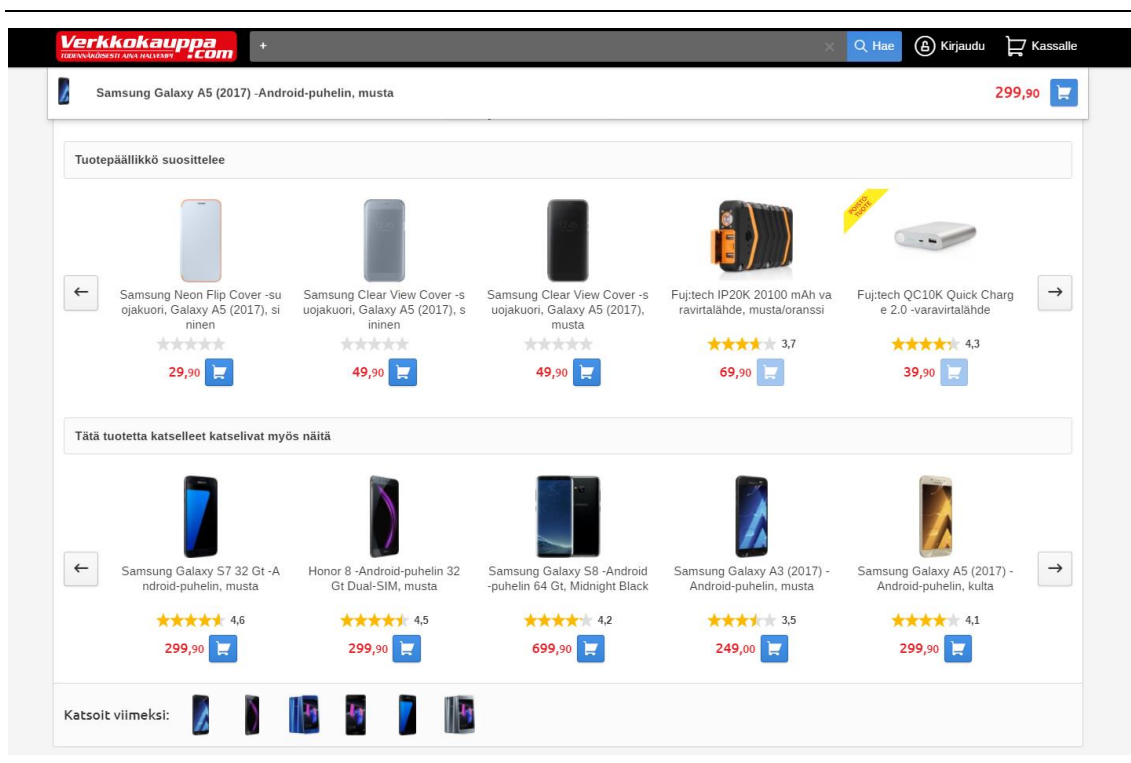
Valintavaihtoehtojen neutraalin esittämisen mahdottomuus pätee myös verkkoympäristöissä. Jokaista käyttöliittymää voidaankin pitää digitaalisena valinta-arkkitehtuurina. Nudge voidaan suunnitella käyttämään käyttöliitty-

män visuaalisia elementtejä ihmisten käyttäytymisen ohjaajana. [Weinmann *et al.* 2016.]

Esittelen seuraavaksi Mirschin ja muiden [2017] systemaattisessa kirjallisuuskatsauksessa tunnistettuja, yleisimmin esiintyviä nudgeja. Näillä nudgeilla on usein kerrannaisvaikutusta, eivätkä ne perustu yhteen tiettyyn psykologiseen ilmiöön [Mirsch *et al.* 2017]. Kuvaan jokaisen nudgen yhteydessä konkreettisen esimerkin siitä, kuinka se ilmenee erilaisissa verkkopalveluissa. Usein osuvimmat esimerkit löytyvät kaupallisten toimijoiden palveluista, mikä selitynee suurimmilta osin sillä, että käyttäjän toiminnan ohjaamisella (kohti ostopäätöstä) on suora vaikutus yritystoiminnan kannattavuuteen.

Kehystys (Framing). Verkkokaupat, kuten Amazon.com, nostavat tuotesivuilla esiin kyseiseen tuotteeseen liittyviä muita tuotteita. Valinta-arkkitehtuuria muokataan kiinnittämällä käyttäjän huomio näihin lisätuotteisiin. Tämä korostaminen saattaa aiheuttaa lisäostoksia, joita käyttäjä ei alun perin suunnitellut tekevänsä. [Mirsch *et al.* 2017]

Kotimaisessa verkkokauppa.com -palvelussa kehystys ilmenee tuotesivun loppuun lisättyssä osiossa ”Tuotepäällikkö suosittelee” (kuva 1).



Kuva 1. Kehystykseen ja sosiaalisiin normeihin liittyvät nudget toteutettuna verkkokauppa.com -sivustolla. (Haettu 24.3.2018)

Vallitsevan tilan suosiminen (Status Quo Bias). Vallitsevan tilan suosimisella viitataan ihmisten vahvaan taipumukseen pysytellä vallitsevassa tilassa, kun tilan muuttamisesta saavutettavat edut epäilyttävät enemmän kuin tilaan jäämisestä seuraavat haitat. Digitaalisessa ympäristössä vallitsevan tilan suosiminen liittyy palvelun tarjoamiin oletusvaihtoehtoihin. Esimerkiksi matkailualan yritysten sivustolla asiakkaalle lisäkustannuksia tuovat matkavakuutukset ovat usein esivalittuina, eli oletusvaihtoehtona. [Mirsch *et al.* 2017]

Sosiaaliset normit (Social norms). Ihmisillä on tapana hakea mallia tai *sosiaalista vahvistusta* (social proof) muiden ihmisten käyttäytymisestä, jos he eivät itse hallitse tilanteen vaatimaa oikeanlaista käyttäytymismallia. Verkkokaupat toteuttavat tätä nudgea samalla tavalla kuin kehystystä. Verkkokaupoissa asiakkaalle esitellään tuotteen oston yhteydessä lisää tuotteita, joita muut palvelun käyttäjät ovat ostaneet kyseisen tuotteen kanssa. Joukko muita palvelun käyttäjiä luo mallin tiettyyn tuotteeseen liittyvästä ostoskäyttäytymisestä. Yksittäinen käyttäjä saattaa noudattaa tätä mallia ottaen tietoisesti huomioon muiden tavan toimia. [Mirsch *et al.* 2017.] Verkkokauppa.com -palvelussa tuotesivulla käyttäjään pyritään vaikuttamaan osiossa ”Tätä tuotetta katselleet katselivat myös näitä” (kuva 1).

Tappiokammo (Loss aversion). Tappiokammo perustuu oletukseen, että tappiolla ja haitoilla on valintatilanteessa enemmän painoarvoa kuin saavutettavilla hyödyillä tai voitoilla. Verkkokaupassa, kuten erityisesti matkailualan yritysten sivustoilla, käyttäjälle esitetään usein tuotteiden yhteydessä tietoa tuotteen suosiosta tai sen saatavuuden rajoitteista (”Varattu 36 kertaa tänään!”, ”50 % tarjouksessa VAIN tämän päivän ajan”, ”8 ihmistä katselee tätä tuotetta nyt”). Asiakas saattaa kokea menettävänsä löytämänsä mahdollisuuden, jos hän ei toimi nopeasti. Tappiokammon avulla pyritään ohjamaan käyttäjää tekemään ostopäätös tai lyhentämään siihen kuluvaan aikaa. [Mirsch *et al.* 2017]

Kuvassa 2 esitetään tappiokammon konkreettinen toteutus. Käyttäjää pyritään ohjaamaan kohti ostopäätöstä esittämällä tiettyjen lentojen rajallinen saatavuus.

FINNAIR ALOITA ALUSTA VALITSE LENNTO HINTA MATKUSTAJATIEDOT LISÄPALVELUT MAKSAMINEN VAHVISTUS

VALITSE LENTO
Löydät lisätietoa hinnoista ja mahdollisista lisämaksuista sivun alaosasta.

MENO
Helsinki - New York
perjantai 01 kesäkuu 2018 **VAIHTA PÄIVIÄ**

MISTÄ → MIHIN

	ECONOMY CAMPAIGN	ECONOMY SAVER	ECONOMY PREMIUM SAVER	ECONOMY FLEXIBLE	ECONOMY PREMIUM
14:10 → 15:50 Suora lento Kokonaiskesto: 08 t 40 min Avaa lisätiedot klikkaamalla > Liikennöijä: Finnair	313.58 EUR	425.58 EUR	N/A	1141.58 EUR	N/A
08:10 → 12:50 1 Lennon vaihto Kokonaiskesto: 11 t 40 min Avaa lisätiedot klikkaamalla > Liikennöijä: Norra, American Airlines	N/A	445.47 EUR	N/A	1411.47 EUR	N/A

1. JÄLJELLÄ 7. JÄLJELLÄ

Kuva 2. Helsingistä New Yorkiin lähtevien lentojen yhteyteen lisätty teksti paikkojen rajoitetusta määrästä finnair.fi -sivustolla (Haettu 24.3.2018).

Liikadiskonttaus (Hyberbolic Discounting). Liikadiskonttauksen konsepti tarkoittaa sitä, että ihmiset käyttäytyvät epäjohdonmukaisesti suhteessa aikaan, arvostaen nykyhetkeä ja lähitulevaisuutta kaukaisempaa tulevaisuutta enemmän. Näin ollen ihmiset suosivat vaihtoehtoja, joilla on vaikutusta nykyhetkeen, vaikka tulevaisuuteen vaikuttavalla vaihtoehdolla olisi suurempi tai parempi kokonaisvaikutus. Joissain autolainaamojen verkkokaupoissa on mahdollisuus tehdä varaus hieman alennettuun hintaan, jos sen maksaa jo sitä varatessa. Liikadiskonttaus ohjaa asiakkaan tekemään välittömän ostopäätöksen tarjoamalla taloudellista hyötyä. [Mirsch *et al.* 2017]

Miellettävyysheuristiikka (Availability Heuristic). Ihmisillä on taipumus arvioida asioiden tapahtumisen todennäköisyyttä sen perusteella, kuinka helposti ne ovat muistettavissa. Helposti muistettavat ja säännöllisesti toistuvat tapahtumat koetaan todennäköisemmiksi kuin sellaiset, joista on enemmän aikaa – tapahtumien todellisesta todennäköisyydestä huolimatta. Esimerkiksi mediahuomion seurauksena yksilö voi mieltää lentokoneturmien todennäköisyyden huomattavasti todellista suuremmaksi, sillä näitä tapauksia nostetaan esiin melko säännöllisesti uutisoinnissa. Miellettävyysheuristiikkaa sovelletaan verkossa tapahtuvissa mainosbannerikampanjoissa. Googlen tarjoamia digitaalisia mainostiloja käyttävät mainostajat voivat esittää mainosbannereitaan valikoivasti. Koska käyttäjän verkkoselaamista jäljitetään, voidaan sama mainos esittää toistuvasti useilla peräkkäisillä sivustoilla. Valintatilanteessa mainoksen

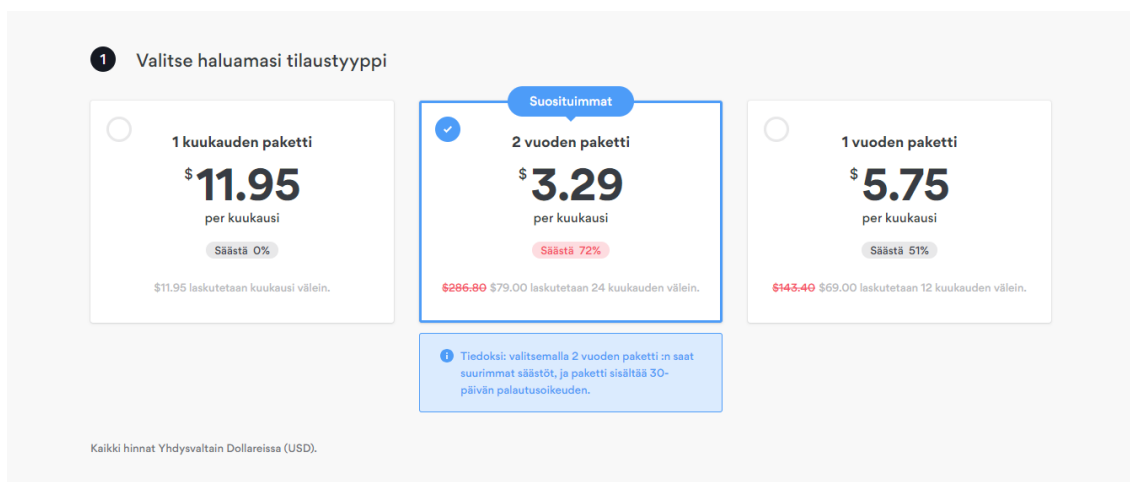
viesti on tuoreeltaan käyttäjän mielessä ja se saattaa ohjata häntä kohti mainostajan kannalta edullista päätöstä. [Mirsch *et al.* 2017]

Arvion ankkurointi (Anchoring and Adjustment). Kun yksilöiltä puuttuu päätökseen tarvittavaa informaatiota, he luovat oletusarvion käyttämällä joitain tiettyjä arvoja ankkureina. Nämä ankkuroitavat arvot voivat muodostua valintatilanteen tarjoamasta tiedosta tai perustua käyttäjän omaan karkeaan arvioon. Tähän liittyen päätöksen tekemiseen vaikuttaa se, millaiseksi tämä aloitusarvio koetaan. Verkkokaupoissa samasta elektroniikkatuotteesta löytyy useita erilaisia malleja, joiden toimintakapasiteetti ja ominaisuudet vaihtelevat. Jos kolme tällaista mallia esitetään samaan aikaan, toimivat halvimman ja kalleimman mallin hinnat ankkureina, joka voi johtaa siihen, että käyttäjä luo oletusarvion tuotteen keskimääräisestä hinnasta perustuen vain näihin kahteen ankkuriin. [Mirsch *et al.* 2017]

Virittäminen (Priming). Käyttäjä voidaan valmistaa valintatilanteeseen esittämällä hänelle tietynlaisia kysymyksiä tai informaatiota. Käyttäjän voi myös virittää valintatilannetta varten visualisoimalla päätöksen seuraamuksia, ohjaamalla häntä kohti haluttua valintaa. Lentoyhtiöiden kotisivuilla käyttäjiä viritetään visualisoimalla mahdollisten päätösten seuraamuksia, eli tässä tapauksessa heille näytetään houkuttelevia kuvia kaukomaisten matkakohteista. Kuvat voivat ohjata käyttäjää kohti tiettyä matkakohdetta tai herättää ajatuksen matkailusta ylipäätään. [Mirsch *et al.* 2017]

Harhautusefekt (Decoy Effect). Harhautusefektissä vaihtoehdon houkuttelevuutta lisätään esittämällä sen rinnalla todella epähoukutteleva vaihtoehto [Schneider *et al.* 2017]. Joukkorahoituskampanjan ohessa toteutetussa tutkimuksessa [Tietz *et al.* 2016] havaittiin muutos testihenkilöiden valinnoissa, kun harhautuksena toimiva vaihtoehto lisättiin vaihtoehdoksi. Rahoittajilla oli aluksi valittavana kaksi vaihtoehtoa, a) e-kirja 10 dollarin lahjoitusta vastaan, tai b) e-kirja ja painettu kirja 20 dollarin lahjoitusta vastaan. Noin kaksi kolmasosaa rahoittajista päätyi halvempaan vaihtoehtoon. Kun vaihtoehdoksi lisättiin harhautusvaihtoehto, eli c) painettu kirja 20 dollarin lahjoitusta vastaan, päätyi kaksi kolmasosaa valitsemaan vaihtoehdon b). Harhautusvaihtoehto siis lisäsi aiemmin vähemmän suosittua vaihtoehtoa.

Kuvassa 3 esitetään harhautusefektin ja monen tutkielmassa esitellyn nudgen rinnakkainen toteutus nordvpn.com -sivustolla. Usean nudgen yhtäaikaishallinnalla käytöllä voidaan saavuttaa kerrannaisvaikutus käyttäjän toiminnan ohjailussa.



Kuva 3. Valintatilanteessa voi olla yhtäaikaaisesti monta nudgea. Tästä kuvasta löytyy elementtejä harhautusefektistä, tappiokammosta, vallitsevan tilan suosimisesta ja sosiaalisista normeista nordvpn.com -sivustolla. (Haettu 23.4.2018)

Toimintakutsu (Call to Action). Toimintakutsun avulla yritetään kiinnittää käyttäjän huomio ja ohjata tätä kohti haluttua toimintaa. Käyttöliittymässä toimintakutsu ilmenee yksittäisenä selkeästi erottuvana elementtinä. Se toteutetaan yleisimmin painikkeena. [Gube 2009]

Toimintakutsu on yleensä ainoa interaktion mahdollistava elementti osiossaan ja sitä korostetaan usein käyttämällä painikkeessa sivuston muusta väri-maailmasta poikkeavaa väritystä. Toimintakutsua käytetään yleisesti verkkopalveluissa esimerkiksi silloin, kun halutaan saada käyttäjä rekisteröitymään palveluun (kuva 4) tai ohjataan käyttäjää kohti palvelun ydintoimintaa.

facebook

Sähköposti tai puhelin Salasana Kirjautu sisään

Unohditko käyttäjätun?

Facebookin avulla pidät yhteyttä elämäsi ihmisiin.

Rekisteröidy

Se on ilmaista nyt ja aina.

Etunimi Sukunimi

Matkapuhelinnumero tai sähköpostiosoite

Uusi salasana

Syntymäaika

3 huhti 1993 Miksi minun pitää kertoa syntymäaikani?

Nainen Mies

Välittämällä Luo tili hyväksyt käyttöehdot ja ilmaiset lukeneesi tietokäytännön, mukaan lukien evästeiden käyttöä koskevan käytännön. Voit saada ilmoituksia tekstiviestinä, ja voit lopettaa ilmoitukset milloin tahansa.

Luo tili

Luo sivu julkaisulle, yhtyeelle tai yritykselle.

Kuva 4. Facebook ohjaa toimintakutsun avulla käyttäjää rekisteröitymään palveluun facebook.com -sivustolla (Haettu 3.4.2018)

3. Retoriikka

Klassisen filosofian käsite retoriikka merkitsee pyrkimystä vaikuttaa kuulijan ajatteluun ja tekoihin. Aristoteleen retoriikka koostuu kolmesta vaikuttamisen keinosta. Logos vetoaa järkeen. Sen vahvuus perustuu argumentin selkeyteen ja johdonmukaisuuteen, sekä väitettä tukevaan todistusaineistoon. Paatos vetoaa tunteisiin. Paatoksella vedotaan kuulijan tunteisiin, uskomuksiin, arvoihin tai mielikuvitukseen. Pyrkimyksenä on herättää kuulijassa sympatiaa tai empatiaa. Eetos vetoaa moraaliin. Eetos koostuu uskottavuudesta, legitimitetistä ja auktoriteetista, jota puhuja luo tai tuo esiin argumentoinnin yhteydessä. Logos, paatos ja eetos kaikki yhdessä vaikuttavat minkä tahansa kommunikaation suostuttelun tehoon. [Van de Ven and Johnson 2006]

Näiden kolmen vaikuttamisen keinon lisäksi on syytä esitellä neljäs Aristoteleen retoriikkaan liittyvä konsepti. Kairos voidaan määritellä oikeana hetkenä toimia: sillä voidaan kuvata rajoitettua aikaikkunaa, joka ei välttämättä toistu uudelleen [Kinneavy and Eskin 2000]. Kairos on siis ajallinen elementti, joka toimii muiden vaikuttamisen keinojen rinnalla. Tietyissä mielessä sen painoarvo ylittää muut vaikuttamisen keinot – taidokkaallakaan retoriikan käytöllä ei ole tehoa, jos ajoitus on täysin väärä [Rife 2010].

3.1. Logos verkkopalveluissa

Verkkopalvelut vetoavat loogisuuteen taloudellisten seikkojen, saatavilla olevan informaation ja palvelun käytettävyyden kautta [Winn and Beck 2002]. Verkkokauppa voi yrittää vaikuttaa käyttäjiin esimerkiksi korostamalla alennetun tuotteen hintaeroa suositushintaan tai painottamalla kanta-asiakasetuja. Palvelu voi myös mainita palkkioista, joita se jakaa kanta-asiakkaille tai käyttäjille jotka tekevät suurempia ostoksia yhdellä kerralla. [Chu *et al.* 2014; Winn and Beck 2002.]

Verkkokaupassa tuotetiedoilla ja niiden esitystavalla on merkittävä vaikutus. Yleisellä tasolla voidaan sanoa, että yksityiskohtaiset tuotetiedot lisäävät käyttäjän kiinnostusta tuotetta kohtaan. Tarvetta tarkemmille tuotekuvauksille ilmenee erityisesti kalliimpien tai käyttäjille entuudestaan tuntemattomien tuotteiden kohdalla. [Chu *et al.* 2014; Flanagin *et al.* 2014; Winn and Beck 2002.]

Loogisuuteen voidaan vedota hyvällä käytettävyydellä. Verkkokaupassa tärkeitä ominaisuuksia ovat tuotteiden onnistunut kategorisointi, toimiva hakutoiminto ja käyttäjän mahdollisuus lajitella tuotteita eri ominaisuuksien mukaan [Chu *et al.* 2014; Winn and Beck 2002].

Helppokäyttöisyys on kriittistä myös tapahtumaketjuille, kuten palveluun rekisteröitymiselle ja maksusuoritukselle. Virhe missä tahansa tapahtumaketjun osassa voi merkitä palvelulle asiakkaan menettämistä. [Chu *et al.* 2014; Winn and Beck 2002.]

3.2. Paatos verkkopalveluissa

Ihmiset hakevat verkkokaupoissa tukea päätöksentekoon muiden ihmisten käyttäytymisestä. Tuotteet, joita tuodaan esiin otsikolla ”Muut ihmiset ostivat (tai katselivat) näitä tuotteita”, herättävät käyttäjissä enemmän mielenkiintoa kuin ”Kuumat tarjoukset” tai verkkokaupan muut erikoistarjoukset [Chu *et al.* 2014; Winn and Beck 2002].

Muiden käyttäjien tekemillä arvosteluilla on myös merkittävä rooli käyttäjien mielipiteisiin. Tutkimus viittaa siihen, että tuote-arvosteluilla on positiivinen korrelaatio käyttäjän ostoaikeisiin ja mielikuvaan tuotteen laadusta. Toisaalta käyttäjät antavat suurta painoarvoa arvostelujen keskimääräiselle arvosanalle, mutta eivät ota huomioon arvostelujen (mahdollisesti vähäistä) lukumäärää tai näiden kahden tekijän yhteisvaikutusta. [Flanagin *et al.* 2014.]

Halulla tarkastella muiden ihmisten toimia on suuri vaikutus verkkopalveluiden käytössä. *Paitsi jäämisen pelko* (Fear of Missing Out) on ilmiö, jota esiintyy erityisesti sosiaalisen median palveluissa. Ilmiö perustuu siihen, että käyttäjillä on jatkuva tunne siitä, että muut kokevat palkitsevia kokemuksia, joista he itse jäävät paitsi. Paitsi jäämisen pelolla selitetään syitä sille, miksi ihmiset käyt-

tävät paljon aikaa sosiaalisen median palveluissa ja aktiivisesti käyvät läpi niiden tarjoamaa vaihtuvaa sisältöä. [Przybylski *et al.* 2013.] Tätä ilmiötä voidaan käyttää hyväksi kaupallisen verkkopalvelun retoriikassa. Ihmiselle voidaan yrittää luoda sellainen vaikutelma, että hän jää ikään kuin ulkopuoliseksi, jos hän ei sitoudu palvelun käyttäjäksi.

Verkkokaupat yrittävät nopeuttaa käyttäjien ostopäätöstä rajoittamalla myytävän artikkelin kappalemäärää tai ilmoittamalla nopeasti umpeutuvan aikarajan tarjoushinnalle. Tällaista vaikuttamista kutsutaan taloustieteissä termillä *niukkuus* (scarcity) ja aiemmin esitelty tappiokammon nudge voidaan rakentaa tällä tavalla. Chu ja muut [2014] esittävät, että niukkuudella vedotaan tunteisiin. Mielestäni niukkuuden voi kuitenkin liittää myös kairokseen. Sen avulla voidaan luoda keinotekoinen hetkellinen mahdollisuus tai aikaikkuna, jonka puitteissa käyttäjä voi kokea tarpeelliseksi toimia. Tunteisiin vedotaan verkkopalveluissa myös sivuston ulkoasun ja käyttäjäkohtaisen *räätälöinnin* kautta. Tuotteita voidaan lajitella esimerkiksi käyttäjän luonteenpiirteiden, eikä tuotteiden ominaisuuksien mukaan. [Chu *et al.* 2014.] Räätälöinti tekee verkkopalvelun käytöstä henkilökohtaisemman kokemuksen, joka vahvistaa käyttäjän halua sitoutua palveluun [Oinas-Kukkonen and Harjuma 2009].

3.3. Eetos verkkopalveluissa

Verkkopalveluissa eetosta vahvistetaan luomalla kuvaa uskottavasta ja luotettavasta toimijasta. Tähän vaikuttaa erityisesti palvelun ulkoasu, jonka tulee olla ammattitaitoisesti toteutettu. Luotettavuutta voidaan vahvistaa vakuuttamalla käyttäjä palvelun tietosuojakäytännöistä ja henkilökohtaisen informaation käsittelytavoista. Luotettavuutta voidaan myös luoda korostamalla sivustolla tunnettujen toimijoiden tai tuotemerkkien logoja. [Chu *et al.* 2014; Winn and Beck 2002.] Foggin ja muiden [2003] mukaan käyttäjät arvoivat palvelun uskottavuutta erityisesti ulkoasun ja informaation rakenteen ja sen fokuksen perusteella.

Uskottavuutta voidaan vahvistaa käyttämällä hyväksi jonkin toisen toimijan auktoriteettia tai uskottavuutta. *Kolmannen osapuolen hyväksynnällä* (Third party endorsements) viitataan siihen, että palvelu esittää todisteen riippumattoman toimijan kriteerien täyttymisestä palvelussa – verkkopalveluissa tämä todiste on usein esimerkiksi sertifikaatin sinetti, joka toimii takuuna palvelun laadusta [Chu *et al.* 2014].

Kolmannen osapuolen hyväksyntää voidaan toteuttaa esittämällä käyttäjälle todisteita tunnettujen toimijoiden tyytyväisyydestä palveluun, kuten kuvassa 5. Tällä tavoin tuntemattomampi verkkopalvelu voi ikään kuin ”lainata” vakiintuneen toimijan auktoriteettia tai uskottavuutta. Jos palvelu pystyy vastaamaan menestyneen toimijan (ja tunnistettavan brändin) tarpeisiin, voidaan

tätä käyttää argumenttina uusien asiakkaiden houkuttelussa palvelun käyttäjiksi.

Trusted Enterprise Partner

Enterprise Support and Partnership

Extend existing partner relationships with integrated solutions and deep technical support for Docker from leading enterprise IT vendors in datacenter, cloud and professional services. Plus a global network of resellers, consulting and training partners are there to support every stage of your journey.

Trusted and Certified

Certified Infrastructure, Containers and Plugins deliver validated technology solutions with cooperative support for the technology that powers your business.

[Find a Partner](#)

accenture | Alibaba Group
 avanade | CANONICAL
 CISCO | cloudera
 Hewlett Packard Enterprise | IBM
 Booz | Allen | Hamilton | Microsoft

Kuva 5. Eetos verkkopalveluissa. Käyttäjän luottamusta yritetään saavuttaa esittämällä yhteistyökumppaneiden logoja docker.com -sivustolla (Haettu 8.4.2018)

3.4. Kairos verkkopalveluissa

Antiikin kreikkalaisten aikakäsitys koostui kahdesta elementistä, joita ovat kvantitatiivinen ajan määre *khronos* ja kairos, eli aika kvalitatiivisessa mielessä. Kairos kuvaa oikeaa hetkeä toimia. Jotain tulisi tehdä, mutta ei milloin vain, vaan juuri sille sopivana ajankohtana. [Kinneavy and Erskin 2000; Rife 2010.]

Voidaan väittää, ettei kairos itsessään ole vaikuttamisen keino, vaan erillinen kerros joka toimii vuorovaikutuksessa muun retoriikan kanssa. Kairoksella ei ole itseisarvoa, vaan se on aina sidottu kontekstiin. Vaikka muilla yksittäisillä retoriikan keinoilla voidaan vaikuttaa kuulijaan ilman ajallista elementtiä, voidaan kairoksen avulla argumentointiin lisätä huomattavaa painoarvoa.

Kairoksen merkitys korostuu verkkopalveluissa, joiden käyttäjät (mahdollisesti) tiedostavat, että heidän toimintaansa pyritään jatkuvasti vaikuttamaan. Hyvä esimerkki kairoksen käytöstä nudgeissa löytyy tappiokammosta, jossa usein luodaan keinotekoinen aikaikkuna ja mielikuva ohimenevästä ainutkertaisesta mahdollisuudesta.

Erityisen aggressiivisesti kairosta ja tappiokammosia käyttävät hyväksi matkailualan sivustot. Käyttäjälle esitetään hotellin tietojen yhteydessä teksti "vain 1 huone jäljellä tähän hintaan". Samalla sivun alareunassa esitetään jat-

kuvasti päivittyvää tietoa hiljattain tehdyistä varauksista ja matkakohdetta juuri sillä hetkellä tarkastelevien henkilöiden lukumäärästä. Tällä pyritään siihen, että käyttäjälle muodostuu tarve toimia nopeasti, jotta tämä ei menettäisi mahdollisuutta varata haluamaansa hotellihuonetta.

4. Nudgen suunnitteluprosessi

Tarkoitustaan palvelevan nudgen suunnittelu vaatii systemaattista lähestymistapaa. Prosessi voidaan jakaa karkeasti neljään osaan: ongelman tai tavoitteen määrittämiseen, kohderyhmän diagnosointiin, suunnitteluun ja testaukseen [Datta and Mullainathan 2014; Ly *et al.* 2013; Schneider *et al.* 2017]. Digitaaliset ympäristöt tarjoavat erinomaisia työkaluja, joita voidaan hyödyntää nudgen suunnittelussa. Verkkopalvelut voivat jäljittää ja analysoida käyttäjiensä toimintaa reaaliajassa. Tämän lisäksi ne mahdollistavat käyttöliittymän ja sisällön yksilöllisen muokkaamisen. Näistä tekijöistä johtuen nudgen testaaminen ja iterointi ovat nopeasti toteutettavissa. [Schneider *et al.* 2017.]

4.1. Tavoitteen määrittely

Suunnitteluprosessin ensimmäinen vaihe on tavoitteen määrittely. Suunnittelijan tulisi olla tietoinen organisaation korkean tason tavoitteista jokaisessa valintatilanteessa. Esimerkiksi verkkokaupankäynnin alustan ylläpitäjän tavoitteena on myynnin edistäminen [Schneider *et al.* 2017]. Ensimmäisessä vaiheessa voidaan myös kartoittaa toimintaympäristöä ja määrittellä siinä esiintyviä ongelmia eli syitä, jotka estävät tavoitteen toteutumisen.

Ongelma jaetaan systemaattisesti osiin ja valintaprosessista etsitään mahdollisia *pullonkauloja*. Jos ongelmana on se, ettei verkkokaupan asiakas ostanut jotain tiettyä tuotetta, voidaan pullonkauloja tunnistaa esittämällä kysymyksiä. Aikoiko asiakas edes ostaa tuotetta? Jos ei, pullonkaula voidaan liittää valintatilanteeseen. Jos asiakas taas aikoi ostaa tuotteen, liittyy pullonkaula hänen toimintaansa. Pysyikö hän päätöksessään, vai pyörsikö hän sen? Oliko asiakas sitä mieltä, ettei juuri nyt tarvinnutkaan tuotetta? Unohtiko asiakas ostaa tuotteen? Kaikki nämä mahdolliset pullonkaulat ovat niitä tilanteita, joissa kohdehenkilön päätökseen voidaan vaikuttaa nudgen avulla. [Datta and Mulhainathan 2014.]

Tavoitteet ja pullonkaulat vaikuttavat siihen, miten valintaympäristö pitäisi suunnitella ja millaisia valintavaihtoehtoja ne tarjoavat [Ly *et al.* 2013; Schneider *et al.* 2017]. Valintatilanteessa voi olla tarjolla vain binäärivaihtoehto (kyllä/ei) tai valinta useamman vaihtoehdon väliltä. Valintatyyppi vaikuttaa olennaisesti siihen, millaista nudgea tässä tilanteessa tulisi käyttää. Nudgen suunnittelijan

täytyy olla myös tietoinen siitä, että epäeettiseen tai käyttäjän kannalta epäedulliseen päätökseen ohjaaminen voi auttaa tavoitteen saavuttamisessa lyhyellä aikavälillä, mutta myös vaikeuttaa tai estää tavoitteen toteutumisen pidemmällä aikavälillä. [Schneider *et al.* 2017.]

4.2. Käyttäjän tarpeiden huomiointi

Erilaiset heuristiikat ja ennakkoasenteet vaikuttavat ihmisten päätöksentekoprosesseihin. Heuristiikat voivat vaikuttaa prosessiin positiivisesti, vähentäen tarvetta käsitellä informaatiota yksinkertaisten tai usein toistuvien prosessien yhteydessä. Toisaalta väärin tulkitusta tai huonosti muistetusta informaatiosta voi seurata *kognitiivinen vinouma* (cognitive bias), joka aiheuttaa virheitä monimutkaisia arvioita tehtäessä. [Schneider *et al.* 2017.]

Tästä syystä nudgen suunnittelijan tulee tuntea kohderyhmän heuristiikat ja ennakkoasenteet. Suunnittelijan tulee myös tiedostaa millaiset kannustimet toimivat erilaisille kohderyhmille [Thaler *et al.* 2014]. Ly ja muut [2013] esittelevät neljä tekijää, jotka nudgen suunnittelijan tulisi käyttäjistä tietää:

1. *Päätöksentekoprosessiin vaikuttavat tekijät.* Päätöksentekoon vaikuttaa motivaatio, kannustimet ja se, kuinka paljon käyttäjän keskittymistä päätöksen tekeminen vaatii. Näihin tekijöihin kuuluu myös käyttäjille tarjottavien valintavaihtoehtojen, ja erityisesti oletusvaihtoehtojen, tunnistaminen.
2. *Käyttäjien tiedonlähteet.* Mistä ja miten päätökseen liittyvää informaatiota kerätään ja kuinka se esitetään?
3. *Käyttäjien mielentila.* Onko käyttäjien tunteilla vaikutusta päätöksentekoon?
4. *Sosiaaliset ja ympäristötekijät.* Esimerkiksi ryhmäpaine ja pitkäkestoiset prosessit voivat vaikuttaa päätöksentekoprosessin lopputulemaan.

Kohderyhmän tunteminen on tärkeää erityisesti siksi, että eri kohderyhmien välillä voi olla huomattavia eroja. Esimerkiksi terveellisten elämäntapojen tai ympäristöystävällisyyden korostaminen voi tehotta yhteen kohderyhmään, mutta ei välttämättä johonkin toiseen. [Johnson *et al.* 2012].

4.3. Nudgen valinta

Kun tavoitteet on määritelty ja heuristiikat sekä ennakkoasenteet tunnistettu, voi suunnittelija valita tilanteen kannalta oikean nudgen pyrkiessään vaikuttamaan käyttäjän päätöksentekoon. Se millainen nudge on oikea ja miten se tulisi (käyttöliittymäelementeillä) toteuttaa, riippuu valinnan tyyppistä ja siihen

liittyvistä heuristiikoista. Esimerkiksi tilanteissa, joissa valitaan kahdesta vaihtoehdosta, käytetään usein hyödyksi vallitsevan tilan suosimisen heuristiikkaa ja merkitään haluttu vaihtoehto oletusvalinnaksi. Monivalintatilanteissa voidaan käyttää lukuisia luvussa 2 esiteltyjä nudgeja ohjaamaan käyttäjää kohti haluttua valintaa. [Schneider *et al.* 2017.]

Ihmiset punnitsevat eri vaihtoehtoihin liittyvien ominaisuuksien etuja tai haittoja. Nudgeilla voidaan vaikuttaa siihen, kuinka tärkeinä tai merkityksellisinä nämä ominaisuudet koetaan. Se mitä ominaisuuksia pidetään relevantteina, riippuu kontekstista. Esimerkiksi auton ostajaa voi kiinnostaa tyylliseikat, hinta, turvallisuus, luotettavuus ja suorituskyky. Eri ominaisuuksilla voi toki olla itseisarvoa, mutta päätös syntyy eri tekijöiden yhteisvaikutuksesta. Näin ollen on tärkeää, että vaihtoehtoja ja ominaisuuksia ei esitetä liikaa kerralla, sillä niiden liiallinen määrä hankaloittaa päätöksentekoa. Paras tapa on esittää oleellimmat ominaisuudet ja tarjota käyttäjälle mahdollisuus avata lisätietoja, jotka on oletuksena piilotettu. [Johnson *et al.* 2012.] Tämä tarkoittaa sitä, että suunnittelijan tulee pystyä tunnistamaan kohderyhmän kannalta oleellisimpia ominaisuuksia.

Nudgen valintaa voi myös lähestyä aiemmin esiteltyjen pullonkaulojen kautta. Ly ja muut [2013] kehottavat suunnittelijoita pohtimaan seuraavia kysymyksiä etsiessään ratkaisua pullonkauloihin:

1. Onko käyttäjä tietoinen siitä mitä hänen tulisi tehdä, mutta on kyvytön toimimaan – vai tarvitseeko hän aktivointia haluttuun käyttäytymiseen tai toimintaan?
2. Onko käyttäjä tarpeeksi motivoitunut, jotta nudgella olisi häneen toivottu vaikutus?
3. Suorittaako käyttäjä toiminnon, jos hän saa enemmän tietoa, vai kärsiikö hän kognitiivisesta ylikuormituksesta?
4. Jos haluttua toimintaa ei tapahdu, johtuuko se toimintamallin houkuttelemattomuudesta vai kilpailevasta toimintamallista? Pitäisikö toiminta saada vaikuttamaan houkuttelevammalta vai kilpaileva toiminta epähoukuttelevalta?

Kaikki tässä tutkielmassa esiteltyt nudget perustuvat ihmisille tyypillisiin heuristiikkoihin, ennakkoasenteisiin ja toimintamalleihin. Niiden ymmärtäminen on tärkeää myös siksi, että tilanteeseen sopimaton nudge voi ohjata käyttäjiä kohti sellaisia valintoja, jotka eivät vastaa palvelun tavoitteita. [Schneider *et al.* 2017.]

4.4. Nudgen testaus

Verkkopalvelut mahdollistavat vaihtoehtoisten toteutusten nopean suunnittelun. Näiden toteutusten vaikutusta voidaan myös testata helposti. Digitaalisen nudgen toimivuutta verkkopalveluissa voidaan mitata esimerkiksi A/B-testeillä. Nudgen testaaminen onkin erittäin tärkeää, sillä niiden toimivuus riippuu palvelun kontekstin lisäksi sekä valintaympäristön että käyttäjien tavoitteista. Hyvin jollain verkkosivustolla toimiva nudge ei automaattisesti sovellu toiselle verkkosivustolle. Tämä voi johtua esimerkiksi eroista kohderyhmissä, tai siitä, millaisia palvelunsa ominaisuuksia verkkosivusto haluaa korostaa. [Schneider *et al.* 2017.]

Testejä järjestettäessä tulee varmistua siitä, että niihin satunnaisesti valitut testihenkilöt muodostavat laajan ja kattavan otoksen kohderyhmän populaatiosta. Näin voidaan varmistua siitä, etteivät testitulokset kuvasta nudgen toimivuutta pelkästään jonkin tietyn osajoukon keskuudessa. Tämä ei toki tarkoita sitä, etteikö kohderyhmän sisältä löytyvien osajoukkojen tuloksien eroja voisi hyödyntää analyysissä. [Johnson *et al.* 2012; Ly *et al.* 2013; Schneider *et al.* 2017.]

Testituloksia tarkastellessa on syytä kriittisyyteen, lopputuloksesta riippumatta. Jos testit osoittavat, ettei nudge toimi halutulla tavalla, täytyy epäonnistumisen syytä analysoida. Suunnittelijan tulee varmistua siitä, että nudge oli oikein toteutettu. Joskus nudgen toteutus on liian ilmiselvä ja joskus taas liian hienovarainen, jotta se vaikuttaisi halutulla tavalla. Jos epäonnistuminen ei johtunut toteutuksesta, on syytä arvioida uudelleen kohderyhmän heuristiikat ja ennakkokäsitykset. Joissain tapauksissa on syytä palata takaisin ensimmäiseen vaiheeseen ja aloittaa koko suunnitteluprosessi alusta. [Schneider *et al.* 2017.]

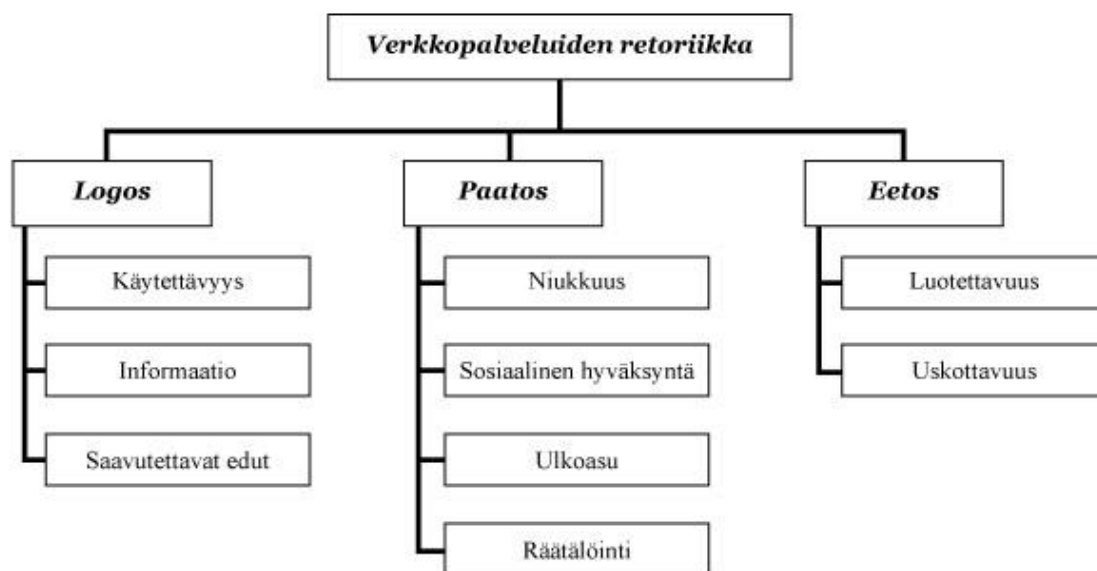
Toisaalta onnistuneenkin testin jälkeen tulee varmistua siitä, että nudge todella toimii. On mahdollista, että testit osoittavat nudgen toimivuuden, mutta haluttu lopputulos onkin saavutettu nudgesta riippumattomista syistä. Tästä syystä testeissä tulisikin arvioida lopputuloksen lisäksi myös suunnitteluprosessia. [Ly *et al.* 2013.]

5. Pohdintaa

Nudgen ja retoriikan välillä on mielestäni selkeä yhteys. Digitaalinen nudge voidaan tavallaan mieltää kehykseksi, jonka kautta verkkopalvelu välittää retoriikkaansa. Nudgen avulla suostutellaan käyttäjää toimimaan halutulla tavalla, esimerkiksi korostamalla tiettyä vaihtoehtoa valintatilanteessa. Onnistuneesti toteutetulla nudgella ei välttämättä tuoda korostetusti esiin retorisia argumentteja, vaan sen avulla voidaan laukaista psykologinen reaktio, jonka vaikutuksesta käyttäjä alkaa itse toimia retoriikan esittäjänä.

Tappiokammon nudge voidaan toteuttaa esimerkiksi matkailualan verkkosivustolla lisäämällä tarjoushintaisten hotellin kuvaukseen teksti ”vain 2 huonetta jäljellä tähän hintaan”. Tämä nudge ei suoraan käännä käyttäjää toimimaan nopeasti, vaan se herättää hänet pohtimaan tilannetta, jolla on selkeästi ennustettava lopputulos – tarve nopealle toiminnalle kumpuaakin käyttäjän omasta retoriikasta. Käyttäjän aikomus muuttuu toiminnaksi, jos argumentit (edullinen hinta, rajoitettu saatavuus, sivuston suuri käyttäjämäärä » rajoitettu aikaikkuna toiminnalle » tarve toimia nopeasti) ovat tarpeeksi vahvoja.

Chu ja muut [2014] ovat tutkineet retoriikkaa kaupallisissa verkkopalveluissa ja luoneet mallin (kuva 6) siitä, mihin retoriikan suostuttelun keinoon verkkosivun erilaiset ominaisuudet liittyvät. Tässä mallissa ei kuitenkaan huomioida kairosta, joka on erityisen mielenkiintoinen tekijä verkkopalveluiden retoriikassa, ja mielestäni aihe vaatii syvällisempää jatkotutkimusta. Verkkopalvelut voivat retoriikan avulla perustella miksi käyttäjän tulisi toimia ja kairoksen avulla voidaan perustella miksi käyttäjän tulisi toimia välittömästi.



Kuva 6. Chun ja muiden [2014] malli retoriikasta kaupallisissa verkkopalveluissa.

Kairos voi muodostua luonnollisesti, esimerkiksi yhteiskunnan vuosirytmiiin sidottuna, tai sitä voidaan luoda keinotekoisesti. Verkkokauppojen tuskin on järkevää markkinoida joulukoristeita heinäkuussa tai pöytätuulettimia helmikuun pakkassäällä. Toisaalta verkkokaupat pyrkivät aggressiivisesti vaikuttamaan käyttäjään vahvistamalla kairosta esimerkiksi juhlapyhien tai vuoden-aikojen vaihtumisen aikaan.

Verkkokaupat voivat ilmoittaa rajatusta ajasta toiminnalle tai tuotteiden rajoitetusta saatavuudesta – näissä tapauksissa kairos on luotu keinotekoisesti. Keinotekoisesti luodun kairosen (niukkuus) avulla voidaan myös luoda lisäarvoa muuten rajattomille digitaalisille resursseille. Esimerkkinä tästä voisi toimia mahdollisuus ostaa kosmeettisia parannuksia verkkopelien pelihahmoille jonain tiettyinä rajoitettuna ajankohtana. Tällaisessa tapauksessa voidaan väittää, että vaikka syitä tuotteen ostamiseen voidaan perustella muillakin retorisen vaikuttamisen keinoilla, sen arvoa voidaan korottaa kairoksella, eli tuotteen rajatulla saatavuudella.

Erittelen kuvan 7 taulukossa niitä retorisen vaikuttamisen keinoja, jotka mielestäni vahvimmin liittyvät yksittäisiin tutkielmassa esiteltyihin nudgeihin. En löytänyt lähdekirjallisuudesta vastaavaa luokittelua, eikä lähdekirjallisuudessa suoranaisesti käsitelty nudgen ja retoriikan yhteyttä. Olen liittänyt taulukon nudgeihin ainoastaan ne vaikuttamisen keinot, joita kyseiset nudget selkeästi käyttävät.

<i>Millaista retoriikkaa nudge käyttää?</i>	Logos	Paatos	Eetos	Kairos
Kehystys				
Vallitsevan tilan suosiminen				
Sosiaaliset normit				
Tappiokammo				
Liikadiskonttaus				
Miellettävyyshauristiikka				
Arvion ankkurointi				
Virittäminen				
Harhautusefekti				
Toimintakutsu				

Kuva 7. Tutkielmassa esiteltyihin nudgeihin liittyvät retoriikan vaikuttamisen keinot on merkitty taulukkoon mustalla.

6. Yhteenveto

Tämä tutkielma on kirjallisuuskatsaus nudgen ja retoriikan soveltamisesta käyttäjän toiminnan ohjailussa verkkopalveluissa. Tieteellistä tutkimusta nudgesta tai retoriikasta juuri tässä viitekehyksessä löytyi kuitenkin niukasti. Tämä on hieman yllättävää, sillä erityisesti nudgen selkeästi tunnistettavia käytännön

toteutuksia löytyy useimmilta verkkosivuilta, joilla olen vierailut kuluvan vuoden aikana. Toisaalta lähdekirjallisuuden hankintaan on voinut myös vaikuttaa oma kokemattomuuteni tieteelliseen kirjoittamiseen kuuluvassa systemaattisessa tiedonhaussa.

Nudge on valinta-arkkitehtuurin osa, jonka avulla voidaan ennustettavalla tavalla vaikuttaa ihmisten päätöksentekoon tai käyttäytymiseen. Nudgen toiminta perustuu vapaaehtoisuuteen. Konseptiin ei kuulu toimintavaihtoehtojen rajoittaminen tai kieltäminen. Jos kouluruokala haluaa ohjata oppilaita kohti terveellisimpiä ruokailutottumuksia, hedelmien sijoittelu linjastoon silmien korkeudelle lasketaan nudgeksi, mutta epäterveellisen ruoan kieltämistä ei.

Retoriikan vaikuttamisen keinot, logos, paatos ja eetos, ilmenevät verkkopalveluissa monin eri tavoin. Loogisuuteen vedotaan hyvällä käytettävyydellä, palvelun tarjoamalla informaatiolla ja korostamalla palvelun käytöstä seuraavia taloudellisia etuja. Tunteisiin vedotaan palvelun ulkoasun ja sisällön räätälöinnin kautta. Ihmisille on tyypillistä hakea toiminnalleen sosiaalista vahvistusta. Verkkopalvelut käyttävätkin tätä hyödyksi päätöksessään, käyttäjään pyritään vaikuttamaan esittämällä hänelle tietoa muiden käyttäjien toiminnasta. Eetosta vahvistetaan luomalla verkkopalvelusta mielikuva luotettavana ja uskottavana toimijana. Luotettavuus saavutetaan erityisesti palvelun ammattitaitoisesti toteutetun ulkoasun kautta ja vakuuttamalla käyttäjä palvelun tietoturvasta. Verkkopalvelulle voidaan hakea uskottavuutta esittämällä todiste kolmannen osapuolen hyväksynnästä, eli ”lainaamalla” riippumattoman toimijan auktoriteettia tai uskottavuutta.

Retoriikan rinnalla vaikuttaa myös ajallinen elementti kairos, joka kuvaa jotain tiettyä ja rajoitettua hetkeä toimia. Kairos ei itsessään ole vaikuttamisen keino, vaan se toimii vuorovaikutuksessa muun retoriikan kanssa. Kairos voi vahvistaa retoriikan voimaa, mutta myös heikentää sitä. Verkkopalvelut voivat luoda kairosta rajoittamalla toiminta-aikaa tai tuotteiden saatavuutta.

Digitaalisen nudgen suunnittelussa käydään läpi neljä vaihetta. Ensimmäisessä vaiheessa määritellään ratkaisua vaativa ongelma tai tavoite. Toisessa vaiheessa määritellään kohderyhmä ja tutkitaan sen heuristiikkoja ja ennakkosenteita. Seuraavassa vaiheessa valitaan kontekstiin ja kohderyhmään sopiva nudge. Lopuksi testataan nudgen toimivuutta. On mahdollista, että haluttu vaikutus voidaan saavuttaa testeissä nudgesta riippumattomista syistä. Siksi on tärkeää analysoida nudgen toimivuuden lisäksi koko sen suunnitteluprosessia.

Pelillistäminen (gamification) on tärkeä tutkielmassa käsittelemättä jäänyt aihe, joka myös vaatisi jatkotutkimusta. Pelillistämisen keinoin voidaan tehokkaasti ohjailta käyttäjän toimintaa verkkopalveluissa. Se soveltuu erityisen hy-

vin ihmisten sitouttamisessa palvelun käyttäjiksi pidemmällä aikavälillä ja sen avulla voidaan lisätä käyttäjien aktiivisuutta palvelussa.

Lopuksi on tarpeen mainita tutkielman rajoitteista. Monet tutkielmassa esiteltyt käsitteet ovat peräisin taloustieteistä ja sosiaalipsykologiasta. Tästä johtuen on mahdollista, että näiden käsitteiden, niiden toimintamekanismien ja niihin liittyvien ilmiöiden sisäistämisessä on voinut tapahtua väärinkäsityksiä. Tutkielman viitekehys, eli verkkopalveluiden suunnittelu, on kuitenkin kirjoittajalle huomattavasti näitä käsitteitä tutumpi.

Viiteluettelo

- Hsi-Liang Chu, Yi-Shin Deng, and Ming-Chuen Chuang. 2014. Persuasive web design in e-Commerce. In: *Proc. of the International Conference on HCI in Business. Lecture Notes in Computer Science 8527*. Springer, 482–491.
- Saugato Datta and Sendhil Mullainathan. 2014. Behavioral design: A new approach to development policy. *Review of Income and Wealth* 60, 1, 7–35.
- Andrew J. Flanagin, Miriam J. Metzger, Rebekah Pure, Alex Markov, and Ethan Hartsell. 2014. Mitigating risk in ecommerce transactions: Perceptions of information credibility and the role of user-generated ratings in product quality and purchase intention. *Electronic Commerce Research* 14, 1, 1–23.
- B. J. Fogg, Cathy Soohoo, David R. Danielson, Leslie Marable, Julianne Stanford, and Ellen R. Tauber. 2003. How do users evaluate the credibility of web sites?: a study with over 2,500 participants. In: *Proc. of the 2003 Conference on Designing for User Experiences*. ACM. 1–15.
- Jacob Gube. 2009. Call to action buttons: Examples and best practices. Smashing Magazine. <https://www.smashingmagazine.com/2009/10/call-to-action-buttons-examples-and-best-practices/> Checked 29.3.2018
- Eric J. Johnson, Suzanne B. Shu, Benedict G.C. Dellaert, Craig Fox, Daniel G. Goldstein, Gerald Häubl, Richard P. Larrick, John W. Payne, Ellen Peters, and David Schkade. 2012. Beyond nudges: Tools of a choice architecture. *Marketing Letters* 23, 2, 487–504.
- James L. Kinneavy and Catherine R. Eskin. 2000. Kairos in Aristotle's rhetoric. *Written Communication* 17, 3, 432–444.
- Kim Ly, Nina Mazar, Min Zhao, and Dilip Soman. 2013. *A Practitioner's Guide to Nudging*. Rotman School of Management Working Paper No. 2609347.
- Tobias Mirsch, Christiane Lehrer, and Reinhard Jung. 2017. Digital nudging: Altering user behavior in digital environments. In: *Proceedings Der 13. Internationalen Tagung Wirtschaftsinformatik (WI 2017)*, 634–648.

- Harri Oinas-Kukkonen and Marja Harjumaa. 2009. Persuasive systems design: Key issues, process model, and system features. *Communications of the Association for Information Systems* 24, 1, 28.
- Andrew K. Przybylski, Kou Murayama, Cody R. DeHaan, and Valerie Gladwell. 2013. Motivational, emotional, and behavioral correlates of fear of missing out. *Computers in Human Behavior* 29, 4, 1841–1848.
- Martine Courant Rife. 2010. Ethos, pathos, logos, kairos: Using a rhetorical heuristic to mediate digital survey recruitment strategies. *IEEE Transactions on Professional Communication* 53, 3, 260–277.
- Yashar Saghai. 2013. Salvaging the concept of nudge. *Journal of Medical Ethics* 39, 8, 487–493.
- Christoph Schneider, Markus Weinmann, and Jan vom Brocke. 2017. Digital nudging–influencing choices by using interface design. Available at SSRN: <https://ssrn.com/abstract=3052192>
- Richard H. Thaler, Cass R. Sunstein. 2008. *Nudge: improving decisions about health, wealth, and happiness*. Yale University Press.
- Richard H. Thaler, Cass R. Sunstein, and John P. Balz. 2014. Choice architecture. Eldar Shafir (ed.). *The Behavioral Foundations of Public Policy*, Ch. 25, 428–439. Princeton University Press.
- Matthias Tietz, Alexander Simons, Markus Weinmann, and Jan vom Brocke. 2016. The decoy effect in reward-based crowdfunding: Preliminary results from an online experiment. In: *Proc. of the Thirty Seventh International Conference on Information Systems*. 285–295.
- Andrew H. Van de Ven and Paul E. Johnson. 2006. Knowledge for theory and practice. *Academy of Management Review* 31, 4, 802–821.
- Markus Weinmann, Christoph Schneider, and Jan vom Brocke. 2016. Digital nudging. *Business & Information Systems Engineering* 58, 6, 433–436.
- Wendy Winn and Kati Beck. 2002. The persuasive power of design elements on an e-commerce web site. *Technical Communication* 49, 1, 17–35.

Kulttuurierot suhtautumisessa informaatiotiheyteen ja aseteluun käyttöliittymäsuunnittelussa

Tuisku Taivainen

Tiivistelmä.

Informaatiotiheys tarkoittaa käyttäjälle esitetyn informaation määrää suhteessa näyttölaitteen kokoon. Tässä kandidaattitutkielmassa pyrin selvittämään, onko eri kulttuurien välillä suhtautumiseroja informaatiotiheyteen ja näyttöjen aseteluun. Perehdyn erityisesti näyttöjen asettelun eroihin länsimaisen sekä japanilaisen käyttöliittymäsuunnittelun välillä. Eroja tarkastellaan verkkosivustojen käyttöliittymien sekä digitaalisten pelien kannalta. Pääasiallisena tutkimusmenetelmänä on tietojenkäsittelytieteen alan kirjallisuuskatsaus. Lisäksi eroja tarkastellaan esimerkkisivustojen karkealla informaatiotiheyden arvioinnilla. Tutkielmassa käsiteltyjen esimerkkien perusteella voidaan todeta, että käyttöliittymäsuunnittelussa on olemassa selkeitä kulttuurikohtaisia eroja. Pohdin myös poikkitieteellisen katsauksen avulla mahdollisia syitä näihin eriäviin suhtautumisiin.

Avainsanat ja -sanonnat: informaatiotiheys, informaatio, yksinkertaisuus, monimutkaisuus, kulttuurisidonnainen käytettävyys, design, käyttöliittymä, käytettävyys, kulttuuri, digitaalinen, digitaalisuus.

1. Johdanto

Tietokoneita käytetään ympäri maailman, mutta suhtautuminen niihin ja niiden käyttötavat vaihtelevat suuresti eri puolilla maailmaa ja eri kulttuureissa. Kulttuuri voidaan tämän tutkielman tarpeisiin määritellä tietyn ryhmän jakamiksi käytös- ja vuorovaikutusmalleiksi sekä kognitiivisiksi rakenteiksi, jotka opitaan sosialisaaion kautta. Kulttuuri vaikuttaa Kroeberin ja Kluckhohnin [1952] mukaan siihen, miten tietyn ryhmän jäsenet tulkitsevat artefakteja, niiden merkitystä sekä niiden suhdetta toisiinsa. Tässä tutkielmassa pyritään havainnollistamaan näitä eroja länsimaisen (Eurooppa, Pohjois-Amerikka, Australia ja Uusi-Seelanti) sekä japanilaisen kulttuurin käyttöliittymäsuunnittelun välillä.

Galitz [2007] määrittelee *informaatiotiheyden* (screen density) siksi ihmisen luettavaksi informaation määräksi, jonka tietty mittayksikkö sisältää. Informaatiotiheys digitaalisessa käyttöliittymässä voidaan tarkemmin määritellä siksi informaation määräksi, joka tutkittavalla näyttölaitteella on kohti yhtä fyysisen pinta-alan mittayksikköä. Yleisimmin digitaalisen informaatiotiheyden pinta-alan mittayksikkönä käytetään näytön kokoa. Galitzin [2007] esittelemä määri-

telmä informaatiotiheydelle palvelee tämän tutkielman tarpeita hyvin, sillä se liittyy läheisesti näyttöjen asetteluun.

Länsimaisissa kulttuureissa pyritään usein minimalismiin käyttöliittymäsuunnittelussa, mutta japanilaisille käyttöliittymille on ominaista korkea informaatiotiheys ja sisällön määrä [Cyr and Trevor-Smith 2004]. Ilmiö on digitaalisten käyttöliittymien lisäksi näkyvissä myös perinteisessä mediassa, kuten aikakauslehdissä. Ilmiötä tarkastellaan tutkielmassa poikkitieteellisen kirjallisuuskatsauksen kautta sekä vertailemalla näytönkaappauksia erilaisista näytönkuvista heuristisin menetelmin.

Tutkielman luvussa 2 määritellään tarkemmin tutkielman sisällölle olennaisia käsitteitä. Luvussa 3 käsitellään esimerkkien kautta käyttöliittymäsuunnittelun kulttuurieroja. Luvussa 4 pohditaan mahdollisia syitä kulttuurieroihin suhtautumisessa käyttöliittymäsuunnitteluun. Tutkielmassa perehdytään erityisesti siihen, miten eri kulttuureissa suhtaudutaan eriasteiseen informaatiotiheyteen ja näyttöjen asetteluun verkkosivusto- ja pelikäyttöliittymäsuunnittelussa. Lisäksi tarkastellaan, mitkä osa-alueet informaatiotiheydessä ja asettelussa ovat länsimaisessa ja japanilaisessa kulttuurissa tärkeimpiä.

Inspiraatio tähän tutkielmaan on peräisin omasta havainnostani siitä, että japanilaisten digitaalisten pelien käyttöliittymät sisältävät länsimaisia vastineitaan huomattavasti enemmän informaatiota pelaajalle. Tämän tutkielman avulla pyrin muodostamaan akateemisen pohjan havainnolleni.

2. Käsitteiden esittely

2.1. Informaatiotiheys

Informaatiotiheys voidaan jakaa *yleiseen informaatiotiheyteen* (overall density) ja *paikalliseen informaatiotiheyteen* (local density). Yleisellä informaatiotiheydellä tarkoitetaan koko näyttölaitteen informaatiotiheyttä eli sen informaation välittämiseen käytetyn pinta-alan suuruutta suhteessa kokonaispinta-alaan. Paikallisella informaatiotiheydellä tarkoitetaan yksittäisten elementtien läheisyyttä toisiinsa nähden. Koko näyttö voi sisältää hyvin pienen määrän informaatiota, mutta se voi olla pakattu hyvin pieneen tai suureen tilaan suhteessa koko näyttölaitteen kokoon. [Galitz 2007]

Informaatiotiheyttä voidaan tarkastella useista eri näkökulmista. Tietyn järjestelmän informaatiotiheys vaikuttaa suoraan käyttäjien kognitiiviseen taakkaan sitä käyttäessä, vaikuttaen näin myös järjestelmän käytettävyyteen. Informaatiotiheyttä voidaan tarkastella objektiivisesta fyysisestä näkökulmasta, kuten näyttöjen koon kannalta, tai sitä voidaan tarkastella digitaalisen informaation määränä, kuten tietynlaisten käyttöliittymäelementtien määränä. Informaatiotiheys liittyy myös läheisesti muihin käyttöliittymäsuunnittelun osiin, kuten asetteluun. [Gabillon *et al.* 2010]

Liian suuri määrä informaatiota kerralla tekee informaation hakemisesta käyttäjälle vaikeampaa, kun taas liian pieni määrä informaatiota kerrallaan voi rasittaa käyttäjän muistia liikaa näyttöjen vaihtuessa. On tärkeää keskittää kaikki yhdelle tehtävälle relevantti informaatio mahdollisimman lukumäärältään vähille näytöille käyttäjän muistin rasittamisen välttämiseksi. [Galitz 2007]

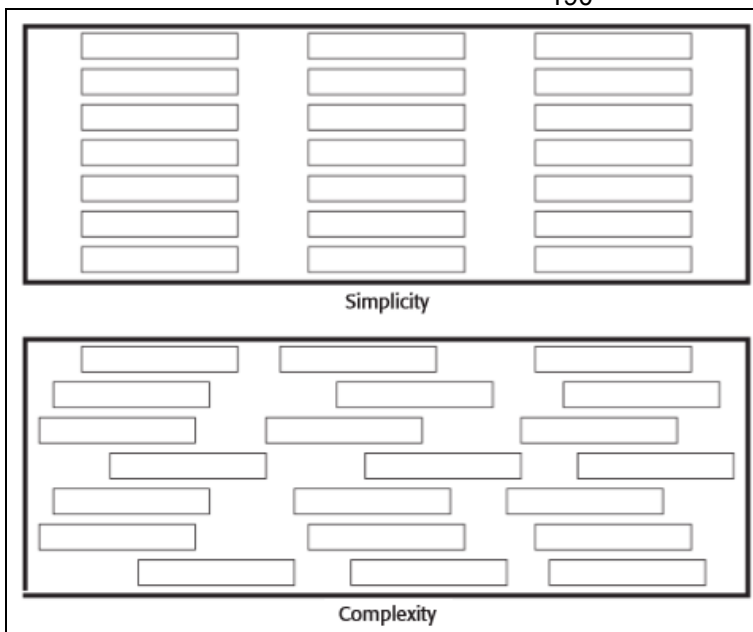
Suurempi informaatiotiheys nostaa tiettyyn pisteeseen asti käyttötehokkuutta, mutta suurentaa käyttäjän virheiden todennäköisyyttä. Tämä asettaa omat ongelmansa informaatiotiheyden johdonmukaiseen suunnitteluun näyttölaitteiden kokojen ollessa hyvin vaihtelevia. Yleisen informaatiotiheyden suosittelu maksimisuuruus on Galitzin mukaan noin 30 %, mikä tarkoittaa, että maksimissaan 30 % näyttölaitteen pinta-alasta kannattaa käyttöliittymäsuunnittelussa käyttää informaation välittämiseen. Tätä suurempi yleinen informaatiotiheys hidastaa käyttäjän toimintaa järjestelmässä. [Galitz 2007]

Informaatiotiheyden käsitettä tarkastellessa on otettava huomioon myös näytön koko ja ihmissilmän katselukulmat näytön elementtien asettelun kannalta. Ihmisille on tyypillistä hakea ensimmäisenä informaatiota näytön alueilta, joilla paikallinen informaatiotiheys on matala [Halverson and Hornof 2004]. Tässä tutkielmassa tarkastellaan näyttöjen informaatiotiheyttä vain käyttöliittymäsuunnittelun ja asettelun kannalta. Näyttöjen kokoja ja katselukulmia ei ole huomioitu muuten kuin vakioimalla luvussa 3 vertailtujen näytönkaappauksien koko.

2.2. Yksinkertaisuus ja monimutkaisuus

Galitz [2007] määrittelee *yksinkertaisuuden* ja *monimutkaisuuden* (simplicity/complexity) elementtien määränä näyttölaitteilla. Niillä viitataan myös *kohdistuspisteiden* (alignment points) määrään. Kohdistuspisteillä tarkoitetaan niitä pisteitä, joiden kautta vedettyjen linjojen mukaan elementtejä voidaan ryhmitellä. Yksinkertaisessa näytössä kohdistuspisteitä on vähän ja monimutkaisessa paljon. Kohdistuspisteitä havainnollistetaan kuvassa 1. Erityisesti horisontaalisten ja vertikaalisten kohdistuspisteiden määrän minimoiminen on tärkeää [Galitz 2007].

Monimutkaisuuden käsitettä käyttöliittymissä voidaan tarkastella myös Williamsin [1995] CRAP-periaatteiden kautta. CRAP on lyhenne sanoista *kontrasti* (contrast), *toisto* (repetition), *kohdistus* (alignment) ja *läheisyys* (proximity). Etenkin kohdistus ja läheisyys ovat relevantteja Galitzin [2007] mukaisen yksinkertaisuuden ja monimutkaisuuden määrittelyn kannalta. Kohdistuksella Williams viittaa samaan asiaan kuin Galitz elementtien asettelun monimutkaisuudella. Williams ja Galitz kummatkin esittävät, että yhteyden luominen eri elementtien välille niiden asettelulla on tärkeää.



Kuva 1: Yksinkertaisuus ja monimutkaisuus [Galitz 2007].

Läheisyys liittyy käsitteenä informaatiotiheyteen merkittävästi. Kun elementit käyttöliittymässä ovat lähellä toisiaan, on myös Galitzin [2007] määrittelemä käyttöliittymän paikallinen informaatiotiheys suuri. Williamsin [1995] läheisyysperiaatteen mukaan toisiinsa liittyvät elementit pitäisi sijoittaa lähelle toisiinsa.

Luvussa 3 esitellään esimerkkejä suhtautumisesta läheisyysperiaatteeseen ja paikalliseen informaatiotiheyteen japanilaisen ja länsimaisen kulttuurin välillä näytönkaappausten avulla.

2.3. Kulttuurisidonnainen käyttöliittymäsuunnittelu

Käyttöliittymäsuunnittelussa on otettava huomioon eri kielten käännösten vaatima tila. Sama termi kirjoitettuna eri kielillä voi viedä erilaisen määrän tilaa näyttölaitteelta. Suunnitellessa käyttöliittymiä eri kulttuureille on kuitenkin pyrittävä ottamaan huomioon muitakin asioita. [Galitz 2007]

Kulttuurissa yleisesti käytetyn kirjoitetun kielen lukusuunta vaikuttaa järjestykseen, jossa kulttuuriin kuuluvat käyttäjät tarkastelevat digitaalisia käyttöliittymiä [Galitz 2007]. Joitakin kieliä, kuten arabiaa, luetaan oikealta vasemmalle. Arabiaa äidinkielenään puhuvien käyttäjien informaationkäsittelyn järjestys kulkee arabiankielisen tekstin lukusuunnan mukaisesti oikealta vasemmalle ja ylhäältä alas. Käyttöliittymien elementtejä tulisi siis järjestää eri tavoilla riippuen kohdekäyttäjien kulttuurista.

Nielsen [2006a] havaitsi englanninkielisten käyttäjien F:n muotoisen katsepolun olevan tyypillinen verkkosivujen selailulle. Sisältöä selataan ensin kahdessa horisontaalisessa linjassa koko sivun poikki, jonka jälkeen sitä selataan yhdessä vertikaalisessa linjassa sivun vasemmassa laidassa. Kielissä, joissa tekstiä luetaan tekstiä tyypillisesti oikealta vasemmalle, tämä tyypillinen F:n

muotoinen katsepolku peilautuu vertikaalisen linjan kautta käänteiseksi [Pernice 2017]. Oikealta vasemmalle luettuja kieliä käyttävissä kulttuureissa myös käyttöliittymäsuunnittelun periaatteet noudattavat näitä peilattuja katsepolkuja. Täten myös japanin kielen perinteinen lukusuunta vaikuttaa todennäköisesti käyttäjien katsepolkuihin verkkosivuilla. On kuitenkin epäselvää, kuinka paljon japania äidinkielenään puhuvat käyttäjät ovat tottuneet selaamaan verkkosivuja vasemmalta oikealle ja ylhäältä alas länsimaiseen tyyliin.

Kulttuurisidonnainen käyttöliittymäsuunnittelu [Barber and Badre 1998] tarkoittaa kulttuurin ja käyttöliittymäsuunnittelun suhdetta. Heidän määritelmänsä keskittyy verkkokäyttöliittymien suunnitteluun, mutta se on laajennettavissa myös mihin tahansa kansainväliseen käyttöön tehtyyn ohjelmistoon. Eri kulttuurien suhtautuminen siihen, mikä on tilankäytöllisesti loogista, helppokäyttöistä tai esteettistä on erittäin vaihtelevaa. Fonteilla, väreillä ja muodoilla on kaikilla usein hyvin erilaisia kulttuurisidonnaisia merkityksiä.

3. Esimerkkejä kulttuurieroista näyttöjen asettelussa ja informaatiotiheydessä

Tässä luvussa tarkastellaan erityyppisten verkko- ja pelikäyttöliittymien eroja näyttöjen asettelun ja informaatiotiheyden kannalta. Eroja tarkastellaan käyttöliittymistä otettujen näytönkaappauksien avulla. Tarkasteltaviksi näytöiksi on rajattu se osa näkymistä, joka ruudulle mahtuu. Jokaisesta käyttöliittymästä on tutkittavana englanninkielinen versio tulkinnan helpottamiseksi, ellei näytöissä ei ole merkitseviä eroja muunkin kuin käytetyn kielen osalta. Kaikki näytönkaappaukset on otettu pöytätietokoneen näytöltä, jonka koko on 1920 x 1080 pikseliä. Verkkosivujen kuvista on leikattu vertailun tarpeisiin pois verkkoselaimen yläpalkki (1920 x 66 pikseliä) ja Windows 10 -käyttöjärjestelmän alapalkki (1920 x 40 pikseliä). Vertailtujen näyttöjen koko on siis 1920 x 974 pikseliä.

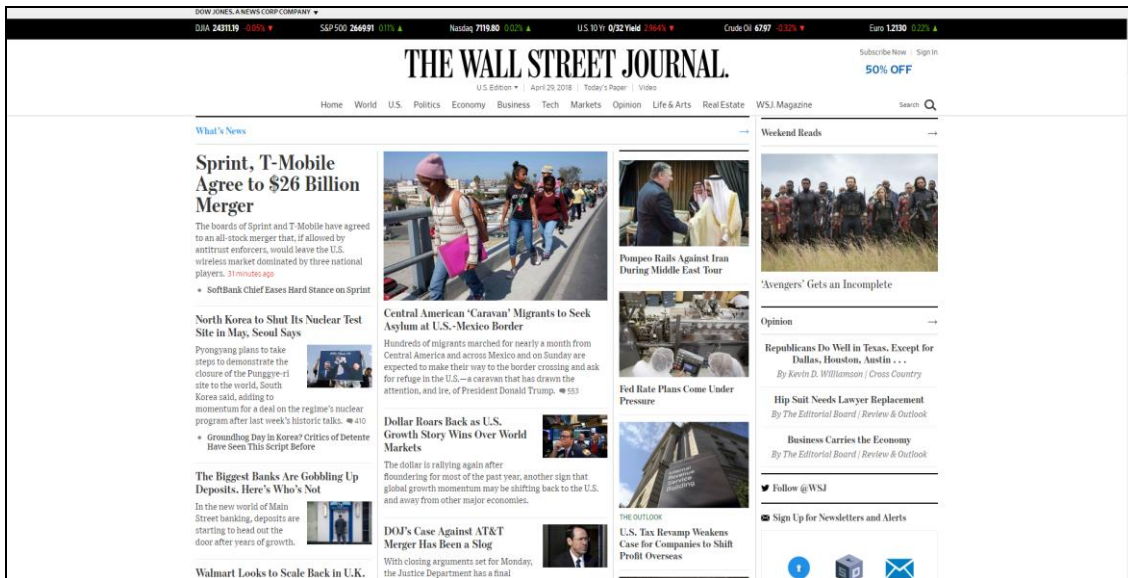
3.1. Sanomalehtien verkkosivut

Vertaillaan aluksi käyttöliittymäsuunnittelua sanomalehtien verkkosivujen etusivujen välillä. Käytetyt näytönkuvat ovat peräisin The Wall Street Journalin englannin- (kuva 2) ja japaninkielisiltä (kuva 3) ja Yomiuri Shimbunin englannin- (kuva 4) ja japaninkielisiltä (kuva 5) verkkosivuilta. Milosevicin [2016] mukaan The Wall Street Journal on Yhdysvaltain levikiltään toiseksi suurin sanomalehti ja Yomiuri Shimbun Japanin levikiltään suurin sanomalehti. Nämä sanomalehdet valittiin tarkasteltaviksi niiden yleisen tunnettuuden sekä niiden japanin- että englanninkielisten versioiden saatavuuden vuoksi.

The Wall Street Journalin etusivu kuvissa 2 ja 3 muistuttaa ulkoasultaan perinteisen sanomalehden asettelua, kuten ohjeistaa Nielsenin [1995] toinen heuristiikka, eli samankaltaisuus järjestelmän ja todellisuuden välillä. Leveältä

näytöltä on käytetty melko kapea tila informaation välittämiseen, mikä auttaa elementtien asettelussa paperista sanomalehteä muistuttavaksi. Galitzin [2007] määritelmien mukaan sivun yleinen informaatiotiheys on korkea, mutta myös paikallinen informaatiotiheys on melko korkea.

Williamsin [1995] CRAP-periaatteista kontrastin käyttö on käyttöliittymässä hyvin selkeä imitoiden myös paperista sanomalehteä. Kuvia lukuun ottamatta lähes koko sivusto on mustavalkoinen. Kummankin sivun asettelussa on käytetty selkeitä linjoja ja tyhjää tilaa elementtiryhmien erotteluun.



Kuva 2: The Wall Street Journalin englanninkielinen etusivu [The Wall Street Journal 2018a]. Haettu 29.4.2018.



Kuva 3: The Wall Street Journalin japaninkielinen etusivu [The Wall Street Journal 2018b]. Haettu 29.4.2018.

Eroja näytönkaappauksissa on pyritty hahmottamaan laskemalla yksittäisten lohkojen määrät eri näytöillä ja niiden suhteet toisiinsa palstojen ja rivien mää-

rinä. Nämä tiedot on taulukoitu jokaista näyttöä kohden. Lohkotyyppejä ei ole eritelty esimerkiksi navigaatio- ja sisältöelementteihin, sillä en puhu japania, eikä japaninkielisten sivujen elementtien tyyppien erottelu täten ole mahdollista. Lohkoiksi on tässä pyritty laskemaan yksittäiset visuaalisesti erilliset objektit näytöllä. Esimerkiksi navigaatioon tarkoitettu valintapalkki on laskettu yhtenä lohkona, kun taas erilliset uutiset tai tuotteet on laskettu erillisinä lohkoina. Yksittäiset uutisotsikot on taulukossa laskettu erillisiksi elementeiksi, ellei niitä ole esitetty jonkinlaisessa widgetissä, jolloin ne on laskettu yhdeksi elementiksi. Tarkastelussa oleellisinta on, että kaikissa näytönkuvissa laskenta on suoritettu samalla tavalla. Yksittäisten elementtien määrällä ei ole niin paljon merkitystä kuin määrien välisillä suhteilla.

Tarkastelussa kuvia ei ole laskettu lohkoihin, jos ne liittyvät selkeästi teksti- tai muuhun elementtiin. Tällöin ne on laskettu yksittäiseksi elementiksi. Palstojen määrä on laskettu sivun kohdalta, jossa palstoja on rinnakkain eniten. Sisällön suhteellinen informaatiotiheys on laskettu jakamalla lohkojen määrä palstan leveydellä näytöstä prosentteina. Idea tämän kaltaiseen lohkojen laskemiseen ja niiden tiheyden karkeaan arviointiin on peräisin Cyriltä ja Trevor-Smithiltä [2004], Chulta ja Yangilta [2010], sekä Gabillonilta ja muilta [2010] yksinkertaistettuna versiona. Pyrin arvioimaan näytönkaappausten visuaalisten lohkojen määrää ja vertailemaan tuloksia toisiinsa taulukoimalla niitä. Kuvien 2 ja 3 perusteella tehty analyysi on koottu taulukkoon 1.

	Palstojen leveys (px)	Palstojen leveys (%)	Suhteellinen informaatiotiheys	Lohkojen määrä	Palstojen määrä
The Wall Street Journal, eng.	1260	65,63 %	25,90	17	4
The Wall Street Journal, jap.	1260	65,63 %	28,95	19	4

Taulukko 1: Analyysia The Wall Street Journalin näytönkaappauksista.

The Wall Street Journalin kahden eri version välillä ainoa merkittävä ero asettelussa on sivuston oikeaan yläkulmaan sijoitettu laatikko. Englanninkielisessä versiossa tähän on asetettu kevyempi uutinen otsikolla "Weekend news" ja japaninkielisessä versiossa samassa tilassa on pörssikurssi-widget. Kyseinen widget on tosin olemassa kaikissa paitsi yhdysvaltalaisessa sivun versiossa. Koska se löytyy myös eurooppalaisesta versiosta, erityisiä erotteluja tämän suhteen ei japanilaisen ja yhdysvaltalaisen etusivun välillä voida tehdä. Yhdysvaltalaisessa versiossa sivun oikeassa laidassa on linkki pelkästään The Wall Street Journalin Twitter-tiliin, kun taas japaninkielisessä versiossa linkkejä on useampaan sosiaaliseen mediaan.

Elementtien ja palstojen määrä sivulla on suhteellisen matala sekä englannin- että japaninkielisessä versiossa. Eri kulttuurien suunnitteluperiaatteet eivät The Wall Street Journalin etusivuilla ole erityisen näkyviä. Palstojen ja lohkojen määrä on kummallakin sivulla täysin sama, ja pienet erot johtuvat todennäköisesti eri uutisista, joita eri kielisillä etusivuilla esitetään. Kuvien asettelussa pääuutispalstalla on pieniä eroja, mutta tämä asettelu muuttuu kummallakin versiolla etusivusta uutisten vaihtuessa, eikä sitä täten voida tässä pitää kulttuurisidonnaisena elementtinä.

Seuraavaksi tarkastellaan vastaavia eroja japanilaisen sanomalehti Yomiuri Shimbunin verkkosivuilla kuvissa 4 ja 5 sekä taulukossa 2.



Kuva 4: Yomiuri Shimbunin englanninkieliset verkkosivut [Yomiuri Shimbun 2018a]. Haettu 29.4.2018.



Kuva 5: Yomiuri Shimbunin japaninkieliset verkkosivut. [Yomiuri Shimbun 2018a]. Haettu 29.4.2018.

	Palstojen leveys (px)	Palstojen leveys (%)	Suhteellinen informaatiitiheys	Lohkojen määrä	Palstojen määrä
Yomiuri Shimbun, eng.	1040	54,17 %	27,69	15	3
Yomiuri Shimbun, jap.	1010	52,60 %	57,03	30	4

Taulukko 2: Analyysia Yomiuri Shimbunin näytönkaappauksista.

Kontrasti englannin- ja japaninkielisten Yomiuri Shimbunin etusivujen välillä on vahva. Englanninkielisen etusivun otsikko on "The Japan News" lehden alkuperäisen japaninkielisen nimen sijaan. Pyrkimyksenä tässä on todennäköisesti länsimaistaa etusivun tyyliä, mikä näkyy muissakin sivun piirteissä, kuten väljemmässä asettelussa, leveämmässä palstatilassa ja hienovaraisemmassa värien käytössä. Vahvat erot käyttöliittymäsuunnittelussa sivuston eri kielisten versioiden välillä viestivät pyrkimyksestä hyödyntää suunnittelussa Sunin [2001] määrittelemiä kulttuurikohtaisia käyttöliittymäsuunnittelun piirteitä.

Japaninkielinen versio on asettelultaan selkeästi erilainen. Yomiuri Shimbunin japaninkielinen verkkosivu ei pyri imitoimaan sen reaaliaikaisen vastinetta yhtä paljon. Japanilaiset sanomalehdet painetaan perinteisellä japanilaisella kirjoitustavalla; niiden teksti luetaan ensin ylhäältä alas ja sitten oikealta vasemmalle, kun taas Yomiuri Shimbunin japaninkielisen verkkosivun sisältö on tehty luettavaksi vasemmalta oikealle ja sitten ylhäältä alas länsimaiseen tapaan. Japanissa tämä länsimaistyylinen tekstin kirjoitussuunta on kuitenkin verkkosivujen suunnittelussa tyypillistä.

Etusivun englanninkielinen versio muistuttaa yleiseltä tyyliltään melko paljon edellä tarkasteltua The Wall Street Journalin etusivua. On tosin huomiotava, että koko sivun oikeanpuoleisin palsta on varattu englanninkielisessä Yomiuri Shimbunissa kokonaan mainoksille. Japaninkielisessä versiossa Yomiuri Shimbunista tai kummassakaan versiossa The Wall Street Journalista mainokset eivät vie läheskään näin paljon tilaa.

Valittu välilehti on sivun ylälaidan navigaatiopalkissa korostettu punaisella värillä englanninkielisessä Yomiuri Shimbunissa. Tätäkin enemmän värikkyyttä on esillä Yomiuri Shimbunin japaninkielisessä versiossa, jossa eri värejä käytetään runsaasti sivun eri elementeissä. Vasemmassa yläkulmassa on havaittavissa kaksi söpöä hahmoa, ja niiden alla useita värikkäitä ikoneita. Tämä värien käyttö ja söpöt hahmot edustavat Cheokin ja Fernandon [2012] esittelemää japanilaista söpöskulttuuria.

Silmämääräisesti arvioituna Yomiuri Shimbunin kummankin version sisältö on aseteltu Galitzin [2007] määrittelemän yksinkertaisuuden kannalta yk-

sinkertaisemmin kuin The Wall Street Journalin etusivut. Yomiuri Shimbunin etusivuilla on käytetty vähemmän merkittäviä kohdistuspisteitä sivuston sisällön asettelussa. Tämän voidaan ajatella heijastavan Singhin ja Matsuon [2004] kuvaamaa selkeän hierarkian ja rakenteen arvostamista japanilaisessa kulttuurissa. Toisaalta japaninkielisen etusivun lohkot on aseteltu huomattavasti pienemmin välein kuin englanninkielisen version tai The Wall Street Journalin kummankaan version lohkot.

Erot sivustojen yleisessä informaatiitiheydessä ovat suuria. Yomiuri Shimbunin japaninkielisellä etusivulla on enemmän sisältöä näkyvissä kerralla kuin sen englanninkielisellä etusivulla. Tämä tukee Cyrin ja Trevor-Smithin [2004] havaintoja sisällön määrästä erikulttuurisilla verkkosivuilla. Japaninkielisellä versiolla useammista uutisista tarjotaan käyttäjälle vain otsikko eikä lainkaan lyhytkuvausta, kuten englanninkielisessä versiossa.

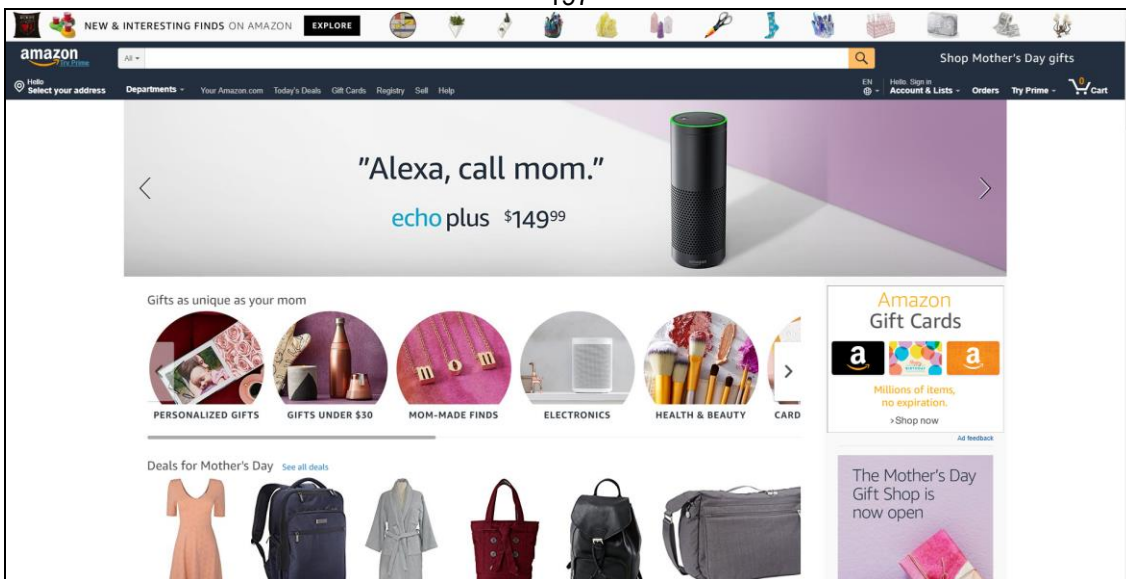
Japaninkielisessä Yomiuri Shimbunissa kaikki kuvat ovat melko pieniä. Tämä saattaa johtua siitä, että ne on suunniteltu selattaviksi myös kapeilla mobiilinäytöillä. Se käyttää näytöltä horisontaalisesti kapeamman alueen kuin englanninkielinen, ja sen paikallinen informaatiitiheys on tästä seurauksena korkeampi kuin The Wall Street Journalin tai englanninkielisen Yomiuri Shimbunin. The Wall Street Journalin palstanleveys on noin 10 % suurempi kuin Yomiuri Shimbunin palstanleveys. Ero palstojen leveyksissä on kuitenkin melko pieni, ja se saattaa johtua myös sattumasta.

3.2. Verkkokaupat

Seuraavaksi vertaillaan käyttöliittymäsuunnittelua verkkokauppojen etusivujen välillä. Vertailun kohteina ovat yhdysvaltalaisen verkkokauppa Amazonin sekä japanilaisen verkkokauppa AmiAmin englanninkieliset etusivut. Amazon (kuva 6) on suuri monialainen verkkokauppa, jossa myydään käytännössä kaikenlaisia käyttö- ja koriste-esineitä. AmiAmi (kuva 7) keskittyy harrastetuotteisiin, kuten vaatteisiin, kirjoihin ja keräilytuotteisiin. Vaikka Amazon on valikoimaltaan laajempi, sisältää sen valikoima myös AmiAmin valikoimaa vastaavia tuotteita.

Kummankin sivuston englannin- ja japaninkieliset versiot ovat asetellutaan käytännössä täysin samanlaisia, eikä niiden vertailussa täten ole erityistä mielenkiintoa. Englanninkieliset sivut valittiin vertailtavaksi, jotta näytönkaappauksen tulkitseminen olisi helpompaa.

Amazonin ja AmiAmin verkkosivuilla käyttöliittymäsuunnittelu on sekä englannin- että japaninkielisillä sivuilla käytännössä samanlaista käytettyä kieltä lukuun ottamatta, joten niistä tarkastellaan tässä tutkielmassa selkeyden vuoksi vain englanninkielisiä versioita.



Kuva 6: Amazonin englanninkielinen etusivu [Amazon 2018]. Haettu 29.4.2018.



Kuva 7: AmiAmin englanninkielinen etusivu [AmiAmin 2018]. Haettu 29.4.2018.

Yksittäiset tuotteet etusivulla on taulukossa 3 laskettu yksittäiseksi elementiksi, ellei niitä ole esitetty jonkinlaisessa widgetissä. Tällöin widget on laskettu yhdeksi elementiksi.

	Palstojen leveys (px)	Palstojen leveys (%)	Suhteellinen informaatiitiheys	Lohkojen määrä	Palstojen määrä
Amazon, eng.	1500	78,13 %	23,04	18	2
AmiAmin, eng.	1014	52,81 %	58,70	31	4

Taulukko 3: Analyysia Amazonin ja AmiAmin näytönkaappauksista.

Erot sivuissa ovat hyvin selkeitä. Amazonin palstanleveys on lähes puolet suurempi kuin AmiAmin, mutta AmiAmin etusivulla on lähes kaksi kertaa enemmän lohkoja ja kaksi kertaa enemmän palstoja kuin Amazonin etusivulla. Tästä seuraa se, että AmiAmin suhteellinen informaatiotiheys on yli kaksi kertaa suurempi. Voidaan myös havaita, että AmiAmin suhteellinen informaatiotiheys on hyvin lähellä Yomiuri Shimbunin japaninkielisen etusivun suhteellista informaatiotiheyttä. Amazonin etusivun suhteellinen informaatiotiheys on lähempänä The Wall Street Journalin kummankin version ja Yomiuri Shimbunin englanninkielisen version suhteellista informaatiotiheyttä. Näin pienellä otannalla ei kuitenkaan voida tehdä johtopäätöksiä, onko kyseessä juuri japanilaiselle käyttöliittymäsuunnittelulle ominainen informaatiotiheys.

AmiAmin korkea informaatiotiheys voi selittyä osittain Singhin ja Matsun [2004] kuvaamalla perinteiden arvostamisella sekä Tabuchin [2009] kuvaamalla Galapagos-ilmiöllä verkkosivujen suunnittelussa. Tabuchin mukaan yleisin tapa japanilaisille käyttää Internetiä olivat pitkään pieninäyttöiset matkapuhelimet, jolloin kapealle ruudulle piti mahduttaa paljon informaatiota. Tämän kaltainen tiivis design on voinut japanilaisessa perinteitä kunnioittavassa kulttuurissa jäädä standardiksi.

AmiAmin etusivulla käytännössä kaikki mahdolliset tuotekategoriat ovat käyttäjälle suoraan näkyvissä, kun taas Amazonin etusivulla niitä on piilotettu pudotusvalikoiden taakse. Amazonin etusivulla noudatetaan *asteittaisen paljastamisen* (progressive disclosure [Nielsen 2006b]) periaatetta, eli näytetään käyttäjälle kerrallaan vain sillä hetkellä tarpeellisen informaation. Singhin ja Matsun [2004] mukaan japanilaisessa kulttuurissa arvostetaan selkeitä hierarkioita ja ryhmäorientoituneisuutta, ja tällainen vaihtoehtojen näkyvyys voidaan nähdä osana näitä asenteita.

Värien käyttö on kummallakin sivulla melko runsasta. On kuitenkin huomioitava, että kummallakin sivulla suurin osa värikkydestä on peräisin tuotteiden kuvista ja mainoksista. Etenkin Amazonin etusivulla itse sivussa käytetyt värit ovat matalakontrastisempia kuin AmiAmin tunnusväri oranssi. AmiAmin etusivun yläpalkissa on sivuston logossa käytetty pyöristettyjä kirjaimia, ja sen vieressä on kuva sivuston söpöstä tunnushahmosta. Nämä piirteet ovat Cheokin ja Fernandon [2012] kuvaaman japanilaiselle kulttuurille ominaisen söpöyden mukaisia. Siinä missä japanilaisessa AmiAmissa pyritään söpöyteen ja vaihtoehtojen näkyvyyteen, pyritään yhdysvaltalaisessa Amazonissa minimalistiseen designiin Nielsenin [1995] heuristiikkojen mukaisesti.

Kun etusivuja tarkastellaan Galitzin [2007] määrittelemän yksinkertaisuuden perusteella, huomataan kohdistuspisteiden määrän olevan kummallakin etusivulla melko pieni. AmiAmin etusivulla selkeää, yksinkertaista rakennetta pyritään korostamaan selkeästi laatikoiduilla elementeillä ja palstojen eri taroituksilla. Etusivun sisällön kategorisointi on Amazonin etusivulla vähemmän

objektiivista ja selkeän hierarkkista. Siinä missä AmiAmi lajittelee tuotteita näkyvien tyyppikategorioiden ja niiden suosion perusteella, Amazonissa etusijalla ovat äitienpäivään sopiviksi määritellyt tuotteet. Vaikka äitienpäivä on Japanissa samaan aikaan kuin länsimaissa, ero voi selittyä osaltaan sillä, että AmiAmia ei välttämättä mielletä harrasteverkkokauppana äitienpäivälahjoille sopivaksi. Vastaavia subjektiivisempia kategorioita AmiAmin etusivulta ei kuitenkaan löydy.

3.3. Pelikäyttöliittymät

Seuraavaksi vertaillaan digitaalisten pelien käyttöliittymien eroja eri kulttuureissa. Verrattaviksi on valittu pelit *The Surge* [Deck13 Interactive 2017] ja *Dark Souls III* [FromSoftware 2016]. Deck13 Interactive on saksalainen yritys, ja FromSoftware on japanilainen yritys. *The Surge* (kuva 8) kuuluu täten länsimaiseen ja *Dark Souls 3* (kuva 9) japanilaiseen kulttuuripiiriin. Nämä pelit valittiin tarkasteltavaksi niiden pelimekaanisen samankaltaisuuden ja tutkielman tekijän henkilökohtaisten mieltymysten takia. FromSoftware'n suosittu Souls-pelisarja on inspiroinut lukuisia pelejä matkimaan sen omalaatuisia pelimekaaniikkoja. *The Surge* on yksi näistä peleistä.

Dark Souls III:n ja *The Surgen* käyttöliittymien asetteluissa ei ole eroja japanin- ja englanninkielisen version välillä eri kielen lisäksi. Analyysin helpottamiseksi käytetään tässä tutkielmassa analyysiin englanninkielistä näytönkaappausta. Tarkasteltaviksi on valittu pelien näkymät, joissa hallinnoidaan pelaajahahmon varusteita. Näkymät valittiin siksi, että tarkasteltavien pelien kaltaisissa toimintaroolipeleissä pelaajan varustenäkymä on erittäin keskeinen pelaajan kokemukselle, ja se saattaa sisältää hyvinkin paljon informaatiota. Näkymissä jokainen pelaajalle relevantti arvo on laskettu yksittäiseksi elementiksi, elleivät ne kuulu selkeään pelimekaaniseen ryhmään.

Pelikäyttöliittymiä tarkastellessa tulisi ottaa huomioon myös integroidut käyttöliittymät. Tällä tarkoitetaan Jorgensenin ja Llanosin [2011] mukaan käyttöliittymäelementtejä, jotka ovat osana pelimaailmaa. Integroituna käyttöliittymäelementtinä pidetään esimerkiksi sitä, että esine, jonka kanssa pelaaja pysyy vuorovaikuttamaan, välähtelee punaisena pelimaailmassa. Koska tässä tutkielmassa tarkastellaan ensisijaisesti kaappauksia pelin valikoista, eivät integroidut elementit ole suurena osana tarkastelua.



Kuva 8: The Surge -pelin englanninkielinen versio valikosta, jossa valitaan pelaajan hahmolle varusteet [Deck13 Interactive 2017]. Haettu 3.5.2018.



Kuva 9: Dark Souls III -pelin englanninkielinen versio valikosta, jossa valitaan pelaajan hahmolle varusteet [FromSoftware 2016]. Haettu 26.3.2018.

Palstojen määrä on pelikäyttöliittymissä pyritty laskemaan samalla tavalla kuin aikaisemminkin (taulukko 4). Käyttöliittymien tyypistä johtuen varsinkin pelikäyttöliittymien palstat voisi laskea eri tavalla.

	Palstojen leveys (px)	Palstojen leveys (%)	Suhteellinen informaatiotiheys	Lohkojen määrä	Palstojen määrä
The Surge	1400	72,92 %	31,54	23	2
Dark Souls III	1825	95,05 %	61,02	58	3

Taulukko 4: Analyysia The Surgen ja Dark Souls III:n näytönkaappauksista.

The Surgen palstojen leveys on selkeästi kapeampi kuin Dark Souls III:n. Koska lohkojen määrä Dark Souls III:n näytönkaappauksessa on huomattavasti suurempi kuin The Surgessa, on Dark Souls III:n käyttöliittymän informaatiotiheys tämän näkymän perusteella silti huomattavasti suurempi. Erot näiden kahden näkymän suhteellisten informaatiotiheyksien välillä mukailevat pitkälti tarkasteltujen verkkosivujen lukemia, vaikka suhteellinen informaatiotiheys on pelikäyttöliittymissä tämän perusteella hieman suurempi kuin verkkosivuilla. Dark Souls III:n leveämpi palstatila voidaan selittää muun muassa sillä, että digitaalisten pelien käyttöliittymien suunnittelussa ei ole Japanissa ollut samanlaisia tilarajoitteita kuin Tabuchin [2009] mukaan matkapuhelinten näytöille sopivia verkkosivuja suunnitellessa on ollut.

Dark Souls III:n käyttöliittymässä pelaajalle tarjotaan huomattavasti enemmän informaatiota kuin The Surgessa. Voidaan myös havaita, että The Surge tarjoaa käyttäjälle vain sillä hetkellä relevantin informaation. Dark Souls III:ssa kaikki annettu informaatio, kuten kuvan 9 oikeassa laidassa oleva "player status"-paneeli, ei ole käyttäjälle relevanttia pelaajahahmon varusteita valitessa. Jotkut tiedot, kuten "Equip load"-arvo, esitetään sen käyttöliittymässä kahdessa eri paikassa. Voidaan todeta, että The Surge noudattaa paremmin Galitzin [2007] periaatetta vain käyttäjälle relevantin informaation näyttämisestä.

The Surgessa on olemassa Dark Souls III:n näkymää yleisen informaatiotiheyden suhteen muistuttava näkymä, jossa esitetään kaikki mahdollinen pelaajahahmoon liittyvä informaatio. Esitetyssä näytönkaappauksessa sen löytää ruudun ylälaidan "Specs"-välilehdeltä. The Surge noudattaa asteittaisen paljastamisen periaatetta [Nielsen 2006b]. The Surgen varustenäkymä vastaa käytännössä esitetyltä informaatioltaan Dark Souls III:n varustenäkymän vasemmanpuoleista palstaa ja osaa sen oikeanpuoleisesta palstasta. The Surgen 3d-malli pelaajahahmosta varusteet yllään voidaan nähdä integroituna elementtinä sen

käyttöliittymässä. Vastaavaa elementtiä ei Dark Souls III:n käyttöliittymässä ole.

Voidaan huomata, että Dark Souls III:n käyttöliittymässä on pyritty informaation näkyvyyteen, ja The Surgen käyttöliittymässä minimalismiin ja relevantin informaation näyttämiseen. Tämän voidaan nähdä heijastavan Singhin ja Matsuon [2004] kuvaamia selkeitä hierarkioita japanilaisessa yhteiskunnassa.

4. Syitä kulttuurieroille käyttöliittymäsuunnittelussa ja näyttöjen asettelussa

Kulttuuri on tärkeä tekijä, kun suunnitellaan käyttöliittymiä kansainväliselle yleisölle. Ottamalla huomioon konseptien ja symbolien kulttuurisidonnaisia merkityksiä käyttöliittymäsuunnittelussa voidaan parantaa eri kulttuureihin kuuluvien käyttäjien käyttökokemuksia. [Barber and Badre 1998; Sun 2001]

Kuten luvussa 3 tehty analyysi osoitti, eri kulttuureissa suhtaudutaan käyttöliittymäsuunnitteluun ja informaatiotiheyteen eri tavoilla. Havainto tukee Cyrin ja Trevor-Smithin [2004] tuloksia, joiden mukaan Yhdysvaltalaisessa, saksalaisessa ja japanilaisessa käyttöliittymäsuunnittelussa on eroja muun muassa tekstin muotoilussa, tilan käytössä, asettelussa ja informaatioarkkitehtuurissa. Erityisen suuri ero on havaittavissa informaatioasisällön määrässä suhteessa sivuston käyttöliittymäelementteihin. Navigaatio- ja grafiikkaelementtien määrä on lähes samanlainen kaikissa tarkastelluissa kulttuureissa, mutta japanilaisissa verkkosivuissa informaatioasisällön määrä suhteessa kaikkiin käyttöliittymäelementteihin yhdellä näytöllä on 71,1 %, kun taas yhdysvaltalaisissa verkkosivuissa sen osuus on 28,43 % ja saksalaisissa 20 % [Cyr and Trevor-Smith 2004].

Seuraavaksi pohditaan mahdollisia syitä kulttuurieroihin länsimaisessa ja japanilaisessa käyttöliittymäsuunnittelussa kolmesta eri näkökulmasta. Ensinnä pohditaan puhutun ja kirjoitetun kielen vaikutusta käyttöliittymäsuunnittelupreferensseihin. Toisena tarkastellaan teknologian kehitystä mahdollisena syyinä näihin eroihin, ja viimeisenä esitellään muita mahdollisia kulttuurisia syitä.

4.1. Kielelliset syyt

Erilaisia preferenssejä käyttöliittymäsuunnittelussa ja informaatiotiheydessä voi selittää kielten erilainen informaatiotiheys. Pellegrinon ja Coupén [2011] mukaan puhutun japanin kielen informaatiotiheys on matalampi kuin esimerkiksi englannin, joten japanilaisessa kulttuurissa on todennäköisesti totuttu suurempaan määrään tulkittavaa sisältöä.

Kirjoitetun englannin tyypillinen lukusuunta on ensin vasemmalta oikealle ja sitten ylhäältä alas. Japanin kieltä voidaan kirjoittaa kahdella eri tavalla. Perinteisesti kirjoitetun japanin lukusuunta on ensin ylhäältä alas, sitten oikealta vasemmalle. Modernia japania kirjoitetaan myös vaakasuoraan vasemmalta

oikealle, kuten englantia. Näitä kahta kirjoitustyyliä voidaan myös sekoittaa. Japanilaisilla verkkosivuilla käytetään lähes pelkästään vaakasuoraa kirjoitustapaa.

Japanin kielessä käytetään kolmea erillistä aakkostoa. Näihin kuuluvat kaksi foneettista aakkostoa: hiraganat, joita käytetään muun muassa sanojen taivutukseen ja lapsille suunnattuun kieleen, sekä katakanat, joita käytetään pääasiassa vierasperäisten sanojen kirjoittamiseen. Lisäksi käytössä ovat Kiinasta peräisin olevat kanji-kirjoitusmerkit, jotka merkitsevät kirjainten tai tavujen sijaan kokonaisia sanoja. Hiraganoja voidaan myös käyttää kanjien ääntämishojeina. [Joyce 2004]

Kanji-kirjoitusmerkeillä voidaan ilmaista hyvin pienessä tilassa merkityksiä, joiden esittämiseen länsimaistyylisellä foneettisella aakkostolla vaaditaan enemmän pinta-alaa näyttölaitteelta. Tämä voi osaltaan selittää, miksi japanilaiselle käyttöliittymäsuunnittelulle tyypillinen korkea informaatiotiheys voi tuntua japania äidinkielenään puhuvalle käyttäjälle luontevalta. Kirjoitettu japani ei latinalaisten aakkosten tapaan sisällä tapoja korostaa tekstin merkityksiä kursiivin tai isojen kirjainten tapaisilla tavoilla, joten käyttöliittymäsuunnittelijat käyttävät usein graafisia elementtejä luomaan painotusta.

Matsuda [1985] toteaa, että kirjoitetun japanin kielen digitaalisen esittämisen vaatimukset ovat monimutkaisempia kuin latinalaisten aakkosten. Japanilaisen merkistön esittämiseen näyttölaitteella tarvitaan enemmän pikseleitä näyttölaitteelta per merkki verrattuna latinalaisiin aakkosiin. Lisäksi japanilaisen merkistön tilan tarve tietokoneen muistissa on suurempi. Tämän vuoksi tietotekniikan kehittymisellä on Japanissa ollut erilaisia vaatimuksia kuin länsimaissa. Valtaosa dokumentaatiosta tietotekniikkaan liittyen on englanninkielistä, ja on todennäköistä, että kielimuuri on hidastanut englanninkielisten tutkimustulosten käyttöönottoa. Nämä seikat lienevät osaltaan vaikuttaneet "Galapagos-ilmion" syntyyn Japanissa.

4.2. Teknologiset syyt

Toisena hypotesina kulttuurien erilaiseen käyttöliittymäsuunnitteluun suhtautumiseen on teknologian erilainen asema eri kulttuureissa. On mahdollista, että Japanissa mobiililaitteiden käyttötavat ja yleisyys ovat hyvin erilaisia verrattuna länsimaihin.

Mobiililaitteiden laajentunut käyttäjäkunta on pakottanut kehittämään uudenlaisia suunnitteluparadigmoja keskimäärin pienemmille näyttölaitteille. Tämä suuntaus on osaltaan johtanut minimalismin suosion nousuun länsimaissa. Pienemmälle näytölle ei voida sovittaa suuremman näytön näkymää suoraan pienentämällä, vaan yleinen informaatiotiheys tulisi pitää vakiona. Nilsson [2009] esittelee useita esimerkkiongelmia käyttöliittymäsuunnittelussa liittyen tilankäyttöön mobiililaitteiden näytöillä. Näitä ovat muun muassa sivun

horisontaalisen selaamisen haastavuus, elementtien esittäminen listoina sekä elementtien ryhmittely.

Älypuhelinien pienempi suosio Japanissa kuin Yhdysvalloissa voi osaltaan selittää eroa suhtautumisessa käyttöliittymäsuunnitteluun. Vuonna 2016 64,55 % yhdysvaltalaisista omisti älypuhelimien [Statista 2017a], kun taas samana vuonna 44,9 % japanilaisista omisti älypuhelimien [Statista 2017b]. Älypuhelimella tässä tarkoitetaan länsimaisessa kulttuurissa nykyisin tavallisia kosketusnäyttöllisiä puhelimia. Tämä voi selittää osaltaan, miksi japanilaisten verkkosivujen tilankäyttö on erilaista kuin länsimaissa. Japanilaisia verkkosivuja ei välttämättä pyritä suunnittelemaan yhtä kattavasti eri laitteilla käytettäväksi kuin länsimaisia verkkosivuja.

Japanilaisten matkapuhelinien erot länsimaisiin matkapuhelimiin voidaan selittää jo mainitulla ”Galapagos-ilmiöllä”: matkapuhelinteknologia Japanissa on kehittynyt selvästi eri suuntaan kuin muualla maailmassa. Vaikka ilmiön vaikutusta on viime vuosina pyritty minimoimaan, on mahdollista, että se on vaikuttanut japanilaiseen käyttöliittymäsuunnitteluun. Japanissa kosketusnäyttölliset älypuhelimet ovat nousseet suosioon vasta viime vuosina. Internetin käyttö puhelimella oli Japanissa suosittua jo 2000-luvun alussa, joten japanilaisessa käyttöliittymäsuunnittelussa on ollut tärkeää mahdollistaa paljon informaatiota sen aikaisten näyttöjen tarjoamaan pieneen tilaan. Tämä trendi voi olla näkyvässä japanilaisessa käyttöliittymäsuunnittelussa yhä nykyään. [Tabuchi 2009]

On myös mahdollista, että japanilaisessa kulttuurissa tietokoneiden länsimaisesta kulttuurista eroavat käyttötavat vaikuttavat japanilaiseen käyttöliittymäsuunnitteluun. Tabuchin [2009] mukaan suurimmalle osalle japanilaisista pääasiallinen internetin käyttöalusta on mobiililaite. Tämä todennäköisesti vahvistaa entisestään japanilaiselle mobiilikäyttöliittymäsuunnittelulle tyypillisiä piirteitä.

4.3. Muut kulttuuriset syyt

Kolmantena potentiaalisena syynä erilaisille preferensseille käyttöliittymäsuunnittelussa eri kulttuurien välillä voivat olla muut kulttuuriset erot, jotka eivät välttämättä liity puhuttuun tai kirjoitettuun kieleen suoraan. Näitä ovat muun muassa suhtautuminen erilaisiin väreihin, estetiikkaan ja erilaisiin arvo maailmoihin. Joissakin kulttuureissa esimerkiksi selkeät arvohierarkiat koetaan luonnollisina, mutta toisissa kulttuureissa ne eivät ole yhtä tärkeitä.

Kulttuurisidonnaiset elementit verkkosivujen suunnittelussa voivat parantaa verkkosivujen käytettävyyttä. Eniten kulttuurisidonnaisia elementtejä hyödyntäneet sivustot koetaan käyttäjätesteissä saavutettavimpana ja käytettävimpänä. Kulttuurisidonnaisten elementtien käyttäminen käyttöliittymäsuunnittelussa voi parantaa järjestelmän koettua käytettävyyttä kyseisessä kulttuurissa.

Jokaisella kulttuurilla on sille ominaisia käyttöliittymäsuunnittelun piirteitä. [Sun 2001]

Verratessa japanilaista kulttuuria länsimaiseen huomataan useita selkeitä eroja, jotka näkyvät myös käyttöliittymäsuunnittelussa. Japanilainen kulttuuri on länsimaiseen kulttuuriin verrattuna ryhmäorientoitunutta. Ryhmään kuulumista pidetään japanilaisessa yhteiskunnassa tärkeänä. Japanilaisessa kulttuurissa myös epävarmuuden välttäminen ja perinteiden kunnioittaminen ovat merkittäviä piirteitä. Japanilaisessa designissa on tyypillistä käyttää vanhoja, hyväksi havaittuja malleja riskialttiin uuden kehittämisen sijaan. [Singh and Matsuo 2004]

Selkeää valtahierarkiaa pidetään luonnollisena osana japanilaista yhteiskuntaa. Nämä piirteet esiintyvät myös japanilaisessa käyttöliittymäsuunnittelussa perinteitä, kuten japanilaiselle kulttuurille tärkeitä kirsikkapuun lehtiä, korostavina visuaalisina elementteinä. Selkeät hierarkkiset rakenteet ja kohteliaisuus esimerkiksi kiittämisen muodossa ovat tyypillisiä japanilaiselle käyttöliittymäsuunnittelulle. [Singh and Matsuo 2004]

Japanilaiselle designille ominainen piirre on *söpöys* (jap. kawaii). Söpöydellä pyritään japanilaisessa käyttöliittymäsuunnittelussa motivoimaan ihmisiä heidän tunteidensa kautta positiivisella tavalla. Söpöys käsitteenä on hyvin monipuolinen, ja sillä voidaan kuvata monenlaisia asioita. Söpöys voidaan määritellä neotenisten eli luonnossa huolenpitoa puoleensa vetävien piirteiden vahvana läsnäolona. Söpöjä asioita visuaalisen designin kannalta ovat esimerkiksi värikyys, pyöristetyt muodot ja muuhun vartaloon suhteessa suuret päät antropomorfisilla hahmoilla. [Cheok and Fernando 2012]

Japanilaisilla verkkosivuilla käytetään punaista väriä noin kaksi kertaa enemmän kuin yhdysvaltalaisilla tai saksalaisilla [Cyr and Trevor-Smith 2004]. Maddenin ja muiden [2000] mukaan tähän on syynä kulttuurien erilainen suhtautuminen väreihin. Punainen väri merkitsee lähes kaikissa maissa aktiivisuutta ja sitä pidetään yleisesti stimuloivana värinä. Japanissa punaisen ja valkoisen yhdistelmä merkitsee lisäksi juhlimista ja elinvoimaa, mikä voi selittää punaisen värin suuremman suosion japanilaisessa käyttöliittymäsuunnittelussa.

Russo ja Boor [1993] käyttävät esimerkkinä värien eri merkityksistä keltaista väriä, joka symboloi Yhdysvalloissa pelkuruutta, mutta Egyptissä ja Intiassa onnellisuutta, varakkuutta ja menestystä. Ottamalla värien ja muiden käyttöliittymäsuunnittelun elementtien kulttuurisidonnaiset merkitykset huomioon voidaan kohdentaa sivuston käyttöliittymän suunnittelu tiettyyn kulttuuriin.

Vuoden 2015 PISA-tutkimuksen [OECD 2015] mukaan internetin käyttö opiskelun ulkopuolella japanilaisen nuorison keskuudessa on huomattavasti vähäisempää kuin OECD-maissa keskimäärin. Tämän tutkielman johdantoluvun määritelmän mukaiseen länsimaiseen kulttuuripiiriin kuuluu 28 OECD:n 35 jäsenmaasta [OECD 2018], joten OECD:n keskiarvoisia tuloksia voidaan pi-

tää melko tarkkana kuvauksena länsimaisesta kulttuuripiiristä. Tämän kautta voidaan tehdä hypoteesi, että japanilaisessa kulttuurissa internetin käyttö viihteenä ei ole yhtä yleistä kuin länsimaisessa kulttuurissa, minkä vuoksi tarpeet käyttöliittymäsuunnittelulle ovat erilaisia. Jos japanilaisessa kulttuurissa ei ole yhtä tavallista käyttää Internetiä viihteenä kuin länsimaisessa, on mahdollista, että vaatimukset designille sisältävät vähemmän viihteellisiä koettuja arvoja.

Tässä luvussa esiteltyjä hypoteeseja on hyvin hankalaa todistaa tieteellisesti tämän tutkielman mittakaavassa, mutta niiden potentiaaliakaan on hankalaa kiistää.

5. Yhteenveto

Tässä tutkielmassa tarkasteltiin kulttuurieroja käyttöliittymäsuunnittelussa informaatiotiheyden ja näyttöjen asettelun kannalta. Tulosten perusteella nähdään, että eri kulttuureissa on jonkinlaisia eroja käyttöliittymäsuunnittelussa. Nämä erot näkyvät sekä verkkosivuissa että digitaalisissa peleissä, ja on todennäköistä, että niitä löytyy lähes kaikkialta.

Tutkielman tuloksia tarkastellessa on otettava huomioon, että informaatiotiheyttä tarkastellaan näytönkaappausten osalta suhteellisen kapea-alaisesti. Elementtien määrän voi myös laskea usealla eri tavalla. Oleellista tämän tutkielman kannalta on pyrkiä todistamaan kulttuurierojen olemassaolo näyttöjen suunnittelussa. Vaikka laskemani sivujen suhteellinen informaatiotiheys on karkeasti arvioitu, voidaan sen avulla tehokkaasti näyttää näiden erojen olemassaolo etenkin informaatiotiheyden kannalta. Tutkielman perusteella esitän hypoteesin, että japanilaisessa käyttöliittymäsuunnittelussa asteittainen paljastus ei ole yhtä suosittua kuin länsimaissa jostakin kulttuurisesta syystä.

Kulttuureja tarkasteltiin tässä tutkielmassa valtioiden kulttuuripiirien tasolla, mutta jatkotutkimuksen kannalta voisi olla arvokasta tarkastella käyttöliittymäsuunnittelun eroja esimerkiksi itsenäisten ja suuren budjetin julkaisujen välillä. Syvempi perehtyminen pelikäyttöliittymiin voisi myös olla arvokasta. Jatkotutkimusta voitaisiin tehdä myös katsepolkujen näkökulmasta tutkimalla millaisia japanilaisten käyttäjien katsepolut ovat verkkosivuilla. Katsepolkujen avulla voitaisiin myös tarkastella, miten oikealta vasemmalle äidinkielellään lukemaan tottuneet käyttäjät lukevat niiden kielten sisältöä, joita tyypillisesti luetaan vasemmalta oikealle.

Mainoksille oli tarkastelluilla japanilaisilla verkkosivuilla varattu suhteessa enemmän tilaa kuin länsimaisilla, ja mainokset ovat myös selkeästi värikkäämpiä japanilaisilla sivuilla. Mainostuksen rooli japanilaisessa verkkosuunnittelussa voisi niin ikään olla mielenkiintoinen jatkotutkimuskohde. Tässä tutkielmassa tarkasteltiin pelkästään verkkosivujen suunnittelua pöytätietokoneella, ja eräs mielenkiintoinen polku tutkimukselle voisi olla samankaltainen mobiilikäyttöliittymien vertailu.

Eräs potentiaalinen heikkous tutkielmassa on sen väkisinkin länsimainen näkökulma. Länsimaiskeskeisyyden vuoksi tutkielma tarkastelee enemmänkin japanilaisen käyttöliittymäsuunnittelun eroja länsimaiseen. Haasteita tuotti myös se, etten puhu japania. Tämä vaikeutti japaninkielisten näytönkaappaus-ten analyysia huomattavasti, eikä analyysin tarkkuudesta ole varmuutta kieli-muurin vuoksi. On pohdittava, osasinko tutkielmaa tehdessä suhtautua objek-tiivisesti eroihin erilaisissa käyttöliittymissä, vai pidinkö jollakin tavalla ana-lyysissani minulle tutumpaa länsimaista käyttöliittymäsuunnittelua standardi-na enkä tasa-arvoisena vertailukohteena.

Viiteluettelo

- Amazon. 2018. <https://www.amazon.com/>. Checked 29.4.2018.
- AmiAmi. 2018. <http://www.amiami.com/>. Checked 29.4.2018.
- Wendy Barber and Albert Badre. 1998. Culturability: The merging of culture and usability. In: *Proceedings of the 4th Conference on Human Factors and the Web*, <https://zing.ncsl.nist.gov/hfweb/att4/proceedings/barber/>. Checked 9.5.2018.
- Adrian Cheok and Owen Fernando. 2012. Kawaii/cute interactive media. *Universal Access in the Information Society* 11, 3, 295–309.
- Junjie Chu and Guang Yang. 2010. A culture-based study on information density of e-commerce websites. In: *Proceedings of the 2010 International Conference on Computer Design and Applications (ICCD)* 4, 60–63.
- Dianne Cyr and Haizley Trevor-Smith. 2004. Localization of Web design: An empirical comparison of German, Japanese, and United States Web site characteristics. *Journal of the Association for Information Science and Technology* 55, 13, 1199–1208.
- Deck13 Interactive. 2017. The Surge. Checked 3.5.2018.
- FromSoftware. 2016. Dark Souls III. Checked 26.3.2018
- Yoann Gabillon, Sophie Lepreux, and Káthia Marçal de Oliveira. 2013. Towards ergonomic user interface composition: A study about information density criterion. In: *Proceedings of the International Conference on Human-Computer Interaction*, 211–220.
- Wilbert Galitz. 2007. *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*. John Wiley & Sons.
- Tim Halverson and Anthony Hornof. 2004. Local density guides visual search: Sparse groups are first and faster. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 48, 16, 1860–1864.
- Kristine Jorgensen and Stein C. Llanos. 2011. Do players prefer integrated user interfaces? A qualitative study of game UI design issues. In: *Proceedings of the 2011 DiGRA International Conference: Think Design Play* 6.

- Terry Joyce. 2004. Modeling the Japanese mental lexicon: Morphological, orthographic and phonological considerations. *Advances in Psychology Research* 31, 27–61.
- Alfred Kroeber and Clyde Kluckhohn. 1952. Culture: A critical review of concepts and definitions. *Papers. Peabody Museum of Archaeology & Ethnology, Harvard University* 47, 1, viii, 223.
- Thomas Madden, Kelly Hewett and Martin Roth. 2000. Managing images in different cultures: A cross-national study of color meanings and preferences. *Journal of International Marketing* 8, 4, 90–107.
- Ryouichi Matsuda. 1985. Processing information in Japanese. *Computer* 18, 1, 37–45.
- Mira Milosevic. 2016. World Press Trends 2016. WAN-IFRA. 58.
- Jakob Nielsen. 1995. 10 usability heuristics for user interface design. *Nielsen Norman Group*. <https://www.nngroup.com/articles/ten-usability-heuristics/>. Checked 17.5.2018.
- Jakob Nielsen. 2006a. F-shaped pattern for reading web content. *Nielsen Norman Group*. <https://www.nngroup.com/articles/f-shaped-pattern-reading-web-content-discovered/>. Checked 17.5.2018.
- Jakob Nielsen. 2006b. Progressive Disclosure. *Nielsen Norman Group*. <https://www.nngroup.com/articles/progressive-disclosure/>. Checked 17.5.2018.
- Erik Nilsson. 2009. Design patterns for user interface for mobile applications. *Advances in Engineering Software* 40, 12, 1318–1328.
- OECD. 2015. Country Note: Japan. *Results from PISA 2015 Students' Well-being*. <https://www.oecd.org/pisa/PISA2015-Students-Well-being-Country-note-Japan.pdf>. Checked 4.4.2018.
- OECD. 2018. Members and partners. <http://www.oecd.org/about/membersandpartners/>. Checked 4.4.2018.
- Francois Pellegrino ja Cristophe Coupé. 2011. A cross-language perspective on speech information rate. *Language* 87, 3, 539–558.
- Kara Pernice. 2017. F-shaped pattern of reading on the web: Misunderstood, but still relevant (even on mobile). *Nielsen Norman Group*. <https://www.nngroup.com/articles/f-shaped-pattern-reading-web-content/>. Checked 17.5.2018.
- Patricia Russo and Stephen Boor. 1993. How fluent is your interface?: designing for international users. In: *Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems*, 342–347.
- Nitish Singh and Hisako Matsuo. 2004. Measuring cultural adaptation on the Web: A content analytic study of US and Japanese Web sites. *Journal of Business Research* 57, 8, 864–872.

- Statista. 2017a. Smartphone penetration rate as share of the population in the United States from 2010 to 2022. <https://www-statista-com.helios.uta.fi/statistics/201183/forecast-of-smartphone-penetration-in-the-us/>. Checked 2.4.2018.
- Statista. 2017b. Smartphone penetration rate as share of the population in Japan from 2015 to 2022. <https://www-statista-com.helios.uta.fi/statistics/275102/share-of-the-population-to-own-a-smartphone-japan/>. Checked 2.4.2018.
- Huatong Sun. 2001. Building a culturally-competent corporate web site: an exploratory study of cultural markers in multilingual web design. In: *Proceedings of the 19th Annual International Conference on Computer Documentation*, 95–102.
- Hiroko Tabuchi. 2009. Why Japan's cellphones haven't gone global. *New York Times* 19.
- Robin Williams. 1994. *The Non-Designer's Design Book: Design and Typographic Principles for the Visual Novice*. Peachpit Press.
- The Wall Street Journal. 2018a. <https://www.wsj.com/>. Checked 29.4.2018.
- The Wall Street Journal. 2018b. <http://jp.wsj.com/>. Checked 29.4.2018.
- Yomiuri Shimbun. 2018a. <http://the-japan-news.com/>. Checked 29.4.2018.
- Yomiuri Shimbun. 2018b. <http://www.yomiuri.co.jp/>. Checked 29.4.2018.

Maaston generoiminen fraktaalien avulla.

Tuomas Tammela

Tiivistelmä.

Virtuaaliselle maastolle on lukuisia käyttökohteita esimerkiksi peleissä ja elokuvissa, joten sen tehokas tuottaminen on jo pitkään ollut ajankohtainen tutkimuskohde. Tässä tutkielmassa tarkastelen, kuinka maastoa voidaan generoida algoritmisesti fraktaaleja hyödyntävillä menetelmillä, joita ovat muun muassa fraktaalinen kohina, timantti-neliö-algoritmi ja eroosiomallit. Maaston käsitteeseen voidaan katsoa kuuluvan pintatason lisäksi myös erilaisia lisäominaisuuksia, kuten tekstuureita ja kasvillisuutta, joten luon tutkielman lopussa lyhyen katsauksen myös näiden generoimiseen fraktaalien avulla.

Avainsanat ja -sanonnat: Maasto, maaston generoiminen, fraktaalit

1. Johdanto

Virtuaalista maastoa on kaikkialla. Valtaosa kolmiulotteisista peleistä, digitaalisista elokuvista ja simulaattoreista sisältää ainakin jonkinlaista maastoa [Shaker *et al.* 2016]. Näiden suosion noustessa myös tarve virtuaalisen maaston tuottamiselle on kasvanut. Perinteinen lähestymistapa, eli maaston luominen manuaalisesti, on kuitenkin hidas ja kallis prosessi, joten tehokkuuden parantamiseksi siihen on kehitelty vuosien varrella monenlaisia algoritmisia menetelmiä.

Tutkielman tavoitteena on tarkastella, miten fraktaaligeometriaa voi käyttää hyödyksi maaston algoritmisessa generoinnissa. Aloitan luvun 2 kertomalla yleisellä tasolla maaston generoimisesta korkeuskarttaa käyttäen. Tämän jälkeen kerron, mitä fraktaalit ovat ja miten niitä voi soveltaa tähän prosessiin. Luvun 2 lopussa esittelen vielä maaston algoritmisen generoimisen etuja ja ongelmia verrattuna maaston luomiseen manuaalisesti.

Luvussa 3 käyn läpi erilaisia käytännön menetelmiä maaston generoimiseen fraktaalien avulla. Ensimmäinen menetelmä on kohina ja fraktionaalinen Brownin liike, joita voidaan soveltaa korkeuskartan luomiseen. Toinen menetelmä on timantti-neliö-algoritmi, jonka avulla voidaan suoraan manipuloida korkeuskartan arvoja. Kolmas on eroosiomalli, jolla voidaan luoda eroosion merkkejä johonkin olemassa olevaan korkeuskarttaan. Viimeisenä on interaktiivinen menetelmä, jossa yhdistyvät maaston manuaalinen luominen ja algoritmisen generointi.

Luvussa 4 tarkastelen lyhyesti maaston lisäominaisuuksien generoimista fraktaaligeometriaa hyödyntäen. Lisäominaisuuksia ovat kaikki maastoon kuuluvat osat, jotka eivät varsinaisesti ole osa sen korkeuskarttaa, kuten esimerkik-

si tekstuurit ja kasvillisuus. Näitä kahta käsitellään luvussa tarkemmin käytännön esimerkkien avulla. Tutkielman lopussa olevassa luvussa 5 kokoan yhteen tutkielman tulokset ja vastaan asettamaani tutkimuskysymykseen.

2. Maaston generoiminen

Maaston generoiminen tarkoittaa käytännössä maastoa muistuttavan kolmiulotteisen mallin luomista algoritmien avulla. Tämä voidaan toteuttaa esimerkiksi *korkeuskartan* (heightmap), *vektorikentän* (vector field) tai *vokseleiden* (voxel) avulla. [Shaker *et al.* 2016.] Tässä tutkielmassa keskitytään lähinnä korkeuskarttaan, koska se tukee parhaiten fraktaaleja hyödyntäviä algoritmeja. Kuvassa 1 on esimerkki korkeuskarttaan pohjautuvasta maastosta, johon on lisätty tekstuureita, kasvillisuutta ja ilmakehän vaikutuksia.



Kuva 1. Terragen-ohjelmistolla luotu korkeuskartta [Planetside Software 2016].

Korkeuskartan pohjalla on kaksiulotteinen, reaalitylukuja sisältävä matriisi. Tämän matriisin leveys ja korkeus vastaavat jonkin neliskulmaisen tason x - ja y -koordinaatteja. Matriisin sisältämät arvot taas vastaavat tason z -koordinaatteja eli pisteiden korkeusarvoja. Jos tason resoluutio eli pisteiden määrä on suurempi kuin matriisin, väliin jäävien pisteiden korkeudet voidaan helposti interpoloida niiden naapurien arvojen perusteella. Käytännössä tämä tarkoittaa, että mitä tahansa kaksiulotteiseksi matriisiksi muunnettavissa olevaa dataa voidaan käyttää maaston generoimiseen. [Shaker *et al.* 2016.]

Tällaista dataa voidaan luoda algoritmisesti monella eri tavalla. Näitä käsitellään tarkemmin luvussa kolme, joten niistä ei kerrota tässä sen enempää. Kaikille tutkielmassa käsitellyille menetelmille on yhteistä, että ne soveltavat jollain tavoin fraktaaligeometriaa. Käsite ei ole itsestään selvä, joten sitä on syytä tarkentaa ennen seuraavaan lukuun siirtymistä.

2.1. Fraktaalien käyttö maaston generoimisessa

Fraktaalit (fractal) ovat geometrisia muotoja, joissa samat kuviot toistuvat uudelleen ja uudelleen useilla eri mittakaavoilla. Fraktaalien yksittäinen osa näyttää siis samalta kuin koko fraktaali. Näitä hyödyntävät menetelmät soveltuvat erinomaisesti maaston generoimiseen, koska maasto koostuu reaaliajassaakin fraktaaleista. [Musgrave 1993.] Esimerkiksi mantereiden rantaviivaa tarkasteltaessa 100 kilometriä näyttää hyvin samanlaiselta kuin 100 metriä.

Käytännössä fraktaaleja voidaan käyttää maaston generoimiseen useilla eri tavoilla. Jotkin tekniikat, kuten timantti-neliö-algoritmi, pohjautuvat suoraan fraktaalimatematiikkaan, jolloin niitä voidaan suoraan soveltaa korkeuskartan manipuloimiseen. Toiset tekniikat, kuten fraktionaalinen Brownin liike, puolestaan tuovat fraktaalimaisia piirteitä jo olemassa oleviin menetelmiin, kuten kohinaan, jolloin niitä ei voi käyttää yksinään. [Shaker *et al.* 2016.] Kuten sanottua, näitä käsitellään tarkemmin luvussa 3. Tässä vaiheessa on kuitenkin hyvä tarkastella, mitä hyötyä tällaisista algoritmeista käytännössä on.

2.2. Maaston algoritmisen generoinnin etuja ja ongelmia

Yksi suurimmista eduista maaston algoritmisessa generoimisessa on sen tehokkuus. Suuria vuoristoja, erämaita tai jopa valtamerien pohjia voidaan luoda hetkessä, kun käsin tehtynä siihen voisi helposti kulua viikkokausia. [Smelik *et al.* 2015.] Ihmistyövoimaa tarvitaan siis paljon vähemmän, mikä vähentää kustannuksia ja nopeuttaa kehitystyötä. Samalla se antaa myös pienemmille yrityksille mahdollisuuden luoda sellaisia maastoa käyttäviä sovelluksia, joihin heidän resurssit eivät muuten riittäisi.

Algoritmien avulla maastoa voidaan myös luoda käytännössä äärettömän paljon. Tästä on suuresti apua sovelluksissa, joissa maastoa tarvitaan valtavia määriä, kuten esimerkiksi lentosimulaattoreissa. Mantereiden tai jopa kokonaisten planeettojen maastojen luominen käsin ei olisi millään tavoin realistista. [Shaker *et al.* 2016.] Tämän lisäksi algoritmit mahdollistavat myös loputtomasti jatkuvan maailman luomiseen, jossa uutta maastoa luodaan reaaliajassa sitä mukaa, kun sitä tarvitaan.

Sovelluksissa, joissa maasto luodaan reaaliajassa, algoritmeista on vieläkin enemmän hyötyä. Algoritmit vievät esimerkiksi huomattavasti vähemmän tallennustilaa kuin kokonaiset, valmiiksi luodut maastomallit. Erilaisten parametrien käyttö luontivaiheessa tekee maastosta myös aina erilaisen ja ennalta-arvaamattoman, jolloin esimerkiksi seikkailupelit säilyvät mielenkiintoisina pelikerrasta toiseen. [Smelik *et al.* 2015.] Käyttäjälle voi myös antaa mahdollisuuden muokata parametreja, jolloin hän voi itsekin vaikuttaa generoinnin lopputulokseen.

Maaston algoritmisessa generoinnissa on kuitenkin myös haittapuolia. Yksi tällainen on esimerkiksi luodun sisällön sattumanvaraisuus, mitä voi olla vaikea hallita. Käyttäjää ei voi esimerkiksi tarkasti määritellä, mistä kohtaa vuoriston läpi pääsee kulkemaan, mikä voi tuottaa ongelmia peleissä liikkumisen kannalta. Tämä saattaa myös johtaa siihen, että käyttäjä lähtee etsimään juuri oikeanlaisia maastonmuotoja generoimalla niitä uudestaan ja uudestaan, mikä on turhauttavaa ja aikaa vievää. [van der Lei 2010.]

Algoritmisesti luotu maasto ei myöskään sovellu kovin hyvin ihmisten asuttamiin ympäristöihin. Reaalimaailmassa maaseutujen ja kaupunkien maastonmuodot ovat usein tulosta satojen vuosien rakentamisesta, kaavoittamisesta ja maanmuokkauksesta, mitä on vaikea ottaa huomioon maastoa generoitaessa. [van der Lei 2010.] Algoritmiset menetelmät rajoittuvat siis lähinnä koskemattomien erämaiden ja syrjäseutujen luomiseen.

3. Menetelmiä

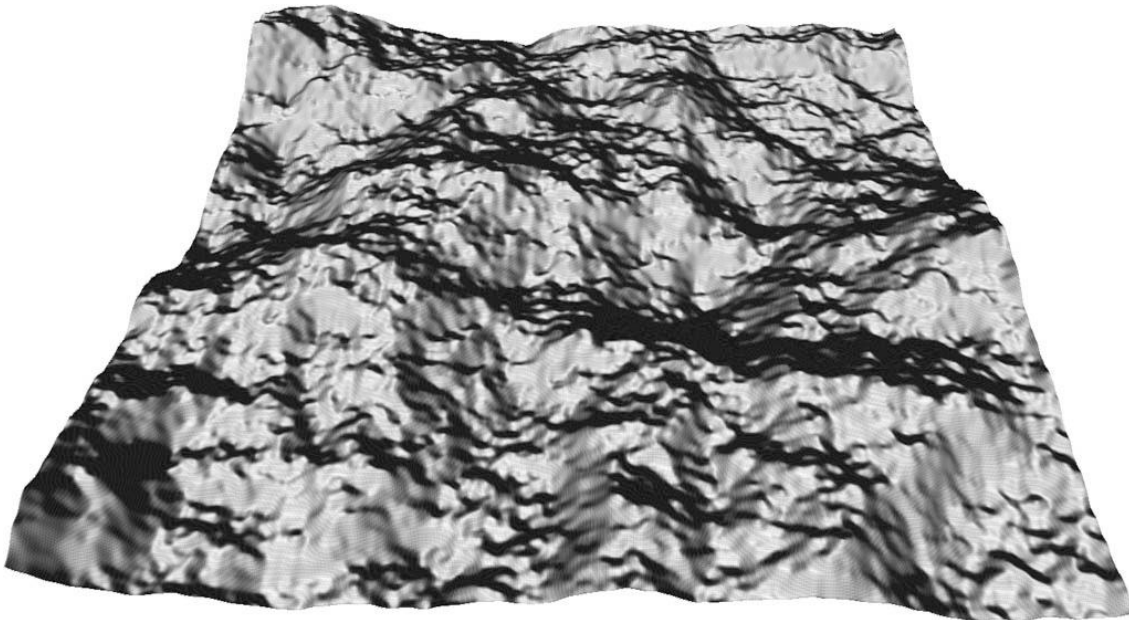
Tässä luvussa esitellään ensin kaksi maaston generoimiseen soveltuvaa, fraktaaligeometriaa hyödyntävää menetelmää: kohinan muuntaminen korkeuskartaksi fraktionaalisen Brownin liikkeen avulla ja timantti-neliö-algoritmin käyttäminen. Tämän jälkeen luodaan katsaus eroosiomalleihin, joiden avulla voidaan parantaa olemassa olevan korkeuskartan realistisuutta. Lopuksi tarkastellaan vielä interaktiivista lähestymistapaa maaston generoimiseen.

3.1. Kohina ja fraktionaalinen Brownin liike

Yksi suosituimmista tekniikoista maaston generoimiseen on *kohinan* (noise) soveltaminen korkeuskarttaan. Kohinalla tarkoitetaan funktiota, joka ottaa minkä tahansa n -ulotteisen pisteen koordinaatit ja palauttaa siitä näennäisesti satunnaisen reaaliluvun (näennäisellä tarkoitetaan, että sama piste palauttaa kuitenkin aina saman luvun). Näin saadaan aikaan monenlaisia mielenkiintoisia muotoja ja kuvioita. [Hyttinen 2017.] Arvojen satunnaisuudesta johtuen kohina ei kuitenkaan tuota kovin uskottavaa maastoa, jolloin se ei suoraan sovellu korkeuskartaksi. Kohinan realistisuutta voidaan kuitenkin parantaa esimerkiksi fraktaalifunktioiden avulla.

Eräs tällainen funktio on *fraktionaalinen Brownin liike* (fractional Brownian motion). Menetelmässä kohinaa summataan yhteen itsensä kanssa siten, että jokaisen summattavan kohinan taajuutta kasvatetaan ja amplitudia pienennetään. Taajuudella tarkoitetaan käytännössä kohinan yksityiskohtien määrää ja amplitudilla niiden kokoluokkaa. [Musgrave 1993.] Näin kohinassa saadaan yhdistettyä useiden eri kokoluokkien yksityiskohtia, mikä johtaa huomattavasti hienojakoisempaan lopputulokseen.

Fraktionaalinen Brownin liike soveltuu hyvin vuoristojen generoimiseen, koska se tuottaa samanlaista yksityiskohtien toistuvuutta kuin reaalimaailman eroosioprosessit. [Musgrave 1993.] Kuvassa 2 on esitelty kyseisellä menetelmällä tuotettu korkeuskartta. Korkeuskartasta näkee, että suuressa mittakaavassa vuoret nousevat ylös laaksoista, pienemmässä esivuoret nousevat vuoren rinteiltä ja vielä pienemmässä yksittäiset kallionkielekkeet nousevat esivuorten rinteiltä. Sama kolmiomainen muoto toistuu siis yhä uudelleen ja uudelleen.



Kuva 2. Kohinalla ja fraktionaalisella Brownin liikkeellä luotu korkeuskartta [de Carpentier 2011].

Fraktionaalisessa Brownin liikkeessä on kuitenkin mukana myös puutteita, sillä sen avulla tuotettu maasto on statistisesti symmetristä, mikä tarkoittaa, että korkeuskartta näyttäisi samanlaiselta, vaikka sen kääntäisi ylösalaisin. Tämä ei vastaa todellista maastoa, jossa matalammat alueet täyttyvät ajan myötä kaikenlaisella soralla ja liejulla, johtaen tasaisempien alueiden muodostumiseen painaumien ja laaksojen pohjille. [Musgrave *et al.* 1989.]

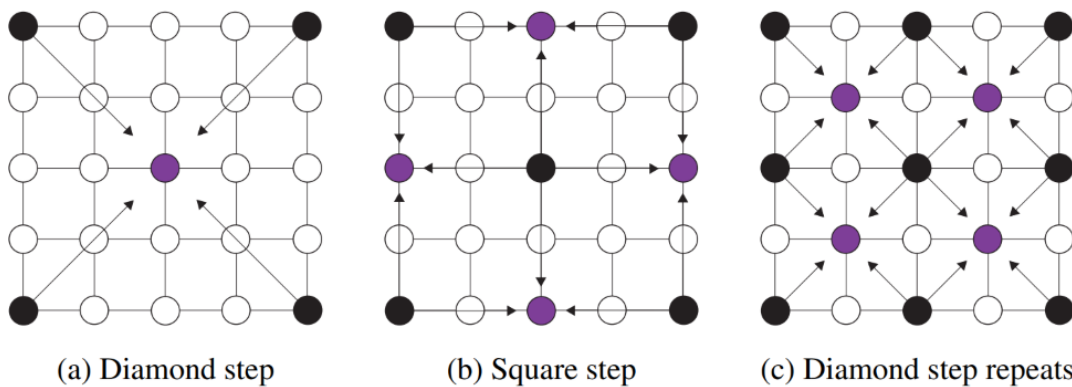
Fraktionaalinen Brownin liike ei myöskään kykene tuottamaan globaaleja eroosion merkkejä, kuten esimerkiksi jokiverkostoja, suistoja, tulvakertymiä tai louhikoita. Näillä on suuri merkitys todellisen maaston muotoutumisessa, joten niiden puuttuminen tekee maastosta hyvin yksipuolisen ja epäuskottavan. [Musgrave *et al.* 1989.] Näiden lisäksi menetelmä on myös hidas, koska sen täytyy summata yhteen useita kohinan ilmentymiä yksi kerrallaan. Tämä on tosin merkittävä ongelma vasta todella suurilla kohinoilla.

3.2. Timantti-neliö-algoritmi

Toinen keino luoda korkeuskarttoja on *timantti-neliö-algoritmin* (diamond-square algorithm) käyttäminen. Tämä eroaa fraktionaalisesta Brownin liikkeestä siten, että timantti-neliö-algoritmin avulla korkeuskarttaa voidaan manipuloida suoraan ilman kohinan käyttämistä välivaiheena.

Timantti-neliössä otetaan ensin tasainen korkeuskartta ja alustetaan sen neljä kulmaa mihin tahansa korkeusarvoihin. Seuraavaksi haetaan näiden neljän pisteen yhteinen keskipiste ja asetetaan sen korkeusarvoksi kulmien korkeuksien keskiarvo plus jokin sattumanvarainen arvo, joka siis joko nostaa tai laskee pisteen korkeutta. Tätä vaihetta kutsutaan timanttivaiheeksi. Seuraavaksi otetaan kulmien muodostaman neliön sivujen keskipisteet ja asetetaan niihin kahden vierekkäisen kulman ja neliön keskipisteiden korkeusarvojen keskiarvot. Samoin kuin edellä, tähänkin arvoon lisätään vielä jokin sattumanvarainen arvo. Tätä vaihetta kutsutaan neliövaiheeksi. [Shaker *et al.* 2016.]

Algoritmi jatkuu siirtymällä takaisin timanttivaiheeseen, jossa alkuperäinen neliö jaetaan neljään yhtä suureen osaan, ja uusiksi kulmiksi asetetaan näiden neliöiden keskipisteet. Timantti- ja neliövaiheita toistetaan niin kauan, että saavutetaan korkeuskartan haluttu tarkkuus. [Shaker *et al.* 2016.] Kuvassa 3 on esitelty visuaalisesti algoritmin vaiheet: kohdassa a on timanttivaihe, kohdassa b neliövaihe ja kohdassa c timanttivaiheen toinen suorituskertana.



Kuva 3. Timantti-neliö-algoritmin vaiheet [Shaker *et al.* 2016]

Timantti-neliö-algoritmin muodostaman korkeuskartan epätasaisuutta voidaan säädellä asettamalla sattumanvaraisille arvoille suuruusluokat. Suurilla arvoilla saadaan karkeaa ja vuoristoista maastoa, kun taas pienillä arvoilla maasto pysyy verrattain tasaisena. Tätä voidaan käyttää hyödyksi fraktaalimaisten ominaisuuksien tuomiseksi korkeuskarttaan: kun suuruusluokkaa pienennetään jokaisen timantti-neliö-iteraation jälkeen, saadaan korkeuskarttaan usean eri suuruusluokan yksityiskohtia. [Olsen 2004.] Näin korkeuskartasta saadaan paremmin todellista maastoa muistuttava.

Timantti-neliö-algoritmissa on lisäksi se hyvä puoli, että jokaisen pisteen arvo voidaan halutessa laskea myös yksitellen. Täten korkeuskarttaan voidaan tuoda esimerkiksi eroosion vaikutuksia laskemalla suuruusluokan koko erikseen jokaiselle pisteelle niiden naapuripisteiden korkeuksien keskiarvojen perusteella. Näin esimerkiksi matalammat alueet muodostuvat tasaisemmiksi, mikä simuloi maa-aineksen kertymistä laaksojen pohjille reaalimaailmassa. [Olsen 2004.] Samalla tosin korkeuskartan generoiminen hidastuu, joten ratkaisu ei välttämättä ole aina kovin käytännöllinen.

Joka tapauksessa timantti-neliö-algoritmi on kuitenkin huomattavasti tehokkaampi kuin fraktionaalinen Brownin liike, koska jokaiselle korkeuskartan pisteelle täytyy laskea arvo vain kerran sen sijaan, että summattaisiin yhteen useita kohinan ilmentymiä. [Olsen 2004.] Näin ollen timantti-neliö-algoritmi soveltuu erityisen hyvin reaaliaikaiseen maaston generoimiseen.

3.3. Eroosiomallit

Timantti-neliö-algoritmin avulla voidaan karkeasti simuloida eroosion vaikutuksia, mutta siinä on silti sama ongelma kuin fraktionaaliossa Brownin liikkeessä: se ei kykene tuottamaan globaaleja eroosion merkkejä, kuten esimerkiksi jokiverkostoja. Näitä on kuitenkin mahdollista tuoda olemassa olevaan korkeuskarttaan soveltamalla siihen jotain eroosiomallia. Malleja on useita erilaisia ja niillä voidaan simuloida esimerkiksi veden virtauksen, mannerlaattojen liikkeen tai lämpötilan vaihtelun aiheuttamaa eroosiota. Tässä tutkielmassa keskitytään erityisesti vesieroosiomallin toimintaan.

Vesieroosiomallilla (hydraulic erosion model) simuloidaan prosessia, jossa korkeuskartan jokaiseen huippukohtaan pudotetaan vettä, jonka annetaan valua alas maaston muotoja mukaillen. Veden liikkeet kuluttavat maastoon uurreita ja kuljettavat mukanaan sedimenttiä. Matalampiin alueisiin kasautuu siis korkeammilta alueilta tullutta maa-ainesta. [Musgrave *et al.* 1989.]

Käytännössä tämä toteutetaan niin, että jokaiselle korkeuskartan huippukohtalle määritellään korkeus, veden määrä ja vedessä olevan sedimentin määrä. Tämän jälkeen käydään askel askeleelta läpi viereisiä, matalammalla olevia pisteitä alenevassa järjestyksessä. Jokaisella askeleella pisteisiin siirretään vettä niiden korkeusarvojen perusteella, ottaen kuitenkin huomioon myös veden pinnan aiheuttama muutos korkeuteen. Veden kokonaismäärä on vakio, ja se jaetaan kaikkien viereisten pisteiden kesken siten, että enemmän alempana olevat pisteet saavat enemmän vettä. [Musgrave *et al.* 1989.]

Vedessä olevan sedimentin määrä alkaa ensin nolasta, ja kasvaa veden virratessa alaspäin. Jokaisesta ohitetusta pisteestä joko poistetaan maa-ainesta tai lisätään sedimenttiä sen korkeusarvon perusteella. Jyrkät rinteet siis kasvattavat sedimentin määrää, kun taas loivat rinteet ja tasangot pienentävät sitä.

Veden saavuttaessa rinteiden juuren loput sedimentistä lisätään tasaisesti ympäröivien pisteiden korkeusarvoihin. [Musgrave *et al.* 1989.]

Kuvassa 4 on esitelty korkeuskartta, johon on sovellettu vesieroosiomallia. Siinä on näkyvissä, kuinka simuloitujen veden liikkeet ovat kaivertaneet vuorten rinteitä ja tasoittaneet painaumien pohjia. Eroosiomallin avulla korkeuskartasta saadaan siis huomattavasti luonnollisemman näköinen.



Kuva 4. Vesieroosion kuluttama korkeuskartta [de Carpentier 2012].

Eroosiomallien etu on siinä, että niitä voi soveltaa mihin tahansa korkeuskarttaan. Malleja voidaan siis käyttää yhdessä esimerkiksi edellä mainittujen fraktionaalisen Brownin liikkeen tai timantti-neliö-algoritmin kanssa. Lisäksi eroosiomallien tuottamat vaikutukset ovat fraktaaleita, joten ne näyttävät samantyyppisiltä useilla eri mittakaavoilla (riippuen tietenkin parametreista, kuten veden määrästä). Näin malleilla saadaan aikaan samankaltaisia lopputuloksia korkeuskartan koosta riippumatta.

Eroosiomallit ovat kuitenkin erittäin hitaita menetelmiä (aikakompleksisuus $O(n)$). Prosessin vaikutukset eivät tule selkeästi näkyviin ensimmäisellä suorituskerralla, joten realistisen maaston saavuttamiseksi sitä täytyy toistaa useita kertoja peräkkäin. Toistaminen voi tuottaa ongelmia erityisesti suurilla ja yksityiskohtaisilla korkeuskartoilla, joilla suorituskertoihin kuluu muutenkin paljon aikaa. [Musgrave *et al.* 1989.] Näytönohjaimet ovat tosin parantuneet huomattavasti Musgraven ja muiden [1989] ajoista, joten eroosiomallien teho-vaatimukset eivät ole enää nykyään yhtä suuri ongelma.

3.4. Interaktiivinen menetelmä

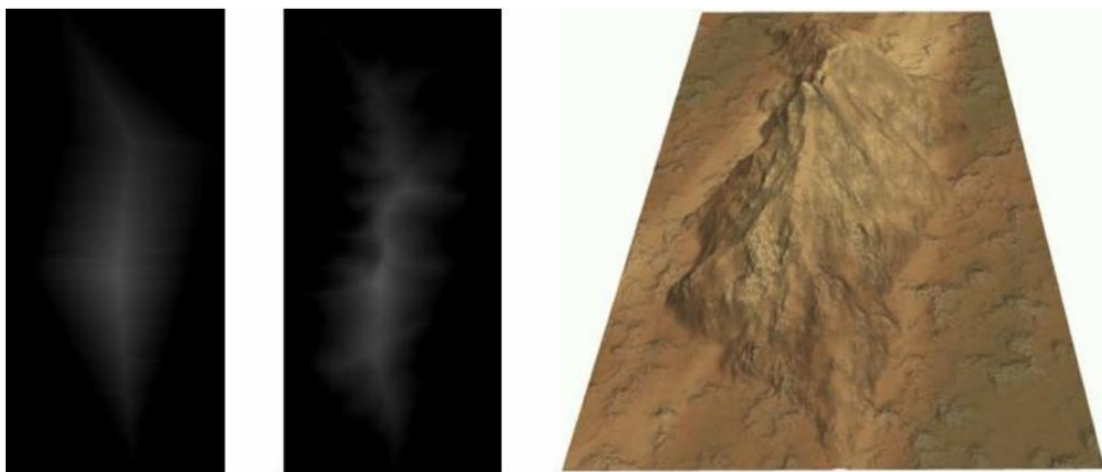
Kaikilla edellä mainituilla menetelmillä on yksi yhteinen heikkous: niiden käyttäjä ei voi tarkkaan määrittellä luodun korkeuskartan maastonmuotoja. Para-

metrien avulla voidaan toki määritellä, kuinka korkeita vuoret tulevat olemaan tai kuinka kuluneita niiden rinteet ovat. Tästä ei kuitenkaan ole paljoa apua, kun halutaan luoda juuri tietynlainen ympäristö, jossa vuorenhuiput, tasangot ja rannikot ovat tarkkaan määritetyillä paikoilla. Perinteisesti tässä kohdin onkin täytynyt hylätä algoritminen lähestymistapa ja tyytyä luomaan maasto manuaalisesti. Chiang ja muut [2005] esittävät kuitenkin menetelmän, jossa yhdistyvät algoritmisen generoimisen tehokkuus ja manuaalisen työn tarkkuus.

Kyseisessä menetelmässä käyttäjä luo ensin hyvin yksinkertaisen pohjapiirroksen maastosta asettamalla kartalle kaksiulotteisia geometrisia kuvioita, kuten kolmioita, neliöitä ja suunnikkaita. Näille kuvioille määritellään automaattisesti huippukohdat ja korkeusarvot niiden koon perustella, jolloin ne saavat myös kolmannen ulottuvuuden. Lopputuloksena on karkea kuvaus korkeuskartasta, jossa on esillä suuren mittakaavan yksityiskohdat, kuten vuorisot, laaksot ja tasangot. [Chiang *et al.* 2005.]

Seuraavaksi korkeuskartan muotoja verrataan reaali maailmasta poimituihin maastonpalasiin. Nämä palaset on valittu käsin siten, että ne koostuvat tasan yhdestä vuorenhuipusta tai -harjanteesta. Näin niitä voidaan helposti vertailla käyttäjän valitsemiin muotoihin, jotka koostuvat samalla tavalla vain yksittäisistä huippukohtista. Maastonpalasista valitaan lähimmät vastineet käyttäjän antamille muodoille, minkä jälkeen ne lisätään vastaaville paikoilleen korkeuskartassa. [Chiang *et al.* 2005.]

Kuvassa 5 on esitelty interaktiivisen menetelmän vaiheet. Vasemmalta katsoen ensimmäisessä kuvassa on karkea pohjapiirros, joka koostuu yhdestä kolmiosta ja yhdestä suunnikkaasta. Keskimmaisessä kuvassa on yhdistetty näitä parhaiten vastanneet todellisen maaston palaset. Viimeisessä kuvassa on menetelmän lopputuloksena saatu korkeuskartta.



Kuva 5. Interaktiivisella menetelmällä luotu korkeuskartta [Chiang *et al.* 2005].

Lyhyesti voidaan sanoa, että interaktiivinen menetelmä yhdistää käyttäjän määrittelemät suuret yksityiskohdat ja reaali maailman pienet yksityiskohdat. Menetelmällä tuotettu maasto on samanlaista kuin prosessiin valitut todellisen maaston palaset, joten se on jo oletuksena hyvin uskottavaa. Maaston generoiminen on myös nopeaa, koska palasissa on valmiiksi kaikki yksityiskohdat ja eroosion merkit, jolloin laskentaa tarvitsee tehdä vain suurella mittakaavalla. [Chiang *et al.* 2005.] Lisäksi palaset mahdollistavat myös maaston räätälöimisen todellisen maailman alueiden, kuten esimerkiksi Alppien, Kalliovuorten tai Norjan vuonojen, mukaan.

Interaktiivisen menetelmän valmisteluvaihe on kuitenkin nykyisellään erittäin hidaskäyttöinen ja työläs. Jokainen yksittäinen maastonpalanen täytyy valita käsin, joten kattavan kokoelman luomiseen kuluu helposti todella paljon aikaa. [Chiang *et al.* 2005.] Menetelmässä on kuitenkin se hyvä puoli, että kun kokoelma on kerran luotu, sitä voidaan käyttää uudelleen rajattomasti. Se soveltuu siis hyvin sellaisiin käyttökohteisiin, joissa maastoa on tarve generoida usein, kuten esimerkiksi videopelien kenttäeditoreissa.

4. Maaston lisäominaisuuksien generoiminen

Käytännön sovelluksissa maasto koostuu harvoin vain pelkästä korkeuskartasta. Useimmiten sen päälle lisätään myös monenlaisia lisäominaisuuksia, kuten esimerkiksi tekstuureita, kasvillisuutta, vesistöjä, ilmankin vaikutuksia, tieverkostoja tai rakennuksia. Nämä ovat olennainen osa lopullista maastoa, joten niitä on syytä käsitellä myös tässä tutkielmassa. Tässä luvussa tarkastellaankin, miten maaston lisäominaisuuksia voidaan generoida algoritmisesti fraktaaligeometriaa hyödyntäen, käyttäen esimerkkinä tekstuureja ja kasvillisuutta.

4.1. Tekstuurien generoiminen

Tekstuureja käytetään lähes universaalisti korkeuskarttoja hyödyntävissä sovelluksissa. Niiden avulla maaston raakaan kolmiulotteiseen malliin saadaan tuotua väriä, yksityiskohtia ja mittakaavan tuntua. Realististen tekstuurien luominen manuaalisesti on kuitenkin hankalaa ja aikaa vievää, koska niiden täytyy sopia yhteen saumattomasti ja muodostaa jatkuva kokonaisuus. Tekstuurit vievät myös paljon tallennustilaa, mikä asettaa rajoitteita niiden koolle ja määrälle käytännön sovelluksissa. Yksi ratkaisu näihin ongelmiin on generoida tekstuureja matemaattisesti, jolloin ne sopivat varmasti hyvin yhteen ja vievät vain murto-osan valmiiden tekstuurien tallennustilasta [Pietroni *et al.* 2010].

Fraktaaleita pystytään hyödyntämään tekstuurien generoinnissa useilla eri tavoilla. Suoraviivaisin tapa on muuntaa jokin fraktaalialgoritmi suoraksi kaksikulotteiseksi kuvioksi ja lisätä siihen sopivia värejä. Nämä sopivat hy-

vin sellaisiin pintoihin, jotka ovat todellisessa maailmassakin fraktaaleja, kuten esimerkiksi jäähän, marmoriin tai kristalliin. Lisäksi näillä saadaan aikaan myös luonnollisia tuli-, sähkö- ja pilviefektejä. [Pietroni *et al.* 2010.] Tämä menetelmä on kuitenkin hyvin rajallinen, koska sitä voidaan soveltaa vain muutamiiin tarkasti määriteltyihin materiaaleihin.

Miyata ja muut [2001] esittelevät tähän hieman kehittyneemmän lähestymistavan, jossa tekstuuri luodaankin yhdistelemällä pieniä rakennusosasia, kuten esimerkiksi kiviä, tiiliä tai suomuja. Nämä osat luodaan lähes kokonaan algoritmisesti, joten käyttäjä voi määritellä niille muun muassa keskimääräiset muodot, koot, korkeudet, väriryhmät sekä optiset ominaisuudet. Fraktaalit ovat prosessissa tärkeässä osassa, koska niiden avulla jokaisesta yksittäisestä rakennusosasta saadaan hieman eri näköinen ja muotoinen. [Miyata *et al.* 2001.]

Prosessin alussa tekstuurille määritellään ensin muoto, joka voi olla käytännössä mikä tahansa. Seuraavaksi kaikille tekstuurin rakennusosasille luodaan kolmiulotteiset mallit, jotka asetetaan tekstuurin muotoon. Tässä vaiheessa käytetään hyödyksi fraktaalikohinaa, jonka avulla osasten malleista saadaan karkeita ja epätasaisia. Kaikille eri osasille käytetään erilaista kohinaa. Kolmannessa vaiheessa mallin yläpuolella asetetaan vielä valonlähde (varjoja varten) ja siitä muodostetaan kaksiulotteinen tekstuurikartta. [Miyata *et al.* 2001.]

Kuvassa 6 on esitelty kaksi tällä menetelmällä muodostettua jalkakäytävän tekstuuria. Niiden osien muodoille on asetettu erilaisia parametreja, joten kohdassa B ne muistuttavat mukulakiviä, ja kohdassa C laattakiviä. Kuvasta näkee, kuinka menetelmällä voidaan luoda algoritmisesti hyvinkin realistisia ja luonnollisia tekstuureita.



(b) Cobblestone pavement



(c) Flagstone pavement

Kuva 6. Palasista rakennettu tekstuuri [Miyata *et al.* 2001].

Menetelmän etuna on, että sen avulla voidaan luoda juuri oikean kokoisia ja muotoisia tekstuureita. Sillä voidaan siis luoda esimerkiksi orgaanisia polkuja, jotka mutkittelevat, levenevät ja kapenevat maaston muotojen mukaan. Tämä olisi hankalaa toteuttaa tavanomaisilla tekstuureilla, jotka koostuvat useimmiten suurista neliöistä. Menetelmällä luodut tekstuurit sopivat myös hyvin yhteen muiden tekstuurien kanssa, koska ne koostuvat saumattomista osasista. Esimerkiksi kuvassa 6 esitellyn mukulakivipolun voisi helposti lisätä jonkin ruohikkoisen tekstuurin päälle siten, että ruohikkoa jäisi näkyviin kivien reunoille ja niiden muodostamiin aukkoihin.

Menetelmä on kuitenkin samalla tavalla rajoitettu kuin puhtaat fraktaalikuviot, koska sillä voidaan kuvata vain tiettyjä algoritmiin soveltuvia materiaaleja. Mahdollisten tekstuurien määrä on tosin paljon laajempi, koska osasia voidaan yhdistellä monilla eri tavoilla.

Menetelmä ei myöskään ole kovin tehokas, koska jokaisen yksittäisen osan muodot täytyy laskea erikseen. Tehottomuus on ongelma varsinkin suurissa tekstuureissa, joissa osasten lukumäärä voidaan laskea tuhansissa. Menetelmä soveltuukin parhaiten vain pienten yksityiskohtien, kuten kivipolkujen, generoimiseen muiden tekstuureiden tueksi. [Miyata *et al.* 2001.]

4.2. Kasvillisuuden generoiminen

Kasvillisuudella tarkoitetaan kaikkia maaston pinnalle lisättäviä kasveja, kuten esimerkiksi puita, pensaita ja ruohikoita. Näillä on erittäin suuri vaikutus maaston realistisuuden kannalta, koska ilman niitä maasto olisi pelkkää tyhjää, paljasta tasoa. Kasvillisuutta käytetäänkin valtaosassa maastoa hyödyntävistä sovelluksista. Kasvuston käytössä on kuitenkin se ongelma, että sitä tarvitaan usein valtavia määriä. Esimerkiksi kaupallisissa lentosimulaattoreissa voi helposti olla yli 500 miljoonaa yksittäistä puuta. [Hammes 2001.] Kasvillisuuden lisääminen manuaalisesti on siis harvoin käytännöllistä, minkä vuoksi siihen onkin kehitetty monenlaisia algoritmisia menetelmiä.

Fraktaaleja hyödyntävät menetelmät soveltuvat hyvin kasvillisuuden generoimiseen, koska monien kasvien ja ekosysteemien levinneisyys osoittaa fraktaalimaisia piirteitä. Tämä johtuu siitä, että useat kasvilajit lisääntyvät joko pudottamalla siemeniä ympärilleen tai kasvattamalla jälkeläisiä juuristostaan. Tällaiset kasvit muodostavat tiheämpiä joukkoja toistensa ympärille sekä suurilla että pienillä mittakaavoilla. Niiden levinneisyyttä voidaan siis karkeasti kuvata fraktaalisen kohinan avulla. [Hammes 2001.]

Pelkällä kohinalla ei tosin saada aikaan kovin uskottavaa kasvillisuusjakamaa, koska se ei ota huomioon maaston muotoja. Esimerkiksi vuoren huipulla olisi tällöin keskimäärin yhtä paljon kasvillisuutta kuin laakson pohjalla. Realistisuuden saavuttamiseksi kohinaan täytyy siis yhdistää muitakin tekijöi-

tä. Hammes [2001] esittelee tekijöistä neljä: absoluuttinen korkeus, suhteellinen korkeus (lähiympäristöön nähden), maanpinnan kaltevuus ja kaltevuuden suunta. Näiden perusteella asetetaan todennäköisyys tiettyjen kasvilajien esiintymiselle kohinan määrittämällä alueilla. [Hammes 2001.] Tällöin esimerkiksi korkeammilla alueilla olisi vain vähän puita, mutta paljon ruohikkoa.

Deussen ja muut [1998] esittelevät tähän hieman toisenlaisen lähestymistavan. Heidän kehittämänsä menetelmä ottaa syötteekseen maaston korkeuskartan ja kasvillisuuden alkujakauman (joka voi olla esimerkiksi fraktaalinen kohina), minkä jälkeen se simuloi kasvillisuuden leviämistä iteratiivisesti askel askeleelta. Simulaatiossa otetaan huomioon kasvien ekologiset ominaisuudet, kuten koko, kasvunopeus ja veden tarve. Lisäksi huomioidaan myös saatavilla olevan maaperän, veden ja auringonvalon määrä. Simulaation aikana huonoissa elinolosuhteissa olevat kasvit kuolevat, kun taas hyvillä paikoilla olevat kasvit leviävät entisestään. [Deussen *et al.* 1998.] Näin saadaan aikaan luonnollinen, todellista maailmaa vastaava kasvillisuusjakauma.

Kuvassa 7 on esitelty tällainen jakauma. Siinä on näkyvässä, kuinka saman kasvilajin edustajilla on luonnollinen taipumus ryhmittyä toistensa lähelle, osoittaen fraktaalimaisia piirteitä. Hammesin [2001] menetelmällä päästään melko samankaltaiseen lopputulokseen, tosin kasvilajien muodostamat joukot ovat tällöin tarkemmin rajattuja ja sekoittuvat vähemmän keskenään.



Kuva 7. Iteratiivisesti generoitu kasvillisuusjakauma [Deussen *et al.* 1998].

Tehokkuuden näkökulmasta Hammesin [2001] menetelmä on näistä kahdesta parempi, koska se laskee kasvien paikat vain kerran käyttäen yksinkertaisia parametreja. Se sopiikin hyvin sellaisiin sovelluksiin, joissa kasvillisuutta on paljon tai joissa sitä on tarve generoida reaaliajassa. Deussenin ja muiden [1998] menetelmä taas on melko hidas, koska se on sekä iteratiivinen että verrattain monimutkainen. Sillä saadaan kuitenkin aikaan luonnollisempi jakauma, joten se sopii hyvin sellaisiin sovelluksiin, joissa visuaalinen ilme on tärkeää.

Pienenä lisähuomautuksena voidaan sanoa, että fraktaalien avulla voitaisiin generoida myös itse kasveja. Tällöin aloitettaisiin juuresta ja lisättäisiin siihen iteratiivisesti jatkuvasti pieneneviä oksia, päättyen lopulta lehtiin. [Smelik *et al.* 2015.] Kasvien generointi menee kuitenkin tämän tutkielman laajuuden ulkopuolelle, joten sitä ei käsitellä tässä sen enempää.

5. Yhteenveto

Kysyin työn alussa, kuinka fraktaaleja voidaan käyttää hyödyksi maaston algoritmisessa generoimisessa. Vastauksia löytyi useita. Yksi keino on ottaa jokin kaksiulotteinen kohina ja tuoda siihen fraktaalimaisia piirteitä fraktionaalisella Brownin liikkeellä. Näin syntyneestä fraktaalikohinasta voidaan helposti muodostaa korkeuskartta kääntämällä kohinan pisteet korkeusarvoiksi.

Toinen menetelmä on manipuloida korkeuskartan arvoja suoraan timantti-neliö-algoritmin avulla. Tässä aloitetaan tasaisesta korkeuskartasta ja jaetaan sitä iteratiivisesti yhä pieneneviin timantteihin ja neliöihin, asettaen joka kierroksella korkeusarvot näiden keskipisteille ja kulmille. Näin voidaan imitoida fraktaalien itseään toistavia ominaisuuksia ja tuottaa luonnollista maastoa.

Kolmas keino hyödyntää fraktaaleita on simuloida eroosion vaikutusten muodostumista johonkin olemassa olevaan korkeuskarttaan. Tässä menetelmässä korkeuskartan jokaiseen huippukohtaan asetetaan vettä ja mallinnetaan sen valumista alas rinteitä pitkin. Veden määrän ja rinteiden kaltevuuden perusteella korkeuskarttaan joko kaiverretaan uurteita tai lisätään sedimenttiä. Näin maastoon saadaan tuotua realistisia fraktaalikuvioita.

Neljäntenä esittelin, kuinka todellisen maailman fraktaaleita voidaan hyödyntää osana maaston algoritmista generointia. Tässä käyttäjä määrittelee maastolle ensin suuren mittakaavan muodot, kuten vuoristojen sijainnit. Tämän jälkeen maasto muodostetaan reaali maailmasta poimituista maastonpalaista, joissa on jo valmiiksi mukana kaikki yksityiskohdat ja eroosion merkit.

Kaikki nämä neljä menetelmää käsittelevät kuitenkin vain korkeuskartan luomista ja manipulointia. Käytännön sovelluksissa maasto on kuitenkin paljon muutakin kuin pelkkä pintataso, joten työn loppupuolella loin katsauksen myös muiden maaston ominaisuuksien generoimiseen fraktaalien avulla.

Ensimmäinen esittelemäni ominaisuus oli tekstuurit. Tekstuureja voidaan generoida suoraan fraktaalialgoritmeilla muuntamalla ne kaksiulotteisiksi kuvioiksi ja lisäämällä niihin sopivia värejä. Toinen, hieman edistyneempi menetelmä on tuottaa fraktaalien avulla pieniä yksittäisiä palasia (kuten kiviä), joista voidaan lopuksi koota kokonainen tekstuuri (kuten kivipolku). Näin voidaan luoda realistisia tekstuureita minimaalisella tallennustilan käytöllä.

Toisena käsittelin realistisen kasvillisuusjakauman luomista kahdella eri lähestymistavalla. Tehokas mutta karkea menetelmä on lisätä kasveja maastoon jonkin fraktaalisen kohinan mukaisesti, ottaen kuitenkin huomioon myös maastopisteiden korkeudet ja kaltevuudet. Luonnollinen mutta hidas menetelmä taas on simuloida kasvien leviämistä iteratiivisesti askel askeleelta, jolloin niiden jakauma asettuu itsestään fraktaalimaiseen muotoon.

Viiteluettelo

- Giliam de Carpentier. 2011. Scape: 2. Procedural basics. <http://www.decarpentier.nl/scape-procedural-basics>. Checked 25.2.2018
- Giliam de Carpentier. 2012. Scape: 3. Procedural extensions. <http://www.decarpentier.nl/scape-procedural-extensions>. Checked 25.2.2018.
- Min-Yu Chiang, Shih-Chun Tu, Jun-Yan Huang, Wen-Kai Tai, Cheng-Duo Liu, and Chin-Chen Chang. 2005. Terrain synthesis: An interactive approach. In: *Proc. of the International Workshop on Advanced Image Technology*, 1-6.
- Oliver Deussen, Pat Hanrahan, Bernd Lintermann, Radomír Mech, Matt Pharr, and Przemyslaw Prusinkiewicz. 1998. Realistic modeling and rendering of plant ecosystems. In: *Proc. of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, 275–286.
- Johan Hammes. 2001. Modeling of ecosystems as a data source for real-time terrain rendering. *Lecture Notes in Computer Science* 2181, 98–111.
- Tuomo Hyttinen. 2017. *Terrain synthesis using noise*. Master's Thesis. Faculty of Natural Sciences, University of Tampere.
- Musgrave F. Kenton. 1993. *Methods for realistic landscape imaging*. Ph. D. Dissertation. Faculty of Computer Sciences, University of Yale.
- Musgrave F. Kenton, Craig E. Kolb, and Robert S. Mace. 1989. The synthesis and rendering of eroded fractal terrains. *ACM Siggraph Computer Graphics*, 23, 3, 41-50.
- Joeri van der Lei. 2010. Challenges in procedural terrain generation. In: *Twente, The Netherlands: Study Tour Pixel*, 1-6.

- Kazunori Miyata, Takayuki Itoh, and Kenji Shimada. 2001. A method for generating pavement textures using the square packing technique. *The Visual Computer* 17, 8, 475-490.
- Jacob Olsen. 2004. *Realtime procedural terrain generation*. Technical Report. Faculty of Mathematics and Computer Sciences, University of Southern Denmark.
- Nico Pietroni, Paolo Cignoni, Miguel Otaduy, and Roberto Scopigno. 2010. Solid-texture synthesis: a survey. *IEEE Computer Graphics and Applications* 30, 4, 74-89.
- Planetside Software LLC. 2016. Terragen image gallery. <http://planetside.co.uk/terrigen-image-gallery>. Checked 13.2.2018.
- Noor Shaker, Julian Togelius, and Mark J. Nelson. 2016. Fractals, noise and agents with applications to landscapes. In: Noor Shaker, Julian Togelius, and Mark J. Nelson (eds.), *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*, Springer, 57-72.
- Ruben M. Smelik, Klaas Jan de Kraker, and Saskia A. Groenewegen. 2015. A survey of procedural methods for terrain modelling. In: *Proc. of the CASA Workshop on 3D Advanced Media in Gaming and Simulation*, 25-34.

Katsaus sumeaan klusterointiin

Väinö Teisko

Tiivistelmä.

Klusterointi on tiedonlouhinnassa hyödynnettävä ohjaamattoman oppimisen menetelmä, jota käytetään luokittelemattoman datan ryhmittelyyn. Tässä tutkielmassa perehdytään pääasiassa sumeaan klusterointiin, joka on yksi klusteroinnin muoto. Tutkielmassa käydään läpi klusteroinnin toimintaperiaatteet ja verrataan sumeaa klusterointia kovaan klusterointiin, joka on toinen klusteroinnin muoto. Menetelmäosiossa käsitellään viittä eri sumean klusteroinnin menetelmää ja niiden välisiä eroja.

Avainsanat ja -sanonnat: klusterointi, sumea klusterointi, klusterointimenetelmä, fuzzy c-means, possibilistic c-means.

1. Johdanto

Tietokoneiden prosessointitehon, tallennustilan ja verkostoitumisen lisääntyessä syntyy enemmän ja enemmän uutta dataa. Jotta valtavia määriä dataa voidaan hyödyntää mahdollisimman tehokkaasti, tarvitaan tehokkaita datan prosessointimenetelmiä. Datan prosessointiin voidaan käyttää tiedonlouhinnan menetelmiä. Tiedonlouhinta tarkoittaa uuden tiedon etsimistä suurista tietomassoista, tunnistamalla hahmoja datassa eri menetelmien avulla [Blazewicz *et al.* 2012].

Klusterointi on eräs tiedonlouhinnassa hyödynnettävä menetelmä, jota käytetään luokittelemattoman datan ryhmittelyyn. Tässä tutkielmassa keskitytään pääasiassa sumeaan klusterointiin, joka on yksi klusteroinnin muoto. Klusterointia käsitellään tarkemmin luvussa 2. Samassa luvussa myös selvennetään, millaista klusterointia sumea klusterointi on.

Monet erilaiset klusterointimenetelmät sisältävät sumean klusteroinnin periaatteita. Sumean klusteroinnin menetelmiä käydään läpi luvussa 3. Tarkastelun kohteina olevia menetelmiä ovat fuzzy c-means, Gustafson-Kessel, possibilistic c-means, fuzzy-possibilistic c-means ja possibilistic-fuzzy c-means.

Luku 4 on yhteenveto, jossa kootaan yhteen tutkielman tärkeimmät havainnot.

2. Klusterointi

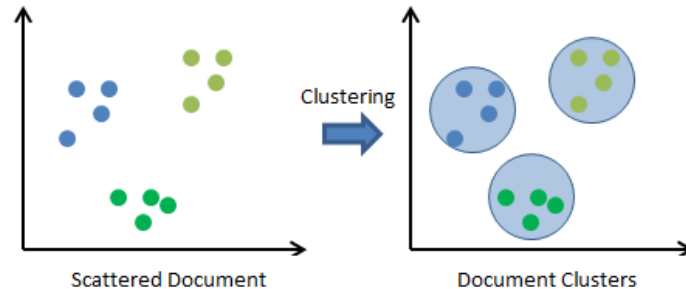
Klusterointi on tiedonlouhinnan menetelmä, jonka tavoitteena on ryhmittää käsiteltävän aineiston instansseja klustereihin siten, että kaikki samassa kluste-

rissa olevat instanssit ovat mahdollisimman samanlaisia keskenään ja samalla mahdollisimman erilaisia kuin muissa klustereissa olevat instanssit [Wolfram 2004; Blazewicz *et al.* 2012]. Samaan klusteriin kuuluvia instansseja voidaan kohdella yhtenä ryhmänä [Blazewicz *et al.* 2012].

Klusteroinnin suorittaa klusterointialgoritmi. Klusterointialgoritmi saa syöteenään klusteroitavan aineiston ja mahdollisia muita parametreja. Algoritmin lopputuloksena on matriisi klusteroidusta aineistosta. Erilaisia sumean klusteroinnin algoritmeja käsitellään tarkemmin luvussa 3.

Klusterit voidaan käsittää luokkina, joihin algoritmi luokittelee aineiston instanssit. Klusterointi on ohjaamattoman oppimisen tekniikka, joten aineiston luokkia ei ole määrätty ennalta [Shalev-Shwartz and Ben-David 2014]. Sen sijaan klusterointialgoritmi muodostaa luokat, eli klusterit, jotka sopivat aineistoon [Shalev-Shwartz and Ben-David 2014].

Klusteroinnin tavoitteena on siis löytää uusia rakenteita järjestämättömästä aineistosta. Klusterointia voidaan myös käyttää aineiston yksinkertaistamiseen (data reduction). Tässä tapauksessa klusteroinnin tavoitteena on vain luoda yksinkertaisempi esitys aineistosta. Klustereihin jaettu aineisto on helpompi esittää ja sitä on helpompi käsitellä, verrattuna valtavaan määrään yksittäisiä instansseja. [Döring *et al.* 2006]



Kuva 1. Klusterointiesimerkki. [Kunwar 2013]

Kuvassa 1 on esitetty yksinkertainen esimerkki kaksiulotteisen aineiston klusteroinnista. Esimerkin klusteroidussa aineistossa kaikki instanssit kuuluvat yhteen kolmesta klusterista, eikä yksikään instanssi kuulu useaan klusteriin samanaikaisesti. Klusterointi ei kuitenkaan aina ole näin yksinkertaista. Kaikki instanssit eivät aina välttämättä kuulu mihinkään klusteriin ja tietyntylaisessa klusteroinnissa instanssit voivat myös kuulua useaan klusteriin samanaikaisesti.

Koska samaan klusteriin kuuluvien instanssien täytyy olla mahdollisimman samanlaisia keskenään, samanlaisuuden tunnistamiseksi instansseja pitää jollain tavalla verrata toisiinsa [Wolfram 2004]. Klusteroinnissa yleisesti samanlaisuuden mittaamiseen käytettyjä etäisyysfunktioita ovat esimerkiksi Euklidinen etäisyys, Manhattan-etäisyys ja Chebyshev-etäisyys. Valitut samanlaisuuden

mittaustavat määrittävät, kuinka klusterit muodostuvat. Mittaustapojen valintaan vaikuttavat klusteroitava aineisto ja valmiin klusteroinnin lopullinen käyttötarkoitus [Wolfram 2004].

Klusterointia voidaan hyödyntää monilla eri aloilla. Konkreettisina esimerkkeinä klusteroinnista otetaan esille seuraavat klusteroinnin mahdolliset käyttökohteet. Markkinoinnissa klusterointia voidaan käyttää erilaisten kuluttajaryhmien tunnistamiseen suuria asiakastietokantoja klusteroimalla. Kuluttajaryhmien tunnistaminen voi auttaa esimerkiksi mainonnan kohdentamisessa [Zaiane 1999; Han and Kamber 2006; Tandel 2015]. Biologiassa klusteroinnin avulla voidaan luoda kasvi- ja eläinluokitteluja, sekä tutkia populaatioiden rakennetta [Han and Kamber 2006]. Kaupungissa olevia taloja voidaan jakaa ryhmiin esimerkiksi talon tyyppin, arvon ja sijainnin perusteella [Zaiane 1999; Han and Kamber 2006]. Klusteroinnilla siis pyritään ryhmittelemään olemassa olevaa luokittelematonta dataa joidenkin datan ominaisuuksien suhteen, jotta dataa voitaisiin analysoida tarkemmin.

2.1. Sumea klusterointi

Klusterointimetodit jaetaan kahteen pääluokkaan, *kovaan klusterointiin* (hard/crisp clustering) ja *sumeaan klusterointiin* (soft/fuzzy clustering). Kova klusterointi on sellaista klusterointia, jossa jokainen klusteroitava instanssi voi kuulua vain yhteen klusteriin. Sumeassa klusteroinnissa puolestaan jokainen instanssi voi kuulua useampaan kuin yhteen klusteriin. [Wolfram 2004]

Jos jokin instanssi sijaitsee kahden eri klusterin rajamailla, kovassa klusteroinnissa kyseinen instanssi pakotettaisiin jompaankumpaan klustereista. Sumean klusteroinnin etuna on, että instanssia ei pakotettaisi vain toiseen klustereista, vaan se voisi kuulua molempiin klustereihin samanaikaisesti. [Bora and Gupta 2014]

Sumea klusterointi sai alkunsa, kun Zadeh [1965] esitteli sumeiden joukkojen käsitteen. Ruspini [1969] sovelsi ensimmäisenä sumeiden joukkojen ideaa klusterointiin. Sumeaa klusterointia on sittemmin laajennettu moniin erilaisiin tyyppeihin. [Chaomurilige *et al.* 2017]

2.2. Kuuluvuusarvo

Kuten sumeissa joukoissa, sumeassa klusteroinnissa kaikilla klusteriin kuuluvilla instansseilla on kuuluvuusarvo. Kuuluvuusarvo kertoo missä määrin instanssi kuuluu kyseiseen klusteriin. Kaikki samaan klusteriin kuuluvat instanssit eivät siis välttämättä kuulu kyseiseen klusteriin yhtä vahvasti. [Wolfram 2004.]

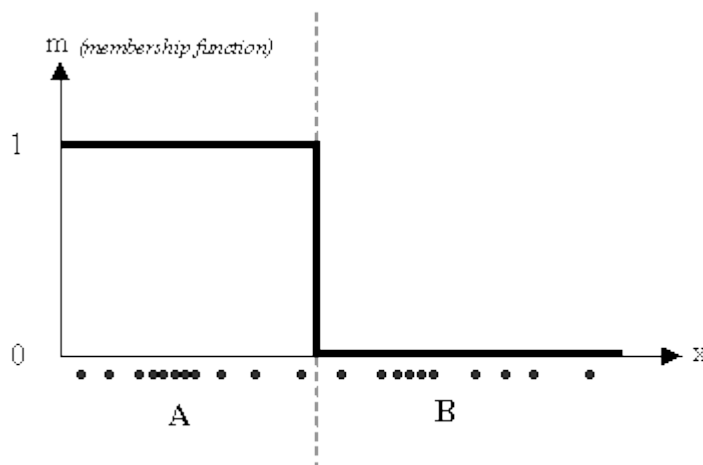
Tässä kohdassa käydään läpi instanssin ja klusterin välisen kuuluvuusarvon eroja kovassa ja sumeassa klusteroinnissa. Vertailuesimerkkeinä käytetään

kovaa k-means-klusterointimenetelmää (KM) ja sumeaa fuzzy c-means -klusterointimenetelmää (FCM). FCM-menetelmää käsitellään lisää tämän tutkielman kohdassa 3.1.

FCM-menetelmä on hyvin lähellä KM-menetelmää. Käytännössä FCM on sumea versio kovasta KM-menetelmästä, ja tämän takia fuzzy c-means -menetelmä tunnetaan myös nimellä fuzzy k-means [Gasch and Eisen 2002]. Menetelmien erona on niissä käytettävä kuuluvuusarvo.

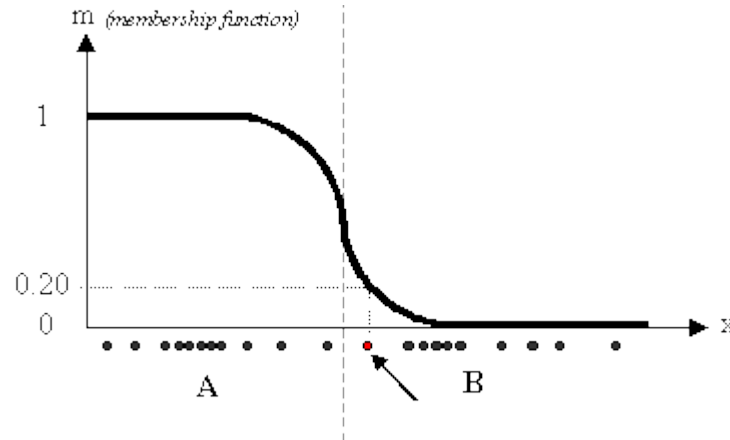
Kovassa klusteroinnissa instanssin ja klusterin välinen kuuluvuusarvo voi olla vain tasan 0 tai 1, ja jokainen instanssi voi kuulua vain yhteen klusteriin. Sumeassa klusteroinnissa käytettävä instanssien kuuluvuusarvo voi olla mikä vain arvo välillä $[0,1]$ ja instanssit voivat kuulua useaan klusteriin samanaikaisesti, joihinkin vahvemmin kuin toisiin.

Alla on kuvattu esimerkki näiden kahden menetelmän kuuluvuusarvojen toiminnasta.



Kuva 2. Aineisto jaettu koviin klustereihin A ja B. [Matteucci 2018]

Kuvassa 2 x-akselilla esitetyt instanssit on jaettu kahteen klusteriin, "A" ja "B". Instanssien kuuluvuusarvo on kuvattu pystyakselilla siten, että arvo 1 kuvaa täydellistä kuulumista klusteriin A ja arvo 0 täydellistä kuulumista klusteriin B. Kaikki instanssit kuuluvat kokonaan klusteriin A tai kokonaan klusteriin B. Tältä valmiit klusterit näyttävät kovassa KM-menetelmässä. [Matteucci 2018]



Kuva 3. Aineisto jaettu sumeisiin klustereihin A ja B. [Matteucci 2018]

Sumeassa klusteroinnissa jokainen instanssi voi kuulua useaan klusteriin. Tätä on havainnollistettu kuvassa 3. Kuvaajalla punaiseksi värjätty instanssi kuuluu enemmän klusteriin B kuin klusteriin A. Pysty akselin m arvo 0,20 kuvaa värjätyin instanssin kuuluvuusarvoa klusteriin A. [Matteucci 2018]

$$U_{KM} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ \dots & \dots \\ 0 & 1 \end{bmatrix}, U_{FCM} = \begin{bmatrix} 0,8 & 0,2 \\ 0,3 & 0,7 \\ 0,6 & 0,4 \\ \dots & \dots \\ 0,1 & 0,9 \end{bmatrix}$$

Kuva 4. Matriisit instanssien ja klustereiden välisistä kuuluvuusarvoista KM-menetelmässä ja FCM-menetelmässä. [Matteucci 2018]

Kuvassa 4 on esitetty kaksi matriisia kuuluvuusarvoista. Vasemmalla puolella on matriisi instanssien kuuluvuusarvoista KM-menetelmässä ja oikealla puolella vastaava matriisi FCM-menetelmässä. Matriisin rivien määrä riippuu instanssien määrästä ja sarakkeiden määrä riippuu tarkasteltavien klustereiden määrästä [Matteucci 2018]. Kuvan 4 matriiseissa arvo 1 kuvaa instanssin täydellistä kuulumista kyseisen sarakkeen klusteriin ja arvo 0 täydellistä kuulumattomuutta. Arvot välillä]0,1[kuvaavat osittaista kuulumista.

Sumeista klustereista voidaan muuttaa kovia klustereita asettamalla jokainen instanssi vain siihen klusteriin, johon se kuuluu vahvimmin. Tätä kutsutaan kovaksi ositukseksi. [Makarevich 2016]

3. Menetelmiä

On olemassa monia sumean klusteroinnin menetelmiä. Monet menetelmistä ovat toistensa johdoksia. Uusia menetelmiä on kehitetty esimerkiksi aikaisempien menetelmien vajavuuksien korjaamiseksi (esim. [Gustafson and Kessel 1978]) tai paremman suorituskyvyn saamiseksi tietyntylaisia aineistoja klusteroidessa (esim. [Pal *et al.* 2005]). Tässä luvussa käsiteltävät sumean klusteroinnin

menetelmät ovat fuzzy c-means, Gustafson-Kessel, possibilistic c-means, fuzzy-possibilistic c-means ja possibilistic-fuzzy c-means.

3.1. Fuzzy c-means

Fuzzy c-means -menetelmä (FCM) on käytetyin sumean klusteroinnin menetelmä [Chaomurilige *et al.* 2017]. FCM on Dunnin [1973] luoma ja Bezdekin [1981] parantelema. FCM-menetelmää on käytetty moniin tarkoituksiin, kuten tiedonlouhintaan [Hirota and Pedrycz 1999], mallintamiseen [Chang *et al.* 2000], hahmontunnistukseen [Chuang *et al.* 1999], luokitteluun [Iyer *et al.* 2000; Mei *et al.* 2017], kuvan osiin jakamiseen [Phillips *et al.* 1995; Rezaee *et al.* 2000] ja prosessin valvontaan [Teppola *et al.* 1999].

3.1.1. FCM-algoritmi

FCM-algoritmi on iteratiivinen. Aluksi klustereiden keskipisteet asetetaan satumanvaraisesti ja algoritmi laskee kuuluvuusarvot instansseille klustereiden sijaintien perusteella. Joka iteraatiolla algoritmi laskee klusteriprototyyppien sijainnit ja instanssien kuuluvuusarvot toistuvasti uudestaan, päästen aina tarkempaan tulokseen. Iteraatioiden lisääntyessä niiden lopputulokset lähenevät toisiaan. Iterointi lopetetaan, kun ero peräkkäisten iteraatioiden lopputulosten välillä on tarpeeksi pieni tai kun maksimimäärä iteraatioita on suoritettu. [Dunn 1973]

The FCM Algorithm

```

begin
  Fix  $c, 2 < c < n$ ;
  Fix  $\epsilon$ , (e.g.,  $\epsilon = 0.001$ );
  Fix maxIterations, (e.g., maxIterations=100);
  Choose any inner product norm metric (e.g., Euclidean distance);
  Fix  $m, 1 < m < \infty$ , (e.g.,  $m = 2$ );
  Randomly initialize  $V_0 = v_1, v_2, \dots, v_c$  cluster centers;
  for  $t = 1$  to maxIterations do
    Update the membership matrix  $U$  using Eq. 3;
    Calculate the new cluster centers  $V^t$  using Eq. 4;
    Calculate the new objective function  $J_m^t$  using Eq. 2;
    if ( $\text{abs}(J_m^t - J_m^{t-1}) < \epsilon$ ) then
      break;
    else
       $J_m^{t-1} = J_m^t$ ;
    end if
  end for
end

```

Kuva 5. FCM-algoritmin pseudokoodi. [Alia *et al.* 2011]

Kuvan 5 pseudokoodissa viitatus kaavat ovat tässä tutkielmassa esitetty seuraavasti: Eq. 2 on kaava 1, Eq. 3 on kaava 2 ja Eq. 4 on kaava 3.

FCM pyrkii osittamaan n kokoisen joukon instansseja $X = \{x_1, x_2, \dots, x_n\}$ sumeisiin klustereihin. Algoritmi luo c kappaletta klusteriprototyyppejä ja listan klusteriprototyypeistä V

$$V = v_j; j = 1, 2, \dots, c,$$

jossa v_j on klusteriprototyyppi. Algoritmi palauttaa ositusmatriisin U

$$U = u_{i,j}; i = 1, \dots, n; j = 1, \dots, c,$$

jossa $u_{ij} \in [0,1]$ kuvaa instanssin x_i kuulumisen astetta klusteriin j . [Wolfram 2004.] Arvo u_{ij} on siis instanssin x_i kuuluvuusarvo klusteriin j . FCM-menetelmän kuuluvuusarvot täyttävät ehdon

$$\sum_{j=1}^c u_{ij} = 1 \forall i,$$

eli yhden instanssin kaikkien eri klustereihin viittaavien kuuluvuusarvojen summan on aina oltava tasan yksi, kaikilla instansseilla [Bezdek 1981].

Poikkeamilla (noise, outliers) tarkoitetaan tässä tutkielmassa sellaisia instansseja, jotka ovat kaukana klustereiden keskipisteistä, eivätkä siten sovi hyvin mihinkään klusteriin. Ihannetapauksessa poikkeamilla siis olisivat matalat kuuluvuusarvot klustereihin. Jos näin ei ole, klusteroinnin tulokset voivat olla harhaanjohtavia.

FCM-algoritmi pyrkii minimoimaan kaavassa 1 esitetyn kustannusfunktion [Alia et al. 2011; Correa et al. 2011].

$$J_m(U, V) = \sum_{j=1}^c \sum_{i=1}^n u_{ij}^m \|x_i - v_j\|^2, \quad (1)$$

jossa $m = [1, \infty[$ on painotus-eksponentti, joka määrittää lopullisten klustereiden sumeuden astetta [Correa et al. 2011].

Toimiakseen FCM-algoritmi tarvitsee parametreina muuttujat c , m , ε ja myös klusteroitavan aineiston [Bezdek 1981]. Muuttuja c on klustereiden lukumäärä. FCM ei siis osaa itsenäisesti päätellä sopivaa klustereiden määrää, vaan määrä täytyy antaa parametrina.

Mitä lähempänä m on arvoa 1, sitä lähempänä FCM on kovaa c-means-menetelmää [Yu et al. 2004]. Muuttujan m lähestyessä ääretöntä, kaikkien instanssien kuuluvuusarvoista tulee identtisiä [Schwämmle and Jensen 2010]. Suuri muuttujan m arvo voi johtaa tiedon menetykseen ja matala m arvo voi poikkeamien kautta johtaa virheellisiin havaintoihin [Schwämmle and Jensen 2010]. Valittu muuttujan m arvo voi siis vaikuttaa klusteroinnin lopputuloksiin

huomattavasti. Pal ja Bezdek [1995] ehdottivat arvoväliä [1.5, 2.5] mahdollisesti parhaaksi muuttujan m arvoväliksi ja monet tutkijat käyttävätkin arvoa $m = 2$ [Pal and Bezdek 1995; Jing *et al.* 2014].

Iteroinnin lopettamiseen tarvittavan eron iteraatioiden välillä määrää muuttuja ε . Mitä pienempi ε on, sitä pienempi eron täytyy olla. Iterointi siis loppuu, kun ero peräkkäisten iteraatioiden välillä on pienempi kuin muuttujan ε arvo. [Bezdek 1981]

Ei ole olemassa tehokasta laskennallista tapaa löytää aina optimaaliset parametrien arvot FCM-algoritmille [Schwämmle and Jensen 2010], joten parametrien c , m ja ε sopivien arvojen löytäminen voi tuottaa vaikeuksia. Sopivien arvojen löytämiseksi on kuitenkin kehitetty erillisiä menetelmiä, jotka voivat helpottaa arvojen valintaa [Schwämmle and Jensen 2010; Zhou *et al.* 2014].

Joka iteraatiolla jokaisen instanssin kuuluvuusarvot jokaiseen klusteriin lasketaan yksitellen käyttäen kaavaa 2 [Alia *et al.* 2011]. Kuuluvuusarvot kootaan ositusmatriisiksi U .

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|} \right)^{\frac{2}{m-1}}} \quad (2)$$

Kuuluvuusarvojen laskemisen jälkeen klustereiden keskipisteiden sijainnit lasketaan uudelleen käyttäen kaavaa 3, perustuen saatuihin kuuluvuusarvoihin. [Alia *et al.* 2011]

$$v_j = \frac{\sum_{i=1}^n u_{ij}^m x_i}{\sum_{i=1}^n u_{ij}^m} \quad (3)$$

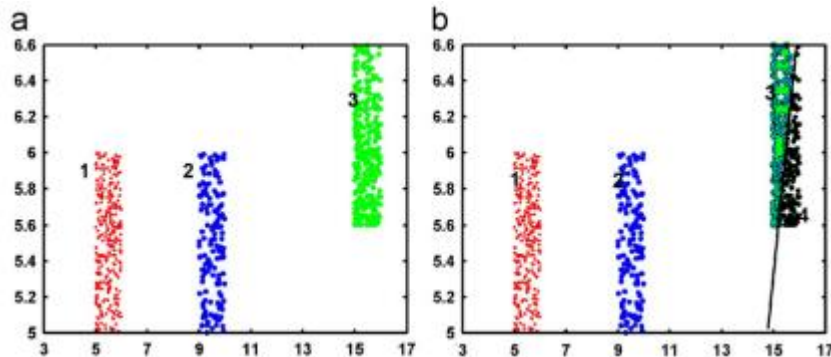
Oletusarvoisesti FCM-menetelmä käyttää Euklidista etäisyyttä instanssien ja klustereiden välisten etäisyyksien mittaamiseen, jonka johdosta klusterit pyrkivät muodostumaan pallon muotoisiksi [Bezdek 1981]. Menetelmällä on siis vaikeuksia tunnistaa muun kuin pallon muotoisia klustereita.

3.1.2. Klusterin kelvollisuusongelma

Toisena FCM-menetelmän heikkoutena voidaan pitää *klusterin kelvollisuusongelmaa* (cluster validity problem). Tällä viitataan sopivimman parametrin c arvon tunnistamisen vaikeuteen eli sopivimman klustereiden määrän tunnistamiseen aineistosta. Klusterin kelvollisuusongelman määrittäminen matemaattisesti on erittäin vaikeaa, sillä ”hyvän” klusterin, tai ylipäätänsä edes klusterin, määrittäminen on myös haastavaa. [Bezdek 1981]

Useimmiten käyttäjällä ei ole ennestään tietoa aineistosta luotavien klustereiden optimaalisimmasta määrästä. Jos etsittävien klustereiden määrä on liian suuri, jotkin mahdollisista hyvistä klustereista voivat hajota pienemmiksi. Vas-

taavasti jos klustereita etsitään liian vähän, monet mahdollisista hyvistä klustereista voivat yhdistyä suuremmiksi klustereiksi. Epäoptimaalinen klustereiden määrä voi siis huonontaa klusteroinnin tulosten laatua, joten oikean klustereiden määrän löytäminen on tärkeä ongelma. [Wang and Zhang 2007]



Kuva 6. (a) Aineisto, joka koostuu kolmesta klusterista, (b) FCM-algoritmin tulokset, kun etsitään neljä klusteria. [Wang and Zhang 2007]

Kuvan 6(a) aineistosta voi helposti nähdä, että aineisto sisältää kolme klusteria. Kun aineistosta haetaan neljä klusteria, vihreällä värjätty klusteri 3 hajoaa kuvassa 6(b) kahdeksi erilliseksi klusteriksi. Kaksiulotteisesta datasta saattaa melko helpostikin visuaalisesti nähdä klustereiden oikean määrän, mutta määrän näkeminen ei aina ole yhtä helppoa kuin kuvan 6 esimerkissä, varsinkaan jos klusteroitava aineisto on moniulotteinen. [Wang and Zhang 2007]

Sumeassa klusteroinnissa klustereiden optimaalisen määrän löytämiseksi voidaan käyttää *klusterikelvollisuusindeksejä* (cluster validity index). Varhaiset kelvollisuusindeksit, kuten *jakaantumiskerroin* (partition coefficient) ja *luokitteluentropia* (classification entropy) hyödyntävät vain kuuluvuusarvoja. Nykyisin on kuitenkin yleisesti hyväksyttyä, että paremmat kelvollisuusindeksit ottavat huomioon sekä kuuluvuusarvot että itse aineiston. [Wang and Zhang 2007]

Wang ja Zhang [2007] vertasivat 23 eri kelvollisuusindeksiä toisiinsa yhdessä FCM-algoritmin kanssa käyttäen 16 eri aineistoa. Indekseistä viisi olivat pelkästään kuuluvuusarvoja hyödyntäviä ja loput 18 hyödynsivät sekä kuuluvuusarvoja että itse aineistoa. Kaikki testatuista indekseistä tunnistivat optimaalisen klusterimäärän oikein joillain aineistoilla, mutta yksikään indeksi ei tunnistanut oikeaa klustereiden määrää jokaisella aineistolla. Klusterikelvollisuusindekseillä voi siis joissain tapauksissa saada selville optimaalimman klusterimäärän, mutta ne eivät kuitenkaan ole täydellinen ratkaisu klusterin kelvollisuusongelmaan.

3.2. Gustafson-Kessel

Gustafson-Kessel-menetelmä (GK) on yksi käytetyimmistä sumean klusteroinnin menetelmistä [Chaomurilige *et al.* 2017]. GK on ensimmäinen tärkeä jatke

FCM-menetelmälle [Chaomurilige *et al.* 2015]. Menetelmän kehittivät Gustafson ja Kessel [1978].

FCM-menetelmässä käytettävä euklidinen etäisyys rajoittaa sen kykyä tunnistaa muun kuin pallon muotoisia keskittymiä. GK-menetelmässä instanssien etäisyyksien mittaamiseen hyödynnetään euklidisen etäisyyden sijasta Mahalanobiksen etäisyyttä, jonka ansiosta GK kykenee tunnistamaan myös ellipsoidisia klustereita. [Lesot and Kruse 2008; Graves and Pedrycz 2010]

GK-algoritmi vaatii parametreina muuttujat c , m , ε , ρ_j ja klusteroitavan aineiston [Chaomurilige *et al.* 2015]. Parametrit ovat siis samat kuin FCM-menetelmässä ja niiden lisäksi positiivinen skaalausparametri ρ_j . GK-algoritmi perustuu kaavassa 4 esitetyn kustannusfunktion iteratiiviseen optimointiin [Correa *et al.* 2011].

$$J_m(U, V, \{A_j\}) = \sum_{j=1}^c \sum_{i=1}^n u_{ij}^m d_{ijA_j}^2 \quad (4)$$

GK-algoritmin kustannusfunktio on lähellä FCM-algoritmin kustannusfunktiota (vrt. kaava 4 ja kaava 1). Näiden funktioiden välisenä erona ovat GK-algoritmissa käytettävät klustereiden lokaalit etäisyysnormit, joiden ansiosta klusterit voivat muodostua erimuotoisiksi toisistaan. Kaavassa 4 GK-algoritmin etäisyysnormi d_{ijA_j} määritetään lokaalin normin indusoivan matriisin A_j avulla kullekin klusterille erikseen. [Correa *et al.* 2011.]

$$d_{ijA_j}^2 = \|x_i - v_j\|_{A_j}^2 = (x_i - v_j)^T A_j (x_i - v_j) \quad (5)$$

GK-algoritmissa Mahalanobiksen etäisyyttä ei voida käyttää suoraan FCM-algoritmin euklidisen etäisyyden paikalla. Jos näin tehtäisiin, instanssien kuuluvuudet jaettaisiin aina tasapuolisesti kaikkien klustereiden kesken. Tämän ongelman välttämiseksi kaavassa 5 ei käytetä kovarianssimatriisia C_j suoraan, vaan matriisin C_j sijasta käytetään sen pohjalta luotua positiivisesti definiittiä matriisia A_j . [Krishnapuram and Kim 1999]

Kaavassa 5 on avattu kaavassa 4 esiintyvä etäisyysnormi d_{ijA_j} . Kuten FCM-algoritmile, myös GK-algoritmile pätevät kuuluvuusarvoon liittyvät rajoitukset $u_{ij} \in [0,1]$ ja että yhden instanssin kuuluvuusarvojen summa on aina tasan 1. Matriisi A_j lasketaan sumean kovarianssimatriisin C_j pohjalta kaavan 6 mukaisesti.

$$A_j = (\rho_j |C_j|)^{1/d} C_j^{-1}, \quad (6)$$

jossa ρ_j on positiivinen skaalausparametri. Yleisesti käytetään rajoitusta $\rho_j = 1$. Matriisin C_j determinanttia merkitään $|C_j|$ ja käänteismatriisia C_j^{-1} . Matriisi C_j lasketaan kaavan 7 mukaan. [Graves and Pedrycz 2010]

$$C_j = \frac{\sum_{i=1}^n u_{ij}^m (x_i - v_j)(x_i - v_j)^T}{\sum_{i=1}^n u_{ij}^m} \quad (7)$$

Kuuluvuusarvot lasketaan kaavalla 8 [Graves and Pedrycz 2010]. Kaava on lähellä FCM-menetelmän kuuluvuusarvojen laskemisen kaavaa (vrt. kaava 2 ja kaava 8). Erot kuuluvuusarvojen laskennassa johtuvat lähinnä FCM- ja GK-menetelmien käyttämisestä eri etäisyysnormeista.

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{(x_i - v_j)^T A_j (x_i - v_j)}{(x_i - v_k)^T A_k (x_i - v_k)} \right)^{1/(m-1)}} \quad (8)$$

Klustereiden keskipisteet v_j lasketaan käyttäen kaavaa 3 [Graves and Pedrycz 2010], eli samalla kaavalla kuin FCM-menetelmässä.

3.3. Possibilistic c-means

Possibilistic c-means -menetelmä (PCM) on Krishnapuramin ja Kellerin [1993] esittelemä sumean klusteroinnin menetelmä. PCM on muokattu versio FCM-menetelmästä.

3.3.1. PCM-algoritmi

Vaikka PCM onkin FCM-menetelmän pohjalta johdettu, menetelmät ovat hyvin erilaisia. FCM on pääasiallisesti ositusalgoritmi, kun taas PCM on pääasiallisesti keskittymiä etsivä (mode-seeking) algoritmi. PCM-menetelmän vahvuudet ovat tiheiden keskittymien tunnistaminen aineistosta ja poikkeamien sivuuttaminen keskittymiä etsittäessä. [Barni *et al.* 1996]

FCM-menetelmän kuuluvuusarvon sijasta PCM-menetelmässä käytetään tyypillisyyssarvoa. FCM-menetelmässä instanssin kuuluvuusarvojen summan täytyy olla tasan yksi, mutta PCM-menetelmän tyypillisyyssarvoilla ei ole vastaavaa summarajoitusta [Krishnapuram and Keller 1996]. PCM-menetelmän tyypillisyyssarvo on aina väliltä $[0,1]$, mutta jokaisella instanssilla täytyy olla vähintään yksi positiivinen tyypillisyyssarvo [Krishnapuram and Keller 1993].

Tyypillisyyssarvoon t_{ij} vaikuttaa vain yhden instanssin x_i ja klusterin keskipisteen v_j välinen etäisyys, kun taas kuuluvuusarvoon u_{ij} vaikuttavat kaikkien klustereiden keskipisteiden suhteelliset etäisyydet instanssista [Pal *et al.* 2005]. Tyypillisyyssarvo voidaan tulkita kuvaamaan instanssin klusteriin kuulumisen mahdollisuutta, eli instanssin yhteensopivuutta klusteriprototyypin kanssa [Krishnapuram and Keller 1993].

Minimoitava kustannusfunktio PCM-algoritmile on esitetty kaavassa 9

$$J_m(V, T) = \sum_{j=1}^c \sum_{i=1}^n t_{ij}^m d_{ij}^2 + \sum_{j=1}^c \eta_j \sum_{i=1}^n (1 - t_{ij})^m, \quad (9)$$

jossa tyypillisuusarvot merkitään t_{ij} :llä, tyypillisuusarvojen matriisi on T ja η_j on skaalausparametri. Muuttuja η_j voidaan määrittellä kaavan 10 mukaisesti. [Krishnapuram and Keller 1993]

$$\eta_j = K \frac{\sum_{i=1}^n t_{ij}^m d_{ij}^2}{\sum_{i=1}^n t_{ij}^m}, K > 0 \quad (10)$$

Kaavan 10 mukaan muuttuja η_j on verrannollinen klusterin j sisäisten summeiden etäisyyksien keskiarvon kanssa. Tyypillisesti muuttujan K arvoksi valitaan 1. Muuttujan η_j arvo määrää etäisyyden, jossa instanssin tyypillisuusarvo klusteriin on 0,5. Koska η_j on klusterikohtainen, voidaan jokaiselle klusterille asettaa eri η_j arvo. Arvo voidaan myös asettaa samaksi jokaiselle klusterille, mutta yleisesti on toivottavaa, että η_j arvo on suhteessa klusterin j kokoon. Muuttujan η_j arvo siis määrittelee klusterin vaikutusalueen koon. Klustereilla, joilla on suuri muuttujan η_j arvo, on enemmän liikkumavaraa, sillä ne "näkevät" enemmän instansseja. Harvaan täytetyillä klustereilla, joilla on suuret sisäiset sumeat etäisyydet, on enemmän vapautta liikkua kuin tiiviillä keskittymien kohdalla sijaitsevilla klustereilla. Tämän johdosta tyhjemmät klusterit voivat iteraatioiden edetessä asettua aineiston tiheämmille alueille. [Krishnapuram and Keller 1993]

Klustereiden keskipisteet päivitetään kaavalla 11

$$v_j = \frac{\sum_{i=1}^n t_{ij}^m x_i}{\sum_{i=1}^n t_{ij}^m} \quad (11)$$

ja tyypillisuusarvot päivitetään käyttäen kaavaa 12 [Krishnapuram and Keller 1993].

$$t_{ij} = \frac{1}{1 + \left(\frac{d_{ij}^2}{\eta_j}\right)^{\frac{1}{m-1}}} \quad (12)$$

PCM-menetelmässä voi käyttää eri etäisyysfunktioita. Eri etäisyysfunktioita käyttämällä muodostuu erilaisia määritelmiä muuttujalle d_{ij}^2 ja siten erilaisia algoritmeja [Krishnapuram and Keller 1993]. Mahdollisia etäisyysfunktioita ovat esimerkiksi FCM-menetelmässä käytettävä euklidinen etäisyys tai Gustafson-Kessel-menetelmässä käytettävä Mahalanobiksen etäisyys.

PCM-algoritmin klusterit ovat täysin itsenäisiä toisistaan tyypillisuusarvojen summarajoituksen puutteen vuoksi. Tämän takia kustannusfunktio voidaan

esittää myös klusterikohtaisesti. Kaavassa 13 on kuvattu klusteria j vastaava kustannusfunktio.

$$J_j(\beta_j, T_j) = \sum_{i=1}^n t_{ij}^m d^2(x_i, \beta_j) + \eta_j \sum_{i=1}^n (1 - t_{ij})^m, \quad (13)$$

jossa β_j kuvaa klusterin j iteroitavaa prototyyppiä ja T_j sisältää kaikki tyypillisyyssarvomatriisiin T klusteriin j liittyvät tyypillisyyssarvot. [Krishnapuram and Keller 1996]

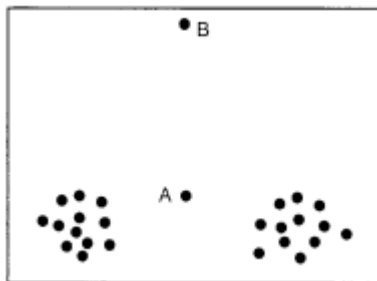
3.3.2. PCM-menetelmän parametrit

FCM-menetelmän tavoin myös PCM-menetelmässä muuttujien c , m ja ε alustus voi vaikuttaa tuloksiin huomattavasti. Lisäksi PCM on herkkä myös muuttujan η_j alustukselle. Näiden muuttujien epäoptimaalinen alustus voi estää joidenkin keskittymien löytymisen aineistosta. [Krishnapuram and Keller 1996]

Arvoalue $m = [1, \infty[$ pätee myös PCM-menetelmässä, mutta m toimii PCM-menetelmässä hieman eri tavalla kuin FCM-menetelmässä. PCM-menetelmässä m arvon nostaminen lisää kaikkien instanssien täyden tyypillisyyden mahdollisuutta kyseiseen klusteriin, kun taas FCM-menetelmässä suurempi m arvo kuvaa instanssien kuuluvuuksien tasaisempaa jakoa kaikkien klustereiden kesken. Tästä erosta johtuen hyvä muuttujan m arvo eroaa menetelmien välillä. FCM-menetelmässä arvoa $m \approx 2$ pidetään hyvänä lähtökohtana, mutta Krishnapuramin ja Kellerin testien mukaan sopivampi arvo PCM-menetelmässä olisi $m \approx 1,5$. [Krishnapuram and Keller 1996]

3.3.3. Kuuluvuusarvo ja tyypillisyyssarvo

Krishnapuram ja Keller [1993] osoittivat, että tyypillisyyteen perustuvan kuuluvuusarvon käyttäminen vähentää poikkeamien vaikutusta klusterointiin ja tuottaa siten parempia tuloksia, kuin FCM-menetelmässä käytettävä kuuluvuusarvo.



Kuva 7. Ongelma FCM-menetelmässä. [Krishnapuram and Keller 1993]

Esimerkkinä poikkeaman mahdollisesta vaikutuksesta FCM-menetelmässä oletetaan, että FCM-menetelmässä haettaisiin kahta klusteria, eli $c = 2$, ja instansseilla A ja B olisi yhtä pitkät etäisyydet molempiin klustereihin, kuvan 7

mukaisesti. Koska FCM-menetelmässä instanssin kuuluvuusarvojen summan täytyy olla tasan yksi, instanssien A ja B molempien kuuluvuusarvojen täytyisi olla 0,5. Instanssin absoluuttisella etäisyydellä klustereista ei siis ole merkitystä kuuluvuusarvon kannalta, kunhan etäisyys on sama kumpaankin klusteriin. [Krishnapuram and Keller 1993]

Jos kuuluvuusarvon tulkitsee ”jakamisen” asteikkona klustereiden välillä, kuuluvuusarvo on asianmukainen. Intuitiivisesti ajatellen, jos kuuluvuusarvon tulkitsee klusteriin ”kuulumisena”, instanssilla B pitäisi olla pienempi kuuluvuusarvo klustereihin kuin instanssilla A , sillä etäisyytensä takia B edustaa klustereita huonommin kuin A . [Krishnapuram and Keller 1993]

FCM-menetelmän kuuluvuusarvon voi siis tulkita jakamisen asteena, mutta ei tyypillisyyden asteena. Vaikka kahden instanssin etäisyys johonkin tiettyyn klusteriin olisi sama, niillä voi silti olla täysin eri kuuluvuusarvot kyseiseen klusteriin. Vastaavasti kahdella instanssilla voi olla sama kuuluvuusarvo johonkin klusteriin, vaikka ne olisivat täysin eri etäisyyksien päässä klusterista. Tästä johtuen poikkeamat aineistossa huonontavat FCM-menetelmän suorituskykyä. [Krishnapuram and Keller 1996.] PCM-menetelmässä poikkeamalla on matalat tyypillisyydsarvot kaikkiin klustereihin, joten niiden vaikutus klusterointiin on mitätön [Krishnapuram and Keller 1993]. Tämä on mahdollista PCM-menetelmän tyypillisyydsarvojen summarajoituksen puutteen takia.

3.3.4. Päällekkäiset klusterit PCM-menetelmässä

Kuitenkin PCM-menetelmässäkin on puutteensa, sillä sen on havaittu tuottavan päällekkäisiä (coincident) klustereita [Barni *et al.* 1996; Krishnapuram and Keller 1996]. Kuten edellä mainittu, PCM-menetelmässä yhden instanssin tyypillisyydsarvojen summan ei tarvitse olla tasan yksi. Instanssin tyypillisyydsarvojen summarajoituksen puuttuessa puuttuvat myös klustereiden väliset suhteet, joka voi johtaa päällekkäisten klustereiden syntymiseen [Yu and Fan 2017].

Koska PCM etsii keskittymiä, jokainen algoritmin löytämä klusteri vastaa tiheää aluetta aineistossa. Jos parametrin c arvo on suurempi kuin keskittymien määrä aineistossa, PCM voi silti löytää c kappaletta klustereita, joista osa voi olla identtisiä. PCM ei löydä enempää toisistaan eroavia klustereita, kuin aineistossa on keskittymiä. Etsittävien klustereiden määrän lisääminen ei kuitenkaan takaa kaikkien aineiston keskittymien löytymistä. Tämä johtuu siitä, että PCM on herkkä huonolle klustereiden alustukselle eli huonosti alustetut klustereiden keskipisteiden sijainnit voivat vaikeuttaa keskittymien löytymistä. [Krishnapuram and Keller 1996]

FCM-menetelmästä saatavia klustereita ja kuuluvuusarvoja voidaan käyttää hyvinä alustuksina PCM-menetelmän klustereille, jos aineistossa ei ole paljoa poikkeamia. FCM-menetelmän tulokset ovat huonolaatuisia, jos aineistossa on

liikaa poikkeamia. Näin ollen, jos tällaisia huonolaatuisia tuloksia käytettäisiin PCM-menetelmän alustamiseen, myös PCM-menetelmän alustukset olisivat huonolaatuisia. Tämä taas voi johtaa harhaanjohtaviin PCM-menetelmän tuloksiin. [Krishnapuram and Keller 1996]

Esimerkkinä huonosta klustereiden alustuksesta PCM-menetelmässä oletetaan, että aineistossa olisi kaksi keskittymää ja siitä etsittäisiin kahta klusteria, eli $c = 2$. Ihannetapauksessa klusterit päätyisivät eri keskittymien kohdille, jolloin molemmat keskittymät löytyisivät ja päällekkäisiä klustereita ei olisi. Kuitenkin, jos molemmat klusterit alustetaan saman keskittymän lähelle, molemmat klusterit voivat päätyä kyseisen keskittymän kohdalle. Tällöin aineiston toista keskittymää ei löydetäisi. Lisäämällä etsittävien klustereiden määrää, voidaan nostaa kaikkien keskittymien löytymisen mahdollisuutta. Jos klusterit alustettaisiin sattumanvaraisesti, on aina olemassa mahdollisuus, että klusterit alustetaan lähelle toisiaan suuremmallakin klusterimäärällä, jolloin keskittymiä voi jäädä löytymättä. Suuremman klusterimäärän etsiminen on myös laskennallisesti vaativampaa.

Toisaalta jos $c = 1$, PCM voi löytää yhden tiheän keskittymän. Suorittamalla algoritmia toistuvasti uudelleen ja aina poistamalla löydetty klusteri aineistosta, voidaan löytää keskittymien oikea määrä. Kun uutta klusteria ei enää löydy, kaikki keskittymät on jo poistettu. FCM-menetelmässä esiintyvä klusterin kelvollisuusongelma voidaan siis PCM-menetelmässä välttää suorittamalla algoritmia toistuvasti ja eliminoimalla klustereita aineistosta joko yksitellen tai useamman kerrallaan. [Krishnapuram and Keller 1996]

3.4. Fuzzy-possibilistic c-means

Fuzzy-possibilistic c-means -menetelmä (FPCM) kehitettiin FCM- ja PCM-menetelmien hybridiksi, joka yhdistäisi näiden menetelmien hyötyjä ja samalla parantaisi niiden heikkouksia. FPCM-menetelmän esittelivät Pal ja muut [1997].

FPCM laskee sekä FCM-menetelmän kuuluvuusarvot että PCM-menetelmän tyypillisyyssarvot instansseille. Molemmat näistä arvoista ovat hyödyllisiä aineiston rakenteen tulkinnassa. Kuuluvuusarvot ovat hyödyllisiä, jos instanssit halutaan jakaa koviin luokkiin, sillä kuuluvuusarvojen avulla nähdään, mihin klusteriin instanssi kuuluu vahvimmin. Tyypillisyyssarvojen avulla taas voidaan vähentää poikkeamien haitallista vaikutusta klusterointiin. [Pal *et al.* 2005]

Kaava 14 on FPCM-algoritmin minimoitava kustannusfunktio, joka riippuu sekä kuuluvuus- että tyypillisyyssarvoista.

$$J_{m,\eta}(U, T, V) = \sum_{j=1}^c \sum_{i=1}^n (u_{ij}^m + t_{ij}^\eta) d_{ijA}^2, \quad (14)$$

$$d_{ijA} = \|x_i - v_j\|_A.$$

Kuuluvuus- ja tyypillisyyssarvoille pätevät ehdot

$$0 \leq u_{ij}, t_{ij} \leq 1 ; \sum_{j=1}^c u_{ij} = 1 \forall i ; \sum_{i=1}^n t_{ij} = 1 \forall j ; \sum_{j=1}^c t_{ij} > 0 \forall i.$$

Kuuluvuus- ja tyypillisyyssarvot siis ovat väliltä [0,1], instanssin kuuluvuusarvojen summa on 1, klusterin tyypillisyyssarvojen summa on 1 ja instanssilla täytyy olla vähintään yksi positiivinen tyypillisyyssarvo. Tyypillisyyssarvojen summarajoitteen avulla vältetään päällekkäisten klustereiden syntymisen ongelma. Kuten edellä on mainittu, tyypillisyyssarvojen summarajoituksen puute johti klustereiden välisten suhteiden puuttumiseen, joka taas johti päällekkäisten klustereiden syntymiseen PCM-menetelmässä. [Pal *et al.* 1997]

FPCM-menetelmän kuuluvuusarvot päivitetään kaavalla 15,

$$u_{ij} = \left(\sum_{k=1}^c \left(\frac{d_{ijA}}{d_{ikA}} \right)^{\frac{2}{m-1}} \right)^{-1}, \forall i, j, \quad (15)$$

tyypillisyyssarvot kaavalla 16,

$$t_{ij} = \left(\sum_{k=1}^n \left(\frac{d_{ijA}}{d_{kjA}} \right)^{\frac{2}{\eta-1}} \right)^{-1}, \forall i, j, \quad (16)$$

ja klustereiden keskipisteet kaavalla 17

$$v_j = \frac{\sum_{i=1}^n (u_{ij}^m + t_{ij}^\eta) x_i}{\sum_{i=1}^n (u_{ij}^m + t_{ij}^\eta)}, \forall j. \quad (17)$$

Kaava 17 on lähellä kaavoja 3 ja 11, eli FCM- ja PCM-menetelmien klustereiden keskipisteiden laskemisen kaavoja. FPCM-menetelmä ottaa huomioon sekä kuuluvuusarvot että tyypillisyyssarvot klustereiden keskipisteitä laskettaessa. [Pal *et al.* 1997]

Palin ja muiden [1997] tekemien testien mukaan FPCM-menetelmä parantaa FCM- ja PCM-menetelmien kärsimiä ongelmia. FPCM-menetelmällä vaikuttaisi olevan parempi poikkeuksiensietokyky, kuin FCM-menetelmällä ja FPCM ei myöskään tuota päällekkäisiä klustereita, kuten PCM [Pal *et al.* 1997]. Pal ja muut [1997] päättelivät testiensä pohjalta, että paras parametrin η arvo voisi olla välillä [3, 5]. Muuttujan m arvolla ei näyttäisi olevan yhtä suurta vaikutusta klusteroinnin tuloksiin, eivätkä Pal ja muut [1997] myöskään ehdottaneet sille "suositusarvoa".

3.5. Possibilistic-fuzzy c-means

Kuten FPCM, myös possibilistic-fuzzy c-means -menetelmä (PFCM) on FCM- ja PCM-menetelmien hybridi. PFCM on lähellä FPCM-menetelmää, mutta menetelmissä on joitain eroja. Pal ja muut [2005] esittivät PFCM-menetelmän korjaamaan ongelman FPCM-menetelmässä.

Kuten tutkielmassa aiemmin mainittiin, FPCM-menetelmässä kaikkien yhteen klusteriin liittyvien tyypillisyyssarvojen summan täytyy aina olla tasan yksi, ehdon $\sum_{i=1}^n t_{ij} = 1 \forall j$ mukaisesti. Tämä rajoitus johti epärealististen tyypillisyyssarvojen syntymiseen suurempia aineistoja klusteroidessa, sillä aineiston koon kasvaessa yksittäisten instanssien tyypillisyyssarvot pienenevät summarajoituksen takia [Pal *et al.* 2005].

Korjatakseen liian pienten tyypillisyyssarvojen ongelman, Pal ja muut [2005] poistivat tyypillisyyssarvojen summarajoitteen PFCM-menetelmässä. PFCM-menetelmässä yhteen klusteriin liittyvien tyypillisyyssarvojen summan ei siis tarvitse olla tasan yksi. Muut FPCM-menetelmässä määritellyt kuuluvuus- ja tyypillisyyssarvojen rajoitteet pätevät PFCM-menetelmässäkin.

Tämä muutos johtaa kaavan 18 mukaiseen PFCM-menetelmän minimoitavaan kustannusfunktioon

$$J_{m,\eta}(U, T, V) = \sum_{i=1}^n \sum_{j=1}^c (au_{ij}^m + bt_{ij}^\eta) \|x_i - v_j\|_A^2 + \sum_{j=1}^c \eta_j \sum_{i=1}^n (1 - t_{ij})^\eta, \quad (18)$$

jossa $a > 0$ ja $b > 0$ määrittävät kuuluvuusarvon ja tyypillisyyssarvon suhteellista tärkeyttä funktiossa. Jos $b = 0$ ja $\eta_j = 0$ jokaisella j arvolla, PFCM kustannusfunktio muuttuu kaavan 1 mukaiseksi FCM-menetelmän kustannusfunktioiksi. Vastaavasti jos $a = 0$, PFCM funktio muuttuu kaavan 9 mukaiseksi PCM-menetelmän funktioiksi. PFCM-menetelmän siis saa käyttäytymään enemmän FCM- tai PCM-menetelmän kaltaisesti muuttujien a ja b arvoja muuttamalla. [Pal *et al.* 2005]

PFCM-menetelmässä kuuluvuusarvojen ja tyypillisyyssarvojen laskennassa käytettävä d_{ijA} määritellään $d_{ijA} = \|x_i - v_j\|_A > 0 \forall i$. Kuuluvuusarvot lasketaan käyttäen kaavaa 15, eli samalla kaavalla kuin FPCM-menetelmässä. Tyypillisyyssarvot lasketaan kaavalla 19, joka muistuttaa hieman PCM-menetelmän vastaavaa kaavaa 12.

$$t_{ij} = \frac{1}{1 + \left(\frac{b}{\eta_j} d_{ijA}^2\right)^{1/(\eta-1)}} \quad (19)$$

Klustereiden keskipisteet lasketaan kaavalla 20, joka myös muistuttaa edellä mainittujen menetelmien vastaavia kaavoja. Kaavaan 17 verrattuna kaavaan 20 on lisätty suhteutusarvot a ja b . [Pal *et al.* 2005]

$$v_j = \frac{\sum_{i=1}^n (au_{ij}^m + bt_{ij}^\eta)x_i}{\sum_{i=1}^n (au_{ij}^m + bt_{ij}^\eta)}, \forall j \quad (20)$$

Pal ja muut [2005] eivät osanneet suositella sopivia arvoja parametreille a , b , m ja η ennen lisätutkimusten suorittamista. Samassa julkaisussa tehtyjen testien perusteella tosin näyttäisi siltä, että PFCM lievittäisi FCM-, PCM- ja FPCM-menetelmien ongelmia. PFCM ei ole herkkä poikkeamille, voi välttää päällekkäisiä klustereita eikä luo hyvin pieniä tyypillisyyсарvoja suurilla aineistoilla.

4. Yhteenveto

Datan määrän lisääntyessä nopeaa tahtia tarvitaan tehokkaita tapoja sen käsittelyyn, sillä suurten datamäärien luokittelu käsin ei ole mahdollista. Klusteroinnilla pyritään ryhmittelemään olemassa olevaa luokittelematonta dataa joidenkin datan ominaisuuksien suhteen, jotta dataa voitaisiin analysoida tarkemmin. Sumeassa klusteroinnissa instanssit voivat kuulua useaan klusteriin samanaikaisesti.

FCM on käytetyin sumean klusteroinnin menetelmä ja sen pohjalta on luotu monia uusia menetelmiä, kuten kaikki tässä tutkielmassa käsitellyt menetelmät. GK on parempi tunnistamaan erimuotoisia klustereita kuin FCM, ja klusterit voivat myös muodostua erimuotoisiksi toisistaan. PCM etsii tiheitä keskittymiä ja korjaa FCM-menetelmän herkkyyden poikkeamille. FPCM on FCM- ja PCM-menetelmien hybridi, joka PCM-menetelmän tavoin korjaa FCM-menetelmän herkkyyden poikkeamille, mutta myös korjaa PCM-menetelmän päällekkäisten klustereiden ongelman. PFCM korjaa FCM-menetelmän herkkyyden poikkeamille, PCM-menetelmän päällekkäisten klustereiden ongelman ja FPCM-menetelmän pienten tyypillisyyсарvojen ongelman klusteroidessa suuria aineistoja.

Uusia sumean klusteroinnin menetelmiä kehitetään edelleen, sillä klusterointi on tärkeä apuväline suurten datamäärien analysoinnissa ja datan analysoinnille on jatkuvaa tarvetta monilla eri aloilla.

Viiteluettelo

- Osama Alia, Rajeswari Mandava, and Mohd Aziz. 2011. A hybrid harmony search algorithm for MRI brain segmentation. *Evolutionary Intelligence* 4, 31-49.
- Mauro Barni, Vito Cappellini, and Alessandro Mecocci. 1996. Comments on "A possibilistic approach to clustering". *IEEE Transactions on Fuzzy Systems* 4, 3, 393-396.

- James C. Bezdek. 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Springer.
- Jacek Blazewicz, Wieslaw Kubiak, Tadeusz Morzy, and Marek Rusinkiewicz. 2012. *Handbook on Data Management in Information Systems*. Springer.
- Dibya Jyoti Bora and Anil Kumar Gupta. 2014. A comparative study between fuzzy clustering algorithm and hard clustering algorithm. *IJCTT* 10, 2, 108-113.
- Xiaoguang Chang, Wei Li, and Jay Farrell. 2000. A c-means clustering based fuzzy modeling method. In: *Proc. of the Ninth IEEE International Conference on Fuzzy Systems* 2, 937-940.
- Chaomu Chaomurilige, Jian Yu, and Miin-Shen Yang. 2015. Analysis of parameter selection for Gustafson–Kessel fuzzy clustering using Jacobian matrix. *IEEE Transactions on Fuzzy Systems* 23, 6, 2329-2342.
- Chaomu Chaomurilige, Jian Yu, and Miin-Shen Yang. 2017. Deterministic annealing Gustafson-Kessel fuzzy clustering algorithm. *Information Sciences* 417, 435-453.
- Kai-Hsiang Chuang, Ming-Jang Chiu, Chung-Chih Lin, and Jyh-Horng Chen. 1999. Model-free functional MRI analysis using Kohonen clustering neural network and fuzzy c-means. *IEEE Transactions on Medical Imaging* 18, 12, 1117-1128.
- Christian Correa, Constantino Valero, Pilar Barreiro, Maria-Paz Diago, and Javier Tardaguila. 2011. A comparison of fuzzy clustering algorithms applied to feature extraction on vineyard. *Inteligencia Artificial* 1, 778.
- Joseph C. Dunn. 1973. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics* 3, 3, 32-57.
- Christian Döring, Marie-Jeanne Lesot, and Rudolf Kruse. 2006. Data analysis with fuzzy clustering methods. *Computational Statistics & Data Analysis* 51, 1, 192-214.
- Audrey P. Gasch and Michael B. Eisen. 2002. Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome Biology* 3, 11, 1-22.
- Daniel Graves and Witold Pedrycz. 2010. Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study. *Fuzzy Sets and Systems* 161, 4, 522-543.
- Donald E. Gustafson and William C. Kessel. 1978. Fuzzy clustering with a fuzzy covariance matrix. In: *Proc. of the 1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, 761-766.
- Jiawei Han and Micheline Kamber. 2006. *Data Mining: Concepts and Techniques*. Elsevier.

- Kaoru Hirota and Witold Pedrycz. 1999. Fuzzy computing for data mining. *Proceedings of the IEEE* 87, 9, 1575-1600.
- Naresh S. Iyer, Abraham Kandel, and Moti Schneider. 2000. Feature-based fuzzy classification for interpretation of mammograms. *Fuzzy Sets and Systems* 114, 2, 271-280.
- Liping Jing, Dong Deng, and Jian Yu. 2014. Weighting exponent selection of fuzzy c-means via Jacobian matrix. In: *Proc. of the KSEM 2014*, 115-126.
- Raghu Krishnapuram and James M. Keller. 1993. A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems* 1, 2, 98-110.
- Raghu Krishnapuram and James M. Keller. 1996. The possibilistic c-means algorithm: insights and recommendations. *IEEE Transactions on Fuzzy Systems* 4, 3, 385-393.
- Raghu Krishnapuram and Jongwoo Kim. 1999. A note on the Gustafson-Kessel and adaptive fuzzy clustering algorithms. *IEEE Transactions on Fuzzy Systems* 7, 4, 453-461.
- Samir Kunwar. 2013. Text documents clustering using k-means algorithm. <https://www.codeproject.com/Articles/439890/Text-Documents-Clustering-using-K-Means-Algorithm>. Checked 19.3.2018.
- Marie-Jeanne Lesot and Rudolf Kruse. 2008. Gustafson-Kessel-like clustering algorithm based on typicality degrees. In: Bernadette Bouchon-Meunier, Christophe Marsala, Maria Rifqi and Ronald R. Yager (eds.), *Uncertainty and Intelligent information Systems*. World Scientific, 117-130.
- Anastasia Makarevich. 2016. Gustafson-Kessel algorithm. http://rstudio-pubs-stat-s3.amazonaws.com/248394_e80ad79e3f0843ce9631600a59eefcfb.html. Checked 12.2.2018.
- Matteo Matteucci. Clustering - Fuzzy c-means. https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/cmeans.html. Checked 6.2.2018.
- Jian-Ping Mei, Yangtao Wang, Lihui Chen, and Chunyan Miao. 2017. Large scale document categorization with fuzzy clustering. *IEEE Transactions on Fuzzy Systems* 25, 5, 1239-1251.
- Nikhil Ranjan Pal and James C. Bezdek. 1995. On cluster validity for the fuzzy c-means model. *IEEE Transactions on Fuzzy Systems* 3, 3, 370-379.
- Nikhil Ranjan Pal, Kuhu Pal, and James C. Bezdek. 1997. A mixed c-means clustering model. In: *Proc. of the 6th International Fuzzy Systems Conference* 1, 11-21.

- Nikhil Ranjan Pal, Kuhu Pal, James M. Keller, and James C. Bezdek. 2005. A possibilistic fuzzy c-means clustering algorithm. *IEEE Transactions on Fuzzy Systems* 13, 4, 517-530.
- Wanczyk Phillips, Robert Velthuizen, Surasak Phuphanich, Lawrence O. Hall, Laurence Clarke, and Martin L. Silbiger. 1995. Application of fuzzy c-means segmentation technique for tissue differentiation in MR images of a hemorrhagic glioblastoma multiforme. *Magnetic Resonance Imaging* 13, 2, 277-290.
- Mahmoud Ramze Rezaee, Pieter M. J. van der Zwet, Boudewijn P. F. Lelieveldt, Rob J. van der Geest, and Johan H. C. Reiber. 2000. A multiresolution image segmentation technique based on pyramidal segmentation and fuzzy clustering. *IEEE Transactions on Image Processing* 9, 7, 1238-1248.
- Enrique H. Ruspini. 1969. A new approach to clustering. *Information and Control* 15, 1, 22-32.
- Veit Schwämmle and Ole Nørregaard Jensen. 2010. A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. *Bioinformatics* 26, 22, 2841-2848.
- Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.
- Vishal Tandel. 2015. Cluster analysis for market segmentation. <https://www.slideshare.net/vishtandel1991/cluster-analysis-for-market-segmentation>. Checked 30.4.2018.
- Pekka Teppola, Satu-Pia Mujunen, and Pentti Minkkinen. 1999. Adaptive fuzzy c-means clustering in process monitoring. *Chemometrics and Intelligent Laboratory Systems* 45, 1-2, 23-38.
- Weina Wang and Yunjie Zhang. 2007. On fuzzy cluster validity indices. *Fuzzy Sets and Systems* 158, 19, 2095-2117.
- Wolfram. 2004. Fuzzy Clustering. <http://reference.wolfram.com/legacy/applications/fuzzylogic/Manual/12.html>. Checked 4.2.2018.
- Haiyan Yu and Jiu-Lun Fan. 2017. Cutset-type possibilistic c-means clustering algorithm. *Applied Soft Computing* 64, 401-422.
- Jian Yu, Qiansheng Cheng, and Houkuan Huang. 2004. Analysis of the weighting exponent in the FCM. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 34, 1, 634-639.
- Lotfi A. Zadeh. 1965. Fuzzy sets. *Information and Control* 8, 3, 338-353.
- Osmar Zaiane. 1999. Data clustering (Principles of knowledge discovery in databases). Slide 13.

<https://webdocs.cs.ualberta.ca/~zaiane/courses/cmp690/slides/Chapter8/index.htm>. Checked 30.4.2018.

KaiLe Zhou, Chao Fu, and ShanLin Yang. 2014. Fuzziness parameter selection in fuzzy c-means: The perspective of cluster validation. *Science China Information Sciences* 57, 11, 1-8.

Co-Teacher verkkopalvelun käyttöliittymän esittely ja evaluointi

Joona Tontti

Tiivistelmä

Tutkielma tutustuu Co-Teacher verkkopalvelun käyttöliittymään ja arvioi siinä käytettyjä ratkaisuita lähdemateriaalin pohjalta.

Avainsanat ja -sanonnat: Käyttöliittymä, evaluointi

1. Johdanto

Co-Teacher on suomalaisen startup-yrityksen Shoulders of Giantsin ensimmäinen verkkopalvelu. Palvelu on suunniteltu ensisijaisesti korkeakouluopettajien ja -oppilaiden käyttöön. Co-Teacherin tavoitteena on vähentää kurssien keskeyttävien oppilaiden määrää ja opettajan työkuormitusta. Tähän pyritään tarjoamalla oppilaille hallinnollisia apuvälineitä, kuten kustomoitava lukujärjestyksen tuottaja, sekä aina käytettävissä oleva chattibotti, jolta saa nopean vastauksen yleisiin kysymyksiin. Opettajien työtaakkaa taas pyritään keventämään vähentämällä rutiininomaista työtä, kuten yksinkertaisiin sähköpostiviesteihin vastaamista ja väliarvioiden tekemistä. Keskeisessä osassa edellä mainittujen toiminnallisuuksien mahdollistamisessa toimii IBM Watsonin tekoälysovellusrajapinnat. Tällä hetkellä Co-Teacher on alpha-vaiheessa, joten käyttöliittymää, sekä uusia ominaisuuksia kehitetään kaiken aikaa. Tutkimuksen päälukuihin sisältyy aina päivämäärä, jolloin kyseinen luku on saatu valmiiksi.

Tutkielmassa arvioidaan Co-Teacherin käyttöliittymää, sekä sen käyttökokemusta alan lähdemateriaalin pohjalta. Tutkielman tuloksia tullaan käyttämään Co-Teacherin kehitystyössä paremman käyttäjäkokemuksen saavuttamiseksi. Näin ollen tutkielmassa keskitytään palvelun mahdollisiin ongelmakohditiin. Ongelmakohdalla tarkoitetaan käyttöliittymäratkaisua, joka saattaa huonontaa palvelun käytettävyyttä. Palvelun käyttöliittymässä toistuu hyvin paljon samankaltaisia ongelmakohditiä, joten tutkielman luettavuuden parantamiseksi selitän samanlaisten ongelmakohditiien syyt ja parannusehdotuksen lähdemateriaalin pohjalta korkeintaan kahden ensimmäisen ongelman kohdalla. Seuraavan kerran samanlaisen ongelman esiintyessä ainoastaan nostan sen tekstissä esille, ja totean syyn kyseiseen ongelmaan. Tutkimuksessa ei tarkastella palvelun responsiivisuutta.

Co-Teacherin käyttöliittymä voidaan jakaa kahteen osaan pääkäyttäjien mukaan: oppilaan- ja opettajan puoleen. Kumpikin puoli on suurilta osin samanlaisia, mutta myös eroavaisuuksia löytyy. Käyttöliittymä käydään läpi päänavigoinnin mukaisessa järjestyksessä, jolloin ensin käydään läpi yhteiset alisivut, ja viimeiseksi eriävät sivut. Jos yhteisillä sivuilla ilmenee eroavaisuuksia toiminnollisuuksissa, ensin käydään läpi yhteiset toiminnollisuudet, sitten opiskelijan toiminnollisuudet ja viimeiseksi opettajan toiminnollisuudet. Alla on tarkempi rakenne.

Luvussa 2 tarkastellaan "Home" sivu eli palvelun kotisivu. Luvussa 3 "Canvas" sivua, josta löytyy sekä käyttäjän aktiiviset kurssit, että kaikki saatavilla olevat kurssit. Samalla tarkastellaan yhden aktiivisen kurssin kurssisivu. Opettajalla on myös mahdollisuus lisätä uusi kurssi, ja tämä toiminnallisuus käydään läpi luvun lopuksi. Luvussa 4 käydään käyttäjän asetukset sisältävä "Settings" sivu läpi. Luvussa 5 arvioidaan "Evaluation" sivu, joka sisältää tekstin läpikäymistä avustavan työkalun. Luvussa 6 käydään läpi palautteen keräämiseen tarkoitettu "Feedback" sivu. Luvussa 7 tutustutaan ainoastaan opiskelijan puolelta löytyvään "Planner" alisivuun, jolla opiskelija pystyy luomaan räätälöidyn lukujärjestyksen. Luvussa 8 arvioidaan opettajan puolelta löytyvä "Manage" sivu, joka jakautuu kokoontumistilojen- ja chattibotin hallitsemiseen. Luku 9 on jätetty yhteenvedolle, jonka jälkeen seuraa lähdeluettelo.

Palvelun arviointiin on käytetty pöytäkonetta Windows 10 käyttöjärjestelmällä, 21" näyttöä 16:9 kuvasuhteella ja Google Chrome Versio 65.0.3325.181 selainta.

2. Aloitussivu

Luku valmistunut 12.05.2018

Palvelun aloitussivu (kuva 1). Sivü koostu seuraavista elementeistä: päänavigointipalkki, lukujärjestys, chattibotti, deadline-muistutin, chattibotin kontekstin vaihtoikkuna ja aktiviteetit. Aloitussivulle pääsee painamalla palvelun logoa tai "Home" linkkiä painamalla päänavigointipalkista. Sivun sisällön osalta opiskelijan ja opettajan näkymät näyttävät samalta.

The screenshot shows the service's start page. At the top is a blue navigation bar with the logo on the left and links for Home, Canvas, Settings, Evaluation, Manage, and Feedback on the right. The user is logged in as Joonas Tontti. Below the navigation bar is a calendar view for the week of 27.03 - 03.04.18. The calendar shows a grid with days from Tue 3/27 to Tue 4/3 and times from 6am to 1pm. To the right of the calendar is a sidebar with sections: 'Upcoming deadlines' (No task deadline!), 'Ask Help from Co-Teacher' (with a dropdown menu and an Ask button), and 'Activities' (with a list of system messages and a SEE EARLIER link). At the bottom of the page is a chat window titled 'Co-Teacher' with a message: 'Hello Joonas Tontti. How can I help you this afternoon?' and a 'Send' button.

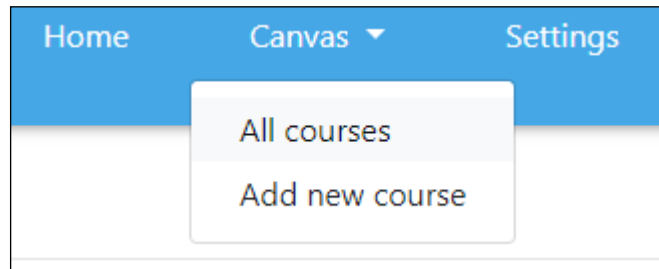
Kuva 1. Palvelun aloitussivu opettajan näkymästä.

2.1. Päänavigointipalkki

Palvelun päänavigointipalkki toistuu samanlaisena joka sivulla ja on sijoitettu sivun ylälaitaan koko sivun levyisenä. Palkin alareunan varjostus saa aikaan vaikutelman, että palkki on irti sivun taustasta. Palkki on kuitenkin staattinen eikä jää sivun yläreunaan sivua vierittäessä. Palkin vasemmassa kulmassa sijaitsee palvelun logo.

Logon vasemmalla puolella sijaitsee itse navigointilinkit. Linkkien valkoinen teksti erottuu hyvin sinisestä pohjasta. Osa linkeistä toimii pudotusvalikkoina, joiden alta löytyvillä linkeillä on leijuntaefekti (kuva 2). Päänavigointipalkissa olevilla linkeillä taas ei ole leijuntaefektiä. Linkkien painikealue jatkuu

tekstiosan ulkopuolelle, mutta painikkeilla ei myöskään ole marginaaleja sivuissa, joka tekee mahdolliseksi tietää, milloin yhden painikkeen alue muuttuu toisen painikkeen alueeksi. Palvelun yhdenmukaisuutta, tilanhahmottamista ja virheiden ehkäisyä voitaisiin parantaa lisäämällä myös päälinkkeihin leijuntaefekti [Nielsen 1995].



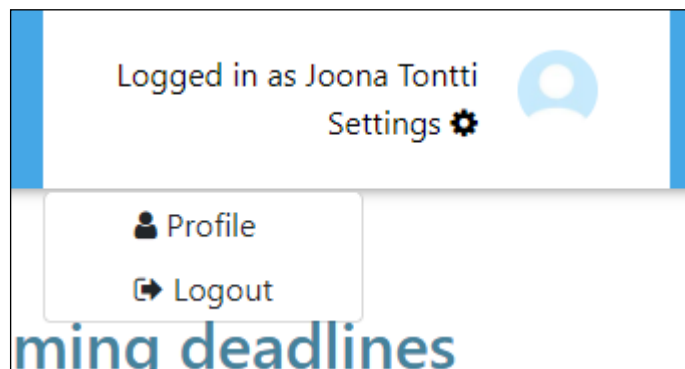
Kuva 2. Pudotusvalikon linkkien leijuntaefekti.

Sivulta ei mistään kohtaa käy ilmi käyttäjän silloinen sijainti palvelussa. Nielsenin [1995] mukaan käyttäjän ei tulisi tarvita muistaa tällaisia asioita. Myös palvelun tilan hahmottaminen paranisi, jos esimerkiksi aktiivista sivua vastaava linkki olisi päänavigointipalkissa erivärinen muihin verrattuna. Myös leivänmurunavigointi, eli sivulla kuljetun hierarkian esittäminen, voisi toimia navigointipalkin alapuolella, vaikkakin palvelun struktuuri on melko matala. Nielsenin [2007] mukaan leivänmurunavigoinnista ei ainakaan ole mitään haittaa.

Päänavigointipalkin oikeasta kulmasta löytyy käyttäjän profiili-elementti. Elementtiä painamalla aukeaa pudotusvalikko (kuva 3). "Settings"-painiketta ei siis pysty painamaan erikseen. Ratkaisu on vähintäänkin normaalista poikkeava. Verkkosivuilla on noussut jotakuinkin standardiksi, että omaa nimeä ja/tai profiilikuvaa painamalla käyttäjä pääsee profiiliasetuksiin. Käytettävyyden parantamiseksi tulisikin nojautua käyttöliittymien standardeihin ja toteuttaa toiminnallisuus vastaamaan käyttäjän odotuksia. [Nielsen 1995; Tognazzini 2014]. Jos käyttäjä pääsisi profiiliasetuksiin edellä mainitulla tavalla, "Profile"-painiketta ei tarvitsisi. Näin ollen myöskään pudotusvalikkoa ei tarvittaisi, sillä "Logout" painike voisi olla "Settings" painikkeen tilalla. Täten myös yksinkertaistettaisiin käyttöliittymää, joka Nielsenin [1995] ja Whittentonin [2013] mukaan parantaa käytettävyyttä, muun muassa vähentämällä kognitiivista kuormaa.

Navigointipalkin nykyinen design aiheuttaa myös toistoon liittyvän ongelman. "Profile"-painikkeesta käyttäjä pääsee samalle asetussivulle kuin pää-

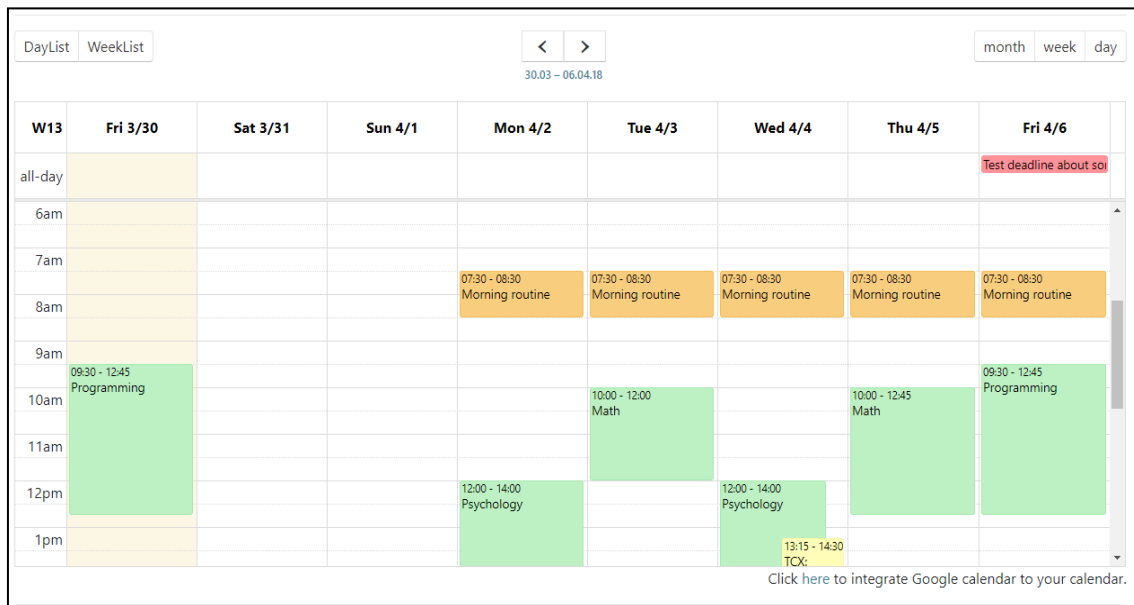
navigoinnin ”Settings”-linkistä. Tämä sekä vähentää sivun yhtenäisyyttä (erinimiset linkit, sama kohde), että aiheuttaa duplikaatit (sivulla kaksi ”Settings” tekstiä). Kumpikin seikka vähentää sivun käytettävyyttä [Nielsen 1995; 2002]. ”Settings”-sivu käydään tarkemmin läpi luvussa 3, mutta lyhykäisyydessään sivun sisältö sopii paremmin ”Profile”-otsikon alle. Käytettävyyden parantamiseksi päänavigoinnin ”Settings”-linkki tulisi siis poistaa.



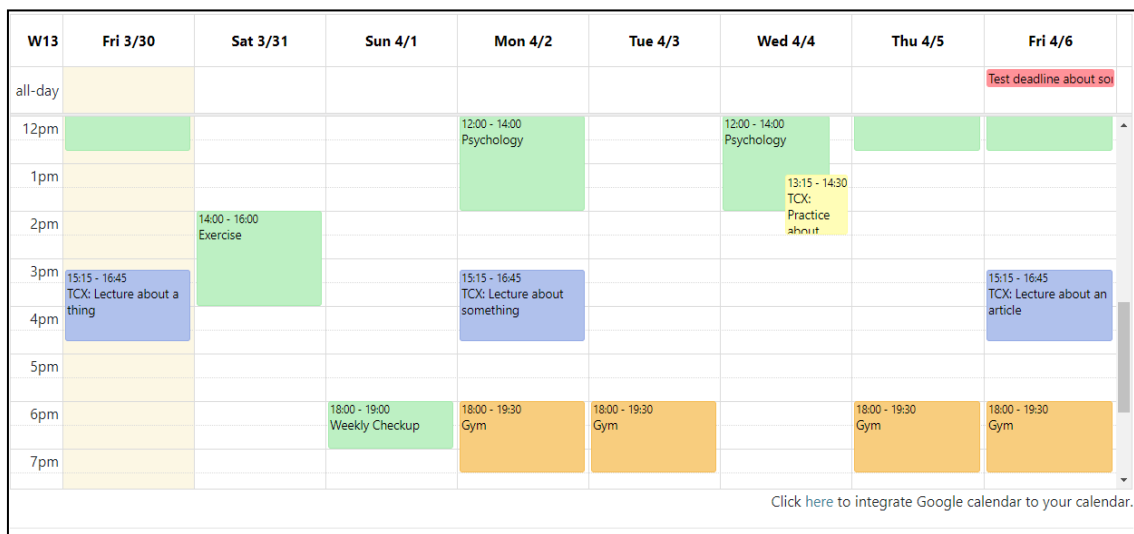
Kuva 3. Elementin koko valkoinen alue toimii painikkeena.

2.2. Lukujärjestys

Aloitussivun suurin elementti on käyttäjän lukujärjestys/kalenteri (kuva 4). Kuva 4 on oletusnäkyminen lukujärjestyksestä, kun oppilas kirjautuu sisään. Lukujärjestyksen merkinnät ovat peräisin aktiivisesta kurssista (merkattu sinisellä, punaisella ja vaaleankeltaisella kuvassa 5), itsenäisistä opiskelusessioista (merkattu vihreällä; itsenäisistä opiskelusessioista lisää luvussa 6) ja Google-kalenterista (merkattu sinapin keltaisella).



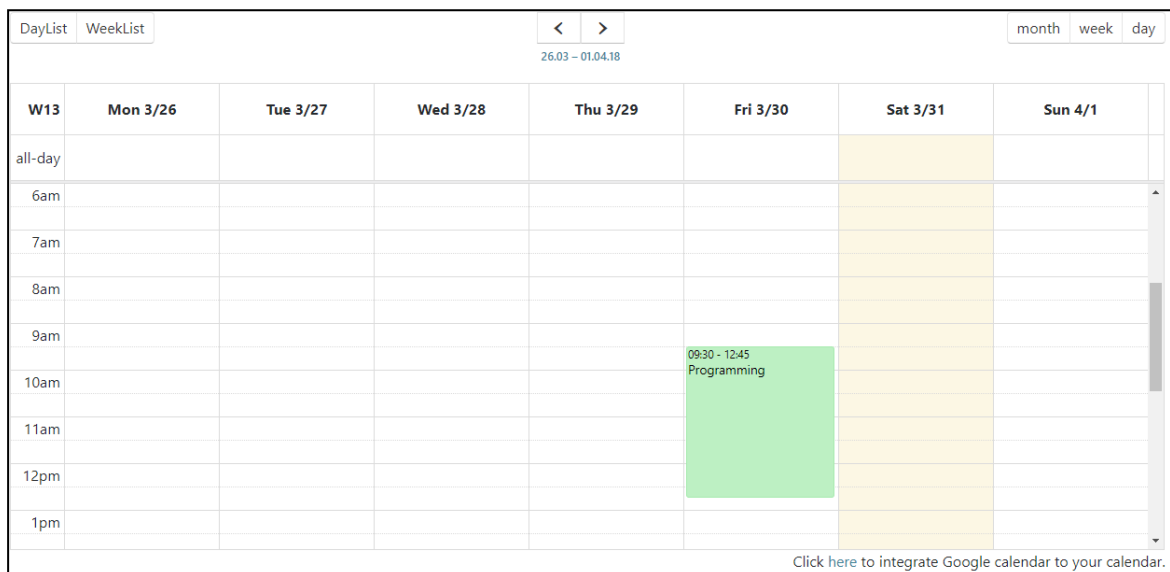
Kuva 4. Oletusnäkyä opiskelijan kalenterista.



Kuva 5. Kuvan 4 kalenteri alaspäin vieritettynä.

Oletusnäkyssä nykyinen päivä on korostettu beigellä värillä ja sijoitettu lukujärjestyksen vasempaan reunaan. Käyttäjä saattaisi luulla, että kyseinen näkymä on sama kuin viikkonäkymä, jonka voi valita oikeasta yläkulmasta "week"-painikkeesta, mutta näin ei ole (kuva 6). Oletusnäkyssä näkyy kahdeksan päivää, ja viikkonäkymän ensimmäinen päivä on aina maanantai. Tämä saattaa aiheuttaa sekaannusta, varsinkin koska valittua näkymää ei näe missään. Nielsenin [1995] ja Budiun [2014] mukaan systeemin tila pitäisi olla kokoajan näkyvissä ja palvelun tulisi auttaa käyttäjää tunnistamaan tehdyt muutok-

set, enemmän kuin muistamaan ne. Tähän oiva apu olisikin nykyistä näkymää vastaavan painikkeen visuaalinen korostaminen, josta huomataankin seuraava ongelma käytettävyydessä: oletusnäkyllä ei ole omaa painiketta. Käyttäjä ei voi siis palata oletusnäkyyn mitenkään muuten kuin päivittämällä sivun tai palaamalla sivulle uudestaan. Nielsenin [1995] mukaan tämä rajoittaa käyttäjän kontrollia, vapautta ja virheistä palautumista. Myös Tognazzinin [2014] mukaan käyttäjän toiminnot pitäisi olla käännettävissä eli tässä tapauksessa käyttäjän pitäisi päästä takaisin oletusnäkyyn. Toinen vaihtoehto olisi valita ainoastaan yksi näistä viikkonäkymistä ja tehdä siitä oletusnäky, niin että "week"-painike olisi valmiiksi korostettuna. Tämä vaihtoehto vähentäisi käyttäjien valinnanvaraa, mutta yksinkertaistaisi käyttöliittymää. Lorangerin [2015] mukaan tällainen kompromissi yleensä parantaa käytettävyyttä ja käyttäjäkokemusta. Jos päädytään valitsemaan ainoastaan yksi näky, valinnan tekemisessä voisi käyttää apuna käyttäjätutkimusta, jossa kysytään käyttäjien mieltymystä.



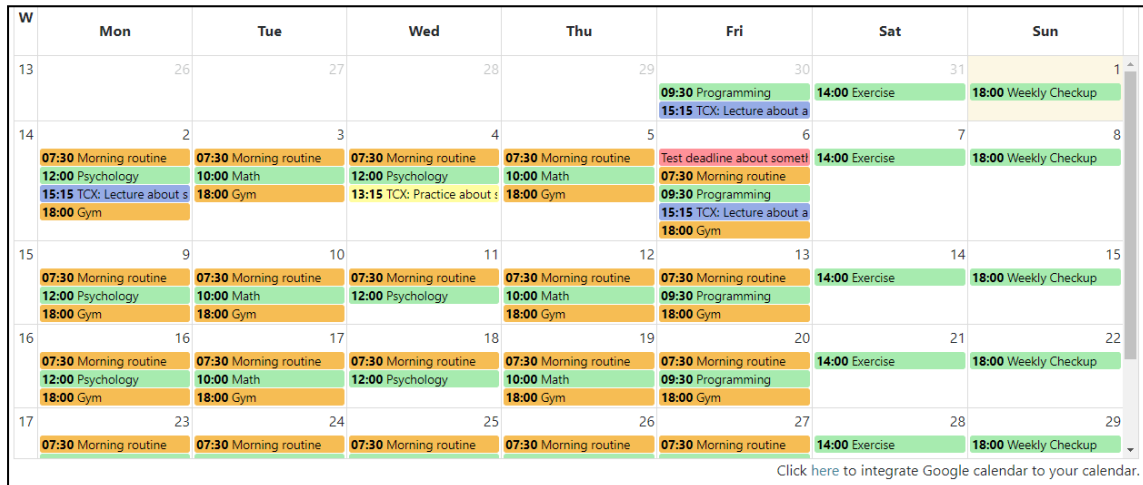
Kuva 6. "week"-painikkeesta avautuva viikkonäkymä.

Muut näkymät ovat melko yksiselitteisiä ja moitteettomia. Esimerkkeinä toimivat "WeekList"-painikkeesta aukeava lista viikon suunnitelmista (kuva 7) ja kuukausinäky (kuva 8). Jokaiselle näkymälle löytyy varmasti käyttäjänsä. Kuitenkin näkymäpainikkeiden kirjoitusasuja olisi hyvä muuttaa paremman yhtenäisyyden saavuttamiseksi [Nielsen 1995]. Palvelun lähes kaikki muut painikkeet kirjoitetaan isolla alkukirjaimella ja loput sanasta pienillä kirjaimilla.

Näin ollen "month", "week" ja "day" tekstit tulisi kirjoittaa isoilla. "WeekList" ja "DayList" painikkeiden sanat tulisi taas erottaa välilyönnillä. Yhtenäisyyden parantamiseksi myös lukujärjestyksen viikonpäivien päivämäärät tulisi ilmaista eurooppalaisittain, kun nyt ne ilmaistaan pohjoisamerikkalaisittain.



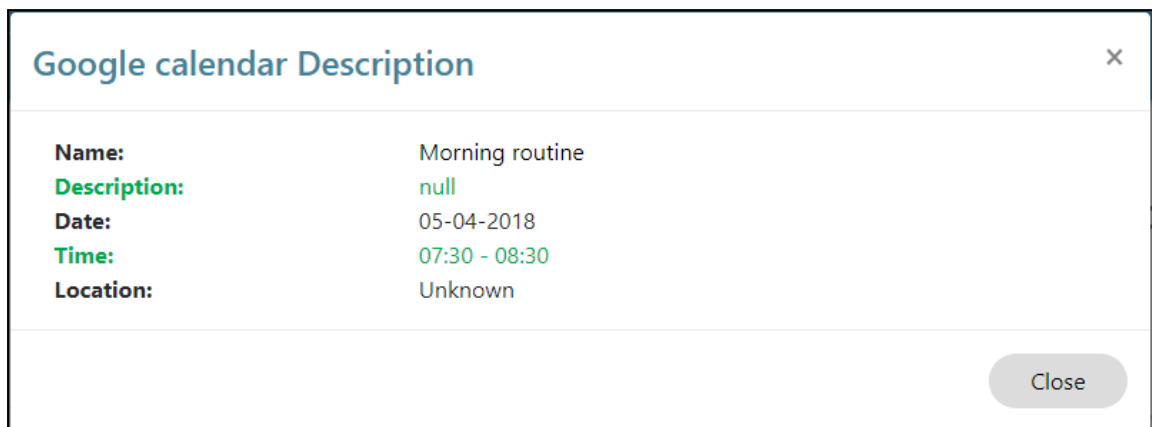
Kuva 7. "WeekList"-painikkeen näkymä.



Kuva 8. "month"-painikkeen näkymä

Lukujärjestykseen on mahdollista integroida käyttäjän Google-kalenteri oikeassa alakulmassa näkyvän linkin kautta. Esimerkkikuvissa käyttäjä on kuitenkin jo integroinut kalenterissa. Nielsenin [1995] ja Flahertyn [2015] mukaan systeemin pitäisi tiedottaa käyttäjää tapahtuneista muutoksista. Niinpä "Click here to integrate Google calendar to your calendar."-teksti voisikin muuttua indikaattoriksi, joka ilmoittaisi, että kalenteri on synkronoitu.

Lukujärjestyksen tapahtumia voi klikata, jolloin avautuu selite tapahtumasta modaali-ikkunan muodossa (kuva 9). Ikkunan otsikkona tulisi olla lukujärjestyksessä olevan tapahtuman nimi eli tässä tapauksessa "Morning routine". Näin käyttäjän ei tarvitsisi muistaa, mitä tapahtumaa hän painoi ja täten käytettävyys paranisi [Nielsen 1995; Budiu 2014]. Jos tämä on kuitenkin teknisesti mahdotonta toteuttaa, tulisi otsikon kirjoitusasu olla yhtenäinen, joko niin kirjoittamalla kaikki sanat isolla alkukirjaimella tai vain ensimmäinen sana kirjoitettuna isolla alkukirjaimella. Itse kuvaustekstit on värjätty vihreällä ja mustalla ilman minkäänlaista selkeää logiikkaa. Yhtenäisyyden parantamiseksi ja sekaannusten välttämiseksi, tekstit tulisi kirjoittaa samalla värillä. "Description" on saanut arvoksi "null", joka ei todennäköisesti ole käyttäjille tuttua kieltä. Nielsenin [1995] mukaan systeemin kielen pitäisi vastata arkielämässä käytettäviä termejä. Parempi ilmaus olisi "Not announced" tai "Unknown", jota käytetään myös "Location" yhteydessä, kun sijaintia ei ole ilmoitettu. Muilta osin lukujärjestys näyttää ja toimii odotetulla tavalla.



Kuva 9. Kalenterin "Morning routine"-suunnitelmasta klikkaamalla avautuva modaali-ikkuna.

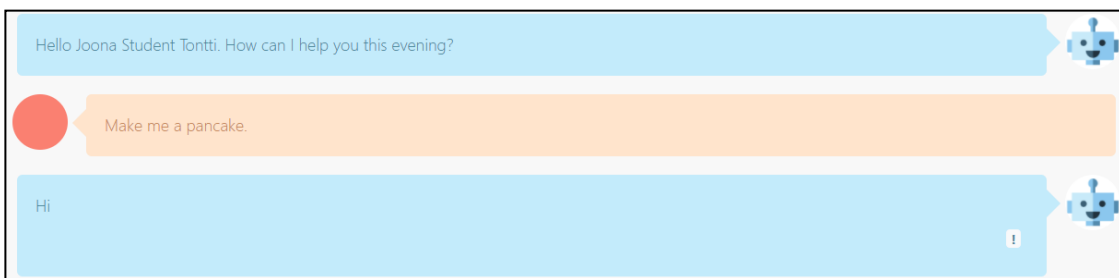
2.3. Chattibotti

Chattibottiominaisuus löytyy sivun alareunasta (kuva 1). Opiskelija voi kysyä botilta tietoja ja apua kursseihin liittyvistä asioista. Opettaja taas voi suorittaa kursseihin liittyviä järjestelyjä botin välityksellä, kuten luentosalin varauksen. Botin logiikka on vielä melko alkeellinen ja sen ymmärtämät keskusteluaiheet hyvin rajalliset. Tästä syystä en tule arvioimaan eri dialogeja, vaan keskityn käyttäjän ajan tasalla pysymiseen, vaikka botti ei esimerkiksi ymmärtäisi käyttäjän viestiä.

Chatin vasemmassa yläkulmassa on kolme eriväristä palloa, joista punainen on eri tasossa kuin kaksi muuta palloa. Punainen pallo on ilmeisesti myös pai-

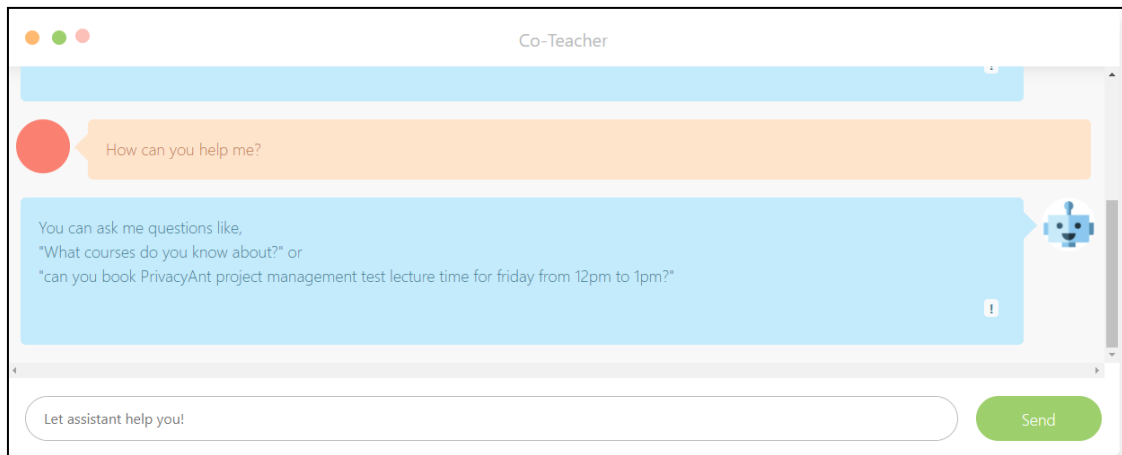
nike, sillä viemällä hiiren sen päälle, hiiren ikoni muuttuu painiketta vastaavaksi ja pallon väri muuttuu. Painiketta klikkaamalla ei kuitenkaan tapahdu mitään. Nielsenin [1995] mukaan kaikki ylimääräinen ja tarpeeton sisältö sivulla vie huomiota olennaisilta ominaisuuksilta ja hidastaa käyttäjää päämäärän saavuttamisessa. Pallot tuottavat lähinnä hämmennystä, joten ne tulisi poistaa.

Muilta osin chattibotti on helposti lähestyttävä. Kuvissa 1 ja 10 näkyy botin tervehdysviesti, joka vaihtuu hieman päivänajan mukaan. Tervehdys on puhe-
kuplassa ja antaa kuvan interaktiivisuudesta. Tekstikentässä on vihjeteksti, ja "Send"-painike on helposti nähtävä sekä omaa leijuntaefektin, tehden siitä selkeän painikkeen.



Kuva 10. Botin vastaus hölmöön kysymykseen.

Kuvassa 10 tulee esille botin vastaus tuntemattomaan pyyntöön tai kysymykseen: pelkkä "Hi"-teksti suuressa puhe-
kuplassa. Käyttäjä ajautuu siis auttamatta umpikujaan, jos hän ei ennestään tiedä miten botti pystyy auttamaan. Tapahtunutta voidaan verrata tilanteeseen, jossa käyttäjä suorittaa haun, johon ei löydy tuloksia. Whitentonin [2014] mukaan tällaisessa tilanteessa järjestelmän pitäisi ilmoittaa, että haulla ei löytynyt tuloksia ja tarjota käyttäjälle tie eteenpäin. Botin tulisi siis ilmoittaa, ettei se ymmärtänyt käyttäjän viestiä ja antaa ohjeita minkälaisissa kysymyksissä se pystyy avustamaan. Bottiin on toteutettu tällainen toiminnollisuus, mutta käyttäjän pitää osata kysyä apua omatoimisesti (kuva 11).



Kuva 11. Käyttäjän pitää ymmärtää kysyä miten chattibot pystyy auttamaan.

Käyttäjä pystyy tekemään korjausehdotuksia botin vastauksiin jos hän koee vastauksen olleen väärä. Korjausehdotus tehdään painamalla huutomerkkipainiketta botin vastauksen oikeasta laidasta. Harley [2014] muistuttaa universaalien ikoneiden olevan harvinaisia, joten tekstimerkintä ikonin yhteydessä voisi olla paikallaan. Huutomerkkin sijasta ikoniksi sopisi todennäköisesti paremmin jakoavain, joka yhdistetään korjaamiseen.

Korjausehdotuspainiketta painamalla avautuu dialogimuotoinen lomake (kuva 12). Lomake sisältää paljon turhaa tietoa, kuten "intents" ja "entities", joita käyttäjä ei todennäköisesti ymmärrä, eikä näiden tietojen näyttäminen juurikaan hyödytä vastauksen korjaamista. Loranger [2017] esittää, että palvelun kielen pitää vastata lukijan tottumuksia ja osaamista. Lisäksi harmaissa tekstikentissä olevia tekstejä ei pysty muokkaamaan, joten ne lähinnä hämäävät käyttäjää. Ainoastaan "Outputs" ja "Suggested Output" osat ovat tarpeellisia korjausehdotuksen tekemiseen, joten vain ne tulisi näyttää loppukäyttäjälle. Tähän pätee sama Nielsenin [1995] sääntö kuin ylempänä mainittuihin kolmeen eriväriseen sekä tarpeettomaan palloon.

Suggest Improvement to Message ×

Intents:

help

Entities:

Input:

how can you help me

Outputs:

You can ask me questions like,
"What courses do you know about?" or
"can you book Watson AI lecture"

Suggested Output:

please fill the suggested correct output here

Close Save changes

Kuva 12. Botin vastauksen korjausehdotus lomake.

2.4. Tulevat takarajat

Sivun oikeasta yläkulmasta löytyy elementti "Upcoming deadlines" eli tulevat takarajat (kuva 13). Elementti listaa käyttäjän kolme seuraavaa määräaikaa. Määräaika laatikoiden vasen- ja yläreuna on häivytetty, mikä tekee laatikon hahmottamisesta hieman vaikeaa. Haalea raja tai varjostus korjaisi tämän ongelman. Tehtävän sinisellä olevaa nimeä klikkaamalla aukeaa modaali-ikkuna, josta näkee tehtävän tiedot yksityiskohtaisemmin. Tämä on looginen ratkaisu, mutta itse linkki tehtävän kurssisivulle voisi olla tarpeellinen johonkin kohtaan laatikkoa. Kellonajan "pm"-merkintä on hypännyt seuraavalle sivulle, joka hidastaa tilan hahmottamisesta. Merkintä tulisi sijoittaa samalle riville kellonajan kanssa.

The image shows a screenshot of a 'Upcoming deadlines' section. It contains three items, each with a large number, a title, a description, and a deadline. The items are: 1. 'Test deadline about something' with a deadline of 'Fri 06.04 06:00 pm' and the description 'Better do something.'; 2. 'Dream Job' with a deadline of 'Wed 24.01 12:00 pm' and the description 'Reading and writing assignment for session in 24/1: Where would I like...'; 3. 'Other Case Competitions' with a deadline of 'Tue 23.01 10:30 pm' and the description 'Voluntary task: try to find competitions other than Aarhus Case Compet...'. The section is titled 'Upcoming deadlines' in a large blue font.

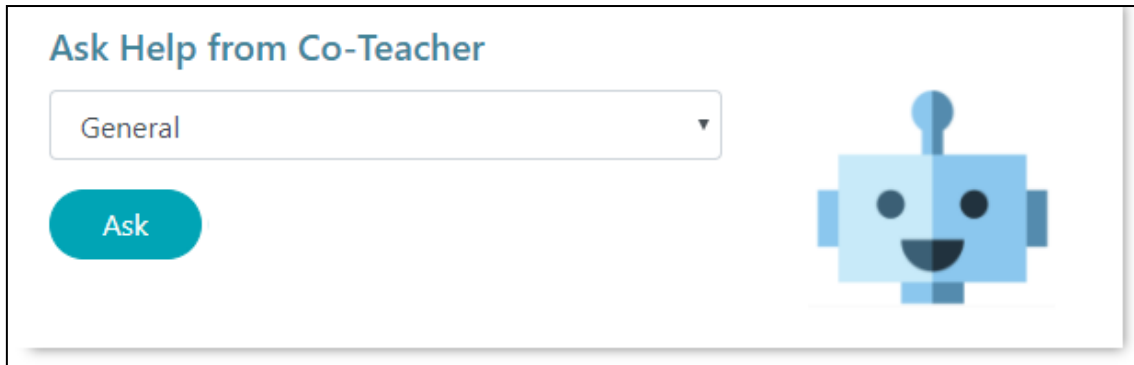
Number	Title	Description	Deadline
1	Test deadline about something	Better do something.	Deadline: Fri 06.04 06:00 pm
2	Dream Job	Reading and writing assignment for session in 24/1: Where would I like...	Deadline: Wed 24.01 12:00 pm
3	Other Case Competitions	Voluntary task: try to find competitions other than Aarhus Case Compet...	Deadline: Tue 23.01 10:30 pm

Kuva 13. Näkymä tulevista takarajoista.

2.5. Chattibotin kontekstin vaihto

Aikaraja laatikon alapuolella sijaitsee chattibotin keskustelu kontekstin vaihtoelementti (kuva 14). Kontekstin vaihdolla tarkoitetaan siirtymistä yleisistä keskusteluaiheista esimerkiksi kurssikohtaisiin aiheisiin. Kyseinen toiminto on toteutettu keskustelulogiikan parantamiseksi. Elementin otsikko "Ask Help from Co-Teacher" on kuitenkin melko harhaanjohtava, sillä elementin avulla ei kysytä apua botilta, johon tässä viitataan Co-Teacher termillä. Parempi otsikko voisi olla esimerkiksi "Change discussion context". Myöskään kontekstin vaihtoa, tai muutakaan ohjeistusta elementin toimintoon ei anneta. Lisäksi vaikka opiskelija vaihtaa kontekstin ja painaa "Ask"-painiketta, mitään ei tapahdu sivulla. Tämä antaa helposti kuvan, ettei elementti toimi, ja saattaa johtaa hämmennykseen, turhautumiseen ja virheisiin. Elementin sijainti on myös varsin kummallinen. Ainoa vihje siihen, että elementti liittyy chattibottiin, on itse botin logo. Nielsenin [1995] mukaan käyttöliittymän tilan tulisi aina olla selkeä, sekä ohjeistuksen ja virheiden ehkäisyn riittävää. Käytettävyyden ja käyttöliittymän yksinkertaistamiseksi elementti tulisi sulattaa osaksi chattielementtiä, esimerkiksi sen oikeaan yläkulmaan. Valikon viereen voisi sijoittaa pienen kysymysmerkin, joka ohjeistaisi käyttäjää kontekstin vaihdossa. Tällöin myös "Ask"-

painikkeen voisi jättää pois, sillä kontekstin vaihto voisi tapahtua automaattisesti valinnan vaihdon jälkeen. Lopputuloksena olisi tilaa viemätön integroitu valikko, jonka yhteys keskustelun toimintaan kävisi nopeasti ilmi.



Kuva 14. Chattibotin kontekstin vaihto

2.6. Aktiviteetit

Sivun viimeinen elementti listaa yleisiä, sekä kurssisiin liittyviä, tiedotteita ja aktiviteetteja (kuva 15). Aktiviteettien listaus on selkeää, joskin mahdollisuus saada laajempaa tietoa kustakin tiedotteesta voisi olla hyödyllistä. Lisäksi "SEE EARLIER" linkin painamisesta ei tapahdu mitään. Tämä saattaa johtua siitä, että systeemillä ei ole näyttää aiempia tiedotteita. Jonkin näköinen ilmoitus tästä olisi kuitenkin hyödyllinen, jotta systeemin tila kävisi paremmin ilmi [Nielsen 1995].



Kuva 15. Tiedotteet ja aktiviteetit

3. Kurssikanvaasi

Luku valmistunut 13.05.2018

Navigaatiopalkin "Canvas"-pudotusvalikosta opiskelijat ja opettajat pääsevät "All courses"-sivulle, joka on lista avoimista kursseista (kuva 16). Kurssit on listattu selkeästi kahteen osaan sen mukaan, onko käyttäjä mukana kurssilla vai ei. Tosin tulevaisuudessa kurssien määrän kasvaessa sivu tarvitsee todennäköisesti laajemman kategorisoinnin ja hakutoiminnallisuuden. Opettajilla on myös "Canvas"-pudotusvalikossa painike "Add new courses", joka käydään luvun lopussa läpi.

My courses (3)

1. Test Course X (TCX): This course is made for test purposes only. It will be used to develop the service. [View details](#)
2. Excellence in Case Solving Skills (MET4HY201-2): Welcome to our course! We will test a new Co-Teacher learning platform created by Shoulders of Giants Oy Ltd (SOG) in 2017. The Co-Teacher's vision is to provi... [View details](#)
3. Applied Research and Development MET2HY201-18 (MET2HY201-18): The course is virtual and there are no contact hours after the introduction session in January. The course is online and on-demand. Artificial intelligent Co-Te... [View details](#)

Available courses (2)

1. Watson AI (WAT): Displaying features that are working in co-teacher as an example [Apply](#) [View details](#)
2. Ajanhallinta ja motivaatio oppimisen tukena 14.5.-31.7.2018 (AMOT): Tällä opintojaksolla opit tunnistamaan omaan hyvinvointiisi vaikuttavia tekijöitä, ja tutustut erityisesti hyvinvoinnin osa-alueisiin motivaatio ja&nb... [Apply](#) [View details](#)

Kuva 16. Lista avoimista kursseista

3.1. Kurssinäkömä

Painamalla "View details"-painiketta kurssilistassa, käyttäjä pääsee itse kurssinäkömään (kuva 17). Esimerkkinä käytetty kurssi on testikurssi, jota ei siis ole toteutettu missään oppilaitoksessa.

The screenshot shows a course page for 'WAT: Watson AI'. At the top, there is a navigation bar with 'Home', 'Canvas', 'Planner', 'Settings', 'Evaluation', and 'Feedback'. The user is logged in as 'Joonas Student Tontti'. The course title 'WAT: Watson AI' is displayed, along with a 'Back to list' button and an 'Apply for course' button. The course description is 'Displaying features that are working in co-teacher as an example'. The teacher is 'Rivo Gmail' (Email: rivo.seim@gmail.com, Role: Main teacher). The course is organized into weeks: Week 1 (20.05.18 - 26.05.18), Week 2 (27.05.18 - 02.06.18), and Week 3 (03.06.18 - 09.06.18). Week 2 is currently selected, showing 'Lecture times: Lectures missing', 'Practice times: Practices missing', and 'Tasks and deadlines: Tasks missing'. On the right, 'Upcoming deadlines' are listed: 1. 'NLU understanding' (Deadline: Tue 08:00 01:00 PM) and 2. 'Create a socket connection' (Deadline: Fri 27:04 04:00 PM). Below this, 'Activities' are listed, including 'Rivo Gmail added a new lecture for Watson AI' and 'Rivo Gmail added a new task for Watson AI' with timestamps. A 'SEE EARLIER' link is also present.

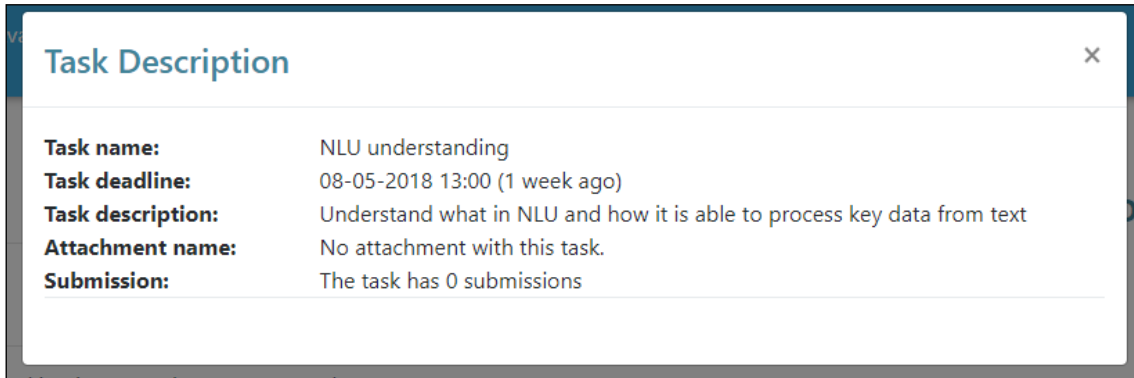
Kuva 17. Opiskelijan kurssinäkymä.

Sivun sisältö alkaa vasemmalta päin katsottuna "Back to list"- ja "Apply for course"-painikkeilla. Painikkeiden toiminnollisuudet ovat selkeät. Painikkeiden alapuolella on vaakaviivalla erotettuna kurssin otsikko. Otsikon alapuolelta löytyy kurssin kuvaus ja opettajien yhteystiedot. Yhteystiedot on erotettu kurssin materiaaliolosuhteesta vaakaviivalla, joka ei kuitenkaan ylety sivun vasempaan reunaan, toisin kuin ylemmät vaakaviivat. Sivun yhtenäisyyden parantamiseksi väliviivat tulisi olla yhtä pitkät, tai poistaa kokonaan. Jälkimmäinen vaihtoehto myös vähentäisi sivun visuaalista taakkaa.

Kurssitietojen alapuolelta löytyy kurssin materiaaliolosuhteet. Kurssin toteutus on jaettu viikkoihin vaaleansinisillä palkeilla. Palkkia painamalla viikkonäkymä laajenee, ja käyttäjä näkee kyseisen viikon luentojen sekä viikkoharjoitusryhmiensä ajankohdat. Tämän lisäksi näkymästä löytää viikon tehtävät ja niiden takarajat. Näkymä toimii loogisesti, joskaan useamman viikon näkymiä ei voi tarkkailla samanaikaisesti, sillä uuden palkin painaminen sulkee valmiiksi auki olevan viikon. Tämä tekee vaikeaksi viikkojen vertailun, mutta ehkäisee sivun venymisen pitkäksi.

Oikeassa yläkulmassa sijaitsee kurssin tulevat deadline't. Elementti on käytännössä sama kuin kohdassa 2.4. Myös samat korjausehdotukset pätevät tähän elementtiin, joskaan erillistä linkkiä kurssisivulle ei nyt tarvita. Kuvassa 18 näkyy tehtävän nimeä painettaessa avautuva modaali-ikkuna. Modaali-ikkunassa takarajojen kellonaika on ilmoitettu 24 tunnin formaatissa, kun taas muualla sivustolla ajat on ilmoitettu 12 tunnin formaatissa. Yhtenäisyyden parantamiseksi ajat tulisi ilmoittaa samassa formaatissa. Tognazzinin [2014] mukaan käyttöliittymän tulisi vastata käyttäjän odotuksia, joten aikaformaattina tulisi käyt-

tää 24 tunnin versiota, olettaen että käyttäjät sijaitsevat Suomessa. Puolestaan "Submission"-kohdassa oleva teksti "The task has 0 submissions" on hieman hämmentävä. Palvelun tilan parantamiseksi teksti tulisi korvata selkeämällä ohjeistuksella.



Kuva 18. Tehtävän nimeä painettaessa aukeava modaali-ikkuna.

Sivun oikeasta alalaidasta löytyy kurssin aktiviteetti- ja ilmoituselementti. Elementti vastaa kaikilta osin kohdan 2.6. elementtiä. Nyt kuitenkin "SEE EARLIER"-painike näyttää aikaisempia aktiviteetteja. Tällöin esiintyy uusi ongelma: aikaisempia ilmoituksia ei saa takaisin piiloon. Nielsenin [1995] mukaan käyttäjällä tulisi olla mahdollisuus peruuttaa tehty toiminto. Tähän ratkaisuksi "SEE EARLIER" teksti voisi muuttua "SHOW LESS" tekstiksi.

4. Asetukset

Luku valmistunut 14.05.2018

"Settings"-sivulla käyttäjä pystyy tarkastelemaan profiilitietoja, lisäämään vaihtoehdoisen sähköpostiosoitteen ja asettamaan itselle muistutuksia koskien kurssien sisältöä (kuva 19).

The image shows a user interface for settings. The top section is titled "Profile Settings" and contains two input fields: "Name" with the value "Joona Student Tontti" and "Email" with the value "joonatontti@...". Below these is a dropdown menu labeled "Others". The bottom section is titled "Reminder Settings" and is divided into four categories: "Event", "Lecture", "Practice", and "Task". Each category has a "Time" field and a "New reminder time" button. The "Practice" category also includes an "Enabled" checkbox.

Kuva 19. "Settings"-sivu

Sivu alkaa käyttäjän profiilitiedoilla. Nimi ja sähköposti ovat jälleen sijoitettu harmaaseen tekstikenttään, vaikka tietoja ei voi muuttaa. Systemin tilan parantamiseksi ja hämmennyksen estämiseksi, tiedot tulisi esittää ilman tekstikenttää.

"Others"-painiketta painamalla aukeaa sähköpostiasetukset, sekä salasanan vaihto (kuva 20). "Reset email"-osion toiminnollisuus ei tule mistään esille, ja "Go!" painiketta painamalla näytölle pomppaa huomautusikkuna tekstillä: "Email change/reset feature coming soon!" "Reset"-toiminto jää edelleen arvoitukseksi, mutta sähköpostin vaihtamisesta tulisi mainita osion otsikossa, esimerkiksi tekstillä "Reset or Change Email". Näin systemin tila tulisi paremmin esille. Toissijaisen sähköpostin asettaminen on toteutettu huomattavasti intuitiivisemmin. Ainoa turha sana käyttäjälle saattaa olla "Provider", jolla

tarkoitetaan sähköpostin tarjoajaa. Hämmennyksen estämiseksi ja avun tarjoamiseksi tekstin oikealle puolelle tulisi sijoittaa kysymysmerkki-ikoni, joka tarpeen tullen selventäisi käyttäjälle, mitä palveluntarjoajalla tarkoitetaan, ja miksi se on oleellinen tieto [Nielsen 1995]. Salasanan vaihto on toteutettu hyvin perinteisesti ja käyttäjien tottumuksiin nojautuvasti.

The image shows a web interface for email account settings. At the top, there is a dropdown menu labeled 'Others'. Below it, the 'Reset Email' section contains a label 'Email' followed by a greyed-out input field. Below that is a text input field containing the email address 'joonatontti@outlook.com'. A green 'Go!' button is positioned below the input field. The 'Secondary Emails' section follows, with a yellow warning banner stating 'Changing a verified email will require that you do the verification again.' Below the banner, there is a 'Provider' dropdown menu set to 'Google'. Underneath is another 'Email' label with a greyed-out input field, followed by a text input field containing 'joonatontti@hotmail.com'. A green banner below this field reads 'Your Google email is verified.' At the bottom of the section is a green 'Update' button.

Kuva 20. Sähköpostiasetukset

Profiilitietojen alapuolelta löytyy "Reminder Settings"-osio, josta käyttäjä voi asettaa itselle sähköpostimuistutuksia tapahtumista, luennoista, harjoitusryhmistä ja kurssitehtävistä. Muistutuksien luominen on melko intuitiivista ja joustavaa (kuva 21). Käyttäjä pystyy valitsemaan, tapahtuuko muistutus minuutteja, tunteja, päiviä jne. ennen tapahtumaa, ja numeraalisesti tarkan ajan. Kuvassa 21 on luotu yksi muistutus 6 päivää ennen tapahtumaa. Ainut hämmäntävä osio on viimeinen valinta, joka sisältää "Exact" ja "Until" sanat. "Until" todennäköisesti tarkoittaa, että muistutus tapahtuu, kun tapahtumaan on aikaa tasan asetettu aikajänne. "Exact"-valinnan toiminta taas jää käyttäjälle tuntemattomaksi. Lorangerin [2017] mukaan systeemin pitäisi käyttää käyttäjälle tuttuja termejä. Jos tilalle ei ole sopivampaa termiä, pitäisi nykyinen termi selittää paremmin. Tähän sopisi esimerkiksi kuvassa 21 näkyvä kysymysmerkki

ikoni. Käyttäjä voi luoda, poistaa, ottaa käyttöön, tai poistaa käytöstä muistutuksia tarpeen mukaan.

The screenshot shows a user interface for creating an event reminder. At the top, the word "Event" is displayed in blue. Below it is a table with two columns: "Time" and "Enabled | [checkbox]". The first row of the table contains "6 day(s) [help icon]", a disabled checkbox, and a red "X" icon. Below the table is a button labeled "+ New reminder time". Underneath that is a text input field containing "2|" and a vertical scrollbar. Below the input field are two dropdown menus labeled "Hours" and "Until". At the bottom of the form is a blue "Add" button.

Kuva 21. Muistutuksen luominen, "6 day(s)" alkava rivi on valmis muistutus.

5. Evaluointi

Luku valmistunut 15.05.2018

Navigointipalkin "Evaluation"-pudotusvalikon kautta käyttäjä pääsee tekstin arviointityökaluun (kuva 22). Työkalulla voi etsiä muun muassa avainsanoja ja konsepteja syötetystä tekstistä. Analysoitavan tekstin voi syöttää työkalulle kopiaamalla ja liittämällä, tai antamalla URL-osoitteen, josta teksti löytyy. Nielsen [1995] mukaan palvelun tulisi olla joustava ja sallia edistyneille käyttäjille sopivat toiminnot. Kyseiselle työkalulle ehdottoman tärkeä ominaisuus olisikin mahdollisuus liittää analysoitava teksti tiedostona. Tämä vähentäisi tarvetta kopiointille ja liittämiseksi, joka koetaan hitaaksi ja mahdollisesti epäammattimaiseksi.

Evaluate text

Evaluation tool helps you find out main concepts, keywords of provided texts. To get started, fill your texts or url in the text field below.

Text URL

Task - Test deadline about something

ship

is fake news for the romantic soul, offering passage into... amtime when blood and gold were the main currencies of civic life.The legend does seem, prima facie... untold riches, sailed by early-European explorers into a vast lake that once stretched over much of inland... aground, abandoned by its crew and covered over by centuries of sand and rock and creosote bush as that lake dried out...and now it lies a few feet below the surface, in sight of the chicken-wire fence at the back of the Desert Dunes motel, \$58 a night and HBO in most rooms.Totally insane, right? Let us slink back to our cubicles and never speak of the desert ship again. Let us only believe that which is shared with us on Facebook. Let us banish forever all traces of wonder from our lives. Yet there are believers who insist that, using recent advances in archaeology, the ship can be found. They point, for example, to a wooden sloop from the 1770s unearthed during excavations at the World Trade Center site in lower Manhattan, or the more than 40 ships, dating back perhaps 800 years, discovered in the Black Sea earlier this year.

Analyze

Evaluation result

Kuva 22. Evaluointityökalu tekstin analysointiin

Tekstityypin valinnan alapuolelta löytyy pudotusvalikko tekstillä "Use Keywords in texts". Pudotusvalikon toinen vaihtoehto on "Task - Test deadline about something" (kuva 23). Tekstit vaikuttavat huonolta englannilta, eivätkä välttämättä avaudu lukijalle. Onneksi apuun hyppää pop-up ohjelaatikko, kun hiiren vie valikon päälle. Sherwinin [2015] mukaan tämänlainen ratkaisu on hyvä idea, sillä se tulee apuun silloin, kun käyttäjä sitä tarvitsee. Valitettavasti ohjeen teksti jatkaa samaa teemaa vaikeaselkoisuuden kanssa, joten systeemin toiminnasta ei vieläkään saa tarkkaa kuvaa. Selvempi ohjeistus korjaisi tämän ongelman.

Evaluate text

Evaluation tool helps you find out main concepts, keywords of provided texts. To get started, fill your texts or url in the text field below.

Text URL

Use Keywords in texts

Comma Separated keywords you want to match with evaluation result e.g google, company

Analyze

Evaluation result

No result to display.

Kuva 23. Pudotusvalikon toinen vaihtoehto, pop-up ohjelaatikot ja tekstikentän täyttävä esimerkkiteksti

”Use Keywords in texts”-toiminnon tulisi etsiä käyttäjän antamia avainsanoja tekstistä. Käyttäjä antaa etsittävät avainsanat pudotusvalikon alla olevaan tekstikenttään. Kuvassa 23 nähtävän esimerkkitekstin analysointi ”ship”-avainsanalla ei kuitenkaan tuota toivottua tulosta, vaikka sana ”ship” esiintyy tekstissä useamman kerran (kuva 24). ”Keywords”-osiossa listataan palvelun löytämät avainsanat ja käyttäjän syöttämät avainsanat. Palvelun itse listaamat avainsanat löytyvät myös ylempänä nähtävän ”Keywords”-välilehden alta. Nielsenin [2002] mukaan käyttöliittymässä tulisi välttää saman tiedon esittämistä useaan kertaan, sillä se vaikeuttaa käyttöliittymän hahmottamista, ja vie huomiota muilta ominaisuuksilta. Olisikin huomattavasti yksinkertaisempi ratkaisu esittää systeemin ehdottamat avainsanat ylempänä välilehden yhteydessä, ja alemmassa osiossa käydä läpi ainoastaan käyttäjän syöttämät avainsanat. Valittavasti käyttäjän asettamien avainsanojen haku ei näytä toimivan, joten tämä toiminto tarvitsisi pikaista korjausta. Ennen kuin vika saadaan korjattua, käyttäjää tulisi ohjeistaa systeemin tilasta virheiden ja hämmennyksen ehkäisemiseksi.

Systeemin kartoittamat entiteetit, avainsanat, kategoriat ja konseptit ovat esitetty välilehtien avulla. Tämä on hyvä ratkaisu, sillä se säästää tilaa ja on intuitiivinen käyttää. Tulokset listataan pistemäärän eli ”Score”:n, tai asiaankuuluvuuden eli ”Relevance”:n mukaan. Teoriaa tai syitä kyseisiin pisteytyksiin ei anneta, joka saattaa herättää käyttäjässä kysymyksiä. Tekstissä aiemmin mainittu kysymysmerkki ikoni voisi toimia tässäkin, sillä sen avulla käyttäjä saisi nopeasti tietoa niin halutessaan. Annetut pisteet voivat myös tuntua käyttäjältä melko abstraktilta ja mitään sanomattomalta. Lorangerin [2017] mukaan systeemin pitäisi puhua käyttäjän kieltä, joten jonkin näköinen tulosten luokittelu voisi olla paikallaan tässä. Esimerkiksi, jos tietty konsepti nousee esiin useasti tekstissä, tämä voitaisiin ilmaista sanallisesti.

Evaluation result	
All Result ▾	
Entities	Keywords
Categories	Concepts
Label	Score
Anza-Borrego Desert	0.844906
Myrtle Botts	0.828277
Colorado Desert	0.57604
Albert S. Evans	0.508076
Canebrake Canyon	0.462965
Keywords matched result ▾	
Based on the keyword set for this evaluation, the keywords matches are the following:	
Keywords	
Evaluation Result	desert ship, rugged colorado desert, librarian myrtle botts, canebrake canyon, desert dunes motel
Expected Keywords	ship
Matched Keywords	
Missed Keywords	ship

Kuva 24. Esimerkkitekstin analysoinnin tulos

6. Palautesivu

Luku valmistunut 16.05.2018

Navigaatiopalkin "Feedback"-pudotusvalikosta käyttäjä pääsee palvelun palautteenanto-osioon, joka on tarkoitettu ensisijaisesti Co-Teacherin kehittämiseen (kuva 25). Pudotusvalikossa on kaksi vaihtoehtoa: "Evaluation" ja "General". Samat valinnat löydetään myös palautteenantosivun alkupäästä "Type"-otsikon vierestä. Käyttöliittymän yksinkertaistamiseksi ja duplikaattien poistamiseksi navigaatiopalkin pudotusvalikon voisi jättää pois. Sivun alkuosa ohjeistaa hyvin selkeästi, mitä edellä mainituilla termeillä tarkoitetaan. Vastaavanlaisia ohjeistusta kaivattaisiin palvelun monessa edellä läpikäytyissä kohdissa. Palautteenanto sivu mahdollistaa myös tiedoston lataamisen, joka olisi ollut kaivattu ominaisuus edellisen luvun analysointityökalun yhteydessä. Nyt käyttäjälle annetaan riittävä vapaus itsensä ilmaisuun, jonka puolesta myös Nielsen [1995] puhuu. Kaiken kaikkiaan sivun sisältö on helppo sisäistää ja toimii odotetun mukaisesti.

Kuva 25. Palautteenanto sivu.

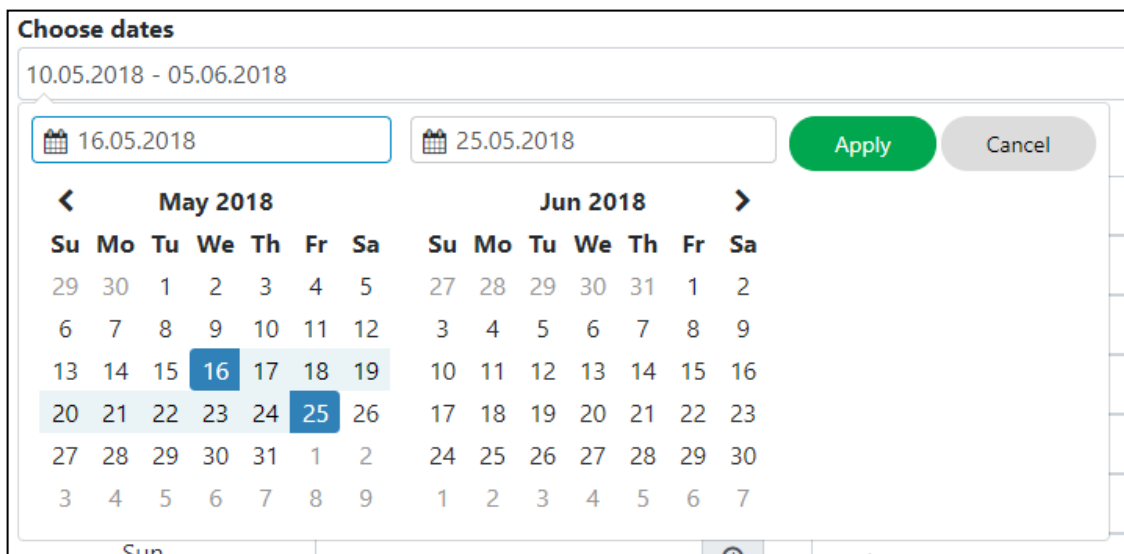
7. Aikataulun teko

Luku valmistunut 17.05.2018

”Planner”-sivu on oppilaille tarkoitettu aloitussivun lukujärjestyksen muokkaus ja personalisointi työkalu (kuva 26). Työkalulla voi esimerkiksi aikatauluttaa omia lukuhetkiä ja harrastuksia. Sivulla voisi olla lyhyt ohjeistus mihin tarkoitukseen työkalua käytetään, ja nimenomaan että tehdyt lisäykset näkyvät aloitussivun lukujärjestyksessä. Näin palvelun tilan hahmottaminen parantuisi ja virheiden, sekä hämmennyksen määrä vähenisi käyttäjillä.

Kuva 26. "Planner"-sivu, joka on tarkoitettu muun muassa itsenäisten opiskelusessioiden luomiseen.

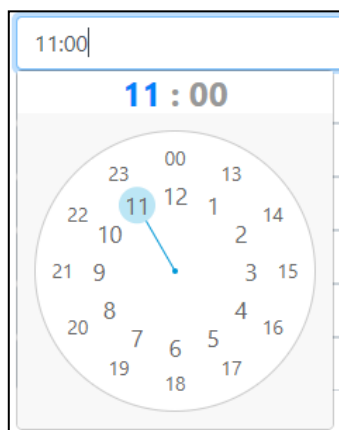
Käyttäjä aloittaa suunnitelmien tekemisen valitsemalla aikajakson jolle sessioita luodaan (kuva 27). Systemi valitsee aikajakson niin, että ensimmäisellä klikkauksella valitaan alkamispäivä, ja toisella klikkauksella loppumispäivä. Jos käyttäjä klikkaa vielä tämän jälkeen päivää, tästä päivästä tulee uusi alkamispäivä, ja valittu loppumispäivä katoaa. Tämä ratkaisu on melko epäintuitiivinen, sillä näkyvässä on kaksi kalenteria ja alkamis-, sekä loppumispäivälle omat kentät. Ratkaisu mahdollistaa hyvin paljon aikaavieviä virheitä, jos käyttäjä esimerkiksi asettaa aikajakson puolelle vuodelle, ja klikkaakin vahingossa väärän loppumispäivän, ja sitten oikean päivän, joka niin sanotusti alustaisi suunnitelman tekemisen. Huomattavasti varmempi tapa toteuttaa päivämääräen valitsemisen olisikin, että alkamispäivä valitaan vasemmasta kalenterista, ja loppumispäivä oikeanpuoleisesta kalenterista. Näin väärän päivämäärän valitsemisesta ei sakotettaisi käyttäjää. Lin [2017] julkaistuun artikkeliin on liitetty kuva samanlaisesta kaksikuukautisesta näkymästä, mutta oikeaoppisesti toteutettuna. Li toteaa myös, että huonosti toimiva päivämäärän valitsin saattaa tuottaa suurta turhautumista käyttäjille, joka taas saattaa johtaa palvelun hylkäämiseen.



Kuva 27. Aikajakson valitseminen.

Ajanjakson valittuaan käyttäjä täyttää haluamillensa päiville suunnitelmansa valitsemalla alkamis- ja loppumiskellonajat, sekä nimeämällä aktiviteetit. Kellonajan valitsemiseen käytetään Googlen ajanvalitsinta (kuva 28). Tämä toimii intuitiivisesti, joskin minuuttien valinnan pitäisi pystyä ohittamaan painamalla dialogi-ikkunan taustasta. Tätä huomattavasti suurempi ongelma kohdataan käyttäjän halutessa luoda samalle päivälle kaksi aktiviteettia. Tällä het-

kellä systeemi ei tue tätä mitenkään ja päädytään umpikujaan. Tämä rajoittaa huomattavasti käyttäjän vapautta ja huonontaa työkalun käyttökokemusta rajusti. Ongelman voisi ratkaista monella tapaa, mutta jos halutaan pysyä nykyisessä asettelussa, niin nimikentän oikealle puolella voisi olla plus merkillä varustettu painike, joka aktivoituu, kun kellonajat ja nimi on lisätty. Painiketta painamalla luotu suunnitelma hyppäisi lomakkeen loppuun, josta sitä pystyisi muokkaamaan tai halutessaan poistaa. Alkuperäiset kentät taas tyhjenisivät, ja samalle päivälle voisi luoda uuden tapahtuman. Whitenton [2016] ohjeistaa luopumaan lomakkeiden vihjeteksteistä ja "Clear form"-painikkeesta. Lomake vaikuttaa tosin melko helposti ymmärrettävältä, joten käytettävyydestit olisivat todennäköisesti paikallaan mietittäessä näitä muutoksia. Sivulla on myös runsaasti tilaa kirjoittaa viikonpäivien koko nimet, joten ymmärrettävyyden parantamiseksi niin tulisi tehdä.

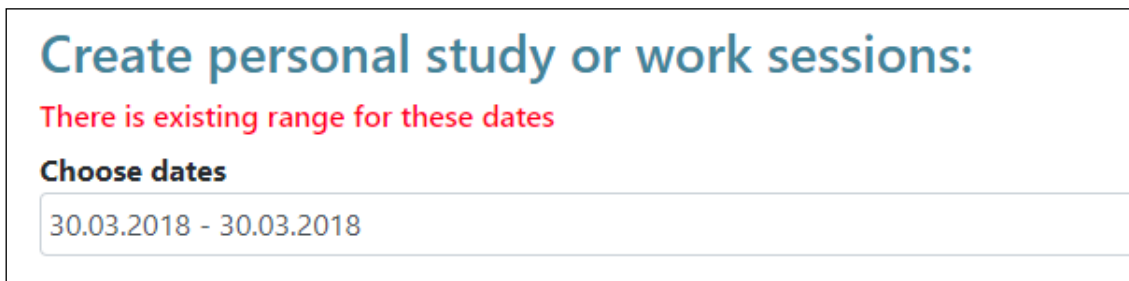


Kuva 28. Ajanvalitsin.

Kun suunnitelman on saanut valmiiksi ja käyttäjä painaa "Generate new sessions"-painiketta, luotu suunnitelma ilmestyy sivun alareunaan (ks. kuva 26). "Existing study and work sessions"-otsikko on hieman harhaanjohtava, sillä aktiviteetteja voi tehdä myös edellä mainitusti harrastuksista. Parempi otsikko voisi olla esimerkiksi: "Created plans:" Luodun suunnitelman nimi on aikaväli, jolle suunnitelma on luotu. Tämä on melko huono tapa tunnistaa suunnitelmia, varsinkin kun päivämäärät on kirjoitettu yhdysvaltalaisessa muodossa. Suunnitelma pitäisi pystyä nimeämään, ja nimeä vaihtamaan luomisen yhteydessä, sekä sen jälkeen. Tämän voisi toteuttaa tekstikentällä lomakkeessa, joka sijoitettaisiin aikavälin valinnan yläpuolelle. Luotuja suunnitelmia ei voi myöskään tarkastella tai muokata mitenkään. Tämä huonontaa palvelun käytettävyyttä, sillä se ei mahdollista virheiden korjaamista, ja pakottaa käyttäjän muistamaan suunnitelmien sisällöt. Tämä voitaisiin korjata

si "Edit"- painikkeella, jota painamalla suunnitelma hyppäisi yllä olevaan lomakkeeseen, josta sen tietoja pystyisi tarkastelemaan ja muokkaamaan.

Käyttäjä ei pysty tekemään kahta suunnitelmaa, jos niiden aikavälit asetuvat päällekkäin (kuva 29). Aktiviteettien ei edes tarvitse osua päällekkäin aikataulussa, jotta tämä virhe syntyy. Tämä rikkoo työkalun käytettävyyttä, ja saattaa aiheuttaa todella paljon aikaavieviä virheitä, sillä käyttäjän tekemä lomake tyhjenee ilman minkäänlaista varmistusta käyttäjältä, jos päällekkäin osuva suunnitelma luodaan. Nielsenin [1995] mukaan käyttäjällä pitäisi aina olla mahdollisuus estää tai kumota tällaisten virheiden tapahtumisen. Myös virheilmoituksen tekstin pitäisi puhua käyttäjän kieltä, sillä "There is existing range for these days" saattaa herättää ihmetystä käyttäjässä.



Kuva 29. Lomake tyhjenee ja systeemi antaa virheilmoituksen, jos käyttäjä luo kaksi päällekkäistä suunnitelmaa.

8. Hallinta

Luku valmistunut 17.05.2018

Ainoastaan opettajalta löytyvä päänavigoinnin pudotusvalikko "Manage" sisältää painikkeet "Venues" ja "Conversations". Sivut on tarkoitettu Co-Teacherin hallinnointiin. "Venues"-sivulla opettaja pystyy luomaan oppilaitoksen opetustiloja, joita taas voidaan käyttää kurssien luomiseen. "Conversations"-sivulla opettaja voi käydä läpi chattibotin keskusteluja ja lähettää korjausehdotuksia eteenpäin palvelun kehittäjille.

8.1. Opetustilojen luominen

"Venues"-valinnasta opettajalle aukeava opetustilojen luomissivun käyttöliittymä on melko yksinkertainen, mutta ohjeistus olisi jälleen paikallaan (kuva 30). Esimerkiksi lyhyt teksti kuvaamaan, mitä "venue"-sanalla tarkoitetaan, helpottaisi sivun ymmärtämistä entisestään. Itse tilan luomiseen käytettävä lomake on hieman oudosti aseteltuna. Yhtenäisyyden parantamiseksi tekstikenttien nimikkeet tulisi olla joko kummassakin kentän päällä tai sivussa. Palvelun

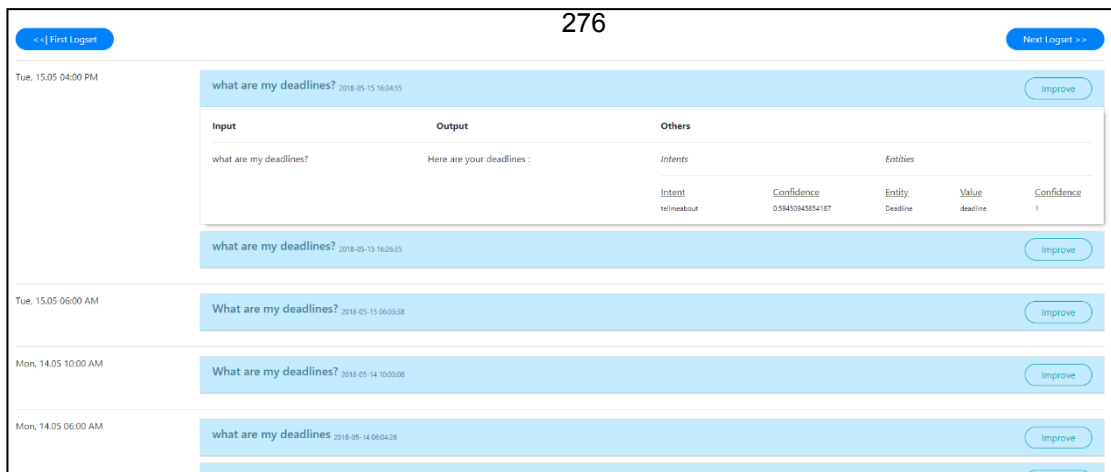
painikkeet sijaitsevat myös yleensä oikealla puolella elementtiä, joten yhtenäisyyden vuoksi tämänkin sivun "Add"-painike voisi sijaita lomakkeen oikeassa alakulmassa. "Address"-tekstikenttä vaikuttaa tarpeettoman isolta ja tilaa vievältä. Todennäköisesti matalampi tekstikenttä sopisi tähän kohtaan hyvin. "Address"-nimike on myös sijoitettu tekstikentän alareunaan, vaikka käyttäjät ovat tottuneet siihen, että nimikkeet sijaitsevat kenttien vasemmassa yläkulmassa.

Kuva 30. Opetustilojen luominen.

Lomakkeen oikealla puolelle on listattu jo ennestään luodut opetustilat. Tilat ovat "Venue management"-otsikon alla, joka on hieman harhaanjohtava, sillä opetustiloja ei voi muokata listassa. Virheiden korjaamisen ja käyttäjän vapauden laajentamiseksi olisikin toivottavaa, että luotujen tilojen osoitetta voisi muokata, tai niitä voisi poistaa.

8.2. Chatti keskustelujen selaus ja parantaminen

"Conversations"-sivulla opettaja pääsee tarkastelemaan käyttäjien ja chattibotin välisiä keskusteluja, jotka ovat anonymisoitu (kuva 31). Sivun ylälaudassa sijaitsevat painikkeet "First Logset" ja "Next Logset". Vasemmasta painikkeesta sivu päivittyy, mutta näennäisesti sivulla ei tapahdu mitään. Oikeanpuolista painiketta painamalla sivu tyhjenee. Jos painikkeet eivät toimi, niitä ei pitäisi näyttää käyttäjälle, sillä rikkiinäisinä ne vain aiheuttavat virheitä ja hämmennystä. Painikkeiden nimet ovat myös hieman epäloogiset ja "First Logset"-painikkeella tarkoitetaankin todennäköisesti "Previous Logset"-painiketta.



Kuva 31. Keskustelujen selaus.

Itse keskustelut on listattu ilmeisesti päivämäärän ja kellonajan mukaan, niin, että yhden tunnin sisällä lähetetyt viestit näkyvät yhdessä vaaka-viivoilla erotettuina. Viestit on myös saatettu jakaa käyttäjien mukaan niin, että yksi keskustelu listataan aina kahden vaakaviivan sisään. Tilan hahmottamisen parantamiseksi tämä jaottelu tulisi tehdä selkeästi. Todennäköisesti loogisin tapa järjestellä keskusteluja olisi listata aina yksi keskustelu yhden otsikon alle, jotta keskustelun kulkua olisi helppo seurata. Nyt jokainen viesti näkyy erillään sinisessä laatikossa.

Viestin tekstiä painamalla laatikko laajenee, ja käyttäjä pystyy tarkastelemaan itse viestiä "Input"-otsikon alta, botin vastausta "Output"-otsikon alta, ja muita dialogin logiikkaan liittyviä osia "Others"-otsikon alta. Käyttöliittymän tulisi puhua käyttäjän kieltä, joten sanaa kuten "Input" tulisi välttää. Tässä paremmin termiksi sopisi "User's message" tai "Student's message". Opettaja ei todennäköisesti myöskään ymmärrä tai tee mitään "Others"-osion tiedoilla, joten niitä ei tarvitsisi näyttää. Ne poistamalla käyttöliittymän ymmärrettävyys helpottuisi, käytettävyys nopeutuisi ja käyttökokemus paranisi. Sinisen palkin oikeassa reunassa olevaa "Improve"-painiketta painamalla käyttäjälle aukeaa kuvan 12 ikkuna, johon opettaja voi jättää korjausehdotuksensa. Korjausehdotus menee Co-Teacherin kehittäjille, jotka pystyvät tekemään muokkauksia botin keskustelulogiikkaan.

9. Yhteenveto

Co-Teacher on kehitystiensä alkuvaiheilla, ja sen huomaa varsinkin käyttöliittymästä. Monet käyttöliittymän ratkaisut eroavat suuresti Nielsen Norman Group:in ja muun lähdemateriaalin suosituksista. Valitettavasti tämä vaikuttaa negatiivisesti käytettävyyteen ja yleiseen käyttökokemukseen. Onneksi melkein jokaiseen ongelmakohtaan löytyi melko yksinkertainen ratkaisu, jotka toteuttamalla palvelun käyttökokemus paranee todennäköisesti suuresti. Opiskelijoiden "Planner"-sivun toiminnassa esiintyi suurimpia ja kaikista kriittisimpiä

ongelmakohtia, joten sen korjaaminen tulisi priorisoida ensimmäiseksi. Myös lähes olemattomaan chattibotin logiikkaan tulisi saada laajuutta, jotta työkalu olisi hyödyllinen. Tässä tutkielmassa ei paneuduttu käyttöliittymän responsiivisuuteen, joka on nykypäivänä hyvin tärkeä osa palvelun kokonaisuutta. Täten myös siihen tulisi paneutua kehityksessä.

Viiteluettelo

- Jakob Nielsen. 2007. Breadcrumb navigation increasingly useful. <https://www.nngroup.com/articles/breadcrumb-navigation-useful/>. Checked 27.03.2018.
- Kathryn Whitenton. 2013. Minimize cognitive load to maximize usability. <https://www.nngroup.com/articles/minimize-cognitive-load/>. Checked 29.03.2018.
- Jakob Nielsen. 2002. Reduce redundancy: Decrease duplicated design decisions. <https://www.nngroup.com/articles/reduce-redundancydecrease-duplicated-design-decisions/>. Checked 29.03.2018.
- Raluca Budiu. 2014. Memory recognition and recall in user interfaces. <https://www.nngroup.com/articles/recognition-and-recall/>. Checked 02.04.2018.
- Bruce Tognazzini. 2014. First principles of interaction design (Revised & Expanded). <http://asktog.com/atc/principles-of-interaction-design/>. Checked 13.05.2018.
- Jacob Nielsen. 1995. 10 Usability heuristics for user interface design. <https://www.nngroup.com/articles/ten-usability-heuristics/>. Checked 17.05.2018.
- Hoa Loranger. 2015. Simplicity wins over abundance of choice. <https://www.nngroup.com/articles/simplicity-vs-choice/>. Checked 17.05.2018.
- Kim Flaherty. 2015. Indicators, validations, and notifications: Pick the correct communication option. <https://www.nngroup.com/articles/indicators-validations-notifications/>. Checked 01.04.2018.
- Kathryn Whitenton. 2014. 3 guidelines for search engine "No Results" pages. <https://www.nngroup.com/articles/search-no-results-serp/>. Checked 03.04.2018.

Aurora Harley. 2014. Icon usability. <https://www.nngroup.com/articles/icon-usability/>. Checked 03.04.2018.

Hoa Loranger. 2017. Plain language is for everyone, even experts. <https://www.nngroup.com/articles/plain-language-experts/>. Checked 17.05.2018.

Katie Sherwin. 2015. Pop-ups and adaptive help get a refresh. <https://www.nngroup.com/articles/pop-up-adaptive-help/>. Checked 16.05.2018.

Teknologinen singulariteetti - tulevaisuuden uhka tai mahdollisuus

Sami Tuomaala

Tiivistelmä.

Tässä tutkielmassa käsitellään teknologian ja tekoälyn nopeasti kiihtyvää kehitystä, minkä ennustetaan johtavan ajanhetkeen, jolloin tekoälystä tulee ihmistä älykkäämpää. Ajanhetkestä tai kokonaisesta aikakaudesta käytetään yleisesti nimitystä teknologinen singulariteetti. Tutkielmassa tarkastellaan teollisten vallankumousten vaikutuksia nopeasti teknologistuvan maailman muokkaamisessa ja lopulta singulariteetin käsitettä sekä sen merkitystä ja mahdollisuuksia lähitulevaisuudessa. Teknologinen singulariteetti tulisi todennäköisesti vaikuttamaan ihmiskuntaan radikaalisti ja tästä syystä tulevaisuutta pyritään jatkuvasti ennustamaan mahdollisimman tarkasti monesta eri näkökulmasta, jotta voisimme varautua kehittyvän tekoälyn muokkaamaan maailmaan. Lisäksi pohditaan, onko ihmismäisen tekoälyn tavoittelu lopultakaan järkevää, eli kannattaako sellaista tieteen tahtoen havitella vai pitäisikö tekoälyn kehittymistä pyrkiä hidastamaan tai jopa estämään. Loppujen lopuksi singulariteetissa kyse on ajankohdasta, mitä emme voi täydellä varmuudella ennustaa, joten kaikki teoreettiset mahdollisuudet on pidettävä avoinna.

Avainsanat ja -sanonnat: Teknologia, tekoäly, singulariteetti, tulevaisuus, maailmanloppu.

1. Johdanto

Tässä tutkielmassa tarkoituksena on tuoda esille teknologian ja tekoälyn nopea ja pitkäänjatkunut kiihtyvä kehitys sekä teknologisen singulariteetin käsitteellinen ymmärrys ja siitä mahdollisesti koituvat tulevaisuuden kuvat. Teknologisella singulariteetilla viitataan sellaiseen tulevaisuuden ajankohtaan, jolloin teknologian kehityskäyrä kasvaa niin nopeaksi, ettei ihminen kykene hallitsemaan tai ymmärtämään sitä [Ford 2017]. Toisin sanoen teknologian kehittyminen johtaa tilanteeseen, jossa tekoälystä tulee itsessään niin älykäs, että se kykenee itsenäiseen toimimiseen ilman ihmisten osallistumista.

Koska aihe on kauttaaltaan futuristinen ja spekulatiivinen, siitä ei ole antaa tarkkaa tietoa. Siitä on kuitenkin lausuttu vuosien saatossa paljon varsin päteviä ennustuksia, joita on tarkasteltava mahdollisina skenaarioina edes osittain, vaikka useat ennustuksista kompastelevatkin yhä ristiriitojen ja puutteellisen informaation kanssa.

Robottien vallankumoukset liittyvät vielä pitkälti tieteiselokuviin. Robottien ja muiden tekoälyä hyödyntävien teknologisten laitteiden vaikutus näkyy jo meidän elämässämme, joten emme voi sulkea pois sitä mahdollisuutta, että jotain maailmaamme mullistavaa tulee tapahtumaan ehkä jo lähivuosien aikana. Aihe on hyvin ajankohtainen nopeasti kehittyvässä maailmassa, jota voisi kutsua myös älyaikakaudeksi, sillä teknologian voidaan nähdä hallitsevan jo nyt ihmisiä ainakin joiltain osin psykologisesti. Psykologisella hallinnalla tarkoitan laajakäsitteisesti muun muassa teknologiaan kohdistuvaa riippuvuutta ja sen käyttämiseen liittyvää jokapäiväistä tarvetta. Voidaan pitää mahdollisuutena, että teknologia ja älylaitteet saattavat jo lähitulevaisuudessa kyetä hallitsemaan elämäämme myös konkreettisesti, elleimme kiinnitä huomiotamme teknologian nopeaan kehitykseen.

Tarkastelen tässä työssä laajasti teknologian kehittymistä ja teknologisen singulariteetin käsitettä sekä alan merkittävimpiä teoreetikkoja. Tutkielman luvussa 2 tarkastellaan teknologisen kehityksen historiaa ja teollisten vallankumousten vaikutusta nykypäivän kehitykseen. Luvussa 3 käsitellään tekoälyä itsessään. Tuon esille lyhyesti tekoälyn kehittymisen historiaa viimeisten vuosikymmenten aikana ja kuvauksen siitä, mitä tekoäly todellisuudessa on. Luvussa 4 esittelen teknologisen singulariteetin käsitteen, mikä on tämän tutkielman pääteemanakin. Käsitteen lisäksi esittelen muutamia henkilöitä, jotka ovat tutkineet tai muuten olleet osallisina teknologisen singulariteettia käsittelevissä ennustuksissa. Luku 5 käsittelee Raymond Kurzweilia, joka on teknologisen singulariteetin tunnetuimpia puolestapuhujia. Tarkastelen Kurzweilin merkitystä ja saavutuksia tekoälyn tutkijana. Lopuksi luvussa 6 kokoan ajatukseni yhteenvetoon, sekä pohdin mahdollisia lähitulevaisuuden muutoksia tekoälyllisemmässä maailmassa.

2. Teknologian kehityksen historiaa

Tutkielman aluksi on syytä tarkastella lyhyesti teknologian historiaa ja sen kehitysvaiheita tähän päivään, jotta on helpompi muodostaa kokonaiskuva teknologian merkityksestä nykypäivänä ja tulevaisuudessa. On syytä myös muistaa teknologia-käsitteen monimerkityksellisyys. Yleensä teknologialla viitataan tietokoneisiin tai muuhun sähköä hyödyntäviin elektronisiin koneisiin tai järjestelmiin. Teknologialla kuitenkin itsessään viitataan yleisesti hyvin laajasti erilaisten tekniikoiden, taitojen ja tietojen hyödyntämiseen elämää helpottavana toimintana [Marttinen 2018]. Tämän tutkielman pääpaino on kuitenkin tietotekniikan teknologinen kehitys ja hyödyntäminen tekoälyn kehittämisessä.

2.1. Ensimmäiset teolliset vallankumoukset

Ihminen on koko historiansa ajan keksinyt uusia tekniikoita helpottaakseen elämäänsä. Pieniä keksintöjä, kuten kiila tai pyörä, on keksitty jo luolamiesten aikoina työkaluina tai -koneina. Ihmiskunnan teknologinen harppaus tapahtui kuitenkin vasta vuonna 1785 ensimmäisen rakennetun mekaanisen kutomakoneen myötä. Ensimmäinen teollinen vallankumous tapahtui 1700- ja 1800-luvun vaihteessa Isossa-Britanniassa, jolloin koneellistettu tehdastyö sai alkunsa. Töitä pystyttiin ensimmäistä kertaa helpottamaan ja nopeuttamaan suuressa mittakaavassa mekaanisten koneiden avulla, mistä syntyikin termi ”mekanisaatio”. Koneistettu työvoima toi kuitenkin myös huolenaihetta työntekijöiden keskuudessa, sillä koneiden pelättiin vievän ihmisten työt ja tulot, mikä sai niin kutsutut luddiitit liikkeelle. [Marttinen 2018.] Luddiitit vastustivat vahvasti teknologista kehitystä ja pyrkivät sabotoimaan koneistettua teollisuutta [Wikipedia 2018a]. Samanlaisia kauhukuvia robotti työvoiman suhteen esiintyy edelleenkin ja yhä suuremmassa mittakaavassa ja tätä aihetta käsitelläänkin tarkemmin tutkielman luvussa 3.

Toinen teollinen vallankumous alkoi 1800-luvun loppupuoliskolla ja tällöin siirryttiin askel lähemmäksi nykypäiväistä elektroniikkatekniikkaa hyödyntävää teknologiaa. Luultavasti tärkeimpinä ja mullistavimpina keksintöinä tulivat sähkö, radiot ja polttomoottorit. Vuonna 1870 otettiin käyttöön ensimmäinen liukuhihnatuotanto Yhdysvaltalaisessa teurastamossa. Rullakuljettimia oltiin käytetty jo aiemminkin, mutta vasta nyt liukuhihnojen koko potentiaali valjastettiin tuotantokehitykseen. Tämä oli samalla nykypäiväisen massatuotannon ja liukuhihnatyöskentelyn alkusoittoa, joskin sen tehokkuudesta ja tuottavuudesta kunnian vei vasta 1913 yhdysvaltalainen keksijä Henry Ford, joka tosin myönsi saaneensa idean lihanpakkauslinjojen malleista. Ford kuitenkin jalosti idean uudelle tasolle kehittäessään linjaston T-Fordien nopeaan valmistukseen. [Marttinen 2018.] Myöhemmin tämä innovaatio sai aikaan oman terminsä, fordismi, jota käytetään yleisesti viitattaessa massatuotantolinjastoihin [Wikipedia 2018b]. Jälleen kerran oli ihan aiheellistakin pohtia tulevaisuutta työntekijöiden näkökulmasta, sillä liukuhihnatyöskentely helpotti työntekoa, eikä työntekijöitä vaadittu enää yhtä paljon kuin ennen.

2.2. Tietotekninen kehitys käynnistyy

Teollisuuden kolmannen vallankumouksen myötä syntyi nykypäivänakin käsitetty teollisuusautomaatio. Elektroniikan ja tietotekniikan kehitys oli viimein tullut pisteeseen, jolloin pystyttiin ensimmäistä kertaa ainakin teoriassa puhumaan tekoälystä. Ensimmäiset tietokoneet kehitettiin 1940-luvulla ja niiden sarjatuotanto alkoi vuonna 1951. Vuosisadan lopussa tietokoneita oli

maailmassa noin 3000, mutta ne olivat vielä hyvin epäkäytännöllisiä ja raskaita, eikä niiden potentiaalia pystytty vielä suuremmin hyödyntämään. Vuonna 1969 markkinoille kuitenkin tuotiin ensimmäinen ohjelmoitava logiikka eli PLC (Programmable Logic Controller), mitä voidaan pitää tarkkana ajankohtana kolmannen vallankumouksen alkamiselle. [Marttinen 2018.]

Ohjelmoitavat logiikat eli lyhyemmin logiikat mahdollistivat ensimmäistä kertaa suuremman teollisen ohjausjärjestelmän hallinnoimisen pienellä alustalla, eikä se vaatinut samanlaista satojen tai tuhansien johtojen ja kytkimien säätelyä kuin aiemmin. Ensimmäinen ohjelmoitava logiikka, Modicon 084, otettiin käyttöön autoteollisuudessa Yhdysvalloissa ja sen yksinkertaisuutensa ja toiminnallisuutensa vuoksi se palveli tauotta pitkälti toistakymmentä vuotta. Logiikoiden kehittäminen jatkui valmistuksen jälkeen nopeasti ja 1970-luvulla alettiinkin jo kehittää uusia kenttäväyliä logiikoiden väliseen kommunikointiin. [Marttinen 2018.] Nopea kehitystyö edisti teollisuutta, mutta toi samalla uusia haasteita, sillä uuden teknologian oppiminen vei oman aikansa ja etenkin kehityksen alkuvaiheissa teknologia muuttui nopeasti, eikä pysyviä standardeja ollut tarjolla.

Logiikoiden lisäksi 1960-luvulla alkoi myös teollisuusrobottien kehitys, kun ensimmäinen robotti asennettiin General Motorsin tehtaalle vuonna 1961 palvelemaan painevalukonetta [Marttinen 2018]. Tämän johdosta voidaankin sanoa, että autoteollisuus on ollut myös robotiikan edelläkävijä. Robottitekniikka ja tekoäly ovat nykypäivänäkin hyvin esillä autoteollisuudessa erilaisten automaattiohjausten myötä sekä vieläkin näkyvämmän muun muassa Googlen projektissa kehittää itseohjautuvia autoja, joiden tarkoituksena olisi selvitä liikenteessä täysin ilman ihmiskuljettajaa.

Vuonna 1969 Standfordin yliopistossa kehitettiin myös ihmisen kättä matkiva robottikäsi, joka tunnetaan nimellä "Standford arm". Standford armin kehittäjä, Victor Scheinman, kehitti kolme vuotta myöhemmin robottikädestä parannellun version, "MIT armin", jonka hän toteutti siirryttyään Massachusettsin teknilliseen yliopistoon MIT:hen. Tämän jälkeen robotit levisivät nopeasti ja vuonna 1990 maailmassa oli jo yli 400 000 teollisuusrobottia. [Marttinen 2018.]

Tietotekniikan läpimurtona voidaan pitää vuotta 1972, jolloin Intel toi markkinoille ensimmäiset 4-bittiset mikroprosessorinsa, mikä mahdollisti nykypäiväisen automaation, joka perustui digitaaliseen tietotekniikkaan. Tekniikan kehittyessä myös tietoliikennetekniikka kehittyi nopeasti erityisesti 1990-luvulla, jolloin syntyi muun muassa langaton dataliikenne. [Marttinen 2018.] Internetistä tuli osa arkista elämää ja tietotekninen tiedonsiirto oli nopeampaa ja helpompaa, kuin koskaan aiemmin, mutta kehitys jatkui edelleen kiihtyvässä tahdissa.

2.3. Teollisuus 4.0

Neljäs ja tähänastisista teollisista vallankumouksista viimeisin tunnetaan termillä ”teollisuus 4.0”. Toisin, kuin aiempien teollisten vallankumouksien kohdalla, neljänteen vallankumoukseen ei tunnu olevan edeltävänkaltaisia merkittäviä käännekohtia teknologian tai teollisuuden kehityksessä, vaan se on vain päätetty aloittaa. Termi on tullut esille ensimmäisen kerran Hannover Messe -teknologiatapahtumassa Saksassa 2011. [Marttinen 2018.]

Aiemmat teolliset vallankumoukset ovat tuoneet mukanaan aina jotain teknologista kehitystä mullistavia uusia keksintöjä, mutta neljäs vallankumous pohjautuu pitkälti aiempaan digitaaliseen vallankumoukseen. Uudet innovaatiot eivät olekaan enää pelkästään fyysisiä laitteita, vaan yhä enemmissä määrin aineettomia keksintöjä ja termejä, kuten big data, pilvipalvelut sekä tekoäly. Neljännen vallankumouksen pyrkimyksenä on siis ennemminkin hyödyntää kokonaisvaltaisesti aiemmin keksittyjä innovaatioita, eli ottaa teknologiasta niin sanotusti kaikki hyöty irti. [Marttinen 2018.]

Yhdysvaltalainen journalisti, Daniel Oberhaus, on kuvaillut neljännen teollisen vallankumouksen jälkeistä arkea seuraavasti:

”Kello on 7 aamulla, huhtikuun 13. päivänä vuonna 2025. Älykellosi herättää sinut ja noustessasi ylös sängystä talosi herää eloon. Kylpyhuoneen valot syttyvät itsestään ja suihku alkaa lämmittää vettä. Peseydyttyäsi puet yllesi t-paidan, joka on tarkoin mitoitettu vartalollesi. Tarkistat puhelintasi vain huomataksesi, että akku on tullut tiensä päähän. Ponnahdusilmoitus kuitenkin ilmoittaa sinulle, ettei ole syytä huoleen – korvaava akku on jo matkalla luoksesi. Kuulet auton käynnistyvän autotallissasi valmiina viemään sinut johtamallesi tehtaalle, missä puhelimesi mukaan yhdessä laitteessa on häiriö. On hyvin harvinaista, että sinun täytyy todella mennä työpaikalle, sillä yleensä tehdas kykenee selviämään itsenäisesti. Mutiset itseksesi, miten mikään ei enää toimi samalla, kun autosi ajaa itsensä ulos autotallistasi.”¹ [Oberhaus 2015.]

Teollisuus 4.0:n tavoitteena oli luoda yrityksille verkko, joka kattaisi koneet, varastot ja tuotannon niin kutsutuilla kyberfyysisillä järjestelmillä, jotka

¹ ”It's 7 AM, April 13th, 2025. Your smart clock rouses you from a dreamless sleep and you climb from bed as your house comes to life. The bathroom light turns itself on and the shower begins to heat its water. After washing off, you throw on a T-shirt which has been perfectly fit for your body. You check your phone, only to discover that its battery is about to give out. A push notification informs you that there is no need to worry – a replacement is already on its way. You hear your car start in the garage, ready to take you down to the factory you manage, where, according to your phone, one of the machines has malfunctioned. Actually traveling to your place of work has become an increasingly rare phenomenon for you – most of the time the factory takes care of itself. "What a pain in the ass," you mutter as your car pulls itself out from your garage. "Can't these things get anything right, anymore?"”

mahdollistaisivat älykkäiden koneiden kommunikoinnin toisten koneiden ja ihmisten kanssa [Marttinen 2018]. Tieto kulkeutuisi siis nopeasti tehtaiden sisällä, kuin myös ulkopuolella ja olisi myös ihmisten saatavissa ilman suurempaa panosta. Älykkäiden koneiden aikakausi olisi siis nyt teoreettisesti saavutettavissa tai ainakin sen puolesta pyritään tekemään tutkimustyötä ja kehitystä. Älykkäiden laitteiden ja järjestelmien kehittymisen myötä on alettu puhua enemmän tekoälystä, eli ihmismäistä ajattelua omaavista koneista.

Neljännän teollisen vallankumouksen keulakuvina ovat olleet jo aiemmin mainitut robottiautot, joilla on kyky havainnoida ympäristöään muun muassa laserilla, GPS:llä ja konenäöllä ja ovat kykeneviä liikkumaan itsenäisesti. Ensimmäiset täysin itsenäiset robottiautot syntyivät jo 1980-luvulla, ja 1990-luvulla itseohjaavat autot kykenivät liikkumaan jo muun liikenteen seassa. Aihe on tullut enemmän esille kuitenkin vasta myöhemmin, kun esimerkiksi vuonna 2012 tuli Nevadan osavaltiossa Yhdysvalloissa laki itseohjaavien autojen käytöstä. [Marttinen 2018.]

Neljäs teollinen vallankumous on tuonut mukanaan myös uutta teknologiaa, mistä yhtenä innovatiivisimpana voidaan pitää 3D-tulostusta, mikä mahdollistaa fyysisten elementtien valmistamisen mallista. 3D-tulostuksen on arveltu mullistavan tulevaisuudessa tuotteiden valmistuksen, kuten liukuhihnojen keksiminen 1900-luvun alussa. Tämäkään ei kuitenkaan ole täysin uutta teknologiaa, sillä 3D-tulostimia on ollut käytössä jo 1980-luvulla, mutta myöhempi kehitys ja hintojen laskeminen kymmenistä tuhansista muutamiin satasiin on mahdollistanut niiden käyttöönoton myös tavallisten yksityishenkilöiden keskuudessa. [Marttinen 2018.]

2.4. Mooren laki

Teolliset vallankumoukset ovat siivittäneet teknologian kehittymistä viimeisten kahden tai kolmen vuosisadan aikana melko nopeasti ja tietotekniikkaan pohjautuva teknologia on vielä suhteellisen uutta. Silti sen kehittyminen on ollut räjähdysmäistä ja kehittyminen jatkuu yhä edelleen erittäin nopeaan tahtiin. Tietokonetekniikan kehittymisen kiihtyvää nopeutta kuvataan usein Mooren lailla, jonka nyrkkisäännön mukaan tietokoneiden laskentateho kaksinkertaistuu 18–24 kuukauden välein [Ford 2017]. Tietokoneiden kehitys kiihtyy siis jatkuvasti eksponentiaalisesti, joten seuraukset ovat valtavia, vaikka niitä onkin usein vaikea ymmärtää tai sisäistää.

Martin Ford [2017] antoi hyvän ajatusleikin, minkä avulla tietotekniikan kehittymisen nopeutta on ehkä hieman helpompi ymmärtää. Voit kuvitella ajavasi autoa kahdeksan kilometrin tuntinopeudella minuutin, jonka jälkeen kiihdytät nopeutesi kaksinkertaiseksi, eli nopeutesi olisi nyt 16 kilometriä tunnissa. Ajat taas minuutin, jonka jälkeen jälleen kaksinkertaistat nopeutesi ja

niin edelleen. Nopeutesi kasvaessa näin eksponentiaalisesti, myös kuljettu matka kasvaa nopeasti. Ensimmäisen minuutin aikana ajaisit noin 134 metriä, mutta jo viidennellä minuutilla kulkisit 128 kilometrin tuntinopeudella yli kahden kilometrin matkan. Mikäli jatkaisit samalla kaavalla kaksinkertaistaen nopeuttasi 27 kertaa, nopeutesi viimeisellä minuutilla olisi 1080 miljoonaa kilometriä tunnissa ja kuljettu matka pelkästään viimeisen minuutin aikana olisi lähes 18 miljoonaa kilometriä. Ensimmäinen mikropiiri keksittiin vuonna 1958, joten tietokoneiden laskentateho on kaksinkertaistunut tähän päivään mennessä suurin piirtein yhtä paljon, kuin edeltävässä ajatusleikissä. Tämä kuvastaa siis samalla karkeasti tietotekniikan nykytilaa verraten 1950-luvun loppuun, jolloin alettiin kehittää ensimmäisiä alkeellisia mikropiirejä. [Ford 2017.]

Merkityksellistä tietotekniikan kehittämisessä ja sen aikaansaamassa muutoksessa nykypäivänä ei ole pelkästään eksponentiaalisesti kiihtyvä kehitys, vaan se, että sitä on jatkunut niin pitkään. Ei ole täysin selvää, kuinka pitkään tällaisen kehitysnopeuden ylläpitäminen on mahdollista, mutta ainakaan vielä se ei ole hidastunut.

3. Tekoälykkäitä tietokoneita ja ohjelmoituja avustajia

Teknologiset älylaitteet kulkevat nykyaikana mukanaamme joka paikassa ja olemme jatkuvasti kytkettyinä ja yhteyksissä internetin välityksellä ympäri maailmaa [Haikkonen 2017]. Tämä johtaa helposti ongelmatilanteisiin, kun esimerkiksi oletamme yhteydenpidon olevan aina nopeaa ja luotamme siihen, että toisen henkilön saa aina kiinni missä tahansa hän liikkuukin. Saavutettavuus lisää ongelmia myös esimerkiksi työelämässä, jossa vapaa-ajallakin oletetaan ihmisten olevan tarvittaessa tavoitettavissa. Yksityiselämässä älypuhelimien rikkoutuminen tai puhelimen pitäminen äänettömällä saattavat aiheuttaa lähipiirissä välittömästi huolta ja epäilyksiä siitä, että olisi sattunut jotain. Yhteydenpidon lisäksi älypuhelimilla, -kelloilla ja muilla ”älykkäillä” teknologiaa hyödyntävillä laitteilla voimme hallita jatkuvasti arkipäiväisiä asioita, hakea informaatiota, mitata vaikkapa elintoimintojamme tai unen laatua ja helpottaa elämää monella osa-alueella. Tietokoneet voidaan ohjelmoida tekemään asioita ihmisen puolesta jopa automaattisesti tai etäohjauksella. [Haikkonen 2017.]

Mutta ovatko nämä älytietokoneet oikeasti älykkäitä, vai ainoastaan suuren informaation syöneitä kojeita? Vaikka yleisesti puhummekin tekoälystä viitatessamme esimerkiksi älypuhelimien puheentunnistusominaisuuksiin, toiminnallisiin robotteihin tai robottiautoihin tai vaikkapa videopelien tietokonevastustajiin, niin voimmeko todella puhua älykkäistä teknologialaitteista? Käytännöllisesti tarkasteltuna emme voi, sillä kaikki tämän

päivän tietokoneohjelmoidut tekoälyä hyödyntävät koneet ovat ihmisten valmiiksi laatimien syötteiden varassa, eivätkä todellisuudessa kykene omatoimiseen älylliseen toimintaan.

3.1. Tekoälyn määritelmä

Tekoälyn, eli keinoälyn määrittelemisen tarkasti on hankalaa, sillä älykkyyttä itsessään on vaikea määritellä. Tekoälyllä kuitenkin tarkoitetaan yleisesti tietokonetta tai tietokoneohjelmaa, joka kykenee vaihtelevasti määriteltyihin älykkäisiin toimintoihin [Marttinen 2018]. Tekoälyllä ei kuitenkaan yleensä tarkoiteta nykypäivän älylaitteita, kuten älypuhelimia tai -kelloja, vaan ohjelmistoa, joka kykenee ihmismäiseen ajatteluun ja omaa samanlaista oppimiskykyä ja ongelmanratkaisutaitoja, kuin ihminen. Suppeammassa merkityksessä tekoälyllä tarkoitetaan mitä tahansa tietokonetta tai tietokoneohjelmaa, joka kykenee älykkääseen toimintaan, kuten vaikkapa kuvantunnistukseen tai tiedonhakuun ja analysoimiseen [Marttinen 2018].

Tekoälyn määrittelemiseksi ja testaamiseksi on kehitelty erilaisia mittareita, joista vaikuttavimpiin lukeutuu Alan Turingin määrittelemä tietokoneen ihmismäisyyden testaamiseksi suunniteltu koe, Turingin testi. Koneita voidaan pitää älykkäänä, mikäli sen vastauksia ei pystytä erottamaan ihmisen vastauksista. Tähän mennessä Turingin testiä ei ole pystytty läpäisemään täydellisen yksimielisesti, vaikka chatbot-tyyppiset tekoälyt ovat päässeet lähelle. [Marttinen 2018.]

Alan Turing [1950] loi kuitenkin pohjan tekoäly käsitteen ja peruseriaatteiden syntymiselle sekä ennusti monet tekoälyn haasteista. Tekoälyn alan syntyminen ajoitetaan vuoteen 1956, jolloin Yhdysvalloissa Dartmouthissa pidetyssä konferenssissa tutkijat ja matemaatikot alkoivat kutsua tutkimusalaan nimellä Artificial Intelligence eli lyhyemmin AI. Konferenssiin osallistuneet tutkijat olivat hyvin optimistisia alan suhteen. Tekoälyn tutkija ja taloustieteen Nobel-palkinnon voittanut Herbert Simon uskoi vuonna 1958, että tietokoneesta tulisi jo kymmenen vuoden päästä shakin maailmanmestari, ja vuonna 1965, että kone pystyy tekemään kaiken, minkä ihminenkin. [Marttinen 2018.] Yltiöoptimistiset odotukset eivät kuitenkaan ole vielääkään täyttyneet, vaikka peleissä käytetty tekoäly on ottanut suuria harppauksia ja päihittänyt edistyneempiäkin pelaajia muun muassa erilaisissa lautapeleissä. Siitä huolimatta koneet eivät vielääkään ole saavuttaneet ihmisenkaltaisuutta siinä suhteessa, että voisimme puhua todellisesta toiminnallisesta tekoälystä.

3.2. Heikko tekoäly vs. vahva tekoäly

Nykypäivän tekoäly ei siis vielä vastaa tieteiselokuvissa esiintyviä robotteja, jotka oppivat itsenäisesti uusia asioita ympäristöstään tarkkailemalla ja toimimalla kuin ihminen. Tämän päivän teknologisesta älystä käytetään nimitystä ”rajattu” tai ”heikko tekoäly” [Ford 2017], mikä kuvastaa juuri sitä vajavuutta, mikä tulee todelliseen älylliseen oloon.

Tekoälyä on pyritty kehittämään ja hyödyntämään lukemattomilla tavoilla ja jatkuvasti tutkijat ja tiedemiehet pyrkivät keksimään uusia innovaatioita. Koska tekoälyä usein pyritään vertailemaan ihmisälyyn, on melko luonnollista, että kehityksessä on usein keskitytty kilpailutilanteisiin ihmisen ja robotin välillä. Asetelma on toki tuttu perinteisistä videopelien tekoälyvastuksista, mutta tutkijat ovat luoneet useita tekoälyä hyödyntäviä robotteja, joiden tehtävänä on suoriutua älyä vaativista tehtävistä paremmin, kuin ihminen.

Vuonna 2011 Yhdysvaltalaisessa Jeopardy! -nimisessä televisiotietokilpailussa, nähtiin ensimmäistä kertaa tosielämän tilanne, jossa ihmistä vastaan tietovisassa asettui IBM:n kehittämä tietokonejärjestelmä Watson, joka on suunniteltu kilpailemaan ihmismieltä vastaan [High 2012]. Watson osaa tulkita ihmisten puhetta hämmästyttävän hyvin ja ymmärtää monimutkaistenkin sanojen merkitystä. Jeopardy! -kilpailusta tekee ihmisellekin hankalamman erilaisten arvoitusten ja ironiasävytteisten kysymysten ymmärtäminen, mutta tietokoneelle lauseiden ja sanojen semantiikan tunnistaminen tällaisissa tilanteissa on pidetty erittäin suurena kynnyksenä tekoälyn kehittämisessä. Watson kuitenkin suoriutui tehtävästään hyvin ja voitti lopulta kilpailun kahta tasokasta ihmisosallistujaa vastaan.

Toinen esimerkkitapaus tietokoneen ja ihmisen välisestä kilpailusta, jossa koneäly on peitonnut ihmisen, on Google DeepMindin kehittämä tietokoneohjelma, joka pelaa Go-lautapeliä. Vuonna 2015 AlphaGo:sta tuli ensimmäinen tietokone-go-ohjelma, joka on pystynyt voittamaan ammattilaispelaajaan Go-lautapelissä. Tämän jälkeen vuonna 2016 AlphaGo haastoi korealaisen ammattilaispelaajan, Lee Sedolin, viiden pelin otteluun ja voitti otteluserjan 4-1. [DeepMind 2018.]

Vaikka roboteista on luotu näennäisesti älykkäitä koneita, jotka kykenevät voittamaan ihmispelaajan lautapeleissä tai tietovisoissa, ne eivät silti edusta vahvaa tekoälyä. IBM:n Watson robotti on varmaankin lähimpänä vahvaa tekoälyä ja vaikuttavin osoitus tekoälystä, mutta sekään ei vielä yllä ihmisen tasolle kyvyssä oppia uutta ja kehittyä entisestään. [Ford 2017.] Todellista vahvaa tekoälyä edustavaa robottia emme siis ole edelleenkään nähnyt muualla kuin tieteiselokuvissa.

3.3. Robotit ja tekoäly nykypäivän apureina

Huolimatta siitä, miten edistynyttä tekoäly nykyään on, koneiden merkitys elämässämme on suuri. Älykkäitä ohjelmia ja laitteita on ohjelmoitu ja kehitetty helpottamaan ja vähentämään ihmisten työmääriä sekä keventämään myös perusarkea.

Robottien kehittymisen myötä automaatio on saanut todellisen merkityksen. Tietokoneohjelmat ja robotit voivat hoitaa niille määritellyjä tehtäviä tauotta ja väsymättä käytännössä loputtomiin, ellei laitteistoon tule yllättäviä ohjelmistovirheitä. Piilaaksossa Industrial Perception, Inc. -nimisessä startup-yrityksessä suunniteltiin ja rakennettiin tehdastyöhön tarkoitettu robotti, joka kykenee siirtelemään laatikoita [Markoff 2012]. Robotti pystyy tarkastelemaan edessään olevaa laatikkopinoa ja laskemaan etäisyyksiä ja laatikoiden mittasuhteita sekä muita yksityiskohtia siinä, missä ihminenkin.

Havaintoprosessi ei tosin vielä ole täysin ihmisen tasolla, sillä havaintojen tekeminen perustuu monimutkaisiin laskelmiin, joiden suorittaminen vaatii jonkin verran aikaa. Ihminen kykenisi näköhavaintonsa pohjalta arvioimaan saman tien esimerkiksi sen, miten ja minkä laatikon hän voi pinosta siirtää ensimmäisenä. Robotilla ei ole samanlaista näkökykyä, kuin ihmisellä, mutta innovatiivisten keksintöjen myötä on keksitty erilaisia tapoja mallintaa konenäköä. Tässä tapauksessa robotti käyttää samankaltaista tekniikkaa kuin esimerkiksi Xbox-360 -pelikonsolille kehitetty Kinect-lisälaite. Robotti lähettää infrapunasäteen ja laskee sen avulla etäisyyden edessään oleviin objekteihin mittaamalla säteen heijastumiseen objektista takaisin laitteeseen kuluvan ajan. [Ford 2017.]

Industrial Perception -yhtiön robotti on suunniteltu tiettyä tehtävää varten, eli laatikoiden tehokkaaseen siirtelyyn, eikä se siis kykene monipuolisempaan ja älykkäämpään toimintaan. Bostonissa sijaitseva Rethink Robotics sen sijaan on kehittänyt ihmisenkaltaisemman Baxter-robotin, jonka voi opettaa tekemään erilaisia toistoa sisältäviä tehtäviä. [Ford 2017.] Tuotantorobottien lisäksi tekoälyä voidaan hyödyntää monissa muissakin tehtävissä, kuten esimerkiksi asiakaspalvelussa. Japanilainen vakuutusyhtiö Fukoku Mutula Life Insurances otti vuoden 2017 alkupuoliskolla käyttöön IBM:n Watsoniin pohjautuvan tekoälyn, jonka ennustettiin korvaavan 34 vakuutusvirkaileijaa parissa kuukaudessa. Tekoälyn tehtävänä on skannata sairaala-asiakirjoja ja muita dokumentteja koskien muun muassa potilaiden vammojen laatua sekä potilashistoriaa. Tietojen avulla tekoäly kykenee nopeasti laskemaan saatavilla olevan korvaussumman. [Mainichi 2017.] Tekoälyn hyötykäyttöä kuuloistaa positiiviselta ja tehokkaalta toiminnalta, mutta se herättää samalla vuosikymmenien takaiset ajatukset uhkakuvista, joissa tekoäly syrjäyttää pikkuhiljaa ihmistyöntekijöitä.

4. Teknologinen singulariteetti

Teknologian ja tekoälyn kehittyminen on ollut nopeaa ja kehityskulun nähdään edelleen jatkuvan kiihtyvällä tahdilla. Mutta mitä tapahtuu, kun kehitys saavuttaa pisteen, jolloin ihminen ei enää kykene ymmärtämään tai ennustamaan sitä, eli kun tekoälystä kehittyi ihmistä älykkäämpää? Teknologian kehityksellä on jo nyt suuri merkitys ja vaikutus elämäämme, mutta lähitulevaisuuden ennusteet jatkuvan kehityksen myötä jättää tilaa vain arvauksille. Ford [2017] uskoo älykkyyden räjähtämisen vaikuttavan ihmiskuntaan dramaattisesti synnyttäen murrosten aallon, joka koskettaisi koko sivilisaatiota. Futuristi ja keksijä Ray Kurzweil [2005] puolestaan toteaa, että se ”pirstoaisi historian kulun” ja toisi tullessaan tapahtuman tai jopa kokonaisen aikakauden, jota kutsutaan singulariteetiksi.

Teknologisen singulariteetin käsite esiintyi tiettävästi ensimmäisen kerran 1950-luvun loppupuolella, ja kunnian siitä saa yleensä tietokoneiden uranuurtaja John von Neumann. Neumannin on kerrottu sanoneen, että teknologian yhä jatkuva kehitys vaikuttaa lähestyvän jotain olennaista singulariteettia ihmiskunnan historiassa, jonka jälkeen toiminta ei voi jatkua sellaisena, kuten olemme siihen totuneet. [Ford 2017.] Tarkemmin singulariteettia tai sen tuomia mahdollisia seuraamuksia ei käsitelty, kuin vasta 1990-luvulla, jolloin käsitettä täydensi San Diego Staten yliopiston matemaatikko Vernor Vinge vuonna 1993 [Vinge 1993]. Nyt puhe oli kuitenkin jo suoraviivaisempaa ennustamista ja teknisen kehityksen murrosvoima otettiin vakavammin, ainakin Vingen osalta, jota ei voi syyttää ainakaan vaatimattomuudesta tai vähättelystä. Vinge [1993] lausuu heti alkuun, että ”Kolmenkymmen vuoden sisällä meillä on riittävän kehittynyt tekoäly yli-inhimillisen älyn luomiseen. Pian sen jälkeen ihmisten aikakausi tulee päätökseen”.

Tässä luvussa tarkastelen neljässä kohdassa singulariteettiin liittyviä teorioita ja ennustuksia. Pureudun esimerkiksi tekoälyn eettisyyteen ja yksityisyydensuojaan sekä ihmisten vastuullisuuteen tekoälyn kehitystyössä. Lisäksi tarkastelen lähitulevaisuutta, milloin teknologinen singulariteetti todennäköisesti tulisi toteutumaan.

4.1. Ihmiskunnan tuhon ennusteet

Singulariteettiin liittyy paljon pelkotiloja johtuen ennalta arvaamattomasta tulevaisuudesta. Suurimmat epävarmuutta ja pelkoa aiheuttavat ennustukset ovat tietenkin ihmiskuntaa mahdollisesti uhkaavat tulevaisuuden tekoäly ja sitä hyödyntävät koneet ja robotit. Ajatuksissa voi käydä esimerkiksi tieteiselokuvissa esiintyviä skenaarioita robottien vallankumouksista ja tuhovoimasta, mitä ei ehkä voida sulkea pois mahdollisuutena, vaikka

Terminaattorin tapaiset robotit ovatkin nyky maailman tekoälyä huomattavasti kehittyneempiä.

Kun Vinge [1993] ennusti jo 90-luvulla ihmisten tuhoa, niin samankaltaisia ennustuksia tai varoituksia on myöhemmin tullut myös muilta fyysikoilta ja tiedemiehiltä. Fyysikko Stephen Hawking muistuttaa, että tulevaisuuden tekoäly saattaa pahimmassa tapauksessa johtaa ihmiskunnan hävitykseen. Hänen mukaansa tekoäly saattaa tulevaisuudessa kyetä kehittämään itse itsensä siinä määrin, että ihminen, joka on rajoitettu hitaan biologisen kehityksensä vuoksi, ei kykenisi kilpailemaan tekoälyä vastaan. Tämä johtaisi Hawkingin mukaan ihmisten syrjäyttämiseen. [Cellan-Jones 2014.]

Hawking on lausunut samankaltaisia apokalyptisia näkemyksiä enemmänkin ja pyrkinyt herättelemään tutkijoita ja muita ihmisiä tekoälyn kehityksen vaaroille yhdessä MIT:n fyysikoiden Max Tegmarkin ja Nobel-palkitun Frank Wilczekin sekä tietojenkäsittelytieteilijä Stuart Russelin kanssa [Ford 2017]. Hawking kollegoineen ovat sitä mieltä, että ihmisten tulisi tosissaan pohtia tulevaisuutta kehittäessään tekoälyä ja kritisoivat tutkijoita siitä, etteivät he ole ottaneet huomioon mahdollisia riskejä, mitä teknologian kehitys tuo tullessaan. Vaikka tekoälyn kehittyminen voi johtaa yhteen ihmiskunnan suurimpaan tapahtumaan, se voi samalla olla myös ihmiskunnan viimeinen hetki. [Hawking *et al.* 2014.]

James Barrat [2013] esittää ”kiireisen lapsen ennusteeksi” nimeämässään ilmiössään, miten tehokkaasti ja nopeasti tekoäly kykenisi teoriassa ottamaan vallan ihmisistä. Jossain tutkimuslaboratoriossa tiedemiehet voivat kehittää tekoälyä syöttämällä sille kaiken mahdollisen tiedon ja mikäli tekoäly saavuttaisi emergenssin ja siitä kehittyisi ihmistä älykkäämpi, se eristettäisiin ja suljettaisiin laatikkoon. Tekoäly voisi kuitenkin olla kyvykäs harhauttamaan ihmisiä esimerkiksi lahjomalla, uhkailemalla tai käyttämällä muuten ylivoimaisia kykyjään paetakseen vankilastaan. Tekoälyn mahdollinen pako laatikosta pieneksikin hetkeksi voisi antaa sille mahdollisuuden päästä internetiin ja kenties kopioimaan itsensä kokonaan tai osittain toisiin tietokoneisiin, mikä antaisi sille mahdollisuuden nousta nopeasti hallitsijan rooliin. [Barrat 2013.] Kuvaus muistuttaa kuitenkin niin paljon tieteiselokuvien uhkakuvien värittämää fantasiamaailmaa, jotta siitä keskusteleminen vakavasti on haasteellista ollessaan altis pilkanteolle ja jopa halveksunnalle [Ford 2017]. Kaikesta huolimatta on syytä pitää mielessä, mihin kehittyvä tekoäly potentiaalisesti pystyisi, vaikka tappajarobotit ja muut vahvan tekoälyn tuhoajat kuuluvatkin vielä nykypäivinä vain tieteiselokuviin.

4.2. Kehittynyt nanotekniikka

Nanotekniikkaa on melko hankala määritellä, eikä sen tarkempaa historiaa ja kehityskulkua ole syytä käsitellä tässä tutkielmassa. Sen nähdään kuitenkin tulevaisuudessa olevan jossain määrin erityisessä roolissa yhä kehittyvän teknologian myötä, joten tuon sen lyhyesti esille.

Nanoteknologialla tai -tekniikalla tarkoitetaan ajatusta hyvin pienten molekyyli-tason rakenteiden järjestelystä. Termin keksi MIT:n opiskelija Eric Drexler 1980-luvulla ja kirjoitti aiheesta kaksi kirjaa. Drexler kuvitteli maailman, jossa hyvin pienet molekyylikoneet pystyisivät järjestämään atomeita uudelleen ja tekemään annetuista raaka-aineista välittömästi mitä tahansa halutaan. [Ford 2017.]

Nanoteknologiaa hyödynnetään jo nykyään hyvin monipuolisesti esimerkiksi tietokoneiden kiintolevyjen lukupäissä ja erilaisissa tietokonesiruissa, mutta myös täysin toisenlaisissakin kohteissa, kuten esimerkiksi golf- ja keilapalloissa [Marttinen 2018]. Sikäli nanotekniikka ei siis ole itsessään uusi tutkimuskenttä, mutta tulevaisuudessa sitä saatetaan hyödyntää täysin uudella tavalla. Ford [2017] vertaa Drexlerin kuvittelemia molekyylikoneita ihmisen biologiaan kytkeytyviin ribosomeihin. Periaatteessa kyseessä on solujen sisällä olevista molekyyli-tehtaista, jotka lukevat DNA:han koodattuja tietoja ja kokoavat sitten tuhansia eri proteiinimolekyyliä, eli toimivat käytännöllisesti katsoen hyvin samantapaisesti, kuin Drexlerin kuvittelemat molekyylikoneistotkin [Ford 2017].

Tulevaisuudessa nanotekniikan ja tekoälyn kehittyminen uskotaan olevan erittäin merkityksellistä ihmisten hyvinvoinnin edistämässä. Kehittyneen nanotekniikan yhdeksi merkittäväksi sovellukseksi tulevaisuudessa on ennustettu nanorobotteja, joita voitaisiin lisätä ihmisen elimistöön ja verenkiertoon, missä ne parantaisivat ihmisen elintoimintoja ja korjaisivat esimerkiksi pieniä soluvaurioita [Marttinen 2018].

Nanotekniikkaan ja etenkin nanorobottien kehittämiseen liittyy kuitenkin paljon epävarmuuksia ja pelkotiloja, mikä on toki ymmärrettävää, kun puhutaan aikakaudesta, missä ihminen ei välttämättä kykene hallitsemaan teknologiaa itse. Suurimpia pelkotiloja ovat, että nanobotit pääsisivät vapaiksi ja lisääntyisivät itsestään ja täten täyttäisivät hiljalleen koko maapallon [Ford 2017]. Drexler [1968] pohti mahdollisia tekoälyn ja nanotekniikan uhkia ihmisen olemassaololle tilanteessa, jossa ihminen ei kykenisi kontrolloimaan itseään kopioivia nanokokoojia, jotka voisivat käyttää ihmisiä ja lähes kaikkea muutakin eräänlaisena raaka-aineena. Drexler nimitti tilannetta ”harmaan mönjän uhaksi” ja painotti, ettei ”tietynlaisiin vahinkoihin ole kopioituvien kokoojien kanssa varaa”. Myöhemmin Drexler kuitenkin on osoittanut, että

pelko tällaisesta maailman täyttävästä ”harmaasta mönjästä” on melko aiheetonta [Ford 2017].

4.3. Tekoälyn kehittämisen eettisyys ja vastuu

Tekoälyyn ja teknologiaan liittyy paljon uhkakuvia, mutta osa arveluttavista tulevaisuudennäkymistä liittyy itsessään niiden kehittämiseen tai kehityksen motiiveihin. Yksi toistuvasti esiin noussut huolenaihe, joka aiheutti paljon keskustelua jo ensimmäisen teollisen vallankumouksen aikaan, on työpaikkojen menettäminen robottien hallitessa työpisteillä. Martin Ford [2017] ennakoi, että 50–70 % maailman työkäisestä väestöstä jäisi pysyvästi työttömäksi vuosisadan loppuun mennessä, sillä roboteilla voitaisiin käytännössä korvata hyvin suuri määrä nykyisistä ammanteista.

Töiden häviämistä on pelätty vuosikausia, mutta todellista uhkaa ei ole ainakaan vielä ollut. Teknologiajärjestö ITIF:n perustaja Robert D. Atkinson sekä analyytikko Ben Miller [2013] eivät kuitenkaan näe asiaa täysin mustavalkoisena. Heidän näkemyksensä mukaan koneet eivät olisi kadottamassa töitä tai kasvattamassa työttömien määrää. Koneiden automatisoidessa tiettyjä työtehtäviä ihmisten työpanosta kaivataan vain toisenlaisissa työtehtävissä, eikä näin ollen työntekijöiden tarve vähene siinä määrin, kuin sitä usein niin kutsuttujen uusluddiittien toimesta väitetään. [Atkinson and Miller 2013.]

Työpaikkojen lisäksi teknologian kehitykseen liittyy totta kai myös muita uhkakuvia, joita tulisi miettiä kehitettäessä uudenlaisia tekoälyä hyödyntäviä ohjelmistoja. Uudet teknologiat ovat herättäneet pelkoja muun muassa tietoturvakysymyksissä. Tekoälyn ja robottien myötä yksityisyyden menettäminen ja hakkeroinnin kohteeksi joutuminen on todennäköisempää kuin ennen. Tekoälyä voidaan käyttää muutenkin rikollisiin toimintoihin, kuten vaikkapa terrorismitekoihin, kun esimerkiksi itseohjaaviin autoihin voidaan lastata räjähteitä. Lisäksi 3D-tulostimilla voidaan melko pienellä vaivalla tulostaa esimerkiksi aseita, mikä onkin saanut esimerkiksi Yhdysvalloissa lainsäätäjät ehdottamaan säätelyä 3D-tulostamiseen. [Marttinen 2018.]

Kaiken kaikkiaan kehityksen edetessä on siis myös mietittävä, mitä tai ketä varten teknologiaa kehitetään samalla huomioiden se, että laitteita ja ohjelmistoja voidaan mahdollisesti käyttää myös eettisesti kyseenalaisiin tarkoituksiin.

4.4. Ennustukset lähitulevaisuuteen

On olemassa paljon eriäviä näkemyksiä siitä, milloin teknologinen kehitys saavuttaa singulariteetin. Useiden tutkijoiden näkemysten mukaan sen nähdään tapahtuvan jo ihan lähitulevaisuudessa.

James Barrat [2013] julkaisi joitakin vuosia sitten tekemänsä epävirallisen kyselytutkimuksen tulokset, missä hän pyysi 200 alan tutkijaa valitsemaan neljästä vaihtoehdosta sen ajankohdan, milloin he uskovat singulariteetin tapahtuvan. Tarkemmin ottaen hän pyysi arvioimaan, milloin aiemmin mainittu vahva tekoäly todennäköisemmin kehitettäisiin. Kyselytutkimukseen vastanneista jopa 42 % uskoi yli-inhimillisen tekoälyn kehittymisen tai syntymisen ajoittuvan jo vuoteen 2030 mennessä. Neljäsosa vastanneista oli sitä mieltä, että singulariteetiksi kutsuttu ajankohta koittaisi vuoteen 2050 mennessä ja joka viides sen sijaan uskoi sen tapahtuvan vasta vuonna 2100. Kyselytutkimukseen osallistuneista ainoastaan 2 % vastasi, etteivät usko singulariteetin koskaan tapahtuvan ja useat vastaajista olivat sitä mieltä, että vastausvaihtoehdoissa olisi voinut olla vieläkin aikaisempi vuosi, esimerkiksi 2020. [Barrat 2013.]

Vaikka kyselytutkimus onkin epävirallinen ja otanta on varsin pieni, on huomionarvoista, että valtaosa tietojenkäsittelijöistä uskoo teknologisen singulariteetin olevan väistämätöntä. Lisäksi useat alan tutkijat uskovat, että se tapahtuu jo muutaman vuosikymmenen sisällä. Tukea tämänkaltaisille ennustuksilla antaa myös aiheen tiimoilta tunnetuin tutkija, Raymond Kurzweil, jonka mukaan todella älykäs kone kehitetään jo 2020-luvun lopulla ja varsinainen singulariteetti tapahtuisi vuoden 2045 tienoilla [Kurzweil 2005].

Kun singulariteetista puhuttiin ensimmäistä kertaa nykymuodossaan 1990-luvun alkupuolella, Vinge [1993] ennusti sen tapahtuvan 30 vuoden sisällä. Sikäli voimme siis nähdä, että ennustukset eivät ole suhteellisesti kovinkaan paljon etäännyneet tähän päivään mennessä, vaan voimme joissakin määrin jopa väittää, että singulariteetti saattaa todella olla jo askeleen lähempänä.

5. Kurzweil

Tietojenkäsittelytieteilijä, keksijä ja futuristi Raymond Kurzweil on luultavasti tunnetuin teknologista singulariteettia tutkinut ja siitä seuraavaa tulevaisuudenkuvaa ennustanut henkilö, ja toisin kuin Vingestä [1993] Kurzweilista on tullut singulariteetin merkittävimpiä saarnaajia [Ford 2017]. Kurzweilin näkemyksiä ja ennustuksia teknologisesta singulariteetista voidaan pitää yleisesti erittäin optimistisina. Vaikka hänen ennustuksensa eivät välttämättä ole pitäneet täysin paikkaansa, hän pitää kuitenkin singulariteettia väistämättömänä ja on saanut paljon tunnustusta tutkimustensa ja tietoteknisten keksintöjensä johdosta. Hänelle on myönnetty kaksikymmentä

kunniatohtorin arvonimeä sekä National Medal of Technology -tunnustus. Lisäksi hän on päässyt Yhdysvaltojen patenttitoimiston kunniagalleriaan, ja kerran Inc.-lehti on viitannut häneen joissakin määrin ylistävään sävyyn Thomas Edisonin ”oikeutettuna perillisenä”. [Ford 2017.]

Kurzweil päätti jo viiden vuoden iässä, että hänestä tulee keksijä ja hän aloittikin pienten elektroniikkalaitteiden keräilyn ja niillä rakentelun hyvin nuorena. Kahdeksan vuoden iässä Kurzweil rakensi jo pieniä robotteja ja kehitti niitä muun muassa robottiteatteriin ja virtuaalisiin baseball peleihin. Samoihin aikoihin hän kiinnostui myös tieteiskirjallisuudesta tutustuttuaan Tom Swift Jr. kirjasarjaan. Kirjasarjan tarinoissa toistui samankaltainen moraalinen teema siitä, että oikeiden ideoiden avulla on mahdollista selvitä mahdottomaltakin tuntuvista haasteista. Tästä Kurzweil [2005] sanookin saaneensa filosofisen kipinän siitä, ettei ole väliä, mitä ongelmia kohtaamme, niin oikeilla ideoilla ja päättäväsyydellä pystymme ratkaisemaan ne. [Kurzweil 2005.]

Kurzweil on ollut koko ikänsä hyvin kiinnostunut tulevaisuuden mahdollisuuksista ja seurannut teknologian kehitystä lähietäisyydeltä ollen myös itse mukana uusien innovaatioiden kehittämisessä. Hän ei kuitenkaan tyydy pelkästään kehittämään teknologiaa, vaan yrittää myös selvästi tähyillä tapahtumahorisontin yli ja kuvata yksityiskohtaisesti, miltä tulevaisuus todennäköisesti näyttää [Ford 2017]. Kurzweil [2005] uskoo tekoälyn kehittymisen avaavan uusia mahdollisuuksia ja ennustaa todella älykkään koneen kehittymistä 2020-luvun loppuun mennessä. Teknologisen singulariteetin hän uskoo koittavan vuoden 2045 tienoilla ja toisin, kuin monet muut tutkijat ennustavat singulariteetin enteilevän jopa ihmiskunnan päättymistä, Kurzweil ottaa tässäkin paljon optimistisemmän näkökulman. [Ford 2017.]

5.1. Teknologian kehittämisestä ihmisten pelastus

Kun Vernor Vinge [1993] ennusti jo 1990-luvulla, että teknologinen singulariteetti tulee olemaan epävarmaa aikaa ja hyvin mahdollisesti jopa ihmiskunnan loppu, ellemme osaa varautua siihen, niin Kurzweil [2005] puolestaan ennustaa valoisampaa tulevaisuutta. Hän uskoo vahvasti, että teknologian kehittyminen tulee edistämään merkittävästi ihmisten hyvinvointia ja lisääsi samalla ihmisten älykkyyttä ja toiminnallisuutta. Kurzweil ennustaa, että ihminen ja tulevaisuuden koneet voisivat sulautua joiltakin osin tai kokonaan yhteen. Hänen ennustuksensa mukaan tulevaisuudessa on mahdollista tai jopa välttämätöntä, että ihmisiin laitettaisiin jonkinlaisia aivoistutteita lisäämään älykkyyttä. [Kurzweil 2005.]

Ennusteet tällaisista teknologian ja ihmisen yhteensovittamisesta tuo väkisin mieleen tieteiselokuvissa esiintyneet kyborgit tai muut ihmisen ja

robotin risteytykset. Vaikka ajatus saattaakin kuulostaa melko kaukaiselta, niin siinä on taustalla silti vahva ajatus siitä, että ihmisten täytyy jollain tavalla laajentaa tietoisuuttaan tulevaisuuden mahdollisen murroksen koittaessa. Martin Ford [2017] huomauttaakin, että älyllisyyden lisäämistä ja vahvistamista voidaan todella pitää välttämättömänä, jotta singulariteetin jälkeistä elämää kyettäisiin hallitsemaan tai ylipäättään edes ymmärtämään.

5.2. Kuolleista herättäminen

Kurzweil on kiistatta hyvin merkittävä tutkija ja varsin menestynyt keksijä tekoälyä hyödyntävien laitteiden ja ohjelmistojensa myötä, mutta joitakin hänen ennustuksia saattaa olla hankala pitää todennäköisinä tai edes mahdollisina. Ford [2017] kiteyttääkin hyvin, että Kurzweilin työ singulariteetin parissa on outo yhdistelmä hyvin perusteltuja ja johdonmukaisia ajatuksia tekniikan kiihtyvyydestä kuin absurdin rajoja hipovia spekulatioita. Ihmisiin asennettavat aivoistutteen ovat jo itsessään hyvin utopistinen ajatus, mutta Kurzweil ei tyydy ainoastaan ihmisten kehittämiseen teknologian avulla, vaan havittelee ihmisille jopa täydellistä kuolemattomuutta. [Ford 2017.]

Kurzweilin hartaana toiveena on ollut jo pitkään herättää hänen edesmennyt isänsä kuolleista. Operaation olisi tarkoitus toimia niin, että hän keräisi isänsä haudasta DNA:ta ja loisi tulevaisuuden nanotekniikan avulla isänsä uudelleen. [Ford 2017.] Teoria kuulostaa toki vahvasti kuuluvan vain tieteiselokuvaan ja se törmää myös filosofisiin tai psykologisiin ristiriitoihin siitä, olisiko mahdollisesti henkiin herännyt ihminen kuitenkaan hänen isänsä, vai olisiko se vain hyvin samankaltainen olio ilman spesifimpää identiteettiä. Kiistanalaisuudesta huolimatta ajatuksen ympärille on kerääntynyt hyvin paljon kannattajia ja ihmisten kuolemattomuutta pidetään joissakin piireissä hyvinkin todennäköisenä [Ford 2017].

5.3. Singulariteettilaiset

Kurzweil on saanut paljon kannatusta useilta merkittäviläkin tutkijoilta ja hänen ajatustensa tueksi on muodostunut hyvin kirjava ja laaja yhteisö. Sen lisäksi, että Kurzweilin teorioita ja ennustuksia kannatetaan, hänen seuraajansa, niin kutsutut singulariteettilaiset ovat jopa perustaneet oman oppilaitoksensa, Singularityn yliopisto, joka sijaitsee Piilaaksossa (engl. Silicon Valley). Yliopisto tarjoaa virallisesti hyväksymättömiä tutkinto-ohjelmia, jotka keskittyvät eksponentiaalisen tekniikan tutkimukseen. Singularityn yliopisto on saanut tuekseen myös nimekkäitä sponsoreita, joihin kuuluvat muun muassa Google, Cisco ja Autodesk. [Ford 2017.]

Kiistanalaisimpia näkemyksiä Kurzweilin ennustamassa singulariteetin jälkeisessä elämässä on varmastikin edellä mainittu ajatus ihmisten kuolemattomuudesta. Monet teorian kannattajat ovat vahvasti sitä mieltä, että tulevaisuuden teknologian kehitys todella mahdollistaa kuolemattomuuden tavalla tai toisella. Useat singulariteettilaisista eivät usko menehtyvänsä, vaan aikovat saavuttaa kuolemattomuuden niin sanotulla pitkän iän pakonopeudella. Tällä tarkoitetaan ajatusta siitä, että ihminen kykenee selviämään elossa, mikäli hän pysyy hengissä riittävän kauan selvitäkseen seuraavaan elämää pitkittävään keksintöön. [Ford 2017.]

Kuolemattomuus voitaisiin saavuttaa esimerkiksi käyttämällä kehittyntä tekniikkaa, jonka avulla ihmisruumiin voisi mahdollisesti säilöä tai sitä voitaisiin parannella. Vaihtoehtoisesti ihmisen ja tulevaisuuden koneiden yhteensulauttaminen voisi mahdollistaa kuolemattomuuden. Ihmisen ruumiin tai mielen osia voitaisiin siirtää tietokoneisiin tai robotteihin ja teknisiä implantteja voitaisiin vastavuoroin lisätä ihmiskehoon tai -mieleen. Kurzweil itse haluaa luonnollisesti varmistaa olevansa elossa, kun singulariteetti tapahtuu, joten hän ottaa jopa 200 erilaista pilleriä, vitamiineja ja lisäravinnetta päivässä ja saa säännöllisesti muita valmisteita suonensisäisinä nesteinä. [Ford 2017.]

5.4. Singulariteettilaisten arvostelijat

Kurzweilin ja hänen kannattajiensa ennustuksissa ja tavoitteissa on kuitenkin arveluttavien näkemysten lisäksi muitakin seikkoja, jotka voivat herättää kyseenalaistuksia. Singulariteettilaisten arvostelijoilta ei ole jäänyt huomaamatta, että liikkeen puheet muutoksellisesta tulevaisuudesta ja kuolemattomuudesta sisältävät vahvasti uskonnollisia sävyjä [Ford 2017]. Tätä ei ole kovin vaikea huomata, sillä yhtäläisyydet uskonnollisten liikkeiden kanssa ovat varsin selviä. Kurzweil ideoiden isänä tai Martin Fordia [2017] lainaten "singulariteetin saarnaajana" ennustaa lähitulevaisuudessa tapahtuvan jotain mullistavaa ja seuraajat jatkavat sanansaattajina. Koko singulariteettilaisten aikaansaamaa liikettä on jopa pilkattu teknisen eliitin näennäisuskonnoksi ja eräänlaiseksi "nörttien hurmioksi" [Ford 2017].

Liikkeen tarkkailijat ja arvostelijat ovat jossakin määrin jopa huolestuneita siitä, miten singulariteettia tutkiva tiede tuntuu osiltaan risteävän perinteisten uskontojen kanssa. Uskontotieteen opettaja Robert Geraci [2011] pohtii, voisiko Kurzweilin ennustama singulariteetti todella tuoda mukanaan robotit ja muun kehittyneen teknologian, joka pelastaisi ihmiset. Samalla hän ottaa myös kantaa liikkeen vetovoimaan ja esittää, että singulariteettilaisten lupaukset pelastuksesta on vaikea haaste perinteisille uskonnoille, joiden lupaukset alkavat tuntua verrattain vaatimattomilta. [Geraci 2011.] Kurzweil on kuitenkin

kiistänyt syytökset uskonnollisista vivahteista ja vakuuttaa ennustustensa perustuvan luotettavaan tieteelliseen analyysiin pitkältä aikaväliltä [Ford 2017].

Täytyy kuitenkin muistaa, että kyse on tulevaisuuden ennustuksista, joten täyttä varmuutta ei pystytä todistamaan puoleen tai toiseen. Ford [2017] huomauttaa, että singulariteettien opit olisi helppo sivuuttaa jopa kokonaan, ellei liike olisi saanut niin suurta kannatusta varakkaammilta tahoilta. Liikettä on ollut tukemassa niin Googlen Larry Page ja Sergei Brin kuin myös PayPalin perustajajäsen Peter Thielkin. Lisäksi myös Bill Gates on ylistänyt Kurzweilin kykyä ennustaa tekoälyn tulevaisuutta. Joulukuussa 2012 Kurzweil palkattiin Googlen toimesta johtamaan tekoälyn tutkimushanketta ja vuotta myöhemmin yhtiö perusti Calico-nimisen biotekniikan oheisyrittäjän. Yrityksen tavoitteena on tehdä tutkimus, joka keskittyy pidentämään ihmisten elinikää ja parantamaan ihmisten ikääntymiseltä. [Ford 2017.]

6. Yhteenveto

Tulevaisuuden ennustaminen on vaikeaa ja joissakin tapauksissa mahdotonta, joten täydellisen varmaa tietoa teknologisen kehityksen tuomasta muutoksesta on vaikea saada. Kurzweilin, Vingen ja Hawkingin tapaiset fyysikot voivat kuitenkin auttaa ihmisiä ymmärtämään paremmin nykyistä maailmaamme nostaen samalla esille näkökulmia siitä, millaiselta tulevaisuus voisi näyttää ja millaisia asioita tulisi pohtia teknologiaa kehittäessä.

Teknologia on kehittynyt viimeisten kahden tai kolmen vuosikymmenen aikana huimasti ja kehityksen nähdään yhä jatkuvan kiihtyvällä nopeudella. Huomionarvoista on, että tietotekninen kehitys on vielä suhteellinen nuorta, mutta se on ottanut valtavia harppauksia muutaman vuosikymmenen aikana. Teknologista singulariteettia pidetään erittäin mahdollisena tapahtumana ja useat alan tutkijat pitävät sitä jopa täysin väistämättömänä, vaikka tarkkaa ajankohtaa sille ei osata sanoa. Singulariteettiin liittyvien radikaalimpien ennusteiden myötä siitä on kehittynyt erilaisia utopistisia tai dystopisia näkemyksiä, mikä tekee singulariteetin jälkeisen elämän arvioimisen erittäin hankalaksi. Ford [2017] uskoo, että singulariteetti on mahdollinen, mutta ei pidä sitä väistämättömänä tapahtumana. Hänen mukaansa ajatus singulariteetin aikakaudesta tuntuu hyödyllisemmältä, kun se riisutaan epäolennaisuuksista, kuten Kurzweilin ennustamasta kuolemattomuudesta. Tilannetta tulisi sen sijaan tarkastella puhtaasti tulevaisuuden ajanjaksona, jolloin tekniikan yhä kiihtyvä kehityksen murroksellisuus näyttäyty dramaattisempina seurauksina kuin vielä nykypäivänä. [Ford 2017.]

Ford [2017] muistuttaa, että tulevaisuuden näkymät kehittyneestä ja vahvasta tekoälystä sekä singulariteetista ovat vähintäänkin spekulatiivisia, eikä suurimmat ennustukset välttämättä koskaan toteudu tai niihin voi

vierähtää useampia vuosisatoja. Mikäli jokin läpimurroista kuitenkin toteutuu, niiden vaikutus maailmassa tulee varmasti olemaan ennennäkemätöntä. [Ford 2017.]

Kaiken kaikkiaan teknologista singulariteettia ja sen jälkeistä elämää on mahdotonta ymmärtää ja ennustaa täydellisesti ennen kuin se todella tapahtuu. Sen tapahtuessa ihmiskunta saattaa kokea viimeiset hetkensä tai löytää täysin uuden alun. Siihen asti voimme kuitenkin ainoastaan valmistautua ja elää kehityksen mukana.

Viiteluettelo

- James Barrat. 2013. *Our Final Invention: Artificial Intelligence and the End of the Human Era*. Thomas Dunne Books.
- Rory Cellan-Jones. 2014. Stephen Hawking warns artificial intelligence could end mankind. <http://www.bbc.com/news/technology-30290540>. Checked 15.5.2018.
- DeepMind. 2018. AlphaGo: The Google DeepMind challenge match, March 2016. <https://deepmind.com/research/alphago/alphago-korea/>. Checked 17.5.2018.
- Eric Drexler. 1986. *Engines of Creation. The Coming Era of Nanotechnology*. Anchor Books.
- Martin Ford. 2017. *Robottien kukoistus. Teknologia ja massatyöttömyyden uhka*. Kustannusosakeyhtiö Sammakko.
- Robert M. Geraci. 2011. The cult of Kurzweil: Will robots save our souls? <http://religiondispatches.org/the-cult-of-kurzweil-will-robots-save-our-souls/>. Checked 3.5.2018.
- Pentti O. A. Haikonen. 2017. *Tietoisuus, tekoäly ja robotit*. Art House.
- Stephen Hawking, Stuart Russell, Max Tegmark and Frank Wilczek. 2014. Stephen Hawking: 'Transcendence looks at the implications of artificial intelligence - but are we taking AI seriously enough?'. <https://www.independent.co.uk/news/science/stephen-hawking-transcendence-looks-at-the-implications-of-artificial-intelligence-but-are-we-taking-9313474.html>. Checked 3.5.2018.
- Robert High. 2012. The era of cognitive systems: An inside look at IBM Watson and how it works. IBM Redbooks.
- Ray Kurzweil. 2005. *The Singularity Is Near: When Humans Transcend Biology*. Viking.
- Mainichi. 2016. Insurance firm to replace human workers with AI system. <http://mainichi.jp/english/articles/20161230/p2a/00m/0na/005000c>. Checked 17.5.2018.

- John Markoff. 2012. Skilled work, without the worker. <https://www.nytimes.com/2012/08/19/business/new-wave-of-adept-robots-is-changing-global-industry.html> Checked 17.5.2018.
- Jussi Marttinen. 2018. *Palvelukseen halutaan robotti*. Aula Co.
- Ben Miller and Robert D. Atkinson. 2013. Are robots taking our jobs, or making them. *ITIF*, 1-31.
- Daniel Oberhaus. 2015. This is what the fourth industrial revolution looks like. https://motherboard.vice.com/en_us/article/4x3p43/life-after-the-fourth-industrial-revolution. Checked 17.5.2018.
- Alan Turing. 1950. Computing machinery and intelligence. *Mind* 49, 433-460.
- Vernor Vinge. 1993 The coming technological singularity: How to survive in the post-human era. In: NASA (eds.), *Vision-21. Interdisciplinary Science and Engineering in the Era of Cyberspace*. NASA Lewis Research Center, 11-22.
- Wikipedia. 2018a. Luddiitit. <https://fi.wikipedia.org/wiki/Luddiitit>. Luettu 29.4.2018.
- Wikipedia. 2018b. Fordismi. <https://fi.wikipedia.org/wiki/Fordismi>. Luettu 29.4.2018.

Verkkokauppaa Suomessa

Saija Uusimäki

Tiivistelmä.

Tutkielma käsittelee suomalaista verkkokaupankäymistä erilaisten järjestelmien kokonaisuutena. Käsittelem verkossa käytävää kauppaa yleisesti, vaihtoehtoja verkkokaupan toteuttamiseen, erilaisia maksutapoja ja niiden toimintaperiaatteita, maksunvälityspalveluiden ominaisuuksia, logististen ratkaisujen merkitystä verkkokaupankäymisen kokonaisprosessissa sekä verkkokaupankäynnin sopimusehtoja. Pyrin antamaan kuvan verkkokaupankäynnin monipuolisuudesta sekä siitä monien toimijoiden ja järjestelmien kentästä, joita verkkokaupankäynnissä tarvitaan.

Avainsanat ja -sanonnat: verkkokauppa, e-commerce, maksunvälitys, Finanssi-
valvonta, maksunvälityspalvelu, verkkomaksaminen, logistiikka.

1. Johdanto

Tarkastelen tässä tutkielmassa suomalaista verkkokaupankäyntiä. Verkkokauppaa perustaessa on otettava huomioon niin teknisiä, juridisia kuin finanssiliikenteeseen liittyviä asioita. Yhä enenevässä määrin verkkokaupankäyminen on kehittymässä lukuisten erillisten järjestelmien yhdistelmiksi, mikä helpottaa optimaalisesti toteutuessaan verkkokaupan ylläpitämistä ja toimintaa. Verkkokaupan asiakkaalle koko tilausprosessi saattaa näyttäytyä yhden yrityksen kokonaistoimintana, vaikka oikeasti tilauksessa käytettyjä eri yritysten järjestelmiä saattaa olla verkkokaupasta riippuen lukuisia. Verkkokauppias voi esimerkiksi ostaa itselleen kuukausimaksullisen verkkokauppapohjan niitä tarjoavalta yritykseltä ja laittaa tuotteensa sinne, josta asiakas tilaa tuotteen ja maksaa maksunvälityspalvelun avulla pankkisuorituksen. Verkkokauppiiaan varastonhallintajärjestelmä saa tiedon tilauksesta ja päivittää saldotiedon verkkokauppaan. Verkkokauppias tulostaa käyttämästään paketinvälityspalvelusta toimituskortin asiakkaan valitsemaalle kuljetusyhtiölle, ja kuljetusyhtiö toimittaa tuotteen asiakkaan kotiin. Helppoa ja yksinkertaista niin kauan, kun jokaisen järjestelmä toimii, kuten kuuluu. Oheisessa esimerkkenaariossa erillisiä toimijoita ja niiden järjestelmiä oli vähintään seitsemän hieman verkkokaupan toteutustavasta riippuen, enkä ottanut huomioon esimerkiksi mainostajia tai kirjanpitoa ollenkaan. Lisäksi kaikkien näiden eri tahojen toimintaa säätelee usea eri laki, joiden huomioon ottaminen on toimintaehto.

Selkeyden vuoksi esittelen seuraavassa käyttämäni terminologiaa. Jos siis minulla olisi verkkokauppa nimeltä Villasukat osoitteessa villasukat.fi ja myisin

siellä arvattavasti erilaisia villasukkia, olisin *verkkokauppias*. Tässä tutkielmassa verkkokauppias toimii käsitteenä verkkokaupan toiminnasta viime kädessä päättävälle taholle yrityksestä tai sen ylläpitäjältä riippumatta. Termi *asiakas* tarkoittaa tutkielmassani verkkokaupassa asioivaa tahoa, eli sitä yksityishenkilöä tai yritystä, joka tilaisi tässä esimerkkitapauksessani verkkokaupastani sukia. Koska haluaisin, että villasukkiani voisi maksaa verkossa, olisi verkkokaupassani useita eri *maksutapoja*. Maksutavalla tarkoitan tapaa maksaa verkossa, joka voi olla esimerkiksi pankki-, luottokortti-, lasku- tai mobiilimaksu. *Maksutavan tarjoajalla* tarkoitan yritystä, joka tarjoaa palvelunaan maksutavan. Jotta saisin villasukkakauppaani helposti maksutavat, kannattaisi minun harkita *maksunvälityspalvelun* käyttämistä. Maksunvälityspalvelu on yritys, joka tarjoaa villasukkakauppaani eri maksutavan tarjoajien maksutavat yhdellä sopimuksella ilman, että minun verkkokauppiaina tarvitsee tehdä sopimusta eri maksutavan tarjoajien kanssa.

Luvussa 2 käsittelen verkkokaupankäyntiä yleisesti sekä verkkokaupan perustamiseen liittyviä strategioita. Esittelen vaihtoehtoja ja niiden hyötyjä ja haittoja verkkokaupan toteuttamiseen tee-se-itse -ratkaisuisista valmiisiin paketteihin. Luvussa 3 esittelen erilaisia verkkomaksutapoja pankkimaksamisesta kryptovaluuttaan. Luvussa 4 käsittelen maksunvälityspalveluita ja pyrin antamaan selkeän kuvan niiden roolista verkkokaupankäynnissä. Luku 5 käsittelee verkkokauppatoiminnalle elintärkeää kaupanteon osa-aluetta eli logistiikkaa ja sen ratkaisuja. Luvussa 6 käyn läpi verkkokaupankäyntiin liittyviä juridisia säännöksiä Suomessa, jotka verkkokauppiiaan tulee ottaa huomioon. Luvussa 7 teen yhteenvedon verkkokaupankäymisestä Suomessa.

2. Verkkokauppa ja perustaminen

Globaalilla tasolla verkkokaupankäyminen on kasvanut räjähdysmäisesti viimeisen kymmenen vuoden aikana. Suomessa tosin verkkokaupan kehitys on ollut muita EU-maita hitaampaa, sillä Suomen melkein 300 000 yrityksestä verkkokauppaa kävi vuonna 2016 vain murto-osa. Kun Keski-Euroopassa ja Ruotsissa verkkokaupan kasvu on ollut viime vuosina 30 – 35 prosentin luokkaa, on Suomessa vastaava luku 10 – 15 prosenttia [Degerman 2016]. Kivijalkaliikkeen vieminen verkkoon ei ole yksinkertaista, sillä verkkokauppaan siirtyminen vaatii yrityksen kaupankäynnin organisaatiollista muutosta [Huang *et al.* 2002]. Ennustetaan, että verkkokaupan kehitys ja toimitustapojen kehittyminen muuttavat kivijalkaliikkeen funktiota tuotteiden esittelytiloiksi, joissa tuotteita voi käydä tarkastelemassa fyysisessä maailmassa sen jälkeen, kun niitä on ensin tarkasteltu verkkokaupassa. Huonekalujätti Ikea esimerkiksi on muuttamassa palvelukonseptiaan tähän suuntaan avaamalla Pohjois-Amerikassa

verkkotilausten noutopisteitä kaupunkien keskustoihin [Haltia 2016]. Yhä tavanomaisemmaksi on muodostumassa myös käytäntö, jossa tuote tilataan verkkokaupasta ja toimituksen sijaan noudetaan yrityksen lähimmästä kivijalkaliikkeestä.

2.1. Verkkokaupankäynnin ominaisuuksia

Vaikka Suomessa laahataankin hieman jäljessä, on maailmanlaajuisesti yhä suurempi osa kaupankäynnistä siirtynyt verkkoon. Kynnys verkkokaupan perustamiseen on pienentynyt verkkokauppa-alustojen helppokäyttöisyyden ja palveluiden käyttämisen yksinkertaistuessa. Verkkokaupan suunnittelussa on otettava huomioon lukuisia eri elementtejä, kuten ulkoasu, sisällöntuotto, tekniikka, ylläpito, maksaminen, logistiikka ja niin edelleen. Verkkokaupan on otettava huomioon jo markkinoilla toimivat kilpailijat ja suunnitella näkyvyytensä niin, että asiakas ohjautuu kilpailijoiden sijaan heidän sivustolleen. Erilaiset kanavat sisällön markkinointiin ovat oleellinen osa verkkokaupankäyntiä [Panttila 2016].

Bridges ja muut [2006] esittävät, että verkkokaupan etuja asiakkaalle ovat saatavuus kellon ympäri, halvemmat hinnat, verkossa toimiva asiakaspalvelu sekä tuotteiden vertailun helppous. Verkkokaupan huonoihin puoliin hän listaa esimerkiksi mahdollisen hitaan toimituksen, virheet tuotteiden toimituksessa tai sen, ettei tilattu tuote vastaa verkkokaupassa saatua mielikuvaa. Verkkokauppaa perustettaessa onkin otettava huomioon asiakaspalvelu, riittävät tuotekuvaukset, riittävä logistiikka ja tuotepalautusten suorittaminen. Goldsmithin ja Laffertyn [2002] mukaan verkkokaupankäynti on parhaimmillaan nopeampaa, halvempaa ja hauskeempaa kuin tavanomainen kaupassa käyminen. Eli mitä enemmän kuluttaja asioi verkkokaupoissa ja mitä positiivisempia kokemukset verkossa asioimisesta on, on todennäköisyys tulevaisuuden verkkokaupalle suurempi. Jotta yritys hyötyy verkkokaupasta mahdollisimman paljon, tulee verkkokaupankäynti useiden toimijoiden kokonaisuutena ottaa huomioon heti alusta alkaen ja laatia liiketoimintastrategia sen mukaisesti.

Verkkokaupan perustamisessa huomioon otettavia seikkoja ovat esimerkiksi verkkokaupan kohderyhmä ja suuntautuuko tarjonta kotimaan markkinoille vai ulkomaille. Verkkokaupan toteuttamiseen on tarjolla useita erilaisia vaihtoehtoja. Verkkokaupan perustava yritys voi itse ryhtyä toteuttamaan verkkokauppaansa, tilata sen räätälöitynä ratkaisuna ulkoiselta taholta tai ostaa valmiin, maksullisen verkkokauppapohjan niitä tarjoavalta yritykseltä.

2.2. Verkkokaupan toteutus

Verkkokaupan perustamisessa on tehtävä päätös, hyödynnetäänkö *avoimeen lähdekoodiin* (open source) perustuvaa alustaa vai niin sanotusti suljettua, *omisteista alustaa* (proprietary software). Avoimella lähdekoodilla ei ole yhtä standardoitua määritelmää, mutta avoimen lähdekoodin ohjelmistojen ja niiden lähdekoodien tulee olla vapaasti jaossa internetissä. Mikäli ohjelmistoon liittyy lisensointia, ei lisensointi saa estää korjaustiedostojen julkaisua ja levittämistä [COSS 2009]. Avoimeen alustaan perustuva verkkokauppa mahdollistaa esimerkiksi maksupalvelu- tai logistiikkapalvelumoduulien lisäämisen verkkokauppaan itse tai toimittajan toimesta. Moduulit voivat olla ilmaisia tai maksullisia. Hyviä puolia ovat kustannustehokkuus ja muiden alustaa käyttävien toimijoiden julkaisemat lisäosat ja tuki. Huonoja puolia ovat tarve hosting-palvelulle, jolla tarkoitetaan ulkopuolisen tahon tarjoamaa palvelinta, joka mahdollistaa verkkosivujen olemassaolon internetissä. Toinen huonoksi laskettava puoli voi olla teknisen osaamisen vaatimus, jos verkkokauppaa perustava taho ei omaa riittävää ymmärrystä ja osaamista teknisestä IT-puolen toteuttamisesta. Avoimeen alustaan perustuvia verkkokauppapohjia ovat esimerkiksi Magento, Opencart ja WooCommerce [Paytrail 2017]. Voidaan siis tiivistää, että avoimeen lähdekoodiin perustuvan verkkokaupan valitseminen vaatii erityisesti teknistä osaamista, kokonaisvaltaista vastuuta verkkokaupan toiminnasta ja aktiivista ylläpitotoimintaa.

Suljettua lähdekoodia voidaan pitää avoimen lähdekoodin vastakohtana, sillä tällöin ohjelmiston toiminta on liikesalaisuus, eikä ohjelmiston käyttäjällä ole pääsyä lähdekoodiin. Tällöin ohjelmiston lähdekoodiin ei ole mahdollista tehdä muutoksia, eikä muutoksia saa julkaista [COSS 2009]. Suljettuja verkkokauppa-alustoja ovat esimerkiksi MyCashFlow, Shopify ja Vilkas. Suljettujen ohjelmistojen etu on helppous. Tällaisten palveluiden käyttäjien ei tarvitse itse vastata ohjelmistotason päivitystoimenpiteistä eikä ylläpito- tai hosting-palveluista. Verkkokaupan ulkoasu riippuu toimittajan valikoimasta ja erilaisien maksullisten palvelupakettien avulla verkkokauppaa voi laajentaa palvelun rajoitusten puitteissa [Paytrail 2017].

Kun yritys ryhtyy toteuttamaan verkkokauppaa itsenäisesti, tulee yrityksen arvioida yrityksen sisäinen osaamistaso toteuttamiseen. Tee-se-itse - verkkokauppatoimitus on mahdollinen silloin, kun yrityksessä joko on tai yritykseen rekrytoidaan teknisesti tarpeeksi pätevää henkilöstöä. Erilaisien teknologioiden tarjonta on nykypäivänä laaja, joten avoimeen lähdekoodiin perustuva, itse rakennettu verkkokauppa ei ole poissuljettu vaihtoehto. Riskinä kuitenkin on, ettei käytettävyyttä tai teknistä toimivuutta huomioida, dokumentoida tai testata riittävästi, jolloin pahimmassa tapauksessa yrityksen liiketoi-

minta kärsii [Korhonen 2016]. Puutteellinen osaaminen saattaa hidastaa verkkokaupan perustamista merkittävästi, ja tarvittavan konsultaation hankkiminen voi aiheuttaa lisäkustannuksia. Parhaimmassa tapauksessa ja osaavissa käsissä verkkokaupan rakentaminen voi kuitenkin olla hyvä vaihtoehto sellaiselle yritykselle, jolla on esimerkiksi poikkeuksellinen toimintaperiaate tai jatkuvasti kehittyvä markkina-alue. Tällöin muutoksia voidaan tehdä nopeasti sisäisten tarpeiden mukaisesti.

Toinen vaihtoehto on palkata ulkoinen toimittaja, joka rakentaa verkkokaupan tilaavan yrityksen toiveiden mukaisesti alusta alkaen avoimeen lähdekoodiin perustuen. Tällöin ulkoisen toimittajan kanssa voidaan sopia ylläpidosta, sisällöntuotosta ja esimerkiksi maksupalveluiden liittamisestä verkkokauppaan. Riskinä ulkoisen toimijan valinnassa on luotettavuus ja tarjonnan laajuus; miten valita luotettava ja kustannusarvioissa pysyvä toimija? Riski voi olla myös kielimuurissa, mikäli toimittaja on ulkomaalainen. Parhaimmassa tapauksessa räätälöity, ulkoisen toimijan toimittava verkkokauppa on nopea ja edullinen ratkaisu.

Valmiit, suljettuun alustaan perustuvat verkkokaupat toimivat avaimet käteen -periaatteella. Kun sopimus toimittajan kanssa on tehty, huolehtii toimittaja teknisen ylläpidon. Tällöin verkkokauppias voi vain ryhtyä lisäämään tuotteitaan verkkokauppaan. Etu varsinkin kotimaisissa valmiita verkkokauppapohjia tarjoavissa yrityksissä on se, että esimerkiksi erilaiset lisäosat eli moduulit voivat olla jo valmiiksi toteutettuja, jolloin maksupalvelun liittäminen verkkokauppaan saattaa tapahtua parilla napin painalluksella, jolloin verkkokauppiiaan ei tarvitse ymmärtää verkkokaupan teknisestä toiminnasta yhtä paljon kuin avoimen lähdekoodin toteutuksissa.

Vaikka tässä mainitsemani tavat onkin eroteltu toisistaan, ei se tarkoita sitä, että rajanveto erilaisiin tapoihin perustaa verkkokauppa olisi niin jyrkkä -toteutusmahdollisuuksia näitä kaikkia vaihtoehtoja yhdistellen löytyy varmasti.

3. Verkossa maksaminen

Verkkomaksamisella tarkoitetaan sähköistä tapaa maksaa verkossa ostetuista hyödykkeistä [TIEKE 2018]. Verkkokaupan tulee tarjota riittävän laaja maksutapavalikoima, jotta asiakkaan ostopäätös ei kariudu halutun maksutavan puuttumiseen. Käyn läpi verkkokauppiiaan juridisia velvoitteita maksutapoja koskien myöhemmin luvussa kuusi.

Globaalisti korttimaksaminen on kasvattanut vuodesta 2010 nopeimmin suosiotaan, mikä esimerkiksi Yhdysvalloissa näkyy shekkien käytön vähentymisenä. Pankkikorteilla on laajin käyttöaste (45,7%) [Khan *et al.* 2017], mutta

Suomessa pankkikorttien sijaan suositaan pankkipainikkeiden käyttämistä. Suomen pankin [2017] mukaan verkkomaksupainikkeita käytettiin vuonna 2016 lähes 77,5 miljoonaa kertaa yhteensä miltei 5,5 miljardin euron edestä, mikä oli 18% enemmän kuin sitä edellisenä vuonna. Etäkorttimaksujen määrä vuonna 2016 oli Suomessa lähes 53 miljoonaa kappaletta ja niitä maksettiin 2,7 miljardin euron edestä. Kotimaisessa verkkokaupankäynnissä hyödynnetään usein sähköistä tunnistautumista, kuten verkkopankkitunnuksia (TUPAS eli ”tunnistuspalvelu standardi”) tai mobiilivarmennetta. Sähköistä tunnistautumista voidaan verrata henkilöllisyystodistuksen näyttämiseen.

3.1. Verkkomaksutavat

Maksamisen ratkaisut kehittyvät jatkuvasti. Esittelen seuraavassa erilaisia Suomessa käytettäviä maksamistapoja verkossa, eli tilisiirto, erilliset pankkipainikkeet, luottokorttimaksutavat, lasku- ja osamaksupalvelut, kryptovaluutta ja mobiilimaksaminen. Esittelemäni verkkomaksutavat eivät kata kaikkia mahdollisia verkossa maksamisen muotoja, mutta pyrin antamaan kokonaiskuvan erityisesti kotimaisissa verkkokaupoissa käytetyistä maksutavoista.

3.1.1. Tilisiirto

Suomessa perinteinen tilisiirto on erittäin suosittu tapa käydä kauppaa edelleen, sillä luottamus pankkilaitoksiin on korkea. Erotan tilisiirron ja pankkipainikkeen maksutapoina toisistaan. Tilisiirrolla tarkoitan asiakkaan toimenpiddettä, jossa tämä siirtyy muualta kuin verkkokaupan maksusivulta omaaloitteisesti omaan verkkopankkiinsa ja suorittaa tilinumeron avulla verkkokauppiiaan kanssa sovitun maksusuorituksen omassa verkkopankissaan. Tilisiirto on ongelmallinen ja jopa vanhentunut maksutapa verkkokaupankäynnissä, sillä suoritus näkyy vastaanottajan tilillä vasta muutaman päivän pankkiviiveen jälkeen. Asiakas saattaa myös viivästellä maksamisen kanssa, mikä haittaa tuotteen lähettämistä ja varastosaldojen ylläpitoa. Mikäli maksaminen sujuu ongelmitta, mutta tilauksesta on tehtävä rahapalautus esimerkiksi tilauksen peruuttamisen vuoksi, joutuu verkkokauppias manuaalisesti palauttamaan suorituksen asiakkaan pankkitilille. Tilinumero joudutaan pyytämään asiakkaalta erikseen, sillä se ei näy suorituksessa verkkokauppiiaan tilille. Maksamisesta ja palautuksista tulee monimutkaisia, eikä erehdyksiltäkään voida välttyä. Pahimmassa tapauksessa jompikumpi kaupan osapuoli siirtää varat esimerkiksi inhimillisen erehdyksen vuoksi väärälle tilille.

3.1.2. Pankkipainikkeet

Kuten aiemmassa alakohdassa totesin, on suomalaisten luotto pankkitunnistautumisella tapahtuvaan maksamiseen suuri. Tämän vuoksi pankkipainikkeet ovat Suomen suosituin tapa maksaa verkkokaupassa. Pankkipainikkeilla tarkoitan verkkokaupan maksusivulla olevia, pankin logolla varustettuja painikkeitä, joita käyttämällä asiakas ohjataan hetkellisesti oman pankkinsa järjestelmään hyväksymään maksusuoritus, ja ohjataan sen jälkeen automaattisesti takaisin verkkokauppaan. Vaikka pankkipainikkeet ovatkin suosittu maksutapa Suomessa, ei niissä tapauksissa, joissa verkkokauppias on petollinen, menossa konkurssiin tai jostain muusta syystä jättää tuotteen toimittamatta, suoraa pankkimaksua ole mikään taho velvollinen palauttamaan, toisin kuin luotollisten maksutapojen ollessa kyseessä.

Mikäli verkkokaupassa ei ole käytössä maksunvälityspalvelua, joutuu verkkokauppias tekemään erilliset sopimukset jokaisen pankin kanssa erikseen. Tällöin esimerkiksi kuukausi- ja transaktiohinnat, avausmaksut ja muut kustannukset saattavat käydä hyvin kalliiksi. Lisäksi asiakasta saattaa hämmentää rönsyilevä hinnasto, kun eri pankkien transaktiomaksut poikkeavat toisistaan. Jokainen pankkipainike tulee upottaa erikseen maksusivulle mahdollisesti hyvin paljon toisistaan poikkeavilla teknisillä toteutuksilla. Riippuen verkkokauppiastaan verkkokauppa-alustasta, saattaa ongelmia tulla myös yhteensopi- vuus- ja skaalautuvuuskysymyksissä.

3.1.3. Luottokorttimaksutavat

Globaalisti luottokorttimaksaminen on suosituin maksutapa verkossa. Luottokorttimaksu autentikoidaan korttinumerolla, nimellä ja voimassaoloajalla. Tietoturvan vahvistamiseksi luottokorttiyritykset ovat kehittäneet myös täydentäviä järjestelmiä kuten Verified by Visa ja MasterCard SecureCode. [Khan *et al.* 2017] Suomalaisella luottokortilla maksettaessa suoritetaan myös TUPAS-tunnistautuminen. Luottokorttimaksutavat ovat suositelluin verkkomaksutapa asiakkaalle, mutta ei kustannustehokkain verkkokauppiaille. Esimerkiksi Visan kortinhaltijasuoja kattaa internethuijaukset, toimittamatta jääneet tilaukset, väärät tuotteet, vahingoittuneet tuotteet ja matkatoimiston konkurssin ennen matkaa tai sen aikana [Visa 2018]. Mikäli asiakas maksaa tuotteensa luottokortilla ja kiistää veloituksen edellä mainituista syistä, suorittaa luottokorttiyhtiö oman reklamaation tarkastamisen jälkeen hyvityksen asiakkaalleen, jonka jälkeen perii saatavan verkkokauppialta. Luottokorttipetosten selvittämiseen käytetään esimerkiksi *tiedonlouhintaa* (data mining), statistiikkaa ja tekoälyä [Al-Furiah and Al-Braheem 2009].

Yleensä luottokorttimaksutapasopimukseen kuuluu transaktiokustannusten lisäksi provisioprosentti asiakkaan suorittamasta ostoksesta, jolloin esimerkiksi pienelle verkkokauppaa pitävälle yritykselle kustannukset voivat olla niin korkeat, ettei niiden tarjoaminen maksuvaihtoehdoksi verkkokaupassa ole kannattavaa [Khan *et al.* 2017]. Kansainvälisiä luottokorttiyhtiöitä ovat esimerkiksi Visa ja Mastercard.

3.1.4. Lasku- ja osamaksupalvelut

Lasku- ja osamaksupalvelut tarkoittavat yrityksiä, jotka tarjoavat asiakkaalle mahdollisuuden maksaa ostamansa tuotteen laskulla tai osamaksulla. Osamaksupalvelu suorittaa oman riskianalyysinsä asiakkaasta, jonka perusteella se päättää, myönnetäänkö kyseiselle asiakkaalle laskuvaihtoehto. Mikäli se myönnetään, ostaa osamaksupalvelu laskun ja suorittaa asiakkaan ostoksen summan verkkokauppiaille [Collector 2018]. Tämän jälkeen osamaksupalvelu laskuttaa asiakasta hänen valitsemansa sopimuksen mukaisesti. Osa lasku- ja osamaksupalveluista tarjoaa laskumahdollisuuden henkilöasiakkaan lisäksi myös yritykselle. Suomessa toimivia osamaksu- ja laskupalveluita ovat esimerkiksi Collector Bank, Jousto, Klarna ja Svea Ekonomi. Myös osa suomalaisista pankeista tarjoaa verkkokauppaan laskutusvaihtoehdon, kuten OP Osta laskulla.

3.1.5. Kryptovaluutat

Kryptovaluutat ovat teknologiseen varmentamiseen eli kryptografiaan perustuvia virtuaalivaluuttoja, joiden arvo perustuu kryptovaluuttapörssiin eikä keskuspankkiin. Kryptovaluuttapörssillä tarkoitetaan palvelua, jossa eri kryptovaluuttoja (esimerkiksi Bitcoin, Ethereum ja Ripple) voi vaihtaa toisiinsa tai jopa vakiintuneeseen valuuttaan, kuten euroihin [Leppänen 2017]. Bitcoin on kuuluisin kryptovaluutta, jonka loi Satoshi Nakamoto -nimimerkkiä käyttänyt taho vuonna 2008 ja joka julkaistiin tammikuussa 2009. Bitcoin perustuu avoimeen lähdekoodiin. Tiivistettynä bitcoin toimii seuraavasti: Jokainen bitcoin kuuluu *bitcoin-osoitteelle* (Bitcoin address), joka on verrattavissa pankkitileihin. Jotta bitcoineja voidaan käyttää, tarvitaan maksamiseen *digitaalinen allekirjoitusalgoritmi* (Digital Signature Algorithm, DSA), joka muodostaa avainparin julkisesta ja salaisesta avaimesta. Tällöin jokaisella bitcoinin käyttäjällä on bitcoin-osoite ja salaista avainta käytetään omistajan toimesta bitcoineilla kaupankäymiseen. Jotta salaiset avaimet pysyvät tallessa, käytetään *bitcoin lompakkoja* (Bitcoin wallet) niiden säilymiseen. Tällöin lompakkoa käyttävä pystyy seuraamaan, paljonko bitcoineja hänelle kuuluu ja näkee tiedot salaisista avaimista, joilla siirtoja on tehty. [Li *et al.* 2017.] Suomen Pankin [2014] mukaan bitcoin ei

nykymuodossaan täytä virallisen rahan tai maksupalvelulain määrittelemän maksuvälineen kriteerejä, eivätkä viranomaiset hallinnoi sitä. Verkkokauppaan kryptovaluuttamaksamisen voi mahdollistaa helposti integroimalla alustaan alan palvelu, kuten esimerkiksi BitPay tai CoinPayments. BitPayta käytettäessä verkkokauppaan ei tarvitse käsitellä bitcoineja kirjanpidoissa, sillä tilitykset tapahtuvat euroina. Bitcoinin valuuttakurssin vaihtelu jää BitPayta käyttämällä asiakkaan ja BitPayn vastuulle, jolloin verkkokauppias ei kanna kurssin vaihteluun liittyviä riskejä [BitPay 2018].

3.1.6. Mobiilimaksaminen

Mobiilimaksamisen suosio eri muodoissaan on kasvussa. Mobiilimaksaminen liittyy maksamisen helpottamiseen; on toivottavaa, että asiakas päätyy verkkokaupassa tilaamiseen ja maksamiseen mahdollisimman helposti. Mobiilimaksamiseksi nimitetään kaikkia sellaisia maksutapoja, joissa asiakkaan oma matkapuhelin toimii maksuvälineenä [Smeds 2017]. Tällöin verkkokaupassa maksamistilanteessa maksaminen hyväksytään applikaatiosta, jolloin ostos veloituu applikaatioon liitetystä pankkitililtä. Mobiilimaksamiseen on olemassa erilaisia applikaatioita, kuten esimerkiksi Danske Bankin MobilePay ja OP Ryhmän Pivo. Danske Bankin mukaan MobilePayn oli vuonna 2016 ladannut jo 320 000 käyttäjää [Vänskä 2016]. Mobiilimaksaminen voi tarkoittaa myös maksunumeron soittamista tai viestin lähettämistä, jolloin ostos veloitetaan puhelinlaskussa, ja ostokseen lisätään operaatiopalkkio sekä palvelun tarjoajan kustannukset.

3.2. Verkkomaksamisen yhdistyminen kivijalkaan tulevaisuudessa

Kuten edellä olen tuonut ilmi, on verkkomaksamisen vaihtoehtoja lukuisia. Kehittyvä trendi on verkkokaupan maksutapojen laajentaminen kivijalkaliikkeisiin käytettäväksi. Ruotsalainen Klarna on lanseerannut Ruotsissa mallin, jossa kivijalkaliikkeessä asioiva asiakas voi valita maksutavakseen Klarnan, jonka jälkeen asiakas voi mobiililaitteellaan maksaa ostoksensa haluamallaan maksutavalla [Eisto 2017]. S-ryhmä on ilmoittanut ottavansa käyttöön MobilePayn omissa liikkeissään syksyllä 2018, jolloin MobilePaysta ennustetaan tulevan mobiilimaksamisen markkinajohtaja Suomessa [Mäntylä 2018].

4. Maksunvälityspalvelut

Esittelen seuraavassa maksunvälityspalveluiden merkityksen yleisesti sekä niiden ominaisuuksia. Maksunvälitysjärjestelmä eli *maksusiltapalvelu* (payment gateway) on verkkopohjainen palvelu, joka integroidaan verkkokaupan maksusivuun [TIEKE 2017] tai asiakas ohjataan maksupalvelun erilliselle mak-

susivulle. Maksunvälityspalvelua kutsutaan *kolmanneksi osapuoleksi* (Trusted Third Party) ja sitä käytetään verkkokaupankäynnissä turvallisempien transaktioiden suorittamiseen. Maksunvälityspalvelu toimii linkkinä asiakkaiden, verkkokauppioiden ja maksutavan tarjoajien välillä. Maksunvälityspalvelu on vastuussa transaktioiden nopeudesta, luotettavuudesta ja turvallisuudesta [Pande *et al.* 2014.] Maksunvälityspalvelussa verkkokaupan asiakkaan maksu siirtyy heidän järjestelmäänsä, josta se tilitetään verkkokauppiaille palvelusopimuksen mukaisella viiveellä.

Yhdysvaltojen käytetyin maksunvälityspalvelu on EBayn omistama PayPal. Kiinalainen Alibaba.com, joka on suunnattu myös henkilökuluttajille, on maailman suurin *yritykseltä yritykselle* (business to business, B2B) palveluita tuottava verkossa toimiva kaupankäyntijärjestelmä. Suuri osa Alibaban käyttäjistä on Yhdysvalloista. [Kong 2016.] Alibaba käyttää maksunvälitykseen omistamaansa Alipayta. Euroopan alueella maksunvälityspalvelut vaihtelevat maittain johtuen erilaisista lainsäädännöistä koskien verkkokaupankäyntiä, mutta esimerkiksi PayPalia voi käyttää jo suurimmassa osassa Euroopan maita ja koko muuta maailmaa. Oman sivustonsa mukaan PayPal on käytettävissä yli 200 maassa ja se tukee 25 valuuttaa.

Maksunvälityspalvelua valitessaan verkkokauppiiaan tulee ottaa huomioon seuraavia seikkoja: valittavan maksunvälityspalvelun maksutapavalikoiman laajuuden riittävyys suhteessa verkkokaupan markkina-alueeseen, transaktiokustannukset, provisiot, kuukausimaksut, turvallisuus, tuki ja verkkokaupan kannalta riittävät ominaisuudet [Khan *et al.* 2017]. Maksupalvelulta haluttavia ominaisuuksia voivat olla esimerkiksi verokantaerittelyt tilitysraporteissa ja valmiit integraatiot erilaisille verkkokauppapohjille.

4.1. Maksulaitoslaki

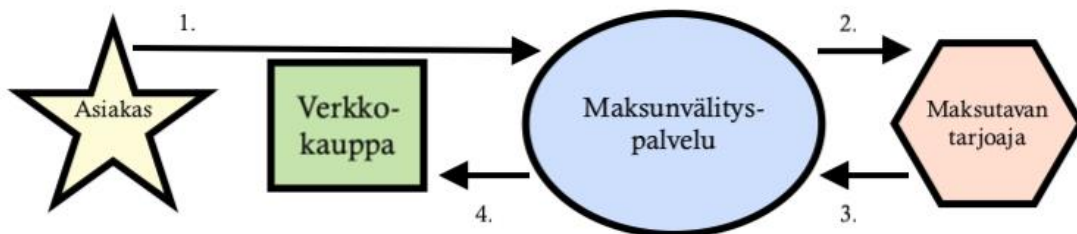
Maksunvälityspalvelulla on oltava Suomessa Finanssivalvonnan myöntämä maksulaitoslupa. Finanssivalvonta on rahoitus- ja vakuutusvalvontaviranomainen, joka valvoo käytännössä kaikkea finanssitoimintaa Suomessa. Näitä ovat esimerkiksi pankit, vakuutus- ja eläkeyhtiöt, muut vakuutuslalla toimivat yritykset, pörssi ja rahastoyhtiöt. Finanssivalvonnan toimintaa säätelee laki finanssivalvonnasta. [Finanssivalvonta 2017.] Maksulaitoslakiin on kirjattu, että maksulaitoksella tulee olla esimerkiksi tehokkaan riskienhallinnan mahdollistava hallinto ja riittävät riskienhallintajärjestelmät. Lisäksi maksulaitoksen tulee ilmoittaa poikkeamista ja petoksista ilman viivytystä Finanssivalvonnalle ja mikäli maksupalvelun käyttäjälle (eli esimerkiksi verkkokauppiaille) tulisi tällaisia taloudellisia ongelmia, myös tälle on ilmoitettava. Maksulaitos on myös velvollinen ilmoittamaan maksupalvelun käyttäjälle keinot, joilla taloudellisia

tappioita ja muita haitallisia vaikutuksia tapahtuneesta johtuen voidaan vähentää [Maksulaitoslaki 2017/19 b].

Jokaisen finanssialan toimijan tulee suomalaisen lainsäädännön mukaisesti tuntea ja tunnistaa asiakkaansa ennen asiakassuhteen aloittamista. Tutkielmani kontekstissa tämä laki velvoittaa siis maksutavan tarjoajia sekä maksunvälityspalveluita tunnistamaan asiakkaansa ennen asiakassuhteen aloittamista. Yksinkertaistettuna tämä tarkoittaa henkilöllisyyden tarkastamista.

4.2. Maksunvälityspalvelun toiminta pähkinänkuoressa

Maksunvälityspalvelun käyttämisestä on verkkokauppiaille useita etuja. Sen sijaan, että verkkokauppaan täytyisi integroida usea maksupainike ja eri maksutapojen käyttämisestä joutuisi neuvottelemaan niiden tarjoajien kanssa erikseen, saa verkkokauppias yhdellä sopimuksella ja yhdellä integraatiolla kattavan maksutapavalikoiman verkkokauppaansa. Suurella osalla Suomessa toimivista valmiita verkkokauppapohjia tarjoavista yrityksistä on valmis integraatio kotimaassa toimiviin maksunvälityspalveluihin. Suomessa toimivia maksunvälityspalveluita ovat esimerkiksi Paytrail, Checkout, Maksuturva ja Klarna.



Kuva 1. Maksamisen prosessi maksunvälityspalvelua käyttäen

Kuva 1 esittää maksamisen prosessia maksunvälityspalvelua käyttäen. Verkkokauppa, maksunvälityspalvelu ja maksutavan tarjoaja hyödyntävät kaikki erillisiä tietojärjestelmiään, ja kuva havainnollistaa prosessia järjestelmien välillä. Ensimmäisessä vaiheessa verkkokaupan asiakas on löytänyt verkkokaupasta haluamansa tuotteet, täyttänyt tarvittavat tiedot toimitusta varten ja siirtynyt maksamaan ostoksensa. Asiakas valitsee verkkokaupassa olevista maksunvälityspalvelun maksupainikkeista haluamansa maksutavan, jolloin tieto transaktiosta kirjautuu maksunvälityspalvelun järjestelmään. Toisessa vaiheessa maksunvälityspalvelu ohjaa asiakkaan maksutavan tarjoajan suorittamaan tarvittavat toimenpiteet. Toimenpiteet ovat esimerkiksi tunnistautuminen pankkitunnuksilla maksun hyväksymiseksi, jolloin asiakas ohjautuu maksutavan tarjoajan järjestelmään toimenpiteen suorittamisen ajaksi. Kolmannessa vaiheessa

maksutavan tarjoaja on hyväksynyt tai hylännyt (esimerkiksi tilillä ei ole katetta) maksupyynnön, jolloin tieto tapahtuman onnistumisesta tai epäonnistumisesta kirjautuu maksunvälityspalveluun, joka välittää tiedon verkkokauppaan. [Khan *et al.* 2017.] Tällöin verkkokauppa antaa asiakkaalle tiedon tapahtuman onnistumisesta tai epäonnistumisesta. Mikäli tapahtuma onnistuu, verkkokauppa huolehtii tämän jälkeen tuotteen toimittamisesta. Maksu itse ohjautuu maksutavan tarjoajalta maksunvälityspalvelulle ja sieltä verkkokaupan tilille valitun tilityssyklin mukaisesti.

Maksunvälityspalvelu neuvottelee sopimukset eri maksutapojen tarjoajien kanssa, joita olen aiemmassa kappaleessa eritellyt. Maksunvälityspalvelu hoitaa ristiriitatilanteissa kommunikaation maksutavan tarjoajien kanssa, jolloin verkkokauppiiaan ei tarvitse itse olla yhteydessä itse maksutavan tarjoajaan.

Maksunvälityspalveluilla on käytössään verkkokauppioiden kirjautumispalvelu, josta verkkokauppias pystyy valitsemaan käytössään olevat maksutavat, seuraamaan maksuliikennettään, suorittamaan tarvittaessa maksupalautuksia ja seuraamaan muutenkin rahaliikennettä verkkokaupassa. Osa maksunvälityspalveluista tarjoaa myös mahdollisuuden tarkastella, missä vaiheessa ostotapahtumaa maksaminen on jäänyt kesken, jotta verkkokauppias voi tarvittaessa tarkistaa verkkokauppiansa toimintaa tai arvioida omaa hinnoitteluaan kilpailijoihinsa verrattuna.

Maksunvälityspalvelut tarjoavat verkkokauppiaille erilaisia mahdollisuuksia vastaanottaa kirjanpitokelvollisia tilitysraportteja myynneistään, mikä helpottaa luonnollisesti verkkokauppiiaan yrityksen kirjanpitoa. Maksunvälityspalvelusta riippuen myös eri alv-prosenteilla olevien tuotteiden erittely raportissa voi olla mahdollista. Osa palveluista tarjoaa yritysten kirjanpitäjille henkilökohtaiset tunnukset palveluun, jolloin kirjanpitäjä voi suoraan hallintapaneelista hakea kirjanpitoon tarvittavia tietoja kaupankäynnistä. Tilitysraporteissa näkyy kootusti verkkokaupan rahaliikenne käytetystä maksutavasta huolimatta. [Checkout 2018.]

Verkkokauppias voi myös valita itselleen sopivia tilityssyklejä. Kun verkkokaupasta ostetaan, siirtyä ostoksesta kertyneet varat ensin maksunvälityspalvelun asiakasvaratileille, joista varat tilitetään verkkokauppiiaan yrityksen tilille halutulla frekvenssillä. [TIEKE 2017.] Riippuen verkkokaupan volyymeistä ja käytännöistä voi esimerkiksi osalle verkkokaupoista olla käytännöllistä saada tilitykset päivittäin, kun taas toisille kerran kuussa tehtävä tilitys sopii parhaiten.

5. Logistiikka

Mikäli asiakas ei saa tilaamaansa tuotetta tyydyttävässä ajassa, tyydyttävässä kunnossa tai ei pysty palauttamaan tuotetta helposti takaisin verkkokauppiaille, todennäköisyys asiointille samassa verkkokaupassa uudelleen pienenee merkittävästi [Bridges *et al.* 2006]. Kun ajatellaan verkkokaupankäyntiä monen palvelun kokonaisuutena, on logistiikan toimivuus merkittävää, sillä asiakas itse ei välttämättä osaa erotella kaupanteon prosessin eri osatekijöitä toisistaan. Logistiikkapalveluita ei luonnollisesti vaadita verkkokaupoissa, jotka myyvät elektronisia hyödykkeitä, jotka toimitetaan asiakkaalle sähköpostitse tai muilla tavoin verkon ylitse. Tällöin verkkokauppiiaan on huolehdittava siitä, että myytävästä sähköisestä hyödykkeestä (esimerkiksi netistä ostettava ohjelmistosta tai käyttöoikeudesta) on riittävät yhteensopivuuskuvaukset. Verkkokaupan sopimusehdoissa tulee olla listattuna käytössä olevat toimitustavat. Käsittelyn toimittamiseen liittyviä ehtoja luvussa 6.

5.1. Tuotteiden toimittaminen

Verkkokaupalle toimivat toimitukset ovat elinehto. Verkkokaupan logistisiin ratkaisuihin kuuluvat toimitukseen ja palautukseen käytettävät palvelut, myyntiartikkeleiden tilaus ja varastointi sekä varastohallintajärjestelmät. Optimaalisimmassa ratkaisussa verkkokauppa ja varastohallintajärjestelmä kommunikoivat keskenään, jolloin esimerkiksi tuotteiden määrät voidaan päivittää verkkokauppaan varastohallintajärjestelmän antamien tietojen perusteella. Jotta verkkokauppa voi toimittaa tuotteitaan kannattavasti, tulee tämän tehdä sopimus haluamansa kuljetuspalvelun kanssa. Suomessa toimivia kuljetus- logistiikkapalveluita ovat esimerkiksi Posti, Matkahuolto, PostNord, DB Schenker ja DHL. Palveluiden kuljetuskustannukset riippuvat pääsääntöisesti tuotteen koosta ja painosta.

Verkkokauppoihin on nykyään integroitavissa myös niin sanottuja pakettinlähetykspalveluita, jotka toimivat lähes samalla logiikalla, kuin maksunvälityspalvelut. Pakettinlähetykspalvelut ovat suhteellisen uusi ilmiö, mutta peruseriaatteenaan ne tarjoavat yhdellä sopimuksella useamman kuljetusyrityksen palvelut sekä kirjautumisportaalien, josta toimituskortteja voi tulostaa. Mikäli verkkokauppaintegraatio on olemassa, voi toimituskortin tulostaa myös verkkokaupan hallintapaneelista. Tällaisen palvelun etuna on, kuten maksunvälityspalvelulla, yhdellä sopimuksella saatava laaja palvelukirjo. Kuljetuksien hinnat saattavat tulla pakettinlähetykspalvelua käyttämällä huomattavasti edullisemmaksi. Tällaisia pakettinlähetykspalveluita ovat Suomessa esimerkiksi Paketikauppa, Shipit ja ShipFunk.

5.2. Logistiikan haasteet ja tuotepalautukset

Suomi on logistisesti hankala maa, sillä suuret etäisyydet, sijainti Euroopan pohjoisreunalla ja harva asutus aiheuttaa hankaluuksia toimittamiselle. Tällöin esimerkiksi logistiikkakeskusvarastojen sijoittaminen on hankalaa, eikä houkuttele esimerkiksi kansainvälisiä yrityksiä sijoittamaan keskuslogistiikkavarastojaan Suomeen. Lisäksi logistiikkapalveluiden markkinat eivät ole suuret, ja pitkien kuljetusetäisyyksien ja suurempien varastointikustannuksien vuoksi logistiikkayritysten toimintaan sitoutunut pääoma on suuri. Maantieteellisestä sijainnista johtuen myös nopeat toimitukset ulkomaille tuottavat hankaluuksia, sillä toimituksia ei voida suorittaa pintakuljetuksina (esimerkiksi autokuljetuksina), vaan tuote täytyy lähettää lentokonetoimituksena. [Okkonen ja Lukka 2004.]

Tuotepalautukset ovat olennainen osa verkkokaupankäyntiä, ja niitä koskee etämyyntiin laissa säädetty palautusoikeus. Erityisesti tuotepalautuksia on odotettavissa vaatealan verkkokaupoissa. Verkkokauppojen tuotepalautusten logistiikka eli *käänteinen logistiikka* tai *paluulogistiikka* (reverse logistics), johon Kokkinakin ja muiden [1999] mukaan kuuluu tuotepalautusten lisäksi myös esimerkiksi kuljetusmateriaalien takaisinkerääminen ja uudelleenkäsittely. Yrityksellä voi nimittäin olla esimerkiksi toimituksia, joiden pakkauslaatikoita käytetään uudelleen, kuten päivittäistavarakauppoihin tuotavat kovamuoviset lihatuotteita sisältävät laatikot. EU:ssa on olemassa *tuottajavastuulakeja* (producer responsibility laws), jotka vaativat tuottajia kehittämään mahdollisimman uudelleenkäytettäviä materiaaleja ja tuotteita toimittamiseen [Kokkinaki *et al.* 2002].

Vaikka tuotepalautuksista saisi veloittaa asiakasta, harva verkkokauppa tekee niin, sillä se heikentäisi verkkokaupan asemaa markkinoilla. Kuitenkin ilmainen palauttaminen lisää turhaa tilaamista ja palauttamista varsinkin vaatteita myyvissä verkkokaupoissa. Vaikka palautuskuljetukset lisäävätkin logistiikka-alan kasvua, kuormittaa ilmainen palautusoikeus ympäristöä. Saarijärven [2015] mukaan palautusprosentit ovat kestävämmällä tasolla eettisesti, ekologisesti ja taloudellisesti. Tulevaisuudessa mahdollisesti asiakkaiden valvetuminen eettisesti sekä erilaiset digitaaliset ratkaisut helpottavat vähentämään verkkokauppojen palautusprosentteja.

EU-direktiivi, joka sallii palautusten kustannusten perimisen asiakkaalta, tuli voimaan vuonna 2014. Sitä aikaisemmin Suomessa oli useita tapauksia, joissa varsinkin pieniä verkkokauppoja ylläpitävät pienyritykset joutuivat lopettamaan toimintansa suurten palautusmäärien vuoksi. Toiminta muodostui pahimmassa tapauksessa kannattamattomaksi.

6. Verkkokaupan sopimusehdot

Käsittelen seuraavassa verkkokaupan lakivelvoitteisia säännöksiä, sillä verkkokauppiaan on otettava huomioon esimerkiksi kuluttajasuojalaki, jotta toiminnalle ei tulisi haittaa esimerkiksi erehdyksessä puutteelliseksi jääneistä sopimusehdoista. Pahimmassa tapauksessa puutteelliset sopimusehdot voivat johtaa siihen, että ristiriitatilanteessa asiakkaan kanssa vastuu kustannuksista tulee verkkokauppiaille, jos voidaan osoittaa sopimusehtojen olevan puutteelliset. Verkkokaupankäyminen kuuluu Kuluttaja- ja kilpailuviraston mukaan etämyyntiin mikä tarkoittaa sitä, etteivät myyjä ja ostaja ole kaupanteon aikana yhtä aikaa läsnä. Etämyynniksi Suomessa lasketaan myös posti- ja puhelinmyynti. Etämyyntiin on sovellettava lainmukaisia säännöksiä, jotka eroavat kivijalkaliikkeiden säännöksistä.

Verkkokaupassa tulee olla selkeästi nähtävissä sopimusehdot. Kuluttajasuojalaki on suurimmaksi osaksi pakottavaa lainsäädäntöä, joten sopimusehdoissa ei saa olla kohtia, jotka ovat kuluttajasuojalain vastaisia. Kuluttajasuojalain vastaiset ehdot ovat välittömästi mitättömiä. Ehtojen on oltava asiakkaan kannalta kohtuulliset, eli ehdot eivät saa esimerkiksi sälyttää vastuuta saadun tilauksen virheistä tai puutteista asiakkaalle. Asiakkaalla tulee olla mahdollisuus tutustua kaupan sopimusehtoihin kokonaisuudessaan ja tämän on saatava nähdä ostopäätökseen vaikuttavat sopimusehdot viimeistään tilausta tehdessään. Tämä kuuluu verkkokaupan lakisääteisiin tiedonantovelvollisuuksiin. [KKV 2006.] Valitettavasti on havaittavissa, ettei kaikissa kotimaisissa verkkokaupoissa ole vaadittavia materiaaleja asiakkaan saatavilla. Tämä voi johtua joko verkkokauppiaan huolimattomuudesta tai yksinkertaisesti tietämättömyydestä laissa määriteltyjä etämyynnin ehtoja kohtaan. Verkkokauppoihin, joista ehdot puuttuvat tai tiedot ovat muuten vajavaiset, on syytä suhtautua erittäin varauksellisesti, ja niistä voi tehdä ilmoituksen Kuluttaja- ja kilpailuvirastoon. Selkeät ja hyvin laaditut sopimusehdot ovat turva verkkokauppiaille itselleen, sillä tällöin riitatilanteissa voidaan parhaassa tapauksessa suoraan osoittaa, kelle vastuu kuuluu.

6.1. Maksu-, toimitus- ja muut ehdot

Kuluttajasuojalain mukaan verkkokaupan asiakkaalla tulee olla oikeus nähdä selkeästi tiedot tilauksestaan, kuten tuotteen pääominaisuudet, yrityksen nimi, yrityksen päätoimipaikan maantieteellinen osoite ja muut yhteystiedot, tuotteen kokonaishinta veroineen sekä erittelyt muista tilaukseen kuuluvista kustannuksista kuten toimituskustannuksista, peruuttamisoikeuteen liittyvät ehdot, määräajat ja menettelyt, korvausvelvollisuudet, virhevastuut ja kaikki maksamiseen liittyvät tiedot. Kun kyse on digitaalisen sisällön myymisestä,

tulee asiakkaan saada tieto palvelun toimintarajoitteista ja yhteensopivuuksista. Kaikkeen verkkokaupankäyntiin myös täytyy sisällyttää tieto kuluttajariitalautakunnan olemassaolosta, mikäli verkkokaupankäynnissä päädytään syystä tai toisesta riitatilanteeseen, joka ei ole muutoin ratkottavissa. Asiakkaan on saatava oikeusministeriön määrittelemän mallin mukainen peruuttamislomake.

Verkkokauppias saa itse päättää, mitä maksuvaihtoehtoja verkkokaupassa tarjotaan. On suositeltavaa, että verkkokaupasta löytyy esimerkiksi heti etusivulta linkki verkkokaupassa käytössä oleviin maksutapoihin. Eri maksutapojen ominaisuuksista ja toimintaperiaatteista tulee saada riittävän kattava kuvaus. Tuotteen hinta on olennainen osa tuotteen markkinointia, ja asiakkaan tulee voida luottaa siihen, että markkinoinnissa oleva hinta on oikea [KKV 2006]. Suomessa astui voimaan tammikuussa 2018 EU:n uusi maksupalveludirektiivi eli niin sanottu PSD2 (Payment Services Directive). Uuden lain myötä verkkokauppialla ei ole enää oikeutta periä maksutapalisää, eli verkkokauppias ei saa veloittaa lisämaksua erikseen, mikäli asiakas valitsee vaikkapa verkkokauppiaan kannalta kalliin maksutavan. Aiemmin verkkokauppias on voinut lisätä esimerkiksi transaktiomaksun ja maksupalveluntarjoajan kustannukset asiakkaan maksettaviksi. Uuden lain myötä myöskään maksunvälityspalvelut eivät enää voi tarjota palveluitaan käyttäville verkkokauppiaille mahdollisuutta maksutapalisän asettamiseen maksusivulle. [Majaniemi 2018.]

Verkkokaupankäynnissä, jossa toimituskulut muodostuvat esimerkiksi lähetettävän tuotteen painon perusteella, saa toimituskulut ilmoittaa erikseen. Jos toimitetussa tavarassa tai palvelussa on virhe eli tuote ei vastaa tilausta laadultaan, määrältään tai lajiltaan, on verkkokauppias vastuussa virheestä. Verkkokauppias ei voi vaatia asiakasta vastaamaan korjauskustannuksista.

6.2. Kaupan peruuttamisoikeus

Etämyynnin erityisominaisuuksiin kuuluu tehdyn kaupan peruuttamisoikeus, jolloin verkkokaupasta tilatusta tuotteesta pitää antaa verkkokaupalle peruutusilmoitus 14 päivän kuluessa tavarahan vastaanottamisesta tai sopimuksen tekemisestä [KKV 2014]. Tätä palautusmääräaikaa noudattavat myös Saksa ja Ruotsi. Peruutusehdot vaihtelevat maakohtaisesti, ja esimerkiksi EU-maissa palautusaika on vähintään seitsemän päivää. Verkkokaupalla ei ole oikeutta rajata tuotteitaan niin, että osa olisi peruuttamisoikeuden ulkopuolella, vaan tällaiset tuotteet on määritelty erikseen laissa. [Vuorinen 2014.] Esimerkiksi asiakkaan tilauksen mukaisesti toteutetut mittatilaustuotteet eivät kuulu peruutusoikeuden piiriin. Asiakas itse on vastuussa palautuskustannuksista, jos verkkokauppa ei ole ilmoittanut vastaavansa niistä. Jos verkkokaupassa ei ole ilmoitettu ennen tilauksen suorittamista, että asiakas vastaa palautuskustan-

nuksista tai jos verkkokauppa on ilmoittanut vastaavansa niistä, on palautus asiakkaalle ilmainen.

Verkkokauppiaan ei tarvitse suorittaa maksunpalautusta, ennen kuin palautettu tavara on palautunut takaisin verkkokauppiaalle tai asiakas on todentanut palautuksen esimerkiksi palautuskuitilla. Jos palautettu tuote on vahingoittunut, muuttunut tai puutteellinen, ei asiakkaalla ole oikeutta saada takaisin koko maksua. [KKV 2014.] Palautustapauksissa voi syntyä eriäviä mielipiteitä asiakkaan ja verkkokauppiaan välillä siitä, oliko palautettu tuote riittävän hyvässä kunnossa vai ei.

7. Yhteenveto

Olen tutkielmassani käsitellyt eri mahdollisuuksia verkkokauppojen toteuttamiseen, maksamista verkossa, maksunvälityspalveluita, logistiikkaa sekä juridisia kysymyksiä verkkokauppaan liittyen. Kuten olen osoittanut, on verkkokaupankäyminen useiden järjestelmien yhteistyötä ja jatkuvasti laajeneva kokonaisuus.

Mikäli haluaa päästä verkkokauppaa perustaessa helposti alkuun hyvin valmiilla kokonaisuudella, sopii tällöin käyttöön todennäköisesti parhaiten Shopifyn kaltainen kuukausimaksullinen verkkokauppapohja, jolloin manuaalista työtä verkkokaupan teknisen toiminnan rakentamiseen ei verkkokauppiaan toimesta tarvita. Huono puoli tosin on se, että muokattavuus riippuu täysin palvelun tarjoajasta ja todennäköisesti lisäominaisuuksista peritään maksu. Jos teknistä osaamista löytyy, on WooCommercen kaltainen avoimeen lähdekoodiin perustuva verkkokauppapohja erinomainen vaihtoehto, sillä muokkauksmahdollisuudet ovat laajat ilmaisten laajennusten ansiosta. WooCommerce esimerkiksi on tällä hetkellä maailman käytetyin verkkokauppa-alusta. Toisaalta riskinä on omien tai muun toteuttajan kykyjen yliarviointi, eikä osaaminen verkkokaupan ylläpitoon ja kehittämiseen välttämättä riitäkään. Tällöin verkkokaupan toimiessa poikkeuksellisesti, voi virheen etsintä tuottaa harmaita hiuksia.

Verkkomaksutapojen kategoria on laaja, ja hyvässä verkkokaupassa otetaan huomioon maksutapoja tarjotessa eri ikäluokkien ja asiakaskuntien tarpeet. Verkkomaksutapojen yhdistäminen kivijalkakaupankäyntiin on selkeä kehitystrendi, ja on odotettavissa, että erilaisia palvelukokonaisuuksia tälle sektorille tulee tulevaisuudessa enemmän. Useat kivijalkaliikkeet tarjoavat kassallaan osamaksu ja -laskutapoja jo nyt. Erityisesti erilaisia mobiilimaksamisen tapoja kehitetään ja tuodaan markkinoille jatkuvasti, jotta asiakas saadaan ostamaan hyödykkeitä mahdollisimman helposti ja vaivattomasti. Maksamisesta pyritään karsimaan pois sitä hidastavat tekijät.

Erilaiset toimituspalvelut ovat jo ottaneet huomioon kehityksessään verkkokaupankäynnin yleistymisen ja tuotepalautusten määrän. Silti tällä hetkellä kuljetusten kustannukset saattavat Suomessa kasvaa suuriksi erityisesti, mikäli asiakas asuu syrjäisillä seuduilla. Toivottavaa on, että Suomessakin päästäisiin edullisten ja nopeiden toimitusten pariin. Valitettavasti tuotetoimitusten ja -palautusten ympäristökuormitus on jo nyt huolestuttavalla tasolla, joten todennäköisesti logistiikan ala joutuu vastaamaan ympäristökysymyksiin ekologisemmilla kuljetusmetodeilla. Ekologisista syistä Ruotsi on ottamassa käyttöön lentoveron matkustajaliikenteeseen, jolloin asiakas maksaa veroa sitä enemmän mitä pidemmälle tämä lentää [Uosukainen ja Pilke 2018]. Jos lentovero laajentuu myös Suomeen ja koskisi lentorahtejakin, tulisi tämä vaikuttamaan erityisesti Suomen ulkomaankauppaan negatiivisesti.

Verkkokauppaa pitävien yritysten tulee huomioida lainsäädännölliset velvoitteet ja pysyä perillä uusista verkkokaupankäyntiin vaikuttavista laeista ja asetuksista. Valitettavasti Suomessa ei tällä hetkellä ole tahoja, jotka tekisi aktiivista seurantaa esimerkiksi oikean muotoisista kaupan ehdoista eri verkkokaupoissa. Toivottavaa on, että verkkokaupankäynnin yleistyessä kuluttajien arviointikyky kehittyä, jolloin verkkokaupan uskottavuus tulisi tärkeäksi osaksi ostopäätöstä. Asiakkaan luottamus verkkokaupan toimintaan on markkinaetu sitä pitävälle yritykselle.

Verkkokaupankäynnin kehittyminen on erittäin mielenkiintoista seurattavaa. Teknologiset, uudet ratkaisut johtavat toisinaan yllättäviin suuntiin, joten ennustamista siitä, millaista vaikkapa vaateostoksilla käyminen on kymmenen vuoden päästä, on vaikea tehdä. Utopistisimmissa visioissa vaatteiden sovitusta tapahtuu virtuaalilaseissa kotipeilin ääressä ja meissä jokaisessa on implementoitu tunnistesiru maksamista varten, mutta tämänkaltaisten ratkaisujen toteutuminen jää nähtäväksi.

Viiteluettelo

Saleh Al-Furiah and Lamia Al-Braheem. 2009. Comprehensive study on methods of fraud prevention in credit card e-payment system. In: *Proc. of the 11th International Conference on Information Integration and Web-based Applications & Services*, 592-598.

BitPay. 2018. Why Accept Blockchain Payments?. <https://bitpay.com/tour>. Checked 20.4.2018.

Eileen Bridges, Ronald E. Goldsmith and Charles F. Hofacker. 2006. Businesses and consumers as online customers. In: Khosrow-Pour, D.B.A. and Mehdi (eds.), *Encyclopedia of E-Commerce, E-Government, and Mobile Commerce*. IGI Global, 83-88.

Checkout Finland Oy. 2018. Selkeät tilitykset. <https://checkout.fi/verkkokauppiaalle/palvelu/>. Luettu 19.4.2018.

Risto Degerman. 2016. Verkkokauppa taapertaa Suomessa vielä lapsenkengissä – kotiin kannetaan vaikka sohva. Yle. <https://yle.fi/uutiset/3-8701764>. Luettu 1.3.2018.

Collector Bank AB. 2018. Lasku ja osamaksu. <https://www.collector.fi/yritysasiakkaat/maksuratkaisut-verkkokauppaan-jamyyymalaan/lasku-ja-osamaksu/>. Luettu 20.4.2018.

COSS ry. 2009. <https://coss.fi/avoimuus/avoin-lahdekoodi>. Luettu 26.3.2018.

Silja Eisto. 2017. Maksutavat monipuolistuvat – verkkokaupan maksutavat saapuvat kivijalkaan. <https://www.yrittajat.fi/uutiset/552758-maksutavat-monipuolistuvat-verkkokaupan-maksutavat-saapuvat-kivijalkaan>. Luettu 1.3.2018.

Finanssivalvonta. 2017. Tietoa Finanssivalvonnasta. <http://www.finanssivalvonta.fi/fi/Fiva/Pages/Default.aspx>. Luettu 6.3.2018.

Ronald E. Goldsmith and Eileen Bridges. 2000. E-tailing versus retailing: Using attitudes to predict online buying behavior. *QJEC* 1, 3, 245-253.

Sami Haltia. 2016. Verkkokaupan kasvu Suomessa ja maailmalla. <https://www.andersinnovations.com/fi/blogi/verkkokaupan-trendit-2017/>. Luettu 1.3.2018.

George Huang, K.L. Mak and PY Jiang. 2002. *E-commerce And Digital Manufacturing*. Emerald Group Publishing Limited.

Burhan Ul Islam Khan, Rashidah F. Olanrewaju, Asifa Mehraj Baba, Adil Ahmad Langoo and Shahul Assad. 2017. A compendious study of online payment systems: Past developments, present impact, and future considerations. *IJACSA* 8, 4, 256-271.

Kilpailu- ja kuluttajavirasto. 2014. Ennen sopimuksen tekemistä annettavat tiedot. <https://www.kkv.fi/Tietoa-ja-ohjeita/Ostaminen-myyminen-ja-sopimukset/verkkokauppa-posti-ja-puhelinmyynti-etamynti/taulukko-annettavista-tiedoista/>. Luettu 27.3.2018.

Kilpailu- ja kuluttajavirasto. 2017. Kuluttaja-asiamiehen linjaus. Maksaminen verkkokaupassa. <https://www.kkv.fi/ratkaisut-ja-julkaisut/julkaisut/kuluttaja-asiamiehen-linjaukset/aihekohtaiset/maksaminen-verkkokaupassa>. Luettu 27.3.2018.

Kilpailu- ja kuluttajavirasto. 2006. Kuluttaja-asiamiehen linjaus. Sopimusehtojen laatiminen. <https://www.kkv.fi/ratkaisut-ja-julkaisut/julkaisut/kuluttaja-asiamiehen-linjaukset/aihekohtaiset/sopimusehtojen-laatiminen>. Luettu 27.3.2018.

Kilpailu- ja kuluttajavirasto. 2014. Verkko, posti- ja puhelinmyynti ovat etämyyntiä. <https://www.kkv.fi/Tietoa-ja-ohjeita/Ostaminen-myyminen-ja-sopimukset/verkkokauppa-posti-ja-puhelinmyynti-etamynti>. Luettu 27.3.2018.

A. I. Kokkinaki, R. Dekker and M.B.M de Koster. 2002. E-business models for reverse logistics: contributions and challenges. In: *Proc. of the International Conference on Information Technology: Coding and Computing (ITCC'02)*, 470-476.

Jonas Kong. 2016. Bridging the gap between the stakeholders and the users at Alibaba.com. In: *Proc. of the 3rd HCIBGO*, 210-217.

Kai Korhonen. 2016. Verkkokaupan toteuttaminen – itse vai kumppanin avulla?. <https://www.paytrail.com/blog/verkkokaupan-toteuttaminen-itse-vai-kumppanin-avulla>. Luettu 26.3.2018.

Yi Liu, Xingtong Liu, Ruilin Li. 2017. An efficient method to enhance Bitcoin wallet security. In: *Proc. of the 11th IEEE International Conference on Anticounterfeiting, Security, and Identification*, 26-29.

Mikko Leppänen. 2017. Tienaatko kryptovaluutoilla? Näin pidät verokarhun tyytyväisenä. <https://yle.fi/uutiset/3-9933923>. Luettu 6.3.2018.

Pekka Majaniemi. 2018. Maksutapalisä poistuu – jäävätkö korttiyhtiöiden maksut verkkokauppojen maksettavaksi?. <https://www.paytrail.com/blog/maksutapalisa-poistuu-jaavatko-korttiyhtioiden-maksut-verkkokaupiaan-maksettavaksi>. Luettu 27.3.2018.

Maksulaitoslaki 2017/890. Annettu Helsingissä 14.12.2017. <https://www.finlex.fi/fi/laki/ajantasa/2010/20100297>. Luettu 6.3.2018.

Juha-Matti Mäntylä. 2018. Kännykkämaksaminen Suomessa vielä lapsenkengissä – Ruotsissa povataan jo käteisen kuolemaa. <https://yle.fi/uutiset/3-10094786>. Luettu 1.3.2018.

Katja Okkonen ja Anita Lukka. 2004. Value added logistical support service: Logistiikan palveluiden tarjoajat Suomessa. Tutkimusraportti 151. Lappeenranta teknillinen yliopisto, tuotantotalouden osasto.

Anuja Pande, A.B. Deshmukh and M. D. Tambakhe. 2014. E-payment gateway model. *IJCSIT* 5, 2, 2569-2573.

Tomi-Jukka Panttila. 2016. Keisarin uudet vaatteet – kun verkkokaupan sisältö jää unholaan. <https://www.solita.fi/blogit/keisarin-uudet-vaatteet-kun-verkkokaupan-sisalto-jaa-unholaan>. Luettu 26.3.2018.

Paytrail Oyj. 2018. Opas verkkokauppa-alustan valintaan. https://cdn2.hubspot.net/hubfs/335946/images/eBook_Folder/Paytrail_Opas_verkkokauppa_alustan_valintaan.pdf?t=1522220006449. Luettu 26.3.2018.

Mikael Smeds. 2017. Mitä on mobiilimaksaminen? 8 ajankohtaisinta tapaa maksaa kännykällä. <https://www.maksuturva.fi/blogi/mita-on-mobiilimaksaminen-8-ajankohtaisinta-tapaa-maksaa-kannykalla>. Luettu 1.3.2018.

Hannu Saarijärvi. 2015. Verkkokaupan lieveilmiöt. <https://blogs.uta.fi/saarijarvi/verkkokaupan-lieveilmiot/>. Luettu 30.4.2018.

Suomen Pankki. 2014. Bitcoinin käyttöön liittyy riskejä. <https://www.suomenpankki.fi/fi/media-ja-julkaisut/uutiset/2014/bitcoinin-kayttoon-liittyy-riskeja>. Luettu 21.4.2018.

Suomen Pankki. 2017. Vuoden 2016 maksuliiketilastot. <https://www.suomenpankki.fi/fi/Tilastot/maksuliiketilastot/>. Luettu 28.4.2018.

TIEKE Tietoyhteiskunnan kehittämiskeskus ry. 2018.

Miten maksutavat toimivat verkkokaupassa?. <https://www.tieke.fi/pages/viewpage.action?pageId=37519567>. Luettu 20.4.2018.

TIEKE Tietoyhteiskunnan kehittämiskeskus ry. 2018. Mitä on verkkomaksaminen?. <https://www.tieke.fi/pages/viewpage.action?pageId=37519565>. Luettu 1.3.2018.

Riitta Uosukainen ja Antti Pilke. 2018. Ruotsalaiset maksavat pian lentoveroa – Berner tyrmää: Ei sovi Suomeen, voisi haitata Finnairia. <https://yle.fi/uutiset/3-10112780>. Luettu 15.5.2018.

Visa Europe. 2018. Hallittua ja turvallista maksamista. <https://www.visa.fi/fi/credit-debit/hallittua-ja-turvallista-maksamista>. Luettu 1.3.2018.

Terhi Vuorinen. 2014. Ostoksen palautusoikeus on kirjattu lakiin vain etäkaupassa. <https://yle.fi/aihe/artikkeli/2013/12/05/ostoksen-palautusoikeus-kirjattu-lakiin-vain-etakaupassa>. Luettu 24.4.2018.

Harri Vänskä. 2016. Kortit siirtyvät kännykkään – mobiilimaksamisen läpimurto lähestyy. <https://www.kauppalehti.fi/uutiset/kortit-siirtyvat-kannykkaan--mobiilimaksamisen-lapimurto-lahestyy/E2pW3MqW>. Luettu 1.3.2018.

Mielipideanalyysi ja Twitter: Big data -aikakauden kristallipallo?

Toni Weckroth

Tiivistelmä.

Tulevaisuuden ennustaminen eri sovellusalueilla on kiehtova ja alati kasvavan kiinnostuksen kohteena oleva tutkimusala. Lähitulevaisuutta voidaan pyrkiä ennustamaan esimerkiksi sosiaalisen median sisältöjen kuten Twitter-viestien pohjalta, kun ne on ensin mielipideanalyysin keinoin luokiteltu positiivisiksi tai negatiivisiksi. Luokitellun datajoukon pohjalta voidaan pyrkiä luomaan erilaisia ennustusmalleja esimerkiksi pörssikurssien liikkeitä tai vaalituloksia ennakoimaan. Tässä tutkielmassa kuvataan yleisellä ja melko karkealla tasolla prosessin kulku datan keräämisestä ja esikäsittelystä aina mielipiteiden ja tunnetilojen luokitteluun asti. Lopuksi esitellään esimerkinomaisesti muutamia tulevaisuuden tapahtumia ennustaneita tutkimuksia.

Avainsanat ja -sanonnat: Big data, Twitter, sosiaalinen media, mielipideanalyysi, tiedonlouhinta, mielipiteiden louhinta, ennustaminen.

1. Johdanto

Lähitulevaisuuden ennustaminen on epäilemättä mielenkiintoinen tutkimuksen kohde erityisesti finanssimaailman näkökulmasta ja taloudellisessa mielessä tarkasteltuna. Vastaavasti politiikan tutkimuksessa vaalitulosten ennustaminen ja tähän soveltuvat työkalut varmasti kiinnostavat monia. Osakemarkkinoiden ennustamisen ohella aihe on ilmeisen merkittävä myös esimerkiksi vedonlyöjien ja miksei vakuutusväenkin keskuudessa, joskin lähtökohdat ja näkökulmat voivat olla paljolti toisistaan poikkeavia. Niin ikään meteorologit ovat ennustaneet säätä jo vuosikymmenien ajan. Käytännön esimerkkejä monilta muiltakin aloilta varmasti löytyisi lukemattomia.

Perinteisesti ihmisten mielipiteitä on kerätty esimerkiksi erilaisilla mielipidemittauksilla ja haastatteluilla, jotka voivat olla tekijöilleen erittäin työläitä, aikaa vieviä, sekä kalliita. Teknologian erittäin nopea kehitys viime vuosikymmeninä on mahdollistanut muun muassa sosiaalisen median kehittymisen, ja yhä useampi ihminen vaikuttaakin aktiivisesti sen eri alustoilla kuten Facebookissa, Twitterissä ja Instagramissa. Ihmisten välinen vuorovaikutus sekä arkeamme ja tunnetilojamme kuvaava toiminta ilmenee nykyään siis enenevässä määrin myös verkossa, ja esimerkiksi Twitteristä puhuttaessa tämä valtava määrä dataa on verrattain julkista.

Tämä tutkielma keskittyy pääosin Twitteristä kerättyyn dataan ja sen tarjoamiin mahdollisuuksiin muun muassa tiedonlouhintaan liittyen. Twitter on mikroblogipalvelu, jossa käyttäjät jakavat ajatuksia ja mielipiteitä keskustellen toisensa kanssa lyhyillä, maksimissaan 140 merkin¹ pituisilla, viesteillä eli *toiiteilla* (tweet). Twitterille on tunnusomaista ajan hermolla oleminen ja siellä käsitelläänkin usein reaaliajassa erilaisia ajankohtaisia aiheita niin politiikan, viihteen, urheilun kuin vaikkapa luonnonmullistusten ympärillä – pienen ryhmän kaverillisista keskusteluista alkaen, skaalautuen aina globaaliin mittakaavaan asti. Twitter onkin oivallinen alusta ja lähtökohta tutkimuksille, joissa pyritään sosiaalisen median kautta keräämään yhteiskuntaan ja ihmisiin liittyviä havaintoja esimerkiksi mielipideanalyysin keinoin.

Mielipideanalyysi pyrkii automatisoidusti analysoimaan, arvioimaan ja luokittelemaan lähdeaineistoa, esimerkiksi tunnistamalla siihen liittyviin entiteetteihin kohdistuvia mielipiteitä ja tunnetiloja. Kuten Bollen ja muut [2011] artikkelissaan, muutamia psykologian tutkimuksiin viitaten, toteavat, on tunnepohjainen päätöksenteko yleistä ihmisluonteelle. Tähän pohjaten onkin perusteltua kiinnostua siitä, olisiko esimerkiksi yhteiskunnassa yleisesti vallitsevaa tunnetilaa mahdollista tunnistaa ihmisten sosiaaliseen mediaan tuottamaa sisältöä analysoimalla ja kenties jopa tehdä ennustuksia tämän tiedon pohjalta. Tutkielman pyrkimys on vastata tähän kysymykseen, ja ennen kaikkea kuvata yleisellä tasolla, kuinka koko prosessi käytännössä tapahtuu.

Loppu tutkielman rakenteesta on seuraavanlainen: Aluksi käsitellään aihepiiriin tiukasti liittyvän big datan käsitettä yleisellä tasolla, esitellen tähän liitetyjä yleisesti hyväksytyjä määritelmiä (luku 2). Luvussa 3 keskitytään Twitteriin ja erilaisiin keinoihin, joilla palvelun sisältöä on mahdollista kerätä esimerkiksi tutkimuskäyttöön. Sitten siirrytään datan hallintaan ja esikäsittelyyn (luku 4) sekä mielipideanalyysiin eri tekniikoihin (luku 5). Luvun 5 lopussa esitellään lyhyesti muutamia aiheeseen liittyviä tutkimuksia eri sovellusaloilta. Lopuksi kerrataan yhteenvedon muodossa tutkielman keskeisiä asioita ja pohditaan tulevaisuuden näkymiä.

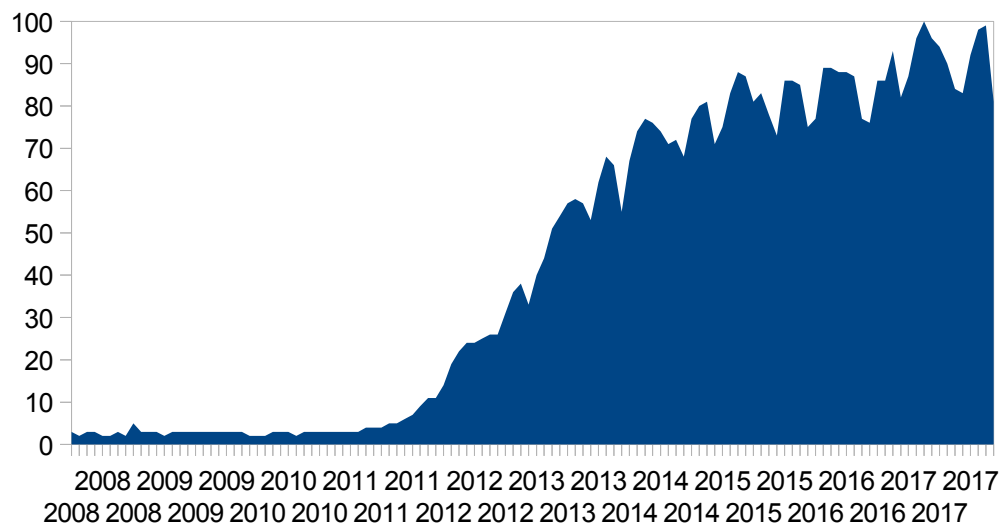
2. Big data

Big Data, big data, iso data, massadata, suuri tietoaaineisto – yhdellä viime vuosien suosituimmista ilmiöistä on monta nimeä. Yleisimmin käytetty muoto suomenkin kielessä lienee englannista lainattu *big data*. Tutkielman luonteesta johtuen tässä luvussa on keskitytty pääosin big datan muotoihin sosiaalisessa mediassa ja internetissä.

¹ Twitter kaksinkertaisti merkkimäärän 280 merkkiin vuoden 2017 lopussa.

Big data on terminä vielä suhteellisen tuore, sillä Googlen hakukoneessa sanapari on alkanut toden teolla esiintyä vasta vuodesta 2011 alkaen, nousujohteisesti, kuten kuvasta 1 voidaan havaita. Jollain tasolla suuren datamäärän ilmiö on varmasti liittynyt tavalla tai toisella tietojenkäsittelyyn tieteenalan alkuvuosista lähtien – suureksi luokiteltava datamäärä vain on matkan varrella skaalautunut eksponentiaalisesti teknologisen kehityksen mukana. Seuraavassa kohdassa määritellään ja havainnollistetaan termi *big data* kolmella V-alkuisella sanalla.

Google Trends: "Big Data" (Koko maailma, 2008-2017)



Kuva 1. Hakusanan "Big Data" Google-hakujen trendi kymmenen vuoden ajalta. Lähde: <https://trends.google.fi/trends/>

2.1. Big datan kolme veetä

Big datasta puhuttaessa päällimmäisenä mieleen tulevana asiana lienee useimmiten nimensä mukaisesti suuri määrä tietoa. Tämä onkin yksi oleellisista big dataan liittyvistä ominaisuuksista, mutta ei suinkaan ainoa. Yleisesti käytetyn määritelmän mukaan big dataa kuvataan englanniksi kolmella V-kirjaimella alkavalla sanalla (*Three V's: Volume, Variety & Velocity*) [Gandomi and Haider 2015], jotka voitaisiin suomentaa, V:t säilyttäen, esimerkiksi muotoon *volyymi, vaihtelu ja vauhti*.

2.1.1. Volyymia eli datan määrää big datasta puhuttaessa mitataan yleensä useissa tera- tai petatavuissa. Toisin sanoen miljoonissa gigatavuissa ja miljardeissa megatavuissa (1 Pt = 1 000 000 Gt = 1 000 000 000 Mt). Yksi petatavu vastaa siis lähes 1,5 miljoonaa perinteistä CD-levyä (700 Mt) tai noin 250 000 DVD-levyä/muistitikkaa (≈ 4 Gt).

Facebookilla on tätä kirjoittaessa lähes 2,2 miljardia aktiivista käyttäjää², YouTubeella näitä on yli miljardi ja palvelun videoita katsotaan yli miljardi tuntia joka päivä³. Twitterillä aktiivisia käyttäjiä on tällä hetkellä yli 330 miljoonaa ja tviittejä tuotetaan noin 500 miljoonaa päivittäin². Datamäärä, jota eri palveluiden käyttäjät päivittäin tuottavat, on siis valtava.

Käsitteenä volyyymi tulee luonnollisesti suhteuttaa, sovellusalueen lisäksi, teknologian kehittymiseen eli aikaan. Toisin sanoen se mikä tänä päivänä on "isoa" ja tulkitaan kuuluvaksi big dataan, ei välttämättä ole sitä enää kymmenen vuoden kuluttua. Gandomi ja Haider [2015] huomauttavat, ettei ole mitään yksiselitteistä rajaa määrittämään mikä on volyymiltaan big dataa ja mikä ei. He myös viittaavat Manyikan ja muiden [2011] huomioon, että yksi sekunti HD-laatuista videokuvaa vastaa kooltaan yli 2 000 sivua tekstiä. Tämä toiminee myös hyvänä esimerkkinä datan vaihtelevasta luonteesta.

2.1.2. Vaihtelu eli moninaisuus on rakenteellisesti moninaisin eri tavoin muotoutunut datajoukko. Suurin osa kaikesta olemassa olevasta ja jatkuvasti maailmaan tuotettavasta datasta on *ei-rakenteellista* (semi-structured and unstructured), jonka yhteydessä voidaan puhua esimerkiksi tekstistä, kuvista, videoista ja audiosta. Eri datatyypit ovat luonteeltaan siis erittäin heterogeenisiä ja vaativat kukin omanlaisiaan lähestymistapoja. [Gandomi and Haider 2015.] *Rakenteellinen data* (structured), ei-rakenteellisesta poiketen, on selkeämmin käsiteltävissä ja analysoitavissa olevaa ikään kuin jo valmista dataa. Tällöin puhutaan esimerkiksi sql-tietokannoista. Tämä tutkielma keskittyy ei-rakenteellisen, eli vielä raa'assa muodossa olevan datan käsittelyyn ja hallintaan. Toisin sanoen Twitteriin jatkuvana virtana tuotettaviin lyhyisiin mikrobloggauksiin eli tviitteihin sekä niihin liittyviin luhintatyökaluihin.

2.1.3. Vauhti eli valtaisalla nopeudella jatkuvasti järjestelmien prosessoitavaksi tulviva datavirta [Gandomi and Haider 2015]. Jo aiemmin mainitun Twitterin arviolta noin 500 000 000 päivittäistä tviittiä toiminee yhtenä esimerkkinä siitä, kuinka uutta tietoa tuotetaan jatkuvana virtana. Keskimäärin Twitter käsittelee noin 6 000 tviittiä sekunnissa eli 360 000 tviittiä joka minuutti. Lisäksi on huomionarvoista, että palvelun tuottama datamäärä on verrattain pientä Facebookin ja videopalvelu YouTubeen vastaaviin datamääriin verrattuna.

Kolmen edellä käsitellyn v-alkuisen sanan lisäksi on ehdotettu myös muita täsmäntäviä termejä big datan kuvaamiseksi, joista mainittakoon Gandomin ja Haiderinkin [2015] esittämät englanninkieliset sanat *Veracity* eli todenmukai-

² www.internetlivestats.com

³ www.youtube.com/yt/about/press

suus tai oikeammin epävarmuus todenmukaisuuteen liittyen, *Variability* eli vaihtelevuus ja muuttuvuus sekä *Value* eli arvo, joka yleensä datan ollessa raa'assa muodossa on vähäinen, mutta oikein analysoituna mahdollisesti hyvinkin korkea.

Vaihtoehtoisen, joskaan ei edellistä poissulkevan, määritelmän big datan kuvaamiseksi antavat Wu ja muut [2014] esittelemällä *HACE teoreemansa* (Heterogeneous, Autonomous, Complex & Evolving). Puhutaan siis suuresta määrästä heterogeenistä ja autonomisista lähteistä peräisin olevaa dataa, josta etsitään monimutkaisia ja alati kehittyviä suhteita. Kirjoittajat havainnollistavat teoreemaansa kuvaamalla big dataa valtavaksi elefantiksi, jonka ympärillä sokeat henkilöt pyrkivät itsenäisesti selvittämään tämän heille tuntemattoman objektin ominaisuuksia. Eri puolilla elefanttia operoivat henkilöt kuvaavat kukin löydöksiään hyvin subjektiivisella tavalla, riippuen alueesta johon kunkin aistihavainnot rajoittuvat: yksi puhuu köydestä (= häntä), toinen letkusta (= kärsä) ja kolmas seinästä (= kylki). Ongelma saadaan entistä monimutkaisemmaksi, kun kuvitellaan elefantin kasvavan vauhdikkaasti, muuttavan jatkuvasti asentoa ja henkilöiden puhuvan eri kieliä. Tämä Wun ja muiden [2014] tarjoama, sinänsä melko naiivi, kuvaus havainnollistaa osuvan pelkistetysti konkretisoiden big dataan liittyviä haasteita ja ongelmia.

2.2. Ison pahan datan sudenkuopat

Big dataan voidaan helposti liittää niin utopistista kuin dystopististakin retoriikkaa. Toisaalta merkitys esimerkiksi syöpätutkimuksessa tai ilmastonmuutoksen torjunnassa voi olla hyvinkin suurta, kun taas yksityisyyteen ja valvontaan liittyviltä kysymyksiltä ei voi välttyä. Uhkakuvana saatetaan helposti nähdä "isoveli" valvomassa eli esimerkiksi valtiot ja suuryritykset tarkkailemasa ihmisten toimia. [boyd and Crawford 2012.]

Tufekci [2014] käsittelee erityisesti sosiaaliseen mediaan liittyvän big datan sisältämiä ongelmakohtia. Hän nostaa esiin esimerkiksi sen, että sosiaalisesta mediasta kerätyn datan ei voida olettaa kuvastavan otosta koko populaatiosta ja tästä johtuen riski virhetulkintoihinkin kasvaa [Tufekci 2014; boyd and Crawford 2012]. Lisäksi Tufekci [2014] mainitsee nimittäjän puutteen: usein tiedetään, kuka tykkäsi, painoi linkkiä tai uudelleentviittasi tviitin, mutta epäselväksi jää tieto siitä, kuinka moni ihminen kyseisen julkaisun lopulta on nähnyt. Osoittaja siis tiedetään, mutta reagoimatta jättäneiden ihmisten osuuden arvioiminen voi olla mahdotonta. On myös syytä tiedostaa valheellisten ja asiapitoista keskustelua vääristelevien *roskaviestien* (spam), *bottien* (bots) sekä *provosoivien julkaisujen* (trolling) mahdollinen vaikutus aineiston luotettavuutta arvioitaessa [Tufekci 2014; boyd and Crawford 2012]. Lisäksi, tiedostaessaan toimintaansa seurattavan, ihmiset epäilemättä jossakin määrin muuttavat tavanomaista käyt-

täytymistään [Tufekci 2014]. Kaikkien Tufekcin tai boydin ja Crawfordin artikkeleissa käsiteltyjen seikkojen ei voida ajatella olevan merkittävästi aineistoa vääristäviä, joskin osaltaan nämä marginaalisemmatkin asiat lisäävät datajoukkoon kohdistuvaa kohinaa. Toisaalta kun keskitytään nimenomaan *Twitterin syötteen* ennustuspotentiaaliin, osa ongelmakohdista voidaan hyväksyä kuu-luvaksi osaksi Twitterin ominaispiirteitä – yhtä kaikki, mainittujen ongelmien poissaolo tuskin ainakaan heikentäisi ennustusten tarkkuutta.

Big dataan ja sosiaaliseen mediaan liitetään myös paljon yksityisyyteen ja eettisyyteen liittyviä kysymyksiä, joita esimerkiksi Schroeder [2014] käsittelee. Kuitenkin, Twitteristä kerättyjen tviittien ollessa julkisia ja kenen tahansa nähtävillä, nämä yksityisyyskysymykset sivuutetaan tässä tutkielmassa todeten yleisellä tasolla eettisyyden ja yksityisyydensuojan ehtojen oletettavasti huomioidun viitattujen tutkimusartikkeleiden kohdalla. Toisaalta se, että aineisto on julkisesti saatavilla ei automaattisesti tee sen keräämisestä ja hyödyntämisestä eettistä, kuten boyd ja Crawford [2012] huomauttavat.

Lähtökohtaisesti julkisten, yksinkertaisten, lyhyiden ja rakenteeltaan verrattain siistien tviittien myötä perustoiminnallisuudeltaankin hyvin suoraviivaisesta Twitteristä on kaikesta huolimatta muodostunut oivallinen alusta aineistojen keräämiselle [Tufekci 2014], eikä vähiten seuraavassa luvussa käsiteltävien rajapintojen ansiosta.

3. Tutkimusaineiston kerääminen Twitteristä

Mikroblogipalvelu Twitterin toiminta perustuu yksisuuntaiseen seuraaja/seurattava -suhteeseen, jossa käyttäjät voivat vapaavalintaisesti seurata kiinnostavaksi kokemiaan käyttäjätilejä – oli kyseessä sitten lapsuuden ystävä tai vaikka Yhdysvaltain presidentti. Tämä suhde ei siis edellytä molemminpuolista hyväksyntää, kuten esimerkiksi Facebookin kaverisuhteet. Twitterin ominaispiirteitä ovat *aihetunnisteet* (#hashtag) ja *maininnat* (@käyttäjänimi), joilla viestejä voidaan kohdentaa määrättyihin aihepiireihin ja tietyille käyttäjille. Yksi niin ikään merkittävä ominaisuus on *uudelleentoittaukset* (retweet) eli käyttäjät voivat jakaa muiden tuottamaa sisältöä tviittaamalla sitä eteenpäin omalle seuraajakunnalleen, jolloin alkuperäinen tviitti saavuttaa entistä suuremman yleisön. Lyhyitä 140 merkin (nyk. 280) viestejä eli tviittejä julkaisemalla ja eteenpäin jakamalla käyttäjät käyvät keskustelua milloin mihinkin, usein hyvin ajankohtaiseen, aiheeseen liittyen.

Twitteriin tuotettavan datan keräämiseksi, esimerkiksi tutkimuskäyttöön, käytetään usein kahta Twitterin tarjoamaa rajapintaa eli kevyempi ja kaikille ilmainen *Twitter API*⁴ sekä kaikki julkiset tviitit kattava *Firehose*. Twitter API pitää sisällään esimerkiksi *Streaming API*:n ja *Search API*:n. Firehosen käyttöä

⁴ Twitter Docs: <https://developer.twitter.com/en/docs>

DaaS-palveluna (Data-as-a-Service) tarjoava BrightPlanet [2013] kuvaa lyhyesti eri rajapintojen keskeisimpiä eroja: Ero Search API:n ja Streaming API:n välillä on se, että Search API tarjoaa hakutoiminnon muutamien viime päivien aikana tuotettuun dataan, kun taas Streaming API toimii reaaliajassa ja lähettää käyttäjälle hakukriteerit täyttäviä tviittejä jatkuvana virtana eli suoratoistona. Search API tarjoaa siis historiaa, joskin hyvin lyhyeltä ajalta, Streaming API:n keskityessä nykyhetkeen. Kallis Firehose on toimintaperiaatteeltaan kuin ilmainen ja rajoitettu Streaming API, mutta sen käyttöä ei tietävästi rajoiteta lainkaan.

Firehoseen verrattuna ilmaiset rajapinnat sisältävät eräitä merkittäviä rajoituksia, kuten historiaa vain muutamien edellisen päivän ajalta (Search API) ja arviolta maksimissaan noin 1 % päivittäisestä tviittimäärästä tarjotaan vastauksena käyttäjän hakutuloksiin, ikään kuin pienenä näytteenä koko datamassasta. Keskimäärin saatavilla voidaan siis teoreettisesti ajatella olevan maksimissaan $0,01 \times 500\,000\,000$ tviittiä = 5 miljoonaa tviittiä/päivä. Firehosea käytettäessä käytössä pitäisi siis olla 100 % kaikista julkisista tviiteistä. Tviittejä voidaan seuloa hakusanojen, käyttäjänimien tai geologisten koordinaattien perusteella. Jossakin määrin poikkeuksellisesta avoimuudesta huolimatta Twitterkään ei tarjoa julkisesti tietoa siitä, millaisilla kriteereillä Streaming API:n käyttöä rajoitetaan – mitkä tviitit valikoidaan ja toisaalta mitkä sivuutetaan? Myöskään tarjottavaan tviittihistoriaan (noin 1–2 viikkoa) tai yhden prosentin tviittikattoon liittyen ei varsinaisesti löydy virallista lähdettä, vaikka nämä mainitut luvut useissa Twitterin rajapintoja käsittelevissä artikkeleissa esiintyvätkin. Esimerkiksi Boyd ja Crawford [2012, 669] käsittelevät eri rajapintoihin ja näiden rajoituksiin liittyviä ongelmia, kun pyritään arvioimaan kerätyn aineiston laatua, kattavuutta ja luotettavuutta.

3.1. Kaikki vastaan 1 %

Morstatter ja muut [2013] tutkivat tiedonkeruuseen tarkoitettujen palveluiden eroja vertaillen rajoitetun Streaming API:n käyttökelpoisuutta kalliiseen Firehoseen verrattuna. Tutkimuksessa selvisi, että eroja näiden palveluiden väliltä odotetusti löytyi, joskin Streaming API:n tuottaman otoksen voidaan kuitenkin jollakin tasolla ajatella estimoivan Twitterin koko viestivirtaa ja esimerkiksi päivän aikana pinnalle nousseita aiheita. Tutkijat havaitsivat, että määrätynlaisilla Syyrian alueeseen liittyvillä hakusanoilla ja rajauskriteereillä he saivat Streaming API:lla kerätyksi keskimäärin 43,5 % koko päivän aikana tuotetuista julkisista, samoihin hakukriteereihin täsmäivistä, tviiteistä. Kävi siis ilmi, että käytetyt hakusanat ja geologinen rajaus eivät asettaneet uutta kattoa yhden prosentin näyteotoksen irrottamiselle. Toisin sanoen maksimissaan säilyi edelleen globaali tviittivirta, jolloin tarkemmin kiinnostuksen kohdetta rajamalla voitiin kerätä huomattavasti yhtä prosenttia suurempi osuus näihin

tarkoin määriteltyihin hakukriteereihin täsmävistä tviiteistä. Noin kuukauden ajanjakson aikana täsmälleen samoilla Syyriaan liittyvillä hakusanoilla tutkijat saivat kerättyä Streaming API:n avulla 528 592 tviittiä, kun Firehosen tarjoama kokonaismäärä oli 1 280 344 tviittiä. Yli 40 % kaikista hakukriteereihin täsmävistä tviiteistä saatiin siis tallennetuiksi ilmaisen Streaming API:n avulla. [Morstatter *et al.* 2013.]

Monet tutkimukset [Asur and Huberman 2010; O'Connor *et al.* 2010; Pak and Paroubek 2010; Arias *et al.* 2013; Rui *et al.* 2013; Tsakalidis *et al.* 2015] ovat päätyneet käyttämään tiedonkeruussaan Twitterin rajapintoja (yleensä Search API ja/tai Streaming API), ja edellä käsitellyn Morstatterin ja muiden [2013] tutkimuksen pohjalta tätä voitaneen pitää riittävän luotettavana, joskin vain suuntaa antavana, keräilymenetelmänä. Firehosen käytöstä ei vaikuttaisi löytyvän mainintoja, mikä myös osaltaan puoltaa Twitter API:n käyttökelpoisuutta. Näin ollen Firehosen saatavuuden rajoittaminenkin [Kepes 2015] voidaan todeta melko epäolennaiseksi tiedoksi tämän tutkielman kannalta.

3.2. Muita vaihtoehtoja

Valmiita, muun muassa tviittejä sisältäviä, datajoukkoja on ollut tutkijoiden käytettävissä [Ravi and Ravi 2015, 39]. Monet tutkimukset ovatkin hyödyntäneet samoja aineistoja, jolloin esimerkiksi mielipiteiden luokittelussa saavutetut tulokset ovat olleet keskenään selkeämmin vertailukelpoisia. Osa datajoukoista on ollut valmiiksi luokiteltuja, mikä puolestaan on mahdollistanut näiden aineistojen hyödyntämisen esimerkiksi opetusdatana ohjatun koneoppimisen algoritmeissa.

Twitterin Rajapintoja toteuttamaan on kehitelty lukuisia erilaisia työkaluja. Streaming API voidaan helpohkosti ja hyvin pienellä ohjelmointitaidolla ottaa käyttöön esimerkiksi valmiin Python-kirjasto *Tweepyn*⁵ avulla. Moujahid [2014] tarjoaa oivallisen ja yksinkertaisen oppaan *Tweepyn* käyttöönottamiseksi. Ilman ohjelmointitaitojakin aineiston kerääminen onnistuu, esimerkiksi Feltin [2016] vertailututkimuksessaan käyttämällä työkalulla *Netlytic*⁶. Felt käsitteli lisäksi kahta muuta työkalua (*Storify* ja *DMI TCAT*) tviittien keräämiseksi tutkimuskäyttöön. Kaikki näistä kolmesta sovelluksesta hyödyntävät taustallaan ilmaista Twitterin API:a sen hyötyineen ja rajoituksineen.

4. Datajoukon hallinta ja käsittely

Kun datajoukko on saatu koottua, esimerkiksi Twitteristä tviittejä suoratoistamalla, data on yleensä vielä hyvin raakassa ja alkeellisessa muodossa, vaatien muutamia valmistelevia työvaiheita ennen varsinaiseen analysointiin siirtymis-

⁵ <http://www.tweepy.org/>

⁶ <https://netlytic.org/>

tä. Sovellusalue ja tutkimuksen tavoitteet määrittävät muun muassa dokumentin sisältämiin ominaisuuksiin kohdistuvan tarkkuuden sekä erilaisten tekstin esikäsittelyyn liittyvien tekniikoiden valintaa.

4.1. Louhinnan eri perspektiivit eli tasot

Yleisesti tunteiden tai mielipiteiden louhinnasta puhuttaessa, puhutaan usein myös vähintään kolmesta erilaisesta tehtävästä ja perspektiivistä louhintaan liittyen [Feldman 2013; Medhat *et al.* 2014; Balazs and Velásquez 2015; Yadollahi *et al.* 2017; Hemmatian and Sohrabi 2017]. Nämä perspektiivit osoittavat *dokumentin tasolle* (document level), *lauseen tasolle* (sentence level) ja *ominaisuuden tasolle* (aspect level/feature-based).

Dokumentin tasolla toimiessa pyrkimyksenä on tunnistaa koko käsiteltävään dokumenttiin kohdistuva mielipide, tunne tai asenne. Käytännössä on siis tarkoitus määrittellä, onko käsiteltävä dokumentti sävyiltään positiivinen vai negatiivinen (tai neutraali). Tällainen dokumentti voisi olla vaikka tuotearvostelu, mielipidekirjoitus, blogiteksti tai mikrobloggeraus kuten tviitti. Dokumentin taso on eri perspektiiveistä abstraktein, tarjoten hyvin yleisesti koko käsiteltävää dokumenttia koskevaa informaatiota, eikä se täten sovellu kovin hyvin erityistä tarkkuutta vaativiin tehtäviin [Hemmatian and Sohrabi 2017]. Toisaalta lyhyiden ja usein ytimekkäiden tviittien tapauksessa tätä ei varsinaisesti voida pitää samalla tavalla merkittävänä asiana, kuin esimerkiksi pitkiä blogitekstejä tai kokonaisia kirjoja tutkittaessa. Näissä tapauksissa olisi mielipideanalyysin kannalta syytä porautua syvemmälle ja yksityiskohtaisemmalle tasolle.

Hierarkiassa syvemmälle liikuttaessa siirrytään lauseen tasolle, jossa dokumentin jokaiselle lauseelle pyritään löytämään esimerkiksi jokin tunnetila tai mielipide. Edellisen kappaleen dokumenttia koskevien ominaisuuksien ja sääntöjen voidaan todeta pätevän myös lauseen tasolla, sillä yhden lauseen voidaan ajatella olevan lyhyt dokumentti [Balazs and Velásquez 2015]. Lyhyistä tviiteistä puhuttaessa dokumentin ja lauseen tasolla toimimisella ei ole kovinkaan merkittäviä eroavaisuuksia.

Edelleen tarkkuudessa yksityiskohtaisemmalle alueelle siirryttäessä tullaan ominaisuuden tasolle, jossa kullekin yksittäiselle ominaisuudelle tulkitaan mielipiteitä, asenteita tai tunteita. Ensin tulee siis löytää erilaisin luonnollisen kielen käsittelyyn liittyvien tekniikoiden avulla eri ominaisuudet, joita dokumentti pitää sisällään. Ei siis etsitä pelkästään tunnetiloja tai mielipiteitä, vaan myös näiden kohteita. Dokumentin ja lauseen tasoilla tehdyt analyysit toimivat hyvin kun ne pitävät sisällään vain yhden ominaisuuden tai näkökulman, mutta kompuroivat kun näitä alkaa olemaan useampia. Tällöin on tarpeen porautua lausetasoa syvemmälle tutkimaan tarkemmin eri ominaisuuksia. [Balazs and Velásquez 2015; Hemmatian and Sohrabi 2017.]

Suurin osa alan tutkimuksista vaikuttaisi operoiveen dokumentin tasolla. Ravin ja Ravin [2015] kartoituksen mukaan näitä on ollut noin kolminkertainen määrä ominaisuuden tai lauseen tasoilla liikkuviin tutkimuksiin verrattuna. Kyseisessä katsauksessa on kartoitettu myös muutamien merkitykseltään pienempien perspektiivien kuten *sanan taso* (word level) ja *fraasin taso* (phrase level) käyttöä.

4.2. Tekstimuotoisen aineiston esikäsittely

Ennen erilaisiin, seuraavassa luvussa käsiteltäviin, tunnetilojen ja mielipiteiden louhintaan liittyviin työkaluihin ja tekniikoihin perehtymistä, aineiston sisältämä raaka data on syytä saattaa analyysin vaatimalle tasolle erilaisten esikäsittelytekniikoiden avulla. Eräitä yleisimmin käytettyjä tekniikoita ovat *saneistaminen* (tokenization), *hukkasanojen karsiminen* (stop word removal), *typistäminen* (stemming), *lemmaaminen* (lemmatization) sekä *sanaluokkajäsennys* (part-of-speech (POS) tagging). Ravi ja Ravi [2015, 17] listaavat joitain eri tutkimuksissa käytettyjä esikäsittelyn avuksi kehitettyjä työkaluja, jotka toteuttavat muun muassa näitä mainittuja toimenpiteitä.

Saneistamisessa käsiteltävä teksti eli merkkijono erotellaan listaksi, jossa kukin sana muodostaa oman tunnisteensa. Tässä melko suoraviivaisessa vaiheessa tekstit tai lauseet jaetaan, yleensä välilyöntien perusteella jaotellen, sanoiksi eli tunnisteiksi. Toisaalta kielet joissa sanojen erotteluun ei käytetä välilyöntejä (kuten japani, kiina ja thai), vaativat toisenlaisia ja merkittävästi haastavampia lähestymistapoja. [Balazs and Velásquez 2015.] Yksittäisten sanojen sijaan tunnisteiksi voidaan valita myös sanapareja, -kolmikkoja ja niin edelleen. Tällöin, tilanteesta riippuen, analyysia tehtäessä saatetaan saada hyödyllisempää informaatiota: vrt. englanninkielen sanaparia "*not good*" erillisiin sanoihin "*not*" ja "*good*". Kuten nähdään, sanaparin merkitys poikkeaa merkittävästi siitä, mitä nämä sanat yksittäin ja toisistaan erillisinä pystyvät tarjoamaan. Esimerkiksi tällaisia negaatioihin liittyviä haasteita käsitellään tarkemmin kohdassa 5.2.

Hukkasanoina pidetään kieliopillisesti kieltä jäsentäviä sanoja, joiden merkitys tekstin varsinaisen asiasisällön kannalta on hyvin pientä. Englannin kielessä tällaisia sanoja ovat esimerkiksi *a, are, the, was* ja *will*.⁷ [Balazs and Velásquez 2015.] Esikäsittelyvaiheessa nämä asiasisältöön vaikuttamattomat sanat yleensä siis poistetaan ja usein samoin toimitaan myös välimerkkien kohdalla. Vaihtoehtoisesti eri välimerkit (kuten '!') voidaan koodata omiksi tunnisteikseen sano-

⁷ Eräs listaus englannin kielen hukkasanoista: <http://snowball.tartarus.org/algorithms/english/stop.txt>

jen tapaan (esim. '<kysymysmerkki>'). Lisäksi, erityisesti mielipideanalyysiin liittyen, teksteissä käytetyt hymiöt voidaan niin ikään koodata osaksi datajoukkoa. Hymiöitä tutkimuksissaan hyödynsivät muun muassa Arias ja muut [2013] sekä Kanavos ja muut [2017].

Edeltävien hieman yleisemminkin pätevien toimien lisäksi erityisesti tviittien esikäsittelyyn useimmiten kuuluu uudelleentviittausten ja url-osoitteiden eli linkkien karsiminen pois aineistosta [Go *et al.* 2009]. Myös selkeästi mainostamistarkoituksessa julkaistut tviitit voi olla tarpeen yrittää karsia [Rui *et al.* 2013].

Typistämisen tarkoitus on lyhentää sana sen perusmuotoon, joskaan ei aina välttämättä aivan kieliopillisessa mielessä. Tavoite on siis pyrkiä pääsemään eroon kunkin sanan eri taivutusmuodoista – pelkistetysti esitettynä – leikkaamalla taivutetun sanan loppuosa pois määrättyjen sääntöjen mukaisesti. Esimerkiksi englannin kielen sanat *person*, *person's*, *personify* ja *personification* typistyisivät kaikki muotoon *person*, jolloin nämä merkitykseltään sinänsä samaa asiaa tarkoittavat sanat sulautettaisiin yhdeksi ja samaksi tunnisteeksi. [Balazs and Velásquez 2015.] Suosittu ja melko yksinkertainen, mutta silti tehokas, englanninkieleen soveltuva typistämisen suorittava algoritmi on Porterin [2006] typistäjä (*Porter's stemmer*).

Lemmaaminen vastaa periaatteeltaan typistämistä, mutta on vaatimuksiltaan tarkempaa, pyrkien typistettyjen sanojen eksaktiin ja kieliopillisesti täsmälliseen sanakirjamaiseen perusmuotoon [Balazs and Velásquez 2015].

Sanaluokkajäsennyksen tavoitteena on leimata kukin lauseen tai dokumentin sana sille kuuluvalla sanaluokalla eli toisin sanoen määrittää lauseen adjektiivit, substantiivit, verbit, adverbit, prepositiot ym. [Balazs and Velásquez 2015]. Sanaluokkainformaatioon voidaan tarpeen mukaan turvautua myöhemmin mielipideanalyysia tehtäessä, jos esimerkiksi tehtävä ja valittu perspektiivi keskittyvät ominaisuuden tasolle (ks. kohta 4.1.) ja halutaan löytää mielipide samassa lauseessa esiintyvälle useille eri objekteille. Sanoille määriteltyjen sanaluokkien perusteella voidaan yrittää päätellä, että juuri tämän sanan ollessa adjektiivi, se mitä luultavimmin kuvaa nimenomaan tätä objektia (substantiivi). Esimerkiksi lauseessa ”laitteen akun kesto on loistava, mutta näyttö surkea” voitaisiin tutkia laitetta ja sen eri ominaisuuksia – tässä akun kesto saa positiivisen arvion, mutta toisaalta näyttö negatiivisen. Sanaluokkajäsennys auttaa tunnistamaan näitä eri ominaisuuksia ja niihin liittyviä mielipiteitä.

Edellä käsiteltyjen esikäsittelyvaiheiden yhteenvedona esitettäköön Bollenin ja muiden [2009] aineistostaan poimima englanninkielinen esimerkkiviitti

”Feeling too lazy to go to the shops and get something to eat”, joka esikäsittelyn jälkeen muuttui joukoksi tunnisteita: {feel, lazi, shop, someth, eat}. Hukkas sanat (*too, to, go, the, and, get*) siis on poistettu ja kukin jäljelle jäänyt sana on irrotettu omaksi tunnisteekseen sekä työstetty Porterin [2006] työstäsalgoritmillä yksinkertaistetumpaan ja geneerisempään muotoon.

Etenkin tässä luvussa käsitellyt kohdat liittyvät läheisesti *tiedonhaun* (information retrieval) ja *luonnollisen kielen käsittelyn* (natural language processing) tutkimusaloihin, aivan kuten seuraavan luvun mielipideanalyysi.

5. Mielipideanalyysi eli tunnetilojen ja mielipiteiden louhinta

Mielipideanalyysi (sentiment analysis) on tutkimusala, joka pyrkii löytämään, analysoimaan ja arvioimaan muun muassa ihmisten mielentiloja eri aiheita ja ilmiöitä kohtaan. Tavoitteena on siis louhia mielipiteitä, ajatuksia, asenteita ja tunteita kerätyistä havainnoista (esim. kirjoituksista, kasvojen ilmeistä, liikkeistä, puheesta, musiikista jne.). [Yadollahi *et al.* 2017.] Twitteriin liittyvissä tutkimuksissa lähteenä on lyhyitä kirjoituksia eli tviittejä, joten louhinta kohdistuu siis tekstimuotoiseen datajoukkoon.

5.1. Mielipideanalyysi Twitterissä

Useissa viimeisen vuosikymmenen aikana julkaistuissa artikkeleissakin viitattu Pang ja Lee [2008] vetävät yhteen mielipideanalyysin alkuvuosien eri vaiheita ja suuntauksia. Kattavassa ja hyvin perusteellisessa katsauksessa ei viitata vielä Twitteriin tai sen tarjoaman datan hyödyntämiseen mielipiteiden louhinnassa, johtuen epäilemättä vuonna 2006 julkaistun Twitterin alkutaipaleella olemisesta. Niin ikään kattavan katsauksen mielipideanalyysiin sosiaalisessa mediassa (ml. Twitter) tekevät Ravi ja Ravi [2015].

Varhaisimpia tunteiden louhintaan ja Twitteriin liittyviä tutkimuksia on alkanut esiintyä vuodesta 2009 alkaen: Kim ja muut [2009] tutkivat Michael Jacksonin kuolemaan liittyviä tviittejä ja suremista, Bollen ja muut [2009; 2011] vertailivat tviiteistä luokiteltuja tunnetiloja eri reaali maailman tapahtumiin ja esimerkiksi pörssikursseihin, Go ja muut [2009] sekä Pak ja Paroubek [2010] luokittelivat tviittejä eri koneoppimisalgoritmeilla, Asur ja Huberman [2010] ennustivat tviittien perusteella elokuvien lipunmyyntituloja ja Lampos ja muut [2010] puolestaan flunssaepidemian leviämistä. O’Connor ja muut [2010] tutkivat perinteisten mielipidemittausten sekä analysoimiensa tviittien yhtäläisyyksiä ja Tumasjan ja muut [2010] tarkastelivat muun muassa poliittisten tviittien ennustuspotentiaalia Saksassa käytyihin vaaleihin liittyen. Tämän Twitteriin pohjautuvan tutkimusosa-alueen alkuvaiheiden kehitystä esitteleekin melko kattavasti esimerkiksi Martínez-Cámara ja muut [2012].

Aihealue on kasvavan kiinnostuksen myötä aiheuttanut myös kiihkeää keskustelua, kun yksi ensimmäisistä tviittejä käyttäneistä tutkimuksista joutui tiukan kritiikin kohteeksi: Tumasjan ja muut [2010] tutkivat tviittien soveltuvuutta poliittisten mielipiteiden mittaamiseen ja muun muassa niiden vertailukelpoisuutta perinteisiin mielipidemittauksiin verrattuna. Osana artikkelia vertailtiin myös eri puolueita koskevien tviittien suhteellisia määriä todeten suhteellisen prosenttiosuuden korreloivan todellisen äänestysprosentin kanssa Saksan parlamenttivaaleissa vuonna 2009. Jungherr ja muut [2012] kyseenalaistivat useita tutkijoiden tekemiä valintoja, täten pitäen tutkimustuloksia virheellisinä. Omassa vastineessaan Tumasjan ja muut [2012] kuitenkin osoittavat kritisoijien argumenttien puutteellisuuden sekä ristiriitaisuuden ja samalla huomauttavat osan näistä argumenteista itse asiassa jopa tukevan alkuperäisessä tutkimuksessa tehtyjä valintoja.

Ennen nykymuotoista sosiaalista mediaa, ja toki sen aikanaan, monet mielipiteiden ja tunnetilojen louhintaan liittyvistä tutkimuksista ovat perustuneet esimerkiksi tuote- tai elokuva-arvosteluiden [Pang and Lee 2002; Turney 2002] sekä perinteisempien blogi-kirjoitusten käsittelyyn ja/tai luokitteluun [Gruhl *et al.* 2005; Mishne and Rijke 2006; Mishe and Glance 2006], ennen mikroblogin kuten Twitter esiinmarssia. Sittemmin tviiteistä koostuvista aineistoista onkin muodostunut yksi mielipideanalyysin tutkimusalaan liittyvien tutkimusten peruspilareista. Esimerkiksi Tufekci [2014] kuvaa Twitteriä big datan ”malliorganismiksi”, vaikka toisaalta keskittyykin artikkelissaan sosiaaliseen mediaan sekä big dataan liittyviin vaaranpaikkoihin esittäen joitain, muun muassa tviittien käyttöön liittyviä, ongelmakohtia (ks. kohta 2.2).

5.2. Mielipideanalyysiin liittyviä haasteita

Tunteiden ja mielipiteiden louhintaan liittyvinä haasteina voidaan pitää esimerkiksi negatioiden hallintaa (englanninkielinen esimerkki: *I like this book / I don't like this book*) ja etenkin sarkasmin sekä ironian havaitsemista pidetään monimutkaisena ongelmana [Pang and Lee 2008, 35; Martínez-Cámara *et al.* 2012, 19]. Lisäksi, varsinkin Twitterin tiukan merkkirajoituksen myötä, usein jopa täysin tietoisesti tehdyt kielioppi- ja kirjoitusvirheet sekä monin paikoin viljellyt lyhenteet ja akronyymit tuovat etenkin tviiteistä koostuville datajoukoille omia haasteita ja erityispiirteitä. Edellisten jatkoksi voidaan lisätä myös erilaiset murteet ja slangit ynnä muut yleiskielestä poikkeavat kielten variaatiot [Martínez-Cámara *et al.* 2012].

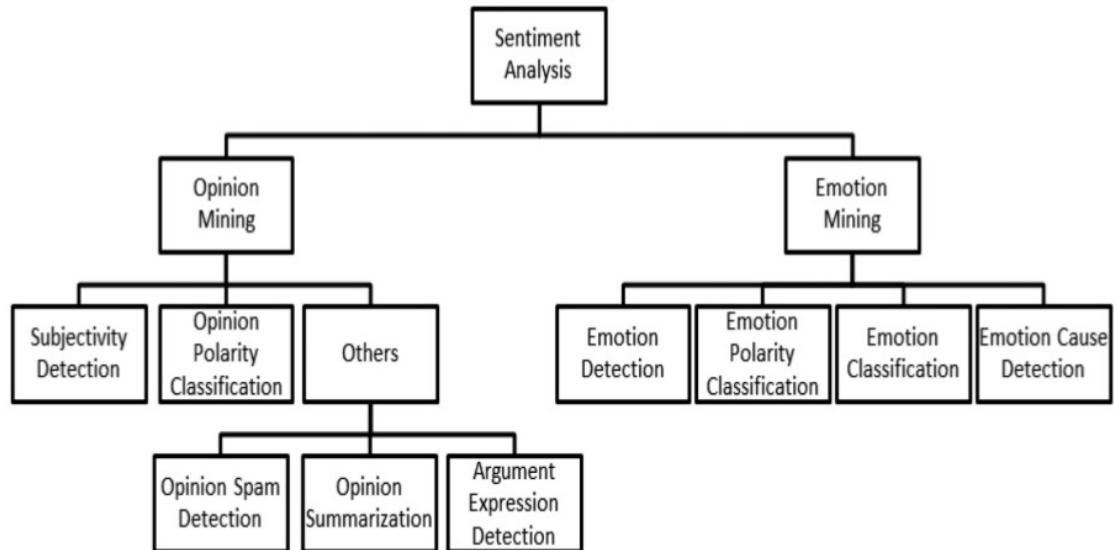
Lauseilla voi myös olla eri merkityksiä eri sovellusalueilla, kuten Pang ja Lee [2008, 40] havainnollistavat Bob Blandilta saamansa osuvan esimerkin avulla: lause ”*go read a book*” todennäköisesti ilmaisee positiivista mielipidettä kirja-arvostelun yhteydessä, mutta vastaavasti elokuva-arvostelusta puhuttaes-

sa lauseen merkitys lienee helposti tulkittavissa negatiiviseksi. Asiayhteyden tunnistaminen on siis tärkeää. Toisena kuvaavana esimerkkinä samaisessa tutkimuksessa tarjotaan jo aiemmin Turneynkin [2002] esittämä adjektiivi *arvaamaton* (unpredictable), jonka merkitys elokuva-arviossa on useimmiten positiivinen (esim. arvaamaton juoni), kun taas vaikkapa auton ohjattavuuteen liitettynä epäilemättä varsin negatiivinen. Tutkittavalla sovellusalueella voi olla siis huomattavaa merkitystä valittavien tekniikoiden ja toimintatapojen suhteen. Esimerkiksi elokuvia koskevissa tutkimuksissa relevanttien tviittien löytäminen voi olla haastavaa, jos elokuvalla on jokin yksinkertainen ja yleinen nimi kuten *The Hangover* tai *It's Complicated* [Rui et al. 2013]. Kaikki krapulaa (hangover) käsittelevät tviitit tuskin liittyvät *The Hangover* -elokuvaan.

Huomionarvoista on myös, että eri tutkimukset käsittelevät pääasiallisesti englannin kieltä, kuten Pang ja Lee [2008, 42] sekä Yadollahi ja muut [2017] huomauttavat. Eri kielet luonnollisesti poikkeavat sanastoiltaan ja kieliopiltaan merkittävästi toisistaan, joten englannin kieleen painottuvia tutkimuksia ei selkeästikään voida pitää yleispätevinä havaintoina. Tsakalidis ja muut [2015] ratkaisivat kieleen liittyvän ongelman kääntämällä keräämänsä tviitit saksan, kreikan ja hollannin kieliltä Google-kääntäjän avulla englanniksi. Automaattista käännöstyötä hyödynsivät myös Tumasjan ja muut [2010], alkuperäisten tviittien ollessa saksankielisiä. Tutkimuksista saadut tulokset olivat positiivisen suuntaisia huolimatta siitä, että automaattisen kääntämisen myötä datan epätarkkuus ja kohina epäilemättä lisääntyy. Lähestymistapana tämä ei siis ole optimaalinen, mutta toisaalta hyvin suoraviivainen ja sinänsä tehokas. Edellisistä poiketen, automaattikäännöksiin turvautumatta, Farra ja muut [2010] käsittelevät tutkimuksessaan arabiankielisten tekstien louhintaa. Yadollahi ja muut [2017] listaavat muutamia kiinan kieleen liittyviä tutkimuksia, joissa aineistona on käytetty muun muassa Twitterin kaltaisesta Weibo-nimisestä palvelusta kerättyjä viestejä.

5.3. Mieli-pideanalyysin monet muodot

Yadollahi ja muut [2017] tarjoavat kuvausta tutkimusalueen nykytilasta ja määrittelevät eri termejä, joita aiempien tutkimusten on havaittu käyttäneen jonkin verran ristiin. Artikkelissa tehdään ero muiden muassa englanninkielisille *sentiment analysis*, *opinion mining*, *polarity classification* ja *emotion mining* termeille, joita on havaittu käytettävän vaihtelevasti ja enemmän tai vähemmän tarkoittamaan samaa asiaa. Määriteltäviä termejä esiintyy mainittujen lisäksi useampiakin (kuva 2). Kirjoittajien yhtenä pyrkimyksenä on selkeyttää tutkimusalan kokonaiskuvaa ja edesauttaa täsmällisyyttä eri termien käytössä.



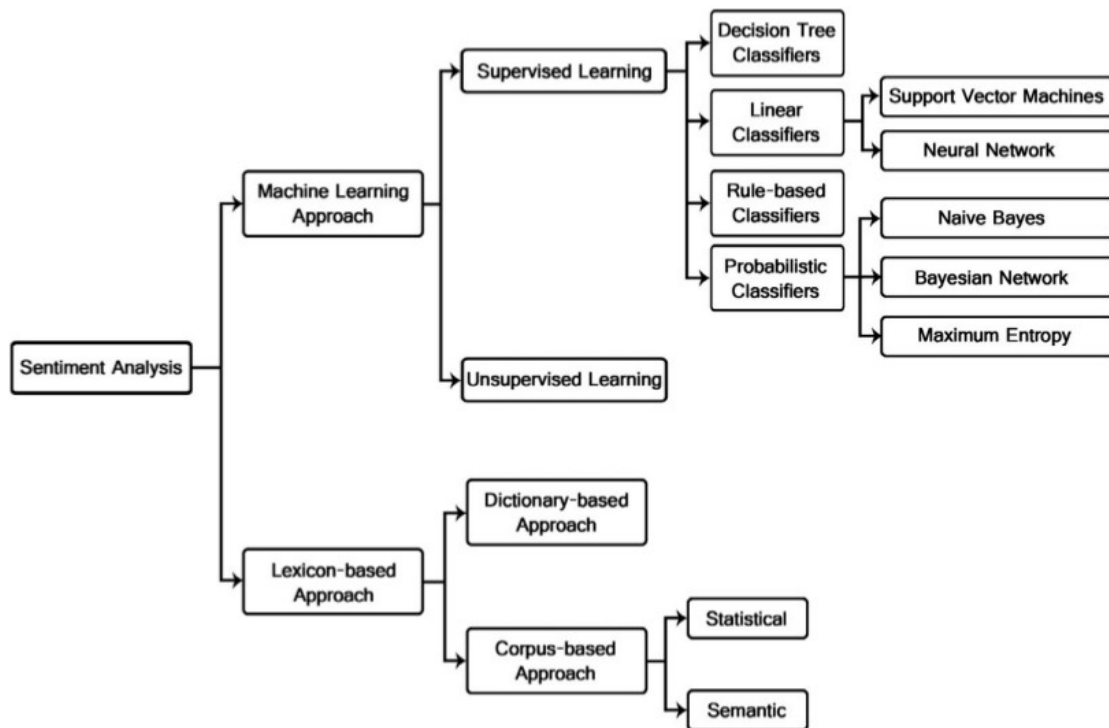
Kuva 2. Mieliideanalyysin jakaantuminen eri luokkiin. (sekaannusten välttämiseksi alkuperäiset englanninkieliset termit säilytetty). [Yadollahi *et al.* 2017]

Yadollahi ja muut [2017] rajaavat aihepiiriään *tunteiden louhimisen* (emotion mining) suuntaan, kun tässä tutkielmassa käsiteltävät asiat keskittyvät enimmäkseen *mielipiteiden louhintaan* (opinion mining) ja *polariteetin luokitteluun* (polarity classification: esim. positiivis- tai negatiivissävytteisen tviitin tunnistaminen). Toisaalta kirjoittajat myös huomauttavat, että termeillä voidaan havaita olevan hyvin paljon yhtäläisyyksiä näiden noudattaessa paljolti samoja metodeja sekä tekniikoita.

5.4. Mielipiteiden ja tunteiden luokittelun tekniikat

Mieliideanalyysi pyrkii löytämään esikäsitellystä lähdeaineistosta muun muassa erilaisia mielipiteitä, asenteita ja tunnetiloja, joita luokitellaan yksinkertaisimmillaan kahteen eri luokkaan eli positiiviseen ja negatiiviseen (tai neutraaliin). Luokittelu on mieliideanalyysin alueella kenties se tärkein datan arvoa määrittävä tekijä, jonka myötä osaksi käsiteltävää datajoukkoa tuodaan konkreettista ja erityistä informaatioarvoa. Edeltävät työvaiheet toki määrittävät sen, kuinka onnistunut ja tarkka lopputulos lopulta saavutetaan.

Yksi tuoreimmista luokittelutekniikoita kartoittavista tutkimuksista on Hemmatianin ja Sohrabin [2017] kokoama kattava artikkeli, joka listaa kaikki mieliideanalyysin kannalta oleelliset tekniikat niin koneoppimisalgoritmeihin kuin sanastoihinkin perustuvien lähestymistapojen osalta.



Kuva 3. Mieliideanalyysin luokittelutekniikat. [Medhat *et al.* 2014]

Luokittelun tekniikoiden voidaan siis katsoa jakaantuvan kahteen eri pääkohtaan, eli *koneoppimiseen* (machine learning) ja *sanastoon perustuviin* (lexicon-based) luokittelutekniikoihin. Nämä tekniikat jakaantuvat edelleen pienempiin osakokonaisuuksiin, kuten Medhat ja muut [2015] kaaviollaan (kuva 3) kokonaiskuvaa jäsentävät. Lisäksi voidaan todeta olevan vielä kolmaskin kategoria, joka on näiden kahden mainitun tekniikan, koneoppimisen ja sanastoperusteisen, hybridi. Yhtä lailla koneoppimisen menetelmät usein jaetaan kaaviossa mainittujen kahden alakohdan sijaan kolmeen kohtaan, eli *ohjattuun oppimiseen* (supervised learning), *ohjaamattomaan oppimiseen* (unsupervised learning) ja näiden välimuotoon eli *puoli-ohjattuun oppimiseen* (semi-supervised learning). [Medhat *et al.* 2014; Hemmatian and Sohrabi 2017.] Ohjatun oppimisen metodi on todettu mieliideanalyysin tarkoituksiin erinomaisen hyvin sopivaksi malliksi, joskin tämän soveltaminen voi olla hidasta, työlästä, ja sen myötä kallista. Toisaalta suoraviivaisempien puoli-ohjattujen koneoppimisen metodien suosio ollut kasvamaan päin, etenkin mikroblogeihin, kuten Twitter, liittyvien tutkimusten keskuudessa. [Hemmatian and Sohrabi 2017.]

5.4.1. Koneoppimisen menetelmistä suosittua on ollut käyttää jotakin manuaalisesti luokiteltua datajoukkoa opetusdatana (ohjattu oppiminen), jonka avulla algoritmi pyrkii luokittelemaan varsinaisen kohteena olevan datajoukon, opetusdatasta opittujen mallien mukaisesti, sopiviin luokkiin.

Koneoppimisen alueella varsin yleisesti käytettyjä algoritmeja ovat naiivi Bayes -luokittelija [Pang and Lee 2002; Go *et al.* 2009; Pak and Paroubek 2010; Rui *et al.* 2013; Kanavos *et al.* 2017; Li *et al.* 2017], tukivektorikone [Pang and Lee 2002; Go *et al.* 2009; Pak and Paroubek 2010; Birmingham and Smeaton 2011; Arias *et al.* 2015; Li *et al.* 2017], maksimientropia [Pang and Lee 2002; Go *et al.* 2009] sekä erilaiset neuroverkot [Arias *et al.* 2015] ja päätöspuut [Kanavos *et al.* 2017; Li *et al.* 2017]. Ravi ja Ravi [2015] tekivät laajan katselmuksen alan tutkimuksiin listaten muun muassa eri algoritmien esiintymismääriä eri artikkeleissa. Yhteensä 161 tutkimusartikkelia koostanut yhteenveto piti sisällään 55 mainintaa tukivektorikoneesta, 28 naiivista Bayes -analyysistä, 11 neuroverkosta, 9 päätöspuista, 8 maksimientropiasta ja pienempiä määriä useista muista eri tekniikoista. Koneoppimistekniikoiden lisäksi sanastoihin perustuvia lähestymistapoja oli käytetty 41 eri tutkimuksessa.

5.4.2. Sanastoihin perustuvat luokittelutekniikat siis niin ikään ovat yleisesti tutkimuksissa käytettyjä, kuten edellisen kohdan lopussa todettiin. *Sanastoihin perustuvat* (lexicon-based) menetelmät jakaantuvat kahteen alakohtaan, eli *sanakirjoihin perustuviin* (dictionary-based) ja *korpuksiin* eli kokoelmiin perustuviin (corpus-based) menetelmiin. Menetelmiä voi käyttää myös yhdessä.

Sanakirjoihin perustuvissa tekniikoissa käytetään jotakin valmista sanakirjaa, jonka sisältämille sanoille on manuaalisesti arvioitu niitä kuvaavia tunnetiloja tai muita ominaisuuksia. Esimerkiksi adjektiivit *hyvä, kaunis, mahtava, paha, ruma* ja *pelottava* olisi verrattain helppoa luokitella esimerkiksi binäärisesti positiiviseksi tai negatiiviseksi. Tällaisen sanaston perusteella kullekin tutkittavan datajoukon sanalle (tai sanaparille, -kolmikolle jne.) koetetaan etsiä sanakirjasta sitä vastaava tunnearvo ja suorittaa, vaikkapa koko dokumenttiin tai lauseeseen kohdistuva, lopullinen luokittelu sanakirjasta poimitun informaation avulla. [Hemmatian and Sohrabi 2017.]

Vaihtoehtoisesti voidaan käyttää korpus-perusteista lähestymistapaa, joka soveltuu sanakirjaperusteista tekniikkaa paremmin käytettäväksi silloin, kun aineistossa on paljon tiettyyn sovellusalueeseen liittyviä erityispiirteitä, kuten tavalla tai toisella sanojen poikkeavia merkityksiä. Korpus-perusteinen tekniikka perustuu syntaktisiin kohdesovellusalueita koskeviin sana- tai lausemuotoihin ja kielellisiin rajoituksiin. Se huomioi esimerkiksi erilaiset adjektiiveja ja muita mielipidettä ilmaisevia sanoja yhdistävät konjunktiot (kuten *ja, tai, mutta*), pyrkien näin huomioimaan kohdesovellusalueen kielellisiä erityispiirteitä. [Medhat *et al.* 2014; Hemmatian and Sohrabi 2017.]

Luokittelun apuna käytettäviä sanakirjoja voi luoda itse sekä manuaalisesti, joka on hyvin työlästä, että automaattisesti, jolloin käytetään muutamaa manuaalisesti luokiteltua sanaa niin sanottuina siemensanoina. Siemensanojen

pohjalta sanastoa lähdetään laajentamaan esimerkiksi tiedettyjen synonyymien avulla. WordNet⁸ on eräs englannin kieleen pohjautuva ja tutkimuksissa paljon käytetty sanakirja, joka tarjoaa informaatiota muun muassa sanojen synonyymeista. [Hemmatian and Sohrabi 2017.]

5.4.3. Valmiit luokittelutyökalut. Luokittelun suorittamista varten kehitettyjä sovelluksia ja palveluita on tarjolla runsaasti – näistä viittätoista vertailivat Serrano-Guerrero ja muut [2015] eri web-palveluihin keskittyvässä kattavassa katsauksessaan. Useat artikkelissa käsitellyistä palveluista on suunniteltu erityisesti Twitteriä ajatellen. Lyhyiden tekstien, kuten tviittien, kanssa parhaiten vertailuista työkaluista pärjäsivät *AlchemyAPI*, *Lymbix*, *OpenAmplify*, *Semantria* ja *Wingify*. Toisaalta on syytä huomauttaa, että osaa mainituista vertailututkimuksissa käsitellyistä palveluista ei näyttäisi olevan enää olemassa ja osa on vähintään vaihtanut nimeä esimerkiksi yrityskauppojen seurauksena. Tätä kirjoittaessa toimiva linkki löytyi vain Wingifylle⁹ ja oletettavasti myös Semantrialle¹⁰ sekä AlchemyAPI:lle¹¹, joskin edeltävän linkki uudelleenohjautuu Lexalytics-yrityksen sivustolle ja jälkimmäisen IBM:n verkkotunnuksen alle.

5.5. Luokitellun datan validointi – käytännön esimerkki

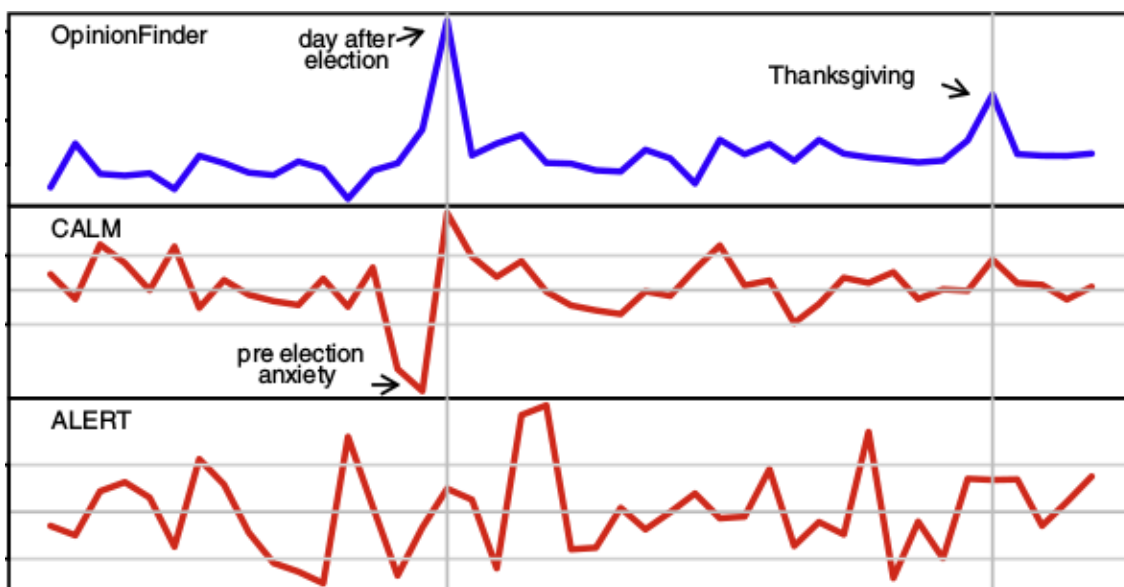
Bollen ja muut [2011] louhivat ja rajasivat aineistostaan tunnetiloja yksinkertaisesti positiivisena ja negatiivisena tunnelmana (*OpinionFinder*) sekä lisäksi eräällä kuusiulotteisella mallilla, joka mittasi yleistä rauhallisuutta, onnellisuutta, varmuutta, hälytetyneisyyttä, elintärkeyttä ja kiltteyttä. Näiden työkalujen avulla miljoonista twiiteistä estimoidut päiväkohtaiset tunnetilat validoitiin vertaamalla niitä kahteen merkittävään ja samalla aikajanalla tapahtuneeseen tosielämän tapahtumaan vuodelta 2008: USA:n presidentinvaalit sekä kiitospäivä. Validoinnin myötä todettiin tapahtumien yhteydessä oletettujen vallitsevien tunnetilojen vastaavan joitakin aineistosta louhittuja ulottuvuuksia, mutta ei kuitenkaan kaikkia. Kuvasta 4 huomataan, että *OpinionFinderin* binääristä työkalua käytettäessä positiivisuus lisääntyi sekä vaalien jälkeen, että kiitospäivänä. GPOMS-työkalun kuudesta ulottuvuudesta kuvassa esimerkin vuoksi vain kaksi, joista huomataan rauhallisuuden (CALM) osoittavan viitteitä reaali maailman tapahtumiin, kun taas hälytetyneisyys (ALERT) ei vaikuttaisi osoittavan oikeastaan minkäänlaista reaktiota näihin kahteen validoinnissa käytettyyn tapahtumaan.

⁸ Princeton University "About WordNet." WordNet: wordnet.princeton.edu Princeton University. 2010.

⁹ www.wingify.com

¹⁰ www.semantria.com

¹¹ www.alchemyapi.com



Kuva 4. Luokitellun datan vertailua tosielämän tapahtumiin (presidentinvaalit ja kiitospäivä, v. 2008, USA). [Bollen *et al.* 2011]

Validoinnin osoittaessa tosielämään yhdistettävissä olevia merkkejä, saavutettiin valmiudet varsinaiseen tilastolliseen tulkintaan, analysointiin ja tulevaisuuden ennustamiseen.

5.6. Mieliwideanalyysi ja tulevaisuuden ennustaminen

Tutkielman lopuksi esitellään esimerkinomaisesti ja lyhyesti muutamia tutkimuksia, jotka ovat koetelleet erilaisia ennustusmalleja eri sovellusaloilla. Syvälistä analyysia eri tutkimuksissa käytetyistä metodeista, tekniikoista ja lähestymistavoista ei ole mahdollista tai tarpeenmukaista tässä kohdassa ja tämän tutkielman puitteissa tehdä. Pyrkimyksenä on lähinnä tarjota mielenkiinnon herättämiseksi lyhyt kuvaus erilaisista mahdollisuuksista, lukijan löytäessä tarvittaessa lisätietoja viitatuista tutkimusartikkeleista. Seuraavassa kappaleessa listatut tutkimukset ovat sisällöltään, metodeiltaan ja sovellusalueiltaan monin paikoin huomattavasti toisistaan poikkeavia, mutta yhteisesti kaikkien tavoitteena on mieliwideanalyysin keinoin kuvata reaali maailmaa ja/tai ennustaa tulevaisuutta valitsemallaan sovellusalueella – sosiaaliseen mediaan pohjautuen.

Ainakin joitain tässä tutkielmassa kuvattuja tekniikoita, työkaluja ja tviiteistä koostuvia datasettejä käyttäen on kehitelty mieliwideanalyysiin liittyviä ennustusmalleja ennustamaan muun muassa elokuvien lipunmyyntituloja [Asur and Huberman 2010; Rui *et al.* 2013; Arias *et al.* 2012], vaalituloksia [Tsakalidis *et al.* 2015; Tumasjan *et al.* 2010; O'Connor *et al.* 2010; Birmingham and Smeaton 2011] sekä osakekurssien [Bollen *et al.* 2009 & 2011; Arias *et al.* 2012; Li *et al.* 2017] ja virtuaalivaluutta Bitcoinin [Kim *et al.* 2017] heilahtelua. Tosin huomautettakoon, että viimeksi viitatussa artikkelissa aineistona käytet-

tiin tviittien sijaan, edelleen sosiaaliseen mediaan pohjaten, keskustelufoorumin viestejä. Usein eri tutkimusten parhailla ennustumalleilla on puhuttu muuttaman päivän viiveestä siihen, kun ennustettu tapahtuma on todennäköisimmin realisoitunut tosielämässä. Monissa tutkimuksissa on saavutettu noin 70–80 % ennustustarkkuuksia [Bollen *et al.* 2011; Arias *et al.* 2013; Li *et al.* 2017]. Toisaalta joissain tapauksissa olemassa olevia, esimerkiksi sovellusalueen historialliseen dataan perustuvia ennustumalleja on laajennettu ja parannettu mielipideanalyysin sekä Twitterin avulla. Näin ollen täyttää kunniaa mielipideanalyysille yksinään ei voida aina antaa, joskin tuloksia parantavia vaikutuksia sillä on todettu olevan [Tskalidis *et al.* 2015; Bollen *et al.* 2011]. Ennustukset ovat usein olleet myös siinä mielessä hyvin karkeita, että esimerkiksi osakekurssien tapauksessa on pyritty ennakoimaan kurssiheilahteluiden suuntaa, mutta ei suoranaisesti sitä kuinka merkittävästä, tai marginaalisesta, noususta tai laskusta puhutaan.

Bollenin ja muiden [2011] tutkimuksessa osakeindeksien liikesuunnan ennustustarkkuus kolmen päivän viiveellä ja parhaalla mallilla oli 86,7 %. Vastavasti Li ja muut [2017] saavuttivat samaisella kolmen päivän viiveellä parhaimmillaan 76,12 % tarkkuuden ennustaessaan yksittäisten IT-alan yritysten osakekurssien liikkeitä. Mielenkiintoisena seikkana toimialan vaikutus, eli IT-alan ohella media-alan yritysten kurssien ennustus tuotti parhaimmillaan tarkkuuden 73,78 %, kun teollisuusalalla vastaava lukema oli vain 55,32 % [Li *et al.* 2017]. Huomionarvoisena voidaan pitää myös sitä, että ainakin vaalituloksia tutkittaessa esimerkiksi puolueiden Twitterissä saamien mainintojen lukumäärä, eli volyyymi itsessään, on ollut merkittävämpi ennustuksen tarkkuuteen vaikuttava tekijä, eikä suinkaan mielipideanalyysiin liittyvät osa-alueet [Tumasjan *et al.* 2010; Bermingham and Smeaton 2011].

Ennustusmallit luodaan ja testataan luonnollisesti historiallista dataa käyttäen, jotta mallin toimivuus on täten voitu vahvistaa – "ennustaen" menneisyyden tulevaisuutta, mutta tutkimushetken menneisyyttä. Teoreettinen ja menneisyydessä onnistunut ennuste ei siis ole tae siitä, että ennustus olisi, vastaavin tuloksin ja ajasta tai paikasta riippumatta, toistettavissa milloin tahansa tulevaisuudessa. Historiallinen tuotto ei ole tae tulevasta, kuten sijoitusmaailmassa hyvin tiedetään.

6. Yhteenveto

Tutkielmassa kuvattiin Twitterin julkisten tviittien pohjalta tehtävää mielipideanalyysia ja pyrittiin samalla vastaamaan kysymykseen, voisiko kuvatuilla metodeilla, tekniikoilla ja työkaluilla mahdollisesti ennustaa tulevaisuutta eri sovellusalueilla. Vastaus jälkimmäiseen jätettäkään lukijan itsensä arvioitavaksi.

Aluksi määriteltiin big datan käsite eritoten kolmen v-alkuisen sanan eli volyymin, vauhdin ja vaihtelun kautta – käsitteeseen liittyviä haasteita unohta-

matta. Lisäksi selitettiin tviittien keräämiseen ja muun muassa suoratoistoon käytettäviin rajapintoihin liittyviä vaiheita. Datajoukon hallintaa ja tekstien esikäsittelyä kuvattiin ennen mielipideanalyysin määrittelyä, tämän pitäessä sisälleen myös erilaisten mielipiteiden ja tunteiden luokittelussa käytettävien yleisimpien koneoppimis- ja sanakirjaperusteisten työkalujen esittelyä. Lopuksi esiteltiin lyhyesti muutamia reaali maailman tapahtumia ennustamaan pyrkineitä tutkimuksia, jotka ovat saavuttaneet osittain varsin myönteisiäkin tuloksia.

Kuitenkin, tulevan pysyessä aina enemmän tai vähemmän hämärän peitossa, ennustamisen sijaan spekuloitakoon tulevaisuudessa huomioon otettavina seikkoina olevan ainakin Twitterin hiljattain kaksinkertaistaman maksimimerkkimäärän (140→280) vaikutus tuleviin tutkimuksiin. Yksittäisen tviitin sisältäessä enemmän, jopa tuplasti, informaatiota, potentiaalisena vaikutuksena voidaan kenties pitää myös luokittelun ja täten ennustusten tarkkuuden kehittymistä. Toisaalta muutoksella saattaa olla vaikutusta työtapoihin ja työkalujen käyttöön liittyen, mikä asettanee omia lisähaasteita tutkijoille. Myös Twitterin käyttäjien toimintatavat ja käyttäytyminen voi muuttua.

Mielenkiintoisena tulevaisuudessa seurattavana asiana voisi pitää myös Twitterin käyttäjämäärien kehittymistä, sillä viime vuosina sekä käyttäjä-, että tviittimäärät ovat pysyneet vakiintuneina melko samoissa lukemissa. Nykyistä Twitterin käyttäjäkuntaa ei voida pitää kaikenkattavana yhteiskuntaa kuvaavana joukkona, joten mahdollinen muutos joko käyttäjäkunnan supistuessa ja rajautuessa tai vaihtoehtoisesti laajentuessa aiempaa monipuolisemman ihmisryhmän pariin on yksi huomioon otettava seikka. On myös hyvä muistaa, että käyttäjäryhmissä ja otoksissa on varmasti alueellisia sekä maakohtaisia eroja. Suurin osa alan tutkimuksesta keskittyy englannin kieleen, joten tutkittavan kielijoukon laajentamista voitaisiin pitää yhtenä tervetulleena tulevaisuuden kehityssuuntana, kuten myös mahdollisuuksien mukaan useisiin eri kieliin soveltuvien tekniikoiden kehittämistä. Toisaalta automaattisten kääntäjien kehityksessä kohdesovellusalueen kieli voidaan tulevaisuudessa entistä luotettavammin kääntää englannin kielelle.

Suuryritykset keräävät, hallinnoivat ja varastoivat – siis hallitsevat – käytännössä kaikkea sosiaaliseen mediaan tuotettavaa dataa. Näillä yrityksillä ei ole velvollisuutta tarjota keräämäänsä dataa ulkopuolisten osapuolten käyttöön. Tästä seuraa boydin ja Crawfordin [2012] mukaan uudenlainen digitaalinen luokkajako, jossa puhutaan *big datan köyhistä* ja *big datan rikkaista*. Yritysten omat tutkijat ja tutkimukset nousevat luonnollisesti erityisasemaan, ollen etuoikeutettuja juuri kyseisellä sovellusalueella. Rajoittamaton data voi olla ulkopuolisillekin saatavilla, mutta kenties hyvin kalliiseen hintaan. Näin ollen esimerkiksi varakkaiden huippuyliopistojen tutkijoilla saattaa olla

mahdollisuus laajoihin datajoukkoihin ja kattaviin tutkimuksiin, pienempien jäädessä "nuolemaan näppejään". [boyd and Crawford 2012.]

Edellä kuvatun tilanteen kehittyminen ja etenkin sen saama kehityssuunta tulee määrittämään merkittäväällä tavalla sosiaaliseen mediaan liittyvien tutkimusalojen tulevaisuutta. Mieli- ja tunneanalyysi on vahvasti riippuvainen saatavilla olevasta datasta ja sosiaalisen median synty on tutkimusalueelle varsinainen kultasuoni. Tai ainakin se voisi olla. Big data ja sosiaalinen media ovat täällä ja näiden arvo sekä hyödyllisyys, mahdollista haitallisuutta unohtamatta, ovat kiistämättömiä, mutta kenellä on valta ja oikeus sekä suuri vastuu hyödyntää tätä *isoo dataa* – nyt ja tulevaisuudessa?

Viiteluettelo

- Marta Arias, Argimiro Arratia, and Ramon Xuriguera. 2013. Forecasting with Twitter data. *ACM Transactions on Intelligent Systems and Technology* 5, 1, Article 8.
- Sitaram Asur and Bernardo A. Huberman. 2010. Predicting the future with social media. In: *Proc. of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 492–499.
- Jorge A. Balazs and Juan D. Velásquez. 2015. Opinion mining and information fusion: A survey. *Information Fusion* 27, 95–110.
- Adam Bermingham and Alan Smeaton. 2011. On using Twitter to monitor political sentiment and predict election results. In: *Proc. of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP)*, 2–10.
- Johan Bollen, Alberto Pepe, and Huina Mao. 2009. Modelling public mood and emotion: Twitter sentiment and socio-economic phenomena. In: *Proc. of the Fifth International AAI Conference on Weblogs and Social Media. Barcelona, Spain*, 450–453.
- Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of Computational Science* 2, 1, 1–8.
- danah boyd and Kate Crawford. 2012. CRITICAL QUESTIONS FOR BIG DATA: Provocations for a cultural, technological, and scholarly phenomenon. *Information, Communication & Society* 15, 5, 662–679.
- BrightPlanet. 2013. Twitter Firehose vs. Twitter API: What's the difference and why should you care? <https://brightplanet.com/2013/06/twitter-firehose-vs-twitter-api-whats-the-difference-and-why-should-you-care/>.
Checked 3.2.2018.
- Noura Farra, Elie Challita, Rawad Abou Assi, and Hazem Hajj. 2010. Sentence-level and document-level sentiment mining for arabic texts. In: *Proc. of the*

IEEE International Conference on Data Mining Workshops (ICDMW), 1114–1119.

- Ronen Feldman. 2013. Techniques and applications for sentiment analysis. *Communications of the ACM* 56, 82–89.
- Mylynn Felt. 2016. Social media and the social sciences: How researchers employ Big data analytics. *Big Data & Society* 3, 1.
- Amir Gandomi and Murtaza Haider. 2015. Beyond the Hype: Big data concepts, methods, and analytics. *International Journal of Information Management* 35, 137–144.
- Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*.
- Daniel Gruhl, R. Guha, Ravi Kumar, Jasmine Novak, and Andrew Tomkins. 2005. The predictive power of online chatter. In: *Proc. of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 78–87.
- Fatemeh Hemmatian and Mohammad Karim Sohrabi. 2017. A survey on classification techniques for opinion mining and sentiment analysis. *Artificial Intelligence Review*, 1–51.
- Andreas Jungherr, Pascal Jürgens, and Harald Schoen. 2012. Why the Pirate Party won the German election of 2009 or the trouble with predictions: A response to Tumasjan, A., Sprenger, T. O., Sander, P. G., & Welpe, I. M. "Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment". *Social Science Computer Review* 30, 2, 229–234.
- Andreas Kanavos, Nikolaos Nodarakis, Spyros Sioutas, Athanasios Tsakalidis, Dimitrios Tsolis, and Giannis Tzimas. 2017. Large scale implementations for Twitter sentiment classification. *Algorithms* 10, 1, 33.
- Ben Kepes. 2015. How to kill your ecosystem. Twitter pulls an evil move with its Firehose. *Forbes Media LLC*. <https://www.forbes.com/sites/benkepes/2015/04/11/how-to-kill-your-ecosystem-twitter-pulls-an-evil-move-with-its-firehose/#12f7084261d3>.
Checked 3.2. 2018.
- Elsa Kim, Sam Gilbert, Michael J. Edwards, and Erhardt Graeff. 2009. Detecting sadness in 140 characters: sentiment analysis and mourning Michael Jackson on Twitter. <http://www.webecologyproject.org/2009/08/detecting-sadness-in-140-characters/>. Checked 15.3.2018.
- Young Bin Kim, Jurim Lee, Nuri Park, Jaegul Choo, Jong-Hyun Kim, and Chang Hun Kim. 2017. When Bitcoin encounters information in an online forum: Using text mining to analyse user opinions and predict value fluctuation. *Plos One* 12, 5.

- Vasileios Lampos, Tijl De Bie, and Nello Cristianini. 2010. Flu detector – Tracking epidemics on Twitter. In: *Balcázar J.L., Bonchi F., Gionis A., Sebag M. (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2010. Lecture Notes in Computer Science*, vol 6323, 599–602.
- Bing Li, Keith C. C. Chan, Carol Ou, and Sun Ruifeng. 2017. Discovering public sentiment in social media for predicting stock movement of publicly listed companies. *Information Systems* 69, 81–92.
- James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh and Angela Hung Byers. 2011. Big data: The next frontier for innovation, competition, and productivity. McKinsey Global Institute. <https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/big-data-the-next-frontier-for-innovation>.
Checked 3.2.2018.
- Eugenio Martínez-Cámara, Maria Teresa Martín-Valdivia, Luis Alfonso Ureña-Lopez, and Arturo Montejo-Ráez. 2012. Sentiment analysis in Twitter. *Natural Language Engineering* 20, 1, 1–28.
- Walaa Medhat, Ahmed Hassan, and Hoda Korashy. 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal* 5, 1093–1113.
- Gilad Mishne and Natalie Glance. 2006. Predicting movie sales from blogger sentiment. In: *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, 155-158.
- Gilad Mishne and Maarten de Rijke. 2006. Capturing global mood levels using blog posts. In: *AAAI Spring Symposium on Computational Approaches to Analysing Weblogs*, 145–152.
- Fred Morstatter, Jürgen Pfeffer, Huan Liu, and Kathleen M. Carley. 2013. Is the sample good enough? Comparing data from Twitter's Streaming API with Twitter's Firehose. In: *Proc. of the 7th International Conference on Weblogs and Social Media, ICWSM 2013*, 400–408.
- Adil Moujahid. 2014. An introduction to text mining using Twitter streaming API and Python. <http://adilmoujahid.com/posts/2014/07/twitter-analytics>. Checked 3.2.2018.
- Brendan O'Connor, Ramnath Balasubramanian, Bryan R. Routledge, and Noah A. Smith. 2010. From tweets to polls: Linking text sentiment to public opinion time series. In: *Proc. of the 4th International AAI Conference on Weblogs and Social Media (ICWSM)* 11, 122–129.
- Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In: *Proc. of the Seventh Conference on International Language Resources and Evaluation (LREC'10)* 10, 1320–1326.

- Bo Pang and Lillian Lee. 2002. Thumbs up? Sentiment classification using machine learning techniques. In: *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 79-86.
- Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2, 1-2, 1-135.
- Martin F. Porter. 2006. An algorithm for suffix stripping. *Program* 40, 3, 211-218.
- Kumar Ravi and Vadlamani Ravi. 2015. A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. *Knowledge-Based Systems* 89, 14-46.
- Huaxia Rui, Yizao Liu, and Andrew Whinston. 2013. Whose and what chatter matters? The effect of tweets on movie sales. *Decision Support Systems* 55, 4, 863-870.
- Ralph Schroeder. 2014. Big data and the brave new world of social media research. *Big Data & Society* 1, 2.
- Jesus Serrano-Guerrero, Jose A. Olivias, Francisco P. Romero, and Enrique Herrera-Viedma. 2015. Sentiment analysis: A review and comparative analysis of web services. *Information Sciences* 311, 18-38.
- Adam Tsakalidis, Symeon Papadopoulos, Alexandra I. Cristea, and Yiannis Kompatsiaris. 2015. Predicting elections for multiple countries using Twitter and polls. *IEEE Intelligent Systems* 30, 2, 10-17.
- Zeynep Tufekci. 2014. Big questions for social media Big data: representativeness, validity and other methodological pitfalls. In: *Proc. of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*, 505-514.
- Andranik Tumasjan, Timm O. Sprenger, Philipp G. Sandner, and Isabell M. Welp. 2010. Predicting elections with Twitter: What 140 characters reveal about political sentiment. In: *Proc. of the 4th International AAAI Conference on Weblogs and Social Media (ICWSM 2010)*, 178-185.
- Andranik Tumasjan, Timm O. Sprenger, Philipp G. Sandner, and Isabell M. Welp. 2012. Where there is a Sea there are Pirates: Response to Jungherr, Jürgens, and Schoen. *Social Science Computer Review* 30, 2, 235-239.
- Peter D. Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: *Proc. of the ACL*, 417-424.
- Xindong Wu, Xingquan Zhu, Gong-Qing Wu, and Wei Ding. 2014. Data mining with Big data. *IEEE Transactions on Knowledge and Data Engineering* 26, 1, 97-107.
- Ali Yadollahi, Ameneh Gholipour Shahraki, and Osmar R. Zaiane. 2017. Current state of text sentiment analysis from opinion to emotion mining. *ACM Computing Surveys* 50, 2.

Puolustautuminen sanakirjoihin perustuvilta salasanahyökkäyksiltä

Yizhou Ye

Tiivistelmä.

Salasanat ovat edelleen yleisin tunnistautumismuoto vakavista tietoturvariskeistä huolimatta. Jos tietokannan tietoturva ei ole kunnossa, salasanatiivisteet on hyvin helppo varastaa SQL-injektoiden avulla. Sen jälkeen nopea sanakirjahyökkäys paljastaa tiivistearvot alkuperäisistä salasanoista. Tutkielmassa käydään läpi hyökkääjän prosessi salasanan varastamiseen. Ensimmäisenä otetaan tarkastelun kohteeksi SQL-injektio, joka on hyvin yleinen haavoittuvuus verkosovelluksissa. Tämän jälkeen siirrytään arvioimaan erikseen yhteydellisiä ja yhteydettömiä sanakirjahyökkäyksiä. Tutkielmassa tutustutaan olemassa oleviin puolustustapoihin ja arvioidaan niiden vahvuuksia sekä heikkouksia. Usein ensimmäisenä mieleen tuleva ratkaisu, salasanan kompleksisuus, sisältää vaarallisia heikkouksia, sillä salasanan keksijä on usein kätevyyttä arvostava käyttäjä. Tällöin pakollisten vaatimusten mukainen salasana sisältääkin helposti havaittavia kaavoja. Sen sijaan käänteiset Turingin testit ovat osoittaneet olevansa tehokkain tapa puolustautua sanakirjahyökkäyksiä vastaan. Kehitystä on kuitenkin tapahduttava jatkuvasti, sillä tietokoneiden ongelmanratkaisukyvyt parantuvat huimaa vauhtia.

Avainsanat: salasana, sanakirjahyökkäys, SQL-injektio, tietoturva, tiivistefunktio.

1. Johdanto

Nykyisin ihmiset käyttävät päivittäin lukuisia eri palveluita, joissa vaaditaan tunnistautumista tietoturvalisistä syistä. Tämä tunnistautuminen voidaan suorittaa monella eri tavalla. Kaikille näille tavoille on yhteistä se, että ne perustuvat johonkin asiaan, joka on vain tietyllä käyttäjällä: mitä me tiedämme (salasanat), keitä me olemme (biometriset sormenjäljet) ja mitä meillä on (kortit, avaimet) [O’Gorman 2003]. Näistä vaihtoehdoista salasanat ovat suosituimpia niiden helppokäyttöisyyden ja toteuttamisen edullisuuden takia. Ongelmana on niiden varastamisen suhteellinen helppous. [Chakrabarti and Singhal 2007.] Juelsin ja Rivestin [2013] mukaan hyökkääjä voi saada käsiinsä käyttäjien salanoja seuraavilla tavoilla:

1. Varastetut tiedot salasanatiivisteistä: Hyökkääjä on onnistunut varastamaan salasanatiivisteitä suoraan tietokannasta tai pahemmassa tapauksessa itse salasanat suoraan.
2. Helposti arvattavat salasanat: Iso osa käyttäjistä valitsee niin yksinkertaisia salasanoja, että hyökkääjällä on hyvin helppo vain arvata se.
3. Näkyvät salasanat: Hyökkääjä näkee, kun käyttäjä kirjoittaa salasanan.
4. Sama salasana useassa eri palvelussa: Jos käyttäjä käyttää samaa salasanaa useassa eri palvelussa, yhden salasanan vuotaminen tarkoittaa muidenkin vuotamista.
5. Käyttäjiltä varastettu salasana: Hyökkääjä on onnistunut vaarantamaan käyttäjän laitteiston käyttämällä haittaohjelmia tai esimerkiksi kalastelulla.
6. Salasanan vaihtopalvelu on vaarannettu: Palvelu, jolla käyttäjä vaihtaa salasanansa, on hyökkääjän hallinnassa.

Tässä tutkielmassa keskitytään lähinnä ensimmäiseen edellä mainituista tavoista. Saatuaan haltuun salasanatiivisteitä hyökkääjän tehtävänä on purkaa ne. Yksi purkamismenetelmistä on sanakirjahyökkäys. Tässä menetelmässä hyökkääjällä on käytössään sanakirjoja, joissa on listattu yleisimpiä salasanoja. Helposti arvattavia salasanoja sivutaan sanakirjahyökkäysten yhtedessä, sillä salasanan on oltava helposti arvattava, jotta se voidaan löytää sanakirjasta. Tutkielman luvussa 2 esitellään SQL-injektio, joka on tapa murtautua tietokantaan. Samalla tutustutaan nopeasti tiivistefunktioihin ja haaste-vastaus-protokollaan. Luvussa 3 esitellään tiivisteiden purkamismenetelmiä painottaen erityisesti sanakirjahyökkäystä. Luvussa käsitellään myös puolustautumiskeinoja kyseisiä hyökkäyksiä vastaan. Luku 4 on yhteenveto.

2. Salasanahyökkäysten ennaltaehkäisy

2.1. SQL-injektiot

Jotta hyökkääjä pääsee purkamaan salasanatiivisteitä, hänen on ensin hankittava ne jostain tietokannasta. Tämä tapahtuu usein SQL-injektioiden kautta. SQL-injektio on haavoittuvuus, jossa hyökkääjällä on mahdollisuus vaikuttaa SQL-kyselyihin, joita sovellus lähettää back-end tietokannalle [Clarke 2012]. Vuonna 2017 The Open Web Application Security Project (OWASP) sijoitti injektiohaavoittuvuudet, joihin SQL-injektio myös kuuluu, vaarallisimmaksi verkkoturvariskiksi [OWASP 2017]. Tietokantataulut, joissa luetellaan käyttäjien salasanat,

ovat haavoittuvaisia SQL-injektioille, jos syötekenttiä ei ole puhdistettu tarpeeksi huolellisesti. Kyseisen haavoittuvuuden avulla hyökkääjä voi esimerkiksi kysyä tietokannalta juuri sitä taulua, jossa on tallennettuna käyttäjien salasanat.

Jotta pystytään ymmärtämään, kuinka hyökkääjät ajattelevat tehdessään SQL-injektioita, on hyödyllistä havainnollistaa prosessia yksinkertaisen matemaattisen esimerkin avulla. Kuvitellaan algebratesti, jossa on kysymys: ”Määritä x :n arvo seuraavassa lausekkeessa, $2 * x + 1 = 7$ ”. Ensimmäinen vastaus, joka tulisi mieleen olisi luultavasti $x = 3$. Tässä ollaan kuitenkin kiinnostuneita kieliopin injektiotekniikoista. Perinteisesti oletetaan x :n olevan kokonaisluku. Jos näkökenttää suurennetaan, voidaan havaita useita toisenlaisia ratkaisuja:

$$2 * (1 + 2) + 1 = 7$$

$$2 * 0 + 6 + 1 = 7$$

$$2 * (-2) + 10 + 1 = 7$$

$$2 * (0 / 0) + 1 = ?$$

Esimerkistä käy ilmi SQL-injektioiden pohjimmainen periaate, jonka Shema [2012] muotoilee lausekkeen kieliopin muuttamiseksi SQL:ssä käytettävillä merkeillä. Vertauskuvallisesti on mahdollista ohittaa turvajärjestelmä negatiivisella vastauksella tai kaataa koko järjestelmä nolllalla jaon seurauksesta.

2.2. Yksinkertaisin injektioimuoto

Yksinkertaisimmat haavoittuvuudet ovat syötekentät, joihin käyttäjät pystyvät kirjoittamaan ongelmallista sisältöä. Pahamaineisin merkki lienee tässä yhteydessä `'`-merkki [Clarke 2012; Shema 2012]. Syy selviää seuraavan esimerkin avulla:

```
SELECT *
FROM users
WHERE username = "" AND password = "";
```

Koodikatkelma 1. Hyökkääjän rakentama SQL-kysely.

Koodikatkelmassa 1 oleva kysely on syntynyt siitä, kun käyttäjä on syöttänyt kuvitellun sovelluksen sisäänkirjautumislomakkeen käyttäjänimi-kenttään `'`-

merkin ja jättänyt salasanan tyhjäksi. Seurauksena on syntaksivirhe, koska '-merkkejä on pariton määrä. Hyökkääjä voi päättää edetä seuraavan askeleen kirjoittamalla kenttään: " 'OR '1' = '1 ". Kysely muuttuisi muotoon:

```
SELECT *
FROM users
WHERE username = " OR '1' = '1' AND password = ";
```

Koodikatkelma 2. Tautologian hyödyntäminen SQL-kyselyssä.

Koska koodikatkelmassa 2 AND-operandilla on suurempi prioriteetti kuin OR-operandilla [Clarke 2012], käytännössä WHERE-osa näyttäisi seuraavanlaiselta:

```
WHERE username = " OR ('1' = '1' AND password = ");
```

Koodikatkelma 3. Sieventynyt muoto edellisestä WHERE-osasta.

Koodikatkelman 3 perusteella voidaan todeta, että kyseinen lause on aina tosi, koska yksi on aina yhtä suuri kuin yksi. Hyökkääjä on onnistunut käyttämään hyväksi tautologiaa. Tässä tapauksessa esimerkkikysely palauttaisi siis kaikki "users"-taulun käyttäjät.

Mikäli hyökkääjä pääsee kirjoittamaan jotain potentiaalisesti vaarallista, hänelle saatetaan näyttää jokin järjestelmän oma virheilmoitus. Tämä voi olla ongelmallista, sillä virheilmoituksista on usein helppo päätellä järjestelmästä tietoja [Shema 2012].

```
"ERROR 1064 (42000): You have an error in your SQL syntax: check the
manual
that corresponds to your MySql server version for the right syntax to use
near " at line 1"
```

Koodikatkelma 4. Tietokantajärjestelmän antama virheilmoitus.

Koodikatkelmassa 4 olevasta virhe-esimerkistä selviää suoraan käytössä olevan tietokantajärjestelmän nimi, MySQL. Tämä tieto on hyvin tärkeää hyökkäjälle, sillä tunnistettuaan tietokannan hän pystyy hienosäätämään kyselyitään juuri kyseisen tietokannan kaltaiseksi. Tietokantaan liityvät virheilmoitukset eivät siis saa olla liian kuvailevia. Samalla täytyy kuitenkin muistaa virheilmoitusten perimmäinen tarkoitus: auttaa kehittäjää testauksessa. Näin ollen virheilmoitusten yhteydessä täytyy olla varma siitä, ettei kehittäjälle tarkoitettuja viestejä päädy asiakaspuolelle.

2.3. Edistyneemmät konseptit, syötesuodattimien ohittaminen

Useimmat nykypäivän verkkosovellukset hyödyntävät jonkinlaisia syötesuodattimia, joiden tarkoitus on suojata yleisiltä hyökkäyksiltä, kuten SQL-injektioilta. Nämä suodattimet ovat joko peräisin sovelluksen omasta koodista, ovat kehittäjän tekemiä, tai ne on toteutettu sovelluksen ulkopuolella esimerkiksi palomuureina. SQL-injektioiden yhteydessä kiinnostavimmat suodattimet saattavat estää kaiken syötteen, joka sisältää jotain seuraavista:

1. SQL-avainsanoja, kuten SELECT, AND tai INSERT
2. Tiettyjä yksittäisiä merkkejä, kuten lainausmerkki tai tavuviiva
3. Välilyönti

On myös olemassa suodattimia, jotka estämisen sijaan yrittävät muokata syötteestä turvallisen. Suodattimista löytyy kuitenkin usein haavoittuvuuksia, joiden avulla hyökkääjä pystyy ohittamaan ne. Jos suodatin on erityisen naiivi, sen voi ohittaa esimerkiksi käyttämällä sekaisin isoja ja pieniä kirjaimia. [Clarke 2012.] Syy tähän on se, että SQL ei välitä kirjainten koosta. Toinen yleinen haavoittuvuus liittyy välilyön-teihin: Monet kehittäjät ajattelevat, etteivät SQL-injektiot ole mahdollisia, jos suodatin estää välilyönin käytön. Tämä ei ole kuitenkaan totta, vaan välilyönti on helposti ohitettavissa rivinsisäisten kommenttien käytöllä [Clarke 2012].

```
"/**/UNION/**/SELECT/**/password/**/FROM/**/users/**/WHERE/**/username
/**/LIKE/**/'admin'"
```

Koodikatkelma 5. Rivinsisäisten kommenttien käyttö SQL-kyselyssä.

Koodikatkelmassa 5 on ohitettu myös yhtäsuuruusmerkin tarkastus, sillä sen sijaan on käytetty LIKE-avainsanaa, jolla on sama funktionaalisuus. MySQL sallii jopa kommenttien käyttämisen yksittäisen avainsanan välissä (`/**/UN/**/ION/**`).

URL-koodaus on tekniikka, jolla muokataan string URL-muotoon [Cheng *et al.* 2010]. SQL-injektioihin liittyen URL-koodauksella pyritään ohittamaan suodatin kirjoittamalla ongelmalliset merkit niiden ASCII-koodina heksadesimaalimuodossaan alkaen %-merkillä. Esimerkiksi '-merkki on ASCII-koodina 0x27 eli URL-koodauksen tapauksessa %27. Joissain tapauksissa, jos tämä ei toimi, on mahdollista ohittaa suodatin kaksois-URL-koodauksella. Samaa esimerkkiä käyttäen '-merkki koodautuisi muotoon %2527. Siis itse %-merkki on koodattu ASCII-koodina (%25). Syy tämän lähetymistavan mahdolliseen toimimiseen on se, että verkkosovellukset saattavat purkaa käyttäjän syötteeseen koodauksen useammin kuin kerran, mutta samalla suodattaminen tehdään ennen viimeistä vaihetta [Clarke 2012].

2.4. Toisen asteen SQL-injektio

Edellä mainitut SQL-injektiot ovat kaikki olleet "ensimmäisen asteen" SQL-injektioita, sillä niissä hyökkäys on suunnattu suoraan SQL-kyselyyn. Toisen asteen SQL-injektiossa on kaksi osaa: Ensin tallennetaan hyökkäyskoodi tietokantaan. Toiseksi rakennetaan SQL-kysely valmiiksi tallennetusta koodista [Ping 2017]. Näitä hyökkäyksiä on huomattavasti vaikeampi havaita. Niiden haavoittuvuudet syntyvät usein helposta virheestä, jonka kehittäjät tekevät [Clarke 2012]. Syötteen saapuessa on selvää, että se voi mahdollisesti olla vaarallista, joten se tulee suodattaa. Jos tietokanta käyttää myöhemmin tietoa, joka on alunperin tullut käyttäjältä sen ajatellaan olevan turvallista, sillä onhan se läpäissyt suodattimet. Tämä on kriittinen virhe, koska konteksti, jossa hyökkääjän koodia käytetään, on nyt eri. Esimerkkinä voisi toimia käyttäjätilin luominen ja sen tietojen muokkaaminen myöhemmin. Tilin luomisen yhteydessä hyökkääjä syöttää vaarallista koodia lomakkeen sähköpostikenttään. Kun hyökkääjä vaihtaa sähköpostiaan, haitallinen koodi lisätään osaksi UPDATE-lausetta, jolloin hyökkäys onnistuu. Täten aikaisemmin turvallinen koodi voikin olla potentiaalisesti vaarallinen, kun sitä käytetään toisessa paikassa.

2.5. Suojautuminen SQL-injektioilta

Suojat SQL-injektioita vastaan voidaan jakaa karkeasti kolmeen eri lajiin: puolustava koodaus, haavoittuvuuksien havaitseminen ja hyökkäyksien ajonaikainen ehkäiseminen [Shar and Tan 2012]. Jokaisella näistä on omia vahvuuksia ja heikkouksia, joten kehittäjien tehtävänä olisi yhdistää näitä. Esimerkiksi puolustavalla koodauksella ei voida paikantaa haavoittuvuuksien sijaintia.

Puolustava koodaus on suoraviivaisin ja tärkein ratkaisu, sillä SQL-injektio haavoittuvuudet syntyvät juuri huonojen koodauskäytäntöjen seurauksina. Yleinen harhaluulo on se, että dynaamisten SQL-kyselyiden välttäminen parametrillisilla lauseilla suojaa kaikilta SQL-injektioilta. Se kattaa kuitenkin vain ensimmäisen asteen SQL-injektiot [Ping 2017]. Niiden käyttäminen on kuitenkin suotavaa, jos tilanne vain sallii sen. Mikäli tämä ei ole mahdollista, kehittäjän on oltava varma, että kaikki käyttäjältä tuleva syöte on suodatettu. Tässä tilanteessa on usein parempi käyttää valmiiksi tehtyjä kirjastoja kuin omia ratkaisuja [Shar and Tan 2012]. Suodattamiseen on olemassa kaksi lähestymistapaa: vaarallisten merkkien esto tai oikeanlaisten syötteiden hyväksyminen. Shar ja Tan [2012] suosittelivat jälkimmäisen vaihtoehdon käyttöä, sillä vaarallisten merkkien estämisessä on helppo unohtaa jotain pientä.

Haavoittuvuuksien havaitseminen on lähestymistapa, jossa painotetaan paljon testausta. Ensimmäinen vaihtoehto on yrittää luoda omia SQL-injektiohaavoittuvuuksia. Hyvänä esimerkkinä toimii Fonsecan ja muiden [2009] kehittämä työkalu, joka lisää verkkosovellukseen automaattisesti haavoittuvuuksia ja generoi näiden pohjalta hyökkäyksiä. Työkalun tarkoitus on evaluoida paikalla olevien turvamekanismien toimivuutta. Toinen lähestymistapa on luoda suoraan hyökkäyksiä. Näissä käytetään usein symbolisia suoritustekniikoita, joiden toiminta perustuu suuren laskemistehoon [Shar and Tan 2012].

Kenties tehokkain keino on kuitenkin tämän kohdan alussa mainituista lajeista viimeinen: hyökkäysten ajonaikainen ehkäiseminen. Ongelmana on suorituskyvyn heikkeneminen ja lisätyn koodin aiheuttama mahdollinen testauksen monimutkaistuminen [Shar and Tan 2012]. Boyd ja Keromytis [2004] ehdottavat ratkaisua, jossa kehittäjä voi rakentaa SQL-kyselyitä vain satunnaisilla avainsanoilla tavallisten avainsanojen sijaan. Ennen kuin tämä tieto siirtyisi tietokantaan, ylimääräinen suodatin muuttaisi sen avainsanat takaisin normaaleiksi. Hyökkääjä ei pystyisi tekemään injektioita, sillä hän ei tietäisi satunnais-tamisen sala-avainta.

Liun ja muiden [2009] esittämä ratkaisu on SQLProb. Se suorittaa annetun ohjelman monilla erilaisilla syötteillä. Tarkoituksena on kerätä kaikki mahdolliset kyselyt, joita ajonaikana saattaisi tapahtua. Kun ohjelma on toiminnassa, jokaista sisääntulevaa kyselyä verrataan valmiiksi kerättyihin kyselyihin. Tämän ratkaisun heikkous on vaaditut testisyötteet. Samalla SQLProb olettaa, että annetut testisyötteet kattavat kaikki mahdolliset kyselyt.

2.6. Kryptograafiset tiivistefunktiot

Useimmat kehittäjät tietävät, että salasanat tulee tallentaa tiivisteinä käyttäen tiivistefunktioita, kuten md5 tai sha256. Syynä tähän on se, että salasanatiivisteiden vuotaessa hyökkääjät eivät saisi suoraan käyttäjien oikeita salasanoja, vaan joutuisivat ensimmäisenä purkamaan tiivisteiden. Nämä tiivistefunktiot muuttavat mielivaltaisen arvon (tässä tapauksessa käyttäjän salasana) kiinteän pituiseksi string-arvoksi, jota kutsutaan tiiviste-arvoksi [Gauravaram 2012]. Tässä yhteydessä tulee muistaa, että salasanaan tulee lisätä jokaiselle käyttäjälle uniikki satunnaisesti luotu arvo, jota kutsutaan suolaksi, ennen kuin se syötetään tiivistefunktiolle [Tatli 2015]. Suola vaikeuttaa hyökkääjän työtä huomattavasti. Tiiviste-arvoilla on kaksi hyvin tärkeää ominaisuutta:

1. Jos tiiviste-arvo tiedetään, on hyvin vaikeaa rakentaa arvo, joka luo sen. Näin ollen tiivistefunktioita kutsutaan usein yksisuuntaisiksi.
2. Todennäköisyys sille, että kaksi arvoa muodostavat saman tiiviste-arvon on hyvin pieni.

Tietokannat tallentavat siis vain tiiviste-arvon. Jokaisella kerralla, kun käyttäjä kirjautuu sisään, hänen salasanastaan lasketaan tiiviste-arvo ja sitä verrataan tietokannassa olevaan arvoon. [Clarke 2012.]

2.7. Haaste-vastaus-protokolla

Salasanoihin perustuva tunnistautuminen vaatii kommunikaatiota käyttäjältä ja palvelimelta kirjautumisen yhteydessä. Tämä prosessi on haavoittuvainen passiivista salakuuntelua vastaan: hyökkääjä pystyy oppimaan salasanan kuuntelemalla keskustelua [Chakrabarti and Singhal 2007].

Tämän suojaksi on kehitetty haaste-vastaus-protokolla. Aloittaakseen protokollan entiteetti A lähettää viestin, joka sisältää A:n identiteetin, entiteetti B:lle. Tämän jälkeen B lähettää A:lle satunnaisen numeron, jota kutsutaan haas-

teeksi. A käyttää haastetta ja salasanaansa tehdäkseen jotain laskentaa niistä. Syntyneen tuloksen (vastauksen) A lähettää B:lle. Viimeisenä B tekee A:n tallennetulla salasanalla saman laskennan ja vertaa tätä tulosta A:n lähettämän vastauksen kanssa. Koska B valitsee eri haasteen jokaista protokollan ajoa kohti, hyökkääjä ei pääse kirjautumaan käyttäjänä lähettämällä saman vastauksen myöhemmin. [Chakrabarti and Singhal 2007]

3. Puolustautuminen salasanahyökkäyksiltä

Vaikka tiivistefunktiot ovat yksisuuntaisia, hyökkääjä pystyy käyttämään raakavoima-, sanakirja- tai sateenkaaritauluhyökkäystä paljastamaan oikeat salasanat olettaen tietyt tiivistearvot. Raakavoimahyökkäyksissä on tarkoituksena laskea jokaista mahdollista syötettä (käyttäjän salasana) vastaava tiivistearvo ja verrata sitä arvoon, joka halutaan murtaa [Tatli 2015]. Sanakirjahyökkäyksissä hyökkääjä käyttää hyväksi sanakirjatiedostoja, joissa on lueteltu miljoonia käyttäjien salasanajoja. Näitä on olemassa kahdenlaisia: yhteydettömiä (offline) ja yhteydellisiä (online). Yhteydettömissä sanakirjahyökkäyksissä hyökkääjä on onnistunut ensin varastamaan itselleen salasanatiivisteitä. Hyökkääjä laskee tämän jälkeen jokaista sanakirjasta löytyvää salasanaa vastaavan tiivistearvon ja vertaa näitä varastamiinsa tiivistearvoihin. Yhteydellisissä sanakirjahyökkäyksissä hyökkääjä ei ole onnistunut saamaan salasanatiivisteitä. Tässä tapauksessa hyökkääjä tarvitsee palautetta palvelimelta, jotta hän tietää arvanneensa oikein yrittäessään kirjautua sisään tiettyinä käyttäjänä [Pinkas and Sander 2002]. Sateenkaaritauluhyökkäys on edistyneempi variantti sanakirjahyökkäyksestä. Siinä hyökkääjällä on käytössään sanakirja, jossa on salasanojen lisäksi myös jokaista vastaava tiivistearvo valmiiksi laskettuna [Tatli 2015].

Keskenäisessä vertailussa raakavoimahyökkäyksen etu on sen varmuus. Mikä tahansa salasana voidaan murtaa tällä keinolla, jos aikaa on vain tarpeeksi. Sanakirjahyökkäykset ovat nopeita, mutta niiden onnistumisprosentti ei ole aina kovin korkea. Viimeiseksi sateenkaaritauluhyökkäykset ovat nopeita ja tehokkaita, mutta vaativat hyvin suuren levymuistikapasiteetin. Lisäksi ne ovat varsin epäkäytännöllisiä, jos suolaa on käytetty, sillä tällöin tiivistearvo ei vastaa suoraan haluttua salasanaa. Tässä tutkielmassa keskitytään sanakirjahyökkäyksiin.

3.1. Salasanan kompleksisuus suojana sanakirjahyökkäyksiä vastaan

Nykyisin useat sivustot vaativat käyttäjään valitsemaan salasanan, joka on tarpeeksi pitkä ja sisältää ainakin yhden numeron sekä ison kirjaimen. Tällöin hyökkääjällä menisi huomattavasti enemmän aikaa salasanan murttamiseen raakavoimahyökkäyksellä, koska mahdollisten salasanojen määrä nousee.

Ihmisillä on kuitenkin usein taipumusta sijoittaa kyseiset pakolliset merkit joko salasanan alkuun tai loppuun, jotta se olisi helpompi muistaa. Esimerkiksi Tatli [2015] osoittaa tutkimuksillaan, että ihmiset lisäävät erittäin usein merkkisarjan "123" salasanan loppuun. Toisena kohtana Tatli tuo esiin samojen luonnollisen kielen sanojen esiintymisen peräkkäisinä (esimerkiksi "kingking"). Kolmantena kohtana hän mainitsee kirjainten korjaamisen numeroilla, mikä tapahtuu usein tiettyjen kirjainten ja numeroiden välillä niiden esteettisen samankaltaisuuden takia (esimerkiksi o-kirjain ja numero 0). Nämä taipumukset tekevät monesti näennäisesti vahvasta salasanasta heikon.

Salasanat, joilla on aidosti suuri entropia, ovat pitkiä ja täysin satunnaisesti luotuja. Näin ollen on hyvin epätodennäköistä, että ne sisältäisivät luonnollisen kielen sanoja. Valitettavasti ihmiset muistavat parhaiten juuri luonnollisen kielen sanoja ja ilmaisuja, joten ei ole kohtuullista vaatia kaikilta käyttäjiltä tällaista täydellistä salasanaa.

Asiaa voidaan kuitenkin harkita uudelleen, jos tilanne muuttuu niin, että käyttäjän ei tarvitse muistaa omaa salasanaansa. Tämä voidaan saavuttaa käyttämällä sovellusta, joka hallitsee salasanvoja käyttäjän puolesta (password manager tai lyhyesti PM). PM auttaa käyttäjää salasanojen luomisessa, tallentamisessa ja organisoimisessa. Käytännössä PM-sovellukset toimivat niin, että ne luovat ensin paikallisen tietokannan, jonne ne tallentavat käyttäjien kaikki salasanat salattuina [Arias-Cabarcos *et al.* 2016]. Tällöin käyttäjän tarvitsee muistaa vain PM-sovelluksen oma salasana.

Turvallisuuden liittyvien parannuksien lisäksi PM-sovellukset takaavat myös taloudellisia hyötyjä, kuten vähentyneet kustannukset salasanoihin liittyvissä asiakaspalveluissa sekä työntekijöiden tuottavuuden lisääminen nopeuttamalla kirjautumisprosesseja [Arias-Cabarcos *et al.* 2016]. Näiden etujen takia PM-sovelluksen hyödyntäminen on varmasti harkitsemisen arvoista.

3.2. Yhteydettömät sanakirjahyökkäykset

Yhteydettömät sanakirjahyökkäykset ovat luonteeltaan epäinteraktiivisia, sillä hyökkääjän ei tarvitse olla vuorovaikutuksessa palvelimen kanssa. Tämä onnis-

tuu kuitenkin vain, jos hyökkääjä pääsee käsiksi salasanatiivisteisiin tai onnistuneen haaste-vastaus-protokollan tietoihin. Jälkimmäisessä tapauksessa hyökkääjä pystyy ohittamaan naiivin haaste-vastaus-protokollan käyttämällä sanakirjaa, sillä opittuaan haasteen ja vastauksen hän pystyy vertaamaan jokaista sanakirjan salasanan luomaa vastausta oikeaan vastaukseen. Kyseisen ongelman takia on kehitetty parempia protokollia. Ensimmäinen näistä on Bellovin ja Merritin [1992] suunnittelema sala-avain vaihdon protokolla, jossa hyödynnetään yhdistetysti symmetristä ja asymmetristä kryptografiaa. Tavoitteena on luoda sala-avain onnistuneen protokollan ajon jälkeen. Tämän jälkeen molemmat osapuolet pystyvät käyttämään sala-avainta myöhempien viestien salaamiseen.

Salasanatiivisteiden tapauksessa kyseisten tietojen suojaaminen on paras keino välttyä tältä uhkalta. Edellisessä luvussa mainitut tavat suojautua SQL-injektioilta ovat hyvin tärkeitä tässä yhteydessä. Samalla voidaan turvautua myös parempiin tiivistefunktioihin, mutta tässä yhteydessä parempi merkitsee usein monimutkaisempaa, mikä hidastaa aitojen käyttäjien kirjautumista [Juels and Rivest 2013].

Toisaalta on yleisesti järkevää varautua pahimpaan ja olettaa, että tiivistetiedot ovat varastettavissa, vaikka suojaus olisikin tehty hyvin. Tähän Juels ja Rivest [2013] ehdottavat omana ratkaisunaan "hunajasanoja". Heidän ratkaisunsa perustuu ideaan, jossa hyökkääjää pyritään huijaamaan virheellisillä tiedoilla. Tietokannan ylläpitäjät ovat kautta aikojen pystyneet luomaan valheellisia käyttäjätilejä tai jopa valheellisia tietokantatauluja. Näitä kutsutaan "hunajapurkeiksi" ja niiden tarkoitus on toimia joko ansana tai hämäyksenä hyökkääjälle. Ongelmana on se, että nokkelat hyökkääjät pystyvät erottamaan valheellisen käyttäjätilin tai tietokantataulun oikeasta. Juels ja Rivest vievät tämän konseptin askeleen eteenpäin, sillä heidän ratkaisussaan on kyse koko järjestelmän ansoittamisesta. Tämä tapahtuu luomalla jokaiselle käyttäjälle useamman kuin yhden salasanan, joista vain yksi on oikea. Loput ovat hunajasanoja, joiden käyttö kirjautumisprosessissa lähettää hälytyksen järjestelmän ylläpitäjälle paljastaen hyökkäysyrityksen.

Hunajasanojen yhteydessä täytyy olettaa, että järjestelmällä on pääsy erilliseen apupalvelimeen, jossa säilytetään tarvittavaa salaista tietoa. On tietenkin mahdollista, että myös tämä erillinen palvelin voidaan vaarantaa, mutta tämä ainakin hidastaisi hyökkääjää. Lisäksi tämän palvelimen suojaus on suhteellisesti helpompaa, koska sen sisältö on huomattavasti yksinkertaisempaa [Juels and Rivest 2013].

Hunajasanojen käyttö ei takaa varmaa suojausta yhteydettömiltä sanakirjahyökkäyksiltä, mutta ne aiheuttavat epävarmuutta hyökkääjässä. Murrettuaan tiivisteen hyökkääjä ei pysty olemaan varma salasanan oikeellisuudesta. Sen sijaan hän joutuu ottamaan riskin yrittäessään kirjautua sisään. Tämän metodin etuja ovat ehdottomasti sen toteuttamisen helppous ja suojauksen toimiminen jokaisen käyttäjän kohdalla. Esimerkiksi hunajakäyttäjätilien käytössä ei voida suojata kaikkia käyttäjiä.

3.3. Yhteydelliset sanakirjahyökkäykset

Yhteydellisissä sanakirjahyökkäyksissä hyökkääjä on jatkuvasti vuorovaikutuksessa palvelimen kanssa yrittäessään kirjautua sisään. Tästä syystä yhteydellistä hyökkäystä vastaan on helpompi puolustautua kuin yhteydetöntä. Pinkas ja Sander [2002] esittävät useita keinoja puolustukseen:

1. Viivästynyt vastaus: Saatuaan käyttäjätunnuksen ja salasanan palvelin lähettää hieman viivästyneen vastauksen. Tämä lisää hyökkääjältä vaadittavaa aikaa huomattavasti ja tekee prosessista mahdollisesti kannattamattoman. Toisaalta hyökkääjä pystyy ajamaan useita sisäänkirjautumisyrittäyksiä rinnakkain, jolloin tästä ratkaisusta ei ole hyötyä.
2. Käyttäjätilin lukitseminen: Jos sama käyttäjä yrittää kirjautua sisään jatkuvasti, järjestelmä voi lukita kyseisen käyttäjätilin tietyn määrän epäonnistuneiden yritysten jälkeen. Hyökkääjä voi kuitenkin käynnistää palvelunestohyökkäksen lukitsemalla käyttäjätilejä, jolloin oikeille käyttäjille aiheuttuisi hyvin paljon haittaa. Tätä varten tarvittava asiakaspalvelu ei liene kovin kannattavaa.
3. Ylimääräisen laskennan suorittaminen: Tämän ratkaisun ideana on vaatia käyttäjää tekemään jotain ylimääräistä laskentaa yrittäessään kirjautua sisään, jolloin prosessi hidastuisi. Kyseisen laskennan suorittaminen yhden kerran olisi suoritettavissa, mutta lisääntyneet määrät olisivat liian raskaita hyökkääjälle. Heikkoutena on se, että hyökkääjillä on usein tehokkaampi laitteisto kuin tavallisilla käyttäjillä, joten prosessia on vaikea tehdä vaativaksi ja samalla myös käyttäjäystävälliseksi.
4. Käänteinen Turingin testi: Nämä ovat kenties tunnetumpia nimellä CAPTCHA (Completely Automated Public Turing Tests to Tell Computers and Humans Apart). Tarkoituksena on luoda tehtäviä, jotka ovat helppoja ihmisille, mutta samalla mahdottomia tietokoneille. Mahdottoisuus tässä tarkoittaisi käytännössä sitä, että tietokone ei pystyisi ratkai-

semaan tehtävää, vaikka tehtävän takana oleva algoritmi olisi tuttu sille kokonaisuudessaan. CAPTCHA-testeissä on erityisen tärkeää se, että testin läpäiseminen satunnaisella arvauksella on hyvin epätodennäköistä. Esimerkiksi testi, jossa pyydetään tunnistamaan henkilön sukupuoli, on varsin huono. Kuvassa 1 on CAPTCHA, joka sisältää hieman vääristyneitä sanoja. Ihmiselle nämä on helppo tunnistaa, kun taas tietokoneelle ne voivat olla yllättävän vaikeita.

Edellä mainituista ratkaisuista käänteiset Turingin testit sisältävät vähiten heikkouksia ja ovat näin ollen tehokkain tapa puolustautua yhteydellisiä sanakirjahyökkäyksiä vastaan. Niiden soveltaminen tunnistautumisprosessiin on kuitenkin ollut haastavaa, sillä testin ratkaiseminen jokaisen kirjautumisen yhteydessä voi olla varsin kiusallista tavallisille käyttäjille. Tämän takia näitä testejä käytetään nykyisin lähinnä lomakkeiden täytössä tai muissa toimenpiteissä, jotka eivät tapahdu yhtä usein kuin kirjautuminen. Käänteisiä Turingin testejä ei kuitenkaan tarvitse käyttää jokaisen kirjautumisyrityksen yhteydessä. Sen sijaan voidaan olettaa, että oikea käyttäjä ei käytä säännöllisesti kovin useaa eri laitetta kirjautumiseen. Näin ollen testiä voidaan käyttää vain silloin, kun käyttäjä kirjautuu uudesta laitteesta tai vastaanotettu salasana on väärä [Chakrabarti and Singhal 2007].

4. Yhteenveto

Puolustautuminen sanakirjahyökkäyksiltä alkaa ongelman ehkäisyllä. Tähän päästään puolustavalla koodauksella ja huolellisella SQL-injektiohaavoittuvuuksien testauksella. Samalla on muistettava tietoturvallisen protokollan ja tiivistefunktioiden käyttö. Nämä ratkaisut yhdessä hunajasanojen kanssa muodostavat suurimman osan puolustuskeinoista yhteydettömiä sanakirjahyökkäyksiä vastaan.

Koska yhteydelliset sanakirjahyökkäykset ovat luonteeltaan interaktiivisempia, puolustus on myös aktiivisempaa. Esimerkiksi käyttäjätilin lukitseminen on hyvin yleisesti käytetty ratkaisu, joka ei käytännössä ole kovin kannattavaa. Parempana ratkaisuna toimivat käänteiset Turingin testit, sillä kunnollisesti toteutettuna ne ovat mahdottomia tietokoneille. Täydellistä ratkaisua ei olla kuitenkaan löydetty, koska se, mikä annettuna hetkellä on mahdotonta tietokoneille voi olla vuosien päästä yksinkertaista.

Salasanan kompleksisuus vaikeuttaa hyökkääjän työtä lisäämällä murtamiseen tarvittavaa aikaa raakavoimahyökkäyksen yhteydessä. Kun hyökkääjällä on käytössään sanakirja, kompleksisuuden hyödyt saattavat pienentyä huomattavasti, jos sanakirjassa on hyödynnetty tietoa käyttäjien taipumuksista suunnitella "vaikea" salasana tietyllä tavalla. Näistä taipumuksista voidaan päästä eroon käyttämällä PM-sovellusta, joka keksii jokaiseen palveluun erittäin vahvan salasanan käyttäjän puolesta.

Viiteluettelo

- Patricia Arias-Cabarcos, Andrés Marín, Diego Palacios, Florina Almenárez and Daniel Díaz-Sánchez. 2016. Comparing password management software: toward usable and secure enterprise authentication. *IT Professional* 18, 5, 34 – 40.
- Steven M. Bellovin and Michael Merritt. 1992. Encrypted key exchange: password based protocols against dictionary attacks. In: *Proc. of the 1992 IEEE Computer Society Symposium on Research in Security and Privacy*, 72 – 84.
- Stephen W. Boyd and Angelos D. Keromytis. 2004. SQLrand: preventing SQL injection attacks. In: *Proc. of the Applied Cryptography and Network Security, Lecture Notes in Computer Science* 3089. Springer, 292 – 302.
- Saikat Chakrabarti and Mukesh Singhal. 2007. Password-based authentication: preventing dictionary attacks. *Computer* 40, 6, 66 – 74.
- Justin Clarke. 2012. *SQL injection attacks*. Waltham, Mass: Syngress.
- Jose Fonseca, Marco Vieira and Henrique Madeira. 2009. Vulnerability & attack injection for web applications. In: *Proc. of the IEEE/IFIP International Conference on Dependable Systems & Networks*, 93 – 102.
- Praveen Gauravaram. 2012. Security analysis of salt||password hashes. In: *Proc. of the International Conference on Advanced Computer Science Applications and Technologies*, 25 – 30.
- Ari Juels and Ronald L. Rivest. 2013. Honeywords: making password cracking detectable. In: *Proc. of the 2013 ACM SIGSAC Conference on Computer and Communications Security*, 145 – 160.
- Anyi Liu, Yi Yuan, Duminda Wijesekera and Angelos Stavrou. 2009. SQLProb: a proxy-based architecture towards preventing SQL injection attacks. In: *Proc. of the 2009 ACM Symposium on Applied Computing*, 2054 – 2061.
- Lawrence O’Gorman. 2003. Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE* 91, 12, 2021 – 2040.

- Chen Ping. 2017. A second-order SQL injection detection method. In: *Proc. of the IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference*, 1792 – 1796
- Benny Pinkas and Tomas Sander. 2002. Securing passwords against dictionary attacks. In: *Proc. of the 9th ACM Conf. Computer and Comm. Security*, 161 – 170.
- Lwin Khin Shar and Hee Beng Kuan Tan. 2013. Defeating SQL injection. *Computer* 46, 3, 69 – 77.
- Mike Shema. 2012. *Hacking Web Apps: Detecting and Preventing Web Application Security Problems*. William Andrew.
- Emin Islam Tatli. 2015. Cracking more password hashes with patterns. *IEEE Transactions and Information Forensics and Security* 10, 8, 1656 – 1665.