



Rakennussähkösuunnittelun automatisointi Revit Dynamolla

Jaakko Kontunen

OPINNÄYTETYÖ
Helmikuu 2022

Talotekniikan ylempi tutkinto-ohjelma

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Talotekniikan ylempi tutkinto-ohjelma

KONTUNEN, JAAKKO:
Rakennussähkösuunnittelun automatisointi Revit Dynamolla

Opinnäytetyö 74 sivua, joista liitteitä 67 sivua
Helmikuu 2022

Rakennussähkösuunnittelua on automatisoitu yleisesti hyvin vähän Revitin Dynamolla. Opinnäytetyön tavoitteena oli auttaa Granlundia ymmärtämään Revit-ohjelman Dynamo-skriptien hyödyt ja haasteet rakennussähkösuunnittelun automatisoimisessa. Kyselytutkimuksella selvitettiin eniten toivottuja automatisointeja. Työssä kehitettiin Dynamolla uusia tarpeellisia skriptejä, joilla voidaan nopeuttaa suunnittelua, parantaa laatua tai vertailla vaihtoehtoja. Skriptejä testattiin todellisella suunnittelukohteella ja pyrittiin mittaamaan hyödyt, etsimään ongelmakohtia ja todentamaan skriptien hyödyntäminen käytännössä.

Laaduntarkastukseen luotiin skripti, jolla etsitään puuttuvia elementtejä tiloista halutuilla vaatimuksilla sekä skripti, jolla tutkitaan elementtien ja johtoteiden korkeuksia alakattojen suhteen. Automaattipiirtoon luotiin skripti, jolla voidaan piirtää elementtejä referenssimallien avulla ja kopioida siihen haluttuja tietoja. Automaattipiirtoon luotiin myös skripti, jolla ohjelma siirtää elementit, jotka eivät ole alakatossa, oikeaan korkoon sekä skripti, joka kirjoittaa tekstiä elementin parametrille, jotka ovat alakaton yläpuolella. Määrälaskennan tekeminen suunnittelun alkuvaiheessa ratkaistiin luomalla mallitilat erinimisistä tiloista, jolloin voidaan skriptillä kertoa niiden tuotteiden lukumäärät vastaavan tyyppisten tilojen lukumäärällä sekä myös niiden pinta-alojen suhteella. Skriptit todettiin toimiviksi todellisessa suunnitteluympäristössä. Johtoteiden osalta huomattiin rajoitteita, koska niiden sijainti määritellään keskipisteen perusteella. Jos arkkitehti ei käytä Revit-ohjelmaa suunnitteluun, heikentää se automaattipiirron mahdollisuuksia.

Revit-ohjelman Dynamo-työkalulla voidaan luoda skriptejä, joilla nopeutetaan suunnittelua ja parannetaan laatua sekä voidaan tehdä kattava määrälaskenta yleissuunnittelun jälkeen kohtuullisella lisävaivalla. Dynamo voidaan käyttää Playerin avulla, joka mahdollistaa kaikille suunnittelijoille skriptien käyttämisen hallitusti, kun järjestetään lisäksi suunnittelijoille koulutusta ja tehdään skripteille ohjeet. Skriptit voidaan luoda hyvin tarkasti tekemään vain tiettyä asiaa, sekä niitä voidaan yleistää, jolloin ne sopivat useampiin tilanteisiin. Laaduntarkastus-skriptejä tulee kehittää jatkossa luomalla tarkempia vaatimuksia niihin. Automaattipiirto tehostuu merkittävästi, kun saadaan vakiodut tietosisällöt käytäntöön, jolloin aikaa ei kulu sopivien parametrien tutkimiseen.

Asiasanat: rakennussähkösuunnittelu automatisointi revit dynamo

ABSTRACT

Tampere University of Applied Sciences
Master's degree in Building Services Engineering

KONTUNEN JAAKKO:
Automation of building electrical design with Revit Dynamo

Bachelor's thesis 74 pages, appendices 67 pages
February 2022

Automation of building electrical design is not widely used by Revit Dynamo. The aim of the thesis was to help Granlund Oy understand the benefits and challenges of Revit Dynamo scripts in automation of building electrical design. Need for Revit automation tasks was asked with the survey of Granlund electrical designers. New useful scripts were made in the studies. Scripts speed up design and improve the qualities of plans. The scripts were tested with real projects. The aim was to measure and verify benefits and look for problems in designing.

For quality inspection were made scripts that find missing elements in rooms. Also were made scripts that inspect if the elements height is the same as the ceilings and fix those faults. For automatic drawing were made Dynamo-scripts that can automatically draw elements by using other companies' models and copy information from them. For elements calculation of quantity in the early stages of design was made a Dynamo-script that can estimate number of elements in whole project. Script calculates elements in the example rooms and multiplies those with total number of same named rooms and also with area ratios of those. Dynamo-scripts were proved to be useful but some scripts have limitations. Cable tray has limitation because its locations can only be spotted by its insertion point. If architect doesn't use Revit it reduces opportunities of automatic drawings.

With Dynamo-scripts you can speed up and improve quality of your designing. Elements calculation of quantity can be made in the early stages of design by doing little extra work. Script can be used also with Player. Player helps everyone to use script if simple manuals have been made. Script can be made for a one job or those can be made to use very widely by generalizing them. It's possible to improve accuracy inspection scripts by adding formulas of demanding in those. Automatic drawings will work much better when standardization of parametric information is made.

Key words: building electrical design automation revit dynamo

SISÄLLYS

1	JOHDANTO.....	7
2	SÄHKÖSUUNNITTELU JA SÄHKÖSUUNNITTELUN AUTOMATISOINTI REVIT DYNAMOLLA	9
2.1	Sähkösuunnittelu ja sähkösuunnittelun tehtävät ja työvaiheet.....	9
2.2	Tietomallipohjainen suunnittelu (BIM)	10
2.2.1	Revit ja MagiCAD	10
2.2.2	Yleiset tietomallivaatimukset 2012.....	13
2.3	Sähkösuunnittelun automatisointi Revitin Dynamolla.....	13
2.3.1	Revitin Dynamo ja Python-ohjelmointikieli.....	13
2.3.2	Ohjelmallinen ajattelu ja skriptin hahmottaminen	18
2.3.3	Sähkösuunnittelun automatisointi Dynamolla Granlundilla. 19	
2.3.4	Tutkimukset Dynamon hyödyntämisestä suunnittelussa	19
3	KYSELYTUTKIMUS JA ASIAANTUNTIJAHAASTATTELUT	21
3.1	Kyselytutkimus sähkösuunnittelun automatisoimisen tarpeista....	21
3.2	Asiantuntijahaastattelut	24
3.2.1	BIM Developer Marko Stenmanin haastattelu	24
3.2.2	Varatoimitusjohtaja Antti Törmälän haastattelu	25
3.3	Yhteenveto kyselytutkimuksesta ja haastatteluista	27
4	DYNAMO-SKRIPTIEN KÄYTTÄMINEN, LUOMINEN JA TESTAAMINEN.....	29
4.1	Dynamo-skriptien hallinta ja käyttötavat.....	29
4.2	Dynamo-skriptien käyttäminen suunnittelun eri vaiheissa.....	31
4.3	Dynamo-skriptien luominen ja testaaminen	33
4.3.1	Skriptien luomisen vaiheet.....	33
4.3.2	Testattavien kohteiden tiedot.....	34
4.3.3	Piirustus- ja tulostusnäkyvien automaattinen luominen.....	34
4.3.4	Määrälaskenta mallitilojen avulla	37
4.3.5	Elementtien automaattinen mallintaminen muiden suunnittelualojen tietomallien perusteella	43
4.3.6	Elementtien automaattinen piirtäminen alakattoihin	45
4.3.7	Elementtien asennuskorkeuden tarkastaminen ja muuttaminen alakaton perusteella	46
4.3.8	Johtoteiden asennuskorkeuden tarkastaminen alakaton perusteella.....	50
4.3.9	Puuttuvien elementtien listaaminen automaattisesti.....	51
4.4	IFC-tiedoston kääntäminen Revit-tiedostomuotoon.	54
4.5	Luodut Dynamo-skriptit ja niiden onnistumiset.....	55

5	DYNAMO-SKRIPTIEN HYÖDYNTÄMINEN SUUNNITTELUN AUTOMATISOINNISSA	58
5.1	Dynamo-skriptien käyttöönottoaminen Granlundilla.....	58
5.2	Määrälaskennan tekeminen varhaisessa suunnitteluvaiheessa Dynamo-skriptien avulla	58
5.3	Suunnittelun nopeuttaminen ja laadun parantaminen Dynamo-skripteillä.....	61
5.4	Dynamon ohjelmalliset rajoitukset.....	63
5.5	Dynamo-skriptien käytön haasteet	64
6	JOHTOPÄÄTÖKSET JA POHDINTA.....	66
6.1	Ajatuksia päätulosista	66
6.2	Tutkimuksen luotettavuus.....	69
6.3	Jatkotutkimustarpeet	70
	LÄHTEET	72
	LIITTEET	74
	Liite 1. Kyselytutkimus sähkösuunnittelun automatisoimisesta Dynamolla.....	74
	Liite 2. Dokumentti Dynamo-skriptien luomisesta	74
	Liite 3. Skripti: Piirustus- ja tulostusnäkyvien automaattinen luominen 74	
	Liite 4. Skripti: Määrälaskenta mallitilojen avulla.....	74
	Liite 5. Skripti: Elementtien automaattinen mallintaminen muiden suunnittelualojen tietomallin perusteella.....	74
	Liite 6. Skripti: Elementtien asennuskorkeuden tarkastaminen ja muuttaminen alakaton perusteella	74
	Liite 7. Skripti: Johtoteiden asennuskorkeuden tarkastaminen alakaton korkeuden suhteen.....	74
	Liite 8. Skripti: Puuttuvien elementtien listaaminen automaattisesti....	74

ERITYISSANASTO

API	Ohjelmointirajapinta (Application Programming Interface)
BIM	Rakennuksen tietomalli (Building Information Modeling)
Category	Familyt jaetaan eri kategoriaeihin. Esim. valaisimet
Dataset	MagiCADin tuotekirjasto
Dynamo	Visuaalinen ohjelmointityökalu
Dynamo Player	Työkalu, jolla voidaan käyttää Dynamoa ilman ohjelmointinäkömää
Family	objekteja, joilla on saman tyyppisiä ominaisuuksia
In-PlaceFamily	Family, joka on luotu yksinomaan tietylle projektille.
Loadable Family	ladattavat ulkoiset Familyt. Ladataan RFA-tiedostoista
Magicad	Talotekniikan suunnitteluohjelmisto
MC System Name	Sähköjärjestelmän nimi
MC System Code	Sähköjärjestelmän koodi
Node	Dynamossa solmu, joka tekee yhden toiminnon
Python	Ohjelmointikieli
Revit	BIM-ohjelmisto, jolla tehdään suunnittelua
Skripti	Ohjelman käsikirjoitus
System Family	Revitiin valmiiksi määritellyt Familyt
TAG	Merkintä piirustuksen elementin tunnistamiseksi
TATE18	Taloteknisen suunnittelun tehtäväluettelo
Template	Ohjelman mallipohja
YTV 2012	Yleiset tietomallivaatimukset 2012

1 JOHDANTO

Rakennussähkösuunnittelua on automatisoitu yleisesti hyvin vähän. Opinnäytetyön tavoitteena oli auttaa Granlundia ymmärtämään Revitin Dynamo skriptien hyödyt ja haasteet talosähkösuunnittelun automatisoimisessa kehittämällä Dynamolla uusia tarpeellisia skriptejä, joilla voidaan nopeuttaa suunnittelua, parantaa laatua tai vertailla vaihtoehtoja sekä kokeilla niitä todellisella suunnittelukohteella. Opinnäytetyössä pyrittiin vastaamaan seuraaviin tutkimuskysymyksiin:

1. Miten dynamo skriptit toimivat todellisessa suunnittelukohteessa osana projektia?
2. Minkälaisilla dynamo skripteillä voidaan säästää aikaa tai parantaa laatua?
3. Voidaanko dynamo skripteillä tehdä kustannustehokasta vertailua ilman suurta työmäärää?
4. Kuinka monimutkaisia dynamo skriptejä suunnittelijat uskaltavat käyttää?
5. Minkä tyyppisiä skriptejä voidaan käyttää yleissuunnitteluvaiheessa ja minkälaisia toteutussuunnitteluvaiheessa.
6. Voidaanko laskentamateriaalia tuottaa skriptien avulla jo varhaisessa vaiheessa

Työssä keskityttiin vain Revitin Dynamon käyttöön rakennussähkösuunnittelun automatisoimisessa. Tutkimus tehtiin Granlundin kehitys ja data -osastolle sekä sähkösuunnittelijoille, jotka käyttävät MagiCAD for Revit -ohjelmistoa.

Työssä selvitetiin, voidaanko tehdä kattavia määrälaskentoja suunnittelun alkuvaiheessa, pystytäänkö käyttämään automaattipiirtoa muiden suunnittelualojen tietomallia hyödyntäen, miten voidaan automatisoida toistoa vaativia toimenpiteitä, sekä voidaanko tehdä tarkastusskriptejä, joilla etsitään virheitä ja puutteita. Samalla kerättiin tietoa eteen tulevista ongelmakohtista sekä pyrittiin ratkaisemaan niitä.

Tutkimustyössä perehdyttiin Dynamon ohjelmointityökaluun sekä Python-ohjelmointikieleen. Tämän jälkeen kehitettiin uusia Dynamo-skriptejä. Uudet skriptit

testattiin todellisilla kohteilla sekä mitattiin ja tutkittiin niiden hyötyjä. Työssä tarvittiin ymmärrystä sähkösuunnittelusta sekä erilaisten suunnitteluprojektien läpivientiprosesseista, jotta voitiin ymmärtää, minkälaisista skripteistä on hyötyä missäkin suunnitteluvaiheessa.

Tutkimuksen aikana tehtiin kyselytutkimus Granlundin sähkösuunnittelijoille, jonka avulla selvitettiin, minkälaisia sähkösuunnittelun automatisointeja heidän mielestään on tarvetta kehittää. Selvitettiin Granlundin nykyinen tilanne sähkösuunnittelun automatisoinneista Revit Dynamolla. Haastateltiin rakennuttamisen asiantuntijaa ja keskusteltiin, millä tavoilla voitaisiin tehostaa sähkösuunnittelua Dynamo-skripteillä rakennuttajan näkökulmasta.

2 SÄHKÖSUUNNITTELU JA SÄHKÖSUUNNITTELUN AUTOMATISOINTI REVIT DYNAMOLLA

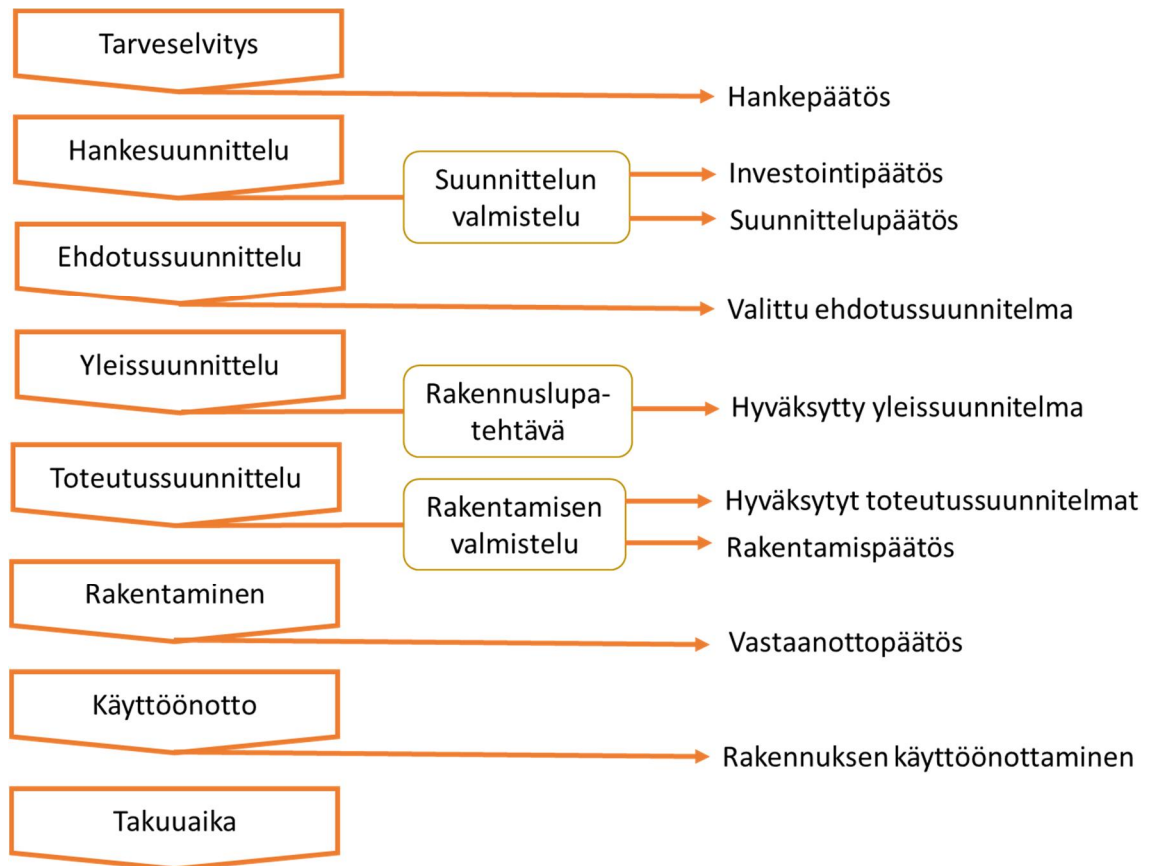
2.1 Sähkösuunnittelu ja sähkösuunnittelun tehtävät ja työvaiheet

Sähkösuunnittelijan tehtäviin kuuluu rakennuksen sähköjärjestelmien ja tietoteknisten järjestelmien suunnittelu tilaajan vaatimuksien mukaisesti. Sähköjärjestelmiin kuuluvat ovat esimerkiksi rakennuksen jakokeskukset, pistorasiat ja valaisimet sekä niiden kaapelointi. Tietoteknisiin järjestelmiin kuuluvat esimerkiksi antennirasiat, yleiskaapelointirasiat eli ATK-rasiat, paloilmaisimet ja valvontakamerat. Suunnittelu sisältää usein selvittelyitä, keskusteluita, laskelmia, vertailuja, piirustuksia, tietomallinnusta, kaavioita, selostuksia, yhteensovitusta sekä paljon muuta.

Suunnittelutehtävät määritellään suunnittelusopimuksessa usein RT-ohjekorteilla, joissa määritellään suunnittelijan tehtävät ja tehtävälaajuudet. (Junnonen, J-M. Kankainen J. 2017). Sähkösuunnittelun eri työvaiheet on esitetty taloteknisen suunnittelun tehtäväluettelossa TATE18. Suunnittelu jakautuu vaiheisiin tarveselvityksestä takuu aikaan. Hankekohtaisesti määritellään tarvittavat tehtävät ja niiden suorittajat. (RT 10-1129. 2013). Suunnitteluun kuuluva järjestelmälaajuus ilmoitetaan TATE18 tehtäväluettelon liitteessä 1, jossa tilaaja voi määrittellä kuuluvat järjestelmät valitsemalla laajuuden rakennustyyppin perusteella tai sitten projektikohtaisesti rastittamalla halutut järjestelmät. Esimerkiksi määrälaskenta ei kuulu sähkösuunnittelijan normaaleihin työtehtäviin. (RT 10-1129. 2013)

Hankesuunnittelussa asetetaan rakennushankkeelle tavoitteet ja teetetään tarvittavat selvitykset. Ehdotussuunnittelussa tehdään vaihtoehtoisia suunnitteluratkaisuja. Yleissuunnittelussa kehitetään ehdotussuunnitelmaa valitun ehdotussuunnitelman mukaisesti. Yleissuunnittelussa voi myös olla erilaisia vaihtoehtoja. Rakennuslupatehtävissä selvitetään suunnittelijoiden kelpoisuudet ja rakennuslupa-asiat sekä laaditaan rakennuslupa-aineisto. Toteutussuunnittelussa tehdään hyväksytystä yleissuunnittelusta toteutuspiirustukset rakentamista ja hankintoja varten. Käyttöönötossa pidetään tarkastukset ja todetaan järjestelmien oikeat toiminnot. Takuuajana tarkastellaan toimivuutta ja tehdään tarvittavat korjaukset ja säädöt. (RT 10-1129. 2013).

Rakennushanke alkaa päätöksestä, kun halutaan rakentaa uutta tai korjata vanhaa. Rakennushankkeen hallitsemista ja oikean aikaista päätöksen tekoa varten hankea jaetaan ajallisesti eri vaiheisiin. Kuviossa 1 on esitetty rakennushankkeen vaiheet. (Junnonen, J-M. Kankainen J. 2017).



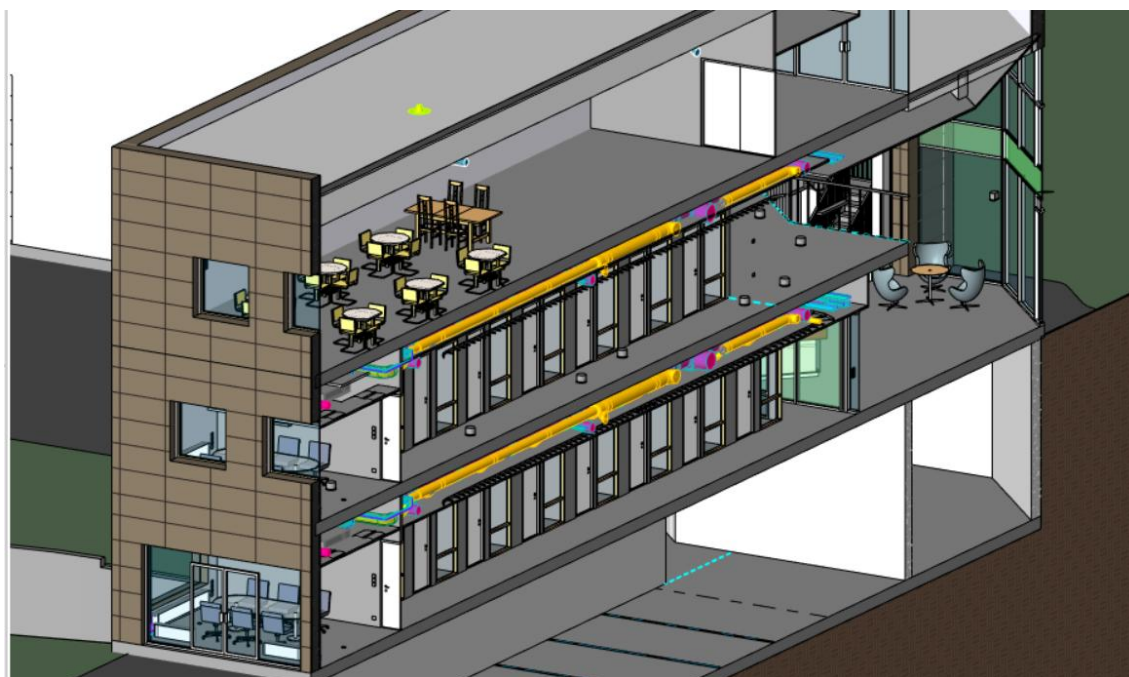
kuvio 1. Talonrakennushankkeen vaiheet. Alkuperäinen lähde (Junnonen, J-M. Kankainen J. 2017)

2.2 Tietomallipohjainen suunnittelu (BIM)

2.2.1 Revit ja MagiCAD

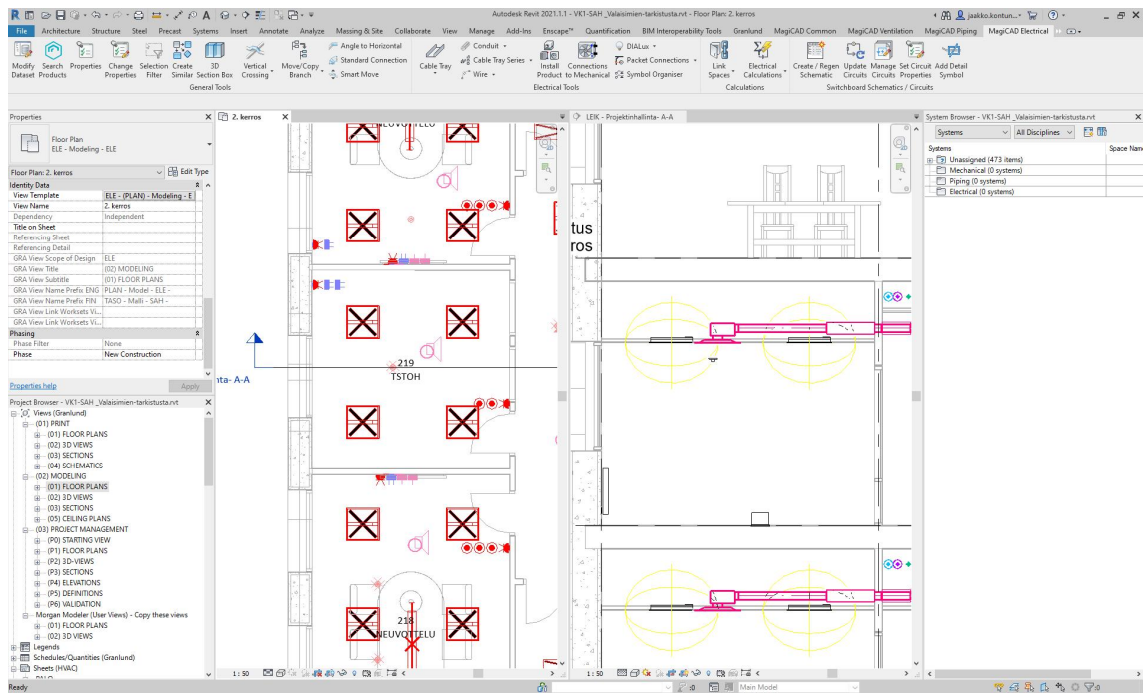
BIM on lyhenne, joka tulee sanoista Building Information Modelling, jolla tarkoitetaan rakennusten tietomallinnusta. Tietomalli sisältää rakennuksen 3D-mallin lisäksi suuret määrät informaatiota. (MagiCAD, Mitä on BIM. 2021). Autodesk Revit on BIM-suunnitteluohjelma, jolla tehdään arkkitehti-, rakenne-, ja talotekniikkasuunnittelua. 3D-grafiikassa näkyvät elementit sisältävät tietoa. (Arcance

System. Revit. 2021). Revit voi koostua yhdestä tai monesta tietokannasta. Tietokannat voivat olla paikallisia tai linkitettyjä. Revitissä käyttäjä hallitsee tietokantaa Revitin sisällä, jolloin muutokset tarvitsevat tehdä vain kerran. AutoCAD puolella tietokannan muutokset on päivitettävä jokaiseen piirustukseen erikseen. (MagiCAD for Revit – PV opetusmateriaali. 2018). Kuvassa 1 on esitetty Revitin 3D-tietomallin näkymä.



Kuva 1. Revitin 3D-tietomallin näkymä

MagiCAD on ohjelmisto, jolla tehdään LVI- ja sähkösuunnittelua. MagiCAD Electrical sähkösuunnitteluohjelma toimii integroituna Revit-ohjelmistoalustaan. Ohjelmalla voidaan tehdä sähkö-, valaistus-, tele-, ja datajärjestelmien suunnittelua ja laskentaa. (MagiCAD, MagiCAD Electrical. 2021). Kuvassa 2 on esitetty Revitin käyttöliittymä, jossa ylhäällä on valikot ja työkalut, vasemmalla näkyvät Properties-valikon parametrit ja keskellä näkyy piirustusnäkymä. Piirustusnäkyymiä voi olla useita kerralla, joissa esitetään samaan aikaan esimerkiksi piirustus-, leikkaus- ja taulukkonäkymä.



Kuva 2. Revit-ohjelman käyttöliittymän näkymä

Revit-ohjelmassa tietoa käsitellään parametreinä. Parametrit sisältävät informaatiota, laskennallista tietoa ja ulkoasuun liittyviä ominaisuuksia. Parametrien avulla elementit myös liittyvät toisiin elementteihin ja päivittävät tietoa niiden tarvittaessa. Suunnittelijalla on mahdollisuus muokata parametreja missä tahansa näkymässä ja tiedot ovat aina ajan tasalla. Esimerkiksi 3D-mallissa tehtävät muutokset näkyvät pohjakuvissa. (Arcance System. Revit. 2021)

Revitissä elementtejä lajitellaan erilaisten parametrien avulla. MagiCAD kirjoittaa eri järjestelmille "MC System Code" ja "MC System Name" -parametreille järjestelmätietoa. Näiden tietojen avulla määritellään mihin järjestelmään elementit kuuluvat sekä hallitaan näkymiä eri tilanteissa. (Stenman, M. 2021). S2010-sähkönimikkeistön päätarkoitus on toimia kattavana jäsentelyä ja luokitteluna rakennus- ja kiinteistöalalle koko sen elinkaaren ajan. Tietomallipohjaisessa suunnittelussa sähköjärjestelmät luokitellaan sähkönimikkeistön mukaan. Sähkönimikkeistöä käytetään suunnittelun lisäksi laajasti eri käyttötarkoituksissa, kuten kirjoissa, määrälaskennoissa ja kaupankäynnin järjestelmänkoko- naisuuksien luokitteluissa. (ST 70.12. 2017). Tietosisältöjen vakioiminen pohjautuu myös sähkönimikkeistön käyttöön (Stenman, M. 2021).

2.2.2 Yleiset tietomallivaatimukset 2012

Tietomallinnuksen päätavoitteena on suunnittelun ja rakentamisen laadun, tehokkuuden ja turvallisuuden tukeminen (Building Smart Finland. 2012. Yleiset tietomallivaatimukset YTV 2012). Suunnittelusopimuksissa viitataan poikkeuksetta YTV 2012 mukaisiin määritelmiin. Yleiset tietomallivaatimukset ovat kehittämissankkeen COBIMin tulos. Joka on tehty yhteistyössä monien ammattiosaajien kesken, jotka koostuvat suunnitteluyrityksistä, rakennuttajista, rakennusyryksistä ja ohjelmistokehittäjistä. (Building Smart Finland. 2012. Yleiset tietomallivaatimukset YTV 2012 Osa 1).

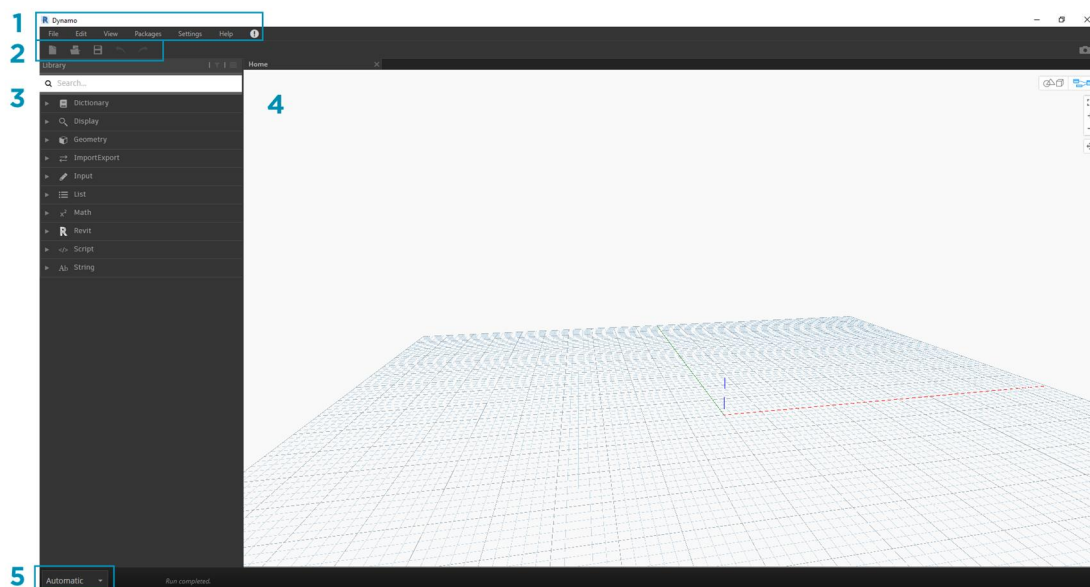
YTV on jaettu 14 osaan, jossa talotekniikalle on tehty oma osio 4. Talotekninen suunnittelu -osiossa käydään läpi eri suunnitteluvaiheiden tehtäviä sekä laatuvaatimuksia ja -tavoitteita. YTV:n mukaan talotekniikkasuunnitteluvaiheet jakautuvat kahteen osaan, joista ensimmäinen osa on ehdotus- ja yleissuunnittelu ja toinen osa on toteutussuunnittelu. Ehdotussuunnitteluvaiheessa ei ole vielä tarvetta tehdä tietomallia, mutta yleissuunnitteluvaiheessa tehdään energiasimulointeja, palvelualuekaavioita, järjestelmävalintoja sekä suunnitellaan mallihuoneita ja talotekniikan pääreittejä, joiden avulla on tarkoitus tuottaa riittävästi tietoa arkkitehti- ja rakennesuunnittelijoille. Toteutussuunnitteluvaiheessa rakennukselle tehdään kattavat järjestelmämallit. (Building Smart Finland. 2012. Yleiset tietomallivaatimukset YTV 2012 Osa-4).

2.3 Sähkösuunnittelun automatisointi Revitin Dynamolla

2.3.1 Revitin Dynamo ja Python-ohjelmointikieli

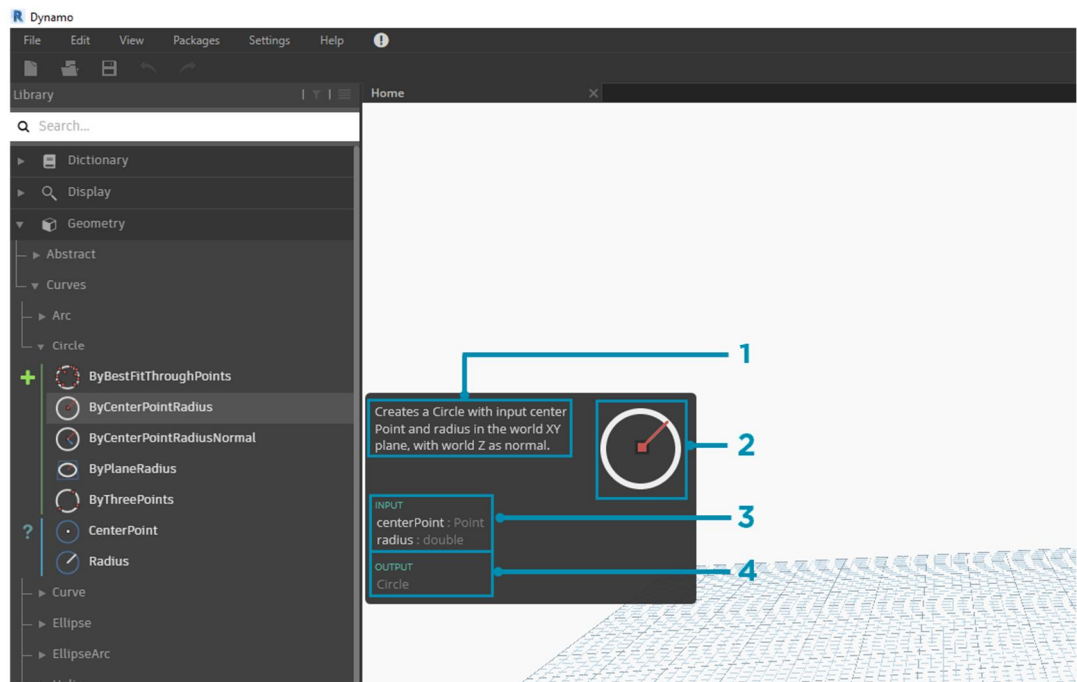
Dynamo on Revit-ohjelman oma sisäänrakennettu visuaalinen avoin ohjelmointityökalu. Ohjelmoinnissa yhdistetään elementit toisiinsa suhteiden ja toimintosarjojen perusteella, jotka muodostavat mukautettuja algoritmeja. Dynamon avulla voidaan tehdä uusia työkaluja, joilla voidaan hyödyntää ulkoisia kirjastoja ja Autodeskin tuotteita, joissa on API-rajapinta. Ohjelmisto käyttää avointa lähdekoodia. (Dynamo Primer, What is Dynamo? 2021)

Dynamo-ohjelman käyttöliittymä jakaantuu viiteen osa-alueeseen. Kuvassa 3 on esitetty käyttöliittymän näkymä. Ylhäältä löytyvät valikko (1) ja työkalut -palkit (2). Vasemmassa reunassa on kirjasto (3), josta löytyvät haettavat solmut (Nodes). Iso näkymä on työskentelytila (4) ja vasemmassa alanurkassa on toteutuspainike (Run) (5). (Dynamo Primer, The Dynamo User Interface. 2021)



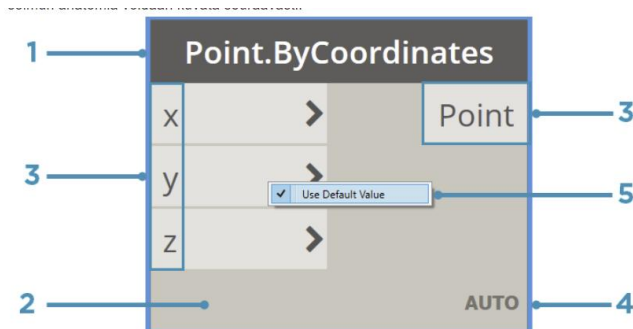
Kuva 3. Dynamon käyttöliittymä (Dynamo Primer, The Dynamo User Interface. 2021)

Kirjastosta (Library) löytyy kahdeksan eri luokkaa. Dynamo esittää kirjastossa solmun (Noden) tietokentät toiminnoista, jotka auttavat ymmärtämään mitä solmu tekee, mitä tietoja sille pitää antaa ja missä muodossa on ulostulo. Kuvassa 4 on esitetty Dynamon kirjaston tietokentän näkymä. Ohjelmassa on myös FIND-toiminto, jolla voidaan etsiä hakusanoilla solmuja.



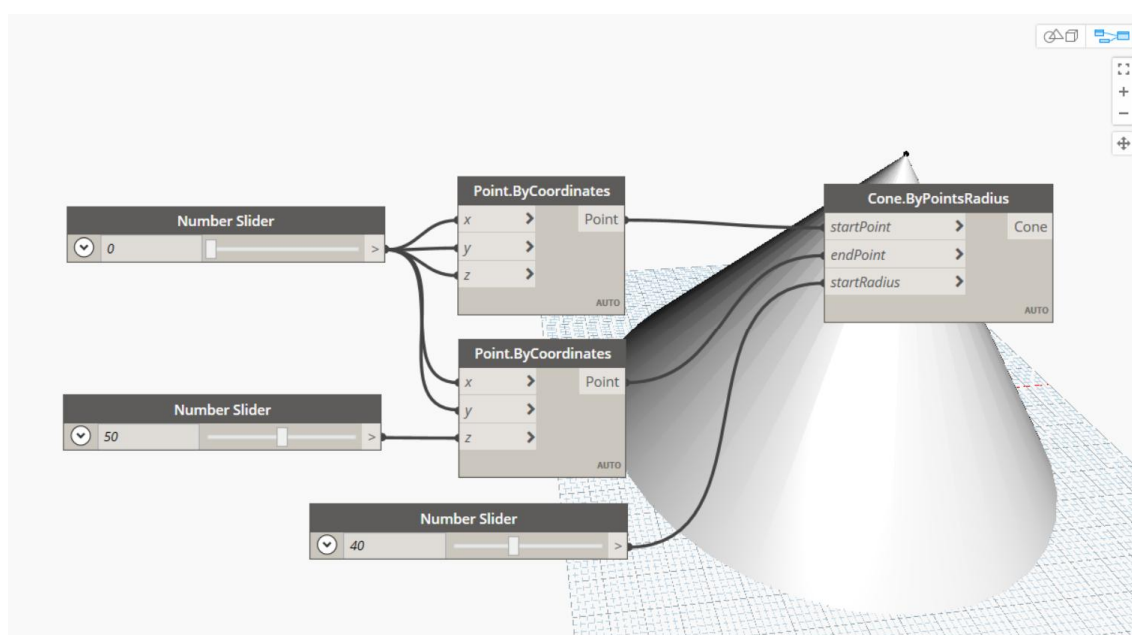
Kuva 4. Dynamon kirjasto ja sen esittämät tietokentät solmusta (Dynamo Primer, The Dynamo User Interface. 2021)

Dynamossa solmu koostuu viidestä osasta. Kuvassa 5 on esitetty solmun osat. Ylhäällä on solmun nimi (1), alhaalla on solmun runko (2), vasemmassa reunassa on solmun sisääntulot ja oikeassa reunassa solmun ulostulo (3), Oikeassa alakulmassa on Lacing (4) ja hiiren oikealla painikkeella voidaan valita oletusarvo (default) (5). (Dynamo Primer, Nodes. 2021). Solmu ilmoittaa virheestä väreillä. Keltainen väri tarkoittaa, että solmu ei voinut tehdä kaikille tuleville elementeille toimintoa, mutta suoritti tehtävään loppuun. Esimerkiksi tyhjiällä (null) arvoilla käy näin. Punainen väri tarkoittaa, että solmu ei toimi, ja tällöin skriptin suorittaminen keskeytyy. Solmuja voi ryhmitellä graafisessa näkymässä ja antaa ryhmälle otsikon ja värin. Tämä on suotavaa varsinkin, jos ohjelma kasvaa suureksi. (Dynamo Primer, Managing Your Program. 2021).



Kuva 5. Dynamo-solmun osat

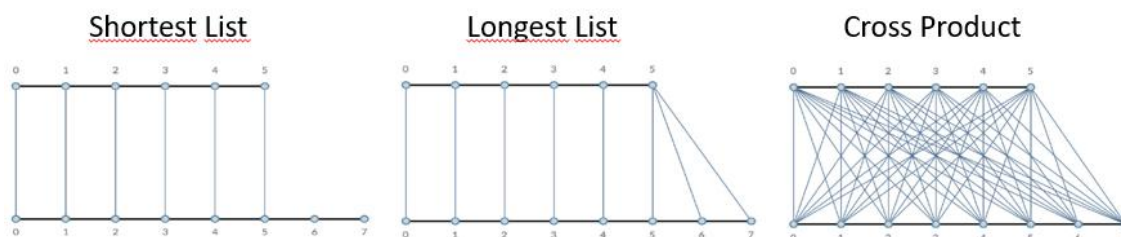
Dynamossa solmut yhdistetään toisiinsa johdoilla (Wires), joilla määritellään tietovirtauksen polku. Ohjelman suorittaminen voidaan tehdä manuaalisesti tai automaattisesti. Dynamo esittää visuaalisen koodin lisäksi 3D-grafiikkaa, kun luodaan geometriaa. Grafiikka esitetään taustalla, jota on myös mahdollista tarkistella kääntämällä ja tarkentamalla näkymää. (Dynamo Primer, Getting Started. 2021). Kuvassa 6 on esitetty yksinkertainen koodi, jolla luodaan kartio. Kartio näkyy taustalla 3D-grafiikkalla. Taustan koordinaatisto ja mittasuhteet vastaa Revit-ohjelman koordinaatistoa, jolloin geometrisia tapahtua voi tarkastella myös Dynamon näkymässä esimerkiksi uusien piirrettävien elementtien sijaintipisteet.



Kuva 6. Dynamo-näkymä, jossa luodaan skriptillä takana esiintyvä kartio.

Tietoa käsitellään Dynamossa listoina, jos tietoa on enemmän kuin yksi. Listojen käsittelyssä on tärkeää ymmärtää solmun Lacing eli tietojen käsittelytapa, joita on neljä erilaista. Shortest eli lyhin suorittaa tehtävät molempien listojen järjestysnumeron mukaisesti. Jos toinen lista on pidempi, sen loput arvot jäävät suorittamatta. Longest eli pisin suorittaa tehtävät molempien listojen järjestysnumeron mukaisesti, mutta pidemmän listan loput arvot se suorittaa lyhemmän listan viimeisimmän arvon perusteella. Cross eli risti suorittaa tehtävät ristiin molempien listojen kaikille järjestysnumeroille, jolloin saadaan kaikki mahdolliset arvot tulokseksi. Kuviossa 2 on havainnollistettu erilaiset käsittelytavat. (Dynamo Primer, What's a List. 2021). Listat voivat sisältää toisia listoja. Tällöin listoille muodostuu

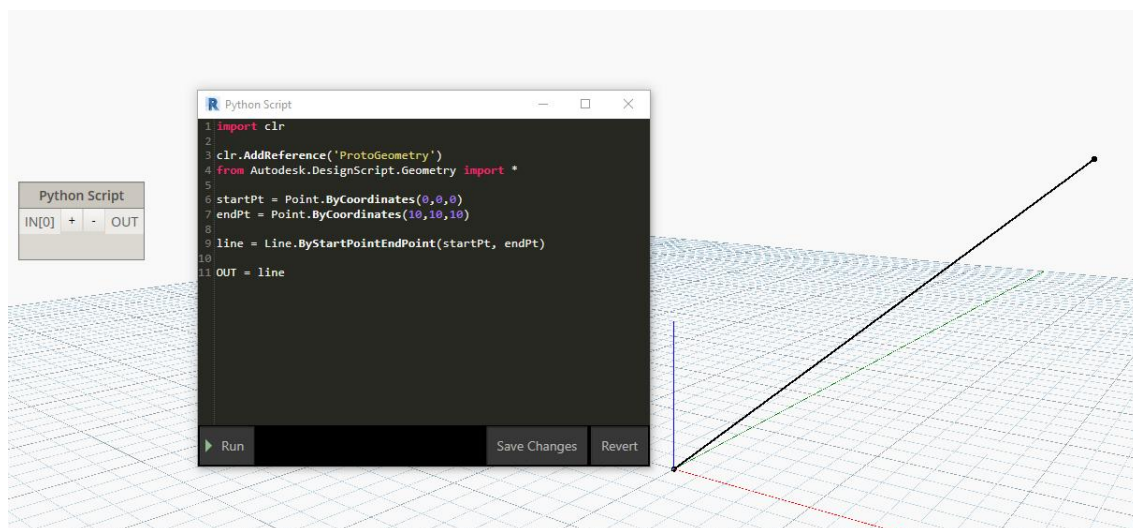
eri tasoja. Eritasojen käsittelyyn voi solmussa vaikuttaa muuttamalla arvoa, jolla määritellään mitä tasoa (Level) tutkitaan tai valitsemalla "Keep list structure", joka säilyttää listan muotoilun. (Dynamo Primer, Lists of Lists. 2021). Listojen erilaisien käsittelytapojen sekä tasojen valintojen avulla voidaan listoja käsitellä aina tarpeen mukaisella tavalla Dynamossa.



Kuvio 2. Listojen erilaiset käsittelytavat (Lacing) Dynamo solmussa. Alkuperäiset kuvat (Dynamo Primer, What's a List. 2021)

Dynamon virallinen sivusto on dynamobim.org. The Dynamo Primer sivuilla kerrotaan Dynamosta sekä esitellään kattavasti erilaisia toimintoja. Sivuilta löytyy harjoituksia sekä esimerkkiskriptejä, joita voi vapaasti ladata ja muokata. Dynamoon ladattavat hyödylliset lisäosat eli paketit, joissa on valmiiksi tehtyjä ohjelman osia, ovat löydettävissä Primerin kohdasta 14.3 Useful Packages. (Dynamo Primer, 2021.) Dynamo-Forum alusta on luotu ihmisten välisiin keskusteluihin (Dynamobim forum. DynamoBIM forum. 2021). Forumilla henkilöt keskustelevat, jakavat tietoa sekä auttavat toisia ihmisiä ongelmien ratkaisuisa.

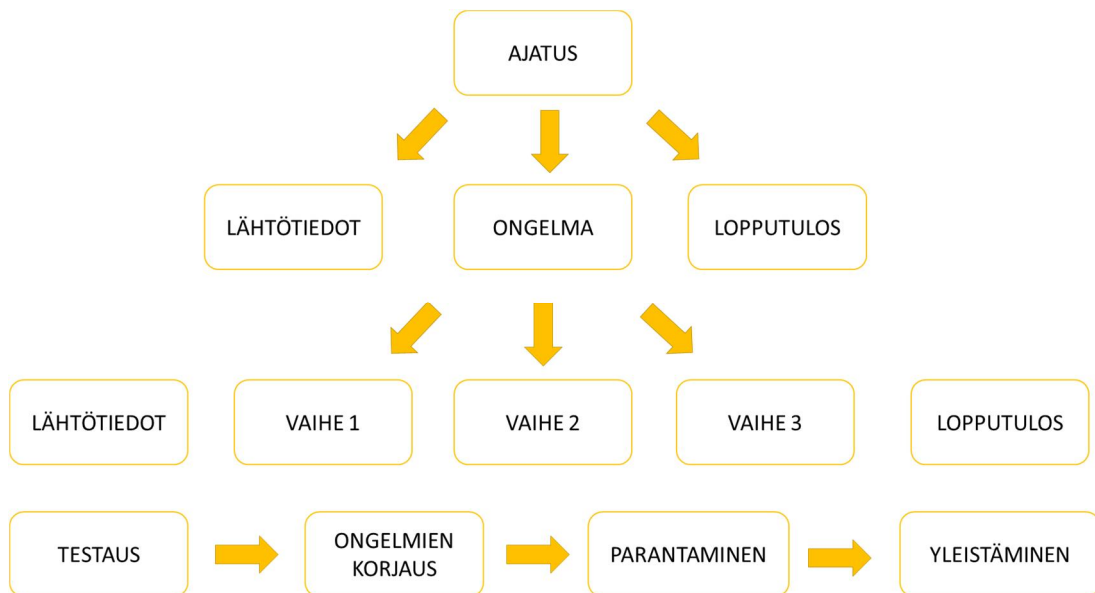
Dynamossa on solmu, jolla voidaan tehdä koodia Python-ohjelmointikielellä. Python on maailmalla laajasti käytetty ohjelmointikieli. Pythonissa voidaan käyttää moduuleita ja paketteja. Kokoelmasta moduuleita muodostuu standardikirjasto, joita voidaan käyttää eri tarkoituksiin. Esimerkiksi moduuli math sisältää matemaattisia funktioita. (Ohjelmointiputka, Oppaat: Python-ohjelmointi. 2021.). Python-ohjelmointi on tehokkaampaa kuin visuaalinen ohjelmointi, varsinkin kun käytetään ehtolauseita ja silmukoita. Tietyissä tilanteissa voidaan muutamalla rivillä koodia tehdä sama kuin monimutkaisella visuaalisella koodilla. (Dynamo Primer. Python Nodes. 2021). Kuvassa 7 on esitetty Dynamon Python-solmun näkymä, jonka avulla piirretään yksinkertainen viiva.



Kuva 7. Python-solmun näkynä Dynamossa, jonka koodilla piirretään yksinkertainen viiva

2.3.2 Ohjelmallinen ajattelu ja skriptin hahmottaminen

Skriptien rakentaminen on ohjelmointia, joka perustuu ohjelmalliseen ajatteluun. Ongelmat ovat ratkaistava vaiheittain, jotta voidaan ratkaista varsinainen ongelma. Jaetaan kolmeen vaiheeseen ratkaisut. Ensimmäiseksi vaiheessa puretaan ongelma pienempiin osiin ja tunnistetaan ratkaisun kannalta merkittävimmät kohdat. Tarvittaessa jaetaan osat vielä pienempiin osiin. Toiseksi muodostetaan eri vaiheet ratkaisun kannalta. Kolmanneksi automatisoidaan ja yleistetään ratkaisu, jolloin sitä voidaan käyttää samanlaisissa tehtävissä (Jouko Järvenpää. 2019). Kuviossa 3 on esitetty skriptin luomisen prosessikaavio, jossa ajatuksesta muodostetaan erivaiheet ja lopuksi skripti testataan, parannetaan ja yleistetään.



Kuvio 3. Skriptin luomisen prosessikaavio.

2.3.3 Sähkösuunnittelun automatisointi Dynamolla Granlundilla

Granlundilla Revitin Dynamoa käytetään Revit-projektien perustamisessa projektitietojen muokkaamisissa kuten kielen ja suunnittelualojen valinnoissa. Skripti poistaa ylimääräiset kielet ja suunnittelualat, jolloin jää jäljelle vain tarvittavat tiedot. Lisäksi käytetään yksinkertaisia skriptejä, joilla lisätään tai siirretään parametritietoa elementeille sekä automaattinen TAG-työkalu, joka tekee merkin elementin tunnistamiseksi vain tuotteille, joilla sitä ei vielä ole. Automaattipiirroksessa on demokäytössä Dynamo-skripti, jolla voidaan piirtää oven pieleen elementtejä kuten valaistusohjauspainike. Automaattipiirto tunnistaa oven sijainnin, mittatiedot sekä aukeamissuunnan, jonka perusteella skripti sijoittaa elementin aina oikealle puolelle. (Stenman, M. 2021)

2.3.4 Tutkimukset Dynamon hyödyntämisestä suunnittelussa

Arkkitehti- ja rakennesuunnittelun osalta löytyy useita tutkimuksia Dynamon hyödyntämisestä suunnittelussa. Rakennuspuolen automatisointeja ei voida kuitenkaan suoraan käyttää sähkösuunnittelussa, koska tarpeet ovat erilaiset. Säh-

kösuunnittelussa Dynamon hyödyntämisestä löytyy vähän tutkimuksia. Suomenkielisiä diplomitoita tai opinnäytetöitä Dynamon hyödyntämisessä sähkösuunnittelusta onnistuttiin löytämään vain yksi.

2019 vuonna Lappeenrannan-Lahden Teknillisessä Yliopistossa Henri Leskinen oli tehnyt Ramboll Finland Oy:lle diplomityön rakennussähkösuunnittelun automatisoimisesta. Työssä oli tutkittu vastaavasti Revitin Dynamon hyödyntämistä sähkösuunnittelun automatisoimisessa. Työssä oli tutkittu valaisimien, paloilmalaitteiden ja johtoteiden automaattista sijoittelua malliin sekä rasia-asenteiden tuotteiden automaattista putkitusta. Yhteenvedossa osa Dynamo-skripteistä todettiin mahdollisiksi käyttää projekteissa. Dynamo-skriptejä suositeltiin tehtäväksi yksinkertaisten ja toistuvien suunnitteluosuuksien automatisointeihin. (Leskinen, H. 2019).

2021 vuonna Metropolia Ammattikorkeakoulussa Teemu Svens oli tehnyt Granolundille opinnäytetyön Revit Dynamon soveltaminen KSL-suunnitteluun. Työssä oli tutkittu parametrien muuttamista, laiteluettelon tulostamista sekä kalusteiden automaattisista sijoittelua pohjakuviin. Laiteluettelon automaattinen tulostaminen sekä parametrien muokkaaminen onnistui työssä. KSL-laitteiden automaattista sijoittelua pohjakuviin ei saatu onnistumaan itsenäisesti. (Svens, T. 2021).

3 KYSELYTUTKIMUS JA ASIANTUNTIJAJAHAASTATTELUT

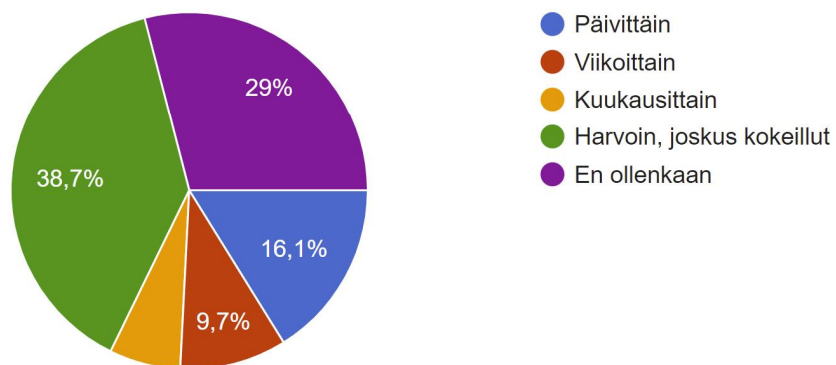
3.1 Kyselytutkimus sähkösuunnittelun automatisoimisen tarpeista

Osana opinnäytetyötä tehtiin Granlundin sähkösuunnittelijoille kyselytutkimus. Kysely tehtiin hyödyntäen Google Forms -ohjelmaa. Kyselyn tarkoitus oli selvittää mihin osa-alueisiin kannattaa työntekijöiden mielestä keskittyä sähkösuunnittelun automatisoinnissa Revitin Dynamolla. Kyselyä varten mietittiin valmiiksi, mitä on mahdollista automatisoida dynamolla, ja sen perusteella tehtiin esimerkkejä kyselyyn. Skriptit jaettiin toimintojen perusteella eri osa-alueisiin. Kaikki kysymykset eivät liittyneet suoraan Dynamolla tehtävään ohjelmointiin, vaan haluttiin selvittää myös Revitin mallipohjan (Template) vakioasetuksia. Kysely on esitetty liitteessä 1.

Kysely lähetettiin kaikilla Granlundin sähköosaston henkilöille, joita oli yhteensä noin 250. Kaikki sähköosaston henkilöt eivät kuitenkaan tee suunnittelua, mutta heille haluttiin antaa mahdollisuus vastata kyselyyn. Kyselyyn vastasi 32 henkilöä. Kuviossa 4 on esitetty vastaavien tiedot. 70 prosenttia on käyttänyt Revit-ohjelmaa. Päivittäin tai viikoittain Revit-ohjelmaa käyttäviä henkilöitä vastanneista oli 26 prosenttia. Vastanneiden suunnitteluvuosijakauma oli tasainen, jolloin vastauksissa näkyvät uusien ja kokeneiden työntekijöiden mielipiteet.

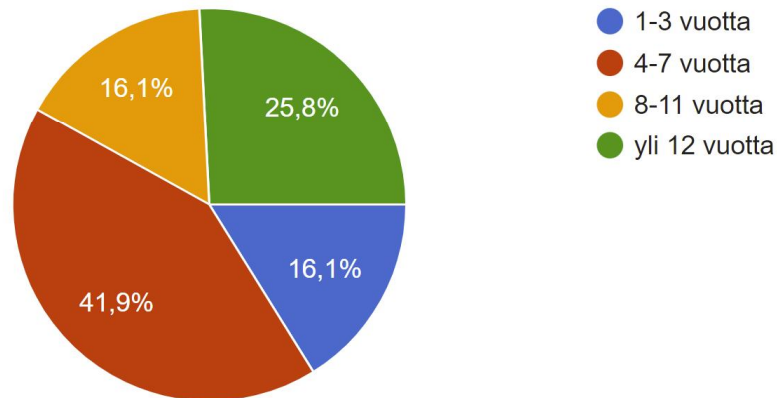
Kuinka usein käytät Revitiä suunnittelussa

31 vastausta



Suunnittelukokemus

31 vastausta

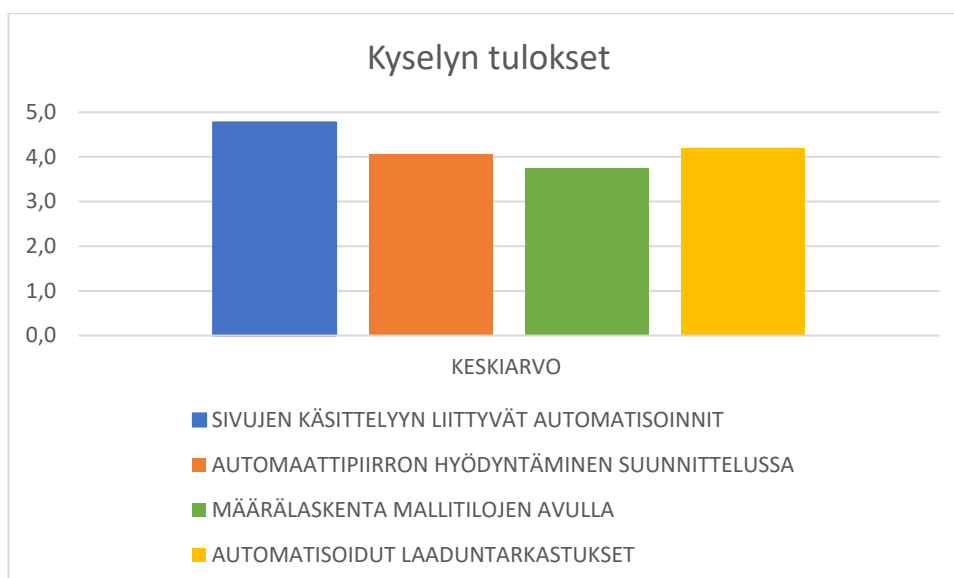


Kuvio 4. Kyselyyn osallistuneiden vastaajien tiedot

Kysely jakautui viiteen eri osa-alueeseen.

1. osassa kysyttiin sivujen käsittelyyn liittyvistä automatisoinneista
2. osassa kysyttiin automaattipiirron hyödyntämisestä suunnittelussa
3. osassa kysyttiin määrälaskennasta mallitilojen avulla
4. osassa kysyttiin automatisoiduista laaduntarkastuksista
5. osassa kysyttiin vastaajan tietoja

Osissa oli ehdotettu erilaisia automatisointimahdollisuuksia ja suunnittelijaa pyydettiin antamaan arvosana yhdestä viiteen sen perusteella, näkeekö hän toimenpidettä hyödyllisenä. Jokaisen osan jälkeen vastaajalle annettiin mahdollisuus kirjoittaa omia ajatuksia, joita tuli myös useita. Kuviossa 5 on esitetty kyselyn eri osa-alueiden keskiarvot ehdotetuille automatisointien hyödyllisyyksille.



Kuvio 5. Sähkösuunnittelun automatisointi -kyselyn eri osa-alueiden hyödyllisyyden pisteytys

Sivujen käsittelyyn liittyviä automatisointeja halusivat lähes kaikki suunnittelijat, joiden arvosanaksi saatiin 4.8/5. Kommenttikentässä oli hyviä tarkentavia huomioita kuten, jos lisätään urakkalaskennan jälkeen jotain automaattisesti, pitää revisioinninkin tapahtua automaattisesti. Revisioinnilla tarkoitetaan merkintätapaa, jolla osoitetaan muuttuneet tiedot suunnitelmissa.

Automaattipiirron hyödyntäminen suunnittelussa sai arvosanaksi 3,9/5. Automaattipiirtotyökaluja nähtiin hyödyllisiksi, mutta osa näki niissä riskejä. Sähkömerkkien piirto LVI-mallin perusteella sai eniten pisteitä. Alakattoihin liittyvissä automaattipiirroissa nähtiin ongelmaksi, että piirrosmerkit on kuitenkin käsin yhteensovittettava ja kohdistettava alakattoruutuihin, joka mahdollisesti aiheuttaa enemmän työtä kuin kertaalleen piirto käsin. Piirretyille tuotteille tiedon lisääminen jälkikäteen oli myös toivottu ominaisuus, kuten yleiskaapelointirasioiden automaattinen numerointi. Omissa ajatuksissa kerrottiin järkeviä toiveita, joita kannattaisi automatisoida, esimerkiksi: ”*erilaisten lisätietomäärittelysten automatisointi, esim. jos asennus jää alaslasketun katon yläpuolelle, automaattinen "AY"-merkintä*”.

Määrälaskenta mallitilojen avulla osio sai arvosanaksi 3,8/5. Esimerkkeinä annettiin massoittelutyökalu ilman tarkkaa sijoittelua, määrälaskenta, jossa mallitilojen

tuotteet kerrottaisiin tilojen määrällä sekä määrälaskenta, jossa mallitilojen tuotteet haettaisiin toisesta kohteesta ja kerrottaisiin tilojen määrällä.

Automatisoidut laaduntarkastukset saivat arvosanaksi 4,2/5. Niissä ei nähty riskejä, koska sinällään ne eivät tee mitään mallille. Laaduntarkistuksissa esimerkeinä annettiin muun muassa tilojen listauksia, joista puuttuu jokin haluttu elementti, korkojen tarkistuksia suhteessa alakattoon tai lattiaan. Vapaissa kommentteissa ehdotettiin alakattokorkojen vertaamista tuotteisiin sekä automaattitarkistuksia asioista, jotka usein unohtuvat tai ovat virheellisiä suunnittelussa.

Kyselyn lopputuloksena huomattiin, että kaikki ehdotetut automatisointi-ideat olivat toivottuja. Osassa ehdotuksia nähtiin riskejä kuten, että automatisointi työllistää enemmän kuin hyödyntää. Näitä riskejä pyrittiin selvittämään skriptien testauksella.

3.2 Asiantuntijahaastattelut

3.2.1 BIM Developer Marko Stenmanin haastattelu

Haastattelun tarkoitus oli selvittää Granlundin tämänhetkinen tilanne sähkösuunnittelun automatisoinnista Revitin Dynamolla sekä keskustella, minkä tyyppisiä skriptejä olisi mahdollisia toteuttaa ja miten skriptien käyttöä tulisi hallita. Haastateltavaksi valittiin Granlundin BIM Developer Marko Stenman, jonka tehtäviin kuului Revitin ja Dynamon kehittäminen Granlundilla. Hän toimi Granlundin kehitys ja data -osastolla ja hänellä on pitkäaikainen kokemus sähkösuunnittelusta sekä hän on osallistunut Revitin kehittämiseen 2016 lähtien.

Granlundilla on tällä hetkellä noin viisi henkilöä käyttää ja kehittää Dynamo-skriptejä. Skriptejä kehitetään tarpeen mukaan uusien projektien yhteydessä. Skriptien tarkoitus on nopeuttaa suunnittelua ja helpottaa toistoa vaativia töitä sekä parantaa laatua. Granlundilla oli käytössä Dynamo skriptejä sähkösuunnittelussa projektin perustamiseen, yksinkertaisia skriptejä parametritietojen lisäämiseen ja siirtämiseen sekä automaattinen TAG-työkalua, joka lisää TAG-tiedot tuotteille, joilta se vielä puuttuu.

Dynamo-skriptien kehitysmahdollisuuksissa Stenman piti tärkeänä tietosisältöjen oikeellisuuden tarkistamista sekä muita laaduntarkastusskriptejä. Dynamon avulla voitaisiin luoda skriptejä, joilla tarkistettaisiin elementtien oikeellisuutta, esimerkiksi vertaamalla elementtien parametrejä valmiisiin Excel-listoihin. Tämä auttaisi elementtien parametrien tietosisältöjen vakioimisessa. Nämä olisivat yksinkertaisia skriptejä, jolloin voitaisiin helposti varmistua niiden toimivuudesta.

Haasteena Stenman koki Dynamon jatkuvan kehityksen. Ohjelman päivityksiä pitäisi ehtiä jatkuvasti seuraamaan, jotta voitaisiin olla tietoisia mitkä asiat ovat muuttuneet. Kaikkia Dynamon solmuja ei löydy vakiona, jolloin tarvitaan kustomoituja paketteja. Tällöin taas skriptit eivät toimi suunnittelijalla, joka ei ole ladanut samoja paketteja. Tämä taas helposti pelästyttää suunnittelijan, joka kokeilee Dynamon käyttöä.

Automaattipiirron haasteista Stenman vastaa, että hän on kehittänyt automaattipiirtoon tarkoitetun skriptin, jolla piirretään oven vierelle sähkötuotteita, ja huomannut, että automaattipiirto vaatii toimiakseen todella paljon huomioimista erilaisista muuttuvista tekijöistä. Esimerkiksi: Oven vieressä ei ole välttämättä riittävästi tilaa sähkörasioille tai rasioiden edessä voi olla pystyviemäri, jolloin niitä ei voida asentaa sen taakse. Ihmisen havainnointikykyä tasopiirustuksen tulkitsemisessä on todella hankala korvata täysin automatisoinnilla, koska tilanteet eivät ole koskaan samanlaisia ja aina löytyy poikkeuksia.

Dynamo-skriptien ohjeiden osalta keskusteltiin, että parhaimpia ohjeita ovat lyhyet videot. Videoissa on esiteltävä skriptin käyttöä riittävän perusteellisesti. Kirjalliset ohjeet ovat myös tärkeä tehdä.

3.2.2 Varatoimitusjohtaja Antti Törmälän haastattelu

Haastattelun tarkoitus oli selvittää, miten rakennuttamisen kannalta voitaisiin hyödyntää sähkösuunnittelun automatisointia Dynamolla. Haastateltavaksi valittiin Granlund Jyväskylän varatoimitusjohtaja Antti Törmälä, jonka tehtäviin kuului

kiinteistöpalveluosaston vetäminen. Osaston tehtäviin kuului muun muassa rakennuttamista, valvontaa, kiinteistöjen teknisiä selvittelyitä sekä huolto- ja kunnossapitoa. Törmälä on koulutukseltaan Diplomi-insinööri ja hänellä on pitkäaikainen osaaminen LVI-puolelta ja energiahankkeista.

Törmälän mukaan rakennushankkeissa on nykyään lyhyemmät hankeaikataulut kuin aikaisemmin. Tällöin joudutaan lyhentämään myös suunnittelu-aikaa, jotta voidaan kilpailuttaa urakoitsijat riittävän ajoissa. Suunnitteluajan lyhentäminen aiheuttaa keskeneräisyyttä suunnitelmiin sekä yhteensovituksen puutteita. Törmälä näki järkevänä kehittämisen aiheena automatisoinnin, joilla voidaan nopeuttaa suunnittelua ja parantaa laatua.

Keskustelimme ajatuksesta, jossa tehtäisiin määrälaskenta koko rakennukselle sähköpisteistä ja johtoteistä mallitilojen avulla. Törmälän mielestä tämä oli toimiva ratkaisutapa talotekniikkaurakoitsijoiden kilpailuttamiselle, mikäli laskentasuunnitelmat täytyisi saada nopealla aikataululla. Määrälaskennan ansiosta saataisiin urakoitsijoille luotua yhdenvertaiset laskentamateriaalit, jotka voitaisiin ottaa sopimuksen pohjaksi. Määrälaskentalistojen määrät ovat myös tilaajan vastuulla, jolloin urakoitsijoiden ei tarvitse myöskään hinnoitella riskiä niin suureksi, joka vähentäisi projektin kustannuspaineita. Määrälaskenta nopeuttaisi myös tarjouslaskijoiden työtä, joka lisäisi kilpailutuksen houkuttelevuutta. Taloudellisessa loppuselvityksessä olisi mahdollista verrata sopimuksen määrälaskentaa valmistuneen kohteen määrälaskentaan, jolloin saataisiin kustannusten seurannalle läpinäkyvyyttä.

Kun keskustelimme sähkösuunnitelmien valvonnasta, Törmälän mukaan tilaaja ei erikseen tarkista suunnitelmia ja sitä myöten puutteita. Vaan suunnittelijat valitaan sopimusperusteisesti siten, että he ovat riittävän päteviä hoitamaan annetut tehtävät laadukkaasti, jolloin vastuu laadusta kuuluu yksinomaan suunnittelijalle. Jos projektilla on sähkövalvoja suunnittelun aikana, on hänen tehtävänsä separata sähkösuunnittelijaa, ehdottaa vaihtoehtoisia ratkaisuja sekä tuoda kustannus- ja aikataulutietoutta. Myöskään sähkövalvojan tehtäviin ei kuulu lähteä etsimään suunnitelmista virheitä ja puutteita.

Törmälän mielestä suunnitelmien laadun parantamisessa voisi keskittyä esimerkiksi puuttuvien tuotteiden, korkovirheiden ja TATE-reititysten oikeellisuuden tarkastamisiin. Törmälää kiinnosti myös, voidaan samoja suunnittelun tarkistuslistoja ja periaatteita käyttää myös työmaan valvonnassa. Voitaisiinko tuottaa listoja, joiden avulla olisi helpompi tarkistaa järjestelmiä tai tilakohtaisesti asennuksia.

Törmälä näki tärkeänä, että kehitystä ei tehdä vain omasta näkökulmasta vaan pyrittäisiin ottamaan huomioon koko rakennushanke ja miettiä miten voitaisiin kehittää rakennuttamista, suunnittelua, valvontaa sekä ylläpitoa. Törmälän mukaan loppujen lopuksi suunnittelu on vain yksi vaihe kokonaisuudessa.

Tilaaajan näkökulmasta, kun yrittää miettiä asiaa, heille kaksi keskeistä teemaa ovat hankeaikataulussa pysyminen sekä kustannusten ennustaminen ja hallinta. Tällöin Granlundin tulee keskittyä kehittämään hankeaikatauluissa pysymistä sekä kustannusten laskemista, ennustamista ja niiden hallintaa. Automatisoinnit, jotka palvelevat näitä, ovat ehdottoman hyödyllisiä Törmälän mielestä.

3.3 Yhteenveto kyselytutkimuksesta ja haastatteluista

Kyselytutkimuksen ja haastatteluiden perusteella nähtiin tarpeelliseksi tutkia ja kehittää erilaisia sähkösuunnittelun automatisointeja Revitin Dynamo -työkalulla. Otettiin tavoitteeksi kehittää erityyppisiä toimivia skriptejä ja kokeilla niitä oikeilla suunnittelukohteilla, jolloin voitaisiin tarkastella monipuolisesti skriptien kipukohtia. Samalla saataisiin selvillä, minkä tyyppisillä skripteillä voitaisiin nopeuttaa suunnittelua tai parantaa laatua. Päätettiin luoda seuraavat skriptit:

- Piirustus- ja tulostusnäkyvien automaattinen luominen. Näkyvien luominen vie normaalisti paljon aikaa ja vaatii paljon toistoa, jolloin automatisoinnin pitäisi toimii hyvin.
- Automaattinen mallintaminen muiden suunnittelualojen perusteella. Voidaan nopeuttaa suunnittelua, kun pystytään mallintamaan kerralla monta elementtiä.

- Asennuskorkeuksien tarkistaminen automaattisesti. Voidaan tarkastaa kerralla iso määrä elementtejä automaattisesti.
- Puuttuvien elementtien etsimisen tiloista. Voidaan tutkia erilaisilla ehdoilla, mistä tiloista puuttuu vielä tarvittavia elementtejä.
- Kattavan määrälaskennan tekeminen mallitilojen avulla. Voidaan tehdä mahdollisimman tarkka määrälaskenta suunnittelun alkuvaiheessa, jolloin saadaan tarkempaa kustannustietoa.

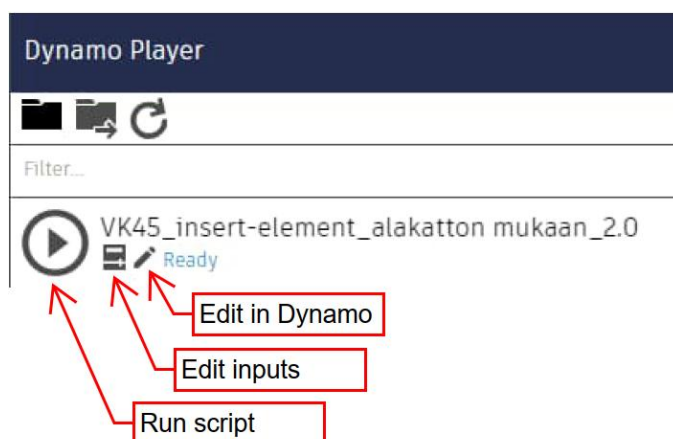
4 DYNAMO-SKRIPTIEN KÄYTTÄMINEN, LUOMINEN JA TESTAAMINEN

4.1 Dynamo-skriptien hallinta ja käyttötavat

Dynamo-skriptiä voidaan käyttää kolmella eri tapaa. Dynamolla ohjelmointinäky-
mässä, Dynamo Playerillä ”Run script” -painikkeella sekä Dynamo-Playerillä ”Edit
inputs” -painikkeella.

Skriptin käyttäminen voi helpoimmillaan olla vain Dynamo Playerin ”Run script” -
painikkeen painaminen. Tällöin kaikki pitää olla valmiiksi valittuna skriptissä ja
skriptin nimen perusteella pitää tietää mitä ohjelma tekee. Tällöin käyttäminen on
äärimmäisen yksinkertaista, mutta kansioon muodostuu paljon erilaisia skriptejä.
Skriptin valinta on tehtävä, kun etsitään juuri sitä oikeaa skriptiä. Lisäksi eri pro-
jekteissa saattaa Familyiden nimet sekä esimerkiksi uppoasennuskoodit vaihtua.
Tällöin on vaikea luoda etukäteen yksilöityjä skriptejä, jotka toimivat kaikissa pro-
jekteissa ilman valintoja.

Tämän opinnäytetyön skripteissä on valittava asioita Dynamo Playerissä, joten
skriptejä käytetään ”Edit inputs” -painikkeella. Kuvassa 8 on esitetty Playerin
”Run script” -painike ja ”Edit inputs” -painike. Dynamo-näkymä voidaan avata
suoraan Revitistä tai Playerin ”Edit in Dynamo” -painikkeesta.



Kuva 8. Dynamo Playerin käyttöpainikkeet

Skriptien määrän vähentämisen suhteen on järkevää luoda yleistettyjä skriptejä,
joissa suunnittelija pystyy itse vaikuttamaan ja valitsemaan asioita. Esimerkiksi

valitsemalla haluamansa Family Typen tai kategorian, muuttamaan suodatettavia parametrejä sekä lisäämään tekstiä.

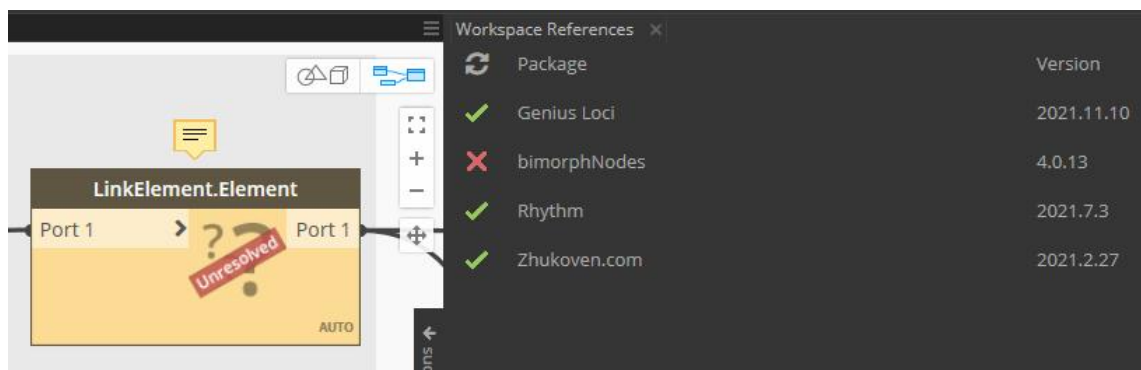
Parhaiten toimivat skriptit, jotka on luotu siten, että ne toimivat kaikissa projekteissa. Jos on muuttuvia tekijöitä, pitää suunnittelijan pystyä itse säätämään sitä. Esimerkiksi tarkistettaessa elementtejä onko ne asennettu samaan korkeuteen kuin alakatto, voidaan elementit ja asennuskoodit hakea saman projektin Schedule-taulukon avulla. Tällöin saataisiin luotua tapa, joka toimii kaikissa projekteissa.

Skriptiin on mahdollista tehdä myös valintapainike, jotta voidaan ensin tarkastella tietoja ja vasta sitten antaa lupa toteuttaa toimenpideoosa. Valintapainike saadaan näkymään myös Dynamo-Playerissä. Esimerkiksi ensin nähdään missä tiloissa valaisimet eivät ole alakattokorossa ja vasta sen jälkeen tehdään päätös, korjataan korkeudet automaattisesti.

Dynamo-Playeriin on viisasta tuoda Watch-solmulla tietoa, mitä ohjelma on tehnyt. Tällöin suunnittelijalle voidaan kertoa, mitä on tapahtunut, missä tilassa on tapahtunut tai kuinka paljon asioita on muuttunut tai luotu. Dynamo Playeriin ei kuitenkaan tule laittaa liikaa erilaisia vaihtoehtoja ja tietoikkunoita tapahtumista, koska se muuttuu sekavaksi, jolloin suunnittelija ei halua käyttää sitä. Järkevänä määränä voidaan pitää vastaavaa kuin Revitin omissa työkaluissakin. Esimerkiksi määrälaskentaa tehdessä on valittava vähintään kolme tietoa, joita ovat Report Template, Range sekä Systems. Lisäksi pitää tietää, mitä Report Template on määritelty sekä osattava muuttaa niitä tarvittaessa. Tällöin voidaan ajatella, että suunnittelija osaa vähintäänkin valita oikean kategorian, Family Typen, kerroksen sekä antaa arvon parametrille.

Skriptin käyttäminen Dynamon ohjelmointinäkyvässä mahdollistaa paljon enemmän tietoa, mutta sillä käyttäminen vaatii myös enemmän koulutusta. Suunnittelijoiden olisi kuitenkin hyvä osata perusasioita Dynamolla ohjelmoinnista, jolloin he itse voisivat tarkentaa skriptiä omien tarpeidensa mukaisesti tai jos skripti ei toimi oikein, niin he osaisivat itse selvittää ongelmaa.

2020 sekä 2021 vuoden Revitin Dynamo Player ei ilmoita puuttuvista kustomoiduista paketeista, joten Playerin avulla on vaikea tietää, jos skripti ei toimi. Kun Dynamo avataan, skripti ilmoittaa puuttuvista kustomoiduista paketeista heti kahdella tapaa. Solmun päällä punaisella pohjalla "Unresolved" -teksti sekä oikeaan reunaan ilmestyvässä ikkunassa punaisella X-merkillä. Kuvassa 9 on esitetty Dynamo-näkymä, jossa ilmoitetaan puuttuvasta paketista. Suunnittelijoita tulee ohjeistaa, että jos skripti ei toimi, niin tällä tavalla voidaan tarkistaa puuttuuko kustomoituja paketteja sekä miten niitä voidaan ladata. Paketteja voidaan etsiä ja ladata Dynamon yläreunan "Packages valikon Search for a Package" -kohdasta. Paketteja voidaan etsiä nimen perusteella ja paketeista voidaan valita eri versioita. Version on oltava yhteensopiva Dynamon vuosiversion kanssa. Vanhemmat paketit toimivat yleensä uudemmissa Dynamon vuosiversioneissa.



Kuva 9. Dynamo ilmoittaa puuttuvasta kustomoidusta paketista

Skriptit, jotka tekevät muutoksia malliin, voidaan muutos kumota Revitin Undo-painikkeella. Toinen vaihtoehto on olla synkronoimatta muutoksia keskusmalliin. Isoissa projekteissa saattaa skriptin ajaminen viedä minutteja aikaa, jos tutkittavia elementtejä on paljon, jolloin myös muutoksen palauttaminen vie vastaavasti aikaa. Isoissa projekteissa voidaan skriptin läpikulkua nopeuttaa esimerkiksi suodattamalla toiminnot kerroksittain, jolloin tutkittavia elementtejä on vähemmän.

4.2 Dynamo-skriptien käyttäminen suunnittelun eri vaiheissa

Suunnittelun eri vaiheissa voidaan käyttää erilaisia skriptejä. Yleissuunnitteluvaiheessa, jossa mallinnetaan pääreitit johtoteille sekä suunnitellaan mallitiloja, ei automaattipiirtoa voida vielä käyttää, koska tietomallit ovat muillakin suunnittelijoilla keskeneräisiä. Tarkistusskriptejä voidaan käyttää ja niiden avulla tutkia onko

elementit mallinnettu alakattoihin tai onko johtoteiden korkeudet alle jonkin sovittun. Elementtien puuttumia voidaan tutkia vain mallitiloista lisäämällä skriptiin suodatuksen, jossa otetaan huomioon vain nimetyt mallitilat.

Yleissuunnitteluvaiheen jälkeen, kun mallitilat ovat tehty, voidaan suorittaa kattava määrälaskenta mallitilojen avulla. Niiden perusteella saadaan tarkempaa tietoa kustannuksista. Projektinjohtourakoissa, tämän toimenpiteen avulla, saadaan pienemmällä vaivalla tarkemmat laskentasuunnitelmat.

Tasopiirustuksien piirustus- ja tulostusnäkyvät tehdään yleissuunnitteluvaiheessa. Jos kohteessa on monta kerrosta tai osaa, piirustus- ja tulostusnäkyvien -skriptillä voidaan nopeuttaa tätä toimenpidettä. Tele- ja turvajärjestelmien tasopiirustusmuotoiset tulosteet tehdään usein vasta toteutussuunnitteluvaiheessa, joita on keskikokoisissakin kohteissa paljon. Piirustus- ja tulostusnäkyvien -skriptillä voidaan nopeuttaa merkittävästi toistoa vaativaa toimenpidettä, jolloin jää enemmän aikaa itse suunnitteluun.

Toteutussuunnitteluvaiheen loppupuolella voidaan käyttää automaattipiirtoa hyödyksi muiden suunnittelijoiden tietomallien avulla. Tämä kannattaa suorittaa vasta, kun suunnittelu on riittävän pitkällä ja mallit eivät enää muutu. Usein kohteissa on sovittu, että arkkitehti- ja LVI-suunnitelmat ovat valmistuttava 2 viikkoa ennen sähköurakan laskentapiirustuksia. Tällöin on sopiva aika käyttää automaattipiirtoa.

Laaduntarkastusskriptejä on suotavaa käyttää jatkuvasti, jolloin voidaan nopeuttaa omaa työtä ja parantaa laatua. Ennen urakkalaskentaa ja toteutussuunnitteluvaiheen loppupuolella projektipäällikkö voivat tehdä kattavia tarkastuksia skriptien avulla nopeasti ja luotettavasti, esimerkiksi selvittämällä puuttuuko tarvittavia elementtejä tiloista.

Ennen tietomallin julkaisua, voidaan Dynamo-skriptien avulla tarkistaa nopeasti, onko alakattoelementit oikeissa koroissa tai johtotiet alakattojen yläpuolella. Elementtien tarkastus alakaton suhteen -skriptiä voidaan täydentää projektikohtaisesti lisäämällä skriptiin kerralla kaikki tarvittavat Family Typet esimerkiksi tieto-

mallivaatimuksien perusteella. Tai tekemällä Revitin Schedule-taulukko, joka listaa alakattoon mallinnettavien elementtien tyypet, jonka perusteella haetaan skriptiin elementit mallista.

4.3 Dynamo-skriptien luominen ja testaaminen

4.3.1 Skriptien luomisen vaiheet

Uuden skriptin luomisessa on ensin selvittävä mitä halutaan tehdä. Sen jälkeen on mietittävä, mitkä ovat lähtötiedot. Tämän jälkeen skriptin eri vaiheet puretaan osiin, joita voi olla todella monta. Näiden toimenpiteiden jälkeen voidaan aloittaa skriptin luominen. Luomisessa voidaan hyödyntää jo aikaisemmin tehtyjä skriptejä, joista on mahdollista kopioita osia tai koko skripti. Skriptin luomisen prosessikaavio on esitetty kappaleessa 2.3.2 Ohjelmallinen ajattelu ja skriptin hahmottaminen.

Skriptiä tulee testata Dynamolla samalla, kun sitä rakennetaan. Tällöin nähdään, tuleeko virheitä sekä käyttäytyykö skripti toivotulla tavalla. On kuitenkin muistettava, että Dynamo tekee Revit-malliin kaikki muutokset, joten skriptin kehittäminen kannattaa tehdä aina testimallilla. Hyödyllinen toiminto testausvaiheessa on myös solmun jäädyttäminen "Freeze" -komennolla, jolloin solmu ei suorita tehtäväänsä. Skriptin eri vaiheiden ratkaisuja parannetaan jatkuvasti, kun tulee parempia ratkaisutapoja mieleen. Tällöin skripti kehittyy koko ajan.

Skriptiä voidaan käyttää Dynamon luomistilassa sekä Dynamo Playerillä. Kun skriptiä rakennetaan, sitä testataan skriptiä luomisnäkyssä Dynamolla. Yleensä vasta lopuksi määritellään sisään- ja ulosmenot Dynamo Playeriin, mikäli halutaan mahdollistaa sellainen käyttöliittymä. Player piirtää aina uuden tuotteen uudella ID-numerolla. Erona luomisnäkyyn on, että luomisnäkyssä tuotteen ID-numero säilyy siihen asti, kunnes Dynamo suljetaan. Tällöin luomisnäkyssä elementin piirrosta voidaan muuttaa asetuksia ilman "Undo" -toimintoa, jolla voidaan Revitissä peruttaa edellinen suoritettu tehtävä.

Lopuksi, kun saadaan toimiva skripti, voidaan kokeilla yleistää sitä. Saadaanko kehitettyä skriptistä, jolla alun perin oli tarkoitus tehdä yksi asia, skripti, jota voidaan käyttää moneen eri tarkoitukseen. Esimerkiksi tehdään aluksi skripti, jolla voidaan mallintaa seinävalaisin peilikaapille, ja yleistetään se toimimaan mille tahansa referenssielementille.

Keskeisimpien skriptien tekeminen dokumentoitiin tarkemmin, joka on esitetty liitteessä 2. Dokumentissa kerrotaan tarkasti skriptin eri vaiheista ja esitellään solmut ja niiden toiminnot. Dokumentoinnin perusteella voidaan luoda samat skriptit uudelleen sekä käyttää niiden ratkaisuja hyödyksi uusien skriptien tekemisissä.

4.3.2 Testattavien kohteiden tiedot

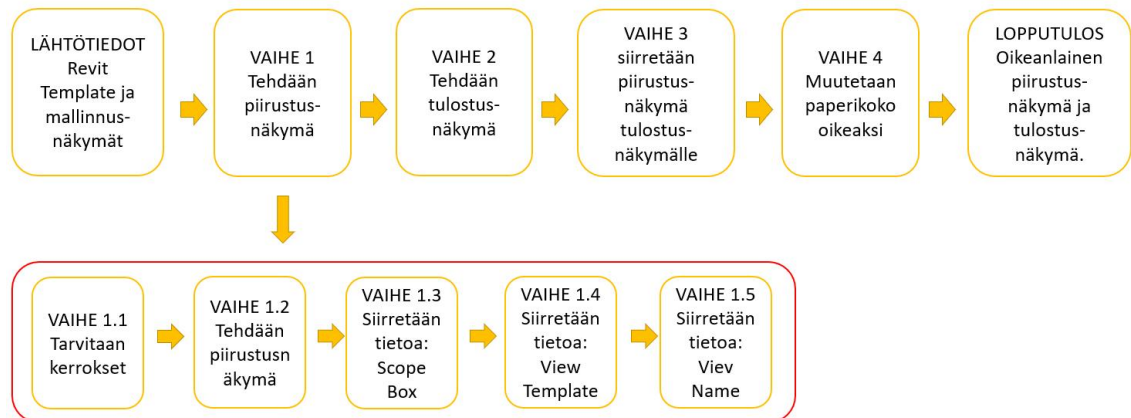
Näkymien automaattinen luominen-, määrälaskenta mallitiloilla- ja puuttuuko tietty elementti tiloista -skriptejä voitiin testata kohteella, jonka arkkitehti suunnittelu oli toteutettu ArchiCADilla. Nämä testaukset tehtiin päiväkodin todellisella suunnittelukohteella, jossa sähkösuunnitelmat olivat toteutettu urakkalaskentavaiheeseen. Urakkalaskenta tasoisilla suunnitelmilla oli myös hyvä vertailla määrälaskennan paikkansa pitävyyttä.

Skriptit, jotka vaativat toimiakseen arkkitehdin tai LVI-suunnittelun Revit-mallia, testattiin MagiCAD-koulutuksen arkkitehdin Revit-tietomallia hyödyntäen. Demoprojektiin oli lisätty Granlundin mallipohja (Template) sekä Granlundin MagiCAD tuotekirjasto (Dataset), jolloin Revit-ohjelma käyttäytyi vastaavasti kuin todellinen kohde. Demoprojektiin oli piirretty valmiiksi eri järjestelmiä LVI-suunnittelusta.

4.3.3 Piirustus- ja tulostusnäkyvien automaattinen luominen

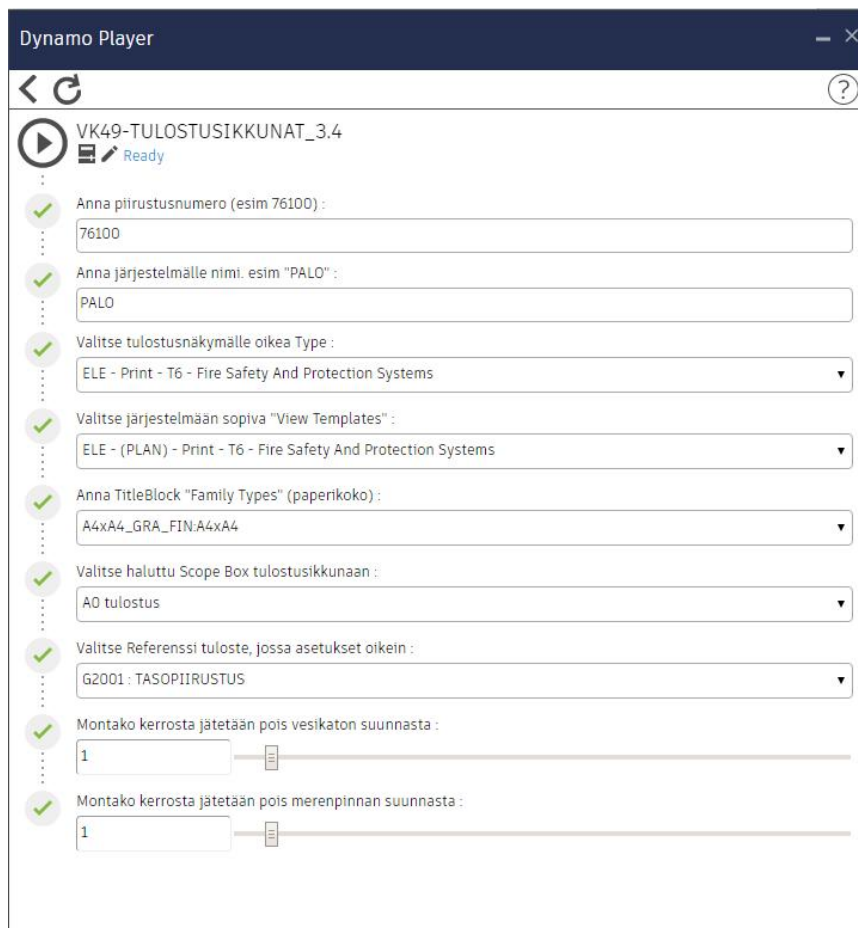
Revit-ohjelmassa on luotava jokaisessa projektissa mallinnus-, piirustus- ja tulostusnäkyvät. Aluksi tehdään mallinnusnäkyvät kaikista kerroksista, joiden avulla luodaan piirustusnäkyvät. Piirustus- ja tulostusnäkyvät on luotava erikseen jo-

kaiselle kerrokselle ja järjestelmälle. Aluksi on tehtävä piirustusnäkyvä, sen jälkeen tulostusnäkyvä, siirrettävä piirustusnäkyvä tulostusnäkyvä ja lopuksi muutetaan paperikoko oikeaksi. Näkymien tekemiseen kuluu aikaa ja sitä voitaisiin helpottaa automatisoimisella. Päätettiin tehdä skripti, joka luo automaattisesti piirustus- ja tulostusnäkyvät. Skripti tekee yhdelle järjestelmälle kerrallaan jokaisesta kerroksesta piirustusnäkyvän ja siirtää sen automaattisesti tulostusnäkyväksi. Skripti tarvitsi toimiakseen yhden valmiiksi tehdyn tulostusnäkyvän, josta se kopioi paperikoon sekä piirustusnäkyvän asettelun. Kehitetty skripti ei kuitenkaan siirtänyt piirustusnäkyvää oikeaan kohtaan, joten se oli vielä lopuksi kohdistettava käsin. Kuviossa 6 on esitetty prosessikaavio skriptin vaiheista. Vaiheesta 1 on esitetty, kuinka se jakautuu vielä pienempiin erillisiin vaiheisiin 1.1 – 1.5.



Kuvio 6. Prosessikaavio piirustus- ja tulostusnäkyvien automaattista luomista varten

Skriptin käyttämisen helpottamiseksi tehtiin Dynamo-Playeriin käyttöliittymä, joka on esitetty kuvassa 10. Näkymien luomista varten Playeriin piti kirjoittaa piirustusnumero ja -nimi, valita järjestelmä, paperityyppi (TitleBlock), tulostusraja (Scope Box), referenssitulostusnäkyvä sekä jätetäänkö kerroksia luomatta merenpinnan tai vesikaton suunnalta. Lopputulos piirustus- ja tulostusnäkyvän automaattisesta luomisesta on esitetty kuvassa 11.



Kuva 10. Dynamo-Playerin näkymä, jolla luodaan valitulle järjestelmälle piirustus- ja tulostusnäkyvät joka kerroksesta



Kuva 11. Revitin uusi tulostusnäkyvä, jossa piirustusnäkyvän sijainti on vielä väärin

Skriptin toimivuutta testattiin demoprojektissa, jossa luotiin seitsemälle eri järjestelmälle piirustus- ja tulostusnäkyvät neljään kerrokseen. Skriptin avulla aikaa kului näkymien luomiseen 3 minuuttia ja 53 sekuntia. MagiCADin koulutuksen mukaisella tavalla aikaa kului 33 minuuttia ja 14 sekuntia. Taulukossa 1 on esitetty vertailutulokset. Skriptillä piirustus- ja tulostusnäkyvien luominen oli yli kahdeksan kertaa nopeampaa verrattuna normaaliin tapaan.

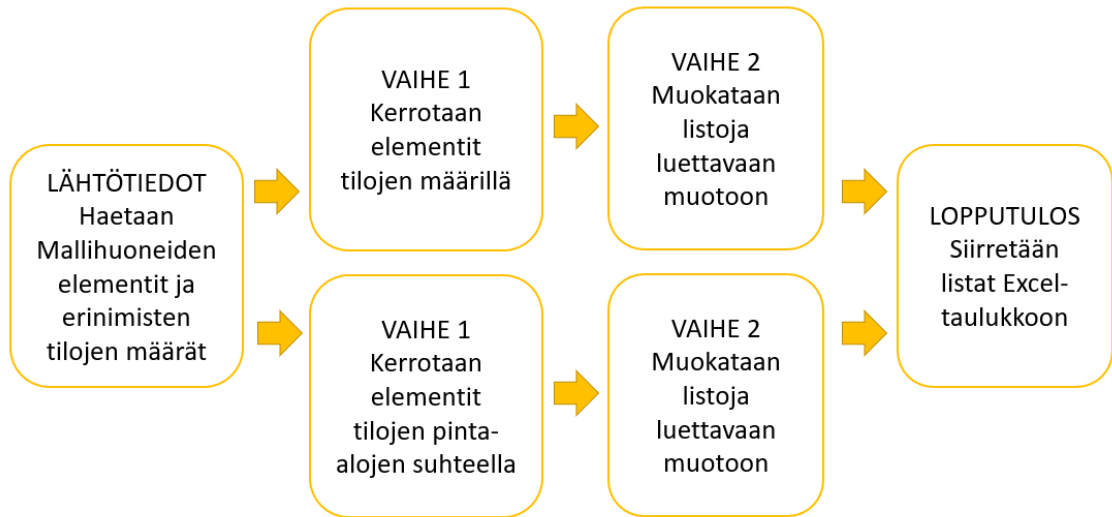
Taulukko 1: Tulostusnäkyvien luomiseen käytetty aika

Tulostusnäkyvien luomiseen käytetty aika				
Teko tapa	Aika	Aika	Näkymien määrä	Aika/Näkymä
	<i>min:sek</i>	<i>sekunti</i>	<i>kpl</i>	<i>sekunti</i>
Skriptillä	03:53	233	27	8,6
Normaalisti	33:14	1994	27	73,9

Skriptin testausvaiheessa ei todettu ongelmia toimivuuden suhteen. Voidaan todeta skripti toimivaksi ja aikaa sääteväksi. Liitteessä 2 on esitetty skriptin tarkempi luominen ja liitteessä 3 on esitetty kuva kokonaisesta Dynamo-skriptistä.

4.3.4 Määrälaskenta mallitilojen avulla

Yksi tutkintokysymyksistä oli selvittää, voidaanko tehdä määrälaskenta varhaisessa suunnitteluvaiheessa. Yleissuunnitteluvaiheessa suunnitellaan sovittujen mallitilojen sähköpisteet tarkasti. Jos suunniteltaisiin kaikkien erinimisten tilojen pisteet kertaalleen, niin niiden tietojen perusteella voitaisiin tehdä kattava määrälaskenta. Päätettiin tehdä skripti, jolla voidaan kertoa mallitilojen pisteiden määrät tilojen määrillä. Skriptiin lisättiin toinen osa, jossa otettiin huomioon pinta-alojen suhde, koska jotkut samannimiset tilat ovat keskenään erikokoisia. Kuviossa 7 on esitetty skriptin prosessikaavio, jota lähdettiin ratkaisemaan vaiheittain.



Kuvio 7. Prosessikaavio määrälaskennan tekemiselle mallihuoneiden avulla

Aluksi rakennettiin skripti toimimaan Revitin Schedule-taulukoiden avulla, koska Schedule-taulukoihin saatiin haettua tilojen pisteet siten, että ne tietävät mihin tilaan kuuluvat. Tilatietoihin (Space) kirjoitettiin, mitkä tilat olivat mallihuoneita. Tiloista muodostettiin toinen Schedule-taulukko, jossa taulukko laski samannimisten tilojen määrät. Pinta-alojen suhteen laskemista varten muodostettiin kolmas Schedule-taulukko, jossa tiloilla näkyivät pinta-alat. Skriptissä listojen käsittelyssä käytettiin Python-koodia, koska sen avulla voitiin hallita listoja helpommin ehtolauseilla ja erilaisilla silmukoilla. Taulukossa 2 on esitetty Schedule-taulukko, jossa on esitetty pistemäisten elementtien tiedot.

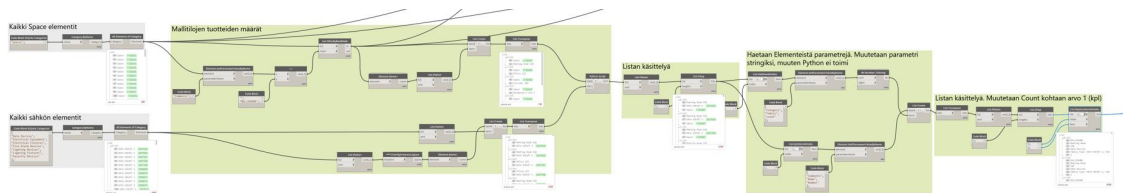
Taulukko 2: Revitin Schedule-taulukko mallitilojen elementtien määristä

<massa_mallihuoneet>					
A	B	C	D	E	F
Space: Comments	Space: Name	Space: Number	Category	Family	Count
MALLIHUONE	KÄYTÄVÄ	117	Communication De	Ovikello	1
MALLIHUONE	VAATENAUL	111	Communication De	Ovikello	1
MALLIHUONE	KÄYTÄVÄ	117	Communication De	Summeri	1
MALLIHUONE	KÄYTÄVÄ	117	Data Devices	Data socket-2	1
MALLIHUONE	LEIKKIHUONE	136	Data Devices	Data socket-2	1
MALLIHUONE	LEPOHUONE	126	Data Devices	Data socket-2	1
MALLIHUONE	NEUVOTTELU	104	Data Devices	Data socket-2	1
MALLIHUONE	SÄH	127	Data Devices	Data socket-2	1
MALLIHUONE	TAUKO	107	Data Devices	Data socket-2	1
MALLIHUONE	TYÖHUONE	105	Data Devices	Data socket-2	1
MALLIHUONE	ET	103	Data Devices	HS - EMERGENCY-STOP-BUTTON	1
MALLIHUONE	TAUKO	107	Data Devices	HS - LAP 1	1
MALLIHUONE	ET	103	Data Devices	KV-OVI	1
MALLIHUONE	KURAEITEINEN	142	Data Devices	KV-OVI	1
MALLIHUONE	ET	103	Data Devices	TE - GRA_RAU_TE	1
MALLIHUONE	KEITTIÖ	121	Data Devices	TE - GRA_RAU_TE	1

Johtotiet kuuluvat System Familyihin, jolloin niille ei tule samanlailla tilatietoa kuin pistemäisille Familyille. Tämä ratkaistiin solmulla, jolla saatiin luokiteltua johtotie

tilaan kuuluvaksi sen insertiopisteen perusteella. Aluksi tilatiedot siirrettiin johtotelementin vapaille parametreille, jotta pysyttiin luomaan Schedule-taulukko, jossa johtoteille oli määritelty mihin tilaan kuuluivat. Johtotien pituus määräytyi elementin kokonaispituuden perusteella, eikä sen perusteella kuinka pitkän matkan se kulki tilassa. Tällöin Johtotien mallintamisessa oli huomioitava, että hylly oli katkaistava mallitilojen reunojen kohdalta, jotta pituus määrittyi oikein. Tämän jälkeen määrälaskenta voitiin suorittaa johtoteille vastaavasti kuin pistemäisille elementeille, mutta Python-koodia ja listojen käsittelyä piti muokata tarvittavilta osilta.

Lopuksi skriptistä poistettiin Schedule-taulukot kehittämällä samat asiat Dynamo-ohjelmoinnilla, koska Schedule-taulukko oli määriteltävä hyvin tarkasti, jotta skripti toimisi oikein. Kuvassa 12 on esitetty, miten monimutkaisella Dynamo-ohjelmoinnin osalla voitiin korvata Schedule-taulukko, jossa oli esitetty mallitilojen pistemäisten elementtien määrät.



Kuva 12. Dynamo-skriptin osa, jonka tehtävä oli korvata yksi Schedule-taulukko, jossa oli koottu mallitilojen pistemäisten elementtien määrät

Määrälaskenta mallitiloilla -skripti laski määrät vain tiloista, joiden tilatiedoille on määritelty parametri "MALLIHUONE". Jotta määrälaskenta antoi oikean tuloksen, oli kaikkia erinimisiä tiloja suunniteltava yksi. Tämän varmistamiseksi skriptiin lisättiin osa, jossa ohjelma tekee listan tilojen nimistä, joista ei ole yhtään nimetty mallihuoneeksi tai oli nimetty useampi kuin yksi.

Saadut määrälaskennan listat siirrettiin Excel-taulukkoon. Excel-taulukkoon siirrettiin lista, jossa samannimisten tuotteiden määrät olivat valmiiksi yhteenlaskettu, sekä toinen lista, jossa määrät olivat laskettu tiloittain. Tiloittain lasketuille määritellä tehtiin Excel-taulukkoon Pivot-taulukko, jonka avulla oli mahdollista tarkistaa kokonaismääriä tiloittain. Pivot-taulukon oli tarkoitus toimia osana laadun- tarkastelua. Excel-taulukkoon vietiin yhteensä yhdeksän eri laskelmaa. Excel-

taulukon mallipohjaan tehtiin lisäksi valmiit Pivot-taulukot tarkastuksia varten. Excel-taulukon ymmärtämistä varten luotiin skriptiin osa, joka tekee Exceliin sisällysluettelon, jossa kerrotaan sivujen tarkoituksista. Sisällysluettelo on esitetty taulukossa 3.

Taulukko 3. Määrälaskenta mallitiloilla -skriptin Excel-taulukon sisällysluettelo, jossa kerrotaan mitä tietoja on Excel-taulukon eri sivuille siirretty

sisällysluettelo	
110	pisteiden määrät suhteessa tilojen määrään
120	pisteiden määrät suhteessa pinta-alojen määrään
130	johtoteiden määrät suhteessa tilojen määrään
140	johtoteiden määrät suhteessa pinta-alojen määrään
210	pisteiden määrät tiloittain suhteessa tilojen määrään
211	pivot-taulukko
220	pisteiden määrät tiloittain suhteessa pinta-alojen määrään
221	pivot-taulukko
230	johtoteiden määrät tiloittain suhteessa tilojen määrään
231	pivot-taulukko
240	johtoteiden määrät tiloittain suhteessa pinta-alojen määrään
241	pivot-taulukko
300	Puuttuvat mallitilat ja kahteen kertaan

Skriptin toimivuutta testattiin päiväkodin projektissa. Vertailun vuoksi suoritettiin ensin määrälaskenta urakkalaskentamallista, jotta voitiin vertailla arvoja todellisiin määriin. Mallista tehtiin kopio, johon nimettiin mallihuoneet tilatietoihin, ja poistetaan muista tiloista kaikki elementit. Tämän jälkeen suoritettiin määrälaskenta skriptin avulla. Vertailu tehtiin ensimmäisen kerroksen tiloihin.

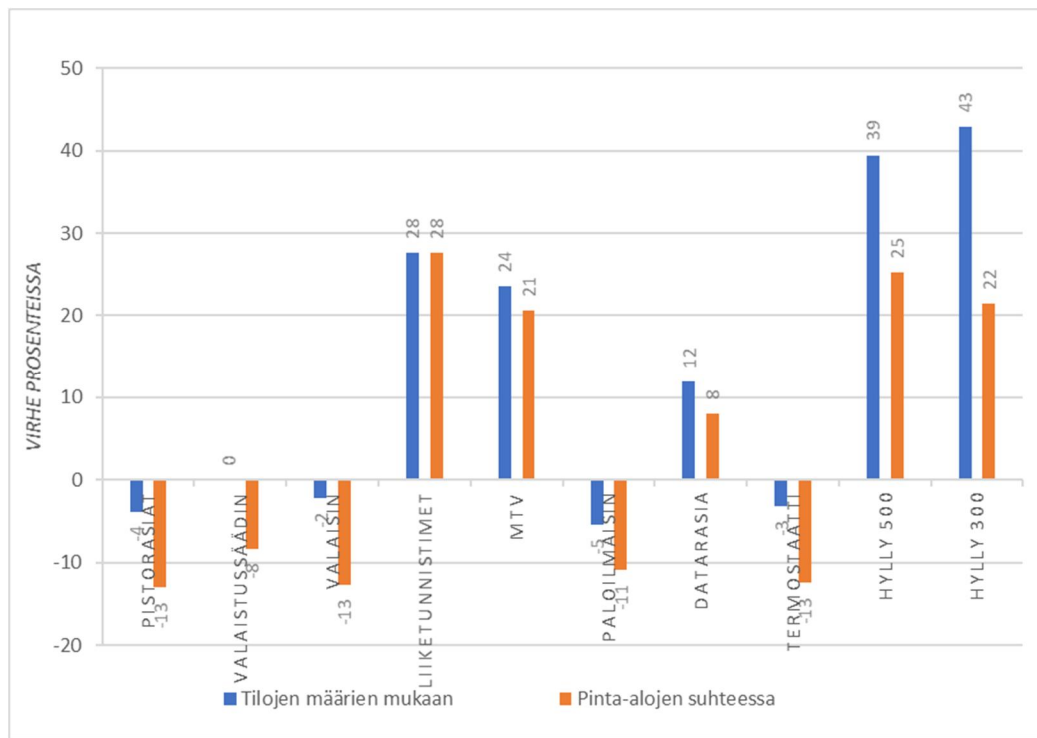
Skriptin kokeilun ensimmäisellä kerralla huomattiin virheitä skriptin tekemisessä määrälaskelmissa. Kaikki johtotiet eivät tulleet mukaan määrälaskentaan, koska arkkitehdin tilaelementin yläreuna oli sama kuin alakaton alareuna, tällöin alakaton yläpuoliset elementit eivät tulleet mukaan laskelmiin. Joidenkin tilojen termostaatit eivät tulleet mukaan määrälaskentaan, koska termostaatin 3D-elementti mallintui erisuuntaan kuin 2D-elementti, jolloin 3D-elementin insertiopiste oli tilaelementin ulkopuolella. Nämä virheet korjattiin ja suoritettiin uudelleen määrälaskenta skriptillä.

Valittiin määrälaskennoista kymmenen erilaista vertailuryhmää, joissa oli laskettu saman tyyppisten tuotteiden määrät, esimerkiksi oli laskettu yhteen kaikki pistorasiat. Vertailuryhmien määriä verrattiin keskenään ja tutkittiin miten ne eroavat toisistaan. Taulukossa 4 on esitetty valitut vertailuryhmät ja niiden määrät eri laskentatavoilla.

Taulukko 4. Määrälaskennan tuloksia eri tavoilla tehtynä

	Määrälaskenta mallitiloilla		Oikea malli
	Tilojen määrien mukaan	Pinta-alojen suhteessa	todellinen kohde
pistorasiat	221	200	230
valaistussäädin	12	11	12
Valaisin	223	199	228
Liiketunnistimet	37	37	29
MTV	42	41	34
paloilmaisoin	87	82	92
Datarasia	28	27	25
Termostaatti	31	28	32
Hylly 500	101	91	73
Hylly 300	136	115	95

Määrälaskentojen antamia tuloksia vertailtiin tekemällä kuvio 8, jossa oli esitetty poikkeamat prosentteina suhteessa todellisen mallin antamiin määriin. Suurimmat poikkeavuudet pistemäisillä elementeillä olivat liiketunnistimilla ja merkki- ja turvavaloilla, joissa ero oli 21-28%. Johtoteillä erot kasvoivat 43% asti. Poikkeamat olivat lähellä toisiaan riippumatta siitä, oliko määrälaskenta mallitiloilla tehty pinta-alojen suhteessa vai tilojen määrien suhteen. Suurin poikkeama oli 300mm leveät sähköhyllyt, jotka oli laskettu tilojen kappalemäärien perusteella.

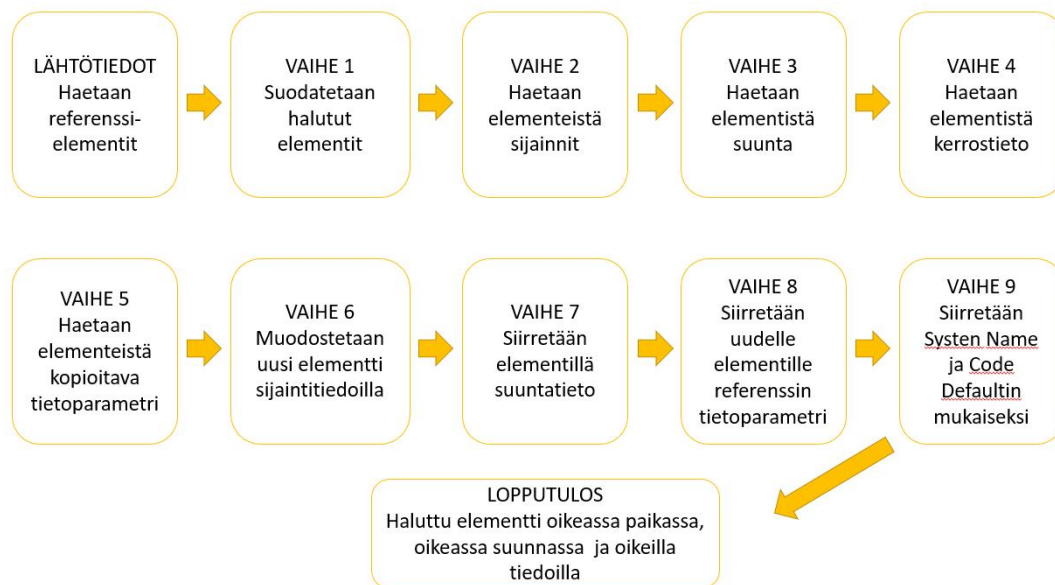


Kuvio 8. Skriptin avulla tehdyn määrälaskennan eroavaisuudet esitettynä prosentteina suhteessa todelliseen kohteeseen.

Todellisen mallin määrälaskennassa oli mukana myös rakennuksen ulkopuoliset asennukset, joita olivat seinävalaisimet, paloilmalaisimet ja pistorasiat. Tämän takia valaisimia, paloilmalaisimia ja pistorasioita oli vähemmän skriptillä toteutetuissa määrälaskennoissa. Johtoteiden ero selitti se, että mallihuoneeksi valittu käytävä oli kapea, jolloin johtoteiden määrä suhteessa pinta-alaan kohti oli suurempi sekä mallitilaksi valitun WC:n ylitse kulki sähköhyllä, mutta muissa WC-tiloissa ei ollut sähköhyllä. Tästä voitiin todeta, että määrälaskennan tuloksiin vaikuttaa mallihuoneiden valinta, sekä tilaelementtien ulkopuoliset asennukset eivät tule mukaan mallihuoneilla tehtävään laskelmaan. Voitiin todeta skripti toimivaksi, mutta useita asioita on otettava huomioon laadun varmistamiseksi. Liitteessä 2 on esitetty skriptin tarkempi luominen ja liitteessä 4 on esitetty kuva kokonaisesta Dynamo-skriptistä.

4.3.5 Elementtien automaattinen mallintaminen muiden suunnittelualojen tietomallien perusteella

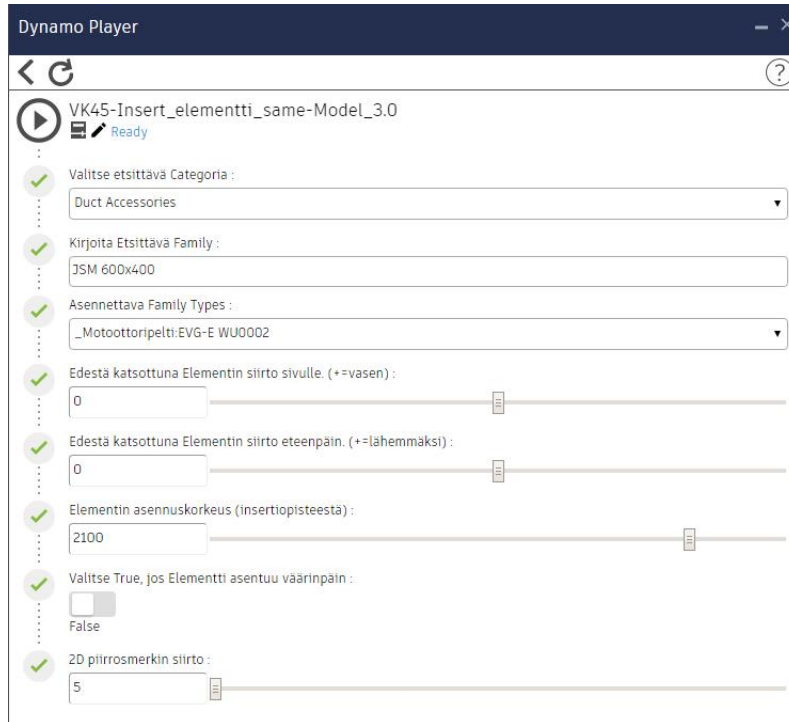
Revit-ohjelmassa, kun piirretään sähkön elementti malliin, valitaan sopiva tuote (Family Type), annetaan sille korkeus ja valitaan sijainti 2D-näkymässä. Tällöin elementti saa oikein sijainnin 3D-näkymässä. Joillekin elementeille kirjoitetaan lisäksi tietoa vapaille parametreilla, joista voidaan muodostaa TAGin avulla tekstitieto näkymään elementin tunnistamista varten. Elementtien automaattisen mallintamisen tarkoitus oli piirtää elementti automaattisesti oikeaan 3D-sijaintiin ja kopioida sille tarvittavia parametreja referenssielementistä. Referenssielementillä tarkoitetaan tässä toisen suunnittelualan tietomallin elementtiä. Tehtiin skripti, jonka avulla voitiin piirtää elementti oikeaan sijaintiin automaattisesti ja kopioida haluttu parametri. Aluksi skripti ratkaistiin toimimaan arkkitehdin peilikaapin suhteen, joka yleistettiin toimimaan kaikkien elementtien kanssa. Kuviossa 9 on esitetty prosessikaavio skriptin tekemisestä.



Kuvio 9. Elementtien automaattisen mallintamisen -skriptin prosessikaavio

Skriptin käyttämisen helpottamiseksi tehtiin Dynamo-Playeriin käyttöliittymä, joka on esitetty kuvassa 13. Referenssielementin määrittelyä varten oli valittava kategoria ja kirjoitettava halutun elementin Family-nimi. Tämän jälkeen voitiin valita piirrettävä elementti ja antaa halutut siirtymät referenssielementin insertiopisteestä edestäpäin katsottuna. Esimerkiksi arkkitehdin peilikaapin insertiopiste oli

lattian korkeudella, seinässä kiinni ja sivusuunnassa keskellä. Joissakin tapauksissa elementti piirtyi väärin päin, jonka takia skriptiin tehtiin osa, jolla voidaan kääntää mallinnettava elementti ympäri. Lopuksi voitiin antaa arvo, jolla siirretään 2D-elementtiä erilleen 3D-elementistä. Tämän jälkeen skripti mallinsi automaattisesti kaikille samannimisille elementeillä sähkön piirrosmerkin oikein.



Kuva 13. Dynamo-Playerin näkymä skriptistä, joka piirtää elementin automaattisesti referenssielementin perusteella

Tästä skriptistä muokattiin toinen versio, jolla voitiin piirtää elementtejä LVI-suunnittelijan Revit-mallin perusteella. Skripti toimi vastaavasti, mutta suurin ero oli, että LVI-suunnittelija piirsivät elementit samaan malliin sähkösuunnittelijan kanssa.

Skriptien toimivuutta testattiin demoprojektissa, koska skriptit vaativat toimiakseen Revit-ohjelmalla tehtyjä muiden alojen tietomalleja. Testauksessa mallinnettiin LVI-suunnittelun ilmansäätöpelleille ja palopelleille sähkösuunnitelmiin moottoripellin elementti ja kopioitiin halutut parametrit, joiden avulla saatiin yksilöllinen tieto elementille. Aluksi piti tutkia mallista sopivat parametrit, joiden avulla löydetään kaikki referenssi elementit sekä mitä tietoja halutaan kopioida. Mallin tutkimiseen ja skriptin muokkaamiseen kului 20 minuuttia aikaa. Automaattipiirtoon kului aikaa vain kaksi minuuttia. Lisäksi tarkistettiin piirrettyjä elementtejä ja

siihen kului aikaa neljä minuuttia. Tällöin automaattipiirroilla kului yhteensä aikaa 26 minuuttia ja saatiin piirrettyä 20 elementtiä. Tehtiin normaalilla suunnittelutavalla sama työ ja aikaa kului 21 minuuttia.

Yhteenvetona testauksesta todettiin, että molemmissa tapauksissa oli selvitetävää, miten elementit löytyvät ja mitä tietoja haluttiin kirjoittaa piirrettävälle elementille. Automatisoinnissa piti määritellä kertaalleen oikeat parametrit skriptiin, mutta käsin piirtäessä tiedot oli tarkistettava joka kerta. Skriptin muokkaamiseen ja testaamiseen kului ylimääräistä aikaa, mutta jos suunnittelijalla ei ole Dynamo-osaamista, olisi hänen pyydettävä apua Dynamo-osaajalta. Automaattipiirrosta myös huomattiin, että osa moottoripellin elementeistä oli piirtynyt valaisimen kohdalle. Näiden elementtien sijainnit piti lopuksi siirtää käsin parempaan kohtaan. Voitiin todeta, että pienissä kohteissa automaattipiirtoa ei kannata käyttää, jos skripti ei ole valmiiksi yhteensopiva. Granlundilla nämä asiat ovat kuitenkin aina samanlailla uusissa projekteissa, joten sellaisessa tapauksessa tällä skriptillä olisi voinut säästää aikaa. Automaattipiirron etuna on, että se täyttää parametri-tiedot aina oikein. Käsin kirjoittaessa on mahdollista, että tulee kirjoitusvirheitä. Liitteessä 2 on esitetty skriptin tarkempi luominen ja liitteessä 5 on esitetty kuva kokonaisesta Dynamo-skriptistä

Samalla testattiin skriptin eri vuosiversioiden yhteensopivuutta sillä, että skripti oli luotu 2020-versiolla ja skriptiä kokeiltiin 2021-versiolla. Dynamo ilmoitti selkeästi mitkä kustomoidut paketit puuttuivat. Niiden lataaminen onnistui muutamassa minuutissa. Muuten skripti toimi oletetusti.

4.3.6 Elementtien automaattinen piirtäminen alakattoihin

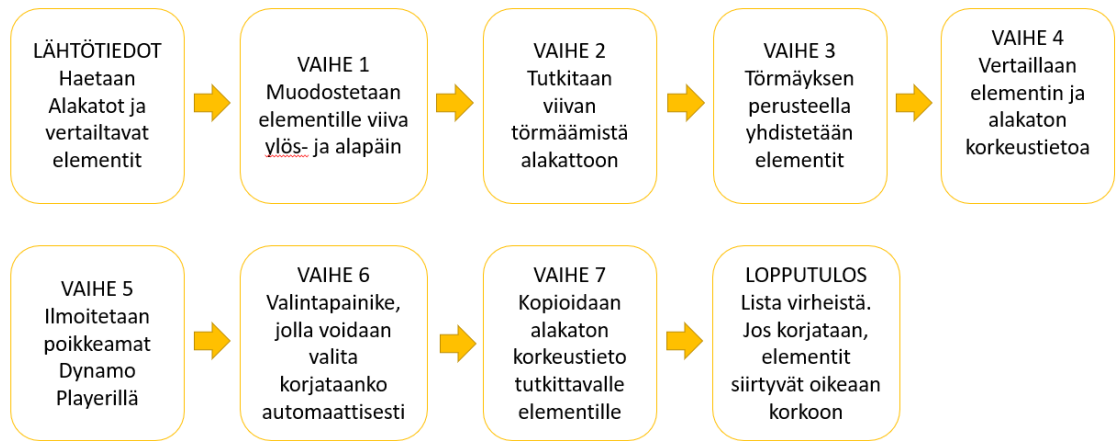
Tehtiin skripti, jonka avulla voitiin piirtää automaattisesti elementtejä alakattoon halutun nimisiin tiloihin. Elementtien sijoittelussa käytettiin matemaattista kaavaa, joka laski, kuinka monta elementtiä tilaan mahtuu. Elementtien keskinäinen etäisyys annettiin lähtötiedoiksi. Esimerkiksi voitiin piirtää valaisimet 1800mm jaolla tilaan ja ohjelma laski, kuinka monta valaisinta sinne mahtuu näillä ehdoilla. Elementtien asennuskorkeudeksi tuli alakaton korkeus.

Muokattiin skriptistä toinen versio, jolla voitiin piirtää erilaisia elementtejä alakattoon. Skriptille annettiin siirtymä alakaton keskipisteestä X- ja Y- suunnassa, elementtien määrä, elementtien välinen etäisyys, sekä haluttu asennuskulma. Tämän skriptin avulla oli mahdollista piirtää esimerkiksi paloilmaisin tilaan ja antaa siirtymä pois keskipisteestä.

Päädyttiin lopettamaan skriptin tekeminen, koska elementtien sijoittelua huoneeseen on äärimmäisen monimutkainen operaatio, jos halutaan ottaa huomioon kaikki tarpeelliset asiat. Tultiin siihen lopputulokseen, että pisteiden sijainnit katossa on kuitenkin tarkistettava ja yhteensovittettava uudelleen. Tällöin tarkistamiseen ja yhteensovitukseen saattaisi kulua jopa enemmän aikaa kuin kertaalleen piirto normaalilla tavalla.

4.3.7 Elementtien asennuskorkeuden tarkastaminen ja muuttaminen alakaton perusteella

Revit-ohjelmassa on annettava piirrettävälle elementille korkeus. Suunnitteluvaiheessa usein alakattokorkeudet eivät ole vielä lopullisia tai ne puuttuvat kokonaan. Tällöin suunnitelmissa esiintyy alakattoon asennettavia elementtejä, jotka eivät ole oikealla korkeudella. Päätettiin tehdä skripti, jolla voidaan selvittää, mitkä alakattoon tarkoitetut elementit eivät ole oikealla korkeudella. Skripti tutki jokaisen elementin kohdalta, onko sen ylä- tai alapuolella alakatto. Jos skripti löysi alakaton, se vertaili, oliko asennuskorkeus sama kuin elementillä. Väärällä korkeudella olevat elementit listattiin ja esitettiin Dynamo-Playerissä. Samalla Player ilmoitti tilat, joista ei voitu tutkia elementtejä sekä elementit, joiden kohdalla ei ollut alakattoa. Tilanteissa, joissa on kaksi alakattoa päällekkäin tai on viisto alakatto, ovat tiloja, joita skripti ei osannut tutkia. Skriptiin tehtiin valintapainike, jolla annettiin skriptille lupa muuttaa elementtien korkeus samaksi kuin sen löytämä alakatto. Samalla kuin skripti korjasi korkeuden se kirjoitti elementin Comments-parametrille tekstin ”korjattu korkeus automaattisesti”. Kuviossa 10 on esitetty prosessikaavio skriptin tekemisestä.



Kuviossa 10: Elementtien asennuskorkeuden tarkastaminen ja muuttaminen alakaton perusteella -skriptin prosessikaavio

Skriptin käyttämisen helpottamiseksi tehtiin Dynamo-Playeriin käyttöliittymä, joka on esitetty kuvassa 14. Tässä versiossa voitiin valita tutkittavat tuotteet kategorian mukaan ja määritellä alakattoon asennettavat tuotteet asennuskoodin perusteella. Dynamo-Playerilla oli valittava arkkitehdin tietomalli, tutkittava kerros, kategorian ja uppoasennuskoodi sekä vertailu tarkkuus.

Dynamo Player - X

< ↺ ?

VK1-Tarksitä-Categoria_onko_alakatossa_Solid_4.7
Run completed with errors

✓ Valitse arkkitehdin Revit-malli, jossa Ceilings (alakatot) :
 ARK - Yleisnimet.rvt : 6 : location <Not Shared>

✓ Valitse tutkittava Level (kerros) :
 2. kerros

✓ Categories :
 Lighting Fixtures

✓ Anna vertailutarkkuus (mm) (alakatto vs elementti) :
 0

✓ Korjataanko elementin korkeus alakaton mukaiseksi? :
 True

✓ Kirjoita oppoasennuksen koodi (MC Installation Code Type) :
 U

1. Näissä tiloissa tutkittavat elementit ovat väärässä korkeudessa :
 List ▼
 0 Tilassa 216 on 1 elementtiä väärässä korkeudessa
 1 Tilassa 218 on 4 elementtiä väärässä korkeudessa
 2 Tilassa 226 on 1 elementtiä väärässä korkeudessa
 3 Tilassa 228 on 3 elementtiä väärässä korkeudessa

2. Näissä tiloissa tutkittavien elementtien asennuskorkeus korjattu automaattisesti :
 List ▼
 0 Tilassa 216 muutettu 1 elementtiä
 1 Tilassa 218 muutettu 4 elementtiä
 2 Tilassa 226 muutettu 1 elementtiä
 3 Tilassa 228 muutettu 3 elementtiä

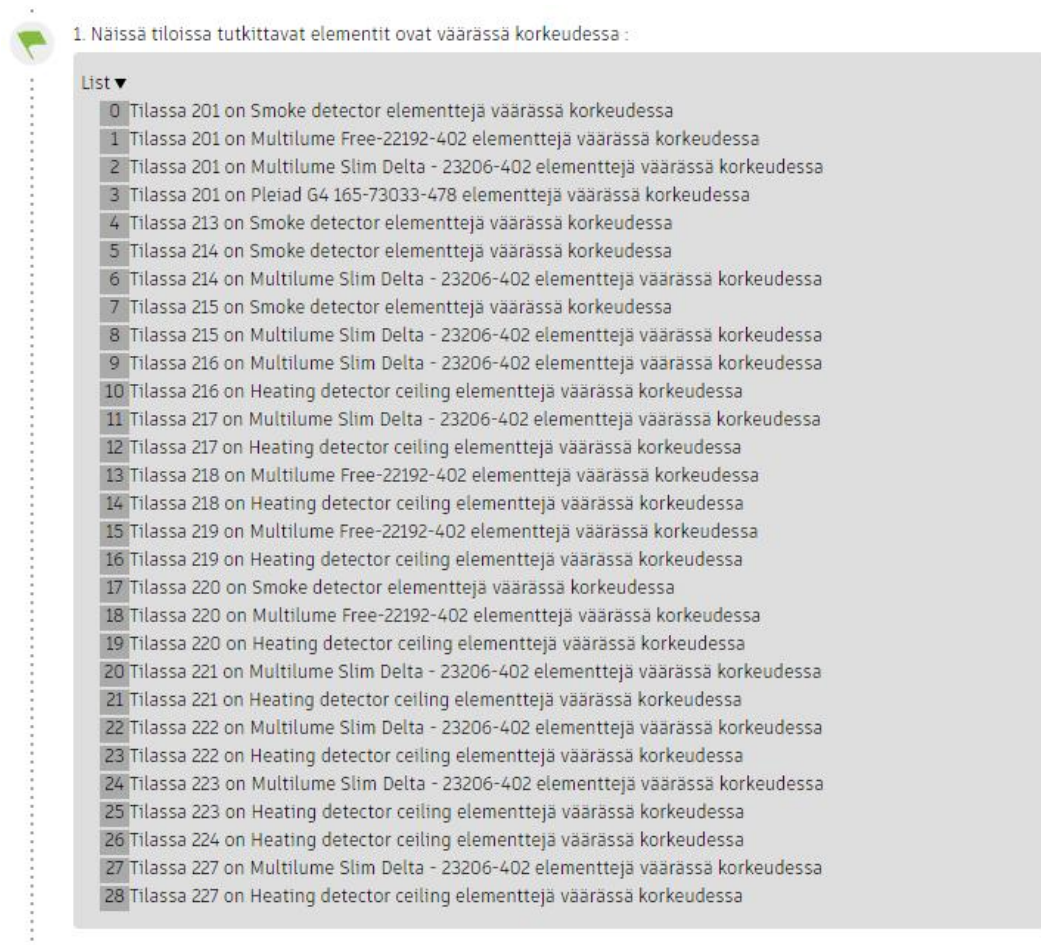
3. Näistä tiloista ei voida tutkia elementtejä. Elementti löytää monta alakattoa samalta kohtaa :
 List ▼
 0 Tilassa 217 ei voida tutkia kaikkia elementtejä

4. Näissä tiloissa on tutkittavia elementtejä, mutta ei alakattoa :
 List ▼
 0 Tilassa 202 on 2 elementtiä, joilta ei löydy alakattoa

Viistoja alakattoja, ei voida tutkia :
 List ▼
 0 Tilassa 215 on viisto alakatto. Ei voida tutkia

Kuva 14. Dynamo-Playerin näkymä, jossa vertaillaan ja korjataan elementin korkeudet alakaton korkeuden mukaisesti

Skriptistä muodostettiin useampi versio. Yhdessä versiossa haettiin Family Ty-
pen perusteella elementit. Tällöin voitiin yksilöidä tarkemmin tutkittavat tuotteet.
Voitiin etsiä esimerkiksi, onko savunpoistoilmalaitteet tai uppokaiuttimet asennettu
alakattoon. Lopuksi kehitettiin skriptiä siten, että saatiin haettua kerralla kaikki
alakattoon asennettavat elementit. Revitin Schedule-taulukon avulla, listattiin eri-
tyyppiset tuotteet, joita voitiin vertailla skriptillä. Skriptin loppuosan Watch-solmua
muutettiin siten, että se ilmoitti myös minkä tyyppinen elementti oli väärällä kor-
keudella. Kuvassa 15 on esitetty Dynamo-Playerin Watch-solmun tietokenttä.



Kuva 15. Dynamo Playerin Watch-solmun tietokenttä elementeistä, jotka eivät ole alakaton korkeudella

Skriptien toimivuutta testattiin demoprojektissa, jossa siirrettiin 2.kerroksen ala-
kattoon asennettavat elementit tarkoituksella väärälle korkeudelle. Lisäksi arkkiteh-
din malliin tehtiin alakattoja päällekkäin, viistottain sekä monimuotoisia alakat-
toja eri korkeuksille, jotta voitiin löytää mahdollisia virheitä. Skripti löysi elementit,
jotka olivat väärällä korkeudella ja korjasi niiden asennuskorkeudet automaattisesti
oikein. Alakattoon kuulumattomat elementit pysyivät muuttumattomina.

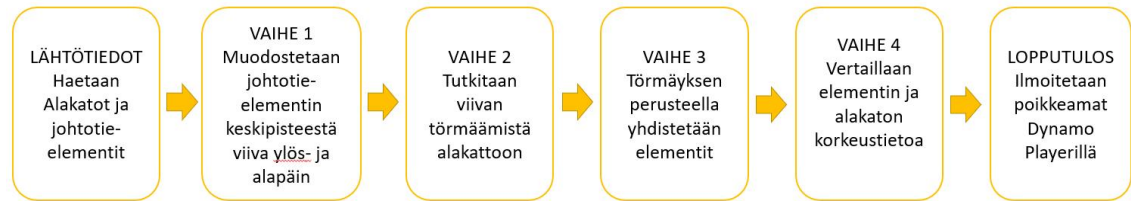
Skriptin testausvaiheessa ei todettu ongelmia toimivuuden suhteen. Tilat, joita skripti ei osaa tutkia, on kuitenkin melko harvinaisia. Tällöin voidaan todeta, että skripti nopeuttaa suunnittelun tarkistusta ja parantaa laatua. Liitteessä 2 on esitetty skriptin tarkempi luominen ja liitteessä 6 on esitetty kuva kokonaisesta Dynamo-skriptistä.

Tämän skriptin perusteella muodostettiin toinen skripti, jolla voitiin kirjoittaa alakaton yläpuolisille elementeille tekstiä. Skriptissä vertailtiin tutkittavien elementtien ja alakaton korkeuksia ja alakaton yläpuolisille elementeille pystyi kirjoittamaan tekstiä haluamalleen parametrille. Tällä saatiin ratkaistua kyselytutkimuksessa ehdotettu automatisointi, jolla haluttiin kirjoittaa alakaton yläpuolisille elementeille teksti "AY".

4.3.8 Johtoteiden asennuskorkeuden tarkastaminen alakaton perusteella

Johtoteitä mallintaessa on Revit-ohjelmassa annettava niille korkeus. Usein johtotiet mallinnetaan alakaton yläpuolelle tikashyllyinä. Suunnittelun alkuvaiheessa olevat alakatot eivät kuitenkaan ole vielä lopullisilla korkeuksilla, vaan ne tarkentuvat suunnittelun edetessä. Johtoteiden korkeuksien tarkastuksia tehdään yleensä erillisillä tietomalliohjelmilla kuten Solibrilla ja Navisworksilla. Erillisten ohjelmien käyttämiseen kuluu kuitenkin ylimääräistä aikaa. Päätettiin tehdä skripti, jolla voidaan selvittää, oliko sähköhyllyjä mallinnettu alakaton alapuolelle. Muodostettiin skriptille prosessikaavio, joka on esitetty kuviossa 11. Skripti tutki johtotien asennuskohtaa sen insertiopisteen perusteella ja vertaili, oliko sen ylä- tai alapuolella alakatto. Jos elementti alakaton, se vertaili, oliko alakatto sen alapuolella tai liian lähellä sitä. Väärällä korkeudella olevat elementit listattiin ja esitettiin Dynamo-Playerissä. Samalla Player ilmoitti tilat, joista ei voitu tutkia elementtejä sekä elementit, joiden kohdalla ei ollut alakattoa. Tilanteissa, joissa on kaksi alakattoa päällekkäin tai on viisto alakatto, ovat tiloja, joita ei voida tutkia. Skriptiin tehtiin suodatuksia, joiden avulla voitiin lajitella johtokanavat ja pystyhyllyt pois vertailusta. Tutkittavien johtoteiden Comments-parametrille kirjoitettiin arvo "alakaton alapuolella", jos se ei ollut tarpeeksi korkealla. Johtotiet, jotka olivat tarpeeksi korkealla, nollattiin Comments-parametrin arvo kirjoittamalla siihen tyhjä teksti. Skriptin käyttämisen helpottamiseksi tehtiin

Dynamo-Playeriin käyttöliittymä, joka ilmoitti huonenumerot ja määrät tiloista, joissa johtotiet olivat liian matalalla.



Kuviossa 11. Johtoteiden asennuskorkeuden tarkastaminen alakaton korkeuden suhteen -skriptin prosessikaavio

Skriptien toimivuutta testattiin demoprojektissa samalla kuin pistemäisten elementtien asennuskorkeuksien tarkistusta. Tiloihin piirrettiin sähköhylyjä alakaton alapuolelle sekä eri pituisia pystyhylyjä. Skripti löysi sähköhylyt, jotka olivat väärällä korkeudella. Comments-parametrien avulla tehtiin suodatus Revin näkymien hallinnassa ja saatiin muutettua alakaton alapuoliset sähköhylyt punaisiksi. Skriptin testausvaiheessa ei todettu ongelmia toimivuuden suhteen, kun tiedostettiin, että johtoteitä tutkittiin vain niiden insertiopisteiden perusteella. Voidaan todeta, että skripti nopeutti tarkastuksia ja parantaa laatua. Liitteessä 7 on esitetty kuva kokonaisesta Dynamo-skriptistä.

4.3.9 Puuttuvien elementtien listaaminen automaattisesti

Isojen kohteiden suunnittelussa on haastavaa selvittää, puuttuuko jostain tilasta jokin elementti. Sähkösuunnitelmissa on paljon piirrosmerkkejä, jolloin yksittäisten elementtien huomaaminen on haastavaa. Päätettiin tehdä skripti, jolla voitiin selvittää tilat, joista puuttui halutun tyyppinen elementti. Skripti haki kaikki tilat ja luokitteli tutkittavat elementit niihin. Elementtien vertailuille tehtiin suodatuksia ja matemaattisia kaavoja, joiden avulla voitiin tarkemmin määritellä puutteita. Skripti käytti listojen vertailussa Python-koodia, koska sillä voitiin tehdä paremmin ehtolauseita ja silmukoita. Seuraavia suodatuksia ja matemaattisia ehtoja sisällytettiin skripteihin:

- Suodatettiin kerroksen (Level) mukaan elementit
- Suodatettiin tilan nimen (Space Name) mukaan elementit
- Tehtiin numeroliukuri, jolla voidaan muuttaa määrää, montako tuotetta on löydyttävä tilasta
- Tehtiin matemaattinen kaava, jonka avulla tutkitaan, onko riittävän monta elementtiä tilan pinta-alaa kohden.
- Tehtiin matemaattinen kaava, jonka avulla tutkitaan, onko tilan valaisimien kokonaisteho pinta-alan suhteen riittävän suuri.

Muodostettiin skriptistä neljä erilaista versiota. Skriptin käyttämisen helpottamiseksi tehtiin Dynamo-Playeriin käyttöliittymä, jossa pystyi valitsemaan suodattukset. Kuvassa 16 on esitetty Dynamo-Playerin käyttöliittymä, jossa voidaan tutkia, onko riittävän monta elementtiä tilan pinta-alaa kohti sekä montako elementtiä on vähintään oltava tilassa. Dynamo-Playerin voi pitää samalla päällä, kun tutkii mallista tiloja ja tekee muutoksia niihin. Tämä helpottaa puutteiden etsimistä ja niiden korjaamista.



Kuva 16. Dynamo Playerin näkymä, kun etsittiin, puuttuiko savuilmaisain tai lämpöilmäisin tiloista

Skriptin toimivuutta testattiin päiväkodin suunnittelussa. Yhdestä leikkihuoneesta poistettiin testauksen ajaksi kaikki elementit, jotta voitiin varmistua skriptin toimivuudesta. Tutkiminen suoritettiin valaisimille, paloilmäisimille, datarasioille sekä liiketunnistimille. Testauksia kokeiltiin erilaisten ehtojen kanssa. Skriptin avulla löydettiin puutteet tiloista, joista elementit saivat puuttua muutenkin sekä leikkihuone, josta oli tarkoituksella poistettu kaikki elementit. Skripti käytti tilatietojen suhteen Revit-mallin tilatietoja (Space), joten arkkitehdin tilatietojen (Rooms) mahdolliset puutteet eivät vaikuttaneet skriptin toimintaan. Skriptin testausvaiheessa ei todettu ongelmia toimivuuden suhteen. Voitiin todeta skripti toimivaksi

ja tehokkaaksi työkaluksi tutkia puutteita tiloista. Liitteessä 2 on esitetty skriptin tarkempi luominen ja liitteessä 8 on esitetty kuva kokonaisesta Dynamo-skriptistä

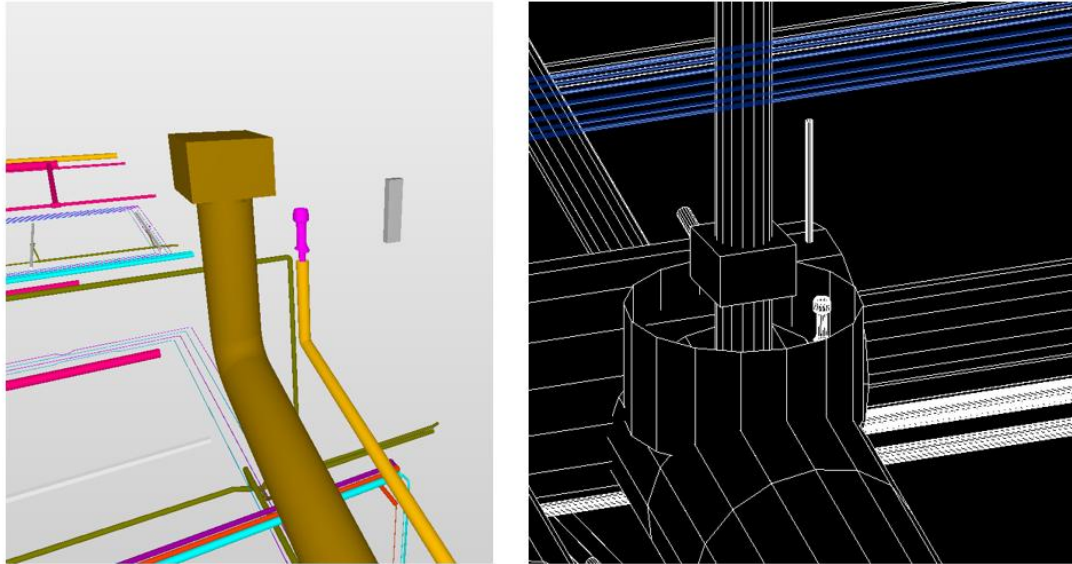
4.4 IFC-tiedoston kääntäminen Revit-tiedostomuotoon.

Revit-ohjelmaa pystytään käyttämään tehokkaasti, vaikka arkkitehti ei tekisi Revit-ohjelmalla. IFC-tiedosto voidaan kääntää Revitin RVT-muotoon avaamalla IFC-tiedosto Open IFC toiminnolla. Ohjelma kääntää IFC-parametrit Revit-parametreiksi Import IFC Options -valikon mukaisesti, johon on määritelty miten elementit lajittelevat. Vertailtiin arkkitehdin ja LVI-suunnittelun tietomalleja, miten IFC:stä RVT-tiedostoiksi käännetty tietomallit erosivat alkuperäisistä Revit-malleista.

Ongelmiksi muodostui virheelliset käännot osassa Familyissä kuten alakatoissa. Osa arkkitehdin tietomallin kalusteista oli väärillä IFC-komponenttitasoilla, jolloin Revitissä myös Familyt menivät väärin. YTV 2012 mukaisesti nämä tulisi arkkitehdin kuitenkin korjata. Käännetyn arkkitehdin Revit-mallin elementit eivät tunnista enää asennussuuntaa. Tällöin automaattipiirtoa ei voitu käyttää tilanteissa, joissa tarvittiin elementin asennussuuntaa.

Osa arkkitehdin IFC:n parametritiedoista kääntyi Revit-malliin, mutta kaikki parametrit eivät kääntyneet. Nämä puutteet saattavat vaikuttaa osassa automatisoinneissa, joissa etsitään tuotteita tiettyjen parametrin perusteella. Tässä opinnäytetyössä ei tutkita syvällisemmin voidaanko asiaan vaikuttaa.

LVI-suunnittelun IFC-tiedosto kääntämisessä RVT-tiedostoksi huomataan, että käänöksessä tuli suuria ongelmia. Kanavapaksuudet ja -pituudet eivät pitäneet paikkaansa. Kuvassa 17 on esitetty näkymä vesikaton ulospuhallushajottimelta. Kuvasta nähdään, että Revit-käännytyssä mallissa kanavapaksuudet muuttuvat suuremmiksi ja kanava ei pääty hajottimeen. Käännetty Revit-malli oli käyttökelpoton. Käännosongelmat voidaan ehkä korjata, mutta sitä ei tutkittu tässä opinnäytetyössä. Todettiin, että nykyisillä class mapping -arvoilla ei voida tehdä toimivaa käännoästä LVI:n IFC-malleille.



Kuva 17. Vasemmalla Solibri-ohjelman näkymä IFC-tiedostosta ja oikealla näkymä Revit-mallista, jossa IFC on käännetty RVT-muotoon.

4.5 Luodut Dynamo-skriptit ja niiden onnistumiset

Taulukossa 5 on esitetty skriptit, jotka voitiin ottaa käyttöön Granlundilla, joiden kehittämistä oli vielä jatkettava ennen kuin ne voidaan julkaista sekä skriptit, jotka eivät tulleet käyttöön, koska eivät toimineet halutusti, olivat ongelmallisia tai muuten vain mahdollisesti vaarallisia projektille. Skriptit on luokiteltu taulukkoon eri käyttötapojen mukaan.

Taulukko 5. Kehitetyt skriptit

Luokka	nimi	Valmis	Kesken	Ei julkaista
Määrälaskenta	Määrälaskenta mallitilojen avulla	x		
Määrälaskenta	Määrälaskenta mallitilojen avulla Schedu-letaulukoilla	x		
Määrälaskenta	Piirrä tuotteita Spacen mukaan (Massoittelu)			x
Laatu	Mistä tilasta puuttuu tietty Categoria, suodatus tilan nimellä ja kappalemäärällä	x		
Laatu	Mistä tilasta puuttuu tietty Family Type, suodatus enimmäismäärällä pinta-alan suhteen	x		
Laatu	Missä tiloissa on liian pieni valaistusteho (W/m^2), suodatus tilan nimellä, valaistusteho voidaan valita	x		
Laatu	Mistä tilasta puuttuu usea eri Family Type, suodatus vähimmäismäärällä pinta-alan suhteen	x		
Laatu	Mistä tilasta puuttuu usea eri Family Type, suodatus Level, tilan nimi, vähimmäis määrä, vähimmäismäärällä pinta-alan suhteen	x		
Laatu	Onko tietty Family Type alakattokorossa	x		
Laatu	Onko tietty Categoria alakattokorossa, suodatus oppoasennuskoodilla	x		
Laatu	Onko sähköhylly alakaton alapuoella	x		
Laatu	Onko sähköhylly alle halutun korkeuden		x	
Automaattipiirto	Piirrä elementti linkitetystä mallista	x		
Automaattipiirto	Piirrä elementti samasta mallista	x		
Automaattipiirto	Piirrä elementti alakaton mukaan			x
Automaattipiirto	Piirrä elementti Space:n mukaan, korko alakatosta jos on.		x	
Automaattipiirto	Piirrä monta elementtiä oven pieleen	x		
Automaattipiirto	tutki ja korjaa onko tietty Family Type alakattokorossa	x		
Automaattipiirto	tutki ja korjaa onko tietty Categoria alakattokorossa, suodatus oppoasennuskoodilla	x		
Automaattipiirto	tutki ja korjaa onko tietty Categoria alakattokorossa, suodatus Schedule-taulukon tyypeillä	x		
Automaattipiirto	Merkitään parametriin teksti, jos Family Type alakaton yläpuolella	x		
Automaattipiirto	Muodosta automaattisesti piirtonäkymät ja tulostusnäkömät	x		
Automaattipiirto	Muodosta Excel-taulukoita käyttäen piirustus- ja tulostusnäkömät		x	
Apuskripti	Siirrä Room nimi ja numero alakatolle	x		x
Apuskripti	Siirrä Room nimi ja numero johtoteille	x		x
Apuskripti	Siirrä Room nimi ja numero Spacelle	x		x
Apuskripti	Käännä IFC-mallit RVT-muotoon automaattisesti			x
Apuskripti	Deletoi hallitusti elementti mallista	x		x

Määrälaskenta skripti mallitilojen avulla toimii niin kuin piti. Tämän avulla on mahdollista tehdä määrälaskentaa varhaisessa vaiheessa. Skripti vaatii toimiakseen tilatiedot eli Space sekä arkkitehdin mallissa Rooms.

Automaattipiirtoskriptit toimivat oikein pistemäisten elementtien suhteen, mutta tilojen ja alakattojen suhteen niissä oli vielä paljon kehitettävää. Automaattipiirrolla on mahdollista säästää aikaa ja parantaa laatua isommissa kohteissa. Skripti vaati toimiakseen referenssimalleista oikeat kategorialuokittelut, sekä jos tarvittiin elementin suuntatietoa, oli linkitetty malli oltava tehty Revit-ohjelmalla.

Automaattinen tulostus- ja piirustusnäkyvien luomisen -skripti toimi oikein ja sillä voidaan säästää aikaa projektissa. Skripti ei vaatinut toimiakseen mitään ylimääräistä.

Laaduntarkastusskripteissä alakattojen suhteen skriptit toimivat hyvin. Tiloja, joita ei voida tarkastella on normaalisti hyvin vähän. Skripti myös ilmoitti tiloista, joista tarkastusta ei voitu tehdä. Tällöin tiedettiin mitkä tilat olivat tarkistettava muulla tavalla. Skriptin automaattikorkojen korjaus toimi myös halutusti, joten se voitiin ottaa käyttöön. Skripti vaati toimiakseen arkkitehdin mallissa oikean kategorialuokittelun alakatolle.

Laaduntarkastusskripteissä puuttuvien elementtien etsiminen toimii halutusti. Suodatuksia oli mahdollista muuttaa ja tehdä useampia, jolloin voitiin tehdä erilaisia versioita tarpeen mukaan. Tarkasteluissa voitiin hyödyntää myös tilatietojen (Space) parametrejä. Skripti vaati toimiakseen tilatiedot (Space) omasta mallista.

Ainoat skriptit, jotka vaativat ehdottomasti Revitillä tehtyä arkkitehtimallia, olivat automaattipiirtoon liittyvät skriptit, joissa oli tarpeen saada myös elementin suuntatieto. Esimerkiksi, kun piirrettiin peilille peilivalaisin ja pistorasia, on tiedettävä mihin suuntaan peilikaappi on asennettu.

5 DYNAMO-SKRIPTIEN HYÖDYNTÄMINEN SUUNNITTELUN AUTOMATISOINNISSA

5.1 Dynamo-skriptien käyttöönottoaminen Granlundilla

Granlundilla oli tehty skripteistä sähkösuunnitteluun oma kansio, jossa oli nimetty skriptit sovitusti. Uusille skripteille tehtiin käyttöohjeet, joissa kerrottiin yksityiskohtaisesti, mitä skripti tekee ja mikä on sen tavoite ja lopputulos. Luotiin skripteille lyhyet opetusvideot, joissa esiteltiin skriptien käyttöä ja toimintaa. Ohjeet sijoitettiin samaan kansioon skriptien kanssa, jotta ne löytyvät helpolla. Tehtiin taulukko, jossa esitettiin sähkösuunnitteluun tarkoitetut toimivat skriptit. Tarkoitus on ylläpitää taulukkoa lisäämällä sinne uudet kehitetyt ja toimivaksi todetut skriptit.

Dynamon käyttö vaatii opettelua kuten muutkin suunnitteluohjelmat. On pidettävä koulutuksia Granlundin sähkösuunnittelijoille skriptien käytöstä sekä ohjeistettava mistä ne löytyvät. Pidetään Granlundin sähkösuunnittelijoille infotilaisuuksia, joissa kerrotaan Dynamon käytön mahdollisuuksista ja opastetaan tulemaan koulutuksiin.

5.2 Määrälaskennan tekeminen varhaisessa suunnitteluvaiheessa Dynamo-skriptien avulla

Määrälaskenta mallitilojen avulla -skriptillä voidaan tuottaa tarkempaa tietoa kustannusten määrittelyä varten yleissuunnittelun lopussa tai esimerkiksi projektinjohtourakoitsijan sähköurakan kilpailutusta varten ilman, että tarvitsee tehdä urakkalaskentatasoisia pistepiirustuksia. Skripti vaatii toimiakseen sen, että kaikki erinimiset tilat ovat kertaalleen suunniteltuja sekä arkkitehdin mallissa on käytettävissä tilatiedot (Rooms).

Määrälaskentaskriptin avulla saatiin pistemäisten tuotteiden kappalemäärät, jotka olivat piirretty mallitiloihin, sekä myös johtoteiden osalta voitiin tehdä määrälaskenta. Elementit lajittuivat järjestelmittäin ja tuotetiedot saatiin Family Typen

tarkkuudella. Määrälaskentatiedot voidaan muodostaa vastaamaan Revitin määrälaskentaa (Bill of Materials). Skripti laski määrät kahdella eri tavalla, joissa tuotteet kerrotaan joko tilojen lukumäärällä tai tilojen pinta-alojen suhteella. Skripti loi Excel-taulukkoon eri laskentatavoista omat sivut, joissa oli laskettu yhteen tuotteiden kokonaismäärät. Määrien perusteella voitiin tutkia, kuinka lähelle skripti antoi saman tuloksen eri laskentatavoilla. Skripti loi Excel-taulukkoon lisäksi sivut, joissa kappalemääriä ei ollut laskettu yhteen. Näiden sivujen määristä voitiin luoda Pivot-taulukko, jonka perusteella voitiin nopeasti tutkia, minkä nimiset tilat antavat poikkeavan määrän.

Normaalisti yleissuunnitteluvaiheessa yhdessä päätetään, mistä tiloista tehdään mallitilat, mutta kaikista erityyppisistä tiloista ei normaalisti suunnitella mallitilaa. Tällöin olisi vietävä suunnitelmat pidemmälle, kuin yleissuunnittelussa on tarvetta. Mutta aikaa ei menisi kokonaisuudessa projektissa enempää, koska kaikki tilat ovat suunniteltava jossain vaiheessa kuitenkin. Kyse olisi siitä, milloin aikaa käytetään.

Määrälaskenta mallitilojen avulla osoittautui testeissä toimivaksi työkaluksi. Virhemarginaali vertailtavissa tuoteryhmissä oli pistemäisten elementtien osalta 0-28% ja johtoteillä 22-43%. Tuotteiden määrään voi vaikuttaa valitsemalla sopivan mallitilan sekä piirtämällä mallitilaan vain sille kuuluvat asiat. Esimerkiksi ylimääräisiä johtoteitä ei tulisi esittää mallitiloissa, mikäli niitä käytetään määrälaskennassa hyödyksi. Virhemarginaalia myös kasvatti se, että ulkoseinien ulkopuoliset asennukset huomioitiin vertailulaskennassa. Ulkopuolen asennukset tulisi arvioida vielä erikseen määrälaskentaan, jotta saataisiin tarkempi tulos.

Määrälaskennassa mallitilojen avulla on huomioitava, että vain tuotteet, jotka ovat tilaelementin sisäpuolella, tulevat mukaan laskelmaan. Tällöin, jos halutaan esimerkiksi kulunvalvontaovet, on piirrosmerkki piirrettävä huoneen puolelle. Tietyissä tapauksissa on hankaluuksia, esimerkiksi jos tilan ovella on ovipielinäyttö. Näyttö liittyy huoneeseen, mutta piirrosmerkki on käytävän puolella. Tämä voidaan ratkaista esimerkiksi siten, että siirretään merkki huoneen puolelle väliaikaisesti tai luodaan sille joku toinen piirrosmerkki huoneen puolelle, jota voidaan hyödyntää määrälaskennassa.

Skriptiin on mahdollista lisätä varmuuskerroin, jolla voidaan kertoa tuotteiden kokonaismäärät halutulla kertoimella. Sama varmuuskerroin tulisi sisällyttää skriptissä mallitilojen määrän perusteella sekä pinta-alojen suhteessa laskelmiin, jotta ne pysyisivät vertailukelpoisina.

Määrälaskentaskriptin avulla ei voida laskea keskusten eikä muiden päälaitteiden määriä, vaan niiden määrät ovat tarkasteltava eri tavalla. Määrälaskennasta saadaan kappalemäärätietoa, jota voidaan hyödyntää tele- ja turvapäälaitteiden määrittämisessä. Esimerkiksi voidaan selvittää, montako tuotetta on paloilmoinjärjestelmässä ja sen perusteella valita oikean kokoinen keskus.

Määrälaskennan kannalta on tärkeää, että tilatiedot (Space) toimii oikein ja ne ovat oikein määriteltäviä. Tilatietojen (Space) nimiä voidaan kuitenkin yhdistellä ja muuttaa, koska ne haetaan ainoastaan omasta Revit-mallista. Arkkitehdin Revit-mallin Room-tiedoista haetaan vain tilanumeroita, joita käytetään johtoteiden vertailuissa. Tilatietojen rajojen on kuljettava tarkasti huoneiden seinien mukaisesti ja tilatietojen on oltava yläpohjaan asti, jotta myös alakaton yläpuoliset elementit tulevat mukaan laskelmaan.

Toteutuspiirustus- tai loppupiirustusvaiheessa vaiheessa voidaan tehdä Revitin määrälaskenta (Bill of Materials) ja verrata sitä mallitilojen avulla tehtävään määrälaskentaan. Molemmissa tuotetiedot voidaan lajitellaan Description sekä MC System Code -parametrien avulla. Jos määrälaskentaa mallitilojen avulla on käytetty sopimusasiakirjoissa, voidaan taloudellisessa loppuselvityksessä tehdä kattava vertailu loppupiirustusten määrälaskennan avulla.

Revit-ohjelmaan on mahdollista määrittellä jokaiselle Family Typelle yksikkökustannus (Cost). Tällöin on mahdollista antaa esimerkiksi järjestelmäkohtaisesti hinta-arvio skriptin avulla tilaajalle. Kustannusten määrittely ei kuitenkaan normaalisti kuulu sähkösuunnittelijan tehtäviin. Kustannusten muutoksien vertailussa olisi kuitenkin helppoa, jos kustannustieto sisällytettäisiin Revit-ohjelmaan. Tällöin ulostulevaan Excel-taulukkoon tulostuvat rivit olisivat aina oikein, jolloin kenenkään ei tarvitsisi käsin määrittellä tuotteiden hinta-arvoja uudelleen. Kun ohjelmaan lisätään uusi tuote, määrälaskentataulukossa sen jälkeisten tuotteiden

rivit siirtyvät alaspäin. Kustannuksia voisi nopeasti vertailla muuttamalla mallitilojen ratkaisuja ja päivittämällä laskelma tai tekemällä uusi vertaileva laskelma.

Vertailua voi myös tehdä siten, että suunnitellaan malliin eri versioita mallihuoneista. Nimetään mallihuoneet eri nimillä, esimerkiksi Mallihuone-1 ja Mallihuone-2. Skriptiin voisi määritellä, minkälaisilla nimillä mallihuoneita etsitään. Yhteisillä tiloilla voidaan käyttää samaa nimeä. Esimerkiksi tehdään laskelma, jossa haetaan "Mallihuone" ja "Mallihuone-1" -nimiset tilat, sekä vertailulaskelma, jossa haetaan "Mallihuone" ja "Mallihuone-2" -nimiset tilat.

5.3 Suunnittelun nopeuttaminen ja laadun parantaminen Dynamo-skripteillä

Suunnittelua voidaan nopeuttaa automatisoimalla toistoa vaativia toimenpiteitä. Näissä on tyypillistä, että asiat tehdään aina samalla tavalla. Esimerkiksi piirustus- ja tulostusnäkyvien luomisen automatisointi -skripti tai skripti, joka kirjoittaa alakaton yläpuolisille elementeille parametriin halutun tekstin, kuten AY (=alakatton yläpuolella).

Suunnittelua voidaan nopeuttaa myös tarkastustoiminnoilla. Tällöin suunnittelijan ei tarvitse itse tehdä tarkastelua Revitin tai muun suunnitteluohjelman avulla. Esimerkiksi voidaan tarkistaa automaattisesti, onko uppovalaisimet mallinnettu samaan korkeuteen kuin alakatto tai onko sähköhyllyjä mallinnettu alakattojen alapuolelle. Silmämääräinen tarkastaminen ohjelmien avulla on aikaa vievää. Kun tarkastuksia tehdään muilla suunnitteluohjelmilla, on silloin Revit-mallista julkaistava IFC-käännös, joka ladattava toiseen ohjelmaan. Tämä toimenpide on myös aikaa vievää sekä korjaukset on tehtävä omalla Revit-mallilla.

Automaattipiirrot muiden suunnittelijoiden referenssielementtien avulla nopeuttaa suunnittelua, jos niitä on runsaasti. Vähäistä määrää ei kannata lähteä tekemään automatisoimalla. Automaattipiirron etuna on myös, että voidaan referenssielementistä kopioida tietoja, kuten ilmansäätöpellin yksilöity koodi. Tällöin koodi on täsmälleen sama kuin referenssielementillä, jolloin ei voi tapahtua inhimillisiä

kirjoitusvirheitä. Automaattipiirron tulokset ovat kuitenkin jossain vaiheessa tarkistettava silmämääräisesti, koska on mahdollista, että elementti piirtyy toisen merkin päälle, tällöin 2D-tulosteesta tulee epäselvä.

Automaattiset toiminnot parantavat laatua, jos skripti on muodostettu ja testattu kunnolla siten, että se tekee luvatut toimenpiteet virheettömästi. Jos skriptillä on rajoitteita, tulee niitä ilmoittaa käyttäjälle selvästi tieto. Esimerkiksi kun skriptillä tutkittiin elementtejä alakaton korkeuden suhteen, Dynamo Player ilmoitti tilat, joissa oli elementtejä, joita ei voitu tutkia alakaton suhteen. Tällöin saatiin tieto, että asia on tutkittava muilla keinoilla.

Automatisointi parantaa myös laatua siten, että se löytää kaikki elementit, jotka skriptiin on ohjelmoitu sekä tekee aina sovitut toimenpiteet kerralla kaikkiin. Normaalisissa suunnittelussa vastaavan työn tekeminen saattaa jäädä kesken erinäisistä syistä johtuen, kuten työpäivän päättymisen tai puhelimeen vastaamisen takia.

Laadunparannusta voidaan tehdä myös tutkimalla mallista puutteita automaattisesti. Voidaan etsiä tehokkaasti eri tiloista, puuttuuko niistä jokin haluttu tuote tai järjestelmä. Esimerkiksi tutkitaan onko kaikissa tiloissa valaisin tai onko toimistohuoneissa yleiskaapelointirasia. Tämän tyyppisiä skriptejä voidaan tarkentaa erilaisilla keinoilla, joissa voidaan hyödyntää laskennallisia ominaisuuksia ja tilatietojen parametrejä.

Suunnittelijan aika on rajallinen, joten tarkastamisiin ei voida käyttää määrättyä enempää aikaa. Tarkastusohjelmien avulla voidaan työtä nopeuttaa, jolloin ehditään tekemään enemmän tarkasteluita. Suunnittelija ei kuitenkaan tee skriptejä vaan käyttää niitä, joten skriptit ja ohjeet on löydettävä valmiiksi. Jos skriptiä on tarkoitus muokata Dynamo-näkymässä, on siitä oltava myös selvä ohje, jotta suunnittelija voi luotettavasti tehdä sen. Lisäksi suunnittelijalle on oltava järjestetty koulutus Dynamon käytöstä.

Määrälaskenta mallitilojen avulla -skripti mahdollistaa suunnittelun nopeuttamista, jos projektinjohtourakassa sovitaan, että sähköurakoisija voidaan kilpailuttaa piirustuksissa esitettyjen mallitilojen ja määrälaskennan perusteella. Tällöin

voidaan välttyä siltä, että tehdään turhan aikaisessa vaiheessa tarkat pistepiirustukset, jotka myöhemmin piirretään uudelleen, kun suunnitelmia kehitetään.

5.4 Dynamon ohjelmalliset rajoitukset

Revitin Dynamolla voidaan tehdä hyödyllisiä skriptejä suunnittelun avuksi, mutta skriptien tekemisessä ja käyttämisessä on haasteita ja ongelmia. Dynamossa on teknisiä haasteita, jotka liittyvät solmujen käyttäytymiseen. Lisäksi on ominaisuuksia, jotka asettavat rajoituksia käyttämisen suhteen.

Johtotiet kuuluvat System Familyihin ja Revit ei määrittele niille tilan nimeä tai numeroa. Yksittäinen johtotie elementti voi kulkea monen tilan lävitse. Jos määrälaskennoissa mallitilojen avulla halutaan huomioida johtotiet, on niiden sijainnit valittava elementin insertiopisteen avulla vastaavasti kuin alakattoihin liittyvissä tarkastusskripteissä. Tällöin pitkä yhtenäinen johtotie määräytyy vain yhteen tilaan sen keskipisteen perusteella. Määrälaskennoissa mallihuoneiden avulla on tällöin ehdottomasti katkaistava johtotiet tilan reunoilta, jotta skripti toimii oikein.

MagiCADin "MC Default System Code" -arvo määritellään Dataset Properties valikossa. Jos esimerkiksi datarasiolle ei määritä Default-arvoa, koska sitä voidaan käyttää useissa järjestelmissä, tällöin automaattinen MC System Code ja Name muodostaminen ei toimi, koska arvo jää Type-parametreihin tyhjäksi. Revit käyttää näkyvyyksien hallintaan näitä parametreja, joten näissä tapauksissa on kirjoitettava arvot käsin jälkikäteen.

Projekteissa, joissa arkkitehti ei käytä Revitiä suunnitteluun, joudutaan kääntämään IFC-tiedosto RVT-tiedostoksi. Automatisoinneissa ongelmiksi muodostui virheelliset käännöt Familyissä kuten alakatot. Osa arkkitehdin tietomallin kalusteista oli väärillä IFC-komponenttitasoilla, jolloin Revit-ohjelmassa myös Familyt menivät väärin. Käännetyn arkkitehdin Revit-mallin elementit eivät tunnista asennussuuntaa, jolloin automaattipiirtoa ei voitu käyttää, jos tarvittiin suuntatietoa. LVI-suunnittelun IFC-tiedosto, kun käännettiin RVT-tiedostoksi, koettiin RVT-tiedosto näissä kokeiluissa hyödyttömäksi, koska käännöksestä aiheutui suuri

määrä virheitä. Tässä työssä ei tutkittu, voidaanko millä tavoilla parantaa mallin kääntämistä.

Dynamo-skriptien toimimisessa huomattiin ongelmakohtia, kun käytettiin skriptejä eri vuosiversioilla. Kaikki solmut eivät toimineet samalla tavalla eri versiolla. Yleensä ongelma oli, että uudemmalla versiolla tehty skripti ei ollut yhteensopiva vanhemman version kanssa, koska uudemman version uusia solmuja ei löydy vanhemmasta versiosta. Ongelmia havaittiin myös, että 2020 versiolla tehty skripti ei toiminut vastaavasti 2021 versiolla. Selvitettiin, että ongelman aiheuttanut solmu ei ilmoittanut virheestä, vaan se tuotti puutteellista tietoa. Tällöin skriptin loppuosa toimi väärin.

5.5 Dynamo-skriptien käytön haasteet

Määrälaskenta mallitiloissa skriptin toiminta vaatii virheettömiä tilatietoja (Space), jotta skripti suorittaa kaikki laskelmat loppuun asti ja lopputulokset ovat virheettömät. Schedule-taulukoilla voidaan tarkastella tilatietoja ja muokata niitä. Virheeliset tilat voidaan poistaa esimerkiksi Dynamon deletointi-skriptillä. Jos arkkitehdin tiloissa on virheitä, samat virheet kopioituvat sähkösuunnittelun Revit-ohjelmaan kun luodaan tilatiedot.

Automaattipiirto voidaan toteuttaa vain kerran projektissa. Esimerkiksi kun piirretään LVI:n ilmansäätöpelleille moottoripellin elementit, on sen jälkeiset muutokset piirrettävä käsin. Jos käytetään uudestaan automaattipiirtoa, tulee jo piirretyille elementeille piirrosmerkki kahteen kertaan.

Arkkitehdin mallin alakatolle (Ceiling) ei määräydy tilatietoa kuten monille muille elementeille. Skriptien tekemisen aikana kokeiltiin hyödyntää alakattoja automatisoimisessa siten, että määriteltiin mihin tilaan alakatto kuuluu. Jostain syystä kaikki alakatolliset tilat eivät kuitenkaan tulleet mukaan lajitteluun. Osa ongelmista johtui väärin mallinnetuista alakatoista, mutta kaikki virheet eivät johtuneet siitä. Väärin mallinnetut alakatot ovat YTV 2012 vastaisia, joten sitä ei varsinaisesti luokitella ongelmaksi, koska arkkitehdin tulisi korjata ne.

Automaattipiirroksessa ongelmana on 2D-piirrosmerkkien päällekkäinen piirtyminen. Testaamisen aikana huomattiin, että moottoripellin elementit piirtyivät valaisimien kanssa samalle kohdalle. Piirustukset ovat lopuksi tarkistettava ja siirrettävä elementit sopivaan kohtaan, jotta 2D-tulosteista tulee luettavia. LVI-suunnitelmista sähköä vaativien komponenttien löytäminen vaatii aikaa ja ammattitaitoa. Normaalisti käytetään hyödyksi toimintakaavioita, joiden perusteella lähdetään etsimään elementtejä tasopiirustuksista. Tätä voitaisiin nopeuttaa, jos LVI-suunnittelija määrittäisi sovitulle parametrille tiedon, jos laite vaatii sähköä tai ohjausta. Tämä ongelma on ratkaistu Granlundin uusissa sisäisissä projekteissa, mutta muiden yritysten kanssa tämä on vielä ratkaisematta.

Arkkitehdin käyttävät eri projekteissa erilaisia nimityksiä elementeille sekä määrittelevät tietoja eri parametreille. Automatisoinnissa on jokainen projekti erikseen käytävä läpi ja tarkistettava, mistä löytyvät oikeat parametrit. Pienissä kohteissa, joissa ei ole paljoa toistoa, on nopeampi tapa piirtää normaalisti.

6 JOHTOPÄÄTÖKSET JA POHDINTA

6.1 Ajatuksia päätuloksista

Opinnäytetyön tavoitteena oli auttaa Granlundia ymmärtämään Revitin Dynamo skriptien hyödyt ja haasteet sähkösuunnittelun automatisoimisessa kehittämällä Dynamolla uusia tarpeellisia skriptejä ja kokeilla niitä todellisella kohteella osana suunnittelua ja selvittää minkälaisilla skripteillä voidaan nopeuttaa suunnittelua, parantaa laatua tai vertailla vaihtoehtoja.

Granlundin Stenmanin asiantuntijahaastattelun perusteella selvitettiin yrityksen lähtötilanne sähkösuunnittelun automatisoimisessa Dynamolla. Todettiin, että Dynamoa ei juurikaan käytetä erikseen sähkösuunnittelun automatisoinnissa, mutta parametritietojen oikeellisuuden tarkistamisissa ja joissain yksinkertaisissa tiedonsiirto skripteissä käytetään Dynamoa. Granlund Jyväskylän Törmälän asiantuntijahaastattelussa nähtiin hyödylliseksi automatisoinnit, joilla voidaan tuottaa tarkkaa määrälaskentaa varhaisessa vaiheessa. Törmälän mukaan laaduntarkastus on suunnittelijan vastuulla, jolloin kaikki laaduntarkastukseen liittyvät automatisoinnit ovat tärkeitä.

Kyselytutkimuksessa selvitettiin tarve, minkälaisia skriptejä suunnittelijat ja projektipäälliköt kokisivat hyödyllisiksi suunnittelun apuna. Tarpeen perusteella toteutettiin useita skriptejä, jotka voitiin todeta toimiviksi kokeilujen jälkeen. Alla on lueteltu merkittävimmät toimivat skriptit.

Sivujen käsittely -skripti:

- Automaattisesti piirustus- ja tulostusnäkyymiä

Automaattipiirtoskriptit:

- Piirrä elementtejä linkitetyn mallin elementin avulla ja kopioi siihen tietoja
- Tarkista ja siirrä alakattoon asennettavat elementit alakaton korkeudelle
- Kirjoita alakaton yläpuoliselle elementille tekstiä haluttuun parametriin

Laaduntarkastusskriptit:

- Etsi mistä tilasta puuttuu tietty elementti
- Etsi onko alakattoon asennettavat elementit alakaton korkeudella
- Etsi onko sähköhyllyjä alakaton alapuolella

Määrälaskentaskripti:

- Tee määrälaskenta mallitilojen avulla

Sähkösuunnittelun automatisointia voidaan Dynamon avulla tehostaa usealla eri tavalla. Voidaan nopeuttaa toistoa vaativien toimenpiteiden tekemistä, jolloin jää aikaa enemmän suunnittelulle, kuten piirustus- ja tulostusnäkyvien automaattinen luominen. Isoissa kohteissa automaattipiirroilla, joissakin asioissa, säästetään aikaa mallintamisessa sekä saadaan tarkalleen samankaltaisia piirrosteknisiä ratkaisuja. Laaduntarkastusskripteillä suunnittelijat ja projektipäälliköt voivat tarkistaa nopeasti tiloja puuttuuko niistä tuotteita ja tarkistaa tuotteiden asennuskorkoja alakattojen suhteen.

Dynamo-skripteistä osa skripteistä toimii itsenäisesti projektista toiseen ja nopeuttaa toimintaa, kuten näkyvien automaattinen luominen ja laaduntarkastusskriptit. Osa skripteistä vaatii tietojen selvittelyä muiden suunnittelualojen tietomalleista sekä käsityötä skripteihin, jotta voidaan löytää halutut parametrit, joita voidaan käyttää automatisoinnissa. Varsinkin pienissä kohteissa on nopeampaa piirtää esimerkiksi sosiaalitulojen peilivalaisimet ja pistorasia kosketusvapaalle hanalle paikoilleen ilman automatisointia, koska samantyyppisten tuotteiden määrä muiden suunnittelualojen mallissa on vähäinen.

Määrälaskenta mallitiloilla -skripti vaatii toimiakseen, että suunnitellaan kertaalleen kaikki erityyppiset tilat. Yleissuunnitteluvaiheessa ei kuitenkaan suunnitella kuin erikseen sovitut mallitilat. Tällöin, jos halutaan tehdä kattava määrälaskenta, on käytettävä enemmän suunnittelu-aikaa yleissuunnitteluvaiheessa. Sinällään aikaa ei kulu projektissa enempää, vaan kyse on, milloin suunnittelu-aikaa käytetään. Tämä skripti on hyödyllinen, jos halutaan laskea aikaisessa vaiheessa tarkempia kustannustietoja, sekä se sopii projektinjohtourakoihin, joissa voidaan sopia, että sähköurakoitsija kilpailutetaan määrälaskennan avulla, joita täydentää mallitilojen tasopiirustukset.

Skriptien soveltuvuus eri suunnitteluvaiheille määräytyy tarpeen mukaan. Laaduntarkastamiseen ja puutteiden etsimiseen tarkoitettuja skriptejä voidaan käyttää koko suunnittelun ajan. Piirustus- ja tulostusnäkyvien automaattisen luomi-

sen -skriptiä voidaan käyttää tarpeen mukaan vain sen verran kuin on sillä hetkellä tarpeellista. Skriptit, jotka piirtävät uusia elementtejä muiden tietomallien perusteella on mahdollista käyttää vain kerran. Tällöin sen käyttö on ajoitettava siihen hetkeen, kun toisen osapuolen tietomalli on valmis.

Kehitystyössä tutkittiin automaattipiirtoa huoneiden alakattoon, mutta päädyttiin lopettamaan tutkiminen, koska elementtien sijoittelu huoneeseen on äärimmäisen monimutkainen operaatio, jos halutaan ottaa huomioon kaikki tarpeelliset asiat. Tultiin lopputulokseen, että pisteiden sijainnit katossa on kuitenkin tarkistettava ja yhteensovittettava uudelleen. Tällöin tarkistamisiin ja yhteensovituksiin saattaa kulua jopa enemmän aikaa kuin kertaalleen piirtoon. Työssä kehitettiin skripti, jolla voidaan piirtää halutun nimiseen tilaan tietty määrä elementtejä alakattoon, antaa niille etäisyydet x- ja y-suunnissa sekä antaa niille asennuskulma.

Suunnittelijoiden voidaan olettaa osaavan käyttää Dynamo Playerillä skriptejä, joiden toiminnot ovat vastaavan tasoisia kuin Revit-ohjelmassa normaalistikin. Tällöin heidän tulee kyetä valitsemaan muutamia opastettuja muuttujia ohjelmassa. Skriptin ohjelmointitilan monimutkaisuus ei sinällään pelota käyttäjää, jos Dynamoä käytetään Playerillä, jossa on kohtuullinen määrä valintoja sekä Playeriin on tehty kenttä, joka ilmoittaa tapahtuneet muutokset ja mahdolliset puutteet. Suunnittelijoille on lisäksi järjestettävä lyhyt koulutus ohjelman käytöstä. Ohjeet on oltava jokaisesta skriptistä, jotta suunnittelija voi varmistua, mitä skripti tekee ja miten sitä käytetään. Video-ohjeet ja esimerkkivideot ovat hyviä tapoja esitellä toimintoja. Kehitetystä skripteistä, joita otettiin käyttöön, on luotu kirjalliset ja video-ohjeet.

Halukkaille suunnittelijoille on järjestettävä myös koulutuksia, joissa opetetaan Dynamon käyttöä. Tällöin suunnittelija saa valmiuden muokata ja kehittää itse skriptiä oman tarpeen ja projektin mukaiseksi. Graafinen ohjelmointi on kuitenkin hyvin erilaista kuin suunnitteluohjelman käyttö, joten kovin vaativista koulutuksista kaikille ei voida puhua. Tavoitteena on enemmänkin, että suunnittelija osaisi selvittää yleisimmät ongelmat, kuten puuttuvat kustomoidut paketit sekä muuttamaan haettavia ja kirjoitettavia parametreja.

Useissa skripteissä käytetään tilatietoja, eli arkkitehdin mallin Rooms tai sähkösuunnittelun Revit-mallin Space. Näiden tilatietojen on oltava paikkansa pitäviä, jotta skriptit voivat toimia oikein. Skriptien luomisen aikana sekä testausvaiheessa tuli eteen useamman kerran tilanne, jossa oli rikkinäisiä tai ylimääräisiä tilatietoja. Myös tilatietojen korkeudet eivät vastanneet yleisen tietomallivaatimuksen mukaista tapaa, jossa korkeus määritellään yläpuolisen laatan alapintaan.

Muiden suunnittelualojen tietomallien ongelmien suhteen on kuitenkin muistettava, että niiltä edellytetään suunnittelusopimuksen mukaista suunnittelun tasoa. YTV 2012 mukainen taso 2 mahdollistaa tämän opinnäytetyön Dynamolla tehtävät suunnittelun automatisoinnit.

6.2 Tutkimuksen luotettavuus

Skriptejä testattiin jatkuvasti kehitysvaiheessa sekä lopuksi skriptit testattiin todellisessa ympäristössä. Skriptien testauksessa pyrittiin löytämään kaikki mahdolliset ongelmat. Kappaleessa 6.1 esitetyt skriptit toimivat luotettavasti ja ne on otettu käyttöön sähkösuunnittelussa. Osa skripteistä todettiin kehitysvaiheessa epätoimiviksi ja niiden kehitys keskeytettiin. Joissakin skripteissä huomattiin testausvaiheessa ongelmia ja päätettiin, että niitä ei julkaista.

Osa skripteistä voitiin kokeilla todellisella suunnittelukohteella. Sen arkkitehtisuunnittelu oli tehty ArchiCADilla, joten osa skripteistä jouduttiin testaamaan demoprojektilla, jossa oli käytettävissä MagiCADin demoprojektin arkkitehdin Revit-mallia. Demoprojektin ympäristö vastasi kuitenkin todellisista kohdetta, koska se oli perustettu uudelleen Granlundin mukaisella tavalla.

Laaduntarkastusskriptien luotettavuuden testauksessa käytettiin hallittuja virheitä, jolloin niiden paikantaminen oli valmiiksi tiedossa. Tällöin voitiin varmistua löytääkö ohjelma kaikki virheet ja puutteet. Laaduntarkastusskriptejä tutkittiin todellisella kohteella, jolla pyrittiin löytämään piileviä ongelmia skriptistä.

Määrälaskenta mallitilojen avulla -skriptin tuloksia verrattiin todellisen kohteen urakkalaskentamalliin. Urakkalaskentamallista tyhjennettiin kaikki muut tilat paitsi

mallihuoneet. Tällöin voitiin keskittyä ohjelman antamiin eroihin, eikä suunnittelun tietosisällön puutteisiin suunnittelun alkuvaiheessa.

Tässä opinnäytetyössä ei annettu muille suunnittelijoille koulutusta Dynamon käytöstä eikä muut suunnittelijat kokeilleet näiden skriptien toimintaa. Tällöin tulkinat Dynamon ja Dynamo Playerin käytöstä osana normaalia suunnittelua perustuvat vain omaan käyttökokemukseen ja testaamiseen.

6.3 Jatkotutkimustarpeet

Laaduntarkastusskripteissä, joissa etsitään puuttuvia elementtejä tiloista, voidaan kehittää entistäkin monipuolisemmiksi sekä kohdentaa ne yleisiin puutteisiin. Tällöin suunnittelijat voisivat jatkossa helpolla ajaa tarkastuksia, ilman että tarvitsee miettiä, mitä halutaan selvittää. Mahdollisesti voisi rakentaa Excel-taulukon, johon listattaisiin eri järjestelmien vaatimuksia ja suorittaa se Dynamolla.

Määrälaskenta mallitilojen avulla -skriptiä on mahdollista kehittää sisällyttämällä siihen kustannustietoa sekä lisäämällä siihen muokattavan varmuuskertoimen, jolla voidaan kasvattaa hallitusti määrää laskentaa varten. Skriptiä on myös mahdollista kehittää siten, että voidaan tehdä vertailevia laskelmia. Mallitiloja voisi piirtää useampia ja määritellä skriptiin, mitkä ovat yhteisiä mallitiloja ja mitkä vaihtoehtoisia mallitiloja. Tämä mahdollistaisi kustannusvertailun tekemisen tehokkaasti.

IFC:n kääntäminen Revit-muotoon ei onnistunut riittävän hyvin. Jatkotutkimusaiheena voisi selvittää, miten voidaan vaikuttaa parempaan käynnöksen lopputulokseen. Miten saadaan arkkitehdin tietomallin elementeistä suuntatietoa sekä kuinka LVI:n tietomalli voidaan kääntää, jotta tieto säilyy oikeanlaisena.

Automaattipiirroksessa muiden suunnittelijoiden tietomallien avulla, voisi tutkia miten voidaan tunnistaa jo piirretyt elementit. Tällöin automaattipiirtoa voisi käyttää myös uudelleen, kun toinen suunnittelija on päivittänyt tietoja omaan malliin. Tässä voisi mahdollisesti hyödyntää tuotteiden ID-koodia, joka kopioitaisiin piirrettävälle elementille. Vakioitujen tietosisältöjen määrittämisen jälkeen, voidaan

skriptien automaattipiirrot kehittää paremmin toimiviksi, kun ei tarvitse tietomallin kautta etsiä sopivia parametreja.

Automaattipiirtoa, jossa piirretään tiloihin ja seinille tuotteita, voisi erikseen tutkia tarkemmin. Miten voidaan elementit kohdistaa suoraan arkkitehdin alakattoruudukkoon ja miten voidaan määrittellä tuotteiden määrät tilatiedon perusteella, jossa huomioitaisiin tilan pinta-ala, henkilömäärä ja valaistusvaatimukset. Seinien osalta voisi tutkia, onko mahdollista hyödyntää sijoittelussa irtokalustusta, tilatietojen henkilömäärää sekä ikkunan alareunan korkeutta ja piirtää niiden perusteella tarvittava määrä johtokanavia, pistorasioita ja yleiskaapelointirasioita. Esimerkiksi kerrostalon mallintamisen voisi pystyä automatisoimaan melko pitkälle.

LÄHTEET

Arcance System. Revit. 2021. Luettu 15.12.2021. <https://arkance-systems.fi/ohjelmistot/autodesk/revit/>

Autodesk. Revit. 2021. Luettu 19.11.2021. https://www.autodesk.fi/products/revit/overview?us_oa=dotcom-us&us_si=6564a08b-730b-4b1a-9d65-2ce13f3c24a3&us_st=revit&term=1-YEAR&tab=subscription#revit-intro

Building Smart Finland. 2012. Yleiset tietomallivaatimukset YTV 2012. Luettu 30.3.2021. <https://buildingsmart.fi/yleiset-tietomallivaatimukset-ytv/>

Building Smart Finland. 2012. Yleiset tietomallivaatimukset YTV 2012 Osa 1 Yleinen osuus. Luettu 30.3.2021. <https://buildingsmart.fi/yleiset-tietomallivaatimukset-ytv/>

Building Smart Finland. 2012. Yleiset tietomallivaatimukset YTV 2012 Osa-3 Arkkitehtisuunnittelu. Luettu 30.3.2021. <https://buildingsmart.fi/yleiset-tietomallivaatimukset-ytv/>

Building Smart Finland. 2012 , Yleiset tietomallivaatimukset YTV 2012 Osa-4 Talotekninen suunnittelu arkkitehtisuunnittelu. Luettu 30.3.2021. <https://buildingsmart.fi/yleiset-tietomallivaatimukset-ytv/>

Dynamobim forum. DynamoBIM forum. 2021. Luettu 3.12.2021. <https://forum.dynamobim.com/>

Dynamo Primer. 2021. Luettu 15.12.2021. <https://primer.dynamobim.org>.

Dynamo Primer, Getting Started. 2021. Luettu 15.12.2021. https://primer.dynamobim.org/02_Hello-Dynamo/2-6_the_quick_start_guide.html

Dynamo Primer, Lists of Lists. 2021. Luettu 19.12.2021. https://primer.dynamobim.org/06_Designing-with-Lists/6-3_lists-of-lists.html

Dynamo Primer, Managing Your Progran. 2021. Luettu 19.12.2021. https://primer.dynamobim.org/03_Anatomy-of-a-Dynamo-Definition/3-4_best_practices.html

Dynamo Primer, Nodes. 2021. Luettu 19.12.2021. https://primer.dynamobim.org/03_Anatomy-of-a-Dynamo-Definition/3-1_dynamo_nodes.html

Dynamo Primer. Python Nodes. 2021. Luettu 15.12.2021. https://primer.dynamobim.org/10_Custom-Nodes/10-4_Python.html

Dynamo Primer, The Dynamo User Interface. 2021. Luettu 15.12.2021. https://primer.dynamobim.org/02_Hello-Dynamo/2-2_the_dynamo_ui.html

Dynamo Primer, What's a List. 2021. Luettu 19.12.2021. https://primer.dynamobim.org/06_Designing-with-Lists/6-1_whats-a-list.html

Dynamo Primer, What is Dynamo? 2021. Luettu 15.12.2021. https://primer.dynamobim.org/01_Introduction/1-2_what_is_dynamo.html

Leskinen, H. 2019. Rakennussähkösuunnittelun automatisointi. Sähkötekniikan koulutusohjelma. Lappeenrannan-Lahden teknillinen yliopisto. Diplomityö.

Junnonen, J-M, Kankainen J. 2017. Rakennuttaminen. 5. korjattu painos. Vaasa: Kirjapaino: Oy Farm Ab.

Järvenpää, J. 2019. Python Ohjelmoinnin opasopettajalle. 1. painos. Keuruu: Otavan Kirjapaino Oy.

Ohjelmointiputka, Oppaat: Python-ohjelmointi. 2021. Luettu 16.8.2021. https://www.ohjelmointiputka.net/oppaat/sarja.php?tunnus=python3#python3_12

MagiCAD for Revit – PV opetusmateriaali. 2018. Versionumero 2018. Progman

MagiCAD, Mitä on BIM. 2021. Luettu 16.12.2021. <https://www.magicad.com/fi/bim/>

MagiCAD, MagiCAD Electrical. 2021. Luettu 1.12.2021. https://www.magicad.com/fi/mc_software/magicad-electrical/#overview

Stenman, M. BIM Developer. Haastattelu 22.12.2021. Haastattelija Kontunen, J. Teams

Svens, T. 2021. Revit Dynamon soveltaminen KSL-suunnitteluun. Konetekniikan tutkinto-ohjelma. Metropolia Ammattikorkeakoulu. Insinööriyö

ST 70.12. 2017. S2010-SÄHKÖNIMIKKEISTÖ. SÄHKÖENERGIAN JAKELUJA KÄYTTÖJÄRJESTELMÄT, TIETOTEKNISET JÄRJESTELMÄT. Sähköinfo Oy

RT 10-11129. 2013. Taloteknisen suunnittelun tehtäväluettelo TATE12. Rakennustieto Oy

RT 10-11290. 2017. Taloteknisen suunnittelun tehtäväluettelo TATE18. Rakennustieto Oy

Törmälä, A. DI, Granlund Jyväskylän varatoimitusjohtaja. Haastattelu 7.1.2022. Haastattelija Kontunen, J. Teams

LIITTEET

Liite 1. Kyselytutkimus sähkösuunnittelun automatisoimisesta Dynamolla

Liite 2. Dokumentti Dynamo-skriptien luomisesta

Liite 3. Skripti: Piirustus- ja tulostusnäkyvien automaattinen luominen

Liite 4. Skripti: Määrälaskenta mallitilojen avulla

Liite 5. Skripti: Elementtien automaattinen mallintaminen muiden suunnittelualojen tietomallin perusteella

Liite 6. Skripti: Elementtien asennuskorkeuden tarkastaminen ja muuttaminen alakaton perusteella

Liite 7. Skripti: Johtoteiden asennuskorkeuden tarkastaminen alakaton korkeuden suhteen

Liite 8. Skripti: Puuttuvien elementtien listaaminen automaattisesti

Kysely sähkösuunnittelun automatisoinnista Magicad for Revit -ohjelman Dynamo-työkalulla

Revitin Dynamo-työkalu mahdollistaa suunnittelun automatisointia graafisella ohjelmointityökalulla. Mitkä asiat mielestäni voisi nopeuttaa suunnittelua ja parantaa laatua? Alla muutamia esimerkkejä automatisoinnista. Kerro ihmeessä ajatuksistasi, joita olet miettinyt voisiko niitä jotenkin automatisoida.

Sivujen käsittelyyn liittyviä automatisointeja

Arvosana 1-5

1. Ohjelmalle kerrotaan halutut järjestelmät ja tiedot. Ohjelma tekee automaattisesti halutut näkymät

Merkitse vain yksi soikio.

	1	2	3	4	5	
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

2. Ohjelmalle kerrotaan halutut järjestelmät. Ohjelma tekee automaattisesti halutut tulosteikkunat

Merkitse vain yksi soikio.

	1	2	3	4	5	
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

3. Ohjelma päivittää automaattisesti nimiöt M-filesin muuttuneiden tietojen kanssa

Merkitse vain yksi soikio.

	1	2	3	4	5	
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

(JATKUU)

4. Omia ideoita?

Ehdotuksia periaatepiirustusten automatisoinnissa.

Arvosana 1-5

5. Olisiko hyödyllistä olla valmiina kattavat periaatepiirustukset templatessa?

Merkitse vain yksi soikio.

	1	2	3	4	5	
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

6. Kerro ideoistasi mitä haluaisit automatisoida.

Ehdotuksia keskuskaavion automatisoinnista.

7. Kerro ideoistasi mitä haluaisit automatisoida.

Automaattipiirron hyödyntämisen mahdollisuudet.

Käytetään hyväksi arkkitehdin tietomallin objektien tietosisältöä sekä tilojen (space/room) tietosisältöjä. Ohjelma piirtää automaattisesti tuotteet (3D-objektit) oikein malliin.
Arvosana 1-5

8. Ovipielituotteiden piirto kerralla kaikkiin haluttuihin tilatyyppeihin. Annetaan tuotteet ja korot. Ohjelma piirtää automaattisesti tuotteet oven vierelle.

Merkitse vain yksi soikio.

	1	2	3	4	5	
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

9. Kattotuotteiden piirto kerralla kaikkiin haluttuihin tilatyyppeihin. Esim paloilmaisin, liiketunnistin, valaisin yms. Annetaan siirtymän suuruus (x,y) huoneen keskikohdasta

Merkitse vain yksi soikio.

	1	2	3	4	5	
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

10. Valaisimien piirto kerralla kaikkiin haluttuihin tilatyyppeihin. Tunnistaa huoneen mitat. Valitaan valaisintyyppi ja valaisimien etäisyydet, ohjelma sijoittaa valaisimet automaattisesti tilaan

Merkitse vain yksi soikio.

	1	2	3	4	5	
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

11. Tunnistaa ARK-mallista erilaisia tietoja. Esim "KV" merkin ovista ja piirtää KV merkin tasoon. Tunnistaa ikkunasta/ovesta "SP" merkin ja piirtää SP-moottorin yms.

Merkitse vain yksi soikio.

	1	2	3	4	5	
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

12. Tunnistaa LVI-mallista erilaisia tietoja. Esim säätöpellin ja piirtää säätöpellin piirrosmerkin tasoon. yms.

Merkitse vain yksi soikio.

	1	2	3	4	5	
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

13. Kerro ideoistasi mitä haluaisit automatisoida.

Piirretyille tuotteille tietojen lisääminen jälkikäteen.

Arvosana 1-5

14. Voidaan lisätä halutuille tuotteille tietoja jälkikäteen. Esim IP44 luokitus märkätiloihin.

Merkitse vain yksi soikio.

	1	2	3	4	5	
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

15. Voidaan lisätä piirretyille tuotteille omat ID-koodit. Voidaan yksilöidä tuotteet.

Merkitse vain yksi soikio.

	1	2	3	4	5	
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

16. Datarasioiden numeroinnin automatisointi.

Merkitse vain yksi soikio.

	1	2	3	4	5	
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

17. Kerro ideoistasi mitä haluaisit automatisoida.

Määrälaskennan tekeminen yleissuunnitteluvaiheessa, kun on suunniteltu vain mallitilat. Tavoite, ettei tarvitse suunnitella kaikkia tiloja kuin vasta toteutussuunnitteluvaiheessa.

Arvosana
1-5

18. Tilojen massoittelu määrälaskennan tekemistä varten. Ohjelmaan listataan halutut tuotteet ja määrät sekä valitaan tilatyyppe (Space/Room). Ohjelma piirtää tuotteet keskelle huonetta. Voidaan tehdä määrälaskenta (Bill of materials) mallista, jolloin saadaan tuotteiden kokonaismäärät. Tasopiirustuksesta näkee tyhjät huoneet.

Merkitse vain yksi soikio.

	1	2	3	4	5	
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

19. Revitiin annetaan tieto, mitkä tilat ovat mallitiloja. Revit tekee määrälaskennan, jossa pisteiden määrät on kerrottu vastaavien tilojen määrillä. Määrälaskennan lisäksi tulee listaus tiloista, joissa ei yhtään tuotetta (laadun tarkistus)

Merkitse vain yksi soikio.

	1	2	3	4	5	
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

20. Määrälaskennassa voitaisiin hyödyntää aikaisemmin tehtyjä kohteita. Esitetään mallihuoneet valmiista kohteesta. Revit tekee määrälaskennan, jossa pisteiden määrät on kerrottu vastaavien tilojen määrillä. Määrälaskennan lisäksi tulee listaus tiloista, joissa ei yhtään tuotetta (laadun tarkistus)

Ongelmana on eri lailla nimetyt tilat ja muuttunut Dataset

Merkitse vain yksi soikio.

	1	2	3	4	5	
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

21. Kerro ideoistasi miten määrälaskentaa voisi parantaa.

Suunnittelun automatisoidut laaduntarkistukset. Merkitse mielestäsi hyödylliset tarkistukset

Arvosana
1-5

22. Ohjelma näyttää sähköhylyt tai tuotteet, jotka ovat jonkin tietyn korkeuden alapuolella tai yläpuolella. esim punaisella sähköhylyt, jotka alle 2300

Merkitse vain yksi soikio.

1	2	3	4	5		
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

23. Ohjelma näyttää tuotteet, jotka ovat alakaton alapuolella. (Esim valaisimet)

Merkitse vain yksi soikio.

1	2	3	4	5		
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

24. Ohjelma näyttää IV-poistoventtiilit, joiden 2 metrin etäisyydellä ei ole paloilmainta.

Merkitse vain yksi soikio.

1	2	3	4	5		
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

25. Listaa huoneet poista puuttuu jokin tuote. (Esim näytä tilat, joissa ei paloilmaisinta)

Merkitse vain yksi soikio.

	1	2	3	4	5	
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

26. Listaa huoneet, joissa enemmän tai vähemmän kuin haluttu määrä. (Esim näytä tilat, joissa yli 4 paloilmaisinta)

Merkitse vain yksi soikio.

	1	2	3	4	5	
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

27. Listaa tilat, joissa valaistuksen neliöteho on jonkin alle. (Esim valaistusteho alle 3W/m²)

Merkitse vain yksi soikio.

	1	2	3	4	5	
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

28. Schedule-taulukkojen automaattinen tekeminen sovitulla tiedoilla. (Esim Johtoteiden pituudet / valaisimien määrät kerroksittain / keskuskien määrät ja luettelot)

Merkitse vain yksi soikio.

	1	2	3	4	5	
Hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Hyödyllinen

29. Omia ideoita?

Vastaajan tietoja.

Täyttö vapaaehtoinen

30. Kuinka usein käytät Revitiä suunnittelussa

Merkitse vain yksi soikio.

- Päivittäin
- Viikoittain
- Kuukausittain
- Harvoin, joskus kokeillut
- En ollenkaan

31. Suunnittelukokemus

Merkitse vain yksi soikio.

- 1-3 vuotta
- 4-7 vuotta
- 8-11 vuotta
- yli 12 vuotta

Google ei ole luonut tai hyväksynyt tätä sisältöä.

Google Forms

Sisällys

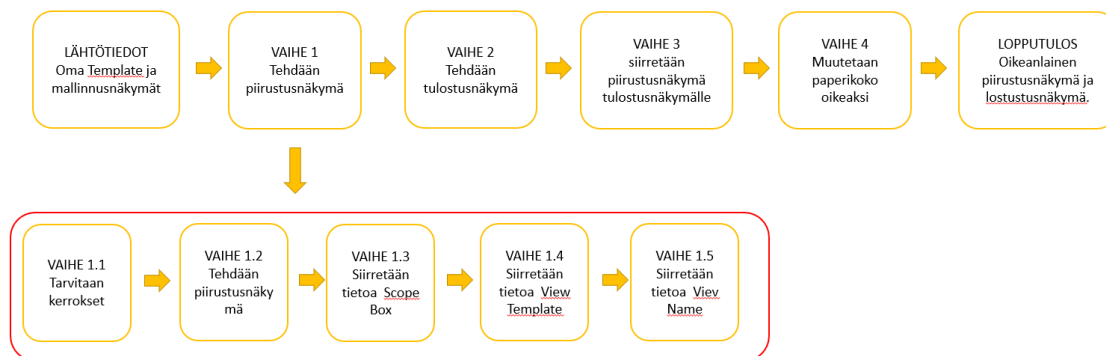
1	Skripti: Piirustus- ja tulostusnäkyvien automaattinen luominen	2
2	Skripti: Määrälaskenta mallitilojen avulla	9
2.1	Määrälaskenta mallitilojen avulla Schedule-taulukolla	9
2.2	Määrälaskenta mallitilojen avulla ilman Schedule-taulukkoa.....	19
3	Skripti: Elementtien automaattinen mallintaminen muiden suunnittelualojen tietomallin perusteella	25
3.1	Tietomallin tutkiminen etukäteen.....	25
3.2	Peilivalaisimen mallintaminen arkkitehdin peilikaapin perusteella	27
3.3	Moottoripellin mallintaminen LVI:n palopellin mukaisesti.....	33
4	Skripti: Elementtien asennuskorkeuden tarkastaminen ja muuttaminen alakaton perusteella.....	36
5	Skripti: Puuttuvien elementtien listaaminen automaattisesti	48

1 Skripti: Piirustus- ja tulostusnäkymien automaattinen luominen

Näkymien automaattisella luomisella on tarkoitus nopeuttaa toistoa vaativa toimenpide, jossa luodaan ensiksi tulostusnäkyvä, sen jälkeen tulostussivu ja siirretään tulostusnäkyvä sivulle, jonka jälkeen muutetaan paperikoko oikeaksi.

Lähdetään rakentamaan skriptiä muiden esimerkkien pohjalta. YouTubeista löytyy opetusvideoita ja Dynamo-foorumilta ohjeita näkymien luomisesta Dynamolla. Mikään ei suoraan vaikuta olevan riittävän toimiva meidän käyttöömme. Yhdistetään ohjeita ja pyritään luomaan helppokäyttöinen ja nopea skripti.

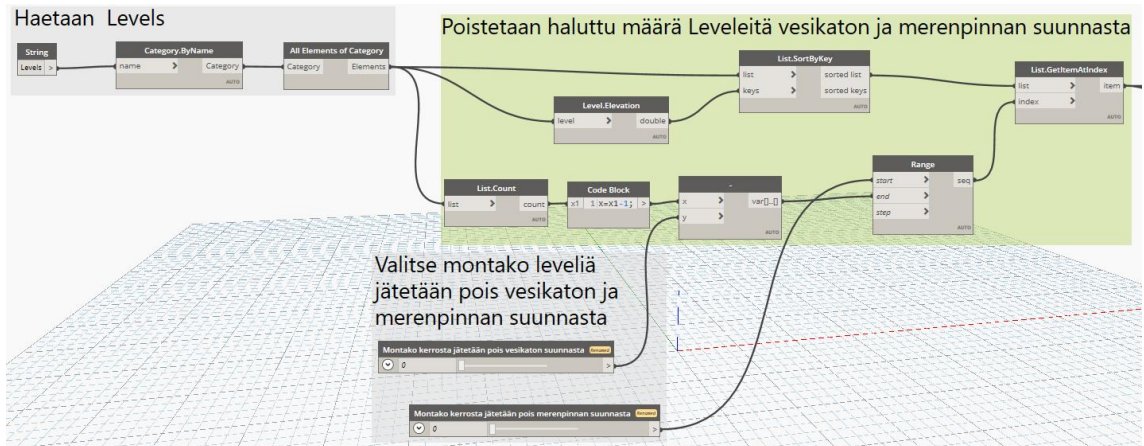
Esimerkkien perusteella otetaan tavoitteeksi luoda skripti, johon syötetään Dynamo-Playerillä haluttu Template, piirustuksen numero ja nimi. Skripti luo automaattisesti kaikista kerroksista oikealla Templatella piirustusnäkymän ja tulostusnäkymän sekä siirtää piirustusnäkymän tulostusnäkymäille ja lisäksi määrittelee sivun koon oikein. Kuvassa 51 on esitelty prosessikaavio, jossa purettu eri osiin. Samalla huomataan, että eri osat jakautuvat vielä pienempiin osiin. Prosessi kaaviossa esitetty Vaiheen 1 eri osat 1.1 – 1.5s.



Kuva 51. Prosessikaavio sivujen automaattisesta luomisesta

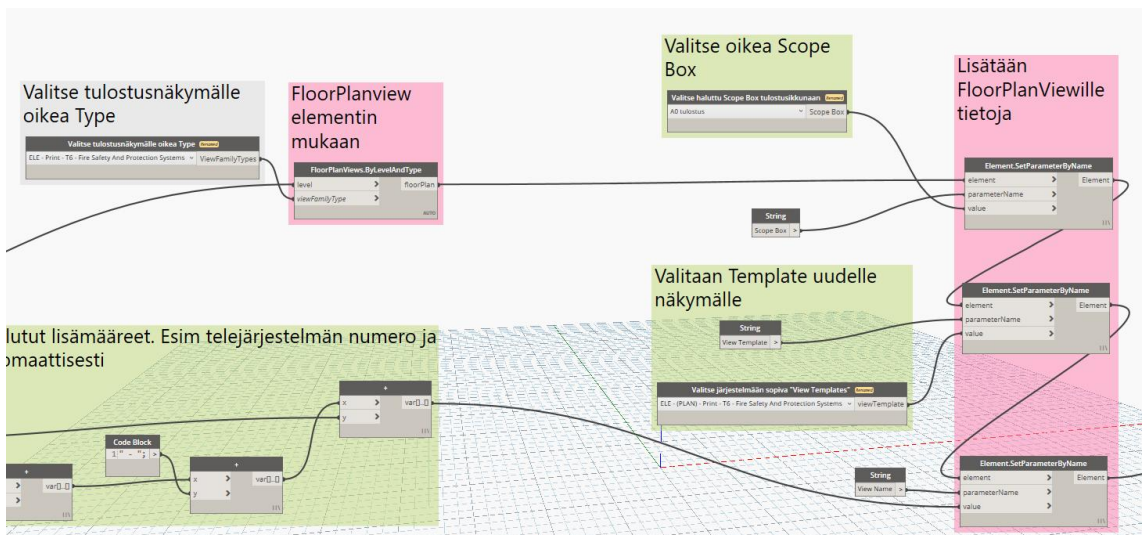
Aluksi määritellään piirustusnäkyvä, joka tehdään solmulla FloorPlanView.ByLevelAndType. Solmu vaatii levelit projektista, jotka saadaan haettua esimerkiksi kuvan xx mukaisella tavalla, sekä valitaan haluttu viewFamilyType. Kuvassa 52 esitetty FloorPlanView.ByLevelAndType-solmu kuvan vasemmassa reunassa. Tehdään lisäksi suodatus, jossa voidaan poistaa haluttu määrä kerroksia alhaalta

ja ylhäältä päin. Tämän avulla voidaan estää sivujen luominen merenpintaan tai vesikattoon, jos sellaiset levelit on määritetty.

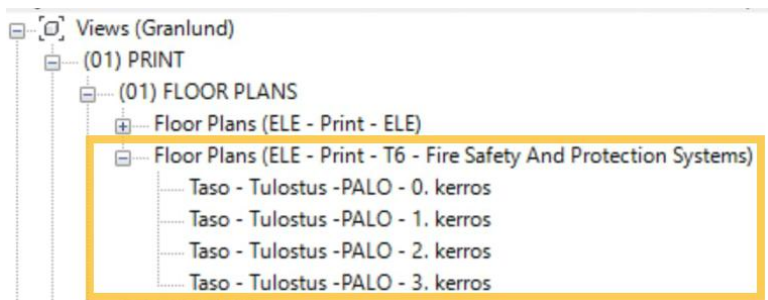


Kuva 52. Skriptin osa, jossa haetaan Levels-tiedot.

FloorPlanView näkymiä muokataan antamalla niille lisätietoa parametreihin. Kuvasssa 53 on esitetty, kuinka lisätään tiedot Scope Box, View Template ja View Name. Näillä tiedoilla saadaan luotua piirustusnäkyä, johon voidaan tehdä kaapelointi ja Tagit. View Template asetus määrittelee oikean näkymän kuvaan. Kuvasssa 54 on esitetty miltä näyttää piirustusasetukset Revitin Project Browserissa.

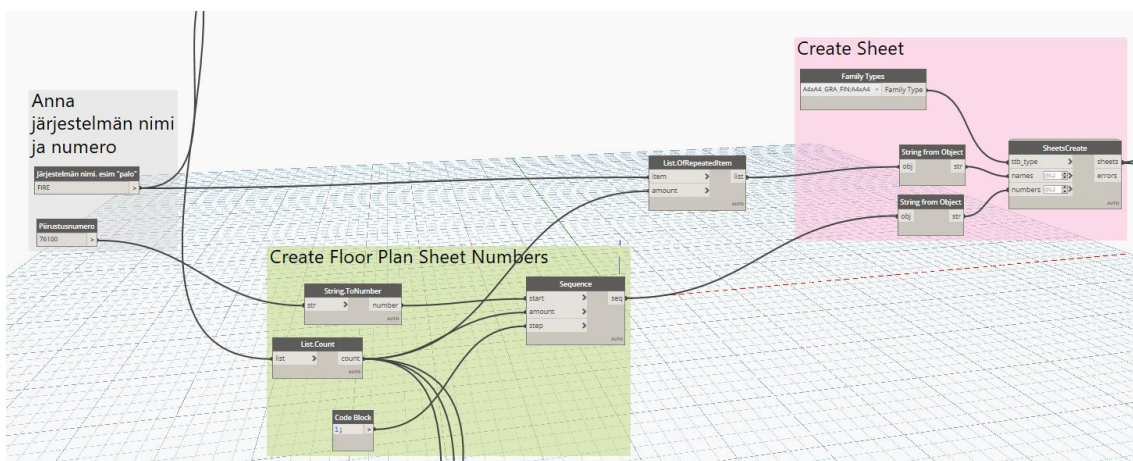


Kuva 53. piirustusnäkyjen luonti ja tietojen automaattinen täyttö



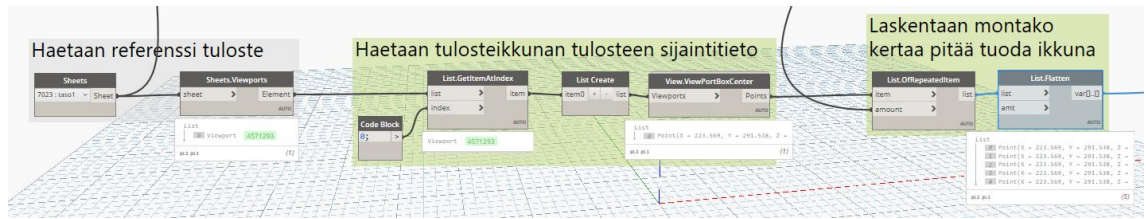
Kuva 54. Automaattisesti luodut piirustusnäkömät.

Tämän jälkeen rakennetaan tulostussivujen (Sheets) skriptin osa. Käytetään solmua SheetsCreate, joka tarvitsee sisääntuloiksi Titleblock Family Typen, nimen ja numeron. Piirustusnumeron on oltava yksilöllinen, joten rakennetaan skriptin osa, jossa tehdään lista, jossa jokainen numero on yhden suurempi kuin edellinen. Piirustuksen nimenä käytetään samaa kaikille. Kuvassa 55 on esitetty skriptin osa.

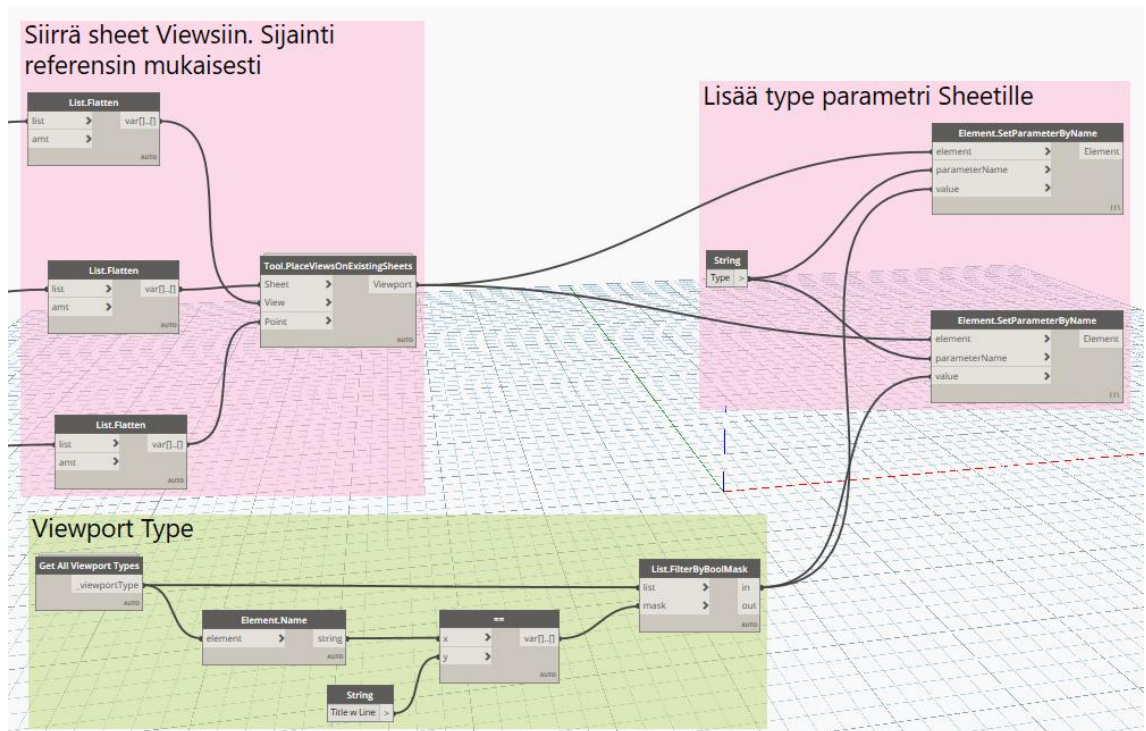


Kuva 55. Skriptin osa, jossa luodaan tulostusnäkömä (Sheets)

Tämän jälkeen on siirrettävä piirustusnäkömä tulostusnäkömälle. Se voidaan toteuttaa solmulla Tool.PlaceViewOnExistingSheets. Solmu tarvitsee lähtötiedoiksi tulostusnäkömän (Sheet), piirustusnäkömän (View) sekä pisteen, johon sijoitetaan piirustusnäkömä. Tulostus- ja piirustusnäkömät saadaan johdettua suoraan edellisistä solmuista. Sijaintitieto (Point) vaatii enemmän työtä, jotta saadaan se oikein. On muodostettava yksi mallipiirustus oikein luotuna, jossa paperikoot yms. ovat oikein. Käytetään sitä referenssitietona, josta saadaan oikea sijaintipiste. Kuvassa 56 on esitetty referenssi pisteen hakeminen ja kuvassa 57 on esitetty yhdistäminen sekä Type parametrin lisääminen.



Kuva 56. Skriptin osa, jossa haetaan piirustusnäkömön sijaintitieto.



Kuva 57. Skriptin osa, jossa siirretään piirustusnäkömön tulostusnäkömälle

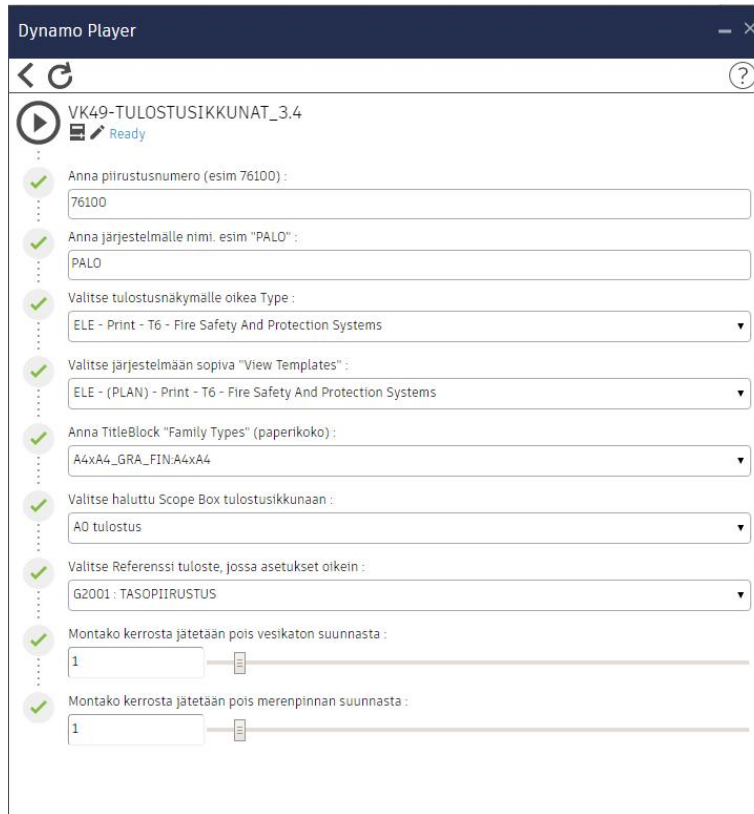
Tämän jälkeen tarvitsee vielä muuttaa paperikoko oikeaksi. Haetaan referenssi kuvan TitleBlockista paperikoon kerrannaiset ja siirretään ne luotavien tulostusnäkömien parametreihin. Kuvassa 58 on esitetty skriptin osa.



Kuva 60. Uusi tulostusnäkyvä, jossa piirustusnäkyvän sijainti vielä väärin, mutta muuten oikein.

Dynamo-Playerin käyttöliittymän tekeminen

Määritellään Dynamo-Playeriä varten sisäänmenot tarvittavista solmuista. Kuvassa 61 on esitetty Playerin näkyvä, josta näkee valitut sisäänmenot. Tämän työkalun avulla on nopea luoda piirustusnäkyvät ja tulostusnäkyvät. Mutta vielä jää käsin tehtäväksi siirtää näkyvä oikeaan kohtaan. Skriptiin on mahdollista luoda lisää kohtia, joilla lisätään nimiöön kopioituvien parametrien tietoja.



Kuva 61. Dynamo-Playerin näkymä, jolla luodaan piirustusnäkyvät ja tulostusnäkyvät yhdellä kertaa

2 Skripti: Määrälaskenta mallitilojen avulla

Skripti mahdollistaa kattavan määrälaskennan tekemisen mallitilojen avulla. Skriptin toimimiseksi on jokainen eriniminen tila suunniteltava kertaalleen. Ohjelma kertoo mallihuoneiden tuotteiden määrät saman nimisten tilojen määrällä sekä toisena vaihtoehtona tilojen pinta-alojen suhteella. Skriptiä voidaan käyttää esimerkiksi, kun halutaan yleissuunnitteluvaiheen jälkeen tehdä määrälaskenta, jota voidaan hyödyntää kustannusten määrittelyissä tai jos halutaan tietää eri järjestelmien pisteiden määrät mitoitusta varten. Skripti luodaan ensin toimimaan Revitin Schedule-taulukoiden avulla. Sen jälkeen skripti muutetaan toimimaan ilman Schedule-taulukoita.

2.1 Määrälaskenta mallitilojen avulla Schedule-taulukolla

Aluksi luodaan Schedule-taulukot. Ensimmäiseen taulukkoon pitää saada mallitilojen elementit. Tämä onnistuu, kun Revitin Space-elementin jollekin parametrimille kirjoitetaan tekstiä, jota suodatetaan Schedulelessa. Päätettiin kirjoittaa Comments-parametriin "MALLIHUONE". Kuvassa 1 on esitetty mallihuoneiden Schedule-taulukko. Toiseen taulukkoon pitää saada erinimisten tilojen määrät. Tämä onnistuu hakemalla taulukkoon Space name ja Count. Suodatetaan Schedule-taulukolla tietoa ja lasketaan samannimisten tilojen määrät. Kuvassa 2 on esitetty erinimisten mallitilojen määrät.

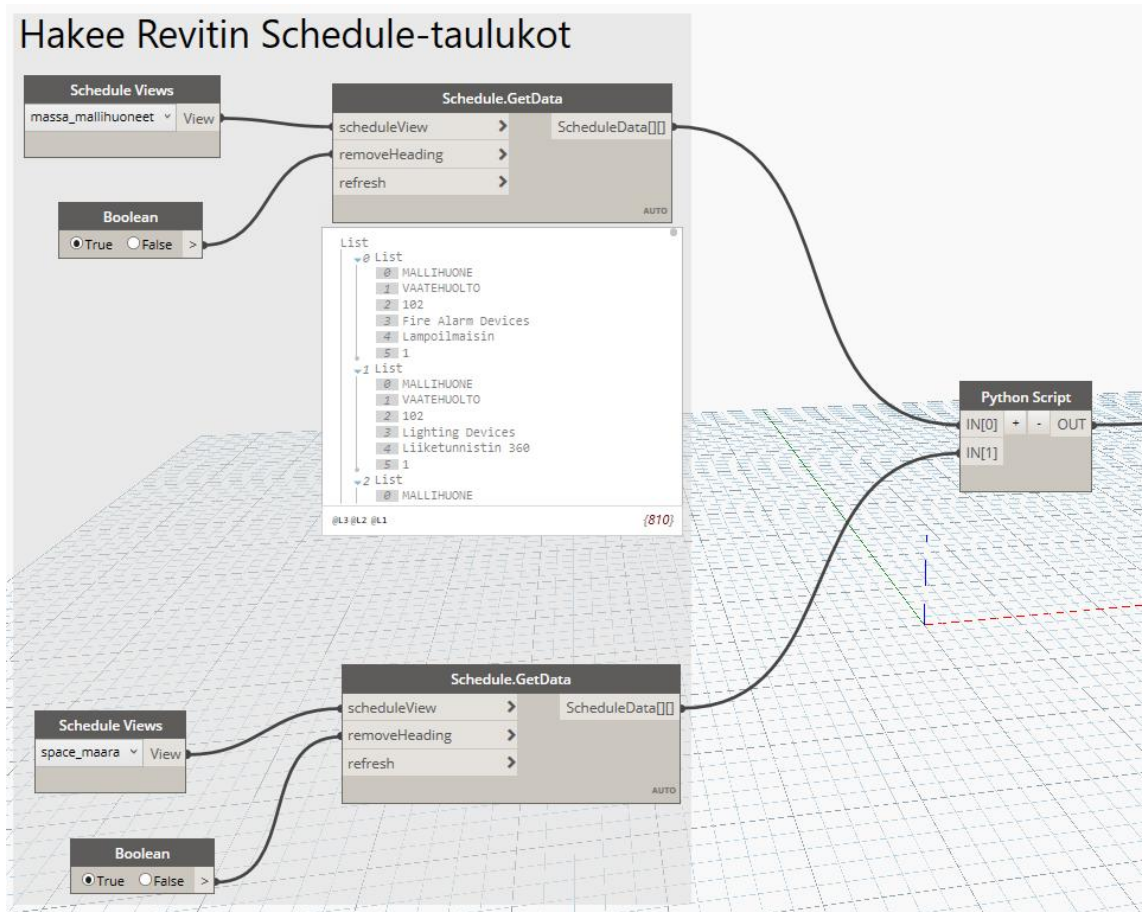
<massa_mallihuoneet>					
A	B	C	D	E	F
Space: Comments	Space: Name	Space: Number	Category	Family	Count
MALLIHUONE	VAATEHUOLTO	102	Fire Alarm Devices	Lampoilmaisin	1
MALLIHUONE	VAATEHUOLTO	102	Lighting Devices	Liiketunnistin 360	1
MALLIHUONE	VAATEHUOLTO	102	Security Devices	Liiketunnistin-katto	1
MALLIHUONE	VAATEHUOLTO	102	Electrical Fixtures	PISTORASIA-UPPO	3
MALLIHUONE	VAATEHUOLTO	102	Lighting Fixtures	siivous, 600x600, O	2
MALLIHUONE	VAATEHUOLTO	102	Electrical Fixtures	TUPLAPISTORASIA-PI	4
MALLIHUONE	VAATEHUOLTO	102	Electrical Fixtures	Turvakytkin 16A	1
MALLIHUONE	VAATEHUOLTO	102	Lighting Fixtures	työtason led-profiili	2
MALLIHUONE	NEUVOTTELU	104	Lighting Fixtures	600x600 MP paneel	2
MALLIHUONE	NEUVOTTELU	104	Data Devices	Data socket-2	1
MALLIHUONE	NEUVOTTELU	104	Electrical Fixtures	Floor box 244x244	1
MALLIHUONE	NEUVOTTELU	104	Electrical Fixtures	Kojerasia, umpipeit	3
MALLIHUONE	NEUVOTTELU	104	Lighting Devices	Kytkin 1-nap, uppo	1
MALLIHUONE	NEUVOTTELU	104	Lighting Devices	Liiketunnistin 360	1
MALLIHUONE	NEUVOTTELU	104	Security Devices	Liiketunnistin-katto	1
MALLIHUONE	NEUVOTTELU	104	Lighting Devices	Painike1uppo	1
MALLIHUONE	NEUVOTTELU	104	Electrical Fixtures	PISTORASIA-UPPO	1
MALLIHUONE	NEUVOTTELU	104	Electrical Fixtures	Recessed socket ho	9
MALLIHUONE	NEUVOTTELU	104	Fire Alarm Devices	Savuilmaisain	1
MALLIHUONE	NEUVOTTELU	104	Fire Alarm Devices	Savuilmaisain, Välitil	1
MALLIHUONE	NEUVOTTELU	104	Electrical Fixtures	TUPLAPISTORASIA-PI	1
MALLIHUONE	NEUVOTTELU	104	Data Devices	Yleiskaapelointiras	2
MALLIHUONE	TYÖHUONE	105	Lighting Fixtures	600x600 MP paneel	2
MALLIHUONE	TYÖHUONE	105	Data Devices	Data socket-2	1
MALLIHUONE	TYÖHUONE	105	Electrical Fixtures	Kojerasia, umpipeit	1
MALLIHUONE	TYÖHUONE	105	Lighting Devices	Kytkin 1-nap, uppo	1
MALLIHUONE	TYÖHUONE	105	Lighting Devices	Liiketunnistin 360	1
MALLIHUONE	TYÖHUONE	105	Security Devices	Liiketunnistin-katto	1

Kuva 1. Mallitilojen määrälaskenta-taulukko

<space_maara>	
A	B
Name	Count
ET	1
KEITTIÖ	1
KUIVAUS	3
KURATEINEN	3
KÄYTÄVÄ	3
LE-WC/HLK	1
LEIKKIHUONE	1
LEIKKIHUONE 12hig	1
LEIKKIHUONE 21hig	1
LEPOHUONE	1
LEPOHUONE 12hig	1
LEPOHUONE 21hig	1
NEUVOTTELU	1
PH	1
PIENRYHMÄ	1
PKH. A	1
PKH. B	1
SIIVOUS	1
SÄH	3
TAUKO	1
TYÖHUONE	2
VAATEHUOLTO	1
VAATENAUH	1
VARASTO	3
WC	3
WC A	1
WC B	1
WC/HLK	1

Kuva 2. Mallitilojen määrälaskenta-taulukko

Kuvassa 3 on esitetty, kuinka Schedule-taulukot haetaan Dynamoon Schedule.GetData solmulla. Tämän jälkeen taulukon tiedot siirretään Python Skriptiin. Schedule-taulukoiden tiedoissa on huomioitava, että ne eivät ole numeroitietoa vaan kirjaimia. Dynamossa sekä Pythonissa on tapa, jolla voidaan muuttaa tekstitieto numeroarvoksi.

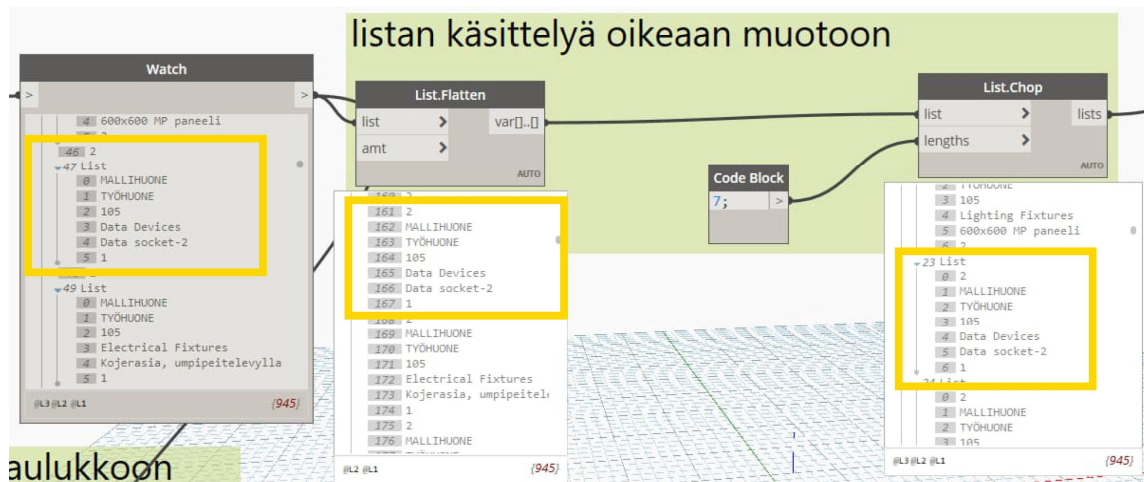


Kuva 3. Skriptin osa Schedule-taulukon hakemisesta

Listojen käsittely

Listojen käsittely Dynamo-solmuilla on monimutkaista, joten listojen käsittely suoritetaan Dynamon Python-Skriptin avulla. Koodissa listoja käsitellään for-silmukalla. Jos tilan nimi on sama molemmissa, kerrotaan pisteiden määrä tilan määrällä.

Python-koodista tulevan listan rakenne on monimutkainen. Listaa voidaan käsitellä erilaisilla tavoilla. Kuvassa 4 on esitetty tapa, jossa moniportainen lista muokataan ensin yhdeksi listaksi, jonka jälkeen takaisin listan listaksi seitsemän jalla. Tällöin saadaan aikaan lista, joka sisältää listoja, jossa on mallihuoneen määrä kerrottuna tilojen määrä

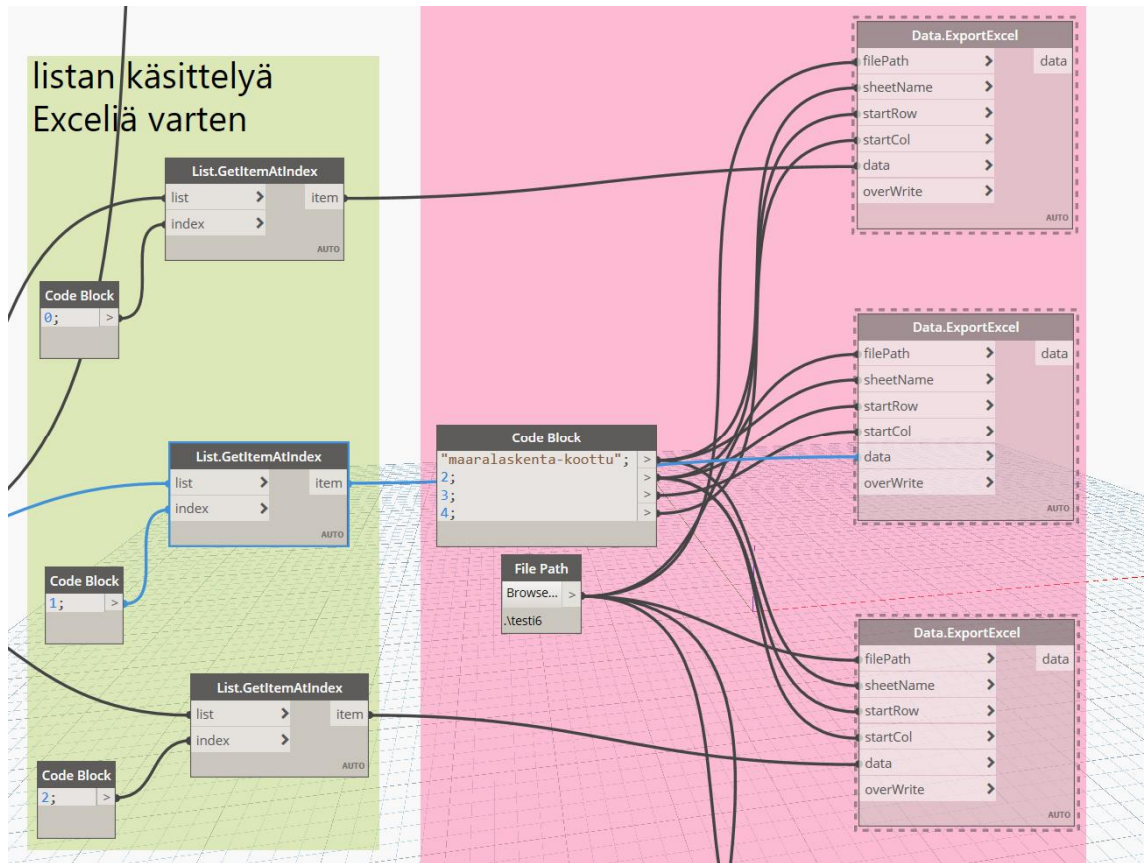


Kuva 4. Skriptin osa listojen muokkaamisesta haluttuun muotoon

Tämän jälkeen lasketaan Python-koodilla saman nimisten tuotteiden kokonaismäärä. Koodi vertailee tuotteen nimeä ja laskee yhteen saman nimiset tuotteet.

Tuloksen käsittely

Lopuksi tiedot siirretään Excel-tiedostoon. Kuvassa 5 on esitetty skriptin osa, jolla voidaan siirtää tietoa. Data.ExportExcel-solmu tarvitsee tiedoiksi tiedostopolun ja nimen, taulukon nimen, rivin ja sarakkeen, josta aloitetaan sekä datan eli tässä tapauksessa listan.



Kuva 5. Skriptin osa tietojen viemisestä Excel-taulukkoon

Lisätään Excel-taulukkoon otsikko ja muotoilu. Rivien ja sarakkeiden sijainteihin ei saa koskea, koska Dynamo alkaa kirjoittamaan päivittäessä listaa 2 riville ja 2 sarakkeeseen. Excel-taulukkoon voidaan päivittää muutokset Dynamon Run -painiketta painamalla milloin vain. Kuvassa 6 on esitetty tämänhetkinen lopputulos.

MÄÄRÄLASKENTA MALLITILOJEN AVULLA		
Category	Family	Count
Fire Alarm Devices	Lampoilmaisin	8
Communication Devices	Ovikello	3
Communication Devices	Summeri	3
Data Devices	Data socket-2	13
Data Devices	Yleiskaapelointirasia, 2-os	4
Data Devices	TE - GRA_RAU_TE	3
Data Devices	KV-OVI	3
Electrical Equipment	muuntaja sähköhana	3
Electrical Equipment	AV-kaappi 600x600	3
Electrical Fixtures	PISTORASIA-UPPO	19
Electrical Fixtures	TUPLAPISTORASIA-PINTA	21
Electrical Fixtures	Turvakytkin 16A	1
Electrical Fixtures	Floor box 244x244	1
Electrical Fixtures	Kojerasia, umpipeitelevyllä	6
Electrical Fixtures	Recessed socket horizontal-2-100x82x17	72
Electrical Fixtures	Liitántarasia	6
Electrical Fixtures	PISTORASIA-PINTA	4
Fire Alarm Devices	Savuilmaisin	20
Fire Alarm Devices	Savuilmaisin, Välitila, merkkilampulla	22
Fire Alarm Devices	Sireeni	3
Fire Alarm Devices	Osoiteyksikkö1	3
Fire Alarm Devices	Palopainike	3
Lighting Devices	Liiketunnistin 360	25
Lighting Devices	Kytkin 1-nap, uppo	6
Lighting Devices	Painike1uppo	4
Lighting Devices	Säädin, uppo	3

Kuva 6. Valmis Excel-taulukko määrälaskennasta.

Tehdään Excel-taulukkoon toinen sivu, johon listataan tuotteen erinimisten tilojen perusteella. Muodostetaan Excelissä tästä taulukosta Pivot-taulukko, jonka avulla voidaan tutkia pisteitä määriä tiloittain. Kuvassa 7 on esitetty Pivot-taulukko.

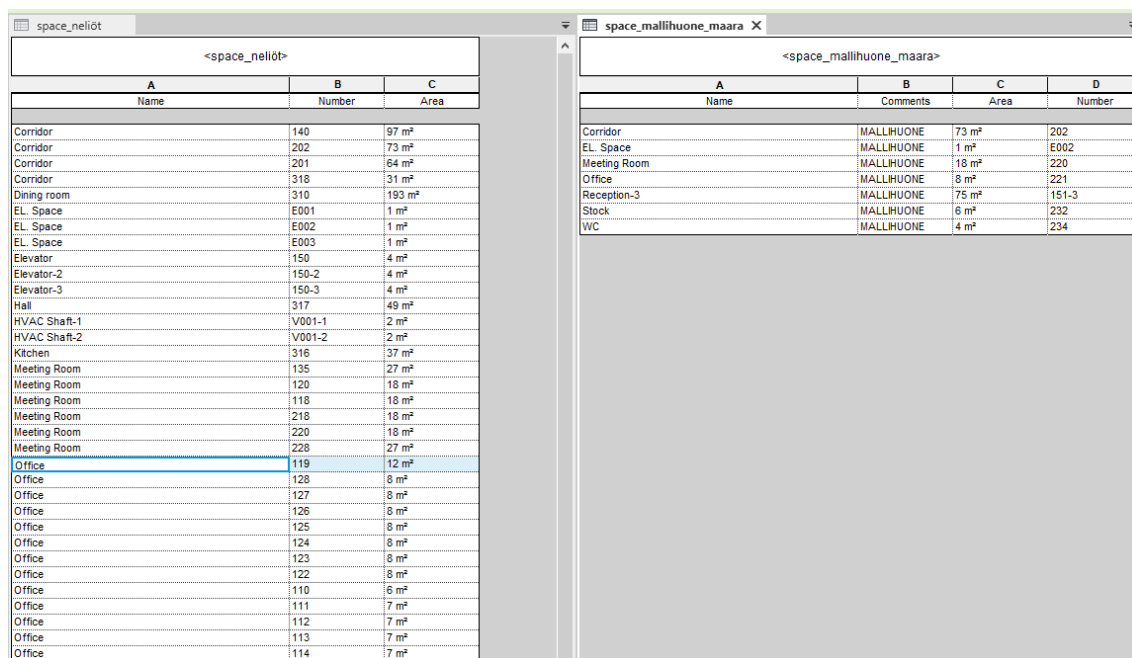
	A	B	C	D	E	F	G	H	I
1									
2									
3	Summa / kokonaismaara	Sarakeotsikot							
4	Riviotikot	Corridor	EL. Space	Meeting Room	Office	Reception-3	Stock	WC	Kaikki yhteensä
5	__DATA-SOCKET-2-0001			22	64			12	98
6	Clock-0001			8					8
7	COVERED-SOCKET-2-0001	25	3	84	183	2	3	21	321
8	EXIT-HERE-WALL	8		8		1		12	29
9	FXXDD-1	41	3	28	64	9	3		148
10	RECESSED-CIRCLE-DOWN-1x32							12	12
11	SMOKE-DETECTOR-0001			8					8
12	Speaker-0001			8	33			12	53
13	Switch-with-one-push-button-0001			14	64				78
14	Kaikki yhteensä	74	6	180	408	12	6	69	755

Kuva 7. Pivot-taulukko määrälaskennan pisteistä

Määrälaskennan kehittäminen pinta-alojen suhteessa

Joissakin tapauksissa on järkevämpää tehdä määrälaskenta vertailemalla tilojen pinta-aloja. Esimerkiksi, jos on paljon eri kokoisia tiloja, kuten käytäviä ja luokkahuoneita. Tehdään uusi skripti, jossa vertaillaan tiloja pinta-alojen suhteessa. Käytetään pohjana aikaisempaa skriptiä.

Muodostetaan kaksi Schedule-taulukkoa. Ensimmäisessä on mallitilojen pinta-alat ja toisessa on kaikkien tilojen pinta-alat. Kuvassa 8 on esitetty Schedule-taulukot. Python-skriptin avulla lasketaan kaikille tiloille kerroin, jolla voidaan kertoa mallihuoneen pistemäärät.



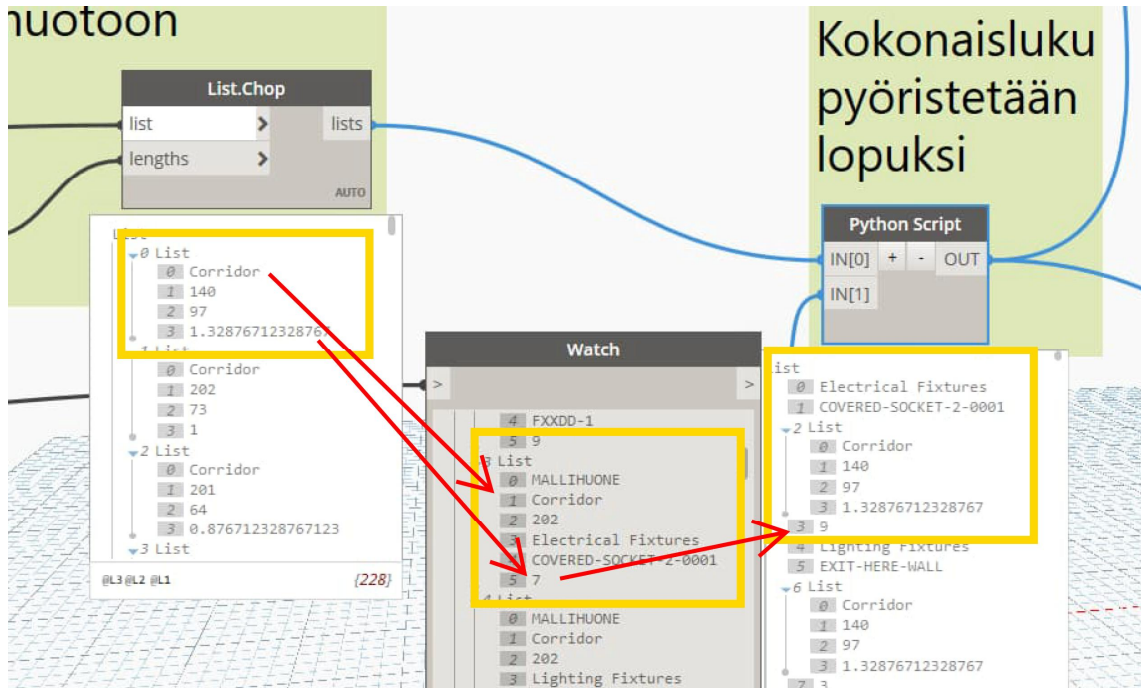
<space_neliöt>		
A	B	C
Name	Number	Area
Corridor	140	97 m²
Corridor	202	73 m²
Corridor	201	64 m²
Corridor	318	31 m²
Dining room	310	193 m²
EL Space	E001	1 m²
EL Space	E002	1 m²
EL Space	E003	1 m²
Elevator	150	4 m²
Elevator-2	150-2	4 m²
Elevator-3	150-3	4 m²
Hall	317	49 m²
HVAC Shaft-1	V001-1	2 m²
HVAC Shaft-2	V001-2	2 m²
Kitchen	316	37 m²
Meeting Room	135	27 m²
Meeting Room	120	18 m²
Meeting Room	118	18 m²
Meeting Room	218	18 m²
Meeting Room	220	18 m²
Meeting Room	228	27 m²
Office	119	12 m²
Office	128	8 m²
Office	127	8 m²
Office	126	8 m²
Office	125	8 m²
Office	124	8 m²
Office	123	8 m²
Office	122	8 m²
Office	110	6 m²
Office	111	7 m²
Office	112	7 m²
Office	113	7 m²
Office	114	7 m²

<space_mallihuone_maara>			
A	B	C	D
Name	Comments	Area	Number
Corridor	MALLHUONE	73 m²	202
EL Space	MALLHUONE	1 m²	E002
Meeting Room	MALLHUONE	18 m²	220
Office	MALLHUONE	8 m²	221
Reception-3	MALLHUONE	75 m²	151-3
Stock	MALLHUONE	6 m²	232
WC	MALLHUONE	4 m²	234

Kuva 8. Schedule-taulukot, joissa pinta-alat

Schedule-taulukoiden pinta-aloissa on huomioitava, että ne sisältä myös päätteen "m2" sekä tieto ei ole numeroita vaan kirjaimia. Tehdään Python-koodi, jolla poistetaan päätteet ja samalla muutetaan pinta-alat numeroarvoiksi.

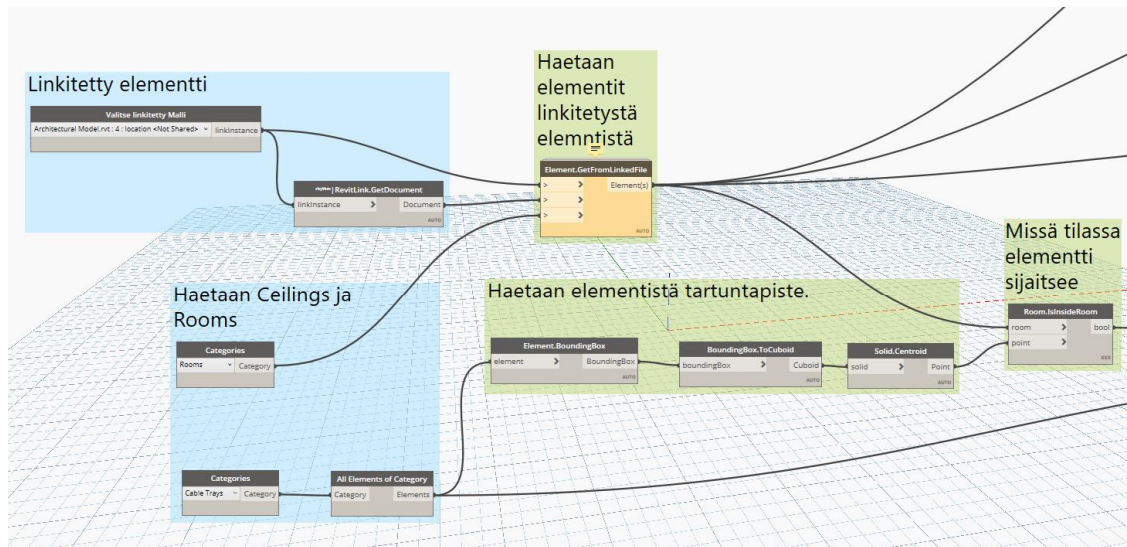
Seuraavaksi kerrotaan mallitilojen pisteiden määrät huoneiden pinta-alojen suhteen mukaisella kertoimella, joka pyöristetään lopuksi kokonaisluvuksi. Laskenta suoritetaan Python-koodilla. Kuvassa 9 on esitetty ajattelumallia. Tämän jälkeen tietoa käsitellään vastaavasti kuin edellisessä määrälaskennassakin.



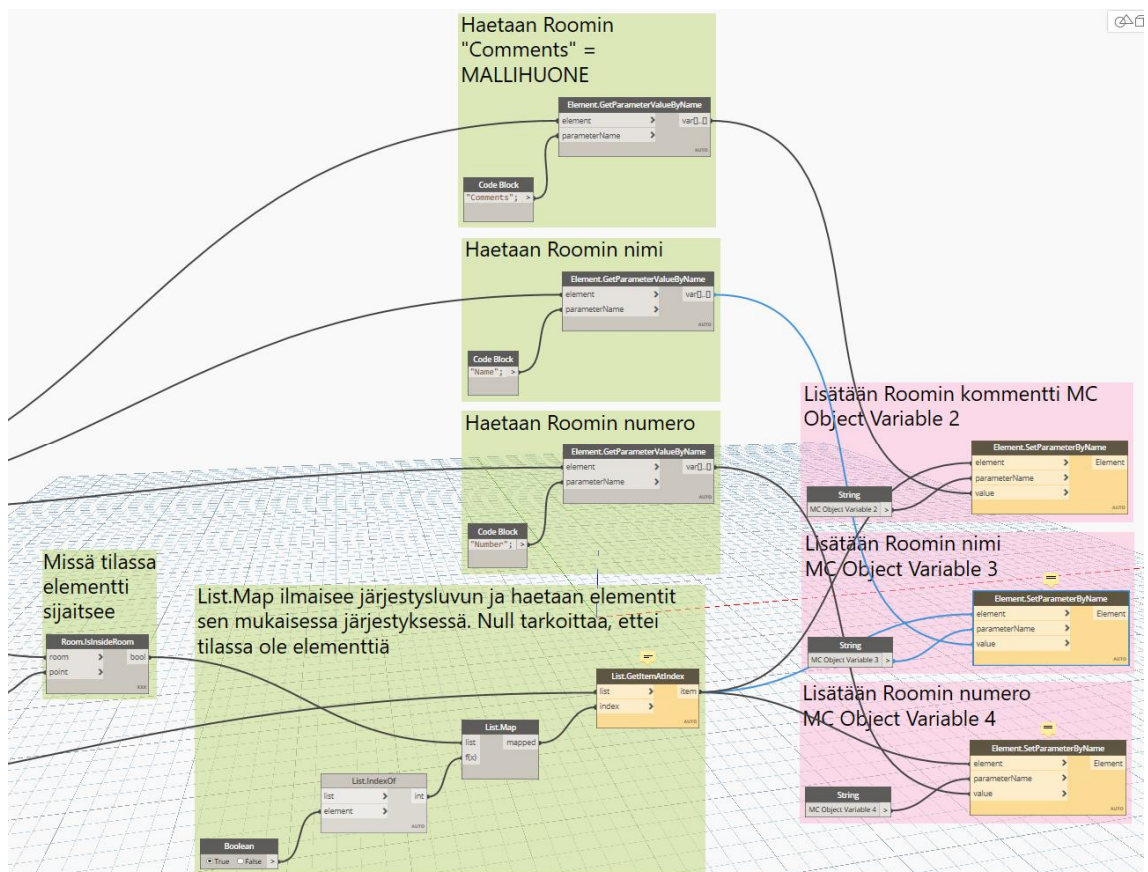
Kuva 9. Skriptin osa listoilla laskemista.

Määrälaskenta johtoteille

Johtotiet kuuluvat system familyihin ja niille ei tule samanlailla tilatietoa kuin Familyille. Tämä on ratkaistava aluksi, jotta voidaan kohdistaa johtotiet tietyllä tilalle. Tehdään skripti, jolla siirretään Space Name ja Number johtoteiden vapaille parametreille. Kuvassa 10 on esitetty skriptin alkuosa, jossa etsitään johtotie Roomista sen keskipisteen avulla käyttäen Room.IsInsideRoom-solmua. Solmu antaa ulos true/false -tietoa. Kuvassa 11 on esitetty, miten saadaan lajiteltua johtotiet samaan järjestykseen kuin Roomit true/false-tietojen avulla sekä miten Roomin tiedot viedään johtotie-elementtien parametreille. Roomin Name, Number ja Comments -parametrit viedään johtoteiden parametreille MC Object Variable 1, 2 ja 3. Tällöin saadaan siirrettyä johtoteille vastaavat tiedot kuin mallitilojen pisteillä ja voidaan muodostaa vastaavanlainen Schedule-taulukko, jota voidaan käyttää määrälaskennassa. Huomioitavaa, että "MALLIHUONE" teksti on kirjoitettava **arkkitehdin** tietomalliin.



Kuva 10. Skriptin osa, jossa etsitään mihin tilaan johtotie-elementti kuuluu



Kuva 11. Skriptin osa, jossa etsitään mihin tilaan johtotie-elementti kuuluu ja siirretään tilaelementin tietoja johtoteille.

Tutkitaan vietyä tietoa johtoteille. Huomataan, että piirtäessä pitkiä hyllyjä monen tilan ylitse, hyllyn sijainti määräytyy hyllyn keskikohtaan perusteella. Tällöin piirroksessa on huomioitava, että hylly on katkaistava mallitilojen kohdalla, jotta tieto siirtyy oikein ja saadaan määrälaskennassa oikeat määrät. Johtokanavien piirroksessa on myös huomioitava, että johtokanavat on katkaistava seinän kohdalla. Tämän jälkeen määrälaskenta mallitilojen avulla voidaan suorittaa vastaavilla tavoilla kuin pistemäisillä elementeillä. Dynamo- ja python-koodia on muokattava tarvittavilta osin. Kuvassa 12 on esitetty lopputulokseksi saatu taulukko, jossa eriteltynä johtotietyyppi, system, koko ja kokonaispituudet.

MÄÄRÄLASKENTA MALLITILOILLA JOHTOTEILLE			
Family and Type	System name	Size	Pituus mm
Cable Tray with Fittings: Cable ladder	Main distribution systems	500 mmx50 mm	203099
Cable Tray with Fittings: Cable tray	Emergency power systems	300 mmx50 mm	63404
Cable Tray with Fittings: Cable ladder	Main distribution systems		39600
Cable Tray with Fittings: Cable trunking	Main distribution systems	70 mmx150 mm	205769

Kuva 12. Määrälaskenta mallitiloilla johtoteillä valmis taulukko

Määrälaskenta johtoteille pinta-alojen suhteessa

Muodostetaan määrälaskenta mallitilojen avulla johtoteille pinta-alojen suhteessa. Käytetään samoja keinoja kuin aikaisemmissakin skripteissä. Lopputulokseksi saadaan taulukko määrästä, joiden kokonaismäärät ovat laskettu pinta-alojen suhteessa.

Määrälaskennan laaduntarkistus

Määrälaskenta laskee pisteet vain mallihuoneista, jolloin on tarkistettava, onko kaikki erinimiset tilat suunniteltu kertaalleen. Lisätään määrälaskentaskripteihin osio, jossa listataan tilat, joista ei ole yhtään nimetty mallihuoneeksi. Käytetään samoja Schedule-tilukoita hyödyksi. Ensimmäisestä taulukosta otetaan lista, jossa on kaikki mallihuoneeksi nimetyt tilat. Toisesta taulukosta, jossa on

eranimisten tilojen määrät, otetaan tilanimet. Tehdään Python-koodi, jolla poistetaan kaikkien eranimisten tilojen listasta mallihuoneiden nimet. Tällöin jäljelle jää lista tiloista, joista ei ole tehty mallihuonetta.

Jos joku tila on nimetty kahteen kertaan mallihuoneeksi, tulee kaksinkertainen määrä pisteitä. Lisätään Python-koodiin osa, jossa listataan tilat, jotka nimetty kahteen kertaan mallihuoneeksi. Viedään tieto samaan Excel-taulukkoon kuin määrälaskenta ja nimetään taulukon sivu ”puuttuvat tilat tai kahteen kertaan”.

Muodostuneen Excel-taulukon määrälaskennoista, joissa tuotteet on listattuna tiloittain, voidaan muodostaa Pivot-tilukko. Pivot-tilukon perusteella voidaan tutkia tarkemmin määriä tiloittain ja varmistua tietojen oikeellisuudesta.

Ongelmat

Yksi määrälaskennan ongelmista on oikein muodostetut Schedule-tilukot. Schedule-tilukoissa on oltava kaikki sarakkeet sekä niiden on oltava oikeassa järjestyksessä. Tilukot on lisäksi suodatettava ja lajiteltava oikealla tavalla, jotta määrälaskenta toimii. Muutoin skripti epäonnistuu tai tulee tyhjää tietoa listoihin.

Yritetään luoda skriptiä, jolla saadaan juuri oikeanlainen Schedule-tilukko. Skriptin luomisessa tulee ongelmia, koska sillä ei saada lajiteltua kuin kahta tietoa kerrallaan. Johtoteiden osalta ei saada muutettua arvoa, jolla lasketaan hyllyjen kokonaispituudet. Ei saada toimimaan Schedule-tilukon automaattista luontia halutulla tavalla.

2.2 Määrälaskenta mallitilojen avulla ilman Schedule-tilukkoa

Kehitetään määrälaskenta mallitilojen avulla -skriptiä poistamalla Schedule-tilukot. Muodostetaan vastaavat tiedot hakemalla elementit ja valitsemalla niistä

tarvittavat parametrit. Parametrien avulla voidaan muodostaa suodatuksia ja lajitteluita. Puretaan skriptin vaiheet osiin. Kuvassa 13 on esitetty prosessikaavio, miten Schedule-taulukon tiedot saadaan listattuna Dynamolla.

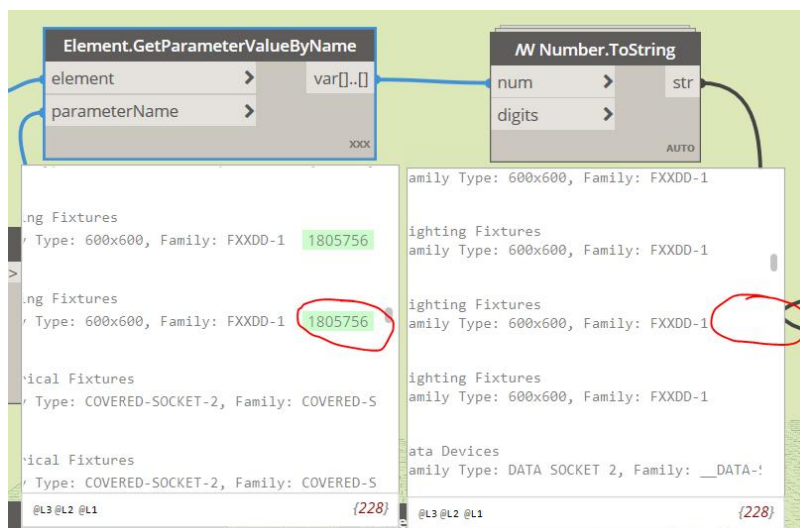


Kuva 13. Prosessikaavio Schedule-taulukon korvaaminen Dynamo-skriptillä

Mallihuoneiden pisteet

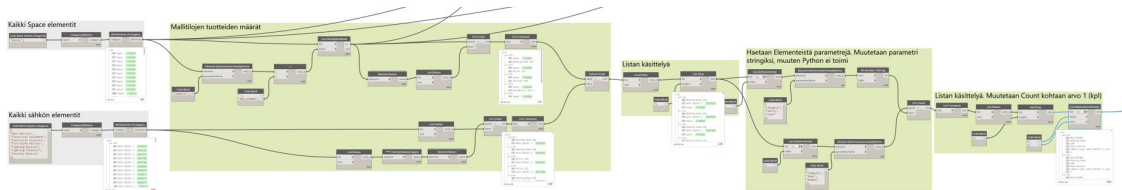
Haetaan mallista kaikki sähkösuunnittelun elementit. Haetaan elementistä tilanimi, johon se on asennettu ja muodostetaan niistä lista1. Haetaan Space-elementit ja suodatetaan niistä mallihuoneet. Tehdään niistä lista2, jossa on Space-elementti sekä sen nimi. Viedään listat Python-koodiin, jolla suodatetaan mallihuoneiden elementit. Sen jälkeen haetaan elementeistä parametrit, joita tarvitaan määrälaskennassa.

Aluksi oli ongelmia, kun Python-koodi ei käsittele halutulla tavalla elementtejä. Tämä ratkaistiin muuttamalla elementit tekstimuotoon. Tähän löytyy Number to String -solmu. Kuvassa 14 on esitetty miltä näyttää "aktiivinen" elementti ja se muutettuna tekstiksi.



Kuva 14. Skriptin osa, jossa elementti muutetaan tekstiksi.

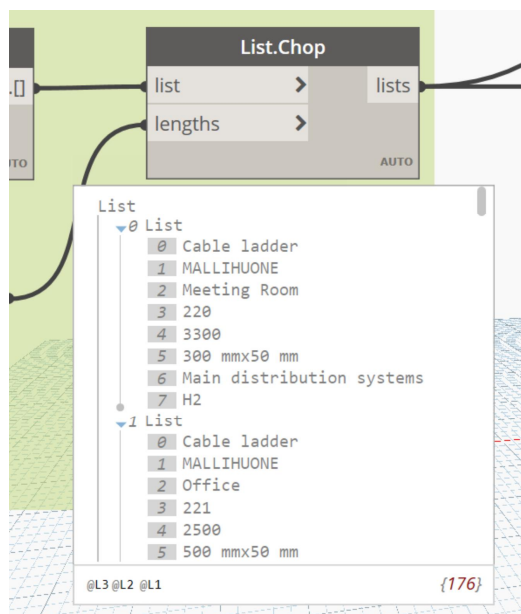
Mallihuoneiden pisteiden Schedule-taulukko voitiin korvata kuvan 15 mukaisella skriptin osalla.



Kuva 15. Skriptin osa mallihuoneiden pisteiden Schedule-taulukon korvaaminen.

Johtoteiden määrät mallitiloissa

Haetaan mallista kaikkien johtoteiden elementit. Haetaan elementeistä parametri, johon aikaisemmin oli siirretty tilaobjektin mallihuone tieto ja suodatetaan ne elementit mukaan. Haetaan näistä elementeistä samat parametrit, kuin aikaisemmin saatiin Schedule-taulukosta. Hyllyelementit muutetaan tekstiksi Element.Name -solmulla. Lopuksi muokataan listaa haluttuun muotoon. Kuvassa 16 on esitetty listan muoto, joka vastaa aikaisempaa Schedule-taulukkoa tietosisällöltään.



Kuva 16. Skriptillä muodostettu lista mallihuoneiden johtoteistä.

	A	B	C	D
1	Määrälaskenta mallitilojen avulla -skriptin pohja		Granlund Jyväskylä Oy	
2			Jaakko Kontunen	
3	sisällysluettelo			
4	110	pisteiden määrät suhteessa tilojen määrään		
5	120	pisteiden määrät suhteessa pinta-alojen määrään		
6	130	johtoteiden määrät suhteessa tilojen määrään		
7	140	johtoteiden määrät suhteessa pinta-alojen määrään		
8	210	pisteiden määrät tiloittain suhteessa tilojen määrään		
9	211	pivot-taulukko		
10	220	pisteiden määrät tiloittain suhteessa pinta-alojen määrään		
11	221	pivot-taulukko		
12	230	johtoteiden määrät tiloittain suhteessa tilojen määrään		
13	231	pivot-taulukko		
14	240	johtoteiden määrät tiloittain suhteessa pinta-alojen määrään		
15	241	pivot-taulukko		
16	300	Puuttuvat mallitilat ja kahteen kertaan		
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				
37				
38				
39				
40				
41				
42				

Kuva 19. Määrälaskenta mallitilojen avulla -skriptin luoman Excel-taulukon asiakirjaluettelo

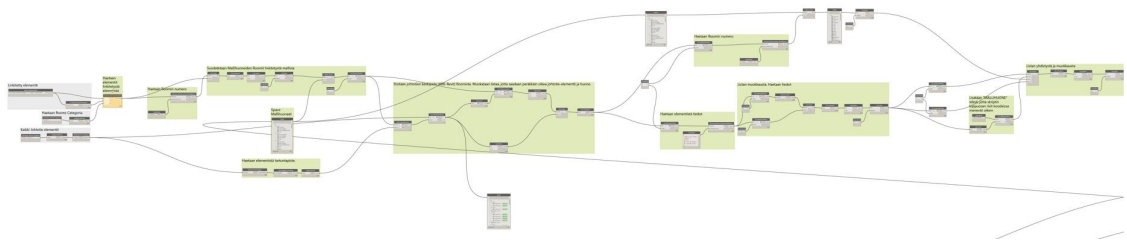
Skriptin kehittäminen, ettei tarvitse erikseen siirtää tilanimiä ja -numeroita johtotien parametreille

Kokeillaan voidaanko johtoteihin liittyvistä skripteistä poistaa kohta, jossa johtote-elementeille siirretään Room Name ja Number sekä "MALLIHUONE"-tieto erillisellä skriptillä. Tällöin voidaan poistaa ongelma, jossa uusille johtoteille ei tule tilatietoa, koska se pitää muistaa aina erikseen tehdä erillisellä skriptillä. Kopioi-

daan määrälaskentaskriptin alkuun tiedonsiirtoskripti. Muokataan sitä, että se hakee linkitetystä mallista tiedot. Tehdään skriptiin Python-koodi, jossa suodatetaan tilanumeroiden perusteella arkkitehdin mallista vain mallitilojen tilat.

Tämän jälkeen voidaan etsiä johtotiet, jotka sijaitsevat mallihuoneissa Room.IsInsideRoom-solmulla. Lopuksi suoritetaan listojen muokkauksia, että saadaan listat samaan muotoon kuin aikaisemmin. Lista ei haeta nimiä arkkitehdin mallista, vaan haetaan Spacen nimi, jotta nimet ovat yhtäläisiä omassa mallissa. Tämä sen takia, että voidaan nimetä halutessaan uudelleen tiloja ja poistaa niistä ylimääräisiä merkkejä. Kuvassa 20 on esitetty skriptin osa.

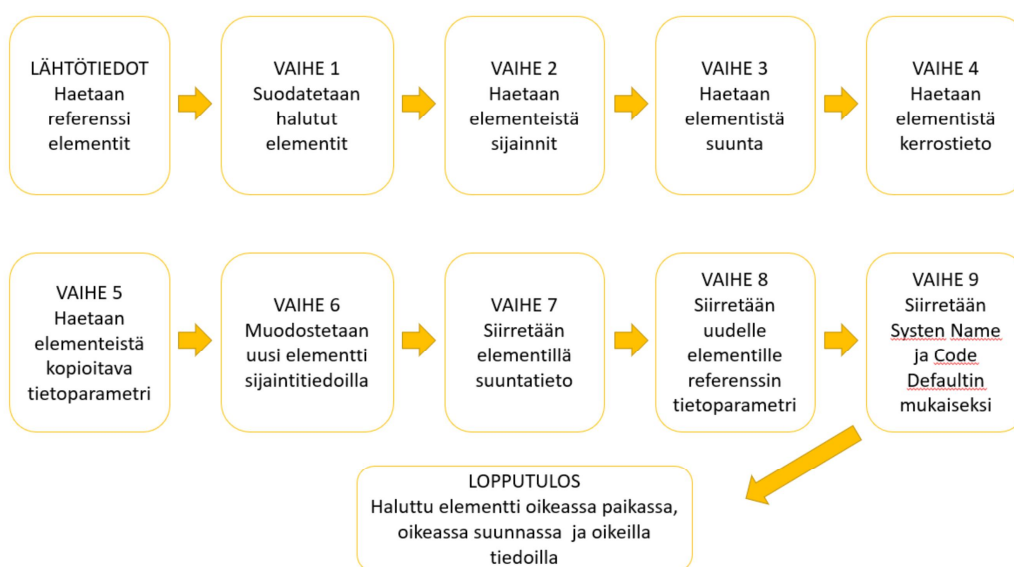
Saadaan skripti toimimaan ilman, että johtoteiden parametreille on erikseen siirrettä tilojen nimiä, numeroita ja "MALLIHUONE"-tekstiä. Haittapuolena skriptin pituus ja monimutkaisuus kasvaa sekä isoissa malleissa mahdollisesti käyttö hidastuu, koska skripti hakee nyt tietoa myös linkitetystä malleista.



Kuva 20. Skriptin osa, jossa etsitään mallitilojen johtotiet ja muokataan listoja haluttuun muotoon.

3 Skripti: Elementtien automaattinen mallintaminen muiden suunnittelu-alojen tietomallin perusteella

Elementtien automaattisella mallintamisella on tarkoituksena sijoittaa sähkösuunnittelun elementtejä malliin hyödyntäen muiden suunnittelualojen malleja. Käytetään skriptistä nimeä automaattipiirto. Piirron lisäksi on tarkoitus siirtää tietoa referenssi elementistä. Kuvassa 21 on esitetty prosessikaaviota ajatuksesta.



Kuva 21. Prosessikaavio elementtien automaattisesti mallintamisesta.

3.1 Tietomallin tutkiminen etukäteen

Ennen automaattipiirtoa on hyvä selvittää mallista mitä elementtejä löytyy ja montako, jotta voidaan tehdä päätös, onko tehokasta käyttää automatisointia hyödyksi. Tämä voidaan tehdä esimerkiksi Revitin Schedule-taulukoilla. Kuvassa 22 on esitetty arkkitehdin mallista vesikalusteisiin liittyvät tuotteet Schedule-taulukossa. Schedule-taulukoita voi tehdä ja muokata itse omien toiveiden mukaa. Peilikaappeja (4830: Gustavsberg ARTic 4830) on projektissa 11 kappaletta.

<Plumbing Fixture Schedule>			
A	B	C	D
Family	Family and Type	Level	Count
4830	4830: Gustavsberg ARTic 4830		11
M_Sink Bathroom_1	M_Sink Bathroom_1: 400 x 500mm		8
M_Sink Bathroom_2	M_Sink Bathroom_2: 650 x 500mm		3
M_WC Seat_2 with Cistern	M_WC Seat_2 with Cistern: 550 x 360mm		13

Kuva 22. Schedule-taulukko arkkitehtimallin vesikalustetuotteista, jossa on eritelty määrät eri "Family and Type" mukaan

Tutkitaan LVI-mallia vastaavasti Schedule-taulukolla kanavatarvikkeista ja nähdään, että palopeltejä (JSM) on erikokoisia ja yhteensä 14 kappaletta sekä säätöpeltejä (IRIS) on erikokoisia ja yhteensä 37 kappaletta. Kuvassa 23 on esitetty kanavien varusteista Schedule-taulukko.

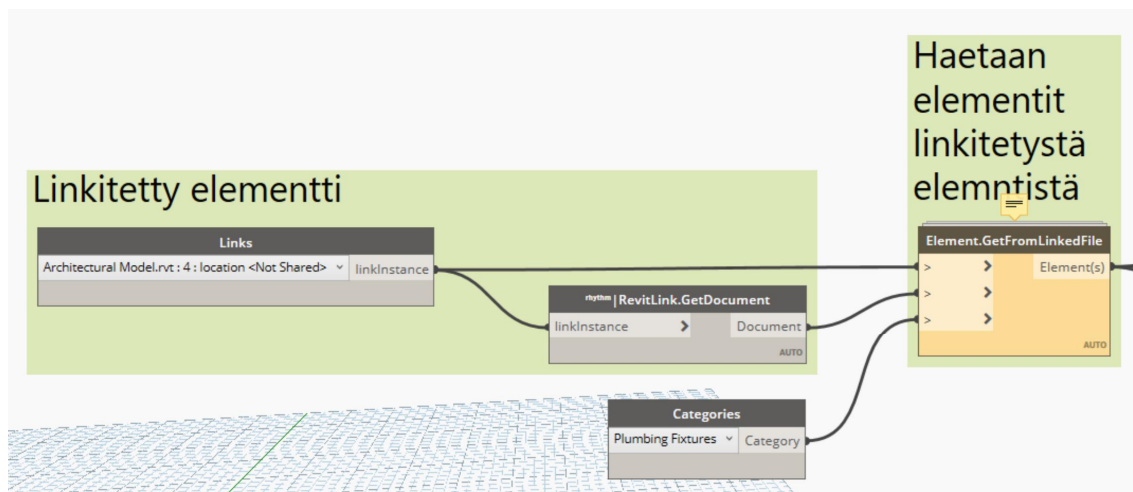
<Duct Accessory Schedule>		
A	B	C
Family	Family and Type	Count
ACCESS PANEL DUCT PART	ACCESS PANEL DUCT PART: Access-panel-400-duct p	1
IRIS	IRIS: IRIS-100	8
IRIS	IRIS: IRIS-125	16
IRIS	IRIS: IRIS-160	8
IRIS	IRIS: IRIS-250	5
JSM	JSM: JSM 400x200	3
JSM	JSM: JSM 500x200	2
JSM	JSM: JSM 600x200	4
JSM	JSM: JSM 600x400	5
MSA100_500	MSA100_500: MSA100-100-4/800x600x500	1
MSA100_500	MSA100_500: MSA100-100-5/1000x300x500	1

Kuva 23. Schedule-taulukko LVI-mallin kanavatarvikkeista, jossa eritelty määrät eri "Family and Type" mukaan

Tällöin voidaan tehdä harkinta, lähdetäänkö etsimään yksittäisiä tuotteita ja piirtämään jokaiselle erikseen vai onko tehokkaampaa käyttää automatisointia.

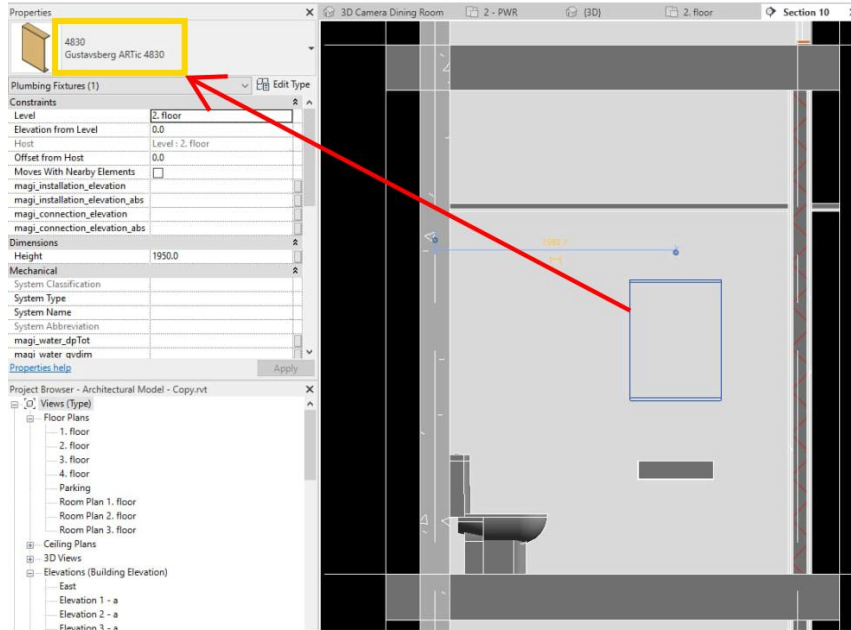
3.2 Peilivalaisimen mallintaminen arkkitehdin peilikaapin perusteella

Lähdetään rakentamaan skriptiä osa osalta, jolla voidaan piirtää arkkitehdin peilikaapin yläpuolelle peilivalaisin. Aluksi haetaan elementit linkitetystä arkkitehdin mallista. Kuvassa 24 on esitetty skriptin alkuosa.

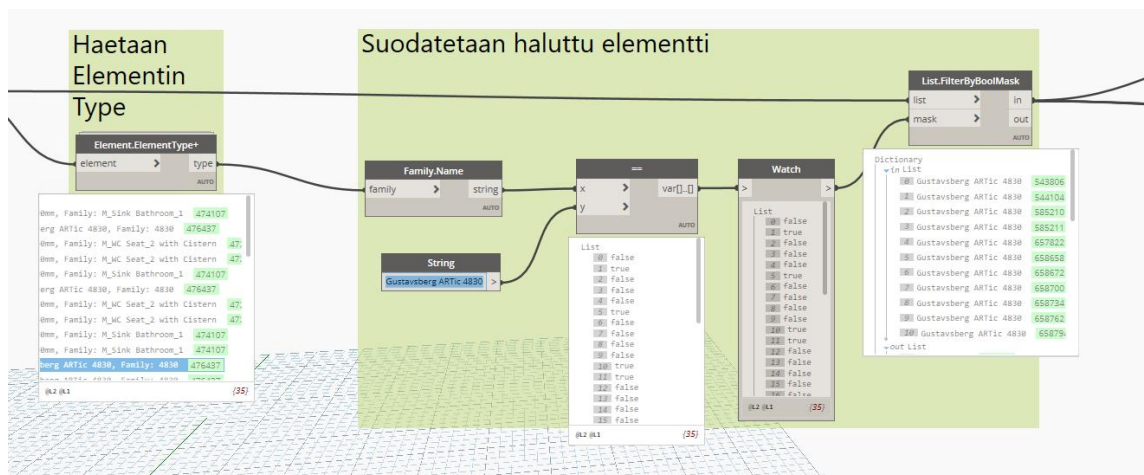


Kuva 24. Skriptin osa, jolla elementit linkitetystä mallista

Nyt meillä on lista elementeistä, joiden kategoria on Plumbing Fixtures. Suodatetaan haluttu elementti family type namen perusteella. Tutkitaan arkkitehdin Revit mallia ja selvitetään, mikä on peilikaapin Family type name. Saadaan vastaukseksi "Gustavsberg ARTic 4830". Family type namen sijainti on esitetty kuvassa 25. Huomion arvoista, että nimi on kirjoitettava virheineen päivinään skriptiin. Tämän tiedon perusteella suodatetaan peilikaappien elementit. Kuvassa 26 on esitetty skriptissä suodatus.

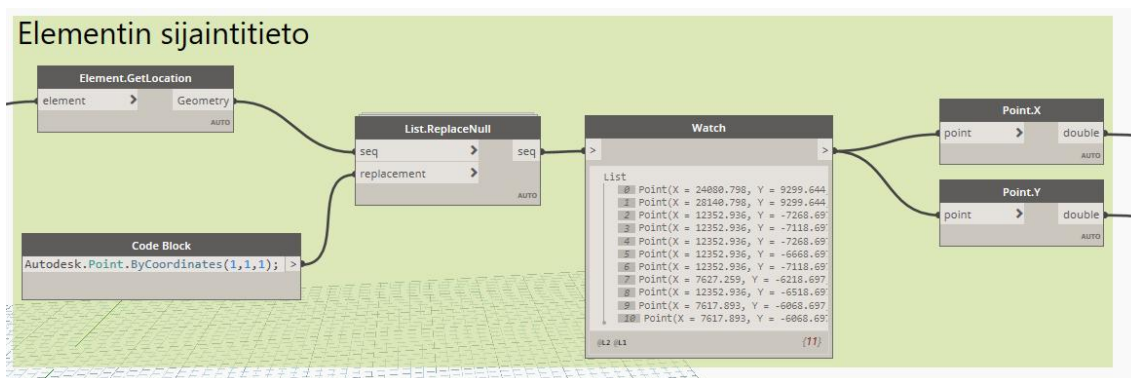


Kuva 25. Peilivalaisimen tietojen tutkimista Revitissä



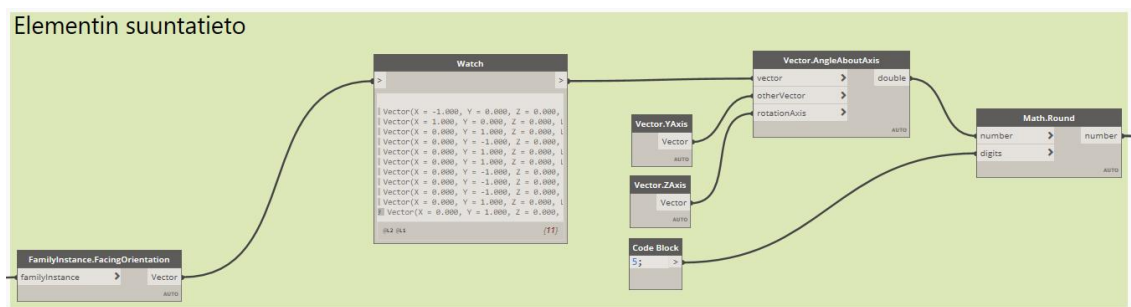
Kuva 26. Skriptin osa, jolla suodatetaan elementit Family type nimen perusteella.

Elementtien sijaintitieto tarvitaan, jotta uusi elementti voidaan kohdistaa samaan paikkaan. Elementin sijaintitieto saadaan kuvan 27 mukaisella skriptin osalla käyttäen Element.GetLocation -solmua. Osa elementeistä saattaa olla väärin arkitehdin mallissa ja tuottaa nolla-arvoa. Muutetaan nolla-arvoille koordinaatti (1,1,1), jotta skripti ei kaadu.



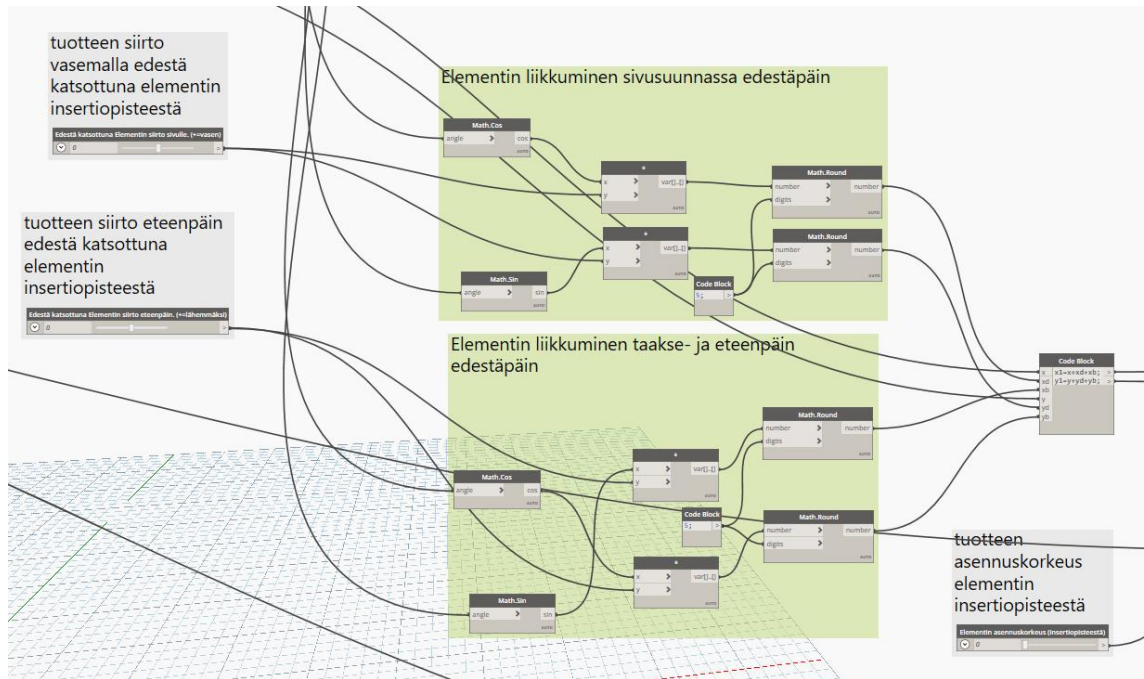
Kuva 27. Skriptin osa, jolla haetaan elementin sijainti

Arkkitehdin mallissa peilikaappien suunta vaihtelee riippuen tilanteesta. Elementin suuntatieto saadaan FamilyInstance.FacingOrientation -solmulla. Kuvassa 28 on esitetty suuntatiedon käsittelyn skriptin osa.



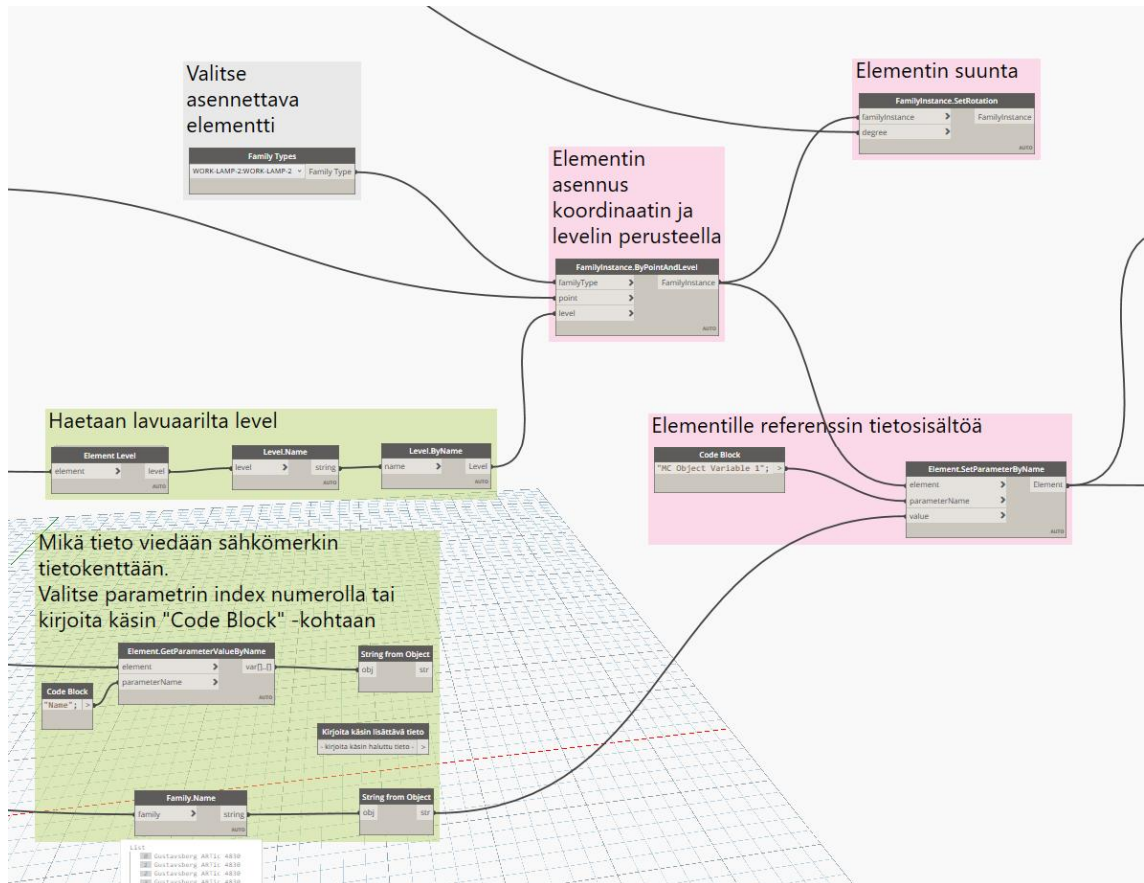
Kuva 28. Skriptin osa, jolla haetaan elementin asennussuunta.

Elementtiä sijoitettaessa on pystyttävä siirtämään sitä sivu-, pituus- ja korkeussuunnassa suhteessa referenssielementin suuntaan nähden. Elementin suuntatiedon perusteella voidaan geometrinen lauseiden avulla tehdä kaava, joilla lasketaan x- ja y-suuntaiset muutokset suhteessa elementin suuntaan. Kuvassa 29 on esitetty skriptin osa, jolla voidaan määrittellä elementille haluttu sijoittelu referenssi elementin insertiopisteeseen nähden. Arkkitehdin mallista nähdään, että peilikaappin sijoituksen korko on nolla. Tällöin valaisin on sijoitettava korkoon 2100mm. X- ja y-suuntainen insertiopiste sijainti selviää kokeilemalla. Useassa kappaleessa x- ja y-suuntainen insertiopiste sijaintee elementin takana keskellä.

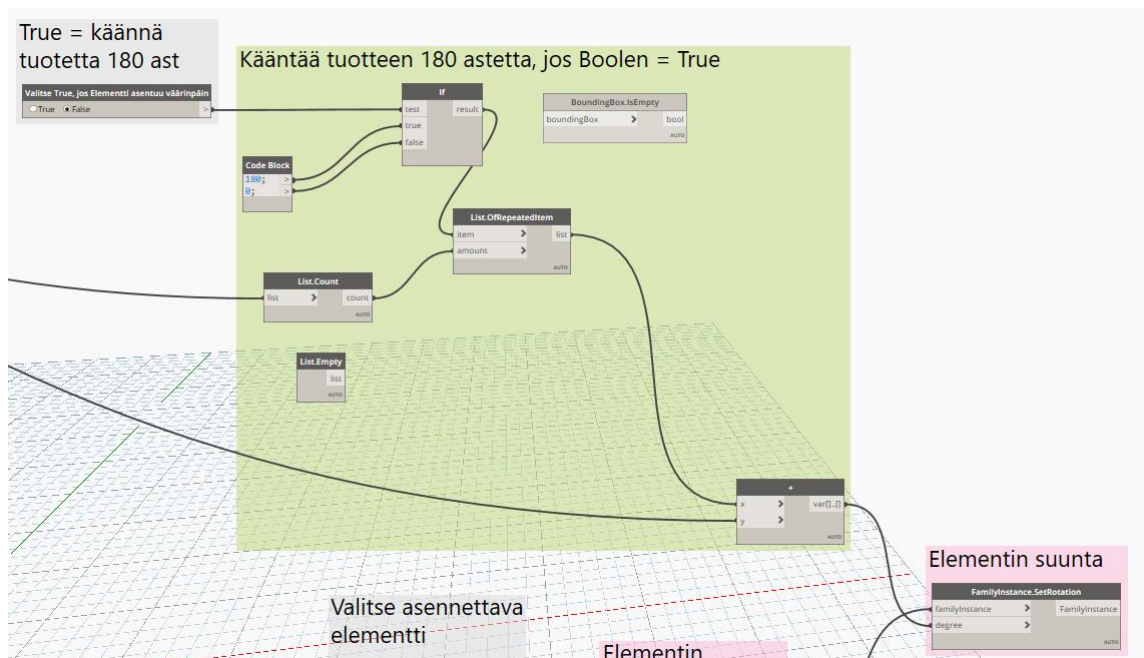


Kuva 29. Skriptin osa, jolla siirretään elementtiä suhteessa referenssi elementtiin

Elementin sijoittamiseen löytyy monta erilaista solmua. Käytetään tässä FamilyInstance.ByPointAndLevel -solmua. Solmu tarvitsee tiedot: Family Type, koordinaatti sekä kerrostiedon. Kerrostieto saadaan referenssielementistä. Kuvassa 30 on esitetty skriptin osa. Node sijoittaa elementin malliin, jonka jälkeen muokataan juuri sijoitetun elementin tietoja kääntämällä sen suunta samaksi kuin referenssielementti lisäämällä parametreihin tietoa. Suuntatietoon lisättiin ominaisuus, jolla voidaan lisätä arvoon 180 astetta, tällöin elementti kääntyy toisinpäin x,y suunnassa. Suuntatiedon lisäys esitetty kuvassa 31. Tietoa voi lisätä usealla eri tavalla. Tässä on linkitetty peilivalaisimen Family name -tieto. Jos peilikaapeilla olisi arkkitehdin mallissa positionumero, se tieto olisi hyödyllisempi, sekä toimisi hyvin myös muille tuotteille. On mahdollista myös kirjoittaa käsin tietoa, joka annetaan String-solmulla.

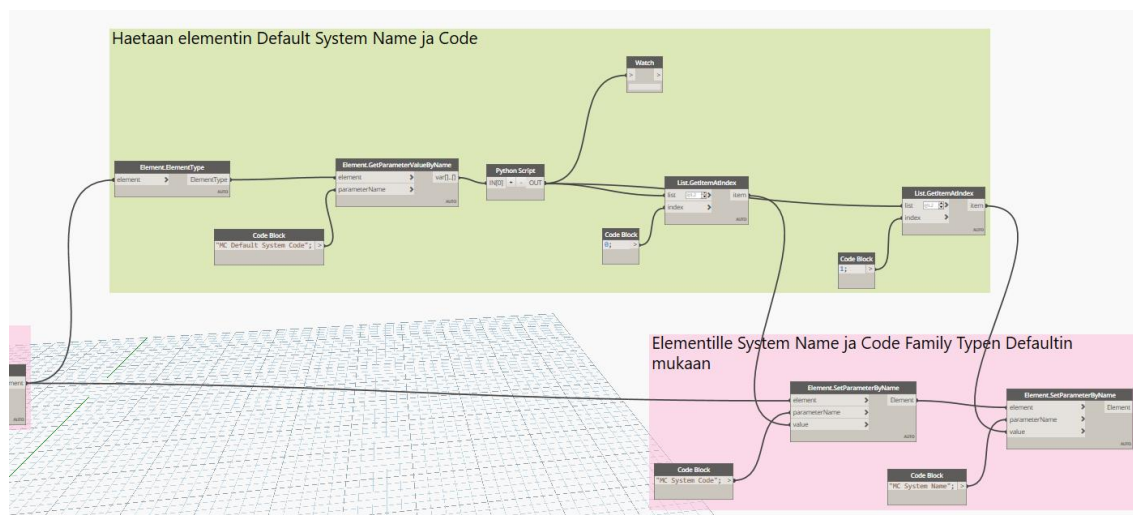


Kuva 30. Skriptin osa, jolla sijoitetaan elementti malliin ja lisätään tietoja



Kuva 31. Skriptin osa, jolla annetaan elementille suuntatieto sekä lisäosa, jolla voidaan kääntää elementti 180 astetta.

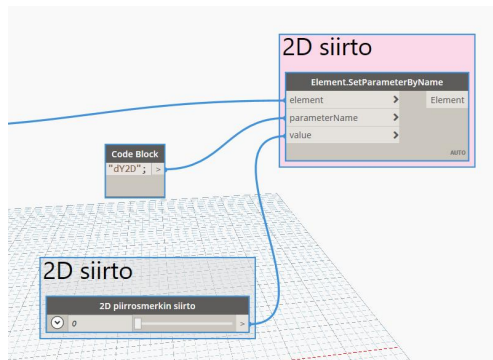
Lopuksi siirretään uudelle elementille MC System Code ja Name -parametrit, jotka saadaan muodostettua Family typen default -arvosta. Kuvassa 32 on esitetty skriptin osa, jolla muutetaan default-arvo kahdeksi arvoksi ja syötetään oikeille parametreille takaisin.



Kuva 32. Skriptin osa, jolla siirretään Magicadin default system name ja code elementin parametrin arvoihin system name ja code.

MagiCADin familyn system defaults -arvo määritellään Dataset Properties valikossa. Jos esimerkiksi datarasialle ei määritä default-arvoa, koska sitä voidaan käyttää myös muissa järjestelmissä, tällöin tämä automaattinen MC System Code ja Name muodostaminen ei toimi, koska arvo jää typeparametreihin tyhjäksi.

Yksi automaattipiirron ongelmia on tuotteiden päällekkäinen piirtyminen 2D-näkymässä. Tätä ongelmaa voidaan vähentää lisäämällä parametriin arvo, joka siirtää 2D-objektin sijaintia 3D-objektin insertiopisteestä. Kuvassa 33 on esitetty skriptin osa, jossa numeroliukurilla voidaan antaa arvo 2D-objektin siirrolle.

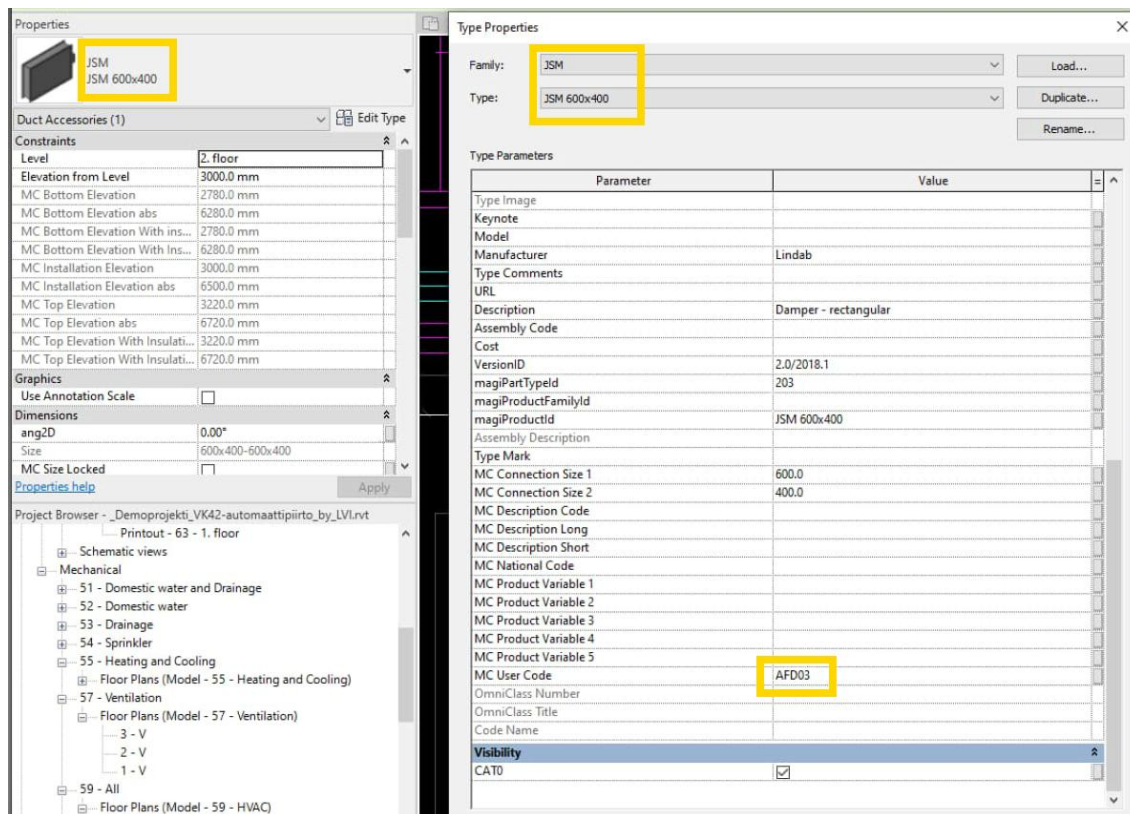


Kuva 33. Skriptin osa, jolla voidaan siirtää 2D-objektia 3D-objektin sijainnista

3.3 Moottoripellin mallintaminen LVI:n palopellin mukaisesti

Tehdään skripti, jolla voidaan piirtää elementtejä LVI-suunnittelijan Revit-mallin perusteella. Skriptin periaatteet toimivat vastaavasti kuin peilivalaisimen piirroksessa. Suurin ero, ettei haeta linkitetystä mallista elementtejä.

Haetaan kaikki elementit kategorioiden mukaan. LVI:n Revit-mallia tutkimalla nähdään, että palopellit kuuluvat kategoriaan Duct Accessories, niiden Family Type on JSM 600x400 ja MC User Code on AFD03. Päätetään suodattaa elementtejä käyttäen User Code. Kuvassa 34 on esitetty malliprojektin properties ja Type properties tietokenttiä. Molempia voidaan käyttää hyödyksi Dynamossa.



Kuva 34. LVI Revit-mallista palopellin parametrejä. Korostettu Family Type ja MC User Code

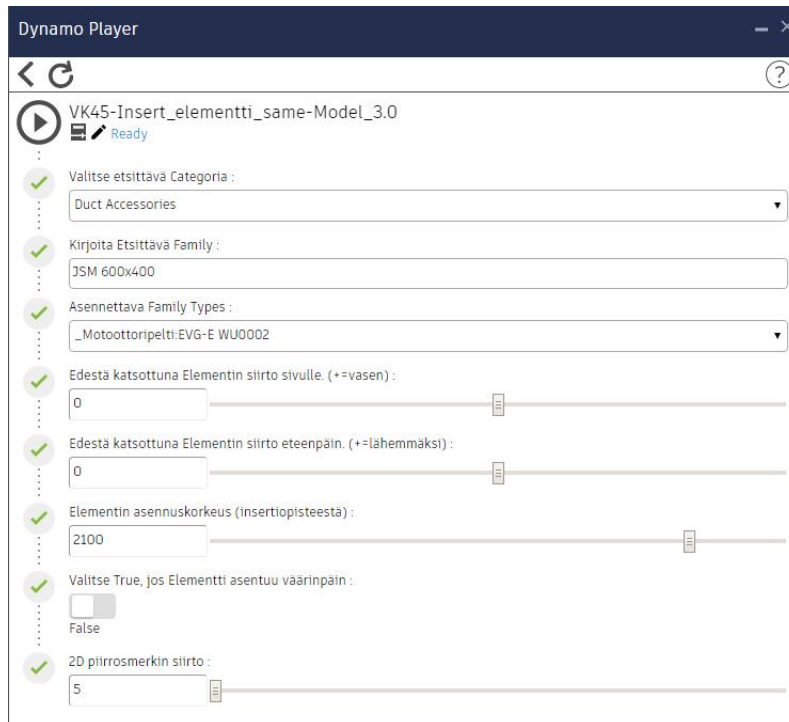
Siirretään piirrettävälle sähkön elementille palopellin position tieto. Suunnittelu-projektissa on sovittava yhteisesti mille parametrille tieto sijoitetaan LVI-mallissa, jotta se kopioituu oikein riippumatta kategoriasta.

Dynamo-Playerin käyttöliittymän tekeminen

Kun saadaan toimimaan skriptit tietyillä tuotteilla, muutetaan ne toimimaan yleisesti sekä lisätään Dynamo-Playeriin sisäänmenot. Yleistämisessä voidaan esimerkiksi tehdä muutoksia skriptiin, jotta voidaan hakea erilaisilla vaihtoehdoilla tuotteita tai muokata niitä.

Valitaan Dynamo-Playeriin sisäänmenot skriptistä. Kuvassa 35 on esitetty Dynamo-Playerin käyttöliittymä, josta näkee sisäänmenot. Ensin valitaan halutun referenssielementin kategoria ja kirjoitetaan elementin Family Type, jonka mu-

kaan elementit lajitellaan. Valitaan mallinnettava tuotteen Family Type ja annetaan niille siirtymät x-, y-, z-suunnissa suhteessa elementin sijaintiin ja suuntaan nähden. Jos elementti piirtyy väärinpäin, voidaan valita elementin kääntö 180 astetta. Lopuksi voidaan siirtää 2D objektin sijaintia 3D objektiin nähden.

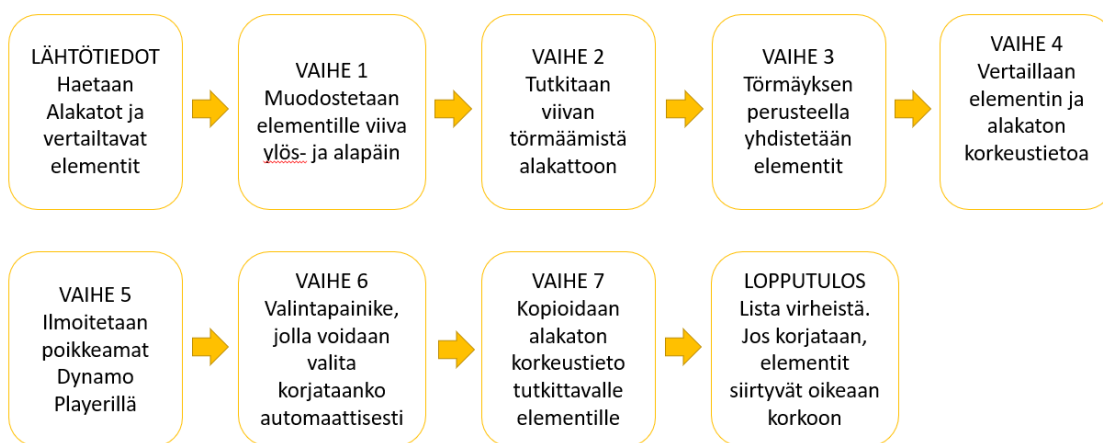


Kuva 35. Dynamo-Playerin näkymä automaattipiirroksessa.

Skriptin käyttöliittymään voidaan kehittää muuttamalla valittavan referenssi tuotteen tietojen hakutapa. Se on mahdollista toteuttaa esimerkiksi siten, että voidaan valita haluttu elementti mallista hiiren painalluksen avulla.

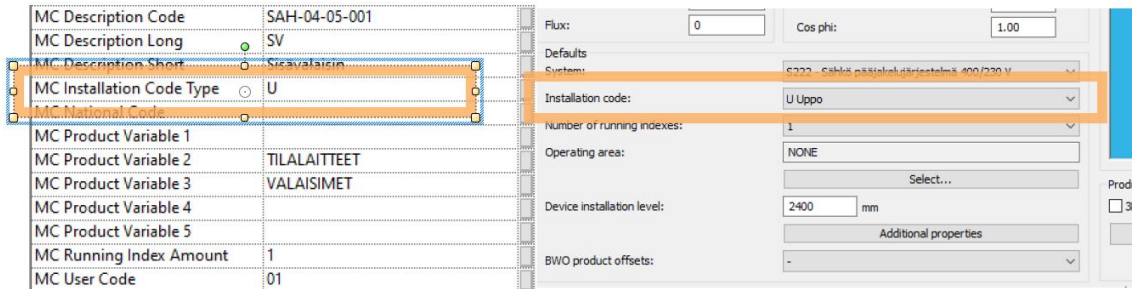
4 Skripti: Elementtien asennuskorkeuden tarkastaminen ja muuttaminen alakaton perusteella

Tutkittiin, miten voidaan vertailla elementtien korkeuksia alakaton suhteen. Muodostettiin prosessikaavio skriptin tekemistä varten. Prosessikaavio esitetty kuvassa 36. Lajitellaan ensin tutkittavat elementit ja piirretään niiden insertiopisteestä viiva ylös- ja alaspäin. Tutkitaan viivan ja alakaton törmäyskohtaa ja yhdistetään ne sen perusteella. Sen jälkeen voidaan hakea molemmista korkeudet ja vertailla niitä keskenään sekä muuttaa alakaton arvo elementin arvolle.



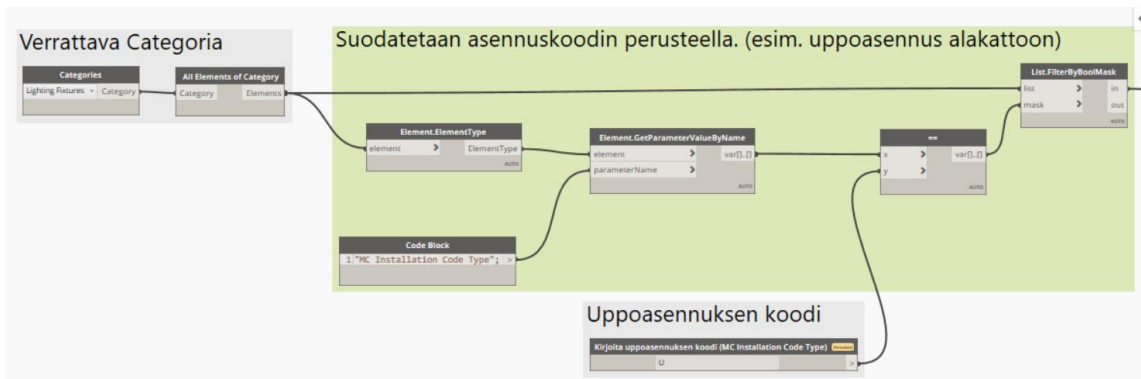
Kuva 36. Prosessikaavio elementtien korkojen muuttamisesta alakaton korkoihin automaattisesti

Jos halutaan tutkia kerralla kaikki valaisimet. On ne luokiteltava jotenkin järkevästi, jotta seinävalaisimet ja ripustetut valaisimet eivät tule mukaan tutkimiseen. Tutkitaan valaisimien parametrejä ja nähdään, että uppovalaisimen asennuskoodi määrittää Revitin parametrille MC Installation Code Type. Jos kaikille alakattoon asennettaville tuotteille käytetään tätä asennuskoodia, voidaan tutkia kerralla koko kategoria. Kuvassa 37 on esitetty tutkittava parametri.



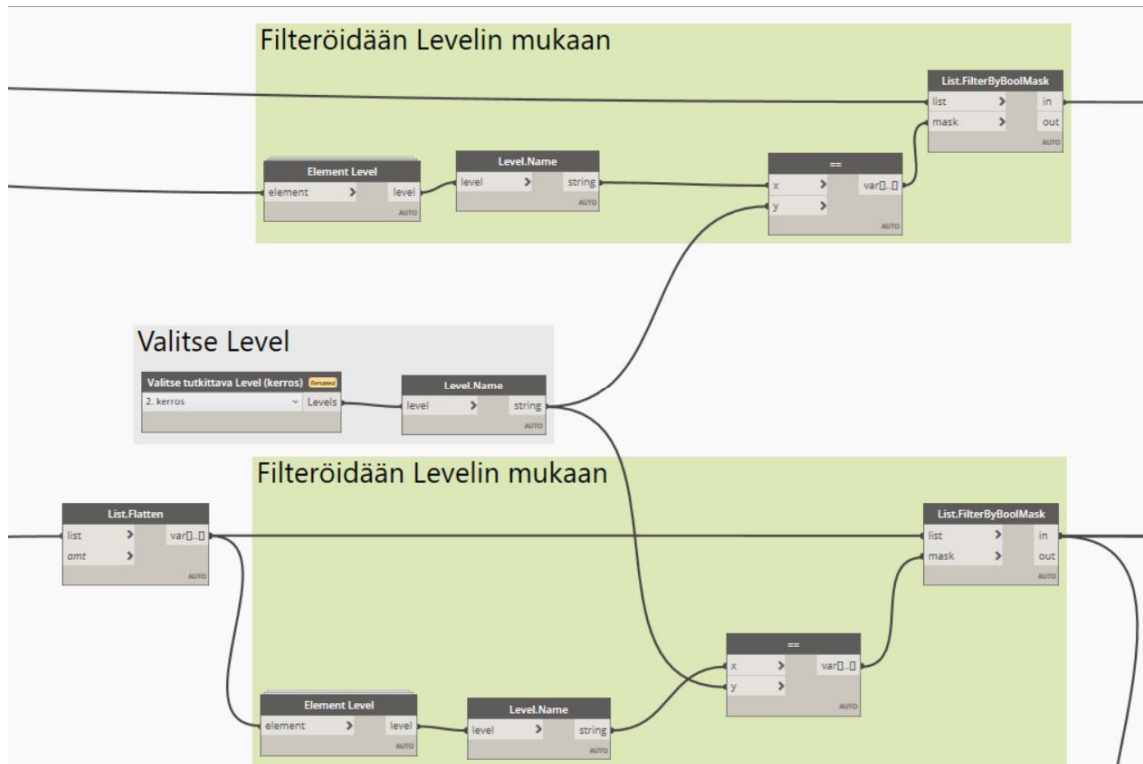
Kuva 37. Revitin ja MagiCADin parametri uppoasennuskoodista

Aluksi haetaan alakatot linkitetystä mallista vastaavalla tavalla kuin automaattipiirto skriptissä haetaan elementtejä. Haetaan tutkittava elementit ja suodatetaan ne uppoasennuskoodin mukaan. Esimerkiksi valaisimet. Skriptin osa on esitetty kuvassa 38.



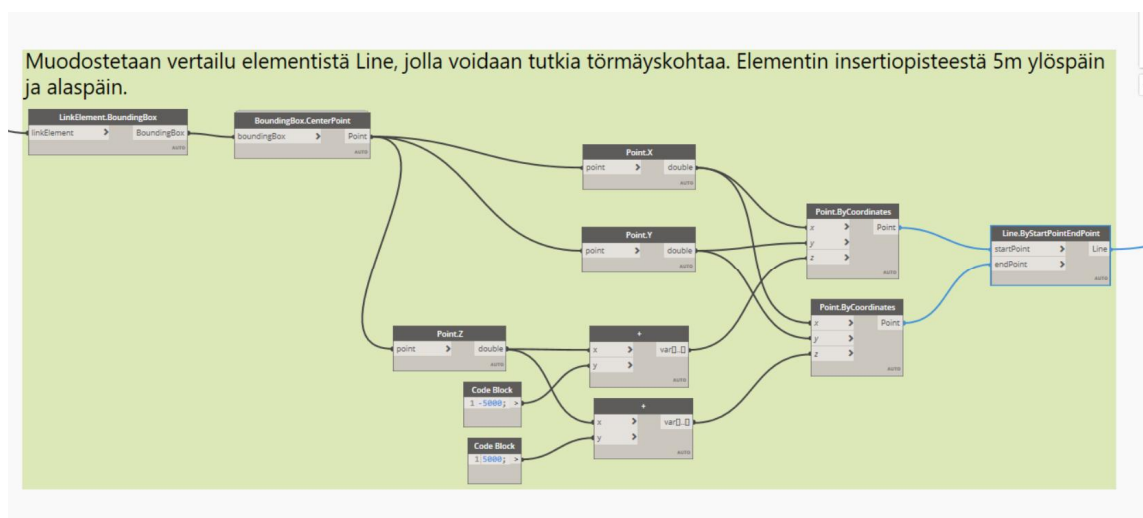
Kuva 38. Skriptin osa, jossa haetaan valaisimet ja suodatetaan uppoasennuskoodin perusteella.

Kun tutkitaan viivan perusteella törmäyskohtaa, on suodatettava kerroksittain elementit, jotta viiva ei löydä törmäyskohtaa ylemmästä tai alemmasta kerroksesta. Kuvassa 39 on esitetty suodatus kerrostiedon perusteella.



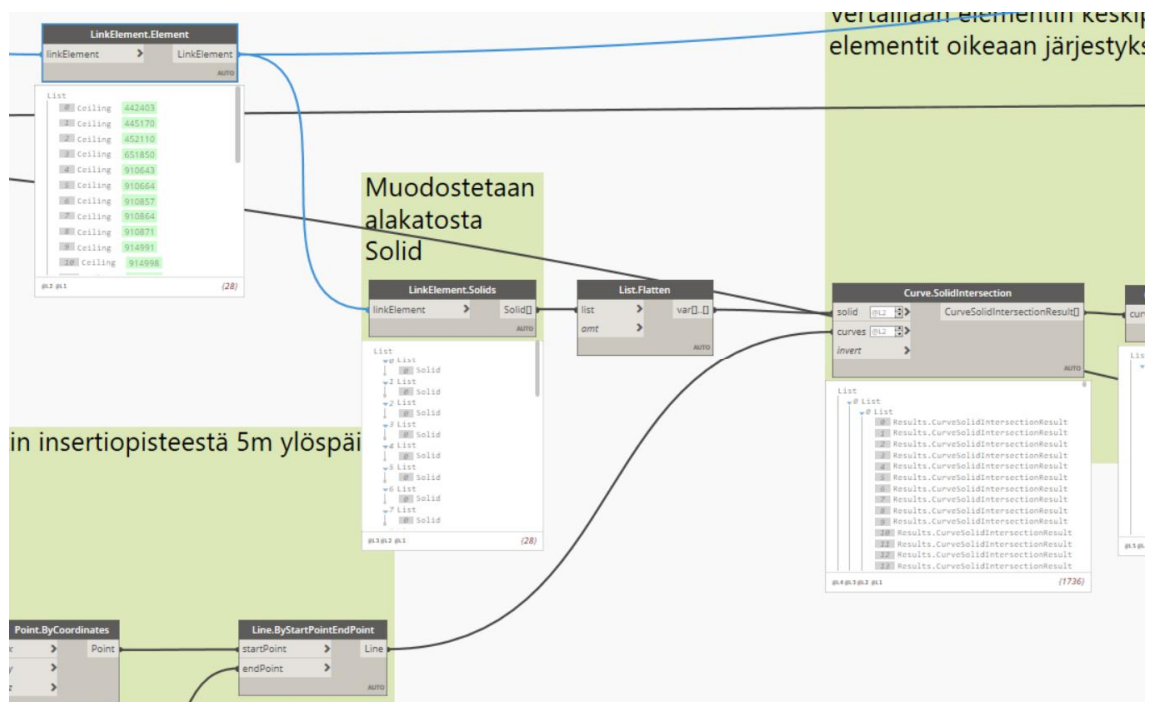
Kuva 39. Skriptin osa, jossa suodatetaan elementit kerrostiedon perusteella

Tutkittavalle elementille piirretään viiva ylös- ja alaspäin 5m, jonka avulla tutkitaan törmäyskohtaa. Kuvassa 40 on esitetty viivan tekeminen elementin insertiopisteestä. Elementistä muodostetaan BoundingBox ja siitä haetaan keskipiste. Määritellään keskipisteen avulla kaksipistettä, joiden x- ja y-arvot ovat samat, mutta z-arvoon listään ja vähennetään 5000. Näiden pisteiden perusteella muodostetaan viiva.



Kuva 40. Skriptin osa, jossa piirretään viiva elementin insertiopisteestä

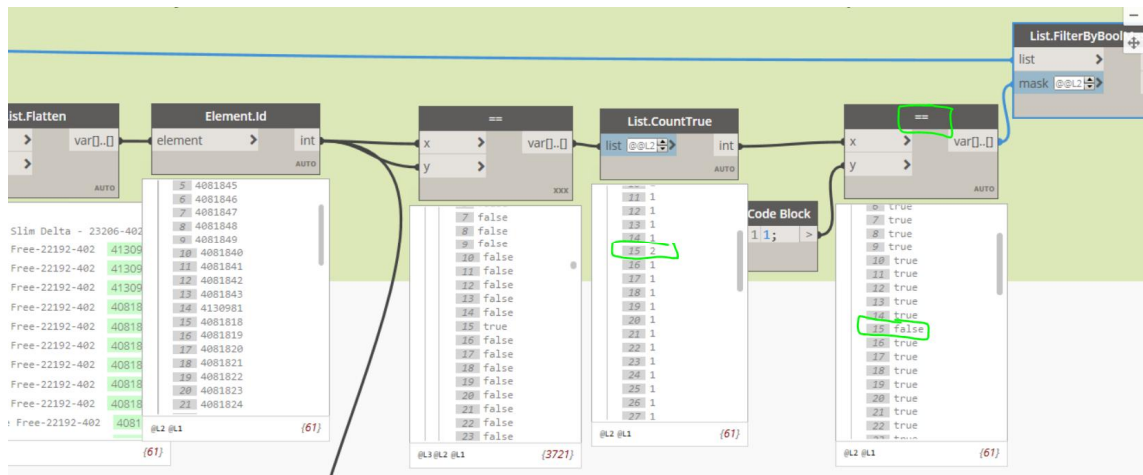
Viivan törmäyskohtaa tutkitaan alakattoon solmulla Curve.SolidIntersection. Solmu haluaa tiedoksi Solidin ja Curvesin. Alakatto muutetaan Solid-muotoon LinkElement.Solid-nodella. Kuvassa 41 on esitetty skriptin osa.



Kuva 41. Skriptin osa, jossa tutkitaan alakaton ja viivan törmäyskohtaa

Curve.SolidIntersection-node ulostulon kääntämiseen löytyy erilaisia solmuja. Valitaan CurveSolidIntersectionResult.CurveSegments-node, jolla saadaan arvoja törmäyskohdasta. Jotta voidaan saada lista FilterByBoolMask varten, tutkitaan, onko törmäyskohdan arvo tyhjä vai ei List.IsEmpty-nodella ja kääntämällä true-false-arvot toisinpäin. Tällöin voidaan filteröidä elementit alakaton mukaiseen järjestykseen. Kuvassa 42 on esitetty skriptin osa.

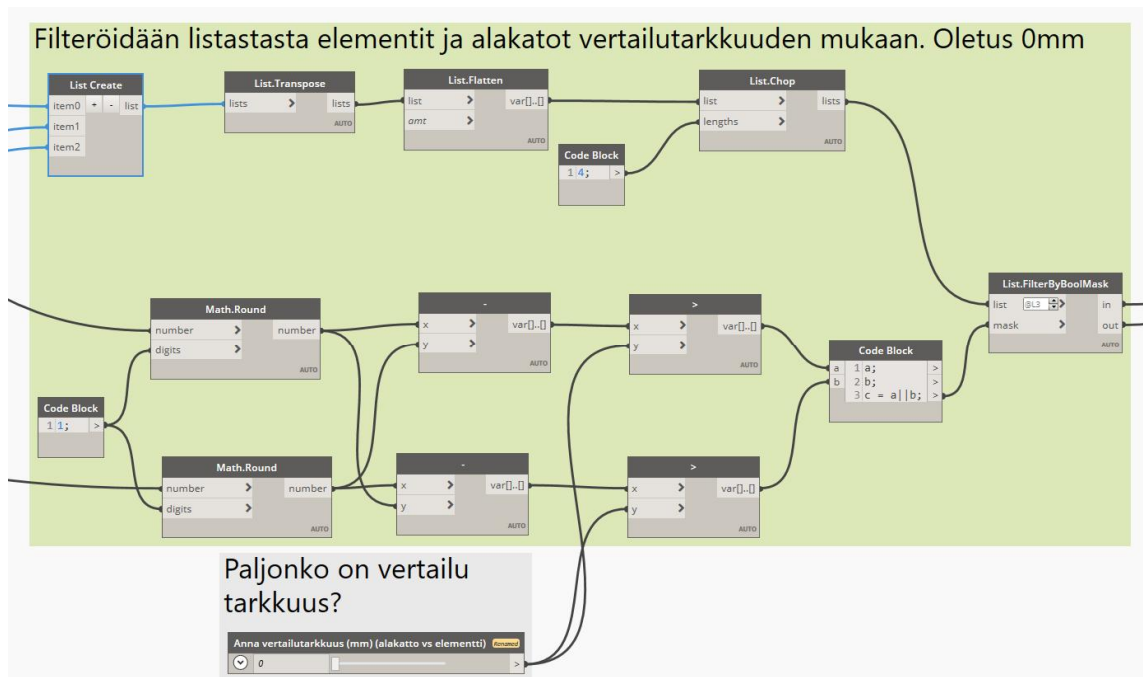
lanteet poistettua. Haetaan valaisimista ID-koodi ja tutkitaan ==-solmulla esiintyykö jossain samaan ID-koodia kahdesti. Tämän perusteella suodataan ne elementit pois. Vastaavalla tyyllillä voidaan tutkia, onko valaisimia, joille ei ole osoitettu ollenkaan alakattoa vertaamalla ID-koodeja listaan, joita ei ole yhdistetty vielä alakattoon.



Kuva 44. Skriptin osa, jossa suodatetaan valaisimet pois, joiden kohdalla on kaksi alakattoa.

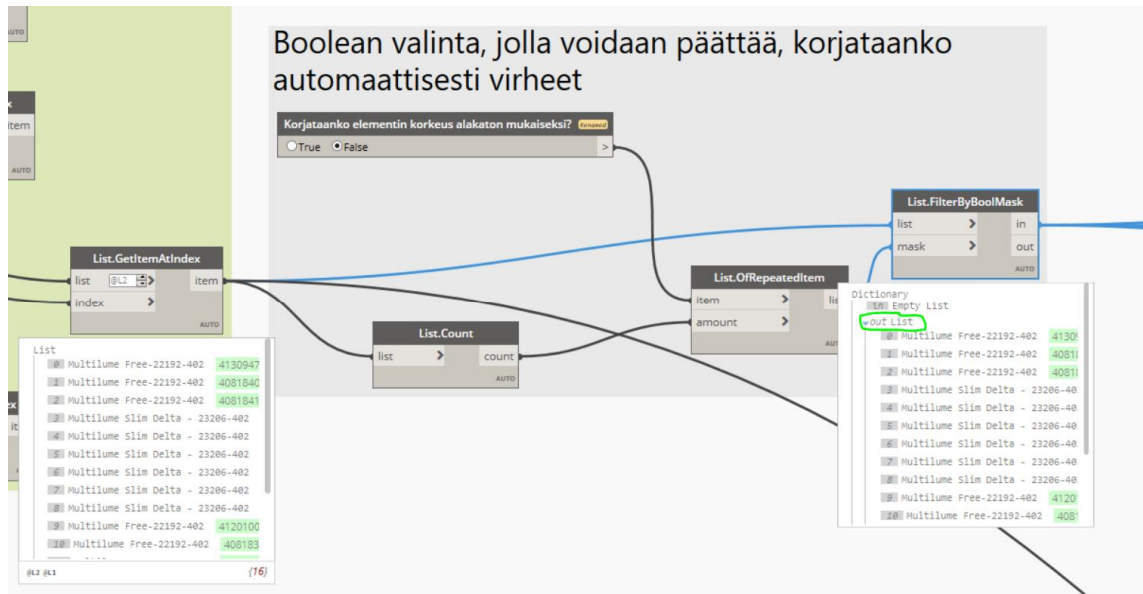
Viistojen alakattojen tutkiminen ei onnistu vastaavasti. Viistot alakatot voidaan filteröidä pois List.FilterByBoolMask-nodella, kun tutkitaan, onko alakaton "Slope" parametri tyhjä vai jokin arvo. Slope-parametri kertoo alakaton kallistuskulman.

Tämän jälkeen meillä on lista, jonka alilistalla on aina valaisin ja sen alakatto peräkkäin. Voidaan hakea valaisimen ja alakaton korkeudet ja vertailla niitä keskenään. Korkeus voidaan hakea Element.GetParameterValueByName-Nodella hakemalla parametriä "Height Offset From Level" alakatosta ja "Elevation from Level" valaisimesta. Valaisimien korkeus ei ole aina tasaluku. Tällöin voidaan käyttää Math.Round-nodea, jolla voidaan pyöristää arvo tasalukuun. Vertailu voidaan suorittaa miinuslaskulla ja suurempi- ja pienempi -merkeillä. Kirjoitetaan lopuksi ehtolausu Code Blockiin. Kuvassa 45 on esitetty skriptin osa, jossa voidaan antaa vertailutarkkuus, jolla tutkitaan eroja. List.FilterByBoolMask-nodella suodatetaan listaa, joka pitää sisällään alakaton korkeuden, valaisimen korkeuden, alakattoelementin ja valaisinelementin.



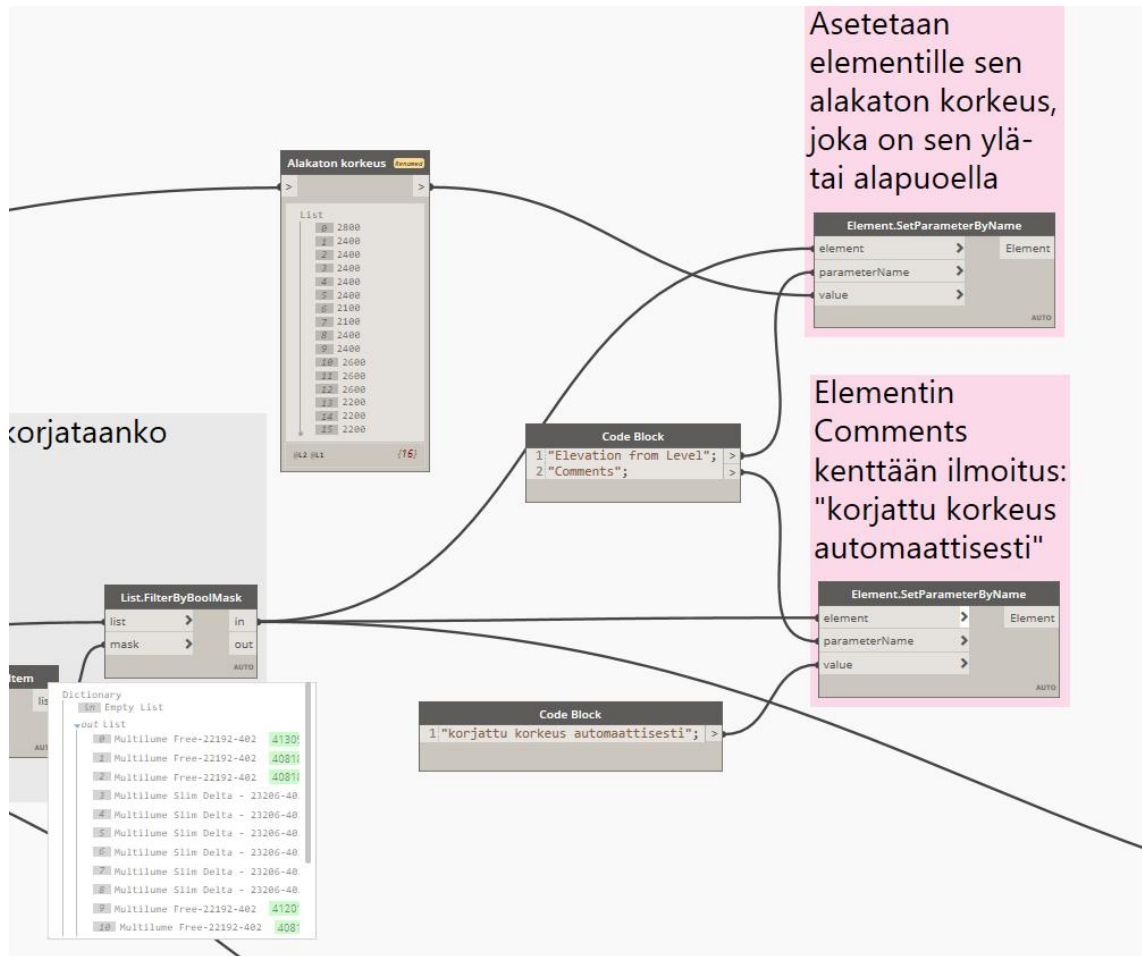
Kuva 45. Skriptin osa, jossa vertaillaan valaisimen ja alakaton korkeutta keskenään

Tutkitaan, miten voidaan tehdä "nappula", jolla voidaan valita, toteutetaanko skriptin loppuosa vai ei. Kokeillaan jo usein käytettyä tapaa muokattuna, jossa suodatetaan listaa totuusarvon mukaisesti. Asetetaan maskiksi käsin valittava boolean-arvo ja lisätään se niin monta kertaa, kuin on listoja. Kuvassa 46 on esitetty toimiva ratkaisu. Kuvasta näkee, että kun valittuna "False", niin tuleva lista menee List.FilterByBoolMask -solmun "out List" puolelle. Lista voidaan päästää läpi valitsemalla "True".



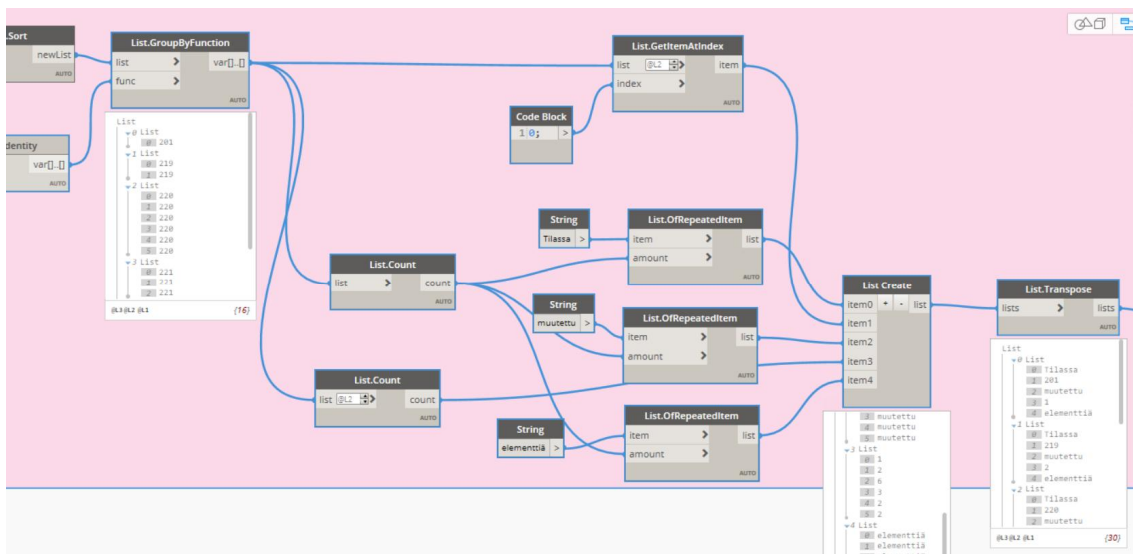
Kuva 46. Skriptin osa, jossa käsin voidaan valita, jatketaanko skriptiä eteenpäin

Siirretään elementille alakaton korkeus parametriin "Elevation from Level" ja siirretään käsin kirjoitetut terveiset "korjattu korkeus automaattisesti" kenttään "Comments". Kuvassa 47 on esitetty skriptin loppuosa.



Kuva 47. Skriptin osa, jossa siirretään listasta alakaton korkeustieto elementin korkeustiedoksi.

Tehdään lopuksi Watch-nodella tietokenttä, jossa kerrotaan mitä on muutettu. Haetaan valaisimista huonetiedot `rhythm|FamilyInstances.Space-` ja `Element.GetParameterByName`-nodella. Muokataan huonenumeroit alilistoihin esiintymismäärän perusteella, jotta voidaan kertoa, montako tuotetta esiintyy missäkin tilassa. Tehdään näistä uusi lista, johon lisätään myös käsin kirjoitettua tekstiä. Kuvassa 48 on esitetty tekstin muodostamista.



Kuva 48. Skriptin osa, jossa muodostetaan Watch-nodeen selkeää viestiä tehdyistä muutoksista.

Jatketaan vielä listan muokkausta, jotta saadaan kirjoitettua yhdelle riville kommentti. Muutetaan listalla esiintyvät sanat peräkkäin String.Join-nodella ja käytetään " "-merkkiä sanojen välillä. Tällä tavalla muokkaamalla saadaan listasta valaisimia tehtyä selkeäkielinen kommentti kenttä tehdyistä muutoksista, esimerkiksi: "Tilassa 220 muutettu 6 elementtiä". Kuvassa 49 on esitetty skriptin loppuosa.

Tehdään lopuksi vastaavanlaiset kommenttikentät tiloille:

- joissa valaisin ei ole alakattokorkeudella
- joissa on kaksi alakattoa päällekkäin
- joissa on viisto alakatto.

Muodostetut Watch-solmut nimetään uudelleen ja määritellään Dynamo-Playerin ulostuloiksi.



Kuva 49. Skriptin osa, jossa muodostetaan Watch-nodeen selkeää viestiä tehdyistä muutoksista.

Dynamo-Playerin käyttöliittymän tekeminen

Määritellään Dynamo-Playeriä varten sisäänmenot tarvittavista solmuista. Kuvassa 50 on esitetty Playerin näkymä, jossa voidaan valita linkitetty malli, tutkittava Family type sekä "nappula", jolla voidaan valita, korjataanko arvot.

Dynamo Player

< ↻ ?

VK1-Tarksita-Categoria_onko_alakatossa_Solid_4.7
Run completed with errors

✓ Valitse arkkitehdin Revit-malli, jossa Ceilings (alakatot) :

ARK - Yleisnimet.rvt : 6 : location <Not Shared>

✓ Valitse tutkittava Level (kerros) :

2. kerros

✓ Categories :

Lighting Fixtures

✓ Anna vertailutarkkuus (mm) (alakatto vs elementti) :

0

✓ Korjataanko elementin korkeus alakaton mukaiseksi? :

True

✓ Kirjoita oppoasennuksen koodi (MC Installation Code Type) :

U

1. Näissä tiloissa tutkittavat elementit ovat väärässä korkeudessa :

List ▼

- 0 Tilassa 216 on 1 elementtiä väärässä korkeudessa
- 1 Tilassa 218 on 4 elementtiä väärässä korkeudessa
- 2 Tilassa 226 on 1 elementtiä väärässä korkeudessa
- 3 Tilassa 228 on 3 elementtiä väärässä korkeudessa

2. Näissä tiloissa tutkittavien elementtien asennuskorkeus korjattu automaattisesti :

List ▼

- 0 Tilassa 216 muutettu 1 elementtiä
- 1 Tilassa 218 muutettu 4 elementtiä
- 2 Tilassa 226 muutettu 1 elementtiä
- 3 Tilassa 228 muutettu 3 elementtiä

3. Näistä tiloista ei voida tutkia elementtejä. Elementti löytää monta alakattoa samalta kohtaa :

List ▼

- 0 Tilassa 217 ei voida tutkia kaikkia elementtejä

4. Näissä tiloissa on tutkittavia elementtejä, mutta ei alakattoa :

List ▼

- 0 Tilassa 202 on 2 elementtiä, joilta ei löydy alakattoa

✓ Viistoja alakattoja, ei voida tutkia :

List ▼

- 0 Tilassa 215 on viisto alakatto. Ei voida tutkia

Kuva 50. Dynamo-Playerin näkymä, jossa vertaillaan elementin korkeutta alakatton korkeuteen ja voidaan korjata se.

5 Skripti: Puuttuvien elementtien listaaminen automaattisesti

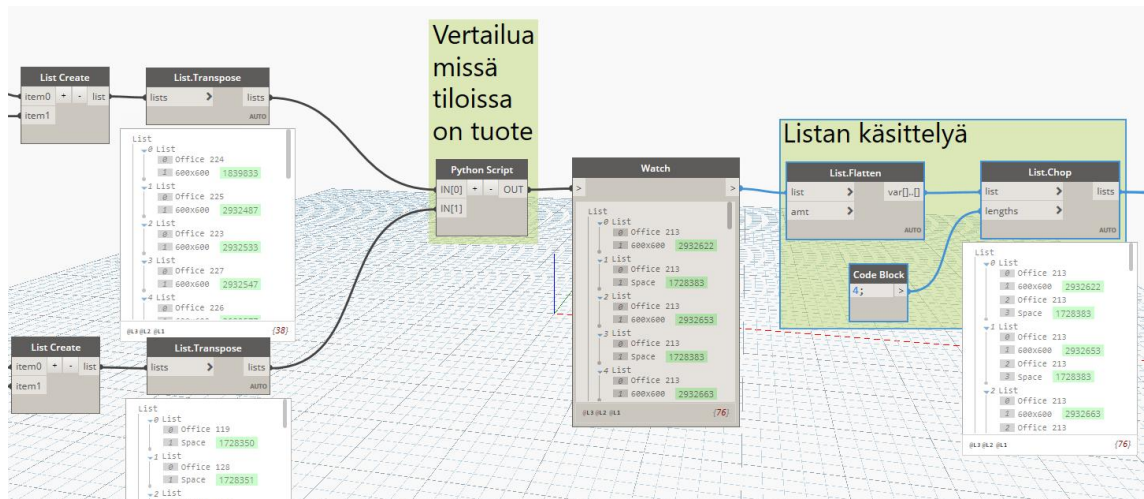
Tehdään laaduntarkastamista varten skripti, jolla voidaan putkia, puuttuu jostain tilasta haluttu elementti. Esimerkiksi tutkitaan mitkä tilat eivät sisällä paloilmainta. Skriptissä on karkeasti kolme osaa. Haetaan elementtien nimet. Vertaillaan keskenään. Lajitellaan tilat, joista ne puuttuvat. Lisätään vertailuun kuitenkin erilaisia ehtoja, että saadaan luotua monipuolinen skripti. Tehdään skriptin luomista varten prosessikaavio ja lähdetään ratkaisemaan skriptiä pala palalta. Prosessikaavio esitetty kuvassa 62.



Kuva 62. Prosessi kaavio skriptin luomisesta, jolla voidaan tutkia puuttuuko elementti jostain tilasta

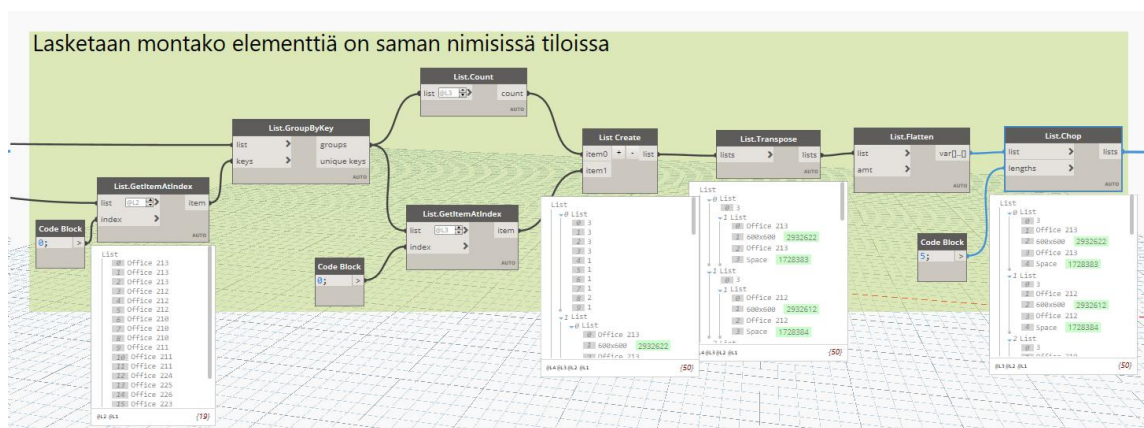
Elementtien ja tilojen hakeminen suoritetaan vastaavasti kuin aikaisemmissa skripteissä. Tutkittavien elementtien tilatieto saadaan `rhythm|FamilyInstances.Space-Nodella`, josta voidaan suodattaa elementit tilanimen perusteella `List.FilterByBoolMask-` Nodella, johon etsitään tilanimet `Element.GetParameterValueByName-` nodella ja vertaillaan sitä käsin kirjoitettavaan arvoon.

Muodostetaan tilanimistä ja elementeistä listat joita vertaillaan keskenään Python-koodin avulla. Python-koodilla pystytään käsittelemään listoja tehokkaasti erilaisilla ehtolauseilla ja silmukoilla. Saatu lista muokataan sopivaan muotoon. Skriptin osa on esitetty kuvassa 63.



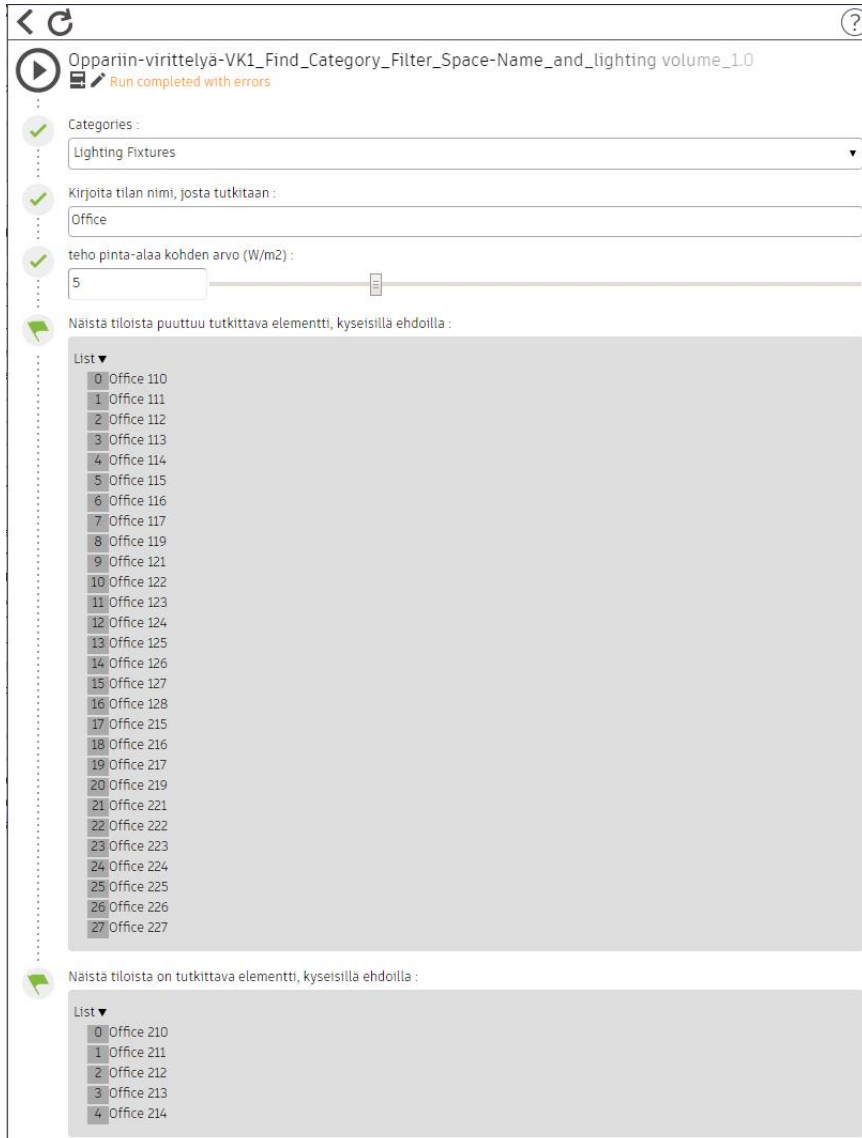
Kuva 63. Skriptin osa, jossa lajitellaan elementit niiden tilatietojen perusteella.

Käsitellään saatuja listoja siten, että saadaan selville montako elementtiä on missäkin tilassa. Tällöin voidaan rakentaa erilaisia matemaattisia kaavoja ja muita ehtolauseita, joilla tarkennetaan vaatimuksia. List.GroupByKey-solmun avulla voidaan ryhmitellä samannimiset tuotteet samalle listalle. Listoista suoritetaan määrälaskenta List.Count-nodella ja yhdistetään saadut listat. Muokataan listoja haluttuun muotoon ja saadaan lista, jossa on kerrottu montako elementtiä on missäkin tilassa. Kuvassa 64 on esitetty kyseinen skriptin kohta.



Kuva 64. Skriptin osa, jossa lasketaan, montako elementtiä on missäkin tilassa

Tämän lista avulla voidaan tehdä monipuolisia vertailuja. Kuvassa 65 on esitetty, miten voidaan tehdä vertailu, jossa tutkitaan, onko tilassa riittävän suuri valaistuksen neliöteho, joka perustuu valaisimelle annettuun tehoon. Haetaan valaisimista tehotieto hakemalla parametrin "MC Active Power"-arvo. Kun tiedetään



< ↻ ?

▶ Oppariin-virittelyä-VK1_Find_Category_Filter_Space-Name_and_lighting volume_1.0
✎ Run completed with errors

✓ Categories :
Lighting Fixtures

✓ Kirjoita tilan nimi, josta tutkitaan :
Office

✓ teho pinta-alaa kohden arvo (W/m2) :
5

✗ Naista tiloista puuttuu tutkittava elementti, kyseisillä ehdoilla :

List ▼

- 0 Office 110
- 1 Office 111
- 2 Office 112
- 3 Office 113
- 4 Office 114
- 5 Office 115
- 6 Office 116
- 7 Office 117
- 8 Office 119
- 9 Office 121
- 10 Office 122
- 11 Office 123
- 12 Office 124
- 13 Office 125
- 14 Office 126
- 15 Office 127
- 16 Office 128
- 17 Office 215
- 18 Office 216
- 19 Office 217
- 20 Office 219
- 21 Office 221
- 22 Office 222
- 23 Office 223
- 24 Office 224
- 25 Office 225
- 26 Office 226
- 27 Office 227

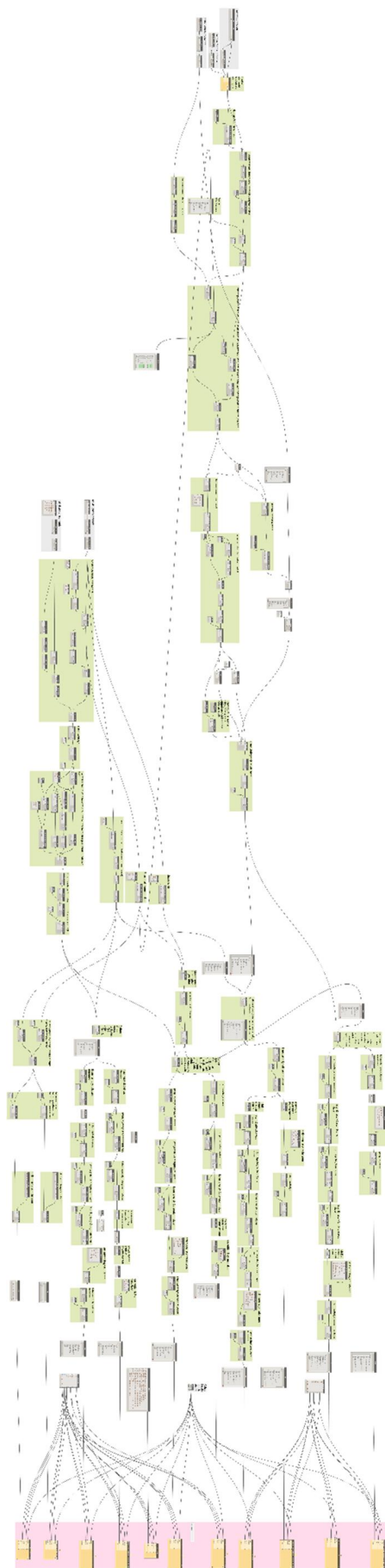
✗ Naista tiloista on tutkittava elementti, kyseisillä ehdoilla :

List ▼

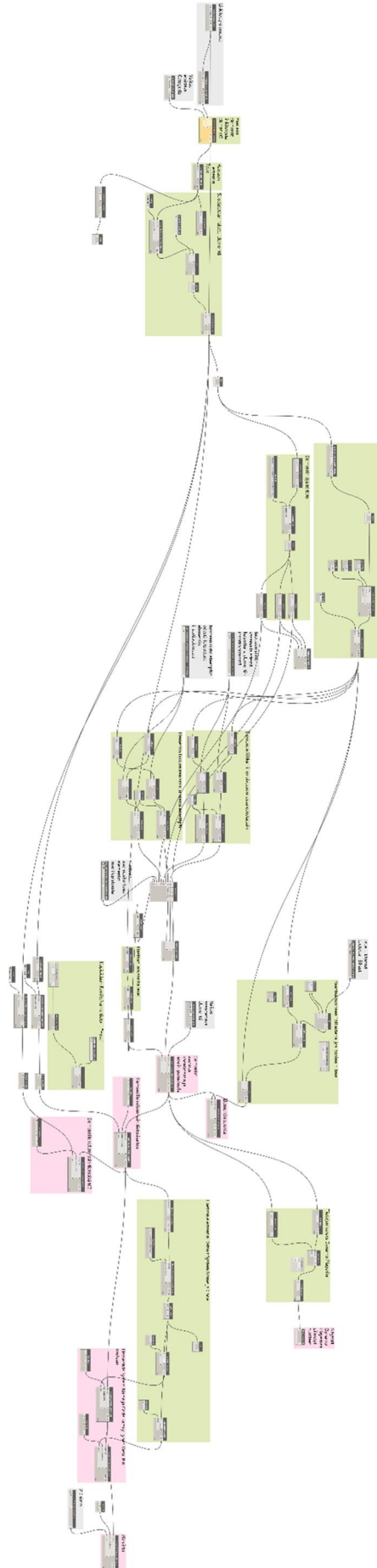
- 0 Office 210
- 1 Office 211
- 2 Office 212
- 3 Office 213
- 4 Office 214

Kuva 67. Dynamo Playerin näkymä, kun etsittiin, puuttuiko savuilmaisin tai lämpöilmaisin tiloista

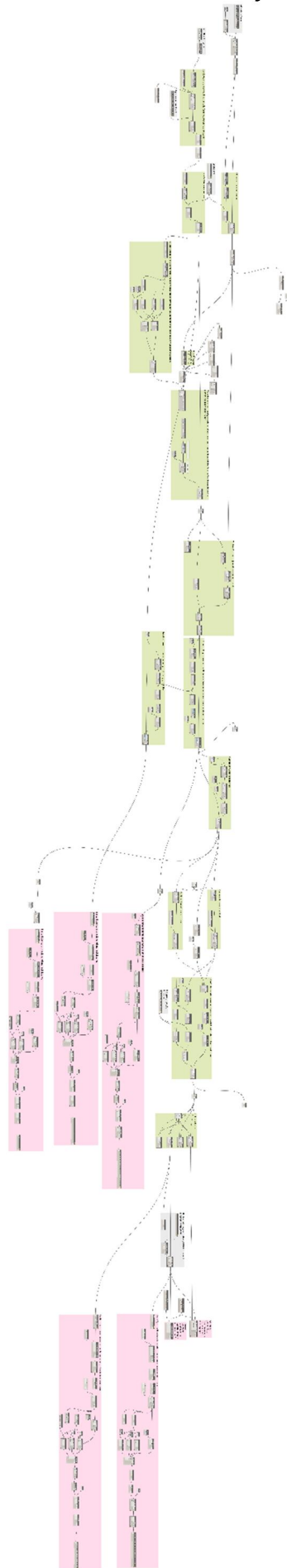
Skripti: määrälaskenta mallitilojen avulla



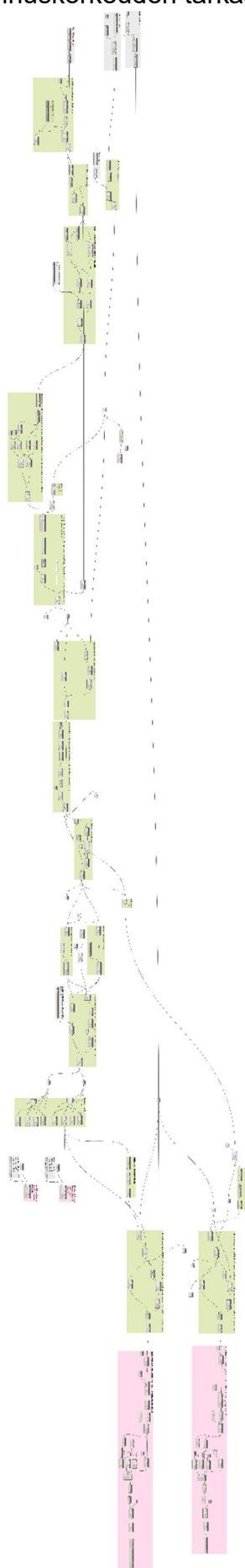
Skripti: Elementtien automaattinen mallintaminen muiden suunnittelualojen tietomallin perusteella



Skripti: Elementtien asennuskorkeuden tarkastaminen ja muuttaminen alakaton perusteella



Skripti: Johtoteiden asennuskorkeuden tarkastaminen alakaton korkeuden suhteen



Skripti: Puuttuvien elementtien listaaminen automaattisesti

