

LAPPEENRANNAN-LAHDEN TEKNILLINEN YLIOPISTO LUT
School of Energy Systems
Sähkötekniikka
2019

Henri Leskinen

RAKENNUSSÄHKÖSUUNNITTELUN AUTOMATISOINTI

Työn tarkastajat: Professori Jarmo Partanen
Tutkijaopettaja Jukka Lassila

TIIVISTELMÄ

Lappeenrannan-Lahden teknillinen yliopisto (LUT)
LUT School of Energy Systems
Sähkötekniikka

Henri Leskinen

RAKENNUSSÄHKÖSUUNNITTELUN AUTOMATISOINTI

Diplomityö

2019

47 s., 19 kuvaa ja 1 taulukko

Tarkastajat: Professori Jarmo Partanen
 Tutkijaopettaja Jukka Lassila

Työn ohjaajat: DI Marko Stenman
 Ins. (YAMK) Keni Peltonen

Hakusanat: sähkösuunnittelu, rakennussähkö, suunnittelun automatisointi, algoritminen suunnittelu

Rakennussähkösuunnittelua ei ole tällä hetkellä juurikaan automatisoitu, vaikka nykyisillä ohjelmilla se on mahdollista. Työn lähtökohtana oli tutkia rakennussähkösuunnittelun automatisoinnin mahdollisuuksia, eli mitä ja miten pitkälle automatisoidaan. Sähkösuunnittelusta pohdittiin osuuksia, jotka olisivat järkeviä automatisoida. Tällaisiksi osuuksiksi valikoituivat valaisimet, paloilmaisimet, johtotiet ja rasiat. Käytettäväksi ohjelmaksi työn vaatimusten mukaan valittiin Autodeskin Revit-ohjelma ja lisäohjelma Dynamo, jolla voi tuottaa graafista- ja tekstimuotoista ohjelmointia.

Alussa valittuja suunnitteluosuuksia toteutettiin valituilla ohjelmilla luomalla komentosarjoja, jotka toteuttavat valittujen objektien lisäyksen suunnitelmiin automaattisesti. Valaisimille luotiin kolme komentosarjaa eri jaoille, yhden, kahden ja neljän valaisimen lisäykseen. Valaisin lisäys onnistui, mutta jokaiselle valaisinmäärälle täytyi luoda oma komentosarja. Kojerasiat lisättiin onnistuneesti uppoasennusten taakse ja kojerasioilta ylöspäin putkitukset kaapeleiden syöttöön. Paloilmaisimille ei onnistuttu tekemään älykästä komentosarjaa.

Komentosarjojen luonti vaatii osaamista rakennussähkösuunnittelusta ja käytettävistä ohjelmistoista. Komentosarjojen käyttö vaatii perehdytyksen käyttöliittymään ja komentosarjan mahdollisiin ongelmiin. Haasteena automaattisen suunnittelun toteutuksessa ovat erilaiset projektit ja arkkitehtimallit. Komentosarjan luomiseen kuluva aika on vaikea arvioida etukäteen. Revit-ohjelmaa käytettäessä kannattaa hyödyntää Dynamon mahdollisuudet toimintojen automatisoinnissa. Rakennussähkösuunnittelun automatisointi kannattaa rajata yksinkertaisiin ja toistuviin suunnitteluosuuksiin.

ABSTRACT

Lappeenranta-Lahti University of Technology (LUT)
LUT School of Energy Systems
Electrical Engineering

Henri Leskinen

AUTOMATION OF BUILDING ELECTRICAL DESIGN

Master's Thesis

2019

47 pages, 19 figures and 1 table

Examiner: Prof. Jarmo Partanen
Associate professor Jukka Lassila

Supervisors: M.Sc. Marko Stenman
M.Eng. Keni Peltonen

Keywords: electrical design, building electrical systems, automation of design, algorithmic design

Building electrical planning hasn't been automated much yet, even though with current programs it is possible. Baseline of this master's thesis was to research the possibilities of automation of building electrical design, what and how far it should be automated. Parts of electrical planning we're considered that would be reasonable to automate. As such parts we're chosen luminaires, fire detectors, cable trays, conduits, mounting boxes and sockets. The program to use was chosen to be Autodesk Revit with add-on Dynamo according to the requirements set. With Dynamo you can produce graphical and textual based programming.

The parts of electrical planning which we're chosen at the beginning we're executed with the chosen programs by creating scripts that add chosen objects to the plans automatically. For the luminaires three scripts we're made for the distributions of one, two and four luminaires. The adding of the luminaires was a success but each distribution of luminaires required its own script. Mounting boxes we're added behind flush mounted objects and conduits we're added upwards from the mounting boxes for the cable installations. A working script for the fire detectors couldn't not be created during this thesis.

The creating of scripts requires knowledge from building electrical planning and the software used. The running of the scripts requires an introduction to the use of the interface and the possible problems when running the script. Challenge in carrying out automated planning comes in the form of different types of projects and architectural models. The time to create a script is difficult to estimate beforehand. When using Revit one should utilize the possibilities of Dynamo in the automation of functions. The automation of building electrical planning should be limited to simple and repetitive parts of planning.

ALKUSANAT

Diplomityön aiheesta ja sen tekoon annetusta ajasta haluan kiittää Ramboll Finland Oy:n sähkösektoria. Aihe oli erittäin mielenkiintoinen ja liittyi konkreettisesti omaan työhöni yrityksessä. Yrityksen puolelta ohjauksesta kiitän Espoossa Marko Stenmania, Keni Peltosta ja Lappeenrannan toimistolta Perttu Valtosta, joka neuvoi Revit-ohjelmiston saloihin. Kiitän myös muita työkavereita neuvoissa sähkösuunnitteluun liittyvissä asioissa. Yliopiston puolelle kiitokset menevät ohjaavalle professorille Jarmo Partaselle.

Tuesta opintojeni suorittamiseen ja diplomityön tekoon haluan kiittää perhettäni, joka on tukenut niitä myös taloudellisesti. Kiitokset myös tyttöystävälleni, joka laittoi opintoihin vauhtia niiden laahatessa. Kiitokset myös vanhoille ja opintojeni aikana saaduille uusille kavereille, jotka järjestivät tekemistä opintojen ulkopuolelle.

SISÄLLYSLUETTELO

Käytetyt merkinnät ja lyhenteet

1.	Johdanto.....	7
1.1	Tutkimuskysymykset.....	7
1.2	Ramboll Finland Oy	8
2.	Rakennussähkösuunnittelu	9
2.1	Nykytilanne	9
2.2	Sähkösuunnitelmista löytyvät automatisoitavat osiot	10
2.2.1	Paloilmaisimet	10
2.2.2	Valaistus	11
2.2.3	Johtotiet.....	11
2.2.4	Rasiat	11
3.	Algoritminen suunnittelu	13
3.1	Suunnitteluohjelman valinta	13
4.	Autodesk Revit.....	14
4.1	Building Information Modeling- BIM.....	14
4.2	MagiCAD for Revit	14
4.3	Family	15
4.4	Näkymät.....	16
4.5	Schedules ja Quantities.....	19
5.	Dynamo	20
5.1	Käyttäjien luomat paketit.....	23
5.2	DesignScript.....	24
5.3	Python Dynamossa	25
5.4	Listat	26
5.4.1	Shortest List.....	27
5.4.2	Longest List.....	28
5.4.3	Cross Product	29
6.	Dynamo toteutus	30
6.1	Arkkitehtimalli	30
6.1.1	Tila- ja huonesuodatus.....	31
6.1.2	Yhteensovitus	32
6.2	Valmiit Dynamo-koodit.....	32
6.3	Objektien sijoitus Dynamolla Revit-projektiin	33
6.3.1	Valaisimet	34
6.3.2	Kojerasiat ja putkitus	36
6.3.3	Paloilmaisimet.....	37
7.	Johtopäätökset.....	39
7.1	Komentosarjojen luonti.....	39
7.2	Komentosarjojen käyttö.....	41
7.3	Haasteet.....	41
7.4	Jatkokehitys.....	42
7.5	Etenemismalli.....	44
8.	Yhteenveto	45
	Lähteet.....	47

Liitteet

KÄYTETYT MERKINNÄT JA LYHENTEET

CAD	Computer aided design
BIM	Building information modeling
LVI	lämpö, vesi, ilmanvaihto
MEP	mechanical, electrical, plumbing

1. JOHDANTO

Rakennussähkösuunnittelu Suomessa tuotetaan pääasiassa tietokoneavusteisilla suunnitteluohjelmilla eli CAD-ohjelmistoilla. Suunnittelua on automatisoitu vähän, eikä älykästä algoritmista suunnittelua vielä ole käytössä. Nykyisillä ohjelmistoilla suunnittelun älykäs automatisointi on mahdollista. Alalle on myös vahvasti tulossa kohteiden BIM-mallintaminen.

Tämän työn tavoitteena oli pohtia, kuinka pitkälle rakennussähkösuunnittelun automatisointi on mahdollista viedä ja kuinka pitkälle se kannattaa viedä. Tavoitteena oli myös löytää sähkösuunnittelun osioita, joita voitaisiin ja kannattaisi toteuttaa algoritmisilla suunnittelutyökaluilla automaattisesti. Työ rajattiin koskemaan rakennussähkösuunnittelua Suomessa ja tarkemmin sähköpisteiden sijoitusta suunnitelmiin.

Työssä esitellään rakennussähkösuunnittelu Suomessa ja sen nykytilanne. Seuraavaksi pohditaan mahdollisia matemaattisia malleja ja vakioita, joita sähkösuunnitelmista löytyy automaattisen suunnittelun käytettäviksi. Käytettävä ohjelma valittiin tutkimuskysymyksen pohjalta. Kun ohjelma on valittu, esitellään ohjelmisto ja sen apuohjelmat ominaisuuksineen ja niiden peruskäyttö. Käytettävän ohjelmiston valinnan jälkeen vastauksia tutkimuskysymyksiin lähdettiin hakemaan testaamalla valitulla ohjelmistolla.

Alussa esitettyjä mahdollisia automatisoitavia suunnitteluosuuksia toteutettiin tai yritettiin toteuttaa valitulla ohjelmistolla eri arkkitehtimalleissa. Toteutukseen käytettiin itse tuotettuja ja valmiita komentosarjoja. Apua komentosarjojen luontiin haettiin valmiista komentosarjoista, ohjelman omista oppaista ja ohjelman käyttäjäfoorumeilta. Komentosarjojen luonnin ja käytön pohjalta saatiin tuloksia suunnittelun automatisaation haasteista, mitä kannattaa ja mitä suunnitteluosuuksia ei kannata automatisoida. Lopuksi pohditaan kuka automaattista suunnittelua toteuttaa ja kuinka aiheen kanssa kannattaa edetä.

1.1 Tutkimuskysymykset

Päätutkimuskysymys:

- Miten pitkälle rakennussähkösuunnittelun automatisointi kannattaa viedä?

Apututkimuskysymykset:

- Minkälaisia matemaattisia malleja ja vakioita sähkösuunnitelmista löytyy ja miten ne voitaisiin kirjata parametreiksi suunnitteluohjelmaan?

- Millä ohjelmalla tai ohjelmilla automatisoitua suunnittelua toteutetaan?
- Mitä lähtötietoja arkkitehtipohjasta saadaan ja mitä haasteita erilaiset pohjat tuovat?
- Kuka automaattista suunnittelua toteuttaa?

1.2 Ramboll Finland Oy

Tämä diplomityö tehdään Ramboll Finland Oy:n sähkösektorille. Ramboll Finland Oy kuuluu Ramboll Group A/S-konserniin. Koko yhtiö työllistää 15 500 työntekijää ja Suomessa 2400 asiantuntijaa. Suomessa toiminta on vahvistunut useiden yrityskauppojen myötä. Ramboll Finland Oy on Suomen suurin suunnittelu- ja konsultointialan yritys. Ramboll Finland Oy toimii kaupunkien, infrastruktuurin, liikenteen, ympäristön ja rakennusten suunnittelussa, rakennuttamisessa, rakentamisessa ja ylläpidossa. Suomessa liikevaihto on 227 miljoonaa euroa. Asiakkaina ovat niin julkinen kuin yksityinen sektori. (Ramboll 2019)

2. RAKENNUSSÄHKÖSUUNNITTELU

Sähkösuunnittelulla rakennusalalla tarkoitetaan sähkösuunnitelmien tuottamista, joiden pohjalta rakennuksen sähköistys voidaan toteuttaa. Sähkösuunnittelu pitää nykyisin sisällään esimerkiksi tasopiirustukset, joissa esitetään sähköpisteet ja johdotus, sähkökeskus- ja piirikaaviosuunnitelmat, turvavalaistus-, tele- ja palosuunnitelmat ja kaiken muun rakennussähköön liittyvän ja tilaajan tilaaman suunnittelun. Sähkösuunnittelu eroaa myös maiden välillä. Joissain maissa sähkösuunnitelmia ei johdoteta, kun taas Suomessa ne lähes poikkeuksetta johdotetaan ja käytetään eri symboleita sähköobjekteille. Tässä työssä keskitytään vahvavirtatasopiirustusten tuottoon Suomessa, joka pitää sisällään pistorasiat, valaistuksen ja muut vahvavirtalaitteet, niiden pistekuvat, johdotuksen ja ryhmityksen. Sähkösuunnitelmat tuotetaan tietokoneohjelmilla kuten CADS- tai AutoCAD-sovelluksilla ja nykyisin on yleistynyt myös projektien tietomallinnus, jossa koko rakennus mallinnetaan kolmiulotteiseksi malliksi. Mallista näkee eri suunnittelualojen asennukset, joka helpottaa tekniikan yhteensovittamista keskenään ja rakennuksen kanssa. Sähkösuunnittelua ohjaavat lait ja standardit kuten SFS 6000.

Sähkösuunnittelijan tehtävä on tuottaa tilaajan toiveiden ja vaatimusten mukaiset suunnitelmat. Suunnitelmien mukaan voidaan toteuttaa asennukset, hankinnat ja toteutukset kohteessa. (Harsia 2004) Suunnittelijan täytyy ottaa huomioon myös muut hankkeessa vaikuttavat suunnittelunalat. Sähkösuunnitteluun vaikuttavat myös arkkitehdin, rakennustekniset ja talotekniikan suunnitelmat. Yhteistyötä eri suunnittelualojen välillä tarvitaan, että sähkösuunnitteluun vaikuttavat rakennus- ja arkkitehtisuunnitelmien muutokset otetaan huomioon ja etteivät esimerkiksi kaapelihyllyt ja talotekniikan putkistot törmää toisiinsa.

2.1 Nykytilanne

Ramboll Finland Oy:n sähkösuunnittelu toteutetaan nykyisin pääosin Autodeskin tuottamalla AutoCAD-ohjelmistolla ja sen Progran Oy:n MagiCAD Electrical lisäosalla. Suunnitelmissa käytetään MagiCAD Electricalia tasopiirustusten ja sen lisäosia Switchboard Schematicsia sähkökeskuksen pääkaavioiden ja Circuit Designeria piirikaavioiden tuottoon. AutoCAD Revit-ohjelmistolla on aloitettu pilottiprojekti.

AutoCAD-ohjelmistolla tuotetaan suunnitelmia kaksiulotteiseen arkkitehtipohjaan, joille voidaan antaa myös korkeus ja tarkastella niitä kolmiulotteisessa näkymässä tai viedä suunnitelmat erilliseen tietomalliohjelmaan. AutoCAD kolmiulotteinen näkymä ei ole kovin kätevä yhteensovitukseen, vaan yhteensovitus tehdään yleensä erikseen tietomallinnusohjelmalla, kuten Autodesk Navisworks- tai Solibri-ohjelmalla. AutoCAD-ohjelmistolla ei voi toteuttaa sähkösuunnitteluun liittyvää automatisoitua suunnittelua, mutta MagiCAD Electrical ohjelmassa on automatisoituja toimintoja kuten ryhmänumeroiden yhdistäminen sähkötasosta sähkökeskuksen pääkaavioon.

2.2 Sähkösuunnitelmista löytyvät automatisoitavat osiot

Algoritmista suunnittelua ei ole järkevää suorittaa, jos algoritmin luomisaika on pidempi kuin luodulla komentosarjalla koskaan säästetty aika. Eli sähkösuunnitelmista tulisi löytää toistuvuuksia, joita hyödyntää ja tuottaa yhdellä komentosarjalla mahdollisimman paljon suunnitteluosuudesta. Tällaisia toistuvuuksia saamme esimerkiksi standardeissa vaadituista asennuksista ja projektikohtaisista vaatimuksista. Standardeista saatuja vaatimuksia ja arvoja voidaan käyttää lähtötietoina ja reunaehtoina automaattiselle suunnittelulle. Seuraavissa kappaleissa on esitetty esimerkkejä yleisistä lähes joka tilan sähköasennuksista, ja niihin löytyviä sääntöjä ja ohjeita, joita voi käyttää lähtötietoina.

2.2.1 Paloilmaisimet

SFS-EN 54-1-standardissa ja ST-ohjeisto 1 on määritelty paloilmaisin, suunnittelu, asennus ja ylläpito. ST-ohjeisto 1 mukaan kaikki palo-osastoon kuuluvat tilat tulee varustaa paloilmaisimilla tietyin ohjeistossa mainituin poikkeuksin. Paloilmaisin vaaditaan useimpiin tiloihin ja voidaan sijoittaa isoon osaan jäljelle jäävistä. Tästä syystä paloilmaisimien automaattinen sijoittaminen on järkevää.

Ilmaisinten sijoittamiseen saadaan käytettäviä lähtötietoja ja reunaehtoja ohjeistosta. Yhdistelmäilmaisimella suurin valvottu alue saa olla enintään 60 m², savuilmaisimella 60 m² ja lämpöilmaisimella 30 m². Ilmaisimien tulee asentaa enintään 2 metrin etäisyydelle koneellisen ilmanvaihdon poistoilma-aukosta, ellei valvottavan tilan pinta-ala ole pienempi kuin ilmaisimen toiminta-ala. Tätä sääntöä ei saa käyttää tiloissa, joissa on virheellisen ilmoituksen mahdollisuus kuten kosteissa, tulipesällisissä tai ruoanlaittomahdollisissa huoneissa. Tarkeimmat määritelmät löytyvät ohjeistosta. Paloilmaisin asennetaan kattoon tai enintään 10% alemmas kuin huonekorkeus. Katto katsotaan alaslasketuksi, eli alakatoksi, jos se on 20%

matalampana kuin huoneen katto. Ilmaisimen paikan määrää se kumpaa on enemmän, alaslaskettua vai normaalia kattoa. Jos alakatto katto on rei'itetty ja 50% tai enemmän on auki sijoitetaan ilmaisimet kattoon. Jos alle 50% on kiinni, tulee ilmaisimet sijoittaa alakattoon ja kattoon, jos alakaton ja katon välinen tila sisältää palokuormaa. Ilmaisimen ympärille on jätävä 0,5 m vapaata ilmatilaa pysty- ja vaakasuunnassa. Ilmaisimen täytyy olla vähintään 0,2 m etäisyydellä kattoon kiinnitetyistä valaisimista ja muista laitteista. (ST-ohjeisto 1 2019)

2.2.2 Valaistus

Valaistus ja sen ohjaus suunnitellaan käytännössä rakennuksen jokaiseen tilaan. Valaisimien vaatimukset löytyvät standardista SFS-EN 60598-1 ja työkohteiden valaistus standardista SFS-EN 12464-1. Jälkimmäisestä standardista löytyvät taulukoidut luksimäärät erilaisille huonetyypeille, joita voi käyttää ohjearvoina huoneen valaistumäärään, ellei suunnittelun tilaaja vaadi muuta. Valaisinohjaukseen voidaan käyttää esimerkiksi perinteisiä valokatkaisijoita, liiketunnistimia tai painonappeja. Katkaisijat ja painonapit sijoitetaan yleensä oven viereen ja liiketunnistimet kattoon niin että ne havaitsevat mahdollisimman suuren alan tilasta.

2.2.3 Johtotiet

Sähköasennukset tarvitsevat johtoteitä, jotta kaapelit saadaan vietyä sähköasennuksille sähkökeskukselta. Johtoteitä ovat muun muassa kaapelihyllyt, valaistusripustuskiskot ja johtokourut. Valaistusripustuskiskoja asennetaan valaistusjaon mukaan ja niihin voidaan asentaa valaisinten lisäksi esimerkiksi pistorasioita, yleiskaapeloinnin rasioita, edellisen kappaleen ohjeiden mukaan paloilmalaisimia ja liiketunnistimia. Samalla ne toimivat johtotienä niillä oleville tai takana oleville laitteille. Johtokouruja asennetaan esimerkiksi toimisto- ja luokkatiloihin ja niihin voidaan uppoasentaa esimerkiksi pisto- ja datarasioita.

2.2.4 Rasiat

Pistorasioita asennetaan yleensä jokaiseen tilaan. Pistorasioiden asennuspaikkoja ovat esimerkiksi siivouspistorasia oven viereen ja pöytien ja toimistojen työpisteiden läheisyyteen. Siivouspistorasioita voidaan asentaa myös tasaisella jaolla esimerkiksi käytäväosuuksille. Jos pistorasioita, valokatkaisijoita tai yleiskaapeloinnin rasioita asennetaan uppoasennuksena vaativat ne myös kojerasian. Samoin yleiskaapeloinnin rasioita asennetaan työpisteiden läheisyyteen.

Jakorasioita käytetään, kun sähkönsyöttöä täytyy jakaa eri laitteille tai kytkeä väliin esimerkiksi kytkimiä. Jakorasioita tarvitaan myös kun tilassa on useita valaisimia ja pistorasioita jos niitä ei ketjuteta, ja esimerkiksi valaisinohjaukseen kytkimille.

3. ALGORITMINEN SUUNNITTELU

Algoritmisella suunnittelulla tässä työssä tarkoitetaan lähtöparametreista algoritmilla tuotettua automaattista suunnittelua. Algoritmisen suunnittelun tässä työssä rajataan vahvavirta sähkötasokuvien pistesijoitteluun, eli vahvavirtatasokuviin tulevien sähköasennuksien sijoittamiseen. Ohjelman tarkoituksena on annettujen lähtötietojen pohjalta lisätä tarvittavat objektit oikeille paikoille automaattisesti. Algoritmisen suunnittelun tarkoitus on nopeuttaa suunnittelua suorittamalla osan pistesijoittelusta automaattisesti, jolloin sähkösuunnittelija voi keskittyä enemmän kohteen haastavempiin osiin kuin mekaaniseen sijoitteluun. Lähtötiedoiksi ohjelmalle annetaan esimerkiksi haluttu sijoitettava objekti, arkkitehtipohja ja haluttu määrä pisteitä. Lisäksi ohjelmalle annetaan reunaehdot suunnittelun näkökannalta kuten standardeista ja ohjeista saatavia arvoja. Ohjelma suorittaa annettujen lähtötietojen ja koodinsa perusteella pisteiden lisäyksen ohjelmalle asetettujen reunaehtojen mukaan.

3.1 Suunnitteluohjelman valinta

Suunnitteluohjelman valintaan vaikuttavat ohjelman alkuperäinen käyttötarkoitus, mahdollisuudet rakennussähkösuunnitteluun, sisältääkö se valmiiksi Suomalaiset sähköpiirrosmerkit, ohjelman muut ominaisuudet ja algoritmisen suunnittelun mahdollisuus ja tehokkuus. Ohjelman alkuperäisessä käyttötarkoituksessa otetaan valintaa tehdessä huomioon lähinnä ohjelmat, jotka ovat tarkoitettu rakennussähkön- tai muuhun rakennussuunnitteluun. Jos ohjelma ei sisällä Suomalaisia sähköpiirrosmerkkejä symbolikirjastossaan olisi ne lisättävä ohjelmaan. Työn tavoitteen ollessa algoritmisen suunnittelun, ohjelmia jotka eivät sisällä suoraa mahdollisuutta siihen ei oteta huomioon.

Vertailuun otettiin seuraavaksi ohjelmistoja, jotka soveltuvat rakennusten talotekniikan MEP-suunnitteluun ja mahdollisuuden algoritmiseen suunnitteluun. Näillä alkuehdoilla vertailtaviksi ohjelmiksi valittiin Bentley OpenBuildings ja Generative Components lisäohjelma, Autodesk Revit ja Dynamo lisäohjelma ja Mcneel Rhinoceros ja Grasshopper lisäohjelma. Tarkasteluhetkellä halutut toiminnot sähkösuunnittelun osalta onnistuvat ainoastaan Autodesk Revit ohjelmistolla ja Dynamo lisäohjelmalla.

4. AUTODESK REVIT

Autodeskin kehittämä Revit ohjelmisto on tarkoitettu rakennusten ja talotekniikan suunnitteluun ja dokumentointiin BIM-ympäristössä, joka tukee suunnittelua, piirroksia ja aikatauluja. Revit malliin luodaan 2d- ja 3d-näkymiä, jotka ovat kaikki samasta virtuaalisesta rakennuksen mallista. Kaikki tehdyt muutokset näkyvät kaikissa näkymissä, joihin se vaikuttaa. Samassa projektissa voi työskennellä useampi suunnittelija samaan aikaan, toisin kuin AutoCAD-ohjelmistossa, jossa vain yhdellä käyttäjällä kerrallaan voi olla sama tiedosto auki. Koko rakennus on siis yhden tiedoston takana, eikä esimerkiksi jokainen kerros ole oma tiedostonsa kuten AutoCAD-ohjelmistossa. Revit myös luo kolmiulotteisen mallin itsestään eikä sitä tarvitse viedä ulos ohjelmasta erikseen ja katsoa tietomallinnusohjelmalla. Seuraavissa kappaleissa on esitetty Revitin tärkeimpiä ominaisuuksia, toimintoja ja lisäohjelmia.

Autodesk Revitissa automaattinen suunnittelu onnistuu Dynamo työkalulla. Dynamo on graafinen ohjelmoinnin käyttöliittymä, jolla ohjelmointiin ei välttämättä tarvitse osata ohjelmoida. Dynamon graafisilla algoritmeilla voidaan esimerkiksi automatisoida suunnittelua ja taulukoiden luontia. (Dynamo Primer 2019, 1.2)

4.1 Building Information Modeling- BIM

BIM on älykäs mallipohjainen prosessi, joka yhdistää eri toimialojen suunnittelijat, jotta rakennuksen suunnittelu, rakentaminen ja käyttö onnistuu tehokkaammin. Aluksi rakennuksesta luodaan kolmiulotteinen malli, joka on kaikkien suunnittelunalojen käytössä koko rakennuksen elinkaaren eri vaiheissa, suunnittelussa, rakentamisessa, käytössä ja huollossa. Tähän malliin liitetään kaikkien suunnittelunalojen suunnitelmat. BIM-mallinnus auttaa eri suunnittelunalojen yhteensovittamisessa projektin suunnitteluvaiheen aikana, koska kaikkien osapuolten muutokset päivittyvät näkymään muille. Revit-projektiin voidaan liittää arkkitehtimallin lisäksi muita rakennuksia, rakenteita ja MEP-malleja, eli muiden suunnittelunalojen suunnitelmia. Liitettyjä malleja voi muuttaa ryhmiksi, Group, ja ryhmiä voidaan muuttaa liitetyiksi malleiksi.

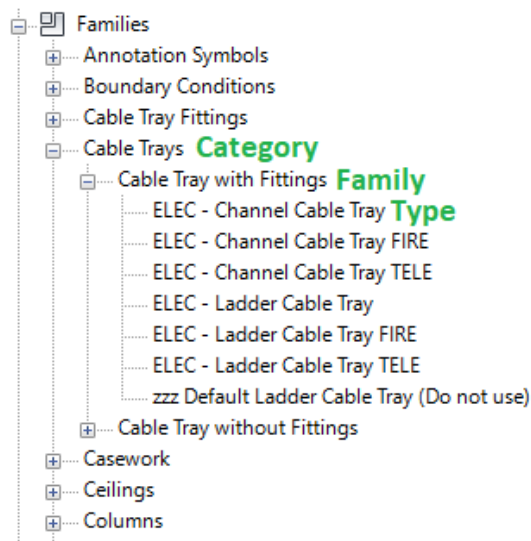
4.2 MagiCAD for Revit

MagiCAD Electrical on MagiCAD Groupin kehittämä lisäohjelmisto Autodesk AutoCAD- ja Revit-ohjelmistoihin. Ohjelmisto sisältää Suomalaisen symbolikirjaston ja laajan BIM-

objektitietokannan, josta voi hakea laitevalmistajien tuotteita. (MagiCAD 2019) Ohjelmistolla onnistuu myös esimerkiksi kaapelihyllyjen ja johtoteiden mallinnus. MagiCAD ohjelmistojä käytetään Ramboll Finland Oy:n sähkösuunnittelussa AutoCAD ja Revit lisäohjelmistoina.

4.3 Family

Kaikki Revitin elementit kuuluvat johonkin perheeseen, Family, ja niillä on oma hierarkiansa. Hierarkian huipulla on kategoriat, Categories, jotka ovat ennalta-asetettuja ja niitä ei käyttäjä pysty lisäämään, poistamaan tai nimeämään uudestaan. Revit sisältää paljon eri kategorioita, mutta tärkeimpänä ovat model- ja annotation-ryhmät. Model-ryhmään kuuluvat kaikki mallin fyysiset elementit, kuten seinät, ovet ja sähköasennukset. Annotation-ryhmään kuuluvat tekstit, merkinnät ja mitat. Kategorioiden alla on perhe. Perhe sisältää objekteja, jotka näyttävät ja käyttäytyvät samalla tavalla. Kuvassa 4.1 on kuvakaappaus perhevalikosta, josta avattuna kaapelihyllyt, jotka kuuluvat kaikki Cable Trays-kategoriaan ja edelleen Cable Tray with Fittings-perheeseen ja itse valittava kaapelihylly on tyyppi, Type.



Kuva 4.1 Revit hierarkia

Perheet jakautuvat kahteen eri ryhmään, järjestelmäperheisiin, System Family, ja komponenttiperheisiin, Component (Loadable) Family. Järjestelmäperheitä ei voi muokata tai poistaa mutta ne voivat sisältää useita tyyppejä, joita voi rajoitetusti muokata. Tyypit ovat hierarkiassa perheiden alapuolella ja ne sisältävät perheen mukaisten ominaisuuksiensa lisäksi muita muuttujia kuten koon, materiaalin tai muita asetuksia. Tyypillä on helppo muokata

useita perheen eri muuttujia kerralla. Tyyppejä voidaan muokata ja lisätä myös järjestelmäperheiden alle. Järjestelmäperheet sisältävät fyysisiä objekteja, näkymiä, projektidataa ja tasoja. Järjestelmäperheet, jotka ovat fyysisiä objekteja ovat isäntiä, Host. Isäntä-objekteja tarvitaan monille objekteille, jotka kuuluvat komponenttiperheisiin, esimerkiksi kattoon asennettavat objektit voivat tarvita katon isännäkseen.

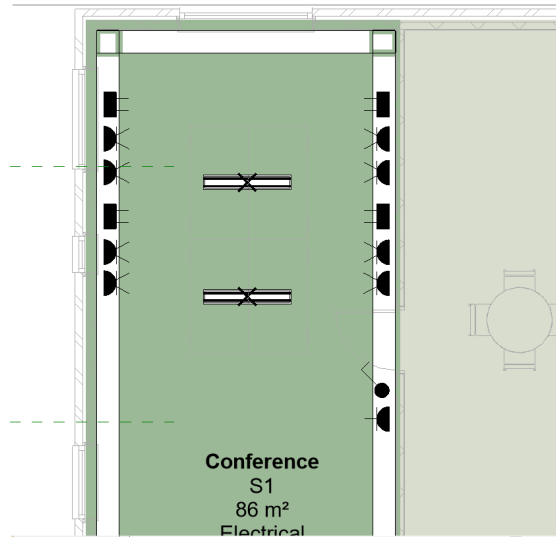
Komponenttiperheet sisältävät kaiken muun, joka ei sisälly järjestelmäperheisiin. Komponenttiperheitä käyttäjä voi lisätä, muokata ja poistaa. Ne voivat vaatia isännän tai olla vapaasti sijoitettavia, jolloin ne eivät vaadi isäntää. Komponenttiperheet jakautuvat myös tyyppihin, joita voi lisätä, muokata ja poistaa. Kolmas perhetyyppi on paikalliset perheet, In-place Family. Nämä ovat projektia varten luotuja yksittäisiä objekteja, joita ei voi viedä kyseisen projektin ulkopuolelle ja niitä voidaan luoda myös järjestelmäperheiden objekteista. Tässä työssä keskitytään komponenttiperheiden tyyppien lisäämiseen ja järjestelmäperheiden komponenttien kuten tilojen, ovien ja kattojen käyttöön lähtötietoina. (Autodesk 2018)

4.4 Näkymät

Revit perustuu yhteen malliin ja sen näkymiin. Revitissa voidaan luoda erilaisia näkymiä rakennusmallista, kuten kaksiulotteisia suunnitelmia, leikkauksia, korkokuvia ja kolmiulotteisia näkymiä. Revit luo näiden tyyppisiä oletusnäkymiä, mutta sähkösuunnittelulle käytetään omia näkymiä, joissa näkyvät myös kaksiulotteiset piirrossymbolit tasokuvissa. Rakennusmallissa kaikki näkymät ja taulukot ovat esitettyinä samasta rakennuksen tiedostosta. Kun muutoksia tehdään yhdessä näkymässä ne vaikuttavat koko projektin muihin näkymiin ja taulukoihin. Muutokset näkyvät muissa näkymissä, jos muutoksen kohde on asetettu niissä näkyväksi. Asetuksilla voi valita mitkä perheet näkyvät suunnitelmassa ja mitkä ovat sammutettuina. Nämä näkymäasetukset voivat olla eri työskentelynäkymille Working Views ja tulostusnäkymille Printing Views. Työskentelynäkymässä voidaan laittaa näkyviin suunnittelua auttavia objekteja kuten kommentteja tai muiden suunnittelualojen suunnitelmia häiritsemättä tulostusnäkyä. (Autodesk, 2017)

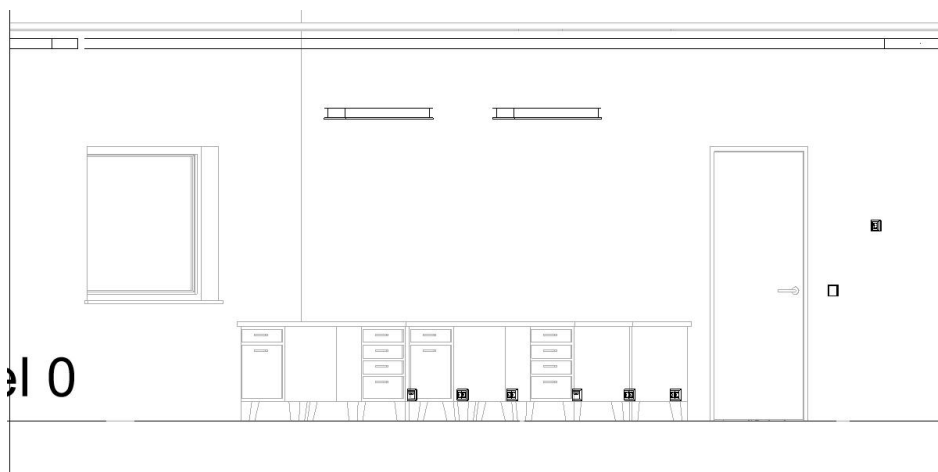
Plan Views sisältää perinteiset kaksiulotteiset piirustukset eli pohjakuvat tai tasokuvat, katosuunnitelmat ja rakennesuunnitelmat. Suurin osa sähkösuunnitelmista esitetään tasokuvissa kaksiulotteisina objekteina, joka näkyy kuvassa 4.2. Näissä pohjakuvissa esitetään kaksiulotteiset symbolit sähkösuunnitelmista ja huone- ja tilamerkinnot. Revit luo tasokuvia automaattisesti, kun rakennukseen lisätään uusia kerroksia. View Range-asetuksella voidaan

määrittää miltä korkeusalueelta kaksiulotteinen tasokuva näyttää objektit arkkitehtimallista. Cut plane-leikkauksella voidaan edelleen määrittää mitä objekteja 2d-pohjakuvassa näkyy valitulta korkeusalueelta.



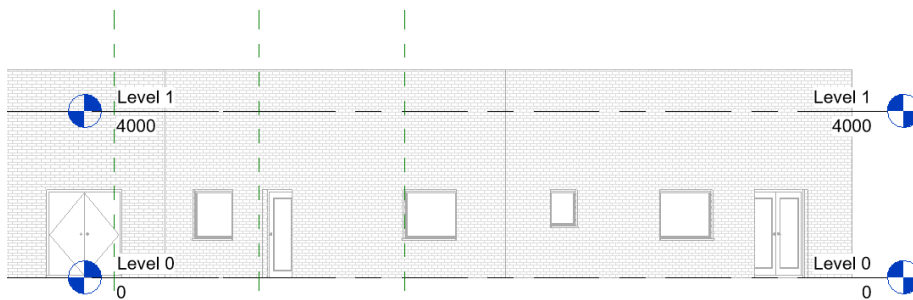
Kuva 4.2 Tasokuva

Section Views sisältää leikkauskuvia rakennuksesta, joita käytetään sähkösuunnittelussa detaljikuvina kohdista, joita on hankala esittää pelkissä pohjakuvissa. Leikkauskuvia voidaan Revitissa luoda kaikista näkymistä. Kuvassa 4.3 on luotu leikkauskuva samasta huoneesta kuin kuvassa 4.2 katsoen vasemmalta alhaalta oikealle ylös, josta näkee saman kaapeliyhdyntä, valaisimet, valokatkaisijan ja pistorasiat. Leikkauskuvat voivat olla kaksi- tai kolmiulotteisia ja niissä voidaan esittää suunnitelmat kaksi- tai kolmiulotteisin symbolein.



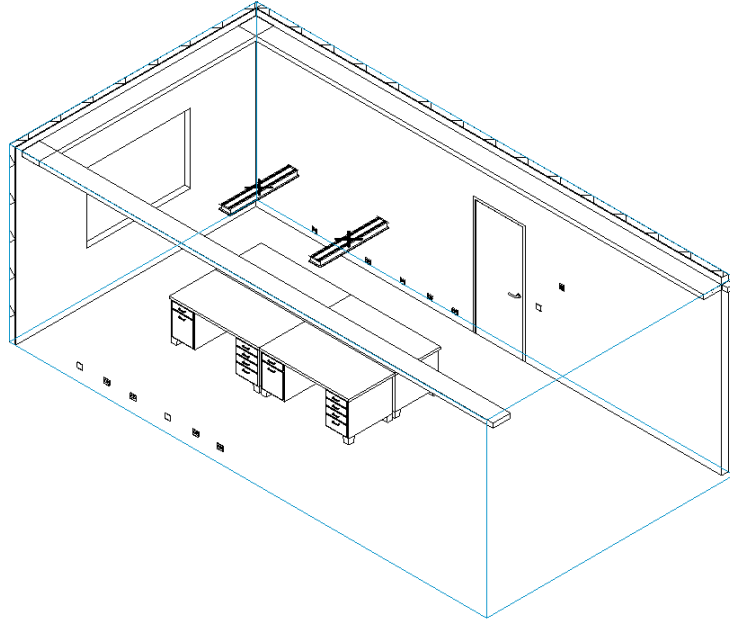
Kuva 4.3 Leikkauskuva

Elevation Views näyttää rakennuksen erilaisia kaksiulotteisia korkeusnäkyymiä. Näissä säädetään myös kerroksien korot kohdalleen vastaamaan arkkitehtimallin korkoja, kuten kuvassa 4.4. Elevation Views sivunäkymässä säädetään kohdalleen sivuttaiset moduuli-viivat. Kerroksien korot, Levels, ovat tärkeitä säätää oikealle kohdalleen, koska Revitissa lisätyt objektit ovat sidottuina kerrokseen, johon ne ovat alun perin lisätty ja ottavat korkonsa kyseisen kerroksen pohjasta alkaen. Korkeusnäkyymiä voidaan tehdä myös rakennuksen sisältä esimerkiksi yhdestä huoneesta, jossa voidaan antaa vielä erikseen sisäisiä korkoja huoneelle. Korkeusnäkymän ja leikkauskuvan erona on, että korkeusnäkymä on kaksiulotteinen ja leikkauskuva voi olla myös kolmiulotteinen.



Kuva 4.4 Korkonäkymä

Kolmiulotteisia näkymiä käytetään rakennuksen tarkastelemiseen kolmiulotteisesti. Kolmiulotteista näkymää Revit sähkösuunnittelussa käytetään sijoitettujen objektien sijaintien tarkastamiseen, valaistuksen jaon tarkastamiseen ja tarkempaan törmäystarkasteluun rakenteiden ja muiden suunnittelunalojen kanssa. Kolmiulotteisessa näkymässä objektit näkyvät oikean kokoisina eivätkä suurennettuina piirrosmerkkeinä kuten kaksiulotteisissa pohjakuvissa. Kolmiulotteisessa näkymässä tehdyt muutokset päivittyvät kaikkiin muihin näkymiin ja muissa näkymissä tehdyt muutokset kolmiulotteisiin näkymiin. Kuvassa 4.5 on sama huone kuin edellisissä kuvissa kolmiulotteisena näkymänä, mutta lähimmät seinät piilotettuina näkymän parantamiseksi.



Kuva 4.5 Kolmiulotteinen näkymä

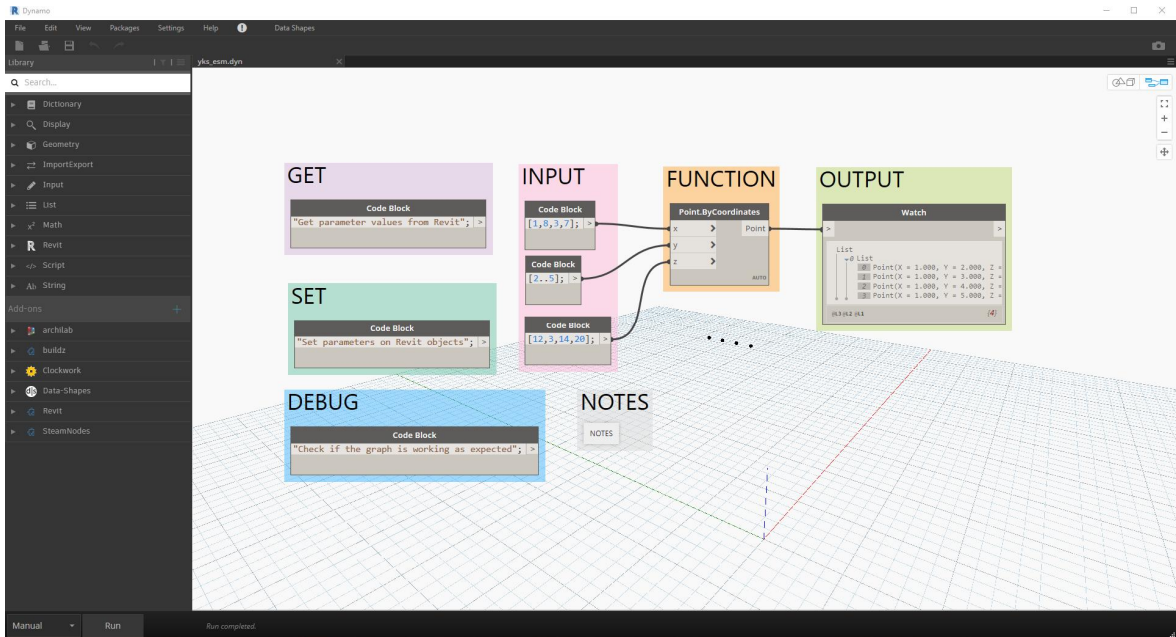
4.5 Schedules ja Quantities

Revitin ominaisuuksiin kuuluu taulukoiden luonti ja määrien laskenta. Luotuihin taulukoihin tiedot päivittyvät automaattisesti muutoksia tehdessä projektiin. Esimerkiksi valaisinmäärää ylläpitävä taulukko päivittyy aina valaisimia lisätessä tai poistettaessa. Taulukoita voidaan viedä Revitista muihin ohjelmiin, kuten taulukko-ohjelmiin. (Autodesk 2019)

5. DYNAMO

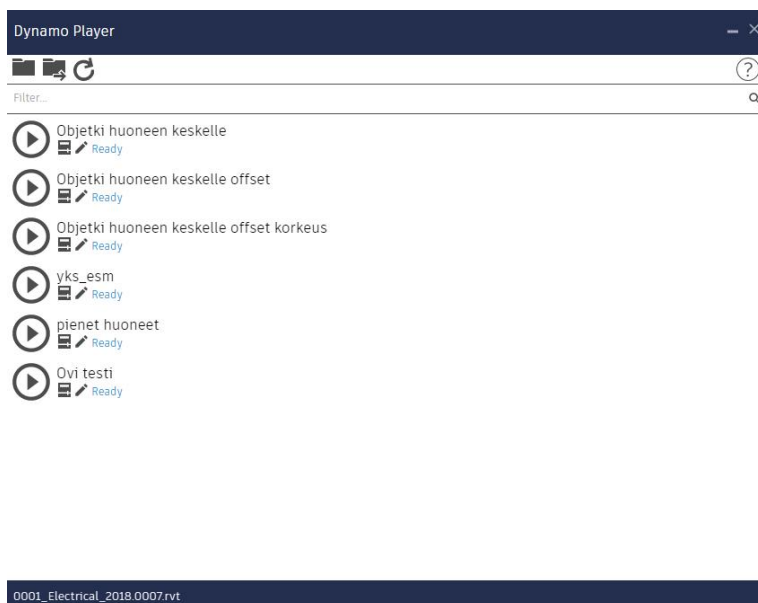
Dynamo on avoimeen lähdekoodiin perustuva graafinen ohjelmoinnin käyttöliittymä suunnitteluun. Tässä työssä käytettiin Dynamo 2.0.3 versiota ja Revit 2019 versiota. Dynamo perustuu komentolohkoihin, joilla on sisään- ja ulostulo, Input ja Output. Komentolohkoja näkyy kuvassa 5.1. Ohjelman taustalla on DesignScript-ohjelmointikieli, jolla komentolohkot toimivat. Lohko suorittaa tietyn komennon sille sisääntulossa annettujen muuttujien pohjalta. Dynamo sisältää itsessään laajan valikoiman komentolohkoja ja lisäksi käyttäjien luomia komentopaketteja, joita voi hakea Dynamon sisältä internet-kirjastosta Packages-valikon alta. Dynamoon voidaan luoda myös omia komentolohkoja erityylyisiä lohkoja yhdistelemällä ja niitä voi julkaista itse internet-kirjastoon. Itse tehdyt tai muiden tekemät lohkot haettuna internet-kirjastosta voivat koostua Dynamon omista lohkoista, Python-koodeista ja DesignScript-koodeista.

Kuvassa 5.1 on kuvakaappaus Dynamon ikkunan näkymästä. Dynamon saa käynnistettyä Revitissa Manage-välilehdeltä. Dynamon voi käynnistää myös itsenäisesti Sandbox Modessa, mutta käyttääkseen komentosarjoja Revitissa on se käynnistettävä Revitin kautta. Kuvassa 5.1 on Dynamolla tehty yksinkertainen komentosarja, joka sijoittaa koordinaatistoon neljä pistettä. Kuvassa on myös esimerkki komentoblokkien ryhmittelystä ja ryhmien värikoodaamisesta Autodeskin ohjeen mukaan, Group, selkeyden ja myöhemmän muokkaamisen helpottamiseksi. Komentoblokit toimivat niitä yhdistelemällä solmuilla, node, aina ulostulosta sisääntuloon. Get-osiossa otetaan tietoja Revit-mallista. Input-osiossa annetaan lähtötiedot toiminnan suorittavalle Function- tai Get-osiolle. Output osiossa voidaan tarkastaa mitä Function-osio teki ja Set-osiossa säätää Function-osiossa tehtyjä toimintoja. Debug-osiossa tarkastetaan, että komentosarja toimi oikein.



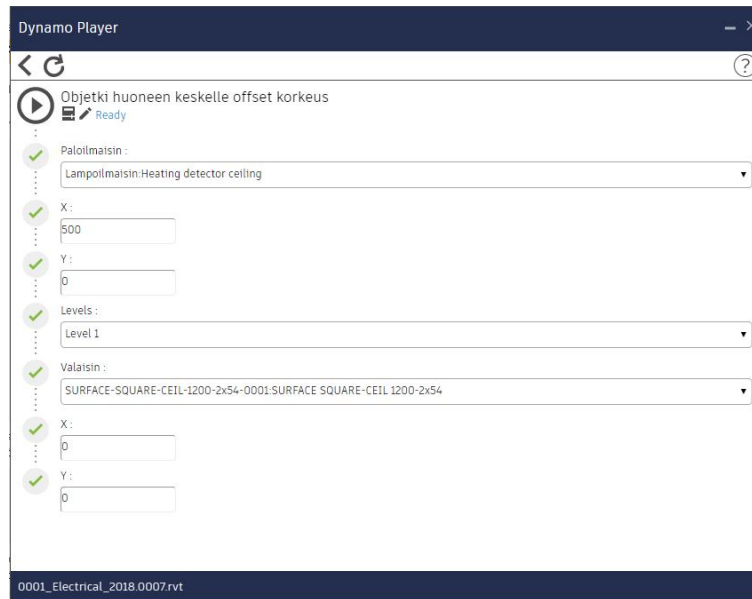
Kuva 5.1 Dynamon käyttöliittymä, yksinkertainen komentosarja ja lohkojen jaottelu ryhmiin

Vasemmalla ikkunassa näkyy kirjasto, Library, jossa on ylempänä Dynamon omat komennot ja alempana käyttäjien luomien pakettien komentovalikot Add-ons palkin alla. Näkymä-asetuksista, Views, voidaan valita näkykö komentosarjan taustalla käynnissä olevan projektin kolmiulotteinen malli vai pelkkä ruututausta. Myös näkymän värejä voi muokata, jotta komentosarja ja solmut eivät huku taustaan. Vasemmalta alhaalta voi valita toimiiko Dynamo-koodi automattisesti, eli ajaa itsensä aina komentosarjan tai lähtöparametrien muututtua vai erillisen Run-painikkeen kautta Dynamo-ikkunan ollessa auki. Dynamon komentosarja voidaan ajaa projektiin myös kuvassa 5.2 näkyvän Revitin Dynamo Player-ikkunan kautta, joka löytyy myös Manage-välilehdeltä. Käyttääkseen komentosarjoja tätä kautta tulee Revitin kautta avatun Dynamon olla suljettuna, mutta Sandbox-ikkuna voi olla auki.



Kuva 5.2 Revit Dynamo Player

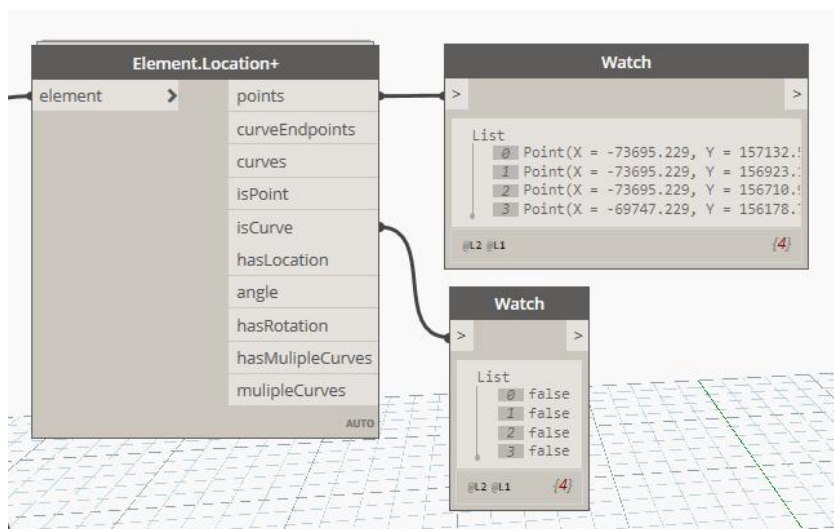
Kansion kuvasta voidaan valita kansio, josta Dynamo Playerissa näkyvät komentosarjat ladataan. Toisesta painikkeesta voidaan avata auki oleva kansio resurssienhallinnan kautta ja kolmannesta päivittää kansio. Ikkuna sisältää haun komentosarjoille. Alareunassa näkyy projektin nimi, johon Dynamo Player on ajamassa komentosarjaa. Play-painikkeesta voidaan ajaa komentosarja projektiin sellaisenaan. Jos komentosarja vaatii käyttäjän valitsemia lähtötietoja, niin Play-painike vie kuvan 5.3 mukaiseen ikkunaan, jossa pyydetyt lähtötiedot voidaan syöttää manuaalisesti. Lähtötiedot valitaan Dynamo komentosarjaan määrittämällä lohko sisääntuloksi klikkaamalla lohkoa oikealla hiirellä ja valitsemalla ”Is input”. Komentosarjalla voi olla myös ulostuloja, jotka näkyvät Dynamo Playerissa komentosarjan käytön jälkeen. Ne valitaan kuten lähtötiedot, mutta valinnalla ”Is output”. Tässä esimerkissä ohjelma tarvitsee paloilmalaitteen ja valaisimen tyyppit, jotka huoneisiin lisätään. Lisäksi X- ja Y-koordinaateilla voidaan määrittää objektin sijoitus huoneen keskipisteeseen nähden ja Levels-valikolla kerros, johon asennukset tulevat. Dynamo Player on yksinkertaisempi käyttöliittymä Dynamon komentosarjoille ja lähtötietojen syöttäminen on selkeämpää kuin suoraan Dynamon ikkunasta. Dynamon komentosarjoille voi luoda myös oman käyttöliittymän ja lisätä sen Revitin Add-Ins-välilehdelle.



Kuva 5.3 Dynamo Player lähtötietojen syöttö

5.1 Käyttäjien luomat paketit

Käyttäjien luomissa paketeissa on monia hyödyllisiä komentoja, jotka täytyisi muuten luoda useilla Dynamon vakiolohkoilla, DesignScript- tai Python-koodilla tai kaikkien yhdistelmillä. Pakettien komennot voivat helpottaa komentosarjan luomista. Kuvassa 5.4 on esimerkki käyttäjien luoman Clockwork-paketin komennosta Element.Location+ jolle on annettu sisääntulona neljä pistettä ja komentolohko antaa ulostulonaan useita tietoja elementin sijainnista ja ominaisuuksista. Samojen ulostulojen saaminen Dynamon vakio komentolohkoilla olisi huomattavasti työläämpää tai mahdotonta.



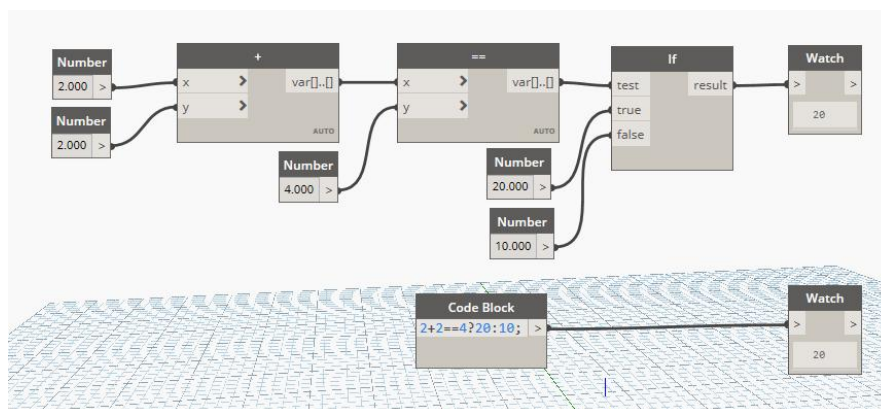
Kuva 5.4 Käyttäjien luoman paketin komentoblokki

Käyttäjien luomien pakettien komentolohkojen käytössä on riskinä Revitin tai Dynamon päivitykset, jolloin käyttäjien luomien komentolohkojen toiminta voi muuttua tai hajota kokonaan. Käyttäjien luomat paketit täytyy tarvittaessa päivittää manuaalisesti, jolloin jokaisella komentosarjojen käyttäjällä tulisi olla sama versio samasta paketista luotujen komentosarjojen toiminnan varmistamiseksi. Dynamon internet-kirjastosta paketteja ladattaessa saa paketin uusimman version, joten jos käyttäjät ovat ladanneet paketit eri aikaan voi heillä olla paketista eri versiot. Osoitteesta www.dynamopackages.com voi hakea haluamansa version paketista. Tästä syystä käyttäjien luomia paketteja tulisi käyttää harkiten ja niitä käytettäessä kaikkien suunnittelijoiden käyttää samoja ja saman version paketteja, koska jos komentosarjan käyttäjältä puuttuu siinä käytetty paketti komentosarja ei toimi.

5.2 DesignScript

DesignScript toimii Dynamon graafisen käyttöliittymän takana. Kaikki Dynamon komentolohkot toimivat DesignScript-ohjelmointikielillä. Ohjelmointikieltä voi käyttää tekstimuotoisena Code Block lohkon kautta. Lohkon voi luoda helpoiten tuplaklikkaamalla kohdasta, johon haluaa lohkon Dynamossa luoda. Lohkoilla voi tehdä samat toiminnot kuin valmiilla Dynamon komentolohkoilla, mutta myös yhdistellä olemassa olevia komentoja tapahtumaan saman lohkon sisällä. Komentolohkoja voi muuttaa DesignScriptin tekstimuotoon Node to Code komennolla valitsemalla niitä yhden tai useamman ja klikkaamalla taustaa oikealla hiirellä. Tämä on kätevä tapa nähdä mitä komentolohkoissa tapahtuu ja näin DesignScript-ohjelmointikieltä on helppo oppia käyttämään. (Dynamo Primer 2019, 7.2)

DesignScript-kielillä on nopeampaa ja selkeämpää luoda esimerkiksi numeroita, merkkijonoja, lukujonoja, tehdä laskutoimituksia, ottaa alkioita listasta, luoda listoja ja tehdä ehtolauseita, kuin Dynamon komentolohkoilla. Tästä esimerkki kuvassa 5.5, jossa kahdeksan lohkoa korvattiin yhdellä Code Block lohkokolla. Ohjelma testaa onko yhteenlaskun tulos neljä ja antaa arvoksi 20 jos tosi ja 10 jos tulos on epätosi. DesignScript-kielillä voidaan myös luoda funktioita, joiden toiminta määritellään Code Block-lohkossa ja tämän jälkeen niitä voidaan käyttää muualla komentosarjassa. Toinen tapa selkeyttää komentosarjoja ja nopeuttaa komentosarjojen luontia vähemmän komentolohkoja käyttämällä on käyttää Python-koodia.

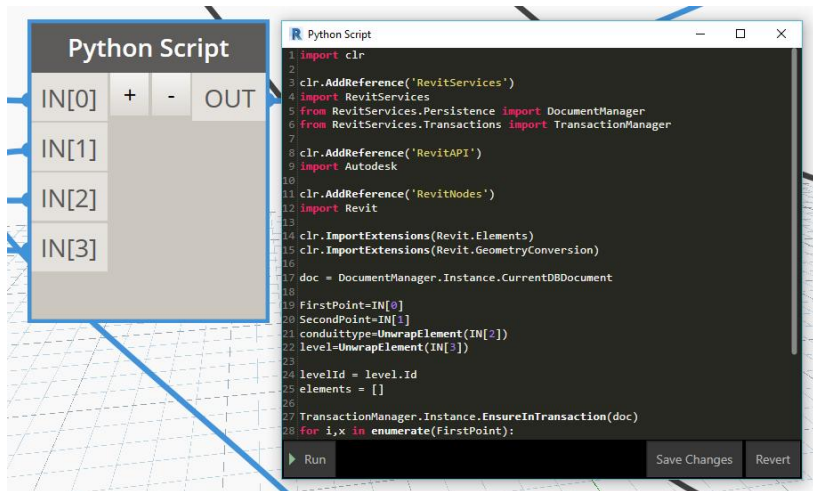


Kuva 5.5 DesignScript vertailu komentolohkoihin

5.3 Python Dynamossa

Python on laajalti käytetty ohjelmointikieli ja se voidaan sulauttaa muihin ohjelmiin. Python on tekstimuotoinen ohjelmointikieli, kun taas Dynamo on visuaalinen ohjelmointikieli. Visuaalisella ohjelmoinnilla ei aina saavuteta haluttavaa lopputulosta ja pitkät komentosarjat voivat mennä sotkuisiksi, kun sama lopputulos saavutetaan Python-koodilla muutamalla koodirivillä. Pitkiä ja sotkuisia visuaalisen ohjelmoinnin komentosarjoja on hankala lukea, muokata ja etsiä virheitä. Python-koodilla on myös helpompaa tehdä ehtolauseita ja silmuikoita, kuin pelkällä Dynamolla, jotka käyvät läpi useita eri arvoja. Python-koodin käyttö Dynamossa vaatii luonnollisesti Python-kielen osaamista. (Dynamo Primer 2019, 10.4)

Pythonia käytetään Dynamossa Python Script lohkolla, joka näkyy kuvassa 5.6. Python-koodi voi käyttää myös Python Script From String lohkolla, jolloin lohkolle annetaan Python-koodi tekstimuodossa sisääntulona. Lohkoissa olevilla lisäys- ja vähennyspainikkeilla voi lisätä tai poistaa sisääntuloja. Itse Python koodia voi muokata tuplaklikkaamalla lohkoa. Tämä avaa kuvassa 5.6 näkyvän Python Script ikkunan.



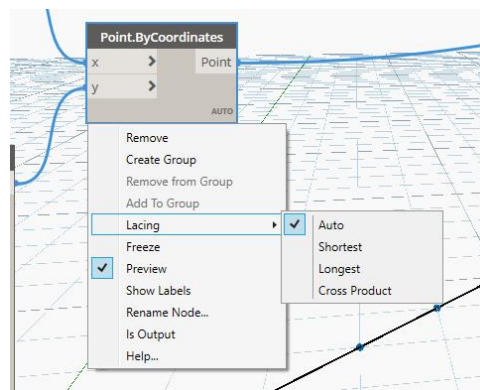
Kuva 5.6 Python Script

Python-koodin alussa voidaan tuoda tarvittavat ominaisuudet Revitista ja määrittää dokumentti johon koodi vaikuttaa. Tämän jälkeen on hyvä määritellä sisääntulot, tehdä toiminnot ja lopuksi Python Script-lohkosta saadaan vain yksi ulostulo. Jos ulostuloja halutaan enemmän kuin yksi tai yksi lista, on halutuista ulostuloista luotava listat ja määriteltävä ulostulo listaksi. Ulostulosta saatuja listoja voidaan edelleen käsitellä Dynamon omilla komennoilla tai toisilla Python-koodeilla.

5.4 Listat

Listat ovat tärkeä osa Dynamon toimintaa. Lista on kokoelma arvoja, tekstiä, tai esimerkiksi Revitista haettuja tietoja kuten lista kaikista huoneista. Listoilla voidaan käsitellä suurta määrää dataa helposti. Melkein kaikki tieto, joka sisältää enemmän kuin yhden arvon, on Dynamossa lista. Dynamossa listan alkiot ovat numeroituja ja numerointi alkaa nolasta. Listoja näkyy vasemmassa laidassa kuvissa 5.5, 5.6 ja 5.7. Dynamossa lista voi sisältää listoja, eli kyseessä on listan lista, jollaisia näkyy edellä mainituissa kuvissa keskellä. Tämä tarkoittaa, että listan alkiot ovat myös listoja. Dynamo kohtelee listoja objekteina listoissa ja listoja objekteina. Tämä tarkoittaa, että kutsuttaessa esimerkiksi ensimmäistä alkioita listasta joka sisältää listoja, Dynamo antaa ulostulona ensimmäisen listan, eikä jokaisen listan ensimmäistä arvoa, kuten joissain muissa ohjelmointikielissä. (Dynamo Primer 2019, 6.3) Listojen listoja voi yksinkertaistaa poistamalla listan lista rakenteen Flatten-komennolla jos monimutkaisen rakenteen käyttö ei ole tarkoituksenmukaista.

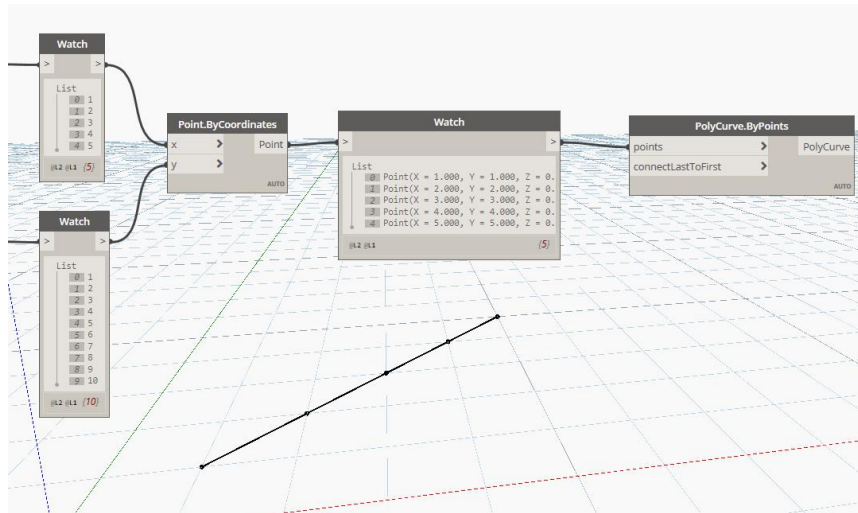
Dynamo käsittelee listoja lohkojen sisääntuloina eri tavoin. Tähän voidaan vaikuttaa Lacing-asetusta muuttamalla. (Dynamo Primer 2019, 6.3) Lacing-asetusta voi muuttaa kaikissa lohkoissa, joissa lukee oikealla alareunassa ”Auto”, kuten kuvassa 5.7. Asetusta muutetaan klikkaamalla oikealla hiirellä blokista Lacing-valikon alta. Lacing-asetus on tärkeä, kun komentosarjassa käytetään listoja ja erityisen tärkeä käytettäessä eripituisia listoja, kuten seuraavissa kappaleissa on esitetty. Eri Lacing-asetuksilla saadaan samoilla listoilla ja komentoblokeilla erilaisia lopputuloksia tai komentosarja ei saata väärillä asetuksilla toimia halutusti tai ollenkaan.



Kuva 5.7 Lacing-asetuksen muuttaminen

5.4.1 Shortest List

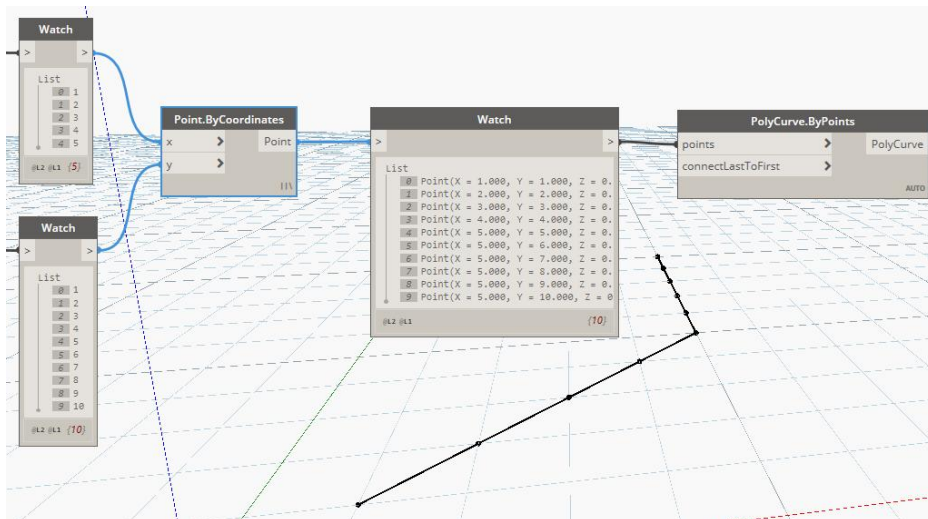
Lacing toimii automaattisella asetuksella yleensä Shortest List-asetuksella. Tällöin eri pituisten listojen yhdistäminen, tässä esimerkissä pisteiksi, lopetetaan kun jostain listasta loppuvat arvot. Esimerkissä kuvassa 5.8 on yksinkertainen komentosarja, joka piirtää pisteitä ja näiden pisteiden välille viivan. Ylemmässä listassa on viisi arvoa ja alemmassa kymmenen, joten lyhin lista asetuksella Dynamo lopettaa pisteiden piirtämisen, kun viisi arvoa on käyty läpi ja piirtää viisi pistettä ja niiden välille viivan. Pisteet yhdistetään molempien listojen samaa alkion numeroa vastaavista arvoista.



Kuva 5.8 Shortest List esimerkki

5.4.2 Longest List

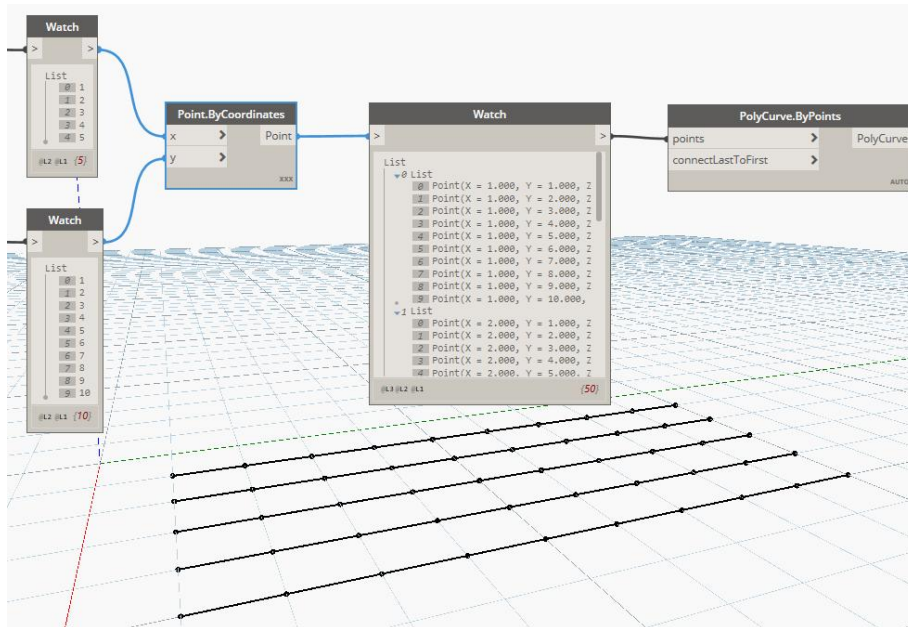
Pisimmän listan asetuksella Dynamo jatkaa lohkon sisääntulojen yhdistämistä kunnes molemmista listoista loppuvat arvot. Asetus käyttää samalla numeroilla olevia alkioita, kunnes lyhyemmästä listasta loppuvat arvot. Tämän jälkeen Dynamo käyttää lyhyemmän listan viimeistä arvoa pidemmän listan loppuille arvoille. Kuvan 5.9 esimerkissä huomataan pisimmän listan asetuksella lohkon käyttävän x-arvoa 5 loppuille viidelle y-arvolle Watch-lohkokossa ja koordinaatistoon piirryneistä pisteistä.



Kuva 5.9 Longest List esimerkki

5.4.3 Cross Product

Cross Product-asetuksella komentolohko käyttää kaikkien sisääntulojen kaikki arvot keskenään. Asetus on kätevä esimerkiksi tilanteissa, joissa kaikkia sisääntuloja verrataan keskenään, eli tässä tapauksessa luo molempien listojen x- ja y-arvoista kaikki mahdolliset pisteyhdistelmät ja piirtää näihin pistejonoihin viivat. Kuvassa 5.10 näkyy myös esimerkki, miten Dynamo käsittelee listojen listoja. Viivat eivät piirry jokaisen pisteen välille, vaan yhden listan listan sisällä olevien pisteiden.



Kuva 5.10 Cross Product esimerkki

6. DYNAMO TOTEUTUS

Dynamoä käytetään Revitissa suunnittelun nopeuttamiseen. Dynamolla voidaan suorittaa nopeasti yksinkertaisia mutta työläitä toimia kuten objektien sijoittelua, luetteloiden luontia tai parametrien vaihtoa. Näin suunnittelijalla kuluu vähemmän aikaa mekaaniseen toistoon ja jää enemmän aikaa projektin monimutkaisempien suunnitteluosuuksien toteuttamiseen. Suunnittelijan tulee tarkistaa Dynamo-koodilla automaattisesti toteutetut suunnitteluosuudet. Seuraavissa kappaleissa on esimerkkejä suunnitteluosuuksista, jotka testattiin Dynamolla.

6.1 Arkkitehtimalli

Arkkitehtimalli toimii tärkeänä lähtöparametrinä Dynamo-koodeille, josta voidaan hakea lähtötietoja. Arkkitehtimallin tulee olla Autodesk Revitilla tehty, jotta siinä voidaan käyttää Dynamo-ohjelmistoa. Tässä työssä vertailtiin kolmeä eri arkkitehtimallia, kahta esimerkkipmallia ja yhtä projektin mallia. Rakennuksen arkkitehtimallista saadaan lähtötietoina sähköasennusten sijoittamiseen vaikuttavat tilat, tilojen korkeus, ovet, ikkunat ja jokaiseen tilaan voidaan lisätä esimerkiksi haluttu valaistuksen luksimäärä. Tiloihin voidaan lisätä myös muita haluttavia parametrejä, kuten huoneen tyyppi, jotka voidaan ottaa lähtötiedoiksi Dynamoon.

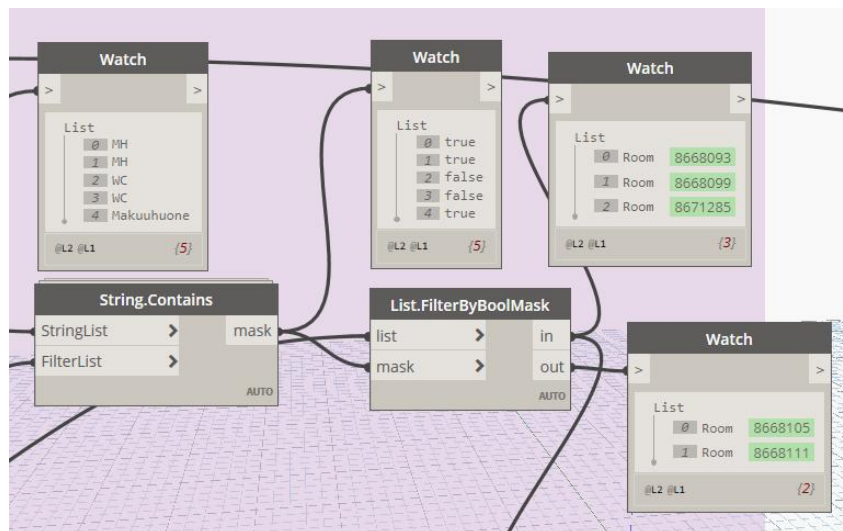
Tärkeä arkkitehtimallista saatava informaatio objektien sijoittelua varten on huoneet, Rooms, tai tilat „Spaces. Suunnittelun helpottamiseksi rakennus tulisi olla jaoteltuina ja nimettyinä huoneiksi tai tiloiksi, esimerkiksi toimistotiloiksi, varastoiksi tai WC:ksi. Jos rakennusta ei ole jaettu tulisi seiniin olla valittuna ”Room bounding”, jotta sähkösuunnittelija saisi jaoteltua ne itse helposti. Kun huoneet ovat nimetty voidaan Dynamolla suodattaa pois muut tilat tai huoneet, jos halutaan ohjelman vaikuttavan vain tiettyihin tiloihin. Objektien sijoitus onnistuu molempiin, mutta automatisoinnin kannalta paras tilanne on, jos rakennus on jaoteltu sekä tiloiksi että huoneiksi.

Arkkitehtimallista saadaan lähtötietoina myös rakenteita, jotka kuuluvat järjestelmäperheisiin kuten ovia, ikkunoita, seiniä ja huonekaluja kuten pöytiä ja peilejä. Projektikohtaisia haasteita voi tulla, jos näissä järjestelmäperheissä käytetään kustomoituja tyyppejä. Tällöin lisättäessä kustomoituja tyyppejä täytyy tyyppin nimi lisätä manuaalisesti Dynamo-koodiin, jotta Dynamo osaa käyttää sitä yhtenä lähtötietona. Toinen ongelmakohta on, jos tyyppien

kokoelmia on yhdistetty yhdeksi perheeksi, esimerkiksi koko keittiön kalusteet tai kokonainen huone niiden monistamisen helpottamiseksi ja jotta muutoksia tehdessä muutos päivittyy kaikkiin samanlaisiin perheisiin kerralla. Tällöin Dynamo ei saa eroteltua perheistä haettua objektia ja käytettyä sen sijaintia lähtötietona. Näin tehtyjä huoneita ei voi myöskään käyttää muiden huoneiden rajaukseen tai niitä jaotella tiloiksi ja huoneiksi. Jos vastaavasti objekteja on koostettu ryhmäksi, ei Dynamo osaa käyttää objekteja lähtötietoina ennen kuin ryhmä on hajotettu.

6.1.1 Tila- ja huonesuodatus

Tilojen tai huoneiden suodatuksen avulla Dynamon komentosarja saadaan vaikuttamaan vain haluttuihin tiloihin. Ilman suodatusta Dynamo vaikuttaa koko projektiin tai kerrosvalinnan ollessa käytössä kokonaiseen kerrokseen. Jos tiloja halutaan suodattaa, tulee jokaiseen tilaan tai huoneeseen lisätä tieto tilan käyttötarkoituksesta, kuten ”Makuuhuone” tai ”WC”. Dynamoön voidaan lisätä lista esimerkiksi erilaisista makuuhuoneiden merkinnöistä tai tuoda lista taulukkosovelluksesta, kuten ”Makuuhuone”, ”MH” ja ”Bedroom” vastaamaan erilaisia merkintätapoja. Tilatiedon mukaan huoneet tai tilat joihin komentosarjan ei haluta vaikuttavan, voidaan suodattaa pois `String.Contains`- ja `List.FilterByBoolMask`-komentoilla joiden yhdistelmä näkyy kuvassa 6.4.



Kuva 6.4 Makuuhuoneiden suodatus

Kuvassa `String.Contains`, joka on archilab-käyttäjäpaketista, lohkolle annetaan suodatettava lista ja lista tilojen nimistä, jotka halutaan jättää tai poistaa. `String.Contains` antaa ulostulo-

naan samanpituisen listan kuin sille annetaan, mutta totuusarvomuuttujina sen mukaan löytyikö listan alkioista ehtojen mukaisia merkkijonoja. `List.FilterByBoolMask` lohkolle annetaan sisääntulona alkuperäinen lista huoneista ja totuusarvomuuttujalista, joiden mukaan se jakaa huoneet in- ja out-ulostuloihin. In-ulostulossa ovat tosiarvoiset huoneet ja out-ulostulossa epätosiarvoiset huoneet. Näin ollen suodatusta voidaan käyttää jatkamalla in- tai out-ulostulon arvoja, riippuen siitä haluttiinko valita vai suodattaa pois tietyt huoneet tai tilat. Kommentosarja toimii samalla tavalla huoneille tai tiloille. Suodatusta täytyy käyttää myös muihin käsiteltäviin listoihin, jotta listojen järjestykset säilyvät oikeina tiloja suodattaessa.

6.1.2 Yhteensovitus

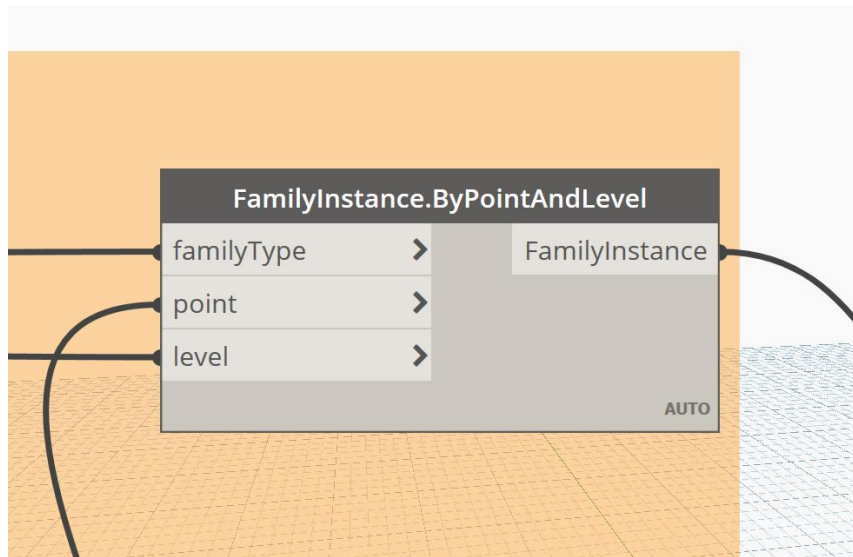
Objektien sijoitus saadaan älykkäämmäksi otettaessa huomioon myös muut suunnittelunalat, kuten rakenteet ja LVI-suunnittelun. Dynamolla ei suoraan objekteja lisätessä varsinainen yhteensovitustarkastelu onnistu, mutta ratkaisu voisi löytyä Python-koodilla. Ainut löydetty ratkaisu tähän oli käyttää Autodesk Nawisworks-ohjelmistoa, jolla tehdä törmäystarkastelu ja tuoda törmäyspisteet Dynamoon. Dynamolla merkataan jokainen törmäyspiste näkyväksi Revittiin esimerkiksi punaisella pallolla.

6.2 Valmiit Dynamo-koodit

Autodeskin kanssa yhteistyössä on ennen tätä diplomityötä Rambollilla tehty Dynamo-komentosarjoja todisteena toimintojen mahdollisuudesta. Kommentosarjat on tehty Dynamon versiolla 1.3.3 ja Revit versioilla 2017 ja 2018. Nämä komentosarjat ovat pääsääntöisesti tehty Python-koodeilla käyttämällä hyvin vähän Dynamon omia komentolohkoja. Kommentosarjat sisältävät komennot, joka luo kaapelihyllyt kiertämään kaikkien tilojen reunat, asettaa siivouspistorasian oven viereen, asettaa valaisimet valittujen pöytätyyppien päälle, asettaa valaisimet valittujen peilien päälle ja kirjaa kaikki tilat ja niissä olevat asennukset Excel-tiedostoon. Kommentosarjat ovat merkattu kokeellisiksi eikä niitä tulisi käyttää sellaisenaan projekteissa. Kommentosarjat ovat luotu toimimaan tietyissä esimerkkiprojekteissa, joten esimerkiksi pöytiin tai peileihin liittyvät komentosarjat käyttävät lähtötietonaan tiettyjä järjestelmäperheisiin kuuluvia projektikohtaisia kalustetyyppejä. Kommentosarjoja apuna käyttämällä, muokkaamalla ja yhdistelemällä tässä työssä luotuihin voidaan luoda toimivia ratkaisuja.

6.3 Objektien sijoitus Dynamolla Revit-projektiin

Yksi Dynamolla tehtävä suunnitteluosuus on objektien sijoitus projektiin. Sijoitettavia objekteja voivat olla kaikki sähkösuunnitteluun liittyvät kuten rasiat, kojeet, johtotiet, keskukset ja viitetekstit. Objekteja voidaan sijoittaa kuvassa 5.1 näkyvällä `FamilyInstance.ByPointAndLevel` lohkokolla, joka tarvitsee sisääntuloiksi sijoitettavan perheen tyyppin, `familyType`, pisteen, `point`, ja kerroksen, `Level`, ja sijoittaa ulostulona objektin sille annetuilla lähtötiedoilla. Lohko osaa sijoittaa useita objekteja kerralla sille annettaessa useita pisteitä listana.



Kuva 6.1 Objektien sijoitukseen tarvittava komento

Ulostulona lohkoista saadaan sijoitetut objektit, joita voidaan tarkastella Watch lohkokolla. Lohko näyttää sille osoitetun lohkon ulostulon. Lohko näkyy kuvassa 5.1, jossa se näyttää koordinaatit johon pisteet lisättiin. Watch lohkoa on hyvä käyttää komentosarjaa luodessa välitulosten tarkasteluun ja valmiissa koodissa lopputulosten tarkasteluun. Watch-lohkon voi asettaa komentosarjan ulostuloksi, jolloin se näkyy Dynamo Playerissa komentosarjan käytön jälkeen.

Perheen tyyppi, eli suunnitelmiin sijoitettava objekti, voidaan valita yksinkertaisesti komentolohkokolla Family Types. Family Types antaa alasvetovalikon, joka näyttää kaikki projektissa olevat tyypit. Jos halutaan käyttää Revitin omaa tai MagiCAD-objektia ja sitä ei ole vielä kertaakaan käytetty projektissa, se ei näy Family Types-lohkossa. Objekti täytyy kerran manuaalisesti sijoittaa mihin tahansa projektissa, jotta sen tyyppi tulee projektiin. Family Types-lohko on kömpelö valikko sijoitettavan objektin valintaan varsinkin, jos projektissa on paljon perheitä. Lohko sisältää hakuominaisuuden, jos sijoitettavan objektin nimi on tiedossa. Toinen vaihtoehto sijoitettavan objektin valintaan on `FamilyType.ByName` komento,

jolle syötetään tyypin nimi täsmälleen oikein ja lohko hakee pyydetyn tyypin projektista. Myös tämän komennon toiminta vaatii tyypin olevan projektissa.

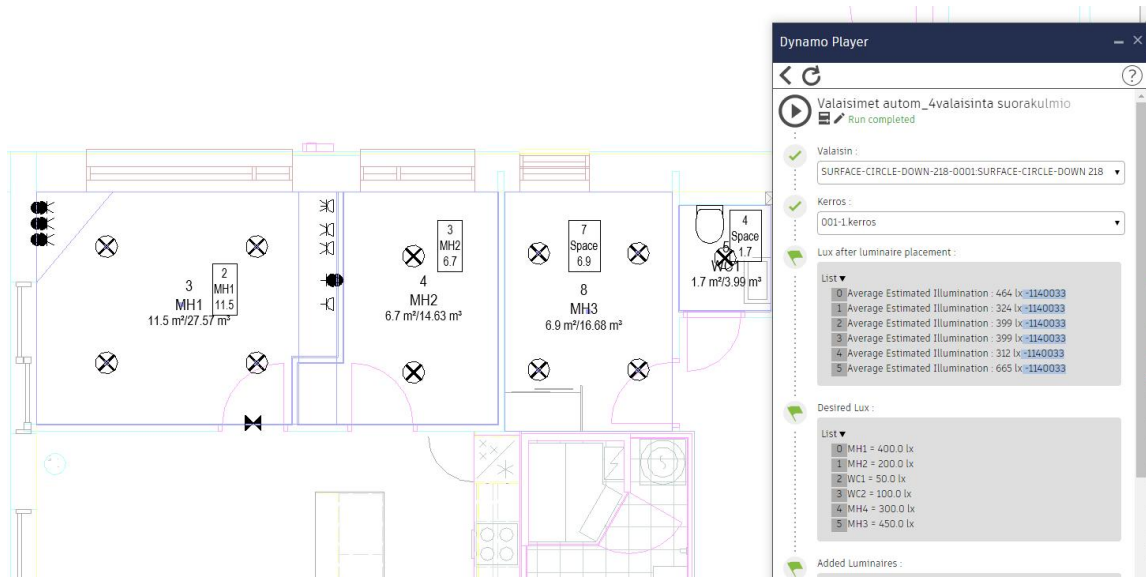
Level eli kerros voidaan valita Levels lohkolla. Piste tai pisteet johon objekti tai objektit sijoitetaan, on objektien sijoitukseen tarvittavista parametreista haastavin. Sijoitettavaan pisteeseen vaikuttavat lähtötietona tai tietona arkkitehtimalli, muiden suunnittelunalojen suunnitelmat ja sähkösuunnitelmat. Arkkitehtimallista pisteen saamiseksi voidaan käyttää apuna esimerkiksi tila tai huonetietoa, kalusteita tai rakenteita kuten ovia tai ikkunoita. Muiden suunnittelunalojen suunnitelmien sijainteja voidaan käyttää lähtötietoina sähkön sijoituksiin ja törmäystarkasteluihin. Sähkösuunnitelmista lähtötiedoiksi saadaan muita sähköasennuksia, jos lisäys tehdään jonkun muun asennuksen mukaan. Objekteja lisätessä tiettyjen elementtien päälle, kuten pöytien tai WC-tiloissa peilin tai allaskaapin päälle, täytyy arkkitehtipohjasta tarkistaa projektikohtaisesti niiden nimet tai tiedot, joilla ne voidaan paikantaa tai niiden täytyy olla aina samoin tavoin nimettyjä.

6.3.1 Valaisimet

Valaistus lisätään yleensä jokaiseen tilaan rakennuksissa ja valaistuksella halutaan saavuttaa tietty valaistusteho. Revit osaa laskea suoraan tilan arvioidun keskimääräisen luksimäärän siihen asetettujen valaisinten perusteella, jos valaisimeen on osoitettu sen oikea valonjakotiedosto. Tilaan voidaan määrittää katon, seinien ja lattian heijastuskertoimet ja haluttu luksimäärä. Tilojen halutun luksimäärän voi asettaa arkkitehti tai sähkösuunnittelija Revitilla tai se voidaan asettaa myös Dynamolla kaikkien huoneiden käyttötarkoituksen perusteella SFS-EN 12464-15 standardissa asetettujen valaistussuosituksen mukaan eri tiloille tai tilaajan toiveiden mukaan. Näitä tietoja voidaan käyttää Dynamossa älykkääseen valaisinaseteluun. Yksinkertaisimmillaan ohjelma lisäisi määrättyä valaisinta asetettuun korkeuteen tietyllä jaolla niin kauan, että haluttu keskimääräinen luksimäärä saavutetaan. Tämä ei kuitenkaan onnistu yhdellä Dynamo-komentosarjalla johtuen Dynamon toiminnasta. Family.InstanceByPointAndLevel-komentoa käytettäessä samalle valaisintyypille uudestaan saman komentosarjan sisällä, komento poistaa kaikki edellisellä kerralla lisätyt objektit. Lisäämällä uusi Family.InstanceByPointAndLevel-komento samalle objektille antaa sekavia tuloksia, jolloin komentosarja ei lisää ja poista objekteja oikein. Helpoin tapa kiertää tämä ominaisuus on luoda jokaiselle valaisinmäärälle oma komentosarja. Tämä myös rajoittaa komentosarjojen käyttöä valaistussuunnitteluun, jokaisen valaisinmäärän vaatiessa oman komentosarjansa

omilla valaistusjaoillaan. Valaisinten sijoitus yhdellä komentosarjalla voisi onnistua Python-koodilla.

Valaisimille tehtiin kolme komentosarjaa, jotka toimivat suorakaiteen tai lähes suorakaiteen muotoisissa huoneissa. Ensimmäinen komentosarja lisää jokaiseen haluttuun tilaan yhden valaisimen keskelle huonetta. Seuraava kahden valaisimen komentosarja lukee tiloihin asetetun halutun valaistustehon ja yhdellä valaisimella saavutetun keskimääräisen valaistustehon ja poistaa jatkokäsittelystä tilat, joissa haluttu keskimääräinen valaisinteho saavutetaan yhdellä valaisimella. Tiloista, joissa yksi valaisin ei riittänyt, poistetaan huoneen keskelle lisätty valaisin. Seuraavaksi komentosarja luo huoneesta kuution sen leveyden, pituuden ja korkeuden mukaan. Pidempi sivu jaetaan neljään osaan ja otetaan tämä pituus vektoriin. Pisteet joihin valaisimet lisätään, saatiin ottamalla huoneen keskipiste ja siirtämällä pistettä pidemmän sivun jaosta saadun vektorin verran. Molempiin pisteisiin lisätään valaisimet. Seuraavassa vaiheessa tarvittaessa poistetaan edellisessä vaiheessa lisätyt kaksi valaisinta ja lisätään neljä. Neljä pistettä saatiin lähtemällä vektorilla huoneen keskipisteestä neljäsosan pituuden ja leveyden verran neljään suuntaan. Kuvassa 6.2 on ensimmäiseen kerrokseen ajettu yhden, kahden ja neljän valaisimen ohjelmat Dynamo Playerilla. Player näyttää myös Dynamossa määritetyt ulostulot, joista nähdään huoneen nykyinen keskimääräinen valaistusteho valaisinten lisäyksen jälkeen ja huoneeseen haluttu valaistusteho. Nykyiseen valaistustehoon ei saanut liitettyä eteen huoneen nimeä Dynamon rajoitusten vuoksi. Yrittäessä ottaa parametrin arvoa erilleen alkutekstistä muuttui arvo nolllaksi. Tiedot ovat samassa järjestyksessä halutun valaistustehon kanssa, joten huoneen nimen voi tarkastaa tästä listasta vastaavasta numerosta. Ohjelma näyttää myös listan lisätyistä valaisimista, joiden piirrosmerkkinä pinta-asennettavina alasvaloina käytetään ympyrää, jonka sisällä on rasti.



Kuva 6.2 Valaisin komentosarjojen käyttö Dynamo Playerin kautta

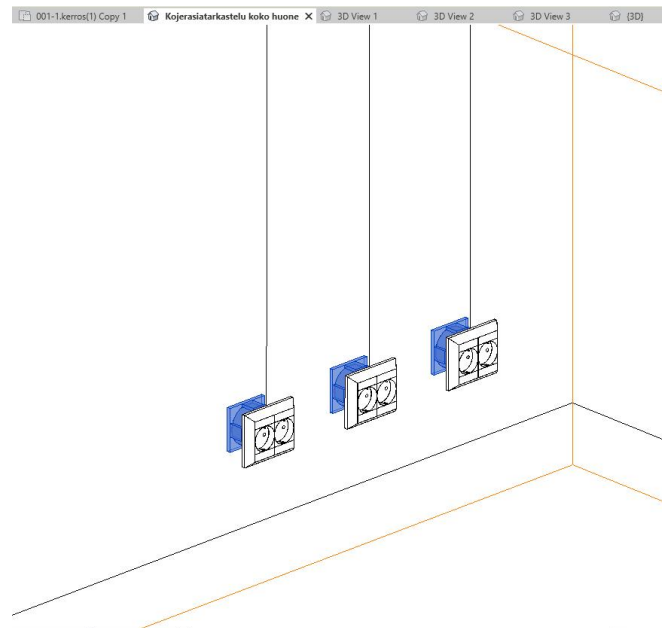
6.3.2 Kojerasiat ja putkitus

Kojerasioita lisätään uppoasennuksiin seinän sisään esimerkiksi pistorasioihin, valokatkaisijoihin ja yleiskaapeloinnin rasioihin. Kojerasioiden mallinnus helpottaa uppoasennusten kartoittamista, jolloin niihin voidaan varautua jo hyvissä ajoin rakennusvaiheessa, eikä porata kaikkia reikiä kojerasioille jälkikäteen.

Dynamolla tämä onnistuu valitsemalla kaikki sähköasennukset ja suodattamalla niistä pois asennukset, jotka eivät tarvitse kojerasiaa. Seuraavaksi haetaan kyseisten asennusten pisteet ja lisätään samaan pisteeseen kojerasia. Kyseinen komentosarja voidaan lisätä suoraan muihin komentosarjoihin, jotka sijoittavat uppoasennuksia tai ajaa niiden lisäämisen jälkeen erikseen. Asennuksista haetaan myös kulma ja käännetään kojerasia samaan kulmaan Dynamon sijoittaessa objektit oletuksena 0° kulmaan.

Uppoasennukset vaativat putkituksen sähkökaapeleille. Revitissa tämä onnistuu Conduits-valikon alta löytyvillä putkituksilla. Putkitus lähtee kojerasiasta ja nousee suoraan ylöspäin kattoon saakka. Dynamolla putkitusten lisääminen kaikkiin lisättyihin kojerasioihin onnistui luomalla tarkoitusta varten Python-koodi, koska tämä ei Dynamon omilla komennoilla onnistu. Koodin ensimmäinen sisääntulo on alkupiste, joka saadaan kojerasiasta, toisena sisääntulona loppupiste, jonka x- ja y-arvot saadaan kojerasian sijainnista ja z-arvolla määrätään putken ylempi korko vastaamaan huonekorkeutta tai lähimmän johtotien korkoa. Kolmantena sisääntulona on putkituksen tyyppi, joka haettiin Element Types- ja edelleen All

Elements of Type-komennoilla, jonka jälkeen valitaan listasta haluttu putkituksen tyyppi ja viedään se Python-koodille. Viimeisenä sisääntulona Python-koodille on kerros eli Level, joka saadaan samasta Levels-valikosta kuin aiemmissa komentosarjoissa. Python-koodi lisää näillä lähtötiedoilla valitun putkituksen. Korko lasketaan valitun kerroksen nollakorosta. Komentosarjan tulos näkyy kuvassa 6.3. Komentosarja lisäsi onnistuneesti edessä näkyvien 2-osaisten uppoasennettavien pistorasioiden taakse kuvassa sinisellä näkyvät kojerasiat ja niille kojerasialta suoraan ylöspäin putkitukset huonekorkoon saakka.



Kuva 6.3 Kojerasiat ja putkitus

6.3.3 Paloilmaisimet

Paloilmaisimien sijoittamiseen liittyvät säännöt on mainittu kappaleessa 2.2.1. Paloilmaisimien sijoitukseen saadaan paljon reunaehtoja standardista ja ohjeesta. Komentosarjan tulee tunnistaa tilasta, eli arkkitehtimallista, tilan pinta-ala, käyttötarkoitus, korkeus ja alaslasketun katon tiedot kappaleessa 2.2.1 mainittujen ehtojen mukaan. Muuta tilasta komentosarjan tulee tunnistaa katossa olevat muut asennukset kuten valaisimet ja putket ja koneellisen ilmanvaihdon poistoaukot, jos tilan pinta-ala on pienempi kuin valitun ilmaisimen toiminta-ala. Lisäksi tulee tarkastaa, että ilmaisimen ympärillä on 0,2 m vapaata tilaa katossa muusta olevasta tekniikasta.

Paloilmaisimien sijoittamiselle löytyi selkeimmät reunaehdot standardeista ja ohjeista valituista sijoitettavista objekteista, mutta reunaehtojen käyttö Dynamossa osoittautui haastavaksi. Alakatot eivät ole Revitissa sidottuina tila- tai huonetietoon ja sama alakatto voi jatkua usean tilan läpi. Tilan alakattotietoa ja tietoa siitä kuinka suuren osan alakatto tilasta kattaa ei onnistuttu saamaan. Koneellisen ilmanvaihdon poistoaukkoja ei saatu jaoteltua kuuluvuksi tiettyihin tiloihin, joten komentosarja vertasi etäisyyksiä kaikkiin tai väärin poistoaukkoihin.

7. JOHTOPÄÄTÖKSET

Halutessa tuottaa automatisoitua rakennussähkösuunnittelua Suomalaisilla piirrosmerkeillä on tällä hetkellä Autodesk Revit Dynamo-lisäohjelmistolla paras ratkaisu Ramboll Finlandin sähköosastolle. Nykyisin käytössä olevalla Autodesk AutoCAD-ohjelmistolla automatisoitu suunnittelu ei onnistu. Sähkösuunnitelmista löytyy toistuvuuksia ja asennuksia, jotka ovat lähes joka tilassa tai aina tietyn tyyppisissä tiloissa.

Dynamo-komentosarjoilla voidaan säästää suunnittelijan aikaa suorittaessa yksinkertaisia ja muuten aikaa vieviä toimia erityisesti isoissa projekteissa, joissa on paljon samankaltaisia ja toistuvia suunnitteluosuuksia. Dynamo-komentosarjaa ei ole järkevää tehdä, jos sen tekemiseen kuluu enemmän aikaa kuin sillä koskaan suunnittelussa säästetään aikaa. Tästä syystä Dynamoa kannattaa käyttää helposti toteutettavissa ja samaan aikaan manuaalisesti aikaa vievissä suunnitteluosuuksissa. Dynamo-komentosarjojen tulee olla riittävän älykkäitä, jottei komentosarjan virheiden korjaukseen mene enempää aikaa kuin suunnittelun hoitamiseen kuluisi manuaalisesti.

7.1 Komentosarjojen luonti

Dynamo komentosarjojen luontiin rakennussähkösuunnittelun automatisoinnin kannalta tekijältä vaaditaan ymmärrys rakennussähköstä ja sen suunnittelusta, Autodesk Revitin perusteet ja itse Dynamon käytön osaaminen. Jos Dynamon mahdollisuudet halutaan hyödyntää täysimääräisesti, on komentosarjojen tekijän myös osattava luoda komentolohkoja Python- ja DesignScript-ohjelmointikielillä.

Komentosarjoja tehdessä huomattiin Python-koodien olevan välttämättömiä automaattisen suunnittelun toteuttamiseen Dynamolla, etenkin listojen arvojen käsittelyssä. Lisäksi kaikki haluttavat toiminnot, kuten putkitusten tai muiden johtoteiden lisäys ei onnistu Dynamo-lohkoilla. Python-koodeilla voidaan myös vähentää käytettävien käyttäjien luomien Dynamo-pakettien käyttöä suorittaessa sama toiminto Python-koodilla. Python-koodien luominen vaatii Python-kielen osaamista. Python- tai DesignScript-koodeilla saadaan Dynamon komentosarjoista selkeämpiä, kun monta komentolohkoa vaativat toiminnot voidaan suorittaa tekstimuotisen koodin sisässä muutamalla koodirivillä. DesignScript-kielen osaamisella komentosarjoja voi tehdä nopeammin kuin Dynamon komentolohkoilla. Kielellä saa myös

luotua erilaisia toimintoja komentolohkoihin verrattuna yhdistelemällä komentolohkojen toimintoja DesignScript-kielellä.

Tämän työn aikana luotiin Dynamolla komentosarjoja objektien sijoittamiseen Revitiin, jotka ovat esiteltynä taulukossa 1. Komentosarjat ovat toimivia alkeellisia ratkaisuja, jotka todistavat suunnittelun automatisoinnin olevan näiltä osin mahdollista. Komentosarjoja voi kehittää pidemmälle ja luoda uusia, mutta nämä komentosarjat riittivät tämän diplomityön osalta vastaamaan tutkimuskysymyksiin. Samaa komentosarjaa voidaan käyttää asetettavaa tai asetettavia objekteja, sijoituspistettä tai esimerkiksi tilasuodatinta muokkaamalla myös muihin vastaaviin sijoituksiin. Käytetty aika sarakkeessa ei ole otettu huomioon Revit ja Dynamo ohjelmien käytön opetteluun kulunutta aikaa vaan ainoastaan kyseisen komentosarjan luontiin kulunut aika.

Taulukko 1. Työssä luodut komentosarjat

Komentosarja	Perustoiminto	Käyttötarkoitus	Kehitettävää	Käytetty aika (h)
Kojerasiat	Sijoittaa objektin samaan pisteeseen valittujen objektien kanssa, kääntää sen ja lisää pisteestä putkituksen haluttuun korkeuteen	Lisää kojerasian merkattuihin uppoasennuksiin, kääntää sen asennuksen taakse seinän sisään ja lisää pisteestä putkituksen haluttuun korkeuteen	Putkitus lähtee 2d-asennuspisteestä, pitäisi siirtää suunnan mukaan myös seinän sisään, korkeus älykääksi	12
Valaisimet	Sijoittaa yhden, kaksi tai neljä objektia tilan kattoon tasaisella jaolla huoneen muodon mukaan, kunnes valittu arvo täyttyy	Lisää valittuja valaisimia tilan kattoon, kunnes haluttu lukumäärä täyttyy	Suurempiin tiloihin jakoa pidemmälle, käytävämalli, valaisinohjaus, valaisinriipustuskiskot	40
Pienet huoneet	Sijoittaa objektin keskelle huoneen kattoa ja toisen sen viereen määrätyn etäisyyden päähän	Pienten huoneiden (WC, varastot) valaisimen ja paloilmaisimen lisäys	Peilivalo (WC-tilat, pistorasiat, valaisinohjaus	8
Paloilmaisimet	Sijoittaa objektin kattoon	Paloilmaisinten sijoitus standardien ja ohjeiden mukaisesti	Alakattoja ei tunnistettu, koneellisen ilmanvaihdon poistojen rajausta vain kyseiseen tilaan	16

7.2 Komentosarjojen käyttö

Alkeellisista komentosarjoista tulee varmistua ennen käyttöä, että ne toimivat kyseisessä projektissa ja arkkitehtimallissa oikein arkkitehtimallien eroavaisuuksien vuoksi. Tästä syystä alkeellisia komentosarjoja projektiin käytettäessä olisi hyvä, että käyttäjä on perehtynyt Dynamoon, käytettäviin komentosarjoihin ja sen mahdollisiin ongelmiin.

Viedessä komentosarjoja pidemmälle voidaan luoda Revitiin lisäosa, jolla olisi yksinkertaisempi käyttöliittymä monipuolisemmilla lähtötietovalinnoilla, helpommilla toiminnoilla ja ohjeilla, kuin Dynamo Player tarjoaa. Dynamo Player tarjoaa melko yksinkertaisen käyttöliittymän komentosarjoille, jossa valitaan syötteet kuten lisättävät objektit, huoneet ja kerros. Player näyttää myös ulostuloiksi määriteltyjen lohkojen tulokset, kuten lisätyt objektit. Kun Dynamo-komentosarjojen tekijä on varmistunut komentosarjojen toimivuudesta projektissa ja suunnittelija on perehdytetty Dynamo Playerin käyttöön, voi komentosarjoilla toteuttaa automaattista suunnittelua. Itse tehtyjen parempien käyttöliittymien arvo valmiiseen Dynamo Playerin tarjoamaan käyttöliittymään on tulkinnanvarainen, koska niiden käyttö vaatii myös perehdytyksen. Jos komentosarjat tehdään tarpeeksi pitkälle esimerkiksi käyttöliittymän ja niiden käytön opastuksen, komentosarjan aiheuttamien virheiden etsinnän ja poiston, yhteensopivuuden tarkastamisen erilaisten arkkitehtimallien kanssa, yhteensovituksen ja ohjeistuksen puolesta, voisi niitä käyttää tulevaisuudessa kaikki suunnittelijat.

Komentosarjoja testattiin myös AutoCAD-sovelluksesta Revitiin tuotuun arkkitehtipohjaan. Komentosarjojen toiminta AutoCAD-arkkitehtipohjassa vaatii Revitin arkkitehtuuristen seinien piirtoa manuaalisesti oikeaan korkeuteen arkkitehtipohjan mukaisesti. Komentosarjat toimivat näin myös AutoCAD-arkkitehtipohjassa. Revitissa automaattisesti tuotetut suunnitelmat voidaan viedä takaisin AutoCAD-sovellukseen, jossa ne näkyvät block-muodossa tai pelkkinä viivoina.

7.3 Haasteet

Dynamo-komentosarjoista on hankala saada jokaisessa projektissa suoraan toimivaa ratkaisua lähinnä arkkitehtimallien ja tilaajan vaatimusten erilaisuuden vuoksi. Arkkitehtimallista riippuen huoneiden tai tilojen luonti puuttuu tai seinien asetusten vuoksi niitä ei voida suoraan luoda. Käytettäessä arkkitehtimallin objekteja kuten huonekaluja lähtötietoina voivat ne olla nimetty eri projekteissa eri tavalla, jolloin ne täytyy hakea ja nimetä komentosarjaan erikseen tai jos ne sisältyvät johonkin luotuun perheeseen Dynamo ei osaa lukea niitä. Myös

järjestelmäperheisiin kuuluvat rakenteet voivat olla tuotettu malliin eri tyypeillä. Arkkitehtimalleille pitäisi luoda perusvaatimukset millaisia niiden tulisi olla Dynamo-suunnittelun onnistumiseksi ilman, että komentosarjoja tai arkkitehtimallia täytyy komentosarjoja käyttäekseen erikseen muokata jokaiseen arkkitehtimalliin sopiviksi. Tilaajalla voi olla kohteesta erilaiset vaatimukset kuin yleensä toteutettava suunnittelu, mutta vaatimuksia voidaan kirjata reunaehdoiksi ja lähtötiedoiksi Dynamoon.

Paras toistuvuus lähtötiedoksi arkkitehtimallista eri projekteissa löytyy luoduista huoneista ja tilatiedoista. Oikein luotu tila tai huone toimii aina samalla tavalla Revitista haettuna lähtötietona Dynamossa. Oikein luodussa tilassa tai huoneessa sen rajat on määritelty oikein vastaamaan arkkitehtimallin seiniä, joka tapahtuu Revitissa automaattisesti, korkeus, haluttu luksimäärä ja nimi. Huoneita ja tiloja voidaan myös suodattaa, kun ne on nimetty, jolloin komentosarja ei vaikuta kerralla koko projektiin tai kerrokseen.

Yksi haasteista on käyttäjien luomien pakettien käyttö. Paketit voivat tarjota nopean ratkaisun komentosarjan luonnin edistämiseen, joka täytyisi muuten tehdä usealla komentolohkolla ja Python- ja DesignScript-kielillä. Pakettien lohkot voivat antaa myös tuloksia, joita ei tavallisilla Dynamo-lohkoilla saavuteta. Jos paketteja halutaan käyttää, tulisi niitä hallita hyvin, jotta jokaisella käyttäjällä olisi samat paketit ja niistä samat versiot. Pakettien päivitys tulee ajankohtaiseksi Revit- ja Dynamo-päivitysten yhteydessä. Pakettien käyttö ei ole välttämätöntä, mutta ne helpottavat ja nopeuttavat komentosarjojen luomista käytettäessä jo valmiita ratkaisuja.

7.4 Jatkokehitys

Yhdistämällä tässä työssä luotuja ja valmiita Autodeskin kanssa yhteistyössä luotuja komentosarjoja saadaan komentosarjoista monipuolisempia. Oven viereen objektin asettavalla komentosarjalla voidaan asettaa siivouspistorasian lisäksi myös valokatkaisija tai painonappi, tai mikä tahansa muu sähköobjekti. Yhdistämällä komento tässä työssä luotuu yhden valaisimen komentosarjaan voidaan valita ohjaustavaksi valokatkaisija tai painonappi ja komentosarja lisäksi valitun ovien viereen. Valittaessa ohjaustavaksi liiketunnistin se sijoitetaan kattoon. Myös jakorasia sijoitetaan kattoon, jonka tarvitsee valaisinohjauksen toteuttamiseen. Jos huoneisiin halutaan erilaisia ohjaustapoja, komentosarjaan tulisi tehdä myös erillinen versio, jossa ohjaukset lisätään vain yhteen valittuun huoneeseen. Valaisimien lisäyksen

lisäksi tulisi luoda komentosarja, joka asettaa halutun luksimäärän jokaiseen tilaan sen käyttötarkoituksen mukaan. Näin valaisimet-komentosarjat toimisivat suoraan ilman manuaalista luksimäärän syöttöä, jos arkkitehti ei ole niitä lisännyt.

Pienet huoneet-komentosarjaan voi yhdistää esimerkiksi WC-tiloihin peilin päälle valaisimen lisäyksen ja siivouspistorasian lisäyksen. Autodeskin luomalla kaapelihylly komentosarjalla voidaan luoda myös johtokourut kiertämään tilan kaapelihyllyjen sijasta tai lisäksi. Komentosarja voisi myös tunnistaa pöydät tai työpisteet ja lisätä niiden mukaan pistorasioita ja yleiskaapeloinnin rasioita johtokouruihin toimistotiloissa.

Kojerasia-komentosarjassa putkitus pitäisi sijoittaa seinän sisään. Tällä hetkellä komentosarja sijoittaa putkituksen alkamaan seinän pinnasta. Putkituksen korkeuden saa älykkääksi ottamalla koron seinän ja yläpuolella olevan johtotien korosta. Kojerasioita ollessa useampi lähekkäin vierekkäin niille riittää yksi putkitus. Tälle etäisyydelle määritetyllä arvolla voi kojerasioiden putkituksia yhdistää.

Paloilmaisien reunaehtojen ottaminen huomioon Dynamolla osoittautui haasteelliseksi. Paloilmaisien on vaadittu kaikkiin palo-osastolla oleviin tiloihin, joten komentosarjan saattaminen käyttökuntoon olisi järkevää. Tämä vaatii alakattojen tunnistamista, eli tilan osalta onko siinä alakattoa, missä korkeudessa, onko alakatto rei'itetty ja kuinka paljon katon pinta-alasta se peittää. Alakaton jatkuessa usean tilan läpi pitäisi alakatosta ottaa huomioon vain tarkasteltavan tilan kattava osa. Koneellisen ilmanvaihdon poistoaukot täytyisi sijoittaa huoneisiin, joissa ne sijaitsevat niiden käyttämiseksi komentosarjassa lähtötietoina.

Lisäkehitystä vaatisi törmäystarkastelun toteutus. Paras ratkaisu olisi, jos Dynamo voisi tehdä törmäystarkastelua objekteja sijoittaessa ja mukautua siihen sijoittamalla objektin eri kohtaan. Tämä ei ole välttämättä Dynamolla mahdollista. Muut vaihtoehdot voisivat olla, että Dynamo merkkaisi törmäyspaikan tai jättäisi objektin sijoittamatta törmäyspaikkaan ja ilmoittaisi siitä.

7.5 Etenemismalli

Autodesk Revit ohjelmaa käyttäessä kannattaa käyttää hyödyksi Dynamon tuomia mahdollisuuksia suunnitteluosuuksien automatisointiin. Tässä työssä luotuja komentosarjoja voi kehittää pidemmälle tai luoda uusia tarpeen mukaan. Sähkösuunnittelusta voi löytää vielä lisää toistuvuuksia, joita hyödyntää Dynamolla.

Komentosarjan tekemiseen kuluvaa aikaa on vaikea arvioida etukäteen. Moni toiminto ei välttämättä toimi kuten alun perin suunniteltu lähdetessä luomaan komentosarjaa. Komentosarjaa tehdessä huomaa mikä ei Dynamolla suoraan onnistu ja sen jälkeen voi arvioida kannattaako kyseistä ongelmaa lähteä ratkomaan esimerkiksi Python-koodilla, etsiä siihen yksinkertaisempi ratkaisu vai jättää komentosarja tekemättä. Vaikka komentosarjan tekemiseen kuluisi enemmän aikaa kuin saman suunnitteluosuuden toteuttamiseen manuaalisesti yhdessä kohteessa, sovellettaessa useampaan projektiin komentosarjalla voidaan säästää suunnittelijan aikaa.

Rakennussähkösuunnittelun automatisointi kannattaa rajata yksinkertaisiin ja toistuviin suunnitteluosuuksiin. Tehtäessä komentosarja erikoisiin tai kertaluontoisiin suunnitteluosuuksiin ei komentosarjalla todennäköisesti säästetä aikaa. Toisaalta muokkaamalla vanhoja komentosarjoja, jos perustoiminto on sama kuin uudessa suunnitteluosuudessa vaadittu, voidaan saada nopeasti uusi ratkaisu. Komentosarjoja voi luoda tulevaisuudessa aina tarpeen mukaan uusia tai vanhoista muokkaamalla, mutta miettiä aina voiko komentosarjalla oikeasti säästää aikaa verrattuna manuaaliseen tekemiseen.

8. YHTEENVETO

Työn lähtökohtana oli tutkia rakennussähkösuunnittelun automatisoinnin mahdollisuuksia. Nykyisin käytössä olevalla Autodesk AutoCAD-ohjelmistolla ei automaattinen suunnittelu ole mahdollista. Halutessa tuottaa automatisoitua suunnittelua Suomalaisilla piirrosmerkeillä, tällä hetkellä paras ratkaisu on Autodesk Revit ja sen lisäohjelma Dynamo, jolla voi tuottaa graafista- ja tekstimuotoista ohjelmointia. Suunnittelun automatisoitavat osuudet tulisivat olla toistuvia, yleisiä projektien välillä ja manuaalisesti hitaita suorittaa. Sähkösuunnittelusta automatisoitaviksi osuuksiksi tässä työssä valittiin paloilmaisimet, valaistus, johtotiet ja rasiat.

Valittuja suunnitteluosuuksia automatisoitiin luomalla Dynamolla komentosarjoja, jotka toteuttavat suunnittelua Revit-ohjelmaan. Komentosarjojen tekijältä vaaditaan ymmärrys rakennussähköstä ja sen suunnittelusta, Autodesk Revitista ja Dynamosta. Lisäksi jos Dynamon mahdollisuudet halutaan hyödyntää täysimääräisesti, on komentosarjojen tekijän osattava myös Python- ja DesignScript-ohjelmointikieliä. Kaikki halutut toiminnot, kuten johtotien lisäykset eivät onnistu ilman Python-koodeja. Ohjelmointikieliä käyttämällä komentosarjoista saadaan selkeämpiä. Komentosarjan tekemiseen kuluva aika on hankala arvioida etukäteen.

Kojerasioiden lisäys uppoasennusten taakse ja niistä ylöspäin putkitusten lisäys onnistui Dynamolla. Valaisimien lisäys onnistui yhden, kahden ja neljän valaisimen jaolla. Jakoja suuremmalle määrälle valaisimia on helppo luoda. Jokaiselle valaistusjaoille jouduttiin tekemään oma komentosarja Dynamon rajoitusten vuoksi, mutta komentosarjat ovat ehkä mahdollisia yhdistää käyttämällä Dynamon komentolohkojen sijasta Python-koodia. Pienille huoneille, kuten WC tai varastoille, suunniteltiin komentosarja, joka sijoittaa kattoon valaisimen ja paloilmaisimen. Komentosarjaan voi lisätä siivouspistorasian, valaistuksenohjauksen ja esimerkiksi valaisimen lisäyksen peilin päälle WC-tiloissa. Paloilmaisimille luodulla komentosarjalla ei onnistuttu vielä tämän työn aikana tekemään älykästä automaattista suunnittelua. Alakattoja ja koneellisen ilmanvaihdon poistoaukkoja ei pystytty sitomaan tilaan ja tunnistamaan. Työn aikana luodut komentosarjat ovat käyttökelpoisia projektikäyttöön olettaen, että niiden tuottamat suunnitelmat tarkastetaan. Komentosarjojen luontiin tähän muotoon kesti 8-40h per komentosarja.

Komentosarjojen käyttö vaatii perehdytyksen. Käyttäjän tulee olla perillä komentosarjan käytöstä ja mahdollisista virheistä. Komentosarjoja voidaan käyttää Revitissa suoraan Dynamon käyttöikkunasta tai Dynamo Player-ohjelmalla. Dynamo Player on yksinkertaisempi käyttöliittymä, kuin Dynamo-ikkuna. Dynamo Playerilla voidaan valita komentosarjaan sääntulot ja se näyttää myös komentosarjan ulostulot. Komentosarjojen käyttöön voidaan luoda myös omia käyttöliittymiä Revitiin, jotka voivat olla selkeämpiä käyttää kuin Dynamo Player, mutta myös itse tehdyt käyttöliittymät tarvitsevat perehdytyksen niiden käyttöön.

Haasteena on saada komentosarjat toimimaan eri projekteissa. Eniten haasteita aiheuttavat eri tavalla tehdyt arkkitehtimallit. Komentosarjoissa kannattaa käyttää lähtötietona arkkitehtimallista huone- ja tilatietoja. Huone- ja tilatietoihin voidaan kirjata tilan käyttötarkoitus, korkeus ja esimerkiksi haluttu valaistusteho. Käytettäessä oikein tehtyjä huone ja tilatietoja toimivat komentosarjat samalla tavalla arkkitehtimallista riippumatta. Haasteena on myös käyttäjien luomien pakettien käyttö ja tarkemmin ylläpito komentosarjojen luonnissa. Ne sisältävät valmiita komentolohkoja, joissa on monipuolisempia toimintoja kuin Dynamon vakiolohkoissa ja ne tarjoavat valmiin ratkaisun ongelmaan. Pakettien hallinta aiheuttaa ongelmia, sillä jos komentosarjassa on käytetty käyttäjien luomien pakettien komentosarjoja, tulisi komentosarjan käyttäjällä olla asennettuna sama paketti ja mielellään samana versiona. Pakettien ylläpitoon ja hallintaan pitäisi olla jokin ratkaisu ennen niiden käyttöä komentosarjoissa.

Autodesk Revit ohjelmaa käytettäessä kannattaa hyödyntää Dynamon tarjoamat mahdollisuudet rakennussähkösuunnittelun automatisointiin. Tässä työssä luotuja ja Autodeskin kanssa yhteistyössä luotuja komentosarjoja yhdistämällä komentosarjoja voidaan monipuolistaa tai niiden pohjalta luoda uusia. Komentosarjoja kannattaa tehdä vain yksinkertaisten ja toistuvien suunnitteluosuuksien toteuttamiseen. Rakennussähkösuunnittelusta tulisi myös etsiä lisää suunnitteluosuuksia, joita voisi automatisoida.

LÄHTEET

Autodesk 2017. Autodesk Knowledge Network. Revit Products, Project Views. [verkkodokumentti]. [viitattu 9.9.2019] Saatavissa <https://knowledge.autodesk.com/support/revit-products/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/Revit-Document-Present/files/GUID-8B1294DC-617E-48F9-AC18-A998F43384C9-htm.html>

Autodesk 2018. Autodesk University. Revit Families: A Step-By-Step Introduction. Aubin, P. [verkkodokumentti]. [viitattu 9.9.2019] Saatavissa <https://www.autodesk.com/autodesk-university/article/Revit-Families-Step-Step-Introduction-2018>

Autodesk 2019. Autodesk Knowledge Network. Revit Products, About Schedules. [verkkodokumentti]. [viitattu 9.9.2019] Saatavissa <https://knowledge.autodesk.com/support/revit-products/learn-explore/caas/CloudHelp/cloudhelp/2019/ENU/Revit-DocumentPresent/files/GUID-73090B70-8A13-4E12-909C-F25D724D5BA7-htm.html>

Dynamo Primer 2019. The Dynamo Primer. Matt Jezyk & Autodesk. [verkkodokumentti]. [viitattu 9.9.2019] Saatavissa <https://primer.dynamobim.org/index.html>

Harsia, P., Autio, I., Leskinen, M., Piikkilä, V., Savuoja, P. & Välimäki, E., 2004. Sähkösuunnittelun käsikirja. Helsinki: Painokurki Oy

MagiCAD 2019. MagiCAD Electrical, ominaisuudet Revitille. [verkkodokumentti]. [viitattu 2.9.2019] Saatavissa https://www.magicad.com/fi/mc_software/magicad-electrical/#ominaisuudet-revitille

Ramboll 2019. Ramboll Finland Oy, Yritys. [verkkodokumentti]. [viitattu 5.11.2019] Saatavissa https://fi.ramboll.com/ramboll_finland_oy

SFS-EN 12464-1 2011. Valo ja valaistus. Työkohteiden valaistus. Osa 1: Sisätilojen työkohteiden valaistus. Helsinki: Suomen Standardisoimisliitto, 2011.

SFS-EN 54-1 2011. Paloilmoittimet. Osa 1: Johdanto. Helsinki: Suomen Standardisoimisliitto, 2011.

SFS-EN 60598-1 2015. Valaisimet. Osa 1: Yleiset vaatimukset ja testit. Helsinki: Suomen Standardisoimisliitto, 2015.

ST-ohjeisto 1 2019. Paloilmoittimen suunnittelu, asennus ja ylläpito. Espoo: Sähkötieto ry.