



Tuotekonfiguraattorin luominen konseptisuunnittelua varten

Ammattikorkeakoulun opinnäytetyö

Konetekniikka, insinööri (AMK)

Kevät 2025

Miikka Lainio

Koulutus	Konetekniikka, insinööri (AMK)
Tekijä	Miikka Lainio
Työn nimi	Tuotekonfiguraattorin luominen konseptisuunnittelua varten
Ohjaaja	Teppo Syrjänaho

Vuosi 2025

Tämän opinnäytetyön ydin on 3D-suunnitteluympäristöön luodussa tuotekonfiguraattorissa. Vaikkakaan opinnäytetyötä ei suoritettu millekään yritykselle, on sen inspiraatio kuitenkin syntynyt työelämän esimerkeistä. Työelämässä olen päässyt vierestä seuraamaan 3D-suunnittelijoiden tekemää konseptisuunnittelua, joka on pääsääntöisesti vaikuttanut työläältä jatkuvien pienten muutosten takia. Muutokset ovat useasti olleet pääkomponenttien mittoihin tai sijoitteluun liittyviä. Tämä sai minut kehittämään tuotekonfiguraattoria konseptisuunnittelun tueksi.

Opinnäytetyö alkaa kartoittamalla ja selvittämällä mikä tuotekonfiguraattori varsinaisesti on, mitä sillä voidaan tehdä ja mitä sillä saavutetaan. Sen jälkeen perehdytään parametrusten 3D-mallien perusteisiin ja miten ne toimivat. Lisäksi havainnollisesta saavutettuja etuja parametrusten ja ei-parametrusten mallien välillä. Näiden jälkeen esitellään käyttöön valikoituneen suunnittelutyökalun sisäinen ohjelmointikieli ja -ympäristö. Kyseistä ohjelmointikieltä hyödynnetään mallin ja konfiguraattorin toiminnassa.

Työn loppupuoliskolla kerrotaan tarkemmin ja pintaa syvemmin, miten rakennettu konfiguraattori ja sen ohjaamat 3D-mallit varsinaisesti toimivat koodi- ja parametrisella. Myös kohdatuista virheistä tai haasteista koskien luotujen mallien parametrusta toimintaa sekä niiden korjauskeinoista on kerrottu. Opinnäytetyö päättyy yksinkertaiseen vertailuun, jossa otettiin aikaa kuinka kauan ennalta määriteltyjen muutoksen sisään ajaminen kestää niin parametriseen, kuin ei-parametriseen malliin. Tällä pyrittiin havainnoimaan opinnäytetyön alussa esitettyjä väitteitä tuotekonfiguraattorin hyötyjen suhteen.

Avainsanat Tuotekonfiguraattori, parametrisuus, 3D-suunnittelu, iLogic
Sivut 23 sivua ja liitteitä 0 sivua

DP Mechanical Engineering
Author Miikka Lainio
Subject Creation of a Product Configurator for Concept Design
Supervisors Teppo Syrjänaho

Year 2025

This thesis' main focus is on a product configurator which was created in a 3D-environment. Even though this thesis was not done for a company, its inspiration from a work-life example. In my work I have been able to witness the conceptual design work of 3D-designers, which has mainly seemed challenging due to constantly ongoing minor changes. The changes mentioned are usually related to the sizes or locations of the main components. This led me to start to develop a product configurator for concept design.

The thesis begins with an investigation and clarification of what is a product configurator, what can be done with it and what can be accomplished with it. After that, the basics of parametric 3D-models are presented with how they function. Also, what is gained with parametric models versus non-parametric models is visualized. After these topics the selected design tool and its internal programming language and environment are presented. This programming language is utilized within the operation of the model and the configurator.

At the latter half of the thesis presents a more detailed view of how the built configurator and the 3D-models it controls work on code and on parametric level. Also, errors or challenges encountered regarding the operation of created parametric models and their mitigation have been presented. The thesis concludes with a simple comparison to see how much time is needed to implement predefined changes in a parametric and non-parametric model. This was done to observe the claims made at the beginning of the thesis regarding the benefits of a product configurator.

Keywords Product configurator, parametricity, 3D-design, iLogic
Pages 23 pages

Sisällys

1	Tuotekonfiguraattorit ja niiden hyödyt.....	3
2	Käytetyt työkalut ja ominaisuudet.....	4
2.1	Parametriset 3D-mallit	4
2.2	Parametrien luominen ja asettele Inventorissa.....	8
2.3	iLogic-ohjelmointi	10
2.4	Sääntöohjelmointi	10
2.5	Tekoälyn hyödyntäminen koodauksessa.....	11
2.6	Forms-työkalu	12
3	Valmis ohjelma.....	14
3.1	Laitetila	16
3.2	Säiliöt.....	18
3.3	Parametrien tuonti pääkokoontasolle	19
3.4	Virheilmoitusten huomiointi koodilla	20
4	Vertailu	20
4.1	Mallien lähtökohta ja niiden muutokset.....	20
4.2	Vertailun tulokset	21
5	Pohdinta.....	22
	Lähteet	23
	Kuva 1. Lähtökohta.....	6
	Kuva 2. Suorakulmion leveyden muuttaminen	7
	Kuva 3. Nurkasta nurkkaan-mitan muuttaminen.....	8
	Kuva 4. Numeerisen parametrin yksikön valintaikkuna	9
	Kuva 5. Parametrien hallintaikkuna.....	10
	Kuva 6. iLogic-ohjelmointiympäristö.....	11
	Kuva 7. Monirivisen ponnahdusikkunan syntaksin kysely tekoälyltä.....	12
	Kuva 8. Forms-työkalu	13
	Kuva 9. Forms-työkalulla luotu valintaikkuna	14
	Kuva 10. Lopullinen kokoonpano	15
	Kuva 11. Toinen versio lopullisesta kokoonpanosta.....	15
	Kuva 12. Säiliöiden Konfigurointi-välilehti.....	18

Johdanto

Laadukkaan esisuunnittelun ja teknisen materiaalin tuottaminen erityyppisten tuotteiden sekä laitosten tarjouskilpailuvaiheessa korostuu nykypäivänä, koska sen avulla toimittaja kykenee ilmaisemaan mahdolliselle tilaajalle visuaalisella ja luotettavalla tavalla toimituskokonaisuutensa toiminnasta. Useasti kuitenkin näihin käytetyt resurssit ja työaika eivät realisoitu liikevaihtona, vaan pelkästään kuluina, jos kauppoja ei saada solmittua. Vaikka haluttuun toimitussopimukseen päästäisiinkin, kasvattaa se kuitenkin toimitettavan asian kustannuksia.

Syntyneitä kustannuksia pyritään minimoimaan monella eri tapaa, yksi näistä tavoista on tässä opinnäytetyössä käsiteltävä tuotekonfiguraattori konseptisuunnitteluvaiheen tueksi. Nopeuttamalla 3D-ympäristössä tehtyä laitostason suunnittelua voidaan säästää merkittäviä kustannuksia ja näin myös tuotteen hinnoittelua voidaan parantaa kilpailukykyisemmäksi. Tuotekonfiguraattori joudutaan rakentamaan ja räätälöimään alusta alkaen käyttäjän tarpeita vastaamaan.

Tässä opinnäytetyössä paneudutaan alkuun tuotekonfiguraattoreihin yleisellä tasolla havainnoiden eri tyyliä konfiguraattoreita, sekä tuodaan ilmi taloudelliset ja laadulliset hyödyt. Opinnäytetyön ydin on kuitenkin luotavassa konfiguraattorissa, joka rakennettiin 3D-suunnittelu-ympäristöön hyödyntäen ohjelman sisäistä ohjelmointikieltä ja käyttöliittymän rakennustyökalua.

1 Tuotekonfiguraattorit ja niiden hyödyt

Tuotekonfiguraattorit ovat työkaluja, joilla mahdollistetaan käyttäjän tarpeen mukainen tuotteen muunneltavuus noudattaen ennalta määrättyjen parametrien raja-arvojen sisällä. Konfiguroitavan massaräätälöityjen tuotteiden tuotantomäärä Martion mukaan olisi tyypillisesti vähintään sata kappaletta, jotta tuotekonfiguraattorin hankinnalle olisi taloudelliset perusteet. Myös monimutkaisimmille konfiguroitaville tuotteille voi myös olla olemassa tuotekonfiguraattoreita. Tällaisia on esimerkiksi laitosten tarjousprosessiin liittyvät tuotekonfiguraattorit, joka tämänkin opinnäytetyön lopputuloksena valmistui. (Martio, 2015, s. 19)

Konfiguraattorilla voidaan helpottaa myös tuotetiedonhallintaa. Niihin voidaan rakentaa sääntöpohjaisia rajoitteita, jolloin käyttäjä ei voi luoda semmoista konfiguraatiota, mitä yritys ei pysty tai halua valmistaa. Kun konfiguraattori luodaan oikein, saadaan aikaiseksi suoraviivaisempi ja tehokkaampi prosessi tuotteen räätälöintiä varten. Konfiguraattoreiden avulla voidaan esimerkiksi generoida osaluettelot, piirustukset ja käyttöohjeet. (Pieper, 2022)

Seuraavissa kappaleissa on listattuna tämän opinnäytetyön konfiguraattoriin verrattavia konfiguraattorityyppejä. Lisäksi esitellään niille tyypilliset ominaisuudet.

Parametrisellä konfiguraattorilla tarkoitetaan semmoista konfiguraattoria, missä käyttäjä voi syötteen avulla ohjata generoitua tuotetta ja se tuottaa lähes reaaliaikaisesti 3D-kuvan käyttäjälle. Ohjattavat parametrit voivat olla esimerkiksi pituus ja korkeus, väri, materiaali. (Houston, 2020)

CPQ- (Configuration, Price, Quote), eli myyntikonfiguraattorit helpottavat myyntityötä tekevien henkilöiden työtä. Niiden avulla voidaan taata tuotteiden eri komponenttien yhteensopivuudet. Jos tuote koostuu useammasta eri ominaisuudesta tai osasta, konfiguraattoriin voidaan määrittää toivottujen ominaisuuksien tuomat lisäkustannukset. Myyntikonfiguraattorilla luodaan myös nopeasti ja yksinkertaisesti tarjous asiakkaalle. (Marquardt, n.d.)

Myyntikonfiguraattorilla voidaan vaikuttaa merkittävästi käytettyyn aikaan tarjouslaskentaprosessissa. Joskus nämä prosessit saattavat olla hyvinkin monimutkaisia kokonaisuuksia, joiden hallintaa konfiguraattori voi helpottaa.

Myyntityöhön käytetyn työtuntien määrän pienentäminen auttaa parantamaan mahdollisesta saavutetun myynnin katetta. Näiden mallit eroavat monesti lopullisista malleista tai osarakenteista, mutta niiden toiminnallisuus on enemmänkin havainnoida myytävää tuotetta loppuasiakkaalle.

Konfiguraattoreiden parametrit ovat myös sidonnaisia tai riippuvaisia toisistaan. Oikeaoppisella parametrimäärittelyllä estetään, että toimintakelvotonta tuotetta ei voida määrittää konfiguraattorilla. Martion mukaan tuottavuuden nousu, kun käytetään konfiguraattoria, voi olla jopa tuhat kertainen. (Martio, 2015, ss. 23, 25, 227)

2 Käytetyt työkalut ja ominaisuudet

Tämän opinnäytetyön ydin on luotavassa tuotekonfiguraattorissa. Se luodaan CAD-ohjelman (Computer Aided Design) sisäistä ohjelmointikieltä hyödyksi käyttäen. Käytetty ohjelmisto on Autodeskin Inventor Professional 2024. Vaikkakin konfiguraattori on CAD-ohjelman sisäinen, on se siltikin helppokäyttöinen ja vähäiselläkin CAD-kokemuksella operoitavissa. Konfiguraattori on tarkoitettu konsepti- ja esisuunnittelun tueksi, eikä sillä ole tarkoitus luoda detaljitason suunnittelua tai piirustuksia. Konfiguraattorilla luotua 3D-mallia voi hyödyntää esimerkiksi myynnin tukena myyntitarjouksen liitteeksi tai tarjouspyyntömateriaalina hankintoja varten.

Kyseinen konfiguraattori on rakennettu toimimaan pääkoonpano-tasolla. Käytettävän ohjelmointikielen (iLogic) avulla yksittäisten osien tai alikoonpanojen parametreja ohjataan pääkoonpano-tasolle luodun käyttöliittymän avulla (Forms). Näin eri laitteiden vaikutusta toisiinsa voidaan tutkia reaaliaikaisesti koko laitosalueella. Esimerkkilaitoksena työssä toimii kaasun höyrystyslaitos, joka tässä tapauksessa koostuu yhdestä tai kahdesta muunneltavasta laitetilasta sekä nesteytetyn kaasun varastosäiliöstä tai -säiliöistä.

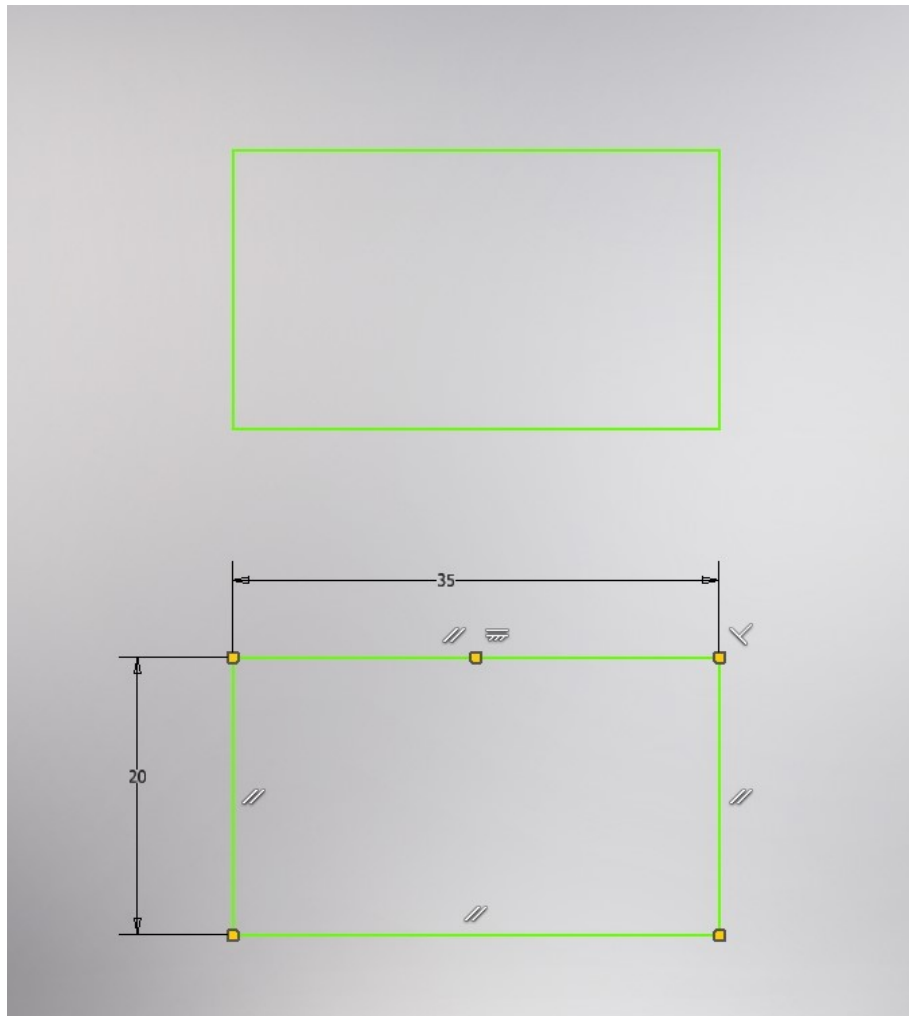
2.1 Parametriset 3D-mallit

Parametrisellä 3D-mallilla tarkoitetaan mallia, jonka piirteitä ja rajoitteita voidaan ohjata parametrin arvoa muuttamalla. Vastakohta tälle on suora mallinnus, jossa geometria luodaan manuaalisesti käyttämällä tai

muokkaamalla piirteitä. Parametreillä voidaan myös luoda sidoksia toisiinsa. Esimerkiksi voidaan määrittää, että parametrin A suuruus on kaksi kertaa suurempi kuin parametrin B arvo. Parametrisuuden suuri etu on joustavuus, piirteen kokoa voidaan helposti muokata vaihtamalla vain parametrin arvoa. (Adobe, 2025) Parametriseen mallintamiseen kuuluu myös vahvasti piirteiden rajoitteiden sekä riippuvaisuuksien määrittäminen toisiinsa. (Autodesk Inc, 2025)

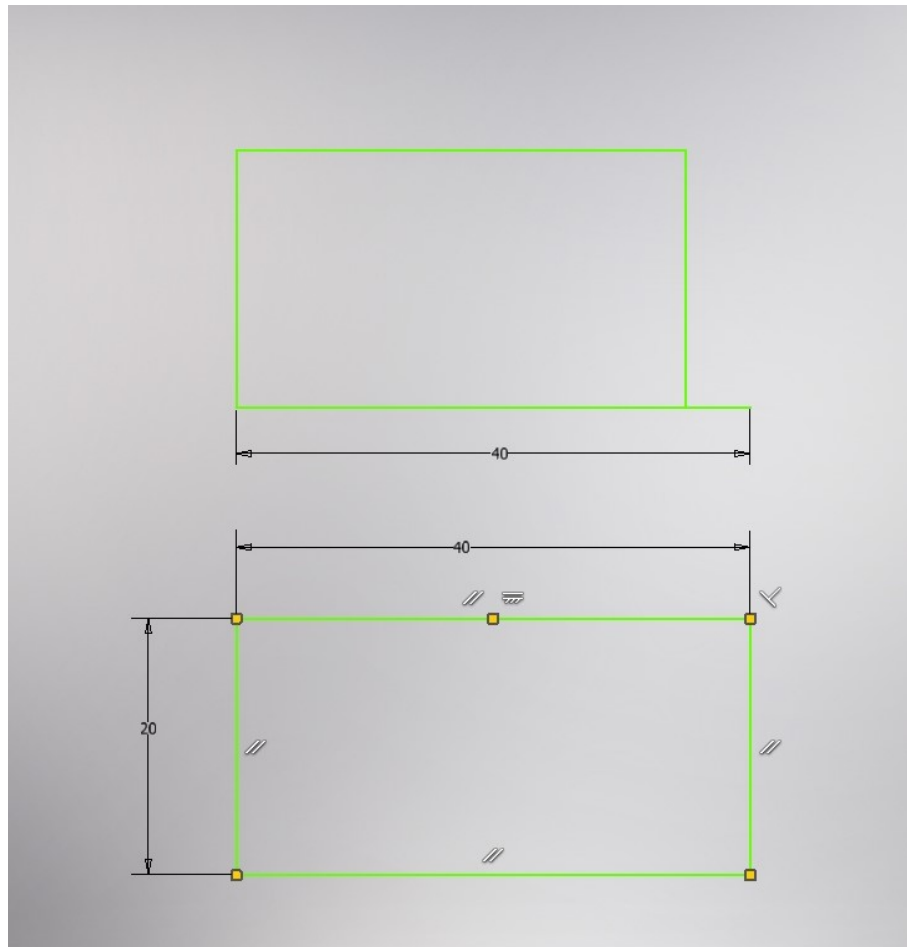
Kuten edellisessä kappaleessa todettiin, parametrisuuden avulla mallin piirteistä saadaan joustavia ja helposti muokattavia. Parametristen rajoitteiden käyttö on havainnoitu yksinkertaisella tavalla seuraavissa kuvissa. Kuva 1 esittää havainnon lähtökohdan, joka on luotu Luonnos-työtilassa (sketch). Kuvassa on kaksi suorakulmiota, joista alempi on luotu parametriseksi kaksipiste suorakulmion avulla. Kyseinen työkalu luo automaattisesti viivojensa rajoitteet, jos yhtäkään kulmaa tai viivaa muokataan, säilyttää se silti muotonsa suorakulmiona. Syntyneet rajoitetyypit ovat yhdensuuntaisuus, vaakasuuntaisuus, sekä kohtisuoruus. Ylempi suorakulmio on kooltaan identtinen, mutta siinä ei ole ollenkaan rajoitteita tai muuta parametrisuutta. Tämä tarkoittaa sitä, että jos jotain pistettä tai viivaa muokataan, suorakulmio ei säilytä muotoansa.

Kuva 1. Lähtökohta



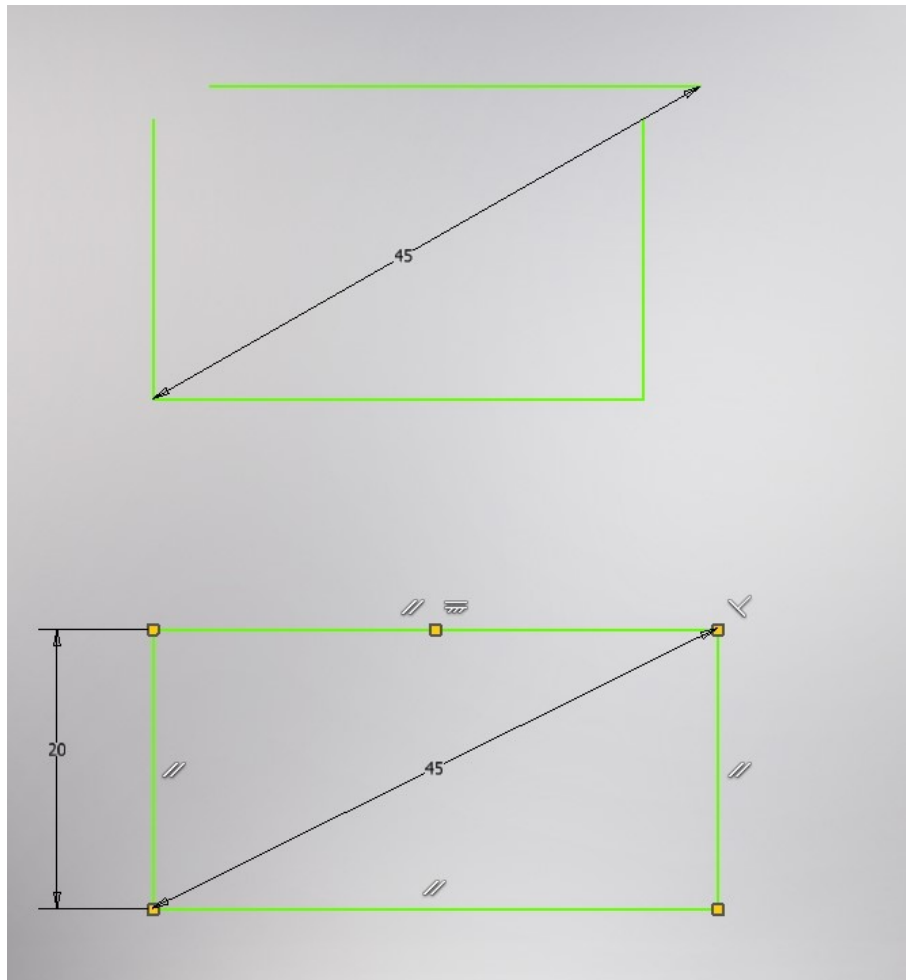
Seuraavassa kuvassa (Kuva 2) esitetään mitä tapahtuu molemmille suorakulmiolle, kun niiden leveyttä muutetaan. Kun suorakulmion jommankumman vaakaviivan pituus muutetaan alkuperäisestä 35 millimetristä 40 millimetriin ylemmän suorakulmion vaakaviiva kasvaa annettuun pituuteen, mutta vastakkainen vaakaviiva pysyy muuttumattomana, eikä oikeanpuoleinen pystyviiva liiku mukana. Alempi, rajoitteilla luotu suorakulmio, sen sijaan säilyttää rajoitteidensa ansiosta geometrisen muotonsa. Kun ylemmän vaakaviivan pituutta on muutettu samat viisi millimetriä, siirtyy oikeanpuoleinen pystyviiva mukana ja vastakkainen vaakaviiva kasvaa pituutta saman verran.

Kuva 2. Suorakulmion leveyden muuttaminen



Vahvistetaan havainnointia vielä toisella esimerkillä. Tässäkin esimerkissä käytetään samaa lähtökohtaa, mutta suorakulmion vaakamitta on korvattu mitalla, joka kulkee nurkasta toisen päädyn vastakkaiseen nurkkaan. Kuva 3 esittää kuinka vapaasti mallinnetun suorakulmion geometria rikkoutuu täysin muutoksen jälkeen, kun taas parametrisempi suorakulmio säilyttää jälleen muotonsa.

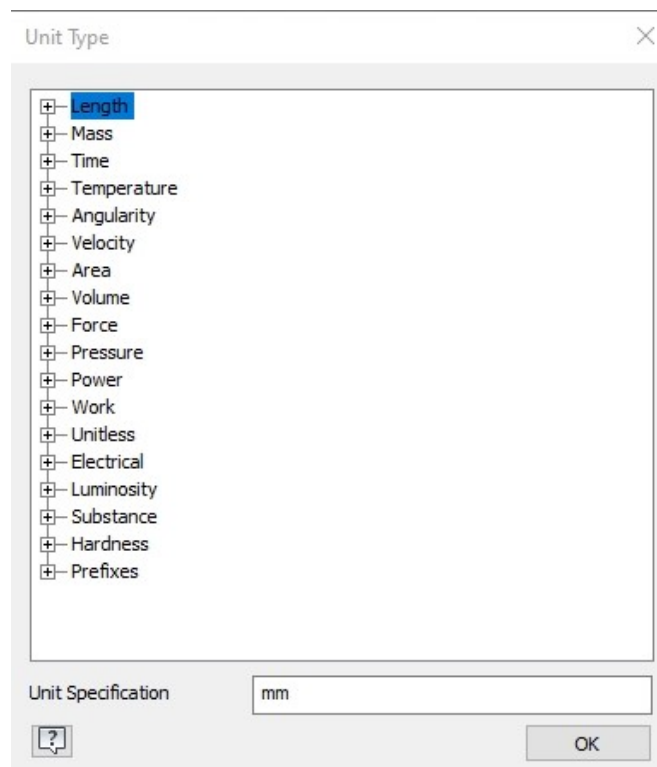
Kuva 3. Nurkasta nurkkaan-mitan muuttaminen



2.2 Parametrien luominen ja asettelu Inventorissa

Autodesk Inventorissa voidaan valita parametrin tyypiksi numeerinen, teksti tai tosi/epätosi. Näistä kolmesta vain numeerinen parametrityyppi tukee lausekkeita. (Autodesk Inc, 2025) Numeerisen parametrin yksikkö voidaan valita hyvin laajasta listasta, joka näkyy Kuva 4. Parametri voi myös olla moniarvoinen, josta kuitenkin vain yksi arvo on kerrallaan käytössä. (Autodesk Inc., 2025) Luodussa konfiguraattorissa käytetään moniarvoisia parametrejä laittilan ovien kanssa, jolla helpotetaan käyttäjän työtä. Kun konfiguraattoriin on esiaseteltu saatavilla olevien ovien koot, ei käyttäjälle jätetä muuta vaihtoehtoa, kuin käyttää niitä tiettyjä mittoja. Tällöin käyttäjän ei tarvitse huolehtia siitä, että onko hänen mitoittamansa ovikoko saatavilla.

Kuva 4. Numeerisen parametrin yksikön valintaikkuna



Inventor luo parametrit automaattisesti jokaiseen piirteeseen tai mitoitettuun viivaan luonnoksessa (sketch). Ohjelma nimeää parametrit alkukirjaimella d, jonka perään tulee juokseva numero (Kuva 5), alkaen arvosta nolla. Juokseva numerointi alkaa nolasta jokaisen uuden tiedoston kohdalla. (Bordino, 2022)

Parametrejä voidaan hallita niitä varten olevan hallintaikkunan avulla (Kuva 5). Työkälun avulla parametrejä voidaan muun muassa etsiä tiettyä parametriä sen nimellä, uudelleen nimetä, muuttaa sen arvoa, lisätä sille tarkentava kommentti. Ne voidaan myös tuoda tai viedä erilliseen taulukkolaskentaohjelmaan. (Autodesk Inc., 2025)

Kuva 5. Parametrien hallintaikkuna

Parameters

Type Keyword...

Parameter Name	Consumed by	Unit/Type	Equation	Nominal Value	Driving Rule	Tolerance	Model Value	Key	Export Parameters	Comment
Model Parameters										
d15	Säiliöiden Pystymeittäus	mm	0,000 mm	0,000000		<Default>	0,000000	<input type="checkbox"/>	<input type="checkbox"/>	
d16	Säiliöiden Vaakameittäus	mm	0,000 mm	0,000000		<Default>	0,000000	<input type="checkbox"/>	<input type="checkbox"/>	
d17	Flush:1	mm	0,000 mm	0,000000		<Default>	0,000000	<input type="checkbox"/>	<input type="checkbox"/>	
d28	Component Pattern 3:1	mm	2,0 mm	2,000000		<Default>	2,000000	<input type="checkbox"/>	<input type="checkbox"/>	
d29	Component Pattern 3:1	mm	Säiliöiden_etiäisyys	5000,000000		<Default>	5000,000000	<input type="checkbox"/>	<input type="checkbox"/>	
d30	Component Pattern 3:1	ul	1 ul	1,000000		<Default>	1,000000	<input type="checkbox"/>	<input type="checkbox"/>	
d31	Component Pattern 3:1	ul	Säilio_QTY	3,000000		<Default>	3,000000	<input type="checkbox"/>	<input type="checkbox"/>	
d32	Säiliöiden kulma	deg	Säilio_rotate	0,000000		<Default>	0,000000	<input type="checkbox"/>	<input type="checkbox"/>	
d35	Laitetila 1 Kulma	deg	0,00000000 deg	0,000000		<Default>	0,000000	<input type="checkbox"/>	<input type="checkbox"/>	
d37	Laitetila 1 X-suunta	mm	15000,00000000 mm	15000,000000		<Default>	15000,000000	<input type="checkbox"/>	<input type="checkbox"/>	
d41	Laitetila 1 Y-suunta	mm	10000,00000000 mm	10000,000000		<Default>	10000,000000	<input type="checkbox"/>	<input type="checkbox"/>	
d42	Laitetila 1 Pohja	mm	0,000 mm	0,000000		<Default>	0,000000	<input type="checkbox"/>	<input type="checkbox"/>	
d43	Mate:11	mm	0,000 mm	0,000000		<Default>	0,000000	<input type="checkbox"/>	<input type="checkbox"/>	
d44	Laitetila 2 Pohja	mm	0,000 mm	0,000000		<Default>	0,000000	<input type="checkbox"/>	<input type="checkbox"/>	
d45	Laitetila 2 X-suunta	mm	Laitetila_2_X	5000,000000		<Default>	5000,000000	<input type="checkbox"/>	<input type="checkbox"/>	
d46	Laitetila 2 Y-suunta	mm	Laitetila_2_Y	10000,000000		<Default>	10000,000000	<input type="checkbox"/>	<input type="checkbox"/>	
d47	Laitetila 2 Kulma	deg	Laitetila_2_angle	0,000000		<Default>	0,000000	<input type="checkbox"/>	<input type="checkbox"/>	
User Parameters										
PlotHeight		m	50 m	50,000000			50,000000	<input type="checkbox"/>	<input type="checkbox"/>	
PlotWidth		m	40 m	40,000000			40,000000	<input type="checkbox"/>	<input type="checkbox"/>	
Laitetila_1_Leveys		m	5 m	5,000000			5,000000	<input type="checkbox"/>	<input type="checkbox"/>	
Laitetila_1_Pituus		m	10 m	10,000000			10,000000	<input type="checkbox"/>	<input type="checkbox"/>	
Laitetila_1_OviseinänKorkeus		m	4 m	4,000000			4,000000	<input type="checkbox"/>	<input type="checkbox"/>	
Laitetila_1_TakaseinänKorkeus		m	3,5 m	3,500000			3,500000	<input type="checkbox"/>	<input type="checkbox"/>	
Laitetila_1_OvernLeveys		mm	910 mm	910,000000			910,000000	<input type="checkbox"/>	<input type="checkbox"/>	
Laitetila_1_OvernKorkeus		mm	2100 mm	2100,000000			2100,000000	<input type="checkbox"/>	<input type="checkbox"/>	

2.3 iLogic-ohjelmointi

iLogic mahdollistaa sääntöpohjaisen suunnittelun ja sitä kautta tarjoaa helpon tavan uudelleen käyttää jo kertaalleen tehtyä työtä. Sitä voidaan käyttää suunnitteluprosessin standardointiin, automatisointiin ja virtuaalituotteiden konfigurointiin. (Autodesk Inc., 2025) Laajuus, mihin iLogicia voidaan hyödyntää, on hyvinkin suuri. Edellä mainittujen asioiden lisäksi sillä voidaan myös ohjata muotoja, kokoja, piirteitä voidaan ohjata päälle ja pois, automatisoida konepiirustusten luomista. (Bordino, 2022)

2.4 Sääntöohjelmointi

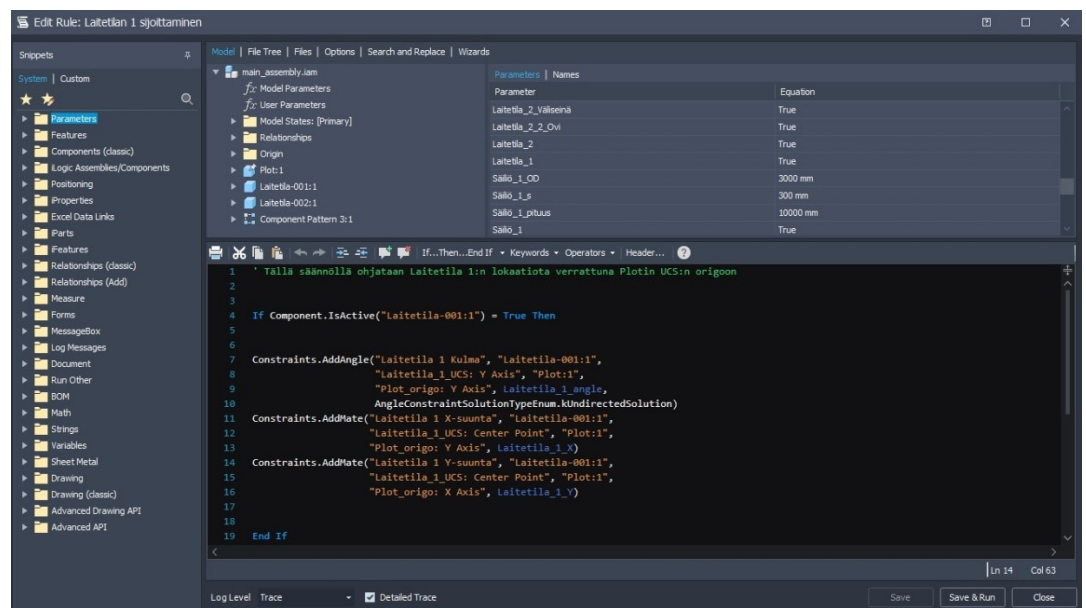
Ohjelmointi suoritetaan ns. sääntöihin, joka voidaan käynnistää manuaalisesti (run rule), ulkoisesti jostain muusta tiedostosta, Forms-työkalulla tai erilaisilla liipaisuilla (triggers). Sääntö voi olla sisäinen tai ulkoinen. Sisäiset säännöt ovat niitä, jotka vaikuttavat vain kyseiseen osaan tai kokoonpanoon, johon ne ovat luotu ja ne ovat osa sitä tiedostoa. Ulkoiset säännöt tallennetaan erilliseen tiedostoon. (Bordino, 2022)

Sääntöjen koodaukselle on Inventorissa oma ympäristönsä (Kuva 6). Käyttäjän työtä on helpotettu ja ohjelman mukana tulee kirjasto, mistä voi valita

tarpeellisen koodinpätkän (snippet) ja tarvittaessa muokata sitä. (Bordino, 2022) Kun sääntö luodaan, tai sitä muokataan, kyseinen teksti konvertoidaan VB.NET-koodiksi ja sen jälkeen se käännetään (compile). (Autodesk Inc., 2025)

Kuva 6 kolme vasemmalla on nähtävissä valmiit snippetit. Keskellä ylhäällä on mallipuu, josta pääsee käsiksi piirteisiin, luonnoksiin, rajoitteisiin yms. Oikealla ylhäällä on parametritaulukko, joka näyttää mallipuusta valitun osan tai alikokoonpaon parametrit. Mustalla taustalla olevaan ikkunaan kirjoitetaan varsinainen koodi.

Kuva 6. iLogic-ohjelmointiympäristö

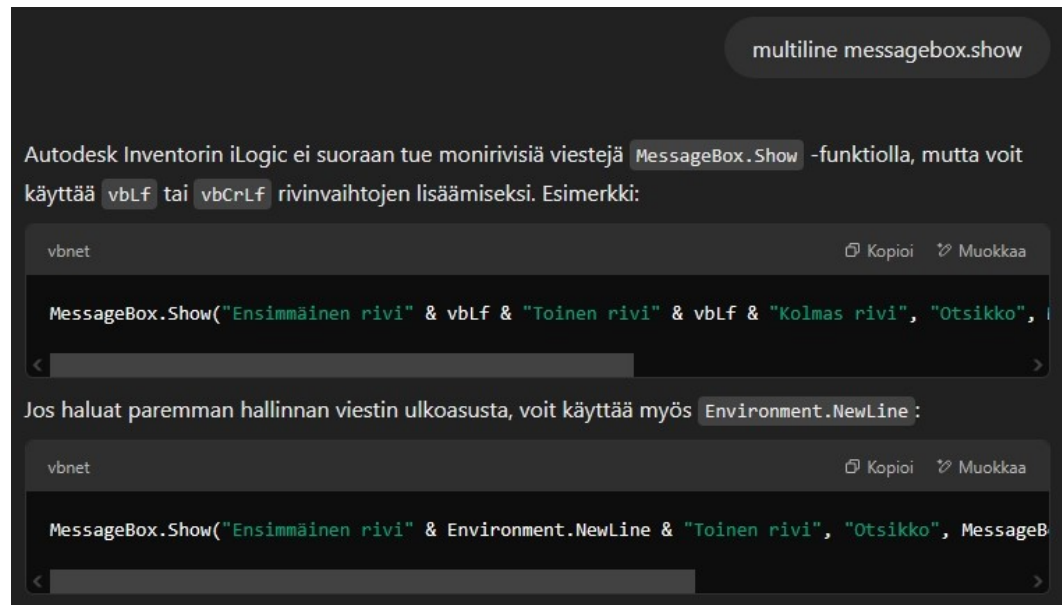


2.5 Tekoälyn hyödyntäminen koodauksessa

Koodin kehityksessä on käytetty hyödyksi tekoälyä nimeltä ChatGPT. Sen luoma koodi ei kuitenkaan ole käytetty sellaisenaan, vaan tekoälyn luoma koodi on käytetty vain suuntaa antavana. Käyttökokemuksen myötä, kyseinen tekoäly ei onnistunut pääsääntöisesti luomaan toimivaa koodin pätkää, vaan sitä jouduttiin lähes jokaisessa tapauksessa muokkaamaan, jotta se toimisi. ChatGPT toimi kuitenkin hyvänä apuna, kun tekijän koodaustaidot rajoittuivat vain perusteisiin eikä kyseisestä ohjelmointikielestä ollut kokemusta.

Merkittävin apu, jonka tekoäly pystyi tarjoamaan, oli ohjeet ohjelmointikielen syntaksiin. Esimerkiksi CAD-ohjelman omat ohjeet eivät sisältäneet ohjetta, jossa neuvottiin miten ponnahtusikkunaan saisi useamman rivin tekstiä. Tätä kysyttiin tekoälyltä (Kuva 7), joka tarjosi vastauksen, että rivien väliin täytyy lisätä `& vbLf`-komento. (OpenAI, 2023)

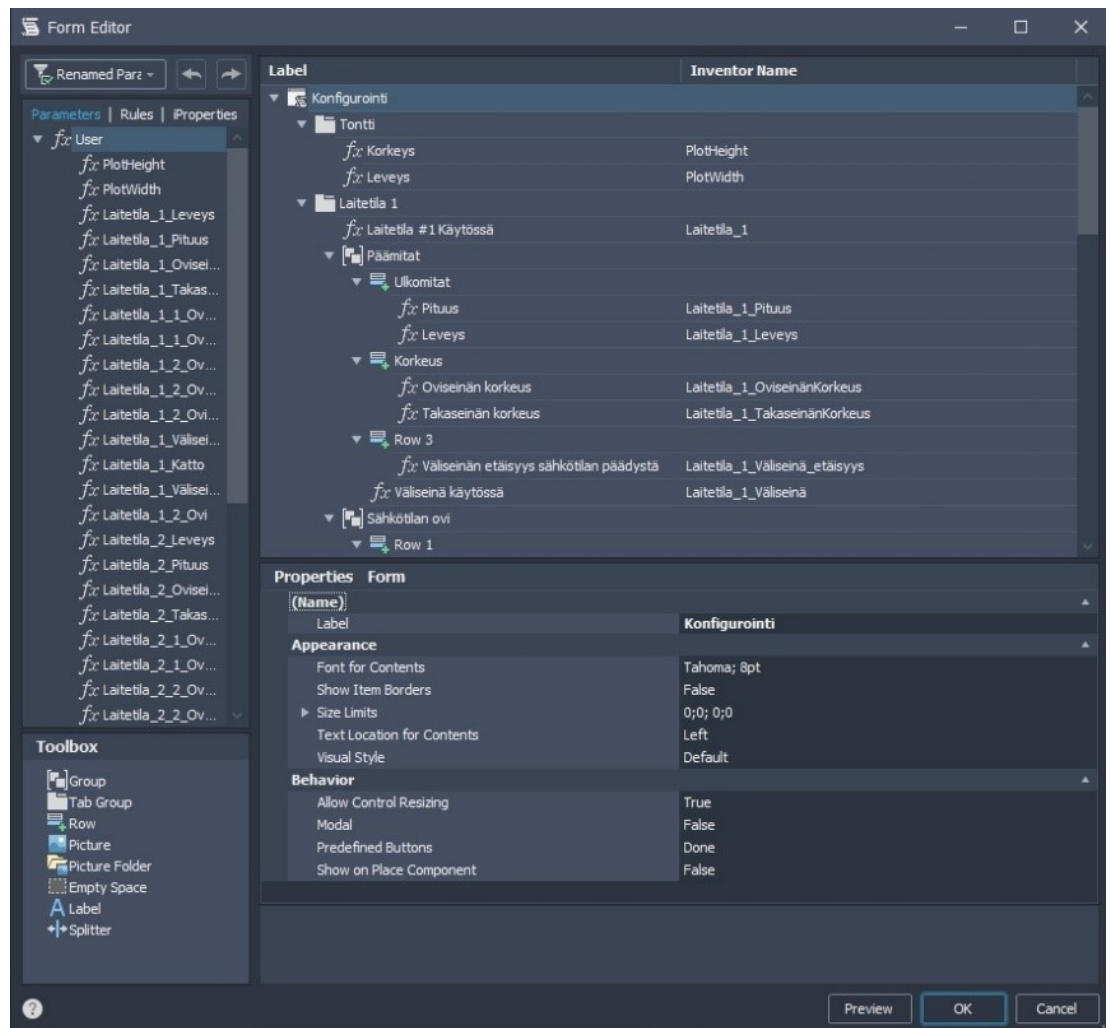
Kuva 7. Monirivisen ponnahtusikkunan syntaksin kysely tekoälyltä



2.6 Forms-työkalu

Formsilla (lomake) voidaan ohjata Inventoria sisäisten käyttöliittymien avulla eri parametrejä, ominaisuuksia ja sääntöjä osissa, kokoonpanoissa ja piirustuksissa. Näiden käyttöliittymien rakentamista varten on oma työkalunsa, joka on esitetty kuvassa 5. Työkalukokoelmasta (Toolbox) voi käyttäjä raahaamalla ja tiputtamalla muokata miten parametrit tai säännöt näkyvät lomakkeellansa. Niitä voi ryhmittää ja sommitella yhtenäiseen ryhmään, asettaa samalla riville tai esimerkiksi luoda kokonaan uusi välilehti niitä varten. Myös kuvia tai lisätekstiä voidaan lisätä. Myös lomakkeen tai parametrin esiintymän arvoja tai tietoja voidaan muuttaa. Fontin kokoa ja tyyliä voidaan editoida, sille voidaan asettaa ylä- ja alaraja, parametrin nimi lomakkeen esiintymää varten voidaan muuttaa, lomakkeen minimi- ja maksimikoko voidaan määrittää, kuvia voidaan lisätä, yms. (Autodesk Inc., 2025)

Kuva 8. Forms-työkalu



Seuraavassa kuvassa (Kuva 9) on nähtävillä laitetilaan liittyvän lomakkeen ulkoasu ja kerrotaan toteutuneen konfiguraattorin avulla, minkälaisia ohjaustapoja erityyppisille parametreille voidaan käyttää. Ohjaustavat eivät kuitenkaan rajoitu näihin esimerkkeihin. Konfigurointinäkylässä pääkomponentit ovat omilla välilehdillään (Tab Group) ja tässä esimerkissä laitetilan sisäiset asiat ovat jaettu viiteen eri ryhmään (Group).

Päämitat-ryhmässä mittoihin liittyviä numeerisia parametreja ohjataan tekstisyötteellä, kun taas Tosi/Epätosi-parametri (Väliseinä käytössä) voidaan kytkeä päälle tai pois raksittamalla. "Sähkötilan ovi"-ryhmässä on hyödynnetty moniarvoista parametria niin oven leveyden kuin korkeudenkin kanssa. "Prosessitilan ovi"-ryhmän numeerisia parametreja ohjataan syötteellä ja alasvetovalikolla, lisäksi "Etäisyys väliseinästä"-parametrille on asetettu erillinen puhekuplaan sijoitettu lisätietoteksti. "Katto"-välilehdellä ohjataan parametria vain päälle ja pois. "Sijainti"-ryhmässä on haluttu tuoda käyttäjälle heti näkyviin

tarvittavaa lisätietoa tekstin muodossa (Label), joka liittyy laittilan paikoitukseen suhteessa tonttiin. X- ja Y-suuntaista liikettä ohjataan numeerisella syötteellä, mutta kontin kulmaa XY-tasossa ohjataan liukupalkilla (Slider). Liukupalkin arvoja on rajattu niin, että sen arvonn täytyy olla 0 ja 360 asteen välillä, ja sitä voidaan kiertää kerralla 45 astetta.

Kuva 9. Forms-työkalulla luotu valintaikkuna

Konfigurointi

Tontti **Laittila 1** | Laittila 2 | Säiliöt

Laittila #1 Käytössä

^ Päämitat

Pituus	10 m	Leveys	5 m
Oviseinän korkeus	4 m	Takaseinän korkeus	3,5 m
Väliseinän etäisyys sähkötilan päädyistä	3,5 m		
Väliseinä käytössä	<input checked="" type="checkbox"/>		

^ Sähkötilan ovi

Leveys	910 mm	Korkeus	2100 mm
--------	--------	---------	---------

^ Prosessitilan ovi

Leveys	2500 mm	Korkeus	1900 mm
Etäisyys väliseinästä	0 mm		

^ Katto

Näkyvissä

Mitta väliseinästä oven oikeaan ulkoreunaan.

^ Sijainti

Mitta tontin UCS:sta laittilan keskelle

X-suunta	15 m	Y-suunta	10 m
Kulma	0 deg		

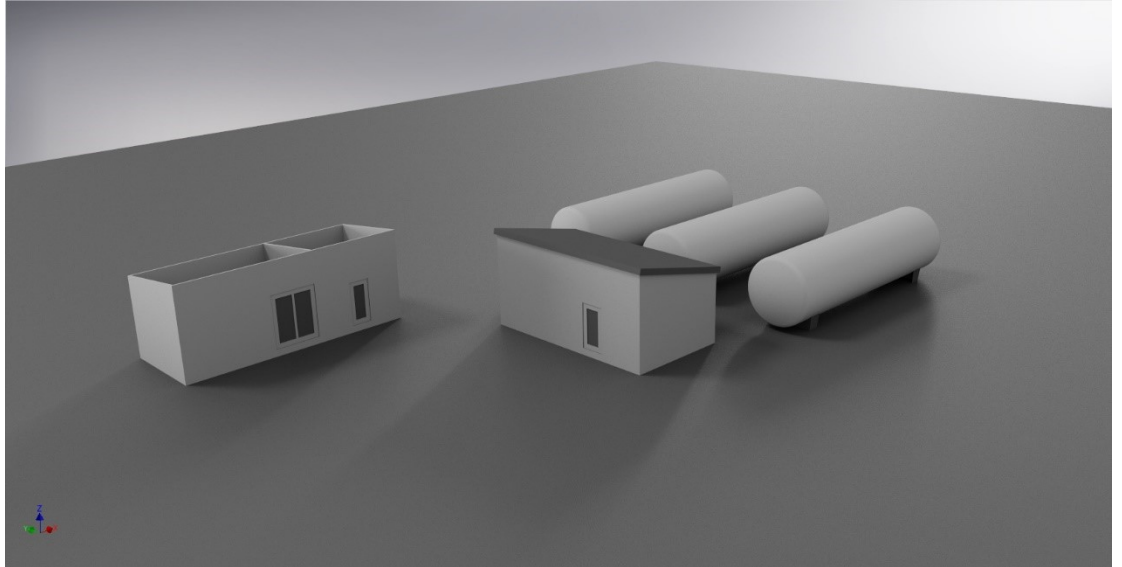
Done

3 Valmis ohjelma

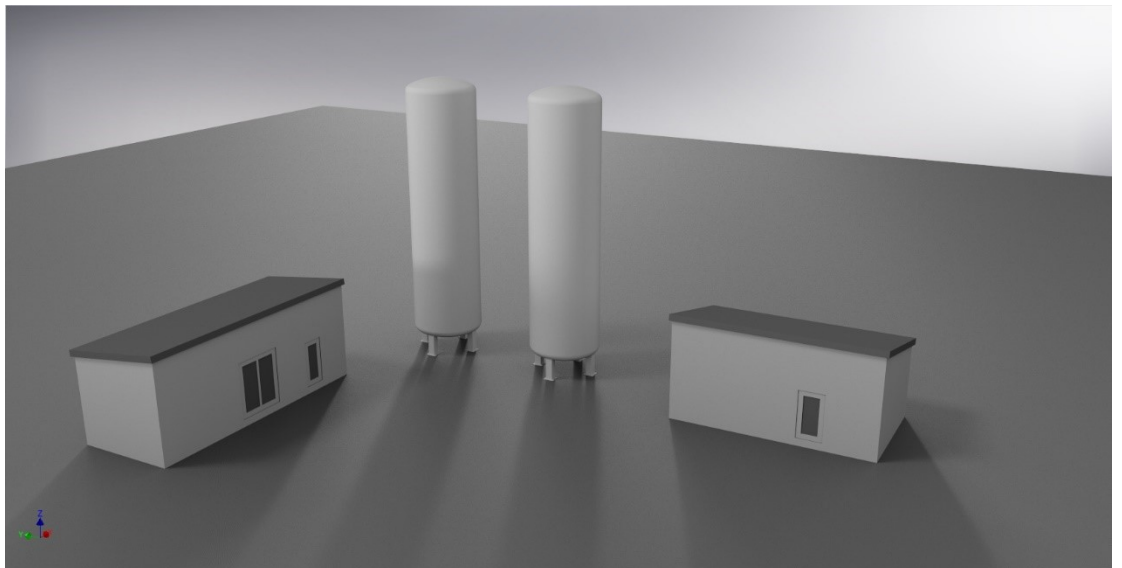
Valmis tuotekonfiguraattori sisältää viisi eri välilehteä, joilta voidaan ohjata pääkomponenttien parametrien arvoja. Laittiloille on luotu omat kaksi eri välilehteä, jotta ne pysyisivät pienempinä. Yhdeltä välilehdeeltä ohjataan tontin

kokoa. Säiliöt-välilehdellä voidaan ohjata säiliöiden parametrejä. Kuva 10 ja Kuva 11 alla ovat näkyvissä lopullisen kokoonpanon komponentit.

Kuva 10. Lopullinen kokoonpano



Kuva 11. Toinen versio lopullisesta kokoonpanosta



Tonttiosan parametreissä on vain vähäisesti kustomoitavaa, sillä se toimii vain esityspohjana. Ainoat ohjattavat parametrit Tonttiosassa ovat sen pituus ja leveys. Useasti laitosten konseptointivaiheessa mahdollisella laitostoimittajalla ei ole tietoa saatavilla tontin todellisesta muodosta, joten tässä tilanteessa yksinkertainen suorakulmio on riittävä. Tonttiosan mittojen muuttaminen aiheutti aluksi ongelmia muissa konfiguraattorin osissa. Ongelman huomattiin liittyvän

tonttiosan paikoittamiseen. Alun perin se oli vain sijoitettu mallinnusavaruuden origoon ja asetettu pysymään paikallaan (grounded). Tämä salli kyllä Tonttiosan mittojen muuttamisen, mutta se rikkoi muiden alikokoonpanojen kiinnitykset ja relaatiot. Tontin yhteen nurkkaan oli jo aikaisemmin alikokoonpanojen paikoittamista varten luotu käyttäjä koordinaatisto (USC, user coordinate system), tämä osoittautui myös ratkaisuksi edellä mainittuun ongelmaan. Tonttiosa sidottiin mallinnusavaruuden origoon luodusta koordinaatistosta. Näin tämä nurkka pysyy koko ajan paikallaan ja Tonttiosan leveys sekä pituus kasvavat tai pienenevät origosta poispäin, eikä haittaa muiden osien paikoitusta.

3.1 Laitetila

Laitetilan konfigurointi näkymä vastaa Kuva 9 olevaa näkymää. Kuten aikaisemmin todettiin, molempien laitetilojen konfigurointi on sijoitettu omille välilehdilleen käyttöliittymän yksinkertaisuuden ylläpitämiseksi. Laitetilojen konfiguraattorinäkymät ovat identtisiä ja ne esitetty tässä raportissa jo aikaisemmin, Kuva 9. Välilehdellä ylimpänä on raxitettava parametri, jolla valitaan, onko kyseinen laitetila käytössä vai ei. Mikäli käyttäjä poistaa ruksin hiiren napautuksella, asettaa se kyseisen Laitetila-osan epäaktiiviseksi sekä deaktivoi sen parametrivalikot konfiguraattorin käyttöliittymästä. Laitetilojen ohjattavat parametrit ovat ryhmitelty viiteen eri alakategoriaan, jotka ovat

- Päämitat
- Sähkötilan ovi
- Prosessitilan ovi
- Katto
- Sijainti

Päämitat-välilehdellä voidaan ohjata laitetilan pituutta, leveyttä, molempien seinien korkeutta erikseen, väliseinäinen paikkaa, sekä onko väliseinä edes käytössä. Mikäli väliseinä ei ole käytössä, ohjelma myös automaattisesti poistaa prosessitilan oven käytöstä. Edellä mainittujen parametrien yksiköksi päätettiin asettaa metrit, eikä millimetrit. Vaikka suunnittelussa normaalisti käytetäänkin millimetrejä, on tässä tapauksessa metrien käyttäminen yksikkönä tekijän mielestä soveliaampaa.

Sähkötilan oven sijoitus on sidottu väliseinän etäisyyteen sähkötilan puoleisesta rakennuksen päädyistä. Oven keskipiste on sidottu rajoittein olemaan aina olemaan tasan pisteen kanssa, joka on puolessa välissä sähkötilan ulkopäätyä ja väliseinää. Oven alareunan korkoa ei voida muuttaa, vaan se on aina vakio. Kyseinen mitta on asetettu oven Pursotus-piirteen luonnokseen. Oven leveyttä ja korkeutta voidaan muuttaa esiaseteltujen, moniarvoisten parametrien avulla. Ovelle on määritetty kaksi vakio leveyttä ja neljä eri korkeutta.

Prosessitilan oven parametrejä pystytään ohjaamaan hieman vapaammin, joskin siinä on joitain samoja yhtäläisyyksiä sähkötilan oven parametrien kanssa. Myös prosessitilan oven alareuna on sidottu vakiomittaan, joka myös on asetettu Pursotus-piirteen luonnokseen. Oven korkeudet ovat myös samat, kuin aikaisemmassa ovenssa. Oven leveyden määrittäminen tapahtuu käyttäjän antaman mitan mukaan ilman rajoitteita. Prosessitilan oven paikoitusta voidaan muuttaa antamalla mitta, joka on määritetty väliseinästä. Mitta on voimassa, vaikka väliseinä ei olisikaan käytössä. Tätä piirrettä varten jouduttiin rakentamaan oma sääntö alikokoonpanotasolle, jotta pursotuksen luonnos ei voi ylittää päätyseinää. Mikäli tätä sääntöä ei olisi luotu, pystyisi käyttäjä asettamaan parametrin arvon liian suureksi. Testauksen perusteella huomattiin, että jos parametrin muutti ensin liian suureksi, ja tämän jälkeen takaisin pienemmäksi, niin piirre lakkasi toimimasta kokonaan. Korjatakseen tilanteen käyttäjän täytyisi aina mennä Laitetila-osan sisälle ja käydä määrittämässä luonnoksen referenssipisteet uusiksi. Luotu sääntö summaa prosessitilan oven etäisyyden väliseinästä, prosessitilan oven leveyden, sähkötilan leveyden sekä 100 mm, joka on kiinteäksi määritetty väliseinän paksuus. Summattuja mittoja verrataan laittilan kokonaispituuteen. Mikäli sen pituus on summan kanssa yhtä suuri, tai pienempi, sääntö avaa ponnahdusikkunan, jossa varoitetaan käyttäjää virheellisestä oven paikotuksesta. Ilmoituksen lisäksi parametri, joka määrittää prosessitilan oven etäisyyden väliseinästä, asetetaan nolnaan millimetriin. Mikäli laittilan kokonaispituus on suurempi kuin edellä mainittu useamman mitan summaus, jää käyttäjän antama mitta voimaan.

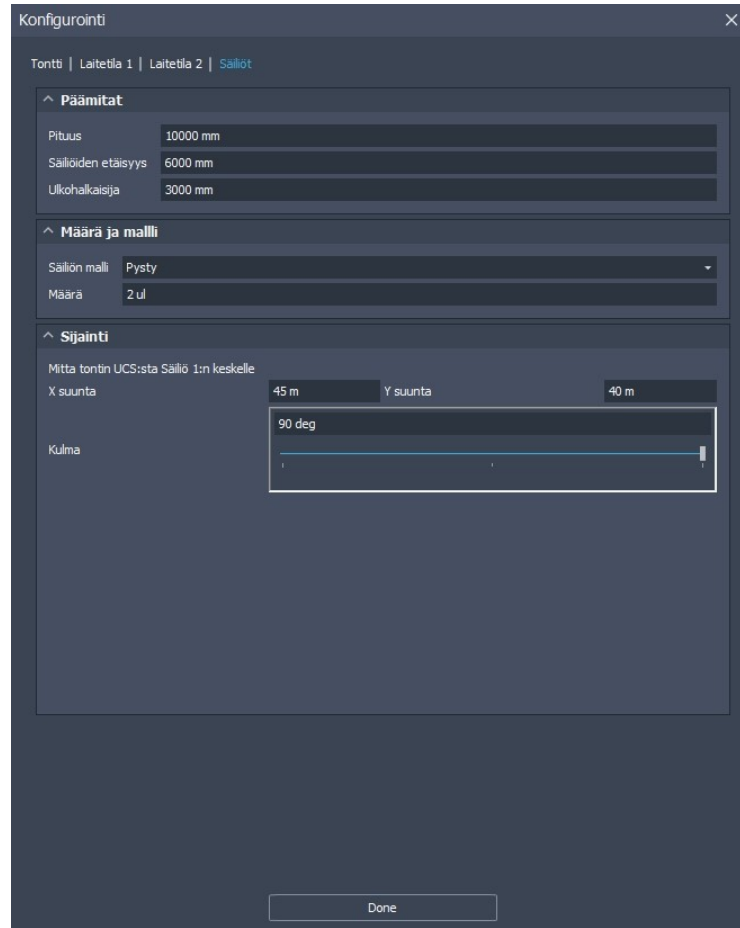
Katon näkyvyyttä ohjataan Tosi/Epätosi-parametrilla, eikä sillä ole sen toiminnallisempaa merkitystä. Tämä ominaisuus päätettiin luoda sen takia, että konfiguraattorin mahdollisen kehitystyön jatkuessa tämän opinnäytetyön jälkeen laittilan sisälle voidaan sijoittaa prosessilaitteistoa, jota voi tämän parametrin avulla helposti tarkastella.

Laitetilan sijaintia voidaan ohjata X- ja Y-suunnassa antamalla syöte metreissä. Tätä ominaisuutta varten laitetilojen osiin luotiin keskelle lattiaa USC. Tämän koordinaatiston keskipiste sidottiin rajoitteella tontin nurkassa sijaitsevaan USC:ään. Laitetilan orientaatio Z-akselinsa ympäri voidaan säätää liukusäätimellä (Slider). Säätimen minimi arvo on nolla astetta ja maksimi 360 astetta. Säätö tapahtuu 45-asteen pykälissä.

3.2 Säiliöt

Säiliöiden esiintymiä ohjataan yhdeltä välilehdeltä, joka on esitetty **Virhe. Viitteen lähde ei löytynyt..** Säiliö on konfiguraattorin ainoa alikokoonpano ja säiliöiden esiintymien määrää ohjataan kuviotyökalun (pattern) avulla. Patternista voidaan käyttöliittymän avulla muuttaa kahta arvoa, esiintymien lukumäärää sekä esiintymien välistä etäisyyttä toisistaan. Etäisyys mitataan esiintymän keskipisteestä, ei ulko- tai sisäreunasta. Säiliön tai säiliöiden sijoitusta ja orientaatio voidaan ohjata samalla periaatteella, kuin laitetilojakin.

Kuva 12. Säiliöiden Konfigurointi-välilehti



Joissain laitoskohteissa suositaan vaakasäiliöitä, kun taas toisissa kohteissa soveliaampi ratkaisu on pystysäiliö tai -säiliöt. Tämä ominaisuus on erittäin tärkeä olla mukana konseptivaiheen suunnittelussa, kun suunnittelija hahmottelee ja mallintaa laitoksen viemää pinta-alaa mahdollisesta rakennuskohteesta. Konfiguraattorissa asia on rakennettu siten, että säiliöiden alikokoonpanossa on niin vaaka-, kuin pystysäiliöinkin jalat valmiina. Molemmat jalkatyypit ovat sidottu rajoittein varsinaiseen säiliöön, siten että säiliön koon muuttuessa jalkojen paikat siirtyvät ennalta määritettyjen mittojen mukana säilyttäen oikean suhteellisuuden säiliöön. Konfiguraattorissa kun valitaan esimerkiksi pystysäiliö, niin ohjelma sääntöinen asettaa pystyjalkojen osan aktiiviseksi ja deaktivoi (supress) vaakajalat.

Pystysäiliön jalkojen mitat ovat vakiot, eli käyttäjä ei voi vaihtaa niiden korkeutta tai muitakaan mittoja. Vaakasäiliön jalat sen sijaan muuttuvat säiliön ulkohalkaisijan mukaan. Vaakasäiliön jalkojen osan luonnos on näin sanottu adaptiivinen luonnos (adaptive sketch) eli sen rajoitteet on sidottu rajoittein ulkohalkaisijan luonnokseen.

3.3 Parametrien tuonti pääkokoonpanotasolle

Forms-työkalu sallii vain kyseisen osan tai kokoonpanon parametrien asettelun käyttöliittymältä. Tämä tarkoittaa sitä, että osien ja alikokoonpanojen parametrit täytyy kutsua pääkokoonpanotasolle, jotta voimme ohjata niitä. Käytännössä asia toteutuu niin, että pääkokoonpanotasolle luodaan parametri, jonka arvo asetetaan olemaan yhtä suuri kuin toivotun piirteen tai parametrin arvo alikokoonpanon puolella. Tähän tarvitaan yksinkertainen iLogic-sääntö, josta seuraava esimerkki kertoo.

Esimerkki: Kyseinen rivi asettaa osassa Laitetila-001:ssa olevan parametrin nimeltä "laitetila_levveys" yhtäsuureksi, kuin pääkokoonpanotasolla oleva parametri nimeltä "Laitetila_1_levveys". Nyt pääkokoonpanotasolla olevan paramaterin suuruudella voidaan ohjata osan tai alikokoonpanon parametria ja siten käyttää sitä Formsissa.

Parameter("Laitetila-001:1", "laitetila_levveys") = Laitetila_1_Leveys

3.4 Virheilmoitusten huomiointi koodilla

Mikäli parametrin tuonti pääkokoontasolle olisi suoritettu vain listaamalla parametrejä yllä olevan esimerkin mukaisesti, voi se aiheuttaa toivomattomia ja turhia virheilmoituksia. Edellistä esimerkkiä hyödyntäen, mikäli Laitetila-001 otetaan kokonaan pois käytöstä, syntyisi tästä virheilmoitus joka kerta, kun ohjelma suorittaa kyseisen komennon. Virheilmoitus täytyisi aina kuitata erillisestä napista ja se aiheuttaisi turhaa vaivaa käyttäjälleen. Asia on ratkaistu käyttäen yksinkertaista IF-lauseketta. Asetettu IF-lauseke tarkastaa Laitetila-001:n esiintymän tilan ja mikäli sillä on aktiivinen esiintymä mallissa, aktivoi se sitten vasta parametrien tuonnin pääkokoontasolle. Mikäli kyseistä esiintymää ei ole, ohjelma ohittaa mainitun komponentin parametrin tuonnin kokonaan. Seuraavassa esimerkissä on esitetty yllä kuvattu IF-lauseke. IF-lauseke tulee päättää End If-komentoon.

```
If Component.IsActive("Laitetila-001:") = True Then
```

```
Parameter("Laitetila-001:1", "laitetila_leveys") = Laitetila_1_Leveys
```

```
End If
```

4 Vertailu

Luodun tuotekonfiguraattorin tehokkuutta muutosten suorittamiseen verrataan ajallisesti perinteisellä tavalla mallinnetun kokoonpanon kanssa. ”Perinteisellä tavalla” tarkoitetaan tässä asian yhteydessä sitä, että minkään alikokoonpanon tai siihen kuuluvan osan mitään piirrettä tai sen mittaa ei ole parametroitu. Toisin sanoen jokainen mittamuutos täytyy käydä tekemässä muuttamalla sketsin mittoja tai ”edit feature”-komennolla piirteen syöttöarvoja muuttaen. Vertailua ei suoritettu useammalla eri käyttäjällä tai moneen kertaan, vaan sen toteutti opinnäytetyön tekijä. Vertailu suoritettiin vain kerran ja sen tarkoitus oli vain havainnollistaa yksinkertaisella tavalla konfiguraattoreiden tehokkuutta.

4.1 Mallien lähtökohta ja niiden muutokset

Molemmissa malleissa, niin konfiguraation kuin ei-parametrisen, lähtökohta on identtinen. Kappaleet ovat sijoiteltu samaan paikkaan, ne ovat samankokoisia

ja niiden esiintymien lukumäärä on yhtä suuri. Niin ikään myös ennalta määritellyt muutokset ovat sen tyyliisiä, että ne voidaan toteuttaa identtisesti molemmissa malleissa. Eli tavalliseen malliin ei lisätä tai siinä ei muokata semmoista asiaa, mitä tuotekonfiguraattorilla ei voida suorittaa.

Muutettavat ominaisuudet ovat seuraavanlaiset:

- Tontin korkeus muutetaan 30 metriin
- Tontin leveys muutetaan 20 metriin
- Laitetila 1:n kokonaispituus muutetaan kymmeneen metriin
- Sen leveys muutetaan viiteen metriin
- Väliseinä otetaan käyttöön
- Sähkötilan oven leveydeksi muutetaan 910 mm ja korkeudeksi 2000 mm
- Väliseinän esiintyminen tarkastamalla deaktivoimalla katto ja tarkastamisen jälkeen katto aktivoidaan takaisin.
- Laitetila 1:stä käännetään 90-astetta
- Laitetila 2 aktivoidaan käyttöön
- Sen pituudeksi ja leveydeksi määritetään neljä metriä
- Molempien seinien korkeudeksi määritellään kolme metriä
- Väliseinä poistetaan käytöstä
- Säiliöiden esiintymien määrä muutetaan kolmeen
- Säiliöiden malli muutetaan vaakaan
- Säiliöiden ulkohalkaisija muutetaan 3500 millimetriin
- Säiliöiden pituus muutetaan 8000 millimetriin

4.2 Vertailun tulokset

Molemmilla tavoilla, perinteisellä ja konfiguraattorilla, tehdyt muutokset tallennettiin videokaappauksen muodossa suoritetuista edellisessä kappaleessa mainituista muutoksista. Perinteisellä tavalla tehdyt muutokset kestivät ajallisesti kuusi minuuttia ja 48 sekuntia. Konfiguraattorilla vastaavien muutoksien tekeminen kesti vain yhden minuutin ja 44 sekuntia. Tästä voidaan siis todeta, että perinteisessä mallinnustavalla työaika kuluu lähes nelinkertaisesti verrattuna konfiguraattoria käytettäessä. Yleisesti konseptivaiheessa joudutaan muutoksia tekemään useita kertoja, jolloin säästetty työaika on erittäin merkittävä.

5 Pohdinta

Nykypäivän trendin mukaisesti jokaiseen prosessiin tai työhön etsitään jatkuvasti tehostavia toimenpiteitä. Töitä mitataan useilla eri mittareilla ja työkaluilla sekä suunnitellaan toimenpiteitä tehokkuuden parantamiseksi. Asioiden kehittäminen voi johtua laadullisista tai taloudellisista hyödyistä, vaikkakin useasti ne kulkevat käsi kädessä.

Eryteisesti työt, joissa ilmenee paljon samankaltaisuutta tai toistoa, ovat mahdollisen kehitystyön puuttuessa aikaa kuluttavia ja jopa taloudellisesti heikkoja. Tuotekonfiguraattorilla voidaan mahdollistaa tasaisemman laadun ja nopeamman työsuorituksen summa. Kun konfiguraattori luodaan huolellisesti, on sen avulla saavutettavat hyödyt erittäin mittavat. Sen avulla voidaan poistaa yhtälöstä paljon työtä, joka toistuisi manuaalisesti tehdessä.

Opinnäytetyön aikana huomasin, että se mitä konfiguraattorin halutaan tekevän, on määriteltävä huolellisesti ja erittäin tarkkaan jo alkuvaiheessa, ennen varsinaisen konfiguraattorin rakentamista. Itseltä tämä jäi tekemättä ja huomasin, että jatkuvien uusien ominaisuuksien ja toimintaideoiden tulva oli valtava. Lähes joka kerta kun rakensin tai koodasin konfiguraattoria tuli aina jokin ajatus uudesta ominaisuudesta tai miten jotakin tiettyä jo luotua ominaisuutta voisi vielä parantaa lisää.

Olen erityisen tyytyväinen, mitä kaikkea opinnäytetyö minulle opetti Inventor-suunnitteluohjelmasta. En aikaisemmin ole ollut iLogic-ohjelmoinnin tai Forms-työkalun kanssa tekemisissä, ja niiden tarjoamat mahdollisuudet osoittautuivat tähän kyseiseen työhön täydellisiksi. Eryteisesti iLogic pystyy tarjoamaan tätäkin enemmän, kuten esimerkiksi ohjelmointi rajapinnan erillisohjelmien kanssa.

Lähteet

- Adobe. (7. 3. 2025). *Adobe.com*. Noudettu osoitteesta <https://www.adobe.com/fi/products/substance3d/discover/parametric-modeling.html>
- Autodesk Inc. (7. 3. 2025). *autodesk.com*. Noudettu osoitteesta <https://www.autodesk.com/solutions/parametric-modeling>
- Autodesk Inc. (23. 2. 2025). *help.autodesk.com*. Noudettu osoitteesta <https://help.autodesk.com/view/INVNTOR/2024/ENU/>
- Bordino, A. (2022). *Autodesk Inventor 2023 Cookbook*. Packt Publishing. Noudettu osoitteesta <https://learning.oreilly.com/library/view/autodesk-inventor-2023/9781801810500/>
- Houston, B. (20. 5. 2020). *Threekit*. Noudettu osoitteesta [threekit.com: https://www.threekit.com/blog/what-is-parametric-configuration](https://www.threekit.com/blog/what-is-parametric-configuration)
- Marquardt, J. (n.d.). *Salesforce*. Noudettu osoitteesta [salesforce.com: https://www.salesforce.com/sales/cpq/what-is-cpq/](https://www.salesforce.com/sales/cpq/what-is-cpq/)
- Martio, A. (2015). *Tuotekonfigurointi ja tuotetiedon hallinta*. Amartekno.
- OpenAI. (1. 3. 2023). *ChatGPT*. Noudettu osoitteesta <https://www.chatGPT.com>
- Pieper, C. (7. 6. 2022). *Threekit*. Noudettu osoitteesta [threekit.com: https://tinyurl.com/4dy6v6nc](https://tinyurl.com/4dy6v6nc)

