



SEINÄJOEN AMMATTIKORKEAKOULU  
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Hannu Laurila

---

## Automaattisen paperirullien korkeavaraston simulointi

Opinnäytetyö  
Kevät 2022  
Automaatiotekniikka



SEINÄJOEN AMMATTIKORKEAKOULU

## Opinnäytetyön tiivistelmä

Tutkinto-ohjelma: Automaatiotekniikka

Suuntautumisvaihtoehto: Sähköautomaatio

Tekijä: Hannu Laurila

Työn nimi: Automaattisen paperirullien korkeavaraston simulointi

Ohjaaja: Toni Luomanmäki

Vuosi: 2022

Sivumäärä: 42

---

Opinnäytetyön toimeksiantajana toimi Pesimal Oy, joka valmistaa varastointi ja pakkausjärjestelmiä paperi-, metalli- sekä rengasteollisuudelle. Tytäryhtiöt, laaja edustajaverkosto ja järjestelmätoimitukset ympäri maailman tekevät yrityksestä kansainvälisen toimijan.

Työn tavoitteena oli luoda simulointimalli suunnitteilla olevasta paperirullavarastosta ja sen avulla testata, riittävätkö korkeavarastohissien ja kuljettimien kapasiteetit suunnitelluille rullamäärille. Lisäksi tavoiteltiin skaalautuvaa mallia, jonka osat olisivat hyödynnettävissä myös tulevilla simulointiprojekteilla.

Työ aloitettiin projektin määrittelyllä, jonka mukaan varaston mallia kehitettiin sekä kartoitettiin halutut lisäominaisuudet. Simulointimalli luotiin Visual Components Professional 4.3 -ohjelmistolla. Mallin luomisprosessin aikana kehitettiin oma WMS-ohjainlaite, jotta lisäominaisuuksia pystyttiin toteuttamaan. Työn aikana perehdyttiin 3D-mallintamiseen, Stackless Python -komentokieleen, taulukoiden ja kirjastojen luomiseen, tiedon tallentamiseen ja muokkaamiseen sekä tiedon siirtoon.

Lopputuloksena saatiin malli varastosta, jonka avulla voitiin määrittellä korkeavarastohissien kapasiteetin sekä kuljettimien määrän riittävyys. Vaadittua skaalautuvuutta saatiin toteutettua WMS-ohjainlaitteeseen ja korkeavarastohisseihin.

<sup>1</sup> Asiasanat: tietokonesimulointi, 3D-mallinnus, Python

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

## Thesis abstract

Degree programme: Automation Engineering

Specialisation: Electric Automation

Author: Hannu Laurila

Title of thesis: Simulation of an Automatic High Bay Warehouse for Paper Rolls

Supervisor: Toni Luomanmäki

Year: 2022

Number of pages: 42

---

The thesis was commissioned by Pesmél Oy, which produces warehousing and packing solutions for paper, steel, and tire industry. Subsidiaries, wide network of representatives and system deliveries around the world make the company an international operator.

The objective of the thesis was to produce a warehouse simulation model. The goal was to test the capacity of stacker cranes and conveyors and an extra objective was to create a model that would have scalable parts. That way the parts of the model would be reusable also in other simulation projects in the future.

This thesis project was started by defining the model of the warehouse and listing the necessary extra features. Based on these specifications the development project was started. The finished model was created by using Visual Components 4.3 Professional software. To be able to execute the desired extra features, a new WMS-controller was created during this project. The thesis focused on 3D-modeling, Stackless Python coding language, creating tables and libraries, as well as saving, editing, and moving data between the components of the model.

The outcome of the thesis was a model of a warehouse which could provide answers to questions concerning the capacity demands of stacker cranes and conveyors. The wanted scalability of the parts was achieved for the WMS-controller and stacker cranes.

<sup>1</sup> Keywords: computer simulation, 3D-modeling, Python

## SISÄLTÖ

<b>Opinnäytetyön tiivistelmä</b> .....	1
<b>Thesis abstract</b> .....	2
SISÄLTÖ .....	3
Kuva- ja kuvioluettelo.....	5
Käytetyt termit ja lyhenteet.....	7
<b>1 JOHDANTO</b> .....	8
1.1 Työn tausta .....	8
1.2 Työn tavoite.....	8
1.3 Työn rakenne .....	9
1.4 Pesmel Oy.....	9
<b>2 SIMULOINTI</b> .....	10
2.1 Mitä simulointi on?.....	10
2.2 Simuloinnin hyötyjä ja haittoja .....	11
2.3 Simulointiprojektin eteneminen.....	12
2.4 Varastonhallintajärjestelmä ja TransRoll-korkeavarasto.....	13
<b>3 VISUAL COMPONENTS</b> .....	15
3.1 Ohjelmisto .....	15
3.2 Stackless Python -ohjelmointikieli .....	16
3.3 Python API .....	17
<b>4 PROJEKTINMÄÄRITTELY</b> .....	21
4.1 Projektin suunnittelu, lähtötiedot ja resurssit .....	21
4.2 Tekniset tavoitteet .....	22
4.3 Dokumentointi .....	22
<b>5 SIMULOINTIMALLIN KEHITTÄMINEN</b> .....	23
5.1 Mallien luonti .....	23
5.1.1 Hylly .....	24
5.1.2 WMS-ohjainlaite.....	26
5.1.3 Korkeavarastohissit.....	26
5.1.4 Tuotteiden luonti .....	27

5.1.5	Ulossyötön ominaisuudet ja tilausjärjestelmä.....	29
5.1.6	Sisäänsyötön ominaisuudet .....	29
5.2	Taulukoiden luominen .....	29
5.2.1	Ohjainlaitteen taulukot .....	30
5.2.2	Korkeavarastohissin kirjasto .....	31
5.3	Tiedonsiirto.....	33
5.3.1	WMS ja korkeavarastohissin välinen tiedonsiirto .....	34
5.3.2	Rullatiedon tallentaminen WMS-ohjainlaitteelle .....	36
5.4	Statistiikka .....	36
6	TULOKSET JA JATKOKEHITYS.....	39
7	POHDINTA.....	41
	LÄHTEET .....	42

## Kuva- ja kuvioluettelo

Kuva 1. Mallinnettu TransRoll-korkeavarasto. ....	13
Kuva 2. Korkeavarastohissi. ....	14
Kuva 3. Visual Components Premium 4.3 -käyttöliittymä.....	15
Kuva 4. PNP-toiminto.....	16
Kuva 5. Python-komentotiedostoesimerkki. ....	17
Kuva 6. Erilaisia lisättäviä behaviour-elementtejä.....	18
Kuva 7. Korkeavarastohissin linkkihierarkia.....	19
Kuva 8. Korkeavarastohissin x-, y-, ja z-akselit.....	20
Kuva 9. Mallin osat ilman hyllyä.....	23
Kuva 10. TransRoll-kanavaprofiili. ....	24
Kuva 11. Komponentin ja geometrian luominen. ....	24
Kuva 12. LinearClone-työkalun käyttö kopiointiin. ....	25
Kuva 13. Valmis varastohylly.....	25
Kuva 14. Hissin geometriat ja skaalaus. ....	27
Kuva 15. Kappalegeneraattorin koodi.....	28
Kuva 16. Ohjainlaitteen hyllytaulukon luonti.....	31
Kuva 17. WorldPositionMatrix-komento.....	31
Kuva 18. Korkeavarastohissin kirjaston luominen.....	32
Kuva 19. Varastokiskojen etäisyydet. ....	33
Kuva 20. Hyllystä luettavia ominaisuuksia. ....	34

Kuva 21. Tiedonsiirto. ....	35
Kuva 22. Korkeavarastohissin ominaisuudet. ....	36
Kuva 23. Malliin toteutettu statistiikka. ....	37
Kuva 24. Statistiikan ominaisuudet. ....	38
Kuvio 1. Korkeavarastohissin tehtävän määrittely. ....	26
Kuvio 2. Korkeavarastohissin kirjaston laskenta. ....	33
Kuvio 3. Signaalit laitteiden välillä. ....	40

## Käytetyt termit ja lyhenteet

<b>API</b>	Application programming interface eli ohjelmointirajapinta ohjelmien väliseen tiedonvaihtoon.
<b>Herkkyysanalyysi</b>	Herkkyysanalyysi on mallin tuloksen epävarmuuden arvioimista. Sen avulla pyritään selvittämään, kuinka herkkä mallin tulos on eri lähtöarvojen muuttamiselle.
<b>WMS</b>	Warehouse management system tarkoittaa tietokonepohjaista varastohallintajärjestelmää.
<b>TransRoll</b>	Paperirullien varastointijärjestelmä, jossa rullat varastoidaan kyljelleen erityisen kiskon päälle.



# 1 JOHDANTO

## 1.1 Työn tausta

Simulointi on todellisen tapahtuman tai järjestelmän mallintamista ajan suhteen (Banks ym., 2010, s. 21). Mallinnusta voidaan suorittaa manuaalisesti tai tietokoneavusteisesti. Näin saadusta keinotekoisesta historiatiedosta voidaan päätellä oikean järjestelmän operaationaalista toimintaa.

Banks ym. (2010, s. 21) mukaan järjestelmän toimintaa ja sen muutosta tutkitaan kehittämällä simulaatiomalli. Mallin luominen aloitetaan tekemällä tiettyjä oletuksia koskien järjestelmän toimintaa. Oletuksien kuvaamiseen käytetään matemaattisia, loogisia ja symbolisia suhteita järjestelmän osien välillä. Kun malli on saatu valmiiksi, sen avulla voidaan tutkia monia erilaisia kysymyksiä koskien oikeaa järjestelmää. Järjestelmään tehtävien muutoksien testaaminen simulointimallin avulla auttaa ennustamaan vaikutusta järjestelmän toimintaan. Simulointimallilla on käyttöä myös suunnitteluvaiheessa, ennen kuin koko järjestelmää on edes oikeasti olemassa. Simuloitua mallia voidaan siis käyttää olemassa olevan järjestelmän analysointiin muutoksia tehdessä sekä uusien järjestelmien suorituskyvyn tarkasteluun.

Älykkään simuloinnin avulla voidaan asiakkaalle toimitettavaa monitahoista järjestelmää testata, sekä sen avulla poistaa mahdollisia pullonkauloja jo ennen kuin järjestelmää aletaan valmistaa (Pesmel Oy, 2018, s. 2). Tämän älykkään simuloinnin avulla saavutetaan kustannussäästöjä ja nopeutetaan järjestelmän käyttöönottoa. Mallit toteutetaan yhteistyössä asiakkaiden kanssa, ja niillä voidaan tutkia esimerkiksi kapasiteetin riittävyyttä tai tehtaan layoutin toimivuutta. Asiakkaalta saadaan tietoa järjestelmän toiminnasta ja tiimin tehtävänä on päättää mallin yksityiskohtaisuudesta. Asiakkaan osallistuminen simulointiprojektiin on ensiarvoista, jotta luotavasta mallista saadaan maksimaalinen hyöty irti.

## 1.2 Työn tavoite

Työllä on tarkoitus tutkia varastoon tulevien kuljettimien ja korkeavarastohissien kapasiteetin riittävyyttä. Lisätavoitteena on luoda yrityksen käyttöön korkeavarastosta simulointimalli, joka olisi helposti muunnettavissa erilaisiin asiakasprojekteihin. Malliin kehitetään varastonhallinta-järjestelmä, joka pitää huolta varastokirjanpidosta, sekä tehtävien jakamisesta toimilaitteille.

Yrityksessä aikaisemmin Visual Components -ohjelmistolla tehdyissä simulointitöissä toiminnallisuus on koodattu suoraan esimerkiksi korkeavarastohissiin eikä erillistä varastohallintajärjestelmää ole mallinnettu. Simulointimalli luodaan käyttäen Visual Components -ohjelmistoa. Lisäksi työssä perehdytään ohjelmassa käytettyyn Stackless Python -ohjelmointikieleen.

### 1.3 Työn rakenne

Opinnäytetyön alussa olevassa johdannossa esitellään työn taustat, tavoitteet ja työn toimeksiantaja Pesmel Oy. Teoriaosuudessa käydään läpi simulointia ja simulointiprojektin etenemistä. Lisäksi perehdytään Pesmel Oy:n TransRoll-korkeavarastoon ja yrityksen WMS-järjestelmän perusteisiin. Teoriaosuudessa esitellään myös työn tekemiseen käytetty Visual Components Professional -ohjelmisto, sekä käydään hieman läpi ohjelman keskeisimpiä ominaisuuksia, kuten Python API.

Työn osuus aloitetaan projektin määrittelyllä, jossa käydään läpi työn teknisiä tavoitteita, sekä muuttujia, joita työn avulla halutaan tutkia. Lisäksi määritellään projektin resurssit, lähtötiedot ja dokumentointi. Viidennessä luvussa käsitellään itse simulointimallin luomista Visual Components -ohjelmistolla, sekä tiedonsiirtoa toimilaitteiden välillä. Myös malliin kirjoitettujen Python-komentotiedostojen sisältöä ja toiminnallisuutta selitetään. Lopussa käydään tulokset läpi ja esitetään jatkokehityskohteita.

### 1.4 Pesmel Oy

Kauhajoella vuonna 1978 perustettu Pesmel on erikoistunut valmistavan teollisuuden materiaalivirtojen ja logistiikan hallintaan (Pesmel Oy, 2022). Yrityksen konseptina on *Material Flow How®*, jonka mukaisesti yritys tarjoaa asiakkailleen automatisoituja varastointi-, pakkaus-, ja käsittelyratkaisuja. Näitä ratkaisuja hyödyntämällä pyritään tehostamaan asiakkaan materiaalivirtoja, varastointia, tilankäyttöä, toimituksia, työturvallisuutta ja läpimenoaikoja.

Pesmel on vuosien saatossa toimittanut yli 400 materiaalin käsittelyratkaisua sellu-, paperi-, metalli-, ja rengasvalmistajille (Pesmel Oy, 2022). Tytäryhtiöitä on Pohjois-Amerikassa, Euroopassa ja Aasiassa. Lisäksi yrityksellä on laaja edustajaverkosto ympäri maailmaa.

## 2 SIMULOINTI

### 2.1 Mitä simulointi on?

Shannonin (1998, s. 7) mukaan simulaatio on tehokas työkalu suunniteltaessa monimutkaisia prosesseja ja järjestelmiä. Simulointi mahdollistaa tutkimisen, analysoinnin ja arvioinnin erilaisissa tilanteissa. Tämän vuoksi simuloinnista on tullut välttämätön ongelmaratkaisutyökalu koko suunnitteluketjulle. Mallin suunnittelussa on otettava huomioon se, että mallin käyttäytyminen jäljittelee mahdollisimman tarkasti oikean järjestelmän käyttäytymistä, jotta saatu tieto on totuuden mukaista. Simulointi on kokeellista ja soveltavaa ja sen tavoitteena on:

- Kuvata järjestelmän toimintaa.
- Mallin avulla ennustaa järjestelmän toimintaa, kun järjestelmään tai sen toimintatapaan tehdään muutoksia.

Joissain tapauksissa on mahdollista ratkaista ongelma kehittämällä järjestelmästä yksinkertainen matemaattinen malli (Banks ym., 2010, s. 21). Ratkaisuja voidaan etsiä käyttämällä esimerkiksi differentiaali- ja todennäköisyyslaskentaa, algebraa sekä myös muita matemaattisia tekniikoita. Näiden avulla voidaan laskea numeerisia suorituskyvyn arvoja ja optimoida niitä. Heliö (2010, s. 17–39) viittaa laskentatehon kasvamiseen, joka mahdollistaa vaativampienkin matemaattisten mallien kehityksen. Matemaattisen mallinnuksen onnistuminen ja tätä kautta järjestelmän toiminnan arvioiminen vaatii järjestelmäyhtälöiden muodostamista ja niiden ratkaisemista. Käytännössä tämä tarkoittaa, että tietokone kokeilee eri muuttujia ja taulukoi syntyneitä ratkaisuja. Matemaattiset mallit ovat tehokas apuväline tutkimus- ja kehitystehtävissä.

Todellisen maailman järjestelmät ovat kuitenkin yleensä niin monimutkaisia syy-seuraussuhteeltaan, että matemaattinen ratkaisu käy mahdottomaksi (Banks ym., 2010, s. 21–22).

Näissä tapauksissa simuloinnin avulla voidaan jäljitellä järjestelmän toimintaa ajan suhteen. Tietoja saadaan simulointimallista samalla tavalla kuin tarkkailtaisiin oikeaa järjestelmää. Kerätyn tiedon avulla voidaan arvioida järjestelmän suorituskykyä.

## 2.2 Simuloinnin hyötyjä ja haittoja

Asiakkaan näkökulmasta simulointi on miellyttävää, koska se jäljittelee oikeaa järjestelmää tai antaa ennakkotietoa suunnitteilla olevasta järjestelmästä (Banks ym., 2010, s. 23). Simulointimallista saadun tiedon tulisi vastata suoraan oikeasta järjestelmästä saatuja tietoja. Simulointimallissa voidaan myös välttää kyseenalaisten oletusten huomioiminen, kuten esimerkiksi matemaattisista malleista tutun keskijakauman jokaiselle sattumanvaraiselle muuttujalle. Simulaatiomalli eroaa optimointimallista siten, että sillä ei pyritä ratkaisemaan mitään tiettyä muuttujaa. Käytössä olevasta tai vasta suunnitteilla olevasta järjestelmästä tehdyllä simulaatiomallilla ajetaan simulaatioita eri parametreillä ja tarkastellaan lopputulosta. Saatuja tuloksia vertaillaan ja parasta suositellaan otettavaksi käyttöön. Simulaatiomallia tulisikin kohdella työkaluna prosessin optimoinnissa (Shannon, 1998, s. 8).

Shannon (1998, s. 7–8) listaa edellä mainittujen lisäksi seuraavia simuloinnin hyötyjä:

- Uusien layoutien ja suunnitelmien testaaminen on halvempaa verrattuna käytännön toteuttamiseen.
- Voidaan kokeilla uusia toimintamenetelmiä ilman, että tuotantoa tarvitsee keskeyttää.
- Tuotannon pullonkauloja voidaan tarkastella kapasiteettia nostettaessa.
- Simuloinnissa voidaan kokeilla, miksi jokin asia tapahtuu.
- Aikaa voidaan kontrolloida ja kuukausien ajo voidaan suorittaa nopeasti.
- Saadaan ymmärrystä, mitkä muuttujat ovat järjestelmän kannalta kriittisiä.
- Simuloinnin suurin vahvuus on, että voidaan kokeilla monenlaisia järjestelmälle epätyypillisiä tilanteita.

Banks ym. (2010, s. 24) kertovat simuloinnin suurimmiksi haitoiksi mallien rakentamisen taidon opetteluun, tuloksien tulkitsemisen, analysointiin käytetyn ajan ja kustannukset sekä turhan simuloinnin. Simulointimallinnus vaatii koulutusta, taitoa ja käytettävän ohjelmiston hallitsemista (Shannon, 1998, s. 8). Mallin hyödyllisyys riippuu suuresti mallin laadusta ja sen tekijän taidoista. Alkutietojen keräämiseen voi kulua paljon aikaa ja vastavuoroisesti tulokseksi saatu tieto on vaikeasti tulkittavaa. Tämän vuoksi mallin tekemiseen ja tulkitsemiseen käytetyistä resursseista säästäminen voi johtaa malliin, jonka tarkkuus ei ole riittävä annetun tehtävän selvittämiseksi (Banks ym., 2010 s. 24). Lisäksi simuloinnin käyttö sellaisissa tilanteissa, joissa olisi järkevämpää käyttää esimerkiksi analyttistä ratkaisumallia syö simuloinnin etuja.

### 2.3 Simulointiprojektin eteneminen

Simulaatio tulisi ottaa osaksi suunnittelu prosessia jo heti projektin alkumetreillä (Pesmel Oy, 2018, s. 3). Simulaatiosta saatavasta hyödyllisestä informaatiosta on apua valmiin järjestelmän optimaalisen toiminnan varmistamisessa. Simulointi antaa hyvän yleiskuvan projektista, niin suunnittelijoille kuin myös asiakkaalle. Kun mallissa vielä otetaan huomioon tehtaan todellinen toiminta, saadaan luotua malli, jonka tarkkuuteen ei yllätä muilla suunnittelun työkaluilla.

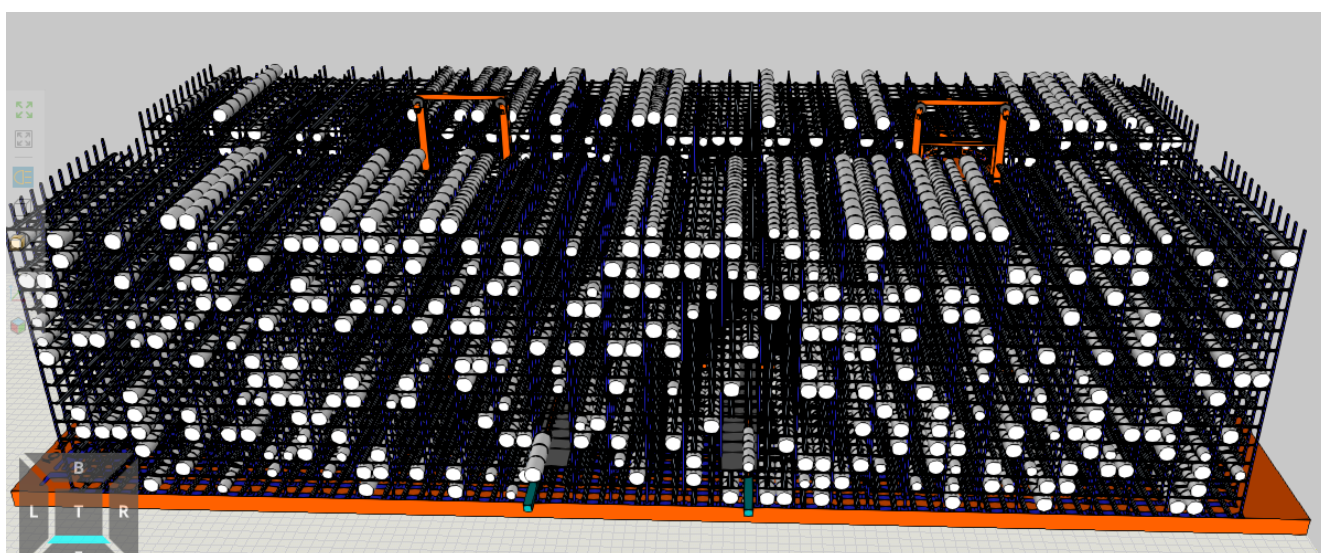
Shannon (1998, s. 9) kertoo simuloinnin tarkoituksiksi päätöksentekijän ongelman ratkaisun helpottamisen. Tästä seuraa, että hyvän simuloijan tulee omata ongelmaratkaisu- ja ohjelmointitaitoja. Hän listaa seuraavat askelmat, jotka jokaisesta simulointityöstä tulisi löytyä:

1. Ongelman selkeä määrittely, josta selviää simuloinnin tarkoitus.
2. Projektin suunnittelussa tulee ottaa huomioon, että käytettävissä on riittävät resurssit, kuten henkilöstö ja työkalut.
3. Järjestelmän määrittelyssä määritellään rajat ja rajoitukset, joiden avulla järjestelmää kuvataan, sekä tutkitaan järjestelmän toimintaa.
4. Luodaan alustava malli lohkokaaaviona, prosessikaaviona tai luonnoskieltä käyttäen, jotta saadaan määriteltyä komponentit, kuvaavat muuttujat ja logiikka järjestelmän taustalla.
5. Päätetään, mitä tutkitaan, ja mitkä muuttujat vaikuttavat tutkittavaan kohteeseen, eli mitä dataa kerätään, missä muodossa ja missä määrin.
6. Tunnistetaan ja kerätään malliin tarvittavaa input-tietoa.
7. Luodaan malli käyttäen työhön sopivaa simulointityökalua.
8. Tarkastellaan mallin toimintaa, jotta voidaan todeta sen toimivan suunnitellulla tavalla. Tästä seuraa, että mallin tuottama tieto vastaa oikean järjestelmän tuottamaa tietoa.
9. Suunnitellaan koeajot, jotta päästään vastaamaan alkuperäiseen tutkimuskysymykseen.
10. Suoritetaan simulaatio halutun datan keräämiseksi ja tehdään herkkyyshanalyysi.
11. Tehdään päätelmät simulaatioajojen tuottamasta tiedosta.
12. Raportoidaan tulokset ja otetaan ne käyttöön. Lisäksi dokumentoidaan myös löydökset ja malli sekä sen käyttö.

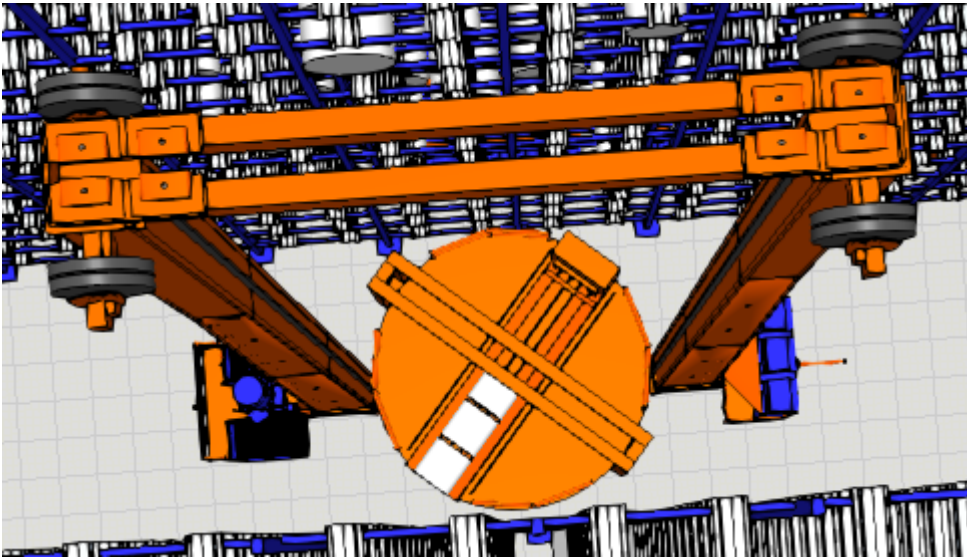
## 2.4 Varastonhallintajärjestelmä ja TransRoll-korkeavarasto

Varastonhallintajärjestelmä, WMS (Warehouse Management System), huolehtii materiaalivirroista tuotannon, varastoinnin, rullien käsittelyn ja lähettämön välillä (Pesmel, 2020, s. 3). Toimintoja ohjaava tietokonejärjestelmä on täysin integroitavissa tehtaalla olemassa oleviin järjestelmiin. WMS-järjestelmän tärkein tehtävä on kontrolloida materiaalivirtoja eri prosessien välillä sekä optimoida varastointia ja logistiikkaa. WMS-järjestelmä nostaa tehtaan kapasiteettia ja vähentää kustannuksia. Lisäksi kyseinen järjestelmä sisältää laajan työkalupakin diagnostiikka ja raportointityökaluja. Tässä työssä WMS-järjestelmään perehdytään luotavan ohjainlaitteen toimintojen hahmottamiseksi.

TransRoll-korkeavarasto (kuva 1) perustuu syväkanavatekniikkaan, joka soveltuu erittäin hyvin isoihin varastointimääriin, ja suuriin kapasiteetteihin (Pesmel, 2020, s. 4). Varasto pystyy käsittelemään rullia riippumatta rullien mitoista tai pakkaustavoista. Varastoissa käytettyjen korkeavarastohissien (kuva 2) normaalikapasiteetin ollessa noin 40–50 työsykliä tunnissa ja rullamäärän 6–8 rullaa kerralla, voidaan sisään- ja ulossyötössä päästä 400 rullaan tunnissa korkeavarastohississä kohden. Työssä luodaan malli TransRoll-korkeavarastosta, jossa on kaiken kaikkiaan 1950 x- ja y-osoitetta. Varastossa on yhteensä noin 34 kilometriä syväkanavaa, sekä kaksi korkeavarastohissiiä.



Kuva 1. Mallinnettu TransRoll-korkeavarasto.



Kuva 2. Korkeavarastohissi.

TransRoll-korkeavarastoa hyödyntämällä aikaan saadut säästöt ja toiminnan tehostuminen perustuvat seuraaviin (Pesmel, 2020, s. 4) ominaisuuksiin:

- yksinkertaisempi layout ja minimaalinen määrä kuljettimia
- varaston korkeus, leveys ja syvyys muokattavissa
- pienin varaston pinta-ala verrattuna vaihtoehtoihin
- erillistä varastorakennusta ei tarvita, vaan seinät ja katto voidaan tukea hyllykköön
- tulipaloturvallisuus osiin jaetulla sprinklerijärjestelmällä.

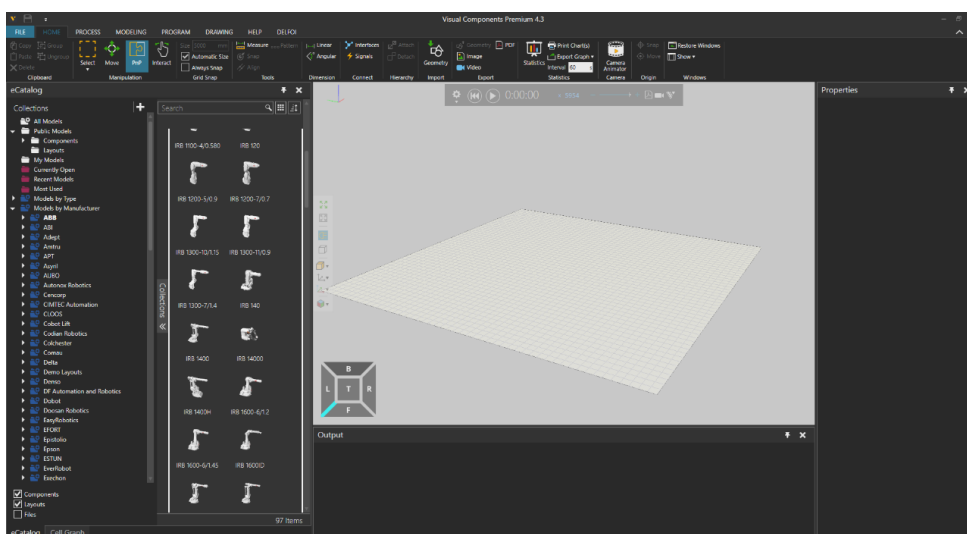
### 3 VISUAL COMPONENTS

Visual Components on vuonna 1999 perustettu yritys, jonka tavoitteena on ollut alusta asti luoda tuotannosuunnitteluun ja simulointiin helppokäyttöinen työkalu kaikenkokoisille organisaatioille (Visual Components Oy, 2022a). Nykyään yritys on markkinajohtaja tuotantosimuloinnin alalla ja monien johtavien brändien kumppani tuotantolinjojen suunnittelussa ja simuloinnissa.

#### 3.1 Ohjelmisto

Ohjelmisto on suunniteltu tuotannon ammattilaisille ja se on rakennettu tehokkaalle, joustavalle ja skaalautuvalle alustalle (Visual Components Oy, 2022b). Ohjelmasta on saatavilla eri laajuisia versioita riippuen mihin käyttöön ohjelmistoa ollaan hankkimassa. Tässä työssä käytän ohjelmiston Professional-versiota. Saatavilla olisi vielä Premium-versio, jossa lisänä tulee rajapinnat Siemensin S7-logiikalle sekä erilaisille robottiohjaimille.

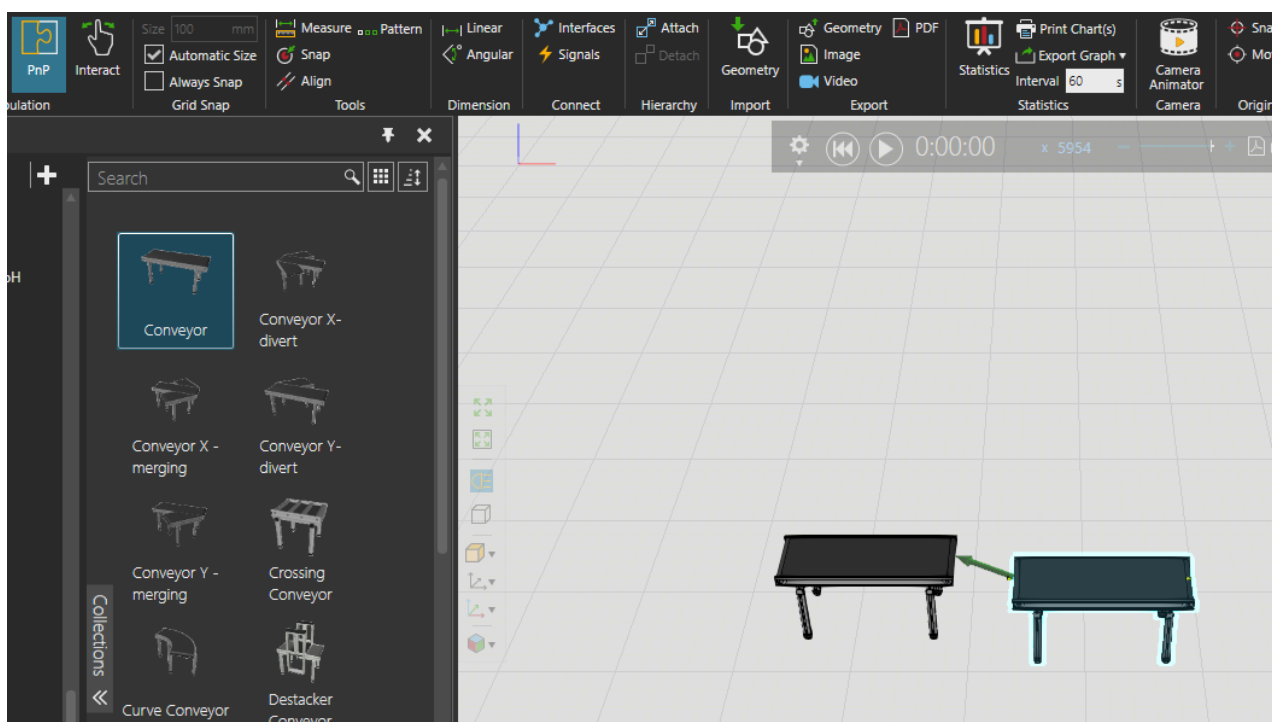
Ohjelman mukana tulee eCatalog-kirjasto, jossa on valmiita toimilaitteita (kuva 3). Tässä työssä ei voitu käyttää valmiita komponentteja, sillä yrityksen valmistamat tuotteet eroavat huomattavasti eCatalog-kirjastossa olevista valmiista toimilaitteista. Työssä esiintyvät korkeavarastohissit ja kuljettimet, sekä pystyynnostajat löytyivät valmiina yrityksen omista kirjastoista, joskin niitä piti työtä varten hieman skaalata. Hyllystä luotiin 3D-malli 2D-piirrustusten pohjalta.



Kuva 3. Visual Components Premium 4.3 -käyttöliittymä.



Ohjelman valmiiden komponenttien käyttäminen on hyvin yksinkertaista, sillä niihin on koodattu tarpeelliset toiminnallisuudet valmiiksi. Niitä voi raahata valikosta malliin ja liittää toisiinsa PNP-toiminnon avulla (kuva 4). Lisäksi niiden ominaisuuksia, kuten esimerkiksi pituutta, korkeutta, nopeutta ja leveyttä, voidaan muuttaa helposti. Käyttöliittymä on hyvin avoin ja valmiiden toimilaitteiden koodia pääsee tarkastelemaan ja tarvittaessa muuttamaan. Mikäli oikeanlaisia komponentteja ei löydy valmiina, ohjelmaan voidaan myös luoda täysin omia toimilaitteita ja niille toiminnallisuutta. Omien toimilaitteiden luonti on helppoa tuomalla valmis 3D-piirros ohjelmaan. Visual Components -ohjelmalla pystyy mallintamaan, mikäli valmista 3D-mallia ei ole. Tässä työssä hyllystö on mallinnettu käyttäen ohjelman omia geometriatyökaluja. Toiminnallisuus näille omille komponenteille luodaan itse käyttäen Python-komentosarjoja.



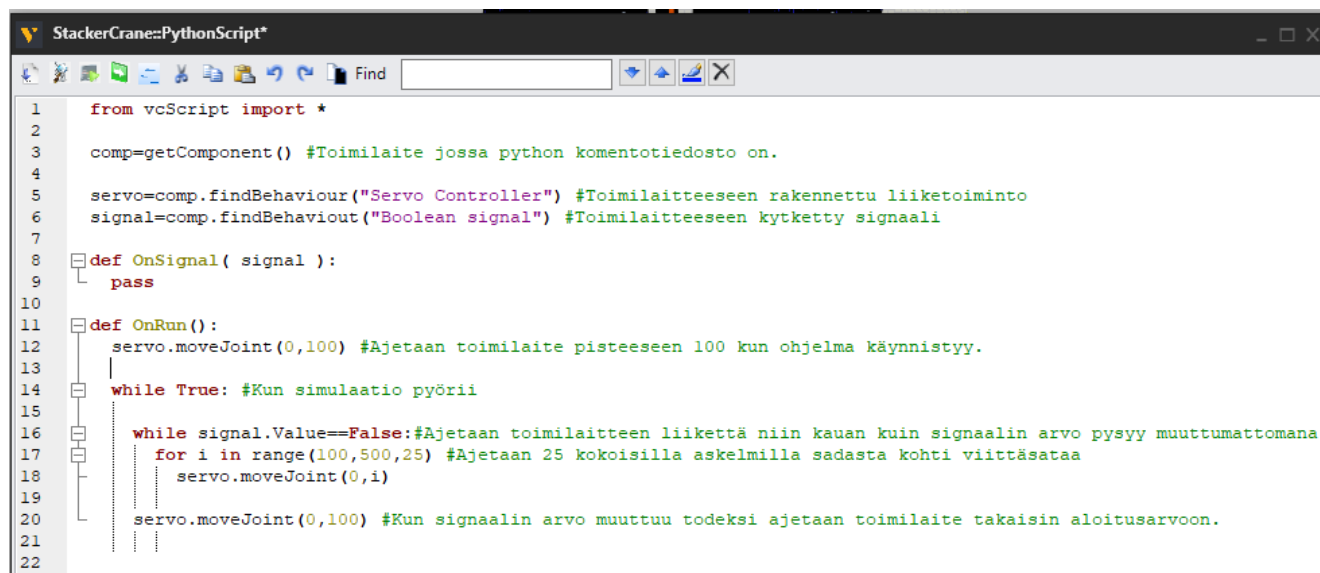
Kuva 4. PNP-toiminto.

### 3.2 Stackless Python -ohjelmointikieli

Stackless Python -ohjelmointikieli on tehostettu versio perinteisestä Python-ohjelmointikielistä (Kruis & Tew, 2021). Se tuo thread-pohjaisen ohjelmoinnin hyödyt ilman niiden monimutkaisuutta ja vaikutusta suorituskykyyn. Tämä ohjelmointikieli mahdollistaa rinnakkaisten tehtävien ajamisen yhtä aikaa, jolloin koodia voidaan hajauttaa eri toimilaitteille.

### 3.3 Python API

Python-ohjelmointikielen ja ohjelmointirajapinnan avulla voidaan toteuttaa omia ratkaisuja (Visual Components Oy, 2022c). Tässä työssä ei käytetty valmiita komponentteja, joten kaikki koodattiin itse. Python-komentosarjoja lisäämällä komponentteihin voidaan luoda toiminnallisuutta ja logiikkaa (Visual Components Oy, 2015). Komentosarjojen avulla voidaan muuttaa komponentin ominaisuuksia tai antaa sille tehtäviä (Visual Components Oy, 2017, s.1). Komponentin sisällä voi olla useita eri komentosarjatiedostoja, joilla jokaisella on oma tehtävänsä, esimerkiksi ensimmäinen ohjaa liikettä ja toinen vaihtaa kappaleen värin, kun liike saavuttaa jonkin tietyn kohdan.



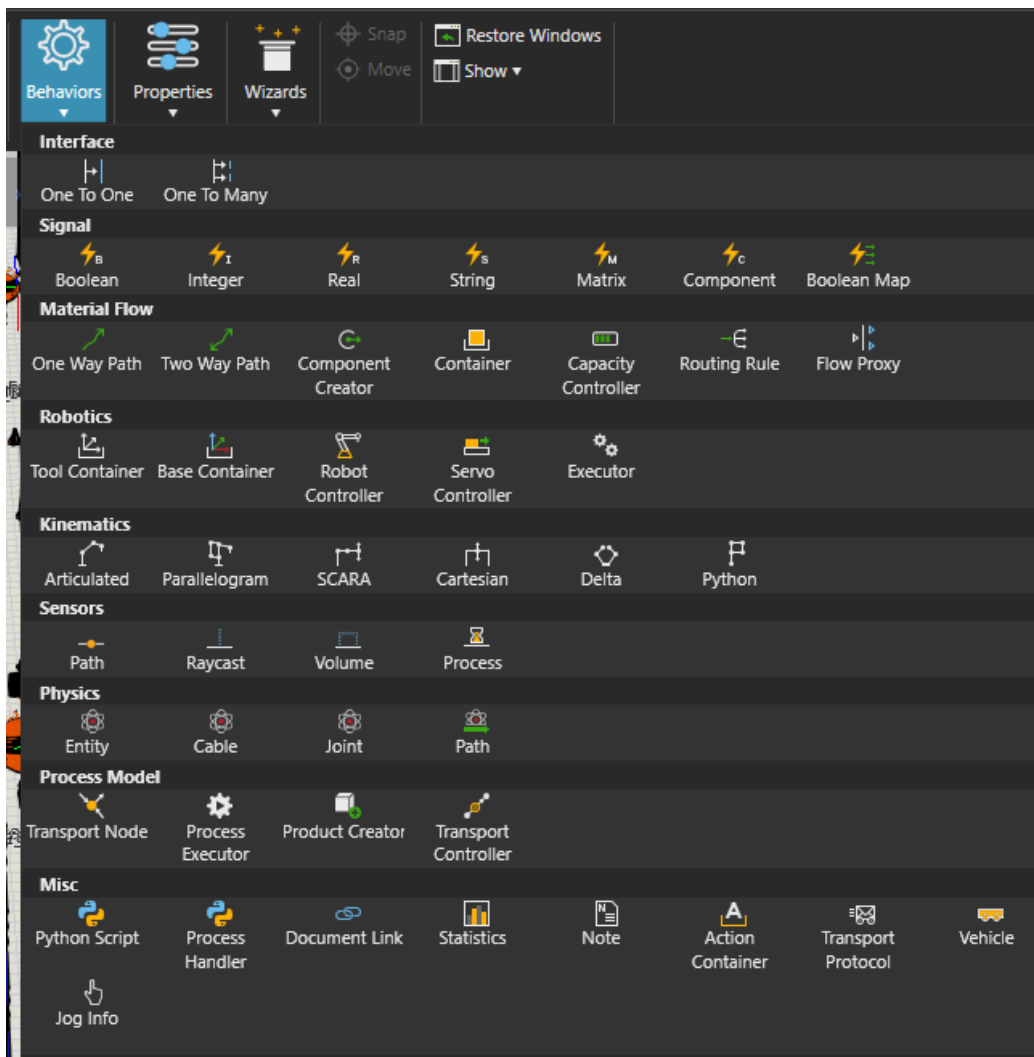
```

1  from vcScript import *
2
3  comp=getComponent() #Toimilaite jossa python komentotiedosto on.
4
5  servo=comp.findBehaviour("Servo Controller") #Toimilaitteeseen rakennettu liiketoiminto
6  signal=comp.findBehaviour("Boolean signal") #Toimilaitteeseen kytketty signaali
7
8  def OnSignal( signal ):
9      pass
10
11 def OnRun():
12     servo.moveJoint(0,100) #Ajetaan toimilaite pisteeseen 100 kun ohjelma käynnistyy.
13
14     while True: #Kun simulaatio pyörii
15
16         while signal.Value==False:#Ajetaan toimilaitteen liikettä niin kauan kuin signaalin arvo pysyy muuttumattomana
17             for i in range(100,500,25) #Ajetaan 25 kokoisilla askelmilla sadasta kohti viittäsataa
18                 servo.moveJoint(0,i)
19
20         servo.moveJoint(0,100) #Kun signaalin arvo muuttuu todeksi ajetaan toimilaite takaisin aloitusarvoon.
21
22

```

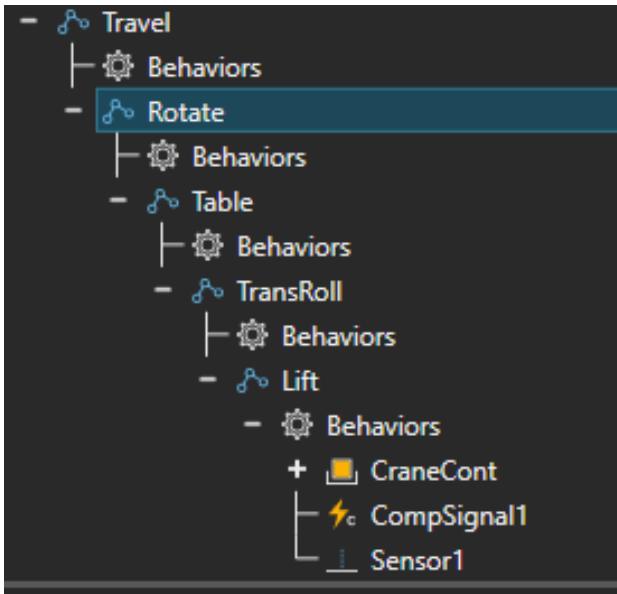
Kuva 5. Python-komentotiedostoesimerkki.

Kuvassa 5 on toteutettu yksinkertainen liikeohjaus. Kuvankaappaus on otettu malliksi toteutusta komentotiedostosta. Koodin alussa haetaan toimilaite GetComponent()-komennolla comp-muuttujaan. Tämän voisi kirjoittaa koodiinkin suoraan, mutta mikäli toimilaitteessa on useampia toiminnallisuksia, tämä lyhentäminen säästää huomattavasti aikaa ja tilaa, sekä tekee koodista helpommin luettavaa. Tämän jälkeen etsitään toimilaitteesta toiminnallisuksia eli behaviour-elementtejä. Niitä voi olla monenlaisia riippuen siitä millaista toiminnallisuutta laitteeseen halutaan (kuva 6), esimerkiksi signaaleja, kuljetinominaisuuksia, robotiikkaa, kinematiikkaa, sensoreita, komentotiedostoja, komponenttigeneraattoreita, yms. Lisäksi käytettävissä on properties-elementtejä, joiden avulla voidaan ilmaista komponenttien ominaisuuksia esimerkiksi mittoja tai vaikkapa väriä.



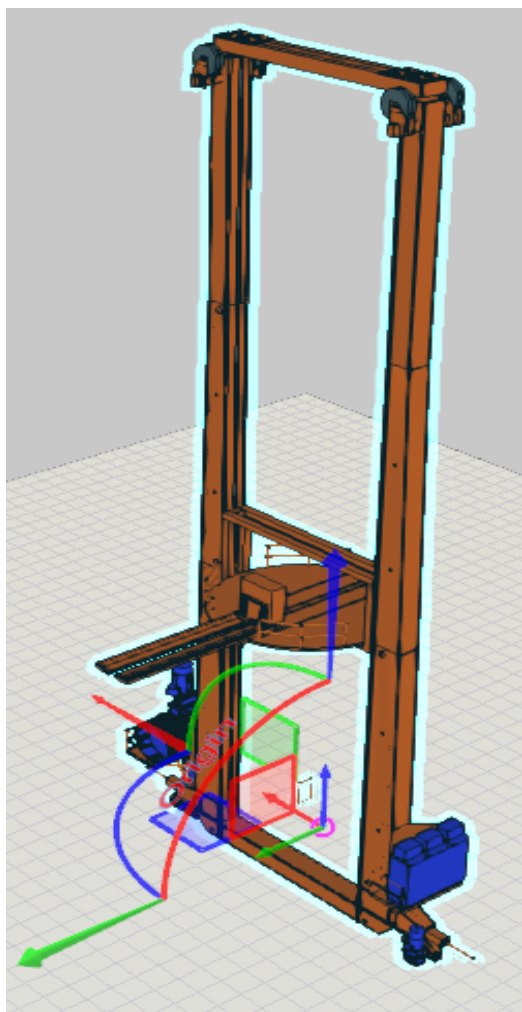
Kuva 6. Erilaisia lisättäviä behaviour-elementtejä.

Esimerkin toimilaitteesta löytyy servo, jonka avulla kontrolloidaan liikettä, sekä signaali, joka voi tulla vaikka simuloitulta anturilta. Tämän jälkeen esimerkissä liikutetaan toimilaitteeseen luotua niveltä servo.moveJoint-komennolla. Samalla tarkkaillaan signaalin arvoa, mikäli se muuttuu, niin ajetaan nivel takaisin alkuasemaan. Niveliä voi olla erisuuntaisia riippuen mihin suuntaan halutaan liikkua, esimerkiksi x-, y- ja z-suunnat sekä pyöritys eri akseleiden ympäri. Niveliä voidaan myös linkittää toisiinsa siten, että jos vaikka yksi osa liikkuu x-akselin suuntaan, yksi y-akselin suuntaan ja yksi z-akselin suuntaan, niin komponentteja yhdistetään näiden linkkien avulla siten, ettei kaikille osille tarvitse luoda esimerkiksi x- ja y-akselien suuntaisia liikkeitä, vaan osa, johon se on linkitetty, kuljettaa sitä mukanaan.



Kuva 7. Korkeavarastohissin linkkihierarkia.

Kuten kuvasta 7 nähdään, korkeavarastohissillä on useita liikkeitä, joiden linkittämisellä saadaan aikaan oikeanlainen toiminnallisuus. Kaikki osat kulkevat mukana, kun hissi kulkee käytävällä. Linkkihierarkiaa luodessa tulee huomioida tarkoin, mitkä osat tulevat minkäkin linkin alle. Esimerkiksi, jos Travel-linkki olisi Rotate-linkin alapuolella, silloin koko korkeavarastohissi pyörisi käytävällä. Antureita käytettäessä tulee myös huomioida niiden sijoittaminen linkkihierarkiaan oikein, jotta nekin kulkevat korkeavarastohissin mukana.



Kuva 8. Korkeavarastohissin x-, y-, ja z-akselit.

Kuvassa 8 on esitetty korkeavarastohissin liikesuuntia. Itse hissi liikkuu punaisen nuolen osoittamalla x-akselilla. Korkeavarastohississä oleva nostokori liikkuu sinisellä z-akselilla ylös alas, ja nostokorin kyydissä oleva TransRoll-vaunu liikkuu vihreän nuolen osoittamalla y-akselilla. Linkkien avulla TransRoll-vaunu liikkuu myös x- ja z-akselien suuntaan korkeavarastohissin kyydissä, näin sille ei tarvitse itselleen rakentaa kaikkia liikkeitä. Nostokori myös pyörii z-akselin ympäri ja TransRoll-vaunussa on nosto-ominaisuus rullien ottamista tai jättämistä varten sekä anturi rullien tunnistamiseen. Huomioitavaa Visual Components -ohjelmiston käytössä on se, että sen maailman koordinaatistossa z- ja y-akseleiden paikat ovat käänteiset, eli z-akseli osoittaa ylöspäin. Tässä työssä tehtiin kooditekniisiä ratkaisuja ja nimettiin z-akselia y-akseliksi, jotta koodi vastaisi yrityksen koordinaatistosuunnille käyttämää nimeämiskäytäntöä.

## 4 PROJEKTINMÄÄRITTELY

Projektin tarkoituksena oli luoda todellista järjestelmää mukaileva simulointimalli asiakkaalle toteutettavasta korkeavarastosta. Mallin avulla oli tarkoitus selvittää, ovatko korkeavarastohissien ja ulossyöttökuljettimien määrät riittävät suunnitelluilla ulossyöttönopeuksilla.

Mallia tullaan hyödyntämään myös tulevissa projekteissa, joten sen skaalautuvuuteen, kuten esimerkiksi korkeavarastohissien määrään, sekä hyllyn mittojen muutettavuuteen tuli kiinnittää huomiota.

### 4.1 Projektin suunnittelu, lähtötiedot ja resurssit

Projektin alussa pidettiin palaveri, jossa käytiin läpi varaston suunniteltua toimintaa. Joistain varaston osista tehtiin myös prosessikaavioita helpottamaan asian ymmärtämistä. Tietoja kerättiin ainakin seuraavista toiminnallisuuteen liittyvistä asioista:

- sisäänsyötön ominaisuudet ja kapasiteetti
- ulossyötön ominaisuudet ja kapasiteetti
- varastointilogiikka
- korkeavarastohissien nopeudet kuormattuna ja ilman kuormaa
- kuljettimien nopeudet
- varaston mitat ja layout
- hissien määrä
- hissien ja ohjaimen välinen kommunikaatio
- kuljettimien määrä
- ulossyötön pystyynkääntäjien toiminta
- rullien välit varastossa

Projektin toteuttamiseksi tarvittiin myös voimassa oleva ohjelmistolisenssi Visual Components -ohjelmistoon, sekä tietokone ohjelmiston pyörittämiseen. Tietokoneelta vaaditaan hieman enemmän suorituskykyä, johtuen mallin koosta ja koodissa suoritetusta varastokirjapidon ylläpidosta. Lisäksi mallissa on paljon geometrioita, joiden laskemisesta koneen tulee suoriutua. Toteutetussa mallissa geometria on yksinkertaistettu jättämällä tukirakenteita

mallintamatta, sekä käyttämällä ohjelmiston omia geometrian ja visualisoinnin optimointityökaluja.

## 4.2 Tekniset tavoitteet

Mallille asetettiin useita teknisiä tavoitteita, sillä tavoitteena oli tehdä toiminnallinen malli. Mallin tulisi jäljitellä oikean järjestelmän toimivuutta, jotta siitä saataisiin hyödyllisiä tietoja varsinaista järjestelmää suunniteltaessa. Lisäksi malliin kehitettyjen komponenttien hyödyntäminen tulevilla projekteilla pidettiin mielessä.

- Tiedonsiirto haluttiin toteuttaa toimilaitteiden välillä oikeaa järjestelmää mukaillen.
- Varastoitavien rullien mittojen eroavaisuuksien huomioiminen.
- Varastokirjanpidon ylläpitäminen WMS-ohjainlaitteella.
- Korkeavarastohisseille haluttiin kanavapaikkojen millimetrisijainnit.
- Korkeavarastohissien ja ohjainlaitteen välisen tehtävien jakamisen haluttiin toimivan kuten oikeassa järjestelmässä.
- WMS-ohjainlaitteesta haluttiin tehdä helposti muokattava, jotta sitä voitaisiin käyttää myös tulevilla projekteilla.
- Statistiikan luominen on tärkeää, jotta voidaan seurata mallin suorituskykyä.
- Alkuvaraston luominen heti simulaation käynnistyessä.
- Tilausjärjestelmän kehittäminen on tärkeää, jotta voitaisiin simuloida ulossyöttöä realistisesti.

## 4.3 Dokumentointi

Eri toimilaitteiden sisältämä Python-koodi kommentoidaan koodin luettavuuden helpottamiseksi. Malliin lisätyn statistiikan avulla tarkastellaan rullien ulos- ja sisäänsyötön kapasiteettia, eli kuinka monta rullaa keskimäärin tunnissa syötetään sisään ja ulos. Lopuksi valmiin mallin eri komponentit tallennetaan itsenäisinä komponentteina kirjastoon, josta niitä on helppo hyödyntää tulevilla projekteilla.

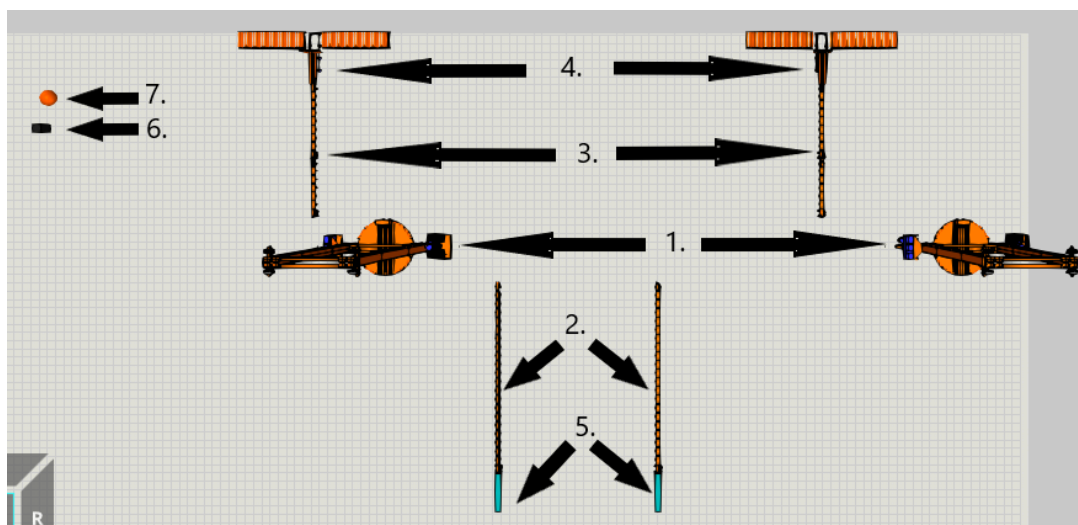
## 5 SIMULOINTIMALLIN KEHITTÄMINEN

Työn taustojen ja tavoitteiden määrittelyn jälkeen päästiin itse asiaan. Työn tekeminen aloitettiin perehtymällä olemassa olevaan suunnitelmaan varastosta ja layoutista. Suunnitelmasta saatiin mitat hyllystölle, kuljettimille, korkeavarastohisseille ja hissien nostokorissa olevalle TransRoll-vaunulle. Lisäksi suunnitelmasta selvisi sisään- ja ulossyöttöjen sekä hissien sijainnit. Mittojen määrittelyn jälkeen tutkittiin yrityksen omat Visual Components -kirjastot ja sieltä saatiin valmiit mallit hisseille, kuljettimille ja pystyynnostajille.

### 5.1 Mallien luonti

Kuvasta 9 selviää simulointimallin toimilaitteiden sijoittelu. Kuvasta on jätetty itse varastohyllystö (kuva 1) pois, jotta toimilaitteet olisi helpompi nähdä. Malli koostuu seuraavista toimilaitteista:

1. Kaksi korkeavarastohissiä
2. Kaksi sisäänsyöttökuljetinta
3. Kaksi ulossyöttökuljetinta
4. Kaksi pystyynnostajaa
5. Kaksi kappalegeneraattoria
6. WMS-ohjainlaite
7. Tilausten tekijä

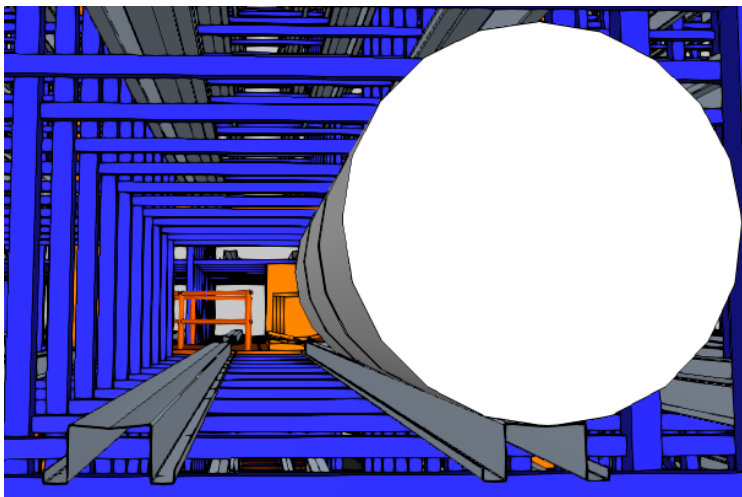


Kuva 9. Mallin osat ilman hyllyä.



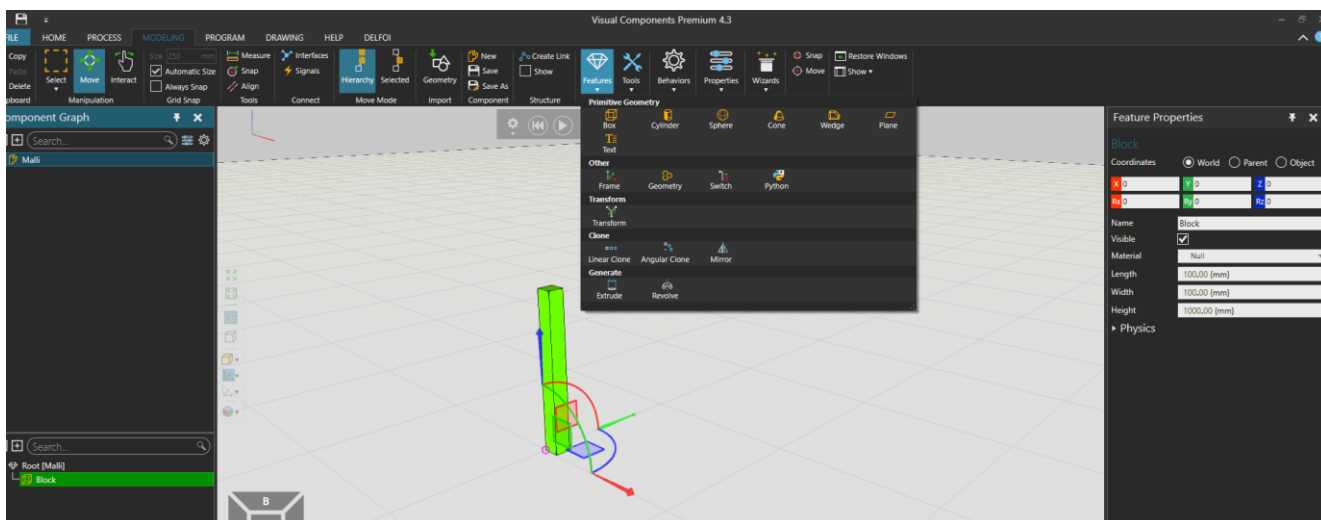
### 5.1.1 Hylly

Hyllystö (kuva 13) mallinnettiin kokonaisuudessaan hyväksi käyttäen Visual Components -ohjelmiston geometriatyökaluja. Ainoastaan TransRoll-kanavaprofiili (kuva 10), jonka päällä rullat hyllyssä lepäävät, tuotiin valmiina kirjastosta. Mallinnus suoritettiin käyttäen suunnitellun varaston mittoja. Kokonaisuudessaan hyllystöissä on 34 kilometriä syväkanavia.



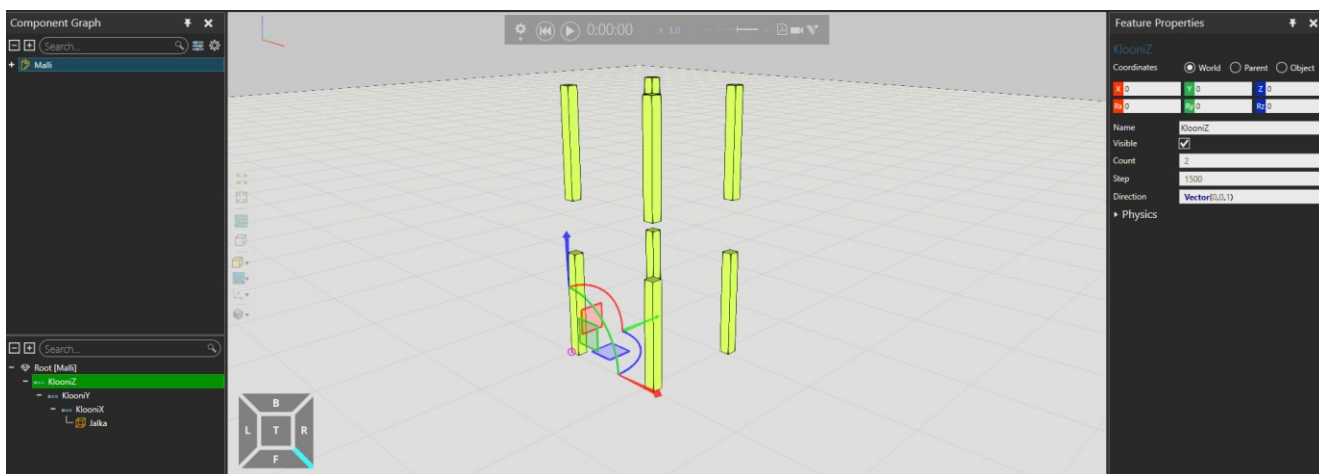
Kuva 10. TransRoll-kanavaprofiili.

Visual Componentsin mallinnustyökalut perustuvat simulointimaailmaan luotaviin geometrioihin, joita muotoillaan sen mukaan, millainen osasta halutaan. Ensin luodaan uusi komponentti, jonka jälkeen Features-valikosta lisätään geometria luodulle komponentille. Osalle annetaan mitat leveys, korkeus ja syvyys (kuva 11).

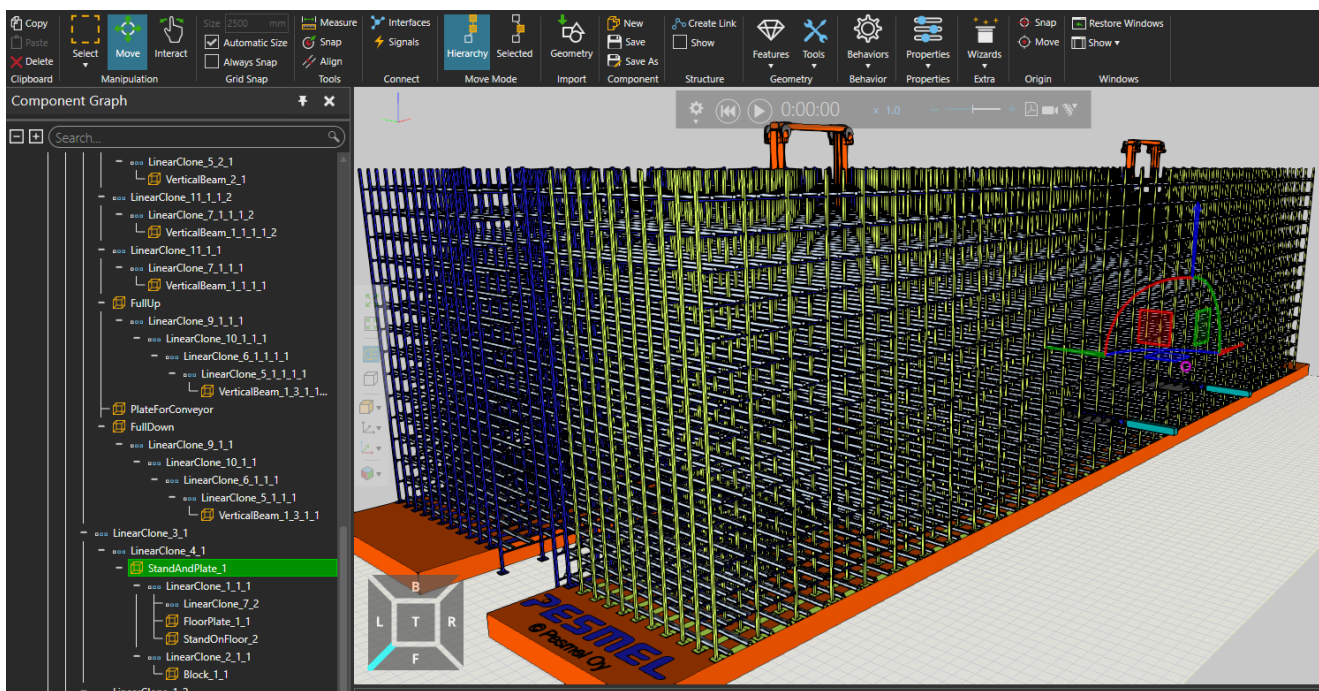


Kuva 11. Komponentin ja geometrian luominen.

Seuraavassa vaiheessa kloonataan kuvassa 11 mallinnettu tolppa lisäämällä valikosta kolme LinearClone-työkalua ja sijoittamalla ne sisäkkäin ja viedään vielä tolpan geometria viimeisen työkalun sisälle. Lopuksi annetaan halutut etäisyydet, suunta ja kopioiden määrä jokaiselle työkalulle. Esimerkissä tolppa on kopioitu jokaisen akselin suuntaan yhden kerran (kuva 12). Koska työkalut ovat sisäkkäin, niin myös työkalun tekemä klooni monistuu. Kyseinen työkalu on käytännöllinen mallinnettaessa suuriakin kokonaisuuksia (kuva 13).



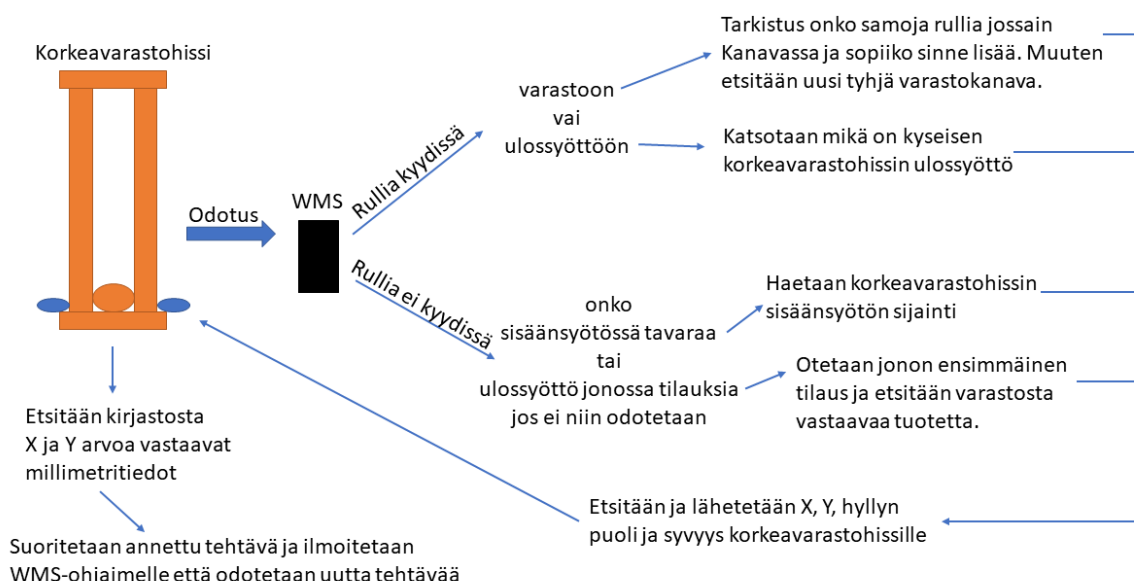
Kuva 12. LinearClone-työkalun käyttö kopiointiin.



Kuva 13. Valmis varastohylly.

### 5.1.2 WMS-ohjainlaite

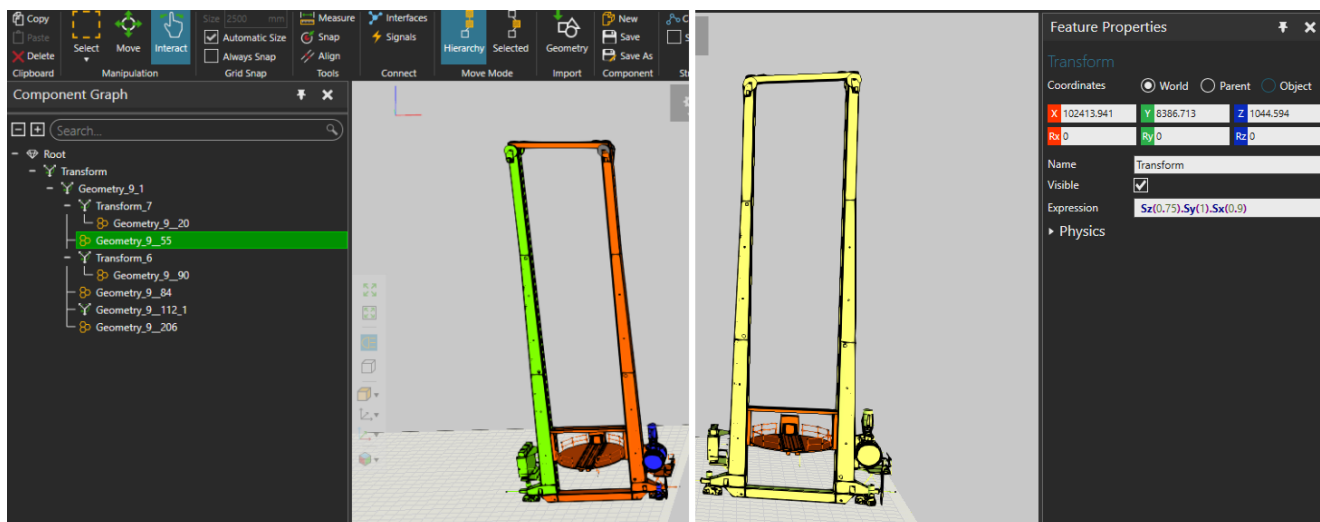
WMS-ohjainlaitteen tehtäväksi määriteltiin varastokirjanpidosta huolehtiminen sekä korkeavarastohissien käskyttäminen (kuvio 1). Ohjainlaitteen rooli on siis hyvin suuri mallin toiminnallisuuden kannalta. Ohjainlaitteen toimintaa käsitellään luvussa 5.3.1. Ensiarvoista oli, että hissit toimisivat yhtäaikaisesti, eli että ne pystyisivät liikkumaan samaan aikaan. Tämän avulla voitiin hissien käyttöastetta parantaa.



Kuvio 1. Korkeavarastohissin tehtävän määrittely.

### 5.1.3 Korkeavarastohissit

Korkeavarastohissien mallit löytyivät yrityksen omasta kirjastosta. Hissien kokoa piti hieman muuttaa, sillä löydetyt mallit olivat liian isot käytävälle. Hissien geometriat jaoteltiin omiin osuuksiinsa ja Transform-työkalulla skaalattiin hissien osat oikean kokoisiksi. Transform-työkalulle annetaan kertoimet, jonka mukaan geometriaa skaalataan, sekä koordinaattiakseli, jonka suuntaan skaalaus tehdään (kuva 14).

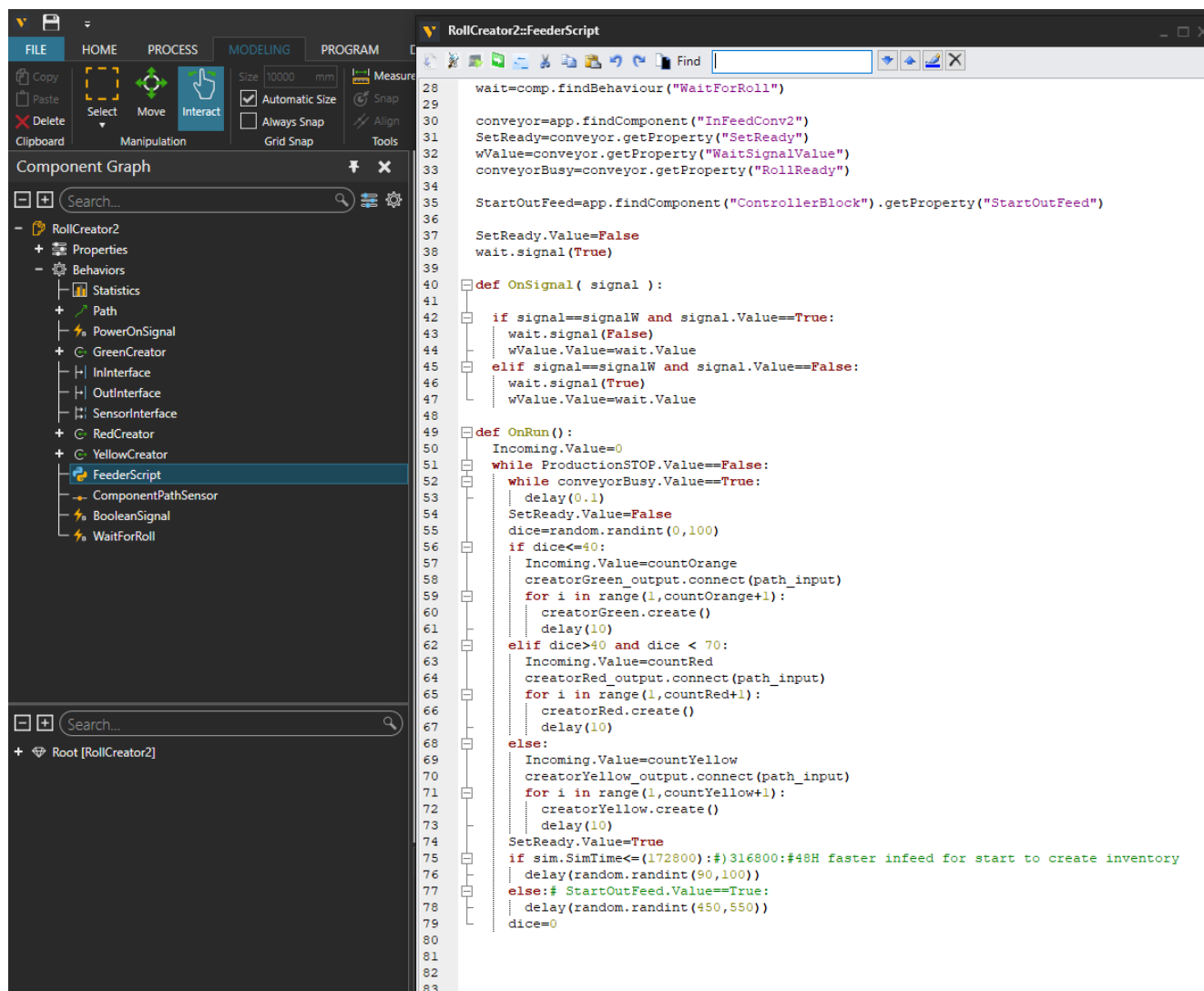


Kuva 14. Hissin geometriat ja skaalaus.

Korkeavastohissin nostokorissa on yksi TransRoll-vaunu per hissi. Hissien nopeudet tyhjänä ja kuormattuna määriteltiin projektinmäärittelyssä. Hissit osaavat väistellä toisiaan boolean signaalin avulla. Esimerkiksi hissien 1 mennessä määritellylle vaara-alueelle, se lähettää hissille 2 signaalin. Hissi 2 odottaa signaalin muuttumista, ja hissien 1 poistuttua vaara-alueelta se lähettää signaalin hissille 2, mikä kertoo hissille 2, että sen on turvallista mennä alueelle. Lisäksi WMS-järjestelmä seuraa hissien sijaintia, ja mikäli hissille ei ole seuraavaa työtä ja edellinen työ oli vaara-alueella, se antaa hissille käskyn liikkua pois vaara-alueelta. Tämä turvajärjestelmä toteutettiin siksi, että toisen hissien ollessa esimerkiksi noutamassa sisään-syötöstä tavaraa on mahdollista, että toinen hissi pääsee törmäämään siihen.

#### 5.1.4 Tuotteiden luonti

Tuotteet luodaan sisään-syötön puolella olevien kahden kappalegeneraattorien sisältämällä koodilla. Nämä generaattorit simuloivat tuotantoa. Projektin määrittelyssä luvussa 4 määriteltiin tuotannosta tulevaisuuden täysiä hissikuormia. Näin ollen tuotteita luodaan kolmesta eri vaihtoehdosta aina täysi hissikuormallinen. Tuotteet ovat rullia, joiden pituus ja halkaisija vaihtelee. Täysi hissikuormallinen on määritelty TransRoll-vaunun kapasiteetin mukaan. Lisäksi sisään-syötön kappalegeneraattori ja kuljetin huolehtivat, että rullien väliin jää toivottu etäisyys, kun rullat ajetaan sisään-syötön kuljettimelle. Itse tuotannon puolta ei lähdetty tarkemmin tässä mallissa simuloimaan, sillä projektinmäärittelystä saatujen tietojen perusteella rullia luodaan tietyllä aikavälillä.



Kuva 15. Kappalegeneraattorin koodi.

Kuvassa 15 esitellään kappalegeneraattorin toiminnallisuudet sekä Python-komentotiedosto. Generaattorilla on kolme ComponentCreator-toiminnallisuutta. Näillä jokaisella on oma kappale, jota tuotetaan. Tuotteita luodaan arpomalla näistä kolmesta toiminnallisuudesta yksi, joka sitten generoi kyseisiä rullia täyden hissikuormallisen. Aluksi rullia generoidaan nopeamman aikataululla, jotta saadaan alkuvarasto luotua. Alkuvaraston luonnin jälkeen rullien generointi tapahtuu 450–550 sekunnin välillä. Mallissa käytettiin aluksi värikoodattuja rullia mallin toiminnan tarkkailun vuoksi. Tämän vuoksi koodissa esiintyy Green-, Yellow- ja RedCreator-toiminnallisuudet.

### 5.1.5 Ulossyötön ominaisuudet ja tilausjärjestelmä

Simulointimalliin kehitettiin tilausjärjestelmä, joka tilaa WMS-ohjainlaitteelta rullia. Projektinmäärittelyssä luvussa 4 määriteltiin ulossyötölle toiminta-ajat ja määrät. Rullia syötetään ulos arkisin 16 tuntia ja lauantaisin noin 14 tuntia. Sunnuntaisin ulossyöttöä ei ole. Python-komentotiedosto tilauksien tekemiseen on sijoitettu erilliseen tilaustentekijälaitteeseen.

Alkuvaraston luomisen jälkeen tilauksia aletaan tehdä WMS-ohjainlaitteelta. Tilausjärjestelmä tilaa aina vuorotellen kaikkia rullakokoja, kuitenkin siten, että suurimmalla rullakoolla on suurin menekki. Hyllystä otettavien rullien määrä arvotaan, ja kun tilaus on valmis, se lähetetään kahdella signaalilla ohjainlaitteelle. Kokonaislukusignaalilla välitetään rullien lukumäärä ja tekstisignaalilla rullien materiaali. Ohjainlaite ottaa tiedon signaaleista vastaan ja tallentaa sen OrderStack-tilauslistaan. Kun korkeavarastohissillä on aikaa, sille annetaan tehtäväksi noutaa varastosta tilauksen mukainen määrä rullia. Mikäli rullat eivät sovi kerralla kyytiin, jätetään toimittamattomat rullat odottamaan OrderStack-listalle.

### 5.1.6 Sisäänsyötön ominaisuudet

Mallissa tuotanto pyörii 24 tuntia vuorokaudessa 7 päivänä viikossa. Alkuvarasto luodaan ajamalla 48 tuntia rullia varastoon, rullia luodaan aina kun kyseinen sisäänsyötön kuljetin on tyhjänä. Tällä varmistetaan tavaran riittävyys varastossa, kun tilauksia alkaa tulemaan. Kun 48 tuntia on kulunut, hidastetaan rullien sisäänsyöttöä vastaamaan tuotantonopeutta. Sisäänsyötöstä tulee täysi hissikuormallinen, sen mukaan mitä rullatyyppiä ollaan tuottamassa ja montako niitä hissien kyytiin sopii.

Mallissa, kuten myös oikeassa järjestelmässä, on sisäänsyöttö etusijalla, eli aina kun tuotannosta tulee tavaraa, korkeavarastohissi käy hakemassa rullat varastoon, jotta tuotanto ei jumiu. Tämä aiheuttaa sen, että tulevat tilaukset pääsevät kertymään. Tulevat tilaukset tallennetaan WMS-ohjainlaitteella listaan, jota molemmat hissit purkavat sitä mukaa kun ehtivät.

## 5.2 Taulukoiden luominen

Mallissa luodaan useita erilaisia taulukoita. Ohjainlaitteella on taulukot varaston sisällölle, sisään- ja ulossyöttöjen sijainneille sekä tilausjonolle. Korkeavarastohisseillä on molemmilla oma kirjastotyyppinen taulukko varastokanavien sijainneista.

### 5.2.1 Ohjainlaitteen taulukot

WMS-ohjainlaitteella luodaan taulukko hyllystä, jonne tallennetaan tiedot kanavapaikassa olevista rullista sekä siitä, kuinka syvällä ne kanavassa ovat (kuva 16). Lisäksi taulukkoon tallennetaan tieto siitä, millaisille rullille kanava on varattuna. Aina kun kanava tyhjenee, se vapautetaan ja sinne voidaan varastoida muuta rullatyyppejä. Taulukosta löytyvät seuraavat tiedot:

- hyllykön puoli
- x-osoite
- z-osoite
- syvyys
- halkaisija
- pituus
- väri
- onko kanavassa tilaa tosi/epätosi
- kanavassa olevien rullien määrä.

Lisäksi luodaan omat listat sisään- ja ulossyöttöjen sijainneille (kuva 16). Myös saapuvat rullatilaukset tallennetaan ohjainlaitteella omaan taulukkoon. Tulevien tilausten taulukossa on, millainen rullan pitää olla ja kuinka monta kappaletta.

```

184 #Create list of Side, X(Length),Z(Height),Y(Depth),Color, Not full / Full and how many in channel
185 #for h in range (0,2):
186 for i in range (0,Columns):
187     for j in range (0,Rows):
188         for h in range (0,2):
189             if h==1 and (i==26 or i==27 or i==36 or i==37) and (j==3 or j==4):
190                 if (i==26 or i ==37) and j==3:
191                     TempList.append(h)
192                     TempList.append(i)
193                     TempList.append(j)
194                     TempList.append(Crane_From_RightSide)
195                     TempList.append(False)
196                     x=list(TempList)
197                     INFEEED.append(x)
198                     del TempList[:]
199             elif h==0 and (i==12 or i==13 or i==48 or i==49) and (j==0 or j==1):
200                 if (i == 12 and j==0) or (i == 48 and j==0):
201                     TempList.append(h)
202                     TempList.append(i)
203                     TempList.append(j)
204                     TempList.append(Crane_From_LeftSide)
205                     TempList.append(False)
206                     x=list(TempList)
207                     OUTFEED.append(x)
208                     del TempList[:]
209             else:
210                 TempList.append(h)#[0] Side
211                 TempList.append(i)#[1] Column X
212                 TempList.append(j)#[2] Row Y
213                 if h==0:
214                     D=BayDepthLeft+Crane_From_LeftSide
215                 else:
216                     D=BayDepthRight+Crane_From_RightSide
217                 TempList.append(D)#[3] Depth Z
218                 TempList.append("Null")#[4] Diameter
219                 TempList.append("Null")#[5] Length
220                 TempList.append("Null")#[6] Colour
221                 TempList.append(True) #[7] If false channel full
222                 TempList.append(0)#[8] How many rolls in channel
223                 x=list(TempList)
224                 RackSXZ.append(x)
225                 del TempList[:]

```

Kuva 16. Ohjainlaitteen hyllytaulukon luonti.

## 5.2.2 Korkeavarastohissin kirjasto

Ennen kuin korkeavarastohissin kirjastoa päästiin luomaan (kuva 18), täytyi selvittää, missä hissi on suhteessa hyllyyn. Tämän selvittämiseksi käytettiin ohjelmistosta löytyvää WorldPositionMatrix-Python-komentoa (kuva 17), jolla komponentin origon sijainti maailmasta löytyy. Näiden sijaintien avulla laskettiin hissin ja hyllykön välisiä etäisyyksiä.

```

ASRS_Position=Rack.WorldPositionMatrix
ASRS_X=ASRS_Position.P.X
ASRS_Z=ASRS_Position.P.Z
ASRS_Y=ASRS_Position.P.Y
ASRS_Crane_Position=comp.WorldPositionMatrix
Crane_X=ASRS_Crane_Position.P.X
Crane_Z=ASRS_Crane_Position.P.Z
Crane_Y=ASRS_Crane_Position.P.Y
Crane_From_Rack=ASRS_Y-Crane_Y

```

Kuva 17. WorldPositionMatrix-komento.

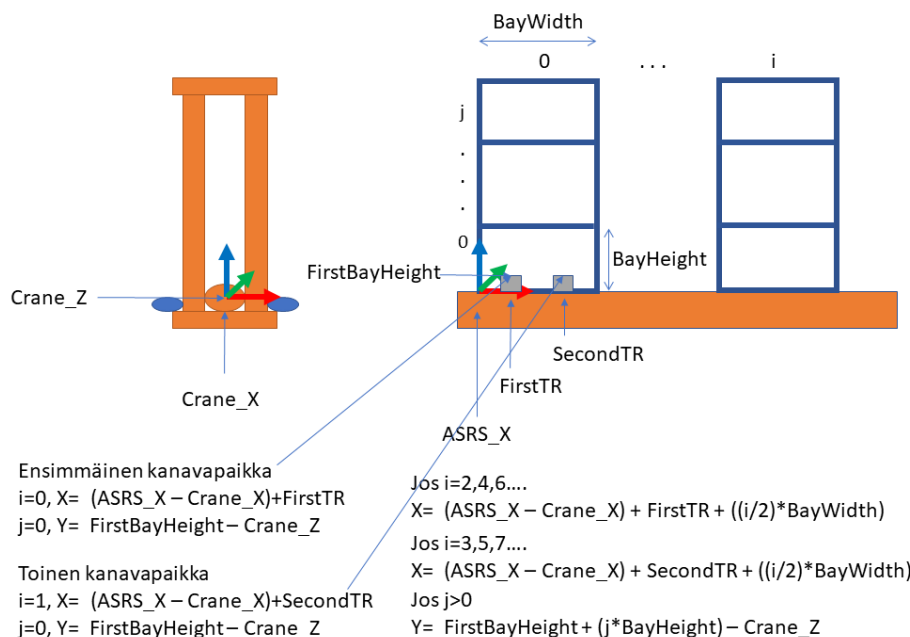


Hissille luotiin kirjastotyyppinen taulukko, josta voidaan hakea x- ja y-koordinaatin perusteella millimetritieto siitä, mihin kohtaan ja kuinka korkealle hissi pitää ajaa, jotta osutaan oikeaan kanavapaikkaan. Näiden millimetritietojen määrittäminen tapahtuu käyttäen hyödyksi hyllykön properties-tiedoista löytyviä kanavien syvyyksiä ja leveyksiä sekä mittaamalla (kuva 19). Korkeavarastohissien origot tulee asettaa samaan kohtaan, jotta sama taulukko käy molemmille. Korkeavarastohissien kirjastoista löytyvät seuraavat tiedot:

- x-osoite millimetreinä
- y-osoite millimetreinä
- sisäänsyöttöjen sijainti
- ulossyöttöjen sijainti.

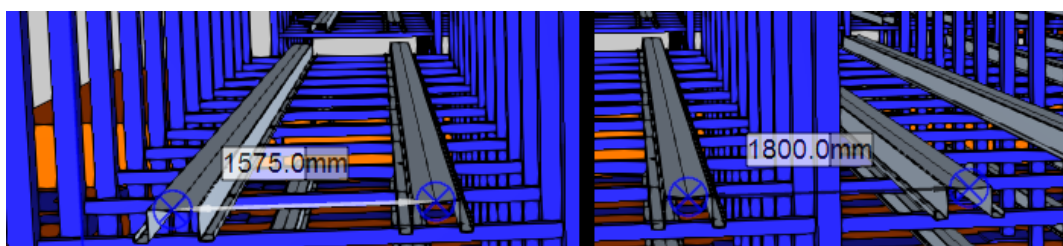
```
#Create dictionary with X,Z key values for mm data of the rack.
#You can access certain places with key values in this case X,Z values coming from controller.
Data={}
for i in range (0,Columns):
  for j in range (0,Rows):
    for k in range (0,2):
      if i==0:
        X=(ASRS_X-Crane_X)+FirstTR
      elif i==1:
        X=(ASRS_X-Crane_X)+SecondTR
      elif i%2 == 0 and i != 0:
        X=(ASRS_X-Crane_X)+FirstTR+((i/2)*BayWidth)
      elif i%2 != 0 and i != 1:
        X=(ASRS_X-Crane_X)+SecondTR+((i/2)*BayWidth)
      if j==0:
        Y=FirstBayHeight-Crane_Z
      elif j>0:
        Y=FirstBayHeight+(j*BayHeight)-Crane_Z
      #InFeeds
      if i ==26 and j==3 and k==1:
        X=59485.773#59442.085
        Y=6344.346
      elif i==37 and j==3 and k==1:
        X=77935.773#77964.812
        Y=6344.346
      #OutFeeds
      if i==12 and j==0 and k==0:
        X=36668.738
        Y=605.951
      elif i==48 and j==0 and k==0:
        X=97398.274
        Y=605.951
      Data[i,j,k]=X,Y
```

Kuva 18. Korkeavarastohissin kirjaston luominen.



Kuvio 2. Korkeavarastohissin kirjaston laskenta.

Haasteita kirjaston luomisessa aiheutti mallin kanavien rakenne, sillä ne eivät ole tasaisella välillä, vaan kaksi kiskoa on samassa kanavassa. Kiskojen etäisyys kanavassa on 1575 mm ja kanavien välillä kiskojen etäisyys on 1800 mm. Tämän vuoksi jouduttiin kehittämään monimutkainen ehtolause (kuva 18), jonka avulla tarkastellaan, mitä kohtaa ollaan kirjastoon lisäämässä (kuvio 2). Tämän jälkeen laskettu millimetritieto tallennetaan hissin kirjastoon ja se on sieltä haettavissa x- ja y-koordinaattien avulla.



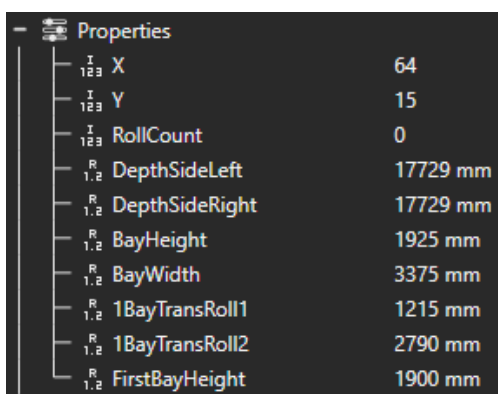
Kuva 19. Varastokiskojen etäisyydet.

### 5.3 Tiedonsiirto

Tiedonsiirto mallin toimilaitteiden välillä toteutetaan hyödyntämällä signaaleja. Signaaleja on käytettävissä useita erilaisia. Työssä tietoa siirretään hyväksikäyttäen kokonaisluku-, reaali-luku- ja boolean-signaaleja. Boolean-signaalien käyttäminen on yksinkertaista, mutta kun

halutaan siirtää esimerkiksi kokonaislukuja signaalin välityksellä, on luotava liitännät molempiin toimilaittekomponentteihin. Tämän liitännän avulla määritellään, millaista ja missä muodossa olevaa tietoa signaalin välityksellä tulee.

Kaikki toimilaitteet on kytketty WMS-ohjainlaitteeseen, jotta se saa tietoa mallin eri osista ja osaa antaa korkeavarastohisseille tehtäviä sen mukaan. Osa tiedoista haetaan suoraan, kuten esimerkiksi hyllykön properties-osiosta (kuva 20). Sieltä saadaan luettua hyllykölle määritettyjä mittoja, kuten esimerkiksi kanavien leveyksiä ja korkeuksia. Lisäksi korkeavarastohisseille on luotu erityinen työvaiheominaisuus, jonka avulla WMS pystyy seuraamaan hissien toimintaa.



Property	Value
X	64
Y	15
RollCount	0
DepthSideLeft	17729 mm
DepthSideRight	17729 mm
BayHeight	1925 mm
BayWidth	3375 mm
1BayTransRoll1	1215 mm
1BayTransRoll2	2790 mm
FirstBayHeight	1900 mm

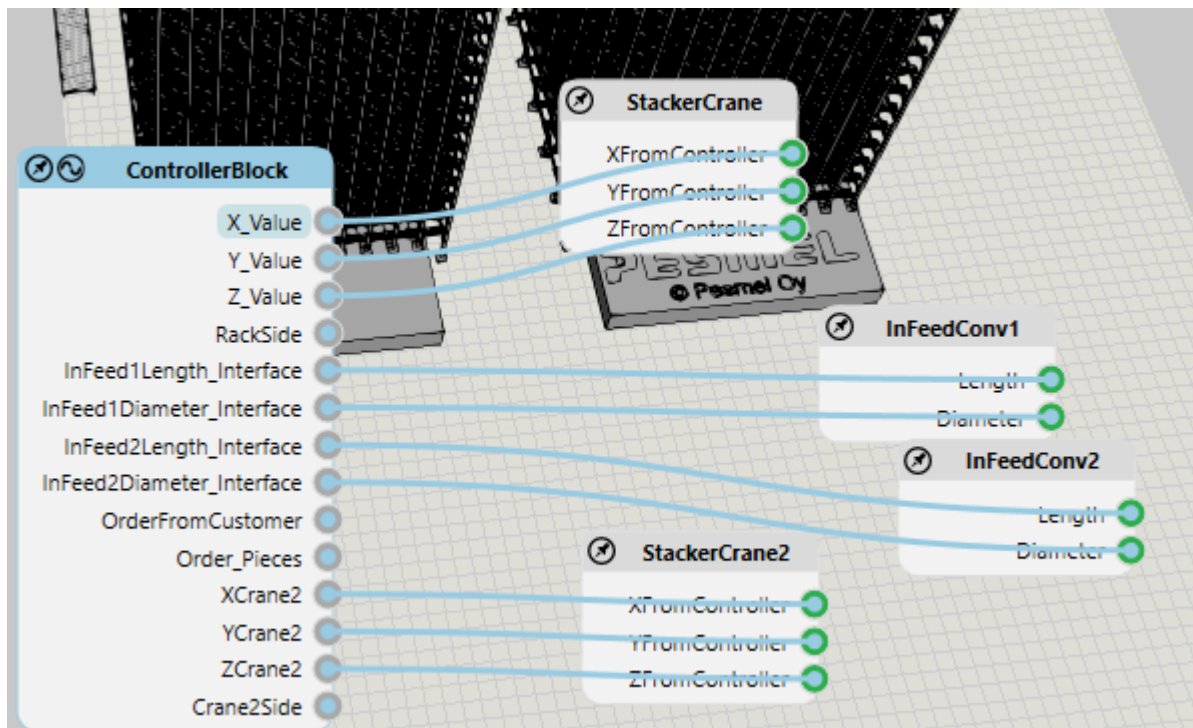
Kuva 20. Hyllystä luettavia ominaisuuksia.

### 5.3.1 WMS ja korkeavarastohissin välinen tiedonsiirto

Sisäänsyötön kuljettimen päässä on anturi, joka ilmoittaa ohjainlaitteelle boolean-signaalin avulla, että siellä on rullia varastoitavaksi. Lisäksi kuljettimelta tulee mittatietoa, millaisia rullia kuljettimella on. Viestin saatuaan WMS-ohjainlaite antaa tehtävän hissille hakea rullat kuljettimelta. Kun hissi on hakenut rullat, se ilmoittaa ohjainlaitteelle suorittaneensa tehtävän, jolloin ohjainlaite etsii rullille varastopaikan, ja antaa hissille tehtäväksi viedä rullat tiettyyn osoitteeseen. Tämä osoite tulee neljällä signaalilla (kuva 21):

- x-kokonaislukuna
- y-kokonaislukuna
- syvyys reaalityyppinä millimetreinä

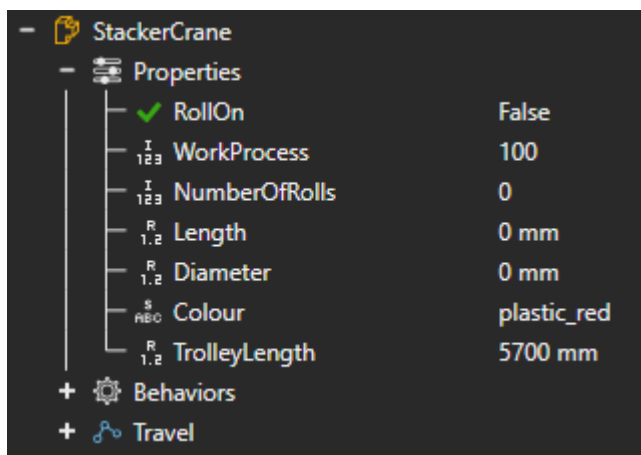
- kumpi hylly kokonaislukuna.



Kuva 21. Tiedonsiirto.

Nämä tiedot saatuaan hissi etsii omasta kirjastostaan x- ja y-kokonaislukua vastaavat reaali-luvut. Nämä reaali-luvut annetaan hissien servoille liikekomennoiksi. Tämän jälkeen korkeavarastohissi liikkuu oikean kanavapaikan kohdalle. Kun oikea kanavapaikka on saavutettu, ajetaan TransRoll-vaunu, jonka kyydissä rullat ovat, kanavapaikkaan sille syvyydelle, jonka WMS-ohjainlaite korkeavarastohissille ilmoitti. Kun rullat on viety kanavaan ja vaunu ajettu takaisin nostokorin kyytiin, ilmoitetaan ohjainlaitteelle, että työ on suoritettu.

WMS-ohjainlaite tarkkailee jatkuvasti korkeavarastohissien työvaiheparametriä, joka kertoo kokonaisluvulla aktiivisen työvaiheen (kuva 22). Työvaiheparametri vaihtelee sen mukaan, mitä työtä hissi on suorittamassa. Työvaiheparametrin arvo menee odottamista kuvaavaan arvoon jokaisen työvaiheen jälkeen. Ohjainlaite huomaa arvon muuttumisen, etsii hissille uuden tehtävän ja tarkastaa hissien tilan, eli onko se kuormattuna vai tyhjänä. Tarkastuksen tehtyään ohjainlaite osaa antaa hissille seuraavan tehtävän.



Kuva 22. Korkeavarastohissin ominaisuudet.

### 5.3.2 Rullatiedon tallentaminen WMS-ohjainlaitteelle

Korkeavarastohissi noutaa sisäänsyötöstä rullat ja ilmoittaa ohjainlaitteelle, että kyydissä on rullia varastoitavaksi. Tämän jälkeen ohjainlaite suorittaa haun taulukostaan, onko kyseiselle rullatyypille varattuna kanavapaikkaa. Mikäli sellaista kanavaa ei löydy, jossa on tilaa, arvotaan listasta uusi tyhjä kanavapaikka. Tämän jälkeen ohjainlaite ottaa kopion kyseisestä kanavapaikasta ja päivittää kanavan uudeksi syvyydeksi alkuperäisen syvyyden, josta on vähennetty rullien vaatiman tilan mitta. Lisäksi päivitetään muut luvussa 5.2.1 mainitut listan kohdat. Kanavapaikan tiedot kopioidaan, jotta tietojen päivittäminen luotettavasti onnistuu. Mikäli tietoa päivittää suoraan taulukkoon, laskenta ei onnistu, sillä vanha tieto ylikirjoittuu heti. Ohjainlaite siis tietää, millä syvyydellä ensimmäinen kanavassa oleva rulla on. Ulosyöttö toimii samoin, mutta käänteisesti, eli kun rullia noudetaan varastosta, kanavan syvyyttä lisätään.

## 5.4 Statistiikka

Malliin kehitettiin статистиikkaa seuraamaan eri toimilaitteiden toimintaa (kuva 23). Korkeavarastohisseille tehtiin kuvaajat seuraamaan käyttöastetta. Varastosta seurataan sinne vietyjen ja sieltä otettujen rullien määrää. Tilauksista seurataan tilauksien kertymistä tilausjonoon.



Kuva 23. Malliin toteutettu statistiikka.

Nämä kaikki on helppo toteuttaa Statistics-elementin avulla. Statistiikan avulla mallin toiminnan tarkastelu on helpompaa, kun saadaan visualisoitua tietoa eri osista ja niiden toiminnasta. Tämä Statistics-elementti on hyvin yksinkertainen työkalu, jonka avulla on helppo mitata haluttuja ominaisuuksia (kuva 24). Tässä työssä korkeavarastohisseillä hyödynnettiin idle- ja busy-komentoja, joiden avulla statistiikka osaa laskea prosentuaaliset arvot näiden tilojen kestolle suhteutettuna simulointiin kuluneeseen aikaan. Aina kun hissi suorittaa tehtävää, se muuttuu arvokseen busy. Kun tehtävä on suoritettu, tila muuttuu idle-arvoon.

Properties			
CraneStats			
Name	CraneStats	IntervalBusyPercentage	0
PartsEntered	15167	IntervalBlockedPercentage	0
PartsExited	15167	IntervalFailedPercentage	0
PartsCurrent	0	IntervalRepairPercentage	0
PartsUtilization	0	IntervalSetupPercentage	0
PartsAverageCount	1.256	StateWarmupPercentage	0
PartsMinCount	0	StateWarmupIntervalPerce...	0
PartsMaxCount	6	StateBreakPercentage	0
PartsTotalTime	765246.956	StateBreakIntervalPercentage	0
PartsAverageTime	50.455	StateSetupPercentage	0
PartsMinTime	23.662	StateSetupIntervalPercenta...	0
PartsMaxTime	147.322	StateIdlePercentage	29.833
PartsIntervalEntered	0	StateIdleIntervalPercentage	100
PartsIntervalExited	0	StateBusyPercentage	70.167
PartsIntervalCurrent	0	StateBusyIntervalPercentage	0
PartsIntervalUtilization	0	StateBlockedPercentage	0
PartsIntervalAverageCount	0	StateBlockedIntervalPercen...	0
PartsIntervalMinCount	0	StateBrokenPercentage	0
PartsIntervalMaxCount	0	StateBrokenIntervalPercent...	0
PartsIntervalTotalTime	0	StateRepairPercentage	0
PartsIntervalAverageTime	0	StateRepairIntervalPercenta...	0
PartsIntervalMinTime	0		
PartsIntervalMaxTime	0		
Utilization	70.167		
TotalStateTime	609420		
BreakPercentage	0		

State Name	System State
Warmup	WarmUp
Break	Break
Setup	WarmUp
Idle	Idle
Busy	Busy
Blocked	Blocked
Broken	Fail
Repair	Repair
Click To Add Row	

Create default states

Kuva 24. Statistikan ominaisuudet.

## 6 TULOKSET JA JATKOKEHITYS

Työn alussa selvitettiin simulointimalliin vaadittavia alkutietoja sekä käytettävissä olevia resursseja, joiden avulla projekti määriteltiin. Työn aikana saatiin luotua projektinmäärittelyn mukainen malli varastosta. Malli saatiin toivottujen ominaisuuksien osalta toimivaksi ja työn aikana kohdattuihin ongelmiin löydettiin ratkaisut. Varsinaiseen kysymykseen sisään- ja ulosyöttöjen kapasiteetin sekä korkeavarastohissien määrän riittävydestä saatiin vastaus. Simulointimallilla suoritettujen ajojen tulokset vastasivat projektin määrittelyssä määriteltyjä kapasiteettien arvoja. Eli varaston sisään- ja ulosyöttö suunnitelluilla rullaa per tunti -arvoilla toteutuu kahdella korkeavarastohissillä.

Varsinaisen työn lisäksi tavoitteena oli kehittää simulointimallia luvussa 4.2 mainittujen teknisten tavoitteiden osalta. Nämä lisätavoitteet laadittiin siksi, että mallista saataisiin skaalautuvampi ja sen osien hyödyntäminen tulevaisuudessa uusissa projekteissa olisi helpompaa. Etenkin WMS-ohjainlaitteen sekä korkeavarastohissin osalta tämä kehitys onnistui. Malli tehtiin aluksi vain yhdellä hissillä ja toinen lisättiin työn edetessä. Koodia muokattiin siten, että tiettyihin muuttujiin luetaan sen hissien ominaisuudet, jota halutaan käskyttää. Tällöin koodia ei tarvitse monistaa, vaan muuttujat vaihdetaan vastaamaan kyseistä hissiä. Hissin koodiin ei tarvinnut tehdä muutoksia, sillä origojen asettaminen samaan kohtaan toisen hissinkin kanssa teki samojen millimetritaulukoiden hyödyntämisen mahdolliseksi.

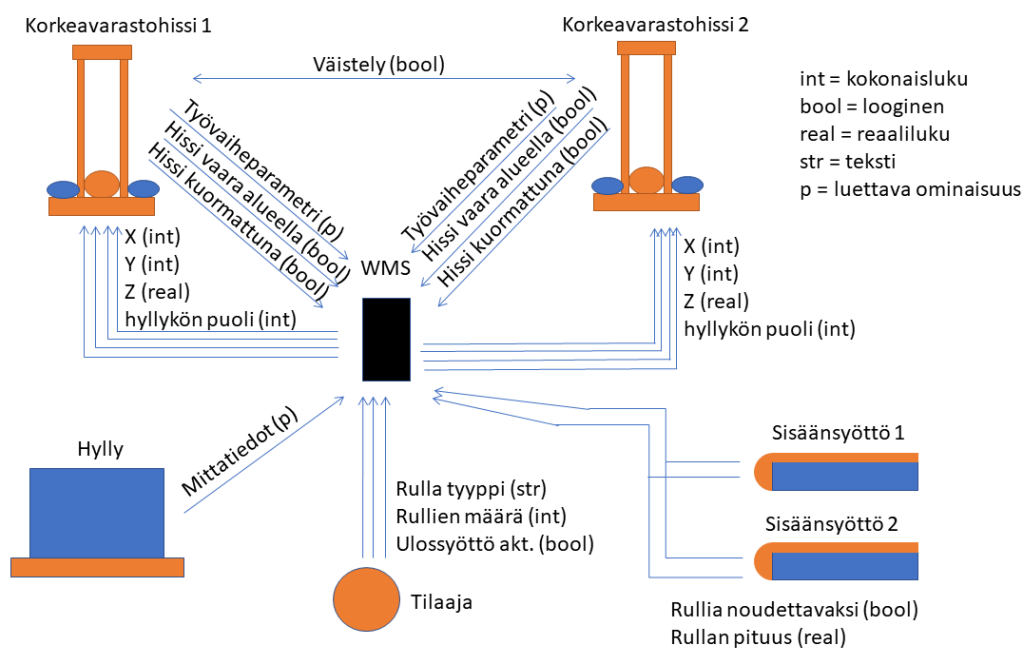
Hyllykkö saatiin mallinnettua käyttäen ohjelmiston omia geometriatyökaluja. Luvussa 5.1.1 esitellyt työkalut osoittautuivat tehokkaiksi ja helppokäyttöisiksi. Hyllykön ensimmäiseen versioon kului miltei viikko aikaa, mutta viimeisin versio valmistui puolessa päivässä. Tästä voidaan päätellä tekijän taitojen kehittyneen. Korkeavarastohissien skaalauksen toteuttaminen ohjelmiston työkaluilla onnistui myös hyvin ja hisseistä saatiin oikean kokoiset.

Kirjastojen ja taulukoiden luomisessa luvussa 5.2 oli haasteita, etenkin mittatietojen tallentamisessa korkeavarastohissin kirjastoon. Kirjasto ja taulukko eroavat toisistaan siten, että kirjastossa on mahdollista antaa tietylle paikalle tunnisteet, joiden avulla paikassa olevaan tietoon päästään käsiksi. Taulukko taas on pitkä lista, jonka sijaintiin päästään käsiksi vain tietämällä oikea kohta listasta. Luvussa 5.3.2 esitelty taulukoiden muokkaaminen vaatii aina listan kohdan kopioimisen, jotta sitä voidaan turvallisesti muokata. Näiden asioiden selvittäminen vaati aikaa, sillä taulukon päivittyminen ennenaikaisesti vaikeutti tiedon hallintaa ja aiheutti mallin toiminnallisuuteen epätoivottuja tapahtumia, kuten rullien ajamista päällekkäin.



Kirjaston käyttäminen oli aivan uutta, mutta selvittelyn jälkeen se havaittiin erittäin toimivaksi etenkin korkeavarastohissien millimetritietojen tallennukseen.

Luvussa 5.3 esitelty tiedonsiirto mallin laitteiden välillä saatiin toimivaksi. Kuviossa 3 esitellään mallin eri osien välillä liikkuvaa tietoa, sekä kerrotaan missä muodossa ja miten tieto siirretään. Kuviossa, lyhenteet int, bool, real ja str, ovat signaaleja, mutta luettava ominaisuus tarkoittaa toimilaitteella olevaa ominaisuutta, jonka WMS-ohjainlaite lukee suoraan.



Kuvio 3. Signaalit laitteiden välillä.

Mallissa alkuvarastoa ei erikseen luoda suoraan hyllyyn, vaan rullia tuotetaan komponenttgeneraattoreilla ja niitä ajetaan korkeavarastohisseillä varastoon. Mallin käytettävyyden kannalta tämä pitäisi saada kehitettyä sellaiseksi, että luetaan alkuvarasto esimerkiksi Excel-taulukosta tai luodaan satunnaisesti rullia varastoon heti kun simulaatio käynnistetään. Jatkokehitykseen kuuluu myös rullien tunnistamiseksi varastossa käytetyn värimääritelmän vaihtaminen rullalle annettavaan id-numeroon, josta selviää esimerkiksi väri, leveys, korkeus ja pituus. Luvussa 5.1.5 esitellyn tilausjärjestelmän tilalle on kehitteillä uusi id-pohjainen tilausjärjestelmä ja WMS-ohjainlaitteen päivittäminen seuraavaan versioon on siis seuraavana työvaiheena.

## 7 POHDINTA

Simulointimallin kehittäminen aloitettiin työharjoittelujaksolla ja harjoittelun päätteeksi siitä saatiin opinnäytetyölle aihe. Mallinnuksesta teki haasteellista tekijän vajavaiset taidot käytetyn ohjelmointikielen sekä ohjelmiston osalta. Kuten Shannon (1998, s.8) toteaa, simulointimallinnus vaatii koulutusta, taitoa ja käytettävän ohjelmiston hallitsemista, ja työharjoittelu on hyvä paikka kehittää itseään. Mielestäni toteuttamani simulointimalli on hyvä esimerkki simulointiprojektista ja sen etenemisestä määrittelystä toteutukseen ja tuloksiin.

Itse mallille asetetut simulointitavoitteet saavutettiin ja sen avulla voitiin todeta tehtävänannon mukaisten kapasiteettien riittävyys. Shannonin (1998, s. 9) mukaan simuloinnin tarkoituksena on päätöksen tekijän ongelman ratkaisun helpottaminen eli tässä tapauksessa varastokuljettimien ja korkeavarastohissien määrän päättäminen. Varastokirjanpidon tekeminen ja sen jatkuva päivittäminen aiheuttivat päänvaivaa, mutta ratkaisut ongelmiin löytyivät yrityksen sisältä, oppilaitoksesta tai ohjelmiston hyvin aktiiviselta keskustelupalstalta. Lisäksi projektin alussa tulleet muutokset suunnitelmiin kuluttivat aikaa, sillä toteutettu malli eroaa huomattavasti ensimmäisestä suunnitelmasta. Vaikka onkin suositeltavaa ottaa simulaatio osaksi suunnitteluprosessia heti alkumetreillä (Pesmel Oy, 2018, s. 3), niin projektin määrittelyn tulee olla valmis ainakin layout-piirrosten osalta. Tällä vältytään turhan työn tekemiseltä.

Lisätavoitteena oli mallin osien kehittäminen siten, että ne olisivat hyödynnettävissä helposti myös tulevissa projekteissa, etenkin WMS-ohjainlaitteen kehitykseen, jonka tärkein tehtävä on kontrolloida materiaalivirtoja eri prosessien välillä (Pesmel, 2020, s. 3). Mielestäni tämä on järkevä tavoite ajatellen projektin vaatimaa työmäärää. Tämä on kuitenkin hankala toteuttaa yhden projektin aikana, sillä lopputulokseen vaikuttaa moni asia. Olen kuitenkin hyvin luottavainen siihen, että ohjainlaite tulee jatkojalostumaan paremmaksi tekijän taitojen kehittyessä. Tarvitaan myös erilaisia projekteja, jotta päästään muokkaamaan ja kehittämään ohjainta sekä löydetään ne kriittiset muuttujat, jotka vaikuttavat ohjainlaitteen toimintaan. Hyödyllistä olisi, jos simulointimallin WMS-ohjainlaitteen kehitykseen osallistuisi myös oikean järjestelmän kehittäjä, jotta saataisiin kehitettyä mallia lähemmäs todellisen järjestelmän toimintaa. Myös liitynnät tehokkaampiin relaatiotietokantoihin, sekä niiden hyödyntäminen, voisi tuoda mallin toiminnallisuuteen lisää käytettävyyttä.

## LÄHTEET

- Banks, J., Carson II, J. S., Nelson, B. L., & Nicol, D. M. (2010). *Discrete-Event System Simulation* (5. painos). Pearson Education, Inc.
- Heiliö, M. (2010). Mallit ja sovellukset. Teoksessa S. Pohjolainen (toim.), *Matemaattinen mallinnus* (s. 17–39). Sanoma Pro Oy.
- Kruis, A. & Tew, R. (2021). *Stackless-Python*. <https://stackless.readthedocs.io/en/3.8-slp/stackless-python.html>
- Pesmel Oy. (2018). *PESMEL MATERIAL FLOW HOW: Simulation and Data Analysis*. [https://pesmel.com/wp-content/uploads/2020/05/Pesmel\\_General\\_Simulation\\_Brochure\\_lowres-1.pdf](https://pesmel.com/wp-content/uploads/2020/05/Pesmel_General_Simulation_Brochure_lowres-1.pdf)
- Pesmel Oy. (2020). *PESMEL MATERIAL FLOW HOW: Handling Systems for Pulp and Paper*. [https://pesmel.com/wp-content/uploads/2020/12/Paper\\_brochure\\_rev4.pdf](https://pesmel.com/wp-content/uploads/2020/12/Paper_brochure_rev4.pdf)
- Pesmel Oy. (2022). *PESMEL: About us*. <https://pesmel.com/about-us/>
- Shannon, E. R. (1998). *Proceedings of the 1998 Winter Simulation Conference: Introduction to the art and science of simulation*. <https://www.informs-sim.org/wsc98papers/001.PDF>
- Visual Components Oy. (2015). *Visual Components Academy: Basics of Python Scripting*. <https://academy.visualcomponents.com/lessons/component-scripting/?course=854>
- Visual Components Oy. (28.2.2017). *Visual Components Academy: Component Scripting*. <https://academy.visualcomponents.com/lessons/component-scripting/?course=854>
- Visual Components Oy. (2022a). *Visual Components: About us*. <https://www.visualcomponents.com/about-us/>
- Visual Components Oy. (2022b). *Visual Components: Products*. <https://www.visualcomponents.com/products/>
- Visual Components Oy. (2022c). *Visual Components Academy: Basics of Python Scripting*. <https://academy.visualcomponents.com/courses/basics-of-python-scripting/>