

Jukka Mattila

**Tuotannonohjauksen tehostaminen simulaation avulla**

Opinnäytetyö

Kevät 2011

Tekniikan yksikkö

Automaatiotekniikka

Koneautomaatio



## SEINÄJOEN AMMATTIKORKEAKOULU

### OPINNÄYTETYÖN TIIVISTELMÄ

Koulutusyksikkö: Tekniikan yksikkö  
Koulutusohjelma: Automaatiotekniikka  
Suuntautumisvaihtoehto: Koneautomaatio

Tekijä: Jukka Mattila

Työn nimi: Tuotannonohjauksen tehostaminen simulaation avulla

Ohjaaja: Jorma Mettälä

Vuosi: 2011 Sivumäärä: 56 Liitteiden lukumäärä: 2

---

Tämä opinnäytetyön kohdeyrityksenä oli Skaala Ikkunat ja Ovet Oy. Työn tavoitteena oli simuloida tehtaan nykytilanteen kautta tulevaisuuden tarpeet materiaalivirtausten ja tuotannonohjauksen suhteen.

Simulaatio rakennettiin Visual Componentsin 3DCreaten avulla. Ensiksi tuotantotilojen layout, koneet, työkalut ja kuljetinhihnat mallinnettiin 3DCreaten mallinnustyökalun avulla. Layoutin mallintamisen jälkeen malliin lisättiin materiaalivirrat ja niitä hallitsevat säännöt. Lopuksi malliin sovellettiin ennalta hankitut läpimenoajat.

Simulaatiomallin avulla tuotannon materiaalivirrasta ja tuotantokapasiteetista saatiin paljon hyödyllistä dataa, jota voidaan käyttää tulevaisuudessa tuotannon suunnittelussa.

Asiasanat: Tuotannonohjaus, materiaalivirta, simulointi

## SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

### **Thesis abstract**

Faculty: School of Technology  
Degree programme: Automation Engineering  
Specialisation: Machine automation

Author: Jukka Mattila

Title of the thesis: Improving Manufacturing Efficiency With a Simulation Model

Supervisor: Jorma Mettälä

Year: 2011                      Number of pages: 56      Number of appendices: 2

---

This final thesis was commissioned by Seinäjoki University of Applied Sciences and Skaala windows & doors. The purpose was to simulate the present state and future needs of material flow and manufacturing efficiency.

The simulation was made with Visual Components 3DCreate. First, the layout of the factory was modelled with 3DCreate modelling tools. All machines, conveyors and tools were modelled. After the layout modelling, the material flow routes and rules were modelled. Finally, valid process times were implemented.

As a result of the simulation, a lot of useful information was received on the material flow and the production capacity. The simulated data can be used in future process planning.

Keywords: Manufacturing execution system, material flow, simulation

## SISÄLLYSLUETTELO

<b>SISÄLLYSLUETTELO .....</b>	<b>5</b>
<b>KÄYTETYT TERMIT JA LYHENTEET .....</b>	<b>7</b>
<b>KUVIO- JA TAULUKKOLUETTELO .....</b>	<b>8</b>
<b>1 JOHDANTO .....</b>	<b>11</b>
1.1 Työn tausta .....	11
1.2 Työn tavoite .....	11
1.3 Työn rakenne .....	12
1.4 Kohdeyritys .....	12
<b>2 TOIMINNANOHJAUS .....</b>	<b>14</b>
2.1 Valmistussuunnitelma .....	15
2.2 Kapasiteetti .....	16
2.3 Läpäisy aika .....	17
2.4 Tuotteiston ja layoutin yhteensovittaminen .....	17
2.5 Tuotantomuoto .....	18
2.5.1 Massatuotanto .....	19
2.5.2 Just in time –tuotanto .....	19
2.5.3 Lean Management .....	20
<b>3 SIMULOINTI .....</b>	<b>21</b>
3.1 Simuloinnin työvaiheet .....	21
3.2 Simulointityökalut .....	23
3.2.1 Delmia V5 .....	23
3.2.2 Flexsim .....	24
3.2.3 Visual Components 3DCreate .....	24
<b>4 TYÖVAIHE .....</b>	<b>25</b>
4.1 Taustatietojen keräys .....	25
4.2 Simulaatioympäristön kehittäminen .....	25
4.3 3DCreate .....	26
4.3.1 Komponentti- sekä layoutkirjasto .....	27

4.3.2	Parametrisuus.....	28
4.3.3	Geometrian mallintaminen .....	28
4.3.4	Sääntöjen luonti .....	29
4.3.5	Parametrien luonti.....	31
4.3.6	Robotin ohjelmointityökalu .....	32
4.4	Simulaatiomallin työvaiheet.....	33
4.4.1	Työpisteen mallintaminen .....	33
4.4.2	Tuotemallintaminen.....	38
4.5	Ympäristön rakentaminen .....	39
4.6	Sääntöjen ja kinematiikan luominen 3DCreatella .....	45
4.6.1	Esimerkki 1: Ovilehden maalaus.....	45
4.6.2	Esimerkki 2: Ovilehden lasitus. ....	46
4.6.3	Esimerkki 3: Alikokoonpanojen yhdistäminen .....	46
<b>5</b>	<b>TULOKSET.....</b>	<b>48</b>
<b>6</b>	<b>POHDINTA .....</b>	<b>51</b>
6.1	Työn aihe .....	51
6.2	Simulaatio .....	51
6.3	Tulokset .....	52
<b>7</b>	<b>LÄHTEET .....</b>	<b>53</b>
<b>8</b>	<b>LIITTEET .....</b>	<b>54</b>

## KÄYTETYT TERMIT JA LYHENTEET

**Component creator** Component creator on 3DCreaten toiminto joka on materiaalin lähtövarastona materiaalivirroille.

**Component container** Component container on 3DCreaten toiminto joka voidaan määrittellä komponentille. Simulaatiossa voidaan tähän Component containeriin liittää muita komponentteja.

**Frame** Ohjauspiste simulaatiomallissa esimerkiksi materiaalivirroille.

**Statistics** 3DCreaten statistiikoille luotu osio, jonka avulla saadaan simulaatiosta helposti kirjallista dataa esimerkiksi Excel-  
taulukkaan.

(3DCreate 2009.)

## KUVIO- JA TAULUKKOLUETTELO

Kuvio 1. Kohteen lähtötilanne .....	11
Kuvio 2. Kohteen tavoite .....	12
Kuvio 3. Toiminnanohjauksen tavoitteiden muodostuminen .....	14
Kuvio 4. Tuotteen läpäisyajan rakenne .....	17
Kuvio 5. Tuotantomuodot .....	18
Kuvio 6. JOT-toiminta kokonaisuutena .....	20
Kuvio 7. Lean management keskittyy tuottamattomien vaiheiden eliminointiin .....	20
Kuvio 8. Tehdassimuloinnin työvaiheet .....	23
Kuvio 9. 3DCreaten komponentti- ja layoutkirjasto .....	27
Kuvio 10. 3DCreaten parametrienhallintavälilehti .....	28
Kuvio 11. 3DCreaten mallintamistyökalu .....	29
Kuvio 12. 3DCreaten Behaviour-välilehti .....	30
Kuvio 13. 3DCreaten parametrien luonnin välilehti .....	31
Kuvio 14. 3DCreaten robotin opettamistyökalu .....	32
Kuvio 15. 3DCreatella mallinnettu työpiste .....	33

Kuvio 16. Blockin luonti.....	34
Kuvio 17. Jalan kloonaus .....	34
Kuvio 18. Kloonauksen parametrit .....	35
Kuvio 19. Jalka kloonattuna .....	35
Kuvio 20. Toisen kloonauksen parametrit .....	35
Kuvio 21. Kloonataan tanko .....	36
Kuvio 22. Sylinterin luonti.....	36
Kuvio 23. Sylinterin kääntö .....	37
Kuvio 24. Sylinterin kloonaus.....	37
Kuvio 25. Piirteiden nosto .....	38
Kuvio 26. Framet.....	38
Kuvio 27. 3DCreatella mallinnettu ovilehti.....	39
Kuvio 28. 3DCreatella mallinnettu layout .....	40
Kuvio 29. Maailman luonti kirjastojen avulla .....	41
Kuvio 30. Reitti komponentin luonti.....	41
Kuvio 31. Reitin määrittely .....	42
Kuvio 32. Reitin linkitys.....	42

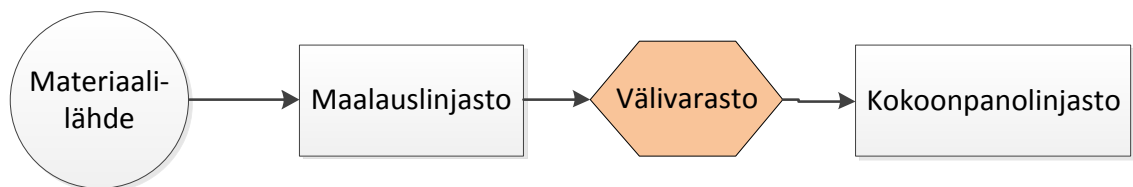


Kuvio 33. Prosessin parametointi .....	43
Kuvio 34. Kappaleen luonnin määrittely .....	43
Kuvio 35. Määritellään reitille sensori.....	44
Kuvio 36. Simulaation ajaminen.....	44
Kuvio 37. 3DCreatella mallinnettu maalaus piste.....	45
Kuvio 38. 3DCreatella mallinnettu lasitus piste .....	46
Kuvio 39. 3DCreatella mallinnettu kokoonpano piste.....	47
Kuvio 40. Kahden eri tuotteen materiaalivirran yhdistyminen .....	48
Kuvio 41. materiaalivirtojen yhdistämisen toteutuminen käytännössä .....	49
Kuvio 42. Kokoonpanolinjaston viimeinen piste .....	50
Kuvio 43. Lopullinen ratkaisu simulaatiossa .....	50

# 1 JOHDANTO

## 1.1 Työn tausta

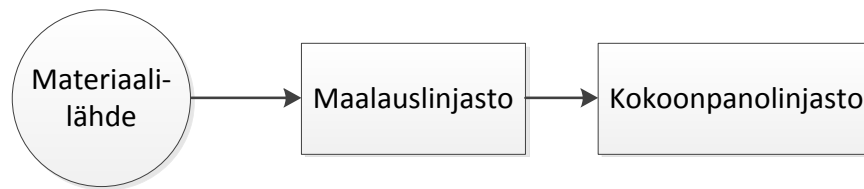
Työn taustana on ovitehtaan tuotannon tehostaminen tarpeettomia vaiheita karsimalla. Tällä hetkellä ovitehtaan tuotannossa on maalauslinjaston jälkeen välivarasto, joka sijaitsee kokoonpanolinjaston vieressä lattialla. Maalauslinjastolta tulleet materiaalit nostetaan välivarastoon, josta ne tarpeen vaatiessa nostetaan kokoonpanolinjastolle. Välivarasto vaatii ylimääräistä tilaa tuotantotiloista, teettää ylimääräistä työtä materiaalin nostelun vuoksi, sekä altistaa materiaalit kolhuille ollessaan suojaamattomana välivarastoalueella. Materiaalivirta on alun perin suunniteltu kulkemaan suoraan maalauslinjastolta kokoonpanolinjastolle, mutta käytännössä tämä on ollut mahdotonta.



Kuvio 1. Kohteen lähtötilanne

## 1.2 Työn tavoite

Skaalan tilausohjautuva JIT-mallinen tuotanto sekä LEAN-ajattelumalli tukevat ajatusta siitä, että ylimääräisiä välivarastoja tulisi tuotannon tehokkuuden ja välitömyyden takia välttää. Tämän takia työ keskittyy maalauslinjaston ja kokoonpanolinjaston väliseen välivarastoon. Työn tavoitteena on tutkia, millä keinoilla välivarasto voidaan poistaa käytöstä.



Kuvio 2. Kohteen tavoite

Tutkimiseen käytetään simulointimallia, jonka rakennetaan Visual Componentsin 3DCreaten avulla. Tavoitteena on saada materiaalivirta kulkemaan suoraan maalauslinjastolta kokoonpanolinjastolle. Simulaatiomallissa hyödynnetään tuotannosta jo ennalta kerättyä tietoa kuten läpimenoaikoja.

### 1.3 Työn rakenne

Työssä johdannon jälkeen kerrotaan tuotantoon liittyvästä teoriasta. Teoriaosuudessa käsitellään toiminnanohjausta ja toiminnanohjaukseen liittyen teoriaa valmistussuunnitelmasta, tuotantokapasiteetista ja läpäisyajasta. Teoriaosuudessa käsitellään myös tuotantomuotojen teoriaa, jossa tuotantomuodon lisäksi esiin tuodaan myös massatuotannon, Lean managementin ja Just in time–tuotannon perusteita. Teoriaosuuden jälkeen luvussa 4 kerrotaan itse työvaiheesta, työkalusta sekä simulaation rakentamisesta 3DCreatella. Kappaleessa käsitellään myös työpisteen mallintamista, tuotemallintamista ja näiden kautta ympäristön mallintamista. Lopuksi luvussa 4 kerrotaan kolmen esimerkin avulla, miten 3DCreatella luodaan materiaalivirroille säännöt, sekä kinematiikat. Luvussa 5 esitellään simulaatiomallin kautta saadut tulokset ja ratkaisut. Lopuksi viidennessä luvussa pohditaan työn onnistuvuutta, Visual Componentsin 3Dcreate ohjelman sopivuutta tähän työhön, simulaatiomallin oikeellisuutta, siitä saatuja tuloksia ja tulosten pohjalta tehtyjen ratkaisujen toimivuutta.

### 1.4 Kohdeyritys

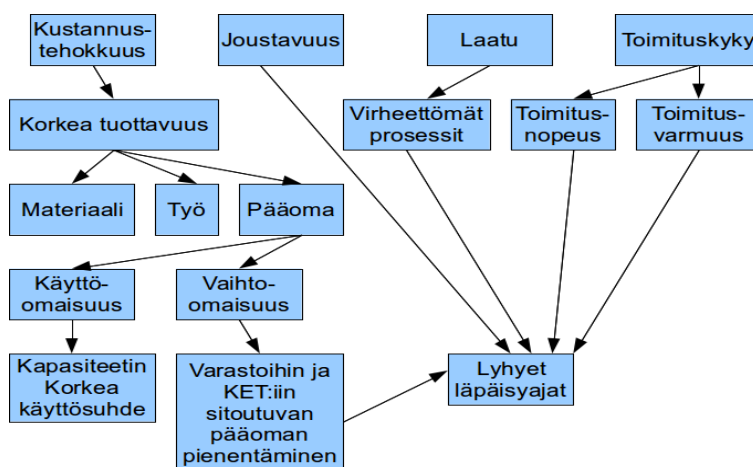
Skaala Ikkunat ja Ovet Oy perustettiin vuonna 1956, ja se on ikkunoiden ja ovien valmistukseen ja markkinointiin keskittynyt yritys. Skaala-konserni on yksi alan

suurimmista valmistajista Pohjoismaissa ja se työllistää noin 350 henkilöä. Suomen lisäksi Skaalan vientiin kuuluu myös Ruotsi, Iso-Britannia ja Venäjä. Ruotsissa ja Iso-Britanniassa toiminta on järjestetty omien tytäryhtiöiden kautta. Skaala tunnetaan energiatehokkaiden ikkuna- ja oviratkaisujen edelläkävijänä. (Skaala. [Viitattu 13.2.2011].)

## 2 TOIMINNANOHJAUS

Toiminnanohjaus-käsitettä käytetään nykyään paljon tuotannonohjaus-käsitteen sijaan, koska tuotannonohjaukseen on nykyään huomioitu konkreettisen tuotannon lisäksi myynti, jakelu, tuotesuunnittelu ja hankinta. Tuotteiden valmistuksen suunnitteluun ja ohjaukseen käytetään nykyään termiä valmistuksenohjaus. (Haverila, Uusi-Rauva, Kouri & Miettinen 2005, 397.)

Toiminnanohjauksen tavoitteet ovat kapasiteetin korkea tuottavuus, toimintaan sitoutuneen vaihto-omaisuuden minimointi, toimitusvarmuus sekä lyhyt läpäisy aika. Kapasiteetin korkealla tuottavuudella tarkoitetaan sitä, että tuotantoon sitoutuneilla koneilla ja tuotantotiloilla on hyötysuhde silloin mahdollisimman suuri, kun tuotanto on suunniteltu siten että resurssit näillä hankinnoilla tulee mahdollisimman tehokkaasti käytettyä. Vaihto-omaisuuden minimoinnilla pyritään valmistusta ja materiaalitoimintoja ohjaamaan siten, että raaka-aineet, keskeneräinen työ ja lopputuotanto sitoo mahdollisimman vähän pääomaa. Toimitusvarmuudella tarkoitetaan yksinkertaisesti sitä, että yritys pitää sovituista toimitusajoista kiinni ja toimittaa tuotteet asiakkaan tarpeen mukaisesti. Lyhyellä läpäisyajalla pyritään pitämään tilausten läpimenoajat mahdollisimman lyhyinä. Tuotteiden lyhyet läpimenoajat vähentävät keskeneräiseen tuotantoon sitoutunutta pääomaa, helpottavat kapasiteetin suunnittelua sekä kehittävät laatua ja toimitusvarmuutta. (Haverila, Uusi-Rauva, Kouri & Miettinen 2005, 402.)



Kuvio 3. Toiminnanohjauksen tavoitteiden muodostuminen (Haverila, Uusi-Rauva, Kouri & Miettinen 2005, 403.)

## 2.1 Valmistussuunnitelma

Valmistussuunnitelman avulla voidaan koordinoita kaikkien tuotantopisteiden oikea-aikaisen tuotteen valmistus. Valmistussuunnitelma pitää sisällään eri vaiheissa tarvittavat materiaalit, aikavälin jolla kukin valmistusyksikkö tekee tehtävänsä, lopullisen tuotteen toimitusajankohdan, verkon sisäiset toimitusajat sekä tilauksen aiheuttamat kuormitukset yksiköittäin. (Lapinleimu 2001, 115.)

Valmistussuunnitelman muoto on riippuvainen tuotteesta ja tuotantotyypistä. Selkeimpiä muotoja ovat:

- tilauksittainen tai projektittainen ajanjaksoihin sidottu Ganttin kaavio
- ajanjaksoille sijoitetut tilaukset tai listat siitä, mitä tuotetaan kunakin ajanjaksona
- kiinteäjärjestyksinen tilausjono. (Lapinleimu 2001, 115)

Taulukko 1. Jaksollinen valmistussuunnitelma (Lapinleimu 2001, 115.)

Tilaukset	Ohjausjaksot					
XX	B	C				
XY	A	B	C			
XZ		A	B	C		
YX		A		B	C	
YY				B	C	
YZ				A	B	C

Taulukossa 1, A, B ja C ovat työvaiheita, esimerkiksi levytyöt, koneistus ja kokoonpano, jotka samalla vastaavat valmistusyksiköitä. Kaikki yksiköt, myös osavalmistusyksiköt, voivat lukea oman aikataulunsa valmistussuunnitelmasta. (Lapinleimu 2001, 115.)

## 2.2 Kapasiteetti

Kapasiteetti on tuotantokyvyn mittari, jolla ilmoitetaan tuotantoyksikön enimmäissuorituskyky aikayksikössä. Mikäli tuotteiden kapasiteettivaatimukset poikkeavat vain vähän toisistaan, voidaan kapasiteetti ilmaista myös tuoteyksiköissä. (Haverila, Uusi-Rauva, Kouri & Miettinen 2005, 399.)

Toiminnanohjauksen karkeasuunnittelussa käytetään kuormitusryhmiä, jotka jaetaan laajoihin osiin kuten tuotantolinja, tuoteverstas, pullonkaulakone tai työntekijäryhmä. Tarkemmassa suunnitteluvaiheessa eli hienosuunnittelussa nämä karkeat kuormitusryhmät jaetaan paljon pienempiin kuormitusryhmiin. (Haverila, Uusi-Rauva, Kouri & Miettinen 2005, 399.)

Kapasiteettia hallitaan työpisteen kapasiteetin sekä suunniteltujen töiden kuormituksen kautta. Kuormitus kertoo kuinka paljon suunniteltu tuotanto varaa kapasiteettia. Kuormitussuhteella taas ilmaistaan tietyn ajanjakson suhteellisen kuormituksen käytettävissä olevaan maksimikapasiteettiin verrattuna. (Haverila, Uusi-Rauva, Kouri & Miettinen 2005, 400.)

$$\text{Kuormitussuhde} = (\text{Kuormitus} * 100\%) / \text{Kapasiteetti.} \quad (1)$$

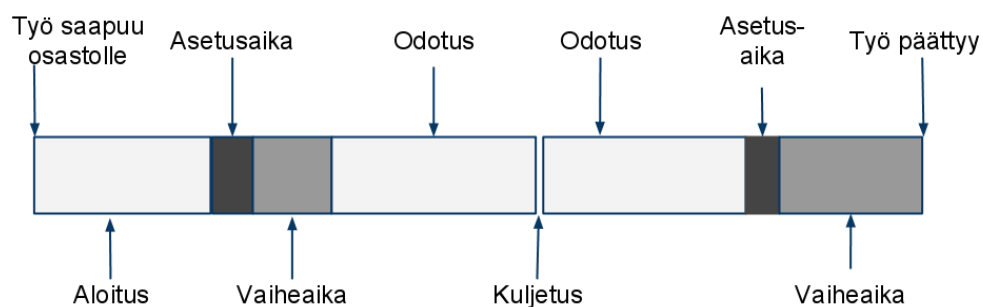
Kuormitusasteen ja kuormitussuhteen vaihtoehtona voidaan mitata suunnitellun kuormituksen sijaan toteutunutta tuotannon määrää ja suhdetta, jotka ilmoitetaan käyttöasteena ja käyttösuhteena. (Haverila, Uusi-Rauva, Kouri & Miettinen 2005, 400.)

$$\text{Toimintasuhde} = (\text{Tuotantomäärä} * 100\%) / \text{Kapasiteetti.} \quad (2)$$

Käytettävissä oleva kapasiteetti voi olla huomattavasti teoreettista maksimikapasiteettia pienempi, sillä erilaiset valmistusprosessin häiriöt, materiaalipuutteet, konerikot, huollot, työnjärjestelyjen puutteet, viallisten tuotteiden valmistukset, koulutukset, sairauslomat ja poissaolot pienentävät todellista kapasiteettia. Usein nettokapasiteetti on vain 50 – 90 % teoreettisesta maksimikapasiteetista. (Haverila, Uusi-Rauva, Kouri & Miettinen 2005, 400-401.)

### 2.3 Lämpäisy aika

Lämpäisyajalla kuvataan toimintaketjun vaatimaa kokonaisaikaa. Tuotteen lämpäisyajalla ilmaistaan tuotteen tilauksesta tuotteen toimittamiseen kulunut aika. Lämpäisy aika ei siis kerro tuotteen valmistukseen käytettyä aikaa, sillä suuri osa ajasta on usein odotusaikaa. Lämpäisy aika lasketaan kalenteriaikana ja se ei ota kantaa mitä tuotteelle tapahtuu lämpäisyajan aikana. Lämpäisy aika ei kuvaa tuottavuutta eikä tuotteen vaatimaa valmistusaikaa. (Haverila, Uusi-Rauva, Kouri & Miettinen 2005, 401.)



Kuvio 4. Tuotteen lämpäisyajan rakenne (Haverila, Uusi-Rauva, Kouri & Miettinen 2005, 401.)

### 2.4 Tuotteiston ja layoutin yhteensovittaminen

Kokonaisvaltaista JIT-toimintaa rakennettaessa tuotteisto, tehtaan layout, sekä asiakastarpeet on sovittava yhteensopiviksi. Tuotannon tulee olla suunniteltu kokonaisuus, jossa mukana myös tehtaan layout. Jos kaikkia tarvittavia osaluokkia ei kehitetä, niin tuotannosta tulee vajavainen. (Tiainen 1996, 63.)

Japanilaisessa tuotantofilosofiassa tuotteenmukainen layout on ehdottomana tavoitteena. Layoutin tuotteen mukaistamisella saavutetaan mahdollisimman pieni valmistukseen sitoutunut pääoma, lyhyet tarkat lämpäisyajat, pieni tilankäyttö, ihmistyön vähentäminen ja virtaustekniikalla saavutettu yksinkertainen ja visuaalisesti helposti seurattava toiminta. (Tiainen 1996, 75.)



## 2.5 Tuotantomuoto

Tuotantomuoto yrityksessä määräytyy tuotteen valmistusmäärien, konstruktion, valmistustekniikan ja jakelutien perusteella. Eri tuotantomuotojen tuotteet ovat peruseriaatteeltaan joko vakiotuotteita tai tilaustuotteita. Vakiotuotteita ovat suurin osa kulutustavarasta, kuten autot, tietokoneet, piirilevyt ja valmisvaatteet. Tilaustuotteisiin kuuluvat ainutkertaiset tuotteet kuten rakennukset, taideteokset, tuotantoautomaatiojärjestelmät sekä ovet. Tuotteen valmistusaloite taas määräytyy sen mukaan onko tuote asiakas- vai varasto-ohjautuva. (Haverila, Uusi-Rauva, Kouri & Miettinen 2005, 353.)



Kuvio 5. Tuotantomuodot (Haverila, Uusi-Rauva, Kouri & Miettinen 2005, 354.)

Varasto-ohjautuvaa tuotetta valmistetaan siten, että tuotetta olisi jatkuvasti saatavilla varastossa, josta se on nopea lähettää asiakkaalle tämän tilatessa tuotetta. Lähes kaikki kulutustavara on varasto-ohjautuvaa kuten esim. elintarvikkeet, vaatteet, elektroniikka ja muut kulutushyödykkeet. (Haverila, Uusi-Rauva, Kouri & Miettinen 2005, 353-354.) Suurina etuina ovat nopeat toimitukset sekä huoltojen määrän minimoituminen. Haittapuolina ovat varastojen jäykistävä ja tehottomuutta peittävä ominaisuus, sekä pääoman sitoutuminen tuottamattomaan tilaan eli varastointiin. (Lapinleimu 2001, 109.)

Asiakas- eli tilausohjausmallinen tuotteen valmistus määräytyy asiakkaan tilausten mukaan. Koska asiakas tilausvaiheessa määrittelee tuotteelle tietyt vaatimukset,

voidaan tuote valmistaa valmiiksi vasta tilauksen saavuttua. Tuotteen valmistus voidaan aloittaa jo ennen tilausta niiltä osin, joissa konstruktio on vakio mallista riippumatta. Hyviä esimerkkiä asiakasohjautuvasta valmistusaloitteesta ovat esimerkiksi ravintolan ruoat ja ovet. (Haverila, Uusi-Rauva, Kouri & Miettinen 2005, 354). Etuina ovat joustavuus myytävien tuotteiden ja kokoonpanojen markkinoille tuonnissa. Lyhyet toimitusajat, sekä pienet valmistuserät ovat erityisen vaativia tällä metodilla. (Lapinleimu 2001, 109.)

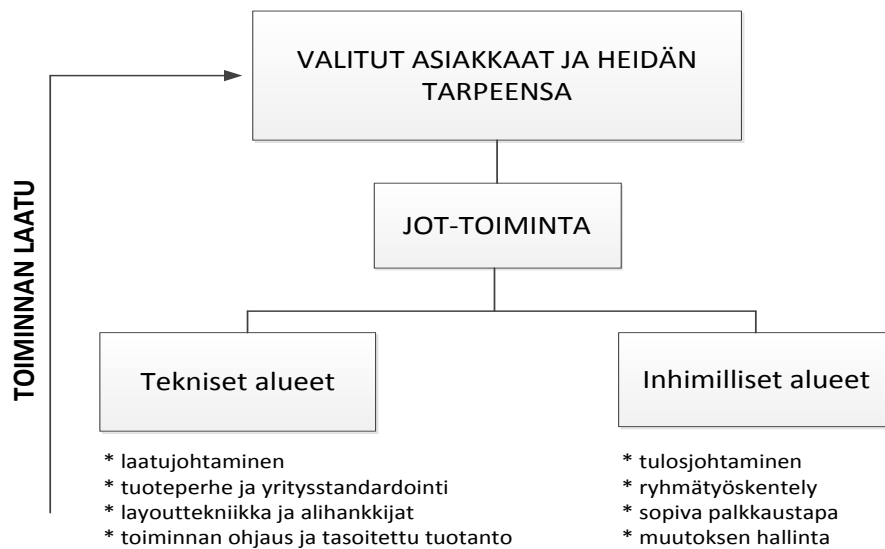
### **2.5.1 Massatuotanto**

Massatuotanto kehittyi 1800-luvulla aseellisuuden kautta. Massatuotannossa osien valmistus- ja kokoonpanovaiheet on jaettu pieniin osiin, niin ettei työntekijöiltä vaadita suurta erikoistumista kyseiseen työvaiheeseen. Näin työvaihe on helposti opittavissa. Osien valmistus on myös standardoitu niin, että valmistus ja kokoonpano voidaan pitää toisistaan irrallaan. Pitkien ja laajojen materiaalivirtojen takia massatuotanto on hidas reagoimaan markkinoiden vaatimiin muutoksiin ja laatuvirheiden lähteiden havaitseminen vaikeutuu. (Haverila, Uusi-Rauva, Kouri & Miettinen 2005, 359-361.)

### **2.5.2 Just in time –tuotanto**

JIT-tuotantomalli on Japanissa syntynyt massatuotannosta poikkeava tuotantometodi. JIT-tuotannossa kapeilla materiaalivirroilla, pienillä tuotantoerillä ja välivarastojen välttämällä päästään välittömään, muutoksiin nopeasti reagoivaan, laadunhallinnallisesti selkeään valmistusmuotoon. Laatuvirheen syntyessä tuotanto jämähtää paikoilleen, mikä pakottaa niin työntekijöitä, kuin johtoporrastakin jatkuvasti kehittämään laadunhallintaa. JIT-tuotannossa pyritäänkin täysin virheettömiin tuotteisiin, mikä vähentää myös muita kuluja. (Haverila, Uusi-Rauva, Kouri & Miettinen 2005, 361-362.)

## JOT-TOIMINTA KOKONAISUUTENA



Kuvio 6. JOT-toiminta kokonaisuutena (Tiainen 1996, 14.)

### 2.5.3 Lean Management

Lean management on keveyteen ja joustavuuteen pyrkivä järjestelmä, jossa mahdollisimman vähillä resursseilla pyritään valmistamaan asiakkaan tuotteet. Tuotannon toiminta pyritään yksinkertaistamaan karsimalla lisäarvoa tuottamatonta työtä. Lean-toimintatavan ansiosta työpanosta ja aikaa käytetään puolet vähemmän aikaisempaan verrattuna. Keskennäköinen tuotanto putoaa murto-osaan ja tuotevarastojen määrä vähenee neljännesosaan. Näin tuotteet saadaan kerralla valmiiksi, eikä tuotteita tarvitse valmistaa useaan kertaan. (Miettinen 1993, 61.)



Kuvio 7. Lean management keskittyy tuottamattomien vaiheiden eliminointiin

### **3 SIMULOINTI**

Tehdassimulointia käytetään tuotantoyrityksissä toiminnanohjauksen simulointiin. Simuloitavina saattavat olla esimerkiksi tuotannon ohjaus, suunnittelu, laskenta tai koulutus. Simulointia käytetään yleensä silloin, kun tutkittava kohde on joko liian vaarallinen, kallis, vaikea tai tapahtuma on liian nopea tai hidas käytännön testeihin. Tietokoneen laskentakapasiteetin ansiosta käyttäjä saa simuloinnista nopeasti sekä numeerisia että graafisia tuloksia. Numeerisena saadun datan avulla halutut parametrit on tarvittaessa helppo laskea ja optimoida. Tehdassimuloinnissa keskitytään yleensä nimenomaan pullonkauloihin, erilaisiin tuotantotekniikoihin sekä materiaalivirtojen ohjaukseen. (Saari & Oijennus 2004, 25.)

#### **3.1 Simuloinnin työvaiheet**

Tehdassimulaatiomallin teko lähtee yleensä jostain tarpeesta. Vaikka tarve ja ongelma yleisellä tasolla olivatkin tiedossa, voi pääongelma olla hämärän peitossa. Ennen simulaation rakentamista onkin pohdittava tarkasti, mikä on simulaatiolla haettavan ratkaisun pääongelma. Pääongelman kautta voidaan lähteä ratkaisemaan muita, pienempiä ongelmia. Tietysti ihan normaalin projektin tavoin, on myös määriteltävä simulaatioprojektille aikataulu ja tavoite. Alussa on määriteltävä mitä parametreja halutaan muuttaa, missä muodossa ja miten tarkasti tulokset esitetään. (Saari & Oijennus 2004, 32.)

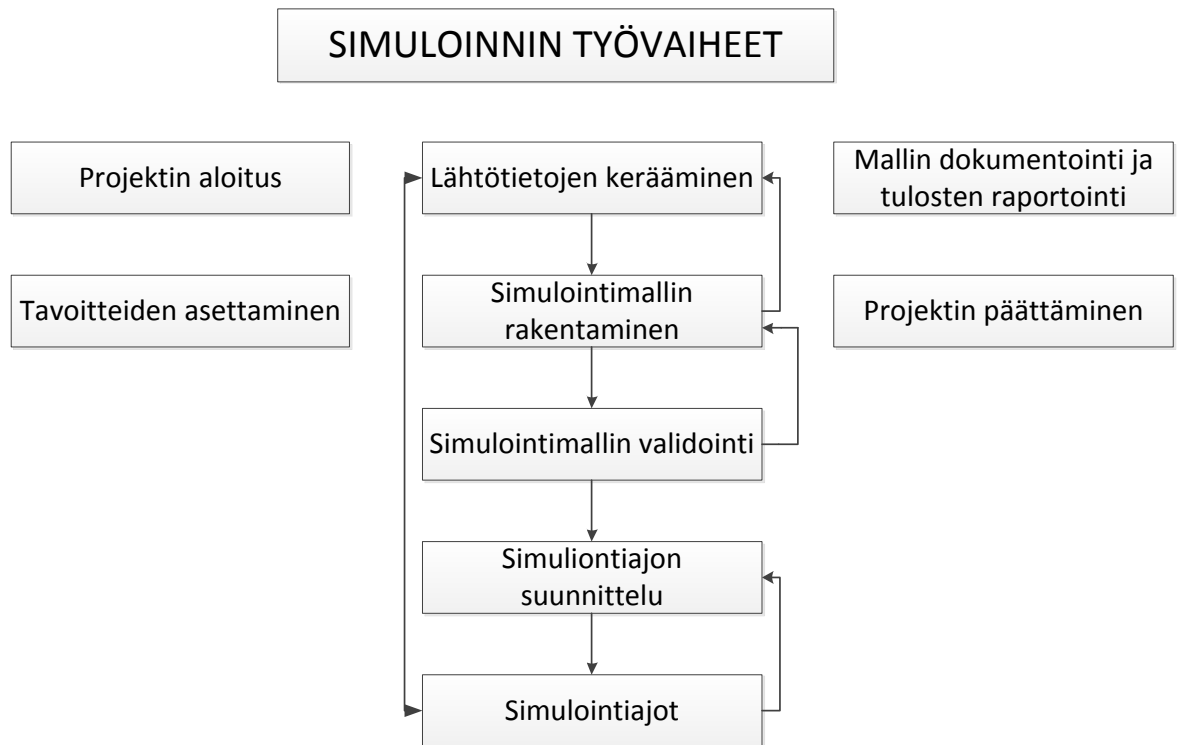
Varsinainen projekti alkaa tietysti lähtötietojen keruulla. Lähtötietoihin sisällytetään esim. läpimenoajat, laitteiden ja tuotantotilojen mitat ja muut ominaisuudet. Kaikkia tietoja ei välttämättä saada kerättyä, jolloin voidaan käyttää keskiarvolukemia tai käytännön arvioita. (Saari & Oijennus 2004, 33.)

Lähtötietojen keruun jälkeen suoritetaan laitteiden 3D-mallinnus ja layoutin teko. Layoutpohjat saattavat olla jo valmiina ja mahdollisesti myös laitteiden 3D-mallit ovat valmiina saatavilla. (Saari & Oijennus 2004, 33.)

Lähtötietojen keruun ja tuotantotilojen staattisen layoutin mallintamisen jälkeen mallinnetaan simulaatiomalliin toiminnallisuus. Tämä vaihe sisältää simuloitavan mallin kannalta tärkeimmät tiedot. Lähtötietojen perusteella annetaan oikeat tiedot ohjelmakoodiin, jotta saataisiin ohjelman logiikka vastaamaan mahdollisimman tarkasti todellisuutta. Ohjelmakoodin yksinkertaisuuteen ja modulaarisuuteen on syytä kiinnittää huomiota, sillä koodia voi syntyä tuhansia rivejä. (Saari & Oijennus 2004, 33.)

Ennen varsinaista simulointia suoritetaan mallin tarkistus eli validointi. Mallia verrataan esimerkiksi historiatietoihin ja tarpeen vaatiessa malliin tehdään muutoksia. Mallin ollessa valmis voidaan aloittaa itse simulointi ja saadun datan analysointi. Simulointi ajetaan useita kertoja monilla parametrien variaatioilla. Ajetaan niin sanottuja ”mitä jos” -tarkasteluja, missä haluttuja parametreja muunnellaan niin kauan että saadaan simulaatio haluttuun tulokseen. Simuloitujen tulosten tarkka analysointi simuloinnin jälkeen on tärkeä ja monesti unohdettu osa simulaatioprojektia. Mallin rakentaminen ja mallin validointi saattaa helposti kattaa alle puolet koko simulaatioprojektin kestosta. (Saari & Oijennus 2004, 34.)

Simulointiprojekti päätetään dokumentointiin ja loppuraporttiin. Tarkistetaan että tehdystä projektista saadaan halutut tiedot ulos ja mahdolliset jatkotoimenpiteet. Varsinaisesti projekti on valmis vasta siinä vaiheessa, kun simuloinnista saatuja tuloksia hyödynnetään käytännössä. Huolellisesti tehdyllä dokumentoinnilla varmistetaan simulointimallin ja tulosten hyödynnettävyys jatkossakin. (Saari & Oijennus 2004, 34.)



Kuvio 8. Tehdassimuloinnin työvaiheet (Saari & Oijennus 2004, 32.)

### 3.2 Simulointityökalut

Simulointityökalut voidaan jaotella kahteen pääryhmään, tapahtumaperusteiseen ja jatkuvaan simulointiin. Simulointiin voi käyttää joko ohjelmointikielen kaltaisia simulointikieliä tai valmiita ohjelmistoja. Simulointikielen käyttö vaatii käyttäjältään vahvaa ohjelmoinnin osaamista, mahdollistaen samalla sovelluskohteesta riippumattomat käyttömahdollisuudet sekä monipuolisten ja yksityiskohtaisten mallien rakennuksen. Simulointiohjelmistot keskittyvät monesti johonkin tiettyyn sovelluskohteeseen, kuten tuotannonohjaus, logistiikka, materiaalivirtaus, robotiikka tai NC-järjestelmät (Saari & Oijennus 2004, 31.)

#### 3.2.1 Delmia V5

DELMIA V5:n avulla valmistajat voivat toimia vuorovaikutuksessa tehtaan prosessien kanssa jo varhain suunnitteluvaiheessa. Insinöörit, johto- ja sidosryhmät voivat kolmiulotteisesti visualisoida suunniteltua tuotantotilaa, sekä suorittaa "mitä

jos” -skenaarioita, joiden avulla he voivat tehdä nopeita muutoksia, optimoida työtehoa sekä tunnistaa ja poistaa suunnittelussa tapahtuneita virheitä. Tämä työkalu helpottaa suunnittelua ja edistää yrityksen innovatiivisuutta. (Dassault Systemes. [Viitattu 18.2.2011].)

### **3.2.2 Flexsim**

Flexsim on tehokas työkalu mallintamiseen, analysointiin, visualisointiin sekä prosessioptimointiin. Mallin voi ohjelmassa rakentaa Flexsimin 3D työkalulla. Flexsimissä on sisäänrakennettu OptQuest-ohjelma, jonka avulla voi kokeilla eri variaatioita järjestelmästä ja maksimoida järjestelmän potentiaalin. Ohjelmaan voi tuoda omat mallit ja CAD-layoutit. Ohjelman avulla voi tuotannosta havaita pullonkaulat, sekä tehdä kuvaajia, raportteja ja tilastoja tuotannosta. (Flexsim. [Viitattu 18.2.2011].)

### **3.2.3 Visual Components 3DCreate**

Visual Components tarjoaa valmiita ohjelmistoratkaisuja myyntiin ja markkinointiin, robotti- ja työsolusimulointiin, tehdassimulointiin, ohjausympäristön validointiin, sekä ohjelmistokehitykseen. Suomalainen Visual Components perustettiin 1999 ja se on kasvanut globaaliksi yritykseksi vuosien saatossa. 3Dcreate-ohjelmalla voidaan tuoda valmiita CAD-kuvia suoraan ohjelmaan ja niiden avulla mallintaa simulaatiomalli tuotantotiloista. Malliin voidaan lisätä parametrit, säännöt, toiminnot ja rakentaa siitä toimiva simulaatio. Näin voidaan luoda kokonainen kirjasto erittäin realistisia uudelleenkäytettäviä simulaatiomalleja. (Visual Components. [Viitattu 15.2.2011].)

## 4 TYÖVAIHE

### 4.1 Taustatietojen keräys

Ennen varsinaista simulaatiotyön aloittamista, täytyy tehdä alustava työ, jossa kerätään eri valmistusvaiheiden läpimenoajat, tuotantoa ohjaavat säännöt ja rajoitteet, kapasiteetit ja ehdot eri työvaiheille. Läpimenoajat saatiin valmiiksi kelloitetuina, mutta ajat sisälsivät myös tauot ja apuajat, joten työpisteiden varsinaista työaikaa täytyy hieman simulaatiomallissa kompensoida. Taustatietojen keruuvaiheessa on myös erittäin tärkeää kerätä mitat mallinnettavista koneista ja komponenteista. Erityisen tärkeää ovat mitat, jotka määrittävät materiaalivirtojen kapasiteettia, kuten esimerkiksi kuljettimien pituudet ja välivarastojen koot. Tehtaan layoutin mitat saadaan helposti tehtaan pohjapiirrustuksesta, jonka päälle voidaan simulaatio suoraan mallintaa. Simulaation validointivaiheessa on tärkeää tietää millaisia, tuotantomääriä tehdas tuottaa esimerkiksi viikon aikana, jotta simulaatiomallia on helpompi vertailla todellisuuteen.

### 4.2 Simulaatioympäristön kehittäminen

Simulaation teko vaiheessa ongelmiksi koitui monimutkaisten toimintasuhteiden ja kinematiikkojen luominen. Visual Componentsin 3Dcreate simulaation rakentamisessa on hankalammin opittavissa spesifimpien konfiguraatioidensa suhteen verrattuna Delmian Questiin. Yksinkertaiset tehtävät kuten materiaalivirtojen yhdistäminen sekä muutokset tuotteisiin prosessin aikana tuottivat ongelmia, koska niissä yleensä tarvittiin sekä python-koodia että ohjelmiston omia komponentteja. Tuotannon parametointi aiheutti myös ongelmia. Tavoitteena olisi, että voitaisiin simuloida oikeita olemassa olevia työeriä mallissa, mutta tuotteiden monimuotoisuus ja poikkeavuudet valmistusprosessissa aiheuttivat todella paljon työtä mallin rakennusvaiheessa. Monet materiaalin valmistusprosessit määräytyvät osittain epämääräisten ja tilannekohtaisten parametrien kautta, näin niiden simulointi tarkasti on mahdotonta.



Simuloinnin tarkoituksena on tutkia materiaalivirtojen eri liikeratojen kapasiteettia ja tarkastella, voidaanko niitä optimoimalla vähentää välivarastojen määrää. Pääkohteena tässä projektissa on maalauslinjaston ja tuotantolinjan välinen välivarasto. Simulaation avulla pyritään selvittämään voidaanko kyseinen välivarasto poistaa käytöstä hyödyntämällä maalauslinjaston kapasiteettia paremmin. Mahdolliset ongelmat aiheutuvat tuotteista joiden läpimenoaika on pidempi kuin toisten tuotteiden läpimenoaika, estäen materiaalin virtauksen tietyiltä osin. Oleellisena osana tässä tulee myös materiaalivirtojen kulku maalauslinjastolla, sekä maalauslinjaston kapasiteetti kappaleiden jonossa.

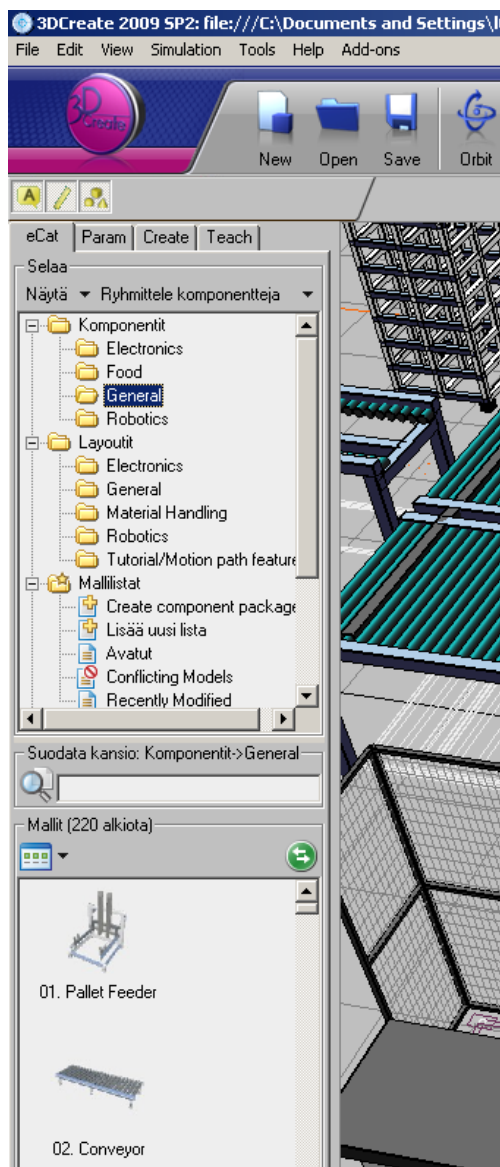
Maalattujen kappaleiden oikein ajoitettu kulku osoittautuikin hieman monimutkaiseksi simulointiohjelman osalta, koska eri vaiheajat ovat hyvin suurpiirteisiä, sekä työjärjestys on käyttäjäkohtainen. Ongelmia tuotti eniten maalauslinjastolla olevien materiaalien käsittely niin, että saataisiin materiaali kulkemaan mahdollisimman realistisesti, koska tämä osio on työn kannalta kaikkein tärkein simulaation oikeellisuuden kannalta.

### **4.3 3DCreate**

Visual Componentsin 3DCreate sisältää myös yksinkertaistetun CAD-mallinnustyökalun, mikä mahdollistaa helpon tuotantotilojen, työkoneiden, työkalujen sekä tuotettavien kappaleiden mallinnuksen. Mallinnustyökalu on kevyt ja yksinkertaistettu versio verrattuna täysimittaisiin 3D CAD-suunnitteluohjelmistoihin, mikä näkyy erilaisten toimintojen määrässä. Esimerkiksi komponenttiin ei voi reikää tehdä suoraan, vaan elementti täytyy rakentaa osista reiän ympärille. Toisaalta mallinnustyökalun yksinkertaisuus on myös ohjelman vahvuus, sillä liian monimutkaiset komponentit tulisivat helposti liian raskaiksi simuloitavaksi. Jokainen osio ohjelmassa on jaettu omalle välilehdelle, joista kerrotaan seuraavassa.

### 4.3.1 Komponentti- sekä layoutkirjasto

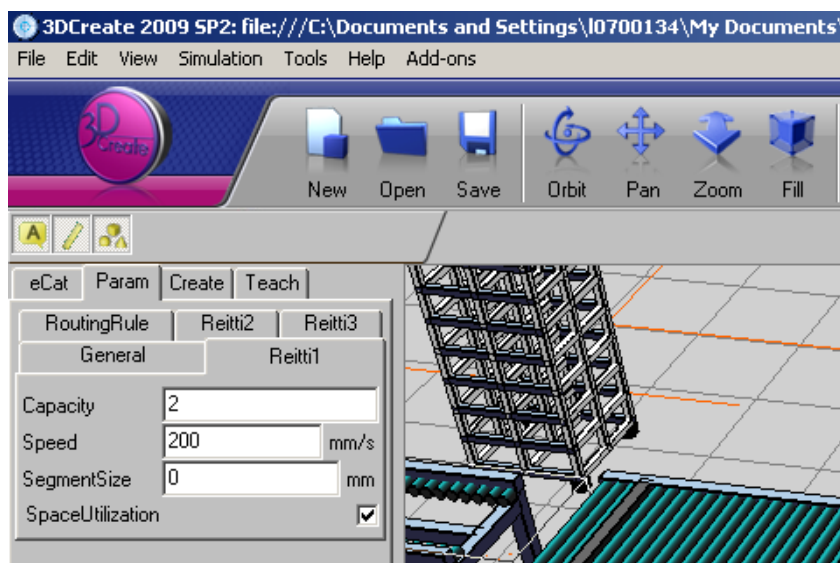
Ohjelman ensimmäinen välilehti eCat sisältää käyttäjän oman sekä esimerkkikirjaston. Kirjastosta löytyy valmiina parametroidut peruskomponentit kuten kuljettimet, trukkilavat, verkkoaidat, toimistotyökalut ja ihmiset. Esimerkkikirjasto sisältää myös valmiita layoutteja mallitehtaista, joista on helppo ottaa vinkkejä omaa mallia tehdessä. Komponenttien ja layouttien siirtäminen malliin onnistuu helposti drag-and-drop-menetelmällä.



Kuvio 9. 3DCreaten komponentti- ja layoutkirjasto

### 4.3.2 Parametrisuus

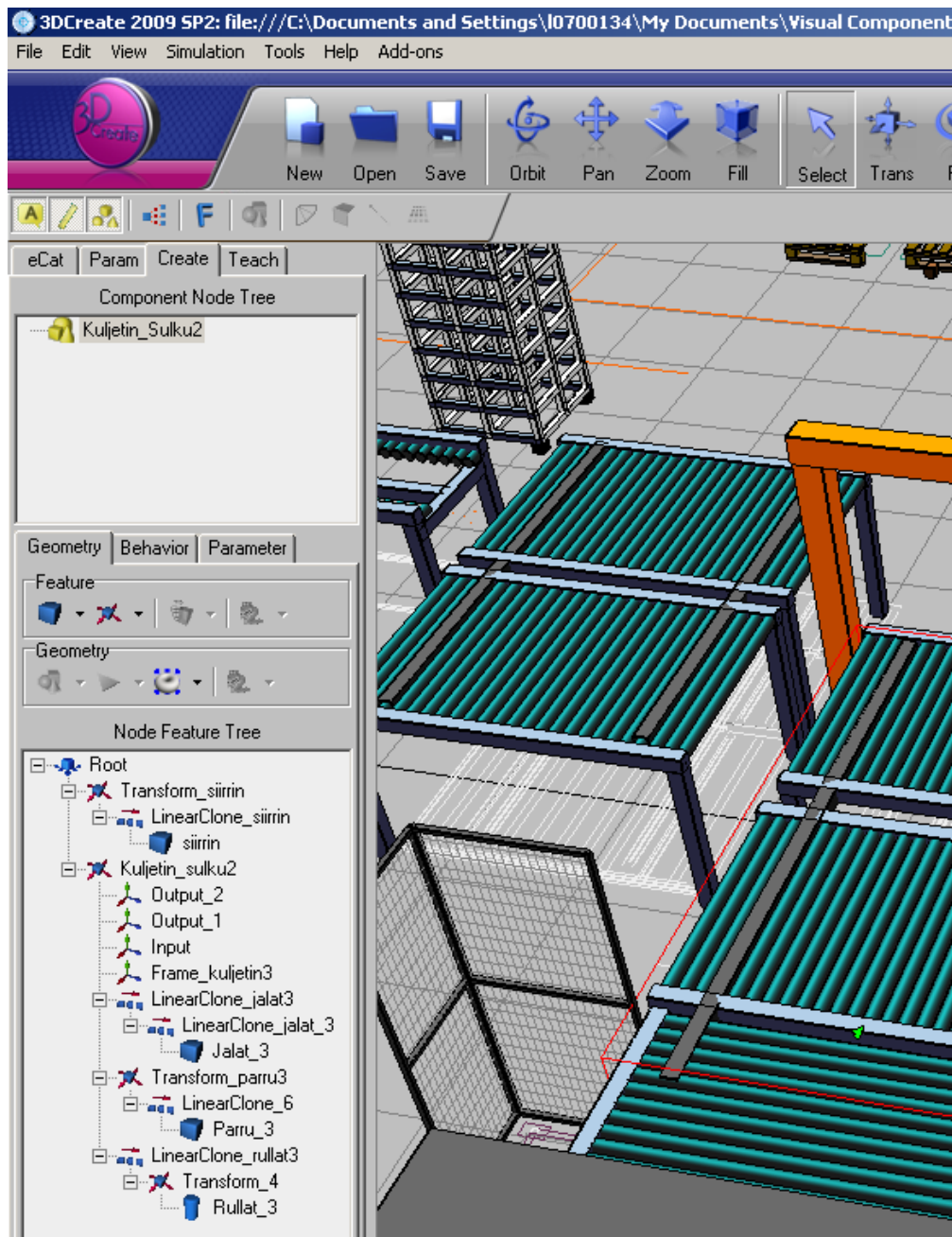
Param-välilehdellä voidaan helposti hallinnoida ja muokata komponenteille aseteltuja parametreja. Parametrien luomisesta kerrotaan myöhemmin. Param-välilehden alta löytyy useampi välilehti, jotka ovat jokainen valitun komponentin eri osioiden parametrit. Parametreilla voidaan hallinnoida esimerkiksi kappaleen geometrisiä mittoja, erilaisia materiaalivirtasääntöjä kuten liikkeen nopeutta tai prosessinopeutta. Parametreja voidaan käyttää myös ihan materiaalivirtaukseen määriteltyjen sääntöjen tukena.



Kuvio 10. 3DCreaten parametrienhallintavälilehti

### 4.3.3 Geometrian mallintaminen

Mallin luomisen kannalta tärkein välilehti on Create-välilehti. Create-välilehdessä näkyy valittu komponentti ja tämän välilehden alla on kolme välilehteä, joista ensimmäinen on Geometry-välilehti. Geometry -välilehden alta löytyy 3D-mallintamiseen työkalut. Feature-valikon alta löytyy valmiita palloja, laatikoita, sylintereitä, sekä muita muotoja, jolle annetaan luomisvaiheessa mitat. Valikoista löytyy myös työkalut komponenttien kloonaukseen, peilaukseen, sekä *Framen* luontiin. Välilehden alaosassa näkyy piirrepuu, jonka kautta voidaan helposti selaila komponentin eri piirteitä.

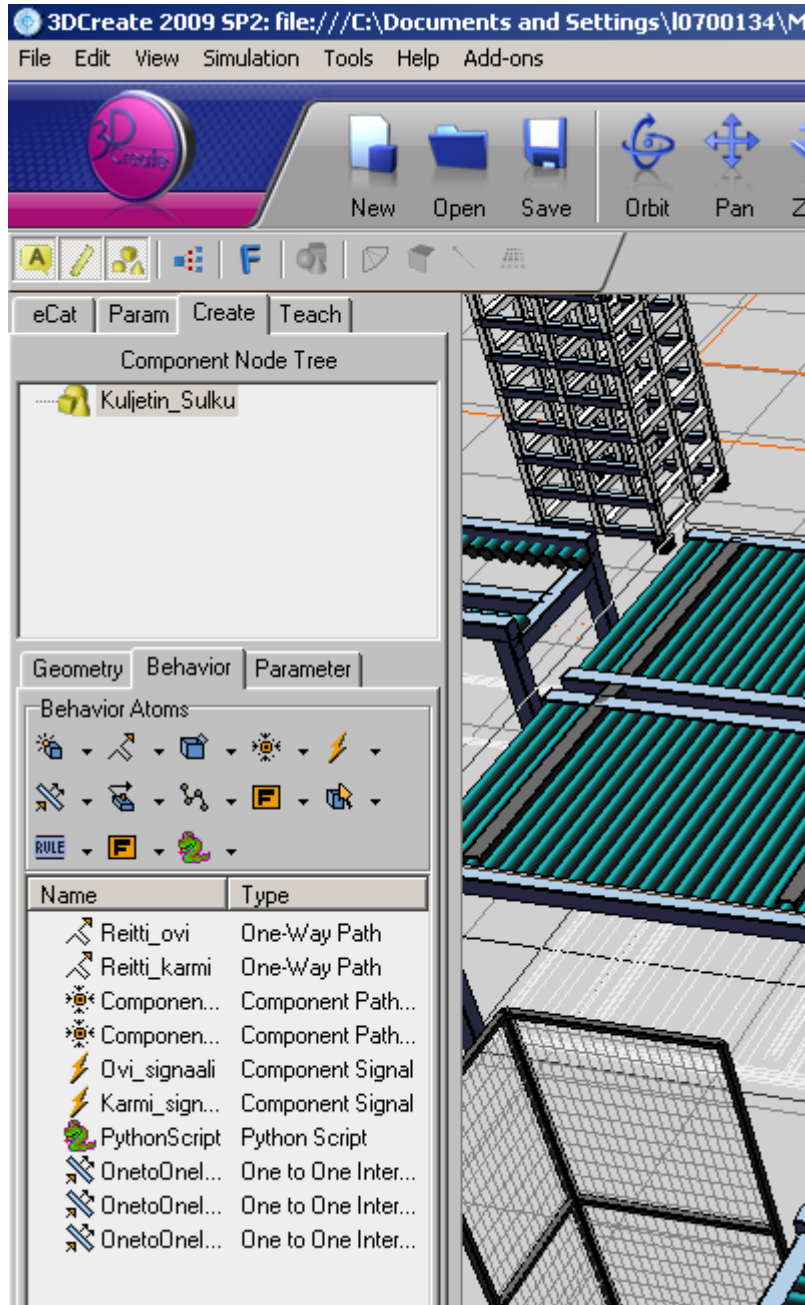


Kuvio 11. 3DCreaten mallintamistyökalu

#### 4.3.4 Sääntöjen luonti

Create-välilehden alta löytyy materiaalivirtojen hallinnoimiseen sääntöjenluontityökalu. Valikoista on nopeasti valittavana materiaalinluojakomponentti, reittikomponentti, signaalikomponentit, reitityssäännöt, sekä muiden sääntöjen lisäksi Python-scripti-komponentti erikoisempien sääntöjen luomiseen. Python-scripti-komponenttiin voi itse helposti liittää esimerkiksi eri antureita (signaaleita) ja tämän

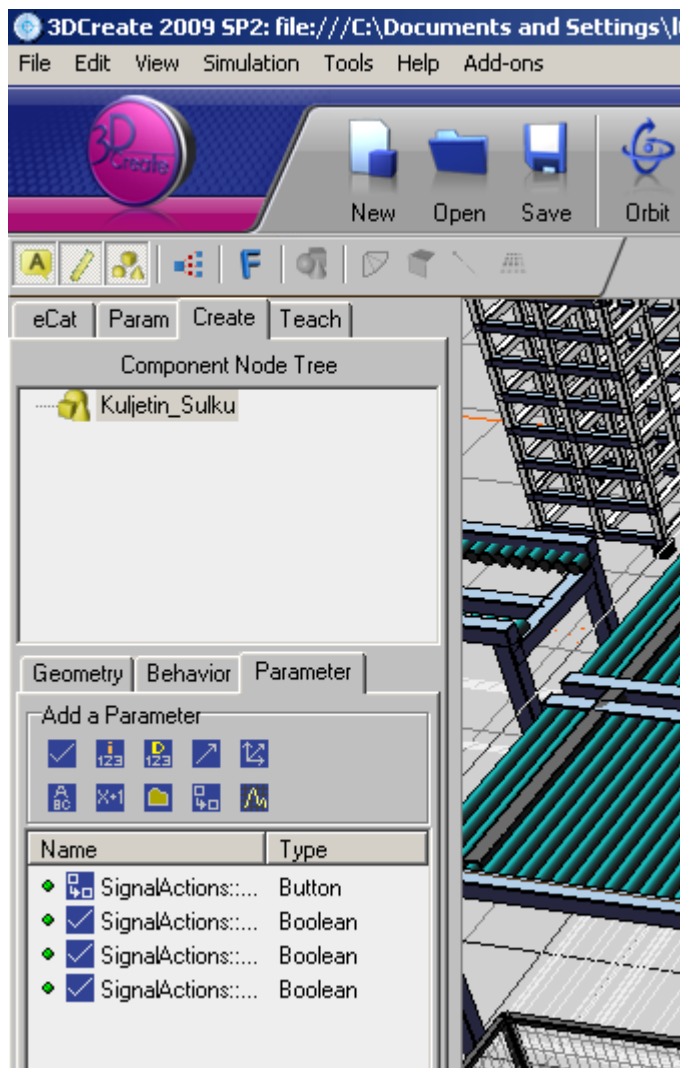
jälkeen Python-ohjelmointikielellä ohjelmoida halutut toiminnot simulaatioon. Kuvassa 12 näkyvät Behaviour Atoms -ikkunan työkalujen alla käyttäjän määrittelemät säännöt.



Kuvio 12. 3DCreaten Behaviour-välilehti

### 4.3.5 Parametrien luonti

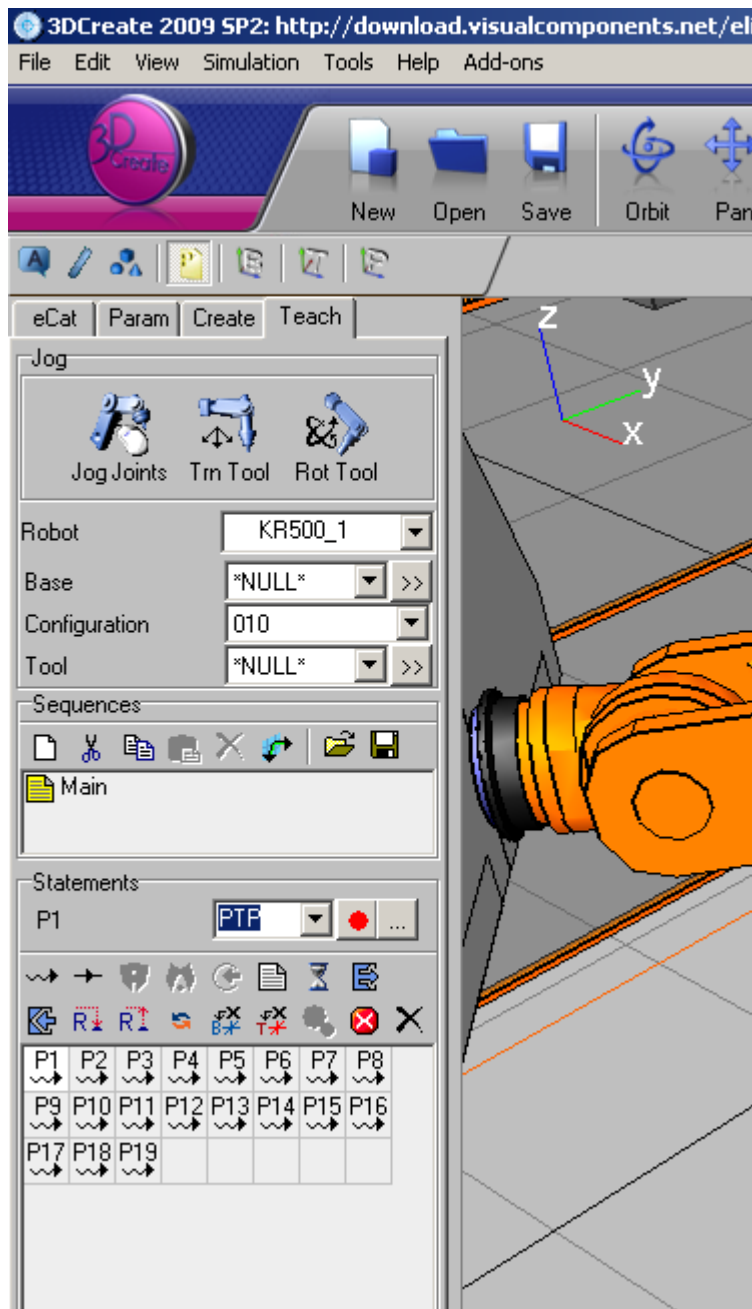
Create-välilehden alla, Parameter-välilehdellä luodaan komponenteille erilaisia parametreja. Valittavana on erimuotoisia parametreja, esim. boolean, integer, string ja numeraali. Luotuja parametreja voidaan hallinnoida joko Param-välilehdeltä tai esimerkiksi Python-scriptistä. Kuvassa 13 käyttäjän määrittelemät parametrit näkyvät Parameter-välilehden alla.



Kuvio 13. 3DCreaten parametrien luonnin välilehti

### 4.3.6 Robotin ohjelmointityökalu

Oikeanpuoleisin välilehti, Teach, keskittyy robottien ohjaamiseen ja ohjelmointiin. Työkalulla voidaan käsitellä malliin tuotua tai itse mallinnettua robottia manuaalisesti Jog-valikon alta. Tämän työkalun avulla robotille voidaan virtuaalisesti opettaa liikeradat, ja simuloida ohjelma ennen robottiin latausta.



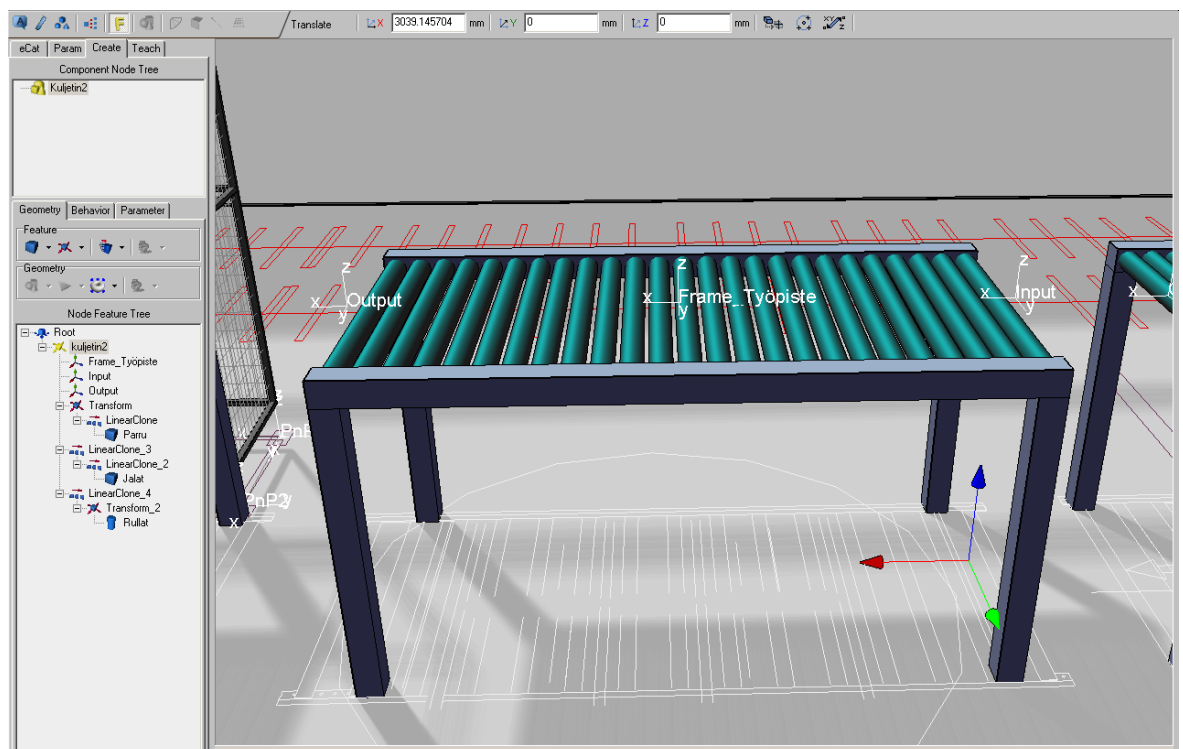
Kuvio 14. 3DCreaten robotin opettamistyökalu

## 4.4 Simulaatiomallin työvaiheet

Lähtötietojen keruun jälkeen aloitetaan simulaatiomallin rakennus. Mallintaminen rakennetaan loogisessa järjestyksessä alottaen työstökoneiden, tuotteiden ja muiden pienien kokoonpanojen mallintamisesta. Valmiit alikokoonpanot kootaan isoksi kokonaisuudeksi muodostaen tuotantotilat.

### 4.4.1 Työpisteen mallintaminen

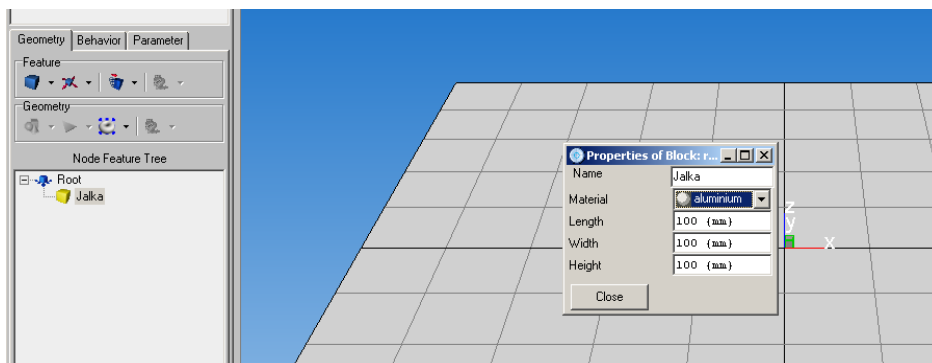
Työpisteen mallintaminen tapahtuu pääosin aina samalla kaavalla, olipa se sitten yksittäinen työpöytä tai tuotantolinjastolla oleva rullakuljetin. Työpisteelle määritellään input josta tavara tulee työpisteeseen. Inputista kappale menee työstettävään kohtaan, eli yleensä työpisteen keskelle. Työlle annetaan parametrit, joiden mukaan kappale viettää aikansa työpisteellä. Annettu parametri voi olla kiinteä aika vaihteluarvoineen tai kappale voi olla riippuvainen jostain toisesta prosessista, joka täytyy suorittaa ensin. Kun kappale on valmis, voi kappale jatkaa työpisteeltä outputtiin, joka on kiinnitetty tuotannon seuraavaan vaiheeseen.



Kuvio 15. 3DCreatella mallinnettu työpiste

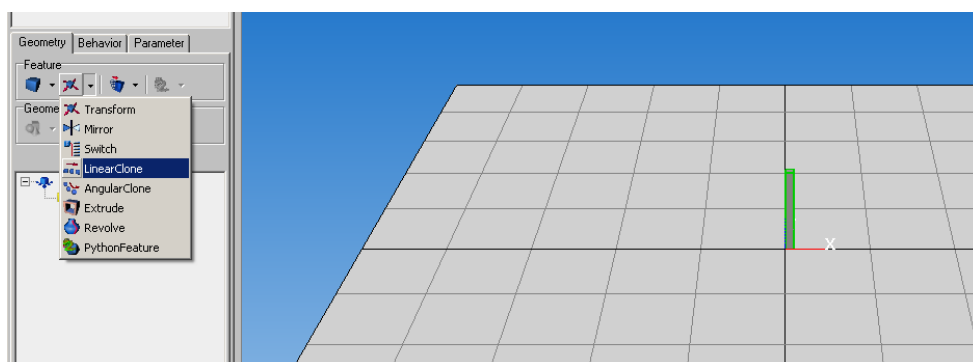


Kuvassa 15 on mallinnettuna rullakuljettimella oleva työpiste. Rullakuljettimen mallintaminen aloitetaan valitsemalla Create-välilehden alta Geometry-välilehti. Geometry-valikon alla on Feature-valikko, jossa heti ensimmäisenä on Block-työkalu. Valitaan Block-työkalu ja klikataan mallinnettavaan maailmaan. Esiin tulee palikalle määritettävät parametrit.



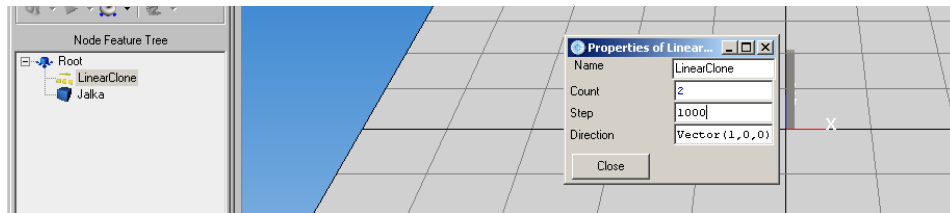
Kuvio 16. Blockin luonti

Syötämme parametrilaatikkoon nimikkokohtaan ”Jalka”, materiaaliksi määrittelemme alumiinin ja korkeus kohtaan korvaamme lukeman 100 mm luvulla 1000 mm. Korkeuden asettamisen jälkeen painamme enteriä ja painamme Close-nappulaa. Seuraavaksi valitsemme Feature-valikosta LinearClone työkalun.



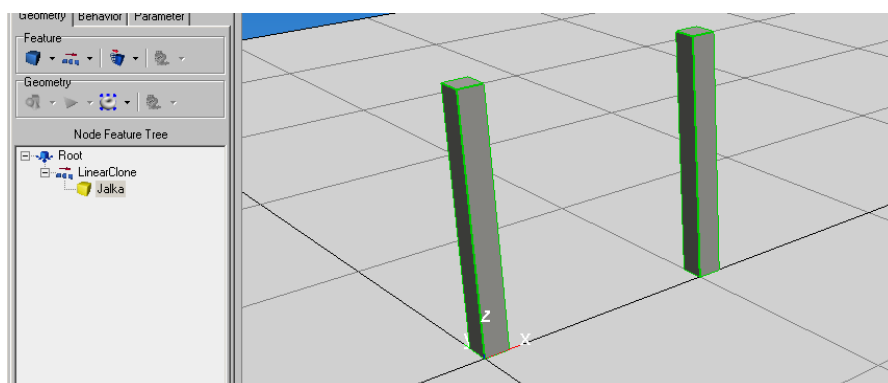
Kuvio 17. Jalan kloonauus

LinearClone valikosta asettelemme Count-tekstiruutuun 2, eli jalkoja on kloonauksen jälkeen 2 kappaletta. Step-tekstiruutuun vaihdamme luvun 1000, eli jalkojen väliseksi etäisyydeksi tulee 1000 mm.



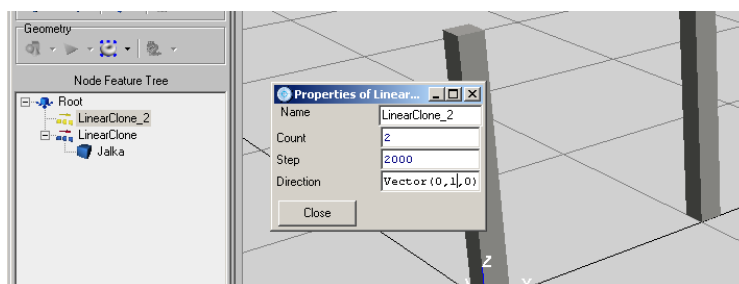
Kuvio 18. Kloonauksen parametrit

Parametrien asettelun jälkeen suljemme taas valikon. Vasemmalla olevassa piirrepuussa siirrämme Jalka-piirteen hiirellä raahaamalla LinearClone-piirteen päälle, jolloin Jalka-piirre on kloonauksen vaikutuksen alainen.



Kuvio 19. Jalka kloonattuna

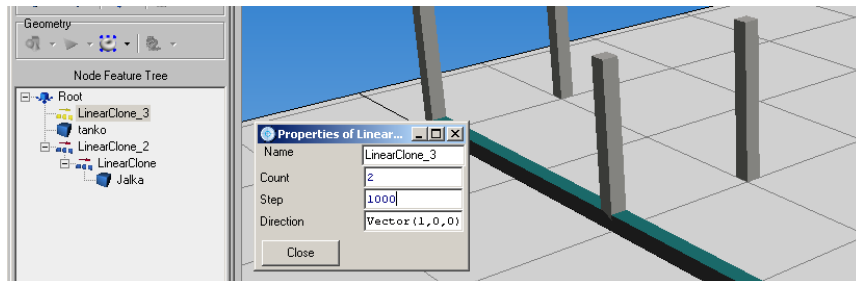
Seuraavaksi kloonamme malliin vielä 4 jalkaa lisää. Valitaan jälleen Feature-valikosta LinearClone, Asetamme parametreihin Count-kohtaan 3, Step-kohtaan 2000 ja Direction-kohdan muutamme muotoon (0,1,0). Tämä tarkoittaa, että kloonaus suoritetaan y-akselin suuntaisesti.



Kuvio 20. Toisen kloonauksen parametrit

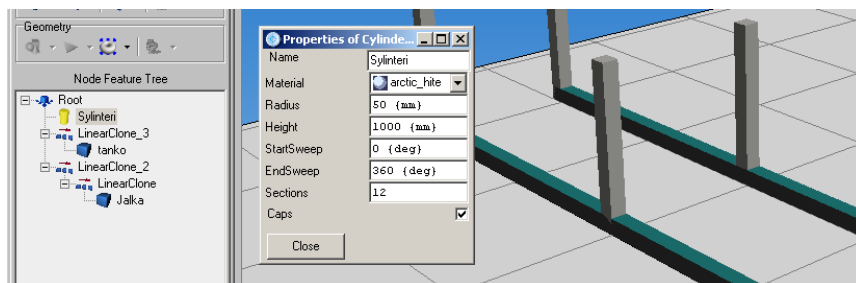
Lopuksi raahaamme Linear Clone-piirteen LinearClone\_2 piirteen alaisuuteen. Seuraavana luomme taas uuden Blockin Feature-valikon alta. Kappaleelle annamme nimen "tanko". Materiaaliksi määrittelemme "Aqua" ja leveydeksi määritte-

lemme 4100 mm. Suoritamme tanko-komponentille kloonauksen toiminnon, eli toimimme samalla lailla kuin ensimmäisessä kloonauksessa. Count-kohtaan annamme luvun 2, Step-kohtaan 1000 ja Direction-kohtaan emme tee muutoksia jolloin kloonauksen suoritetaan x-akselin mukaisesti.



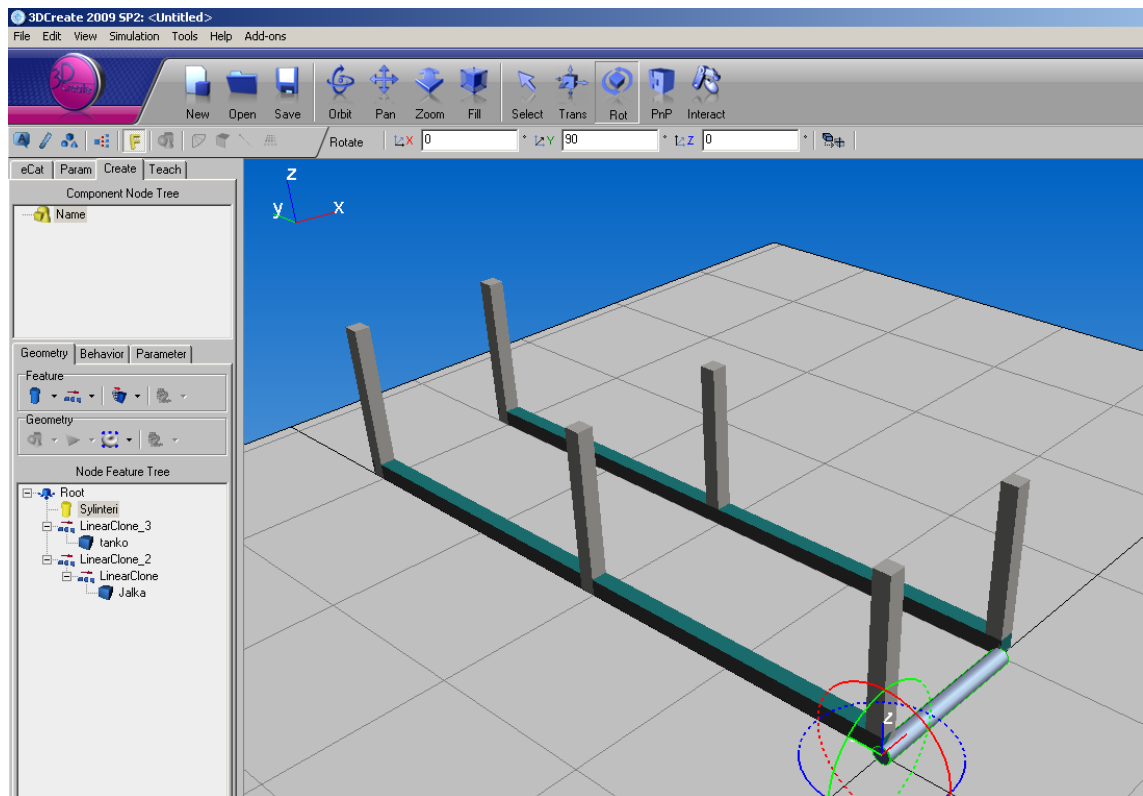
Kuvio 21. Kloonataan tanko

Lopuksi raahataan taas tanko-piirre LinearClone\_3 alaisuuteen. Seuraavaksi valitsemme piirrevalikosta Root-kohdan ja luomme Feature-valikosta Cylinder-komponentin. Sylinterille annamme parametreihin nimi kohtaan "sylinteri", materiaaliksi "Arctic\_hite" ja korkeudeksi 1000.



Kuvio 22. Sylinterin luonti

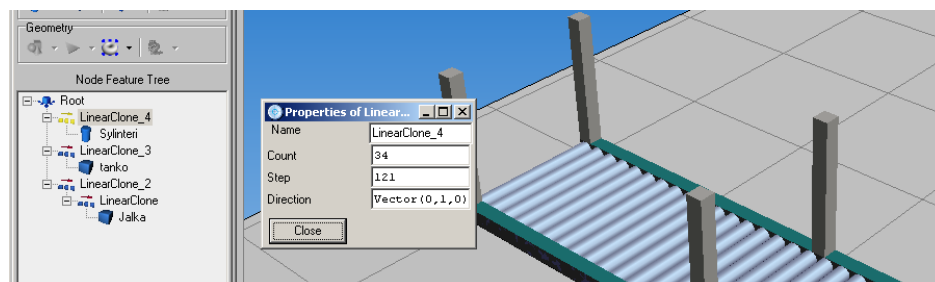
Sylinterin luonnin jälkeen, valitsemme ohjelman ikkunan yläreunasta Rot-työkalu. Tämän työkalun avulla voimme pyörittää kappaletta, joko syöttämällä sille halutun kulman tai pyörittämällä hiirellä. Valitaan-Rot valikosta akseli-y ja annamme sille arvoksi 90, jolloin kappale pyörii 90 astetta suhteessa y-akseliin.



Kuvio 23. Sylinterin kääntö

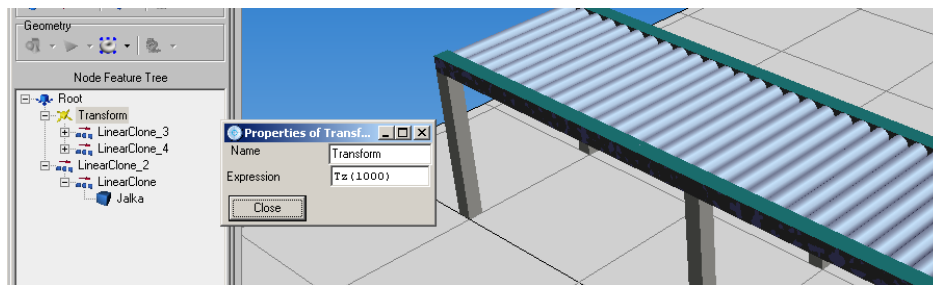
Käännön jälkeen valitsemme Rot-kohdan vasemmalta puolelta Trans-työkalun. Trans-työkalun valikkoon muutamme z-akselin kohdalle arvon 50 ja y-akselin kohdalle arvon 50, jolloin sylinteri nousee 50 mm ylöspäin ja siirtyy 50 mm y-akselilla eteenpäin. Näin saamme mallinnetun kappaleen täysin ”maan pinnalle” ja sopivaan kohtaan jatkoa ajatellen.

Sylinterin noston jälkeen kloonaamme sylinteriä LinearClone-työkalulla. Parametrien arvoiksi annamme Count-kohtaan luvun 34, Step-kohtaan luvun 121 ja Direction-kohtaan arvon (0,1,0). Lopuksi siirrämme taas Sylinterin piirteen kloonauspiirteen alle.



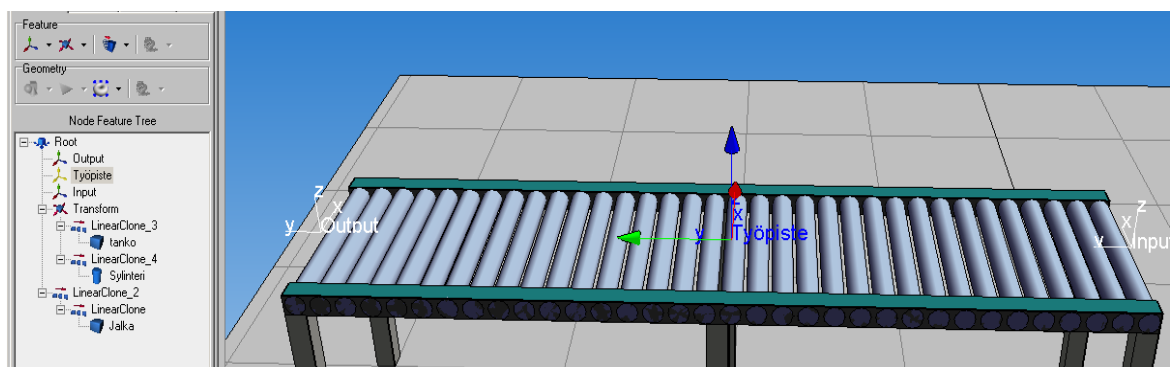
Kuvio 24. Sylinterin kloonaus

Koska rullakuljettimemme taso on vielä maanpinnassa, nostamme sitä Transform-työkalun avulla, joka löytyy Feature-valikosta. Transform-työkalun valikkoon syötämme Expression-kohtaan "Tz(1000)", joka tarkoittaa, että siirretään valittua piirrettä 1000 mm z-akselin suuntaan. Lopuksi siirretään piirrepuussa piirteet LinearClone\_3 ja LinearClone\_4 Transform-piirteiden alaisuuteen.



Kuvio 25. Piirteiden nosto

Lopuksi määritellään rullakuljettimelle vielä Framet. Valitaan Feature-valikon alta Frame-kohta joka lisää yhden Framen malliin. Lisätään Frameja 3 kappaletta. Nimitetään Framet nimillä Input, Output, sekä Työpiste. Asetellaan Input ja Output kuljettimen vastakkaisiin päihin ja Työpiste kuljettimen keskelle.

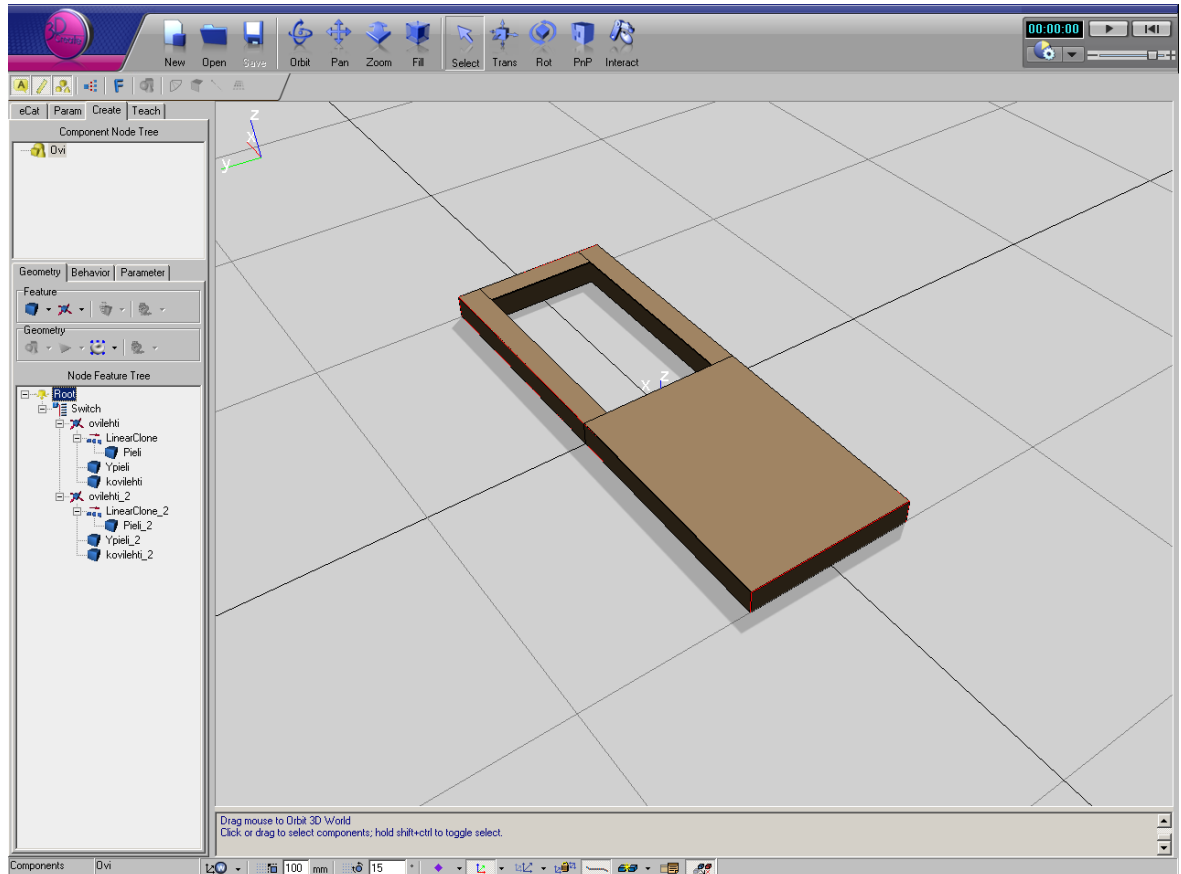


Kuvio 26. Framet

#### 4.4.2 Tuotemallintaminen

Tuotemallintaminen toimii täysin samoilla työkaluilla kuin työpisteen mallintaminen. Kappaleen voi mallintaa suoraan tuotantotilojen lattialle, syöttää parametrit ja sijoittaa oikealle paikalleen. Valmis kappale voidaan sijoittaa suoraan tuotantotiloihin tai kappaleelle voidaan luoda oma *component creator*, joka toimii alkuvarastona

halutulle materiaalivirrälle. Kuvassa 27 oleva ovilehti on yksinkertaistettu versio simuloitavasta tuotteesta. Ovilehden ikkuna-aukko jouduttiin mallintamaan niin, että ovi kasattiin aukon ympärille, koska suoraa materiaalin poisto-ominaisuutta ohjelmasta ei löydy.



Kuvio 27. 3DCreatella mallinnettu ovilehti

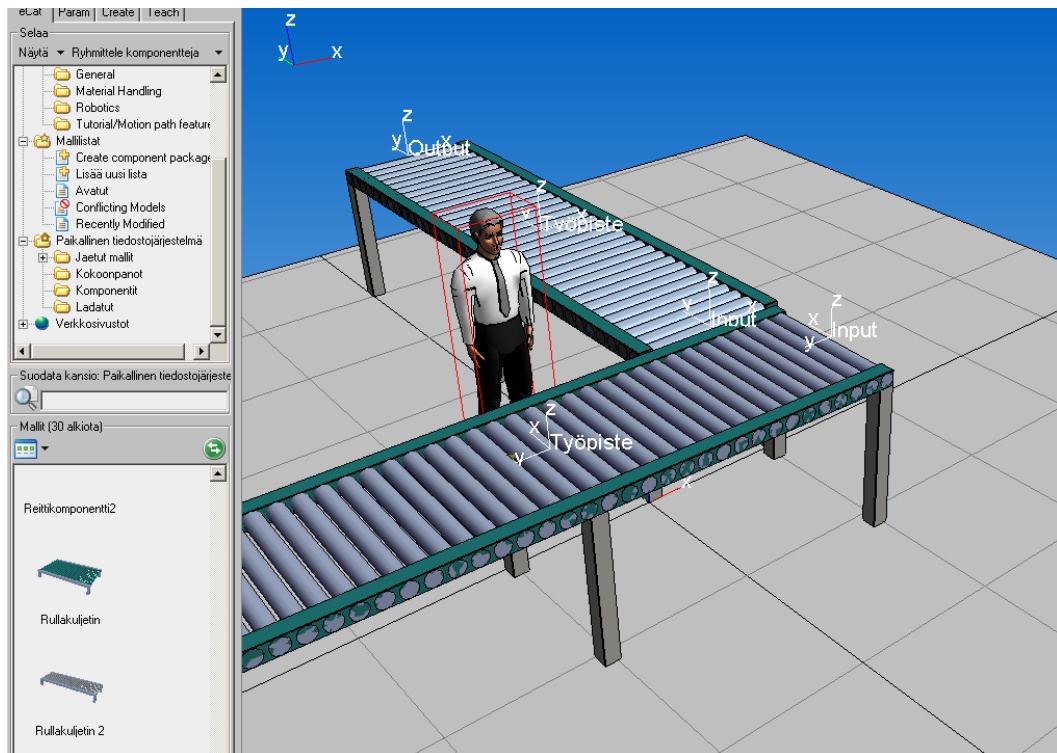
## 4.5 Ympäristön rakentaminen

Tuotantotilojen mallintaminen onnistuu helposti tuomalla dxf-/ tai dwg-muotoinen tuotantotilojen layoutkuva ohjelmaan, kuva voidaan asettaa pohjaksi ympäristölle. Koko ympäristön voi mallintaa suoraan pohjapiirrustuksen päälle. Ympäristön mallintaminen tapahtuu samalla lailla kuin työpisteen tai työkalupöydän mallintaminen. Koneet mallinnetaan yksi kerrallaan, määritellään niille tarvittavat parametrit ja tämän jälkeen linkitetään koneet kiinni toisiinsa ja määritellään riippuvuussuhteet muihin työvaiheisiin.



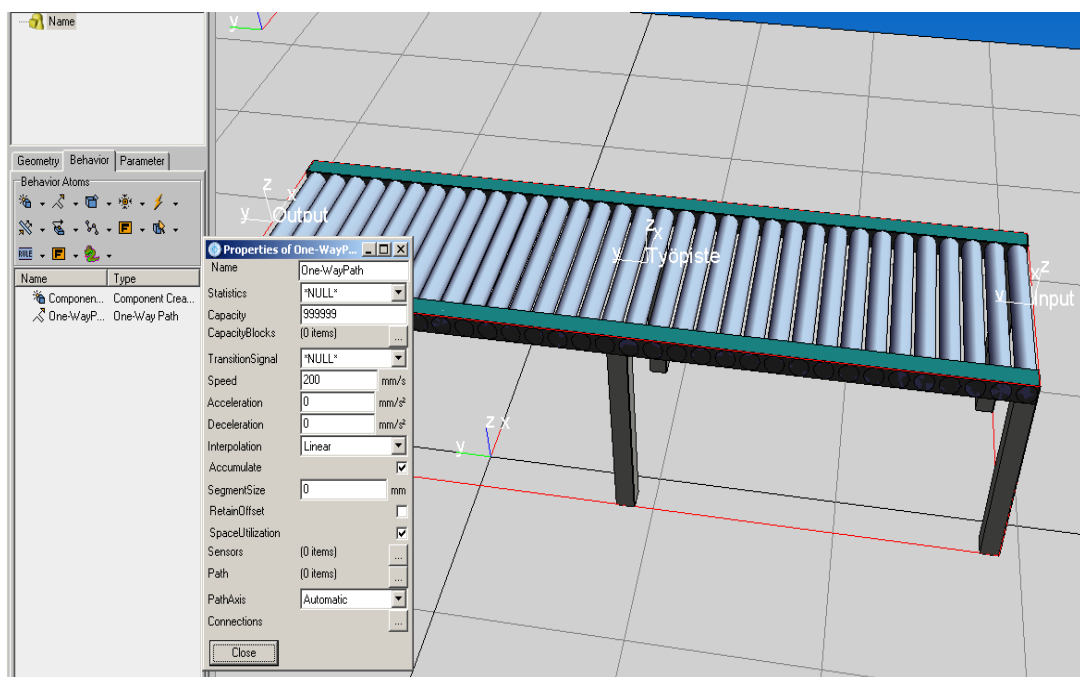
Kuvio 28. 3DCreatella mallinnettu layout

Käytämme aikaisemmin luotua rullakuljetinta ympäristön rakentamisessa. Komponentiksi tallennettu rullakuljetin löytyy 3DCreaten komponenttikirjastosta eCat-välilehden alta. Tuplaklikataan kuljetinta jolloin se ilmestyy mallinnustilan origoon. Kirjastosta voit myös raahata hiirellä komponentteja mallinnettavaan maailmaan.



Kuvio 29. Maailman luonti kirjastojen avulla

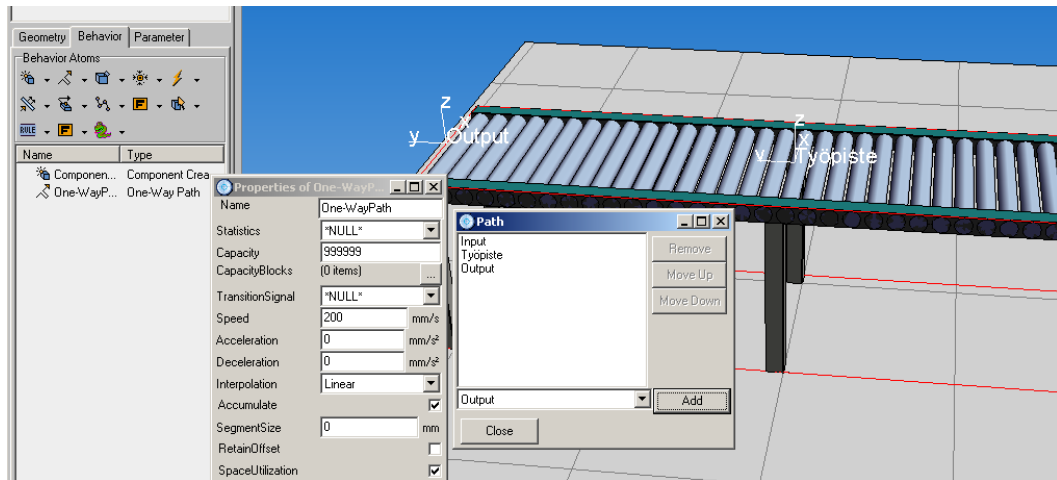
Luodaan selkeyden vuoksi malli jossa on yksi rullakuljetin. Luodaan rullakuljettiin Create – Behaviour välilehden alta Component Creator. Asetukset määritellään myöhemmin. Seuraavaksi luodaan One-WayPath.



Kuvio 30. Reitti komponentin luonti

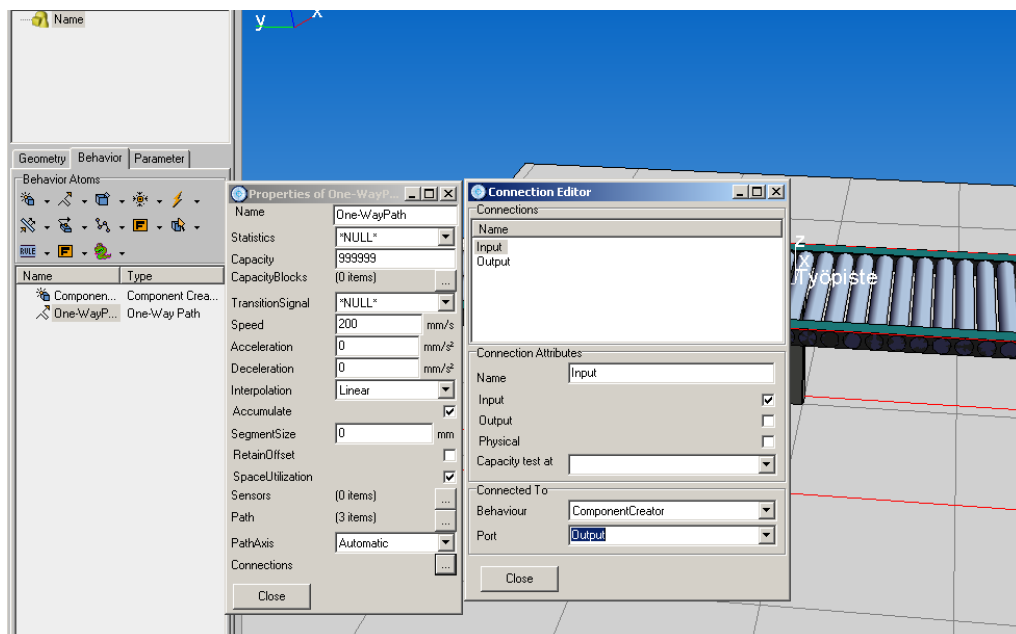


Painetaan One-WayPath ikkunassa Path-kohdan nappia. Valitaan valikosta järjestyksessä pisteet Input, Työpiste, Output ja jokaisen jälkeen painetaan Add-nappia.



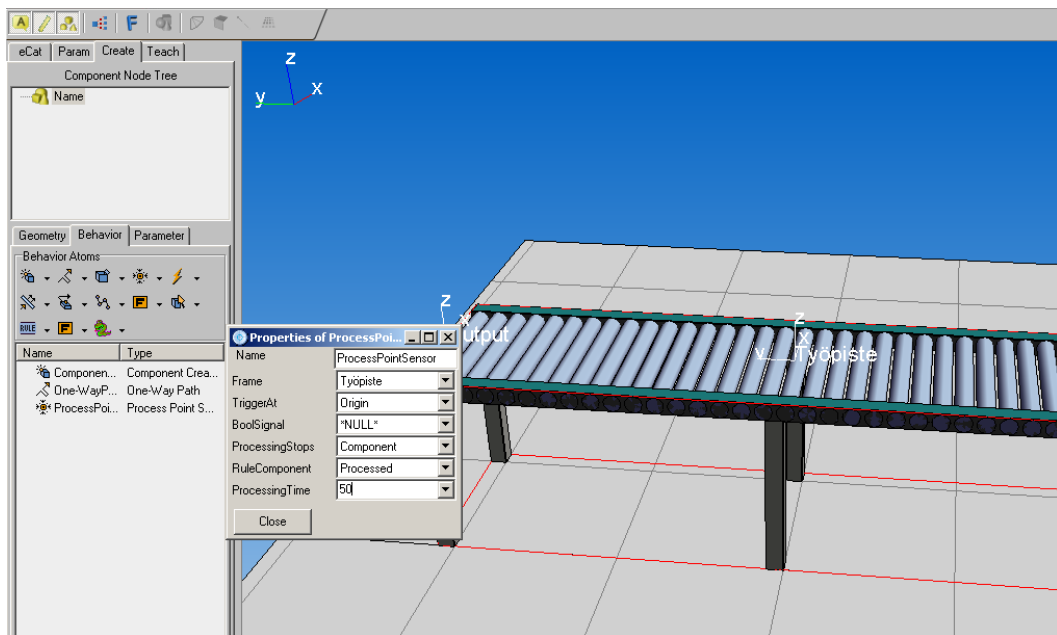
Kuvio 31. Reitin määrittely

Tämän jälkeen suljetaan Path-ikkuna ja avataan One-WayPath-ikkunasta Connections-osio. Connections-osion alta valitaan ensin Input jonka jälkeen alhaalla olevassa Connected To-ikkunassa on Behaviour-/ ja Port-kohdat. Behaviour-kohtaan valitaan listasta ComponentCreator ja Port-kohtaan valitaan listasta Output, eli Component Creatorin ulostulo. Tämän jälkeen suljetaan One-WayPath-ikkuna.



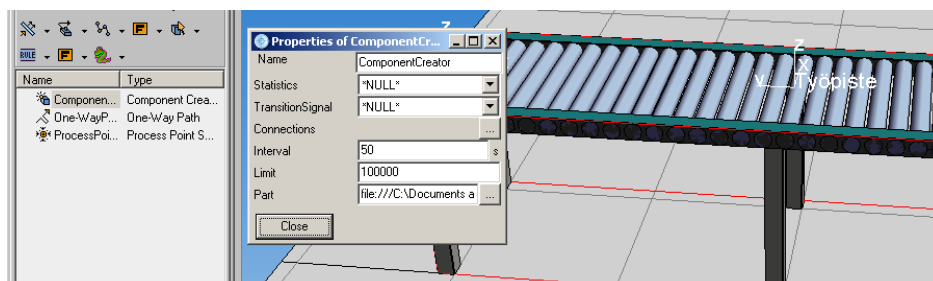
Kuvio 32. Reitin linkitys

Seuraavaksi valitaan Behaviour Atoms-ikkunan alta ProcessPointSensor, johon määrittelemme työstämiseen käytetyn ajan parametrit. ProcessPointSensor ikkunan alle määrittelemme Frame-kohtaan Työpiste, TriggerAt-kohtaan Origin, ProcessingStops-kohtaan Component ja ProcessingTime-kohtaan 50. Annetut parametrit pysäyttävät kappaleen, kun sen origo saavuttaa Framen ”Työpiste” ja pitää kappaletta pysäytettynä 50 sekuntia. Tämän jälkeen komponentti jatkaa normaalisti matkaa.



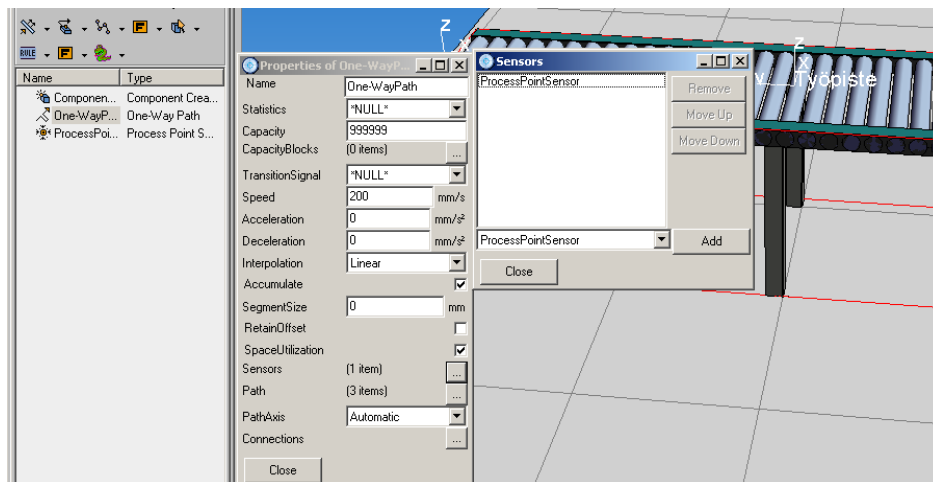
Kuvio 33. Prosessin parametointi

Tämän jälkeen avaamme ComponentCreator-työkalun, määrittelemme Interval-kohtaan kappaleiden luontitahdin, sekä Part-kohdasta voimme valita kirjastosta materiaalin jota komponentti luo.



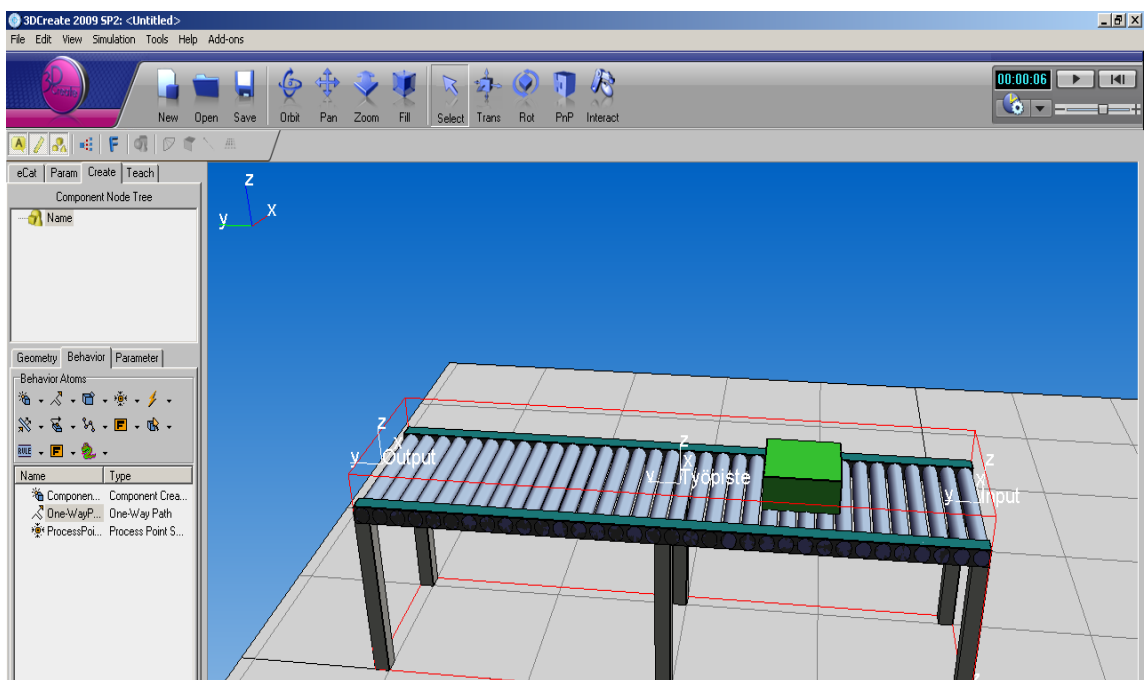
Kuvio 34. Kappaleen luonnin määrittely

Lopuksi avaamme vielä One-WayPath-komponentin ja sieltä Sensors-valikko, jonka alta valitsemme listasta ProcessPointSensor ja painamme Add-nappia.



Kuvio 35. Määritellään reitille sensori

Tämän jälkeen voimme sulkea One-WayPath-ikkunan ja ajaa simulaatio painamalla oikeasta yläkulmasta toistonäppäintä.



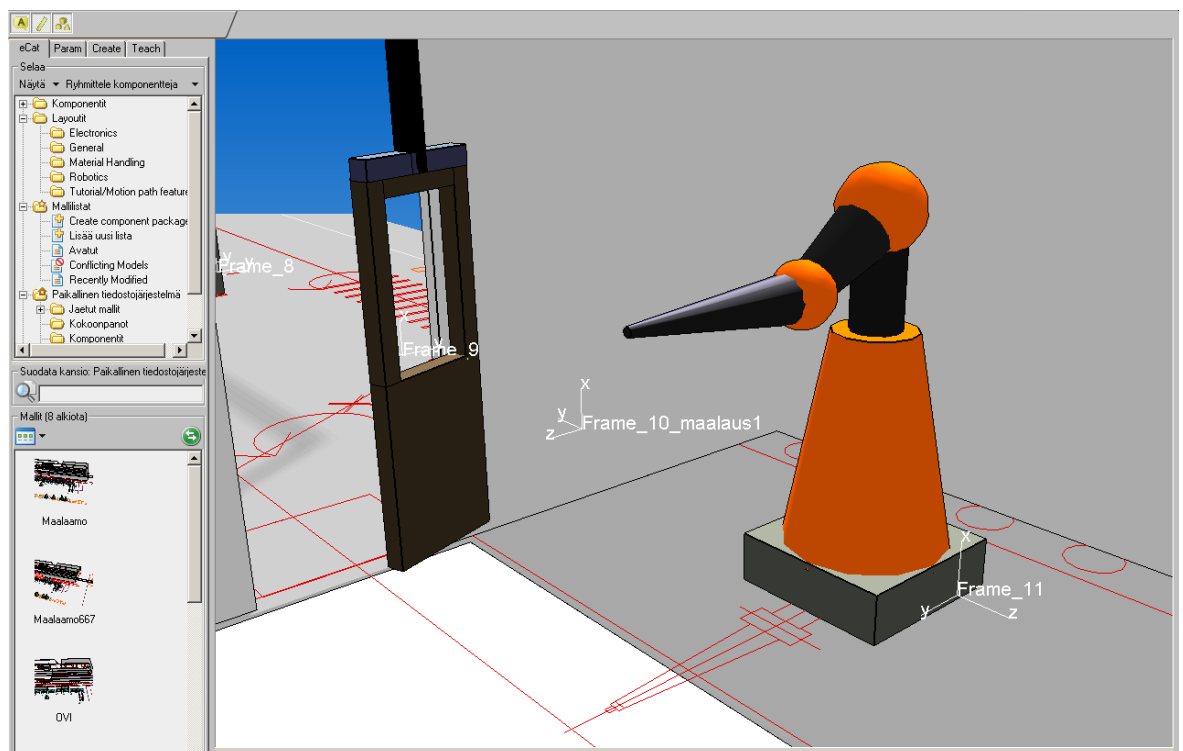
Kuvio 36. Simulaation ajaminen

## 4.6 Sääntöjen ja kinematiikan luominen 3DCreatella

Erilaisten materiaalivirtojen kulkuun liittyvien sääntöjen tekemiseen on valmiita työkaluja, mutta vaativimpiin ratkaisuihin vaaditaan Python-koodia, jonka avulla määritellään tarvittavat parametrit näkyviin muutoksiin, sekä simulaatiossa näkyvämmiksi osiin.

### 4.6.1 Esimerkki 1: Ovilehden maalaus

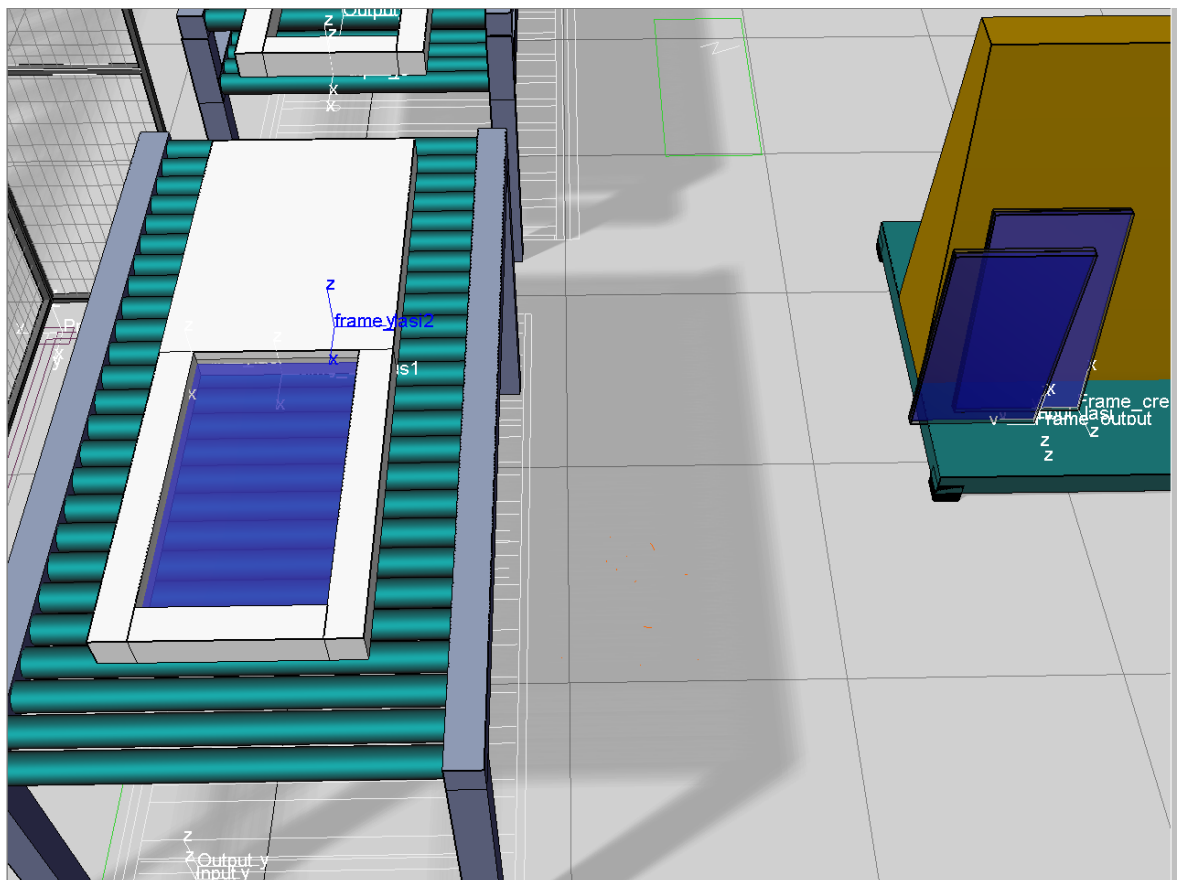
Ovilehden maalaukseen täytyy tehdä valmisteluja jo ovilehteä mallintaessa. Ovilehden tehdään muuttuja, joka on joko päällä tai poissa. Seuraavaksi tehdään ovilehden switch-komponentti, jonka sisälle säilytetään ovilehdestä sekä maalamatonta että maalattua versiota. Ovilehden tullessa maalaamoon ovilehti saapuu maalaukseen, missä maalaus-signaali antaa arvon ovilehden muuttujalle. Ovilehden switch-komponentti tarkkailee muuttujan arvoa ja vaihtaa maalattuun versioon ovilehdestä muuttujan vaihtuessa. (Liite 2).



Kuvio 37. 3DCreatella mallinnettu maalaukspiste

#### 4.6.2 Esimerkki 2: Ovilehden lasitus.

Ovilehden lasitusta ohjaavassa Python-scriptissä odotetaan ovilehden signaalia, eli saapumista lasituspisteelle. Tällöin if-lause käynnistää tapahtuman, jossa lasielementti kulkee reittiään oikeaan kohtaan, jolloin ovilehteen määrätyn säilytystilaan eli containeriin siirretään ikkuna-elementti. Kun ovilehti ja lasielementti ovat halutuilla paikoilla, asetetaan haluttu viive, joka kuvastaa työhön kuluvaan aikaan, jonka jälkeen kokoonpano jatkaa ovilehdelle määritettyä reittiä. (Liite 1).

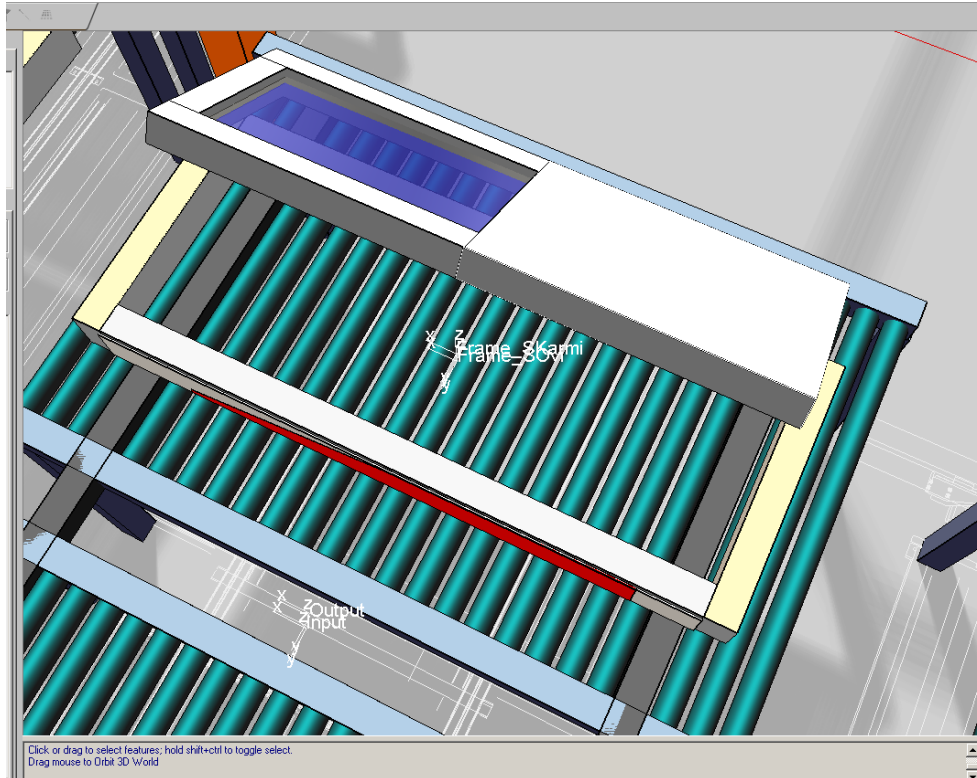


Kuvio 38. 3DCreatella mallinnettu lasituspiste

#### 4.6.3 Esimerkki 3: Alikokoonpanojen yhdistäminen

Kolmas esimerkki on hieman samanlainen kuin esimerkki 2, mutta tässä työpisteelle yhdistyvät kaksi eri alikokoonpanoa. Esimerkissä aiemmin kasatut oven karmit tulevat työpisteelle odottamaan ovilehteä. Ovilehden saapuessa ovilehti nostetaan karmien yläpuolelle ja asennetaan karmeihin yläkautta. Ovilehden olles-

sa paikoillaan, karmit kiinnitetään oven *component containeriin*, käsitellään työhön liittyvä prosessointiaika ja tuote jatkaa matkaa. Signaalit toimivat tässä esimerkiksi samalla periaatteella kuin edellisessäkin, hieman vain eri järjestyksessä.

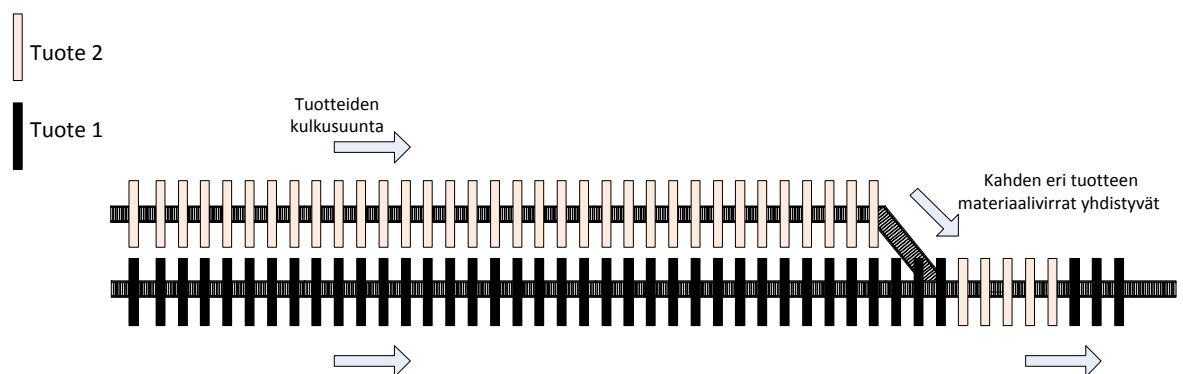


Kuvio 39. 3DCreatella mallinnettu kokoonpanopiste

## 5 TULOKSET

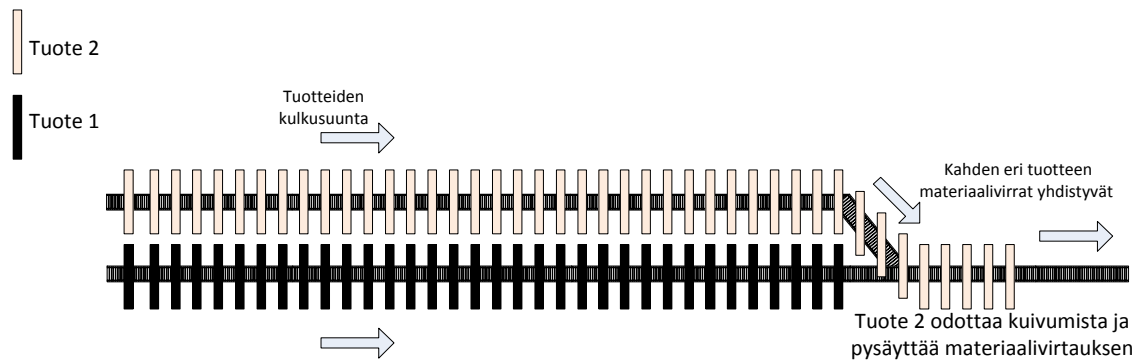
Simulaatiomallin ollessa hyvin suurpiirteinen tuotantoparametreiltaan, ovat myös tulokset suuntaa antavia. Työn lopullisena tavoitteena oli alustavasti suunnitella ja simuloida malli, jonka avulla voidaan tuotannonohjausta muuttaa siten, että maalauslinjan läpi voitaisiin ajaa materiaali suoraan kokoonpanolinjastolle ilman ovilehtien nostelua erilliseen välivarastoon. Lean-mallin mukaan yksinkertaistetaan valmistusprosessia karsimalla tuotannosta lisäarvoa tuottamatonta työtä.

Alkuperäisenä suunnitelmana oli pyrkiä ajamaan kahdet eri tuotteet samoja reittejä ajoittamalla vain tuotteiden valmistussuunnitelmat niin, että tuotteet eivät muodostaisi tukoksia materiaalivirtaan. Näin maalauslinjasto ei olisi vaatinut erillisiä investointeja, vaan välivaraston poisto olisi onnistunut puhtaasti valmistussuunnitelman optimoinnin kautta.



Kuvio 40. Kahden eri tuotteen materiaalivirran yhdistyminen

Tuote nro 2 vaatii oman raiteen missä kuivua, koska niillä on pidempi kuivumisaika. Simulaation sekä tarkempien tutkimusten jälkeen ilmeni, että maalauslinjaston kapasiteetti nykyisellä asetuksella ei riitä sekä tuotteen 1 että tuotteen 2 yhtäaikaan kuivattamiseen. Näin ollen mikäli tuotteet 2 ja tuotteet 1 laitettaisiin samaa reittiä, pysäyttäisivät tuotteet 2 materiaalivirran. (Kuva 22).

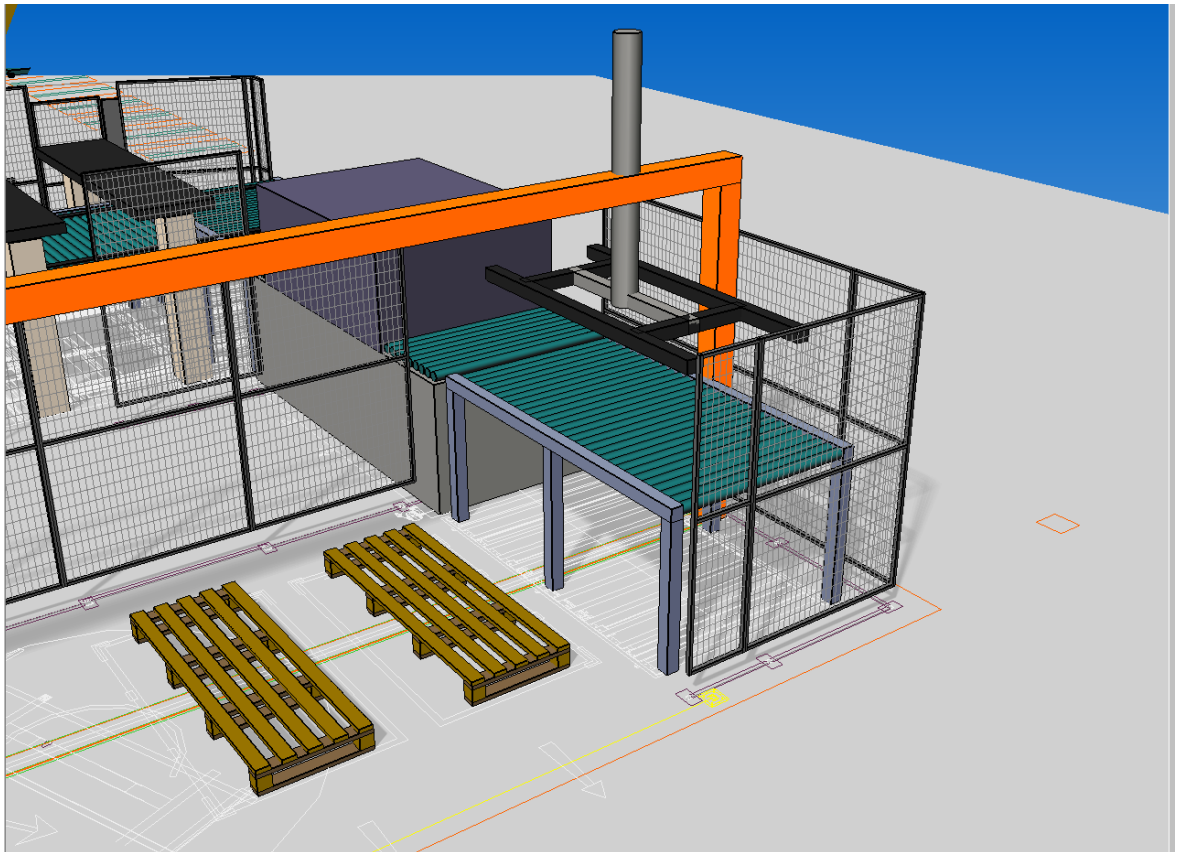


Kuvio 41. materiaalivirtojen yhdistämisen toteutuminen käytännössä

Ratkaisuna syntyi maalaukselinjaston jonotuskapasiteetin lisäys. Todettiin että koska tuote 2 on mahdoton kuivattaa vaadittua aikaa maalaukselinjastolla, ilman että muu tuotanto häiriintyy, täytyy maalaukselinjastoon lisätä jonotuskapasiteettia. Tämä lisäkapasiteetti muodostaa ylimääräisen lenkin maalaukselinjastoon, jolloin tuotteen 2 pitkäkestoisempi kuivumisaika ei aiheuta pullonkaulaa materiaalivirtaan ja sekä tuote 1 että tuotetta 2 voidaan nostaa suoraan maalaukselinjastolta linjalle. Tuotannonohjauksen analogia Lean management -mallin kanssa toteutuu tarkoituksenmukaisesti.

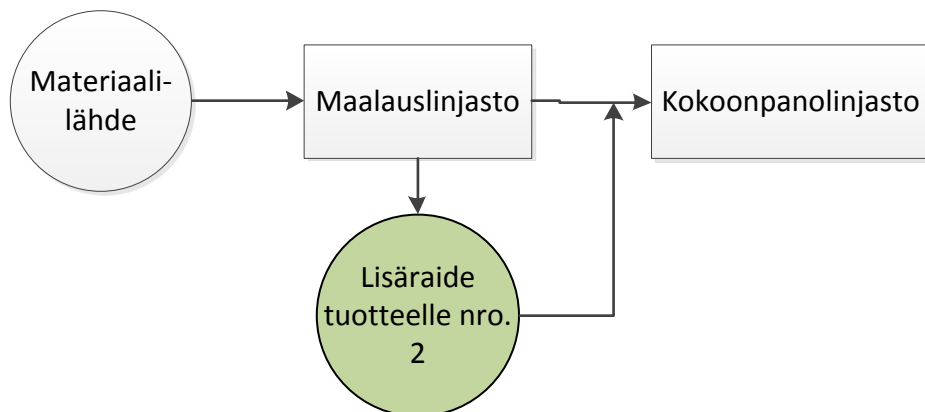
Simulaatiomallin oikeellisuutta tarkkailtiin jatkuvasti sen viikon aikana tuottaman materiaalin mukaan. Kohdeyritykseltä saatiin tuotteiden läpäisyajat, sekä tieto kuinka suuri määrä materiaalia kokoonpanolinjaston läpi menee viikon aikana. Malli rakennettiin toimimaan niin, että se joka vuorokaudessa pyörittää tuotantoa kahden 8 tunnin vuoron ajan, ja yöksi linja tyhjenetään. Materiaalikulun tarkkailupisteinä käytetään kuvassa 23 olevaa pakkausseinän jälkeistä rullakuljetinta.





Kuvio 42. Kokoonpanolinjaston viimeinen piste

Rullakuljettimeen on määritelty ohjelman oma statistics-komponentti, joka on ohjelmoitu näyttämään läpi kulkevan materiaalin määrän. Statistics-komponentti voi näyttää myös muita tietoja pisteestä, esim. korjaamiseen käytetty aika, työhön käytetty aika, odottamiseen käytetty aika sekä käynnistämiseen käytetty aika. Simulaatiossa erikseen määritelty osuus tuotetuista tavaroista tulee valmiina tähän pisteeseen, joten tästä on helppo arvioida vastaako simulaatio todellisuutta.



Kuvio 43. Lopullinen ratkaisu simulaatiossa

## 6 POHDINTA

### 6.1 Työn aihe

Työ aiheena on erittäin haastava. Työ aloitettiin liian laajalta alueelta, ilman kunnan rajausta simuloitavasta alueesta. Alustava suunnitelmassa oli tarkoituksena simuloida koko tuotanto ja etsiä sitä kautta pullonkohdat, mutta pelkästään koko tuotannon simulointi olisi ollut mallina niin laaja, että siitä oltaisiin jouduttu mallin raskauden takia tekemään liian yksinkertaistettu, jolloin mallista olisi mahdollisesti tullut liian epäluotettava. Työn kannalta oli välttämätöntä perehtyä tuotannonohjaukseen, sekä siihen liittyvään teoriaan. Työssä korostuivat simuloitavan aiheen rajaamisen ja fokuksinnin tärkeys, sekä taustatietojen kunnollinen hankinta ennen simulaation rakentamista.

### 6.2 Simulaatio

Simulaatio-ohjelmistoina vaihtoehtoina olivat Delmian Quest tai Visual Componentsin 3Dcreate. Visual Componentsin 3DCreateen päädyttiin laajempien mahdollisuuksien, sekä käyttäjälle luontevamman tuntuisen käyttöliittymän vuoksi. Visual Components tarjoaa myös ilmaisen 3Dvideo-ohjelman, jolla simulaatiomalleja voidaan toistaa. Perustoiminnot 3DCreatella olivat todella helppoja tehdä, sekä mallinnus oli todella yksinkertaista. 3DCreateen vahvuuksia kuitenkin ovat sen monipuolisuus. Python-koodin ansiosta ohjelmalla pystyy luomaan erittäin monimutkaisia ja monipuolisia järjestelmiä. Toisaalta Python-koodi on myös todella rajoittava tekijä sellaiselle, joka ei kyseistä kieltä ole koskaan käyttänyt. Pienimmätkin erikoisuudet materiaalivirrassa vaativat Python-koodia, jolloin kieltä osaamaton ei ilman apua saa mallia toimimaan oikein. Python ohjelmointikielenä on onneksi kuitenkin erittäin helppo ja yksinkertainen kieli oppia.

Itse simulaatiomallia jouduttiin lopussa karsimaan, koska koko tehtaan toiminnot olivat niin raskaat koneen laskettavaksi, ettei malli jaksanut enään pyöriä kunnolla.

Monet alikokoonpanot korvattiin pelkillä component creatoreilla, jotka loivat valmiita komponentteja materiaalireitille. Tärkeintä oli kuitenkin keskittyä maalauslinjaston ja kokoonpanolinjaston realistiseen toimintaan. Toiminta saatiinkin vastamaan melko lähelle todellista toimintaa, mutta tiukan aikataulun takia, kompromisseja jouduttiin tekemään.

### **6.3 Tulokset**

Tuloksia tiukan aikataulun vuoksi ei ehditty kokeilemaan käytännössä, koska se vaatii yritykseltä investointeja ja layoutin muuttamista. Tiukan aikataulun vuoksi kaikkea ei ehditty simulaatioon mukaan ottamaan, joten käytännön toimivuudesta ei voi varmasti simulaatiomallin perusteella tehdä päätöstä. Mallissa kuitenkin yritettiin ottaa simulaatioon mukaan ne tärkeimmät ongelmaan vaikuttavat komponentit. Huomioon täytyy ottaa myös, että työvaiheissa käytetyt läpimenoajat ovat tiedonkeruuvaiheessa olleet yleistettyjä keskiarvoja, joten puolenvuoden/ vuoden vanhoja läpimenoaikoja kannattaa myös tulkita kriittisesti. Ratkaisu lähtöongelmaan silti saatiin, joten tavoitteeseen simuloitavassa aiheessa päästiin. Simulaatiomalli on parametroitu siten, että sillä on helppo tehdä jatkotutkimusta kyseiseen projektiin. Simulaatiossa mallinnettiin myös paljon muuta, mikä loppusimulaatiossa jätettiin pois mallista, joten tehtaan muidenkin toimintojen simuloiminen on valmiin kirjaston vuoksi helppoa.

## 7 LÄHTEET

- Haverila, M., Uusi-Rauva, E., Kouri, I. & Miettinen, A. 2005. Teollisuustalous. Tampere: Infacs Oy
- Saari, H. & Oijennus, M. 2004. Toiminnanohjaus kehityskohteena pk-yrityksessä. Seinäjoki: Seinäjoen ammattikorkeakoulu
- Lapinleimu, I. 2001. Ideaalitehdas. Tampere: Tampereen teknillinen korkeakoulu
- Tiainen, J. 1996. JOT – tie tulevaisuuteen ja menestykseen. Kuhmo: Kuhmon yrityssampo
- Miettinen, P. 1993. Tuotannonohjaus ja logistiikka. Helsinki: ATK-instituutti
- Visual Components. 2010. [www-lähde]. About us. [Viitattu 15.02.2011]. Saatavissa: <http://www.visualcomponents.com/Company/AboutUs>
- Skaala. 2010. [www-lähde]. Yritys. [Viitattu 13.02.2011]. Saatavissa: <http://www.skaala.com/skaala.html>
- Sawka Enterprises. 2010. [www-lähde]. What Is Lean. [Viitattu 15.02.2011]. Saatavissa: <http://www.sawkaenterprises.com/images/7wastes.jpg>
- Dassault Systemes. 2011. [www-lähde]. Delmia V5R20. [Viitattu 18.02.2011]. Saatavissa: <http://www.3ds.com/products/delmia/portfolio/delmia-v5/v5r20/>
- Flexsim. 2011. [www-lähde]. Flexsim Simulation Software – Overview. [Viitattu 18.02.2011]. Saatavissa: <http://www.flexsim.com/products/flexsim/>
- 3DCreate 2009. [Tietokone ohjelma]. Visual Components Oy. 3DCreate Version 2009 SP2

## **8 LIITTEET**

Liite 1: Python-scripti lasitusvaiheesta

Liite 2: Python-scripti maalaamosta

## Liite 1: Python-scripti lasituksesta

```
from vcScript import *

def OnSignal( signal ):
    global ovi, ikkuna
    comp = GetComponent()

    print signal.Name, signal.Value
    signal1 = comp.findBehaviour("Signal1")
    signal2 = comp.findBehaviour("Signal2")
    signal3 = comp.findBehaviour("Signal3")

    if signal == signal1:
        signal1.Value.stopMovement()
        ovi = 1
        print ovi,ikkuna
    if signal == signal3:
        signal3.Value.stopMovement()
        ikkuna = 0
        print ovi,ikkuna
    if signal == signal1:
        signal3.Value.startMovement()
        ikkuna = 0
        print ovi,ikkuna
    if signal == signal2:
        signal2.Value.stopMovement()
        ikkuna = 1
        print ovi,ikkuna

def OnRun():
    global ovi, ikkuna
```

```
comp = GetComponent()
signal1 = comp.findBehaviour("Signal1")
signal2 = comp.findBehaviour("Signal2")
signal3 = comp.findBehaviour("Signal3")

ovi = 0
ikkuna = 0

while True:
    while (ovi == 0 or ikkuna == 0):
        delay(1)
        print "Lasitus OK"
        signal1.Value.findBehaviour("Container").grab(signal2.Value)
        delay( # työhön käytettävä aika )
        signal1.Value.startMovement()
        ovi = 0
        ikkuna = 0
```

Liite 2: Python-scripti maalaamosta

```
from vcScript import *

def OnSignal( signal ):
    comp = GetComponent()
    signaali = comp.findBehaviour("Signal")

    if signal == signaali:
        signal.Value.Maalattu = 1

def OnRun():
    pass
```