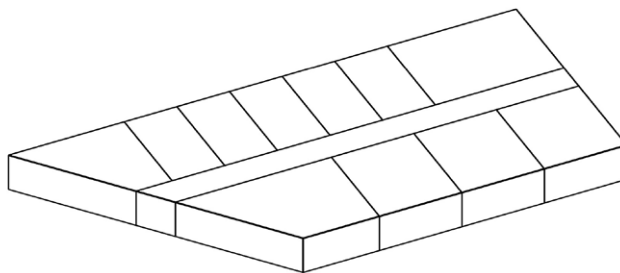
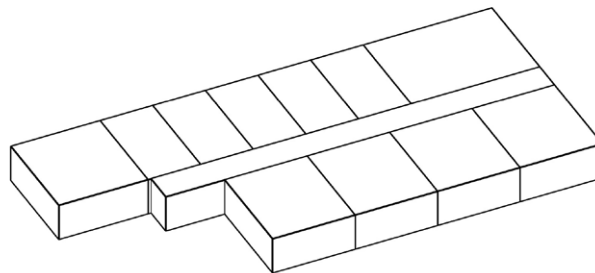
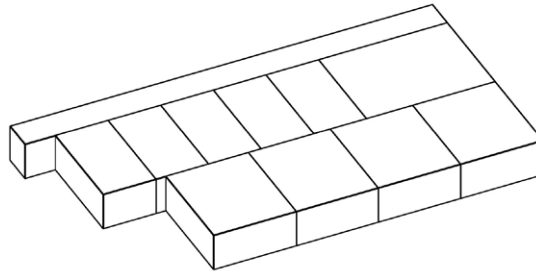
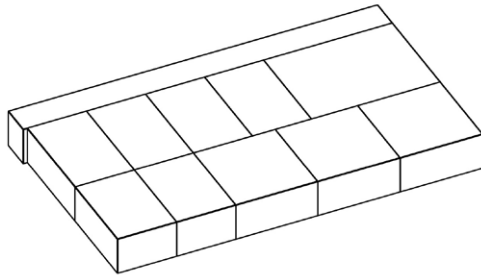
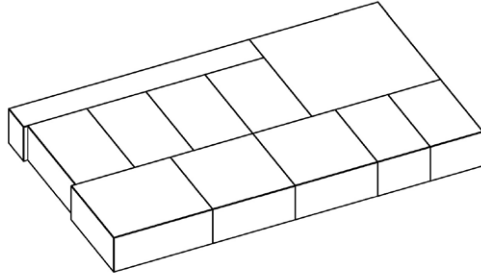


GENERATIIVINEN ALGORITMI SUUNNITTELIJAN TYÖKALUNA

Janne Hänninen / Opinnäytetyö / Metropolia Ammattikorkeakoulu



**GENERATIIVINEN ALGORITMI SUUNNITTELIJAN TYÖKALUNA
ASUINKERROSTALO SOMPASAAREN NIHTIIN**

Opinnäytetyö
Metropolia Ammattikorkeakoulu
Rakennusarkkitehtuuri

Janne Hänninen
2018-2019

“Rajojen koettelmiseen rakennustaiteessa liittyy muiden taiteenalojen tuotteiden käyttäminen arkkitehtonisen muodon generoimiseen. Koko ajatus arkkitehtuurimuodon synnyttämisestä toisen taiteenalan struktuurin avulla on perustavalla tavalla funktionalismin muodon ja funktion välisen pyhän allianssin vastaista. Arkkitehtuurin ilmaisu ei ehkä kasvakaan funktion erittelystä tai edes sosiaalisesta tilauksesta, vaan jostakin kaikkien taiteiden yhteisestä maaperästä. Ja taiteiden välillä vallinnee jonkinlainen struktuuriyhtäläisyys, jota voimme luovassa työssä hyödyntää.”

- Juhani Pallasmaa, arkkitehti-lehti 6/1990

TIIVISTELMÄ

Tekijä: Janne Hänninen
Otsikko: Generatiivinen algoritmi suunnittelijan työkaluna

Sivumäärä: 50 + 5 sivua
Aika: 24.4.2019

Tutkinto: Rakennusarkkitehti (AMK)
Tutkinto-ohjelma: Rakennusarkkitehtuuri

Ohjaajat: Janne Järvinen
Timo Vatanen

Tämän opinnäytetyön aiheena oli tutkia ja hyödyntää tietokoneen laskentaan perustuvaa arkkitehtisuunnittelua. Tietokoneen rooli suunnitteluprosessissa on muuttumassa ohjelmien kehittyessä sekä laskentatehon lisääntyessä jatkuvasti. Uudessa roolissa tietokonetta pyritään käyttämään uuden sisällön tuottamiseen algoritmien, laskennan ja generatiivisten järjestelmien avulla.

Tietokoneen laskentaan ja algoritmeihin perustuva suunnittelu ei ole ilmiönä uusi, mutta se on ottanut viime aikoina suuria harppauksia uusien ohjelmistojen syntymisen ja lisääntyneen käyttäjävälisyyden myötä. Ohjelmistot ja niiden käyttö alkavat lähestyä sitä pistettä, jossa ne alkavat olla suunnittelijoille enemmän hyödyksi kuin haitaksi ja algoritmien ohjelmoinnista voi muodostua yksi suunnittelutehtävistä muiden joukossa.

Generatiivisella suunnittelulla tarkoitetaan prosessia, jossa kyseistä algoritmeihin perustuvaa järjestelmää generoidaan tähän tarkoitukseen tarkoitettulla ohjelmalla, eli generaattorilla. Generointi tuottaa tällöin algoritmin avulla suuren määrän suunnittelun vaihtoehtoja, joista käyttäjän määrittelemillä mittareilla voidaan löytää laskennallisesti paras vaihtoehto.

Opinnäytetyö on jaettu kahteen toisiaan tukevaan osioon. Ensimmäisessä osassa, tutkielmaosuudessa tutustutaan aihealueen teoriaan ja kaikkea siinä opittua pyritään hyödyntämään toisessa osassa, suunnittelun harjoitustyössä.

Lopputuloksena generatiivista suunnittelua hyödynnetään harjoitustyössä arkkitehtisuunnittelijan yhtenä työkaluna; kerrostalon mallikerrospohjan luonnostelussa. Luonnostelun avulla pyritään löytämään ratkaisuja tyyppilliseen arkkitehdin suunnittelutehtävään, huoneistojen sijoitteluun tiettyyn kerrokseen.

Avainsanat: Generatiivinen, suunnittelu, algoritmi

ABSTRACT

Author: Janne Hänninen
Title: Generative Algorithm as Helping Tool for Designer

Page Count: 50 + 5 pages
Date: 24.4.2019

Degree: Bachelor of Construction Architecture
Degree Programme: Construction Architecture

Instructors: Janne Järvinen
Timo Vatanen

The subject of this thesis is to study and use the theory of computation in architectural design. The role of the computer is changing as the design programs are developing and computational power is increasing constantly. In this new role, the designer tries to harness computer as a creative tool to create design solutions together with computing, algorithms and generative systems.

As a subject, algorithmic design is not new, but which has seen minor breakthroughs recently, as new programs are developed and the ease of use of the existing ones is increasing. Algorithmic design programs and their use are starting to close in to the point where they are not seen anymore as burden to their users, but rather a useful helping tool. Programming such algorithms is starting to be one of the design tasks of architects.

Generative design means a process, where these algorithms are generated with a program designed for that, called generators. Generating the algorithm then produces a number of design solutions, that can be sorted with the help of user-created measuring tools that represent design goals.

This thesis is divided to two interlocked and each other supporting parts. The first one studies the theory of the subject and tries to use all of the knowledge learned from it in the second part, the actual practice design study.

The result is in the practice design study, where the designer used generative system as a design tool to find solutions to one level of apartment building. With help of the system, designer found out an optimal solution to the spatial layout of the certain level, a very typical design task in architecture.

Keywords: Generative, design, algorithm

SISÄLLYSLUETTELO

1. SANASTO	8
2. JOHDANTO	
2.1 Alkusanat	10
2.2 Tavoitteet	11
2.3 Rajaus	12
2.4 Tausta	13
3. TEORIA	
3.1 Generatiivinen suunnittelu	16
3.2 Historia	20
3.3 Esimerkkitapaukset	22
3.4 Johtopäätökset	25
4. PROJEKTI	
4.1 Aloitus	26
4.2 Aikataulu	27
4.3 Resurssit	28
5. SUUNNITTELUTYÖ	
5.1 Aloitus	32
5.2 Arkkitehtuurireferenssit	33
5.3 Lähtötiedot	36
5.4 Prosessi	40
5.5 Ratkaisut	48
6. PÄÄTELMÄT	53
7. LÄHTEET	54

1. SANASTO

ALGORITMI

(matematiikka, tietotekniikka.) sarja täsmällisesti määriteltyjä käskyjä tai toimenpiteitä jonkin tehtävän suorittamiseksi.

AUTOPOIEESI

(kreik. autos = itse, poiesis = tekeminen) Niiden eliössä tapahtuvien toimintojen verkosto, joka tuottaa tiettyjen keskenään vuorovaikuttavien biologisten järjestelmien osat; tämä verkosto on lisäksi itse itseään säätelevä

BIOMIMETIIKKA

biologiaa ja teknologiaa hyödyntävä monitieteellinen tutkimusala, joka pyrkii ratkaisemaan käytännön ongelmia ottamalla mallia luonnossa esiintyvistä rakenteista, toiminnoista ja materiaaleista

EVOLUUTIO

Prosessi, jossa erilaiset elävät organismit on uskottu kehittyneen aikaisemmista muodoistaan maapallon historiassa.

GENERAATTORI

Laite tai järjestelmä, joka generoi jotakin.

GENEROIDA

Tuottaa, erityisesti jonkinlaisella generaattorilla.

HEURISTIIKKA

1. Oppi tietoon ja oivalluksiin johtavista menetelmistä
2. Tieteenfilosofiassa metodologisten sääntöjen kokonaisuus pyrittäessä etenemään toivottuihin tai uusiin ideoihin ja tuloksiin

JÄRJESTELMÄ

Järjestelmä eli systeemi (kreik. sýstēma 'yhdistäminen' < synhistánai 'yhdistää') on joidenkin periaatteiden mukainen toiminnallinen kokonaisuus

KOODI

Säännöstö, jonka mukaan kielellisten merkkien (kuten sanojen, äänteiden ym. muotojen) suhde merkityksiin valikoituu esimerkiksi kielessä, murteessa, slangissa tai muussa kommunikaatiojärjestelmässä.

LASKETTAVUUS

Laskettavuus tai laskettavuusteoria on teoreettisen tietojenkäsittelytieteen haara, joka tutkii ongelmien ratkeavuutta ja ratkaisemisen tehokkuutta algoritmisesti.

METAHEURISTIIKKA

Metaheuristiikat ovat sellaisten menetelmien joukko, jotka toimivat korkeamman tason strategioina ohjaamassa optimaalisen ratkaisun hakuprosessia. Ne ovat moniin ongelmiin sopivia yleiskäyttöisiä menetelmiä, jotka tarvitsevat toimiakseen tehtäväkohtaisia algoritmeja ja heuristiikkoja.

OPTIMOINTI

Tietyn funktion suurimman tai pienimmän arvon tuottavien muuttujien arvojen löytäminen. Mm. dynaaminen optimointi, lineaarinen optimointi, matemaattinen optimointi.

PARAMETRI

Kaavoissa, malleissa jne. käytettävä ei-riippumaton muuttuja, apumuuttuja, kerroin tai tekijä.

SUKUPOLVI

Geneettisessä tutkimuksessa sukupolvi käsittää ne yksilöt populaatiossa, jotka ovat peräisin yhteisistä vanhemmista.

SYNTEESI

Yleisessä mielessä analyysiin perustuva yhteenveto tai kokonaisnäkemys, joka yhdistää erilaisia aineksia.

-
- 001 Tieteen termipankki, The Helsinki Term Bank for the Arts and Sciences
 - 002 Oxford Dictionaries Online
 - 003 Kielitoimiston sanakirja, Kotimaisten kielten tutkimuskeskus
 - 004 Hotokka, Pekka. Metaheuristiikat logistiikan optimoinnissa.

2. JOHDANTO

2.1 ALKUSANAT

Ajatus opinnäytetyön aiheesta syntyi kesällä 2018 työskennellessäni arkkitehtitoimistossa tietomallien parissa. Kiinnostukseni ja taitoni tietotekniikassa luultavasti ohjaavat minua rakentamisen alan tietotekniisiin tehtäviin, kuten tietomallintamiseen ja sen koordinointiin. Tästä syystä vahvasti tekniset ja tietokonepohjaiset aihealueet rajautuivat opinnäytetyön ideoiksi.

Aiheesta kiinnostuneena törmäsin kesällä kirjallisuuteen tutustuessani Jiri Hietasen kattavaan kirjaan; *Tietomallit ja Rakennusten suunnittelu, Filosofinen selvitys tieto- ja viestintätekniikan mahdollisuuksista (2005)*, jossa perehdytään aihealueeseen pääosin tietomallintamisen näkökulmasta.

Kirjassa oli kuitenkin lyhyt kappale aiheesta generatiivinen suunnittelu ja tietokoneen rooli luovassa suunnittelutyössä. Aihe oli hyvin kiinnostava sen ajankohtaisuuden ja tulevaisuudennäkymien takia, joten päätin valita sen opinnäytetyöni aiheeksi.

Aihealueesta on tehty maailmalla runsaasti tutkimuksia ja jopa toteutuneita projekteja. Suomessa ko. aihetta käsittelee kattavasti yksi diplomityö (Incessant replication: Computational floor plan generation, Ron Aasholm, 2015) Aihe tuntui sopivalta opinnäytetyölle myös siitä syystä, että sen yhdistäminen itse suunnittelutyöhön tuntui mahdolliselta ja kiinnostavalta haasteelta.

Johtuen opinnäytetyön aikataulusta sekä tiukasta kytköksestä itse harjoitustyöhön, muodostui projektin aikana tutkimuskysymykseksi tällöin:

Voiko generatiivista algoritmia käyttää arkkitehtisuunnittelun työkaluna?

2.2 TAVOITTEET

Opinnäytetyön tavoitteet voidaan opintokokonaisuuden rakenteen takia jakaa kahteen osioon. Molempia osioita käsitellään tässä tutkielmassa yhtenä kokonaisuutena ja tutkielma pyrkii myös yhdistämään molempien osioiden osuudet toisiaan tukeviksi elementeiksi.

Ensimmäisessä osassa tutustun algoritmisen sekä generatiivisen suunnittelun taustoihin ja teoriaan sekä muutamiin aihealueen esimerkitapauksiin maailmalta. Maailmalla on joitakin arkkitehtitoimistoja, joiden yleisiin toimintatapoihin kuuluu hyödyntää algoritmeja sekä laskentaa suunnittelutyössä. Näistä kerron enemmän luvussa 3.3 *Esimerkitapaukset*.

Toinen osa, itse rakennuksen suunnittelu on opinnäytetyön kokonaisuuden merkittävämpi osuus. Rakennusarkkitehdin tutkintoon kuuluva opinnäytetyö oli vahvasti sidottu samaan aikaan käytävään Projektin 12: Syventävä ammatillinen -harjoituskurssiin, jonka tarkoituksena on toimia suunnittelun harjoitustyönä, jota tutkielma tukee.

Harjoitustyössä käytän hyödykseni teoriassa käsiteltyjä asioita ja tuotan ehdotussuunnitelman tasoista sisältöä kerrostalon suunnitelmasta. Johtuen aihealueen prototyypimäisestä ja kokeellisesta luonteesta oli tavoitteenani ensisijaisesti tuottaa kerrostalon suunnitelma ja toissijaisena tavoitteena hyödyntää tässä suunnitelmassa generatiivisia suunnittelumetodeja.

2.3 RAJAUS

Suunnitteluosuuden rajaaminen oli yksi opinnäytetyön vaikeimmista tehtävistä, johtuen aihealueen prototyyppimäisestä luonteesta sekä riippuvuudesta muiden ohjelmistojen käyttäjien luomiin algoritmeihin.

Algoritmeilla tarkoitetaan tässä asiayhteydessä Autodeskin Dynamo-ohjelmaa ja sen graafista ohjelmointiympäristöä. Alusta asti oli selvää, että näin lyhyessä ajassa ei olisi mahdollista ohjelmoida koko algoritmia itse, joten oli tyydyttävä jonkun toisen luomaan, valmiiseen algoritmiin, jota lähtisin muokkaamaan omaan käyttötarkoitukseeni sopivaksi. Tähän räätälöintityöhön kului erityisesti projektin alkuvaiheessa hyvin runsaasti aikaa, sillä en tietoteknisestä kykeneväisyydestäni huolimatta ole juurikaan ohjelmoinut mitään.

Rajauksen vaikeus liittyi esimerkiksi syksyllä kurssin pakollisena harjoituksena tehtyyn tehtävänantoon. Tähän aihealueeseen tehtävänannon luominen tuotti haasteita esimerkiksi algoritmin hyödyntämisen suhteen, käytetäänkö algoritmia rakennuksen kerrospohjan luonnosteluun, julkisivuihin, massoitteeluun vai kaikkiin näihin? Tehtävänantovaiheessa minulla ei ollut vielä tarkkaa käsitystä algoritmin käyttöönottamiseen, ymmärtämiseen, ja hyödyntämiseen kuluva ajasta, jolloin määriteltäväksi jäi lähinnä 'algoritmin hyödyntäminen'.

Myöhemmin, lähdeaineiston keräämisvaiheessa, löysin Ron Aasholmin Aaltoyliopiston arkkitehtuurin laitoksen maisterivaiheen diplomityön '*Incessant Replication: Computational Floor Plan Generation (2015)*'.

Tutkimuksessa Aasholm perehtyy syvällisesti aihealueen metodologiaan sekä rakennuksen pohjapiirroksen generoimiseen laskennallisesti. Diplomityö on 294-sivuinen, ja se auttoi minua käsittämään aihealueen kompleksisuuden ja tarkentamaan omaa tehtävänantoani.

2.4 TAUSTA

Ennen perehtymistä opinnäytetyön aiheen taustoihin, on syytä huomioida aiheeseen liittyvä haaste, eli suomen kieli, jota käytämme. Ymmärtääkseen aihealuetta, lukijan pitää tiedostaa kieleemme vajavaisuus erityisesti tietotekniikkaan liittyvissä asioissa. Kielen asettamat rajoitukset ja ammattikielen kääntäminen ymmärrettävään arkikieleen aiheuttaa usein virheitä sekä vääristyneitä mielikuvia.

Tällaisen mielikuvan antaa jo heti aluksi suomenkielinen sana "tietokone", joka viittaa koneeseen, joka *tietää* asioita (vrt. engl. computer ja computation). Mielikuva persoonattomasta koneesta, josta 'nappia painamalla' saa tietoa, on valmis ja pimentoon jää kokonaan se tieteen- ja taiteenala, jossa kyseinen laite ja sen laskentaan perustuva ohjelmoiminen on ehtymätön luovuuden ja inspiraation lähde lukemattomille ihmisille. Aihealueen ymmärtäminen liittyikin vahvasti kieleen sekä mielikuvien ja todellisuuden väliseen eroon.⁰⁰⁵

Ymmärtääkseen myös tarkemmin opinnäytetyön aihetta, eli laskentaan perustuvaa suunnittelua, pitää ymmärtää laskennan (engl. computation) ja tietokoneistamisen (engl. computerize) ero. Käytännössä nämä kaksi voidaan erottaa niin, että siinä missä laskennan, eli komputaation avulla tuotetaan johtopäätöksiä sekä sisältöä, tietokoneistamisella lähinnä koostetaan sisältöä, jota tietokoneelle syötetään.⁰⁰⁵

Tietokoneet sekä tietokoneavusteinen suunnittelu muuttivat suuresti rakennussuunnittelijoiden tapaa työskennellä tietokoneiden yleistyttyä 1980-luvulla. Siitä huolimatta, että ohjelmistojen kehittäjien ajatuksena oli suunnittelijan ja tietokoneen monitahoinen kanssakäyminen, olivat ensimmäiset laajasti käyttöönotetut ohjelmat lähinnä tietokoneelle käännettyjä piirustusohjelmia. Vaikka ne mahdollistivat suuren tehostamisen suunnittelijoiden työskentelyssä, ne eivät juurikaan muuttaneet heidän työskentelyprosessiaan.⁰⁰⁶

005 Hietanen, *Tietomallit ja rakennusten suunnittelu*, 8.

006 Nagy, Lau et al. *Project Discover: An application of generative design for architectural space planning*, 1.

Vasta viime vuosina on kehittynyt uudenlaisia, helposti lähestyttäviä laskentaan perustuvia suunnitteluohjelmistoja, jotka saattavat perustavanlaatuisesti muuttaa sitä tapaa, jolla suunnittelijat käyttävät tietokonetta.

Uudet graafiseen ohjelmointiin perustuvat ohjelmat mahdollistavat helpon lähestymistavan laskentaan ja algoritmeihin perustuvaan suunnitteluun, joka ei enää pelkästään määrittele lopullista geometristä ratkaisua, vaan koko järjestelmän, millä ratkaisu tuotetaan. Tällaisilla järjestelmillä suunnittelija voi järjestelmää tutkimalla ja manipuloimalla löytää lukuisia suunnitteluratkaisuja.⁰⁰⁶

Laskentaan perustuva suunnittelu tulee todennäköisesti aiheuttamaan alalla suuren muutoksen sekä haasteen suunnittelijoille. Useimmiten haasteen ajatellaan olevan ainoastaan tekninen luonteeltaan, sillä toiminta usein vaatii kehittyneitä ohjelmointitaitoja, joita ei yleisesti opeteta osana alan pääasiallisia tutkintoja. Tästä huolimatta, saavuttaakseen täyden hyödyn laskennallisesta suunnittelusta, on aiheesta kiinnostuneen tutkijan perehdyttävä arkkitehtuurin lisäksi monitieteellisesti sekä matematiikkaan, tietotekniikkaan, systeemiteoriaan että biologiaan.⁰⁰⁷

Tietolähteitä aihealueesta on runsaasti, mutta niiden laadullinen taso sekä painotus tietotekniikan ja rakennustaiteen välillä vaihtelee. Tutkimuksien lukijan on oltava perillä sekä tietotekniikan, että rakennusalan ammattikielistä ymmärtääkseen niitä. Aihealueen monitieteellisyydestä johtuen tutkimukset vilisevät termejä, jotka vaativat usein lisäperehtymistä. Tästä syystä myös tämä tutkimus on avattu laajalla sanastolla.

Valjastaessa tietokonetta uusin käyttötarkoituksiin on kyse usein sen määrittelystä, mitä suunnittelun rooleja tietokoneelle annetaan. Jiri Hietasen mukaan arkkitehtisuunnittelun tehtävät voidaan jakaa neljään kategoriaan;

- Manuaalinen suunnittelu
- Tietokoneavusteinen suunnittelu
- Ihmisavusteinen suunnittelu
- Tietokonesuunnittelu

006 Nagy, Lau et al. *Project Discover: An application of generative design for architectural space planning*, 1.
007 Menges, Ahlquist. *Computational Design Thinking*, 8.

Kategoriat eivät ole tarkkarajaisia ja ne monesti myös limittyvät toistensa kanssa.

Manuaalisella suunnittelulla tarkoitetaan lähtökohtaisesti toimintaa, joka perustuu suunnittelun ikivanhoihin traditioihin. Puhutaan yleisesti siis kynä, viivain ja paperi -tyyppisestä suunnittelusta.⁰⁰⁵ Manuaalisessa suunnittelussa suunnittelija tuottaa sisällön, sekä myös arvioi sen laadun itse, korostaen tällöin suunnittelijan kokemusta.⁰⁰⁸

Tietokoneavusteinen suunnittelu (CAD, *Computer aided design*) on rakennus-suunnittelussa tätä nykyä jo vakiintunut tapa. Tietokoneavusteisessa suunnittelussa suunnittelija syöttää järjestelmään informaatiota, jota se palauttaa takaisin 2d-piirustuksen, 3d-mallin tai osaluettelon muodossa.⁰⁰⁵ Tietokoneavusteisella suunnittelulla voidaan edelleen suunnitella mitä kompleksisempia rakennuksia, mutta siinä tietokoneen rooli on silti rajoittunut reprodusoimaan sille annettuja tietoja.

Ihmisavusteinen suunnittelu on sellainen toimintatapa, jossa prosessi tarvitsee enää ihmisen ohjaamaan järjestelmää. Suunnittelu voi perustua esimerkiksi eräänlaiseen suunnitteluevoluutioon, jossa prosessin antamista ratkaisuista valitaan joko paras ratkaisu ja/tai algoritmia muokataan parempaan suuntaan. Näin suunnitteluratkaisuista syntyy prosessin aikana useita sukupolvia, aina vähän edellistä kehittyneempinä. Tällöin ihminen ja kone yhdessä generoivat suunnitteluratkaisuja.

Tietokonesuunnittelussa tietokone sekä suunnittelee, että arvioi tuottamiensa ehdotusten laadun itse.⁰⁰⁵ Tietokone muuttuu tällöin lähes suunnittelijaksi ja voi suoriutua rajatuista suunnittelutehtävistä itsenäisesti. Utopistiselta kuulostava skenaario saattaa olla tulevaisuudessa mahdollista, ottaen huomioon tekoälyn, koneoppimisen sekä kvanttietokoneiden luomat mahdollisuudet sekä tietokoneen huiman kehityskulun tähän päivään mennessä.

Suunnittelijan rooli tulevaisuudessa on tietysti luoda, tai olla luomatta näitä järjestelmiä sekä kehittää niitä. Nykytekniikan puitteissa on jo täysin mahdollista luoda järjestelmä, joka automaattisesti suunnittelisi 1970-luvun betonielementtistä lähiörakentamista, mutta onko se tavoittelemisen arvoista?⁰⁰⁵

005 Hietanen, *Tietomallit ja rakennusten suunnittelu*, 79.
008 Mitchell, William J. *The Logic of Architecture*, 180.

3. TEORIA

3.1 GENERATIIVINEN SUUNNITTELU

Aluksi on tutkittava teorian käsitteitä. Sana generatiivinen lähtökohtaisesti viittaa johonkin järjestelmään, joka generoi, eli tuottaa sisältöä jonkinlaisella generaattorilla.⁰⁰²

Suunnittelu taas tarkoittaa sanana joko ajatuksen tasolla, tai jotain esittävää formaattia käyttäen tehtävää toimintaa, jolla pohditaan tulevaisuudessa tapahtuvia asioita.⁰⁰²

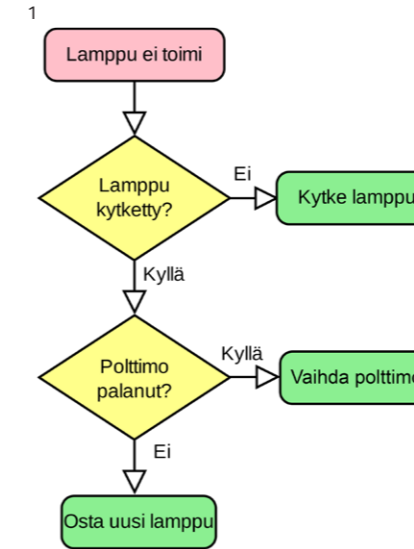
Taitelijia-tutkija Philip Galanter on perehtynyt urallaan generatiiviseen taiteeseen ja on vuonna 2003 määritellyt generatiivisen taiteen osuvasti, lainaus:

“Generative art refers to any art practice where the artist uses a system, such as a set of natural language rules, a computer program, a machine or other procedural invention, which is set into motion with some degree of autonomy contributing to, or resulting in a completed work of art.”⁰⁰⁹

Riippumatta taiteenlajista tai lähestymiskulmasta, generatiivinen taide (tässä tapauksessa arkkitehtisuunnittelu) pitää siis sisällään jonkin automaattisen järjestelmän, jota sen käyttäjä joko osittain tai kokonaan kontrolloi.⁰⁰⁹

Generatiivisten järjestelmien perusyksikköjä ovat algoritmit. Algoritmit ovat kaikkein mukautuvimpia kustomoinnille, koska ne eivät luonnostaan muodosta tiettyä struktuuria, suhteita tai esitystapaa, vaan ne ainoastaan tarjoavat työskentelyolosuhteet suunnitteluprosessin havainnollistamiselle. Algoritmeihin perustuva arkkitehtisuunnittelu ei ole mikään uusi ilmiö, monet nimekkäät arkkitehdit ovat paketoineet suunnittelunsa menetelmiään ja työskentelytapojaan algoritmeihin omiksi muistisäännöstoikseen tai toisten suunnittelijoiden käytettäväksi.⁰¹⁰

Algoritminen suunnittelujärjestelmä koostuu ennakkoon määritellyistä suunnittelun periaatteista, jotka avaavat ja sulkevat suunnittelumahdollisuuksia ja -vaihtoehtoja. Järjestelmään on tarkoitus delegoida suunnittelutehtäviä ja suunnittelijan älyä sekä automatisoidaan näitä tehtäviä.⁰¹¹



Järjestelmä ei muutu tällöin suunnittelijaksi vaan ihmisuunnittelija vain pyrkii ulkopuolistamaan ja algoritmisoimaan omaa suunnittelun prosessiaan tietokoneen ymmärtämään muotoon.^{011, 012}

Hyvin yksinkertaisena suunnitteluprosessina voidaan pitää esimerkiksi mallia esitiedotluonnostelu-viimeistely, joista edellisen on aina oltava valmis ennen kuin seuraavaan voidaan jatkaa ja joista jokainen voi sisältää taas oman prosessinsa.

Arkkitehtisuunnittelussa tällaisella suunnittelujärjestelmällä tarkoitetaan prosessia, jossa ennalta määriteltyjen suunnittelun raja-arvojen ja algoritmisen prosessin avulla syntyvien tulosten mittaamisella pyritään pääsemään parhaisiin suunnitteluratkaisuihin.

Tietokoneen algoritmiksi käännetty suunnitteluprosessi aiheuttaa ihmisen ja tietokoneen välille syntyvän kommunikaation haasteen, sillä siinä missä ihminen pystyy ajoittain vastaamaan kysymyksiin 'ehkä' tai 'noin suurin piirtein', vaatii yksinkertainen algoritmi usein täsmällisiä vastauksia sekä tarkkoja arvoja toimiakseen (sumeaa logiikkaa yrittää ratkaista tätä ongelmaa). Suunnittelualgoritmin kehittäjän yksi tehtävistä onkin suunnittelukysymysten yksinkertaistaminen sellaiseen muotoon, joissa algoritmi voi vastata niihin vain joko muodossa tosi-epätosi tai lukuarvolla.

Manuaalisesta tai ihmisavusteisesta arkkitehtisuunnittelusta poiketen, digitaalisesti generoidut tilat eivät ole suunniteltu tai piirretty tavanomaisesti vaan valitulla, laskentaan perustuvalla tavalla. Sen sijaan, että suunnittelija muotoilisi erilaisten kappaleiden ulkorajapintoja, hän pyrkii hiomaan generatiivista logiikkaa ja sen koodia paremmaksi, jolloin prosessi tuottaisi itse automaattisesti joukon vaihtoehtoja jatkokehitykseen. Koneen kyky tuottaa vaihtoehtoja nopeasti ja suuria määriä voi auttaa löytämään yllättäviä ja luovia ratkaisuja kompleksisiin ongelmiin.⁰¹³

002 OxfordDictionaries.com

009 Galanter, Philip. "Generative Art Theory." A Companion to Digital Art, 9.

010 El-Khaidi, Maher. Mapping Boundaries of Generative Systems for Design, 31.

011 Gürsel, Dino. Creative design exploration by parametric generative systems in architecture, 208.

011 Gürsel, Dino. Creative design exploration by parametric generative systems in architecture, 208.

012 Terzidis, Kostas. Algorithmic Design: A Paradigm Shift in Architecture?, 202.

013 Kolarevic, Branko. Architecture in the Digital Age: Design and Manufacturing, 13

1 Yksinkertainen vuokaavioalgoritmi

Lähde: Wikimedia Commons

Generatiivinen järjestelmä vaatii neljä elementtiä toimiakseen:

- Aloitusolosuhteet ja parametrit eli syötteen (engl. input).
- Generatiivisen mekanismin (säännöt, algoritmit).
- Säännöllä syntyvän tuloksen eli vasteen (engl. output)
- Näiden avulla generoimisen ja parhaimman valitsemisen (mittarit).

Ennen kuin viimeinenkin elementti on suoritettu, järjestelmä ei synnytä tulosta.⁰¹⁰ Prosessin heikkous on sen ehdottomuus ja heikko virheidensietokyky, sillä yksikin väärä arvo voi pilata koko laskujonon. Prosessin kapasiteetti eli kyky tuottaa sisältöä on myös vahvasti sidoksissa sen ohjelmoijan kykyyn ymmärtää ja manipuloida sitä, tehden järjestelmistä varsin henkilökohtaisia työkaluja.⁰¹⁰

Generatiivista suunnittelua on hyödynnetty laajemmin esimerkiksi tuoteteollisuudessa, sillä siellä tuotteen toivotut ja ei-toivotut ominaisuudet ovat usein helpommin määriteltävissä (lujuus, paino, koko). Algoritmien ja laskennan avulla on pystytty suunnittelemaan tuotteita, joissa niille on määritelty tavoitteeksi tietyt lujuusominaisuudet ja ulkomitat, mutta rakenne tuotetaan generatiivisen suunnittelun ja esimerkiksi 3d-tulostamisen avulla.

Tällaisia tuotteita ovat esimerkiksi Autodeskin ja Airbusin yhteistyössä kehittämä lentokoneen alumiininen tilanjakaja-osa, jonka tarkat vaatimukset täyttävä rakenne on tuotettu generatiivisilla suunnittelumetodeilla. Lopputulos on samoilla lujuusarvoilla 45% kevyempi kuin aiempi, yleisesti käytössä ollut osa.⁰¹⁴

Toinen vastaava tuote on skolioosista kärsiville nuorille kehitetty selkärangan tuki, jonka rakenteen suunnittelussa Yhdysvaltalainen Studio Bitonti on hyödyntänyt generatiivista suunnittelua.

Lopullisen tuotteen lääketieteelliset, selkärankaa tukevat arvot ovat samat kuin vanhassa, kovamuovista ja pehmusteesta valmistetulla tuella, mutta tuote on sekä huomattavasti kevyempi, hengittävämpi että kauniimpi.⁰¹⁵



² Generatiivisesti suunniteltu selkärangan tuki skolioosista kärsiville henkilöille.

Lähde: Studio Bitonti/Jason Perry

⁰¹⁰ El-Khaidi, Maher. Mapping Boundaries of Generative Systems for Design, 27

⁰¹⁴ Walmsley, Kean. Autodesk and Airbus generatively design a 3D-printed "bionic" partition for the A320.

⁰¹⁵ Bitonti, Francis, et al. 3D-Printed Body Architecture, 65-69.

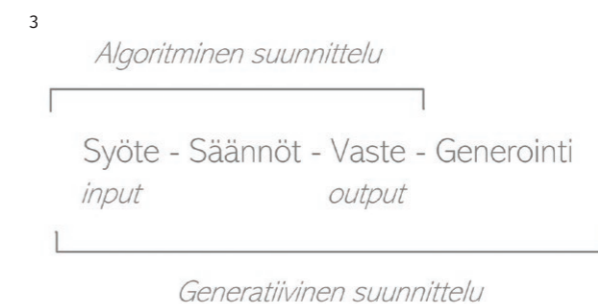
Tällaiset onnistuneet projektit toimivat lupaavina esimerkkeinä generatiivisten algoritmien hyödyntämisestä tarkkojen raja-arvojen sisällä ja vastaavanlaista käyttöä voisi hyvinkin odottaa näkevänsä rakentamisen alalla esimerkiksi rakennessuunnittelussa, joka perustuu myös vahvasti laskentaan ja mitattaviin arvoihin.

Arkkitehtuurin tuotteeseen, eli rakennukseen sen sijaan kuuluu usein sellaisia ominaisuuksia, joita on vaikea mitata (kauneus, sopiminen ympäristöön, kulttuuriperintö) ja joihin on lähes mahdotonta vastata tosi-epätosi -akselilla tai lukuarvoilla, jolloin niiden tulkinta laskennallisesti on hyvin haastavaa.

Tästä syystä generatiivinen järjestelmä tulee tuskin koskaan syrjäyttämään ihmistä arkkitehtisuunnittelussa. Tämän Alvar Aalto osasi kiteyttää jo vuonna 1947. lainaus:

*"...arkkitehtoninen suunnittelu operoi lukemattomilla, usein keskenään ristiriidoissa olevilla elementeillä. Sosiaaliset, humaniset, taloudelliset ja teknilliset vaatimukset yhdistettyinä psykologisiin kysymyksiin, koskien sekä yksilöä että ryhmää, sekä ihmismassojen että yksilöiden liikkeitä ja sisäisiä friktioita, kaikesta tästä muodostuu monisäikeinen vyyhti, joka ei ole rationalistista tai mekaanista tietä ratkaistavissa."*⁰¹⁶

Ristiriitaisuus, monitieteellisyys, humanisuus ovat arkkitehtuuria kuvaavia ominaisuuksia ja kaikki abstrakteja, tietokoneen käsityskyvyn yläpuolisia asioita. Insinöörismielinen opinnäytetyön tekijä voi pyrkiä ratkaisemaan arkkitehtisuunnitteluun liittyviä ongelmia laskemalla ja yksinkertaistamalla suunnittelukysymyksiä, mutta tulee samalla alentaneeksi rakennuksen omaksi yksinkertaiseksi tuotteeksi. Mikäli tämä ei ole ongelma ja rakennus nähdään vain ihmisten asumiseen tarkoitettuna koneena ja mittaustulokset täyttävänä esineenä, voi tulevaisuus olla aihealueelle hyvinkin valoisa.



³ Generatiivisen järjestelmän neljä elementtiä

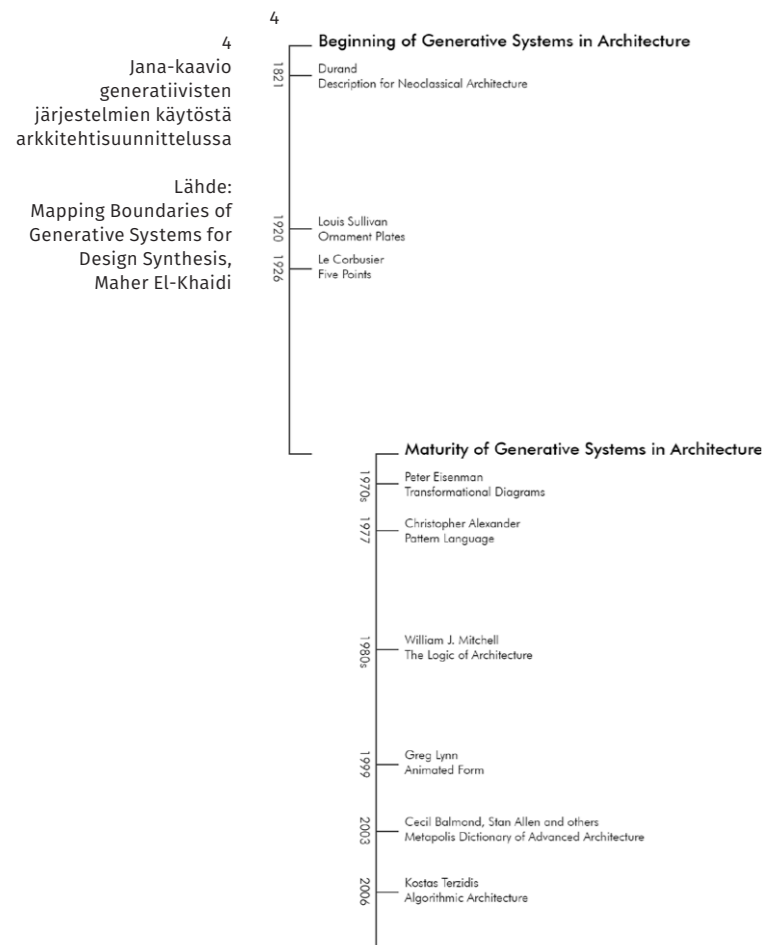
⁰¹⁶ Aalto, Alvar. Taimen ja tunturipuro, 7-10

3.2 HISTORIA

Jo ennen digitalisaatiota arkkitehdit tutkivat algoritmisia suunnitteluprosesseja, joita tietokoneet myöhemmin vain tehostivat ja automatisoivat.

Puhuttaessa generatiivisesta suunnittelusta, parametreista tai algoritmeista, yltää niiden historia kauas ennen tietokoneita tai digitalisaatiota. Generatiivinen logiikka on ollut jo kauan läsnä sekä filosofian, kirjallisuuden, musiikin että arkkitehtisuunnittelun aloilla.⁰¹⁷

Historian saatossa monet maineikkaat arkkitehdit ovat pyrkineet luomaan järjestelmiä tai prosesseja, joilla suunnittelija voisi erilaisista palasista koostaa parhaimman vaihtoehdon rakennukselle, näitä järjestelmiä voidaan kutsua generatiivisten järjestelmien varhaisiksi versioiksi.⁰¹⁰



Reproduktio oli yhteistä kaikille näille järjestelmille, ei niinkään variaatiot tai suunnitteluvaihtoehdot.

Yksi ensimmäisistä järjestelmistä, joissa tutkitaan nimenomaisesti variaatioita generoimisen kautta, on William J. Mitchellin julkaisu *'The Logic of Architecture'* jossa hän lisää algoritmien, evoluution ja logiikan konseptin arkkitehtuurin suunnitteluprosessiin.⁰¹⁰

Tutkimuksessa Mitchell tutkii kuinka rakennuksia voidaan tulkita eri osiensa summana, samalla tavalla kuin kielessä kokonaisia lauseita tulkitaan niiden sisältämien eri sanojensa summana.⁰⁰⁸

017 Aasholm, Ron. Incessant replication: Computational floor plan generation, 58.
010 El-Khaidi, Maher. Mapping Boundaries of Generative Systems for Design, 27, 15.
008 Mitchell, William J. The Logic of Architecture, 137.

Varhaiset generatiiviset suunnittelujärjestelmät synnyttivät aikanaan teoreettista keskustelua ja ajatuksia tilasta, tilaohjelmista, tektoniikasta ja kulttuurista. Yhtenä tällaisten järjestelmien pääasiallisista kehittäjistä voidaan pitää yhdysvaltalaisista arkkitehtia ja teoreetikkoa Peter Eisenmania.

Eisenmanin työt, aina hänen uran alkuvaiheen House-projekteista myöhempiin sisältivät jo 1970-luvulla paljolti samoja prosessin elementtejä kuin myöhemmin yleistyneet, tietokoneen laskentaan perustuvat suunnitteluprosessit.

Prosessissa Eisenmann pyrki tutkimaan arkkitehtuuria omana formaalina kielenään, joka hakee muotoaan transformaation ja dekomposition kautta ja luo tällöin näkyviksi tiloiksi suunnittelun prosessin.⁰¹⁸

Opinnäytetyön harjoitustyössä käytettävää prosessia voidaan verrata tällaiseen suunnittelun teoriaan siinä mielessä, että muodon (kerrostaso) syntymiseen vaikuttaa jokin kieli (graafinen ohjelmointi) jota manipuloimalla ja generoimalla voidaan tuottaa lukematon määrä ennustamattomia vaihtoehtoja.

Henkilökohtaisen tietokoneen yleistymisen jälkeen, vuonna 1992 Eisenman onkin sanonut:

*"Henkilö voi syöttää joukon määrääviä sääntöjä ja lähtötietoja tietokoneeseen ilman etukäteistä tietoa mitä ulostulo tulee olemaan. Tällöin itse prosessista muodostuu yksi suunnittelun algoritmeista ja prosessin koodaamisesta ja korjaamisesta tulee yksi suunnittelutehtävistä."*⁰¹⁸

Työskentelytapojensa takia Eisenmanin rakennuksia ei olekaan tyypillisesti nähty suunnittelemisen lopputuloksena vaan eräänlaisina suunnitteluprosessin representaationa.⁰¹⁹ Tutkimalla näiden järjestelmien historiaa ja toimintaa, pystyin syventämään osaamistani ja käsittämään aihealuetta paremmin.

018 Novak, Marcos. Computational Composition in Architecture, 16.
019 Luce Kristina. Getty Research Journal, no.2, 125-137

3.3 GENERATIIVISEN SUUNNITTELUN ESIMERKITAPAUKSET

Project Discover: An Application of generative design for architectural planning.

Danil Nagy, Damon Lau, John Locke, Jim Stoddart, Lorenzo Villaggi, Ray Wang, Dale Zhao and David Benjamin. The Living, an Autodesk Studio. 2017

Kenties yksi tämän hetken tunnetuimmista generatiivisen suunnittelun tutkimuksista. Autodeskin The Living on itsenäinen, konsernin sisäinen arkkitehtitoimisto, joka pyrkii käyttämään suunnittelussaan hyödyksi generatiivista suunnittelua, biomimetiikkaa sekä uudenlaisia materiaaleja.

Vuonna 2017 he suunnittelivat konsernille uuden toimistorakennuksen Torontoon, Kanadaan. Suunnittelussa lähdettiin liikkeelle alusta alkaen hyvin epäortodoksisella tavalla ottamalla generatiivinen suunnittelu mukaan vahvasti.

Aluksi he määrittivät suunnittelun aloitusparametrit algoritmin syötteeksi. Näitä parametrejä olivat muun muassa rakennuksen fyysiset ulkorajat, tilaohjelma, henkilöstömäärä (300 hlöä) sekä eri yksiköiden määrät ja niiden työskentelytavat. Syötteen perusteella algoritmi pyrki sijoittamaan tiloja kerros pohjaan parhaalla mahdollisella tavalla.

Suunnittelijat lisäsivät suunnittelualgoritmiin eri yksiköille ja tiloille erilaisia tavoitteita ja mittareita, jotka mittaavat kuinka hyvin generoitu tila toteuttaa sen vaatimuksia. Tällaisia tiloja olivat esimerkiksi; normaali yksityinen toimisto, pieni yksityinen toimisto, hiljainen tiimihuone, yhteistyö-tiimihuone tai avokonttoritila. Jokaiselle tilalle oli omat mittarinsa ja tämän lisäksi suunnittelualgoritmissä oli myös kaikille tiloille yhteisiä mittareita.

Tutkielmassa käsitellään erityisesti arkkitehtuurin mittaamiseen liittyvää problematiikkaa ja ehdotetaan uusia, toimistotiloihin liittyviä mitattavia suureita. Näitä ovat esimerkiksi; Tilojen sijainti suhteessa toisiinsa (Adjacency preference), Työskentelytavat (Work style preference), Pöhinä (Buzz), Työtehokkuus (Productivity) ja Auringovalo (Daylight)

Valmis suunnittelualgoritmi syötettiin generointialgoritmiin (MOGA, Multi-objective genetic algorithm) ja se käynnistettiin tavanomaisella MacBookilla, jolla sen annettiin käydä yhtäjaksoisesti 5 päivää.

Generaattori määriteltiin tuottamaan 100 suunnitelmaa, joista jokainen kävi läpi 100 sukupolvea, tuottaen 10 000 vaihtoehtoa. Näistä vaihtoehtoista generaattori pystyi järjestämään parhaimmat vaihtoehdot keräämällä tietoa yksittäisten tilojen sekä kaikkien tilojen yhteisistä mittareista ja laskemalla näistä arvoista jokaiselle vaihtoehdolle keskiarvoisen kokonaislukeman.

Generaattoriin oli mahdollista määrittää sen generointitapa ja tässä projektissa käytettiin ensimmäisiin sukupolviin sattumanvaraista generointia löytämään mahdollisimman paljon variaatioita. Näiden variaatioiden perusteella löydetyt, parhaiten suoriutuneet vaihtoehdot valittiin jatkoon (elitismi) ja niihin käytettiin jälkimmäisissä sukupolvissa joko ristituloa (kahden parhaiten suoriutuneen risteyttäminen) tai mutaatiota (parhaiten suoriutuneen muokkaaminen).

Näin ollen generaattori toiminnallaan matki luonnosta tuttua evoluution konseptia, jossa edellinen sukupolvi määrittää aina seuraavan aloitusparametrit ja tämä ajan ja toiston kautta luo olosuhteet luonnonvalinnalle ja kelpoisimman elonjäämiselle (survival of the fittest).

Generoinnin lopputuloksena tiimi löysi heidän luomiensa mittarien perusteella parhaiten suoriutuneen vaihtoehdon, joka toimi perustana suunnitelmalle.

006, 020

006 Nagy, Lau et al. Project Discover: An application of generative design for architectural space planning.
020 Gerfen, Katie. Autodesk MaRS Office.

Kartal-Pendik Masterplan, Istanbul, Turkey
Zaha Hadid Architects, Patrik Schumacher, 2006

Zaha Hadidin (1950-2016) toimisto on tunnettu poikkeavasta suunnittelustrategiastaan. Toimiston suunnittelumetodeihin kuuluu selkeä suunnittelun prosessikaavio (työskentelyalgoritmi), joka käydään läpi jokaisen projektin kohdalla. Toimisto on tunnettu myös uusien, digitaalisten työkalujen laajasta hyödyntämisestä suunnitteluprosesseissa.⁰²¹

Hadidin pitkäaikainen työskentelykumppani Patrik Schumacher on jopa esittänyt uuden arkkitehtuurin tyylin nimeltä *parametrisismi*. Schumacherin mukaan tyyli on jatkumoa modernistiselle ja post-modernistiselle arkkitehtuurille ja sen juuret ovat digitaalisissa mallinnusohjelmissa. Keskeistä tyyllille on *autopoiesi*, eli suunnittelujärjestelmän eri osien vuorovaikutus toisiinsa ja kyky määrittellä itsensä käyttäytymistä.⁰²²

Vuonna 2006 Zaha Hadid Architects voitti Kartal-Pendikin alueen kaavoituskilpailun Aasian-puoleisessa Istanbulissa. Alue käsitti 55 hehtaaria ja 6 miljoonaa kerrosalaneliometriä entisellä teollisuusalueella, joka oli tarkoitus muuttaa tiiviiksi uudeksi kaupunginosaksi.

Alueen suunnittelussa toimisto keräsi ensin tietoja aluetta ympäröivistä asioista, kuten pääliikenneväylistä ja rakennuskannasta. Kerätyt tiedot toimivat syöteinä algoritmissa, joka hyödynsi Autodeskin Maya-ohjelmaa ja siinä käytettyä dynaamista vähimmäismatkatyökalua.

Työkalu simuloi Otto Frein vähimmäismatkajärjestelmää (*Frei Otto, Apparatus for computing minimal path systems, Institute for Lightweight Structures (ILEK), Stuttgart, 1988*) yhdistämällä eri pisteet (alueen ympäristön tärkeät liikenneväylät) toisiinsa lyhimällä mahdollisella etäisyydellä.

Ohjelma näin imitoi ihmiselle luontaista liikkumismallia, jolloin suunnitelman lopputuloseksi muodostui looginen, vähän kiertoreittejä sisältävä tieverkosto pohjaksi jatkokehitykselle.⁰²²

021 Abdullah, A. A. Zaha Hadid Form Making Strategies for Design. 4
022 Schumacher, Patrik. Parametricism A New Global Style for Architecture and Urban Design, 20.

3.4 JOHTOPÄÄTÖKSET

Tutkimukseni liittyy sarjaan tutkimuksia tietokoneen laskentatehon ja algoritmien hyödyntämisestä suunnittelutehtävissä. Tutkimus myös pyrkii osaltaan selittämään suunnitteluvaiheessa tehtäviä ratkaisuja ja kytkemään nämä kaksi opinnäytetyön osa-aluetta toisiinsa.

Siinä missä algoritmien suunnittelu on hyvin tehokas, mutta kapea työskentelytapa, niin generoimalla sitä tietokonetta ja sen laskentatehoa hyödyntäen laajennetaan sen hyöty moninkertaiseksi.

Tutkimus keskittyy kehittyvään tieteenalaan, jolla on merkittävä kasvupotentiaali ja mahdollisuudet vaikuttaa tulevaisuuden suunnittelutyöhön. Tulevaisuudessa tietotekniikka, tekoäly ja innovatiiviset ohjelmistot tulevat kehittymään, ja mitä todennäköisimmin, vielä kiihtyvällä vauhdilla.

Kehityssuunta tietokoneiden kehityksessä ja innovaatiot, kuten kvanttietokoneet, saattavat mullistaa tietokoneiden laskentatehon erityisesti monimutkaisissa, paljon eri vaihtoehtoja sisältävissä laskutoimituksissa. Graafisten ohjelmointirajapintojen yleistymisen ja helppokäyttöisyys ovat tuoneet ne lähelle sitä pistettä, jossa ne alkavat olla suunnittelijoille enemmän hyödyksi kuin haitaksi ja jolloin laskennan hyödyntäminen sekä ohjelmointi alkavat muodostaa selkeän osan arkkitehtisuunnittelua.

Tutkimukseni perusteella kehitystä aihealueen sisällä selkeästi tapahtuu, se vain on luonteeltaan hyvin kokeilevaa, prototyyppimäistä ja hidasta. Aihealueen ymmärtämiseen vaaditaan usein rakennustaiteellisen kiinnostuksen lisäksi myös teknologista mielenlaatua sekä positiivista suhtautumista teknologiaan.

Tutkimuksen perusteella on myös huojentavaa tehdä havainto, ettei tietokone tai tekoäly kuitenkaan tule koskaan täysin korvaamaan ihmistä arkkitehtisuunnittelussa, johtuen sekä ihmisen ylivoimaisesta luovuudesta ja innovatiivisuudesta, että arkkitehtuurissa esiintyvistä, mittaamattomista arvoista, joita on vaikea tai jopa mahdoton kääntää tietokoneen ymmärtämään muotoon.

4.3 RESURSSIT

Projekti hyvin tietokonepohjaisena vaati toteutukseensa varsin paljon erilaisia ohjelmia ja sisältöä. Onnekasta tilanteessa oli se, että kaikista ohjelmista löytyi joko käytettävissä oleva opiskelijaversio, tai ne olivat ilmaisia ns. 'subscription'-pohjaisia.

Algoritmista ja generatiivisesta suunnittelusta kiinnostuneille löytyy kaikista ohjelmista varsin hyvin malliesimerkkejä jotka auttavat perusteiden ymmärtämisessä. Tutustuminen on hyvä aloittaa yksinkertaisiin algoritmeihin tutustumisella ja niiden luomisella.

Projektissa käytetyt pääasialliset ohjelmat olivat Revit, Dynamo sekä Project Fractal.

Autodesk Revit

<https://www.autodesk.com/products/revit/overview>

Revit on rakennuksen tietomallintamiseen ja suunnitteluun tarkoitettu suunnitteluohjelma joka on tarkoitettu arkkitehdeille, rakennesuunnittelijoille, talotekniikan suunnittelijoille ja urakoitsijoille. Ohjelman kehittäminen alkoi Yhdysvalloissa vuonna 1997 ja AutoCAD-ohjelmista tuttu Autodesk osti ohjelman ja sen kehitystiimin vuonna 2002. Suomessa Revit on arkkitehtisuunnittelussa yksi yleisesti käytössä olevista suunnitteluohjelmista ja sen pääasiallisena kilpailijana toimii Unkarilaisen Graphisoftin ArchiCAD.

Revitin avulla suunnittelija voi tietokoneavusteisesti suunnitella rakennuksen tietomallia kolmiulotteisesti ja ottaa mallista myös 2D-sisältöä. Ohjelmaan voidaan tuoda muiden suunnittelijoiden tietomalleja ja suunnittelu onkin ajateltu tapahtuvan laajassa yhteistyössä eri alojen suunnittelijoiden kesken. Viimeisimpinä uudistuksina Revitillä kyetään nykyään tekemään myös 4D-tietomallintamista, eli tietomallia johon on lisätty yhdeksi määreeksi aika, jolloin mallin avulla voidaan seurata rakentamisen ja rakennuksen eri vaiheita.⁰²³

Revit toimi tämän opinnäytetyön pääasiallisena suunnitteluohjelmana algoritmista saadun kerrostasopohjan jatkojalostuksessa ja valmiiksi suunnitelmaksi työstämisessä.

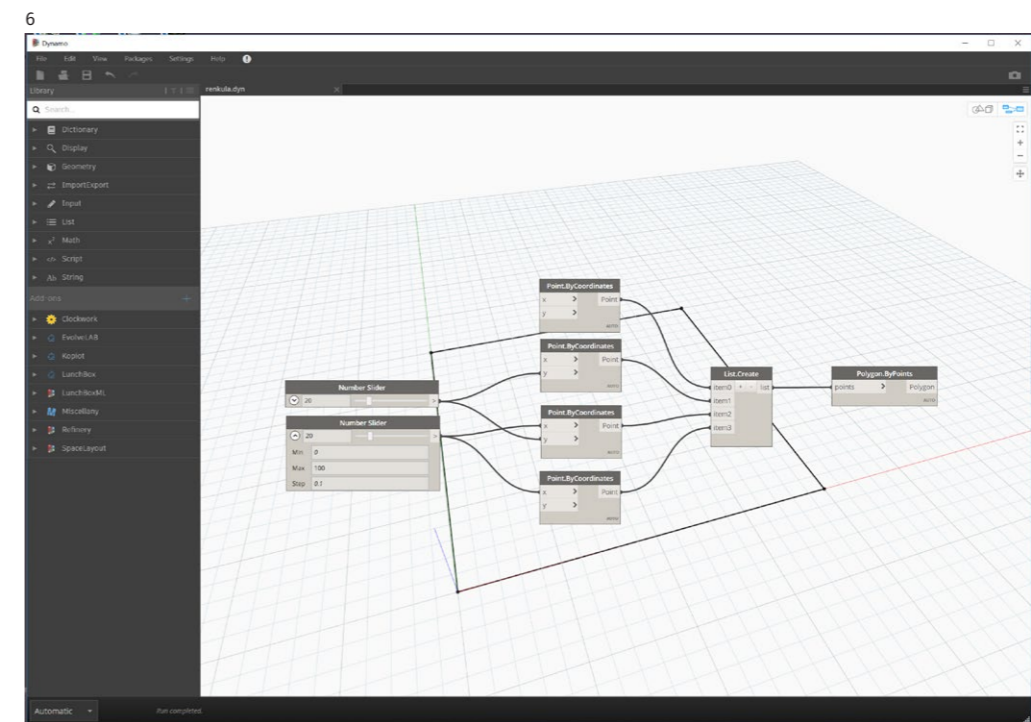
023 Autodesk Revit, www.autodesk.com.

Autodesk Dynamo

<https://www.autodesk.com/products/dynamo-studio/overview>

Autodesk Dynamo on avoimen lähdekoodin graafinen ohjelmointiympäristö, jonka saa joko itsenäisenä Studio Dynamona tai Revitiin liitettynä versiona. Ohjelman pääasiallisena tarkoituksena on antaa käyttäjilleen kyky visuaalisesti ohjelmoida Revitin käyttäytymistä, määrittellä logiikkaa tai ohjelmoida käskyjä yhdistämällä visuaalista sekä tekstipohjasta ohjelmointia.

Ohjelma on kehitetty helppokäyttöiseksi ja helposti lähestyttäväksi ohjelmointikäyttöliittymäksi sekä suunnittelijoille, että perinteisemmille ohjelmointijille.⁰²⁴ Opinnäytetyössä Dynamo toimi algoritmin ohjelmointityökaluna ja josta viimeinen, generatiivisen algoritmin avulla tuotettu vaihtoehto oli helppo siirtää Revitiin jatkotyöskentelyä varten.



6 Tyypillinen Dynamo-näkymä. Informaatio kulkee näkyvässä näkyvien solmujen (nodes) läpi vasemmalta oikealle solmuja yhdistävien lankojen perusteella. Kuvassa näkyvä yksinkertainen algoritmi, joka luo mallitasolle (taustalla) 20 x 20 ruudun kokoisen neliön sen kulmapisteiden perusteella.

024 Dynamo Primer, What is Dynamo?

EvolveLAB

<https://www.evolvebim.com/>

EvolveLAB on rakennussuunnittelun ammattilaisten yhteisö, jonka tarkoituksena on jakaa sekä tietomallintamisen, automaattisten prosessien ja laskentaan perustuvan suunnittelun tietoutta, algoritmeja sekä neuvoja. Yhteisö toimii pääasiallisesti maksuttomana, liittymispakollisena verkkosivuna foorumeineen ja opastusvideoineen. Yhteisö myös myy ammattilaiskäyttöön tarkoitettuja dynamon algoritmeja.

Tässä projektissa käytetty Dynamon algoritmi on EvolveLABin levittämä, mutta raskaasti muokattuna. Algoritmi on osa vapaasti ladattavaa pakettia, joka sisältää rakennuksen massoitteluun, kerrospohjan tilojen sijoitteluun ja julkisivun sommitteluun liittyviä algoritmeja. Paketti on kiinnostuneille löydettävissä DynamoPackagesista hakusanalla 'EvolveLAB'

DynamoPackages

<https://dynamopackages.com/>

DynamoPackages on sekä selaimesta, että ohjelman omasta latausvalikosta löytyvä kirjasto ohjelman käyttäjien luomille ja jakamille räätälöidyille algoritmeille. Kirjastosta löytyy sekä yksittäisten henkilöiden luomia että oikeiden yritysten ja yhteisöjen puhtaasti ammattikäyttöön luotuja algoritmeja. Projektissa käytetty EvolveLABin kehittämä algoritmi on löydettävissä täältä. Muita projektissa vähäisesti käytettyjä tai mainitsemisen arvoisia, kirjastosta löytyviä algoritmeja ovat:

- Space plan Generator (*Subhajit Das, Georgia Institute of Technology*)
- Clockwork for Dynamo 2.x
- Miscellany, Miscellaneous nodes for Dynamo

Project Fractal / Project Refinery Beta

<https://www.autodesk.com/solutions/refinery-beta>

Aloittaessani tätä projektia syksyllä 2018 Project Fractal oli selainpohjainen vaihtoehto/optimointialgoritmi Dynamon itsenäiselle versiolle Studio Dynamolle. Ohjelman kehittäminen lakkautettiin vuoden 2018 lopussa, mutta tilalle tuli uusi, hyvin samankaltainen ohjelma Project Refinery. Refineryn voi asennuksen jälkeen käynnistää suoraan Dynamosta.

Molemmat ohjelmat ovat siis generaattoreita, eli ne suorittavat Dynamossa luotua algoritmia useita kertoja tuottaen lukuisia vaihtoehtoja joita voidaan mitata käyttäjän luomien mittarien perusteella.

Ennen generointia ohjelmalle on tarkoitus kertoa, mitä Dynamossa määriteltyjä aloitusparametrejä (input) se generoi ja miten. Vakio-asetuksina ohjelmassa on vaihtoehtoina optimointi-, ristetulo-, sekä näennäissatunnaisgeneraattorit.

Ennen generointikierrosta generaattoriin määritellään myös variaatio, eli mitkä mittarisolmut määrittelevät lopputuloksen (output) joita se tarjoaa onnistuneen suorituksen jälkeen. Generoitava algoritmi voi myös epäonnistua tehtävässään mikäli generointivaiheessa syötettävistä arvoista syntyy ristiriita (tässä tapauksessa esimerkiksi liian suuri määrä ja/tai liian suuria asuntoja kerrospohjan sisällä)

Onnistuneen generoimiskierroksen jälkeen ohjelma antaa käyttäjälleen mahdollisuuden mittareiden perusteella tutkia, järjestää ja suodattaa algoritmin aikaansaamia vaihtoehtoja paremmuusjärjestykseen.⁰²⁵

5. HARJOITUSTYÖ

5.1 ALOITUS

Suunnittelutyö alkoi tammikuussa 2019. Aloitukseen liittyi sopivien ohjelmistojen valinta, referenssien valinta, algoritmien löytäminen sekä niiden soveltuvuuden testaaminen ja räätälöinti.

Suunnittelun aloituksessa oli haastavaa löytää tasapaino rakennustaiteellisen ja tietoteknisen lähestymistavan väliltä. Siinä missä ensin mainitussa lähestymistavassa on selkeä jo opittu tapa suunnitella referenssien, luonnostelun ja asteittaisen viimeistelyn kautta, on algoritmisessa suunnittelussa isompi rooli heuristisella yritys-erehdys tyyppisellä lähestymisellä. Algoritmin ohjelmoinnissa käytin runsaasti aikaa siihen, että saisin sen toimimaan tismalleen haluamallani tavalla, ennen kuin pystyin luottamaan sen satunnaisesti generoitujen vaihtoehtojen laatuun.

Laadulla tarkoitetaan tämän opinnäytetyön algoritmisissa kerrostasopohjan luonnostelua ja siinäkin suhteessa lähinnä eri asuntojen ja käytävän sijoittamiseen tiettyjen ulkorajapintojen (tontti) sisään järkevästi, tehokkaasti ja jatkokehityksen kannalta hyvin.

Generatiivisella algoritmilla pyrin tällöin luomaan kerrostalorakentamisessa yleisen ns. *mallikerroksen* jolla tarkoitetaan suunnittelun periaatteet määrittävää kerrosta, jota myöhemmissä vaiheissa monistetaan rakennuksessa joko identtisenä tai mahdollisuuksien mukaan varioiden.

5.2 ARKKITEHTUURIREFERENSSIT

1. Baltyk on MVRDV:n ensimmäinen suunnitteluprojekti Puolassa. Poznániin valmistunut rakennus herätti valmistuessaan kaupungissa runsaasti huomiota. Rakennuksen inspiraation lähteenä toimi arkkitehti Marek Leykamin samaiseen kaupunkiin suunnittelema Okrąglak (Rotunda) vuodelta 1954. Rakennuksen kaikki 16 kerrosta ovat erimuotoisia ja saavat aikaan hyvin erikokoisen rakennuksen, riippuen katselukulmasta. Suunnittelutoimiston perustajaosakkaan Nathalie de Vriesin mukaan suunnittelun lähtökohdat olivat luoda edustava toimistorakennus, jossa kaksi ensimmäistä ja ylin kerros on pyhitetty julkisille toimintoille ja loput toimistolle. Julkisine toimintoina rakennuksessa toimivat myymälä, catering-palvelu, kuntosali, panoraamaravintola sekä ylimmän kerroksen jazz-klubi.

Rakennuksen muotokielessä lähtökohtana oli aluksi rakennuksen muotoilu katutasossa sisäpihalle syntyvät aukion ehdoilla, mutta myöhemmin massan muotoilua jatkettiin jokaisessa kerroksessa, luoden ylempien kerrosten eteläosaan diagonaalisen porrastuksen. Porrastuksen ansiosta rakennukseen muodostuu aurinkoisia terasseja toimistojen käytettäväksi. ^{026, 027}

Valitsin tämän kohteen referenssiksi siitä syystä, että se sisältää hyvin paljon yhtäläisyyksiä valitsemani rakennuspaikan kanssa. Tällaisia yhtäläisyyksiä ovat muun muassa melko korkea rakennus (16. krs), erikoisen muotoinen tontti, merkittävä sijainti risteys/torialueella sekä samankaltainen tilaohjelma (liiketilaa katutasossa, ylin kerros muussa kuin pääkäyttötarkoituksessa)

7



7

Baltyk
Poznań, Puola.
MVRDV
2012

Lähde:
Flickr - Wojtek Gurak

Creative Commons Attribution-
NonCommercial 2.0 Generic (CC
BY-NC 2.0)

2. Toinen referenssi kohteena on samaisen arkkitehtitoimiston Rotterdamiin suunnittelema tornitalo The Sax. Kirjoitushetkellä rakennus on vasta suunnitelman ja rakentamisen aloituksen tasolla voitettuaan suunnittelukilpailun Rotterdamin Wilhemina-pierin alueella.

Rakennuskokonaisuus koostuu kahdesta vierekkäisestä asuintornista (70 ja 150m) jotka on yhdistetty toisiinsa 70 metrin korkeudella sijaitsevalla hotellilla. Kokonaisuus käsittää yhteensä 82000 m² kerrosalaa ja tulee sisältämään 450 asuntoa, hotellin, hyvinvointikeskuksen, pysäköintiä sekä kaupallisia toimintoja.

Tornien ulkomuoto muodostuu yksinkertaisesta suorakulmaisesta muodosta, josta työntyy ulos asuntokohtaisia, suuria ulokkeita, joiden määrä harvenee rakennusta ylöspäin noustaessa. Ulokkeet ovat aina olohuoneen yhteydessä ja niiden käyttötarkoituksen voi määrittellä itse; siinä missä yksi asukas voi sijoittaa sinne sohvoryhmän, voi toinen käyttää ulokkeen tilaa vaikkapa suuren ruokapöydän sijoitteluun. ⁰²⁸

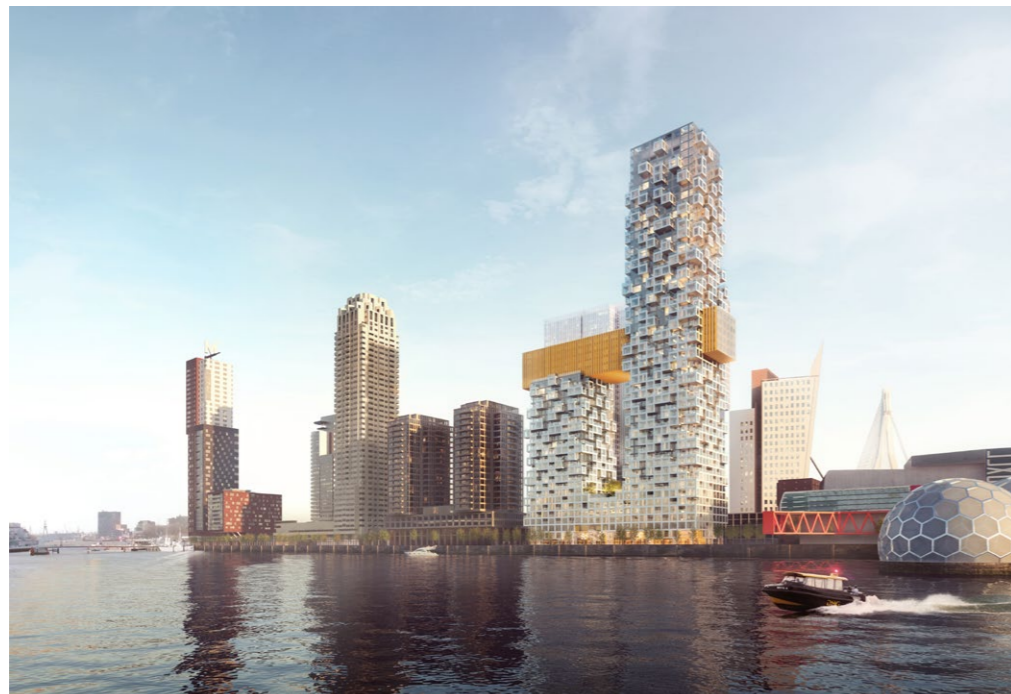
Ulokkeiden ansiosta rakennuksen julkisivuihin saadaan aikaiseksi jännittävä, satunnainen näköinen ja vilkas tunnelma ja sellaista pyrin tavoittelemaan myös omassa suunnitelmassani, sillä mielestäni se sopii hyvin yhteen alueen historian ja merikontti-teeman kanssa.

8

8
The Sax
Rotterdam, Alankomaat
MVRDV
2017-2022

Lähde:
MVRDV Press Room

MVRDV deSax Day
copyright MVRDV



Nihdin korttelit tulevat muodostumaan pääosin suljetuista yksiköistä, joissa yksi kulma on suunniteltu korkeaksi rakentamiseksi (AK-1). Korkean rakentamisen kulmaan ei ole määritelty kerroskorkeutta ja tämän tarkoituksena on aikaansaada kortteleihin selkeästi erottuvaa, hyvin korkeaa rakentamista, riippuen rakennuttajansa kunnianhimesta.

Opinnäytetyön rakennuksen suunnittelualaue, kortteli Spaaki, on Nihdin alueen yksi keskeisimmistä sijainneista. Kortteli, ja sen kaksitoistakerroksinen kulmarakennus muodostavat reunan aukiolle, joka sijaitsee risteyskohdassa, pitkien tielinjojen päässä keskeisellä sijainnilla ja meren rannalla.

Aukiota rajaa pohjoisessa Nihdinkanava sekä sen ylittävä vielä määrittelemätön silta. Itään jatkuu Korkeasaareen ja Laajasaloon vievät Kruunusillat ja etelään käveltävä rantabulevardi. Länteen jatkuva Konttisatamankatu on bulevardityyppinen, nopeilla pyöräteillä varustettu raitiotiekatu, jolla on tulevaisuudessa yhteys Merihakaan.

Rakennuksen edessä sijaitseva aukio tulee olemaan poikkeuksellinen liikennejärjestelyjensä takia sillä aukiolla ei ole ajoneuvoliikennettä. Ajoneuvot käännetään pohjoisesta saavuttaessa Nihdinrantaan ja vain raitiovaunu ja kevyt liikenne saavat jatkaa matkaansa Konttisatamankadulle.

Tontille sijoitettava rakennuskokonaisuus poikkeaa myös kaupunginosan aiemmin mainituista umpikortteleista, sillä se on yhtenäinen, kolmesta massasta koostuva kortteli. Massoista alin on kaksikerroksinen ja sen päälle tulevat sijoittumaan kaksi erimuotoista massaa, toinen neli- ja toinen kaksitoistakerroksisena. Haastattelussa kävi ilmi, että suunnitteluratkaisuna massojen, sekä eri kerrosten ulkonäön on ajateltu poikkeavan toisistaan.

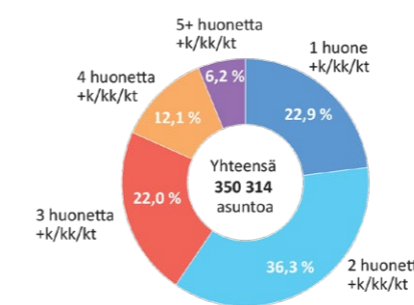
Kahteen alimpaan kerrokseen on suunniteltu koko aluetta tukevia kaupallisia toimintoja, sekä asumista tukevia toimintoja. Kaksikerroksisen massan päälle, asuintornien väliin jäävä alue on suunniteltu kansipihaksi.

Alueen korkeat rakennukset on suunniteltu muistuttamaan merikontteja ulko-verhouksestaan, muistuttaen alueen käyttöhistoriasta vilkkaana merisatamana. Tasaisen aluejulkisivun aikaansaamiseksi rakennuksiin on määritelty sisäänvedetyt parvekkeet

Rakennuksen toiminnoiksi oli ennalta määritelty katutasen suuri (800m²) liiketila sekä ylimmän kerroksen monikäyttötila. Näihin tiloihin suhtauduin suunnitelmassani niin, että pidin niitä irrallisina, muuhun kuin asuinkäyttöön tarkoitettuina tiloina. Pysin tällöin luomaan asuinkerroksille tyypillisen ns. *mallikerroksen* jolla tarkoitetaan suunnittelun periaatteet määrittävää kerrosta, jota myöhemmissä vaiheissa monistetaan rakennuksessa joko identtisenä tai mahdollisuuksien mukaan varioiden. Mallikerroksen asunotajakauman lähteenä toimi Helsingin huoneistotyyppijakauma vuodelta 2015.

11

Kuvio 19. Helsingin huoneistotyyppijakauma 31.12.2015



Jakauman ja kerroksen pinta-alan perusteella määrittelin summittaisen vaihteluvälin eri huoneistotyypeille:

- Yksiö 3 - 5
- Kaksio 4 - 6
- Kolmio 1 - 3
- Neliö 0 - 1
- Viisiö 0 - 1

11 Helsingin huoneistotyyppijakauma

Lähde: Asuminen alueittain Helsingissä 2015, Anna Kajosaari.

Helsingin kaupunki tietokeskus.

Määrän lisäksi, eri asunnoille piti määritellä mitat sekä pinta-alat minimi- ja maksimiarvoineen. Tähän käytin Helsingin kaupungin ohjearvoa eri asunotyyppien huoneistoalalle uudiskohteissa.

- Yksiö 30 - 35 m²
- Kaksio 40 - 45 m²
- Kolmio 58 - 75 m²
- Neliö 75 - 85 m²

Yhdessä asuntojen määrien ja kokojen, sekä tontin ulkoraja-arvojen kanssa minulla oli valmiina algoritmin syötteenä toimivat parametrit.

5.4 PROSESSI

Algoritmi, joka toimi kehittämäni algoritmin pohjana tässä opinnäytetyössä, oli osa EvolveLABin levittämää pakettia, joka sisälsi Dynamoon kehitettyjä algoritmeja käytettäväksi suunnittelun eri vaiheisiin.

Käyttämäni algoritmi Container.Packing ei ole alun perin EvolveLABin kehittämä, vaan heille päätyneet avoimen lähdekoodin ohjelmistojen jakopalvelun GitHubin kautta. Algoritmin alkuperäisen kehittäjän löytäminen teetti minulle salapolii-sityötä, sillä avoimeen lähdekoodiin liittyy usein monimutkainen muokaus- ja kehityshistoria sekä useat kehittäjät.

Selvitystyön tuloksena löysin kuitenkin, että algoritmin alkuperäinen kehittäjä on vänrikki Erhan Baltacioglu Yhdysvaltain ilmavoimien teknillisestä instituutista vuodelta 2001.

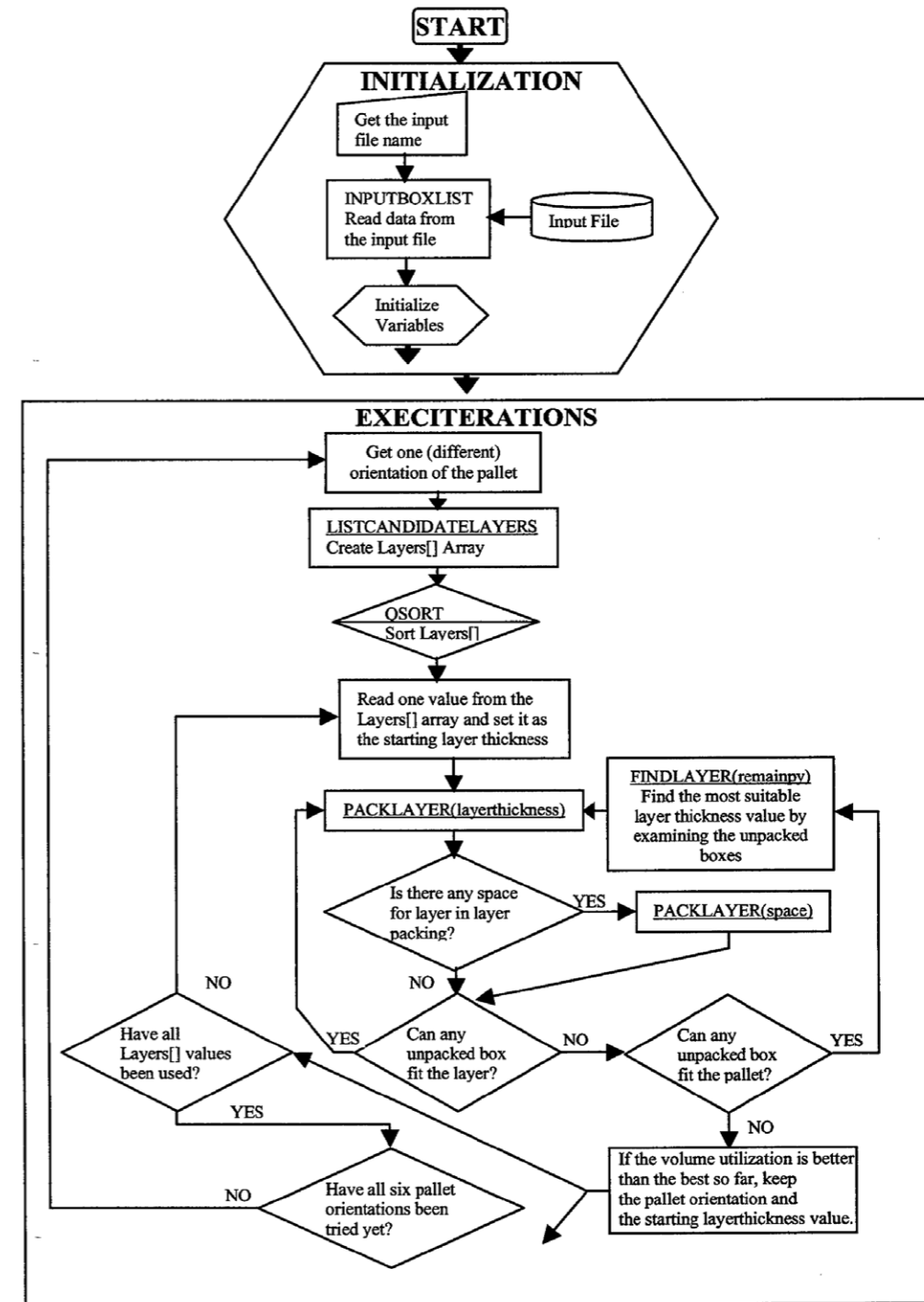
Tutkimuksessa Baltacioglu tutkii tavaran lastauksen näkökulmasta kiinnostavaa, mutta käytännössä erittäin vaikeasti toteutettavaa algoritmia joka vastaa kysymykseen; *'miten pakata mahdollisimman monta erikokoista kappaletta yhden, ennakkoon määritellyn kappaleen sisälle?'*

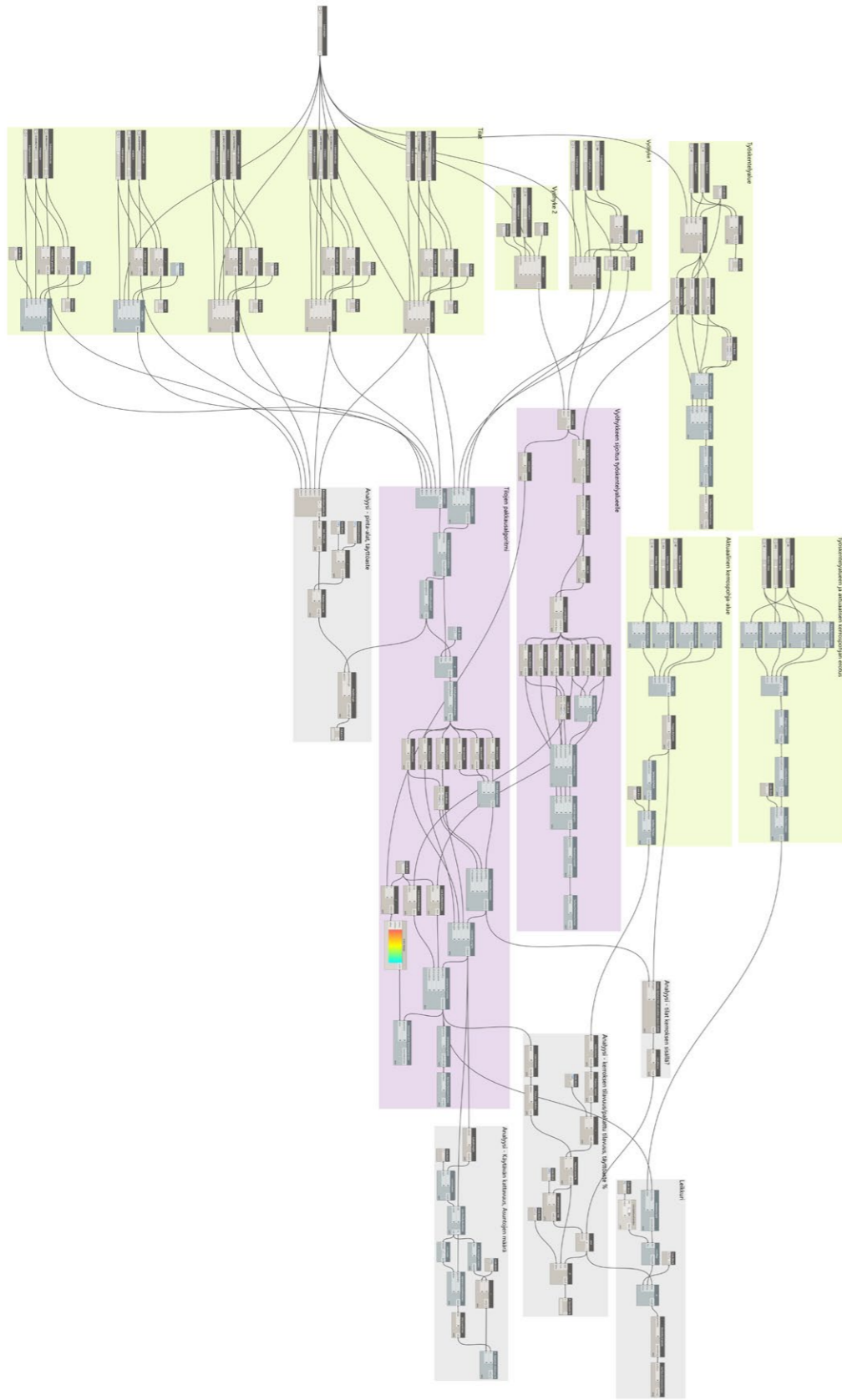
Lopputuloksena on hyvin monimutkainen approksimaatioalgoritmi, joka yrittää löytää pakkaukselle aina parhaimman mahdollisen ratkaisun ihmiselle järkevissä ajassa. ⁰²⁹

Myöhemmin algoritmin ovat löytäneet arkkitehtisuunnittelusta kiinnostuneet Dynamon käyttäjät ja sitä on muokattu eri tarpeisiin ja projekteihin sopivaksi. EvolveLABin rooliksi on tässä siis jäänyt algoritmin yksinkertaistaminen, helppokäyttöisyyden lisääminen ja levitys omalla nimellään.

Algoritmin käyttöönotto vei erityisesti projektin alussa hyvin paljon aikaa. Siitä huolimatta, että Dynamo käyttöliittymänä on hyvin yksinkertainen, voi sen avulla luodut algoritmit olla hyvinkin monimutkaisia ja sisältää paljon toisistaan riippuvia solmuja (engl. node)

⁰²⁹ Baltacioglu, Erhan. Distributor's Three-Dimensional Pallet-Packing Problem: Human Intelligence-Based Heuristic Approach.





Opinnäytetyöhön kehitetty algoritmi valmistui alkuvuodesta 2019. Ohjelmointiin kului noin 200 työtuntia.

Algoritmi toimii karkeasti vasemmalta oikealle ja se käyttäytyy niin että edellinen solmu vaikuttaa aina seuraavaan solmuun, johon se on kytketty. Algoritmi on värikoodattu kolmeen väriin, jotka edustavat sen kolmea prosessin vaihetta (kuten mainittu sivulla 18)

Ensimmäinen vaihe on vihreällä merkityt parametrisolmut, eli syöte (input). Parametrivaiheessa algoritmiin syötetään lukuarvoja suunnittelutehtävästä, tässä tapauksessa kerrostasopohjasta. Tällaista tietoa on esimerkiksi suunnittelualue (hieman rakennusalueen rajoja suurempi alue), rakennusalue, vyöhykkeet, tämän projektin erikoisuus eli yksi vino ulkoraja, eri asuntojen mitat minimi- ja maksimiarvoineen, näistä muodostuvat pinta-alat minimi- ja maksimiarvoineen sekä asuntojen määrät minimi- ja maksimiarvoineen (sivu 39).

Toinen vaihe on liilalla merkityt pakkausalgoritmit, jotka ovat em. EvolveLabin jakamasta paketista voimakkaasti tähän projektiin sopivaksi räätälöityinä. Pakkausalgoritmiin syötetään kaikkien eri asuntojen sisältämät arvot (esimerkiksi yksiö, 7 x 4,5 metriä, 31,5 neliometriä, 4 kappaletta,) joita se yrittää pakata ne annetun suunnittelualueen sisälle kaavion 11 perusteella.

Mikäli algoritmi onnistuu tehtävässään pakata asunnot rakennusalueen sisälle, on lopputulos nähtävissä ohjelman 3d-näkymässä, joka näkyy algoritminäkymän taustalla.

Kolmas ja viimeinen vaihe on harmaalla merkityt mitaussolmut. Mittaussolmut nimensä mukaisesti mittaavat onnistuneesta pakkausvaiheesta syntyneitä malleja.

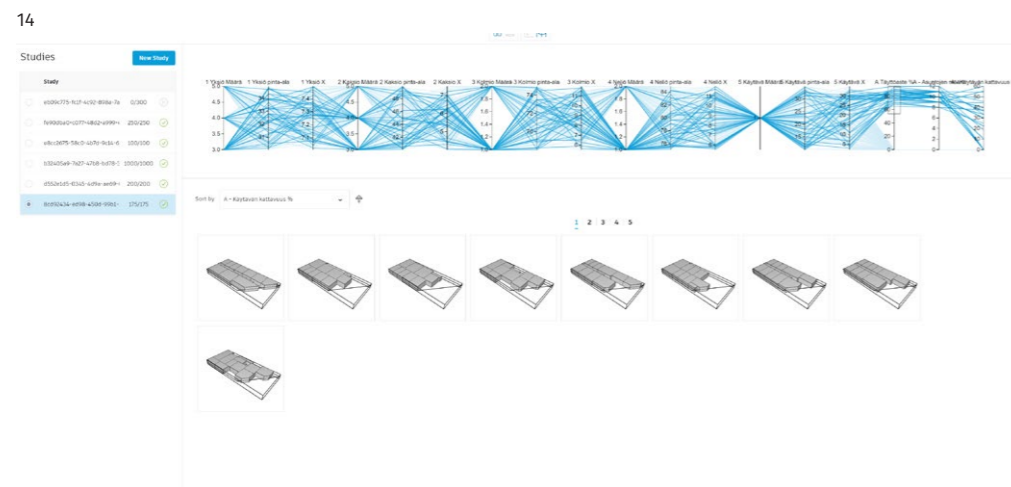
Mitattavia arvoja tässä algoritmissa olivat; eri asuntojen määrä, eri asuntojen koot, käytävien määrä ja niiden koot, pakkauksen täyttöaste (kuinka suuren osan rakennusalueen alasta pakatut tilat täyttävät) sekä käytävän kattavuusaste (kuinka moneen asuntoon käytävä ylettää). Näiden arvojen perusteella generaattorin luomat vaihtoehdot oli helppo järjestää paremmuusjärjestykseen ja valikoida näin jatkokehitykseen sopivin versio.

Prosessin kannalta on tärkeää tietää, että pakkausalgoritmi voi myös epäonnistua tehtävässään. Mikäli syötettyjä asuntoja on liikaa, ja/tai ne ovat liian isoja, eikä pakkausalgoritmi onnistu pakkaamaan niitä suunnittelurajojen sisälle, ohjelma ilmoittaa asiasta ja mallia ei muodostu 3d-näkymään. Samoin mitattavat arvot tippuvat nolliin; seikka, joka mahdollistaa generointivaiheessa kyseisten, epäonnistuneiden pakkausten suodattamisen pois vaihtoehdoista.

Kun algoritmi oli saatu tähän vaiheeseen, oli aika aloittaa sen avulla *generoiminen*. Generointi tapahtui erillisellä, Dynamoon kytketyllä ohjelmalla nimeltä Project Refinery.

Refineryyn on tarkoitus ennen generoimista määrittää tietyt esiasetukset. Näistä ensimmäisenä pitää määrittää mitkä solmut algoritmista toimivat syötteenä ja mitkä vasteena. Tässä työssä syötteenä toimivat algoritmista esiintyvät vihreällä merkityt syötesolmut, ja vasteena harmaalla merkityt mittaussolmut.

Liilalla värillä merkityn pakkausalgoritmin toimintaan Refinery ei ota kantaa. Refineryn roolina on ainoastaan syöttää syötteeseen erilaisia arvoja ja tämän jälkeen lukea mittareista kyseisillä arvoilla syntyviä tuloksia ja toistaa tätä prosessia niin monta kertaa, kun sen käyttäjä on määritellyt.



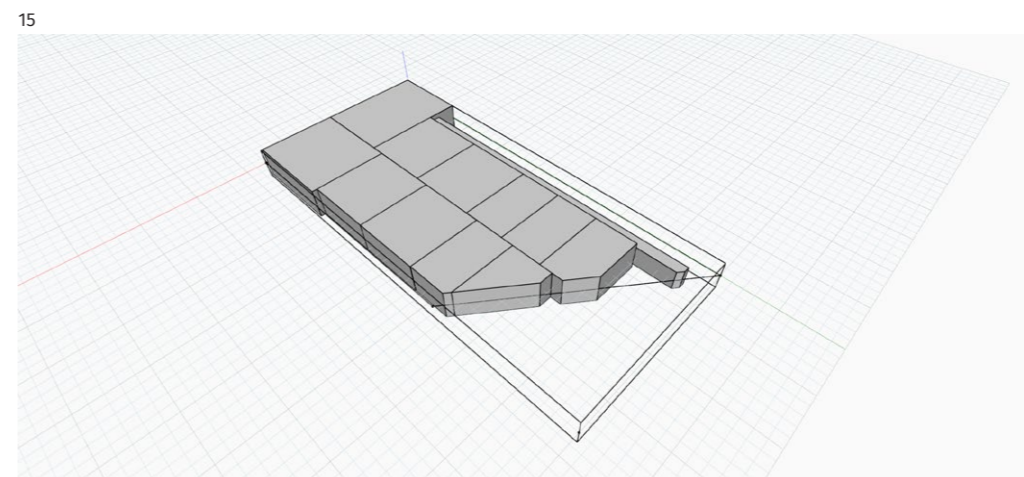
14 Näkymä Refineryssä onnistuneen generointikierroksen jälkeen. Näkymän vasemmassa laidassa näkyvät yksilöidyt generointikierrokset. Oikealla eri vaihtoehdot sekä jokaisen yksittäisen vaihtoehdon matka janakaavion syöteparametrien läpi. Yksityiskohta janakaavion oikeassa reunassa; vaihtoehdoista on rajattu pois ne yksilöt, joilla täyttöaste on 0% eli sellaiset, joissa algoritmi ei ole onnistunut pakkaamaan tiloja kerroksen sisälle.

Syötteen ja vasteen määrittelyn jälkeen Refineryyn pitää vielä määrittellä generointitapa. Generointitapa määrittää sen, millä tavalla ohjelma käsittelee sille syötettyjä arvoja ja pyrkii se esimerkiksi löytämään parhaimman mahdollisen vaihtoehdon vai tuottamaan vain runsaan määrän erilaisia vaihtoehtoja. Vakio-asetuksina ohjelmassa on vaihtoehtoina optimointi-, ristitulo-, sekä näennäissatunnaisgeneraattorit.

Tässä työssä generoin lopulta viisi tuhannen vaihtoehdon sukupolvea, joissa käytin kahta erilaista generointitapaa, näennäissatunnaislukua sekä optimointia.

Näennäissatunnaisgeneraattoria käytin ensimmäiseen, tuhat eri vaihtoehtoa sisältävään sukupolveen. Satunnaisuuteen perustuva generointi tarjosi ensimmäisessä vaiheessa parhaimman mahdollisen kattauksen mahdollisimman erilaisia vaihtoehtoja, sillä se määrittää syötteille lähes täysin satunnaiset arvot ja lopputulos on siten ennustamaton.

Ensimmäistä tuhannen vaihtoehdon joukkoa oli helppo suodattaa Refineryn luettelotyökalulla. Luettelolla voitiin järjestää vaihtoehdot esimerkiksi parhaimman täyttöasteen tai käytävien kattavuuden perusteella. Vaihtoehtojen keskimääräisesti parhaiten suoriutunut yksilö näytti tässä vaiheessa tältä:



15 Vaihtoehdon mittarit antoivat seuraavanlaisia lukemia
 - Asuntojen määrä: 11, 5 Yksiotä (32,8m²), 4 Kaksiota (43,4m²), 1 Kolmio (72,9m²) ja 1 Neliö (80,8m²)
 - Kerroskohjan täyttöaste: 86,4%
 - Käytävän kattavuus: 54,4%

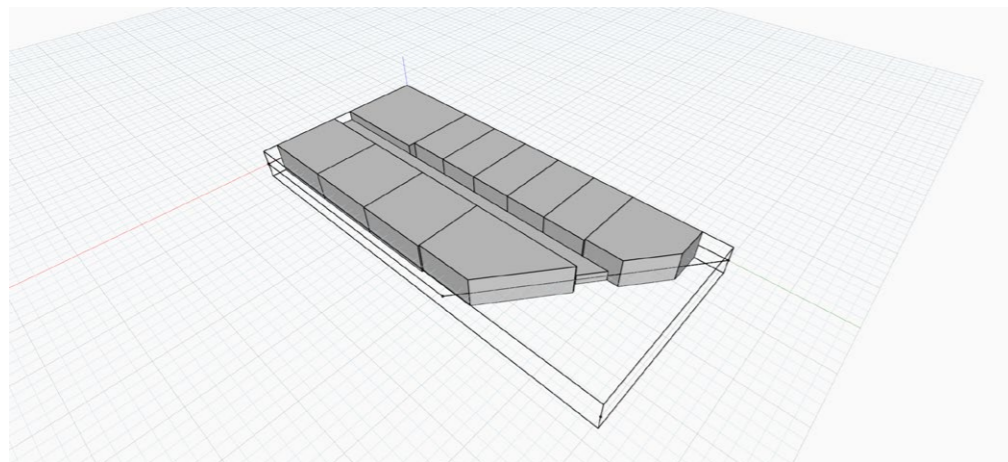
Seuraavan neljän sukupolven generointitapana oli optimointi. Optimointi eroaa satunnaisgeneroinnista siinä, että vasteelle, eli mittareille asetettiin Refineryssä tavoitteita, joita generaattori pyrkii täyttämään. Refineryssä tavoitteiksi voi määritellä yksinkertaiset *maksimointi* ja *minimointi*.

Tässä työssä tavoitteita asetettiin esimerkiksi täyttöasteen, asuntojen pinta-alojen ja käytävän kattavuuden mittareille, joista kaikille asetettiin tavoite maksimointi, sillä kaikki olivat mielestäni seikkoja joissa enemmän = parempi. Tällöin generaattori pyrkii tekemään generoidessaan syötteiden arvoihin sellaisia muutoksia, jotka lisäävät mittarien tavoitetta, tässä tapauksessa esimerkiksi täyttöastetta.

Ensimmäisen, satunnaisesti generoidun sukupolven paras vaihtoehto valittiin jatkokehitykseen ja siitä tehtiin vielä neljä, optimointiin perustuvaa tuhannen vaihtoehdon sukupolvea.

Jokainen sukupolvi oli aina edellistä hieman kehittyneempi. Viimeisen eli viidennen sukupolven parhaiten suoriutunut vaihtoehto muistutti tässä vaiheessa ulkonäöltään jo oikeaa rakennuksen kerrosta ja siitä syystä päätin tässä vaiheessa siirtää sen Revitin puolelle ja jatkaa työskentelyä siellä.

16



16

Viimeisen sukupolven vaihtoehdon mittarit antoivat seuraavanlaisia lukemia
- Asuntojen määrä: 11, 5 Yksiötä (32,5m²), 4 Kaksiota (49,5m²), 1 Kolmio (71,0m²) ja 1 Neliö (79,5m²)
- Kerrospohjan täyttöaste: 89,5%
- Käytävän kattavuus: 100%

Sukupolvien parhaimman vaihtoehtojen valitsemisessa en täysin luottanut mittarien antamaan tulokseen vaan tarkistin mallin aina myös itse. Tähän syynä oli vielä kehitysvaiheessa olevien ohjelmien viriheerkyys ja algoritmin yksinkertaisuus, seikkoja, jotka johtivat monesti hyvin epäkelvojen vaihtoehtojen syntymiseen. Algoritmiin on kuitenkin mahdollista ohjelmoida ehtolauseilla jos-käskyjä asettamaan tiukkoja ehtoja mitattujen suunnittelutavoitteiden täyttymisestä.

Algoritmin avulla luotu mallikerros todisti itsensä jatkokehitysvaiheessa, sillä lopullisessa suunnitelmassa esiintyvä mallikerros (4-11.kerros) muistuttaa vielä hyvin paljon algoritmin tuotosta. Jatkokehitysvaiheessa tulleet muutokset olivat lähinnä kahden seinälinjan siirtyminen esteettömyysvaatimusten täyttämiseksi sekä siitä syystä pienentyneen nelihuoneisen asunnon alentaminen kolmihuoneiseksi.

Kyseisen seikan huomioonottaminen ohjelmointivaiheessa olisi välttänyt tilanteen syntymisen, ja kyseinen seikka myös osoittaa generatiivisten järjestelmien erään heikkouden. Heikkous liittyy takaisinkytkennän työläyteen ja arkkitehtisuunnittelussa tyypilliseen työtapaan jossa hyvin pitkälle edistyneessä suunnitelmassa voi ilmaantua ongelmia, joiden ratkaisu vaatii paluuta suunnittelun alkutekijöihin ja luonnosteluun.

Arkkitehtisuunnittelu ei siis luonnostaan ole lineaarista vaan eri tarkkuustasojen sekä luonnostelu-viimeistely -akselien välissä poukkoilemista, seikka joka on tietokoneelle ja algoritmille täysin vieras. Tästä syystä, oman näkemykseni mukaan, generatiivisten järjestelmien rooli tulevaisuudessa onkin toimia tiettyjen, tarkasti määriteltyjen suunnittelutehtävien avustavina *työkaluina*.

Suhtautumalla näihin järjestelmiin työkaluina ja avustavina tuotteina, voidaan luopua vääristyneistä mielikuvista joissa järjestelmät uhkaavat koko arkkitehtisuunnittelun alaa ja tulevaisuudennäkymiä.

Seuraavassa kappaleessa esittelen suunnitelman planssipiennökset

5.5 RATKAISUT

Spaaki // Generatiivinen algoritmi suunnittelijan työkaluna

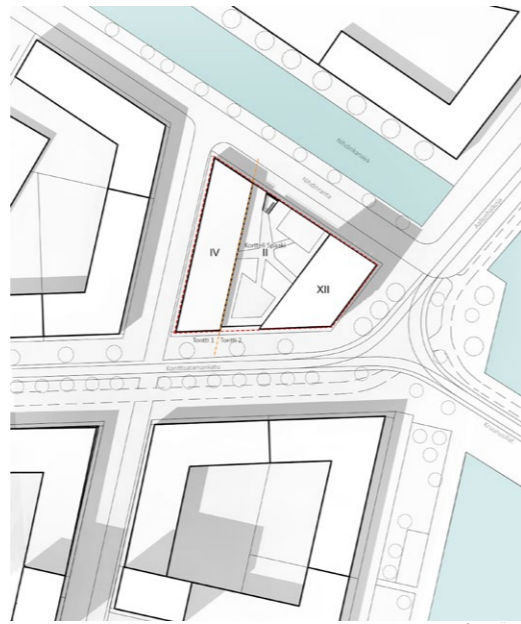
Asuinkerrostalo Sompasaaren Nihtiin



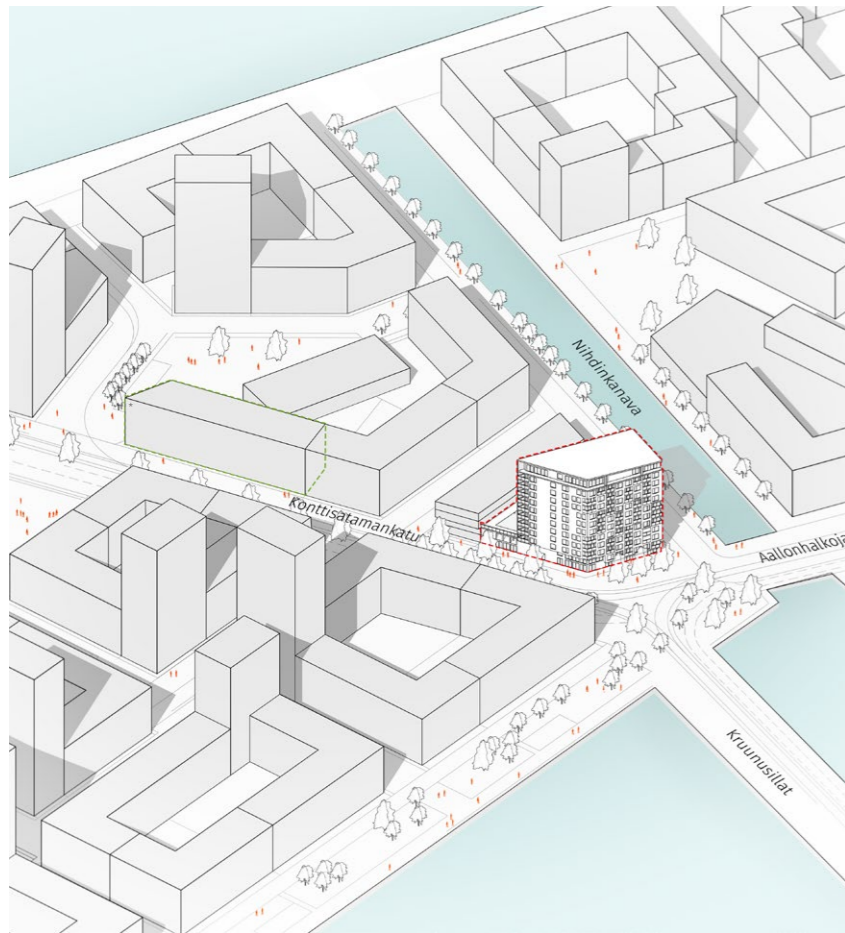
Havainnekuva idästä



Rakennuskartta 1:10 000



Asemapiirros 1:1000



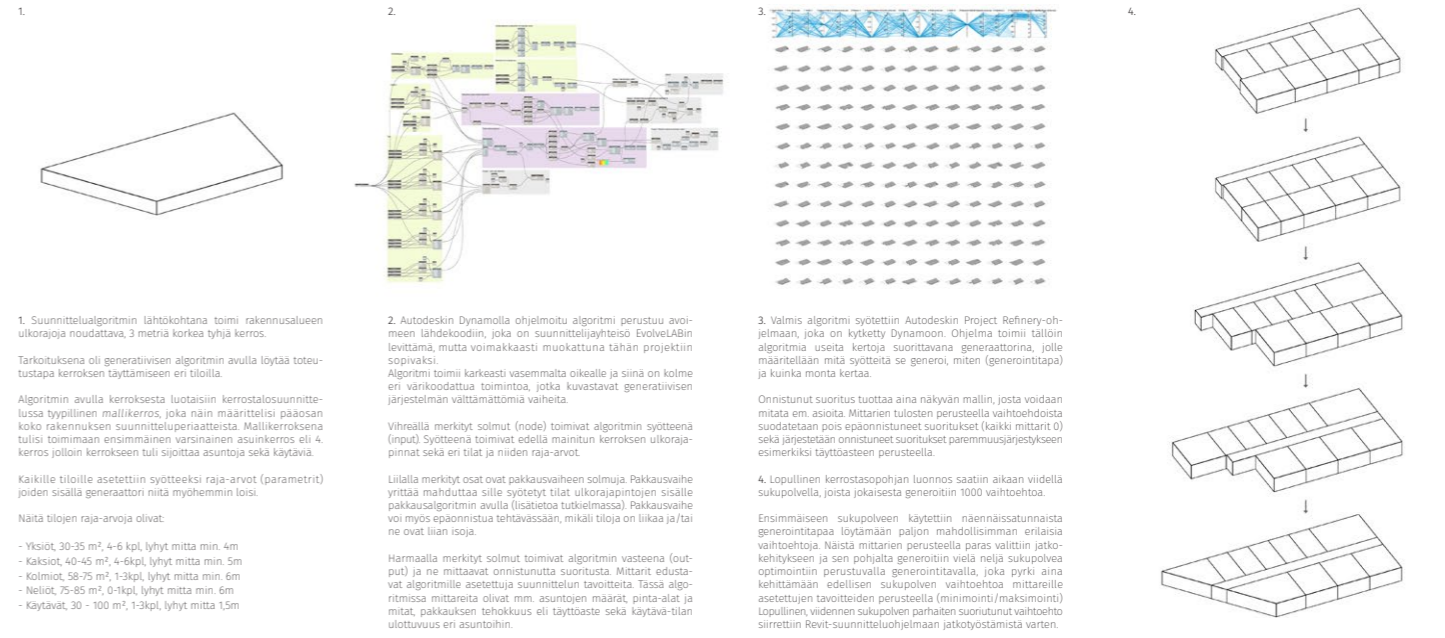
Sijoittuminen ympäristöön



Havainnekuva korttisatamankadulta

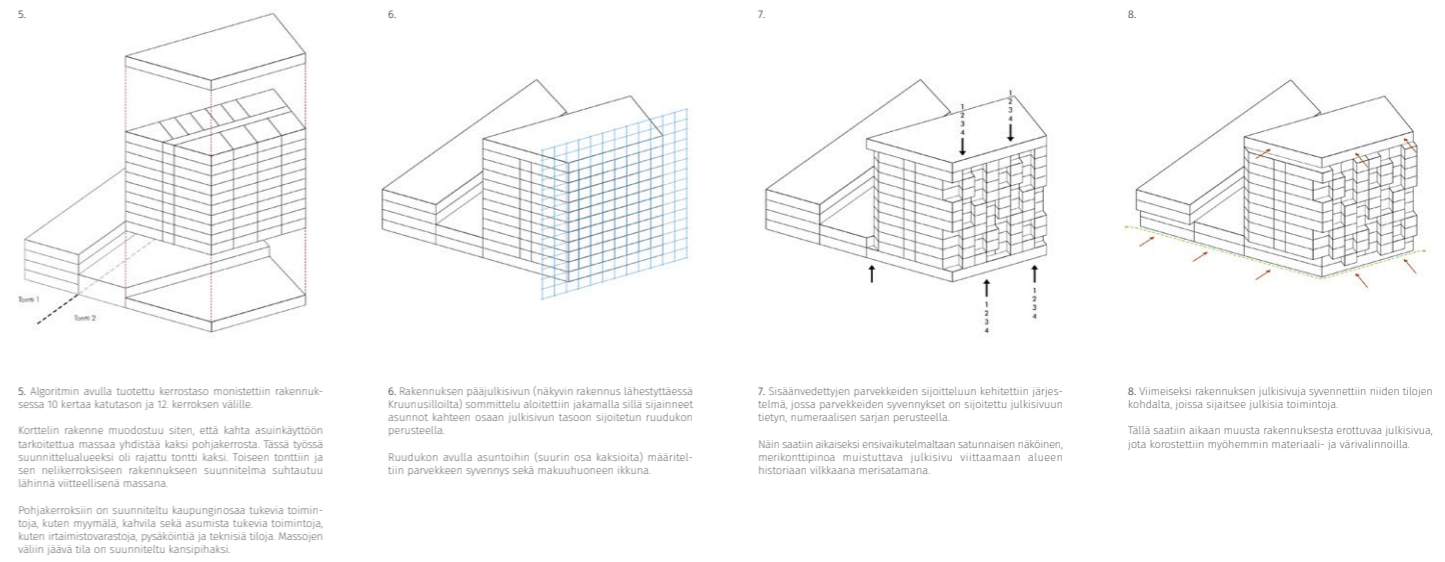
Spaaki // Prosessi

Algoritminen prosessi



Havainnekuva katutasosta

Muodonannollinen prosessi



Rakennuksen värимааlilla on suora viittaus merkintöihin ja alueen käyttöhistoriaan. Värin on valittu merkintäpöytästä otetusta valokuvasta ja jaettu rakennuksessa eri tiloille niiden käytötaroituksen mukaan. Näin rakennukseen saatiin aikaa huomiota herättävä, satunnaisen näköinen ja viikas ulkonäkö.

- - Yleiset, tekniset ja julkiset tilat
- - Kaksiot, alemmat kerrokset
- - Kolmiot
- - Kaksiot PK (pelikuvaa)
- - Yksiot
- - Kolmiot PK
- - Kaksiot, ylempät kerrokset
- - Yksiot PK

Rakennuksen julkiset tilat ovat sijoitettu aukion reunalle, keskelle sijalle. Myymälän ja kahvilan sisäänkäynnit ja ikkunat ovat suunnattu itään, lounasillalle ja meren suuntaan. Sama ideologia toistuu ylimmässä kerroksessa, pääosan asukkaiden käyttöön tarkoitettua tiloissa, joissa ylimmän kerroksen parhaimmat näkymät ovat rakennuksen yhteiskäyttötiloilla ja sauna-alueen näkymät ovat suunnattu länteen, itä-suunnikoon.

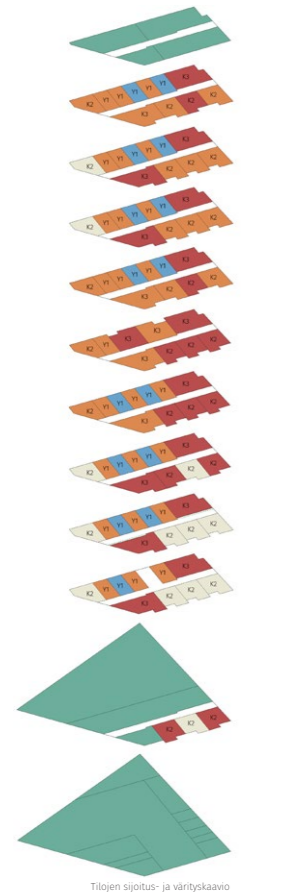
Kansipihan suunnittelussa on otettu huomioon pihan eri käyttäjäryhmät sekä viereinen, samaa pihaa käyttävä rakennus.

Pääosa asunnoista oli pelattavia, seikka jonka ansiosta julkisivun sommitteluun käytetty järjestelmä toimi halutulla tavalla. Asuntojen suunnittelussa on otettu huomioon toistettavuus sekä eri asukkaiden elämäntilanteet.

Algoritmin avulla luotu mallikerros todisti itsensä jatkokehitysvaiheessa, sillä tulopuolella suunnitelmassa esiintyvä mallikerros (4.kerros) muistuttaa vielä hyvin paljolti algoritmin tuotosta.

Jatkokehitysvaiheessa tulleet muutokset olivat lähinnä kahden seinälinjan siirtyminen (merkitty sijoitus- ja värityskaavioon 4. krs) esteettömyysvaatimusten täyttämiseksi sekä siitä syntyä pienentyneen nelihuoneisen asunon alentaminen kolmihuoneiseksi.

Opinnäytetyö vastaa tutkimuskysymykseen "Voiko generatiivista algoritmia käyttää työkaluna suunnitteluun apuna?" myöntävästi ja osoittaa sen tehokkuuden yksinkertaisessa, rajatussa suunnittelutehtävässä.



12. Kerros
 - Yhteiskäyttötilat
 - Suunnistasto
 - Pesula
 - 2 Irtaimistovarastoa
 - IV-konehuone

3-11. Kerros
 - 96 asuntoa
 - Y1 = Yksio
 - K2 = Kaksio
 - K3 = Kolmio

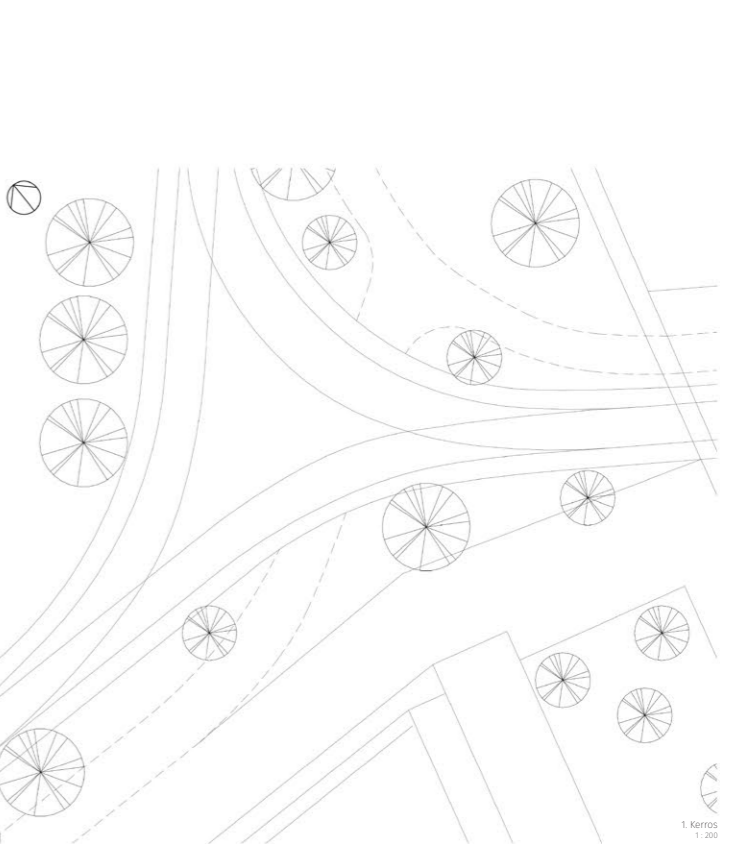
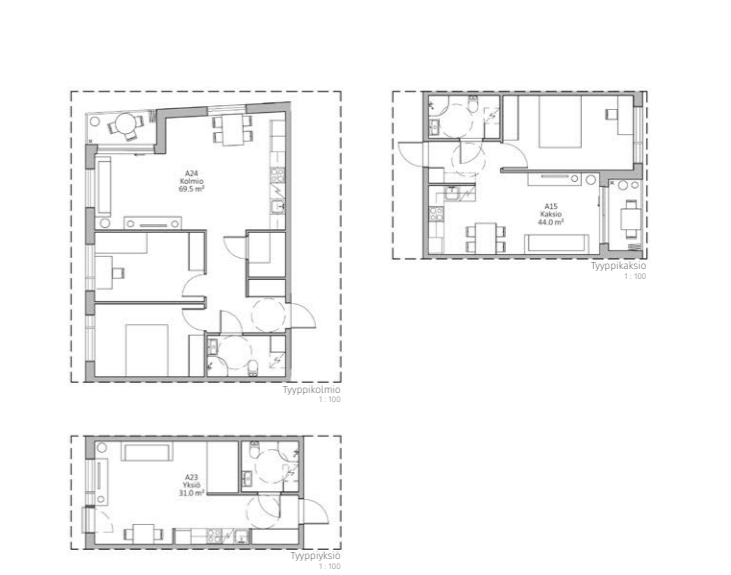
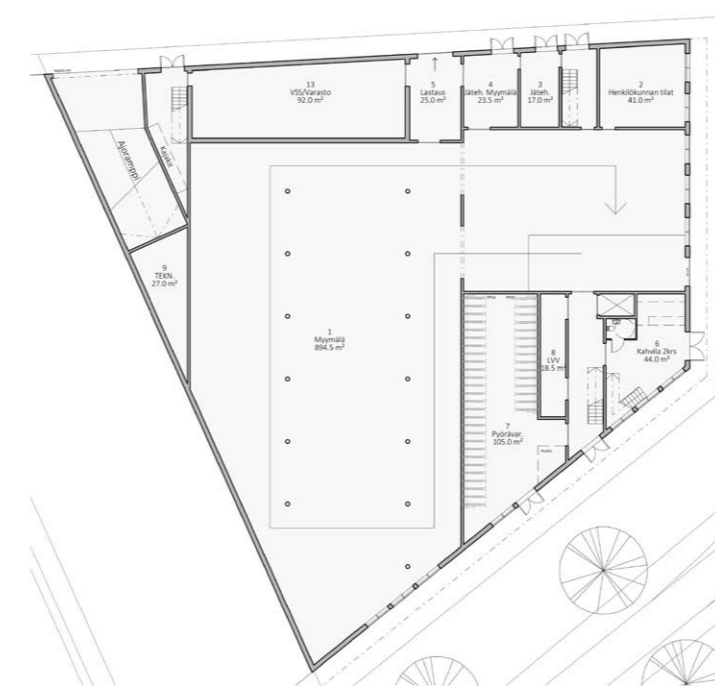
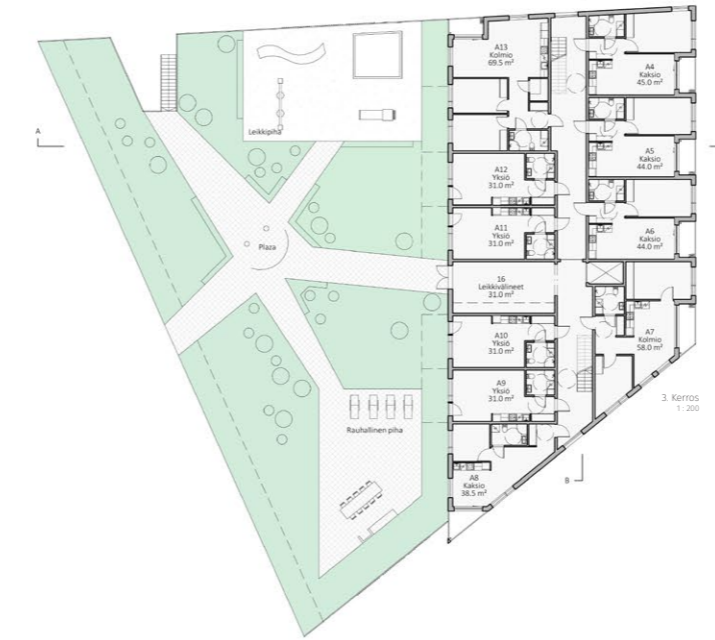
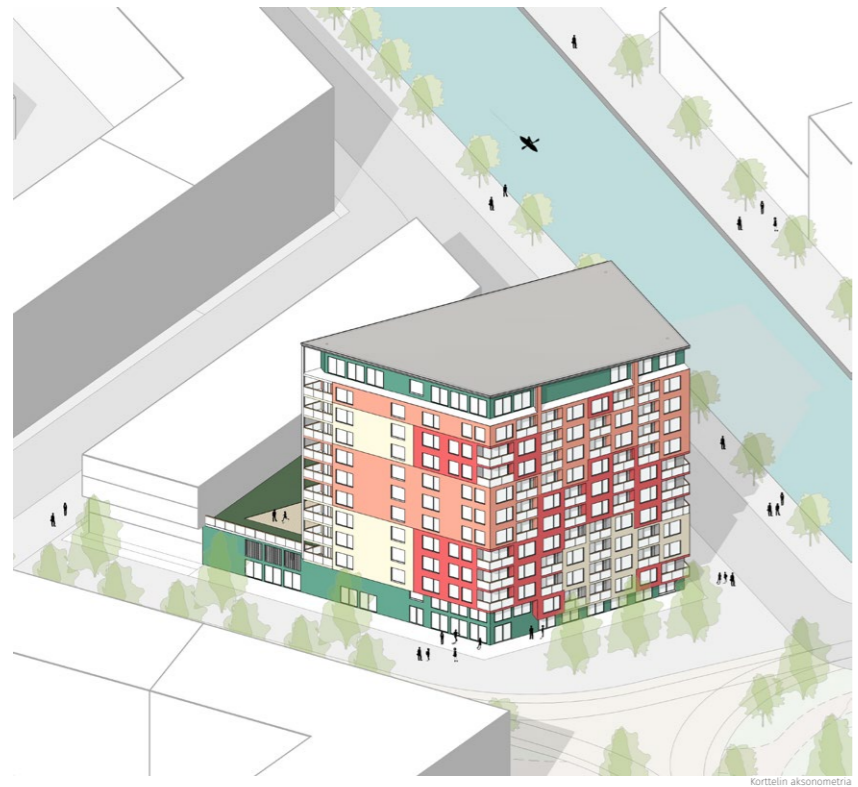
1-2. Kerros
 - Myymälä
 - Kahvila
 - Pysäköinti
 - 4 Irtaimistovarastoa
 - 3 asuntoa
 - Pyörävarastot & huolto

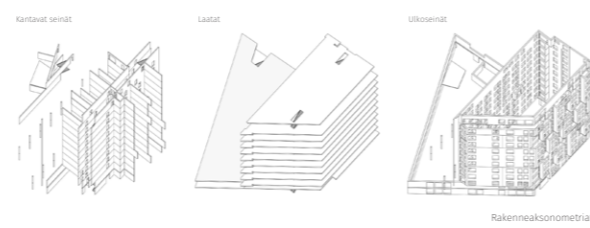
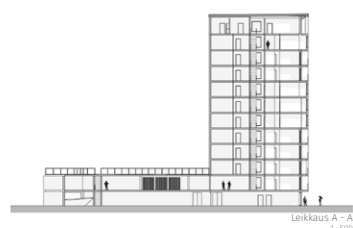
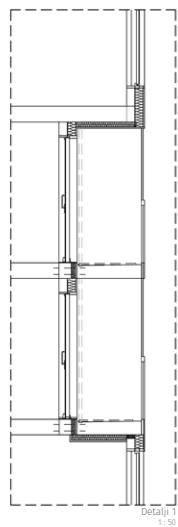
Huoneistot			
Level	Y1	K2	K3
Level 1			
Level 2		3	
Level 3	4	4	2
Level 4	5	4	2
Level 5	5	4	2
Level 6	5	4	2
Level 7	1	4	4
Level 8	5	4	2
Level 9	5	4	2
Level 10	5	4	2
Level 11	5	4	2
Level 12	40	39	20
Yht.	40	39	20
Jakauma %	40.4	39.4	20.2

Kerrosala	
Level	Area
Level 1	1206.5 m ²
Level 2	268.5 m ²
Level 3	566.5 m ²
Level 4	566.5 m ²
Level 5	566.5 m ²
Level 6	566.5 m ²
Level 7	559.0 m ²
Level 8	567.5 m ²
Level 9	566.5 m ²
Level 10	566.5 m ²
Level 11	567.5 m ²
Level 12	125.5 m ²
Yht.	6693.5 m²

(15% ylijys vt. kaavoituksen)

Bruttoala	
Level	Area
Level 1	1532.5 m ²
Level 2	1546.0 m ²
Level 3	606.0 m ²
Level 4	606.0 m ²
Level 5	604.5 m ²
Level 6	606.0 m ²
Level 7	607.5 m ²
Level 8	606.0 m ²
Level 9	606.0 m ²
Level 10	606.0 m ²
Level 11	606.0 m ²
Level 12	558.5 m ²
Yht.	9091.5 m²





6. PÄÄTELMÄT

Miten tästä kaikesta muodostuu rakennus? Opinnäytetyön aika oli rajallinen ja tutkimusosuus suunnitelmalle alisteinen, mikä johti siihen, että tässä opinnäytetyössä generoin vielä varsin yksinkertaisella algoritmilla tietyt lähtöparametrit kerrostasopohjan luonnokseksi jatkokehitykseen. Kysymykseksi muodostuikin tällöin, mihin algoritmi kykenee tilanteessa, jossa sitä käsittelee joku minua huomattavasti ohjelmointitaitoisempi henkilö ilman tiukkoja ulkoisia rajoitteita?

Puhutaan siis tilojen, rakenteiden, objektien ja tekniikan sijoittamisesta tiettyjen ehtojen perusteella, huoneiden generoimisesta ovineen ja ikkunoineen tai julkisivujen generoimisesta niille määriteltyjen ehtojen ja raja-arvojen perusteella. Kaikki nämä ovat vain murto-osa niistä mahdollisuuksista joita algoritmeilla (graafisella ohjelmoinnilla) ja generatiivisilla suunnittelumeto- deilla pystytään tekemään tulevaisuudessa. Järjestelmät nimenomaisesti tarvitsevat toisensa ja yhdessä ne luovat lyömättömän yhdistelmän perustel- tuja ratkaisuja yhdistettynä suureen määrään vaihtoehtoja. Näiden vaihtoehtojen mittaaminen muodostuu tällöin myös eräksi tärkeäksi ohjelmoitavaksi seikaksi algoritmissa.

Tässä työssä vaihtoehtojen mittaus edusti lähinnä luonnosvaiheessa esiintyviä, yksinkertaisia kysymyksiä kuten; 'Kuinka monta asuntoa saan mahdutettua kerrokseen?' 'Kuinka tehokkaasti tilat täyttävät kerroksen?'

Opinnäytetyön ansiosta olen kuitenkin ymmärtänyt ne huimat mahdollisuudet, kun mitattaviksi asioiksi voidaan jatkossa ohjelmoida esimerkiksi palomääräyksiä, esteettömyysmääräyksiä tai jopa tiettyjä asuntojen viihtyisyyteen vaikuttavia seikkoja (valon määrä asunnossa, tilojen suhteet toisiinsa tai kulkureitit asunnon sisällä). Onnistuessaan mittaamaan tällaisia seikkoja algoritmi nostaa itsensä luonnosteluvaiheen viitteellisestä suunnittelusta jo varsinaisiin, rutiininomaisten suunnittelutehtävien pariin.

Yhdessä kehittyvän tekniikan ja ohjelmistojen kanssa aihealueen tulevaisuus näyttääkin lupaavalta ja jättää minut odottamaan innolla uusia läpimurtoja.

7. LÄHTEET

- 001 *Tieteen termipankki*, The Helsinki Term Bank for the Arts and Sciences
<https://tieteentermipankki.fi/wiki/Termipankki:Etusivu/en>, 12/2018
- 002 *OxfordDictionaries.com*.
linkki: <https://en.oxforddictionaries.com/>, 1/2019
- 003 *Kielitoimiston sanakirja*, Kotimaisten kielten tutkimuskeskus
- 004 Hotokka, Pekka. *Metaheuristiikat logistiikan optimoinnissa*.
Jyväskylän yliopisto, Tietotekniikan laitos. 2008
- 005 Hietanen, Jiri. *Tietomallit Ja Rakennusten Suunnittelu: Filosofinen Selvitys Tieto- Ja viestintätekniiikan mahdollisuuksista*. Rakennustieto, 2005.
- 006 Danil Nagy, Damon Lau , John Locke, James Stoddart, Lorenzo Villaggi, Ray Wang, Dale Zhao, David Benjamin. *Project Discover: An application of generative design for architectural space planning*, 2017
- 007 Menges, Achim, and Sean Ahlquist. *Computational Design Thinking*.
John Wiley & Sons, 2011.
- 008 Mitchell, William J. *The Logic of Architecture*. MIT Press, 1994.
- 009 Galanter, Philip. "Generative Art Theory." *A Companion to Digital Art*, 2016, pp. 146–180.
- 010 El-Khaidi, Maher. *Mapping Boundaries of Generative Systems for Design Synthesis*. Master of Science in architecture studies,
Massachusetts Institute of Technology, 2007.
- 011 Gürsel, Dino. *Creative design exploration by parametric generative systems in architecture*. METU JFA, 29(1), pp. 207–224, 2012
- 012 Terzidis, Kostas. *Algorithmic Design: A Paradigm Shift in Architecture?*
Graduate School of Design, Harvard University, 2004
- 013 Kolarevic, Branko. *Architecture in the Digital Age: Design and Manufacturing*. Taylor & Francis, 2003.
- 014 Walmsley, Kean. *Autodesk and Airbus generatively design a 3D-printed "bionic" partition for the A320*. Through the Interface. 2.2.2016.
- 015 Bitonti, Francis, et al. *3D-Printed Body Architecture*. Wiley, 2017.
- 016 Aalto, Alvar. *Taimen ja tunturipuro*. *Domus* 223-225 1947. pp. 7-10
- 017 Aasholm, Ron. *Incessant replication: Computational floor plan generation*.
Aalto-yliopisto, arkkitehtuurin laitos, 2015.
- 018 Novak, Marcos. *Computational Composition in Architecture*.
Graduate School of Architecture and Urban Planning,
University of California, 1988.
- 019 Luce Kristina. *Getty Research Journal*, no.2, 2010.
- 020 Gerfen, Katie. *Autodesk MaRS Office*, *Architect Magazine*, 12.1.2018
- 021 Abdullah, A. A. *Zaha Hadid Form Making Strategies for Design*.
Universiti Teknologi Malaysia, 2013.
- 022 Schumacher, Patrik. *Parametricism A New Global Style for Architecture and Urban Design*. John Wiley & Sons, 2016.
- 023 *Autodesk Revit*, www.autodesk.com.
- 024 *Dynamo Primer, What is Dynamo?*
http://dynamoprimer.com/en/01_Introduction/1-2_what_is_dynamo.html. 1/2019
- 025 Smith, Lilli. *DynamoBIM, Introducing Project Refinery*.
<http://dynamobim.org/introducing-project-refinery/>, 12.11.2018
- 026 *Bałyk, MVRDV'S First project in Poland opens*.
<https://www.mvrdv.nl/news/baltyk-in-poznan-opens>, 2.6.2017
- 027 *Bałyk, MVRD - Projects*, <https://www.mvrdv.nl/en/projects/baltyk>
- 028 MVRDV, *The Sax*. MVRDV Press Room, 2019.
- 029 Baltacioglu, Erhan. *Distributor's Three-Dimensional Pallet-Packing Problem: Human Intelligence-Based Heuristic Approach*.
Air Force Institute of Technology, 2001.

Opinnäytetyö
Metropolia Ammattikorkeakoulu

Janne Hänninen

1. PAINOS, 2019