



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Sami Koivisto

TUOTANTOKONSEPTIN SIMULOINTI

Teollisuusrobotiikka YAMK
2024

TIIVISTELMÄ

| | |
|--------------------|------------------------------|
| Tekijä | Sami Koivisto |
| Opinnäytetyön nimi | Tuotantokonseptin simulointi |
| Vuosi | 2024 |
| Kieli | suomi |
| Sivumäärä | 75 |
| Ohjaaja | Mika Billing |

Tämän opinnäytetyön tarkoituksena on tutkia valitun tuotantokonseptin kyvykkyyttä teollisessa valmistuksessa simuloinnin avulla. Työssä mallinnetaan pääasiassa kokoonpanoa sisältävä tuotantolinja yksityiskohtaisesti käyttäen Siemens Plant Simulation -ohjelmistoa.

Työssä käydään läpi tehdassimuloinnin historiaa ja millaisia hyviä ja huonoja puolia sillä on käytettävänä työkaluna. Työ koostuu suurimmaksi osaksi päiväkirjasta, jossa käydään päiväkohtaisesti läpi simulaatiomallin rakentamista sekä käsitellään siinä vastaan tulleita oleellisia asioita ja haasteita.

Lopputuloksena saatiin hyvin todellisuutta vastaava malli tuotantolinjasta, jolla tehtiin erilaisia simulaatioajoja. Näistä ajoista kerättiin dataa, jota analysoitiin ja pohdittiin sen perusteella miten optimoida kyseistä tuotantoa. Lisäksi pohdittiin, miten kannattaisi jatkaa mallin kanssa sekä arvioitiin simuloinnin mielekkyyttä kyseisessä tehtävässä.

ABSTRACT

| | |
|--------------------|------------------------------------|
| Author | Sami Koivisto |
| Title | Simulation of a Production Concept |
| Year | 2024 |
| Language | Finnish |
| Pages | 75 |
| Name of Supervisor | Mika Billing |

The purpose of this thesis is to investigate the capability of a chosen production concept in industrial manufacturing through simulation. The thesis primarily focuses on modeling an assembly line in detail using the Siemens Plant Simulation software.

The thesis reviews the history of factory simulation, exploring its advantages and disadvantages as a tool. It mainly consists of a diary that chronologically documents the construction of the simulation model, addressing significant issues and challenges encountered.

The result is a realistic model of the production line, used for various simulation runs. The data gathered from these runs was analyzed to determine production optimization strategies. Additionally, the thesis discusses future development of the model and evaluates the relevance of simulation for this type of task.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

| | | |
|-------|---|----|
| 1 | JOHDANTO..... | 9 |
| 1.1 | Danfoss A/S..... | 9 |
| 1.2 | Danfoss Drives Oy Vaasa..... | 10 |
| 1.3 | Taajuusmuuttaja | 10 |
| 2 | SIMULAATIO TEOLLISESSA VALMISTUKSESSA | 11 |
| 2.1 | Historiaa | 11 |
| 2.2 | Sytä simulointimallien käytölle | 13 |
| 2.3 | Simulaatioiden haasteet | 14 |
| 2.4 | Simuloinnin mielekkyys..... | 15 |
| 2.5 | Simuloinnin tulevaisuus | 16 |
| 3 | SIMULAATION TUOTTEET | 18 |
| 3.1 | Yleiskuvaus..... | 18 |
| 3.2 | Avainkomponentit ja simulaation tuotekirjo..... | 19 |
| 3.3 | Tuotantoprosessin kuvaus | 20 |
| 4 | SIMULAATIOMALLIN TEKEMINEN | 21 |
| 4.1 | Tecnomatix Plant Simulation | 21 |
| 4.2 | Koulutus Danfossilla..... | 21 |
| 4.3 | Päiväkirja | 22 |
| 4.4 | Valmistelevat harjoitukset | 23 |
| 4.4.1 | 21.2. – 22.2.2024..... | 23 |
| 4.4.2 | 23.2.2024..... | 24 |
| 4.4.3 | 26.2. – 27.2.2024..... | 25 |
| 4.4.4 | 13.3.2024..... | 26 |
| 4.5 | Viikko 1..... | 27 |
| 4.5.1 | 3.4.2024..... | 27 |
| 4.5.2 | 4.4.2024..... | 29 |
| 4.5.3 | 5.4.2024..... | 29 |

| | | |
|--------|----------------|----|
| 4.6 | Viikko 2..... | 31 |
| 4.6.1 | 8.4.2024..... | 31 |
| 4.6.2 | 9.4.2024..... | 32 |
| 4.6.3 | 10.4.2024..... | 33 |
| 4.6.4 | 11.4.2024..... | 34 |
| 4.6.5 | 12.4.2024..... | 35 |
| 4.7 | Viikko 3..... | 36 |
| 4.7.1 | 15.4.2024..... | 36 |
| 4.7.2 | 16.4.2024..... | 37 |
| 4.7.3 | 17.4.2024..... | 38 |
| 4.7.4 | 18.4.2024..... | 38 |
| 4.7.5 | 19.4.2024..... | 39 |
| 4.8 | Viikko 4..... | 40 |
| 4.8.1 | 22.4.2024..... | 40 |
| 4.8.2 | 23.4.2024..... | 41 |
| 4.8.3 | 24.4.2024..... | 43 |
| 4.8.4 | 25.4.2024..... | 44 |
| 4.8.5 | 26.4.2024..... | 44 |
| 4.9 | Viikko 5..... | 46 |
| 4.9.1 | 29.4.2024..... | 46 |
| 4.9.2 | 30.4.2024..... | 46 |
| 4.9.3 | 2.5.2024..... | 47 |
| 4.9.4 | 3.5.2024..... | 48 |
| 4.10 | Viikko 6..... | 49 |
| 4.10.1 | 6.5.2024..... | 49 |
| 4.10.2 | 7.5.2024..... | 49 |
| 4.10.3 | 8.5.2024..... | 51 |
| 4.10.4 | 9.5.2024..... | 52 |
| 4.11 | Viikko 7..... | 52 |
| 4.11.1 | 13.5.2024..... | 52 |

| | | |
|--------|----------------------------------|----|
| 4.11.2 | 14.5.2024..... | 53 |
| 4.11.3 | 15.5.2024..... | 54 |
| 4.11.4 | 16.5.2024..... | 55 |
| 4.11.5 | 17.5.2024..... | 56 |
| 4.12 | Viikko 8..... | 57 |
| 4.12.1 | 20.5.2024..... | 57 |
| 4.12.2 | 21.5.2024..... | 57 |
| 4.12.3 | 22.5.2024..... | 58 |
| 4.12.4 | 23.5.2024..... | 58 |
| 4.12.5 | 24.5.2024..... | 59 |
| 4.13 | Viikko 9..... | 60 |
| 4.13.1 | 28.5.2024..... | 60 |
| 4.13.2 | 29.5.2024..... | 61 |
| 4.13.3 | 30.5.2024..... | 62 |
| 4.14 | Viikko 10..... | 63 |
| 4.14.1 | 3.6.2024..... | 63 |
| 4.14.2 | 4.6.2024..... | 64 |
| 4.14.3 | 5.6.2024..... | 64 |
| 4.14.4 | 6.6.2024..... | 65 |
| 4.14.5 | 7.6.2024..... | 66 |
| 5 | TULOSTEN ANALYSOINTI | 68 |
| 5.1 | Simulaatioajojen muuttujat | 68 |
| 5.2 | Tulokset..... | 69 |
| 5.3 | Johtopäätökset | 72 |
| 5.4 | Jatkosuunnitelmat | 73 |
| 6 | TYÖN ARVIOINTI | 74 |
| | LÄHTEET | 75 |

KUVA- JA TAULUKKOLUETTELO

| | |
|---|--------|
| Kuva 1. Yleiskuva iC7-kaapitustuotteesta..... | 18 |
| Kuva 2. Tuotannon prosessikaavio | 20 |
| Kuva 3. Ensimmäinen simulaatiomalli yksinkertaiselle kokoonpanolle | 23 |
| Kuva 4. Harjoitusmalli usealla kokoonpanopisteellä ja operaattorilla | 24 |
| Kuva 5. Harjoitusmalli alikokoonpanolla ja usealla operaattorityypillä | 25 |
| Kuva 6. Harjoitusmalli oikeilla kaappilinjan grafiikoilla | 26 |
| Kuva 7. Simulaatiomalli 3.4.2024 | 28 |
| Kuva 8. Simulaatiomalli 5.4.2024 | 30 |
| Kuva 9. Simulointimalli 8.4.2024 | 31 |
| Kuva 10. Simulointimalli 9.4.2024 | 33 |
| Kuva 11. Simulointimalli 10.4.2024 | 34 |
| Kuva 12. Simulointimalli 11.4.2024 | 35 |
| Kuva 13. Simulointimalli 15.4.2024 | 37 |
| Kuva 14. Malliin lisätty operaattorien tehtävälista 17.4.2024 | 38 |
| Kuva 15. Simulointimalli 18.4.2024 | 39 |
| Kuva 16. Simulointimalli 19.4.2024 | 40 |
| Kuva 17. Simulointimalli 22.4.2024 | 41 |
| Kuva 18. Esimerkki tilauskirjataulukosta | 43 |
| Kuva 19. Erilliset kaappipuskurit ennen tailoringia | 45 |
| Kuva 20. Esimerkkejä vaiheajanäytöistä..... | 47 |
| Kuva 21. Tarkemmin mallinnettu systeemimoduulituotanto | 50 |
| Kuva 22. Uudet store-puskurit tailoring-ympäristössä..... | 53 |
| Kuva 23. Tehdas-layoutin tuonti malliin | 54 |
| Kuva 24. Kaappirunkojen purkupiste ja puskurit..... | 55 |
| Kuva 25. Kaikki puskurit store-tyyppiä | 56 |
| Kuva 26. Esimerkki satunnaisesta tilauskannasta | 61 |
| Kuva 27. Esimerkki multi-level experimental designin syöttötaulukosta | 64 |
| Taulukko 1. Operaattorien roolitus vs. tuotos, 5 operaattoria | 70 |

KÄYTETYT LYHENTEET

| | |
|--------|---|
| ACD | Assembly plate for Control Devices |
| AFE | Active Front End |
| CC | Control Compartment (ACD+DiD) |
| DiD | Door-in-Door |
| iC7 ED | iC7 Enclosed Drives |
| | <i>iC7 kaapitus</i> |
| I/O | Input/Output |
| INU | INverter Unit |
| IU | Integration Unit |
| LCL | LCL suodatin AFE:lle, koostuu LC:stä (yläosa) ja L:stä (alaosa) |
| MIS | Main Input Section |

1 JOHDANTO

Tämän opinnäytetyön tarkoituksena oli rakentaa simulointimalli valitusta iC7 ED -tuotantokonseptista, jonka avulla voidaan arvioida sen kyvykkyyttä erilaisissa tuotantotilanteissa. Aihe valittiin, koska simulointiin haluttiin tutustua yleisesti ja kuinka hyödyllistä sen tekeminen olisi yrityksen toiminnassa myös jatkossa. Lisäksi haluttiin hankkia kokemusta simulointimallien rakentamisesta.

Simulointia haluttaisiin käyttää apuvälineenä suunnitteluvaiheessa, kun puntaroidaan erilaisia vaihtoehtoja sekä arvioimaan uusien investointien tuomaa hyötyä verrattuna kustannuksiin.

Opinnäytetyö koostuu simulaation teoriaosuudesta, päiväkirjamaisena tehdystä simulointimallin rakentamisosiosta ja tulosten analysointiosiosta. Lopuksi pohditaan myös jatkokehitysajatuksia ja miten jatkaa opinnäytetyön jälkeen aiheen parissa.

1.1 Danfoss A/S

Danfoss on teknologiateollisuuden yritys, jonka Mads Clausen on perustanut vuonna 1933 nimellä Dansk Køleautomatik- og Apparatfabrik. Yhtiö sai nykyisen nimensä vuonna 1946. Yrityksellä on yli 41 000 työntekijää globaalisti yli 100 tehtaassa ympäri maailmaa. /1/

Yrityksen toiminnot jakautuvat kolmeen päähaaraan, jotka ovat Power Solutions, Climate Solutions ja Power Electronics and Drives. Power Solutionsin tuotteita ovat pääasiassa liikkuvan ja teollisen hydrauliiikan ratkaisut, joita käytetään enimmäkseen laivojen ja työkoneiden voimansiirtoon. Climate Solutions keskittyy energiatehokkaisiin teollisiin lämmitys- ja jäähdytysratkaisuihin, kuten ilmastoinnin jäähdytykseen, kylmäteollisuuteen ja kaukoenergiasovelluksiin.

Power Electronics and Drivesin tuotteita ovat pääasiassa taajuusmuuttajat, joita käytetään sähkömoottorien muuttuvaan nopeudenohjaukseen sekä sähköistykseen auto- ja muissa teollisuuksissa. /1/

1.2 Danfoss Drives Oy Vaasa

Vaasan tehdas oli aiemmin VACON, joka liittyi Danfossiin yritysosaston yhteydessä vuonna 2014. Samalla Vaconin taajuusmuuttajatuotteet tulivat osaksi Danfossin tuoteperhettä. Vaasan tehdas kuuluu Power Electronics and Drives -osioon. Vaasan toimipiste on Suomen suurin, siellä on 627 työntekijää (tammikuu 2023). /2/

Tehdas tuottaa matalajännitteisiä ilma- ja nestejäähdytteisiä taajuusmuuttajia 6 MW asti, sekä irrallisina moduuleina että kaapitettuina. /2/

1.3 Taajuusmuuttaja

Taajuusmuuttaja on moottorinohjain, joka ohjaa sähkömoottorin nopeutta muuttamalla sen tehonsyötön taajuutta ja jännitettä portaattomasti ja tarkasti. Nykyisissä taajuusmuuttajissa on lisäksi verkko- ja diagnostiikkaominaisuuksia, joilla suorituskkyä voidaan hallita paremmin ja joilla tuottavuutta voidaan parantaa. /3/

Useimpia kohteita taajuusmuuttajille ovat erilaiset puhaltimet, pumput ja kompressorit, jotka kattavat noin 75 % kaikista käytössä olevista taajuusmuuttajista. Taajuusmuuttajia voidaan käyttää myös tehonmuunnokseen esimerkiksi akkuvarastointisovelluksissa. /3/

2 SIMULAATIO TEOLLISESSA VALMISTUKSESSA

Simuloinnilla tarkoitetaan todellisen prosessin, järjestelmän, tehtaan tms. toiminnan jäljittelyä. Siihen kuuluu järjestelmän keinotekoisen historian luominen ja tämän historian tarkkailu, jotta voitaisiin tehdä päätelmiä edustetun todellisen järjestelmän ominaisuuksista. Simulointi on ongelmanratkaisumenetelmä monien todellisten ongelmien ratkaisemiseksi. Simulointia käytetään kuvaamaan ja analysoimaan järjestelmän käyttäytymistä, esittämään 'mitä jos' -kysymyksiä järjestelmästä ja auttamaan todellisten järjestelmien suunnittelussa. Sekä olemassa olevia että käsitteellisiä järjestelmiä voidaan mallintaa simulaatiolla. /4/

2.1 Historiaa

Tuotannon ja valmistuksen simuloinnin juuret ovat toisen maailmansodan ajassa 1940-luvulla. Sodanaikaiset operaatiot loivat suuria logistisia ja strategisia haasteita, joiden ratkaisemiseksi alettiin käyttää tieteellisiä ja matemaattisia keinoja näiden ongelmien optimoimiseksi. Tähän aikaan matemaatikot John Von Neumann ja Stanislaw Ulam tutkivat neutronien käyttäytymistä vetypommien yhteydessä. He keksivät keinon käyttää tietokoneita apuna uudella tavalla todennäköisyys- ja tilastomatematiikan avulla, ja heidän saavuttamansa suuri menestys yleistä näiden metodien käyttöä myös muissa ongelmissa. /5/

Sodanjälkeinen teollisuus etsi myös keinoja parantaa valmistuksen tehokkuutta, kun sen tarpeet kasvoivat sekä määrällisesti että lisääntyvän monimutkaisuuden muodossa. Kuluttajat halusivat enemmän erilaisia tuotteita, keksittiin uusia koneita ja laitteita ja hankintaketjut muuttuivat globaaleimmaksi. Tämä kaikki loi haasteita tuotannon optimointiin, lisääntyvien virhemahdollisuuksien hallintaan, sekä tarpeen sille, että osattiin tehdä parempia tietoon perustuvia päätöksiä, mikä osaltaan edesauttoi simulaatiomallien syntyä. /6/

Ensimmäiset simulaatiot 1950-luvulla olivat hyvin rajoittuneita ja perustuivat yleensä fyysisiin malleihin sekä käsin tehtyihin laskutoimituksiin, vaikka alkeellisia

tietokoneitakin oli jo olemassa. Tietokoneilla saatiin laskettua monimutkaisempia asioita, mutta ne olivat harvinaisiana kalliita ja siten harvemmin saatavilla. Tämän ajan simulointi kesti tavallisesti liian kauan, tarvitsi liian monta erikoisosajaa ja tulokset olivat yleensä epämääräisiä, eli oli toisin sanottuna liian kallista saavutettuun hyötyyn nähden. /5/

1960-luvulla simulaatioita ajettiin yleensä niin, että tietokoneeseen syötettiin lähtötiedot ja jätettiin se suorittamaan laskutoimituksiaan, jotka sitten kerättiin myöhemmin, sen sijaan että oltaisiin voitu tehdä näitä asioita reaaliajassa. Tietokoneet olivat edelleen valtavia ns. suurtietokoneita. Tähän aikaan alkoi kuitenkin jo ilmestyä erilaisia ohjelmointikieliä simulointia varten, kuten GPSS, SIMSCRIPT, CSL ja SIMULA. Lisäksi alkoi pikkuhiljaa muodostua yhteisöjä simuloinnin ympärille jakamaan tietoa ja vertailemaan tuloksia. /5/

1970-luvulla tietokoneet kehittyivät, laskentatehoa saatiin lisää ja ohjelmointikieliä jalostettiin, mikä mahdollisti myös simulointimallien kehittymisen. Simulointi oli kuitenkin edelleen suurtietokoneiden tehtävää eli se oli edelleen kallista ja harvoin saatavilla. Tänä aikakautena alettiin nähdä ensimmäisiä graafisia käyttöliittymiä. /5/

1980- ja 1990-luvuilla henkilökohtaisten tietokoneiden (PC) yleistyminen mahdollisti simulointimallien rakentamisen myös muille kuin isoille organisaatioille ja yrityksille, joilla oli suurtietokoneita käytössään. Tätä ajanjaksoa voidaankin pitää eräänlaisena käännekohtana simuloinnin historiassa. Ohjelmistot jatkoivat kehittymistään, ja käyttöliittymät muuttivat graafisiksi ja helppokäyttöisemmiksi, jolloin malleja oli helpompi ja nopeampi laatia. Vaikka PC:iden laskentateho olikin tänä aikana aika rajallinen verrattuna suurtietokoneisiin, saatiin niillä kuitenkin tehtyä yksinkertaisia malleja. /5/

Simuloinnin ”modernina” aikana voidaan pitää aikaa 1990-luvun lopusta tähän päivään. Laskentatehon kasvu on jatkunut eksponentiaalisena, ohjelmistot ovat kehittyneet edelleen, joten yhä monimutkaisemmat ja hienostuneemmat

sovellukset ovat isomman yleisön saatavilla. Internetin yleistymisen on tuonut tietoa aiheesta helpommin saataville ja helpottanut yhteistyötä mallien parissa. /5/

2.2 Syitä simulointimallien käytölle

Simulaatiomallien käytössä voidaan saavuttaa monia hyötyjä.

Mallien avulla voidaan testata uusia asioita virtuaalisessa maailmassa todellisen sijaan ensin. Esimerkiksi voidaan suunnitella uusi tuotantolinja ennen investointeja ja saada hyvä käsitys sen suorituskyvystä jo etukäteen, jolloin voidaan punnita tarvittavien investointien mielekkyyttä.

Mallien avulla voidaan tunnistaa, mihin tuotannon pullonkaulat muodostuvat ja millä keinoilla niitä voitaisiin korjata. /7/

Voidaan tehdä malli olemassaolevasta tuotannosta sen optimoimiseksi. Tällaisella mallilla voidaan esimerkiksi kokeilla muutoksia ilman että oikean tuotantolinjan toimintaa häiritään. /7/

Voidaan kokeilla erilaisia skenaarioita tuotannossa. Esimerkiksi mikä vaikutus olisi, jos henkilöstöä lisätään/vähennetään, jokin kone menee pitkäaikaiseen huoltoon, investoidaan uusi laite, jne. Hyvin rakennetun mallin avulla saadaan vaikutuksista hyvä käsitys etukäteen. /7/

Voidaan kokeilla, millainen vaikutus erilaisilla tuotekirjoilla voisi olla tuotantoon. Mitä jos tehdään monia tuotteita jonkin tuotantolaitoksen kaikilla linjoilla, tai vaihtoehtoisesti vain tiettyjä tuotteita tietyissä paikoissa? Entä jos tehdään tiettyjen tilausten samanlaiset tuotteet aina peräjälkeen sen sijaan että tehtäisiin kaikkia sekaisin? Tällaisiin kysymyksiin saataisiin vastauksia hyvän simulaatiomallin avulla. /7/

Mallin avulla voidaan saada hyvä käsitys tuotannon läpimeno- ja toimitus-ajoista erilaisissa tilanteissa, mikä auttaisi tuotannon suunnittelussa ja ennustamisessa.

/7/

Simulaatiomalli voi helpottaa haluttujen muutosten esittämisessä ja perustelemissa ylemmälle johdolle siten, että on näyttää dataa siitä, millainen vaikutus muutoksilla voisi olla. /7/

2.3 Simulaatioiden haasteet

Simulaatiossa tulee kuitenkin huomioida myös seuraavat haasteet.

Mallin suunnittelu, rakentaminen, prosessin määrittäminen, muuttujien asettelu, ohjelmointi ja muut toimenpiteet, joita simulaatiomalliin tarvitaan vievät paljon aikaa. /8/

Simulaatio-ohjelmistot ovat monimutkaisia ja vaativia käyttäjille. Tästä johtuen tarvitaan yleensä myös merkittävästi koulutusta henkilöstölle ja osaavia resursseja sekä kokemusta hyvien mallien tekemiseen. Tiedon siirto malleista voi olla myös haastavaa käyttäjien välillä, sillä voi olla vaikeaa ymmärtää toisen tekemiä monimutkaisia malleja. Lisäksi ohjelmistolisenssit ovat yleensä kalliita. /8/

Malli ei voi olla parempi kuin lähtötiedot ja data, jonka perusteella se tehdään. Hyvä simulaatiomalli vaatii pohjaksi hyvin tehdyt perustukset, jotta se imitoisi todellisuutta mahdollimman hyvin, esimerkiksi erilaisten vaiheajojen määrittämisessä. Jos malli perustuu väärin lähtötietoihin, ei voida olettaa että sen antamat tiedot olisivat yhtään sen parempia. /8/

Kaikkia tosielämän monimutkaisia toimintoja ei joko kannata tai edes voi aina simuloida tarkasti, johtuen joko siitä että se olisi erittäin vaikeaa ja aikaavievää, tai sitten että ohjelmistot eivät vain taivu siihen. Ylipäätään simuloinnissa tehdään usein yksinkertaistuksia todellisista toiminnoista, mikä voi johtaa siihen, etteivät ne enää vastaa todellisuutta tarpeeksi hyvin. /8/

Mallin antamat tulokset voivat olla vaikeita tulkita. Voi olla vaikeaa sanoa, miksi saatiin sellaisia tuloksia kuin saatiin ja mitkä ovat avaintekijöitä kyseisessä prosessissa. Voi olla myös vaikeaa tulkita, ovatko tulokset oikeita vai onko mallissa esimerkiksi jokin osa-alue, joka vääristää tuloksia. Tästä johtuen jokainen valmis malli tulisi myös verifioida todellisuuden avulla; jos lähtödata on molemmissa sama, niiden antamat tulokset korreloivat ja ovat oikeansuuntaisia. Todennäköisesti mallia täytyy iteroida ja hienosäätää muutamia kertoja, ennenkuin päästään tarpeeksi lähelle todellista tilannetta. Epätarkan simulaation perusteella voidaan tehdä vääriä johtopäätöksiä ja sitä kautta vääriä päätöksiä. /8/

2.4 Simuloinnin mielekkyys

Simulointi ei ole aina automaattisesti paras ratkaisu kaikkiin tapauksiin, mutta seuraavat asiat ovat sellaisia, joiden vuoksi se voi olla hyvä vaihtoehto:

Jos valmistusprosessi tai -järjestelmä on monimutkainen, niin muilla keinoilla voi olla vaikeaa saada hyviä arvioita sellaisen toiminnasta ja suorituskyvystä. Tällaista monimutkaisuutta voi olla esimerkiksi laaja tuotekirjo, jota valmistetaan satunnaisessa järjestyksessä, henkilöstöä hoitamassa useita eri työtehtäviä siten, ettei jokainen työvaihe ole välttämättä aina miehitettynä, useita työpisteitä tai linjoja, jotka ruokkivat esimerkiksi yhteistä seuraavaa työvaihetta, jne.

Jos ollaan suunnittelemassa uusia prosesseja tai ollaan tekemässä isoja investointeja, on asioiden testaaminen etukäteen simuloinnilla houkutteleva vaihtoehto, sillä voidaan saavuttaa suuria rahallisia säästöjä.

Jos edellä mainitut asiat ovat yrityksessä tavallisia asioita, ehkä silloin kannattaisi panostaa simulaation osaamiseen ja ohjelmistoihin, jotta siitä tulisi normaali toimintatapa. Tällöin mallien laatu ja sitä kautta niiden antamat tulokset paranisivat, eikä niiden tekemiseen menisi niin kauan aikaa.

Milloin simulointi ei välttämättä ole hyvä vaihtoehto:

Jos prosessit ovat yksinkertaisia, jotka pystytään ratkaisemaan peruslaskutoimituksilla, niin ei ole syytä rakentaa simulointimallia.

Jos ei ole riittävästi hyvää dataa saatavilla prosessin toiminnasta, niin ei voida olettaa että simulointimallia osaisi jotenkin täyttää nämä aukot ja antaa silti oikeita tuloksia.

Jos tarvitaan nopeita päätöksiä, niin on parempi käyttää joitain muita keinoja. Hyvän simulointimallin rakentaminen vaatii aina aikaa.

Jos yrityksessä ei ole usein tarpeita simuloinnille, eikä valmiiksi osaamista, niin ohjelmistoihin ja koulutuksiin panostaminen ei välttämättä maksa itseään koskaan takaisin.

2.5 Simuloinnin tulevaisuus

Tietokoneiden laskentateho jatkaa kasvuaan ja simulaatio-ohjelmistot kehittymistään, mikä yhdessä mahdollistaa monimutkaisempien ja tarkempien simulaatiomallien tekemisen jatkossa lyhyemmässä ajassa. Kun simulaation hyvät puolet kasvavat samalla kun huonot puolet pienenevät, johtanee se siihen että simulaation käyttäminen jatkaa yleistymistään.

Tekoälyllä tulee varmasti olemaan roolinsa myös simulaation yhteydessä tulevaisuudessa, mutta tällä hetkellä on ehkä vaikea nähdä, missä sen parhaat käyttömahdollisuudet voisivat olla. Ehkä jossain vaiheessa tuotantolinjojen suunnittelussa voidaan antaa tekoälylle vain lähtötietoja tuotteesta ja prosessista, ja se tekee tämän datan perusteella erilaisia vaihtoehtoja, miten sen voisi valmistaa. Vaikka tuskin mikään näistä vastauksista olisi täsmälleen oikea, siinä voitaisiin säästää paljon mallinnus- ja määrittäysaikaa manuaalisena työnä. /9/

Etäsuunnittelu ja pilvipalvelut tarjoavat jatkossa vaihtoehtoja siihen, että simulointityötä tehtäisiin perinteiseen tyyliin omalla tietokoneella. Viimeisen vuosikymmenen aikana on nähty merkittävää kasvua verkkosovelluksissa. Käytetystä teknologiasta riippumatta etäsuunnittelijat tarjoavat tavan keskittää dataa ja kokemusta samalla, kun IT-kustannukset pienenevät. Samalla voidaan saada enemmän laskentatehoa ja sitä kautta sujuvuutta työhön. Resursseista tulee joustavampia ja yhteistyömahdollisuudet laajenevat. /9/

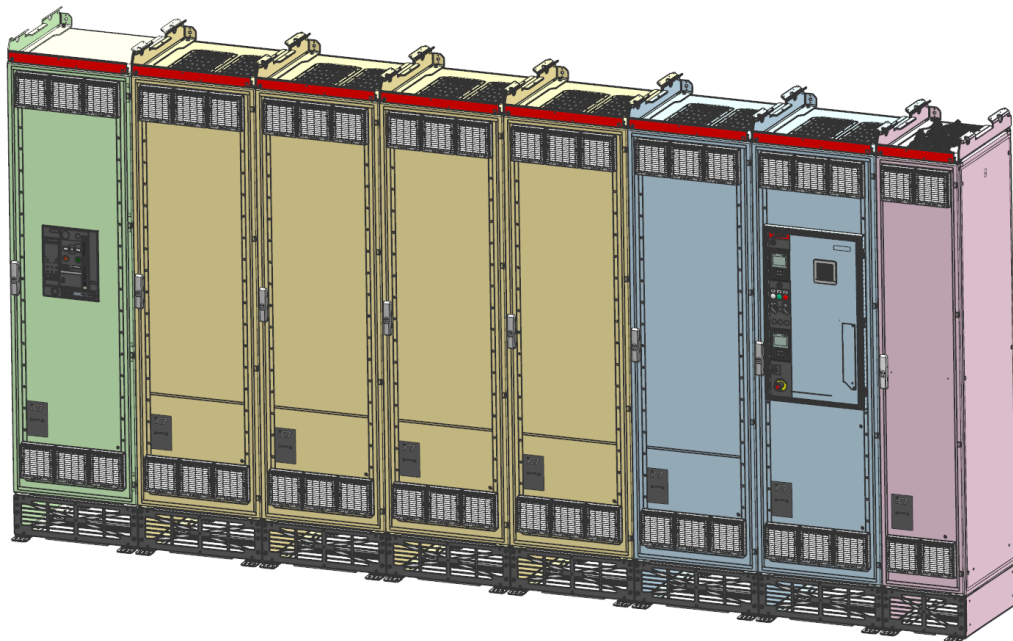
3 SIMULAATION TUOTTEET

3.1 Yleiskuvaus

Tässä simulaatiomallissa käytetyt tuotteet ovat Danfossin iC7-tuoteperheeseen kuuluvia kaapitustuotteita. Nämä tuotteet perustuvat vahvasti modulaarisuuteen.

Tällaisia kenttätyppejä ovat esimerkiksi sisääntulokenttä (MIS), syöttökenttä (AFE), invertterikenttä (INU) ja sisään-/ulostulokenttä (I/O). Yksinkertaistettuna tuote toimii siten, että sisääntulokenttään tulee syöttö vaihtovirtana, AFE:t muuttavat sen tasasähköksi, jota INU moduloi ja muuttaa takaisin vaihtosähköksi.

Kuvassa 1 mainitut kenttätypit ovat eri väreillä vasemmalta oikealle, MIS-AFE-
INU-I/O:



Kuva 1. Yleiskuva iC7-kaapitustuotteesta

3.2 Avainkomponentit ja simulaation tuotekirjo

Simulaatiossa on tarkasteltu erikseen kaappeihin sisääntulevat avainkomponentit ja niistä koostuvat alikokoonpanot, jotka valmistetaan yksilöinä kutakin kaappitilausta vasten. Nämä avainkomponentit ovat:

- Kaapitustuotanto
 - Sisääntulokentät
 - MIS400
 - MIS600
 - Modulikentät
 - LCL10+AFE10+INU10+CC (modulit tyyppiä 10)
 - LCL11+AFE11+INU11+CC (modulit tyyppiä 11)
 - LCL10+AFE10
 - LCL11+AFE11
 - 2xINU10+CC
 - 2xINU11+CC
 - I/O-kentät
 - I/O400
 - Ohjauskokonaisuus (Control Compartment, CC)
 - ACD ja DiD
- Systemimoduulituotanto
 - AFE-systemimoduuli
 - AFE-moduulit ja niiden ilmakehavaosiot (IU)
 - AFE:n ohjausyksikkö ja mahdollinen tähtikytkentäkortti
 - INU-systemimoduuli
 - INU-moduulit ja niiden liitäntäosiot (IU)
 - INU:n ohjausyksikkö ja mahdollinen tähtikytkentäkortti
 - LCL-suodatinkokonaisuus
 - LC-moduuli
 - L-moduuli

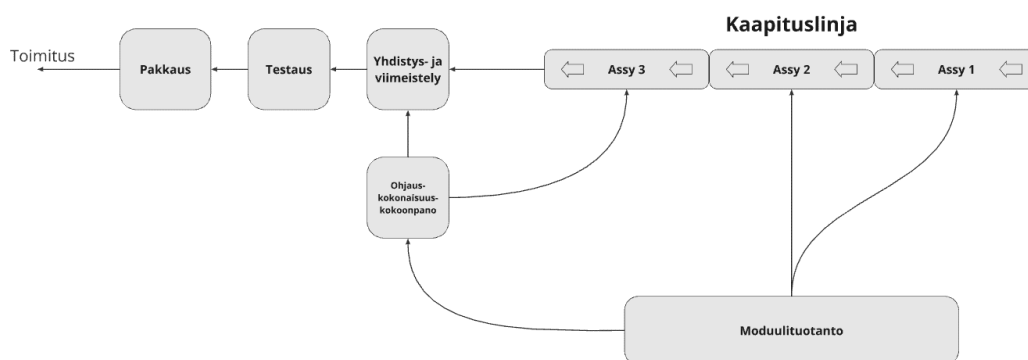
Mallissa simuloitiin seuraavien kaapitustuotteiden tuotantoa:

- A10C10 1xLCL + 1xAFE + 1xINU, modulit tyyppiä 10
- A10C10+IO 1xLCL + 1xAFE + 1xINU + I/O, modulit tyyppiä 10
- A11C11 MIS + 1xLCL + 1xAFE + 1xINU, modulit tyyppiä 11
- A11C11+IO MIS + 1xLCL + 1xAFE + 1xINU + I/O, modulit tyyppiä 11
- 2A102C10 MIS + 2xLCL + 2xAFE + 2xINU, modulit tyyppiä 10
- 2A102C10+IO MIS + 2xLCL + 2xAFE + 2xINU + I/O, modulit tyyppiä 10
- 2A112C11 MIS + 2xLCL + 2xAFE + 2xINU, modulit tyyppiä 11
- 2A112C11+IO MIS + 2xLCL + 2xAFE + 2xINU + I/O, modulit tyyppiä 11

3.3 Tuotantoprosessin kuvaus

Kuten luvussa 3.2 mainittiin, kaikki kaapitustuotteen avainkomponentit valmistetaan yksilöinä päätuotteen tilaukselle. Tämä johtuu siitä, että niistä on niin paljon variaatioita, ettei ole mielekästä yrittää tehdä niitä valmiiksi varastoon.

Kaapitustuotteiden tuotanto jakautuu kuvan 2 mukaisesti moduulituotantoon, kaapitustuotantolinjaan, ohjauskokonaisuuden kokoonpanopisteeseen, kaappikenttien yhdistys- ja viimeistelyalueeseen, testaukseen ja pakkauspisteeseen.



Kuva 2. Tuotannon prosessikaavio

4 SIMULAATIOMALLIN TEKEMINEN

4.1 Tecnomatix Plant Simulation

Simulaatiomalli rakennettiin Siemens Tecnomatix Plant Simulation -ohjelmistolla. Käytetty versio oli 2201, build 2361. Tämä ohjelmisto valittiin sen vuoksi, että se on vakiintunut työkalu Danfossilla, ja sitä on käytetty aikaisemmin monien mallien rakentamiseen. Sen käytöstä löytyy yrityksessä paljon asiantuntemusta simulaatioon keskittyneestä globaalista tiimistä.

Tecnomatix Plant Simulation on ohjelmisto, jolla voidaan rakentaa hyvin jäsenneltyjä 3D-simulointimalleja tuotantotoiminnoista, -linjoista tai vaikka kokonaisista tuotantolaitoksista. Malleja voidaan tehdä erilaisista tuotantoprosesseista käyttäen objektipohjaista arkkitehtuuria ja mallinnusominaisuuksia. /10/

Ohjelmistossa käytetään Siemensin omaa SimTalk-ohjelmointikieltä, jolla voidaan itse ohjelmoida malliin toimintoja ja logiikkaa, joita perusobjekteissa ei ole valmiina, tai jos halutaan että jokin sisäänrakennettu toiminto toimii erilailla.

4.2 Koulutus Danfossilla

Ennen työn aloitusta Danfossilla järjestettiin globaali 12 session koulutusohjelma Plant Simulation -ohjelmiston käyttöön. Näitä tapahtumia oli keskimäärin kerran viikossa ajanjaksolla helmikuu – toukokuu 2024.

Koulutuksessa käytiin läpi perusteita seuraavista asioista:

- Simulaatio-objektit ja niiden käyttäytyminen
- Syöttödata, tilastollinen analyysi ja hyvät käytännöt näihin liittyen
- Operaattorien mallinnus
- SimTalk-ohjelmointikielen esittely
- SimTalk-koodin debuggaus

- SimTalk-koodin käyttö KPI:den keräämiseen ja datan analysointiin
- AGV:iden ja AMR:ien mallinnus
- Koottujen simulaatioajojen ajo ja optimointi
- Simulaatiotulosten esittely ei-asiantuntijoille

Koulutus oli erittäin hyödyllinen mallin rakentamisen alkuun pääsemisessä, koska Plant Simulation ei ole kirjoittajan mielestä mikään erityisen helppotajuinen tai intuitiivinen ohjelmisto. Vaikka koulutus ei tarjonnutkaan vastauksia kaikkiin haasteisiin, joita matkan varrella tuli vastaan, nopeutti se työn etenemistä alkuvaiheessa silti huomattavasti. Koulutus sattui myös ajallisesti oikein hyvään kohtaan siten, että tätä työtä voitiin tehdä samanaikaisesti koulutusjakson ollessa käynnissä.

4.3 Päiväkirja

Seuraavissa luvuissa kuvataan työn etenemistä päivittäisellä tasolla. Ennen varsinaista työn aiheen mallin rakentamista tehtiin useita kokeiluja yksinkertaisemmilla kokonaisuuksilla, jotta saatiin parempi käsitys siitä, miten ohjelmaa voidaan käyttää erilaisten toimintojen aikaansaamiseksi ja minkälaisia haasteita siinä tulee vastaan.

Näitä valmistelevia toimia, harjoituksia ja kokeiluja ohjelmistolla tehtiin 21.2.2024 – 27.2.2024. Varsinainen työn aiheen malli rakennettiin 3.4.2024 – 7.6.2024.

Lukujen lopussa on aina esitetty kuva mallista, jos siinä on jokin muuttunut edelliseen kuvaan verrattuna.

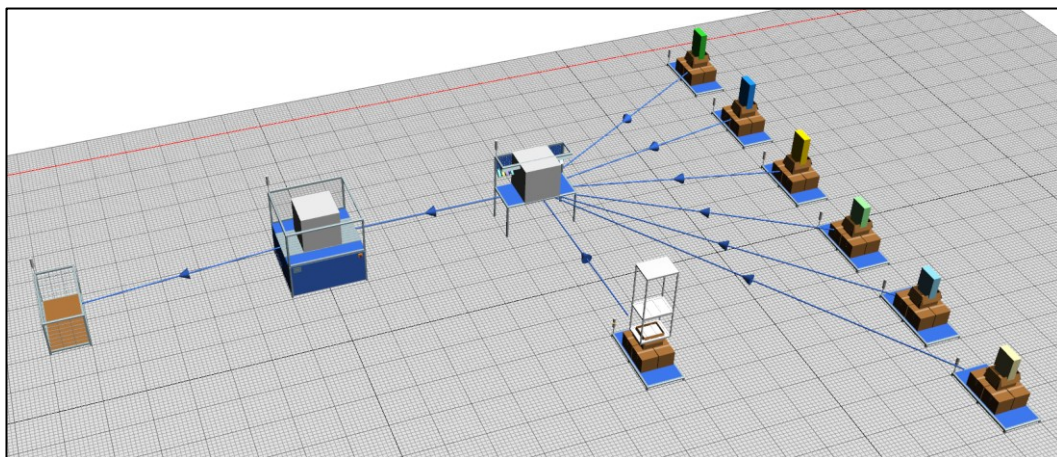
4.4 Valmistelevat harjoitukset

4.4.1 21.2. – 22.2.2024

Ensimmäinen asia, josta lähdettiin alkuun, oli koittaa tehdä yksinkertainen kokoonpano oikentapaisilla rakennuspalikoilla, ilman operaattoreita. Ohjelmisto toimii oletusarvoisesti siten, ettei operaattoreita mallinneta, vaan asemat ja työpisteet toimivat itsenäisesti niille määriteltujen aikojen ja muiden parametrien mukaan. Tällainen tapa simuloida on huomattavasti yksinkertaisempaa, mutta ei myöskään anna yhtä realistisia tuloksia.

Kokoonpanon tekemiseen löytyy ohjelmistosta sisäänrakennettu asema "AssemblyStation". Tässä vaiheessa tutustuttiin myös muihin perustoimintoihin, kuten "Source" (lähde) kappaleiden luomiseen mallissa ja "Drain" (nielu) niiden poistamiseen siitä.

Pääasia oli tutustua siihen, kuinka tällaisia perustoimintoja määritellään. Kokoonpanolle pitää luonnollisesti määritellä, mistä osista se koostuu. Samalla voidaan määrittää, kauanko siihen menee aikaa ym. vastaavia tuotantoparametreja. Tällaiset yksinkertaiset asiat onnistuivat ilman sen suurempia ongelmia. Näistä kokeiluista näkyy malli kuvassa 3.



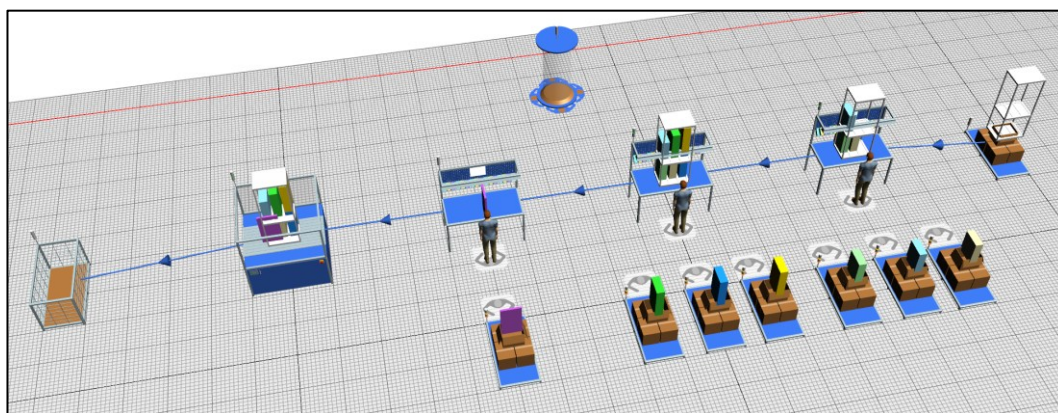
Kuva 3. Ensimmäinen simulaatiomalli yksinkertaiselle kokoonpanolle

4.4.2 23.2.2024

Seuraavaksi harjoitusmallia alettiin jalostaa ottamalla siihen mukaan operaattoreita ja lisäämällä useampia kokoonpanovaiheita muistuttamaan oikeaa kaapituslinjaa. Rakennuspalikoita jäsenneltiin myös tulevan mukaan kokoonpanoon oikeaa valmistustapaa mukaillen.

Operaattoreille annettiin aluksi yksinkertaisia tehtäviä, kuten osien haku kokoonpanopistekohtaisesti sekä niiden kokoonpano. Operaattorien tuominen mukaan malliin lisää huomattavasti sen monimutkaisuutta. Jotta operaattorit saadaan toimimaan halutulla lailla, täytyy kaikki asiat määritellä hyvinkin tarkkaan; mitä tehtäviä niiden pitäisi tehdä, missä järjestyksessä ja millä prioriteetilla sekä miten operaattorien pitäisi toimia, jos niitä on useita toimimassa samalla alueella tai samojen tehtävien parissa.

Mallissa vaihdettiin myös osien perusgrafiikoita muistuttamaan enemmän oikeita iC7-komponentteja. Oikeista komponenteista tehtiin yksinkertaistetut ”palikkamallit” Siemensin NX 3D-mallinnusohjelmalla siten, että niillä oli oikeat mittasuhteet ja värikoodaus, jotta oli helppo tunnistaa eri osat. Simulaatiomalliin ei saada lisäarvoa sillä että jokainen 3D-malli olisi tarkasti mallinnettu, se vaan hidastaisi mallin pyörittelyä ja simulaatioajojen tekemistä. Kuvassa 4 näkyy harjoitusmalli päivän päätteeksi.



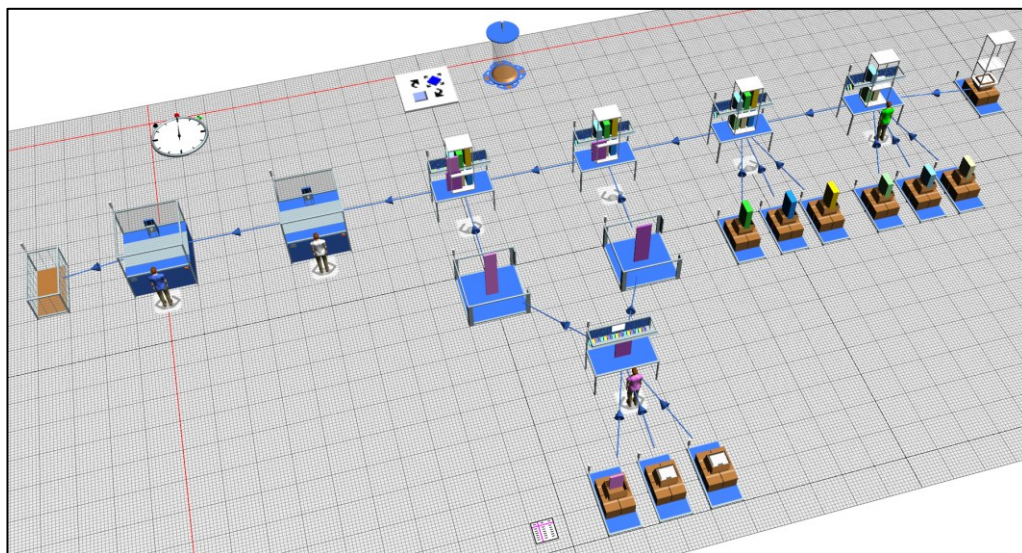
Kuva 4. Harjoitusmalli usealla kokoonpanopisteellä ja operaattorilla

4.4.3 26.2. – 27.2.2024

Seuraava lisäys harjoitusmalliin oli lisätä oma erityinen alikokoonpanopiste kaappien ohjauskokonaisuuden valmistukseen. Tässä oli tarkoitus tutkia sitä, kuinka tehdään alikokoonpano useasta eri komponentista siten, että myös valmiit tuotokset lähtevät prosessissa eri kohteisiin.

Lisäksi operaattorimallinnusta jalostettiin siten, että annettiin operaattoreille eri rooleja ja sitä kautta eri tehtäviä, joita ne tekevät. Tässä mallissa oli kolme eri operaattorityyppiä: kaapituslinjalla toimiva operaattori, viimeistelytoimintojen (tailoring) operaattori ja kaappitestaaja. Kaapituslinjaoperaattori sai hoitaakseen myös kaikki kolme kokoonpanovaihetta ja niiden avainkomponenttien haun lähteistä.

Mallissa tutkittiin myös sitä, miten se käyttäytyy kun lisättiin useita kaapituslinjan operaattoria toimimaan samanaikaisesti. Tässä kohtaa kohdattiin haasteita siinä, miten saadaan operaattorit toimimaan järkevästi yhdessä. Malli oli kuitenkin vielä sen verran yksinkertainen, että se saatiin toimimaan halutulla lailla suhteellisen helposti. Kuvassa 5 on esitetty harjoitusmalli näiden kokeilujen jälkeen.

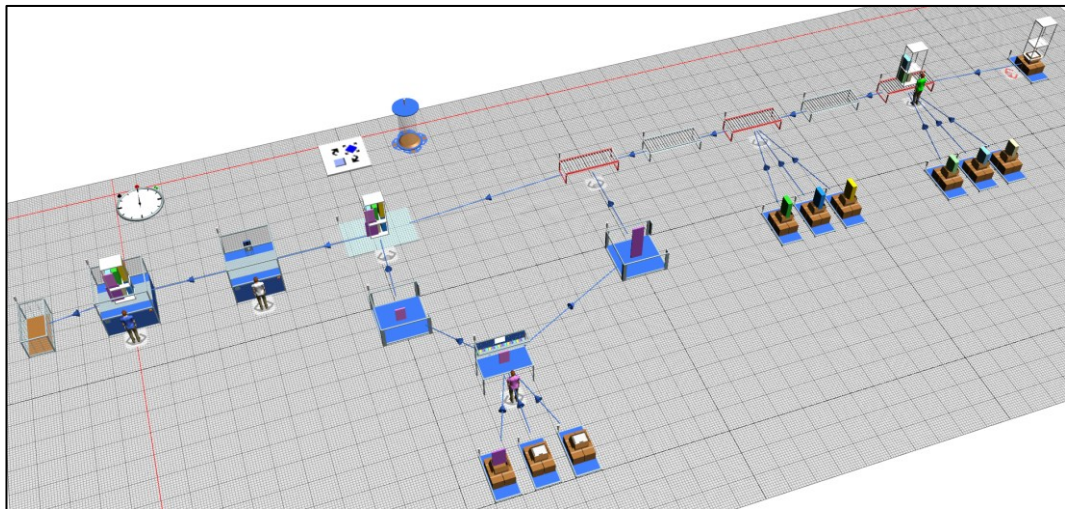


Kuva 5. Harjoitusmalli alikokoonpanolla ja usealla operaattorityypillä

4.4.4 13.3.2024

Mallia kehitettiin vaihtamalla kaapituslinjan kokoonpanopisteiden asennuspöytien ja puskuripöytien grafiikat oikeanlaisiksi sekä lisäämällä operaattorien tekemiä toimenpiteitä, kuten kaappirungon hakeminen varastosta ensimmäiselle asennuspisteelle.

Jostain syystä tätä toimintoa ei saatu toimimaan halutulla lailla. Kun operaattori hakee kokoonpanon perusosan, johon muut osat asennetaan, toiminto ei lähdekään käyntiin. Jos kyse on perusosaan lisättävistä osista, ei ole mitään ongelmaa. Tässä vaiheessa kyseinen ongelma tiedostettiin, mutta kuitenkin sivuutettiin toistaiseksi. Tiedettiin kuitenkin, että aiheen pariin oli palattava myöhemmin, koska tällainen toiminto olisi sellainen, mitä oikeassa tapauksessa tarvittaisiin. Tämän harjoituksen jälkeen päätettiin alkaa muokata mallia kohti oikeaa työn aihetta. Kuvassa 6 on esitetty simulointimalli tässä vaiheessa.



Kuva 6. Harjoitusmalli oikeilla kaappilinjan grafiikoilla

4.5 Viikko 1

4.5.1 3.4.2024

Varsinaisen simulaatiomallin rakennus aloitettiin siten, että jatkettiin em. harjoitusmallin jalostamista, koska siinä oltiin jo tehty asioita oikean tapauksen mukaisesti (oikea yksinkertaistettu prosessi, oikeita työpisteitä, oikeita rakennuspalikoita, jne.).

Aluksi päätettiin yhtenäistää kaikkien asioiden nimeäminen yhtenäiseen tyyliin selkeyttämään mallin hallintaa. Harjoitusmalliin oli jo luonnollisesti päätynyt hieman erilaisia käytäntöjä kun asiaan ei ollut huomattu kiinnittää tarkasti huomiota.

Sitten palattiin kaappirungon kantamisasiaan ensimmäiselle kokoonpanopisteelle. Vaikka kaikenlaista yritettiin ja Danfossin asiantuntijaltakin kysyttiin, niin ei keksitty keinoa, kuinka operaattori saisi kokoonpanon perusosan kannetuksi ensimmäiselle pisteelle kokoonpanoa varten.

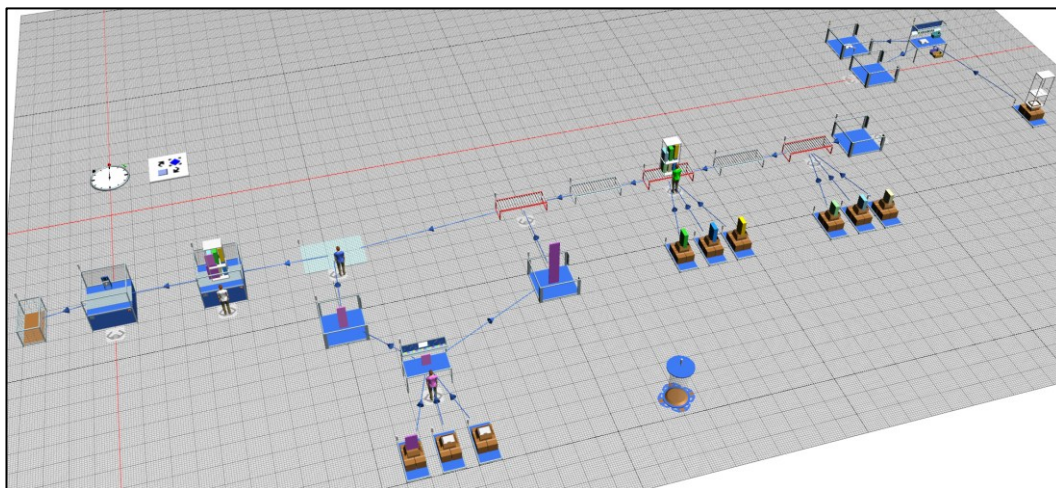
Kokoonpano saadaan alkamaan vain, jos perusosa siirtyy sinne ns. Connectorilla, joten ongelma täytyi kiertää siten, että tehtiin ennen asennuspöytää puskuriasema, johon osan sai kannettua ja josta se siirtyi sitten Connectorilla asennuspöydälle. Tämä oli hieman turhauttavaa, koska tosielämän tapauksessa mitään tällaista puskuria ei ole eikä sitä käytetä, mutta kun muutakaan ei keksitty niin asian annettiin olla.

Kyseisenä päivänä alettiin myös rakentaa toimintoa, jossa kaappirungoista irrotetaan niiden toimituksessa mukana olevat ovet ja takaseinät.

Alkuvaiheessa kohdattiin useita haasteita ja hankaluuksia ohjelmiston käyttämisessä. Vaikutti siltä, että ohjelmiston sisäänrakennetut toiminnot ja logiikat ovat varsin alkeellisia ja että hyvin usein piti itse kirjoittaa SimTalk-koodia, jotta saatiin asioita tapahtumaan halutulla tavalla. Varsinkin operaattorit ajoivat simulaatioajon jumiin hyvin usein, koska niillä ei ole sisäänrakennettuna sellaisia loogisia toimintoja, joita voisi ehkä olettaa olevan itsestäänselvyytenä.

Erityisesti ihmetytti se, ettei nykyaikaisessa ohjelmistossa ole undo-painiketta graafiselle käyttöliittymälle. Tämä tarkoitti sitä, että jos esimerkiksi siirsi vahingossa jotain asemaa, se piti siirtää myös takaisin manuaalisesti. Tämän perustoiminnon puuttuminen johti siihen että välitallennuksia tehtiin aika tiheään.

Toinen iso asia, joka heti alkuvaiheessa tuli mieleen, oli se, miten hallita mallissa kaikille eri asemille ja toiminnoille kirjoitettua SimTalk-koodia, kun malli kasvaa ja monimutkaistuu. Tähänkään ei ohjelmisto itse tarjoa oikeastaan mitään, joten päätettiin pitää itse kopioita kootussa paikassa kaikesta koodista, mitä mallissa on. Tässäkin oli omat sudenkuoppansa kun sitä teki manuaalisesti; välillä unohtui kopioida jotain koottuun tallennuspaikkaan, mikä aiheutti sitten sekaannusta. Kuvassa 7 on esitetty simulointimalli tässä vaiheessa.



Kuva 7. Simulaatiomalli 3.4.2024

4.5.2 4.4.2024

Huomattiin, että yhtenäistetty nimeäminen oli vielä kesken ja liian huonosti määritelty, joten sen kanssa jatkettiin. Syntaksiksi valittiin CamelCase-tyyli, ensin asian tyyppi, sitten sen nimi ja lopuksi järjestysnumero, jos samoja oli useampia, esim. AssyCabinetLine1, BufferTable1, SourceCabinetFrame, jne. Samaa nimeämistyyliä käytettiin tästä eteenpäin kaikkien asioiden kanssa, kuten 3D-mallin asemat, operaattorit, operaattorien toiminnot, SimTalk-koodin muuttujat, jne.

Operaattorien tehtäville määritetyt prioriteetit oli aiemmin harjoitusmallissa numeroitu yhden välein (1, 2, 3, ...), ne muutettiin kymmenen väleille (10, 20, 30, ...), jotta saatiin pelivaraa muutosten helpompaan kokeiluun ja tekemiseen.

4.5.3 5.4.2024

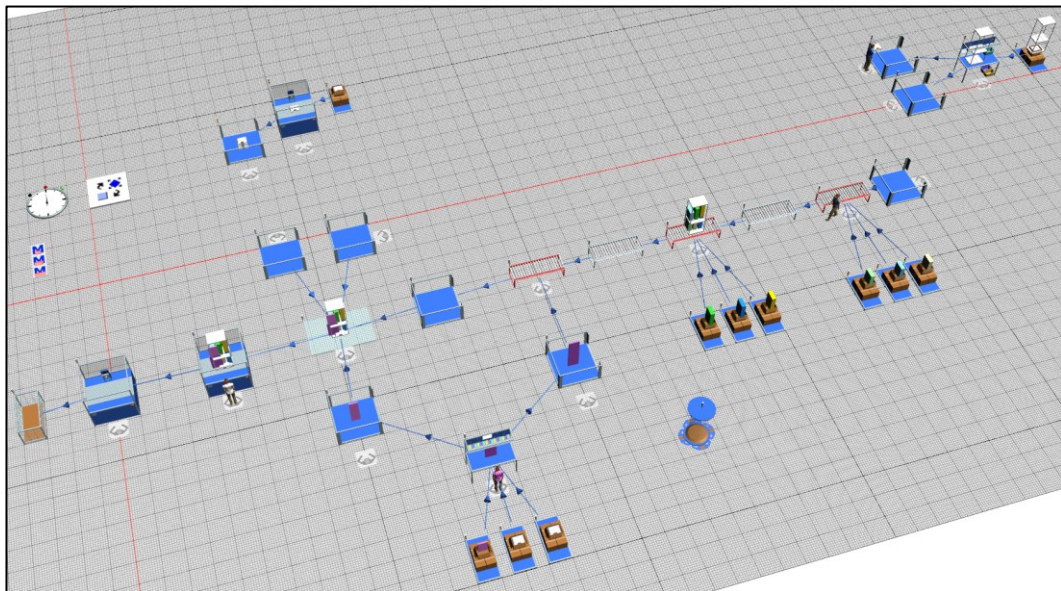
Kyseisenä päivänä taisteltiin kaappirunkojen purkutoiminnon kanssa. Jostain syystä purkuasema käytti operaattoria vain ensimmäisellä kerralla ja sen jälkeen se alkoi toimia itsestään. Ongelma saatiin ratkaistua SimTalk-koodilla lukitsemalla lähteen ulostulo silloin kun purkuasemalle tulee runko ja avaamalla se kun kaappilinjan ensimmäinen vaihe tulee valmiiksi ja kaappi lähtee sieltä pois.

Operaattoritoimintojen prioriteettien kanssa pelattiin, ja kokeiltiin ns. sallittua keskeytystoimintoa, mikä tarkoittaa sitä, että jos se on päällä, operaattori keskeyttää matalamman prioriteetin tehtävän mikäli sille tulee korkeamman prioriteetin tehtävä aktiiviseksi. Lähtökohtaisesti tällaista ei haluttu sallia, vaan työn tekemisen selkeyden vuoksi haluttiin, että jos jokin tehtävä aloitetaan, se myös hoidetaan loppuun.

Lisättiin simulaatioajon ns. perusrutiinit Init, Reset ja EndSim simulointimalliin. Nämä toimivat siten, että Init ajetaan aina silloin kun simulaatioajo aloitetaan, Reset resetoinnin yhteydessä ja EndSim silloin, kun ajo loppuu. Tuntui hieman erikoiselta, että nämä piti itse lisätä malliin.

Muita lisäyksiä olivat kaappipuskurin lisääminen ennen ns. tailoring-vaihetta ja ohjauskokonaisuuskokoonpanon jalostus siten, että operaattori vie valmistuneet osat seuraavien vaiheiden puskureihin, sekä aloitettiin sivuseinien purkutoiminnon tekemistä. Sivuseinien purussa oli ajatus tehdä niiden laitto johonkin ”kärryyn” tai vastaavaan, minkä operaattori siirtäisi sitten tailoring-kokoonpanon alueelle.

Ohjelmistossa on valmiina yksinkertaisia malliesimerkkejä eri toiminnoista, jotka olivat avuksi tässä kohtaa kun uusia asioita tehtiin ensimmäistä kertaa. Vaikka ne hyödyllisiä välillä olivatkin, niin olisi toivonut, että niissä olisi selitetty asioita hieman perustavammin ja kattavammin. Aika usein jäätiin myös ilman vastauksia. Em. kärryn tekeminen sivuseinien tai ovien siirtämiseen vaikutti myös erittäin hankalalta toteuttaa, vaikka kuvittelisi tämäntapaisen toiminnon olevan aika yleinen valmistavassa teollisuudessa. Kuvassa 8 on esitetty simulointimalli tässä vaiheessa.



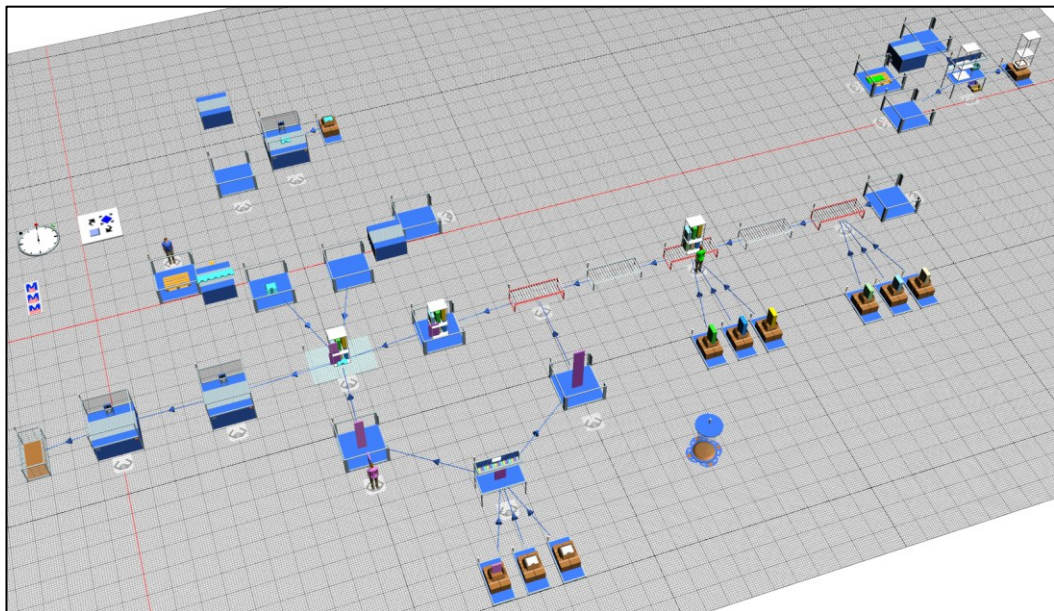
Kuva 8. Simulaatiomalli 5.4.2024

4.6 Viikko 2

4.6.1 8.4.2024

Ovi- ja seinäkärkyjen toteutusta lähdettiin kokeilemaan ns. TransferStationin kautta. Tällä tyyllillä tehtynä suhteellisen yksinkertaisen oloinen toiminto paisui aika monimutkaiseksi. TransferStation vaatii puskurit sekä ennen että jälkeen aseman, joten taas pitäisi lisätä asioita, joilla ei oikeastaan ole tosielämän tapauksessa vastinetta.

Erikoista oli myös se, ettei kyseistä TransferStationia ollut edes saatavilla ohjelmiston perustoimintojen joukosta, vaan se oli ns. add-on-toiminto, joka piti osata hakea omasta paikastaan. Kärkytoiminto saatiin jotenkuten rakennettua tällä tavalla, mutta toteutus jätti myös toivomisen varaa, koska sitten kärkyjä saatiin siirreltyä vain jos ne olivat täynnä. Malliin lisättiin testailuja ja debuggailuja varten ns. jokerioperaattori, joka pystyi tekemään kaikkia mallissa määritettyjä tehtäviä. Kuvassa 9 on esitetty simulointimalli tässä vaiheessa.



Kuva 9. Simulointimalli 8.4.2024

4.6.2 9.4.2024

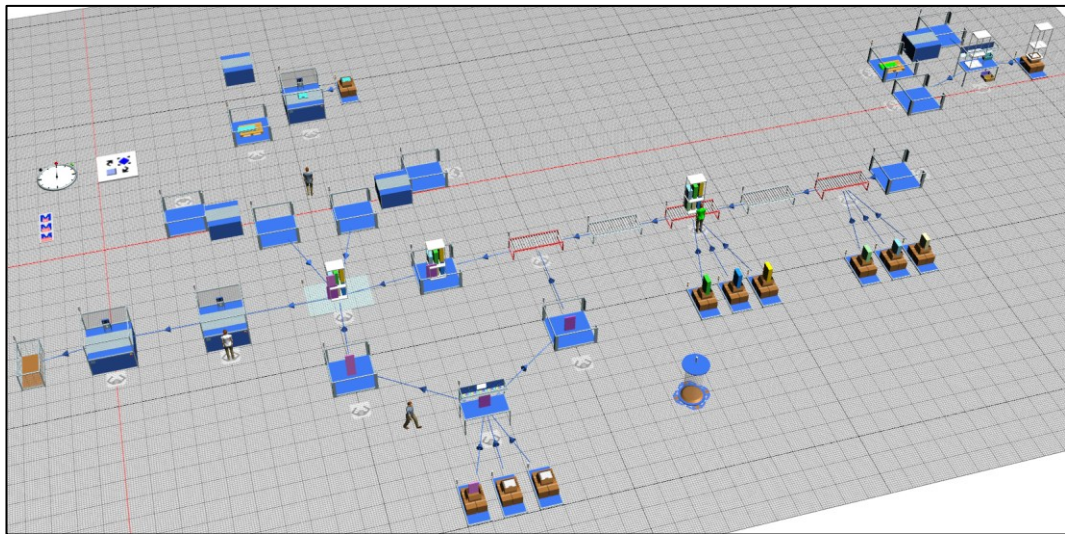
Kyseisenä päivänä keskityttiin SimTalk-ohjelmointiin. Koodia tarvittiin sivuseinien purkutoiminnon kontrollointiin seuraavasti:

- Kun purettu sivuseinäkärky viedään syöntipaikalle, lukitaan purkupaikkabufferin exit.
- Avataan sama silloin kun syöntipaikan bufferi tyhjenee.
- Tämä siksi ettei uutta sivuseinäkärkyä aleta purkamaan ennenkuin puskuri on tyhjä.

Näin tämä saatiin jotenkin toimimaan, mutta todettiin myös, ettei tämä logiikka ole varmaan järkevä todellisuudessa. Todettiin myös, että tämä monimutkainen TransferStation-toteutus on huonompi kuin sitä edeltänyt tapa, jossa varsinaisia kärkyjä ei simuloida, vaan operaattori vaan kantaa osia monta kerralla puskurista toiseen. Vaikka tällainen tapa ei ehkä jäljennä todellisuutta täydellisesti, niin itse toiminto ja sen logiikka on kuitenkin sama.

Tässä kohtaa päätettiin mennä takaisin vanhaan yksinkertaisempaan tapaan, mutta säästettiin TransferStation-versio siltä varalta, että jos jouduttaisiinkin pyörtämään päätös toiseen kertaan.

Sivuseinien purkutoimintoon lisättiin myös se, että yhdellä työvaiheella saadaan viiden kappaleen erä sivuseinäpareja puretuksi. Tämä toiminto piti koodata itse, valmista ominaisuutta tähän ei ollut olemassa. Kuvassa 10 on esitetty simulointimalli tässä vaiheessa.

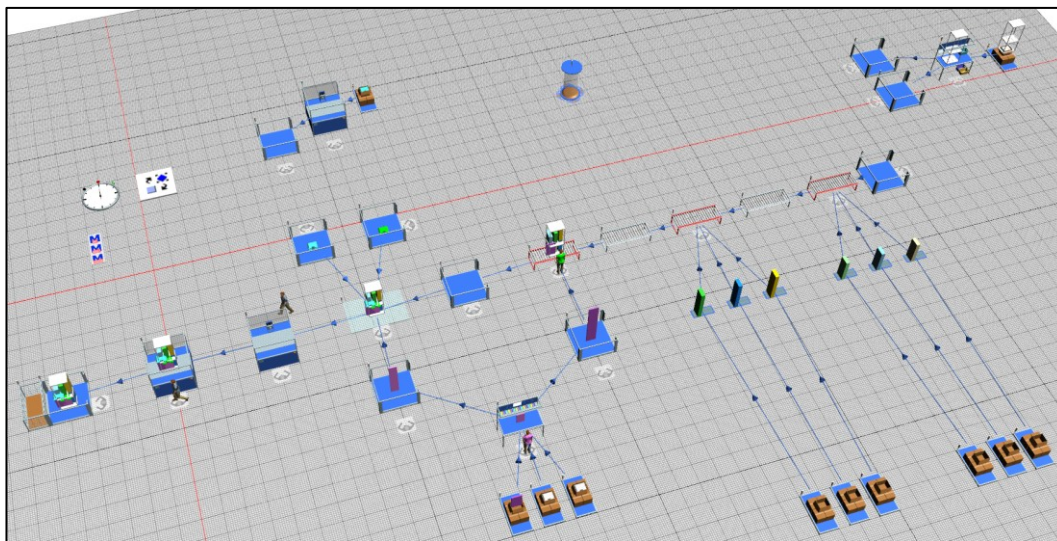


Kuva 10. Simulointimalli 9.4.2024

4.6.3 10.4.2024

Ovien ja seinien purkutoiminto yksinkertaistettiin siten, että otettiin TransferStationit ja sen vaatimat puskurit pois mallista. SimTalk-koodia alettiin siistimään ja tekemään täsmällisemmäksi siten, että korvattiin suoria viittauksia esim. tiettyihin asemiin tai operaattoreihin ns. suhteellisiin viittauksiin erikoismerkeillä kuten "@" ja "?". Näillä saatiin viittaukset suoraan niihin asioihin tai operaattoreihin jotka liipaisevat koodiosuuksia käyntiin, joiden tiedettiin olevan myös välttämättömiä jossain vaiheessa kun malli kasvaa kattavammaksi ja yleismaailmallisemmaksi.

Lisättiin operaattoreille lisää siirtotehtäviä kaappien siirtämiseen kaapituslinjalle, tailoring-vaiheeseen sekä testaukseen. Näitä varten piti myös lisätä asemille ns. Workplaceja, jotka toimivat rajapintana operaattorin ja aseman välillä. Mallia aloitettiin myös laajentaa pelkästä kaapitustoiminnosta myös kohti systeemimoduulilinjaa lisäämällä varastopuskurit sen toimittamille moduuleille. Kuvassa 11 on esitetty simulointimalli tässä vaiheessa.



Kuva 11. Simulointimalli 10.4.2024

4.6.4 11.4.2024

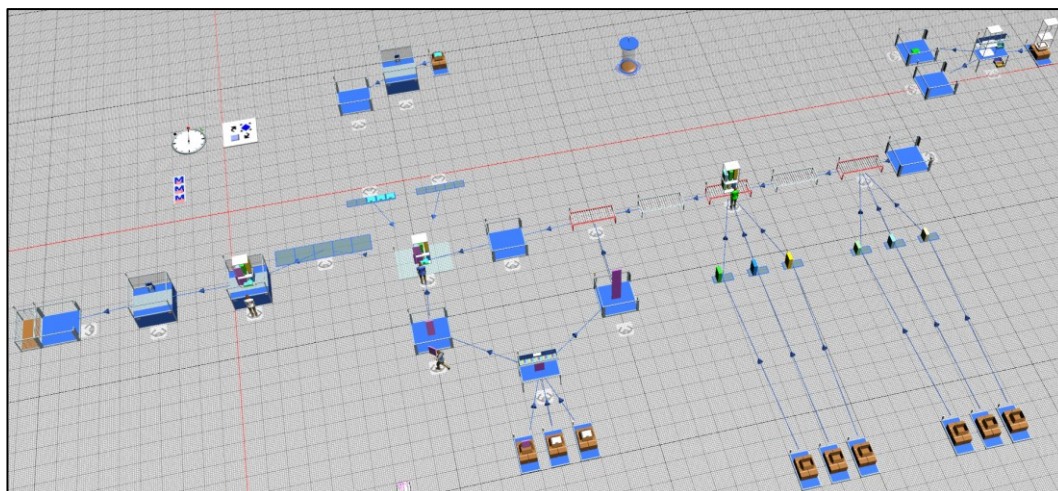
Eräs haaste, joka puskureiden (Buffer) kanssa tuli vastaan, oli se, ettei keksitty mitään keinoa laittaa osia puskureihin muuta kuin päällekkäin. Toiminnallisesti sillä ei ole mitään väliä, mutta mallissa kiinnosti myös tilankäyttö ja paljonko tilaa tarvittiin esim. puolivalmiiden kaappien puskuroinnille.

Mallissa alettiin vaihtaa puskureiden tyyppiä Buffereista Storeiksi, joilla pystytään hallitsemaan selkeämmin ja monin tavoin se miten osat siellä varastoidaan. Siinä on kuitenkin se huono puoli, että Store-tyyppinen varasto sisältää vähemmän sisäänrakennettuja ominaisuuksia, eli sen ohjaaminen tarvitsee enemmän SimTalk-koodia. Malliin lisättiin myös uusi välivarasto (Store) tailoringin ja testauksen väliin, kuten todellisuudessakin.

Malliin piti lisätä koodia hallitsemaan sitä, etteivät operaattorit aja mallia jumiin. Sisäänrakennetulla logiikalla operaattorit yrittävät esimerkiksi viedä osia seuraaville asemille vaikka ne eivät ole vapaana. Tällaisiin tapauksiin piti luoda logiikkaa lukitsemalla asemien ulostuloja testauksen ja pakkauksen välille, sekä tailoringin ja testauksen välille. Nämä toteutettiin siten, että jos osa tulee sisään jälkimmäiseen, laitetaan samalla edellisen aseman ulostulo lukkoon. Lukko

vapautetaan kun osa lähtee jälkimmäisestä asemasta. Tämäntapaisia ratkaisuja tarvittiin siis jokaiseen väliin, missä operaattori vie osia toisille asemille, joten hajaantunutta koodia tuli moneen paikkaan mistä olisi ollut vaikea pysyä perillä jos ei itse olisi kopioinut koodia keskitettyyn dokumenttiin.

Haasteena tänä päivänä oli löytää apua näihin asioihin, joita tehtiin. Ohjelmistossa on kattavanoloinen help-osio, mutta aika usein se silti jättää ilman vastauksia. Internetistä ei myöskään yleensä löydy apua, koska ohjelmisto ei ole mitenkään kovin yleisessä käytössä eikä siitä ole siis kirjoitettu foorumeille juurikaan. Tukea saatiin onneksi Danfossin simulointiexperttien tiimistä. Kuvassa 12 on esitetty simulointimalli tässä vaiheessa.



Kuva 12. Simulointimalli 11.4.2024

4.6.5 12.4.2024

Aiemmin tehdyt ratkaisut operaattorien aiheuttamiin jumeihin asemien ulostuloja lukitsemalla eivät aina riittäneet kaikkien tapauksien estämiseksi, joten piti keksiä jotain lisäystä logiikkaan. Näihin tapauksiin tehtiin koodiin odotuspyyntöjä, että ennen kuin operaattorille annetaan siirtokäsky, odotetaan ensin esim. että puskurissa on tilaa. Tämäntapaiset ratkaisut lisättiin kaikkiin kohtiin, joihin oli jo aiemmin tehty asemien ulostulojen lukituslogiikkaa varmentamaan toimintaa kaikissa tilanteissa.

Haasteina oli se, että jostain syystä Store-tyyppisen puskurivaraston ulostuloa ei voi lukita, mikä aiheutti puolestaan sen, ettei em. logiikkaa voitu sellaisenaan käyttää, vaan koodia piti lisätä erilailla samojen jumien estämiseksi.

Toisena asiana huomattiin, että operaattorien tehtäville annetut prioriteetit eivät oikeastaan merkinneet yhtään mitään, jos tehtäville ei sallittu niiden keskeyttämistä. Eli tilanteessa, jossa operaattori saa jonkin tehtävän valmiiksi ja saa sillä hetkellä uuden korkea-arvoisen tehtävän, niin jos sillä oli jo jonossa aloittamaton alempiarvoinen tehtävä, operaattori aloittaakin sen vaikka toisen periaatteessa samalla hetkellä annetun tehtävän prioriteetti on korkeampi.

4.7 Viikko 3

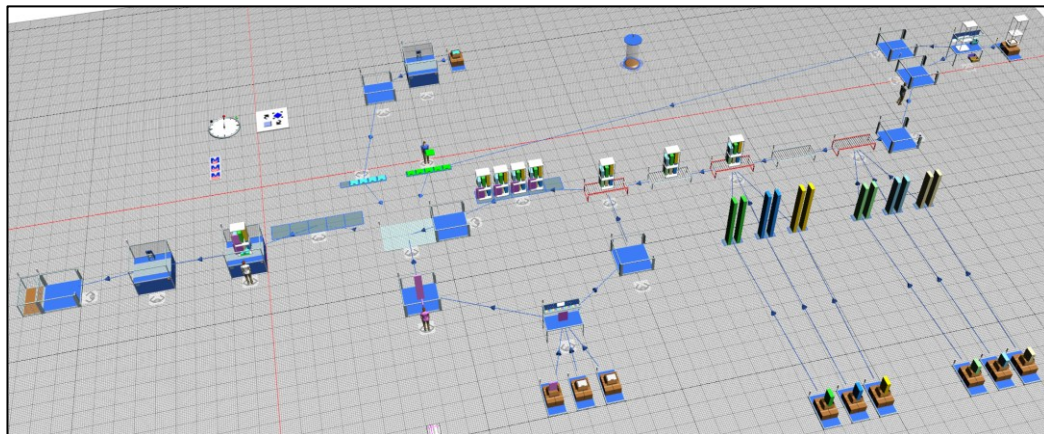
4.7.1 15.4.2024

Kaapituslinjan ja tailoringin välissä ollut puskuri muutettiin Store-tyyppiseksi välivarastoksi. Silti tarvittiin tailoringin yhteyteen myös ”bufferi” aiemmin jo harjoitusmallin kanssa vastaantulleen ongelman vuoksi eli kokoonpanon perusosaa ei voi kantaa kokoonpanopaikalle.

Aiemmin mainittuun prioriteettiongelmaan keksittiin ratkaisu tehdä koodilla sellainen kiertotie, että operaattorit laitettiin aina tehtävän suorittamisen jälkeen ”unohtamaan” kaikki tehtävänsä 0,01 sekunnin ajaksi, minkä jälkeen uudet ja jonossa jo olleet tehtävät tarkasteltiin samalta viivalta, jolloin prioriteettijärjestys alkoikin toimimaan. Tuntui uskomattomalta, että tällainen toiminto piti itse keksiä ja lisätä, jotta operaattorit saatiin toimimaan prioriteettien mukaan, mutta muutakaan tapaa ei löydetty. Tämä ratkaisu lisättiin jokaiselle operaattorityypille.

Tässä kohtaa piti keksiä ratkaisu siihen, että operaattorin pitää välillä siirtää eri määrä kappaleita; joko yksittäisiä kaappeja tai sitten useita ovia tai seiniä kerralla (”siirtää niitä kärryllä”). Tällaista ominaisuutta ei osilla itsellään ollut, mutta operaattorilla oli, joten ongelma ratkaistiin siten, että muutettiin koodilla eri asemille sinne tulevan operaattorin siirtokapasiteettia siellä olevien kappaleiden

mukaisesti eli esim. sivuseinäpuskurilla muutettiin siirtokapasiteetiksi viisi kappaletta ja kaappipuskureissa yhteen kappaleeseen. Lisäksi pelattiin erilaisilla prioriteettiarvoilla, kun haettiin operaattoreille järkevintä tapaa toimia tuotannossa. Kuvassa 13 on esitetty simulointimalli tässä vaiheessa.



Kuva 13. Simulointimalli 15.4.2024

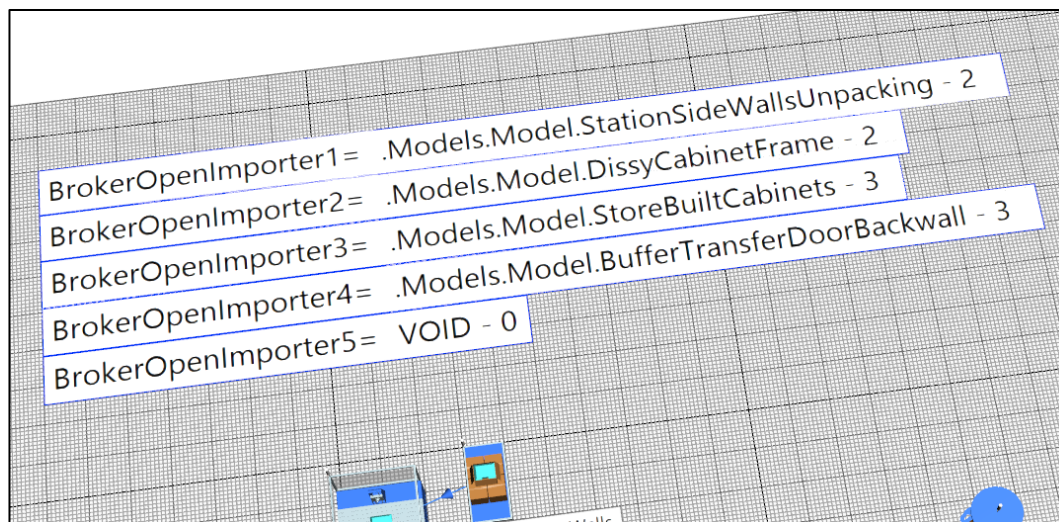
4.7.2 16.4.2024

Kaapituslinjan ja tailoringin välisen kaappipuskurin ehto muutettiin oikeaksi siirron aktivointiin. Aiemmin oli siten, että katsottiin ovatko seuraavat vaiheet "täynnä", mikä ei aktivoitunut jos kaappi oli paikalla, mutta kaikki osat eivät. Vaihdettiin, että katsotaan, onko se "tyhjä" sen sijaan. Tailoring-vaiheen tarpeeton "bufferi" aiheutti ylimääräisen mahdollisen jumipisteen simulaatioajolle. Tähän piti lisätä myös hieman koodia, että se saatiin ratkaistua.

Toinen jumipiste oli ovi- ja takaseinien syöntipiste tailoring-vaiheen yhteydessä. Välillä tuli tilanteita kun sinne yritettiin tuoda enemmän kappaleita kun sinne mahtui. Ensiapuna tähän kasvatettiin vain tämän puskurin kapasiteetti suuremmaksi kuin näiden osien purkupaikalle, jolloin tällainen jumi on mahdoton.

4.7.3 17.4.2024

Tehtiin koodaamalla malliin näkyviin reaaliajassa päivittyvä operaattorien tehtävälista, josta näki tehtävät, joita malli niille antaa. Tämä tehtiin helpottamaan debuggailua operaattorien toiminnan ohjaamisessa, jonka kanssa oli edelleen haasteita. Nämä apumuuttujat mallissa on esitetty kuvassa 14.

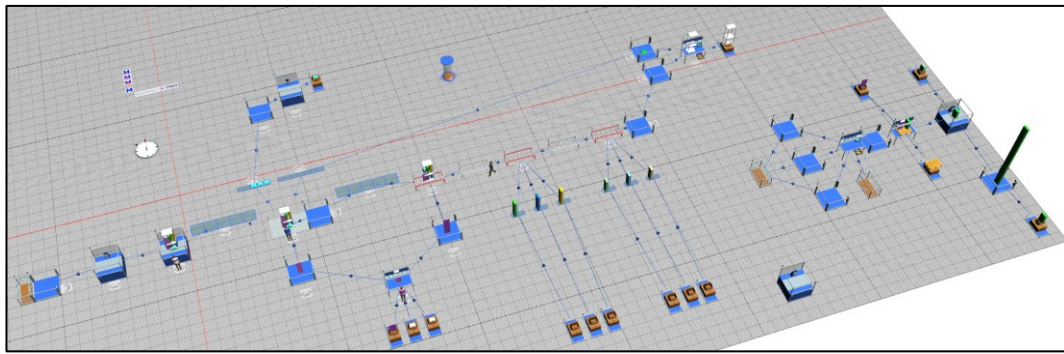


Kuva 14. Malliin lisätty operaattorien tehtävälista 17.4.2024

4.7.4 18.4.2024

Lisättiin SimTalk-rutiini, jolla saatiin simulointiajo pysäytettyä halutulla ajankohdalla. Tämä tehtiin myös debuggailun helpottamiseksi, jotta päästiin helposti aina siihen ajankohtaan simulointiajoa, jossa jokin ongelma tapahtui.

Tämä rutiini laitettiin toimimaan myös siten, että se jätti huomiomatta koodiin asetetut "break pointit" (joihin simulointiajo taukoaa) ajalta ennen haluttua pysäytysaikaa. Tällöin ne break pointit, joita haluttiin ongelmanratkaisussa tarkastella saivat olla siellä mihin ne haluttiin, mutta ne eivät aiheuttaneet turhia taukoja ennen kuin kiinnostava ajanjakso oli käsillä. Malliin alettiin myös luonnostella ja lisätä systeemimoduulilinjan osuutta. Kuvassa 15 on esitetty simulointimalli tässä vaiheessa.



Kuva 15. Simulointimalli 18.4.2024

4.7.5 19.4.2024

Periaatteellinen malli systeemimoduulitoiminnoista saatiin rakennettua. Samalla muutettiin ohjauskokonaisuuden kokoonpano realistisemmaksi siten, että osa sen tarvitsemista rakennuspalikoista tulee systeemimoduulituotannosta, eikä erillisistä lähteistä.

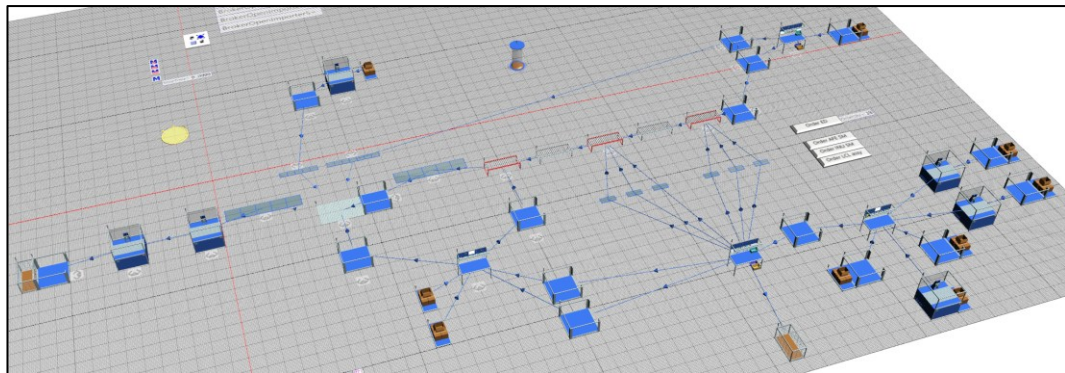
Systeemimoduulipuolen simulaatio päätettiin tehdä yksinkertaistetusti ilman operaattoreita, koska työssä on tarkoitus keskittyä erityisesti kaapituspuoleen, ja se lisäisi mallin monimutkaisuutta sen verran, ettei työhön varattu aika varmaan riittäisi kunnolla. Työvaiheet kuitenkin tehdään oikeilla kellotetuilla vaiheajoilla, joten tästä huolimatta sen pitäisi kuitenkin toimia suhteellisen realistisesti.

Alettiin myös rakentaa systeemimoduulipuolelle tilauspohjaista valmistusta eli sitä, että lähteet toimivat erillisen tilauslistan perusteella. Se laitettiin tekemään tilauksesta AFE- ja INU-systeemimoduulit, LCL-kokonaisuudet sekä toimittaa niiden sisältämät rakennuspalikat oikeille paikoilleen kaapitustuotannon puolelle. Alkuvaiheessa nämä tilaustestailut tehtiin malliin lisätyillä painikkeilla.

Tilauspohjaisessa systeemissä oli haasteita saada asiat tapahtumaan oikeassa järjestyksessä. Jos järjestys oli väärä, systeemimoduulin tapaiset ”alikokoonpanot” eivät esimerkiksi saaneet kaikkia tarvitsemiaan osia, ja malli meni jumiin. Ongelma ratkesi sillä, että koodiin piti lisätä lyhyitä odotuksia luotavien osien välille, jolloin järjestys pysyi haluttuna.

Piti myös lisätä ns. "luontipuskurit" heti lähteiden jälkeen, joihin mahtui kerralla kaikki tilattavat osat, jotta saatiin järjestys pysymään. Niihin saatiin siis luotua oikealla sekvenssillä kaikki tilauksen osat, josta ne sitten siirtyivät oikeassa järjestyksessä eteenpäin. Järjestys meni heti sekaisin jos kapasiteetti ei kerralla riittänytkään.

Pienempien alikokoonpanojen valmistustilausten lisäksi alettiin luonnostella koko kaapitustuotteen tilaamista kerralla sekä lisättiin kaapituspuolelle samanlainen toiminto, että kaappirungot luodaan samalla tilauksella. Järjestelty systeemimoduulituotannon asemia siten, että poistettu ylimääräisiä asioita. Kuvassa 16 on esitetty simulointimalli tässä vaiheessa.



Kuva 16. Simulointimalli 19.4.2024

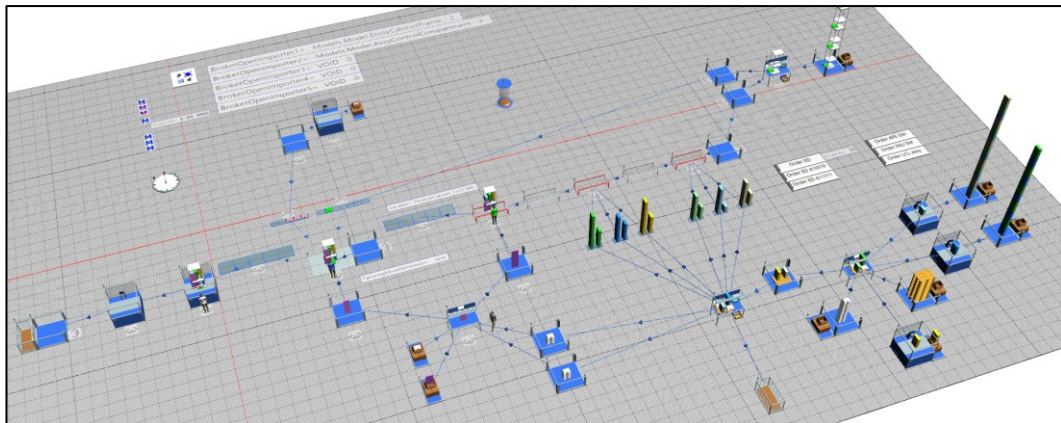
4.8 Viikko 4

4.8.1 22.4.2024

Tähän asti kaikki mallissa on tehty vain yksinkertaisimmalla A10C10-kaapituslopputuotteella ja siihen liittyvillä osilla ja kokonaisuuksilla. Tässä kohtaa alettiin jo valmistautua laajentamaan kohti yleisempää mallinnusta, johon voidaan sitten luoda samoilla valituilla toimintaperiaatteilla haluttu määrä myös muita kaapituslopputuotteita. Tämä aloitettiin jälleen nimeämisasioista eli keksittiin aiemmin käytetyille nimille yleismaailmallisemmat syntaksit, joita ruvettiin käyttämään.

Kaappirunkojen purkupisteelle tehtiin toiminto, jossa rungon luonnin yhteydessä niihin kiinnitettiin koodilla myös kaappirungon tyyppin mukainen ovi- ja takaseinäpaketti simuloimaan todellista tilannetta. (Aiemmin oli vain yksi tyyppi).

Haasteena oli löytää koodin oikeaa syntaksia miten tällainen asia tehdään. Tiedettiin, ettei loogisessa mielessä ole kyse mistään monimutkaisesta tai vaikeasta asiasta, mutta jälleen törmättiin siihen, ettei apua löytynyt ohjelmiston omasta helpistä eikä internetistäkään. Lopulta syntaksi löytyi tutkimalla esimerkkimallien koodia. Ohjauspaneelit lisättiin osaksi systeemimoduulituotantoa ja ohjauskokonaisuuden kokoonpanoa. Kuvassa 17 on esitetty simulointimalli tässä vaiheessa.



Kuva 17. Simulointimalli 22.4.2024

4.8.2 23.4.2024

Lisättiin MIS400-kaapin valmistus rungosta tailoring-vaiheeseen asti. Kaappilinjan kokoonpanopisteille tarvittiin tähän jotain kokoonpantavaa, vaikka siihen ei sisällykään yhtään ns. avainkomponenttia, jotka valmistettaisiin sitä varten systeemimoduulipuolella. Tähän ja muihin vastaaviin tarkoituksiin tehtiin komponentti "PlaceholderAssyPart", joka itse asiassa kelpasi kokoonpanolle määritettäväksi myös lukumäärällä 0, joten sitä käytettiin. Lisättiin myös koko lopputuotteen MIS400+1x11 (A11C11) valmistus tähän asti.

Tässä kohtaa tehtiin lopputuotteiden hallintaa varten mallissa kaapeille jako pääkaappeihin ja komponenttikaappeihin. Pääkaappi tarkoittaa lopputuotteen määrittävää kaappia, jonka valmistus aloitetaan kyseiselle lopputuotteelle ensimmäisenä, ja joita sillä voi olla vain 1 kappale. Komponenttikaapit tarkoittavat kaappeja, joita pääkaappiin liitetään tarvittava määrä, jotta koko haluttu lopputuote saadaan muodostettua.

Käytännössä tämä tarkoittaa sitä, että isommille lopputuotteille MIS-kaapit toimivat pääkaappeina, ja jos sitä ei ole, niin sitten se on modulikaappi. Kokoonpano on määritelty Plant Simulationissa aina niin, että on jokin perusosa, johon muita osia liitetään. Tämä toteutustapa mukailee sitä ja mahdollistaa tailoring-kokoonpanon järkevästi.

Koodiin tarvittiin lisäyksiä tailoring-pisteelle hallitsemaan sitä, että koska sinne saa tuoda kaappeja ja koska ei. Tähän tarkoitukseen tehtiin globaali boolean-muuttuja ”TailoringMovePermission”, jolla oli tarkoitus estää kaappien tuominen sinne silloin kun yhden lopputuotteen kaapit ovat jo siellä ja kokoonpano voitiin aloittaa. Lupa annettiin taas kun kokoonpano valmistui ja kokoonpanopiste vapautui.

Tailoringin kaappipuskuriin piti myös lisätä toiminto sille, että mihin operaattori vie sieltä kaappeja. Tämä liittyi siihen hölmöön ominaisuuteen, ettei kokoonpanon perusosaa voinut viedä suoraan kokoonpanopisteelle, vaan se piti viedä sinne esim. ylimääräisen puskuriaseman kautta, kuten tässä mallissa tehtiin. Koodia piti lisätä siihen, että pääkaapit vietiin kyseiselle puskurille ja komponenttikaapit taas suoraan kokoonpanopisteelle. Tähän tailoringin käsittelyyn jäi vielä joitain bugeja, joita ei tämän päivän aikana saatu ratkaistua.

4.8.3 24.4.2024

Tailoringin siirtoasioiden käsittelyä jatkettiin. Siirtologiikkaa jalostettiin siten, että siirtolupa laitettiin heti pois sen jälkeen kun edellinen siirtokäske on annettu. Tällä pyrittiin estämään se, ettei siirtokäske keritä antaa useita ennen kuin ensimmäistä siirtoa on keretty tekemään, koska sellainen tilanne ajoi mallin jälleen jumiin.

Lisäksi alettiin rakentaa tilauspohjaista logiikkaa, millä ajaa simulaatiomallia todellisuuden mukaisesti. Ensiksi tehtiin taulukko (DataTable), johon täytettiin tilauskirja lopputuotteista mitä halutaan valmistaa, kuinka monta, ja missä järjestyksessä. Sitten tehtiin ohjelma, joka lukee taulukkoa ja tekee sen perusteella asema-/avainkomponenttikohtaiset tilaukset mallin lähteille oikeassa järjestyksessä. Ohjelmarakenne jaettiin lopputuotekohtaisiin aliohjelmiin, joissa määritetään aina sen vaatimat rakennuspalikat. Pääohjelmatasolla sitten kutsutaan näitä aliohjelmia taulukon tietojen perusteella. Tilauspohjainen malli saatiin tässä kohtaa toimimaan pienimmillä lopputuotteilla A10C10 ja A11C11. Kuvassa 18 on esitetty esimerkki tilauskirjataulukosta.

| .Models.Model.OrderTable | | |
|--------------------------|----------------|--------------|
| A10C10 | | |
| | string 1 | integer 2 |
| string | ED End Product | Quantity |
| 1 | A10C10 | 2 |
| 2 | A11C11 | 3 |
| 3 | A10C10 | 4 |
| 4 | A11C11 | 3 |
| 5 | A10C10 | 1 |
| 6 | A11C11 | 3 |

Kuva 18. Esimerkki tilauskirjataulukosta

4.8.4 25.4.2024

Tässä kohtaa todettiin, ettei senhetkinen ratkaisu tailoringin kaappipuskurin käsittelyssä tulisi toimimaan yleisellä tasolla kun isompia ja monimutkaisempia lopputuotteita otetaan malliin mukaan.

Tähän asti se toimi niin, että puskuriin tulevia kaappeja pyrittiin siirtämään automaattisesti kun ne sinne saapuivat, mutta tämä vaihdettiin tilauspohjaiseksi. Eli lisättiin tuotekohtaisiin tilausrutiineihin myös näiden siirtojen tilaus, jolla pyrittiin hallitsemaan tilanne paremmin, siten ettei jumeja enää tulisi. Näihin tilauksiin saatiin myös määritettyä suoraan aiemmin kohdattu monimutkaisuus kaappien eri siirto-osoitteille määrittämällä tilausriville myös kohde mihin tilattiin.

Tässä kohtaa näytti siltä, ettei tilauspohjaisessa ratkaisussa välttämättä tarvittaisi tailoringin lupakäsittelyä ollenkaan, mikä yksinkertaistaisi mallia, mikä olisi aina hyvä asia. Mallia koitettiin ajaa väliaikaisesti ilman lupakäsittelyä, ainakaan heti ei törmätty ongelmiin.

4.8.5 26.4.2024

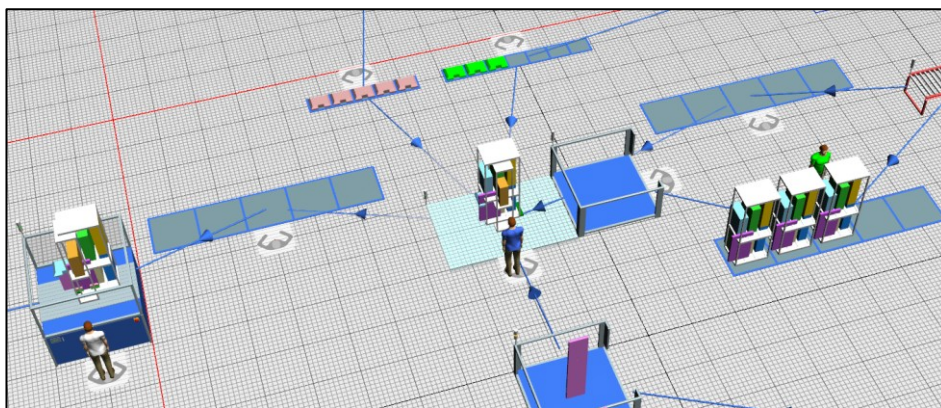
Tässä kohtaa huomattiin, että kaappilinjan ja tailoringin kaappipuskurin väliin tarvittiin myös koodia siirtojen tarkasteluun – muuten puskettiin taas malli jumiin, jos puskuri kerkesi täytyä. Tähän väliin lisättiin samanlainen tarkastelu kuin muissa vastaavissa tapauksissa.

Jatkotestailuissa huomattiin, ettei tilauspohjainen siirtokäsittely tailoringille toiminutkaan halutulla lailla. Kun tilausjonoon laitettiin enemmän lopputuotteita, niin huomattiin ettei etukäteen määritellyllä siirtolistalla mennytkään kaikki oikein, vaan simulaatio ajettiin usein jumiin kun tehtiinkin sellaisia siirtoja mitä ei olisi pitänyt. Tailoringin siirtoluparatkaisu otettiin myös uudelleen käyttöön, mutta se ei ratkaissut ongelmia. Lisäksi siirtokäskyjä jäi joskus väliin jostain syystä, mutta tämän ongelman etsimiseen ei tuhlatu aikaa, koska koko ratkaisu hylättiin tässä kohtaa.

Ongelmaa lähdettiin ratkomaan sillä lailla, että selkeyden vuoksi jaettiin kaappipuskuri omiksi puskureikseen pääkaapeille ja komponenttikaapeille, jolloin siirtoja niistä eteenpäin olisi helpompi hallita. Tämä ajatus toimii myös peilaten todelliseen tilanteeseen, koska nämä puskurit ovat tehtaalla vain varattua lattiatilaa, eivätkä mitään varsinaisia asemia. Lisäksi tällainen jako voisi selkeyttää tilannetta myös oikeassa elämässä, mutta tämä pitäisi arvioida myöhemmin sitten kun ollaan siinä vaiheessa.

Tässä ratkaisussa tehtiin tilauspohjainen siirtolista ainoastaan pääkaapeille, jotta saatiin ohjattua nimenomaan tärkeät lopputuotteet määrittävät perusosat hoidettua oikeassa järjestyksessä tailoringiin. Komponenttikaappipuskuri ei tarvinnut erillistä siirtokäskyä, vaan tailoring-kokoonpano osasi pyytää automaattisesti sen vaatimia osia siinä kohtaa kun sen perusosa (pääkaappi) saapui kokoonpanoon. Tämä oli kieltämättä erittäin yllättävää että tämä tapahtui ns. itsestään, tässä vaiheessa oli kerinnyt tottua jo siihen että kaikki tämäntapainen olisi pitänyt itse koodata.

Tämä ratkaisutapa tarvitsi tailoringin lupakäsittelyn, joten se otettiin takaisin käyttöön. Lisäksi piti tehdä kaappilinjan viimeiselle kokoonpanopisteelle tarkastelu sille, että kumpaan kaappipuskuriin siellä tehty kaappi siirretään sen tyyppin perusteella. Kuvassa 19 on esitetty erilliset kaappipuskurit tailoring-vaiheen ympärillä.



Kuva 19. Erilliset kaappipuskurit ennen tailoringia

4.9 Viikko 5

4.9.1 29.4.2024

IO-kaapit lisättiin mukaan simulointimalliin, mikä tarkoitti 3D-mallien tekemistä, niiden määrittysten lisäämistä kokoonpanopisteille ja tilausrutiineihin, sekä uusien lopputuotteiden määrittävien pääkaappien luomista. Pääkaappityypit oli tuplattava, jotta ne löytyivät sekä versioille, jotka sisältävät IO-kaapin että sellaisille, mitkä eivät sisällä. IO-kaapit eivät vaatineet mitään systeemimoduulipuolella valmistettavia avainkomponentteja.

Tässä kohtaa huomattiin jälleen, että kaivattaisiin yleismaailmallista nimeämissyntaksia pääkaapeille, jotta niiden käsittely olisi selkeää. Tähän valittiin seuraavanlainen syntaksi pääkaapeille:

[kaappisegmentin tyyppi] + "Main" + [lopputuotteen frame] + "_IO"

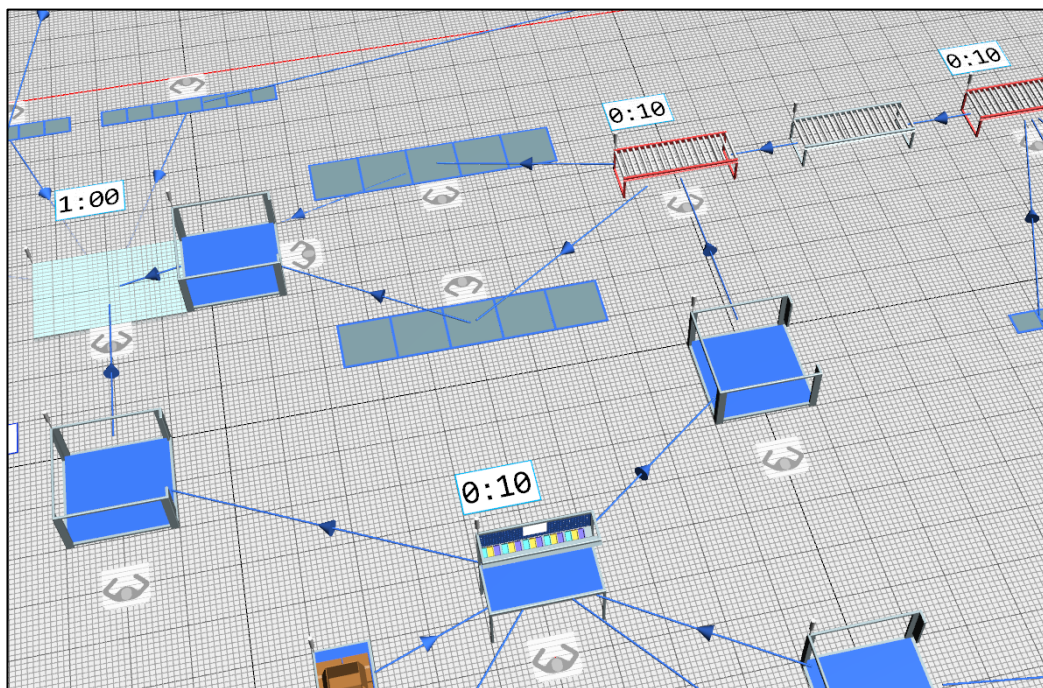
Viimeinen IO-merkintä jätetään luonnollisesti pois jos lopputuote ei sellaista sisällä. Vastaavasti komponenttikaapeilla nimen muodostaa pelkästään ensimmäinen kenttä kaappisegmentin tyylistä. Tänä päivänä myös siivottiin päätilausohjelman koodia, mm. vaihdettiin if-lauseet selkeämmiksi switch-lauseiksi.

4.9.2 30.4.2024

Saatiin idea siitä, että olisi mukava nähdä mallista, kuinka paljon kokoonpanopisteillä on aikaa jäljellä ennenkuin vaihe tulee valmiiksi. Tällainen ratkaisu olisi erityisen mukava silloin, kun mallissa aletaan siirtyä kohti oikeita vaiheaikoja lyhyiden muutamien sekuntien aikojen sijaan.

Tätä lähdettiin toteuttamaan ns. Display-toiminnoilla, joihin sai esimerkiksi aikoja näkyviin. Haluttu toiminta olisi niin, että kun työ aloitetaan, poimitaan siihen tuote- ja vaihekohtainen oikea aika, jota lähdetään laskemaan alaspäin. Kun vaihe on valmis, muutettaisiin ajaksi viimeksi tehdyn tuotteen koko vaiheaika.

Tässä törmättiin pieneen hieman hölmöön ongelmaan siinä, ettei näiden näyttöjen kohdetta voinut muuttaa kuin string-tyyppisenä suoran objekti/muuttuja-viittauksen sijaan. Haluttuun toimintoon tätä piti kuitenkin muuttaa ohjelmallisesti, joten asia piti vain tehdä hieman mutkan kautta. Nämä näytöt ja niiden vaatimat koodit lisättiin kaikille kokoonpanopisteille. Kuvassa 20 on esitetty mallin vaiheajanäyttöjä.



Kuva 20. Esimerkkejä vaiheajanäyttöistä

4.9.3 2.5.2024

Tässä kohtaa projektia oli aika laittaa työvaiheille oikeita vaiheajoja. Minulla oli käytettävissäni kattavasti kelloitetut ajat niistä työvaiheista, joita aiemmissa prototyyppien rakentelussa oltiin pystytty tekemään.

Asiakas voi valita kaapeille paljon erilaisia optioita, joilla on luonnollisesti vaikutusta myös kokoonpanoajoihin. Kellotuksissa tämä asia oli otettu huomioon siten, että jokaisesta vaiheesta oli olemassa aika sille, että jos optioita ei ollut lainkaan, sekä sille että oltiin valittu kaikki mahdolliset optiot. Lisäksi esimerkiksi MIS-kaapeista on olemassa kaksi eri pääversiota, joilla on eri kokoonpanoajat:

syöttökaapelointi alakautta ja kaapelointi yläkautta. Simulaatiossa käytettiin sellaisia laskennallisia keskiarvoaikoja, että sekä optioiden vaikutus että eri kaapin päätyypin ajoista oli otettu painotettu keskiarvot.

Simulaatio-ohjelmistossa olisi mahdollisuus lisätä aikoihin myös satunnaisuutta esim. normaalijakauman tai muiden jakaumien muodossa, mutta koska mallilla halutaan tässä vaiheessa tarkastella tuotantokonseptin teoreettista suorituskykyä, ei tässä vaiheessa haluttu tuoda mukaan satunnaisuutta, joka lisäisi vain yhden monimutkaisuuden asteen analyysihin.

Tietyistä ajoista ei ollut vielä kokemusta eikä kellotuksia, joten ne jouduttiin arvioimaan parhaan tiedon ja asiantuntijoiden arvioiden mukaan. Tällaisia vaiheita olivat tailoring-vaiheet yleisesti kaikkien lopputuotteiden kanssa, testausajat tuotannon testerillä, kaappien ja sivuseinien purkuvaiheet sekä eri lopputuotteiden pakkausajat. Systeemimoduulituotannon ajat olivat jo kellotetut ja tiedossa, mutta niitä ei vielä tämän päivän aikana keritty kerätä ja syöttää malliin.

4.9.4 3.5.2024

Päivä alkoi bugien korjaamisella. Aikanäytöt aiheuttivat häikkää simulaatioajojen välillä, jos ajo pysäytettiin siten, että jokin työvaihe oli kesken. Tällöin tämän työvaiheen aikanäyttö oli asetettu näyttämään meneillään olevan työn jäljelläolevaa aikaa, joten kun uusi ajo aloitettiin eikä siellä ollut mitään, päädyttiin virheeseen. Ongelma korjattiin lisäämällä resetointiohjelmaan kaikkien aikanäyttöjen asetus viimeksi siellä tehdyn vaiheen kokonaisaikaan.

Toinen havaittu ongelma oli jälleen operaattorien ja puskureihin kuljettamisen välillä, tällä kertaa ohjauskokonaisuuden kokoonpanossa. Sinne lisättiin tarkastelu sille, ettei valmista tavaraa lähdetty viemään puskuriin, jos se oli jo täynnä.

Lisäksi tehtiin alustavia testiajoja erilaisilla operaattorien roolituksilla, toisin sanoen rajoitettiin tehtäviä mitä eri operaattorit tekivät, ja koitettiin katsoa mikä

vaikutus sillä oli tuotannon soljuvuuteen. Tästä saatiin hieman suuntaa siitä, että miten tuotantokonsepti toimii, missä on mahdollisesti pullonkauloja jne. Kattavammat testiajot jätettiin kuitenkin suosiolla vielä myöhemmälle, sillä malli ei ollut vielä siinä valmiudessa.

4.10 Viikko 6

4.10.1 6.5.2024

Kaappien purkulogiikka muutettiin siten, että niitä puretaan valmiiksi kaappilinjan alkupään pieneen syöntipuskuriin. Samalla myös ovi- ja takaseinäpuskurien kapasiteettia kasvatettiin, jotta sinne pystyttiin kaikissa tilanteissa mahduttamaan keskeneräisessä tuotannossa olevien kaappien ovet ja takaseinät.

Suuri osa päivästä käytettiin uuden operaattorien prioriteettiongelman parissa. Huomattiin, etteivät operaattorit aina toimineetkaan asetettujen prioriteettien mukaisesti. Prioriteetit oli asetettu siten, että aina pitäisi edistää pisimmällä prosessissa olevia tuotteita, jotta virtaus varmistettaisiin, ja että siihen tulisi mahdollisimman vähän sumia, ja että saataisiin mahdollisimman lyhyt läpimenoaika lopputuotteille. Joissain tilanteissa operaattorit kumminkin ”valitsivat” vähäpätöisemmän tehtävän, vaikka tärkeämpääkin olisi voinut tehdä. Asiaa tutkittiin, mutta syy tähän ei oikein tämän päivän aikana selvinnyt.

4.10.2 7.5.2024

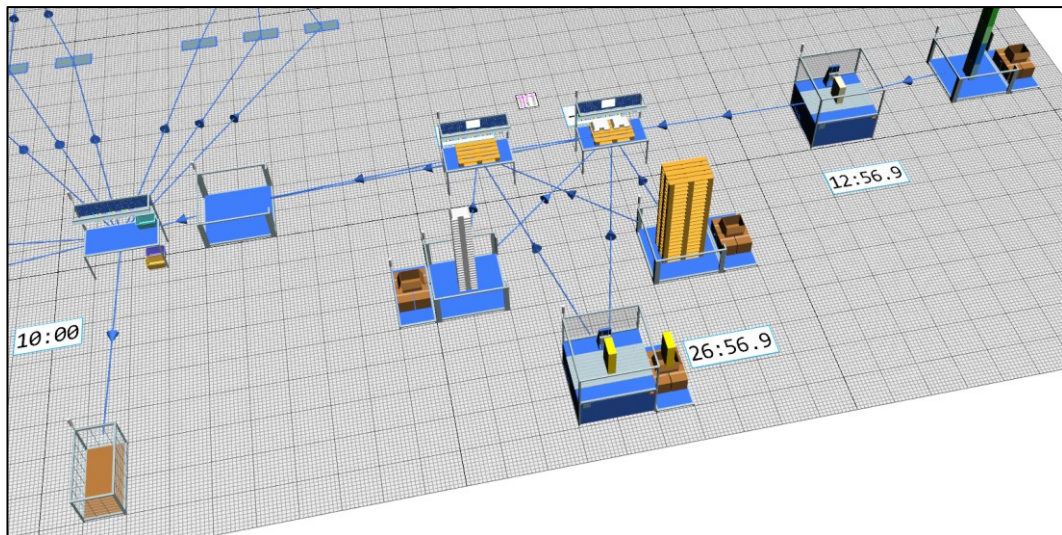
Tässä kohtaa keskityttiin systeemimoduulituotannon jalostamiseen. Aiemmin sen simuloinnin kanssa oltiin tehty ratkaisuja, jotka tekivät siitä helpomman ja selkeämmän tehdä, mutta nyt muutettiin mallin tätä osuutta muistuttamaan enemmän todellisuutta.

Aiemmin ns. ”late dedication” -vaiheella oli mallissa kaksi asemaa, jotka käsittelivät eri tuotteita. Todellisuudessa näitä asemia on kuitenkin vain yksi, jonka läpi kaikki kyseiset tuotteet menevät, joten malli muutettiin myös tämän

mukaiseksi. Lisäksi systeemimoduulin konfigurointiin tehtiin toinen rinnakkainen työpiste samoista syistä.

Systeemimoduulipuolelle lisättiin myös samanlaiset aikanäytöt kuin kaapitustuotannon puolella kaikille työvaiheille, sekä syötettiin niille keskimääräiset oikeat valmistusajat.

Aiemmin mainitun operaattorien prioriteettiongelman lisäksi tuli vastaan uusi ongelma, jossa jotkin simulaatioajot päättyivät virheeseen, sillä jokin operaattori pyrki menemään jo varatulle työpisteelle. Virheen analysoitiin johtuvan aiemmin tehdystä operaattorien prioriteettiongelman kiertotiestä (15.4.2024). Kuvassa 21 on esitetty tarkemmin mallinnettu systeemimoduulituotanto.



Kuva 21. Tarkemmin mallinnettu systeemimoduulituotanto

4.10.3 8.5.2024

Pääasia tänä päivänä oli yrittää ratkaista operaattorien prioriteettiongelma ja virheen aiheuttava varatulle työpisteelle meno-ongelma. Se, että useampi operaattori pyrki hetkellisesti jo varatulle työpisteelle ratkaistiin kaikessa yksinkertaisuudessaan siten, että kyseisten työpisteiden kapasiteetti kasvatettiin yhdestä kahteen operaattoriin. Varsinainen tarkka syy siihen, miksi tällainen ongelma esiintyi, ei oikeastaan täsmällisesti selvinnyt. Kyseessä ei kuitenkaan ollut sellainen tilanne, että toinen operaattoreista olisi mennyt varatulle työpisteelle odottelemaan, vaan kyseessä oli ilmeisesti vaan hetkellinen konflikti operaattorien tehtäväkäskytyksessä. Koska ongelma ratkesi yksinkertaisella korjauksella eikä siinä huomattu mitään negatiivisia vaikutuksia, hyväksyttiin tilanne ja jatkettiin eteenpäin.

Prioriteettiongelmaan kokeiltiin erilaisia ratkaisukeinoja, mutta toimivaa ei saatu vielä otettua käyttöön tämän päivän aikana. Loppupäivästä kuitenkin keksittiin idea, joka näytti alustavien testien perusteella lupaavalta.

Näiden lisäksi tehtiin hieman bugikorjailuja. Systeemimoduulipuolen aikanäytöt oli unohtunut lisätä resetoitiohjelmaan, mikä aiheutti välillä virheitä ja jopa ohjelmiston kaatumisen. Tämä asia korjattiin.

Lisäksi jostain syystä sen jälkeen kun systeemimoduulipuoli oli mallinnettu tarkemmin, kaappilinjan moduulipuskureista ei enää otettu tavaraa kokoonpanoihin automaattisesti kuten aiemmin, vaan ne jäivät sinne vain makaamaan. Kyseisille Store-tyypin puskureille lisättiin koodinpätkä siten, että sinne sisääntuleville kappaleille annettiin heti siirtopyyntö eteenpäin. Tämän jälkeen ne toimivat jälleen, mutta ihmetytti kyllä, miksi tällainen piti tehdä kun ne aiemmin toimivat ilman.

4.10.4 9.5.2024

Operaattorien prioriteettiongelma näytti lopulta ratkeavan. Operaattorien työskentelypisteille (workplace) piti lisätä vielä lisää koodia, jotta prioriteettijonoa tarkasteltiin kuten loogisesti ajatellen sitä pitäisi.

Korjaukseksi tehtiin siten, että kun operaattori lähtee työpisteeltä, niin se asetetaan jälleen hetkeksi ns. "unplanned"-tilaan ja sitten takaisin "planned"-tilaan, jonka jälkeen se tarkastelee tehtävänsä ikäänkuin uudelleen, jolloin ne menevät prioriteettijärjestykseen. Jostain syystä ei siis riittänyt se, että tällainen hetkellinen muistinmenetys tehdään operaattorin vapautuessa tehtävästään, vaan se piti tehdä myös silloin kun se poistuu työpisteeltään.

Tuntui käsittämättömältä että tällaisen asian, jonka pitäisi olla sisäänrakennettu ominaisuus ohjelmassa jota käyttäjän ei tarvitsisi ajatella lainkaan, tekeminen oli näin hankalaa. Kenties tekijältä itseltään oli mennyt jotain ohi pahemman kerran. Yleisesti ottaen kaikki toimiminen operaattorien kanssa tuntui alkeelliselta ja hankalalta ja keskeneräiseltä.

4.11 Viikko 7

4.11.1 13.5.2024

Tänä päivänä oli tarkoitus vaihtaa puskuriasemia tailoring-alueen ympärillä store-tyyppisiksi lähinnä paremman ja realistisemman ulkoasun vuoksi (buffer-tyyppisessä osat sai nähtävästi vain päällekkäin). Tässä kuitenkin möhlittiin jotain ohjauskokoonpanon valmiiden kappaleiden puskurien kanssa, ja osa mallista lakkasi toimimasta oikein. Todettiin että on helpompaa ottaa vanhempi malli uudestaan pöydälle ja aloittaa alusta, kuin yrittää selvittää missä meni vikaan ja korjata ongelmat.

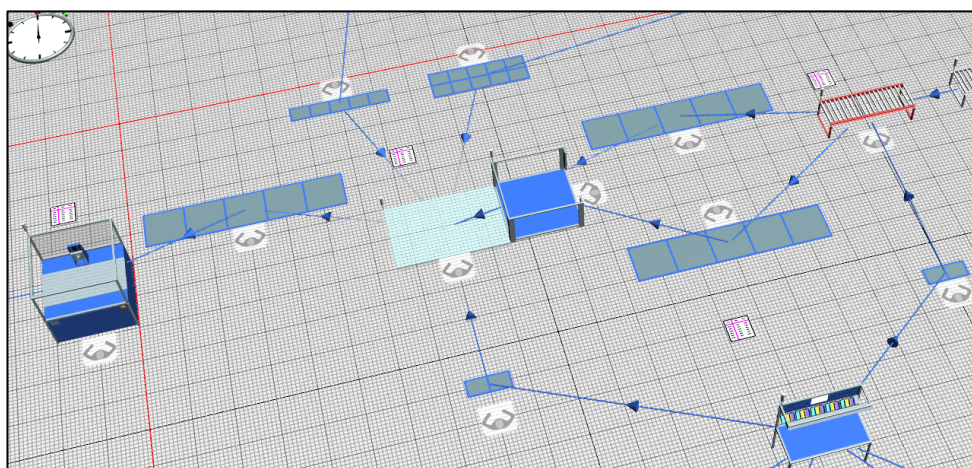
Toimintatapana oli tallentaa joka päivän simulaatiomalli omaksi kopiokseen juuri tästä syystä, että jos tapahtui isoja virheitä, ei kuitenkaan menetettäisi mitään.

Lisäksi toinen syy oli helpottaa tämän työn kirjoittamista, kun voitiin palata tarkastelemaan joka päivän tilannetta uudelleen.

4.11.2 14.5.2024

Edellisen päivän työt unohdettiin ja aloitettiin uudestaan mallin 9.5.2024 pohjalta. Aiemmin mainitut ohjauskokoonpanon valmiiden kappaleiden puskurit saatiin vaihdettua tällä kertaa onnistuneesti store-tyyppisiksi. Pieni kömmähdys kävi siinä että tailoring-vaiheen kokoonpantavien osien "connectorien" järjestysnumerot menivät erilailla kuin ennen, mikä aiheutti sen ettei kaikkia osia otettu kokoonpanoon mukaan. (väärä määritys) Kun tämä hoksattiin, korjaus oli helppoa, "connectorit" numeroitiin uudelleen oikealla lailla.

Osa puskureita skaalattiin myös realistisemmalle kapasiteetille tällä kohtaa; aiemmin oli käytetty vain jotain testilukuja. Bugien korjausten osalta tällä kertaa törmättiin ongelmaan, jossa testaajaoperaattori ajoi itsensä joissain tilanteissa jumiin. Prioriteettien listauksista löydettiin virhe tässä kohtaa, mutta sen korjaaminen ei vielä poistanutkaan ongelmaa, vaan tälle työpisteelle piti myös lisätä 9.5.2024 kuvattu työskentelypisteen koodilisäys. Nämä kaksi asiaa yhdessä korjasivat ongelman. Kuvassa 22 on esitetty uudet store-puskurit.

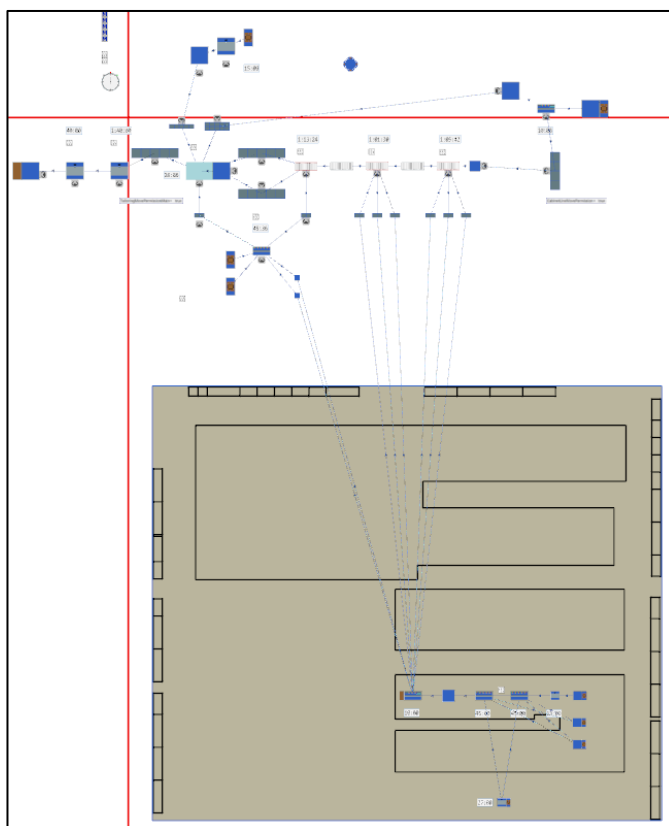


Kuva 22. Uudet store-puskurit tailoring-ympäristössä

4.11.3 15.5.2024

Tässä kohtaa jatkettiin tailoring-ympäristön puskurien vaihtamista buffer-tyypeistä store-tyyppeihin. Tämä saatiin tehtyä kaappirunkopuskureille, joiden osalta tämä varsinkin oli tärkeää, että voidaan arvioida järkevästi sitä minkälaista tilaa ne lattiapinta-alasta voisivat eri tilanteissa vaatia. Storeissa on kuitenkin se huono puoli, että ne tarvitsevat enemmän koodia toimiakseen halutulla tavalla, mutta tarvittavat lisäykset saatiin tehtyä ilman ongelmia. Tässä kohtaa päivitettiin myös ovien ja seinien grafiikat realistisiksi, tähän asti niitä oli mallissa edustaneet vain ohjelmiston oletusgrafiikkapalikat.

Lisäksi malliin tuotiin oikean tehtaan layout pohjaksi, mihin simulaatiomallia alettiin alustavasti sovittaa. Tämän kanssa ei vielä ehditty kovin pitkälle, mutta systeemimoduulipuoli saatiin sinne jotenkuten aseteltua. Kuvassa 23 näkyy tehdas-layout simulointimallissa.



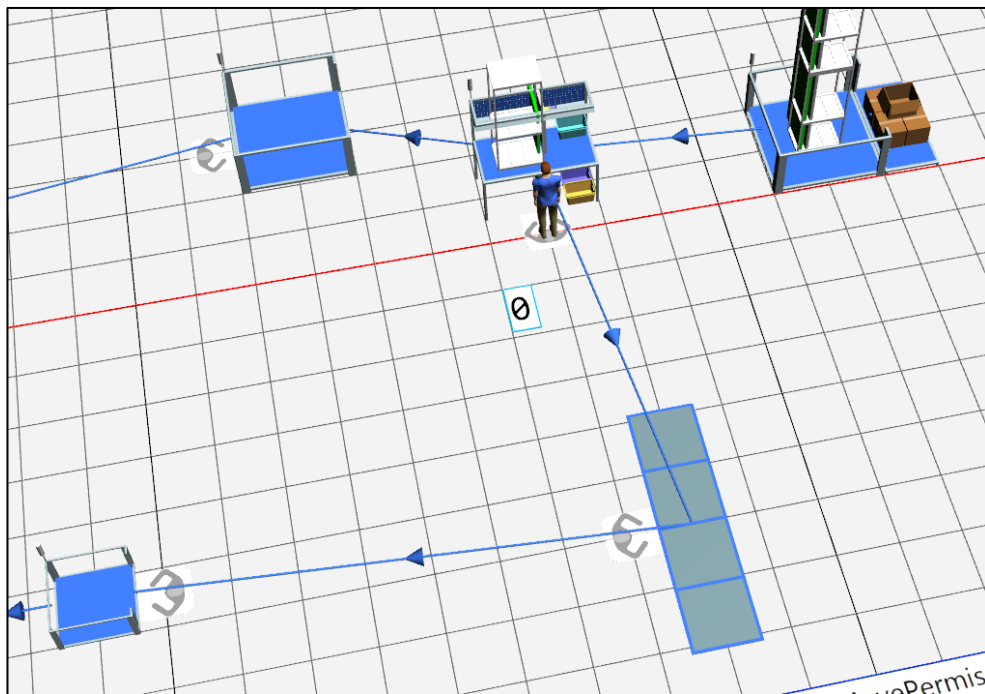
Kuva 23. Tehdas-layoutin tuonti malliin

4.11.4 16.5.2024

Tämä päivä meni kokonaan kaappirunkojen purkupaikan uuden ominaisuuden ongelmanratkaisun kanssa. Purkupaikalle muutettiin toiminta siten, että operaattori kantaa puretut kaapin ja sen ovi- ja takaseinäpaketin eri paikkoihin omiin puskureihinsa. Aiemmin oli jo omat puskurit näille, mutta osat siirtyivät automaattisesti ”connectoreilla”.

Kun alkoi olla aika asetella mallia oikeaan layoutiin, oli selvää, että purkupaikalta tulisi sen verran pitkä matka kaappirunkopuskuriin, etteivät ne voineet enää vain ilmestyä sinne, vaan operaattorin toiminta piti lisätä simulaatioon.

Tämä suhteellisen yksinkertaisen oloinen toiminto aiheutti paljon harmaita hiuksia, sitä ei saatu toimimaan tämän päivän aikana. Oli selvää, että toiminto tarvitsi koodia, mutta oikeaa syntaksia asiaan ei vaan löytynyt. Kuvassa 24 on esitetty kaappirunkojen purkupiste ja siihen liittyvät puskurit.

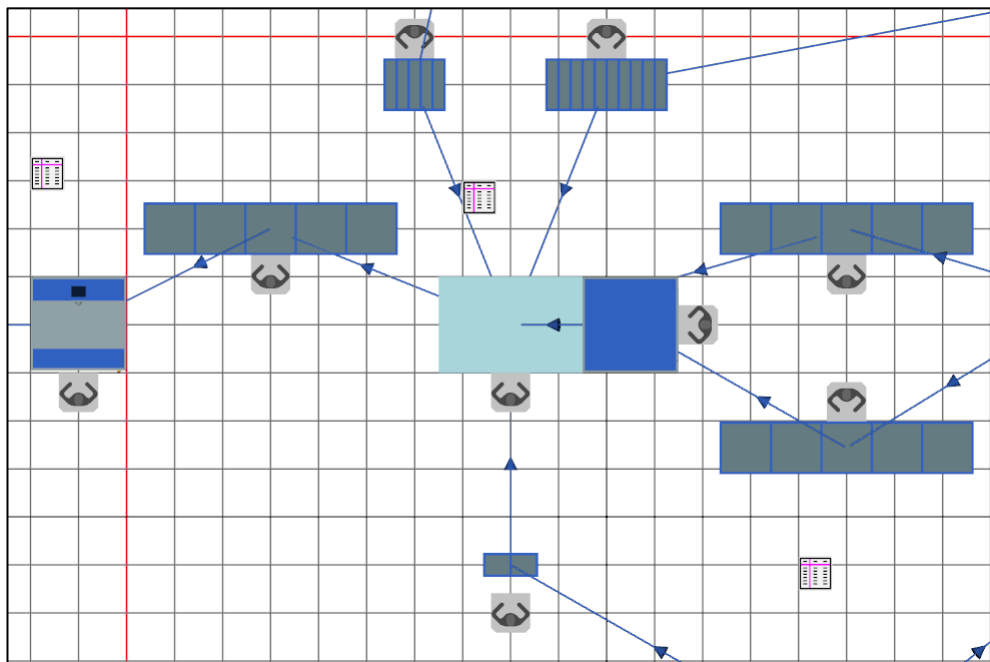


Kuva 24. Kaappirunkojen purkupiste ja puskurit

4.11.5 17.5.2024

Taistelu purkupaikan kanssa jatkui, ja lopulta löytyi sellainen syntaksi millä homma saatiin toimimaan. Jostain käsittämättömästä syystä operaattorin kuljetuksen kohteen määrittävää parametria "MUTarget" ei pysty muuttamaan koodilla kun operaattori saa kuljetustehtävän. Kohde oli kuitenkin vaihdettava, koska puretun ovi- ja takaseinäpaketin ja rungon oli lähdettävä eri paikkoihin. Koodissa pitikin asettaa sen sijaan itse siirrettäville objekteille sellainen parametri kuin "destination".

Tämän päivän aikana saatiin muutettua myös loput buffer-tyypin puskurit store-tyyppiksi tailoring-alueella, eli ovi- ja takaseinäpuskuri sekä sivuseinäpuskurit. Myös nämä tarvitsivat lisää koodia, jotta toiminta pysyi halutunlaisena. Kuvassa 25 näkyvät kaikki tailoring-alueen puskurit.



Kuva 25. Kaikki puskurit store-tyyppiä

4.12 Viikko 8

4.12.1 20.5.2024

Tämä päivä käytettiin mallin sovittamiseen oikeaan layoutiin. Tämä vaati aika paljon aikaa varsinkin asioiden skaalaamiseen sillä lailla, että ne mahtuivat sinne mihin kuuluivat ja että kaikki asioiden nimikyltit ja aikalaskurit saatiin järkevästi ja selkeästi aseteltua.

Malliin lisättiin myös layoutin mukaisesti esteitä niihin paikkoihin, missä on jo esim. kuormalavahyllyjä tai muuta toimintaa, mistä operaattorit eivät saa kulkea.

4.12.2 21.5.2024

Kun layout alkoi olla kohdallaan, ruvettiin sitten tutustumaan siihen, että miten saadaan operaattorien työvuorot ja -ajat määritettyä. Tähän asiaan löytyi sisäänrakennettu toiminto "ShiftCalendar", johon saatiin määritettyä nämä asiat. Näitä kalentereita voisi olla käytössä useampiakin jos niin haluttaisiin. Vuorokalenteriin määritettiin työajat, ja sitten operaattoreille tai asemille voitiin määrittää, että ne käyttävät kyseistä kalenteria toimintaansa.

Aluksi vuorokalenteriin laitettiin pelkästään aamuvuoro siten, että työaika oli klo 7:00-15:00, ja tauot 8:00-8:30, 10:30-11:15, 13:00-13:30, jolloin päivän tehokkaaksi työajaksi jäi 6 tuntia 15 minuuttia. Tämäkin aika on todennäköisesti hieman optimistinen todelliseen keskimääräiseen ihmisen työaikaan, sillä tämä aika ei sisällä mitään ylimääräisiä taukoja, miettimisiä, etsimisiä, yms., joita kuitenkin oikeassa elämässä tapahtuu. Mutta lähtökohdaksi tarvittiin jotain, niin tällä aloitettiin.

Nopeasti huomattiin, että se että pelkästään operaattorien määritettiin käyttävän vuorokalenteria, ei riittänyt. Jos operaattori aloitti esimerkiksi jonkin työvaiheen hetki ennen työpäivän loppua, niin se teki sen kuuliaisesti loppuun vaikka työaika ylittyisi. Jos olisi kyse aina hyvin lyhyistä työvaiheista, tämä voitaisiin ehkä

hyväksyä, mutta ei tässä tapauksessa kun pisimmät työvaiheet ovat jopa 2 tuntia pitkiä.

Kun laitettiin sen sijaan vuorokalenteri käyttöön kaikille asemille, saatiin parempi lopputulos. Lisäksi tämä oli välttämätöntä systeemimoduulipuolen asemien kanssa, koska siellä ei operaattoreita ollut mallinnettuna.

4.12.3 22.5.2024

Malliin lisättiin näytöt näyttämään simulaation aikana reaaliajassa sen, että onko käynnissä työaika vai työajan ulkopuolinen aika. Tämä tehtiin simulaatioajojen seuraamisen helpottamiseksi, ettei tarvinnut ihmetellä että onko koko malli jumissa jos mitään ei näyttänyt tapahtuvan vähään aikaan.

Simulaatioajoissa huomattiin uusi ominaisuus, kun vuorokalenteri oli otettu käyttöön. Aiemmin ohjelma osasi lopettaa ajon itsestään kun kaikki oli tehty tai saavutettiin sellainen jumi, ettei mitään enää tapahtunut. Tämä oli hyvä ominaisuus. mutta nyt ajo jatkui vaan ikuisuuteen asti.

Jälleen törmättiin siis sellaiseen itsestäänselvältä tuntuvaan asiaan, johon piti itse koodata ratkaisu. Malliin lisättiin sitten ohjelma joka pysäyttää simulaatioajon kun viimeinen tilattu lopputuote valmistuu. Systeemimoduulipuolelle lisättiin määritykset kahden AFE:n ja kahden INU:n systeemimoduuleille, mutta tätä ei keretty vielä testaamaan tämän päivän aikana.

4.12.4 23.5.2024

Tässä kohtaa aloitettiin iso urakka rinnanajokaappituotteiden luomisessa malliin. Nämä tuotteet tarkoittavat sellaisia joissa on kahdet modulit ja niiden tarvitsemat muut osat yksien sijaan.

Koska aikaisempien tuotteiden osalta niiden vaatimat asiat olivat jo määritettynä selkeästi ja yleismaailmallisesti, mallin laajentaminen näillä uusilla lopputuotteilla

meni suhteellisen kivuttomasti ja helposti. Rakennuspalikoita ja määrittäksiä oli tehtävänä varsin paljon, mutta niille oli selkeät paikat tiedossa.

Ensin lisättiin lopputuotteet 2A102C10 ja 2A102C10+IO.

Niiden vaatimia lisäyksiä olivat:

- Lisätty 2x systeemimoduulien määrittäys systeemimoduulipuolelle
- Lisätty uudet kaappikentät ja ovet
- Lisätty lopputuotteen tilausrutiini sekä päivitetty päätilausrutiini (ei keritty vielä testata)
- Lisätty oven luonti lähdepuskurille
- Päivitetty kaappirunkojen purkupaikka
 - Purun ja kuljetusten määrittäykset
- Päivitetty kaappilinjan kokoonpanopisteet
 - Kokoonpanojen määrittäykset
- Päivitetty tailoring-piste
 - Kokoonpanojen määrittäykset
- Päivitetty oikeat vaiheajat

Näiden muutosten yhteydessä päivitettiin myös ovi- ja takaseinien siirtologiikka purkupaikalta tailoringiin siten, että siirto pyydetään silloin kun joko tailoring-pisteelle saapuu jokin kaappi, tai sitten jos purkupaikan puskurit täyttyvät.

4.12.5 24.5.2024

Testailua jatkettiin lisättyjen rinnanajokaappilopputuotteiden parissa. Todettiin ongelma siinä, että jos kaappilinjan modulipuskurit pääsevät täyttymään, muiden osien järjestys systeemimoduulipuolelta ohjaus-kokoonpanopisteelle saattaa mennä sekaisin, mikä taas puolestaan aiheuttaa simulaation jumittumisen.

Nopeana korjauksena tähän lisättiin modulipuskureiden kapasiteetti niin suureksi, etteivät ne pääse täyttymään. Tämä ei ollut mikään hyvä pysyvä ratkaisu, mutta

ajoi asiansa tässä kohtaa. Vaikka tietenkään kaappilinjan puskureihin ei määräänsä enempää moduleja oikeasti mahdu, niin voitaisiin ajatella että siinä tapauksessa jos moduleita on saatavilla erittäin runsaasti, niin logistiikka pystyisi hoitamaan ylimenevien varastoinnin jossain muualla sen aikaa, kunnes niille tulisi taas tilaa oikeille syöntipaikoille. Muita uusia ongelmia ei tullut testailussa vastaan, vaan todettiin että uusien tuotteiden lisäys onnistui, ja voitiin jatkaa eteenpäin.

4.13 Viikko 9

4.13.1 28.5.2024

Tässä kohtaa lisättiin lopputuotteet 2A112C11 ja 2A112C11+IO samaan tapaan kuin ensimmäiset rinnanajolopputuotteet. Nämä lopputuotteet vaativat uuden MIS600-tyypin kaapin aiemmin käytetyn MIS400-kaapin sijaan, mutta muuten lisäys meni hyvin samalla kaavalla kuin 23.5.2024.

Lisäyksen jälkeen alettiin testata yleistä simulaatiovarmuutta erilaisilla satunnaisilla tilauskannoilla. Yritettiin siis ajaa mallista kaikki mahdolliset bugit, joita siinä vielä saattoi olla ja korjata niitä sitten. Virheitä löydettiin jonkin verran. Suurin asia oli se, että jossain kohtaa meni tilauksien vaatimien osien järjestys jotenkin sekaisin, mikä sitten ajoi mallin jossain kohtaa jumiin.

Todettiin, että tilausrutiineihin piti lisätä lyhyitä odotuksia tiettyjen osakokonaisuuksien luomisen väliin, jotta niiden järjestys pysyi aina halutunlaisena. Lisäksi ns. luontipuskureiden piti mahtua kerralla kaikki niiden vaatimat osat, koska muuten luontijärjestys meni sekaisin ja taas oltiin jumissa. Luontipuskurit ovat olemassa vain juuri tästä syystä että saadaan syötettyä osia malliin oikeassa järjestyksessä, niillä ei ole tosielämän vastinetta. Luontipuskureiden kapasiteeteiksi asetettiin joka paikkaan 1000 kpl tästä syystä.

Lisäksi löydettiin muutama inhimillinen copy-paste -virhe tilausrutiineista, missä oli jäänyt muuttamatta arvot uuden paikkansa mukaisiksi. Kuvassa 26 on esitetty esimerkki satunnaisesta tilauskannasta.

| .Models.Model.TableOrders | | |
|---------------------------|-----------------------|--------------|
| A11C11+IO | | |
| | string 1 | integer 2 |
| | string ED End Product | Quantity |
| 1 | A11C11+IO | 1 |
| 2 | 2A102C10+IO | 3 |
| 3 | A11C11+IO | 2 |
| 4 | 2A112C11 | 3 |
| 5 | 2A102C10 | 2 |
| 6 | 2A112C11+IO | 3 |
| 7 | A11C11 | 1 |
| 8 | A10C10 | 2 |
| 9 | 2A112C11 | 1 |
| 10 | A11C11 | 3 |
| 11 | A10C10+IO | 1 |
| 12 | A10C10 | 2 |
| 13 | A11C11 | 3 |
| 14 | A11C11 | 1 |
| 15 | A11C11+IO | 2 |

Kuva 26. Esimerkki satunnaisesta tilauskannasta

4.13.2 29.5.2024

Simulaatiovarmuuden ajoja jatkettiin ottamalla vuorokalenteriin myös iltavuoro mukaan ja jatkamalla satunnaisilla tilauskannoilla ajoja. Malli alkoi vaikuttaa toimintavarmalta, bugeja ei enää näyttänyt löytyvän. Tässä kohtaa lisättiin malliin ohjelmistosta valmiina löytyviä työkaluja, kuten "BottleNeckAnalyzer" ja "ExperimentManager".

BottleNeckAnalyzer-työkalulla voidaan yrittää löytää mallista nimensä mukaisesti pullonkauloja. Työkalu näyttää simulaatioajan lopuksi, miten työajat, odotukset ja muut ajat jakautuivat ajossa eri työpisteiden välillä sekä antaa eräänlaisen pisteytyksen näiden perusteella jokaiselle niistä. Isommat pistemäärät kertovat sitten siitä että mitä työpistettä pitäisi sen mukaan kehittää, että saataisiin suurin vaikutus kokonaistehokkuuden parantamiseen. Työkaluun ei kannata sokeasti

luottaa, mutta sen antamia aikajakaumia voi myös itse tarkastella ja analysoida pahimpien pullonkaulojen tunnistamiseksi.

ExperimentManager-työkalu on hyödyllinen apuväline siinä kohtaa, kun halutaan ajaa monia eri simulaatioajoja vaihtuvilla parametreilla. Tällaisia parametreja voivat olla esimerkiksi operaattorien määrä, erilaiset vaiheajat, työvuorojen määrä, tai mitä tahansa vastaavaa. Tässä kohtaa tutustuttiin vain hieman työkalun käyttöön, siitä lisää myöhemmin tulosten analysoinnin yhteydessä.

4.13.3 30.5.2024

Huomattiin, ettei lopputuotteiden läpimenoaikaa (lead time), eli kauanko menee kunkin lopputuotteen valmistukseen menee aikaa, saatu mistään perusominaisuuksista selville helposti. Tähän päätettiin itse rakentaa ratkaisu.

Ratkaisua varten määritettiin jokaiselle pääkaappiobjektille kaksi omaa parametria, "LeadTimeStart" ja "LeadTimeEnd". Ensimmäiseen otettiin ensimmäisen työvaiheen alussa aikaleima ylös, ja sitten kun tuote oli valmis, otettiin aikaleima talteen jälkimmäiseen, ja näistä laskettiin kyseisen tuotteen läpimenoaika. Ajat kirjattiin samalla myös niille tehtyyn omaan taulukkoon jälkitarkastelua varten.

Samassa yhteydessä annettiin kaappiobjekteille myös oma parametri, joka kertoi sen, onko kyseessä pääkaappi vai komponenttikaappi. Tätä parametria hyväksikäyttämällä saatiin yksinkertaistettua ehtolauseiden koodia monessa paikassa, kun sen sijaan että niissä lueteltiin kaikki eri pää- ja komponenttikaapit nimellä erikseen, voitiinkin vertailla vain kaappien tyyppiparametria.

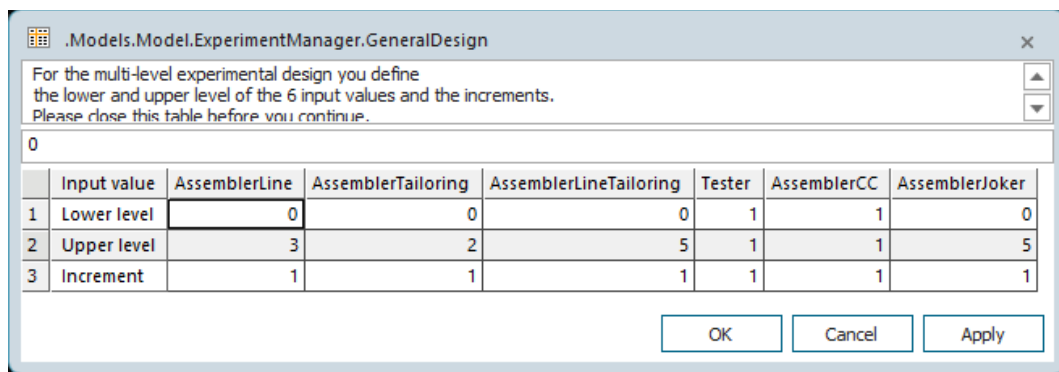
4.14 Viikko 10

4.14.1 3.6.2024

Mallin jalostus jatkui erilaisissa simulaatioajoissa havaittujen epäkohtien parissa. Tänä päivänä alettiin kokeilla ExperimentManagerin käyttöä eri operaattorijakaumien testaamiseen simulaatiomallissa. Tämä ei ollut mitenkään helppoa tai intuitiivista löytää, että kuinka tämä tehtäisiin, mutta lopulta löydettiin oikea toiminto (Multi-level Experimental Design) sekä syöttötaulukoon oikea syntaksi ja asiaa päästiin kokeilemaan. Syöttötaulukoon laitettiin minimi- ja maksimiarvoja kunkin operaattorityypin kohdalle, niin ohjelmisto teki näiden perusteella ”kokeita” (experiment), sellaisen määrän, että kaikki vaihtoehdot käydään läpi ja ajetaan peräjälkeen.

Tässä kohtaa, kun ajettiin ajoja erilaisilla operaattorijakaumilla, huomattiin, ettei kaikkia bugeja ollutkaan saatu vielä kiinni. Lisäksi ohjauskokonaisuuskokoonpanon prioriteettia piti nostaa korkeammaksi kuin kaappilinjan kokoonpanot, koska ohjauskokonaisuuteen kuuluva ACD on yksi viimeisen linjan kokoonpanon rakennuspalikoista. Aiemmin saatettiin joissain tapauksissa törmätä ongelmaan, että operaattori jäi odottamaan kaappilinjalle ikuisiksi ajoiksi ACD:tä, jos operaattorin olisi pitänyt tehdä itse ohjauskokonaisuus sekä kaappilinjan viimeinen kokoonpano.

Tähän ongelmaan liittyy myös se hölmö sisäänrakennettu ominaisuus ohjelmistossa, että operaattori saatetaan kutsua kokoonpanopisteelle, vaikka siellä ei olisi kaikkia siihen vaadittavia osia edes saatavilla. Kuvassa 27 on esitetty esimerkki multi-level experimental designin syöttötaulukosta.



For the multi-level experimental design you define the lower and upper level of the 6 input values and the increments. Please close this table before you continue.

0

| | Input value | AssemblerLine | AssemblerTailoring | AssemblerLineTailoring | Tester | AssemblerCC | AssemblerJoker |
|---|-------------|---------------|--------------------|------------------------|--------|-------------|----------------|
| 1 | Lower level | 0 | 0 | 0 | 1 | 1 | 0 |
| 2 | Upper level | 3 | 2 | 5 | 1 | 1 | 5 |
| 3 | Increment | 1 | 1 | 1 | 1 | 1 | 1 |

OK Cancel Apply

Kuva 27. Esimerkki multi-level experimental designin syöttötaulukosta

4.14.2 4.6.2024

Tässä jatkettiin kokeiluja ExperimentManagerilla. Edelleen löydettiin uusia bugeja kun kokeiltiin erilaisia operaattorijakaumia ja tilauskantoja. Tämä kertoi siitä että simulaatiomalli (ja tuotantokonsepti itsessään) on aika monimutkainen, ja siinä voidaan päätyä hyvin moniin erilaisiin tilanteisiin. Tästä syystä perusteellinen testaus juuri em. tavoilla on hyvin tärkeää, että mallin toimintavarmuus saadaan hyväksi.

Eräs oli sellainen, että jos operaattori sattui kantamaan osaa juuri silloin kun esimerkiksi tauko alkoi, niin päädyttiin virheeseen, sillä operaattori lopetti toimintansa, eikä tiennyt mitä tehdä osalle sen jälkeen. Samantapaiseen ongelmaan oli törmätty jo aiemminkin 21.5.2024. Tässä huomattiin, että operaattorille oli jäänyt virheellisesti määrittäminen käyttää vuorokalenteria, vaikka jo aiemmin oli todettu se huonoksi tavaksi hoitaa asia. Kun nämä huolimattomuusvirheet korjattiin, ongelma katosi. Päivä menikin kokonaan testailun ja bugien korjaamiseen.

4.14.3 5.6.2024

Erilaisten simulaatioajojen testailu jatkui ExperimentManagerilla. Lisäksi lisättiin toiminto sille, että laskettiin jokaisen ajon jälkeen keskimääräinen läpimenoaika valmistuneille tuotteille. Tämä oli suhteellisen helppo lisäys, kun yksittäiset ajat oli

jo kerätty omaan taulukkoonsa aiemmin. Tämä keskimääräinen aika otettiin myös yhdeksi näytettäväksi tulokseksi valmistuneiden tuotteiden lukumäärän lisäksi.

4.14.4 6.6.2024

Tässä kohtaa alettiin ajaa ns. kunnollisia tuloksia mallista. Suunnitelmana oli ajaa ensin ajot kaikilla realistisilla eri operaattorijakaumilla siten, että tilauskannassa on vain yhtä tuotetta. Kun nämä tulokset saadaan kerättyä, niin sen jälkeen ajettaisiin samoja ajoja satunnaisilla tilauskannoilla sen verran kun näyttää järkevältä.

Operaattorijakaumat määritettiin seuraavasti:

- Pelkästään kaappilinjalla toimivia 0-3 kpl
- Pelkästään tailoringissa ja pakkauksessa toimivia 0-2 kpl
- Kaappilinjalla, tailoringissa ja pakkauksessa toimivia 0-5 kpl
- Testaajia aina 1 kpl (vain yksi työpiste, ja ilman testaajaa tuotos 0)
- Pelkästään ohjauskokonaisuustyöpisteellä 0-1 kpl
- Jokereita 0-5 kpl (tekevät kaikkia työvaiheita testausta lukuunottamatta)

Maksimimäärät määräytyivät sen mukaan montako operaattoria maksimissaan mahtuisi kyseisiin toimintoihin.

Koska tuotannossa on seitsemän varsinaista eri työpistettä, sitä pidettiin myös maksimioperaattorimääränä, millä simulaatioajoja tehtiin. Edellämainituilla ehdoilla ExperimentManager loi myös kokeita, joissa oli enemmän operaattoreita, mutta ne karsittiin tarpeettomina pois ajettavien kokeiden joukosta.

Lisäksi osassa kokeita operaattorijakauma oli sellainen, ettei ajoissa valmistu koskaan yhtään valmista lopputuotetta, riippumatta siitä mitä tilauskannassa on. Tällaiset kokeet karsittiin myös pois, jotta saatiin selkeämmät ja siistimmät tulokset aikaiseksi. Toinen hyöty oli se, että ajoihin tarvittava aika lyhenyi.

Simulaatioajoissa käytettiin yhtä vuoroa aikavälillä 7:00-15:00. Taukoajoiksi määritettiin ajat 8:00-8:30, 10:30-11:30 ja 13:00-13:30. Taukoajat saattavat tuntua pitkiltä, mutta niihin otettiin hieman ylimääräistä ottamaan huomioon inhimillistä toimintaa. Lisäksi oltiin tietoisia siitä, ettei simulaatiomalli välttämättä näillä asetuksilla anna ensimmäisellä arvauksella erittäin tarkkoja tuloksia verraten tosielämään, mutta eri simulaatioajot olisivat silti keskenään vertailukelpoisia ja antaisivat silti suuruusluokaltaan oikeansuuntaisen teoreettisen tuloksen.

Ajojen tarkastelujaksoksi määritettiin kaksi viikkoa, joista tulokset kerättiin jälkimmäiseltä viikolta. Tällainen tapa valittiin sen vuoksi, että lopputuotteiden läpimenoajat ovat suhteellisen pitkiä, ja tuotanto tarvitsee aika paljon aikaa päästäkseen tyhjästä alusta kunnolla ns. normaaliin rytmiin. Siksi ajettiin tällainen viikon ”lämmittelyajo”, jotta saatiin tämä normaali rytmi tarkasteltavaksi.

Jäljelle jäi 205 koetta kutakin tilauskantaa kohden. Kerättäviä tuloksia varten tehtiin exceliin pohja, mihin kerättiin eri tilauskantakohtaiset ajot analysointia varten. Tämän päivän aikana kerittiin ajaa kaikille lopputuotetyypeille omat ajonsa, joissa tilauskannassa on vain kutakin yhtä tyyppiä.

4.14.5 7.6.2024

Tämän päivän aikana jatkettiin simulaatioajojen ajoa tuloksia varten satunnaisilla tilausjakaumilla. Niitä tehdessä huomattiin muutamia asioita, mitkä piti ottaa erityisesti huomioon verrattuna yhden tuotteen ajoihin.

Satunnaisissa ajoissa huomattiin, että yhden viikon tulosjakso on liian lyhyt, eikä sillä saatu riittävän tarkkoja tuloksia. Hajontaa tuli aivan liikaa satunnaisuudesta johtuen näin lyhyellä jaksolla, joten näitä ajoja varten otettiin yhteensä viiden viikon ajot siten, että viimeisten neljän viikon tulokset kirjattiin ylös. Tällöin saavutettiin johdonmukaisia tuloksia. Lisäksi nämä tulokset skaalattiin kuitenkin

samaan mittakaavaan aikaisempien tulosten kanssa (jako 4:llä), jotta niitä oli helpompi vertailla keskenään.

Satunnaisia ajoja tehtiin neljä kappaletta, jonka jälkeen huomattiin, ettei niissä ollut juurikaan eroa keskenään. Tässä kohtaa todettiin ettei liene syytä ajaa niitä enää sen enempää siitä syystä.

Tässä kohtaa saatiin työ valmiiksi simulaatiomallin kanssa tätä opinnäytetyötä varten. Simulaatioajojen tulokset ja niiden analysointi esitetään seuraavassa kappaleessa.

5 TULOSTEN ANALYSOINTI

5.1 Simulaatioajojen muuttujat

Tuotekannat, joille ajo tehtiin:

- Pelkästään
 - A10C10
 - A10C10+IO
 - A11C11
 - A11C11+IO
 - 2A102C10
 - 2A102C10+IO
 - 2A112C11
 - 2A112C11+IO
- 4 x satunnainen kanta kaikilla tuotteilla

Ajat:

- Tulokset viikon ajalta viikon lämmittelyajon jälkeen
- Käytössä 1 vuoro, 6 h tehokas työaika
- Kellotetut vaiheajat

Operaattorien määrä:

- 2-7 operaattoria seuraavilla rooleilla:
 - Pelkkä kaapituslinja
 - Pelkkä viimeistely ja pakkaus
 - Kaapitus, viimeistely ja pakkaus
 - Testaaja (aina 1)
 - Pelkkä ohjauskokonaisuus
 - Kaikki paitsi testaus
- 205 tehtäväjakoyhdistelmää

5.2 Tulokset

Yleisesti ottaen tärkein tulos simuloinnista oli se, kuinka kyvykäs tuotantokonsepti maksimissaan on. Toisin sanoen kuinka paljon sillä pystytään tuottamaan eri tuotteita erilaisilla tilauskannoilla. Nämä tulokset on esitetty taulukossa 1, keltainen osio. Tässä huomattavaa on se, että teoreettista maksimimäärää (7 kpl) operaattoreita ei tarvita missään tapauksessa maksimituotoksen saavuttamiseen, vaan siihen päästään aina joko viidellä tai kuudella operaattorilla. Tämä kertoo siitä, että prosessissa on epätasapainoa ja pullonkaula tai useampia, jotka aiheuttavat sen että isommassa määrässä operaattoreita joku vain joutuu odottelemaan edellisten vaiheiden valmistumista.

Operaattorien osalta kiinnostavia tarkasteltavia kohteita ovat luonnollisesti kokonaistuotos, mitä valmiita tuotteita saadaan aikaan, mutta erityisesti myös se että millä määrällä työtä saadaan tehtyä tehokkaimmin. Tätä mitattiin tarkastelemalla valmiiden tuotteiden määrää operaattoria kohden. Nämä tulokset on esitetty taulukossa 1, vihreä osio. Tässä huomattavaa on se, että yleensä paras tehokkuus eri tilanteissa saavutetaan silloin, kun operaattoreita on yksi vähemmän kuin millä maksimituotos saavutetaan. Tämä kertoo myös epätasapainosta prosessista, että myös maksimituotosten tapauksissa esiintyy odottelua. Toki erot tehokkuuksissa näiden operaattorimäärien välillä ovat aika pieniä, mikä antaa viitteitä siitä että nämä epätasapainot eivät kuitenkaan ole kovin suuria.

On myös käytännössä erittäin vaikeaa saada tällaisesta monivaiheisesta prosessista täysin tasapainoista siten, että jokainen vaihe olisi samanpituisen. Tämän valossa tuloksia prosessin tasapainoisuudesta voidaan pitää ihan hyvinä, ottaen huomioon myös sen että tuotannossa ollaan vasta ihan alkuvaiheessa.

Yleisesti ottaen mallin perusteella voitaisiin väittää, että viisi operaattoria vuorossa olisi ihanteellinen määrä tällaisen tuotantokonseptin pyörittämiseen. Se on eräänlainen kompromissi parhaimman tuottavuuden ja tehokkuuden välillä.

Monissa tapauksissa paras tehokkuus saavutetaan yhtä vähemmällä, mutta tätä puoltaisi myös se että silloin tuotannossa olisi hieman enemmän pelivaraa poikkeustilanteita varten.

Taulukosta nähdään, että puolessa tapauksista 5:llä operaattorilla saavutetaan myös maksimituotanto, ja toisessa puolessa tapauksista saavutetaan parempi tehokkuus, vaikka tuotos ei olekaan ihan niin suuri kuin se voisi olla.

Viiden operaattorin mallia tarkasteltiin vielä tarkemmin asentajien työnjaon osalta, että mikä roolitus antaa parhaat tulokset. Kaikki yhdistelmät mitä simuloitiin, listattiin ja tarkasteltiin kullakin tuotekannalla parhaan tuotoksen antava roolitus. Nämä tulokset on esitetty alla taulukossa 1:

Taulukko 1. Operaattorien roolitus vs. tuotos, 5 operaattoria

[illegible]

Näistä teoreettisesti parhaimpana yhdistelmänä voitaisiin pitää vihreällä merkattua roolitusta, jolla saavutetaan kaikki 12 simuloitua tuotekantaa huomioon ottaen paras tuotos yhdeksässä niistä. Jos taas huomioidaan pelkästään satunnaiset tuotekannat (mitä voidaan kuitenkin pitää ehkä lähempänä todellista tuotantotilannetta), paras tuotos tulee kaikissa neljässä. Keltaisella on merkattu taulukkoon muita yhdistelmiä, jotka ovat melkein yhtä hyviä. Ne kaikki voisivat olla testaamisen arvoisia todellisessa elämässä; ehkä jokin niistä toimisikin oikeasti paremmin kuin teoreettisessa mallissa.

Prosessin pullonkauloja arvioitiin simulaatio-ohjelmiston BottleNeckAnalyzer-työkalulla. Työkalun analyysi perustuu siihen, että se katsoo kuinka suuren prosenttiosuuden ajasta työpisteellä työskennellään (working), odotetaan (waiting) tai että se on tukossa (blocked). Viimeinen tarkoittaa siis sitä, että työpisteellä on jokin osa/kokoonpano, mutta sitä ei voida viedä seuraavalle vaiheelle, koska se on varattu. Mukana on myös pause-status, joka tarkoittaa tässä simulaatiossa aikaa asetetun työajan ulkopuolella, mutta se on kaikille vaiheille sama, joten sitä ei tarvitse huomioida. Tämä aika jätetään myös huomioimatta seuraavissa prosenttiluvuissa mitä esitetään, koska oleellisempaa on esittää osuuksia työajasta kokonaisajan sijaan.

Työkalu antaa oletusarvoisesti korkeimmat pisteet vaiheille, joissa työskennellään eniten. Tässä simulaatiossa työkalun mukaan pahin pullonkaula olisi kaapituslinjan kolmas vaihe, jossa käytännössä työskennellään jatkuvasti, eli 99,6 % työajasta. Lisäksi tätä edeltävä kaapituslinjan toinen vaihe on tukossa 41,5 % ja seuraava työvaihe Tailoring odottaa 38,1 % työajasta. Näiden tulosten valossa lienee selvää, että tällä työvaiheella on liikaa töitä, joita pitäisi yrittää jakaa jonnekin muualle.

Toiseksi korkeimmat pisteet saa systeemimoduulipuolen ns. late dedication -vaihe, jossa moduulit ja niiden liitäntäosiöt varioidaan puolivalmisteista lopullisen konfiguraation mukaisiksi. (90,7 % työajasta) Tässäkin seuraava vaihe systeemi-moduulien konfigurointi odottaa varsin paljon, eli 37,8 % työajasta. Tämä vaihe olisi ehkä syytä tuplata tämän ruuhkaantumisen vuoksi.

Kolmanneksi korkeimmat pisteet saa kaapituslinjan ensimmäinen vaihe, jossa työskentelyaika on kuitenkin jo paljon pienempi. (68,7 %) Loppuaika ollaankin sitten melkein kokonaan tukossa oletettavasti linjan kolmannen vaiheen aiheuttaman pullonkaulan vuoksi.

5.3 Johtopäätökset

Simulaatiomallin perusteella voisi siis väittää, että sellaisenaan viisi operaattoria vuorossa voisi olla hyvä määrä tuotannon pyörittämiseen. Kapasiteetin lisäämiseen ensisijainen keino olisi lisätä sitten vuoroja, jolloin esimerkiksi teoriassa iltavuoron lisääminen kaksinkertaistaisi tuotannon kokonaismäärän.

Asiaa voisi kuitenkin tarkastella vielä tarkemminkin sen osalta, jos haluttaisiin aina saavuttaa maksimituotanto lisäämällä vielä yksi operaattori vuoroon, vaikka silloin työskenneltäisiin linjan kannalta huonommalla yksilökohtaisella tuottavuudella (enemmän odottelua). Tämän mielekkyyttä voisi arvioida ottamalla huomioon ja laskemalla ”ylimääräisen” operaattorin aiheuttama kustannus vs. saavutetun lisätuotoksen arvo. Lisäksi asiaa voitaisiin ajatella paremman tuotantovarmuuden kannalta siten, että mahdolliset poissaolot eivät vaikuttaisi tuotantomääriin ihan niin paljon.

Pullonkaula-analyysit näyttivät sen, että tuotantomallissa on epätasapainoisuuksia, joita korjaamalla voitaisiin saada tuotannon virtausta parannettua ehkä paljonkin. Työkalun eri vaiheille antamien odotus- ja tukkoisuusaikojen perusteella voisi alkaa optimoimaan kokonaisprosessia. Yksinkertaistettuna periaatteena voisi ajatella, että jos jollain vaiheella on odotusaikaa, se tarkoittaa että edeltävällä vaiheella on enemmän työtä tehtävänä mitä pitäisi siirtää eteenpäin. Jos taas vaiheella on tukkoisuusaikaa, silloin seuraavalla vaiheella on enemmän työtä. Käytännössä työn siirtäminen vaiheelta toiselle ei kuitenkaan aina ole ihan niin helppoa, mutta työkalun antamat tulokset antavat silti hyvän ohjenuoran siitä mihin suuntaan kannattaisi alkaa kehittää. Koska tässä tuotantomallissa on kyse aika monimutkaisesta kokonaisuudesta, jokaisen muutoksen vaikutus kokonaisuuteen pitäisi tarkistaa, koska muutoksilla voi olla vaikutusta myös sellaisiin asioihin mitä ei osattu ennakoida.

5.4 Jatkosuunnitelmat

Jotta mallin antamiin tuloksiin voitaisiin luottaa, pitäisi sen todenmukaisuus varmistaa vertaamalla sitä todelliseen tuotantoon samoilla lähtökohdilla, paramtereilla ja resursseilla. Vaikka mallin rakentamisessa oltaisiin onnistuttu hyvin, niin todennäköisesti tässä vaiheessa tarvitaan useampi iterointikierros säätöineen ja mahdollisine muutoksineen, ennen kuin päästään lähelle todellista tilannetta. Todellinen tuotanto ei ole vielä kuitenkaan tarpeeksi pitkällä, että tätä vaihetta voitaisiin vielä mielekkäästi toteuttaa.

Toiveena on, että malli saataisiin varmistettua hyvällä tarkkuudella, jolloin sitä voitaisiin parhaassa tapauksessa käyttää myös tuotannonsuunnittelun apuvälineenä. Malliin voi syöttää esimerkiksi tulevan tilauskannan, käytettävissä olevat resurssit ja työajat/vuorot mitä noudatetaan, niin se antaa tuloksena kullekin päätuotteelle odotetun valmistumisajankohdan. Tällainen olisi arvokasta tietoa kirjavan tilauskannan osalta, missä perinteisillä keinoilla saman arvioiminen olisi hankalaa.

Muita jatkosuunnitelmia ovat tulevien suunniteltujen investointien vaikutusten arviointi suorituskykyyn etukäteen ennen kuin sijoitetaan suuria summia rahaa. Tällaisia suunnitelmia ovat esimerkiksi kaapituslinjan muuttaminen 4-vaiheiseksi, toisen kokonaisen kaapituslinjan lisäys kokonaisuuteen ja testauskapasiteetin kasvatus. On vaikea arvioida tällaisten muutosten kokonaisvaikutusta tuotannon toimintaan ilman simulointia, mutta malliin näiden asioiden lisääminen ei maksa muuta kuin vähän aikaa ja vaivaa. Tällaisen analyysin jälkeen, varsinkin jos malli toiminta on varmistettu, on huomattavasti helpompi arvioida investointien mielekkyyttä. Samalla kun kokonaisprosessi muuttuu, pullonkaulat todennäköisesti muuttuvat myös. Simulointimalli näyttäisi myös nämä uudet ongelmakohdat samalla vaivalla, joita kehittämiseen/poistamiseen voitaisiin sitten alkaa jo miettiä keinoja.

6 TYÖN ARVIOINTI

Yleisesti ajatellen arvioisin että simulointimallin tekeminen onnistui hyvin, vaikka hankaluuksia matkan varrella oli kyllä runsaasti. Se oli kylläkin ihan odotettavaa, kun yritetään rakentaa suhteellisen monimutkaista tuotantomallia ensimmäisellä yrityksellä uuden ohjelmiston kanssa. Oman kokemukseni perusteella arvioisin myös, että mallin antamat tulokset ovat oikeansuuntaisia, mutta hienosäätöä pitää varmasti myös tehdä mallin varmistamisen yhteydessä. Olisi liian optimistista ajatella, että kaikki loksautaisi itsestään paikalleen. Olin tyytyväinen siihen, että oikeastaan kaikki asiat mallissa sai simuloitua tavalla, joka on lähellä todellisuutta, eikä tarvinnut ns. oikeaa mutkissa juurikaan.

Simulaatiomallin rakentamisen ajallinen panostus on aika suuri. Tässä tapauksessa kävin koulutusta 12 sessiota, yhteensä noin 16 tuntia. Itse mallin rakentamiseen harjoitteluineen meni noin 70-80 tuntia, eli yhteensä voisi ajatella panostuksen olleen noin kolme viikkoa tehokasta työaikaa. Tällaisella panostuksella voidaan kuitenkin saada aikaiseksi työkalu, mitä ei ollut aiemmin käytettävissä tuotannon suorituskyvyn arviointiin, tuotannon suunnitteluun ja investointien mielekkyyden realistiseen arviointiin. Sanoisin, että tällainen aika kannattaa käyttää monimutkaisten prosessien mallintamiseen, mitä ei pysty kunnolla esimerkiksi excel-taulukon avulla arvioimaan. Esimerkiksi jonkin kalliin investoinnin osalta malli voi maksaa itsensä helposti takaisin. Tietenkin aikaa pitää vielä käyttää lisää, että saadaan aiemmin mainitut jatkosuunnitelmat tehtyä, mutta sanoisin että se on silti sen arvoista.

Mallin rakentamisessa opin paljon Plant Simulation -ohjelmiston käytöstä, ja uskoisin että seuraavien mallien kanssa aikaa ei mene suhteessa läheskään niin paljon enää. Tämä tietysti riippuu paljon uusien juttujen luonteesta, että kuinka helposta tai vaikeasta tapauksesta on kyse simuloinnin kannalta, ja että pitääkö opetella paljon uusia juttuja vai meneekö pääasiassa jo opituilla keinoilla.

LÄHTEET

- /1/ Danfoss yritysesittely, maaliskuu 2024. Viitattu 5.8.2024.
- /2/ Danfoss Vaasan tehtaan esittely, helmikuu 2023. Viitattu 5.8.2024.
- /3/ Mikä on taajuusmuuttaja? (Danfoss.com) Viitattu 5.8.2024.
<https://www.danfoss.com/fi-fi/about-danfoss/our-businesses/drives/what-is-a-variable-frequency-drive/>
- /4/ Jerry Banks, Handbook of Simulation: principles, methodology, advances, applications, and practice, 1998. Viitattu 8.8.2024.
- /5/ Simulation & Modeling Team, IV. Introduction to Modeling and Simulation Systems, A. Historical Perspective. (University of Houston) Viitattu 6.8.2024.
<https://uh.edu/~lcr3600/simulation/contents.html>
- /6/ Post-war reconstruction and development in the Golden Age of Capitalism, World economic and social survey 2017. Viitattu 8.8.2024.
https://www.un.org/development/desa/dpad/wp-content/uploads/sites/45/WESS_2017_ch2.pdf
- /7/ Visual Components, 7 reasons simulation software can help you with manufacturing planning. Viitattu 9.8.2024.
<https://www.visualcomponents.com/blog/7-reasons-simulation-software-can-help-you-with-manufacturing-planning/>
- /8/ Disadvantages of Simulation. Viitattu 13.8.2024.
<https://www.javatpoint.com/advantages-and-disadvantages-of-simulation>
- /9/ Rachael Pasini, The present and future state of 3D simulation in engineering, 18.1.2024. Viitattu 15.8.2024.
<https://www.3dcadworld.com/the-present-and-future-state-of-3d-simulation-in-engineering/>
- /10/ Plant simulation software – Siemens software. Viitattu 11.6.2024.
<https://plm.sw.siemens.com/en-US/tecnomatix/products/plant-simulation-software/>