



Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Atte Lainejoki

Kirjallisuuskatsaus: suunnittelumallit ja mallikielet suunnittelun apuvälineinä

Metropolia Ammattikorkeakoulu

Medianomi (AMK)

Viestinnän koulutusohjelma

Opinnäytetyö

23.4.2019

Tekijä(t) Otsikko	Atte Lainejoki Kirjallisuuskatsaus: suunnittelumallit ja mallikielet suunnittelun apuvälineinä
Sivumäärä Aika	28 sivua 23.4.2019
Tutkinto	Medianomi (AMK)
Tutkinto-ohjelma	Viestintä
Suuntautumisvaihtoehto	Digitaalinen viestintä
Ohjaaja(t)	Lehtori Tero Marin
<p>Tässä opinnäytetyössä on tavoitteena luoda katsaus modulaariseen suunnitteluun liittyviin suunnittelumalleista ja mallikielistä tehtyihin julkaisuihin. Työn menetelmänä käytetään laadullista kirjallisuuskatsausta. Näkökulma tälle työlle on digitaalisten tuotteiden suunnittelu.</p> <p>Kirjallisuuskatsauksen aineistoksi valittiin yhdeksän julkaisua. Aineisto valikoitui verkon aineistohakujen, aiemman aiheeseen tutustumisen sekä lähdeviitteiden kautta, ja siihen kuuluu artikkeleita, kirjoja sekä verkkojulkaisu. Työn alkupuolella esitellään aihepiiriä, kirjallisuuskatsausta menetelmänä sekä menetelmän käyttöä tässä opinnäytetyössä. Neljäs luku sisältää kirjallisuuskatsauksen. Viimeinen luku tiivistää kirjallisuuskatsauksen pohjalta tehdyt johtopäätökset.</p> <p>Kun erilaisten laitteiden määrä kasvaa ja vaatimukset sovelluksille samalla muuttuvat jatkuvasti, digitaalisten tuotteiden suunnittelu monimutkaistuu. Suunnittelu on siirtynyt tämän vuoksi enemmän modulaarisuuteen, jolloin suunnitteluelementtejä pyritään käyttämään uudelleen monipuolisesti mahdollisimman monessa yhteydessä. Modulaarisuus ei ole ajatukseen uusi, vaan erilaisia tapoja toteuttaa modulaarisuutta suunnittelussa on ollut jo pitkään. Eräs näistä tavoista on vuonna 1977 arkkitehtuurin alalla esitelty mallikieli, jota on myöhemmin sovellettu monella eri alalla, erityisesti ohjelmistokehityksessä.</p> <p>Kuten tästä opinnäytetyöstä käy ilmi, suunnittelumallit ja mallikielet ovat konsepteina pysyneet hyvin samankaltaisina aina tähän päivään asti, sillä sama ajatus soveltuu hyvin käytettäväksi erilaisissa tilanteissa. Vaikka suunnittelumallien ja mallikielten käyttö voivat vaatia organisaatiolta merkittävää sitoutumista, ne voivat kuitenkin maksaa itsensä takaisin parantuneen yhteistyön ja kokonaisuuden hallinnan kautta.</p>	
Avainsanat	suunnittelumalli, mallikieli, suunnittelujärjestelmä, modulaarinen suunnittelu, kirjallisuuskatsaus

Author(s) Title	Atte Lainejoki Literature review: design patterns and pattern languages as design tools
Number of Pages Date	28 pages 23 April 2019
Degree	Bachelor of Culture and Arts
Degree Programme	Media
Specialisation option	Digital Media
Instructor(s)	Senior Lecturer Tero Marin
<p>The aim of this bachelor's thesis is to create an overview of design pattern and pattern language publications related to modular design. The thesis was carried out as a qualitative literature review. The perspective of this thesis is the design of digital products.</p> <p>9 publications were selected for the review. The material was selected through database searches, prior familiarisation with the subject, and references in other works. The review includes articles, books and an online publication. At the beginning of the thesis, the topic, the method and the use of it in this thesis are presented. The fourth chapter contains the literature review. The last chapter summarizes the conclusions drawn from the review.</p> <p>As the number and variety of different devices increase and the requirements for applications at the same time change continuously, the design of digital products becomes more and more complex. Thus, design has to be conducted in a more modular manner, with the aim of reusing elements in as many ways as possible. Modularity is not a recent concept, and different ways of implementing modularity in design have long been around. One of these ways is the idea of pattern languages introduced in 1977 in the field of architecture and later applied in many other fields, most notably in software development.</p> <p>The thesis results indicate that design patterns and pattern languages have not changed much as concepts up to this day, as the same idea is well suited for use in different situations. While the use of design patterns and pattern languages may require significant commitment from the organization using them, they can still pay for themselves through improved collaboration and management of the whole.</p>	
Keywords	design pattern, pattern language, design system, modular design, literature review

Sisällys

1	Johdanto	1
2	Aiheen pohjustus	2
3	Kirjallisuuskatsauksen kuvaus	4
3.1	Aiheeseen liittyvää termistöä	4
3.2	Kirjallisuuskatsaus tutkimusmenetelmänä	5
3.3	Kirjallisuuskatsaus tässä opinnäytetyössä	7
4	Kirjallisuuskatsaus: suunnittelumallit ja mallikieliet suunnittelun apuvälineinä	13
4.1	Suunnittelumallien määrittelyä	13
4.2	Mallikielen määrittelyä	14
4.3	Mallikielen hierarkkisuus	16
4.4	Suunnittelumallin rakenne	17
4.5	Suunnittelumallien ja mallikielten kehittyminen	19
4.6	Suunnittelumallien ja mallikielten vaikutus ihmisten väliseen yhteistyöhön	20
4.7	Haasteita suunnittelumalleissa	23
5	Kirjallisuuskatsauksen yhteenveto	24
6	Lopuksi	25
	Lähteet	27

1 Johdanto

Kun erilaisten laitteiden määrä kasvaa ja vaatimukset sovelluksille samalla muuttuvat jatkuvasti, digitaalisten tuotteiden suunnittelu monimutkaistuu. Suunnittelu on siirtynyt tämän vuoksi enemmän *modulaarisuuteen*, jolloin suunnitteluelementtejä pyritään käyttämään uudelleen monipuolisesti mahdollisimman monessa yhteydessä. (Frost 2016, 11–13.)

Tämän opinnäytetyön tavoitteena on tarkastella modulaariseen suunnitteluun liittyviä suunnittelumalleista ja mallikielistä tehtyjä julkaisuja. Yleiskuvan luomiseksi työn menetelmänä on käytetty kuvailevaa kirjallisuuskatsausta. Työn näkökulma on erityisesti digitaalisten tuotteiden suunnittelussa, mutta kirjallisuuskatsauksen aineistoon kuuluu myös muista näkökulmista aihetta tarkastelevia julkaisuja. Työ voi auttaa mallikielen tai suunnittelujärjestelmän luomista suunnittelevaa lukijaa ymmärtämään laajemmin mallikielten vaatimukset, hyödyt ja haasteet.

Tutkin suunnittelujärjestelmiä tutkielmassani syksyllä 2018 tarkoitukseni pohjustaa teoreettista viitekehystä tälle opinnäytetyölle, jonka tavoitteena oli puolestaan alun perin tarjota hyviä käytänteitä suunnittelujärjestelmän luomiselle. Opinnäytteen suunnitteluvaiheessa törmäsin kuitenkin jatkuvasti eri lähteiden kautta arkkitehti Christopher Alexanderin vuonna 1977 esittelemään ajatukseen suunnittelumallikielestä, jossa tuntui olevan paljon yhtymäkohtia nykypäivän suunnittelujärjestelmiin. Niinpä kiinnostuin tutkimaan tarkemmin, kuinka mallikielet ovat vaikuttaneet suunnitteluun eri tieteenaloilla ja millaisia samankaltaisuuksia ja eroavaisuuksia niistä löytyy toisiinsa verrattuna.

Tämän opinnäytetyön toisessa luvussa esitellään modulaarisuuden historiaa sekä aihetta yleisesti. Kolmas luku käsittelee tämän opinnäytetyön menetelmänä olevaa kuvailevaa kirjallisuuskatsausta, menetelmän käyttöä tässä työssä sekä kirjallisuuskatsauksen valittua aineistoa. Neljäs luku sisältää varsinaisen kirjallisuuskatsauksen. Lopuksi tiivistetään kirjallisuuskatsauksen pohjalta tehdyt johtopäätökset ja esitellään mahdollisia jatkotutkimusaiheita.

2 Aiheen pohjustus

Modulaarisuus ei ole järin tuore tai vain teknologiaan ja tietojenkäsittelytieteeseen liittyvä ajatus. Andrew L. Russellin (2012, 257) mukaan modulaarisuutta on konseptina käsitelty monissa tieteenhaaroissa, kuten evoluutiobiologiassa, johtamisen tutkimuksessa sekä kasvatustieteissä. Modulaarisuuden onkin perinteisesti ajateltu auttavan kompleksisten asioiden käsitteellistämässä ja hallitsemisessa (Russell 2012, 259).

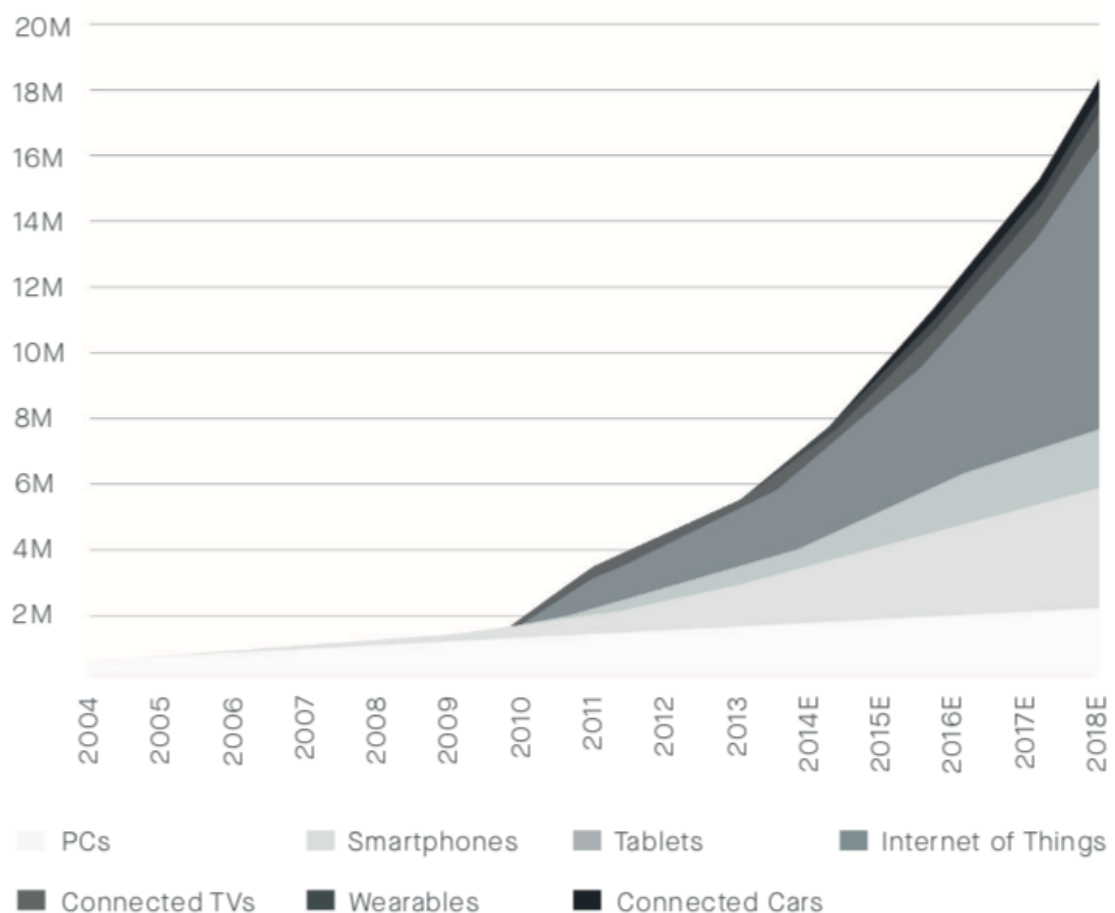
Ohjelmistoalalla modulaarisuuden ajatus lähti liikkeelle 1960-luvulla. Tällöin oltiin tilanteessa, jossa tietokonetekniikan kehitys oli nopeutunut niin paljon, ettei ohjelmistokehitys pysyisi enää pian perässä. Euroopassa asennettujen tietokoneiden määrä kasvoi 25–50 prosenttia vuodessa, ja ohjelmistokehityksen vaatimat henkilötyövuodet olivat saavuttamassa tietokonetekniikan vastaavat. Järjestelmät olivat lisäksi usein myöhässä sovitusta aikataulusta, ne eivät vastanneet etukäteen tehtyä vaatimusmäärittelyä, eivätkä kustannukset pysyneet budjeteissa. Pienikin muutos projektin vaatimukseen tai yksikin havaittu virhe ohjelmoinnissa saattoi aiheuttaa kumuloituvasti suuria muutoksia ohjelman koodissa, mikä puolestaan teki ohjelmien ylläpidosta vaikeaa. (Naur & Randell 1969, 9.)

Naton tiedekomitea järjesti tämän *ohjelmistokriisiksi* (engl. *software crisis* tai *software gap*) kutsutun tilanteen ratkaisemiseksi kaksi konferenssia vuosina 1968 ja 1969. Konferensseissa ohjelmistoalan asiantuntijat kokoontuivat määrittämään, kuinka ohjelmistoja voitaisiin kehittää entistä järjestelmällisemmin. (Aßmann 2013, 6.) Matemaatikko Douglas McIlroy esitti ensimmäisessä konferenssissa ajatuksen komponenttipohjaisesta suunnittelusta ohjelmistokehityksessä. McIlroyn mukaan modulaarisilla, toisistaan riippumattomilla komponenteilla ohjelmistoja kyettäisiin massatuottamaan tehokkaammin. (Naur & Randell 1969, 79–80.)

Douglas McIlroyn tietokonetekniikkaan tuoman komponenttipohjaisen modulaarisen suunnittelun lisäksi myös muilla tieteenaloilla oli samoihin aikoihin pohdittu modulaarisuutta. Christopher Alexander, Sara Ishikawa ja Murray Silverstein esittelivät *A Pattern Language* -kirjassaan (1977) *suunnittelumallien* (engl. *patterns*) konseptin arkkitehtuurin alalla. Teoksessa tunnistetaan yleisiä ympäristössä olevia ongelmia ja tämän jälkeen jokaiseen ongelmaan annetaan ratkaisu, jota voidaan käyttää toistuvasti eri konteksteissa – kuitenkin aina tarvittaessa uudella tavalla (Alexander, Ishikawa & Silverstein 1977, sivu x). Alexanderin ym. tapa jäsentää ongelmia ja niiden ratkaisuja siirtyi perinteisestä arkkitehtuurista ohjelmistokehityksen alalle vuonna 1995 julkaistun *Design Pat-*

terns: elements of reusable object-oriented software -kirjan myötä. Tämän teoksen mukaan suunnittelumallit auttavat niiden käyttäjää löytämään oikean ratkaisun ongelmaan nopeammin (Gamma, Helm, Johnson & Vlissides 1995, 2).

Digitaalisten tuotteiden suunnittelu on tänä päivänä hyvin samankaltaisessa tilanteessa kuin ohjelmistoala oli 1960-luvun lopulla. Erilaisten päätelaitteiden määrä kasvaa jatkuvasti (kuvio 1), ja sovellukset ovat entistä monimutkaisempia. Suunnittelun ajatellaan edelleen olevan pitkälti mittatilauksena tehtäviä ratkaisuja uniikkeihin ongelmiin, ja sen vuoksi suunnittelu ei vaatimusten kasvaessa skaalaudu eli pysy kehityksen perässä. (Suarez 2017.)



Kuvio 1. Maailmanlaajuisesti käytössä olevat laitetypit (tuhatta kappaletta; vuodet 2014–2018 arvioita). (Goldman 2018, 10; alkup. kuvio Danova 2014¹.)

¹ Danova, Tony 2014. The Internet Of Everything: 2014. Business Insider. <<https://www.businessinsider.com/the-internet-of-everything-2014-slide-deck-sai-2014-2>> (luettu 11.4.2019).

Joukko tunnettuja verkkosuunnittelijoita julkaisi vuonna 2011 manifestin, jossa todettiin, että verkon alati muuttuvien käyttötapojen ja jatkuvasti kasvavan laitetarjonnan vuoksi käytössä olevat tuotesuunnitteluprosessit eivät ole sellaisenaan kestäviä (Future-Friendly Manifesto, 2011). Viime vuosina – osin vastauksena manifestin varoitukseen – organisaatiot ovatkin alkaneet kehittää suunnittelunsa avuksi *suunnittelujärjestelmiä* (engl. *design systems*), jotka ovat eräs tapa hyödyntää suunnittelumalleja ja mallikieltä digitaalisten tuotteiden suunnittelussa.

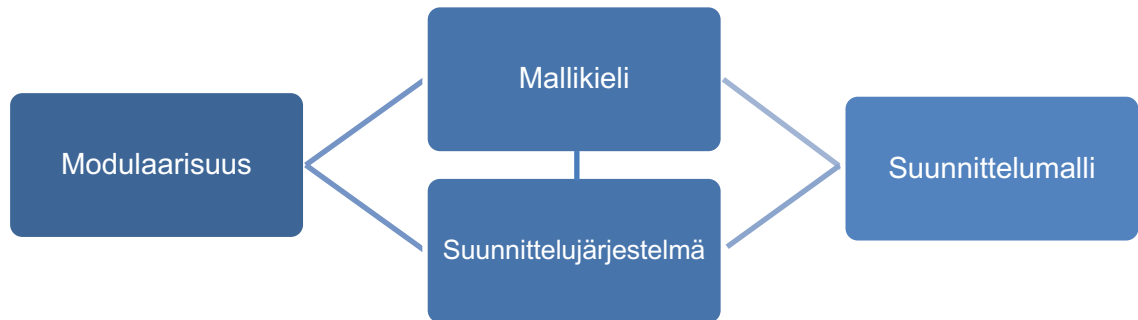
3 Kirjallisuuskatsauksen kuvaus

3.1 Aiheeseen liittyvää termistöä

Modulaarisuus on kattoterminä tämän opinnäytetyön aiheille. Modulaarisuudella voidaan mallintaa monimutkaisia järjestelmiä, joiden rakennusosia moduulit ovat. Moduulit ovat rakenteellisesti irrallaan toisistaan mutta toimivat kuitenkin yhteen. (Baldwin & Clark 2000, 63.)

Työssä käsitellyillä suunnittelumalleilla ja mallikielillä ei ole vakiintunutta määritelmää, ja niiden käyttötapaa vaihtelee tieteenalasta riippuen. *Suunnittelumallit* (engl. *pattern* tai *design pattern*) ovat kuitenkin yleistettynä muodollinen tapa kuvata ongelma ja siihen luotu ratkaisu (Carr ym. 2001, 1). *Mallikieliet* (engl. *pattern language*) ovat puolestaan kokonaisuus suunnittelumalleja, jossa jokainen malli liittyy toiseen malliin muodostaen verkon (Jessop 2004, 458).

Lisäksi etenkin digitaalisten tuotteiden suunnittelun kohdalla puhutaan enenevässä määrin *suunnittelujärjestelmistä* (engl. *design system*), joilla tarkoitetaan joukkoa yhteenkuuluvia suunnittelumalleja ja yhteisiä käytänteitä, jotka on järjestetty johdonmukaisesti niin, että ne palvelevat digitaalista tuotetta (Kholmatova 2017, xii). Suunnittelujärjestelmän määritelmä on siis hyvin lähellä mallikielen määritelmää. Suunnittelujärjestelmiin liittyvät läheisesti *mallikirjastot* (engl. *pattern library*), jotka ovat yksi tapa kerätä ja jakaa tietoa suunnittelumalleista yhtenäisessä muodossa (Kholmatova 2017, xii).



Kuvio 2. Kirjallisuuskatsauksen keskeisimpien termien väliset suhteet.

Kirjallisuuskatsauksessa toisiinsa liittyviä termejä on yhdistetty selkeyden vuoksi yhteen silloin kun se on samanlaisen määritelmän mukaan ollut perusteltua. Esimerkiksi suunnittelujärjestelmästä ja mallikirjastosta on käytetty termiä mallikieli, kun määritelmä on julkaisussa ollut vastaava.

3.2 Kirjallisuuskatsaus tutkimusmenetelmänä

Tässä luvussa esitellään kirjallisuuskatsausta tutkimusmenetelmänä. Luvussa kerrotaan, millaisista syistä kirjallisuuskatsausta käytetään, millaisiin perustyyppisiin metodi voidaan jakaa sekä millainen on tässä opinnäytteessä käytettävä kuvaileva kirjallisuuskatsaus.

Kirjallisuuskatsaus on menetelmä ja tutkimustekniikka, jonka avulla voidaan tutkia tehtyä tutkimusta (Salminen 2011, 4). Menetelmää käytetään monista syistä. Baumeisterin ja Learyn (1997, 312) mukaan kirjallisuuskatsauksille voidaan erottaa viisi varsinaista tavoitetta:

1. Uuden teorian esittäminen ja olemassa olevan kehittäminen.
2. Teorian arvioiminen siihen liittyvän kirjallisuuden avulla.
3. Kokonaiskuvan luominen tietystä aiheesta.
4. Teoriassa tai aihepiirissä olevien ongelmien tunnistaminen ja esilletuominen.
5. Teorian kehityksen ja tutkimuksen historiallinen tutkiminen.

Bearfieldin ja Ellerin (2007, 63–64) mukaan kirjallisuuskatsaus voi esitellä joko tieteenhaaran historiaa tai ideoiden historiaa. Kun kirjallisuuskatsausta lähestytään tieteenhaaran historian kautta, pyritään kartoittamaan kronologisesti, kuinka tiettyä aihetta on käsitelty kyseessä olevaan tieteenhaaraan liittyvissä tutkimuksissa. Ideoiden historian näkökulmasta tehdyssä kirjallisuuskatsauksessa sen sijaan ylitetään tieteenhaarojen rajat ja pyritään paljastamaan tietyn idean yhtäläisyydet ja eroavaisuudet eri tieteenalojen tutkimuksissa. (Bearfield & Eller 2007, 63–64.)

Vaikka katsauksella tarkoitetaan suomen kielessä usein lyhyttä tai tiivistä vilkaisua, ei sellaisesta ole Salmisen (2011, 5) mukaan kirjallisuuskatsauksessa kyse. Kirjallisuuskatsaus-termin englanninkielisissä vastineissa (*review, literature review, research literature review*) sanalla review tarkoitetaan katsauksen lisäksi uudelleen tai kriittisesti tehtävää arviointia, selontekoa, arvostelua, tarkistusta tai historiikkaa (Salminen 2011, 5).

Kirjallisuuskatsaus jaetaan yleensä kolmeen päätyyppiin, jotka ovat kuvaileva kirjallisuuskatsaus, systemaattinen kirjallisuuskatsaus ja meta-analyysi (Salminen 2011, 6). Kuvailevassa kirjallisuuskatsauksessa ei ole tarkoin määriteltyjä sääntöjä ja tutkimuskysymys on usein rajattu väljästi (Salminen 2011, 6). Systemaattinen kirjallisuuskatsaus pyrkii tiivistämään aihepiiristä tehtyjen tutkimusten olennaisimmat sisällöt. Systemaattisessa kirjallisuuskatsauksessa käydään läpi yleensä runsas määrä tutkimusmateriaalia, jotta tarve tutkijan tekemälle omalle tutkimukselle olisi helpompi perustella. Meta-analyysillä on sekä kvalitatiivinen että kvantitatiivinen suuntaus. (Salminen 2011, 9.) Kvalitatiivisella meta-analyysillä on kaksi orientaatiota: metasynteesi ja metayhteenveto. Metasynteessissä pyritään yhtäläisyyksien ja erojen kautta rakentamaan aihealueesta vakuuttava kokonaiskuva, ja metasynteesiin valittujen tutkimustulosten tulkintojen pohjalta pyritään luomaan uusi tulkinta. (Salminen 2011, 12.) Myös metayhteenvedossa tiivistetään laadullisia tutkimustuloksia uuden tulkinnan tekemiseksi mutta kvantitatiivisilla menetelmillä. Metayhteenveto on kvantitatiivisista menetelmistä huolimatta kuitenkin laadullinen tutkimusmenetelmä. (Salminen 2011, 13.) Kvantitatiivinen meta-analyysi on Salmisen (2011, 14) mukaan puolestaan metodisesti vaativin kirjallisuuskatsauksen tyyppi. Meta-analyysissä kvantitatiivisesti tehtyjä tutkimuksia yhdistetään tilastotieteen menetelmin (Salminen 2011, 14).

Tämän opinnäytetyön kirjallisuuskatsauksen tyyppiä valikoitui aiheen, tutkimuskysymyksen ja työn laajuuden perusteella kuvaileva kirjallisuuskatsaus ja tarkemmin sen integroiva alalaji. Integroiva katsaus on kuvailevan kirjallisuuskatsauksen orientaatio, jota voi-

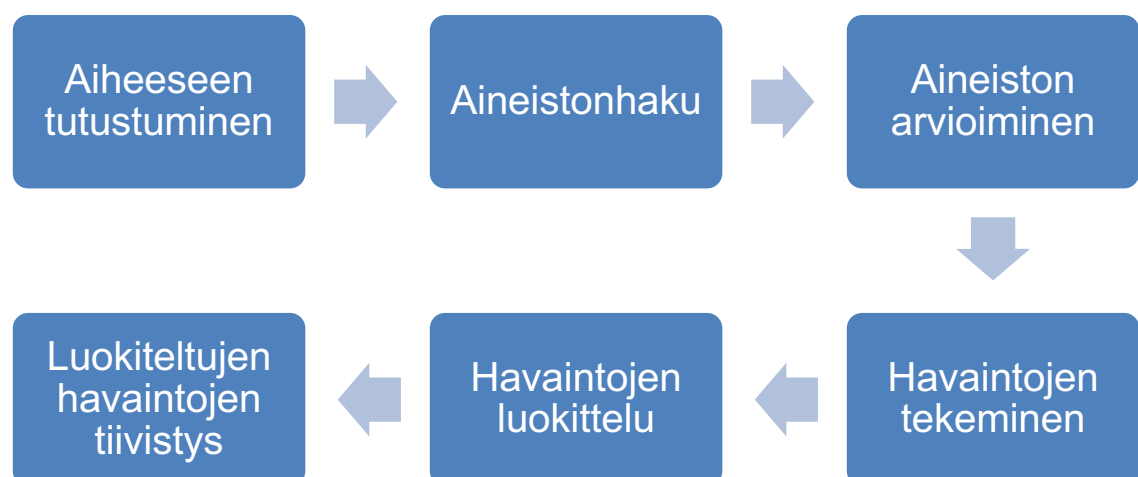
daan käyttää tutkittavan ilmiön kuvaamiseen mahdollisimman monipuolisesti. Vaikka integroiva kirjallisuuskatsaus on lähellä systemaattista katsausta, menetelmälle on ominaista, ettei tutkimusaineistoa seulota niin tarkasti ja että erilaisista menetelmällisistä lähtökohdista tehdyt tutkimukset voidaan ottaa analyysin pohjaksi. (Salminen 2011, 8.) Kirjallisuuskatsaus on tehty myös Bearfieldin ja Ellerin esittämän ideoiden historian näkökulmasta, jotta katsauksessa voidaan tieteenhaarojen rajat ylittämällä löytää aineistosta yhtäläisyyksiä ja eroavaisuuksia.

3.3 Kirjallisuuskatsaus tässä opinnäytetyössä

Tässä alaluvussa esitellään kirjallisuuskatsausta varten tehdyn aineistonhaun prosessia, aineiston läpikäyntiä ja analysointia sekä analyysistä syntyneiden tulosten vertailua.

Tämän opinnäytetyön kirjallisuuskatsauksen prosessi koostui seuraavista vaiheista Cooperin (1989, Salmisen 2011, 8 mukaan) prosessia mukaillen:

1. Aiheeseen tutustuminen ja tutkimusongelman asettelu
2. Aineistonhaku
3. Aineiston arvioiminen
4. Havaintojen analysoiminen luokitellen



Kuvio 3. Kirjallisuuskatsauksen prosessi tässä opinnäytetyössä.

Aineistoa etsittiin pääasiassa tietokantahauilla erilaisilla hakusanoilla sekä niiden yhdistelmillä. Hakuihin käytettiin Googlen Scholar-hakupalvelua, Association for Computing Machinery -yhteisön Digital Library -julkaisutietokantaa, Metropolian kirjaston MetCat-aineistohakua, Helka-kirjastojen Helka-aineistohakua sekä Ammattikorkeakoulujen rehtorineuvosto Arene ry:n Theseus-opinnäytetyötietokannan hakua. Tietokantahauista löydettyjen lähteiden lisäksi aineistoksi valikoitui myös jo aiemman aiheeseen tutustumisen aikana löydettyjä julkaisuja.

Tietokantahauissa käytettiin seuraavia hakusanoja ja niiden yhdistelmiä: design pattern, design system, pattern library, pattern language, modular design, modular web design sekä modular architecture. Lisäksi hakusanoista käytettiin myös suomenkielisiä vastineita sikäli, kun se tietokannan julkaisujen kielen osalta oli tarkoituksenmukaista. Seuraavia suomenkielisiä hakutermejä käytettiin aineistohauissa: suunnittelumalli, suunnittelujärjestelmä, suunnittelumallikirjasto, mallikieli, suunnittelumallikieli, modulaarinen suunnittelu, modulaarinen web-suunnittelu, modulaarinen verkkosuunnittelu sekä modulaarinen arkkitehtuuri.

Hakujen perusteella löytynyt aineisto karsittiin lopulliseksi kirjallisuuskatsauksen aineistoksi arvioimalla erikseen jokaista julkaisua seuraavan kolmen vaatimuksen: kuinka laajasti julkaisu käsitteli tämän kirjallisuuskatsauksen tutkimuskysymyksen aihetta, kuinka merkittävä teos on aihepiirissään ja onko teos vertaisarvioitua tutkimusta. Jokainen näistä syistä saattoi myös kompensoida toista. Julkaisun ei esimerkiksi tarvinnut olla vertaisarvioitu artikkeli, mikäli se oli erityisen merkittävä aihepiirissään. Merkittävyyttä arvioitiin muun muassa teokseen viittaavien lähteiden määrän ja kirjoittajan tunnettuuden perusteella.

Lopulta aineistoksi rajautui yhdeksän erityyppistä julkaisua eri tieteenaloilta (taulukko 1). Julkaisuista neljä on artikkeleita, neljä kirjoja ja yksi puolestaan on verkkojulkaisu. Alat, joita julkaisut edustavat, ovat arkkitehtuuri, tuotesuunnittelu, käyttöliittymäsuunnittelu, ohjelmistokehitys ja operaatioanalyysi.

Taulukko 1. Lopulliseen kirjallisuuskatsaukseen valittu aineisto.

Tekijä(t)	Julkaisun nimi	Julkaisuvuosi	Kuinka löytyi
Alexander, Christopher	The Timeless Way of Building	1979	Aiempi aiheeseen tutustuminen
Anne, Jina, Mounter, Diana, Stanfield, Roy, Suarez, Marco, Saylor-Miller, Katie	Design Systems Handbook	2017	Aiempi aiheeseen tutustuminen
Carr, David A., Granlund, Åsa, Lafrenière, Daniel	A Pattern-Supported Approach to the User Interface Design Process	2001	Lähdeviite
Cline, Marshall P.	The Pros and Cons of Adopting and Applying Design Patterns in the Real World	1996	MetCat-aineistohaku
Frost, Brad	Atomic Design	2016	Aiempi aiheeseen tutustuminen
Gamma, Erich, Helm, Richard, Johnson, Ralph, Vlissides, John	Design Patterns: Elements of Reusable Object-Oriented Software	1994	Lähdeviite
Jessop, Alan	Pattern language: A framework for learning	2004	MetCat-aineistohaku
Kholmatova, Alla	Design Systems	2017	Aiempi aiheeseen tutustuminen
Mahemoff, Michael J., Johnston, Lorraine J.	Principles for a Usability-Oriented Pattern Language	1998	Lähdeviite

Aineiston perusteoksena voidaan pitää Christopher Alexanderin *The Timeless Way of Building* -teosta, sillä monen muun aineiston julkaisun määritelmä suunnittelumalleista ja mallikielistä perustuu siihen. Alexanderin teoksessa mallikieliä kuvataan hyvin utopistisesti ja myös holistisena elementtinä, joka määrittää kaikkea ympäröivässä maailmassa.

Erich Gamman, Richard Helmin, Ralph Johnsonin ja John Vlissidesin teos *Design Patterns: Elements of Reusable Object-Oriented Software*, Brad Frostin teos *Atomic Design* ja Alla Kholmatovan teos *Design Systems* ovat kukin aloillaan määrittäneet laajemmin tapaa, joilla mallikieliä käytetään. Gamma ym. (1994) käsittelevät teoksessaan yleisiä suunnitteluongelmia ohjelmistotuotannossa ja antavat kullekin ongelmalle mahdollisen yleiskäyttöisen ratkaisun suunnittelumallin muodossa. Frostin (2016) ja Kholmatovan (2017) julkaisut käsittelevät molemmat digitaalisten tuotteiden suunnittelun tehostamista. Frost esittelee teoksessaan atomisen suunnittelun (engl. *atomic design*) käsitteen, joka määritelmältään muistuttaa mallikieltä. Kholmatova puolestaan esittelee tavan luoda ja hallinnoida suunnittelujärjestelmiä, jotka ovat myös omanlaisiaan mallikieliä.

Design Systems Handbook -julkaisu on suunnittelutyökaluja tarjoavan InVision-yrityksen kokoelma artikkelimuotoisia oppaita suunnittelujärjestelmien käyttämiseksi organisaatioissa. Vaikka julkaisu on kaupallinen, se valikoitui osaksi kirjallisuuskatsauksen aineistoa, sillä sen kirjoittajat ovat tunnettuja suunnittelujärjestelmäajatuksen kehittäjiä ja teos käsittelee suunnittelujärjestelmiä laajasti.

Artikkelit *A Pattern-Supported Approach to the User Interface Design Process* (David A. Carr, Åsa Granlund ja Daniel Lafrenière), *The Pros and Cons of Adopting and Applying Design Patterns in the Real World* (Marshall P. Cline), *Pattern language: A framework for learning* (Alan Jessop) ja *Principles for a Usability-Oriented Pattern Language* (Lorraine J. Johnston ja Michael J. Mahemoff) käsittelevät kukin suunnittelumalleja tai mallikieliä eri näkökulmista. Carr ym. (2001, 1) esittelevät käyttöliittymäsuunnittelun kontekstissa tavan jäsentää tietoa käyttäjälle suunnittelumallilähtöisellä lähestymistavalla (engl. *pattern-supported approach*). Cline (1996, 47) esittelee käytännön esimerkkejä suunnittelumallien aiheuttamista onnistumisista ja epäonnistumisista ohjelmistosuunnittelussa. Jessopin (2004, 457) artikkelissa mallikielien käyttötapoja esitellään organisaatioiden kehityksen näkökulmasta. Johnston & Mahemoff (1998, 1) puolestaan pohtivat artikkelissaan tapoja ratkaista toistuvia suunnitteluongelmia ohjelmistojen käytettävyyden suunnittelussa.

Edellä esiteltyä aineistoa luettiin systemaattisesti luokitellen tutkimuskysymyksen kannalta merkityksellisiä otteita (esimerkki taulukossa 2). Luokittelutapa luotiin pääasiassa *The Timeless Way of Building* -teoksen pohjalta, sillä alustavan aiheeseen tutustumisen perusteella koin kyseisen teoksen aiheen kannalta merkittävimmäksi ja eniten muuhun aineistoon vaikuttavaksi. Luokkia kuitenkin tarkennettiin myös muun aineiston lukemisen

yhteydessä ja lisäksi yksi luokista luotiin kokonaan vasta myöhemmin läpikäytyjen teosten pohjalta. Neljännen luvun kirjallisuuskatsauksen raportin rakenne pohjautuu näille luokille. Luokat ovat järjestyksessään

- suunnittelumallien määrittely,
- mallikielen määrittely,
- mallikielten hierarkkisuus,
- suunnittelumallien rakenne,
- suunnittelumallien ja mallikielten kehittyminen,
- suunnittelumallien ja mallikielten vaikutus ihmisten väliseen yhteistyöhön sekä
- haasteet suunnittelumalleissa.

Näistä luokista kaikki viimeistä lukuun ottamatta on muodostettu Alexanderin teoksesta kerättyjen otteiden perusteella. Viimeinen luokka syntyi muun aineiston otteiden pohjalta. Luvut, joissa käsitellään Alexanderin teosta, myös alkavat kyseisen teoksen otteilla, sillä historiallisen merkittävyyden ja utopistisen hengen avulla kyseiset otteet antavat hyvän raamin muiden otteiden käsittelylle.

Taulukko 2. Esimerkki luokitelluista otteista.

Julkaisu	Ote	Luokka
The Timeless Way of Building	We have defined the concept of a pattern language clearly. We know that it is a finite system of rules which a person can use to generate an infinite variety of different buildings—all members of a family—and that the use of language will allow the people of a village or a town to generate exactly that balance of uniformity and variety which brings a place to life.	Mallikielen määrittelyä
The Timeless Way of Building	The language will evolve, because it can evolve piece-meal, one pattern at a time. [...] As people exchange ideas about the environment, and exchange patterns, the overall inventory of patterns in the pattern pool keeps changing. [...] Of course, this evolution will never end.	Mallikielten kehittyminen

Otteiden keräämisen jälkeen aineistosta kerätyt luokitellut otteet käytiin uudelleen läpi julkaisu kerrallaan. Tässä läpikäyntivaiheessa otteita pelkistettiin merkitykset säästään ja niitä yhdistettiin samankaltaisuuksien perusteella. Lopputulokseksi muodostui matriisityyppinen taulukko aineistojen keskeisistä havainnoista luokittain jaettuna (taulukko 3).

Taulukko 3. Esimerkki havainnoista tiivistettynä ja yhdistettynä.

Julkaisu / aihe	Mallikielen määrittelyä	Mallikielen hierarkkisuus
The Timeless Way of Building	<p>Yksinkertaisimmillaan suunnittelumallikieli koostuu kahdesta joukosta: joukosta elementtejä ja symboleja sekä joukosta sääntöjä näiden symbolien yhdistelyyn.</p> <p>Mallikieli koostuu rajatusta määrästä sääntöjä, joilla voidaan luoda rajattomasti erilaisia rakennuksia, jotka kuuluvat samaan yhtenäiseen ryhmään.</p> <p>Kaikki rakennukset saavat sekä rakenteensa että henkensä ihmisten käyttämistä mallikielistä.</p>	<p>Suunnittelumallit kattavat jokaisen tason ympäristössämme: suurimmat mallit käsittelevät alueellisia rakenteita, keskitason mallit käsittelevät rakennusten muotokieltä ja siellä tapahtuvia aktiviteetteja, ja pienimmät mallit käsittelevät materiaaleja ja rakenteita, joista rakennukset luodaan.</p>
Design Systems Handbook	<p>Suunnittelujärjestelmät auttavat tiimejä luomaan parempia tuotteita nopeammin tekemällä suunnittelusta uudelleenkäytettävää.</p> <p>Uudelleenkäytettävyys auttaa skaalautuvuudessa.</p> <p>Suunnittelujärjestelmä on kokoelma uudelleenkäytettäviä komponentteja, selkeiden standardien ohjaamana, ja niitä voidaan yhdistää rajattomalla eri tavalla</p> <p>Selkeät standardit poistavat subjektiivisuutta ja monitulkintaisuutta, jotka usein luovat jännitettä ja epätietoisuutta tuotetiimeissä; ilman standardeja päätöksistä tulee sattumanvaraisia ja vaikeita arvioida.</p>	<p>Suunnittelujärjestelmän komponentit ovat toisistaan riippuvaisia.</p> <p>Yhteen paikkaan tehty muutos periytyy läpi koko järjestelmän.</p> <p>Näin ollen esimerkiksi tyylilliset muutokset ovat hyvin helppoja toteuttaa, mutta saavuttavat silti suuren vaikutuksen.</p>

Näiden perusteella luotiin neljäs luku, jossa esitellään kirjallisuuskatsauksessa esiin nousseet havainnot.

4 Kirjallisuuskatsaus: suunnittelumallit ja mallikielet suunnittelun apuvälineinä

4.1 Suunnittelumallien määrittelyä

Christopher Alexander esittelee kirjassaan *The Timeless Way Of Building* syitä sille, miksi toiset paikat tuntuvat eloisilta ja toiset eivät. Lähtökohtana on hänen mukaansa ymmärtää, että jokainen paikka saa luonteensa siellä ilmenevistä toistuvista tapahtumista. Tapahtumat sekä tilojen geometriat muodostavat ne ”atomit ja molekyylit”, joista jokainen rakennus ja kaupunki on tehty. Näitä osasia kutsutaan kirjassa suunnittelumalleiksi. Suunnittelumallit voivat olla eloisia tai elottomia, ja mitä enemmän eloisia malleja rakennuksessa on, sitä enemmän se myös herää eloon kokonaisuutena. (Alexander 1979, ix–x.)

Design Systems -kirjassaan Alla Kholmatova kuvaa käyttöliittymäsuunnittelun näkökulmasta suunnittelumalleja toistuviksi ja uudelleenkäytettäviksi osiksi käyttöliittymää. Malleilla voidaan ratkaista suunnitteluongelmia, tyydyttää käyttäjän tarpeita tai herättää esiin tunnereaktio. Kholmatova rajaa suunnittelumallit tarkasti kahteen erilaiseen ryhmään: *toiminnallisiin* (engl. *functional*) ja *kuvaileviin* (engl. *perceptual*) malleihin. (Kholmatova 2017, x–xi.) Toiminnalliset mallit ovat konkreettisia ja niillä kuvataan käyttäytymistä ja toimintaa. Kuvailevat mallit sen sijaan kertovat, millaisia asiat ovat ja miltä ne tuntuvat. (Kholmatova 2017, 27, 89.)

David A. Carr, Åsa Granlund ja Daniel Lafrenière kertovat artikkelissaan *A Pattern Supported Approach to the User Interface Design Process*, että suunnittelumalli on yksi tapa muotoilla ratkaisuja suunnitteluongelmiin. Mallit ovat muodollisia kuvauksia konsepteista, joiden on todettu ratkaisevan käyttöliittymäsuunnittelussa ilmenneitä suunnitteluongelmia. (Carr, Granlund & Lafrenière 2001, 1.) Myös Lorraine J. Johnston ja Michael J. Mahemoff korostavat *Principles for A Usability-Oriented Pattern Language* -artikkelissaan suunnitteluongelmien merkitystä suunnittelumallien luomisessa. Heidän mukaansa suunnittelumalleilla voidaan huomioida toistuvia suunnitteluongelmia, analysoida niihin liittyviä vaikutteita sekä lopulta löytää näihin vaikutteisiin vastaavat suunnitteluratkaisut. (Johnston & Mahemoff 1998, 2.)

Carrin, Granlundin ja Lafrenièren (2001, 1) mukaan suunnittelumallien dokumentointi ei ole vaatimus niiden käyttämiselle, sillä suunnittelumallit voidaan jakaa implisiittisiin ja eksplisiittisiin malleihin. Implisiittiset mallit ovat sellaisia, joita käytetään päätelemällä

sopiva ratkaisu ongelmaan aiempien onnistuneiden kokemuksien pohjalta. Eksplisiittiset mallit ovat puolestaan jäsenneiltyjä ja dokumentoituja versioita tästä implisiittisestä tiedosta. (Carr ym. 2001, 1.)

Suunnittelumallit eivät välttämättä aina olekaan tietoisesti luotuja. Suunnittelumallien käytöstä ohjelmistotuotannossa kertovassa teoksessaan *Design Patterns: Elements of Reusable Object-Oriented Software* Gamma, Helm, Johnson ja Vlissides toteavat, että suunnittelumalleilla voidaan ilmentää erityisesti sellaisia ratkaisuja, jotka ovat kehittyneet ajan mittaan. Mallit tuovat esiin sellaista näkymätöntä työtä, jota on syntynyt, kun ohjelmistojen osista on pyritty tekemään monikäyttöisempiä ja joustavampia. Pitkällä aikavälillä tapahtuneen kehityksen vuoksi suunnittelumallit eivät useinkaan ole sellaisia, joita suunnittelijat osaisivat luoda intuitiivisesti. (Gamma, Helm, Johnson & Vlissides 1994, 10.)

4.2 Mallikielen määrittelyä

Alexanderin mukaan suunnittelumallit muodostavat mallikielen. Mallikielten avulla on mahdollista luoda rajattomasti erilaisia yksilöllisiä rakennuksia, jotka kuitenkin tuntuvat kuuluvan yhtenäiseen ryhmään. (Alexander 1979, 191.) Kaikki olemassa olevat rakennukset saavat sekä rakenteensa että eloisuutensa rakentajiensa käyttämisestä mallikielistä (Alexander 1979, 224). Alexanderin mukaan mallikielet koostuvat yksinkertaisimmillaan kahdesta joukosta: elementeistä ja symboleista sekä säännöistä, jotka yhdistävät näitä elementtejä. (Alexander 1979, 183–184.)

Myös Marco Suarez näkee artikkelissaan *Introducing design systems*, että mallikielen avulla suunnittelumalleja voidaan yhdistellä rajattomasti eri tavoilla. Hänen mukaansa mallikieli on kokoelma uudelleenkäytettäviä komponentteja, joita ohjaavat selkeät standardit. Standardit ovat ohjeita, joiden mukaan tiimi työskentelee. Standardit ohjaavat komponenttien tarkoitusta, tyyliä ja käyttötapoja, ja ne poistavat subjektiivisuutta ja monitulkintaisuutta, jotka usein luovat jännitettä ja epätietoisuutta tuotetiimeissä. Ilman ohjaavia standardeja komponentteihin liittyvistä päätöksistä voi tulla sattumanvaraisia ja vaikeita arvioida jälkikäteen. (Suarez 2017.)

Kholmatovan (2017, xi) mukaan mallikielet yhdistävät suunnittelumallien kaksi tyyppiä – toiminnalliset ja kuvailevat mallit – yhtenäiseksi kokonaisuudeksi. Mallikielen ominaisuuksiin kuuluu, että suunnitteluosaaminen on hajautettua sen sijaan, että ymmärrys mallikielestä olisi keskitetty yksittäisille ihmisille (Kholmatova 2017, 158).

Johnstonin ja Mahemoffin mukaan mallikieli syntyy, kun rajattu yhdistelmä suunnittelumalleja järjestetään yhtenäiseksi rakenteeksi. Tässä rakenteessa ylemmän tason suunnittelumallit tuottavat yhteyksiä, jotka puolestaan johtavat uusiin vaikutteisiin, jotka ratkaistaan jälleen uusilla yksityiskohtaisemmilla malleilla. Mallikielten päähyöty on niiden kyvyssä luoda uutta. Tällä tarkoitetaan sitä, että suunnittelija voi aloittaa työnsä yksittäiseen kontekstiin liittyvästä suunnittelumallista ja jatkaa siihen liittyvien yhteyksien kautta muihin suunnittelumalleihin, mikä voi helpottaa suunnittelijaa uuden luonnissa. Johnston ja Mahemoff toteavat myös, että jotta mallikieli pysyy koherenttina, sen tulisi pohjautua määriteltyihin periaatteisiin. Suunnittelumallit kanavoivat näitä periaatteita, kun malleilla vastataan suunnitteluongelmista nouseviin vaikutteisiin kyseisten periaatteiden mukaan. (Johnston & Mahemoff 1998, 2.)

Katie Saylor-Millerin *Building your design system* -artikkelin mukaan digitaalisille tuotteille luotu mallikieli noudattaa onnistuessaan seuraavaa viittä periaatetta (Saylor-Miller 2017):

1. Johdonmukaisuus. Mallikielellä on selkeästi määritellyt ja dokumentoidut säännöt, joita kaikki mallikielen parissa työskentelevät myös noudattavat. Hyvät säännöt mahdollistavat mallikielen käyttämisen ja sen kehittämisen.
2. Itsenäisyys. Mallikieli on oma kokonaisuutensa erillään tuotteista, jotka sitä käyttävät. Tällöin mallikielen sisältämät suunnittelumallit eivät ole koskaan sidottuja vain yhteen käyttötapaan.
3. Uudelleenkäytettävyys. Mallikielen suunnittelumallit ovat modulaarisia, koostettavia, yleistettyjä ja joustavia.
4. Saavutettavuus. Mallikielellä toteutetut tuotteet ovat mahdollisimman monen ihmisen käytettävissä.
5. Vakaus. Mallikielen muuttuminen ei vaikuta yllättävillä tavoilla sitä käyttäviin tuotteisiin.

Suarezin mukaan mallikielet mahdollistavat suunnittelun uudelleenkäytettävyyden, mikä puolestaan auttaa suunnittelun skaalaamisessa sellaisessa tilanteessa, jossa suunnitella joudutaan tekemään aina vain enemmän ja nopeammin. Mallikielet voivat auttaa luomaan parempia tuotteita nopeammin. (Suarez 2017.)

Gamma ym. argumentoivat, ettei pelkkä suunnittelumallien luettelointi yksin vielä tee kyseisestä luettelosta mallikieltä, sillä mallikielen määritelmän täyttääkseen luettelon tulisi tarjota sellainen kokoelma malleja, joilla voitaisiin luoda mikä tahansa ohjelmisto. He toteavatkin, ettei tällaisen mallikielen luominen ole välttämättä koskaan mahdollista, etenkin ohjelmistotuotannon alalla. (Gamma ym. 1994, 329.) Suunnittelumallien luettelointi on joka tapauksessa tärkeää, sillä se antaa käytetyille suunnittelutekniikoille standardisoidut nimitykset ja määritelmät (Gamma ym. 1994, 324).

4.3 Mallikielen hierarkkisuus

Alexanderin (1979, 309) mukaan arkkitehtuurin alalla suunnittelumallien tulee kuvata kattavasti jokainen ympäristömme taso. Tässä tapauksessa suurimman mittakaavan suunnittelumallit käsittelevät alueellisia rakenteita, keskitason mallit rakennusten muotokieltä ja niissä tapahtuvaa toimintaa ja pienimmät mallit materiaaleja ja rakenteita, joista rakennukset tehdään.

Niin ikään Brad Frostin käyttöliittymäsuunnitteluun keskittyvässä *Atomic Design* -kirjassa korostuu hierarkkisuus Frostin atomisessa suunnittelumetodologiassa. Frostin metodologiassa suunnittelumallit muodostavat viisi tasoa, jotka toimivat yhteen siten, että niistä voidaan harkiten rakentaa suunnittelujärjestelmä. (Frost 2016, 42.) Suunnittelumetodologian kuvaamat tasot ovat (Frost 2016, 43–52):

1. *Atomit*: Metodologian pienimpiä osasia, joita ei voi pilkkoa enää pienemmiksi ilman, että niiden toiminnallisuus tai käyttötarkoitus kärsisi.
2. *Molekyylit*: Ryhmiä, jotka koostuvat atomeista ja antavat tarkoituksen atomien olemassaololle.
3. *Organismit*: Koostuvat atomeista, molekyyleistä tai muista organismeista. Organismit ovat tässä tapauksessa esimerkiksi rajattuja osia käyttöliittymästä.
4. *Mallit*: Sivun tai muun näkymän kokoisia osia, jotka koostuvat atomeista, molekyyleistä ja organismeista ja ilmaisevat suunniteltua sisältörakennetta.
5. *Sivut*: Koostuvat malleista, mutta joissa sisältö mukailee todellista.

Myös Carr ym. sijoittavat suunnittelumallit eri tasoille, joista kukin kuvaa eri vaihetta käyttöliittymän suunnitteluprosessissa. Jokaisella tasolla on suhde seuraavaan tasoon ja päinvastoin, jotta tason malleissa olevaa tietoa säilyy ja välittyy tasolta toiselle. (Carr ym. 2001, 2–3.) Yksittäisessä suunnittelumallissa voidaan siis viitata sekä edellisen tason malleissa tehtyihin havaintoihin että antaa ratkaisuvaihtoehtoja seuraavan tason malleista, jolloin suunnittelumalleista muodostuu ikään kuin toisiinsa liittyvä ketju. Carr ym. (2001, 3) esittelevät lähestymistapansa suunnittelumallien tasoihin käyttöliittymäsuunnittelun näkökulmasta seuraavasti:

1. *Liiketoiminta-aluemallit* kuvailevat liiketoiminnan tyyppisiä ja tavoitteita. Nämä mallit toimivat hyvänä lähtökohtana mallikielen suunnittelulle.
2. *Liiketoimintaprosessimallit* kertovat edellisen tason mallien tavoitteita mukaillen, millaisia ovat liiketoiminnan tyypilliset prosessit ja toimijat.
3. *Tehtävämalleilla* voidaan välittää tietoa tehtävästä, sen tyypillisistä käyttäjistä sekä mahdollisista vuorovaikutustapojen suunnitteluratkaisuista tehtävään liittyen. Monimutkaiset tehtävät jakaantuvat pienempiin alitehtävämalleihin, jotka helpottavat monimutkaisen tehtävän kuvaamista.
4. *Rakenne- ja navigaatiomallit* kuvaavat tapoja järjestää tietoa ja tietojen välillä siirtymistä, jotta käyttäjän tehtävän vaatimukset täyttyvät. Mallien rakenne pohjautuu tehtävämallien kuvauksiin.
5. *Käyttöliittymän suunnittelumallit* dokumentoivat tehtävämallien sekä rakenne- ja navigaatiomallien pohjalta käyttöliittymään liittyviä suunnitteluongelmia ja niiden mahdollisia ratkaisuja.

Suarez toteaa, että kaikkien digitaalisten tuotteiden mallikielten suunnittelumallien tulee olla toisistaan riippuvaisia. Tämä mahdollistaa sen, että yhteen paikkaan tehty muutos periytyy läpi järjestelmän. Näin ollen esimerkiksi tyylilliset muutokset voivat olla hyvinkin helppoja toteuttaa kuitenkin niin, että muutoksen vaikutus on suuri. (Suarez 2017.)

4.4 Suunnittelumallin rakenne

The Timeless Way Of Building -teoksessa suunnittelumallien rakennetta määrittää mallien ja mallikielen jaettavuus eli se, kuinka helppoa tiedon jakaminen on muille ihmisille.

Jotta mallikieli voisi olla jaettava, kielen yksittäisten mallien tulisi olla samaan aikaan sekä tarpeeksi syvällisiä että tarpeeksi yksinkertaisia. Mikäli malli ei ole tarpeeksi yksinkertainen, se ei voi selvitä suusanallisesta jakamisesta muuttumatta. (Alexander 1979, 229–230.)

Alexanderin (1979, 253) mukaan jokaisen suunnittelumallin tulee olla sääntö, joka koostuu kolmesta osasta ja niiden välisistä suhteista. Yleisessä muodossaan tämä sääntö on yhteys tietyn kontekstin, kontekstissa olevan ongelman ja ongelman selvittävän ratkaisun välillä (Alexander 1979, 247). Kun nämä suhteet on löydetty, ne pitää muotoilla uudelleen itsenäiseksi kokonaisuudeksi, jotta ne olisivat sellaisenaan käyttökelpoisia ja jotta ihmismieli voisi käsitellä niitä helposti. Samasta syystä suunnittelumalli tulee pystyä piirtämään kaaviokuvana ja se pitää nimetä. Suunnittelumalli on mahdollista jakaa muille ihmisille, kun nämä vaatimukset toteutuvat. (Alexander 1979, 265–267.)

Myös Kholmatovan *Design Systems* -teoksessa peräänkuulutetaan suunnittelumallien hyvää nimeämistä. Jos käyttöliittymämallilla ei ole sellaista nimeä, joka on yleisesti tunnettu ja jonka kaikki mallikieltä käyttävät ymmärtävät, ei sellaista suunnittelumallia voi käyttää osana mallikieltä. Lisäksi mikäli suunnittelumallille annettava nimi on liian kuvai-leva, kyseinen nimi voi Kholmatovan mukaan rajoittaa mallin tulevaa kehitystä: esimerkiksi ”vaaleanpunainen painike” ei voi tulevaisuudessa olla väriltään sininen. (Kholmatova 2017, 109.)

Kirjallisuuskatsauksen aineistossa kolme julkaisua antaa tarkan rakenteen suunnittelumalleille. *Design Systems* -kirjassa Kholmatova painottaa, että suunnittelumallin dokumentaation tulee keskittyä rajattuun määrään asioita, jotta se olisi helppolukuinen. Kholmatovan mukaan suunnittelumallin dokumentaation tulee kertoa vähintään suunnittelumallin nimi, tarkoitus, esimerkkejä sen käyttämisestä sekä mallin mahdolliset muunnelmat. (Kholmatova 2017, 223.)

Carr ym. (2001, 3–4) määrittävät *A Pattern Supported Approach to the User Interface Design Process* -artikkelissa, että käyttöliittymän tehtävää kuvaavan suunnittelumallin tulee sisältää mallin nimi, konteksti, jossa sitä käytetään, suunnitteluongelma, jonka malli ratkaisee, esimerkki, vaikutteet, jotka vaikuttavat suunnitteluun sisäisesti tai ulkoisesti, suunnitteluratkaisu, joka vastaa määritettyihin vaikutteisiin sekä viittaukset mahdollisiin alitehtävä-, rakenne-, navigaatio- ja käyttöliittymämalleihin.

Gamma ym. kuvaavat mallien rakennetta pitkälti samoin kuin Carr ym. artikkelissaan, mutta painottavat, että rakenteen tulee noudattaa samaa järjestystä jokaisen mallin kohdalla, jotta mallit ovat helpommin käytettävissä, vertailtavissa ja opeteltavissa. Gamma ym. mukaan suunnittelumallin rakenteeseen kuuluu suunnittelumallin nimi, ongelma, jonka ratkaisemiseksi malli on luotu, muut mahdolliset nimet mallille, motivaatio, joka kuvaa yleisen skenaarion suunnitteluongelmasta, mallin sovellettavuus eri tilanteissa, mallin varsinainen rakenne kaaviokuvana, malliin liittyvät koodiosat, koodiosien keskinäiset suhteet, mallin käytöstä aiheutuvat seuraukset ja kompromissit, mallin toteuttamiseen liittyvät tarkemmat tekniikat ja ohjeet toteutukseen, esimerkkikoodia, tunnetut todelliset toteutukset sekä lopuksi malliin liittyvät muut suunnittelumallit. (Gamma ym. 1994, 19.)

4.5 Suunnittelumallien ja mallikielten kehittyminen

Alexanderin (1979, 340) mukaan mallikielien ja niissä olevat mallit kehittyvät luonnostaan, kun eri alueilla ja eri ihmisillä on erilaisia mallikieliä ja erilaisia versioita mallikielissä olevista suunnittelumalleista. Tämä johtaa ennen pitkää siihen, että kun erilaisia mallikieliä jaetaan ihmisten kesken, kielet kehittyvät itsestään ja niistä häviää huonoja malleja ja tilalle tulee parempia. Kehitys on nimenomaan mahdollista siksi, että yksittäiset mallit ovat kielessä itsenäisiä ja siksi kehitystä voi tapahtua asteittaisesti, yksi suunnittelumalli kerrallaan. Vaikka mallikielen kehittyminen tekeekin mallikielystä jatkuvasti eheämmän, täydellistä ja lopulta muuttumatonta mallikieltä ei kuitenkaan ole mahdollista luoda, sillä mallikielien jatkavat jatkuvasti kehittymistään luonnollisesti. (Alexander 1979, 344–346.)

Frost korostaa tuotteen ja mallikielen välistä yhteyttä. Frostin (2016, 155) mukaan suunnittelujärjestelmän tulee kehittyä tarpeen vaatiessa, mukautua palautteen myötä, kehittyä testaamisen kautta sekä kehittyä sen tuotteen mukana, jota mallikieli tukee. Myös Kholmatova (2017, x) painottaa, ettei mallikieltä voi luoda hetkessä, vaan sitä tulee aina kehittää vähitellen yhdessä tuotteen kanssa. Mallikielen suunnittelumallit kehittyvät, kun niitä testataan tuotteessa ja muokataan testaamisen pohjalta, jotta ne täyttäisivät paremmin tarkoituksensa (Kholmatova 2017, 64). Kehityksen kannalta onkin oleellista, että suunnittelumalleilla on selkeästi kuvattu tarkoitus olemassaoloonsa. Kun on tietoinen etenkin mallikielen ydinmallien osalta niiden tarkoituksista ja tavoitteista, on helpompaa ymmärtää koko mallikielen toimintaa ja sitä kautta estää kielen sirpaloitumista sen kehityksessä (Kholmatova 2017, 68).

Jessop varoittaa artikkelissaan *Pattern language: A framework for learning*, ettei mallikieli kehity aluksi kovinkaan nopeasti. Etenkin ensimmäisten suunnittelumallien luominen on hyvin aikaa vievää, sillä samalla tulee opetella pitämään mallit lyhyinä ja relevantteina sekä kanavoimaan muihin kuin kyseessä olevaan malliin liittyvät keskustelut uusien suunnittelumallien luomiseen. Ajan myötä prosessista tulee nopeampi, mutta siltikään vaadittava aika mallikielen valmiiksi saamiseen tai sen tarkastamiseen ei ole koskaan erityisen pieni. (Jessop 2004, 464.)

Jessopin mukaan on oleellista, että ne henkilöt, jotka suunnittelumalleja käyttävät, myös luovat mallikielen. Tällöin on kuitenkin mahdollista, että kyseiset henkilöt ovat liian lähellä itse prosessia, jolloin heidän ennakoasenteensa voivat liiaksi vaikuttaa myös lopputulokseen. Ulkopuolinen fasilitaattori voi auttaa pienentämään ennakoasenteiden vaikutusta, mutta fasilitaattorin käyttäminen voi kasvattaa suunnitteluprosessiin käytettävää aikaa. (Jessop 2004, 464.)

4.6 Suunnittelumallien ja mallikielten vaikutus ihmisten väliseen yhteistyöhön

Tässä luvussa esitellään kirjallisuuskatsauksessa esiin nousseita merkityksiä, joita suunnittelumalleilla ja mallikielillä on ihmisten väliselle yhteistyölle. Luvun alkupuolella kerrotaan, kuinka tiimien kommunikaatio muuttuu mallikielten myötä. Luvun lopuksi avataan erilaisia tapoja ylläpitää mallikieliä erilaisissa organisaatioissa.

Alexanderin mukaan mallikielen pitää olla ihmisten välistä yhteistyötä parantava ohjeisto, jonka avulla ihmisten yksittäiset teot muodostavat yhden eheän kokonaisuuden. Ohjeiston tulee olla niin selkeä, että kaikki yhteisön jäsenet voivat osallistua yhteisön kehittämiseen: kenen tahansa pitää pystyä muokkaamaan omaa pientä osaansa yhteisöstä niin, että jokainen osa on yksilöllinen, mutta kuitenkin siten, että tästä syntyvä kokonaisuus on myös itsenään eheä. (Alexander 1979, 166.)

Alexander (1979, 229) toteaa myös, että kaupungeissa, joissa on eloisa mallikieli, tämä kieli on niin laajasti jaettu ja tunnettu, että kuka tahansa kaupungissa voi käyttää sitä. Mallikielet voivat myös kuolla, mikäli niistä tulee yksityisiä ja erikoistuneita laajan jakamisen ja tunnettuuden sijaan (Alexander 1979, 231–232). Erikoistumisella tarkoitetaan, että yksittäistä mallia tai mallikieltä osaa käyttää vain siihen erikoistunut henkilö eivätkä ohjeistot ole yleisesti tiedossa. Erikoistuvat mallikielet muuttuvat erikoisalojen sisällä ajan mittaan aina vain yksityisemmiksi, ja tämä aiheuttaa lopulta myös mallikielten sirpaloitumista (Alexander 1979, 232).

Alexanderin ajatusta mukailleen myös Kholmatova huomioi, että yhteinen kieli voi auttaa ihmisiä luomaan tuotteita, jotka tuntuvat eheiltä, vaikka eri ihmiset olisivat luomassa tuotteen eri osia. Toiset voivat osata käyttää kieltä paremmin kuin toiset, mutta yleinen ajatus on, että kaikilla tiimin jäsenillä olisi jonkin asteinen käsitys kielestä ja että tämä käsitys paranisi jatkuvasti, kun tiimi oppii, käyttää ja kehittää kieltä. (Kholmatova 2017, 108.)

Kholmatovan mukaan yhteinen kieli onkin perustavanlaatuinen vaatimus yhteistyölle. Tämän vuoksi myös mallikielen tulee olla jaettu koko tuotteen parissa työskentelevän tiimin kesken. Ilman yhteistä kieltä jokaisella tiimin jäsenellä voi olla lopulta erilainen käsitys siitä, mitä tuotteella yritetään saavuttaa. Edes yhteinen kieli ei kuitenkaan välttämättä riitä, sillä vaikka joskus kuvitellaan yhteisymmärryksen löytyneen ja tiimi käyttäisi-kin samaa mallikieltä, tiimin jäsenillä saattaa olla huomattavia eroja mallien todellisessa ymmärryksessä. Kholmatova esittää, että tämänkaltaiset tulkinalliset ongelmat voidaan välttää, kun jokaisen suunnittelumallin yhteydessä mainitaan myös, kuinka, miksi ja millaisissa konteksteissa kyseistä mallia käytetään ja minkälaisia tarkoituksia mallilla voi olla. (Kholmatova 2017, 28–29.)

Gamma ym. toistavat suunnittelumallien tarjoaman yhteisen kielen hyödyn. Yhteisellä kielellä on mielekästä kommunikoida, dokumentoida sekä tutkia erilaisia suunnitteluvaihtoehtoja. Suunnittelumallien luoma yhteinen kieli myös hälventää kuvaamansa järjestelmän kompleksisuutta, kun järjestelmän konsepteista voidaan puhua yleisemmällä tasolla kuin vain yksittäisestä suunnitteluratkaisusta. Tällä tavoin järjestelmästä tulee helpompi ymmärtää, kun koko suunnittelumallia ei tarvitse kuvailla auki, vaan riittää, että käytetään mallin nimeä. Vaikka aluksi henkilön, joka ei tunne käytössä olevia suunnittelumalleja, pitää etsiä mallien kuvaukset, malleista yleisellä tasolla puhuminen säästää silti aikaa. (Gamma ym. 1994, 324–325.)

Frostin mukaan mallikieli luo perustuksen johdonmukaiselle ja jaetulle sanastolle, joka kannustaa tieteenhaarojen väliseen yhteistyöhön sekä estää viestinnän epäonnistumista. Frost korostaa mallikielen merkitystä etenkin silloin, kun halutaan saada eri sidosryhmät ymmärtämään, millaiset asiat vaikuttavat lopputuotteen taustalla. (Frost 2016, 31.) Mallikielen onkin mahdollisuus toimia koko organisaation hyväksi, sillä se luo yhteisen sanaston jokaista organisaation tuotteiden parissa työskentelevää tieteenalaa varten (Frost 2016, 165).

Myös Cline eriyttää artikkelissaan *The Pros and Cons of Adopting and Applying Design Patterns in the Real World* suunnittelumallien hyödyiksi vakioidun sanaston luomisen

suunnittelulle koko yhteisössä sekä kommunikaation paranemisen, kun suunnittelusta voidaan puhua ylätasolla kaikkien tuotteen parissa työskentelevien osapuolien välillä. Suunnittelumallit ovat erityisen tehokkaita dokumentaatiotyökaluna, kun tiimi ottaa vastaan uusia jäseniä. Mallien avulla tiimin suunnittelufilosofiat kirjoitetaan selvästi auki sen sijaan, että ne olisivat vain alkuperäisen tiimin mielessä. (Cline 1996, 47.)

Kholmatova esittää yhteistyön näkökulmasta mallikielen ylläpitoon kaksi lähestymistapaa: keskitetyn ja hajautetun mallin. Keskitetyn mallin pääajatuksena on, että mallikielen suunnittelumallit ja säännöt ovat pääasiassa yhden ryhmän ylläpitämiä. Tämä tarkoittaa yleensä sitä, että tämä ryhmä määrittelee kaikki mallikielen suunnittelumallit ja säännöt, pitää kehitysprosessissa hallussaan veto-oikeutta sekä ylläpitää mallikielen mallien keräämistä varten luotua suunnittelumallikirjastoa tai muuta tietokantaa. (Kholmatova 2017, 157.) Vaihtoehtoisena tapana on hajautettu malli, jossa kaikki mallikieltä käyttävät ovat myös vastuussa sen ylläpitämisestä ja kehittämisestä (Kholmatova 2017, 157). Tämä malli on siten verrattavissa Alexanderin näkemykseen mallikielen jakamisesta. Kholmatovan (2017, 157) mukaan hajautettu malli tarjoaa yksilöille vapautta sekä oikeuden tehdä päätöksiä. Näin ylläpidetty mallikieli on usein ketterämpi ja kestävämpi: mikäli jokin tiimi ei pysty toteuttamaan jotain, toinen tiimi voi jatkaa. Näin ollen suunnittelutietous ja päätökset luovasta suunnasta ovat hajautettua sen sijaan, että ne olisivat vain tiettyjen ihmisten takana. (Kholmatova 2017, 157–158.)

Jina Annen *Designing your design system* -artikkelin mukaan keskitetyssä mallissa mallikieli pysyy hyvin ylläpidettynä, kun yksi tiimi hallitsee koko suunnittelujärjestelmää kokoaikaisena työnään. Haittana voi kuitenkin olla, että tiimi on irrallaan loppukäyttäjien tarpeista, kun tiimin jäsenet eivät välttämättä ole mukana loppukäyttäjille tehtävissä käyttäjätutkimuksissa. Hajautetussa mallissa sen sijaan tiimillä on usein hyvä näkemys siitä, mitä tuote ja käyttäjät vaativat, mutta tiimi saattaa olla kiireinen työskennellessään jo tuotekehityksessä. Anne esittelee kolmannen mallin, joka pyrkii välttämään keskitetyn ja hajautetun mallin haitat. Tässä yhdistetyssä mallissa yksi ydintiimi hallitsee mallikielen kehitystä, mutta myös muut tiimit voivat tarvittaessa osallistua kehitykseen esittämällä pyyntöjä ja uusia ajatuksia. (Anne 2017.)

Myös Frost esittelee kaksi lähestymistapaa mallikielten ylläpitämiseen yhteistyön näkökulmasta, mutta jaottelee nämä tavat organisaatioiden resurssien ja mallikielen käytön laajuuden mukaan. Frostin (2016, 152) mukaan suuret organisaatiot pystyvät käyttämään mallikielen ylläpitämiseen paljon resursseja, joten kun mallikieli palvelee isoa mää-

rää käyttäjiä, on järkevää varata muutama työntekijä mallikielen kehitykseen ja ylläpitoon. Pienemmissä organisaatioissa puolestaan oman tiimin varaaminen mallikielelle ei yleensä ole mahdollista. Yksittäinen organisaation työntekijä saattaakin sekä kehittää mallikieltä sekä suunnittelumalleja käyttäviä tuotteita. (Frost 2016, 152.) Frost ei ota Kholmatovan tapaan varsinaisesti kantaa siihen, minkälaisella mallilla mallikieltä kannattaa kehittää, vaan keskittyy lähinnä ulkoisiin tekijöihin, kuten organisaation kokoon. Frost (2016, 165) toteaa kuitenkin, että tehokkaan mallikielen luomisen edellytyksenä on ihmisten välinen yhteistyö ja kommunikaatio.

4.7 Haasteita suunnittelumalleissa

Kholmatovan mukaan mallikieli yhdistetään usein nimenomaan parempaan yhteistyöhön. Lopputulos voi kuitenkin olla, että vain pieni määrä ihmisiä käyttää kieltä tai tiimien välisen kommunikaation puutteen vuoksi suunnittelumallit saattavat olla toisistaan irrallisia tai hajanaisia. Kholmatova muistuttaakin, ettei pelkkä mallikieli vielä yksin takaa yhtenäistä ja johdonmukaista suunnittelua, sillä mallikieli on lopulta vain työkalu ja työkalulla itsellään on hyvin vähän vaikutusta loppukäyttäjän kokemukseen tuotteesta, jota mallikieli palvelee. (Kholmatova 2017, 35–36.) Myös Carr ym. (2001, 2) korostaa, ettei suunnittelumalleilla voida koskaan korvata perinteistä suunnitteluosaamista, vaan mallit toimivat vain täydentävänä lähteenä.

Kholmatova nostaa esiin neljä mielestään merkittävintä haastetta mallikielten kehityksessä. Ensinnäkin uudelleenkäytettävien mallien luominen vie lähtökohtaisesti enemmän aikaa kuin kertakäyttöisten ratkaisujen kehittäminen. Tämä ajankäyttö maksaa itsensä takaisin vasta siinä vaiheessa, kun malleja todellisuudessa käytetään toistuvasti uudelleen. Toiseksi koska suunnittelumallien tulee olla yleisluonteisia, jotta ne voivat toimia monessa tarkoituksessa, lopputuloksena voi myös olla vain tehokkuutta korostavaa geneeristä suunnittelua. Kolmanneksi kun modulaarisuuteen on panostettu, suunnittelumallien uudelleenkäyttöä yritetään joskus pakottaa. Tämän vuoksi modulaarisuuden ja suunnittelumallien tuomat hyödyt tulisi aina tasapainottaa tehokkuuden kanssa. Lopuksi Kholmatova nostaa yhdeksi suurimmista haasteista suunnittelumallien saamisen toistensa kanssa yhteensopiviksi. Suunnittelumallien käyttämisen odotuksena usein on, että modulaarisen suunnittelun myötä kaikkia mallikielen osia voi käyttää keskenään ristiin. Tämä voi johtaa paradoksaalisesti siihen, että suunnittelumallien välillä on paljon johdonmukaisuutta, mutta kokonaisuus ei silti ole yhtenäinen. Sen vuoksi malleja suunniteltaessa on suunniteltava myös sen yhteys muihin malleihin: kuinka ne liittyvät toisiinsa,

millainen on niiden välinen hierarkia ja mikä on kunkin mallin rooli kokonaisuudessa. (Kholmatova 2017, 154–155.)

Myös Frost (2016, 34) näkee haasteita suunnittelumallien luomisen vaatimassa ajassa. Mallikielen luominen vaatii organisaatiolta tarvittavia ajallisia ja rahallisia resursseja, jotta mallikielestä saataisiin irti sen tarjoama todellinen hyöty. Koska mallikieli vaatii pitkäjänteisyyttä, organisaation on päästävä eroon kulttuurinsa mahdollisesta lyhytnäköisyydestä. Siksi mallikieltä ei tulisi kohdella vain apukirjastona, vaan osana lopputuotetta. Muussa tapauksessa kielen kehitys voi olla ensimmäisiä karsittavia asioita, kun resursseja joudutaan supistamaan. (Frost 2016, 34–35.)

Frost huomauttaa lisäksi myös, että jos mallikielten ymmärretään olevan vain tiettyjen organisaation ryhmien työkalu tai jos mallikielen luonnissa ei huomioida laajempaa yleisöä, kielistä saattaa tulla liian epämääräisiä tai tiettyyn asiaan keskittyneitä. Se puolestaan vähentää mallikielen näkyvyyttä organisaatiossa ja rajoittaa sen mahdollista vaikutavuutta. (Frost 2016, 35.)

5 Kirjallisuuskatsauksen yhteenveto

Jatkuvasti monimutkaistuvassa digitaalisten tuotteiden suunnittelukentässä voi kokonaisuuden hallinta olla erittäin vaikeaa. Suunnittelumallit ja mallikielet voivat olla tähän eräs ratkaisu. Kuten tästä opinnäytetyöstä käy ilmi, suunnittelumallit ja mallikielet ovat konsepteina pysyneet hyvin samankaltaisina aina tähän päivään asti, sillä sama ajatus soveltuu hyvin käytettäväksi erilaisissa tilanteissa.

Mallikielen ja suunnittelumallien määritelmät ovat aineistossa pääpiirteittäin samanlaisia – Carr ym. (2001, 1), Gamma ym. (1994, 15), Jessop (2004, 458), Kholmatova (2017, 22) ja Johnston & Mahemoff (1998, 2) viittaavat määritelmässään jopa suoraan Alexanderin näkemykseen mallikielistä. Kaikissa aineiston julkaisuissa korostetaan mallikielten koostuvan suunnittelumalleista. Eroavaisuuksia on kuitenkin siinä, kuinka niiden ajatellaan syntyvän. Esimerkiksi Alexander toteaa, ettei mallikieltä varsinaisesti luoda, vaan se syntyy itsestään, kun sille annetaan oikeat olosuhteet (Alexander 1979, xi). Sekä Gamma ym. (1994, 10) että Carr ym. (2001, 1) toistavat tämän ajatuksen suunnittelumallien osalta, sillä heidän mukaansa suunnittelumallit ovat jo valmiiksi olemassa maailmassa, ne pitää vain dokumentoida. Tämä ajattelutapa korostaa siis konventioiden merkitystä suunnittelussa. Muut aineiston julkaisut eivät erityisesti korosta mallien syntymekanismeja, vaan niissä ajatellaan yleisesti, että tiettyyn suunnitteluongelmaan tulee

kehittää sopiva ratkaisu. Näen, että digitaalisten tuotteiden suunnittelun näkökulmasta tunnettujen konventioiden käyttäminen mallikieltä luotaessa on erityisen tärkeää, jotta käyttäjät voivat tutustua tuotteisiin nopeammin. Jotta mallikieli olisi helpompi ottaa käyttöön organisaatiossa, on niissä myös järkevää dokumentoida ennalta tuttuja toimintatapoja ja sitten kehittää niitä.

Vaikka suunnittelumallien ja mallikielten käyttö voivat vaatia organisaatiolta merkittävää sitoutumista, ne voivat kuitenkin maksaa itsensä takaisin parantuneen yhteistyön ja kokonaisuuden hallinnan kautta. Tämä on myös aineistoon valittujen julkaisujen pääsanoma. Hyötynä Alexander (1979, 229), Kholmatova (2017, 108; 2017, 28), Gamma ym. (1994, 324–325), Frost (2016, 31; 2016, 165) ja Cline (1996, 47) esittävät mallikielten tarjoavan vakioidun sanaston ihmisten väliseen kommunikointiin. Kholmatovan (2017, 28–29) mukaan sanaston pitää nimenomaan olla vakioitu, jotta vältetään väärinymmärryksiltä tiimityöskentelyssä. Yhteinen kieli auttaa eri taustoilta tulevia tiimin jäseniä kommunikoidaan keskenään. Mallikielillä voidaan näin helpottaa tiimin kasvattamista, joten myös organisaatioiden ja tuotteiden skaalattavuus paranee.

Merkittävimmäksi haasteeksi suunnittelumallien käyttöönotolle aineistossa nousee tilanne, jossa mallikieli luodaan, mutta se jää vain rajatun ryhmän käyttöön organisaatiossa. Tämän riskin nostaa esiin Alexander (1979, 232), Kholmatova (2017, 35–36) ja Frost (2016, 35). Kun mallikieli on vain lopulta tietyn ryhmän käytössä, ei sillä voida parantaa yhteistyötä organisaatiossa tai tiimeissä, ja lopulta yhteisen kielen käyttäminen sitä myöten myös loppuu.

6 Lopuksi

Koen, että tämä opinnäytetyö onnistuu luomaan yleiskuvaa suunnittelumalleista ja mallikielistä niin digitaalisten tuotteiden suunnittelussa kuin suunnittelussa yleisestikin. Luokija voi saada työstä ohjeita oman mallikielen tai suunnittelujärjestelmän luomiseen. Työ ei kuitenkaan anna valmista pohjaa mallikielelle, vaan esittelee erilaisista näkökulmista joitain keskeisiä vaatimuksia suunnittelumalleille sekä tiettyjen ratkaisujen hyötyjä ja haittoja.

Kirjallisuuskatsauksen tekemisen haasteena oli aineistoa rajatessa määritellä, miten aineistosta saataisiin tarpeeksi kattava yleiskuvan muodostamiseksi mutta kuitenkin sopivan kompakti opinnäytetyön laajuus huomioiden. Digitaalisten tuotteiden suunnitteluun keskittyvien julkaisujen vastapainoksi halusin mukaan vertailukohtaksi ja eräänlaiseksi

ankkuriteokseksi Alexanderin *The Timeless Way Of Building* -kirjan (1979). Lisäksi valitsin mukaan vielä operaatioanalyysin alalta Jessopin (2004) artikkelin. Jälkikäteen keskittyisin kuitenkin vain yhden tieteenalan julkaisuihin, jotta aineisto olisi yhtenäisempi ja aineiston julkaisuja olisi helpompi vertailla keskenään.

Opinnäytetyössä oli alun perin tarkoituksena kartoittaa suunnittelujärjestelmien nykytilaa digitaalisia tuotteita suunnittelevissa organisaatioissa sekä antaa pohja hyvälle suunnittelujärjestelmälle. Työn aineisto oli tarkoitus kerätä puolistrukturoitujen haastatteluiden avulla organisaatioista, jotka käyttävät tai ovat suunnitelleet ottavansa käyttöön suunnittelujärjestelmiä. Haasteeksi osoittautui kuitenkin haastateltavien pieni määrä, sillä suunnittelujärjestelmät ovat edelleen nähdäkseni jokseenkin harvinaisia tai ne eivät ole julkisia. Näin ollen aineistoa ei olisi ollut saatavilla tarpeeksi varsinaista opinnäytetyötä varten.

Koska digitaalisten tuotteiden suunnittelu ei nykytilasta todennäköisesti ainakaan yksinkertaistu, mahdollista jatkotutkimusta kannattaisi kohdistaa mallikielien ja suunnittelujärjestelmien kehityksen tutkimiseen ja helpottamiseen. Koska nähdäkseni sekä mallikielissä että suunnittelujärjestelmissä yksi haastavimmista vaiheista on niiden jalkauttaminen organisaation käyttöön, olisi myös sitä hyödyllistä tutkia. Lisäksi tulisi pyrkiä kehittämään ratkaisuja, joilla suunnittelujärjestelmistä ja mallikielistä tulisi helpommin keskeinen osa organisaation suunnittelutyötä.

Lähteet

Alexander, Christopher 1979. The Timeless Way of Building. Iso-Britannia: Oxford University Press.

Alexander, Christopher, Ishikawa, Sara & Silverstein, Murray 1977. A pattern language: towns, buildings, construction. Iso-Britannia: Oxford University Press.

Anne, Jina 2017. Designing your design system. Marco Suarez, Jina Anne, Katie Saylor-Miller, Diana Mounter, Roy Stanfield (toim.): Design Systems Handbook. New York: Invision. <<https://www.designbetter.co/design-systems-handbook>> (luettu 11.4.2018).

Aßmann, Uwe 2013. Invasive Software Composition. Berliini: Springer Science+Business Media.

Baldwin, Carliss & Clark, Kim 2000. Design Rules: The power of modularity. Cambridge: MIT Press.

Baumeister, Roy & Leary, Mark 1997. Writing Narrative Literature Reviews. Review of General Psychology, 1(3). Washington: American Psychology Association. 311–320.

Bearfield, Domonic & Eller, Warren 2007. Writing a Literature Review: The Art of Scientific Literature. Miller, Gerald, Yang, Kaifeng (toim.): Handbook of Research Methods in Public Administration. Boca Raton: CRC Press. 61–72.

Carr, David A., Granlund, Åsa & Lafrenière, Daniel 2001. A Pattern Supported Approach to the User Interface Design Process. Luettavissa osoitteessa <<http://cieteseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.447.4952&rep=rep1&type=pdf>> (luettu 4.4.2019).

Cline, Marshall P. 1996. The Pros and Cons of Adopting and Applying Design Patterns in the Real World. Communications of the ACM, 39(10). New York: Association for Computing Machinery. 47–49.

Frost, Brad 2016. Atomic Design. Pittsburgh: Brad Frost.

Future-Friendly Manifesto 2011. <<http://futurefriendlyweb.com/>> (luettu 11.4.2018).

Gamma, Erich, Helm, Richard, Johnson, Ralph & Vlissides, John 1994. Design Patterns: Elements of Reusable Object-Oriented Software. 37. painos. Boston: Addison-Wesley.

Goldman, Ben, Avore, Chris, Bridewell, Drew, Campbell, Emily, Godfrey, Andrew, Halberstam, Ehud, Suarez, Marco, Walter, Aaron (toim.) 2018. DSM Field Guide. New York: InVision.

Jessop, Alan 2004. Pattern language: A framework for learning. European Journal of Operational Research, 153(2). Amsterdam: Elsevier B.V. 457–465.

Johnston, Lorraine J. & Mahemoff, Michael J. 1998. Principles for a usability-oriented pattern language. 1998 Australasian Computer Human Interaction Conference, OzCHI '98: proceedings, November 30-December 4, 1998, Adelaide, South Australia. Washington: IEEE Computer Society.

Kholmatova, Alla 2017. Design Systems. Freiburg: Smashing Media.

Naur, Pieter & Randell, Brian 1969. Software Engineering – Report on a conference sponsored by the NATO Science Committee. Luettavissa osoitteessa <<http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF>> (luettu 22.3.2019).

Russell, Andrew 2012. Modularity: An Interdisciplinary History of an Ordering Concept. Information & Culture. 47 (3), 257–287. Luettavissa osoitteessa <<http://arussell.org/papers/47.3.russell.pdf>> (luettu 22.3.2019).

Salminen, Ari 2011. Mikä kirjallisuuskatsaus? Johdatus kirjallisuuskatsauksen tyyppeihin ja hallintotieteellisiin sovelluksiin. Vaasa: Vaasan yliopiston julkaisuja. Luettavissa osoitteessa <https://www.univaasa.fi/materiaali/pdf/isbn_978-952-476-349-3.pdf> (luettu 6.3.2019).

Suarez, Marco 2017. Introducing design systems. Marco Suarez, Jina Anne, Katie Saylor-Miller, Diana Mounter, Roy Stanfield (toim.): Design Systems Handbook. New York: Invision. <<https://www.designbetter.co/design-systems-handbook>> (luettu 11.4.2018).

Saylor-Miller, Katie 2017. Building your design system. Marco Suarez, Jina Anne, Katie Saylor-Miller, Diana Mounter, Roy Stanfield (toim.): Design Systems Handbook. New York: Invision. <<https://www.designbetter.co/design-systems-handbook>> (luettu 11.4.2018).