

Vaiheittaisten ohjeiden rakentuminen suomenkielisissä käyttöohjeissa

Salla Hanhineva
Tampereen yliopisto
Viestintätieteiden tiedekunta
Monikielisen viestinnän ja käännöstieteen maisteriopinnot
Englannin kääntämisen ja tulkkauksen opintosuunta
Pro gradu -tutkielma
Toukokuu 2018

Tampereen yliopisto
Viestintätieteiden tiedekunta
Monikielisen viestinnän ja käännöstieteen maisteriopinnot
Englannin kääntämisen ja tulkkauksen opintosuunta

HANHINEVA, SALLA: Vaiheittaisten ohjeiden rakentuminen suomenkielisissä käyttöohjeissa

Pro gradu -tutkielma, 67 sivua + 1 liite ja englanninkielinen lyhennelmä, 10 sivua
Toukokuu 2018

Tässä pro gradu -tutkielmassa tarkastellaan vaiheittaisten ohjeiden rakennetta suomenkielisissä käyttöohjeissa. Vaiheittaiset ohjeet ovat yksi keskeinen osa teknisen viestinnän lopputuotteita ja siten myös tutkimusta. Van der Meij ym. (2003) ovat luoneet kyseisten ohjeiden tarkasteluun nelikantamallin (*Four Components Model*), jota jakaa niiden piirteet neljään kategoriaan: tavoitteisiin, edellytyksiin, toimintoihin ja reaktioihin sekä ei-toivottuihin tilanteisiin. Tutkielmassa arvioidaan, sopiiko malli suomenkielisten ohjeiden arviointiin.

Nelikantamallin lisäksi tutkimuksen teoreettinen viitekehys koostuu proseduraalisesta diskurssista ja sen taustoista. Tutkielmassa käsitellään etenkin kyseistä diskurssin muotoa ja sitä, miten ongelmanratkaisun mallit ja klassinen retoriikka yhdistyvät siinä.

Tutkimuksen aineisto koostuu yhteensä 36 yksittäisestä vaiheittaisesta ohjeesta, joista puolet on ohjelmisto- ja puolet laitteisto-ohjeita. Aineistoa analysoidaan nelikantamalliin avulla sekä kvalitatiivisesti että kvantitatiivisesti. Tuloksia vertaillaan mallin kehittäjien saamiin tuloksiin, ja erojen ja yhteneväisyyksien mahdollisia syitä tarkastellaan lähemmin.

Tutkimuksessa ilmenee, että nelikantamalli sopii pitkälti sellaisenaan myös suomenkielisten vaiheittaisten ohjeiden analysointiin. Suurin ongelma eivät ole kieli- tai kulttuurierot, vaan mallin paikoittainen tulkinnanvaraisuus. Tämä tutkielma esittääkin ehdotuksia tarkentavista määritelmistä silloin, kun alkuperäinen malli ei ole antanut tarkkoja ohjeita kulloinkin analysoitavan ja laskettavan yksikön määrittelyyn. Näistä määritelmistä tulee olemaan apua, jos nelikantamallia halutaan tulevaisuudessa soveltaa suomenkielisten käyttöohjeiden tarkasteluun. Eräs vaihtoehto jatkotutkimukseen on analyysin toteuttaminen suuremmalla aineistolla ja sen tulosten vertailu tähän tutkielmaan ja nelikantamallin kehittäjien tuloksiin.

Avainsanat: vaiheittainen ohje, vaiheittainen tieto, käyttöohje, proseduraalinen diskurssi

Sisällysluettelo

1	Johdanto.....	1
2	Proseduraalinen diskurssi.....	3
2.1	Ongelmanratkaisu.....	4
2.1.1	Ongelmaympäristöt.....	5
2.1.2	Järjestelmätilat ja toiminnot.....	7
2.2	Proseduraalisen diskurssin retoriset piirteet.....	10
2.3	Yksinkertaistetut vaiheet.....	11
2.4	Proseduraalisen diskurssin yhteenveto.....	14
3	Nelikantamalli.....	16
3.1	Tavoitteet.....	16
3.2	Edellytykset.....	18
3.2.1	Järjestelmätilat ja modulaarisuus.....	18
3.2.2	Käyttäjän taidot ja häivytykset.....	19
3.2.3	Käyttäjän tiedot ja linkitys.....	20
3.3	Toiminnot ja reaktiot.....	21
3.3.1	Itsenäisyys ja neuvominen (<i>Let Go and Support</i>).....	21
3.3.2	Kuvakaappaukset ja järjestelmän reaktiot.....	22
3.3.3	Vaiheet.....	22
3.4	Ei-toivotut tilat.....	23
3.4.1	Varoitukset.....	24
3.4.2	Ongelmanratkaisu.....	25
3.5	Nelikantamallin yhteenveto.....	26
4	Aineisto ja menetelmä.....	29
4.1	Tutkimusaineisto.....	29
4.2	Menetelmä.....	30
5	Aineiston analyysi.....	34

5.1	Tavoitteet	34
5.1.1	Tehtäväanalyysi eli tavoitteen saavuttaminen	34
5.1.2	Vaiheet	37
5.1.3	Otsikot	41
5.2	Edellytykset	43
5.3	Toiminnot ja reaktiot	46
5.3.1	Itsenäisyys ja neuvominen	46
5.3.2	Kuvat	46
5.3.3	Vaiheet	50
5.4	Ei-toivotut tilat.....	53
5.4.1	Varoitukset.....	53
5.4.2	Ongelmanratkaisu.....	57
6	Päätelmät	59
7	Lähteet	63
7.1	Aineisto.....	63
7.2	Kirjallisuus.....	65

1 Johdanto

Tämä pro gradu -tutkielma käsittelee käyttöohjeita, jotka ovat yksi teknisen viestinnän keskeisistä tutkimusalueista. Yksi käyttöohjeissa esiintyvistä informaatiotyypeistä on vaiheittaiset ohjeet (*procedures*), joiden avulla käyttäjää neuvotaan etenemään tehtävissä vaihe kerrallaan ja joka on siten tärkeä huomion kohde paitsi teknisessä viestinnässä, myös käyttäjien arjessa. Eräs teknisen viestinnän alan malli vaiheittaisten ohjeiden piirteiden erittelyyn on van der Meij'n, Blijlevenin ja Jansenin (2003) nelikantamalli (*Four Components Model*), jota käytän analysoidessani vaiheittaisten ohjeiden ominaispiirteitä. Mallin keskeisille termeille *procedure* ja *procedural information* on suomenkielisissä teksteissä useita eri käännöksiä. Käytän niistä tässä tutkielmassa Pohjolan (2007, 6) käännöksiä ”vaiheittainen ohje” ja ”vaiheittainen tieto”, sillä ne kuvaavat nimensä mukaisesti tietyn prosessin etenemistä vaiheesta toiseen.

Van der Meij'n ym. (2003) nelikantamalli jakaa vaiheittaisen tiedon neljään osatekijään: tavoitteisiin, edellytyksiin, toimintoihin ja reaktioihin sekä ei-toivottuihin tilanteisiin. Tavoitteet (*goals*) ja mahdolliset osatavoitteet (*subgoals*) ovat tiloja (*states*), jotka käyttäjä haluaa saavuttaa, ja ne kuvataan usein jo tehtävälistan otsikossa. Edellytykset (*prerequisites*) ovat ehtoja, jotka käyttäjän on täytettävä, jotta hän voi aloittaa tehtävän ja onnistua siinä. Käyttäjä suorittaa toimintoja (*actions*) päästääkseen tavoitteiseensa, ja tuote reagoi (*react*) toimintoihin. Ei-toivotut tilat (*unwanted states*) ovat vaara- ja ongelmatilanteita, joita käyttäjää neuvotaan välttämään ja korjaamaan. Van der Meij ym. (2003) selvittivät tutkimuksessaan, miten vaiheittaisen tiedon eri osatekijät esitettiin laitteisto- ja ohjelmistokäyttöohjeissa ja millaista parannettavaa niissä oli. Nelikantamallin lisäksi käytän tutkielman teoriapohjana proseduraalista diskurssia (*procedural discourse*), johon malli pohjautuu ja joka kuvaa diskurssia, joissa käyttäjiä neuvotaan tehtävän toteuttamisessa (Farkas 1999).

Nelikantamallia ei ole ennen sovellettu suomenkielisten tai nykyaikaisten ohjeiden arvioinnissa, joten tavoitteeni on tuottaa alkuperäisen tutkimuksen kohdalta uudenlainen viitekehys suomenkielisten vaiheittaisten ohjeiden analysointiin. Tutkimuskysymykseni ovat täten seuraavat:

- Millainen rakenne vaiheittaisilla ohjeilla on suomenkielisissä käyttöohjeissa?

- Sopiiko nelikantamalli suomenkielisten ohjeiden analysointiin?

Van der Meij'n ym. (2003) aineisto oli kattava – yhteensä 104 manuaalia – mutta heidän tutkimuksensa on 15 vuotta vanha, ja heidän tutkimansa ohjeet oli julkaistu vuosina 1991–1998. On kiinnostavaa nähdä, näkyykö kielen ja kulttuurin vaikutus vaiheittaisten ohjeiden rakenteessa, sillä erilaisten tekstien rakenne poikkeaa usein suomen ja englannin välillä. Mallia pitänee siis muokata jonkin verran, että se soveltuu suomenkielisiin ohjeisiin. Esimerkiksi otsikoiden (eli tavoitteiden) rakenne on luonnollisesti erilainen englannissa ja suomessa. Tavoitteeni on, että muokattua mallia voitaisiin käyttää pohjana tulevissa tutkimuksissa, jotka käsittelevät suomenkielisten vaiheittaisten ohjeiden piirteitä.

Van der Meij ym. (2003) analysoivat 52 ohjelmisto- ja 52 laitteistokäyttöohjetta, ja vaikka he valitsivatkin vain yhden vaiheittaisen ohjeen jokaisesta käyttöohjeesta, 104 käyttöohjeen tutkiminen olisi ollut liian suuri työ yhdelle henkilölle. Sen vuoksi aineistoni koostuu 18 ohjelmisto- ja 18 laitteisto-ohjeesta eli yhteensä 36 yksittäisestä vaiheittaisesta ohjeesta, jotka olen esitellyt tarkemmin lähdeluettelossa ja liitteessä 1. Koska tutkielmani keskittyy pitkälti nelikantamallin sovellettavuuteen, käytän tutkimusmenetelmänä teoriapohjaista sisällönanalyysia. Toisin sanoen luokittelen aineistoni jo olemassa olevan viitekehyksen pohjalta ja testaan tämän kehykseen toimivuutta uudessa kontekstissa (Tuomi & Sarajärvi 2018, 94–95).

Aloitan nelikantamallin pohjustamisen luvussa 2 proseduraalisen diskurssin ja sen taustan esittelyllä. Pääasiallinen lähteeni on teknisen viestinnän kentällä tunnettu, Farkasin (1999) kirjoittama artikkeli, jossa hän avaa proseduraalisen diskurssin taustoja ja hahmottelee niin kutsuttuja yksinkertaistettuja työvaiheita, joiden pohjalta van der Meij ym. (2003) sittemmin kehittivät nelikantamallin. Esittelen itse mallin yllämainittuine osatekijöineen luvussa 3. Luvussa 4 esittelen hyödyntämäni aineiston sekä tutkimusmenetelmän, joita käytin sitä analysoidessani. Käyn aineiston analyysin läpi luvussa 5, ja luvussa 6 kertaan aineistosta tekemiäni johtopäätöksiä ja pohdin tutkimukseni luotettavuutta, rajoitteita sekä mahdollisia jatkotutkimuksen aiheita.

2 Proseduraalinen diskurssi

Tässä tutkimuksessa keskeisellä sijalla oleva nelikantamalli pohjautuu Farkasin (1999) proseduraalista diskurssia käsittelevään artikkeliin, ja esittelen kyseisen viitekehyksen ja sen taustat tässä luvussa. Nivon nelikantamallin tällä tavoin sen laajempaan kontekstiin eli 1990- ja 2000-lukujen vaihteessa tehtyihin, vaihteista tietoa koskeviin tutkimuksiin. Lisäksi käytän proseduraalisen diskurssin käsitteitä aineistoni analyysin tukena luvussa 5. Proseduraalisen diskurssin kannalta keskeinen termi on *järjestelmä (system)*, jonka kanssa käyttäjät ovat tekemisissä ja joka tarkoittaa keskenään vuorovaikutuksessa olevien osien verkostoa (Skyttner 2005, 45). Järjestelmät voivat olla luonnollisia tai keinotekoisia, mutta tämä tutkielma käsittelee ihmisten valmistamia ja käyttämiä järjestelmiä, kuten ohjelmistoja.

Diskurssi on hyvin laaja käsite, jonka määritelmä vaihtelee tieteenalasta ja muusta kontekstista riippuen. Farkas ei selvennä, mitä määritelmää hän tässä yhteydessä käyttää, mutta Pietikäinen ja Mäntynen (2009, 19) kutsuvat diskurssia kielenkäytöksi, joka on osa sosiaalista toimintaa. Termi *proseduraalinen diskurssi* viittaa puhuttuun ja kirjoitettuun diskurssiin, joka ohjaa ihmisiä tehtävien suorittamisessa (Farkas 1999, 42). Proseduraalisen diskurssin määrittely vain puhutuksi ja kirjalliseksi viestinnäksi tosin on jossain määrin rajoittava kuvaus, sillä diskurssi voi olla ”kirjoitettua, puhuttua, viitottua, piirrettyä tai vaikkapa valokuvattua ja sisältää sekä kielellisiä että muita semioottisia merkkejä” (Pietikäinen & Mäntynen 2009, 19). On tosin mahdollista, että Farkas viittaa kirjoitetulla diskurssilla kaikkeen ei-puhuttuun viestintään, jolloin hänen määritelmänsä sisältäisi myös piirroksia ja valokuvat, joilla on käyttöohjeissa tärkeä rooli paitsi työvaiheiden havainnollistamisen (ks. luku 5.3.2), myös lukijan motivoinnin ja hänen huomionsa kiinnittämisen kannalta (Mäkyne 2012, 27).

Viestinnän tarkasta muodosta riippumatta proseduraalinen diskurssi saa alkunsa tilanteista, joissa käyttäjät tai lukijat haluavat toteuttaa tehtäviä, ja kertoo, kuinka tietystä tilanteesta siirrytään toiseen tilanteeseen. Toisin sanoen se kuvailee erilaisia järjestelmätiloja (*system states*) ja toimintoja, jotka muuttavat nämä tilat toisenlaisiksi. (Farkas 1999, 42.) Määrittelen nämä tilat ja toiminnot tarkemmin luvussa 2.1.2. Vaiheittainen ohje, joka neuvoo yksittäisen tehtävän toteuttamisessa, taas on eräs proseduraalisen diskurssin muodoista. Yksinkertaisimmillaan ohje koostuu vaiheista, joista jokainen neuvoo, kuinka käyttäjä voi muuttaa järjestelmän kulloisenkin

tilan toisenlaiseksi (van der Meij ym. 2003, 130). Farkas (1999, 42) kuitenkin huomauttaa, etteivät kaikki proseduraalisen diskurssin muodot sisällä selkeästi eriteltyjä työvaiheita. Esimerkiksi suullisen ohjeistuksen tarkoitus on auttaa vastaanottajaansa pääsemään tiettyyn päämäärään, mutta se ei useimmiten ole yhtä jäsenneltyä kuin kirjallisesti esitetty, numeroitu vaiheittainen ohje.

Vaikka vaiheittaisten ohjeiden esitystavat ja yksityiskohdat vaihtelevat, niiden rakenne on Farkasin (1999, 42) mukaan aina sama. Tämä olikin varmasti totta artikkelin kirjoittamisen aikaan, josta on kuitenkin tämän tutkielman kirjoittamishetkellä melkein 20 vuotta. Tuotteet, joita käyttöohjeet käsittelevät ja käyttöohjeiden julkaisualustat ovat kuitenkin kehittyneet huomattavasti sitten 1990-luvun lopun. Kuten mainitsin johdannossa, tämän tutkielman tarkoitus onkin selvittää, kuinka hyvin Farkasin ja sitä kautta van der Meij'n ym. oletukset pätevät 2010-luvun lopulla ja suomalaisessa kontekstissa.

Farkasin (1999) tutkimuksen teoreettinen tausta pohjautuu ennen kaikkea retoriikkaan ja ihmisten suorittamaan ongelmanratkaisuun. Esittelen seuraavissa alaluvuissa nämä aiheet siinä määrin kuin ne liittyvät tähän tutkielmaan. Niiden jälkeen esittelen vielä vaiheittaisten ohjeiden yksinkertaistetut työvaiheet, joihin Farkasin artikkeli erityisesti keskittyy.

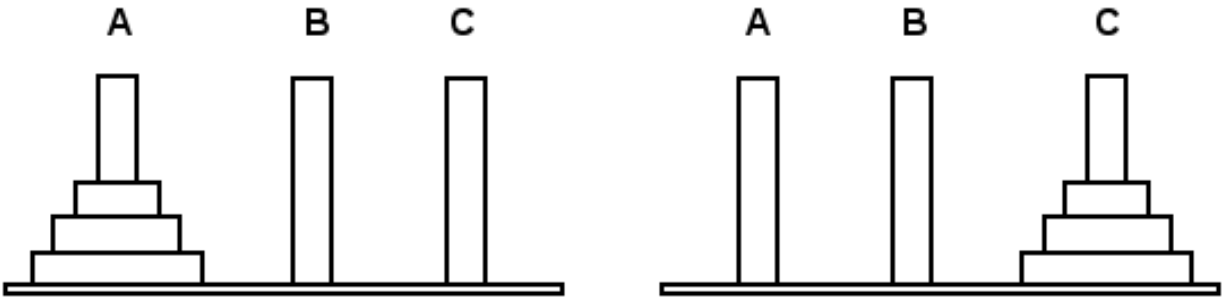
2.1 Ongelmanratkaisu

Farkasin (1999) näkemykset ongelmanratkaisusta perustuvat pitkälti Newellin (1980) artikkeliin, jossa Newell havainnollistaa ihmisten suorittamaa ongelmanratkaisua niin kutsuttujen ongelmaympäristöjen (*problem spaces*) avulla. Hänen käsitteissään on samoja piirteitä kuin Farkasin näkemyksessä proseduraalisesta diskurssista ja sitä kautta van der Meij'n ym. nelikantamallissa, joten esittelen ne tässä yhteydessä. Olen myös täydentänyt vaiheittaisten ohjeiden analyysiani Newellin ja Farkasin käsitteillä niissä tilanteissa, kun nelikantamalli on ollut tulkinnanvarainen tai ei ole antanut tarkkoja ohjeita luokitteluun. Termi *ongelmaympäristö* on oma käänökseni, joka ei täysin vastaa englanninkielistä alkuperäistermiä. Osuvampi käänös olisi *ongelmatila*, joka saattaisi kuitenkin sekoittaa järjestelmä- ja muihin tiloihin, joita käsittelen samassa yhteydessä. Selvyyden vuoksi käänsin sanan *space* siis *ympäristöksi*.

2.1.1 Ongelmaympäristöt

Ongelmaympäristö on eräs tapa tavoitteellisen toiminnan organisointiin (Newell 1980, 4). Se ei nimestään huolimatta ole fyysinen paikka vaan mielensisäinen rakennelma (*mental construct*), joka voi kuitenkin johtaa käyttäjän toteuttamaan fyysisiä toimenpiteitä. Ihmiset päätyvät siihen aloittaessaan toiminnan, ja heidän toimintaansa on mahdollista kuvailla ja ennustaa kyseisen ympäristön perusteella (mts. 5). Ongelmaympäristö koostuu tiloista (mp.) sekä erilaisista käyttäytymisen yksiköistä eli operaattoreista (*operators*) (mts. 7), jotka vaikuttavat ympäristöön. Yksinkertaisimmillaan jokainen operaattori saa jonkin tilan syötteenä (*input*) ja tuottaa tulosteena (*output*) toisenlaisen tilan. Toisiaan seuraavat operaattorit muodostavat polun, joka yhdistää ongelmaympäristön eri tilat jatkumoksi. (Newell 1980, 5.) Vaiheittaiset ohjeet puolestaan opastavat käyttäjiä näiden polkujen läpi. Tämän tutkielman kannalta keskeisiä käsitteitä ovat etenkin operaattorit ja tilat, joihin palaan myöhemmin tässä luvussa sekä luvuissa 3 ja 5.

Ongelmaympäristöjen kontekstissa ongelma muodostuu siis tiloista, tarkemmin määriteltynä alkutiloista, polun rajoitteista ja päämäärätiloista. Käyttäjän toiminnan tavoite on löytää ongelmaympäristöstä pois vievä polku, joka alkaa alkutilasta, noudattaa rajoitteita ja päättyy päämäärätilaan. (Newell 1980, 4–5.) Tehtävän suorittamisessa rajoittaa muun muassa se, että vain yksi operaattori kerrallaan voi olla toiminnassa, mikä tarkoittaa, että tehtävä on suoritettava sarjana toisiaan seuraavia eikä samanaikaisia toimintoja. Lisäksi yksi senhetkinen operaattori vaikuttaa vain yhteen senhetkiseen tilaan, mistä seuraa vain yksi senhetkinen tulos eli uusi tila. Myös mahdollisia tiloja on rajallinen määrä. (mts. 7.) Newell (1980, 5–6) havainnollistaa ongelmaympäristöjen periaatteita Hanoin torni -pelin avulla (ks. tarkempi kuvaus esim. Honkala 2015, 1–2):



Tehtävä ja ongelman kuvaus	Ongelmaympäristö	Ongelma
<p>Lautaan on kiinnitetty kepit A, B ja C. Keppiin A on pinottu N kiekkoa suurimmasta pienimpään. Tehtävänä on siirtää kiekot samassa järjestyksessä keppiin C. Kiekon voi siirtää mihin tahansa keppiin, kunhan se on pinon päällimmäinen. Kiekkoa ei voi siirtää sitä pienemmän kiekon päälle.</p> <p>(Newell 1980, 6.)</p>	<p><i>Tilat:</i> N kiekkoa kolmessa kepillä satunnaisessa järjestyksessä.</p> <p><i>Operaattorit eli käyttäytymisen yksiköt, joiden kautta pelaaja suorittaa tehtävän:</i></p> <ul style="list-style-type: none"> • Kiekon poistaminen kepeistä ja sen siirtäminen toiseen keppiin. • Järjestyksen tunnistaminen kaavan (<i>pattern</i>) esiintymänä. <p>(Newell 1980, 6–7.)</p>	<p><i>Alkutila:</i> Vasemmanpuoleisen kuvan järjestyks.</p> <p><i>Päämäärätila:</i> Oikeanpuoleisen kuvan järjestyks.</p> <p><i>Polun rajoite:</i> Kiekkoa ei voi siirtää pienemmän kiekon päälle.</p> <p>(Newell 1980, 6.)</p>

Kuvio 1. Hanoiin tornin alku- ja päämäärätila ja ongelmaympäristö (Newell 1980, 6–7)

Kuten kuvio 1 osoittaa, Hanoiin tornin tapauksessa ympäristön tiloja ovat kaikki järjestykset, jonne käytössä olevat kiekot voidaan asettaa kolmeen keppiin. Operaattoreita puolestaan on kaksi: Ensimmäinen ottaa syötteenä satunnaisen kiekon ja kepin ja siirtää kiekon kyseisen kepin nokkaan, mistä seuraa tulosteena uusi tila. Toinen operaattori taas tuottaa symbolisen ilmentymän, joka

todentaa, että kiekkojen järjestys sopii tiettyyn kaavaan. Kaavoja ovat esimerkiksi ylläolevissa kuvissa esitetyt järjestykset sekä muunlaiset tilanteet, kuten ”keppi A on tyhjä” ja ”pieni kiekko on isomman kiekon alla, mikä rikkoo pelin sääntöjä”. Lisäksi Hanoin tornin ongelman kuvaus antaa selkeät alku- ja päämäärätilat ja määrittää rajoitukset, joiden puitteissa tehtävä täytyy suorittaa. (Newell 1980, 5.) Tornia voidaan kuvata myös muunlaisten ongelmaympäristöjen avulla. Esimerkiksi operaattori ”Kiekon poistaminen kepistä siirtäminen toiseen keppiin” voidaan jakaa kolmeksi operaattoriksi: ”Kiekon nostaminen”, ”Käden liikuttaminen” ja ”Kiekon asettaminen paikoilleen”. Tornipelin kontekstissa edellä mainitut rajoitteet ilmenevät muun muassa siten, että pelaaja voi pohtia vain yhtä liikettä kerrallaan ja liikkeestä voi seurata vain rajallinen määrä uusia tiloja. (Newell 1980, 7.) Farkas (1999) on määritellyt tilat ja toiminnot tarkemmin omassa artikkelissaan. Käsittelen kyseistä kahta käsitettä tarkemmin seuraavissa alaluvuissa, mutta Farkas ei ottanut teoriaansa mukaan rajoitteita. Niiden rajaaminen pois johtunee siitä, että rajoitteet ovat abstraktimpia kuin toiminnot ja niistä seuraavat tilat, jotka puolestaan ovat fyysisesti havaittavissa. Osa rajoitteista on myös itsestään selviä. Käyttäjälle ei tarvitse esimerkiksi selventää, että hän voi tehdä vain yhden toiminnon kerrallaan.

2.1.2 Järjestelmätilat ja toiminnot

Kuten mainitsin luvun 2 alussa, vaiheittainen ohje koostuu yksinkertaisimmillaan tiloista ja toiminnoista samoin kuin ongelmaympäristötkin (van der Meij ym. 2003, 130). Esimerkiksi Hanoin tornin tapauksessa eräs mahdollinen järjestelmätila on tilanne, jossa jokaisessa kepissä on yksi kiekko. Kiekkojen liikuttaminen puolestaan on toiminto, joka muuttaa kyseistä tilaa. Järjestelmätilat ja toiminnot pohjautuvat Newellin (1980; ks. 2.1.1) käsittelemiin, ongelmaympäristön sisältämiin tiloihin ja operaattoreihin.

Järjestelmätiloja ovat toivottu tila, edellytetty tila, välitilat ja ei-toivotut tilat (Farkas 1999, 42–43), joista osa esiintyy myös Newellin (1980) artikkelissa eri nimillä. Havainnollistan siksi Farkasinkin teoriaa Hanoin tornin avulla. Taulukon 1 kaksi ensimmäistä saraketta pohjautuvat Farkasin työhön, ja viimeinen sarake sisältää omia esimerkkejäni, jotka kehitin Newellin artikkelin pohjalta.

Taulukko 1. Järjestelmätilat (mukailtu Farkasin (1999, 42–43) pohjalta)

Tilan nimi	Tilan kuvaus	Tila Hanoin tornia pelattaessa
------------	--------------	--------------------------------

Toivottu tila	Käyttäjän tavoite.	Kiekot on siirretty kepeistä A keppiin C.
Edellytetty tila	Tila, jossa järjestelmän on oltava, jotta käyttäjä voi siirtyä toivottuun tilaan.	Kiekot on pinottu keppiin A alhaalta katsoen suurimmasta pienimpään.
Välitila tai osatavoite	Tila, joka johtaa kohti käyttäjän lopullista tavoitetta.	Kepissä A on kaksi kiekkoa ja kepeissä C yksi.
Ei-toivottu tila	Tila, jota käyttäjä haluaa välttää.	Suurempi kiekko on pienemmän kiekon päällä.

Esittelen taulukon 2 tiloja muuttavat toiminnot taulukossa 3. Näitä toimintoja ovat käyttäjän toiminnot, järjestelmän toiminnot ja ulkoiset tapahtumat (Farkas 1999, 43), jotka ovat verrattavissa Newellin (1980, 7) käsittelemiin operaattoreihin. Taulukon 2 tapaan olen lainannut kaksi ensimmäistä saraketta Farkasilta ja pohjannut viimeisen sarakkeen esimerkit Newellin teoriaan.

Taulukko 2. Järjestelmätiloja muuttavat toiminnot (mukailtu Farkasin (1999, 43) pohjalta)

Toiminnon nimi	Toiminnon kuvaus	Toiminto Hanoin tornia pelattaessa
Käyttäjän toiminto	Toiminto, jonka avulla käyttäjä pyrkii kohti päämääräänsä.	Kiekon liikuttaminen.
Järjestelmän toiminto	Järjestelmän reaktio käyttäjän toimintoihin, mistä seuraa uusi tila.	Kiekkojen määrä kepeissä C kasvaa.
Ulkoinen tapahtuma	Järjestelmän ulkopuolinen tapahtuma, joka saattaa vaikuttaa järjestelmään, kuten sähkökatkos.	Pöytä, jonka reunalle peli on asetettu keikahtaa, ja peli putoaa lattialle.

Toisin kuin vaikkapa tietokoneohjelmat, Hanoin torni on passiivinen järjestelmä, joten se ei varsinaisesti reagoi käyttäjän tekemisiin esimerkiksi antamalla virheilmoituksia tai estämällä käyttäjää siirtämästä kiekkoa sääntöjen vastaisella tavalla. Farkas (1999, 43) antaa havainnollisemman esimerkin: hän käyttää tavoitteellisesta toiminnasta esimerkkinä

vuorikiipeilyä, missä luonnon järjestelmä eli tässä tapauksessa vuori reagoi kiipeilijän liikkumiseen siten, että lumeen jää kiipeilijän jäljiltä polku. Kuten vuorikiipeilykuvaus osoittaa, tilat ja toiminnot eivät ole esillä vain tuotteissa, vaan kaikissa tilanteissa, joissa ihmiset pyrkivät johonkin päämäärään. Lisäksi tilat ja toiminnot eivät muodosta vain vaiheittaisen tiedon kokonaisuuksia, vaan myös yksittäiset työvaiheet muodostuvat niistä. (Farkas 1999, 43.) Tästä on esimerkkinä vaiheittainen ohje nimeltään ”Yhteyden muodostaminen Wi-Fi-verkkoon”, joka ei ole peräisin aineistostani, vaan erillisestä käyttöohjeesta:

Yhteyden muodostaminen Wi-Fi-verkkoon

1 Avaa ilmoituspaneeli vetämällä tilapalkista alaspäin.

2 Siirry Wi-Fi-asetusnäyttöön koskettamalla [Wi-Fi-kuvaketta] ja pitämällä sitä painettuna.

3 Kytke Wi-Fi päälle. Puhelin luettelee käytettävissä olevat Wi-Fi-verkot.

4 Valitse Wi-Fi-verkko, johon haluat muodostaa yhteyden: Jos verkko on salattu, syötä salasana, kun sitä kysytään. (Huawei 2017, 7.)

Ohje alkaa tilasta, jossa käyttäjä ei ole yhdistettynä Internetiin, mutta myös yksittäisillä vaiheilla on omat alkutilansa: vaihe ”Kytke Wi-Fi” päälle alkaa tilanteesta, jossa käyttäjä on siirtynyt Wi-Fi-asetusnäyttöön, ja verkon valitseminen alkaa tilanteesta, jossa puhelin on etsinyt kaikki käytettävissä olevat verkot.

Vaikka vaiheittaisen tiedon perusrakenne pysyy usein samana, järjestelmän toimikenttä (*domain*) vaikuttaa siihen, mitä toimintoja käyttäjän täytyy suorittaa ja millaisiin tiloihin hän päätyy. Jos järjestelmä on esimerkiksi hyvin monimutkainen, sen käyttöohje saattaa sisältää paljon selitystä ja välitilojen kuvailua, jotta käyttäjä pystyy seuraamaan, missä tilassa järjestelmä kulloinkin on. (Farkas 1999, 43.) Ajan kulumisen asettaa kuitenkin tämänkin Farkasin huomion kyseenalaiseen valoon. Tekniikan kehitys viimeisen kahden vuosikymmenen aikana on pakottanut käyttäjät omaksumaan uusia taitoja nopeampaan tahtiin kuin koskaan aiemmin (Arthur 2013, 143), mikä näkyy esimerkiksi kansainvälisessä PIAAC-tutkimuksessa (OECD 2016), jonka mukaan 80 prosenttia suomalaisista työntekijöistä käyttää tietokonetta työssään (Musset 2015, 28) ja noin 42 prosentilla aikuisista on hyvät tai erinomaiset tietotekniikkataidot. On siis todennäköistä, että 2010-luvun loppupuolen käyttäjät tarvitsevat vähemmän tukea ja järjestelmän kuvailua kuin käyttäjät Farkasin artikkelin julkaisun aikaan. Käyttäjien tarpeiden ennakointi ja arviointi liittyvät läheisesti retoriikkaan, joka on seuraavan alaluvun aihe.

2.2 Proseduraalisen diskurssin retoriset piirteet

Kuten käyttäjillä ja järjestelmillä, myös vaiheittaisella tiedolla on oma sosiaalinen kontekstinsa. Vaiheittaisen tiedon tapauksessa tämä konteksti on pääasiassa tiedon välittäminen. Ihmiset kommunikoivat keskenään ja ohjeistavat toisiaan päämääriin pääsyssä, joten proseduraalinen diskurssi on luonteeltaan retorista, samoin kuin kaikki muukin diskurssi. (Farkas 1999, 43.) Tiedon välittämisen lisäksi vaiheittaisella tiedolla saattaa olla muitakin tavoitteita, kuten käyttäjän vakuuttaminen siitä, että on valinnut itselleen juuri sopivan tuotteen.

Blackburn (2016) määrittelee retoriikan kielenkäytöksi, jonka tavoite on suostutella muita tai vaikuttaa heihin. Farkas (1993, 42) hyödyntää proseduraalisen diskurssin kuvailussa etenkin kahta retoriikan osa-aluetta. Ensimmäinen on klassinen retoriikkaa, jonka tavoite on sen kohteen luottamuksen voittaminen vetoamalla sekä hänen järkeensä että tunteisiinsa (Chandler & Munday 2016). Toinen on retoristen roolien konsepti. Käyttöohjeiden kirjoittajat voivat luoda itselleen ja lukijoilleen retoriset roolit, joiden pohjalta ohjeet kirjoitetaan. Tämän näkökulman mukaan lukijat eivät ole vain tiedon passiivisia vastaanottajia eivätkä kirjoittajat ole etäisiä ohjaajia, vaan molemmat osapuolet osallistuvat aktiivisesti käyttöohjeissa kuvattujen tavoitteiden toteuttamiseen. (Coney & Chatfield 1996, 24–25.) Retoriikan huomiointi on tärkeää etenkin varoituksia laadittaessa, sillä on erityisen tärkeää, että lukijat houkutellaan lukemaan ne.

Jotta käyttöohjeiden lukijat saataisiin klassisen retoriikan tavoitteiden mukaisesti luottamaan vaiheittaiseen tietoon ja noudattamaan sen ohjeita, käyttäjän täytyy ensinnäkin kokea ne itselleen sopiviksi. Uskoisin esimerkiksi, että liian laveasti selitetyt vaiheet saattavat hämmentää kokemattomia käyttäjiä, kun taas kokeneet käyttäjät saattavat kokea yksityiskohtaiset selitykset liiallisena holhoamisena. Vaiheittaisen tiedon täytyy siis sopia käyttäjän taustaan ja tarpeisiin sekä tilanteeseen, jossa he tarvitsevat ohjeita (Farkas 1999, 43). Kirjoittaja ei kuitenkaan aina tunne tarkkaa yleisöään eikä pysty sovittamaan tekstiään tietyille käyttäjille. Tarkemman tiedon puuttuessa eräs mahdollisuus on kirjoittaa käyttöohjeet niin sanottua pienintä yhteisintä tekijää ajatellen eli käyttäjille, joiden opastaminen täytyy aloittaa perustiedoista (Jayaprakash 2007, 151). Tasapainottelu aloittelevien ja kokeneiden käyttäjien tarpeiden välillä on kuitenkin haasteellista (mp.), ja jo vain pienimmän yhteisen tekijän määrittely vaatii käyttäjäkunnan jonkintasoista tuntemista.

Vaiheittaisen tiedon luotettavuutta lisää paitsi lukijan tarpeisiin mukautuminen, myös tiedon uskottavuus. Jos tiedonlähde ei vaikuta uskottavalta, lukija saattaa jättää sen huomiotta ja yrittää tehtävän suorittamista ilman ohjeita. (Farkas 1999, 43.) Ohjeiden uskottavuutta voidaan parantaa muun muassa ammattimaisesti toteutetulla ulkoasulla tai vetoamalla tuotteiden valmistajien maineeseen, mutta myönteiset ennako-odotukset eivät auta, jos itse ohjeet eivät täytä käyttäjien toiveita (mts. 44). On kuitenkin todennäköistä, että uskottavuuden tai luottamuksen puute eivät ole ainoita syitä, miksi käyttäjät eivät lue ohjeita. Niiden lisääminen ei siis takaa sitä, etteivät käyttäjät yrittäisi suorittaa tehtäviä omin päin. Novickin ja Wardin (2016, 16) haastattelemat käyttäjät listasivat online-käyttöohjeiden välttelyn syyksi muun muassa monimutkaisen navigoinnin, liian yksityiskohtaiset tai liian laveat ohjeistukset (mikä liittyy siihen, kokevatko käyttäjät ohjeita itselleen sopiviksi) ja tekniset ongelmat. Toisaalta muutamat käyttäjät moittivat ohjeiden puutteellista ulkonäköä (Novick & Ward 2016, 16), minkä Farkas mainitsee yhtenä osatekijänä, joka saattaa vähentää ohjeiden luotettavuutta. Yksinkertaistetut vaiheet, joihin syvennyn seuraavassa alaluvussa, antavat rajallisesti tilaa lukijan vakuuttamiseen, mutta toisaalta lukijoiden suostuttelu ei mielestäni aina vaadi monisanaisuutta. Pitkälliset vakuutukset saattavat joissakin tilanteissa olla jopa haitaksi, sillä jos käyttäjä on jo ostanut käyttöohjeiden kuvaileman tuotteen, hän kaippaa ensisijaisesti neuvoja, ei vahvistusta sille, että hän on tehnyt juuri oikean valinnan.

2.3 Yksinkertaistetut vaiheet

Kuten olen aiemmin todennut, Farkasin artikkeli (1999, 45) keskittyy erityisesti niin kutsuttuja yksinkertaistettuja vaiheita sisältävään vaiheittaiseen tietoon (*streamlined-step procedure*), jonka vaiheet ovat lyhyitä ja ytimekkäitä ja muotoilut yksinkertaista ja johon van der Meij'n ym. nelikantamalli pohjautuu. Tämänkaltaisille vaiheittaisille ohjeille ovat ominaisia seuraavat piirteet (Farkas 1999, 45):

- **Vaiheet ovat lyhyitä.** Ne sisältävät yhden toimintaa kuvailevan lauseen tai mahdollisesti kaksi toisiinsa läheisesti liittyvää lausetta. Tätä kuvausta saattaa seurata lyhyt selitys siitä, kuinka tuote reagoi käyttäjän toimintaan. Lisäksi vaiheissa voi olla lyhyitä esimerkkejä tai selityksiä.
- Vaikka käyttöohjeissa saatetaan käyttää kuvia tai taulukoita, **tekstin muotoilu on enimmäkseen yksinkertaista.**
- **Yksittäisen vaiheen ydin on imperatiiviverbi,** joka aloittaa kyseisen vaiheen päälauseen.

- **Tehtävä kuvaillaan suhteellisen lyhyesti ennen ohjeita.** Yleensä tehtävän luonne ilmoitetaan vain otsikolla tai otsikon ja lyhyen esittelykappaleen eli konseptuaalisen elementin avulla. Jos lisätietoa on paljon, se ilmoitetaan erikseen vaiheiden jälkeen, ei itse vaiheiden yhteydessä.
- Jos ohjeet on julkaistu sähköisesti, **käyttäjä pääsee hyperlinkkien kautta käsiksi lisätietoon** (esimerkiksi määritelmiin), jota ei ole liitetty vaiheiden yhteyteen.

Yksinkertaistetuilla vaiheilla on lisäksi osatekijöitä, jotka Farkas jakaa pakollisiin, lähes pakollisiin ja valinnaisiin. Hän ei perustele tai määrittele näitä kategorioita, mutta osatekijöiden kuvausten pohjalta uskon, että jaottelu perustuu siihen, kuinka tärkeitä ne ovat lukijalle tehtävän suorittamisen kannalta. Osatekijät ovat vaiheet, otsikko, konseptuaalinen elementti, infinitiivimuotoinen alaotsikko ja huomautukset, ja esittelen ne seuraavaksi tarkemmin.

Vaiheet (pakollinen) koostuvat käyttäjän toiminnan ja tuotteen reaktioiden kuvailusta, ja ne saattavat sisältää lisätietoa, jota käyttäjä tarvitsee tavoitteensa toteuttamiseen. Tällaista lisätietoa ovat muun muassa toiminnan kohteen sijainti, kuinka toiminto täytyy suorittaa (*Drag the handles so that they extend beyond the borders*), mikä toiminnon tarkoitus on ja miten käyttäjä voi saavuttaa vaihtoehtoisia tavoitteita. On kuitenkin tärkeää, ettei vaihtoehtoja anneta liikaa – tällöin käyttäjä joutuu kuluttamaan aikaa ylimääräiseen päätöksentekoon, ja sekä tuote että sen käyttöohje voivat vaikuttaa sekavilta. Käyttäjä voidaan myös ohjeistaa testaamaan, onko järjestelmä ei-toivotussa tilassa, joskin vakavimmat tilat ilmoitetaan virheilmoituksilla ja varoituksilla. Lisäksi käyttöohjeet voivat kuvailla, mitä palautetta järjestelmä antaa käyttäjän toiminnoista ja missä tilassa järjestelmä sillä hetkellä on, mutta tämä ei ole yleensä tarpeen interaktiivisten järjestelmien kohdalla, koska ne antavat itsessään palautetta käyttäjälle. (Farkas 1999, 46–48.) Palautetta voidaan antaa niin teksti- kuin kuvallisessakin muodossa.

Otsikko (lähes pakollinen) esittelee tehtävän, jota vaiheittainen tieto kuvailee ja erittelee sen muista tehtävistä. Jälkimmäinen on erityisen tärkeää, koska käyttäjät eivät lue ohjeita lineaarisesti alusta loppuun vaan etsivät tarvitsemaansa tietoa sisällysluettelosta. (Farkas 1999, 46.) Otsikon on siis oltava paitsi kuvaava, myös tarpeeksi tiivis, jotta lukija voi silmäillä sen nopeasti läpi. Toimintojen ja tilojen näkökulmasta otsikot kuvailevat perustoimintoja sekä vaiheittaisen tiedon päämäärää eli järjestelmän tilaa, johon ohjeiden noudattaminen johtaa. Otsikkoja on

englanninkielisissä ohjeissa neljänlaisia: nominilauseke (esimerkiksi *Macro automation*), gerundi (*Automating your work with macros*), ”juuri” (*Automate your work with macros*) ja infinitiivi (*To automate your work with macros*). Nominilauseke on niin ylimalkainen, ettei sitä juurikaan käytetä, kun taas gerundi on kuvaava ja siten yleisin otsikkotyyppi. ”Juuret” kuulostavat käskyiltä, mikä saattaa saada sisällysluettelot näyttämään vaihteelta ohjeelta. (Farkas 1999, 46.) Sisällysluettelot on kuitenkin lähes aina muotoiltu selkeästi eri tavalla ja sijoitettu eri paikkaan kuin varsinaiset ohjeet, joten lukijan hämmentymisen määrä lienee todellisuudessa pieni. Farkas (1999, 46) jatkaa, että infinitiivien *to*-rakenne puolestaan viestii tiiviistä syy-seuraussuhteesta, joten se edellyttää, että kirjoittaja siirtyy suoraan otsikosta vaihteelliseen tietoon ja jättää mahdollisen konseptuaalisen elementin väliin. Farkas ei kuitenkaan perustele tätä väitettä, eikä *to*-muotoinen otsikko mielestäni ole esteenä konseptuaalisen elementin sisällyttämiselle. Otsikot eivät tietenkään esiinny suomen kielessä täsmälleen edellä esitellyssä muodossa, mutta kerron tästä aiheesta enemmän analyysiluvussa 5.

Konseptuaalinen elementti (valinnainen) on yhden tai parin kappaleen pituinen selitys, joka laajentaa otsikkoa ja antaa muuta lisätietoa, jonka avulla käyttäjä voi päättää, suorittaako hän tehtävän. Lisäksi se saattaa kertoa, mitä käyttäjä pystyy tekemään sen alla olevan vaihteellisen ohjeen avulla varsinaisen päämäärän lisäksi. Konseptuaalisen elementin tulisi myös ilmoittaa tehtävän edellytykset ja mahdolliset ei-toivotut tilat, joita sen suorittamisesta voi seurata. (Farkas 1999, 46.) Ei-toivottu tila ei välttämättä ole niin vakava, että se edellyttäisi varoitusta. Jotkin ei-toivotut tilat ovat kuitenkin suoranaisia vaaratilanteita, joista lukijaa on varoitettava, minkä vuoksi laskisin konseptuaalisen elementin pakolliseksi. Varoitusten pakollisuus on myös säädetty laissa (Tukes 2012, 5).

Infinitiivimuotoinen alaotsikko (valinnainen) erottelee toisistaan tehtävät, jotka on kerätty saman otsikon alle. Tehtävien esittämisestä yhdessä on hyötyä, jos ne ovat läheisesti yhteydessä ja niitä on vaikea selittää erillään. Lisäksi toisiinsa liittyvien tehtävien kerääminen yhteen vähentää käyttöohjeiden aiheiden (*topics*) määrää, mikä helpottaa ohjeiden navigointia ja päivittämistä. (Farkas 1999, 46.) Tässä yhteydessä termi *aihe* viittaa sisältöyksiköihin, jotka erottuvat selkeästi toisistaan ja jotka esiintyvät erillään toisistaan, esimerkiksi eri ikkunoissa (Farkas 1999, 46). En kuitenkaan sanoisi, että alaotsikkojen pitäisi englanninkielisissä ohjeissa olla aina

infinitiivimuodossa. Mielestäni riittää, että samanmuotoisia ylä- ja alaotsikkoja käytetään yhtenäisesti saman käyttöohjeen sisällä.

Huomautukset (valinnainen) sisältävät lisätietoa, joka kuuluu vaiheiden ulkopuolelle. Tällaisia ovat esimerkiksi varoitukset, jotka esitetään joko konseptuaalisessa elementissä tai juuri ennen vaihetta, jota ne koskevat. ”Vihjeiden” tai muun samantapaisen otsikon alle kerätty tieto saattaa olla valinnaista eikä välttämättä kiinnosta kaikkia käyttäjiä. Koska ei-välttämättömät huomautukset pidentävät käyttöohjeita ja saattavat hämmentää käyttäjää, kirjoittajan on kuitenkin harkittava tarkkaan, onko niiden sisällyttäminen tarpeen. (Farkas 1999, 49.) Koska huomautukset voivat Farkasin määritelmään mukaan olla myös varoituksia, luokittelisin ne kuitenkin konseptuaalisen tiedon tapaan pakollisiksi. Kaikkein selkeintä olisi erottaa varoitukset kokonaan omaksi kategoriakseen ja luokitella se pakolliseksi. Näin ei jäisi epäselväksi, kuinka tärkeitä ne ovat sekä käyttöohjeen lukijan ja julkaisijan kannalta. Ihannetilanteessa lukija säästyy vahingolta, ja julkaisija ei joudu vastuuseen kyseisestä vahingosta.

2.4 Proseduraalisen diskurssin yhteenveto

Olen kuvailut edellisissä alaluvuissa proseduraalisesta diskurssista, johon nelikantamalli perustuu. Kyseinen diskurssi kertoo vastaanottajalleen, millaisia toimintoja suorittamalla hän voi siirtyä tietystä tilanteesta toiseen, kunnes hän saavuttaa tavoittelemansa tilan (Farkas 1990, 42). Tutkimani vaiheittaiset ohjeet ovat siten yksi proseduraalisen diskurssin muodoista, sillä ne neuvovat käyttäjiä tehtävien toteuttamisessa. Farkas (1999) perustaa näkemyksensä tällaisesta diskurssista klassiseen retoriikkaan ja ongelmanratkaisuun. Ongelmanratkaisun keskeinen käsite tässä yhteydessä on ongelmaympäristö, johon ihmiset päätyvät ryhtyessään tavoitteelliseen toimintaan. Ongelmaympäristö koostuu tilojen lisäksi operaattoreista eli käyttäytymisen yksiköistä, jotka muuttavat näitä tiloja. (Newell, 1980, 4–5, 7.) Tämän tutkimuksen kannalta keskeisiä termejä ovat *alkutilat* ja *päämäärätilat*. Alkutila on tila, jossa esimerkiksi käyttöohjeen käsittelemä järjestelmä on tehtävän alkaessa, ja päämäärätila on tavoite, jota kohti käyttäjä pyrkii (mp.).

Farkasin (1999, 42–43) järjestelmätilat ja toiminnot puolestaan pohjautuvat edellä mainittuihin ongelmaympäristöihin ja niiden sisältämiin tiloihin ja operaattoreihin. Järjestelmätiloja ovat toivottu tila, edellytetty tila, välitilat ja ei-toivotut tilat. Toivottu tila on käyttäjän tavoite. Edellytetty tila taas on tila, jossa järjestelmän on oltava, jotta käyttäjä voi suorittaa haluamansa

toiminnot ja siirtyä tehtävässä eteenpäin. Välitila tai osatavoite johtaa kohti käyttäjän lopullista tavoitetta, ja ei-toivottu tila on nimensä mukaisesti tila, jota käyttäjä haluaa välttää. Näitä tiloja muuttavat käyttäjän toiminnot, järjestelmän toiminnot ja ulkoiset tapahtumat, joista keskeisiä ovat kaksi ensimmäistä. Käyttäjän toiminto on toiminto, jonka avulla käyttäjä pyrkii kohti päämääräänsä, ja järjestelmän toiminto on reaktio siihen, mitä käyttäjä on tehnyt.

Farkas soveltaa proseduraalisen diskurssin retorisen ulottuvuuden määrittelyssä etenkin klassista retoriikkaa, joka pyrkii voittamaan kohteensa luottamuksen (Chandler & Munday 2016) ja suostuttelemaan häntä (Blackburn 2016). Luottamuksen heräämisen kannalta on tärkeää, että käyttöohjeiden lukija tuntee ohjeet itselleen ja tilanteeseensa sopivaksi. Tiedon on myös vaikutettava uskottavalta, tai lukija saattaa jättää ohjeiden lukemisen kesken (Farkas 1999, 43). Toinen Farkasin hyödyntämä käsite on retoriset roolit, jotka käyttöohjeiden kirjoittaja luo itselleen ja lukijoilleen (Coney & Chatfield 1996, 24–25).

Farkas (1999, 45) hyödyntää edellä mainittuja teoreettisia viitekehyksiä yksinkertaistettujen vaiheiden kuvailussa. Tällaiset vaiheet ovat lyhyitä ja pelkistetyksi muotoiltuja, ja van der Meij ym. (2003) pohjasivat niihin myöhemmin nelikantamallinsa. Pelkistettyjen vaiheiden osatekijät ovat vaiheet, otsikko, konseptuaalinen elementti, alaotsikko ja huomautukset. Vaiheet kuvailevat käyttäjän toimintaa ja siitä seuraavia tuotteen reaktioita, ja otsikko ilmoittaa tilan, johon käyttäjä toivoo pääsevänsä. Konseptuaalinen elementti on lyhyt selitys, joka laajentaa otsikkoa ja kertoo tehtävän edellyttämän järjestelmätilan sekä varoittaa käyttäjää ei-toivotuista tiloista, johon hän on vaarassa päätyä. Alaotsikot erottavat toisistaan tehtävät, jotka ovat saman pääotsikon alla, ja huomautukset sisältävät lisätietoa, joka kuuluu varsinaisten vaiheiden ulkopuolelle. Huomautusten sisältämä tieto ei aina ole välttämätöntä, mutta se saattaa olla tärkeää, kuten varoitusten tapauksessa. (Farkas 1999, 46–49.)

3 Nelikantamalli

Van der Meij, Blijleven ja Jansen (2003) kehittivät Farkasin (1999) yksinkertaistettujen vaiheiden pohjalta nelikantamallin, jonka on tarkoitus auttaa käyttöohjeiden kirjoittajia vaiheittaisten ohjeiden kirjoittamisessa ja tutkijoita niiden kuvailussa. Osa suosituksista pohjautuu puutteisiin, joita he löysivät käyttöohjeista, kun taas osa perustuu ohjeissa esiintyneisiin hyviin käytäntöihin. Van der Meij ja Gellevij (2004) esittelevät alkuperäisen tutkimuksen löydökset toistamiseen myöhemmässä artikkelissa, ja lisäksi he käsittelevät ongelmanratkaisuun liittyviä tuloksia, joita van der Meij ym. (2003) eivät sisällyttäneet artikkeliinsa. Hyödynnän siis työssäni myös vuoden 2004 artikkelia.

Nelikantamalli jakaa vaiheittaiset ohjeet neljään kategoriaan: tavoitteisiin, edellytyksiin, toimintoihin ja reaktioihin sekä ei-toivottuihin tilanteisiin. Tavoitteet kuvailevat, mihin järjestelmätilaan käyttäjä haluaa päätyä, ja ne voidaan ilmoittaa lyhyesti jo otsikossa, kun taas edellytykset ovat nimensä mukaisesti tehtävän aloittamisen tai siinä onnistumisen edellytyksiä. Toiminnot ja reaktiot kuvailevat käyttäjän toimintoja ja sitä, kuinka järjestelmä reagoi näihin toimintoihin. Ei-toivotut tilanteet ovat ongelmia ja vaaratilanteita, joiden välttämiseksi ja korjaamisessa käyttäjää täytyy ohjeistaa.

3.1 Tavoitteet

Tavoitteet ja mahdolliset osatavoitteet kuvataan usein jo vaiheittaisen ohjeen otsikossa (van der Meij ym. 2003, 136), kuten *Näytön lukitseminen* (Huawei 2017, 6). Ne ovat lopullisia tiloja ja niihin johtavia välitiloja, jotka käyttäjä haluaa saavuttaa (van der Meij ym. 2003, 133), mikä heijastelee Newellin ajatusta, jossa ongelmanratkaisu on oikean polun etsimistä ja erilaiset tilat ovat polun etappeja (ks. luku 2.1.1) Toisin sanoen tavoitteet ovat Farkasin (1999) ja Newellin (1980) kuvailemia päämäärä- ja toivottuja tiloja. Tavoitteiden kuvailun lisäksi otsikot antavat lukijalle kokonaiskuvan siitä, millaista järjestelmää hän on ryhtymässä käyttämään ja mitä hän voi tehdä sen avulla (van der Meij ym. 2003, 136). Lisäksi kuvaavat otsikot helpottavat käyttäjän kaipaaman sisällön löytymistä sisällysluettelosta (mts. 137), mikä osaltaan lisää käyttöohjeen selkeyttä ja ottaa siten huomioon käyttäjän odotukset (ks. luku 2.2).

Jotta käyttäjän olisi helppo hahmottaa tavoitteiden vaatimat toimenpiteet, vaiheittaisen ohjeen tulisi muodostua korkeintaan viidestä vaiheesta. Jos tavoite vaatii useita vaiheita, se on hyvä jakaa osatavoitteisiin. (van der Meij & Gellevij 2004, 6.) Tämä on erityisen tärkeää, jos käyttäjän on opetettava vaiheet ulkoa, sillä työmuisti pystyy säilömään kerrallaan 7 ± 2 elementtiä (van der Meij ym. 2003, 134). Osatavoitteet ovat tärkeitä myös siksi, että ne sitovat tehtävät tiettyyn kontekstiin ja tekevät niistä siten tarkoituksenmukaisia, mikä puolestaan motivoi käyttäjiä suorittamaan tehtävän (mts. 136).

Kuten totesin luvussa 2.2, käyttäjä saattaa tarvita suostuttelua. Otsikkoa saattaa siten seurata lyhyt konseptuaalinen elementti, joka esittelee tavoitteen ja myös perustelee, miksi käyttäjän pitäisi yrittää saavuttaa se. Jos tavoitteeseen on useita eri reittejä, kirjoittaja saattaa valita niistä vain yhden tai esitellä niistä osan tai kaikki. Tällöin käyttäjälle täytyy antaa eri keinoista tarpeeksi tietoa, jotta hän voi valita itselleen sopivimman. (van der Meij ym. 2003, 133–134.) Kuten mainitsin luvussa 2.2.2, lukijaa ei kuitenkaan pidä kuormittaa liian monilla päätöksillä (Farkas 1999, 46). Useiden vaihtoehtojen esittäminen voi lisäksi tehdä käyttöohjeiden ulkoasusta sekavan, ja valinnan tekeminen viivyyttää itse tehtävän aloittamista. Viimeksi mainittu on erityisen haitallista, sillä käyttöohjeita luetaan nimenomaan tehtävien suorittamista varten. Jos käyttäjä kokee ohjeiden lukemisen vievän liikaa aikaa, hän saattaa jättää lukemisen kokonaan väliin. Tällöin kirjoittajan on kuitenkin valittava, mikä vaihtoehtoisista reiteistä on suoraviivaisin ja sopivin kohdeyleisölle, missä hän voi käyttää apunaan tehtäväänalyysia.

Tehtäväänalyysi auttaa käyttöohjeiden kirjoittajia hahmottamaan, millaisia tehtäviä käyttäjä haluaa suorittaa ja miten hän pääsee tavoitteeseensa. Tehtäväänalyysin pohjana on mahdollista käyttää esimerkiksi GOMS-mallia, jonka mukaan käyttäjät yrittävät suorittaa tehtäviä toimimalla rationaalisesti tai loogisesti, mutta kognitiiviset rajoitukset saattavat haitata suoritusta (Card, Moran & Newell 1983, 27; ks. myös luku 2.1.1 ja polun rajoitteet). GOMS-malli voidaan esittää lyhyesti seuraavana kaavana: ”Goals + Task + Operators + Inputs + Knowledge + Processing Limits \approx Behavior” (Card ym. 1983, 27). Koska kolme viimeistä elementtiä ovat tilannekohtaisia ja kolme ensimmäistä elementtiä ovat yleisluontoisempia, malli voidaan myös tiivistää seuraavaan muotoon: ”Goals + Task + Operators \approx Behavior” (van der Meij ym. 2003, 134).

Tavoitteet (*goals*) kuvaavat käyttäjän aikomuksia sekä aiemmin käsittelemiäni päämäärä- tai toivottuja tiloja ja välitiloja (Newell 1980; Farkas 1999). Tehtävä (*task*) sisältää menettelytavan

(*method*), jota käyttäjä hyödyntää pyrkiessään tavoitteeseensa eli toisin sanoen sarjan toimintoja, jotka muodostavat vaiheittaisen ohjeen vaiheet. Oikean menettelytavan valinnassa auttavasta tiedosta käytetään tässä kontekstissa termiä *selection rules*. Operaattorit (*operators*) ovat käyttäjän suorittamia, havaittavissa olevia toimia, kuten näytön koskettaminen tai hiiren painikkeen napsauttaminen (ks. Newell 1980). Nämä elementit (*goals, operators, methods, selection rules*) muodostavat yhdessä GOMS-mallin. (van der Meij ym. 2003, 134.) GOMS on erityisen hyödyllinen, koska se paljastaa jo varhaisessa vaiheessa, millaiseksi käyttöohjeiden kirjoittaja olettaa lukijoidensa taitotason. Jos tehtävän kuvailu mainitsee esimerkiksi tietyn operaattorin, kuten ”kuvan lisääminen”, kirjoittajan on mahdollisesti palattava analyysissään askel taaksepäin ja kuvailtava muun muassa se, mitä kohtia ja painikkeita käyttäjän on valittava valikosta, jotta hän voi lisätä kuvan. (van der Meij ym. 2003, 135.) Jos kirjoittaja taas päättää kohdistaa ohjeet edistyneemmälle yleisölle, hän voi todeta, ettei perustoimintojen esittely ole enää siinä vaiheessa tarpeellista. Tämä liittyy edellä mainittuun huomioon siitä, että lukijoiden täytyy kokea käyttöohjeet itselleen sopiviksi ja että kirjoittajan täytyy sen vuoksi mukauttaa tekstinsä lukijoiden tarpeisiin (Farkas 1999, 43). Palaan GOMS-malliin aineiston analyysin yhteydessä luvussa 5.

3.2 Edellytykset


Edellytykset (*prerequisites*) ovat ehtoja, jotka käyttäjän on täytettävä, jotta hän onnistuu tehtävässä. Esimerkiksi ohjelmien käyttö edellyttää, että käyttäjän koneella on kyseistä ohjelmaa tukeva käyttöjärjestelmä ja että käyttäjä on oikeassa näkymässä. Käyttöohjeiden esittelyluku saattaa kuvailla edellytyksiä yleisellä tasolla, kun taas yksittäisten tehtävien edellytykset puolestaan kuvaillaan tarkemmin (van der Meij & Gellevij 2004, 6). Van der Meij ym. (2003, 141) jakavat edellytykset kolmeen luokkaan: järjestelmän tila sekä modulaarisuus, käyttäjän taidot sekä häivytykset ja käyttäjän tiedot sekä linkitys (*mapping*). *Mapping* on haasteellinen termi kääntää, koska sille on kontekstista riippuen useita mahdollisia suomennoksia. Käännän sen tässä tutkielmassa muotoon *linkitys*.

3.2.1 Järjestelmätilat ja modulaarisuus

Jotta käyttäjä voi aloittaa tehtävän, hänen on tiedettävä, mikä järjestelmän senhetkinen tila on (van der Meij & Gellevij 2004, 7) eli toisin sanoen se, mitä järjestelmässä on jo tapahtunut. Käyttöohjeet voivat ilmoittaa tämän sanallisesti tai kuvakaappauksilla. Kuvan 2 esimerkissä tehtävän suorittaminen edellyttää, että käyttäjä on yhdistänyt puhelimensa internettiin (Huawei 2017, 93):

Haitallisten URL-osoitteiden tunnistaminen tekstiviesteistä

Aseta puhelimesi automaattisesti tunnistamaan haitalliset URL-osoitteet henkilökohtaisten ja taloudellisten tietojesi suojaamiseksi.

 Varmista ennen tämän ominaisuuden käyttöönottoa, että puhelimella on Internet-yhteys.

Kuva 1. Esimerkki edellytyksestä

Järjestelmän tiloihin liittyy myös modulaarisuus, joka tarkoittaa, että käyttöohjeen luvut ovat itsenäisiä kokonaisuuksia, jotka viittaavat toisiinsa mahdollisimman vähän. Tämä edellyttää, että kirjoittaja valitsee pisteen, josta kaikki luvut alkavat ja johon ne päättyvät. (van der Meij ym. 2003, 143.) Newellin (1980) termein ilmaistuna luvut alkavat aina samasta alkutilasta, joka on samalla luvun päämäärätila. Tällöin lukijat aloittavat samasta tilanteesta riippumatta siitä, ovatko he lukeneet edellisen luvun vai aloittavatko lukemisen satunnaisesta luvusta (van der Meij ym. 2003, 144). Lisäksi modulaarisuus tarkoittaa, että järjestelmän käsittelemien tuotteiden ei tulisi olla riippuvaisia toisistaan (van der Meij ym. 2003, 144). Van der Meij ym. (2003, 144) antavat jokseenkin vanhentuneen esimerkin kokeilutiedostoista, joita käyttäjä voi muokata ja testata ja joista pitäisi lisäksi olla tarjolla muokkaamattomat alkuperäisversiot jokaista lukua varten, jotta käyttäjä voi aloittaa luvut aina samoista lähtökohdista. Nykyaikaisempi esimerkki voisi olla ohjelmisto, joka antaa käyttäjälle mahdollisuuden tallentaa kustomoituja asetuksia, mutta sallii myös oletusasetusten palauttamisen.

3.2.2 Käyttäjän taidot ja häivytytys

Tiedostojen avaamisen ja tallentamisen kaltaisten perustoimintojen esittely on tarpeen käyttöohjeen alussa, mutta niiden toistaminen läpi koko ohjeiden pidentää vaiheittaisia ohjeita ja voi turhauttaa käyttäjiä. Yksi ratkaisuvaihtoehto on viittaaminen alkuperäisiin, yksityiskohtaisiin ohjeisiin (van der Meij ym. 2003, 144). Ongelma voidaan ratkaista myös häivyttämällä, mikä tarkoittaa, että perustoimintojen kuvaukset ovat sitä ylimalkaisempia, mitä pitemmälle käyttäjä etenee ohjeissa. Van der Meij ja Gellevij (2004, 7) antavat siitä seuraavanlaisen esimerkin:

Taulukko 3. Esimerkki häivyttämisestä

1. versio	1. Valitse valikosta Tiedosto . 2. Valitse Tallenna .
-----------	--

	3. Kirjoita tiedoston nimi. 4. Valitse Tallenna -kuvake.
2. versio	1. Siirry Tiedosto-valikkoon ja valitse Tallenna . 2. Kirjoita tiedoston nimi ja valitse Tallenna -kuvake.
3. versio	1. Tallenna tiedosto

Kuten taulukko 4 osoittaa, vaiheet eivät ainoastaan lyhene, vaan osa niistä on yhdistetty toisiinsa. Koska häivyttäminen vaatii käyttöohjeiden lukijalta jonkin verran ajatustyötä, se auttaa häntä sisäistämään ja oppimaan ohjeiden sisältämän tiedon (van der Meij ym. 2003, 145). Van der Meij ym. perustelevat tätä väitettä Leutnerin (2000, 352) tutkimuksella, jossa selvisi, että testihenkilöt oppivat käyttämään ohjelmaa parhaiten, kun ohjeet olivat aluksi yksityiskohtaisia ja häivytyks tehtiin vähitellen. Aihetta tarkemmin tutkimatta en voi sanoa häivyttämisen yleisyydestä mitään varmaa, mutta se vaikuttaa haasteelliselta toteuttaa, sillä on vaikea ennakoida, missä järjestyksessä lukijat tutustuvat ohjeisiin.

3.2.3 Käyttäjän tiedot ja linkitys

Smith ja Ragan¹ (van der Meij'n & Gellevij'n 2004, 7 mukaan) toteavat, että jo olemassa oleva tieto on tärkein oppimiseen vaikuttava tekijä, sillä se auttaa ihmisiä tulkitsemaan uusia kokemuksia. Käyttöohjeiden täytyy siten löytää tasapaino sen välillä, kuinka käyttäjä ymmärtää tehtävän kokemustensa perusteella ja miten järjestelmä vaatii käyttäjää ymmärtämään sen (van der Meij ym. 2003, 146). Jos käyttöohjeen lukijalla ei ole ennestään tarvittavaa tietoa, hänelle voidaan tarjota se muun muassa ennakkojäsentäjien (*advance organizers*) avulla, jotka linkittävät järjestelmän käyttäjän kokemusmaailmaan esimerkiksi vertausten ja metaforien avulla (van der Meij ym. 2003, 146). Ennakkojäsentäjien kaksi pääkategoriaa ovat vertailevat ja esittelevät jäsentäjät. Vertailevat jäsentäjät selittävät ohjelmistojen elementtejä vertaamalla niitä fyysisen maailman ilmiöihin, kun taas esittelevät jäsentäjät sisältävät lisätietoa konsepteista, joihin järjestelmä pohjautuu tai kertovat mitä tietyn vaihtoehdon valitsemisesta seuraa. (van der Meij ym. 2003, 146–147.) Nelikantamallin ikä näkyy kuitenkin tässäkin tapauksessa. On vaikeaa kuvitella, että moni nykyajan käyttäjä kaipaisi esimerkiksi selattavan tiedoston vertaamista kirjaan.

¹ Smith, Patricia L. & Tillman J. Ragan 1999. *Instructional Design*. 2. painos. New York: John Wiley & Sons, Inc.

Kun käyttäjä saa tehtävästä lisää tietoa ennakkojäsentäjien kautta tai muilla tavoin, hän ymmärtää sen paremmin ja sen suorittamisesta tulee hänelle miellyttävämpää (van der Meij & Gellevij 2004, 7). Lisätiedon antaminen on toisin sanoen myös suostuttelukeino (ks. myös luku 2.2), jonka avulla käyttäjä saadaan seuraamaan ohjeita. Tämä tukee Farkasin (1999, 43) edellä mainittua näkemystä, jonka mukaan vaiheittaisen tiedon välittäminen on retorista toimintaa.

3.3 Toiminnot ja reaktiot

Käyttäjä suorittaa toimintoja päästääkseen tavoitteiseensa, ja tuote reagoi kyseisiin toimintoihin (van der Meij ym. 2003, 160). Toimintojen ja reaktioiden kyseiset määritelmät ovat verrattavissa käyttäjän toimintoihin sekä järjestelmän toimintoihin (Farkas 1999). Ne ovat esillä seuraavassa esimerkissä:

4. Lopeta tallennus koskettamalla [kuvaketta]. Kuvattu video toistetaan nopeutettuna. (Huawei 2017, 2).

Ensin käyttäjä toimii eli lopettaa tallentamisen, ja puhelin reagoi toistamalla juuri kuvatun videon. Van der Meij ym. (2003, 167) sekä van der Meij ja Gellevij (2004, 7) tarkastelevat toimintaa kolmesta näkökulmasta: käyttäjän itsenäisyyden ja neuvomisen tasapainottaminen, kuvakaappausten käyttö toiminnan tukena ja tehtävän vaiheiden kuvailu. Esittelen nämä näkökulmat tarkemmin seuraavissa alaluvuissa.

3.3.1 Itsenäisyys ja neuvominen (*Let Go and Support*)

Tiettyjen toimintojen suorittamisen lisäksi käyttäjää on kehotettava tutkimaan tuotetta omillaan, jotta hän ei oppisi ainoastaan mitä tehdä vaan myös miksi se on tehtävä (van der Meij & Gellevij 2004, 7). Näin käyttäjä tutustuu tuotteen toimintaperiaatteisiin syvällisemmin ja joutuu tukeutumaan käyttöohjeisiin tulevaisuudessa vähemmän. Itsenäisyyden ja ohjeistamisen tasapainottaminen on kuitenkin vaativa tehtävä käyttöohjeiden kirjoittajalle, varsinkin jos hän ei tunne yleisöään hyvin. Lukijan tarpeiden ennustaminen vaikeutuu myös silloin kun yleisö on laaja ja sisältää monen tasoisia käyttäjiä. Aloittelija saattaa tarvita yksityiskohtaisia ohjeita ja vähemmän kehotuksia itsenäiseen työskentelyyn, kun taas kokenut käyttäjä voi kokea ohjaamisen ajantuhlauksena ja liiallisena holhouksena.

3.3.2 Kuvakaappaukset ja järjestelmän reaktiot

Koska palaute on käyttäjälle tärkeää, vaiheittaiset ohjeet siirtyvät usein tiheään tahtiin käyttäjän toimintojen kuvailusta järjestelmän reaktioihin ja päinvastoin (van der Meij ym. 2003, 162). Kuvakaappaukset ovat erityisen hyödyllisiä näiden reaktioiden havainnollistamisessa, ja niistä on käyttöohjeiden lukijalle muutakin apua. Ne muun muassa tukevat tekstuaalista informaatiota ja auttavat mentaalimallin muodostamisessa. (mp.) Kilkin (2013, 50) mukaan mentaalimalli on ”mielen tuottama kuvaus ympäröivästä maailmasta, sen eri osien välisistä suhteista sekä henkilön näkemyksestä hänen omista toimistaan ja niiden vaikutuksista ympäristöön”. Jotta kuvakaappaukset tukisivat käyttäjän mentaalimallin syntymistä, niiden täytyy esittää käyttöliittymän elementit oikeassa kontekstissa, mitä van der Meij ym. (2003, 163) kutsuvat järjestelmätopologiaksi. Lisäksi kuvan tärkeimmät elementit täytyy nimetä, ja muutokset järjestelmän tilassa täytyy kuvailla sanoin tai kuvin. Kuvakaappaukset ovat myös merkki siitä, että käyttöohjeeseen keskittymisen sijaan käyttäjän on aika siirtää huomionsa itse ohjelmaan. (van der Meij & Gellevij 2004, 8.)

Kuvakaappausten avulla käyttäjä voi myös varmistaa, että hän on toiminut oikein ja ohjelma on siinä tilassa missä pitääkin. Tämä motivoi etenkin aloittelevia käyttäjiä, jotka saattavat tarvita enemmän rohkaisua ja positiivista palautetta kuin kokeneet käyttäjät. Lisäksi käyttäjä voi tarkistaa suoraan, onko ohjelma oikeassa tilassa tai minkä näköistä kuvaketta, symbolia tai muuta vastaavaa hänen tulee napsauttaa ja missä kohtaa ruutua se sijaitsee. Näin hänen ei tarvitse kääntää mielessään tekstuaalista tietoa kuvan muotoon, mikä lisäisi kognitiivista rasitusta. (van der Meij & Gellevij 2004, 8.)

3.3.3 Vaiheet

Koska ohjeiden vaiheet ovat kenties tärkein osa käyttöohjeita, niiden on erotuttava muusta tekstistä. Yleisin tapa on esittää ne numeroituna listana, ja tätä konventiota tukevat myös tutkimustulokset (van der Meij & Gellevij 2004, 8) ja standardien suositukset (SFS-EN 82079-1 2012, 52). Van der Meij ym. (2003, 167) perustelevat sitä seuraavin argumentein:

- Numerointi saa tehtäväkeskeisen tiedon erottumaan muusta tekstistä.
- Lukija hahmottaa tehtävän vaiheet helposti.
- Lukijan on helpompi palata samaan vaiheeseen myöhemmin. Todennäköisyys jonkin vaiheen ohittamiseen tai toistoon pienenee.

- Kirjoittaja huomaa nopeasti, onko ohjeesta tulossa liian pitkä.

Yksinkertaisimmillaan nämä vaiheet koostuvat verbeistä ja objekteista. Verbi kertoo, mitä käyttäjän täytyy tehdä, kun taas objekti kertoo, mihin teko kohdistuu. Koska verbi-objekti-pari on työvaiheen tärkein osio, käyttöohjeiden kirjoittajan tulisi korostaa sitä, jotta se erottuu vaiheen muista osista. (van der Meij & Gellevij 2004, 8.) Van der Meij ja Gellevij eivät anna korostamisesta esimerkkejä, mutta yleinen käytäntö on muun muassa se, että käyttöliittymää kuvailevat objektit lihavoidaan ("Valitse **Tallenna**"). Elementit voidaan esittää sanallisen kuvauksen lisäksi tai sijaan myös kuvallisesti. Oikea tapa riippuu kontekstista, ja pääasia on, että elementin visuaalinen ja/tai kuvallinen representaatio käyttöohjeessa vastaa sitä, millaiselta se näyttää järjestelmässä. Tällöin käyttäjän ei tarvitse mielessään kääntää tekstiä esimerkiksi painikkeen kuvaksi, mikä vähentää hänen kognitiivista kuormittumistaan. (van der Meij ym. 2003, 169.) Käyttäjä saattaa muutoinkin joutua ponnistelemaan käyttöohjeita lukiessaan, varsinkin jos tehtävät ovat pitkiä ja monimutkaisia, joten kaikenlaisesta kuormituksen vähentämisestä on hänelle hyötyä.

Koska käyttäjän on tiedettävä toiminnon kohteen sijainti ennen kuin hän voi suorittaa haluamansa toiminnon, sijainti tulisi ilmoittaa ennen toimintoa (van der Meij & Gellevij 2004, 8). Van der Meij ja Gellevij (2004, 8) antavat tästä seuraavanlaisen esimerkin: *In the File manager, select Up*. Tämäntapaiset täsmennykset ovat johdattelevia määritteitä (*facilitating modifiers*), jotka voivat sijainnin lisäksi kertoa esimerkiksi sen, millä tavalla toiminto täytyy suorittaa (van der Meij ym. 2003, 169; Farkas 1999, 48). Toisin kuin suomennokseni kenties antaa ymmärtää, kaikkia määritteitä ei kuitenkaan välttämättä ole sijoitettu ennen varsinaista toimintoa.

3.4 Ei-toivotut tilat

Ei-toivotut tilat (*unwanted states*) ovat ongelmatilanteita, joita käyttäjää neuvotaan välttämään ja ratkaisemaan. Tämä tapahtuu useimmiten varoitusten avulla. Van der Meij ja Gellevij (2004) käsittelevät lisäksi ongelmanratkaisua, joka liittyy tilanteisiin, joissa käyttäjä tai järjestelmä eivät ole vaarassa, mutta järjestelmä on kuitenkin päätenyt tilaan, joka hidastaa tai estää käyttäjää suorittamasta tehtävää. Kerron näiden tilojen välttämistä tarkemmin seuraavissa alaluvuissa.

3.4.1 Varoitukset

Jos järjestelmän valmistaja havaitsee mahdollisen vaaratilanteen, hänellä on kolme vaihtoehtoa: järjestelmä voidaan suunnitella niin, ettei tilanne ole enää mahdollinen, tilanne ja käyttäjä voidaan yrittää pitää erillään toisistaan, tai tilanteen mahdollisuudesta voidaan varoittaa (van der Meij ym. 2003, 151). Varoitusten täytyy ilmoittaa mahdolliset vaaratekijät sekä niiden huomiotta jättämisen seuraukset (SFS-EN 82079-1 2012, 40). Jotta käyttäjä voi välttää nämä vaaratekijät, varoituksen on kerrottava, mitä hänen täytyy tehdä tai olla tekemättä (van der Meij & Gellevij 2004, 10). Van der Meij ja Gellevij (2004, 9) sekä van der Meij ym. (2003, 152) käyttävät varoitusten suunnittelusuositustensa pohjana *see-think-use*-mallia, jossa käyttäjän täytyy ensin nähdä (*see*) varoitus, ymmärtää se (*think*) ja toimia sen mukaan (*use*).

Jotta käyttäjä voi nähdä varoituksen, sen täytyy erottua muusta tekstistä esimerkiksi värin tai koon avulla. Ei myöskään riitä, että käyttäjä näkee sen, vaan sen on kiinnitettävä hänen huomionsa siksi aikaa, että hän ehtii lukea tekstin ja/tai katsoa kuvaa. Tämä vaatii, että varoitus on helposti luettavissa (*legible, readable*). (van der Meij ym. 2003, 153.) Mitä varoitusten sijaintiin tulee, teoria ja käytäntö eroavat tässä aiheessa toisistaan: tutkimusten mukaan varoitukset ovat tehokkaimpia vaiheittaisten ohjeiden yhteydessä varsinkin, jos tehtävä on käyttäjälle uusi, mutta käyttöohjeita julkaisevien yritysten tapana saattaa olla julkaista ne erillisessä osiossa. (van der Meij & Gellevij 2004, 9.) Myös edellä mainitun SFS-EN-standardin (SFS-EN 82079-1 2012, 40) mukaan varoitukset tulisi kuitenkin ilmoittaa siinä yhteydessä, joka saattaa aiheuttaa vaaratilanteen. Van der Meij ym. (2003, 155) ovat samaa mieltä, sillä käyttäjä hyödyntää todennäköisimmin tietoa, josta on hänelle välitöntä hyötyä.

Varoituksen näkemisen lisäksi käyttäjän täytyy myös ymmärtää se, mikä edellyttää, että se on selkeä ja mieleenpainuva. Pelkkä teksti tai pelkkä kuva ei välttämättä riitä, sillä käyttäjä ei ehkä tunne käyttöohjeiden ulkoasua määritteleviä standardeja, kuten symboleja tai sanojen *caution*, *warning* ja *danger*² eroa. Jos teksti on monitulkintainen, sitä tukeva kuva auttaa vähentämään väärinkäsitysten riskiä ja päinvastoin. (van der Meij ja Gellevij 2004, 10.) Van der Meij ym. (2003, 154) kuitenkin lainaavat vuonna 1988 julkaistua tutkimusta, jossa 88 prosenttia koehenkilöistä havaitsi varoituksen, 46 prosenttia luki sen ja vain 27 prosenttia noudatti sitä. Varoitusten on siis oltava suostuttelevia ja vaikutettava uskottavilta, mikä liittyy aiemmin käsiteltyyn klassiseen

² Suomen kielen vastaavat termit ovat *huomio*, *varoitus* ja *vaara* (SFS-EN 82079-1 2012, 12).

retoriikkaan. Lisäksi käyttäjä on motivoituneempi noudattamaan varoituksia, jos haitan välttäminen ei vaadi suurta vaivaa ja jos vaaratilanne on vakava. (van der Meij & Gellevij 2004, 10.) Vakavat vaaratilanteet ovat aiheellinen huoli lähinnä laitteistojen kohdalla, sillä ohjelmistoja käyttäessä tapahtuneet virheet eivät johda fyysisiin vammoihin.

3.4.2 Ongelmanratkaisu

Yksikään tuote ei ole virheetön tai niin intuitiivinen, etteikö käyttäjä kohtaisi ennemmin tai myöhemmin ongelmia. Tilanteet, joissa niitä esiintyy eniten tai joissa niiden korjaaminen on vaikeaa johtuvat usein käyttäjän puutteellisista tiedoista tai siitä, että tuotteen toiminta on ristiriidassa hänen mentaalimallinsa kanssa (van der Meij & Gellevij 2004, 11). Ne eivät ainoastaan vaikeuta tehtävässä etenemistä, vaan ne vaikuttavat käyttäjän motivaatioon sekä mielikuvaan niin tuotteesta kuin sen valmistajastakin (mp.). Tehokkaiden ja suoraviivaisten ongelmanratkaisuehdotusten tarjoaminen on siis paitsi käytettävyyttä, myös imagokysymys. Lisäksi niiden avulla käyttäjä oppii virheistään ja tulee taitavammaksi tuotteen käytössä (van der Meij & Carroll 1995, 252). Tällöin hänen täytyy tukeutua vähemmän käyttöohjeisiin. Eräs vaihtoehto ongelmanratkaisuprosessin kuvailuun on malli, jossa käyttäjä ensin havaitsee (*detect*) ongelman. Seuraavaksi hän arvioi (*diagnose*) ja lopulta korjaa (*correct*) sen. (van der Meij & Gellevij 2004, 11.)

Ensinnäkin käyttäjän on saatava tietää ongelman olemassaolo, mikä voi saada alkunsa sisäisesti tai ulkoisesti. Ensiksi mainitun kohdalla käyttäjä pohtii, onko hän toiminut oikein, ja jälkimmäisessä tuote ilmoittaa ongelmasta. (van der Meij & Carroll 1995, 253.) Ohjelmistoissa tämä tarkoittaa yleensä virheilmoitusta. Jotta ongelma ei ehtisi pahentua ja sen korjaaminen vaikeutua, käyttäjälle tulee ilmoittaa siitä mahdollisimman varhaisessa vaiheessa (van der Meij & Gellevij 2004, 12) muttei kuitenkaan liian aikaisin. Jos mahdollisesta ongelmasta ilmoitetaan liian varhain, pidän vaarana, että käyttäjä saattaa unohtaa sen tai tulla liiankin varovaiseksi. Varoitusten tavoin myös ongelmanratkaisua koskeva ohjeistus tulisi siis sijoittaa käyttöohjeissa siihen kontekstiin, missä sitä todennäköisimmin tarvitaan (van der Meij & Carroll 1995, 253; van der Meij & Gellevij 2004, 11). Tällöin käyttöohjeiden kirjoittaja voi antaa täsmällisiä, tiettyyn ongelmaan keskittyviä neuvoja eikä vain yleisluontoisia ohjeita. (van der Meij & Gellevij 2004, 11–12.)

Ongelman tiedostamisen jälkeen käyttäjän on arvioitava se ja päätettävä mitä tehdä sille. Hän voi arvioida sitä tarkastelemalla järjestelmän senhetkistä tilaa tai ongelmaa edeltäneitä toimia. Jotta

käyttäjä voi ratkaista sen, siinä auttava tieto on esitettävä tarpeeksi yksinkertaisessa ja ymmärrettävässä muodossa. Ohjeistus on myös sopeutettava siihen, minkä tyyppisen ongelman käyttäjä on kohdannut. Jos ongelma on selkeästi määriteltävissä, ohjeiden seuraaminen on helpompaa kuin vaikeasti määriteltävien ongelmien kohdalla, joten jälkimmäisessä tapauksessa tarvitaan yksityiskohtaisempia ohjeita. (van der Meij & Gellevij 2004, 12.)

Van der Meij ja Carroll (1995, 253) huomauttavat, että vain havaitseminen ja ratkaiseminen ovat pakollisia vaiheita, sillä arviointi ei ole yhtä kriittinen ongelman selvittämisen kannalta. Toisaalta suoraan havaitsemisesta ratkaisuun siirtyminen ei välttämättä opeta käyttäjälle tuotteen toimintaperiaatteita siinä määrin kuin ongelman syvällisempi arviointi. Oppimisen kannalta kaikkia kolmea voidaan siis pitää yhtä tärkeinä vaiheina. Lisäksi käyttöohjeiden kirjoittaja voi sisällyttää ongelman arvioinnissa auttaviin osioihin konseptuaalista tietoa, joka saattaa kiinnostaa käyttäjää (van der Meij & Gellevij 2004, 12). Kirjoittajan on kuitenkin muistettava pitää ohjeet käyttäjä- eikä järjestelmälähtöisenä, sillä pääsääntöisesti käyttäjät ovat kiinnostuneita vain ohjelman käytöstä eivätkä lähdekoodin tapaisten taustatekijöiden syvällisestä ymmärtämisestä. Samoin kuin varoituksiin, käyttäjän huomio voidaan kiinnittää myös ongelmanratkaisua koskeviin kohtiin esimerkiksi kuvien ja symbolien avulla (van der Meij & Gellevij 2004, 12).

3.5 Nelikantamallin yhteenveto

Olen esitellyt tässä luvussa nelikantamallin (van der Meij ym. 2003), joka pohjautuu Farkasin (1999) yksinkertaistettuihin vaiheisiin. Malli jakaa vaiheittaisten ohjeiden sisällön tavoitteisiin, edellytyksiin, toimintoihin ja reaktioihin sekä ei-toivottuihin tiloihin.

Tavoitteet ja mahdolliset osatavoitteet kuvailevat, mihin järjestelmätiloihin käyttäjä haluaa päätyä tehtävän suorittamalla. Ne kuvataan usein vaiheittaisen ohjeen otsikossa, mutta niitä saatetaan kuvailla tarkemmin konseptuaalisessa elementissä (ks. luku 2.3). Osatavoitteet eli alaotsikot ovat tarpeen, jos tehtävä on erityisen pitkä, sillä yksittäisessä tehtävässä ei pitäisi olla liikaa vaiheita (van der Meij & Gellevij 2004, 6). Jos samaan tavoitteeseen johtaa useampi polku (ks. luku 2.1.1) ja kirjoittaja esittelee niistä muutamat tai ne kaikki, käyttäjälle on neuvottava, millä perusteella hän voi valita itselleen sopivamman tavan tehtävän suorittamiseen (van der Meij ym. 2003, 133–134). Käyttöohjeiden kirjoittajat voivat käyttää tehtävien kulun hahmottelussa ja käyttäjän toiminnan ennakoimisessa apunaan muun muassa GOMS-analyysia, joka muodostuu käyttäjän tavoitteista,

menettelytavasta, jolla hän suoriutuu tehtävästä, perusteluista, joiden perusteella hän valitsee itselleen sopivimman menettelytavan sekä käyttäjän suorittamista toiminnoista eli operaattoreista (van der Meij ym. 2003, 134).

Edellytykset (*prerequisites*) ovat ehtoja, joita tehtävässä onnistuminen vaatii. Van der Meij ym. (2003, 141) käsittelevät niitä kolmesta näkökulmasta: järjestelmän tila sekä modulaarisuus, käyttäjän taidot sekä häivytytys ja käyttäjän tiedot sekä linkitys. Tehtävän aloittaminen vaatii, että käyttäjä tietää, mikä järjestelmän senhetkinen tila on (van der Meij & Gellevij 2004, 7). Tiloihin liittyy modulaarisuus, jota noudatettaessa käyttöohjeen luvut alkavat aina samasta alkutilasta ja päättyvät samaan päämäärätilaan (van der Meij ym. 2003, 143; Newell 1980). Mitä käyttäjän taitoihin tulee, eräs keino niiden kehityksen tukemiseen on häivytytys, joka tarkoittaa, että ohjeista karsitaan yksityiskohtia käyttöohjeen loppua kohden (van der Meij ym. 2003, 145). Käyttäjän tiedonsaantia puolestaan tukee se, että hänen kokemusmaailmansa ja järjestelmän vaatimusten välinen ero kurotaan umpeen esimerkiksi ennakkojäsentäjillä (van der Meij ym. 2003, 146). Vertailevat ennakkojäsentäjät vertaavat järjestelmää fyysiseen maailmaan, ja esittelevät jäsentäjät antavat lisätietoa järjestelmän toimintaperiaatteista ja siitä, mitä käyttäjä voi saavuttaa eri vaihtoehtojen avulla (van der Meij ym. 2003, 149.)

Toiminnot ovat käyttäjän suorittamia toimintoja, ja tuotteen reaktiot ovat seurausta näistä toiminnoista (van der Meij ym. 2003, 160). Van der Meij ym. (2003, 167) sekä van der Meij ja Gellevij (2004, 7) tarkastelevat toimintoja ja reaktioita käyttäjän itsenäisyyden ja neuvomisen tasapainottamisen, kuvakaappausten sekä tehtävän sisältämien vaiheiden näkökulmasta. Käyttäjä oppii tuntemaan järjestelmän syvällisemmin, jos häntä ei ainoastaan suoraan neuvota tekemään tiettyjä toimintoja, vaan häntä myös rohkaistaan tutkimaan sitä itsenäisesti (van der Meij & Gellevij 2004, 7). Kuvakaappaukset puolestaan havainnollistavat esimerkiksi ohjelmiston reaktioita käyttäjän toimiin ja tukevat hänen mentaalimallinsa muodostumista (van der Meij ym. 2003, 162). Ohjeiden vaiheet taas ovat yksi käyttöohjeiden tärkeimmistä osista, joten ne pyritään erottamaan muusta tekstistä yleensä numeroimalla ne (van der Meij & Gellevij 2004, 8). Vaiheet saattavat koostua pelkästä verbi-objekti-parista (mp.), mutta ne voivat lisäksi sisältää lisätietoa, kuten johdattelevia määreitä, jotka täsmentävät muuan muassa sitä, missä ja miten jokin toiminta täytyy suorittaa (mts. 169).

Käyttäjää ohjeistetaan välttämään ei-toivottuja tiloja varoitusten avulla. Jotta käyttäjä ylipäättään huomaisi varoitukset, ne pitäisi sijoittaa niiden vaiheiden yhteyteen, jossa käyttäjä tarvitsee niitä (SFS-EN 82079-1 2012, 40), ja niiden täytyy erottua muusta sisällöstä ja olla muutoinkin nopeasti hahmotettavissa (van der Meij & Gellevij 2004, 10). Varoitusten on kuvailtava paitsi käyttäjää uhkaavat ei-toivotut tilat, myös varoituksen huomiotta jättämisen seuraukset (SFS-EN 82079-1 2012, 40). Käyttäjälle on lisäksi kerrottava, mitä hänen on tehtävä välttääkseen vaaratilanteen tai millaiset toiminnot johtavat siihen (van der Meij & Gellevij 2004, 10). Kaikki ei-toivotut tilat eivät kuitenkaan ole vaarallisia, ja lievemmat ongelmat ovat usein käyttäjän korjattavissa. Jotta korjaaminen onnistuisi, ongelmanratkaisua koskevan tiedon tulisi varoitusten tapaan olla siinä yhteydessä, missä käyttäjät sitä todennäköisimmin tarvitsee (van der Meij & Carroll 1995, 253; van der Meij & Gellevij 2004, 11), ja sen on oltava riittävän helposti omaksuttavissa (van der Meij & Gellevij 2004, 12).

4 Aineisto ja menetelmä

Käsittelen tässä luvussa tutkimusaineistoani ja -menetelmäni. Ensimmäisessä alaluvussa täsmennän, millaista aineistoa olen tutkinut ja kuinka olen rajannut sen, ja toisessa alaluvussa kerron, mitä menetelmää olen hyödyntänyt aineiston analyysissa.

4.1 Tutkimusaineisto

Yksi aineiston valintakriteereistäni oli, että tutkimieni ohjeiden on oltava tuotteen valmistajan tai muun virallisen tahon julkaisemia. En siis sisällyttänyt aineistoon epävirallisia käyttöohjeita, kuten yhteisön muokkaamia Wiki-sivuja, sillä van der Meij'n ym. (2003) tavoin halusin keskittyä kaupallisiin käyttöohjeisiin. Keräsin käyttöohjeet tuotteiden valmistajien sivuilta 1.12.2017–28.2.2018. Osa ohjeista oli PDF-muodossa ja siten erikseen ladattavissa, mutta suurin osa ohjeista oli luettavissa suoraan valmistajien sivuilla. Poikkeuksia olivat HTML5- (Korpela 2014), Oracle- (Hakkarainen 2011) ja Java- (Peltomäki 2014) oppaat, jotka lainasin kirjastosta. Pyrin sisällyttämään aineistoon mahdollisimman samankaltaisia tuotteita kuin van der Meij ym. (2003), kuten kuvankäsittelyohjelmia, käyttöjärjestelmiä, pelin, näyttöjä, skannerin ja kameran. En kuitenkaan täysin päässyt tavoitteeseeni kielierojen ja tekniikan kehittymisen vuoksi. En löytänyt suomenkielisiä käyttöohjeita esimerkiksi näytönohjaimille tai äänikorteille, ja monet alkuperäisen tutkimuksen aikana suosittu tuotteet ovat lähes kadonneet markkinoilta. Tätä tasapainottaakseni otin mukaan muutamia uudempia tuotteita, joita ei vielä ollut keksitty vuosituhaten vaihteessa, kuten älypuhelimet ja virtuaalilasit. Valikoin aineistoni tuotteiden enkä valmistajan perusteella, mutta koska markkinoita hallitsevat pääasiassa muutamat suuret valmistajat, aineistossani on muutamia samojen valmistajien tuotteita.

Koska tavoitteeni oli tutkia nelikantamallin soveltuvuutta paitsi uuteen kieleen, myös uuteen ajalliseen kontekstiin, valitsin aineistooni ohjeita, jotka on julkaistu 2010-luvulla. Suoraan valmistajien sivuilla luettavat ohjeet eivät kuitenkaan läheskään aina ilmoittaneet, milloin ne on viimeksi päivitetty. Tällaisissa tapauksissa valitsin suosittuja ohjelmistoja ja laitteistoja, joiden ohjeita todennäköisesti pidetään ajan tasalla, kuten Spotify ja Google Chrome, tai varmistin, että ohjeissa käsitelty tuote on julkaistu vuoden 2010 jälkeen. Vaikkei kaikissa käyttöohjeissa ollut julkaisuvuotta, vuoden 2010 jälkeen julkaistun tuotteen käyttöohje on luonnollisesti julkaistu samoihin aikoihin.

Analysoin yhteensä 36 vaiheittaista ohjetta, joista 18 olivat ohjelmistojen ja 18 laitteistojen käyttöohjeista. Aineisto julkaisijoineen on listattu lähdeluettelossa, ja aineiston sisältämien tuotteiden tyypit on listattu liitteessä 1. Toisin kuin loput aineistoista, HTML5- (Korpela 2014), Oracle- (Hakkarainen 2011) ja Java- (Peltomäki 2014) ohjekirjat eivät ole tietyn tuotteen käyttöohjeita, mutta niiden ohjeistavat osiot ovat kuitenkin luokiteltavissa yksittäisiksi vaiheittaisiksi ohjeiksi, vaikka niiden ohjeistuksen muoto poikkeaaakin tuotekohtaisista käyttöohjeista. Samasta syystä aineistoni sisältää muutamia muitakin numeroimattomia ohjeita, vaikkei numeroimattomuus olekaan luvun 3.3.3 suositusten mukaista. Numeroiden puuttumisesta huolimatta tehtävät kuitenkin etenevät vaiheesta toiseen. Olen siis valinnut aineiston informaatiotyypin enkä ulkomuodon perusteella.

Van der Meij ym. (2003) eivät maininneet, millä perusteella he valitsivat yksittäiset vaiheittaiset ohjeet tutkimukseensa, joten valitsin ohjeet satunnaisesti käyttöohjeiden keskivaiheilta. Joissakin tapauksissa oli vaikea tulkita, mitkä osiot muodostivat yhtenäisen vaiheittaisen ohjeen alaotsikkoineen tai vaihtoehtoisine menettelytapoineen (ks. luku 5.1.1) ja mitkä koostuivat erillisistä ohjeista. Käytin määrittelyssä luvussa 2 mainittuja alku- ja päämäärätilojen (Newell 1980) sekä välitilojen (Farkas 1999) käsitteitä. Luokittelin yhtenäisiksi ohjeiksi sellaiset tilanteet, jotka alkavat samasta alkutilasta ja päättyvät samaan päämäärätilaan. Esimerkiksi ohje ”Käyttäjätilin luominen Windowsissa” (Microsoft 2016) sisältää ohjeet Windows 7:lle ja 8.1:lle. Laskin nuo kaksi osiota eri ohjeiksi, sillä ne käsittelevät käyttöjärjestelmän eri versioita eivätkä siten ala samasta alkutilasta tai johda samaan päämäärään. Otsikon ”Windows 8.1” alla on puolestaan vaihtoehtoiset otsikot ”Luo Microsoft-tili” ja ”Luo paikallinen tili”, joilla on sama alkutila (ts. käyttäjä haluaa luoda käyttäjätilin Windows 8.1:een) ja päämäärä (ts. käyttäjä on luonut tilin Windows 8.1:een), mutta alkutilasta päämäärään johtaa erilainen menettelytapa. Microsoft- ja paikallisen tilin luominen ovat siis keskenään vaihtoehtoisia menettelytapoja.

4.2 Menetelmä

Tutkimukseni perustuu kvalitatiiviseen ja kvantitatiiviseen analyysiin, tarkemmin määriteltynä kvalitatiiviseen, pääasiassa teorialähtöiseen sisällönanalyysiin sekä kvantitatiiviseen sisällön erittelyyn. Tuomen ja Sarajärven (2018, 81) mukaan teorialähtöinen sisällönanalyysi pohjautuu nimensä mukaisesti tiettyyn teoriaan tai malliin, joka kuvaillaan tutkimuksessa ja jonka mukaisesti tutkimuksen käsitteet määritellään. Teorialähtöisessä sisällönanalyysissä aineisto siis järjestetään

johtopäätösten tekoa varten tietyn teorian, mallin tai käsitejärjestelmän ohjaamalla tavalla (mts. 94), ja tutkimukseen saattaa liittyä ennestään käytetyn teorian testaaminen uudessa kontekstissa (mp.), mikä on yksi tämän tutkielman lähtökohdista. Toisenlainen näkökulma sisällönanalyysiin on aineistolähtöinen analyysi, jossa tutkija ei ole päättänyt aineistoyksikköjä etukäteen ja jossa luokittelu nousee itse aineistosta (mts. 95). Koska yksi tutkimuskysymyksistäni liittyy suoraan nelikantamallin sovellettavuuteen, tässä tutkielmassa käyttöohjeiden piirteiden luokittelu perustuu suurimmaksi osaksi nelikantamalliin ja sen käsitteistöön. Muutamia luokituksia ovat kuitenkin lähtöisin aineistosta, joten tutkimuksessani on myös teoriaohjaavia piirteitä, mikä tarkoittaa valmiiden mallien ja aineistolähtöisyyden vuorottelua (mts. 81). Kyseinen sisällönanalyysin muoto on kuitenkin tutkimuksessani sivuosassa, joten tutkielmani ei ole kokonaan teoriaohjaava.

Vaikka laadullinen ja määrällinen tutkimus eivät välttämättä ole ristiriidassa keskenään, Tuomi ja Sarajärvi (mts. 100) huomauttavat, että laadullisin menetelmin kerätyt aineistot ovat usein niin pieniä, ettei niiden kvantifointi eli sisällön erittely yleensä tuo tutkimukselle lisäarvoa. Aineiston esittäminen numeerisessa muodossa saattaa kuitenkin selkeyttää sitä ja tuoda esiin sen tarkemman kontekstin (mp.). Sekä van der Meij'n ym. (2003) että edellisessä luvussa esittelemäni aineisto olisi periaatteessa mahdollista kuvailla ainoastaan sanallisesti, mutta kvantifointi helpottaa sen hahmottamista.

Analysoimani yksiköt ovat nelikantamallin osatekijöitä, jotka esittelin luvussa 3. Jaoin nämä osatekijät van der Meij'n ym. (2003) esimerkin mukaisesti kategorioihin, jotka esittelen taulukossa 4. Edellä mainittujen tutkijoiden luokituksia olivat pitkälti käyttökelpoisia sellaisenaan, mutta loin Otsikot-osion yksiköt aineiston pohjalta ja täydensin muutamia tulkinnanvaraisia määritelmiä Newellin (1980) ja Farkasin (1999) esittelemillä konsepteilla (ks. luku 2). Van der Meij ym. (2003, 131; 184) mainitsevat lyhyesti kehittämänsä hollanninkielisen koodikirjan, johon heidän tarkat laskelmansa perustuivat, ja on mahdollista, että kyseinen kirja sisälsi yksiköiden tarkat määritelmät. He eivät kuitenkaan esitele sitä artikkelissaan.

Van der Meij ym. (2003) luokittelivat ohjelmisto- ja laitteisto-ohjeiden löydökset erillisiin taulukoihin, mutta vertailun sujuvoittamiseksi sijoitin ne samoihin taulukoihin. Käsittelen ohjelmisto- ja laitteisto-ohjeiden eroja silloin kun niitä ilmenee, mutta tutkielmani pääasiallinen tavoite on käsitellä niitä kokonaisuutena. Poikkesin van der Meij'n ym. (2003) mallista siinäkin suhteessa, että jätin pois keskihajonnan luvun 5.1.2 laskelmista, sillä se ei ollut tarpeellinen

aineiston havainnollistamisen kannalta. Keskihajontaa käytetään muun muassa laskelmien luotettavuuden arvioinnissa (KvantiMOTV 2004), joten siitä saattaisi olla hyötyä suuremman aineiston arvioinnissa.

Taulukko 4. Analyysiyksiköt

Nelikantamallin osatekijä	Laskettavat yksiköt
Tavoitteet	<p>Tehtävänälyysi eli tavoitteen saavuttaminen:</p> <ul style="list-style-type: none"> • Vaihtoehtoiset operaattorit ja/tai vaihtoehtoiset menettelytavat <ul style="list-style-type: none"> ○ Vaihtoehtoiset operaattorit ○ Vaihtoehtoiset menettelytavat • Perustelut oikean operaattorin tai menettelytavan valintaan <p>Vaiheiden sisältö:</p> <ul style="list-style-type: none"> • Vaiheiden määrä yksittäisissä ohjeissa • Kuinka moni vaihe per ohje sisältää erillistä kuvallista palautetta • Kuinka moni vaihe per ohje sisältää tekstimuotoista tietoa erillään vaiheista • Kuinka moni vaihe per ohje sisältää tekstimuotoista tietoa vaiheen yhteydessä <p>Otsikot:</p> <ul style="list-style-type: none"> • Imperatiivi • IV infinitiivi (-minen) • Kysymys • Perusmuoto • Partitiivi
Edellytykset	<ul style="list-style-type: none"> • Järjestelmän alkutilaa kuvaillaan tai se näytetään • Viittaus muualle • Ennakkojäsentäjät (mukaan lukien päämäärän kuvailut)
Toiminnot ja reaktiot	<p>Itsenäisyys ja neuvominen:</p> <ul style="list-style-type: none"> • Käyttäjää kehoitetaan tutkimaan järjestelmää omin päin <p>Kuvat:</p> <ul style="list-style-type: none"> • Kuva, jonka osat on nimetty • Kuva, joka esittää tapahtumisen järjestyksen (<i>flow</i>) • Kuva, jonka osat on nimetty ja joka esittää tapahtumisen järjestyksen (<i>flow</i>) • Irrallinen kuva (<i>flat</i>) <ul style="list-style-type: none"> ○ Kuvateksti ○ Korostus <p>Vaiheet:</p> <ul style="list-style-type: none"> • Vaiheiden korostaminen <ul style="list-style-type: none"> ○ Numeroidut vaiheet ○ Vaiheita ei ole eritelty muusta tekstistä ○ Vaiheet listattu numeroimattomiin kappaleisiin

	<ul style="list-style-type: none"> • Käyttöliittymäelementtien korostaminen <ul style="list-style-type: none"> ○ Visualisoitu painike ○ Painikkeen nimi ja teksti ”painike” tms. (esim. Enter-painike) ○ Vain painikkeen nimi
Ei-toivotut tilat	<p>Varoitukset:</p> <ul style="list-style-type: none"> • Varoitus erottuu muusta tekstistä • Huomiosana • Uhka tai haitta kuvaillaan • Varoituksen huomiotta jättämisen seuraukset kuvaillaan • Lukijalle neuvotaan mitä hänen täytyy tehdä tai olla tekemättä <p>Ongelmanratkaisu:</p> <ul style="list-style-type: none"> • Ohje sisältää tietoa mahdollisista ongelmista

Analysoitavista yksiköistä riippuen kirjaan ylös joko sen, kuinka monessa valitsemistani vaiheittaisista ohjeista kulloinkin analysoitava yksikkö esiintyy tai sen, kuinka monta kertaa yksikkö esiintyy aineistossa yhteensä. Ilmoitan käyttämäni laskentaperusteet esitellessäni tuloksia tarkemmin luvussa 5.

5 Aineiston analyysi

Käsittelin aineistoani osatekijä ja sen sisältämät yksiköt kerrallaan (ks. taulukko 4). Ensimmäiseksi esittelen käyttäjän tavoitteisiin liittyvät löydökseni, seuraavaksi tehtävien vaatimat edellytykset, sitten käyttäjän ja järjestelmän toiminnot ja reaktiot ja viimeiseksi ei-toivotut tilat, joihin järjestelmä saattaa päätyä ja joista käyttäjän on päästävä pois.

5.1 Tavoitteet

Van der Meij'n ym. (2003) tavoitteiden luokittelu pohjautuu pitkälti luvussa 3.1. esittelemääni GOMS-malliin. Tehtävä voi sisältää yhden tai useamman menettelytavan, joiden valinnassa käyttäjää on ohjeistettava. Menettelytavat taas on jaettu operaattoreita kuvaileviin työvaiheisiin. Näitä vaiheita suorittamalla käyttäjä etenee kohti tavoitteitaan. Sama työvaihe voi sisältää keskenään vaihtoehtoisia operaattoreita. Alkuperäisessä artikkelissa (van der Meij ym. 2003, 138) sekä vaihtoehtoisia menettelytapoja että vaiheiden sisäisiä vaihtoehtoja kutsutaan vaihdellen termeillä *method* ja *approach*, ja vaiheiden sisäiset vaihtoehdot on laskettu *Alternative action steps* -sarakkeeseen. Jotta laskemieni yksiköiden ero olisi selkeämpi, GOMS-mallia mukailen kutsun vaihtoehtoisia tehtäväsarjoja menettelytavoiksi ja samaan vaiheeseen sisällytettyjä, vaihtoehtoisia toimintoja operaattoreiksi. Tarkoitan tässä yhteydessä termillä *suoritustapa* sekä menettelytapoja että operaattoreita.

Esittelen tavoitteista saamani tulokset seuraavissa alaluvuissa. Ensimmäiseksi kuvailen, kuinka paljon vaihtoehtoisia operaattoreita tai menettelytapoja aineistoni sisälsi. Seuraavaksi esitän laskelmani siitä, minkä verran vaiheita tutkimani ohjeet sisälsivät ja millaista lisätietoa vaiheisiin oli sisällytetty operaattoreiden lisäksi. Lopuksi kerron, missä muodossa vaiheittaisten ohjeiden pää- ja alaotsikot oli esitetty.

5.1.1 Tehtäväanalyysi eli tavoitteen saavuttaminen

Laskin vaihtoehtoiseksi operaattoreiksi sellaiset toiminnot, joita ei ole eroteltu omalle rivilleen tai erilliseksi ohjeeksi. En kuitenkaan luokitellut toisensa poissulkevia operaattoreita – esimerkiksi sivuston tai kuvan tallentaminen Chromessa (Google s. d.) – toistensa vaihtoehdoiksi, sillä ne eivät tähtää samaan päämäärään. Alla oleva esimerkki havainnollistaa vaihetta, joka sisältää kaksi vaihtoehtoista operaattoria:

Useimmat tiedostot: Klikkaa latauslinkkiä. Voit myös klikata tiedostoa hiiren kakkospainikkeella ja valita **Tallenna nimellä**. (Google s. d.-b)

Toisin sanoen vaiheen sisällä kerrotaan kaksi erilaista tapaa, joilla käyttäjä voi päästä samaan päämäärään eli tiedostojen tallentamiseen Google Chromessa. Vaihtoehtoiset menettelytavat eli menetelmät (ks. luku 3.1) puolestaan ovat kokonaisia vaiheittaisia ohjeita, joita noudattamalla käyttäjä voi suorittaa saman tehtävän. Koska ne on esitetty saman pääotsikon alla, olen kuitenkin laskenut ne samaksi vaiheittaiseksi ohjeeksi. Seuraavat aineistoesimerkit havainnollistavat ohjeessa ensiksi mainittua menettelytapaa ja sen vaihtoehtoa:

Kuvien ja videoiden tuominen digitaalikamerasta


1. Liitä kamera tietokoneeseen kameran USB-kaapelilla.
2. Käynnistä kamera.
3. Valitse näkyviin tulevassa **Automaattinen toisto** -valintaikkunassa **Tuo kuvat ja videot Windowsilla**.
4. (Valinnainen) Jos haluat lisätä kuviin tunnisteeseen, kirjoita tunnisteeseen nimi **Lisää tunniste seuraaviin kuviin (valinnainen)** -ruutuun.

Kuvien ja videoiden tuontipäivämäärä sekä tunnisteeseen nimi näkyvät kansion nimessä oletusarvoisesti.

5. Valitse **Tuo**.

Uusi ikkuna avautuu ja tuo näkyviin tuodut kuvat ja videot. (Microsoft 2016.)

Entä jos automaattisen toiston valintaikkuna ei tule näyttöön?

1. Avaa tietokone napsauttamalla **Käynnistä**-painiketta  ja valitse **Tietokone**.
2. Napsauta vasemmalla olevan siirtymisruudun **Tietokone**-kohdassa digitaalista kameraa hiiren kakkospainikkeella ja valitse **Tuo kuvia ja videoita**.
3. (Valinnainen) Jos haluat lisätä kuviin tunnisteeseen, kirjoita tunnisteeseen nimi **Lisää tunniste seuraaviin kuviin (valinnainen)** -ruutuun.

Kuvien ja videoiden tuontipäivämäärä sekä tunnisteeseen nimi näkyvät kansion nimessä oletusarvoisesti.

4. Valitse **Tuo**.

Uusi ikkuna avautuu ja tuo näkyviin tuodut kuvat ja videot. (Microsoft 2016.)

Kuvien ja videoiden automaattinen tuominen kamerasta on ensisijainen menettelytapa, mutta ohje kuvailee sille myös manuaalisen vaihtoehdon.

Vaikka van der Meij ym. (2003, 138) eivät kertoneet tehneensä niin, laskin suoritustapojen valinnan perusteluksi käyttäjälle niiden otsikot. Toimin näin siksi, että tuloksista olisi selkeästi havaittavissa, että jotkin ohjeet eivät anna lainkaan viitteitä siitä, millä perusteella käyttäjän tulisi valita tietty vaihtoehtoinen työvaihe tai menettelytapa toisen sijaan. Tämä on ristiriidassa GOMS-mallin kanssa (ks. luku 3.1), jonka mukaan käyttäjälle on annettava tarpeeksi tietoa, jotta hän osaa valita itselleen sopivimman vaihtoehdon. Kaikki löytämäni vaihtoehdot eivät tosin kaivanneet perusteluja – tehtävän suorittamisen kannalta ei ole juurikaan väliä sillä, käytetäänkö siihen hiirtä vai pikanäppäimiä.

Esittelen taulukossa 5, kuinka moni tutkimistani vaiheittaisista ohjeista sisälsi vaihtoehtoisia operaattoreita ja/tai menettelytapoja ja kuinka moni ohjeisti käyttäjää siinä, mikä suoritustapa hänen tulisi valita:

Taulukko 5. Tavoitteen saavuttaminen

Yksikkö	Ohjeet, jotka sisälsivät yksikön (ohjelmistot)	Ohjeet, jotka sisälsivät yksikön (laitteistot)
Vaihtoehtoiset operaattorit ja/tai vaihtoehtoiset menettelytavat	12	12
Vaihtoehtoiset operaattorit	8	11
Vaihtoehtoiset menettelytavat	5	4
Perustelut oikean operaattorin tai menettelytavan valintaan	7	6

Yhteensä 24 tutkimistani 36 vaiheittaisesta ohjeesta sisälsi vaihtoehtoisia suoritustapoja, ja näistä ohjeista 13 perusteli, miksi käyttäjän tulisi valita yksi vaihtoehto ylitse muiden. Kaikista tutkimistani ohjeista 12 sisälsi vain yhden tavan tehtävän suorittamiseen. Tehtävän vaihtoehtoinen suoritustapa esitettiin 19 ohjeessa työvaiheiden sisällä tai muutoin vaiheiden yhteydessä, kun taas kokonaan vaihtoehtoisia menettelytapoja esiintyi 9 vaiheittaisessa ohjeessa. Näissä kahdessa kategoriassa (vaihtoehtoiset operaattorit ja vaihtoehtoiset menettelytavat) oli jonkin verran päällekkäisyyksiä, mutta useimmiten ohje ei tarjonnut joko lainkaan vaihtoehtoja tai se tarjosi molemmat näistä kategorioista. Kun käyttäjälle tarjottiin yksittäisen ohjeen sisällä vaihtoehtoisia operaattoreita tietyn vaiheen suorittamiseen, kyse oli yleensä laite- tai käyttöjärjestelmäkohtaisista eroista. Yleensä eri laitteille ja käyttöjärjestelmille oli kuitenkin omat ohjeensa, eikä niitä ollut sekoitettu samaan vaiheittaiseen ohjeeseen tai vaiheeseen. Toisinaan saman vaiheen pystyi suorittamaan joko valikon tai pikanäppäimien kautta, mutta enimmäkseen ohjeissa esiintyi vain yksi tapa.

Aineistossani oli huomattavasti enemmän vaihtoehtoja tehtävien suorittamiseen ja näiden vaihtoehtojen perusteluja kuin van der Meij'n ym. (2003) aineistossa. On mahdollista, että käyttöohjeiden siirtyminen pitkälti sähköiseen muotoon on tehnyt niistä laajempia, koska julkaisijoiden ei enää tarvitse huolehtia painokustannuksista. Lisäksi teknologian kehittyminen on tehnyt ohjelmistoista ja laitteistoista monimutkaisempia, jolloin käyttäjälle on tarjolla enemmän vaihtoehtoja, mikä taas vaatii pidempiä käyttöohjeita. Eräs mahdollisuus on myös se, että olen käyttänyt jossain määrin erilaisia laskentaperiaatteita kuin van der Meij ym., sillä he eivät kerro, kuinka he täsmälleen määrittelivät vaihtoehdot ja perustelut. Aineistossa on kuitenkin myös yhtäläisyyksiä: vaihtoehdot on molemmissa useimmiten esitetty työvaiheiden sisällä, eikä läheskään kaikkia vaihtoehtoja ole perusteltu.

5.1.2 Vaiheet

Kuten esittelin taulukossa 4, laskin vaiheiden määrän yksittäisissä ohjeissa sekä sen, kuinka moni näistä vaiheista sisälsi operaattoreiden lisäksi kuvallista ja/tai tekstimuotoista, käyttäjälle hyödyllistä palautetta tai muuta tietoa. Melko harva yksittäinen vaihe aineistossani koostui ainoastaan verbistä ja objektista, vaan useimmat sisälsivät myös lisätietoa. Annan tästä esimerkin, joka ei ole lähtöisin aineistostani: ”Valitse **Tulosta**. Dokumenttisi lisätään tulostusjonoon.” ”Valitse **Tulosta**” on päälause, ja ”Dokumenttisi lisätään tulostusjonoon” on sitä tukevaa lisätietoa, joka

ilmoittaa järjestelmän reaktiosta käyttäjän toimintaan. Ohje toimisi periaatteessa vain yhden operaattorin varassa, mutta lisätieto antaa käyttäjän tietää, mitä hänen toiminnastaan pitäisi seurata. Jos kyseistä seurausta ei siis tapahdu, käyttäjä tietää, että jotakin on vialla.

Tekstimuotoinen lisätieto työvaiheiden yhteydessä tarkoittaa muun muassa vaiheen seurausten kuvailua heti vaiheen jälkeen, kuten *Uusi ikkuna avautuu* [...]. (Microsoft 2016) (ks. myös välitilat ja osatavoitteet sekä järjestelmän toiminnot luvussa 2.1). Erillinen lisätieto taas tarkoittaa esimerkiksi vihjelaatikkoa, joka on sijoitettu varsinaisen vaiheittaisen ohjeen jälkeen tai työvaiheiden väliin ja joka on erotettu muusta tekstistä esimerkiksi taustavärillä tai kehyksellä (ks. huomautukset luvussa 2.3). Laskin erilliseksi lisätiedoksi myös kehotukset tutustua käyttöohjeen muihin osioihin. En laskenut kuvalliseksi lisätiedoksi sellaisia kuvia, jotka ainoastaan kuvittivat ohjeita eivätkä havainnollistaneet työvaiheita ja joilla ei siten ollut lisäarvoa tehtävän suorittamisen kannalta. Tällaiset kuvat eivät kuitenkaan ole turhia, sillä kaikenlaiset kuvat kiinnittävät lukijan huomion ja lisäävät hänen motivaatiotaan käyttöohjeiden lukemiseen, vaikkeivat ne suoranaisesti auttaisi häntä tietyssä tehtävässä (Mäkynen 2012, 27).

Kun tein vaiheisiin liittyviä laskelmia, jätin pois laskuista numeroimattomat vaiheittaiset ohjeet. Toimin näin siksi, että leipätekstistä on vaikea erotella yksittäisiä vaiheita ja sitä, mikä lisätieto liittyy mihinkin vaiheeseen. Lisäksi laskin vaihtoehtoiset menettelytavat poikkeuksellisesti erillisiksi ohjeiksi. Erillisiä vaiheittaisia ohjelmisto-ohjeita on täten 16 ja laitteisto-ohjeita 18, ja suuri osa niistä sisälsi vaihtoehtoisia operaattoreita. Taulukko 6 sisältää keskiarvot ja vaihteluvälit luvuille, jotka havainnollistavat työvaiheisiin liittyviä piirteitä.

Taulukko 6. Vaiheet

Yksikkö	Keskiarvo (ohjelmistot)	Vaihteluväli (ohjelmistot)	Keskiarvo (laitteistot)	Vaihteluväli (laitteistot)
Vaiheiden määrä yksittäisissä ohjeissa	4,38	2–9	3,95	1–10

Kuinka moni vaihe per vaiheittainen ohje sisältää erillistä kuvallista palautetta	0,19	0–1	0,78	0–5
Kuinka moni vaihe per vaiheittainen ohje sisältää tekstimuotoista tietoa erillään vaiheista	0,75	0–3	0,50	0–2
Kuinka moni vaihe per vaiheittainen ohje sisältää tekstimuotoista tietoa vaiheen yhteydessä	1,19	0–2	1,22	0–5

Vain yksi ohje kummastakin kategoriasta oli jaettu välitavoitteisiin, jotka johtivat kohti samaa päämäärää. Kummassakin tapauksessa ohjeistus oli pääasiassa leipätekstin muodossa, mutta yksi välitavoitteista oli numeroitu (EA 2017 & Polar Electro 2016, 37–40). Kahdessa tapauksessa yksittäisiä vaiheita ei ollut numeroitu oletettavasti siksi, että niitä oli vain yksi.

Kuten taulukko 6 osoittaa, ohjelmistoja koskevissa vaiheittaisissa ohjeissa vaiheiden määrä oli keskimäärin 4,38, ja niiden vaihteluväli oli 2–9. Laitteistojen käyttöohjeissa vastaavat luvut olivat 3,95 ja 1–10. Keskiarvot vastaavat siis luvussa 3.1 käsiteltyjä suosituksia, joiden mukaan vaiheita tulisi olla enintään viisi. Luvut eivät kuitenkaan anna kokonaiskuvaa vaiheiden määrästä, sillä jotkin vaiheet sisälsivät useamman toiminnon, kuten seuraavista esimerkeistä ilmenee:

1. Etsi Microsoftin hiiri- ja näppäimistökeskuksessa laitteesi ja määritä sitten **Quick Turn** (pikakäännös) painikkeelle [*sic*] valitsemalla toiminto (Microsoft s. d.-c).
4. Lue ja hyväksy terveyst- ja turvallisuusvaroitukset ja käyttöoikeussopimuksen (EULA) ehdot ja napauta sitten SEURAAVA (Samsung 2017, 38).

Toisaalta van der Meij ym. (2003, 145; ks. myös luku 3.2.2) suosittelivat vaiheiden yhdistelyä häivyttämisen yhteydessä, mutta toisaalta he (mts. 141) huomauttavat, ettei yhteen vaiheeseen pidä yhdistää liian monta toimintoa.

Kuvallinen palaute oli ohjelmisto-ohjeissa harvinaista, mutta laitteistojen kohdalla sitä esiintyi enemmän. Vaiheista erillisen tiedon laskeminen oli haastavaa, sillä van der Meij ym. eivät määritelleet sitä tarkemmin tai antaneet siitä esimerkkejä. Päädyin luokittelemaan erilliseksi, tekstimuotoiseksi tiedoksi varsinaisista vaiheista erillään esitetyn tiedon, jonka sisältö kuitenkin selvästi liittyi tiettyyn vaiheeseen, kuten kuvassa 2, jossa hehkulamppusymbolilla merkitty huomautus tarkentaa vaihetta 4.

4. Kirjoita tai valitse haluamasi yhteys henkilön tiedot.

5. Kun olet valmis, napauta **TALLENNA**.

! Kun tallennat yhteystiedon tiliin, kyseinen tili näytetään tallentamisen oletustilinä, kun seuraavan kerran lisäät yhteystiedon. Jos haluat vaihtaa tiliä, jolle yhteystieto tallennetaan, sinun täytyy luoda yhteystieto uudelleen ja tallentaa se sitten oikeaan tiliin.

☛ Jos lisäät plusmerkin ja maatunnuksen yhteys henkilön puhelinnumeron eteen, numeroa ei tarvitse muokata, kun myöhemmin soitat puheluita ulkomailla.

Kuva 2. Tekstimuotoinen lisätieto erillään työvaiheesta (Sony s. d.-c)

Yleensä tekstimuotoinen lisätieto oli kuitenkin sijoitettu vaiheiden yhteyteen. Suurin osa lisätiedosta sopii Farkasin ohjeistukseen yksinkertaistetusta vaiheista (ks. luku 2.3), sillä ohjeen päälauseetta seuraava tieto kuvailee yleensä järjestelmän reaktiota käyttäjän toimintoihin eli sitä, millaiseen tilaan järjestelmä siirtyy (ks. luku 2.1.2). Seuraavat työvaiheet ovat tyypillisiä esimerkkejä ydinlauseesta ja sen yhteydessä esiintyvistä lisätiedosta:

3. Määritä tarkkuus, väritila ja bittisyvyys.

Jos olet kopioinut valinnan Leikepöydälle, kuvan mitat ja tarkkuus ovat automaattisesti näiden kuvatietojen mukaiset. (Adobe 2016.)

1. Avaa apuohjelma, jolla hallitaan WiFi-yhteyksiä reitittimeen liitettävässä tietokoneessa tai WiFi-laitteessa.

Apuohjelma etsii kaikki alueen langattomat verkot. (Netgear 2015, 2.)

Molemmissa tapauksissa jälkimmäinen lause kuvailee, kuinka järjestelmä reagoi toimintaan, johon käyttäjää on opastettu vaiheen ydinlauseessa.

Saamani tulokset eivät juurikaan poikenneet van der Meij'n ym. (2003, 140) laskelmista. Heidän aineistonsa ohjeiden keskimääräiset pituudet olivat 4,47 ja 4,33 vaihetta, ja myös heidän

tuloksissaan tekstimuotoinen, vaiheista erotettu lisätieto oli yleisempää ohjelmisto- kuin laitteisto-ohjeissa. Useimmissa muissa luvuissa on keskenään vain pieniä poikkeamia. Suurin ero oli se, että aineistossani keskimäärin 0,19 ohjelmisto-ohjeiden vaiheista sisälsi kuvallista palautetta, kun taas alkuperäisessä artikkelissa vastaava tulos on 0,85. Aineistoni koon huomioon ottaen kyseessä saattaa olla vain sattuma, mutta on myös mahdollista, että 1990-luvun ja 2000-luvun alun käyttäjien koettiin tarvitsevan enemmän kuvallista tukea kuin nykykäyttäjien.

5.1.3 Otsikot

Van der Meij'n ym. (2003) englanninkielisten otsikoiden luokittelu ei luonnollisesti sovi suomenkielisiin käyttöohjeisiin sellaisinaan, vaan jouduin luomaan uusia kategorioita aineiston pohjalta. Tässä osuudessa analyysini on siis poikkeuksellisesti aineisto- eikä teorialähtöistä. En laskenut huomautuksia tai varoituksia alaotsikoiksi kolmesta syystä: ensinnäkin van der Meij ym. eivät tee niin, ja lisäksi Farkas luokittelee otsikot ja huomautukset eri kategorioiksi (ks. luku 2.3), eivätkä huomautusten tai varoitusten otsikot kuvaile käyttäjän tavoitteita.

Laskiessani alaotsikoita en laskenut sitä, kuinka monessa ohjeessa ne esiintyivät, vaan yksittäisten otsikoiden määrän koko aineistossa. Pääotsikoita oli itsestään selvästi vain yksi jokaisessa vaiheittaisessa ohjeessa, mutta joissakin ohjeissa oli useampi alaotsikko. Taulukko 7 sisältää yhteenvedon siitä, minkä muotoisia otsikoita löysin aineistosta.

Taulukko 7. Otsikot

Yksikkö	Esimerkki	Otsikot (ohjelmistot)	Alaotsikot (ohjelmistot)	Otsikot (laitteistot)	Alaotsikot (laitteistot)
Imperatiivi	Luo Microsoft-tili (Microsoft 2016b) Lopeta pelin jakaminen (Sony s. d.-b)	-	2	2	2
IV infinitiivi (-minen)	Kuvien ja videoiden tuominen digitaalikamerasta (Microsoft 2016) Kuvan luominen (Adobe 2016)	8	4	9	3

Kysymys	Entä jos automaattisen toiston valintaikkuna ei tule näyttöön? (Microsoft 2016) Miten luon pinoja? (Evernote s. d.)	1	6	-	5
Perusmuoto	Kuuntelu offline-tilassa (Spotify s. d.) Valituksen korjaus (Nikon 2016, 115)	8	-	7	8
Partitiivi	Taulukoiden kuvailun tekniikoita (Korpela 2014, 412)	1	-	-	-
Muu	Kun Pelin jakaminen - toimintoon ei saada yhteyttä (Sony s. d.-b) Ennen testiä (Polar Electro 2016, 42)	-	-	-	2
Yhteensä		18	12	18	20

Kuten taulukosta ilmenee 7, 4. infinitiivi eli -minen-pääte oli otsikoiden yleisin muoto – se esiintyi 17 yksittäisessä ohjeessa, mikä on lähes puolet koko aineistosta. Perusmuotoiset otsikot olivat lähes yhtä yleisiä. Niitä oli 15 käyttöohjeessa. Alaotsikkojen kesken oli enemmän vaihtelua, mutta toisaalta niitä esiintyi vain 12 ohjeessa. Ne olivat yleensä eri muodossa kuin yläotsikot. Farkasin (1999) mukaan alaotsikot ovat pääasiassa infinitiivimuodossa (ks. luku 2.3), mutta hänen havaintonsa perustuivat englanninkielisiin otsikoihin, joita ei voi suoraan verrata suomenkielisiin otsikoihin.

Tuloksissani oli enemmän hajontaa kuin van der Meij'n ym. (2003, 143) tuloksissa, sillä heidän aineistossaan gerundi- eli *-ing*-muotoinen otsikko oli selvästi yleisin. Aineistossani gerundia

lähimmin vastaava muoto eli -minen-päätteiset otsikot olivat kyllä yleisimpiä, mutta perusmuotoisia otsikoita oli lähes yhtä paljon. Muu-kategoria sisälsi vain taulukon 7 toiseksi viimeisen rivin kaksi otsikkoa. Van der Meij'n ym. (2003, 141) *Other*-kategoria puolestaan sisälsi yhteensä 15 otsikkoa, joista osa on kysymyksiä, joten on mahdollista, että jotkin muutkin löytämistäni otsikkotyypeistä sisältyisivät siihen. Tutkimani aineiston perusteella suomenkielisillä alaotsikoilla ei siis ole vakiintunutta muotoa.

5.2 Edellytykset

Samoin kuin alkuperäisen nelikantamallitutkimuksen laatijat, en voinut arvioida käyttöohjeiden häivytystä yhden vaiheittaisen ohjeen perusteella. Häivytyksen arviointi vaatisi kokonaisten käyttöohjeiden analysointia.

Tutkin aineistoni modulaarisuutta arvioimalla, kuinka usein vaiheittaiset ohjeet esittelevät järjestelmän alkutilan ja viittaavat muualle. Järjestelmien alkutilat ja viittaukset olivat aineistossa enimmäkseen itsestään selviä, mutta ennakkojäsentäjien tunnistaminen ja siten laskeminen oli haasteellista. Van der Meij ym. (2003, 146) kyllä kuvailevat, mitä ennakkojäsentäjät ovat (ks. myös luku 3.2.2), mutta he eivät anna niistä kovin havainnollisia esimerkkejä. Käytin analyysin tukena seuraavaa tarkentavaa määritelmää:

Ennakkojäsentäjät esitellään aina selittävän opetuksen alussa, jotta ne ”kuroisivat umpeen eron sen välillä, mitä lukija tietää ennestään ja mitä hänen on tiedettävä, jotta hän voi oppia käsillä olevan tehtävän mielekkäällä tavalla” [...]. Ne auttavat oppilaita jäsentämään heille esitellyn uuden tiedon. (O'Donnell, Reeve & Smith 2011, 58; ks. myös Lidwell, Elam, Butler & Holden 2010, 18–19.) [Oma suomennokseni.]

Ennakkojäsentäjä on siis eräänlainen konseptuaalinen elementti (ks. luku 2.3). Se ei kuitenkaan ole vain tiivistelmä sitä seuraavasta tehtävästä (Dell'Olio & Donk 2007, 393). Seuraavat esittelevät ennakkojäsentäjät selventävät ohjelmistojen toimintaa:

The Sims 4 tallentaa useita tallennuspisteitä valittavaksi siltä varalta, että tiedot vioittuvat tai haluat palata aiempaan tallennukseen (EA 2017).

Kuuntele musiikkia ilman internetyhteyttä ja säästä mobiilidataa lataamalla jopa 3 333 kappaletta laitetta kohden korkeintaan kolmelle eri laitteelle (Spotify s. d.).

Päämäärän kuvailu ei ole varsinaisesti ennakkojäsentäjä, mutta se voidaan esittää jäsentäjien yhteydessä, ja alkuperäisessä nelikantamallissa se laskettiin samaan kategoriaan kuin jäsentäjät. Yleensä se tarkoittaa, mitä käyttäjä voi saavuttaa ohjeen vaiheita seuraamalla. (van der Meij ym. 2003, 149.) Seuraava vaiheittaisen ohjeen aloitus on esimerkki päämäärän kuvailusta, ja se on myös yksi harvoista tutkimistani vaiheittaisista ohjeista, joissa lukijan toimintaan viitataan passiivi- eikä imperatiiviverbeillä:

Ostojen raportoinnilla voidaan seurata ja vertailla ostoja sekä tuote- ja toimittajakohtaisia tunnuslukuja (Procountor s. d.).

Esittelen taulukossa 8 kuinka monta aloitustilan kuvausta, viittausta ja ennakkojäsentäjää aineistoni vaiheittaisissa ohjeissa oli yhteensä.

Taulukko 8. Edellytykset ja modulaarisuus

Yksikkö	Kpl (ohjelmistot)	Kpl (laitteistot)
Järjestelmän alkutilaa kuvaillaan tai se näytetään	4	3
Viittaus muualle	9	13
Ennakkojäsentäjät (mukaan lukien päämäärän kuvailut)	8	10

Yhteensä 7 vaiheittaista ohjetta kuvaili tehtävän edellyttämää alkutilaa (ks. luku 2.1.2) muuan muassa seuraavilla tavoilla:

Muistiinpanopinot ovat tällä hetkellä käytettävissä seuraavissa tuotteissa: Evernote Web, Mac, Windows, Android sekä iPhone, iPad ja iPod Touch (Evernote s. d.).

Varmista ennen sovelluksen asentamista, että mobiililaite on yhteydessä Wi-Fi- tai matkapuhelinverkkoon (Samsung 2017, 37).

Toisinaan oli kuitenkin epäselvää, millaiset konseptuaaliset elementit kuvailivat tehtävän edellytyksiä. HUR Labsin (2015, 10) julkaisema käyttöohje sisälsi ennen varsinaista tehtävää tällaisen lauseen:

Ennen kalibrointia kannattaa tehdä nollakohtien mittaus, sillä se on yleisin ongelma väärille tuloksille.

Koska kyseessä on vapaaehtoinen toimenpide (”kannattaa”, vrt. esim. ”täytyy”) eikä pakollinen edellytys kalibroinnin onnistumiselle, en laskenut sitä edellytykseksi.

Ohjelmisto-ohjeista 5 ja laitteisto-ohjeista 8 viittasivat muualle käyttöohjeeseen tai kokonaan erillisiin ohjeisiin. Jotkin ohjeet sisälsivät useamman viittauksen, joten yksittäisiä viittauksia oli yhteensä 22. Enimmäkseen viittaukset olivat linkkejä lisätietoon kulloinkin käsiteltävästä aiheesta, mutta mukana oli myös tapauksia, jotka viittasivat kokonaan muihin käyttöohjeisiin, kuten seuraava esimerkki:

Apua WPS-painikkeen käyttöön saat tietokoneen tai WiFi-laitteen käyttöohjeista tai verkkotuesta (Netgear 2015, 2).

Van der Meij'n ym. (2003, 149) 104 vaiheittaisen ohjeen aineistossa vain 9 vaiheittaista ohjetta sisälsi käyttöohjeiden sisäisiä viittauksia tai viittauksia muihin ohjeisiin, mikä saattaa antaa vihjeitä siitä, että käyttöohjeiden siirtyminen pitkälti sähköiseen muotoon on tehnyt viittaamisesta yleisempää. Näin pienen aineiston pohjalta ei kuitenkaan voi tehdä johtopäätöksiä kyseisen ilmiön yleisyydestä.

Ohjelmisto-ohjeista 8 ja laitteisto-ohjeista 10 sisälsivät ennakkojäsentäjiä. En löytänyt aineistosta vertailevia jäsentäjiä, mikä ei ole järin yllättävää. Kuten arvelin luvussa 3.2.3, nykykäyttäjät tuskin tarvitsevat yhtä paljon tukea teknologian ymmärtämiseen kuin vuosituhanen vaihteen käyttäjät. Selittäviä ennakkojäsentäjiä oli aineistossa yhteensä 18. Laskin seuraavan esimerkin sekä selittäväksi jäsentäjäksi että päämäärän kuvailuksi:

Muistikirjat voidaan ryhmitellä visuaalisesti muistiinpanopinoihin, ja tavallisesti niitä käytetään samankaltaista aihetta tai teemaa käsittelevien muistikirjojen ryhmittelyyn. Muistiinpanopinot voivat sisältää sekä henkilökohtaisia että Business-muistikirjoja, mutta ne ovat vain sinun nähtävissäsi. (Evernote s. d.)

Ensimmäinen lause kuvailee mitä käyttäjä voi tehdä, kun taas kappaleen jälkimmäinen osa selittää, kuinka ohjelmisto toimii. Van der Meij ym. (2003, 150) löysivät aineistostaan yhteensä 7 ennakkojäsentäjää, joista kaikki olivat selittäviä. On jossain määrin yllättävää, että aineistossani oli enemmän jäsentäjiä, mutta kuten mainitsin edellä, jäsentäjiä oli vaikea hahmottaa. Saattaa olla, että olen laskenut jäsentäjiksi sellaisia kohtia, joita van der Meij ym. eivät olisi laskeneet.

5.3 Toiminnot ja reaktiot

Samoin kuin luvussa 3.3, käsittelen toimintoja ja reaktioita kolmessa osassa. Ensin kerron, kuinka käyttäjän neuvominen ja itsenäisyyteen rohkaisu ilmenivät tutkimissani ohjeissa. Seuraavaksi esittelen, millaisilla kuvilla ohjeet täydensivät tekstimuotoista ohjeistusta. Viimeiseksi esitän tulokseni siitä, miten ohjeet erottivat työvaiheet muusta tekstistä ja millaista sisältöä kyseisissä työvaiheissa oli.

5.3.1 Itsenäisyys ja neuvominen

Lähes kaikki vaiheittaiset ohjeet kertoivat suoraan, mitä käyttäjän tulee tehdä eivätkä rohkaisseet käyttäjää tutustumaan tuotteeseen omin päin. Selkeitä poikkeuksia oli kolmessa ohjeessa. Oracle-ohje (Hakkarainen 2011, 237) neuvoo ensin käyttäjää suorittamaan tehtävän väärällä tavalla ja vasta tämän jälkeen oikein. Ohje (mp.) myös selittää, miksi väärä tapa ei toimi ja miksi oikea toimii. Kuten mainitsin luvussa 3.2.2, näin käyttäjä ei ainoastaan opi, mitä hänen täytyy tehdä vaan myös sen, miksi hänen on tehtävä se. HTML5-opas (Korpela 2014, 412) puolestaan esittelee taulukkojen kuvailukeinot, mutta se ei suoraan kerro, mitkä keinot käyttäjän on valittava. Opas (mp.) antaa valintaan suuntaa-antavia ohjeita, mutta jättää lopullisen päätöksen lukijan harkinnan varaan. Windows 7:ää käsittelevä vaiheittainen ohje (Microsoft 2016) taas sisältää lauseen, joka kannustaa käyttäjää ohjelmiston itsenäiseen tutkimiseen:

Voit muuttaa kuvien ja videoiden tuonnin yhteydessä käytettäviä asetuksia niin, että kuvat ja videot nimetään ja järjestetään sinulle sopivalla tavalla.

Kuten totesin luvussa 3.3.1, tasapainottelu käyttäjän suoran neuvomisen ja itsenäisen tutkimisen rohkaisemisen välillä ei ole helppoa, etenkin jos käyttöohjeiden kirjoittajalla ei ole varmaa tietoa siitä, millaisista lukijoista hänen yleisönsä tarkalleen koostuu. Tämä voi olla yksi syy siihen, miksi myös van der Meij ym. (2003, 170) löysivät vain kaksi ohjetta, jotka kehottivat käyttäjää tutustumaan tuotteeseen omillaan. Osasyys kehotusten vähäisyyteen on mahdollisesti sekin, että käyttöohjeita luetaan enimmäkseen silloin, kun käyttäjä tarvitsee tiettyä rajattua tehtävää koskevia neuvoja eikä niinkään ole tutustumassa tuotteen yleisiin toimintaperiaatteisiin.

5.3.2 Kuvat

Jaoin aineistossani esiintyneet kuvat neljään kategoriaan (van der Meij ym. 2003, 170):

- Kuvat, joiden osat on nimetty.

- Kuvat, jotka esittävät tapahtumien kulun (*flow*).
- Kuvat, joissa näkyvät sekä osien nimet että tapahtumien kulku.
- ”Irralliset” (*flat*) kuvat, jotka eivät täytä edellistä kolmea ehtoa.


Van der Meij ym. (mts., 171) laskivat kuitenkin irrallisiksi myös sellaiset kuvat, joiden osioita oli korostettu esimerkiksi kehyksillä tai joissa oli kuvateksti. Loin tällaisille kuville alakategoriat, jotta laskelmistani selviäisi paremmin, millaiset kuvat ovat lähes täysin irrallisia. Aineistossa ei ollut yhtäkään kuvaa, jossa olisi ollut sekä korostuksia että kuvateksti.

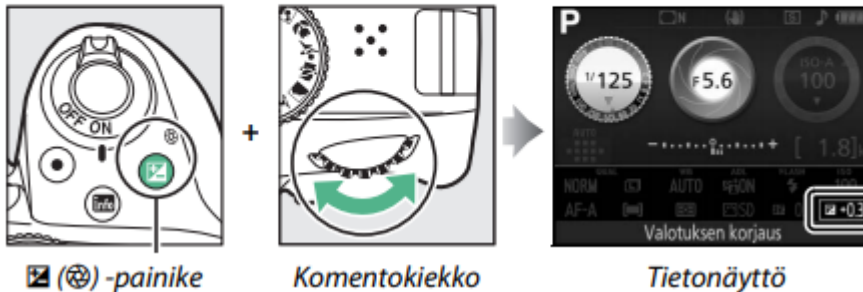
Ohjelmistoja käsittelevistä vaiheittaisista ohjeista 8 hyödynsi kuvia. Laitteistojen kohdalla vastaava luku oli 9. Kuvia oli näissä kahdessa kategoriassa yhteensä 11 ja 23, ja olen eritellyt ne tarkemmin taulukossa 9. Laskin tässä tapauksessa kuviksi vain kuvakaappaukset, valokuvat ja vektorikuvat enkä kuvakkeiden, näppäinten yms. symboleja.

Taulukko 9. Kuvat (ohjelmistot)

Yksikkö	Kpl (ohjelmistot)	Kpl (laitteistot)
Kuva, jonka osat on nimetty	2	2
Kuva, joka esittää tapahtumien järjestyksen (<i>flow</i>)	-	1
Kuva, jonka osat on nimetty ja joka esittää tapahtumien järjestyksen (<i>flow</i>)	-	1
Irrallinen kuva (<i>flat</i>)	9	19
Kuvateksti	4	5
Korostus	1	10
Ei kuvatekstiä tai korostusta	4	4

Tapahtumien järjestystä esittelevät kuvat kuvaavat järjestelmän siirtymistä tilasta toiseen (mp.). Laskin kuvan 3 vektorikuvat yhdeksi kuvaksi, sillä ne esittivät yhden työvaiheen etenemistä:

Valitse valotuksen korjausarvo pitämällä  (Ⓜ) -painiketta painettuna ja kiertämällä komentokiekkoa, kunnes haluamasi arvo on valittuna etsimessä tai tietonäytöllä.

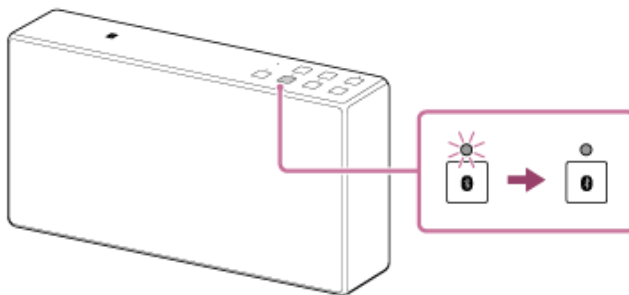


Kuva 3. Tehtävän eteneminen (Nikon 2016, 116)

Toinen esimerkki tämänkaltaisesta kuvasta on kuva 4, joka esittää kaiuttimen ilmaisimen vilkkumisen päättymistä:

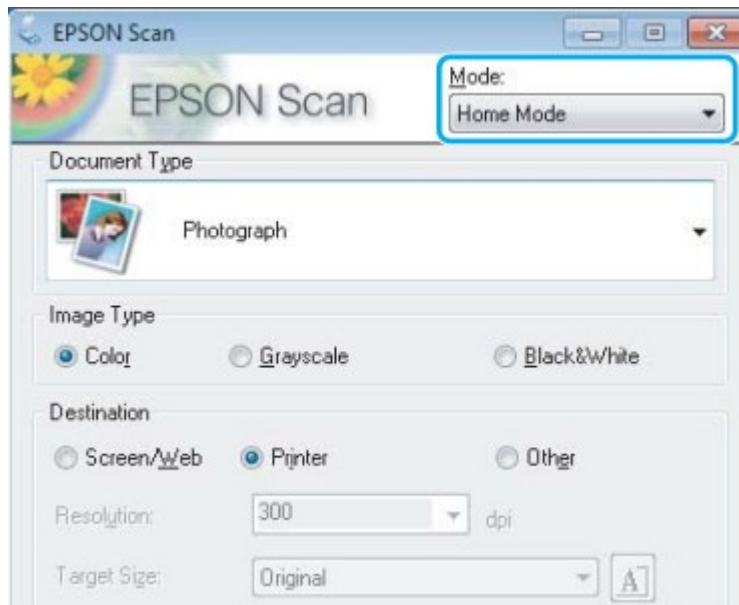
5. Muodosta BLUETOOTH-yhteys BLUETOOTH-laitteesta.

Kun BLUETOOTH-yhteys muodostetaan,  (BLUETOOTH) -ilmaisimen valo jää palamaan.



Kuva 4. Tehtävän eteneminen (Sony 2014)

Vaikka irralliset kuvat ovatkin aineiston yleisimpiä kuvia, suurin osa niistä kuitenkin sisältää korostuksia, tai niiden oheen on liitetty kuvateksti. Ne eivät siis ole niin irrallaan muusta tekstistä kuin kategorian nimi ehkä antaa ymmärtää. Kuva 5 on esimerkki korostetusta kuvakaappauksesta.



Kuva 5. Korostettu kuvakaappaus (Epson 2013, 56)

Samoin kuin van der Meij'n ym. (2003, 173) tuloksissa, irralliset kuvat olivat laskelmissani yleisimpiä. Kuten aiemmin totesin, he eivät kuitenkaan laskeneet kuvatekstillisiä tai korostettuja kuvia omiksi kategorioikseen, joten jää epäselväksi, kuinka irrallisia tekstistä kyseiset kuvat loppujen lopuksi ovat. Omassa aineistossani täysin irrallisia kuvia oli melko vähän: kaikkiaan 34 kuvasta niitä oli 8. Tämä osoittaa, että valtaosassa ohjeista kuvat on sidottu tekstiin jollakin tavalla. Alkuperäisen tutkimuksen tapaan kaksi muuta kategoriaa (ts. kuvat, jotka sisältävät nimettyjä osia ja kuvat jotka esittävät tapahtumien järjestyksen) olivat huomattavasti harvinaisempia. Osien nimeäminen ja järjestyksen esittäminen tukisi kuitenkin käyttäjän mentaalimallin muodostumista, sillä mentaalisen representaation rakentaminen on sitä nopeampaa mitä enemmän kuvien rakenne muistuttaa itse ohjelmiston tai laitteiston rakennetta ja toimintaa (van der Meij ym. 2003, 163; Ganier 2004, 21).

Van der Meij'n ym. (2003, 173) aineistossa ohjelmisto- ja laitteisto-ohjeissa oli lähes saman verran kuvia (30 ja 25), mutta omissa laskelmissani laitteistoja käsittelevät vaiheittaiset ohjeet sisälsivät selvästi enemmän kuvia kuin ohjelmisto-ohjeet. Toisin kuin alkuperäisessä tutkimuksessa, nimettyjä osia sisältäviä kuvia ei kuitenkaan ollut laitteisto-ohjeissa enemmän. Kuvien eriäviä lukumääriä saattaa selittää se, että laitteistojen ohjeet käsittelevät usein konkreettisia esineitä, jotka on helpompi esittää kuvallisessa muodossa kuin ohjelmistojen abstraktit konseptit (Mäkynen 2012, 73). Esimerkiksi valon syttymisen kaltainen järjestelmän reaktio voidaan havainnollistaa

valokuvana tai viivapiirroksena, mutta ”Tiedosto tallentuu” on ohjelman sisäinen toiminto, jota ei voi esittää kuvallisessa muodossa.

5.3.3 Vaiheet

Työvaiheet ovat suurin syy, miksi käyttöohjeita ylipäättään luetaan. Lukijan on löydettävä ne helposti, joten ne pitäisi jollakin tavalla erottaa muusta tekstistä. Van der Meij ym. (2003, 167; ks. luku 3.3.3) suosittelevat tähän numeroitujen listojen käyttöä, ja ne esiintyivätkin aineistossani useimmiten. Kuten taulukoista 17 ja 18 selviää, numeroituja listoja käytettiin 12 ohjelmisto- ja 14 laitteisto-ohjeessa. Jotkin ohjeet käyttivät useampaa tapaa, esimerkiksi numeroituja ohjeita ja numeroimattomia kappaleita, mistä syystä löydösten luvut ovat hieman suuremmat kuin tutkimieni vaiheittaisten ohjeiden kokonaismäärä. Numeroimattomat kappaleet eivät ole van der Meij’n ym. (2003) alkuperäinen kategoria. Van der Meij’n ym. alkuperäiset taulukot eivät myöskään sisältäneet johdattelevia määritteitä (ks. luku 3.3.3), mutta koska he laskivat niiden lukumäärän, esitän myös ne taulukossa 10.

Taulukko 10. Vaiheiden korostaminen

Yksikkö	Ohjeiden määrä (ohjelmistot)	Ohjeiden määrä (laitteistot)
Numeroidut vaiheet	12	15
Vaiheita ei ole eritelty muusta tekstistä	3	1
Vaiheet listattu numeroimattomiin kappaleisiin	4	2
Vaiheissa johdattelevia määreitä	13	12

Numeroimattomat kappaleet kyllä erottavat eri vaiheet toisistaan, mutta ensisilmäyksellä lukijalle saattaa jäädä kuva, ettei kyseessä ole vaiheittainen ohje. Esimerkki tällaisesta ei-toivotusta konventiosta on seuraava ohje:

Siirry näkymään, jossa näet kaikki sovellukset.

Valitse **Asetukset**.

Valitse Asetuksista kohta **Wi-Fi** (vanhemmissa Langattomat ja verkot).

Odoti vähän aikaa, että tabletti etsii verkkoja. [...] (DNA 2015, 15.)

Yhteensä 25 vaiheittaista ohjetta hyödynsi vaiheissaan johdattelevia määritteitä, jotka kertoivat kulloinkin käsiteltävän objektin sijainnin, kuten seuraavassa esimerkissä:

3. Napauta Gear VR -sovelluksen (Oculus) asennusnäytössä **ALOITA** (Samsung 2017, 38).

Suurin osa määritteistä kerrottiin ennen toimintoa, mikä vastaa sitä järjestystä, missä käyttäjä etenee tehtävässä: ennen toiminnon suorittamista hänen on luonnollisesti tiedettävä, missä hänen tarvitsemansa painike tms. sijaitsee. Määritteet ovat erityisen tärkeitä, jos objektin sijaintia ei ilmoiteta kuvallisesti.

Useimmat työvaiheet sisältävät jonkinlaisen käyttöliittymäelementin, kuten painikkeen tai valikon nimen, joka on useimmiten myös imperatiiviverbin objekti. Koska verbi-objekti-pari muodostaa työvaiheen ytimen, sitä pitäisi van der Meij'n ym. (2003, 174) ja van der Meij'n ja Gellevij'n (2004, 8) mukaan korostaa erikseen. Mielestäni kuitenkin riittää, että vain objekti on korostettu – jos lisäksi korostettaisiin verbejä, käyttöohjeiden ulkonäöstä saattaisi tulla sekava eikä lukijan olisi niin helppoa hahmottaa käyttöliittymäelementtejä muusta tekstistä. SFS-standardissa (SFS-EN 82079-1 2012, 60) ei kommentoida, pitäisikö ainoastaan kyseisiä elementtejä korostaa, mutta siinä painotetaan, ettei tärkeiden kohtien korostaminen saa haitata luettavuutta. Yksikään aineistoni vaiheittaisista ohjeista ei korostanut verbejä erikseen, mutta kaikki korostivat painikkeita ja muita elementtejä jollakin tavoin.

Jos mahdollista, käyttöohjeiden kirjoittajien tulisi tavoitella isomorfismia eli esittää käyttöliittymäelementit samassa muodossa kuin ne esitetään järjestelmässä kuvana (van der Meij ym. 2003, 178). Tämä tukee edellä mainittua linkitystä ja käyttäjän mentaalimallin syntymistä, sillä isomorfismi sitoo käyttöohjeen ja itse tuotteen toisiinsa. Elementtiä voidaan kuvailla ainoastaan sen nimellä, lisäämällä sen nimen jälkeen selvennys (-painike, -valikko jne.) tai kuvana (mp.). Laskin, kuinka moni aineistoni vaiheittainen ohje käytti mitäkin elementtien esitystapaa. Jotkin ohjeet käyttivät useampaa kuin yhtä tapaa, kuten InDesignin ohje (Adobe 2017):

Valitse taulukoksi muunnettava teksti tekstityökalulla **T**.

Esimerkissä painike sekä nimetään että esitetään saman näköisenä kuin mitä sen on itse ohjelmistossa. Esittelen laskelmieni tulokset taulukossa 11.

Taulukko 11. Käyttöliittymäelementtien korostaminen

Yksikkö	Ohjeiden määrä (ohjelmistot)	Ohjeiden määrä (laitteistot)
Visualisoitu painike	5	8
Painikkeen nimi ja teksti ”painike” tms. (esim. Enter-painike)	12	10
Vain painikkeen nimi	10	13

Visualisoidut painikkeet olivat suhteellisen harvinaisia, mikä oli jokseenkin yllättävää isomorfismin tärkeyttä ajatellen. Toisaalta monia ohjeissa mainittuja elementtejä ei välttämättä korosteta symboleilla itse järjestelmissä, jolloin niiden kuvittaminen ei toisi ohjeelle lisäarvoa. Visualisointi ei myöskään ollut kovin yleistä van der Meij’n ym. (2003, 179) aineistossa.

Van der Meij ym. (mts., 176) pitävät tärkeänä, että käyttöliittymäelementin nimeen lisätään selkeyden vuoksi -painike, -valikko tms. Nykykäyttäjät eivät tosin välttämättä tarvitse niin paljon ohjeistamista, vaan suurimmalle osalle lienee itsestään selvää esimerkiksi se, että ”Valitse Ctrl+Z” tarkoittaa ”Valitse Ctrl- ja Z-näppäimet”. Yhteensä 22 vaiheittaista ohjetta kuitenkin käytti jälkiliitteitä, mitä selittänevät suomen kielen vaatimukset enemmän kuin tarve selventää käyttäjälle, että kyseessä on käyttöliittymäelementti. Esimerkiksi AOC:n (2014, 32) monitoriohjeesta poimitut esimerkit toimisivat englanniksi ilman jälkiliitteitä (*select MENU, select AUTO*):

2. [...] [S]iirry toimintoon painamalla MENU-painiketta. [...]
5. Poistu painamalla AUTO-painiketta.

”Vain painikkeen nimi” -kategoria (johon laskin 23 ohjetta) ei kuitenkaan tarkoita, etteikö käyttöliittymäelementtiä olisi mitenkään korostettu muuhun tekstiin nähden. Elementtien nimet oli kirjoitettu isolla alkukirjaimella kaikissa ohjeissa, mukaan lukien tässä kategoriassa. Vaikka nämä

ohjeet eivät hyödyntäneetkään elementtien kuvia tai lisänneet niiden nimiin liitettä, suurin osa kyseisistä ohjeista lihavoivat nimet. PlayStation 4:n ohje oli ainoa, joka erotti nimet erikoismerkeillä:

Valitse ryhmänäytöstä [Pelin jakaminen] > [Anna ohjain vierailijalle] (Sony s. d.-b).

Toisin kuin alkuperäisessä tutkimuksessa, aineistossani ei ilmennyt suuria eroja ohjelmisto- ja laitteisto-ohjeiden välillä. Van der Meij'n ym. (mts., 179) saamissa tuloksissa 14 % ohjelmisto- ja 63 % laitteisto-ohjeista sisälsivät painikkeen nimen ja ”painike”-päätteen, kun taas ”Vain painikkeen” -nimi kategorian kohdalla vastaavat määrät olivat 77 % ja 21 %. Omissa laskelmissani käyttöliittymäelementtien esitystavat jakautuvat hieman tasaisemmin, mikä viittaa siihen, etteivät ohjelmistojen ja laitteistojen suomenkielisten ohjeiden konventiot eroa tämän aiheen suhteen merkittävästi, mutta aineistoni koon huomioon ottaen sen pohjalta ei voi tehdä yleistäviä johtopäätöksiä.

5.4 Ei-toivotut tilat

Ei-toivotut tilat ovat paitsi suoranaisia vaaratilanteita, myös lievempiä ongelmia, joita käyttäjää neuvotaan välttämään. Esittelen seuraavissa alaluvuissa, kuinka ei-toivotuista tiloista ilmoittaminen ilmeni analysoimissani ohjeissa.

5.4.1 Varoitukset

Varoituksia esiintyi vain kuuden vaiheittaisen ohjeen yhteydessä, joista kaksi oli ohjelmisto- ja neljä laitteisto-ohjetta. Varoituksia oli yhteensä 8. Esiintymiä on siis hyvin vähän, mutta toisaalta harva ohje sisälsi vaiheita, jotka olisivat tarvinneet oheensa varoituksen. Van der Meij'n ym. (2003, 159) esimerkkiä seuraten luokittelin uhkan kuvailuksi järjestelmän tilat ja seurauksiksi järjestelmän reaktiot (ks. ei-toivotut tilat luvussa 3.4 ja järjestelmän reaktiot 3.3.2). Näitä kahta oli kuitenkin usein vaikea erottaa toisistaan. Esimerkiksi seuraava esimerkki voisi sopia kumpaankin kategoriaan:

VAROITUS: Huono kalibrointi voi antaa vääriä arvoja testejä tehdessä! (HUR Labs 2015, 11.)

Huono kalibrointi on tulkittavissa järjestelmän tilaksi ja väärin arvojen antaminen järjestelmän reaktioksi. Luokittelin sen kuitenkin reaktioksi ja siten haitan seurauksen kuvailuksi, sillä huonon kalibroinnin voi tulkita myös käyttäjän toiminnoksi. Väärin arvojen antaminen puolestaan on selvästi järjestelmän reaktio.

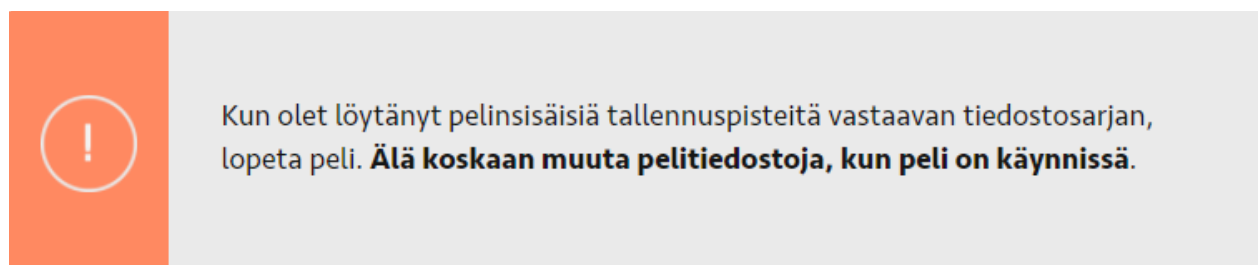
En laskenut muusta tekstistä selvästi erottuvaksi varoitukseksi sellaista varoitusta, joka oli mukana luettelomerkityssä listassa ja siten erillään leipätekstistä mutta joka ei erottunut luettelon muista osista. Tämänlaisia varoituksia oli kahdessa eri vaiheittaisessa ohjeessa yhteensä kolme. Taulukko 12 havainnollistaa, millaisista piirteistä aineistoni varoituksissa kokonaisuudessaan on.

Taulukko 12. Varoitukset

Yksikkö	Kpl (ohjelmistot)	Kpl (laitteistot)
Varoitus erottuu muusta tekstistä	2	2
Huomiosana	1	2
Uhka tai haitta kuvaillaan	-	2
Varoituksen huomiotta jättämisen seuraukset kuvaillaan	1	2
Lukijalle neuvotaan mitä hänen täytyy tehdä tai olla tekemättä	1	6


Kuten yllä oleva taulukko osoittaa, suurin osa löytämistäni varoituksista vain kertoo mitä lukijan ei pidä tehdä, mikä on mielestäni vain varoitusten minimivaatimus. Ne eivät siis seuraa van der Meij'n ym. (mts., 156) suosituksia eivätkä SFS-standardin (SFS-EN 82079-1 2012, 40; ks. luku 3.4.1) vaatimuksia. Lisäksi varoitusten pitäisi erottua muusta sisällöstä paremmin. Eri informaatiotyyppien liiallinen sulautuminen toisiinsa tekee käyttöohjeiden ulkoasusta vaikean hahmottaa nopealla silmäilyllä, mikä mielestäni ei anna ohjeista ammattimaista vaikutelmaa. Epäammattimainen ulkoasu puolestaan on yksi syistä, joka vähentää sen todennäköisyyttä, että käyttäjä jatkaa ohjeisiin tutustumista (Farkas 1999, 44; Novick & Ward 2016, 16; ks luku 2.2).

Kuva 6 on kuitenkin eräs esimerkki, joka sisälsi useimmat taulukon 12 yksiköistä:



Kuva 6. Varoitus (EA 2017)

Kuva on selvästi varoitus. Se on eristetty muusta tekstistä laatikolla, symboli kiinnittää lukijan huomion, ja lukijalle kerrotaan mitä hänen ei pidä tehdä, mutta toisaalta varoitus ei kuvaile sen noudattamatta jättämisen seurauksia tai ei-toivottua järjestelmätilaa. Kuva 7 on aineiston ainoa varoitus, joka täyttää lähes kaikki van der Meij'n ym. (2003, 156) kriteerit:

 **VAROITUS:**

- 160 merkin raja voi vaihdella maittain kielestä ja tekstiviestin koodauksesta riippuen.
- Jos tekstiviestiin lisätään kuva-, video- tai äänitiedosto, se muunnetaan automaattisesti multimediamiestiksi ja siitä laskutetaan tämän mukaan.

Kuva 7. Varoitus (LG 2015, 42)

Kuva 7 erottuu muusta tekstistä taustan ja symbolin avulla, siinä on huomiosana (*Varoitus*), ja se kuvailee mitä toimintoa käyttäjä haluaa mahdollisesti välttää (*Jos tekstiviestiin lisätään kuva-, video- tai äänitiedosto...*) ja miten järjestelmä reagoi mahdolliseen ei-toivottuun toimintoon (*...siitä laskutetaan tämän mukaan.*). Vaikka varoituksia oli aineistossa hyvin vähän, oli silti yllättävää, ettei niistä useammassa perusteltu, miksi käyttäjän tulisi toimia tietyllä tavalla, varsinkin kun seurausten mainitsematta jättäminen ei noudata muun muassa aiemmin lainaamaani SFS-standardia.

Seuraava esimerkki on melko tulkinnanvarainen, mutta laskin sen varoitukseksi, koska siitä on pääteltävissä, että maksimipainon ylittäminen vaurioittaa laitetta:

Huomioi että 200 kg on maksimipaino sensoria kohden (HUR Labs 2015, 11).

Lause esiintyi kuitenkin leipätekstissä, eikä sitä ollut mitenkään eroteltu muusta tekstistä. Lauseen aloittava imperatiivi ("huomioi") on periaatteessa laskettavissa huomiosanaksi, mutta en kuitenkaan luokitellut sitä sellaiseksi, koska se ei ollut erillään itse huomautuksesta.

Aineistosta erottui esimerkki, jonka laskin varoitukseksi sen sisällön vuoksi, vaikkei sitä ollut korostettu huomiosanalla tai symbolilla. Kyseinen kappale yhdistää haitan kuvailun ja tuotteen mainostamisen:

LowBlue-tila on helppo silmille ja lisää tuottavuutta. Tutkimukset ovat osoittaneet, että samoin kuin ultraviolettisäteily voi aiheuttaa vahinkoa silmille, LED-näyttöjen lyhyen aallonpituuden siniset säteet voivat vahingoittaa silmiä ja haitata näköä ajan mittaan. Hyvinvointia varten luotu Philipsin LowBlue-tila käyttää lykstä [*sic*] ohjelmistoteknologiaa, joka vähentää haitallista lyhyen aallonpituuden sinistä valoa. (Philips 2017, 10.)

Klassisen retoriikan (ks. luku 2.2) periaatteiden mukaisesti esimerkkikatkelma vetoaa käyttäjän järkeen, jotta hänet saataisiin noudattamaan valmistajan kehotusta eli käyttämään LowBlue-tilaa: jos käyttäjä ei huomioi valmistajan suositusta, hänen näkönsä saattaa heikentyä. Varoituksen ja tuotteen uskottavuutta pyritään lisäämään vetoamalla tutkimuksiin ja vakuuttamalla, että Philipsin älykäs teknologia on luotu hyvinvointia varten. Pelkkä ”tutkimukset” ilman tarkennuksia on kuitenkin hyvin suurpiirteinen käsite, joka saattaa herättää joidenkin käyttäjien epäilykset luottamuksen sijaan. Käyttöohjeiden kirjoittaja myös asettuu katkelmassa asiantuntijan asemaan, mikä liittyy luvussa 2.1 mainittuihin retorisiin rooleihin.

Löysin lisäksi muutamia kohtia, joita ei ollut merkitty symboleilla tai muuten korostettu, mutta jotka muistuttivat varoituksia. En laskenut esimerkiksi seuraavaa kohtaa varoitukseksi:

4. [...] Valitse **Hylkää**, jos et ole varma ladattavan tiedoston sisällöstä. (Google s. d.-b)

Lause ei noudata *see-think-use*-mallin suosituksia (ks. luku 3.4.1), sillä se ei erotu muusta tekstistä eikä perustele käyttäjälle, miksi ladattavien tiedostojen kanssa on oltava varovainen, mutta se kuitenkin suosittelee, ettei käyttäjän pitäisi ladata tiedostoja, joiden sisällöstä hän on epävarma. Java-ohje (Peltomäki 2014, 68) puolestaan sisältää seuraavan esimerkin, joka on varoituksen kaltainen mutta jota en laskenut varoitukseksi sen tulkinnanvaraisuuden vuoksi:

Yleensä kuitenkin luokan ominaisuudet suojataan ulkopuoliselta käytöltä käyttämällä niiden edessä *private*-avainsanaa.

Luokan ominaisuuksien käytön rajaaminen tekee koodin päivittämisestä helpompaa, mutta tätä ei kuitenkaan todeta kyseisessä vaiheittaisessa ohjeessa. Toisaalta päivityksen vaikeutuminen on pienehkö haitta, josta ilmoittaminen saattaisi kuulua ennemmin konseptuaalisen tiedon tai huomautusten kuin varoitusten piiriin, kun taas haitallisen tiedoston lataamisella saattaa olla hyvinkin vakavia seurauksia. On mahdollista, että Chromen käyttöohjeiden kirjoittajat ovat

olettaaneet, että kaikki nykykäyttäjät ovat perillä epämääräisten tiedostojen lataamisen mahdollisista vaaroista.

Kolmas epäselvä esimerkki on Nikonin (2016, 116) kameraohjeesta:

Kuvaus- ja [pimeänäkö]tilassa valotuksen korjaus nollautuu, kun toinen tila valitaan tai kamera kytketään pois päältä.

Tämäkin on tosin pieni haitta, jota en välttämättä luokittelisi varoitukseksi.

Kuten mainitsin luvussa 3.4.1, sekä SFS-standardi (SFS-EN 82079-1 2012, 40) että van der Meij ym. (2003, 155) suosittelevat, että varoitus sijoitetaan sen tehtävän yhteyteen, josta saattaa aiheutua haittaa tai vaaratilanne. Kolme analysoimaani varoitusta oli vaiheen yhteydessä, kolme vaiheittaisen ohjeen jälkeen ja kaksi ennen vaiheittaista ohjetta. Esimerkiksi kuva 8 oli sijoitettu vasta ohjeen jälkeen eikä tiedostojen liittämistä kertovaan vaiheeseen. Jos käyttäjä ei lue koko tekstiä ennen tehtävän suorittamista, on mahdollista, että päästessään varoitukseen asti hän on tietämättään jo lähettänyt tavallista tekstiviestiä kalliimman multimediaviestin.

Yksikään kokonainen ohjelmistokäyttöohje ei sisältänyt erillistä varoitusosiota, mutta kahdeksassa laitteistokäyttöohjeessa oli sellainen. Kolmessa käyttöohjeessa osio oli lopussa ja viidessä ohjeiden alussa. Tämä antaa viitteitä siitä, että van der Meij'n ja Gellevij'n (2004, 9) huomio yritysmaailman konventioista pitää yhä paikkansa (ks. luku 3.4.1): suositusten mukaan varoitusten pitäisi olla vaiheiden yhteydessä, mutta yrityksillä on tapana sijoittaa ne erillisiin osioihin. On kuitenkin jälleen hyvä huomioida, että kyse on hyvin pienestä otannasta.

5.4.2 Ongelmanratkaisu

Löysin aineistostani 4 vaiheittaista ohjelmisto-ohjetta ja 5 laitteisto-ohjetta, jotka sisälsivät ratkaisuja ongelmiin, joita käyttäjä saattaa kohdata tehtävää suorittaessaan. Toisin kuin Farkas (1999, 46) suosittelee, aineistoni vaiheittaiset ohjeet eivät siis useimmiten kerro käyttäjälle, millaisiin ei-toivottuihin tiloihin järjestelmä voi päätyä. On tietysti mahdollista, että jotkin järjestelmät itsessään antavat kattavia virheilmoituksia ja korjausehdotuksia, mikä vähentää tarvetta sisällyttää ne vaiheittaisten ohjeiden yhteyteen.

Windows 7 -ohje (Microsoft 2016) sisälsi vastauksen mahdolliseen ongelmaan vaihtoehdoisen menettelytavan muodossa:

Entä jos automaattisen toiston valintaikkuna ei tule näyttöön?

1. Avaa tietokone napsauttamalla **Käynnistä**-painiketta ja valitse **Tietokone**. [...]

Käyttäjää myös ohjattiin ongelman sattuessa muualle käyttöohjeeseen:

Jos haluamasi tulostimen vieressä näkyy kuvake, siinä saattaa olla ongelma, joka on ratkaistava. Katso [Kuvakkeet Tulostimen ponnahtusvalikossa](#). (Apple s. d.)

Tämä ei kuitenkaan ole täysin varma menetelmä, sillä yhdessä ohjeessa (Bose s. d.) tämänkaltainen linkki ei toiminut.

Ongelmanratkaisua koskevat tieto ei toisin sanoen ollut aineistossani yleistä, mutta sitä oli kuitenkin enemmän kuin van der Meij'n ym. (2003) aineistossa. Heidän tutkimastaan 104 vaiheittaisesta ohjeesta vain 11 sisälsi ratkaisuja mahdollisiin ongelmiin (van der Meij & Gellevij 2004, 12). Laskelmieni tulokset ovat siis jossain määrin myönteisemmät, mutta ongelmaratkaisuehdotusten soisi silti olevan yleisempiä muun muassa niiden syiden vuoksi, joita käsittelin luvussa: ongelmien kohtaaminen luonnollisesti estää käyttäjän etenemisen kohti hänen päämääräänsä, ja ne myös vaikuttavat haitallisesti hänen motivaatioonsa ja vähentävät hänen luottamustaan tuotetta ja sen valmistajaa kohtaan (mts., 11). Koska luottamuksen puute saattaa yllyttää käyttäjän jättämään käyttöohjeet kokonaan huomiotta (ks. luku 2.2), ohjeiden kirjoittajien tulisi pyrkiä ylläpitämään tätä luottamusta kaikin saatavilla olevin keinoin. Ongelmaratkaisuehdotusten sisällyttäminen vaiheittaisiin ohjeisiin on siihen suhteellisen yksinkertainen keino, varsinkin verkko-ohjeissa, joiden pituudessa ei tarvitse ottaa huomioon painatus- ja jakelukustannuksia. Muutamat tutkimani käyttöohjeet sisälsivät erillisen ongelmanratkaisuosion, mutta kuten varoituksistakin (van der Meij ym. 2003, 155; ks. luku 3.4.1), ongelmia koskevista ohjeista on eniten hyötyä siinä yhteydessä, missä käyttäjä tarvitsee niitä välittömästi.

6 Päätelmät

Edellisessä luvussa kuvailemani analyysin perusteella vaikuttaa siltä, että nelikantamalli sopii hyvin suomenkielisten vaiheittaisten ohjeiden yleiseen analysointiin ja niiden piirteiden erittelyyn. Toisin kuin ennen varsinaisen tutkimuksen aloittamista oletin, suomenkielisten ohjeiden rakenne ei juurikaan poikkea van der Meij'n ym. (2003) tutkimista ohjeista. Nelikantamallin osatekijät ja sitä kautta Newellin (1980) ja Farkasin (1999) käsittelemät aiheet (ks. luku 2) eivät esiintyneet aineistossani tietyssä vakiintuneessa järjestyksessä tai muodossa, vaan niiden sijainti ja ulkoasu vaihtelivat huomattavasti. Kulttuuri- tai kielikohtaisten erojen sijaan kyse lienee siis siitä, ettei vaiheittaisilla ohjeilla ole tiukan vakiintunutta rakennetta, vaan jokainen toimiala ja yritys noudattaa omia konventioitaan. Toisaalta näin on järkeväkin tehdä, sillä eri tuotealojen käyttöohjeilla on jossain määrin erilaiset vaatimukset. Käyttöohjeiden rakennetta yhtenäistää kuitenkin muun muassa lainsäädäntö.

Hypoteesini nelikantamallin muokkaamistarpeesta piti kuitenkin paikkansa. Ilmiselvin esimerkki tästä ovat otsikot (ks. luku 5.1.3), joita en voinut luokitella suoraan van der Meij'n ym. (2003, 143) taulukon pohjalta. Erot alkuperäisen tutkimuksen tuloksiin eivät kuitenkaan loppujen lopuksi olleet suuria. Valtaosa aineistoni otsikoista oli -minen-päätteisiä, ja 4. infinitiivi vastaa suunnilleen englannin gerundia eli *-ing*-päätettä, joka oli van der Meij'n ym. (mp.) laskelmien perusteella yleisin otsikkomuoto. Otsikoissa oli silti yllättävänkin paljon vaihtelua, ja uskon, että koostaan huolimatta aineisto antoi suhteellisen kattavan kuvan suomenkielisten vaiheittaisten ohjeiden otsikoiden variaatiosta.

Kuten johdannossa huomautin, suomenkielisiä käyttöohjeita ja nimenomaan vaiheittaisia ohjeita ei ennen ole arvioitu nelikantamallin avulla, joten suomennettu malli tarjoaa uudenlaisen viitekehyksen vaiheittaisen tiedon piirteiden kartoittamiseen ja analysointiin. Samalla selvisi, että alun perin 1990-luvulla julkaistujen ohjeiden pohjalta laadittu malli soveltuu myös 2010-luvun ohjeisiin. Koska itse työvaiheet ovat ohjeiden tärkein osuus (Farkas 1999, 46; ks. luvut 2.3 ja 3.3.3), niiden kriittiseen tarkasteluun tulisi mielestäni kiinnittää erityistä huomiota. Nelikantamalli sisältää keinot sekä vaiheiden sisällön (ks. luku 5.1.2) että kyseisen sisällön ulkoasun (ks. luku 5.3.3) arviointiin. Aineistossani esiintyneet vaiheet vastasivat osittain van der Meij'n ym. (2003) suosituksia, joskin esimerkiksi painikkeiden ja käyttöliittymäelementtien visualisointi jätti toivomisen varaa. Toinen tärkeä arviointikohde ovat ongelmanratkaisutieto ja etenkin varoitukset.

Jälkimmäiset olivat aineistossani yllättävän pelkistettyjä, mikä ei nelikantamallin suositusten lisäksi noudata standardeja (van der Meij & Gellevij 2004, 10; SFS-EN 82079-1 2012, 40).

Analyysiani eniten hidastavaksi tekijäksi ei osoittautunut nelikantamallin muokkaaminen vaan se, että malli oli melko usein yllättävän tulkinnanvarainen tai suurpiirteinen. Laskettavien yksiköiden tarkka määrittely olisi ollut huomattavasti vaikeampaa ilman luvun 2 teoriaan tukeutumista. Toisenlainen epäselväksi jäänyt seikka oli keskihajonnan sisällyttäminen vaiheiden laskennassa (van der Meij ym. 2003, 140; ks. luku 5.1.2). Sen laskeminen on mahdollista automatisoida, joten sen lisääminen taulukkoon 8 ei olisi tuonut lisävaivaa, mutta jätin sen kuitenkin pois, koska se ei olisi tuonut lisäarvoa näin pienen aineiston analyysiin. Mielestäni se ei olisi ollut pakollinen alkuperäisessä tutkimuksessakaan, mutta toisaalta en tunne kvantitatiivista tutkimusta ja sen vaatimuksia kovin hyvin. Lisäksi haasteita tuotti käsitteiden samankaltaisuus. Termien tarkempi tutkiminen ja käsittekartan laatiminen ennen varsinaisen työn aloittamista olisi auttanut myös niiden suomentamisessa. Jossain määrin kyse oli kuitenkin siitä, että alkuperäisessä tutkimuksessa käytettiin samoista asioista paikoitellen samoja termejä, kuten *step* ja *action step* sekä *approach* ja *method*.

Tässä tutkielmassa tekemäni analyysi olisi mahdollista toistaa samankokoisella aineistolla, mutta luotettavampien tulosten saaminen edellyttäisi laajempaa aineistoa. Vaikka olen tehnyt tulosteni pohjalta tarkempia päätelmiä luvussa 5, olen samalla huomauttanut, että näin pienestä aineistosta ei voi tehdä yleispäteviä tulkintoja suomenkielisistä vaiheittaisista ohjeista. Tässä aineistossa on kuitenkin suuntaa-antavia viitteitä siitä, että varoitusten tulisi olla nykyistä yksityiskohtaisempia ja mahdolliset ongelmat ratkaisuneen pitäisi kuvailla vaiheittaisten ohjeiden yhteydessä eikä erillisessä osiossa. Positiivinen huomio taas oli se, että ohjeiden pituus pysyi enimmäkseen suositusten rajoissa (ks. luku 3.1). Tutkimissani ohjelmisto-ohjeissa oli mukana kuvallista palautetta varsin vähän verrattuna van der Meij'n ym. (2003) aineistoon, mutta kuten arvelin luvussa 5.3.2, eron saattaa olla syynä se, että nykykäyttäjät ovat harjaantuneempia ohjelmistojen käytössä kuin 1990-luvun käyttäjät eivätkä siksi tarvitse yhtä paljon kuvakaappauksia. Syynä saattavat olla myös tiheään tahtiin julkaistavat ohjelmistopäivitykset, joista monet muuttavat ohjelmien ulkonäköä jossain määrin, vaikka varsinaiset toiminnot pysyisivät samana. Tietotekniikan kehittyminen saattaa muutenkin vaikuttaa tulosten eroavaisuuksiin. Kuten totesin luvussa 5.2, esimerkiksi viittaukset muualle ovat luultavasti yleistyneet siksi, että käyttöohjeet

julkaistaan nykyään usein sähköisesti. Pääasiassa van der Meij'n ym. (2003) ja myös Farkasin (1999) käsitykset vaiheittaisten ohjeiden ominaispiirteistä kuitenkin pitävät paikkansa nykypäivänäkin.

Tulosten luotettavuutta ja objektiivisuutta olisi suuremman aineiston lisäksi lisännyt tutkijapari tai -ryhmä. Monet laskemistani yksiköistä olivat subjektiivisia – esimerkiksi varoituksen pitäminen selkeästi erottuvana riippuu lukijan odotuksista, tottumuksista ja näkökyvystä – ja muiden kanssa keskusteleminen olisi selkeyttänyt muun muassa ennakkojäsentäjien määrittelyä. Olisi kiinnostavaa nähdä, kuinka paljon jonkun toisen laskelmat eroaisivat omista tuloksistani ja missä määrin nämä erot johtuisivat tulkintaeroista. Toinen objektiivisuuteen vaikuttava tekijä on se, ettei satunnaiseksikaan tarkoitettu aineiston kerääminen voi olla täysin satunnaista, vaan otannan kerääminen on jossain määrin aina harkinnanvaraista (ks. esim. Tuomi & Sarajärvi 2018, 74). Jos aikaa olisi ollut enemmän, olisin yrittänyt tehdä prosessista satunnaisemman vaikkapa koodaamalla ohjelman, joka olisi valinnut käyttöohjeita ja niistä yksittäisiä vaiheittaisia ohjeita sattumanvaraisesti (ks. esim. Thompson 2012, Olteanu 2017 & Paruchuri 2016).

Uskoakseni termien suomennoksista ja laskentaperiaatteiden täsmentämisestä on hyötyä teknisen viestinnän tutkijoille, jotka haluavat hyödyntää nelikantamallia suomenkielisten ohjeiden arvioinnissa. Mitä nelikantamallin yksiköiden määrittelyyn tulee, hollannintaitoiset tutkijat voisivat ottaa yhteyttä mallin kehittäjiin ja pyytää nähtäväkseen koodikirjaa (ks. luku 4), joka mahdollisesti sisälsi yksiköiden tarkemmat määritelmät. Jos kirja on vielä näin pitkän ajan jälkeen saatavilla, se todennäköisesti selventäisi monia epävarmuustekijöitä, joita kuvailin luvussa 5.

Eräs muutosehdotukseni tulevaa tutkimusta ajatellen liittyy lukuun 5.1.1 eli siihen, millä keinoin käyttäjä pääsee tavoitteeseensa. Van der Meij'n ym. (2003) mallin mukaisesti laskin, kuinka moni vaiheittainen ohje sisälsi perustelut sille, miksi käyttäjän pitäisi valita tietty työvaihe tai menettelytapa useammasta vaihtoehdosta. Toisenlaisen näkökulman tähän tarjoaisi vaihtoehtoisten operaattoreiden ja menettelytapojen laskeminen yksitellen ja sen kirjaaminen, onko näitä vaihtoehtoja perusteltu. Näin aineistosta saisi tämän aiheen suhteen täsmällisemmän kuvan. Myös ongelmanratkaisuohteita voisi tutkia tarkemmin luvun 3.4.2 suositusten eli *detect-diagnose-correct*-mallin pohjalta, ja toimintoja ja reaktioita tutkittaessa on mahdollista määritellä, sulkevatko johdattelevat määreet ja kuvat toisensa pois kenties sillä perusteella, että ne käsittelevät usein samoja asioita, vai käytetäänkö niitä toistensa tukena.

Häivyttäminen on myös eräs jatkotutkimuksen aihe. Kuten sekä alkuperäisessä artikkelissa (van der Meij ym. 2003, 147–148) että luvussa 5.2 todetaan, ohjeiden yksityiskohtien vähittäisen häivyttämisen tutkiminen ei ole mahdollista yksittäisten ohjeiden perusteella, mutta sen voisi toteuttaa tutkimuksessa, joka sisältää kokonaisia käyttöohjeita. Niin laajan tutkimuksen toteuttaminen vaatisi kuitenkin runsaasti aikaa tai kokonaisen tutkijaryhmän, ellei kyseessä sitten olisi pieneen aineistoon syvällisesti paneutuva tapaustutkimus. Kokonaisia käyttöohjeita tutkittaessa olisi lisäksi mahdollista selvittää, ovatko pää- ja alaotsikot kautta ohjeen samanlaisessa muodossa, esimerkiksi pääotsikot aina 4. infinitiivissä ja alaotsikot aina perusmuodossa.

7 Lähteet

7.1 Aineisto

- Adobe 2016. Kuvan luominen. Saatavilla: https://helpx.adobe.com/fi/photoshop/using/creating-opening-importing-images.html#create_an_image.
- Adobe 2017. Taulukkojen luominen: Taulukon luominen alusta asti. Saatavilla: <https://helpx.adobe.com/fi/indesign/using/creating-tables.html>.
- AOC 2014. LCD-monitorin käyttöopas. Saatavilla: https://aoc-pim.s3.amazonaws.com/Public/AOC/Product%20Data%20and%20Pictures/77ID/U3477PQU/Manual/U3477PQU_Finnish.pdf.
- Apple s. d. Dokumenttien tulostaminen. Mac-ohje. Saatavilla: <https://support.microsoft.com/fi-fi/help/17449/windows-7-import-pictures>.
- Bose s. d. Äänikohotteiden kielen valinta/vaihtaminen: Kielen valinta ensimmäisen asennuksen aikana. Saatavilla: <https://www.bose.fi/fi-fi/support/article/selecting-and-changing-a-language-sl-revolve.html>.
- Corel Corporation 2015. Pinnacle Studio 19 – Käyttöopas. Saatavilla: <http://help.pinnaclesys.com/pinnacle/v19/main/fi/user-guide/PinnacleStudio.pdf>.
- DNA 2015. DNA Matkamokkula 4G WLAN E5377. Saatavilla: https://www.dna.fi/documents/753910/853450/DNA_MokkulaE5377_pikaopas_fin_al.pdf/2bc0ce89-8efa-4f55-71f3-86f33a3ae2fe.
- EA 2017. Aiemmin tallennetun pelin palauttaminen The Sims 4 -pelissä. Saatavilla: <https://help.ea.com/fi-fi/help/the-sims/the-sims-4/how-to-restore-a-previous-saved-game-in-the-sims-4/>.
- Epson 2013. Käyttöopas. Saatavilla: <ftp://download.epson-europe.com/pub/download/3768/epson376830eu.pdf>.
- Evernote s. d. Muistikirjojen järjestäminen pinoihin: Windows. Saatavilla: <https://help.evernote.com/hc/fi/articles/208314158>.
- Google s. d. Sovellusten lataaminen Android-laitteeseen: Sovellusten lataaminen Google Playsta. Saatavilla: https://support.google.com/android/answer/7391672?hl=fi&ref_topic=7311596.
- Google s. d.-b. Tiedoston lataaminen: Tietokone. Saatavilla: https://support.google.com/chrome/answer/95759?hl=fi&ref_topic=7439636.
- Hakkarainen, Anssi 2011. *Oracle-tietokannan tehokas hallinta*. Helsinki: Readme.fi.
- HP 2015. HP ElitePad 1000 G2 Rugged -tabletti – Käyttöopas. Saatavilla: <http://h10032.www1.hp.com/ctg/Manual/c04898311>.
- HUR Labs 2015. iBalance SmartTouch- käyttöohje. Saatavilla: http://sd7.staattinen.fi/sites/www.hurlabs.com/files/files/ibalance_smarttouch_manual.pdf.
- Korpela, Jukka K. 2014. *HTML5-käsikirja*. Jyväskylä: Docendo.

- Lenovo 2015. Käyttöopas – Lenovo E31-80. Saatavilla:
https://download.lenovo.com/pccbbs/mobiles_pdf/e31-80_ug_fi.pdf?linkTrack=PSP:ProductInfo:UserGuide.
- LG 2015. Käyttöopas: Northern Europe. Saatavilla: <http://www.lg.com/fi/tuki/tuki-mobile/lg-G3-s-D722>.
- Microsoft s. d. Kuinka voin soittaa puhelun Skypeä? Saatavilla:
<https://support.skype.com/fi/faq/fa10613/kuinka-voin-soittaa-puhelun-skypeessa>.
- Micosoft s. d.-b. Muutosten jäljittäminen Wordissa: 2016, 2013: Tiedoston muutosten (muokkauksien) näyttäminen. Saatavilla: <https://support.office.com/fi-FI/article/muutosten-j%C3%A4ljitt%C3%A4minen-wordissa-197ba630-0f5f-4a8e-9a77-3712475e806a>.
- Microsoft s. d.-c. Pikakäännöksen kalibroiminen Microsoft-hiirellä. Saatavilla:
<https://www.microsoft.com/accessories/fi-fi/support/how-to/mouse/calibrate-qt-mouse>.
- Microsoft 2016. Kuvien tuominen Windows 7:ään. Saatavilla: <https://support.microsoft.com/fi-fi/help/17449/windows-7-import-pictures>.
- Microsoft 2016b. Käyttäjätilin luominen Windowsissa: Windows 8.1. Saatavilla:
<https://support.microsoft.com/fi-fi/help/17449/windows-7-import-pictures>.
- Netgear 2015. Pikaopas. Saatavilla:
http://www.downloads.netgear.com/files/GDC/R6220/R6220_QSG_FN_13Feb2015.pdf.
- Nikon 2016. Digitaalikamera D3400 – Käyttöopas. Saatavilla:
[http://download.nikonimglib.com/archive3/UcLKf00v5GCm0309A9944ZLNKC695/D3400RM_\(Fi\)02.pdf](http://download.nikonimglib.com/archive3/UcLKf00v5GCm0309A9944ZLNKC695/D3400RM_(Fi)02.pdf).
- Peltomäki, Juha 2014. *Pieni Java 8* -kirja. Helsinki: Books on Demand.
- Philips 2017. Käyttöopas. Saatavilla:
https://www.download.p4c.philips.com/files/3/356m6qjab_75/356m6qjab_75_dfu_fin.pdf.
- Polar Electro 2016. Polar A300. Käyttöohje:
https://support.polar.com/e_manuals/A300/Polar_A300_user_manual_Suomi/manual.pdf.
- Procountor s. d. Ostojen raportointi. Saatavilla:
<http://support.procountor.com/fi/raportointi/ostojen-raportointi.html>.
- Samsung 2016. Käyttöopas. Saatavilla:
[http://files.customersaas.com/files/Samsung_A310F_Galaxy_A3_\(2016\)_K%C3%A4ytt%C3%B6hje.pdf](http://files.customersaas.com/files/Samsung_A310F_Galaxy_A3_(2016)_K%C3%A4ytt%C3%B6hje.pdf).
- Samsung 2017. Käyttöopas. Saatavilla:
http://downloadcenter.samsung.com/content/UM/201705/20170523083346485/SM-R324_UM_Open_Fin_Rev.1.0_170511_Screen.pdf.
- Sonos s. d. Sonos PLAY:1. [Kopio saatavilla.]

- Sony s. d. Laiteparin muodostaminen ja yhdistäminen BLUETOOTH-laitteeseen. Saatavilla: <http://helpguide.sony.net/speaker/srs-x5/v1/fi/contents/TP0000379166.html>.
- Sony s. d.-b. Pelaamisen jakaminen. Saatavilla: <http://manuals.playstation.net/document/fi/ps4/share/gameplay.html>.
- Sony s. d.-c. Yhteystietojen lisääminen ja muokkaaminen: Yhteystiedon lisääminen. Saatavilla: <https://support.sonymobile.com/fi/xperiaz5/userguide/adding-and-editing-contacts/>.
- Spotify s. d. Kuuntelu offline-tilassa: Työpöytäsovellus. Saatavilla: https://support.spotify.com/fi/listen_everywhere/on_phone_tablet_desktop/listen-offline/.

7.2 Kirjallisuus

- Arthur, Sarah C.M. 2013. Digital Literacy in a Global Context. Teoksessa Serap Kurbanoglu, Esther Grassian, Diane Mizrachi, Ralph Catts & Sonja Špiranec (toim.), *Worldwide Commonalities in Information Literacy Research and Practice*. Berlin: Springer. 141–147.
- Blackburn, Simon 2016. *The Oxford Dictionary of Philosophy*. Oxford: Oxford University Press. 3. painos.
- Card, Stuart K., Thomas P. Moran & Allen Newell 1983. *The Psychology of Human-Computer Interaction*. Mahwah: Lawrence Erlbaum Associates.
- Chandler, Daniel & Rod Munday 2016. *A Dictionary of Media and Communication*. Oxford: Oxford University Press. 2. painos.
- Coney, Mary B. & Carl S. Chatfield 1996. Rethinking the Author-Reader Relationship in Computer Documentation. *ACM SIGDOC Asterisk Journal of Computer Documentation* 20:2. 23–29. DOI: 10.1145/381815.381826. [Luettu 17.1.2018.]
- Dell’Olio, Jeaninen M. & Tony Donk 2007. *Models of Teaching: Connecting Student Learning with Standards*. Thousand Oaks: SAGE Publications.
- Farkas, David K. 1999. The Logical and Rhetorical Construction of Procedural Discourse. *Technical Communication* 46:1. 42–54. Saatavilla: <http://www.old-classes.design4complexity.com/6715-F15/readings/Farkas.pdf>. [Luettu 4.1.2018.]
- Ganier, Franck 2004. Factors Affecting the Processing of Procedural Instructions: Implications for Document Design. *IEEE Transactions on Professional Communication* 47:1. 15–26. DOI: 10.1109/TPC.2004.824289. [Luettu 16.4.2018.]
- Honkala, Iiro 2015. *Kombinatoriikka*. Luentomoniste. Matematiikan ja tilastotieteen laitos, Luonnontieteiden ja tekniikan tiedekunta, Turun yliopisto. Saatavilla: <https://www.utu.fi/fi/yksikot/sci/yksikot/mattil/opiskelu/kurssit/Sivut/Kombinatoriikka.aspx>. [Luettu 28.2.2018.]
- Huawei s. d. *Honor 8 Lite Käyttöopas*. Saatavilla: <https://consumer.huawei.com/fi/support/phones/honor-8-lite/>. [Luettu 25.1.2018.]
- Jayaprakash, Sajitha 2007. *Technical Writing*. Mumbai: Global Media.

- Kilkki, Kalevi 2013. Ajatuksia todellisuudesta. *Tieteessä tapahtuu* 31:6. 50–52. Saatavilla: <https://journal.fi/tt/article/view/9008/6555>. [Luettu 26.12.2017.]
- KvantiMOTV 2004. Tilastollinen päättely. Saatavilla: <http://www.fsd.uta.fi/menetelmaopetus/paattely/paattely.html>. [Luettu 16.4.2018.]
- Lidwell, William, Kimberly Elam, Jill Butler & Kritina Holden 2010. *Universal Principles of Design: 125 Ways to Enhance Usability, Influence Perception, Increase Appeal, Make Better Design Decisions, and Teach Through Design*. 2. painos. Beverly: Rockport Publishers.
- Leutner, Detlev 2000. Double-fading support – a training approach to complex software systems. *Journal of Computer Assisted Learning* 16:4. 347–357. DOI: 10.1046/j.1365-2729.2000.00147.x. [Luettu 6.4.2018.]
- Mäkynen, Aliisa 2012. A Picture is Worth a Thousand Words - or Is It? The Interplay of Text and Images in Technical Documents. Tampereen yliopisto, pro gradu -tutkielma. Saatavilla: <http://urn.fi/urn:nbn:fi:uta-1-22522>. [Luettu 10.4.2018.]
- Newell, Allen 1980 [2013]. Reasoning, problem solving, and decision processes: The problem space as a fundamental category. Teoksessa Raymond S. Nickerson (toim.), *Attention and Performance VIII*. London & New York: Routledge. 693–718.
- Novick, David G. & Karen Ward 2006. Why don't people read the manual? *Proceedings of the 24th annual ACM international conference on design of communication*. 11–18. DOI: 10.1145/1166324.1166329. [Luettu 1.2.2018.]
- Musset, Pauline 2015. *Building Skills For All: A Review of Finland*. OECD Skills Studies. Saatavilla: <http://www.oecd.org/education/skills-beyond-school/Building-Skills-For-All-A-Review-of-Finland.pdf>. [Luettu 28.2.2018.]
- O'Donnell, Angela M., Johnmarshall Reeve & Jeffrey K. Smith 2011. *Educational Psychology: Reflection for Action*. 3. painos. Hoboken: John Wiley & Sons, Inc.
- OECD 2016. *Skills Matter: Further Results from the Survey of Adult Skills*. OECD Skills Studies. Paris: OECD Publishing. Saatavilla: http://www.oecd.org/skills/piaac/Skills_Matter_Further_Results_from_the_Survey_of_Adult_Skills.pdf. [Luettu 28.2.2018.]
- Olteanu, Alex 2017. *Web Scraping with Python and BeautifulSoup*. Dataquest. Saatavilla: <https://www.dataquest.io/blog/web-scraping-beautifulsoup/>. [Luettu 28.4.2018.]
- Paruchuri, Vik 2016. *Python Web Scraping Tutorial using BeautifulSoup*. Dataquest. Saatavilla: <https://www.dataquest.io/blog/web-scraping-tutorial-python/>. [Luettu 28.4.2018.]
- Pietikäinen, Sari & Anne Mäntynen 2009. *Kurssi kohti diskurssia*. Tampere: Vastapaino.
- Pohjola, Tytti 2007. *Miltä tieto näyttää? Informaatiotyypit ja niiden visuaalinen asu käyttöoppaissa*. Tampereen yliopisto, pro gradu-tutkielma. Saatavilla: <http://urn.fi/urn:nbn:fi:uta-1-17172>. [Luettu 28.2.2018.]
- Skyttner, Lars 2005. *General Systems Theory: Problems, Perspectives, Practice*. 2 painos. Hackensack: World Scientific.

- SFS-EN 82079-1 2012. *Käyttöohjeiden laatiminen. Jäsentäminen, sisältö ja esittäminen. Osa 1: Yleiset periaatteet ja yksityiskohtaiset vaatimukset*. Helsinki: Suomen Standardoimisliitto SFS ry.
- Thompson, Steven K. 2012. *Sampling*. Hoboken: John Wiley & Sons, Inc.
- Tukes 2012. Tuotteiden käyttöohjeet ja turvallista käyttöä koskevat merkinnät. Saatavilla: http://www.tukes.fi/Tiedostot/julkaisut/Tuotteiden_kaytto-ohjeet_opas.pdf. [Luettu 16.4.2018.]
- Tuomi, Jouni & Anneli Sarajärvi 2018. *Laadullinen tutkimus ja sisällönanalyysi*. Uudistettu laitos. Helsinki: Tammi.
- van der Meij, Hans & John M. Carroll 1995. Principles and Heuristics for Designing Minimalist Instruction. *Technical Communication* 42:2. 243–261. [Luettu 3.1.2018.]
- van der Meij, Hans & Mark Gellevij 2004. The Four Components of a Procedure. *IEEE Transactions on Professional Communication* 47:1. DOI: 10.1109/TPC.2004.824292. [Luettu 26.11.2017.]
- van der Meij, Hans, Peter Blijleven & Leanne Jansen 2003. What Makes Up A Procedure? Teoksessa Michael J. Albers & Beth Mazur (toim.), *Content and Complexity: Information Design in Technical Communication*. Mahwah, New Jersey: Lawrence Erlbaum Associates. 129–186.

Liite 1

Valmistaja	Tuotteen nimi	Tuotteen tyyppi
Adobe	Photoshop	Kuvankäsittelyohjelma
Adobe	InDesign	Julkaisuohjelma
AOC	LCD-monitori	Näyttö
Apple	Mac	Käyttöjärjestelmä
Bose	SoundLink Revolve Bluetooth®-kaiutin	Kaiutin
Corel Corporation	Pinnacle Studio 19	Videonmuokkausohjelma
DNA	Matkamokkula 4G WLAN E5377	Mobiiliyhteyden reititin
EA	The Sims 4	Videopeli
Eclipse Foundation	Java 8	Ohjelmointikieli
Epson	XP-sarjat	Tulostin
Evernote	Evernote	Muistiinpanosovellus
Finago	Proucountor	Taloushallintaohjelmisto
Google	Android	Käyttöjärjestelmä
Google	Chrome	Selain
HP	HP ElitePad 1000 G2 Rugged	Tabletti
HUR Labs	iBalance SmartTouch	Tasapainotestiohjelmisto
Lenovo	Lenovo E31-80	Kannettava tietokone
LG	LG G3 s D722	Matkapuhelin
Microsoft	Skype	Viestintäohjelma
Microsoft	Microsoft Word	Tekstinkäsittelyohjelma
Microsoft	Microsoft-hiiret	Hiiri

Microsoft	Windows 7	Käyttöjärjestelmä
Microsoft	Windows 8.1	Käyttöjärjestelmä
Netgear	AC1200 Smart WiFi -reititin	Reititin
Nikon	Digitaalikamera D3400	Kamera
Oracle Corporation	Oracle	Tietokantojen hallintajärjestelmä
Philips	Philips-näytöt	Näyttö
Polar Electro	Polar A300	Aktiivisuusranneke
Samsung	Samsung Galaxy A3	Matkapuhelin
Samsung	Samsung Gear VR	Virtuaalitodellisuuslasit
Sonos	Sonos PLAY:1	Kaiutin
Sony	PlayStation 4	Pelikonsoli
Sony	SRS-X5	Bluetooth-kaiutin
Sony	Xperia Z5	Matkapuhelin
Spotify AB	Spotify	Musiikin suoratoistopalvelu
W3C	HTML5	Merkintäkieli

English Summary

The Components of Procedures in Finnish User Manuals

Introduction

This thesis focuses on manuals, which is a significant area of study in the field of technical communication. I focus on procedures, which is one of the most important information types in manuals due to their significance to the user. One model for studying the characteristics of procedures is the Four Components Model created by van der Meij, Blijleven and Jansen (2003), and this study is the first one that analyzes Finnish, modern procedures using that model.

As its name suggests, the Four Components Model categorizes the content of procedures into four components: goals, prerequisites, actions and reactions, and unwanted states (van der Meij et al. 2003, 130). Goals and possible subgoals are states that the user wants to reach, and they are usually described in procedure titles (ibid., 133–134). Starting and completion of tasks have prerequisites (ibid., 141) – for example, the use of a program requires the correct operating system. The user strives towards their goals by doing actions, and these actions cause the system or product to react (ibid., 160). Unwanted states are hazards that the user is warned of and that they might be able to fix (ibid., 150–151; van der Meij & Gellevij 2004, 11). In addition to the Four Components Model, the theoretical background of this thesis is based on Farkas's (1999) seminal work on procedural discourse.

Following the foundation laid in the above paragraphs, my research questions are:

- What kind of structure do Finnish procedures have?
- Can Finnish procedures be analyzed with the Four Components Model?

Due to the different languages and different writing conventions at play, I presume that the model will have to be adapted into the context of Finnish procedures of the 2010s. The goal of this study is to produce a fresh framework for future analysis of procedures written in Finnish.

Van der Meij et al. (2003) studied 104 procedures in total, but due to the limited scope of this thesis, my material consists of 36 procedures. I analyze these procedures via theory-based content

analysis, which means that I categorize and quantify my data according to an existing framework and test this framework in a new context (Tuomi & Sarajärvi 2018, 94–95) rather than create classifications based on the data itself.

Theoretical Framework

The work of van der Meij et al. (2003) is based on an article by Farkas (1999), who in turn draws inspiration from Newell (1980). Farkas's article focuses on procedural discourse, which he calls "how-to communication". In other words, it is discourse whose focus is relaying information on how to complete tasks. Farkas bases his view on procedural discourse on two theories: rhetoric – particularly classical rhetoric – and human problem-solving.

When it comes to problem-solving, *a problem space* (Newell 1990) is a central concept from which Farkas (1999) draws inspiration. Unlike its name might suggest, a problem space is not a physical space but rather a mental construct that may however drive a person to conduct actions in the physical world. In addition to system states, a problem space consists of operators, which are units of human behavior that change the system states. The interplay of operators and states forms a path that starts from an initial state and ends up in a goal state. The first one is a state where the system is before the user engages in the task at hand, while the latter is the user's goal (Newell, 1980, 4–5, 7).

Farkas (1999, 42–43) bases the following system states and actions in Newell's states and operators. A desired state is a state that the user wishes to realize. A prerequisite state is a state that the system must be in before the user can start the actual task. An interim state leads towards the user's final goal, while an unwanted state is a state that the user wants to avoid, such as system malfunction. These states are changed by human actions and system actions. The user takes actions to move towards their goal, and the system responds with actions of its own.

Procedural discourse also includes rhetorical elements, especially from classical rhetoric, which aims to gain its recipient's trust and persuade them by appealing to their reason along with their emotions (Chandler & Munday 2016; Blackburn 2016). The user of a manual will trust it more if its contents are tailored to their needs and the context in which they use it. The contents of a manual must also appear believable, or the user might dismiss it and attempt tasks on their own (Farkas

1999, 43), which can have undesirable results. Farkas also mentions rhetorical roles that the writer of a manual builds for themselves and their audience (Coney & Chatfield 1996, 24–25). Farkas's (1999) work focuses especially on streamlined-step procedures that later serve as a base for the Four Components Model (van der Meij et al. 2003). According to Farkas (1999, 46–49), the elements of a streamlined-step procedures are action steps, a title, a conceptual element, a subheading, and notes. A procedure might include all these elements or only some of them, but steps are naturally mandatory. They describe what actions the user must take and how the system reacts to these actions. A title describes the state that the user wants to reach, while a conceptual element might elaborate on the title and inform the user of any possible prerequisites and unwanted states. Subheadings separate closely related procedures from one another, and notes include information that should be outside the action steps, either because they are not mandatory, like extra tips, or because the user needs to be able to tell them apart from the rest of the text, such as warnings.

The Four Components Model by van der Meij et al. (2003) is based on the streamlined-step procedures described above. The model divides the content of procedures into four components: goals, prerequisites, actions and reactions, and unwanted states. **Goals** and possible subgoals describe which system states the user wants to achieve. They are often synonymous with the title of a procedure, but a conceptual element might elaborate on them (Farkas 1999, 46). Long procedures can be divided into subgoals (van der Meij & Gellevij 2004, 6). If the same task can be completed using different methods, a user must be advised on the selection of a method that is most appropriate for their needs (van der Meij et al. 2003, 133–134). One way to design the steps of a task and to predict user behavior is GOMS analysis. The analysis consists of user goals, the method that they use to complete the task at hand, selection rules for choosing one method over the other, and user actions (van der Meij et al. 2003, 134).

Prerequisites are conditions that a system or a user must fulfill so that the user can succeed. Van der Meij et al. (2003, 141) discuss them in four parts: system states and modularity, user skills and fading, and user knowledge and mapping. Before the user can begin a task, they must know the current state of the system at hand (van der Meij & Gellevij 2004, 7). Modularity is a concept that relates to these states. If the writer of a manual wants to achieve modularity, they must pick a so-called home base from which the chapters of a manual begin and where they end (van der Meij et

al. 2003, 143). In other words, chapters always start from the same initial state that is at the same their goal state.

One way to help a user become more skilled with a system is fading. It means that as the manual proceeds from one procedure to another, the procedures become less detailed, and some of the simpler actions might be combined into same steps (van der Meij et al. 2003, 145). Van der Meij et al. (2003, 149) add that the manual should attempt to map the user's current knowledge to what the system requires of them. One way to achieve this are advance organizers. Comparative organizers use analogies and metaphors that draw parallels between the system and the real world, while expository organizers elaborate on the inner workings of the system.

Actions are undertaken by the user, and they result in system **reactions**. Van der Meij et al. (2003, 167) and van der Meij and Gellevij (2004, 7) discuss them in three parts: support and let go, screen captures, and action steps. If the user is advised to explore the product on their own and not only told directly what they must do, the user learns more about the system and not only *what* they should do but also *why* they should do it (van der Meij & Gellevij 2004, 7). Screen captures and other images visualize system reactions and help the user to develop a mental model (van der Meij et al. 2003, 162). Since action steps are arguably one of the most important components of a procedure, they are usually set apart from the rest of text by numbers (van der Meij & Gellevij 2004, 8). Action steps can consist of just a verb and a noun, but they may also include additional information, such as facilitating modifiers that let the user know where and how a certain action must be performed (ibid.).

Warnings alert users of unwanted states. Since warnings are of no use if the user cannot see them, they should be placed near the action steps what might lead to an unwanted state (SFS-EN 82079-1 2012, 40), and they must be set apart from the rest of the procedure with signal words, symbols and colors (van der Meij & Gellevij 2004, 10). They must also be easily readable and legible (ibid.). Furthermore, the user must be told what kind of unwanted state the system might end up in and what will happen if they ignore the warning (SFS-EN 82079-1 2012, 40). They must also be told what they must do to avoid danger or what actions might lead into it (van der Meij ja Gellevij 2004, 10). However, not all unwanted states are so serious that they require a warning, and the user might be able correct them with problem-solving information. Like warnings, such information should

be placed in the context where the user is most likely to need it and not in a separate section (van der Meij & Carroll 1995, 253; van der Meij & Gellevij 2004, 11).

Material and Method

I analyzed 18 individual procedures from software manuals and 18 from hardware manuals, but rather than study the differences between these two categories, I mainly focused on analyzing them as a whole, although I took note of significant differences. The manuals were published after 2010, and they were produced either by the concerned products' manufacturers or official third parties. In other words, I did not include unofficial guides, such as community-generated Wiki pages. Like van der Meij et al. (2003), I kept the focus of my study on commercial manuals. I collected most manuals from the manufacturer websites between December 1st, 2017 and February 28nd, 2018. The exceptions were HTML5 (Korpela 2014), Oracle (Hakkarainen 2011) and Java (Peltomäki 2014) guides which I borrowed from a library in Tampere. I chose my study material based on the types of the products rather than on their manufacturers. Product and manual lists are available in Chapter 7 and in the attachment (Liite 1). Following the example of van der Meij et al. (2003), I chose procedures at random, usually around the middle of a manual.

The study method is mainly qualitative, specifically theory-based content analysis. In other words, the categorization of the data was based on an existing model (Tuomi & Sarajärvi 2018, 91) that was tested in a new context (ibid.). I chose this method because one of the aims of this study is to determine whether the Four Components Model can be used to analyze Finnish procedures from the 2010s. However, even with the aid an existing framework, the definition of some terms and units proved to be challenging. I used the work of Farkas (1999) and Newell (1980) to complement the Four Components Model when necessary. I also used quantitative methods to describe my data. The units that I quantified are based on tables that were created for the original study (van der Meij et al. 2003), which in turn were based on the components of the Four Component Model.

Analysis of the Material

I analyzed the collected material one component at a time. Those components were further split into units that I have presented in tables in Chapter 5. The original English versions of these units are available in van der Meij et al. (2003).

The categorization of **goals** was largely based on GOMS analysis. A single task can contain one or more multi-step methods with which the task can be accomplished, and these steps include operators, i.e. user actions. A step can include alternative operators, of which van der Meij et al. (2003, 139) give the following example: 3. Press **Enter** or click *Add*. When alternative methods or operators are present, the user should be given enough information so that they can select the option that suits their needs best. Options were quite common in my material: they were included in 24 out of 36 procedures. Most of the options (19) were limited to a single step, but 9 procedures had alternative methods that spanned several steps. Roughly one third of the procedures aided the user in selecting the correct option, but it should be noted that some options did not have a clear advantage over one another and did not therefore require selection rules. To use an example that is not from my material, it makes little difference whether a user deletes text by backspace or Ctrl+Z.

Few action steps in my material included only a verb-noun pair. Most also had additional textual information, such as the results of the operator at hand. Some of the information was separated into notes, and information was also presented in visual form. Visual feedback was clearly more common in hardware manuals than in software manuals: on average, 0.19 steps per software procedure had images, while the same average for hardware procedures was 0.78. On average, the number of steps per procedure was about 4, which stays well within the capabilities of human working memory (see van der Meij et al. 2003, 134).

The categorization of titles put forth by van der Meij et al. (2003) and Farkas (1999) was naturally unsuitable for Finnish procedures, so I created units based on the titles that I found in my material. There was quite a lot of variety among them, but the fourth infinitive (*-minen*, which closely corresponds to the English gerund) was most common, followed closely by the noun phrase. Additional title categories that I found include questions, the imperative, and the partitive.

When it came to **prerequisites**, analyzing fading would require the study of a whole manual rather than a single procedure, which is why I cannot present results on the subject. I analyzed modularity by counting how often procedures let the user know of the required initial system state and how often the procedures referred elsewhere, either in the same manual or outside it. Modularity also includes the issue of advance organizers. They turned out to be difficult to define and hence count. While goal descriptions (e.g. conceptual elements) are not organizers, I counted them in the same category, as did van der Meij et al. (2003).

Initial states were included relatively rarely, while about half of the procedures had references elsewhere. However, references were more common than in the original study (van der Meij et al. 2003), which might suggest that electronic procedures of the 2010s contain more references than the procedures in the 1990s, but definite conclusions would require a larger sample size. Advance organizers were surprisingly common compared to the original study, but as I mentioned before, they were hard to define, which may have led to erroneous results.

As for **actions and reactions**, procedures favored direct guidance over encouraging exploration. In fact, only three procedures (Hakkarainen 2011, 237; Korpela 2014, 412; Microsoft 2016) included elements that can be interpreted as encouragement to explore the system. One reason for this discrepancy might be that when they need manuals, I would argue that users are more interested in accomplishing a particular task rather than learning about the product or system in general.

Images, which often visualize system reactions, were about twice as common in hardware procedures compared to software procedures (11 vs. 23). One explanation might be that hardware manuals describe physical objects that can be easily captured in a photograph or a vector graphic, while software manuals describe abstract concepts and inner workings of a program that cannot be visualized (Mäkynen 2012, 73). The majority of the images in my material were so-called flat images that do not present the flow of an action or label the parts of a product (see van der Meij et al. 2003, 170). While flat images might not contribute to the user's mental model development to the same extent that labeled or "flowing" pictures do, it should be noted that most flat pictures in my material had a caption or highlights. This means that they did not appear unrelated to the procedures.

User actions are included in action steps. Since I think that steps (along with warnings) are the most important part of a manual, they should be somehow set apart from rest of the text. True to the recommendations of van der Meij et al. (2003, 167) and official standards (SFS-EN 82079-1 2012, 52), this was the most popular method for discerning and organizing action steps. Some manuals mixed unnumbered paragraphs and numbered lists, but most stuck to just numbers. Facilitating modifiers that locate a UI element or other kind of objects were present in 25 procedures out of 36. Such modifiers are especially important if the location an element is not presented in a screen capture, but the correlation between the presence of a modifier and the lack of visual feedback was

outside the scope of this study. Visualization of elements – for example, text tool **T** (Adobe 2016) – was rather rare, which surprised me considering how important isomorphism is for mapping and mental model development (van der Meij et al. 2003, 178).

Unwanted states include dangerous situations that require warnings and milder problems that the user can fix with relative ease. Since few of the procedures that I analyzed carried risks, warnings were rare in my material. There were only 8 warnings, and most of them only let the user know what they should not do, not what might happen if the user ignores the warning. Furthermore, most did not include a signal word, nor were they easy to tell apart from the rest of the text, which means that they do not follow official standards (SFS-EN 82079-1 2012, 43). Problem-solving information was somewhat more common (9), which means that in most cases the user does not know what kind of unwanted states the system might end up in. However, it is possible that systems themselves notify the user of errors, which would reduce the need to list possible errors in procedures.

Conclusion

It appears that the Four Components Model is well-suited for the analysis of Finnish procedures despite the language and time differences between my study and that of van der Meij et al. (2003). The components and their units did not always appear in the same order, which implies that standards of appearance and structure differ between domains and companies.

As I predicted in the Introduction, the model had to be adapted into a new context. The most obvious example were titles: due to the languages used, they naturally differ between English and Finnish procedures. The variety of different types of titles was greater than I anticipated, and while my sample size was rather small, I think that the material gives a good overview on what kinds of titles are found in Finnish procedures. Moreover, the translation of the Four Components Model offers a new kind of framework for the study of Finnish-language procedures. In my opinion, particular attention should be paid to action steps since they are the reason why manuals are read and the means by which users reach their goals. Another important area of study is problem-solving information and warnings, especially since those found in my material could have been more detailed.

Contrary to what I expected, translating and adapting the Four Components Model turned out to be an easier task than interpreting it. Not all units were explained in detail, and in some cases I had to rely on the work of Farkas (1999) and Newell (1980). Furthermore, some of the terms overlapped or were used interchangeably, such as *method* and *approach* (van der Meij et al. 2003, 137–138), and on hindsight, creating a mind map would have made my work easier.

While I have drawn tentative conclusions from my analysis, it should be noted that more reliable results require a larger sample. However, the data I have gathered implies that warnings should include more details about the hazards at hand and that warnings and problem-solving information should be placed more frequently in procedures themselves rather than in separate sections. Visual feedback in software procedures was rarer than in the original study (van der Meij et al. 2003), and references were also more common. Both might be explained by technological advantages made in the last two decades: today's users might need less support than users of the 1990s, while references may be more common because manuals these days are often electronic. Yet, overall the components and units of procedures match the ideas of van der Meij et al. (2003) and Farkas (1999).

In addition to a larger sample, the reliability of my results would have benefited from more researchers. As I mentioned before, I had trouble defining some of the units, and discussing definitions and criteria with someone else would have been enlightening. Therefore, it would be interesting to see what kind of results another researcher or a group would get. Another reason for some of the subjectivity of this study is that random selection by human agents can never be truly random, and I would advise future researches to further randomize their samples in some way, such as by using a combination of web scraping and randomizer scripts (see Thompson 2012, Olteanu 2017 & Paruchuri 2016). If another party wishes to replicate my study, I believe that my translation and definitions will be of use to people who seek a rather straightforward but comprehensive way to analyze procedures in Finnish manuals. I have suggestions for future analyses that I will elaborate on next.

One way to change the calculation of selection rules is noting which individual options include explanations on why the user should select them. An analysis of problem-solving information can be based on the detect-diagnose-correct framework (see van der Meij & Gellevis 2004, 11 and chapter 3.4.2), and another area of interest is how facilitating modifiers and screen captures support or replace one another. As I mentioned before, analyzing fading techniques is not possible while

studying only one procedure per manual, but it can be done if whole manuals are studied. While studying a complete manual, it would also be possible to find out whether its procedures always phrase titles and subtitles in the same manner.