



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Mikael Koivula

KEHITYSTYÖ 3D CAD -SUUNNITTELUN JA TOISTUVIEN TÖIDEN TEHOSTAMISEKSI IN- VENTOR ILOGICIN AVULLA

Tekniikka
2024

ALKUSANAT

Tämä opinnäytetyö on tehty vuosien 2023 ja 2024 välillä Vaasan ammattikorkeakoulun Konetekniikan koulutuksessa ja kehitystyötä on tehty jatkuvasti keväästä 2023 lähtien VEO Oy:n mekaniikkasuunnittelun puolella.

Haluan kiittää VEO Oy:tä tämän opinnäytetyön teon mahdollisuudesta, Jussi Anttilaa luottavasta ja kannustavasta ohjauksesta VEO:lla. Rami Nevalaa haluan kiittää tämän aiheen esittelystä, kattavasta Inventorin käyttöön liittyvästä koulutuksesta, kehitystyöhön liittyvästä yhteistyöstä ja haastattelusta.

Toivon tämän opinnäytetyön olevan inspiraation lähde mekaniikkasuunnittelun jatkokehityksille sekä herättävän mielenkiintoa suunnitteluautomaatioon yleisesti ja keskustelua sen mahdollisuuksista VEO Oy:llä.

Vaasassa 20.12.2023

Mikael Koivula

TIIVISTELMÄ

Tekijä	Mikael Koivula
Opinnäytetyön nimi	Kehitystyö 3D CAD -suunnittelun ja toistuvien töiden tehostamiseksi Inventor iLogicin avulla
Vuosi	2024
Kieli	suomi
Sivumäärä	71 + 6 liitettä
Ohjaaja	Jani Leppämäki

Opinnäytetyön aiheena ja tarkoituksena on käydä läpi keväältä 2023 lähtien tehtyä jatkuvaa kehitystyötä tehostamaan 3D CAD -suunnittelua ja vähentämään toistuvia suunnittelun tehtäviä yrityksessä VEO. Autodesk Inventorin suunnitteluautomaatio-ohjelmistoa nimeltä iLogic käytettiin kehitystyössä. Kehitystyötä tehtiin yhteistyössä Rami Nevalan kanssa, joka on VEO:lla kehittämässä Autodesk Inventorin kautta tehtävää suunnittelua.

Teoreettisen viitekehyksen pohjana on oikeanlaiset 3D-mallinnusperiaatteet suunnitteluautomaation pohjana. Kehittämisen apuna ja sen tarvitseman työvälineohjelman käytön tietoperustan lähteenä toimi Autodesk Inventorin omat dokumentaatiot, yrityksen tarjoamat lähtötiedot ja internet-lähteet. Kehitystyön jälkeisenä aikana toteutettiin käyttäjäpalautekysely, arvioimaan tehdyn kehitystyön tuloksia ja merkityksellisyyttä.

Kehitystyön aikana on saavutettu erittäin isoja askelia suunnittelutyön tehostamisessa ja niiden tuloksia käsitellään myös tehdyn käyttäjäpalautekyselyn perusteella ja omilla henkilökohtaisilla havainnoilla. Opinnäytetyössä tehdään katsaus myös tulevaisuuden mahdollisiin jatkokehityksiin toistuvien töiden vähentämiseksi.

ABSTRACT

Author	Mikael Koivula
Title	Development Work to Streamline 3D CAD Design and Repetitive Tasks Using Inventor iLogic
Year	2024
Language	Finnish
Pages	71 + 6 Appendices
Name of Supervisor	Jani Leppämäki

The subject and purpose of this thesis was to review continuous development work done since the spring of 2023 to make 3D CAD-design more efficient and reduce repetitive design tasks in the company called VEO. The Autodesk Inventor design automation program iLogic was used in the development work. The development work was done in collaboration with Rami Nevala, who is also involved at VEO in the development of design work done through Autodesk Inventor.

The theoretical framework is based on proper 3D modeling principles as the basis for design automation. For the development work and the required knowledge base for the software tool used in development, Autodesk Inventor's own documentation, company-provided initial data and internet acted as the sources. After the development work, a user feedback survey was conducted to evaluate the results and significance of the development work.

Significant steps were made during the development work towards the efficiency of design work and the results will also be discussed with user feedback obtained from the conducted survey. The thesis also gives an overview on future development possibilities to reduce repetitive tasks.

SANASTO

3D CAD	3D Computer Aided Design eli Tietokoneavusteinen 3D-suunnittelu.
BOM	Bill of Materials eli materiaaliluettelo komponenteista, kokoonpanoista ja niiden määristä.
Boolean	Tiedontyyppi millä on vain kaksi arvoa: Tosi tai Epätosi.
Bottom-Up	Menetelmä, missä tuoterakenne rakennetaan alhaalta ylös erikseen mallinnettujen osien perusteella.
Check In	Sisäänkuittaus tuotetiedon hallintajärjestelmään säilytykseen.
Check Out	Uloskuittaus tuotetiedon hallintajärjestelmästä muokkauksia varten.
Composite	iMaten toiminto, joka yhdistää useat rajoitteet yhdeksi itsenäiseksi rajoitteeksi. Liitettävästä mallista täytyy löytyä vastaava rajoite liitosta varten.
Direct Edit	Inventorin toiminto millä voi muokata valmista mallia jälkikäteen luomatta luonnoksia tai piiteitä esimerkiksi venyttämällä tai kiertämällä.
iam	Autodesk Inventorin käyttämä tiedostomuoto kokoonpanoille.
iAssembly	Toiminto millä voi luoda ja hallita Excel-pohjaiseen taulukkoon keskitetyksi muutettavia variaatioita kokoonpanoista.
idw	Autodesk Inventorin käyttämä tiedostomuoto piirustuksille.

iLogic	Inventorin toiminto suunnitteluautomaatio-ohjelmointia varten.
iLogic Form	iLogic toimintoja hallinnoiva lomake tai kyselyikkuna, missä parametrien valintoja sekä mittoja voidaan asettaa ja koodia ajaa.
iLogic Snippet	Snippetit eli koodileikkeet ovat lyhennettyjä koodinpätkiä, jotka on tehty ohjelmoinnin nopeuttamiseksi ja helpottamiseksi.
iMate	Toiminto Inventorissa rajoitteiden tai niiden komposiittien luomiseksi.
Inventor API	Inventor Application Programming Interface eli Inventorin ohjelmointirajapinta.
iPart	On sama kuin iAssembly, mutta vain osatiedostojen variaatioiden taulukkohallintaa varten.
iProperties	Attribuuttitiedot piirustuksissa, kokoonpanoissa ja osissa.
ipt	Autodesk Inventorin käyttämä tiedostomuoto osille.
iTrigger	Komento, joka luo yksiköttömän numeroparametrin Inventorissa koodien käynnistämistä varten.
Kiskoelementti	Alumiinista tai kuparisista virtakiskoista muodostuva kiskosillan osa.
Konfiguraattori	Työkalu millä voi muokata ja räätälöidä haluttua mallia tarpeisiin soveltuvaksi.

Level of Detail	Mallin yksityiskohtaisuuden määrittävä taso. Jokaiselle tasolle voi määrittää mitkä osat tai kokoonpanot ovat mukana.
Middle-out	Menetelmä, joka on Top-Downin ja Bottom-Upin yhdistelmä.
Parametri	Muuttuja mikä määrittää mm. mitan tai halutun funktion mallissa.
PE	Potective Earth eli maadoitusjohdin
PEN	Protective Earth Neutral eli yhdistetty neutraali- ja maadoitusjohdin
Released	Tila tiedostolle, kun se on julkaistu Vaultissa valmiiksi tuotantoa varten.
Rule	Rulet eli säännöt ovat iLogicin tiedostoja mihin kirjoitetaan mm. VB.NET-koodikieltä ja koodinpätkiä niiden ajoa varten
Suppress	Inventorissa suppress-toiminto ottaa pois käytöstä mallien piirteitä tai kokoonpanossa olevia komponenttien osia.
Top-Down	Mallinnusmenetelmä, missä ylimmällä tasolla määritetään rakenne ja kaikki määräävät mitat sekä piirteet.
Vault	Autodeskin Product Data Management eli Tuotetiedonhallinta-järjestelmä.
Vault Copy Design	Vaultin toiminto millä voi kopioida kokonaisia kokoonpanoja ja niiden tiedostoja uusiksi variaatioiksi.
Vault SDK	Vaultin ohjelmistokehitystyökalupaketti.

VB.NET	Visual Basic .NET-koodikieli.
VEBA	VEO Oy:n Pienjännitekiskosilta.
VEDA	VEO Oy:n Pienjännitekojeisto.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO.....	14
1.1	Sisältö.....	14
1.2	VEO Oy	14
2	TEORIA	16
2.1	Parametrinen mallintaminen rajoitteita hyödyntäen	16
2.2	Multi-Body	18
2.3	Ohjausmalli ja Top-Down -menetelmä	20
2.4	Konfiguraattori.....	21
2.5	Suunnittelun kehittäminen yrityksissä	23
3	SUUNNITTELUOHJELMISTO	25
3.1	Inventor.....	25
3.1.1	iLogic.....	26
3.1.2	iLogic Forms.....	27
3.1.3	Event Triggers.....	28
3.1.4	Inventor API.....	29
3.2	Vault.....	30
4	KEHITYSTYÖ JA TUTKIMUSONGELMA	32
4.1	Tausta.....	32
4.2	Tutkimusongelma ja lähtötilanne	32
4.3	Alustava tutkimustyö ja opiskelu	33
5	KEHITYKSEN KOHTEET	35
5.1	VEBA Kiskoelementit.....	35
5.1.1	Katkaisu- ja taivutuspituudet	37
5.1.2	Automaattinen piirustus	39
5.2	Marine-alusta.....	40

	10
5.3 Jäähdytysputki	43
5.4 Virtuaaliset komponentit	47
6 KEHITYSTYÖN VAIKUTUS	49
6.1 Käyttäjien palaute	49
6.2 Tehostus	52
7 KÄYTTÄJÄYSTÄVÄLLISYYS	56
8 JATKOKEHITYS.....	59
8.1 iLogic & Vault	59
8.2 Kiskoelementtien jatkokehitys.....	60
8.2.1 iMate	61
8.2.2 Automaattinen uudelleen nimeäminen ja kopiointi.....	64
8.2.3 iPart & iAssembly	64
8.3 Kehityksen kohteiden parantelu	65
9 ITSEARVIOINTI JA YHTEENVETO	68
9.1 Itsearviointi	68
9.2 Yhteenveto	68
LÄHTEET	70
LIITTEET	72

KUVIO- JA TAULUKKOLUETTELO

Kuva 1. VEO:n liiketoiminta eri alueilla. [3]	15
Kuva 2. Luonnosviivat rajoitettuna keskelle koordinaatistoa.	17
Kuva 3. Huonot tavat.	18
Kuva 4. Parametrit hyviä tapoja noudattaen ja rajoitteita hyödyntäen.	18
Kuva 5. Itsenäiset tilavuusmallit osatiedostossa.	19
Kuva 6. Inventor Professional 2024.	25
Kuva 7. iLogic-ohjelmointiruutu.	27
Kuva 8. iLogic Form Editor.	28
Kuva 9. Event Triggers.....	29
Kuva 10. Inventor 2024 API Help.	30
Kuva 11. Autodesk Vault Professional 2024. [11].....	31
Kuva 12. Tuplataivutettu kiskoelementti.	35
Kuva 13. Esimerkkejä kyselyikkunan välilehdistä ja valinnoista.	36
Kuva 14. Kiskoelementin taivutuksen suunta ja PEN-puolet vaihdettuna.	37
Kuva 15. Esimerkkejä tekstisymboleista.....	39
Kuva 16. Marine-alusta.....	41
Kuva 17. Marine-alustan kyselyikkuna välilehtineen.	42
Kuva 18. Alusta kahdella kentällä ja välipalkki poistettuna.....	43
Kuva 19. Jäähdytysputken kokoonpano.	44
Kuva 20. Jäähdytysputken kyselyikkuna välilehtineen.....	45
Kuva 21. Jäähdytysputki varioituna kahdella laipalla.	46
Kuva 22. Normaali tapa luoda virtuaalinen komponentti.	47
Kuva 23. Esimerkki koodin ehtolauseista.	48
Kuva 24. Komponenttien kyselyikkuna ja lopputulos selaimessa.	48
Kuva 25. Kysymys #6.....	49
Kuva 26. Kysymys #8.....	50
Kuva 27. Kysymys #3.....	50
Kuva 28. Kysymys #5.....	51
Kuva 29. Kysymys #7.....	51

Kuva 30. Kysymys #1.....	52
Kuva 31. Kysymys #2.....	53
Kuva 32. Kysymys #4.....	53
Kuva 33. Kysymys #9.....	54
Kuva 34. Vapaita kommentteja elementtien pituuksien laskennasta.....	54
Kuva 35. iTrigger.	56
Kuva 36. Jäähdytysputken ohjeistus.....	57
Kuva 37. Inventor Professional 2024 Ilogic Vault -koodileikkeet.	59
Kuva 38. MoveFile Method. [14]	60
Kuva 39. iMate Properties Matching list.	62
Kuva 40. Elementtien ja jatkojen lisääminen iMatella interaktiivisesti.....	62
Kuva 41. iLogic & iMate yhteistyössä.	63
Kuva 42. iAssembly-rivin vaihto iLogicilla.	65
Kuva 43. Custom iProperties	66
Kuva 44. Mutka jäähdytysputkelle.	67
Kuva 45. Esimerkki napin lisäämisestä.	67

LIITELUETTELO

LIITE 1. Henkilöhaastattelu: Konsultti Rami Nevala, 12/2023

LIITE 2. Mekaniikkasuunnittelun henkilöstökyselyn kysymykset.

LIITE 3. Henkilöstökyselyn vapaat kommentit.

LIITE 4. iLogic automaatti-ikkunoiden käyttöohje

LIITE 5. iLogic käyttöliittymän luonti ja ohjeistus

LIITE 6. Jäähdytysputki-ohjausmallin käyttö

Liitteitä 4–6 ei julkaista liikesalaisuuksien vuoksi.

1 JOHDANTO

1.1 Sisältö

3D-suunnittelun automatisointi ja toistuvien töiden teon vähentäminen on nykypäivänä teknologiateollisuuden suunnittelutöissä erittäin isossa osassa kilpailukyvyn säilyttämisen, työn tehokkuuden kasvattamisen vuoksi ja projektien läpimeinojen lyhenemisen saavuttamiseksi. Tämä opinnäytetyö käsittelee miten pienellä kehityksellä ja 3D-suunnittelun automaatiolla voi tehdä isoja askelia edellä mainittujen asioiden saavuttamiseksi. Opinnäytetyössä käsitellään teoriaa mikä tukee 3D-suunnittelun ja mallintamisen hyviä tapoja automatisoinnin sekä kehityksen kannalta. Tämän lisäksi esitellään VEO Oy:llä kehitystyössä käytetty suunnitteluohjelmisto ja sen automatisointia tukevia ominaisuuksia.

Opinnäytetyöhön on sisällytetty esimerkkejä tehdyn kehitystyön tuloksista, miten kehitystyötä on aloitettu tekemään ja tutkimaan sen hetkisestä lähtötilanteesta, käyttäjiltä kerättyä palautteiden käsittelyä ja sitä miten käyttäjäystävällisyys on otettu kehitystyössä huomioon. Lopuksi käsitellään miten mahdollisesti tehtyä kehitystyötä voisi jatkokehittää tai parannella ja tuoda ilmi muita mahdollisia tarpeellisia kehityksen kohteita VEO:n mekaniikkasuunnittelun automatisoinnissa.

1.2 VEO Oy

Vuonna 1989 Vaasassa perustettu yritys Vaasa Engineering Oy ja nykyiseltä nimeltään VEO Oy on energia-alan näkyvä toimija ja tarjoaa asiakkailleen sähköistys- ja automaatoratkaisut mm. energian tuotannossa, käytössä, siirrossa ja jakelussa. VEO Oy työllistää n. 500 henkilöä ja yrityksen pääkonttori on Vaasassa. Vaasassa VEO:lla työskentelee n. 160 henkilöä. VEO Oy:n liikevaihto oli vuonna 2022 86,5 miljoonaa euroa ja yrityksen liikevaihdosta puolet tulee uusiutuvan energian ratkaisuksista. Tytäryhtiöitä VEO Oy:llä sijaitsee Norjassa, Isossa-Britanniassa sekä Ruotsissa. [1, s. 19; 2]

VEO Oy:llä on liiketoimintaa monella eri energia-alan alueissa kuten mm. tuuli- ja aurinkovoima, vesivoima, energian talteenotto lämpövoima. VEO Oy:n tuotteisiin kuuluu kiskosiltajärjestelmät, kojeistot- ja ohjausjärjestelmät, magnetointijärjestelmät, automaatio- ja suojausjärjestelmät sekä modulaarinen rakennusaluusta E-House sähkönsyöttöratkaisuille ulkokäyttöön. VEO Oy:n vahvuuksiin kuuluu joustavasti räätälöinti asiakkaiden tarpeiden mukaan. VEO Oy:n tavoitteena on vauhdittaa kasvua keskittymällä tukemaan energiateollisuuden vihreää siirtymää uusiutuvan energian ratkaisuilla kaikissa liiketoiminta-alueissa Isossa-Britanniassa ja Pohjoismaissa. [2] Kuvassa 1 esitetään VEO:n liiketoiminta-alueita.



Kuva 1. VEO:n liiketoiminta eri alueilla. [3]

2 TEORIA

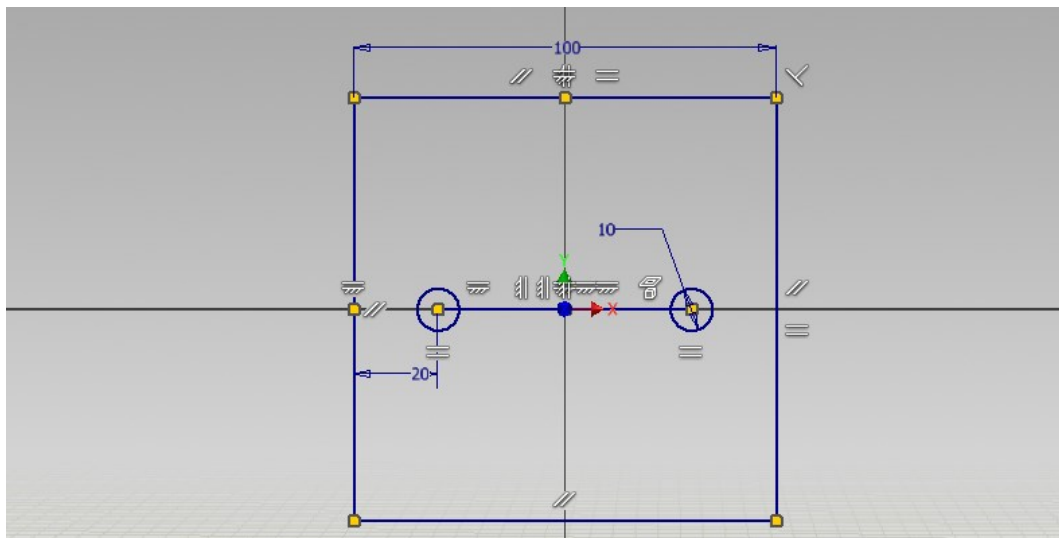
2.1 Parametrinen mallintaminen rajoitteita hyödyntäen

Parametrit ovat 3D-mallinnuksessa erittäin tärkeitä, sillä nykyään jopa 90 % mallinnuksesta tehdään parametrisella mallinnuksella. Parametreihin voi määrittää tärkeitä arvoja, millä mallin mitat ja muut piirteet määräytyvät. Mallien parametrisuus helpottaa mallin muuttamista jälkikäteen. [4, s. 23–25]

Mallia tai suunniteltavaa kohdetta tehdessä ei välttämättä ole vielä kaikki mitat tiedossa, mutta tällaisessa tapauksessa tärkeintä on parametrien käyttäminen eteenpäin katsovalla tavalla. Mallien parametrisuus helpottaa mallin muuttamista jälkikäteen itse parametreilla ja piirteisiin ei tarvitse koskea piirrepuusta. Suunnittelu sisältää paljon muutoksia ja tehtävät muutokset ovat paljon helpompia sekä nopeampia parametrisella mallinnuksella. [4, s. 25]

Mittoja muuttaessa ei myöskään tarvitse luonnosta avata. Tähän riittää, että parametrialikosta muutetaan tiettyä parametria ja malli päivittyy itsenäisesti. Parametrien välille voi luoda relaatioita, jolloin useampi samanlainen mitta muuttuu yhdestä parametrasta ja voi olla myös matemaattisia kaavoja tilanteen mukaan. Parametrisesti mallintaessa tulee muistaa nimetä parametrit asianmukaisesti ja parametreja tulee olla mahdollisimman vähän, mutta riittävästi.

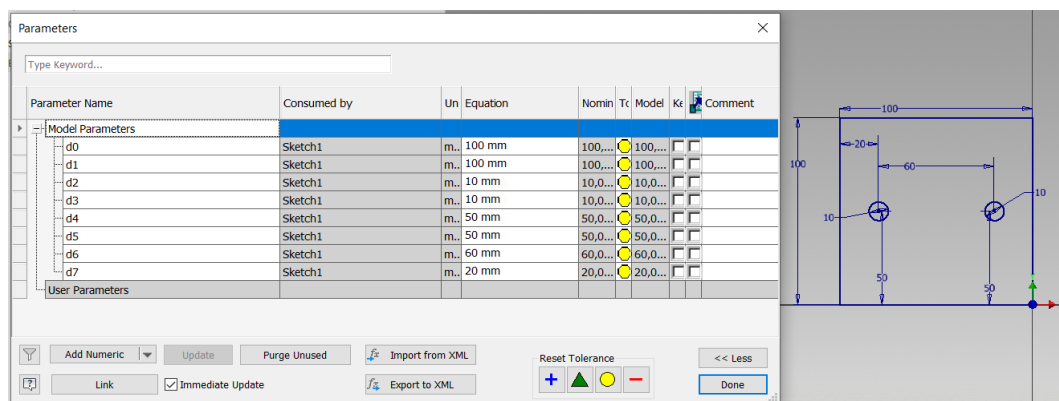
Parametrien käyttö tulee aloittaa jo ensimmäisessä luonnoksessa. Itse parametrien määrään voi tietyssä määrin vaikuttaa ja tähän apuun tulee rajoitteiden käyttö. Luonnostasolla esimerkiksi yhdensuuruus, pysty- ja vaakarajoitteita käyttämällä, vähennät tehokkaasti parametrien määrää systemaattisesti kuten kuvassa 2.



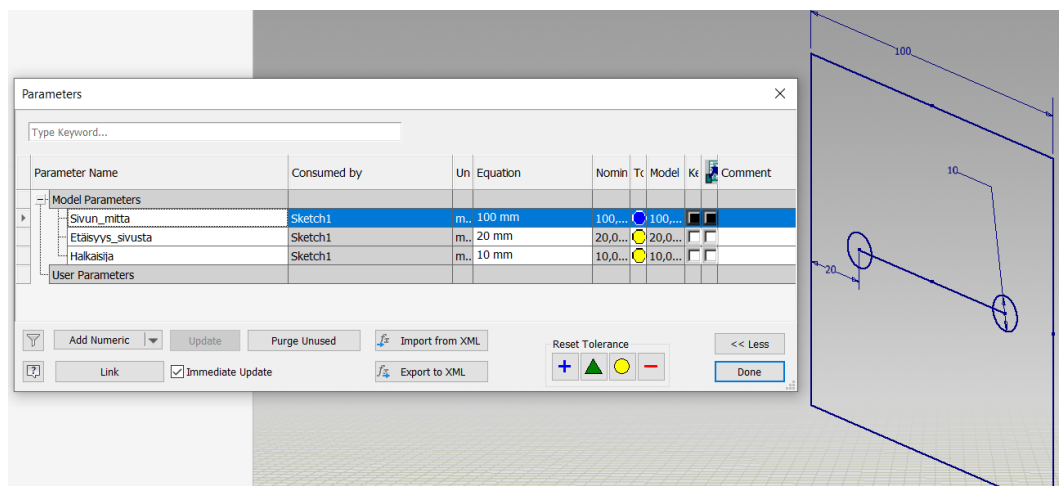
Kuva 2. Luonnosviivat rajoitettuna keskelle koordinaatistoa.

Luonnos kannattaa rajoittaa keskelle koordinaatistoa mallin järkevämpää muokattavuutta ja käytettävyyttä kokoonpanotasolla rajoitteet mielessä pitäen. Tämä on tärkeätä symmetrisen osan kohdalla erityisesti. Rajoittamalla keskelle välttyy myös ylimääräisiltä sijaintimitoilta. Parametrien määrää vähentämällä rajoitteita käyttäen ja parametreja nimeämällä lista pysyy erittäin selkeänä ja on helpompi hyödyntää suunnitteluautomaatiossa.

Mallintajat yleensä kiireen vuoksi valitsee vääränlaisen tien ja ei nimeä parametreja eikä hyödynnä rajoitteita. Tämänlaisen huonon mallintamistavan vuoksi lista paisuu isoksi tarpeettomien mittojen vuoksi kuten kuvan 3 esimerkissä. Mallin seuraava käyttäjä joutuu käydä tarkistamassa kaikki luonnokset, piirteet ja parametrit läpi yksinkertaistakin muokkausta varten. Hyviä tapoja noudattamalla kuin kuvassa 4, voidaan erittäin helposti välttää tällainen tilanne ja säästää aikaa tulevaisuudessa.



Kuva 3. Huonot tavat.

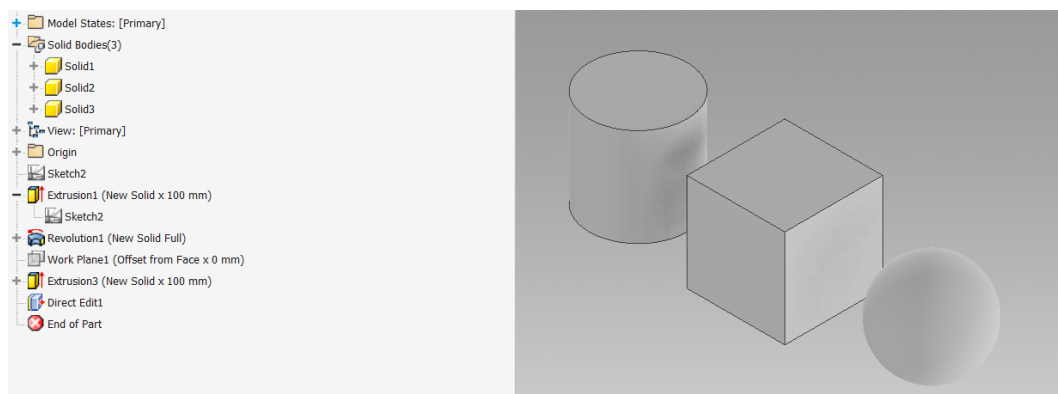


Kuva 4. Parametrit hyviä tapoja noudattaen ja rajoitteita hyödyntäen.

2.2 Multi-Body

Multi-Body 3D-mallit ovat itsenäisiä tilavuusmalleja osatiedostossa, joita voi parametrien avulla muokata erimittaiseksi ja muotoiseksi toisista osatiedostossa sijaitseviin tilavuusmalleihin verrattuna kuten kuvan 5 esimerkissä on esitetty [5]. Tämänkaltainen mallinnustapa on hyödyllinen tapauksissa, jolloin ei haluta esimerkiksi tehdä kokoonpanotiedostoa vaan itsenäisillä tilavuusmalleilla simuloidaan toisia samasta materiaalista tehtyjä osia. Kokoonpanoon tuotaessa voi lisätä esimerkiksi virtuaalisia komponentteja näitä puuttuvia osia paikkaamaan osaluetteloa varten.

Mallinnustapa voi olla hieman riskialttiimpi, koska tiedosto voi helposti kadota tai korruptoitua ja tällaisessa yhdessä tiedostossa voi olla paljon koottua suunnittelu-dataa. Mallin aiempien luonnosten muuttaminen voi myös rikkoa myöhempiä piirteitä. [6, s. 223]



Kuva 5. Itsenäiset tilavuusmallit osatiedostossa.

Multi-Bodyja voi suppress-toiminnolla piilottaa pois käytöstä, jolloin varioitavuuden mahdollisuudet kasvavat. Kun Multi-Body -tyyliset mallit lisätään kokoonpanoon, kaikki kokoonpanossa olevat osat ovat rajoitettavissa myös näihin tilavuusmalleihin lähtökohtaisesti. Multi-Body -malleihin on helppo luoda kokoonpanoihin verrattuna erilaisia piirteitä luonnosten avulla nopeasti [6, s. 223].

Käytännössä suunnittelija ei tarvitse kuin yhden Multi-Body -osatiedoston kokoonpanoon lisätä vähimmillään. Kokoonpanoakaan ei tarvitse, jos osaluettelo ei ole valmistettavuuden kannalta pakollinen, jolloin mahdollisesti itsenäiset tilavuusmallit riittävät kuvaamaan visuaalisesti eri osia. Edellä mainittu tilanne on tietenkin mallista ja valmistajasta riippuva. Multi-Body-mallit ovat suunnitteluautomaatiossa erittäin hyödyllisiä juuri vähäisen tiedostojen määrän vuoksi.

Kun on vain yksi osatiedosto multi-bodyilla mitä muokata, aikaa säästyy erilaisten tiedostojen nimeämisestä ja kopiointia ehkäisemällä. Koska kaikki automaatio ja ohjattavuus voidaan keskittää yhteen osatiedostoon, koodien sekä parametrien luonti vähenee myös huomattavasti. Kokoonpanotasolta pystyy osatiedoston parametreja lisäksi erittäin helposti myös muokkaamaan.

2.3 Ohjausmalli ja Top-Down -menetelmä

Ohjausmalli mahdollistaa 3D-mallin tuoterakenteen ja suunnitteluprosessin kokonaisvaltaisen hallinnan. Ohjausmallilla pyritään ottamaan haltuun tehokkaasti tiedostojen väliset sidokset ja näin myös ohjaamaan mittoja sekä muita piirteitä esimerkiksi kokoonpanotasolta. Ohjausmallina yleensä toimii kokoonpano, mistä piirteitä on linkitetty muihin osiin riippuvaisuuksiksi. Ohjausmalli voi olla myös täysin itsenäinen osatiedosto.

Ohjausmallin hyötynä on, että pelkkää ohjausmallia muuttamalla kaikki muutokset automaattisesti siirtyvät siihen linkitettyihin osatiedostoihin. Muutosten tekoa varten kokoonpanossa voi esimerkiksi olla parametreja, jotka on linkitetty alemmalle tasolle osatiedostoihin tai kokoonpanossa voi olla luurankoluonnos, mistä piirteitä projektoidaan, mitoitetaan, muokataan tai osia paikoitetaan. Ohjausmalli mahdollistaa muokkauksen jälkeen yhteensopivuuden isompaan kokonaisuuteen huomattavasti pienemmällä työllä.

Top-Down -menetelmä on myös ohjausmalleissa erittäin järkevä tapa rakentaa tuoterakennetta ja Nevalan mukaan sen käyttö on jatkuvasti lisääntynyt [5]. Top-Downin ideana on, että kaikki muutokset ja mallinnusta varten tehtävät päätökset tehdään ylemmällä tasolla ja ohjausta sekä muuta suunnitteluprosessia viedään hierarkiaa noudattaen kokoonpanosta osatiedostoihin. Osien välisiä sidoksia vältetään, koska niistä voi koitua myöhemmässä vaiheessa riippuvaisuuksia mistä ei ole hyötyä vaan päinvastoin. Toinen tapa mallintaa on menetelmä nimeltä Bottom-Up. [6, s. 223]

Bottom-Upissa mallit luodaan erillään ja rajoitetaan lopuksi vasta kokoonpanorajoitteilla. On olemassa myös kolmas menetelmä nimeltä Middle-Out, joka on yhdistelmä Top-Downia sekä Bottom-Upia ja on yleisimmin käytetty menetelmä. Middle-Outissa valmiiseen Top-Down -tyyliseen tuoterakenteeseen suunnitellaan tai tuodaan kokoonpanoja sekä osia lisää. Top-Downissa on etuna se, että suunnittelija pystyy havainnoimaan osien välisiä riippuvaisuuksia helpommin, mutta

menetelmä on kuitenkin alkuun aikaa vievä ja tarkempaa suunnittelua vaativa. [6, s. 410] Tuholana ja Viitasen mukaan ”Onnistuneen suunnittelun perusvaatimus on, että valmistuneet osat ja kokonaisuudet sopivat sekä toisiinsa, että kokoonpanoihin” [7, s. 33].

Ohjausmalleista voi tehdä erittäin monimutkaisia kokonaisuuksia, mutta kun noudattaa hyviä tapoja kuten Top-Down -menetelmää, mallin hajoamisen mahdollisuus vähenee virheiden takia. Virheiden korjaus vie aikaa ja se on kallista. Nevalan mukaan ”Ohjausmallin avulla säästetään aikaa ja minimoidaan virheet” [5].

Tuoterakenne tulisi tehdä aina oikeaa logiikkaa käyttäen ja määritettyä tapaa ei aina ole yrityksissä lähdetty muodostamaan. Kokonaisuus on yleensä lähtökohtaisinkin sekava, vaikka sen pitäisi perustua aitoon fyysiseen kokoonpanorakenteeseen. Fyysiseen kokoonpanoon perustuva suunnittelurakenne on eduksi työvaiheille ja myös ystävällinen tehtäville asennuksille. [8, s. 122]

2.4 Konfiguraattori

Kun halutaan ohjausmalleja viedä seuraavalle tasolle, on konfiguraattorit seuraava askel. Konfiguraattori on työkalu millä voidaan 3D CAD -maailmassa räätälöidä määrättömästi erilaisia malleja halutuilla valinnoilla ja tietyillä konfigurointiin ennalta suunniteltujen ja määritetyin säännöin. Malleista voidaan esimerkiksi suunnitella konfiguraattorin avulla automatisoidusti sopivia tuotteita vain yksilölliseen asiakastarpeeseen nopeasti ilman, että konfiguroitavan tuotteen erilaisia variantteja olisi valmistettu valmiiksi varastoon. Nopeuteen tietenkin vaikuttaa myös tuotteen valmistus- ja toimitusaika, mutta kaikki säästetty aika lyhentää toimitusta loppuasiakkaalle. [8, s. 20, 122]

Konfiguroitavan tuotteen variaatioiden määrä on suoraan yhteydessä valintoihin, jotka kerrotaan jokaisen vaihtoehdon lukumäärällä toisiinsa. Tämän vuoksi variaatiota ei ole järkevää suunnitella tai valmistaa valmiiksi varastoon, vaan tarpeeseen.

Tietokoneistettu konfigurointi on nykypäivänä tärkeää ja se estää ihmisten tekemiä virheitä, koska lopputulos on aina sama samoista annetuista arvoista. [8, s. 23]

Automatisoitu konfigurointi on niin nopea, kuin käytetyn tietokoneen tehot sallivat sen olevan. Kustannussäästöt ovat isot, koska tietokone tekee saman työn monta kertaa nopeammin kuin ihminen manuaalisesti. Martion mukaan ”Tyypillinen tuottavuuden kasvu on 1000-kertainen” [8, s. 23]. Moninkertaiset kasvut ovat siksi syynä miksi konfiguraattoreihin siirrytään tuottavuuden nousua tavoitellen. [8, s. 206]

Suunnittelussa ensimmäisenä määritellään konfiguroitavan tuotteen ominaisuudet ja millä arvoilla niitä muutetaan. Nämä arvot ovat parametreja ja parametri voi olla minkä tahansa muuttuva arvo tuotteessa. Konfiguraattoriin voi myös määritellä, että parametrit voi olla tietyssä arvossa, jos toisessa parametrissa on jokin muu arvo. [8, s.27, 30, 206] Parametreille voi antaa maksimiarvot mistä ei poiketa. Tähän voi olla syynä jokin fyysinen rajoite kuten esimerkiksi lämmönkestävyys materiaalilla tai se voi olla edullista toimitusta rajoittava, kuten pituus ja leveys.

Konfiguraattorien avulla tehdään paljon muutakin kuin muutetaan parametreja. Konfiguraattorit voivat tuoda osia kokoonpanoon, vaihtaa niitä tai jopa luoda uusia osia parhaimmassa tapauksessa. Konfiguraattorit pystyvät myös muuttamaan tiedostojen attribuuttitietoja tai luoda sekä päivittää piirustuksia. Hallinta yleensä tapahtuu 3D CAD -ohjelmissa niitä varten luoduissa formeista eli esimerkiksi Autodesk Inventorin tapauksessa lomakkeen tapaisista kyselyikkunoista.

Konfiguraattoreilla voi myös tiedostojen määrää vähentää ja valmiita variaatioita malleista ei tarvitse olla olemassa turhaan. Konfiguraattoreita voi räätälöidä erilaisiin tarpeisiin ja jos niihin yhdistää malleihin luotua koodia, suunnittelu nopeutuu merkittävästi. Konfiguraattorit ovat yleistyneet viimeisen 15 vuoden aikana ja niitä yleensä näkee yritysten verkossa sijaitsevien web-konfiguraattoreiden muodossa. Koska konfiguraattoreiden käyttö nopeuttaa suunnittelua ja niiden muokattavuus vie mallinnuksen eri tasolle, kiinnostus niiden hyödyntämiselle yritysmaailmassa

kasvaa. Konfiguroitavat mallit tarjoavat nopeutensa vuoksi yrityksille kilpailuedun. [8, s. 36, 255]

Epäedullisesti yrityksissä ohjelmointitaitoja vaativa konfiguraattori-mallien ylläpito on suurin haaste ja yleensä syynä mallien käytön lopettamiselle. Jos mallia ei ylläpidetä se ei välttämättä vastaa nykyisiä tarpeita. Erittäin huonoissa tapauksissa konfiguraattori voi olla jo käyttökelvoton tai vanhentunut. [8, s. 263]

2.5 Suunnittelun kehittäminen yrityksissä

Yrityksissä tapahtuva suunnittelun kehittäminen voi olla vaikeata. Yrityksissä yleensä on valmiita toimintatapoja, mitä on vuosia hyödynnetty ja ne eivät välttämättä ole oikeanlaisia tapoja. Yksittäiset työntekijät voi omalla asenteellaan olla juurtuneita vanhoihin toimintatapoihin ja heiltä ei välttämättä löydy halua niitä muuttaa omien periaatteiden takia. Yrityksissä voi myös olla vanhoja 3D-malleja, mitkä ei ole luotu eteenpäin katsovalla tavalla vaan ne ovat kiireellä mallinnettuja. Mallinnustavoissa voi myös olla puutteita osaamisessa tai tietotaidossa, koska ei tiedetä ohjelmiston monista mahdollisuuksista ja lopuksi käytetään vain kymmenesosaa kaikista saatavilla olevista ominaisuuksista.

Suunnittelun kehittämisessä ensiksi täytyy korjata nykyisten työntekijöiden asennetta ja poistaa heitä vanhoista rutiineista. Asenteiden korjaamisen ei löydy helpoja työkaluja, mutta tuomalla esiin uusia toimintatapoja näyttämällä niiden hyödyllisyyden konkreettisesti, asenteet voivat poistua. Pelisääntöjä pitää luoda, että mahdollisia huonoja toimintatapoja voi pysäyttää ja että ne eivät aiheuta enempää vahinkoa, kuin mitä jo ovat todennäköisesti tehneet.

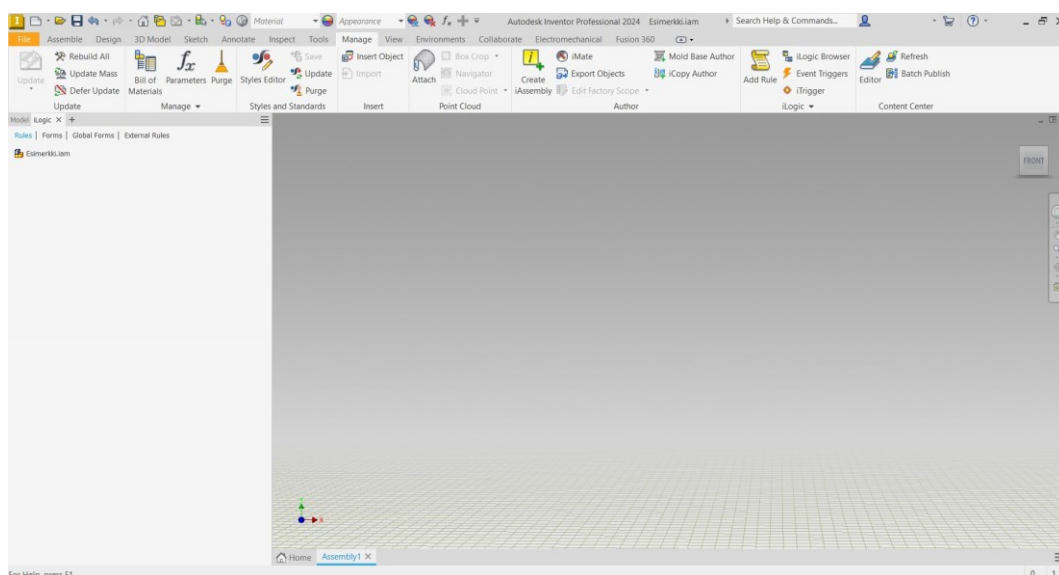
Toimintatapoihin pitää kehittäjän ensimmäiseksi itse tutustua, että voi myös huonoja tapoja tunnistaa. Yritykset eivät itse välttämättä ole huomanneet tietynlaisia huonoja toimintatapoja tai mahdollisuuksia. [5] Yrityksissä on paljon hiljaista tietoa, mitä kaikki työntekijät eivät tiedä ja vain muutamat henkilöt tietävät. Kun tällaiset työntekijät lähtevät ja vievät osaamista pois, syntyy tiedolle tietynlainen

tyhjiö. Dokumentoinnin ja ohjeistuksien tekoon pitää panostaa, että ei synny suunnittelukatkoksia tiedon puuttumisen vuoksi ja tämä voi olla erittäin kallista. Työntekijöitä pitää osata viedä pois toistuvista ja turhista työtehtävistä mitkä voi helposti automatisoida nykypäivänä.

3 SUUNNITTELUOHJELMISTO

3.1 Inventor

Autodesk Inventor (Kuva 6) on vuonna 1999 julkaistu 3D CAD -ohjelmisto, missä on nykyään erittäin kattavat ominaisuudet mekaniikkasuunnitteluun. Inventorista löytyy integroidut ominaisuudet ohutlevyjen, runkopalkkien, putkistojen sekä joh-tojen suunnitteluun. [9] Inventor saa vuosittain päivityksiä uusilla ominaisuuksilla ja ohjelma on tunnettu myös lisäosistaan. Autodesk Inventorissa on käytännössä suuri osa ominaisuuksista mitä mekaniikkasuunnittelija voisi tarvita.



Kuva 6. Inventor Professional 2024.

Autodesk Inventor on erittäin helppokäyttöinen CAD-ohjelmisto ja Autodesk tarjoaa myös sivuillaan erittäin laajat ohjeistukset Inventorin käyttöön. Ohjelmiston suosion vuoksi myös verkosta löytyy todella paljon ohjeistusvideoita käyttäjiltä ja eri yrityksiltä. Inventorissa on myös panostettu varsinkin suunnitteluautomaatioon viime vuosikymmenellä. Suunnitteluautomaatiota varten Inventorista löytyy käyttäjälle ohjeistukset Inventorin ohjelmointirajapintaan eli API:hin ja siihen löytyy myös suoraan ohjelmasta sekä verkosta pääsy sen dokumentointiin. Inventorissa on myös Inventor API:ta hyödyntävä toiminto, joka on kehitetty suunnitteluautomaation helpottamiseksi nimeltä iLogic.

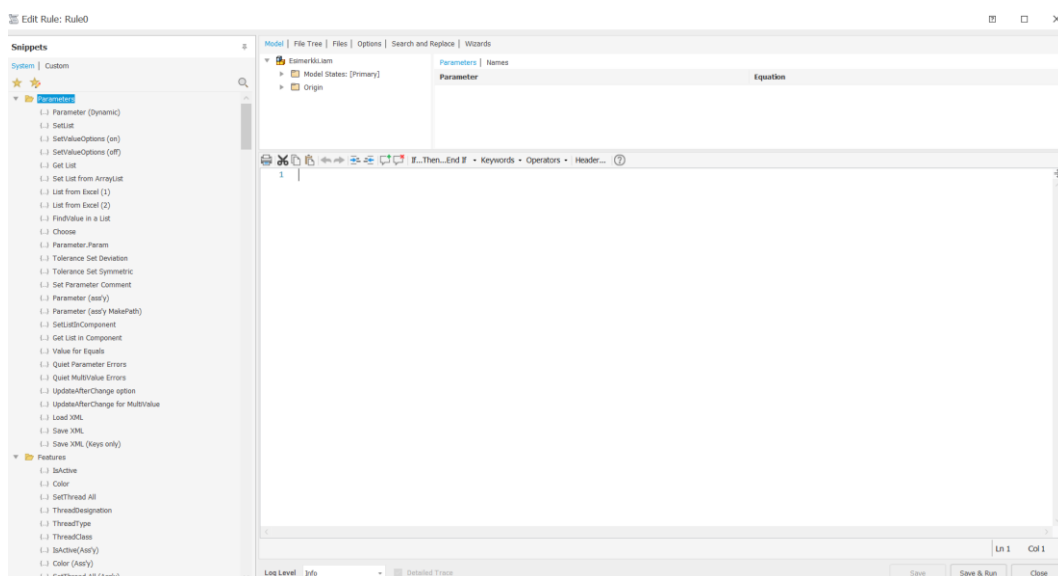
3.1.1 iLogic

Inventorin iLogic oli alun perin Autodeskin ulkopuolinen ohjelmisto, jonka Autodesk osti vuonna 2008 Logimetrix-nimiseltä ohjelmistoyritykseltä. [10] iLogic mahdollistaa suunnitteluautomaation Inventorissa, millä voi tehdä sääntöpohjaista ohjelmointia toistuvien tehtävien ja töiden vähentämiseksi. iLogic mahdollistaa 3D-mallien varioinnin ja konfiguroinnin koodin avulla. Parametrinen mallintaminen on siksi iso osa iLogicia, koska muutamaa tai jopa yhtä mallin parametria muuttamalla voi iLogicin avulla määrätä ehtoja muille toiminnoille mitä käyttäjä normaalisti joutuisi käsin erikseen muokkaamaan.

iLogic tarjoaa alustassaan (Kuva 7) yleisesti käytettyjä snippettejä eli koodileikkeitä ohjelmointia helpottamaan, koska koodileikkeiden sisältämät koodit normaalisti voivat olla moninkertaisia leikkeisiin nähden kooltaan. Koodin luonti iLogicissa on helppoa sillä käyttäjän ei tarvitse olla hyvä ohjelmoija, koska jo If Else -ohjelmoinnilla pystyy saavuttamaan halutun lopputuloksen. iLogic voi myös hyödyntää Inventorin API:ssa olevia koodiesimerkkejä, jos käyttäjä kokee koodileikkeiden olevan riittämättömiä suunnitteluautomaatiota varten.

iLogicissa voi olla kahta eri tyyppiä koodia sisältäviä ruleja eli iLogicin tapauksessa VB.NET-tyyppisiä sääntöjä. On mallien sisältämät rulet, mitkä toimivat vain kyseisen mallin kautta ja sitten on malliin sitoutumattomat external rulet eli ulkopuoliset säännöt. Ulkopuoliset säännöt ovat hyvä siinä tilanteessa, kun haluaa samaa koodia käyttää useammassa tiedostossa ilman niiden sisällyttämistä kaikkiin tiedostoihin.

Parametrit itsessään ovat myös iso osa iLogicia ja niillä voi hallinnoida kaikkea iLogicin ajamaa koodia. iLogic ei kuitenkaan välttämättä tarvitse yhtäkään parametria riippuen ajettavasta koodista. Parametrien ja sääntöjen hallinnointia varten iLogicissa on myös kyselyikkunoiden tekoa varten ominaisuus nimeltä Forms.

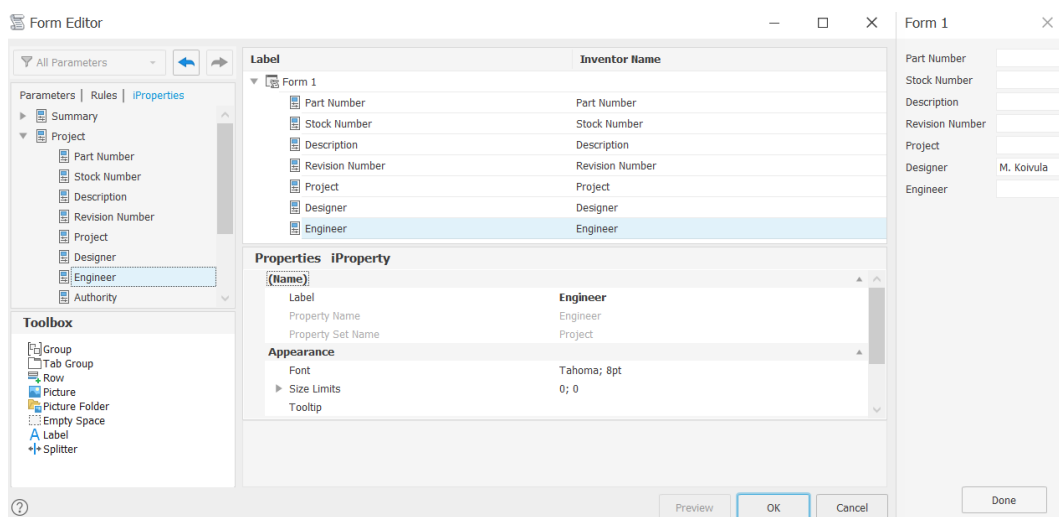


Kuva 7. iLogic-ohjelmointiruutu.

3.1.2 iLogic Forms

iLogicin Forms-ominaisuus tarjoaa helpon tavan hallinnoida käyttäjän määrittelemiä parametreja ja niiden esillepanoa käyttöä varten itse kyselyikkunassa. Kyselyikkuna toimii käyttöliittymänä mallien konfiguroinnille ja niihin on myös mahdollista sisällyttää iLogicin käyttämät rulet, nappeina helpottamaan koodin ajamista vain tietyissä tilanteissa. Tätä varten tietenkin pitää asetuksista varmistaa koodin ajautuvan vain nappia painaessa.

Kuvassa 8 olevan kyselyikkunoiden käyttöliittymän editori pääosin on hieman rajoittunut, mutta toiminto tarjoaa kuitenkin päätarpeet muokkaukselle. Parametrit ja napit voi laittaa vierekkäin tai allekkain, eri välilehdelle ja ryhmitellä laatikkoon. Halutessaan voi lisätä myös tekstiä itse käyttöliittymään tai vihjeitä parametreihin käyttäjää avustamaan. Käyttöliittymään voi myös lisätä kuvia ja niiden vaihtuvuutta voi myös parametrien avulla hallinnoida.



Kuva 8. iLogic Form Editor.

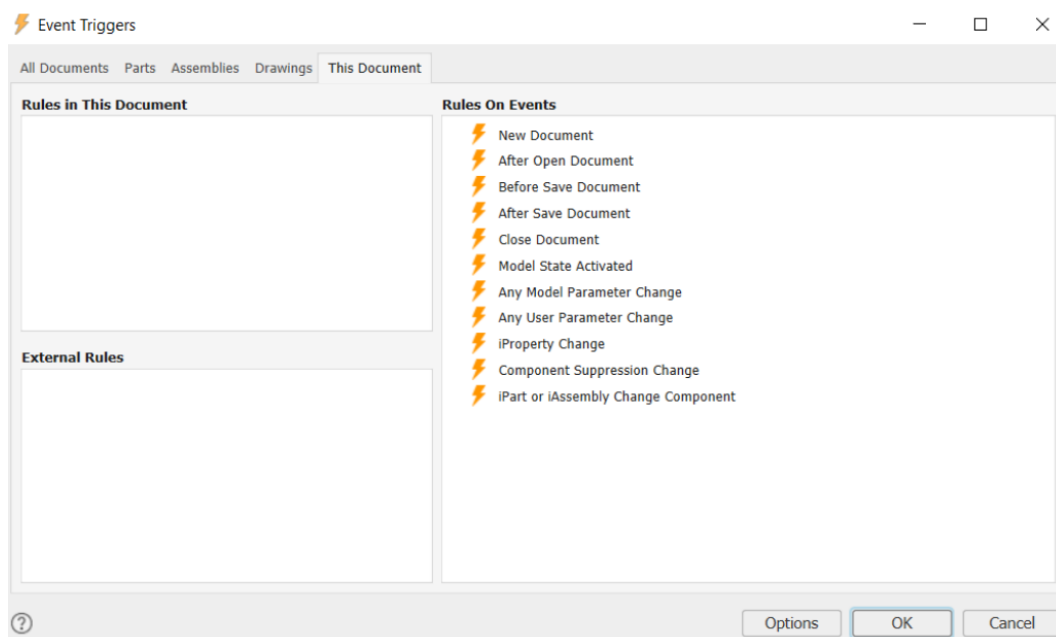
Parametrien hallinnointityyliä voi myös asetuksista vaihtaa. Tyylien määrä ja saatavuus on kaikille parametrityypeille eri. Boolean-tyyppiselle eli True/False -parametrille on kaksi hallinnointityyliä: Ruksattava ruutu tai alasvetovalikko. Tekstiparametrille on tekstikenttä ja valintanappiryhmä. Valintanappiryhmäryhmä on perinteinen valintapallukoilla varustettu radiopainikenippu eri vaihtoehdoille. Numero-tyyppiselle parametrille on tekstikenttä, radiolaatikko tai liukusäädin.

3.1.3 Event Triggers

Event Triggers on iLogicin toiminto, missä tehtyä koodia eli iLogicin tapauksessa sääntöjä voi määrittää ajettavaksi tietyissä Inventorin tapahtumissa. Säännöt voi asettaa tapahtumaan nykyisessä tiedostossa, kaikissa tiedostoissa, osien tiedostoissa, kokoonpanoissa ja piirustuksissa. Event Triggers on hyödyllinen työkalu, jos säännöt haluaa pakottaa ajautumaan näissä tapahtumissa käyttäjän tarvitsematta itse tehdä mitään liittyen koodin ajamiseen.

Kehitystyön aikana huomioimme, että säännöt eivät välttämättä ainakaan Inventor 2021 -tapauksessa ajaudu kokonaan, kun tiettyä parametria muuttaa mikä on linkitetty koodissa olevaan pitkään jonoon ehtolauseita. Tämä korjautuu, kun säännön asettaa kuvassa 9 näkyvään "Any User Parameter Change"-tapahtumaan, jolloin koodi ajautuu kokonaan läpi parametrien muuttuessa toiseen arvoon.

Kaikkia sääntöjä ei tietenkään kannata näihin tapahtumiin linkittää, koska koodissa voi olla jokin ehto mikä tallentaa, vaihtaa tietyn osan kokoonpanossa tai tekee kertaaluontoista muutosta. Tällaisissa tapauksissa käyttäjä voi laittaa yleensä on ohjelmointiruudussa ruksin ”Don’t run automatically”, mutta Event Triggers yliajaisi tämän asetetun ehdon.



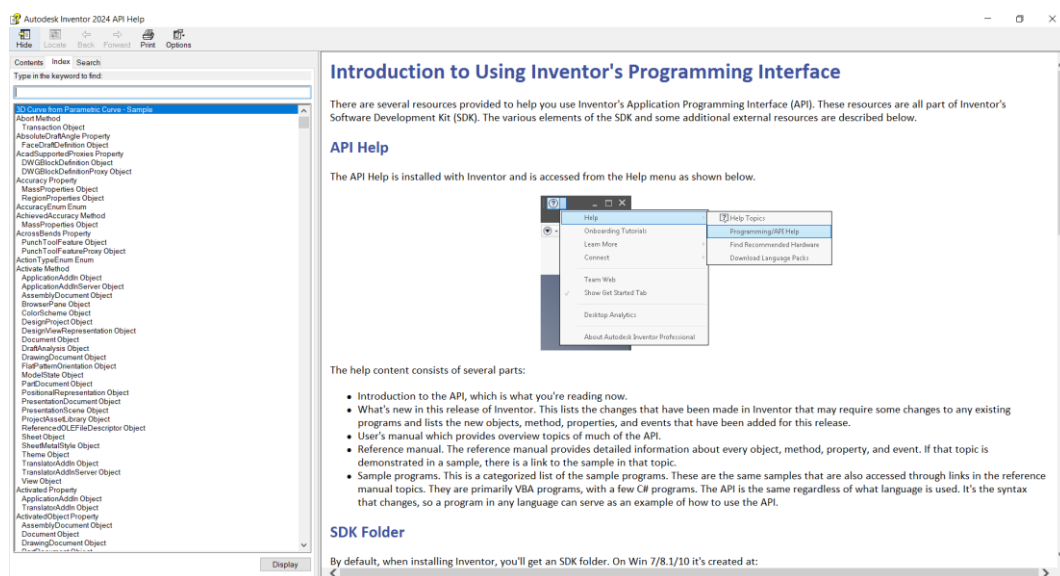
Kuva 9. Event Triggers.

3.1.4 Inventor API

Inventor tarjoaa ohjelmointia varten ohjelmointirajapintaan helpon pääsyn, ja sitä varten Autodesk on tehnyt erittäin kattavan dokumentaation eri käskyille, koodileikkeille ja muulle sen toiminnalle. Inventor API tarjoaa ison mahdollisuuden automatisoida ja jopa parannella Inventorin toimintaa. API:n avulla käyttäjä voi iLogicin puolella luoda pienenä esimerkkinä tiedoston tallennus/avausruutuja, käynnistää Inventorin yläpalkissa olevia eri toimintoja, automatisoida eri tiedostotyyppien luontia käyttäjän määrittelemänä ja ohjeistus tarjoaa jopa opastusta, miten lisäosan voi Inventoriin luoda.

API:ta pääsee tutkimaan Inventorista suoraan yläreunan Help-kuvakkeesta ja sen kautta pääsee avaamaan Programming/API Help-dokumentaation.

Dokumentaatio löytyy myös Autodeskin sivuilta. Inventorin API voi tarjota suuret mahdollisuudet korkean luokan suunnitteluautomaatiolle, mutta se vaatii hieman taitoa ja kokemusta ohjelmoijalta API-sisällön hyödyntämistä varten. Kuvassa 10 näkyvässä dokumentaatiossa on myös kokonaisia esimerkkikoodeja tietyille toiminnolle mitä voi iLogicin avulla käyttäjä kokeilla. Kehitystyössä API:ta hyödynnettiin vain hieman, mutta tulevaisuudessa mahdollisesti se tulee olemaan isommassa osassa.

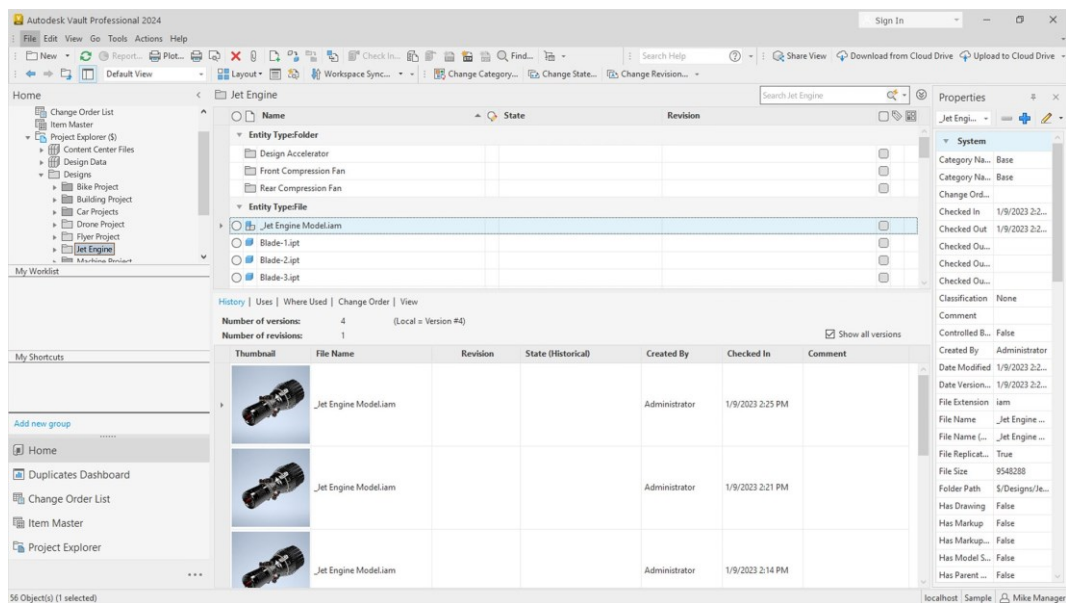


Kuva 10. Inventor 2024 API Help.

3.2 Vault

VEO Oy käyttää tuotetiedon hallintajärjestelmänään Autodeskin Vault-ohjelmiston WorkGroup-versiota. Vault (Kuva 11) toimii Inventorin kanssa oman lisäosansa avulla ja Inventorista voi suoraan malleja, piirustuksia ja kokoonpanoja tuoda sisään Vaultiin ja myös avata niitä sen kautta. Vault tarjoaa saumattoman toiminnan Inventorin kanssa ja VEO:n kokoisessa yrityksessä Vault on välttämätön, koska tiedostoja on satojatuhansia ja työntekijöillä pitää olla pääsy tarvittaessa kaikkiin tiedostoihin mitä muutkin työstävät. Omalla koneella tiedostojen säilyttäminen ei ole vaihtoehto tai ei ainakaan suotavaa.

Kehitystyön kohteena olevat mallit eivät sisällä mitään toimintoja Vaultin kanssa vaan niitä käytännössä vain säilytetään, otetaan käyttöön ja julkaistaan tuotantoa varten Vaultin kautta. Released-tila on iLogic-malleille erittäin tärkeä, koska muulloin kuka tahansa voi mennä muuttamaan alkuperäisestä mallista jotain mikä vaatisi korjaamista seuraavalle käyttäjälle. Vault itsessään tarjoaa myös automaatiota työprosessorinsa ansiosta. Työprosessori automatisoi tarvittavien valmistusta varten tehtävät toisioformaatit kuten .stp, .pdf ja .dwg ”Released”-tilassa. Vault tekee myös BOM Excelin työprosessorissa.



Kuva 11. Autodesk Vault Professional 2024. [11]

4 KEHITYSTYÖ JA TUTKIMUSONGELMA

4.1 Tausta

Kehitystyö aluksi lähti siitä, kun VEO Oy:llä mekaniikkasuunnittelua kehittämässä oleva konsultti Rami Nevala otti yhteyttä silloiseen opettajaani Juha Hantulaan. Nevala tiedusteli, jos Vaasan ammattikorkeakoulusta löytyisi halukas oppilas lähtemään tutkimaan ja kehittämään hänen kanssaan VEO Oy:llä Inventor iLogicin avulla suunnitteluautomaatiota mekaniikkasuunnittelussa.

Koska olin jo aiemmin sopinut kesätyöt VEO:n kanssa mekaniikkasuunnitteluun, Hantula ehdotti tätä minulle ja hänen kehotuksesta otin puhelimitse yhteyttä Nevalaan. Nevalan kanssa sovimme tapaamisen, missä hän esitteli mitä on VEO:lla tekemässä ja mitä minun pitäisi saada aikaiseksi. Tapaamisessa puhuimme, että tutkisin ja opettelisin vapaa-ajallani iLogicia ennen kesätöideni alkua. Hyväksyin tämän toimeksiannon ja Nevala lupasi mentoroida minua Inventor-ohjelman kanssa vastapalvelukseksi, koska hänellä oli siitä vuosien työ-, konsultointi- ja koulutuskokemusta.

Tarkoituksena oli se, että kehittäisin Nevalan kanssa jatkossa suunnitteluautomaatiota VEO:lla kun olen iLogicia opetellut ja myös opettelisin lisää jatkossakin. Nevala itse koki, että hän ei iLogicia ehdi lähteä opiskelemaan ja oli sitä mieltä, että nuorempi sukupolvi saa sen tehdä. Tilannetta helpotti se, että opiskelijat saavat Inventorin ilmaiseksi käyttöönsä, joten kotona opiskelu onnistui vaivatta.

4.2 Tutkimusongelma ja lähtötilanne

Alkuperäinen tutkimusongelma oli selvittää, miten iLogicin avulla saisi Nevalan tekemiä malleja ja niihin luotuja parametreja automatisoitua ja niiden tulisi kuitenkin olla jatkossakin säädettävissä. Lisäksi näitä malleja varten piti saada tehtyä käyttöliittymä helpottamaan suunnittelijaa. Koska VEO:lla ei aiemmin ole ollut tämänkaltaisia konfiguraattoreita ja varioitavat mallit on luotu käytännössä tyhjästä tarpeen mukaan, vaikutus olisi jatkoa ajatellen merkittävä.

VEO:n malleihin Nevala käytännössä oli luonut ja nimennyt alustavat parametrit toimintaa varten, mitä on sitten käyttäjä muuttanut projekteissaan. Jotkin toiminnot olivat parin parametrin muutoksen alla ja käyttäjän kannalta hieman hankala käyttää. Käyttäjän kannalta Inventorin parametrien valikko ei ole ulkonäöltään, sijainniltaan ja toiminnaltaan käyttäjäystävällisin. Parametreja voi olla pahimmassa tapauksessa listattuna satoja ja tilanne on vielä pahempi, kun parametreja ei ole nimetty asianmukaisesti.

Kun avattavaa käyttöliittymää ei ole, mallit käytännössä pitää avata ja etsiä parametrivalikon kautta tarvittava parametri. Inventor 2021 -versio, joka VEO:lla on nykyisin käytössä, ei tarjoa hakutoimintoa parametrivalikossa. Tilanne on huono projektivetoiselle yritykselle, jonka pitäisi saada räätälöitäviä malleja näin aikaa tuhmaavan prosessin vuoksi. Tämän vuoksi erillinen käyttöliittymä olisi erittäin käytännöllinen suunnittelijan tarvitsemille parametrien näkyvyydelle sekä säädölle ja tietenkin suunnittelijan tehokkuudelle.

4.3 Alustava tutkimustyö ja opiskelu

Nevala halusi käyttöliittymän ilmestyvän aina esille ruudulle, kun ohjattava malli avataan ja olisi siitä heti helposti suunnittelijan käytössä muutoksia varten. Tämä käyttöliittymän avaaminen mallissa oli ensimmäinen haaste, mitä lähdin ratkaistaan. Samalla aloitin iLogicin koodikielen opiskelun ja tutkin miten se toimii inventorin puolella käytännössä.

Tutkimuksissani aluksi suuntasin katsomaan Nevalan lähettämiä linkkejä videoihin ja myös Autodeskin sivuille. Heti alussa huomasin ohjeistuksia olevan kattavasti, mutta niitä oli määrällisesti paljon ja hieman hajanaisesti, että tietyn tiedon etsintään voi mennä aikaa. Tutkimusteni alussa löysin jo ensimmäisen tunnin aikana Inventor iLogicin toiminnon ”Event Triggers” ja ensimmäinen ongelma oli käytännössä ratkaistu. Toiminnon avulla sain automatisoitua käyttöliittymän ilmestymään aina tiedoston avatessa. Koodikieleksi selviytyi VB.NET, joka oli helposti opeltavissa.

Nevala tarjosi minulle yhden 3D-mallin mihin pystyin testaamaan ja koodaamaan iLogicin avulla toimintoja. Pidin hänet ajan tasalla oppimastani, vaikka hän ei varsinaisesti iLogicia aikonut opetella, mutta hän halusi kuitenkin tietää sen tarjoamat mahdollisuudet tulevien kehitysten kannalta. Tutkimuksissani keskityin perustointojen käyttöön ja koodin syntaksiin eli miten ohjelman koodia tulee kirjoittaa. Alkumetreillä testailujen aikana kävi jo ilmi, että iLogic tulee tarjoamaan merkittäviä mahdollisuuksia toiminnoiltaan. Tätä alustavaa tutkimusta ja opiskelua tein 1,5 kk kesätöiden alkuun asti.

Kesätöiden aikana pääsin soveltamaan oppimaani käytännössä ja ensimmäiset mallit olivat valmiita käyttöön tästä ”varaslähdöstä” johtuen jo ensimmäisen viikon aikana. Mallit otettiin käyttöön kuitenkin suunnittelijoiden toimesta vasta myöhemmin. Suunnittelijoilta heidän kanssaan keskusteltua sai arvokasta tietoa, mitä valintoja piti olla malleissa ja miten niiden tulisi toimia. Kuten monessa muusakin yrityksessä myös VEO:lla on ”hiljaista” tietoa ja tietyn tiedon saa vain tietyiltä työntekijöiltä, joten tiedustelu oli avainasemassa.

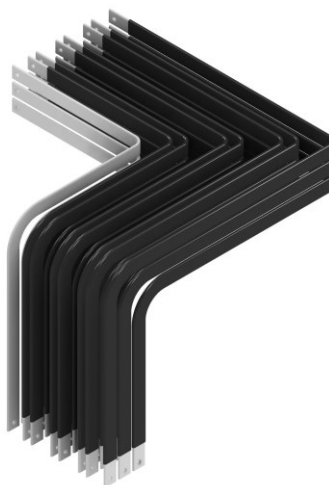
Kehitettäviä malleja oli useampia kesätöiden aikana ja moneen piti etsiä taulukoista standardien määrittelemiä kokoja. Tämänkaltaisissa tapauksissa piti huomioida, miten esimerkiksi standardimukaiset mitat laitetaan koodiin mukaan ja tietenkin käyttäjälle valittavaksi käyttöliittymään varioitavaksi eri asiakastarpeisiin.

Tietyissä malleissa piti ottaa huomioon VEO:n käyttämä lajimerkkiavain, joka on upotettu iProperties-tietoon nimeltä Part Number eli osanumeroon. Tällä lajimerkkiavaimella VEO:lla tavarat tilataan suunnittelijan toimesta ja malleissa piti olla oikea määritelty lajimerkkiavain oikealle variaatiolle. Nämä lajimerkkiavaimet tuli asettaa mallien mittojen mukaan, kun tietty mitta on aktiivinen tai jos jokin muu ehto täyttyy.

5 KEHITYKSEN KOHTEET

5.1 VEBA Kiskoelementit

Ensimmäiset ohjelmoitavat kehityksen kohteet olivat VEBA-kiskoelementit, joiden määrä kehityksen aikana kasvoi 7 erilaiseen elementtiin. Kaikki nämä elementit ovat taitoksiltaan erilaisia, mutta opinnäytetyössä on rajattu käsittely vain yhteen kiskoelementtiin. Kiskoelementit ovat pääpiirteittäin samanlaisia, mutta niiden taivutukset ovat erilaisia. Opinnäytetyöhön esiteltäväksi valikoitiin haastavin näistä eli kuvan 12 tuplataivutettu kiskoelementti.



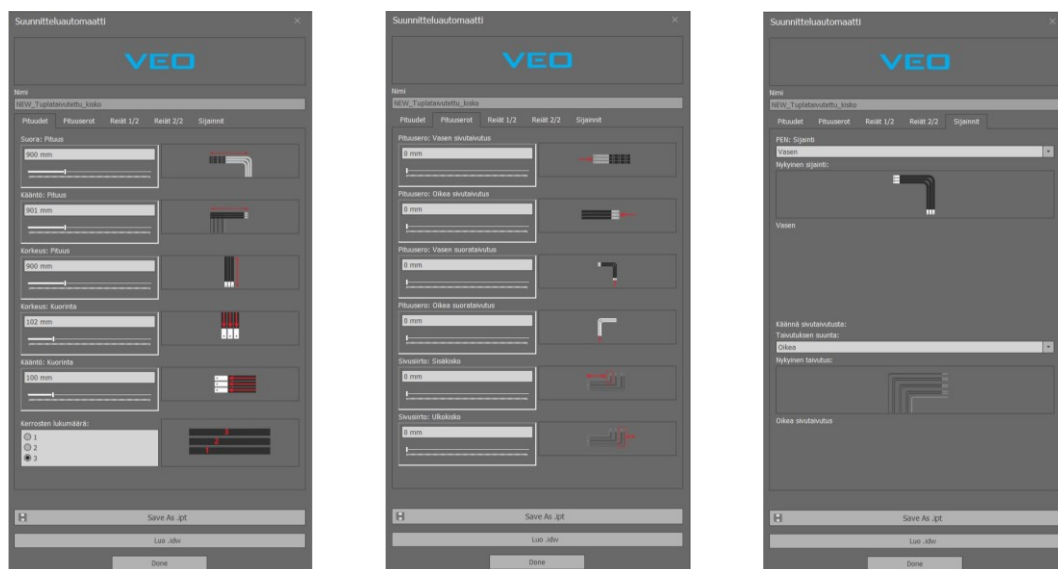
Kuva 12. Tuplataivutettu kiskoelementti.

Kiskoelementissä piti olla erilaisia valintoja koskien niiden liitettävyyttä muuntajiin ja toisiin kiskoelementteihin. Näihin tiedot sain kiskosiltasuunnittelijoilta Inventoria ja projekteja käsittelevässä palaverissa [12]. Samalla elementeissä piti pystyä kerrosten määrää muuttamaan sekä siirtämään PE- ja PEN-kiskoja sijaintia puolelta toiselle. Tuplataivutetun kiskoelementin kohdalla piti myös pystyä kätisyyttä muuttamaan.

Kiskoelementtejä varten laadin jokaiseen helppokäyttöisen oman käyttöliittymän kyselyikkunoihin ja lisäsin kaikki koodia hyödyntämättömät perusmittoja

muuttavat parametrit niihin. Nämä olivat helposti suoraan vedettävissä parametrien listasta kyselyikkunaan. Käyttöliittymässä on myös muutama olennaisesti kiskolementtejä muuttava koodia vaativa parametri.

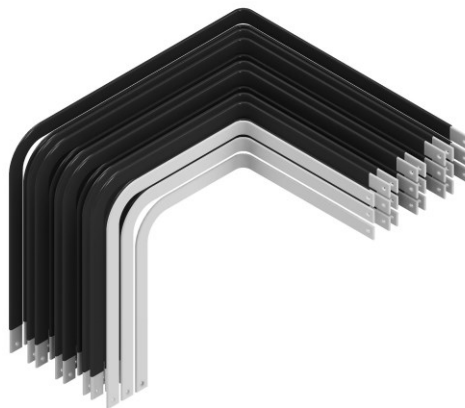
PE- ja PEN-puolen muutoksen sain alustavissa tutkimuksissani muuttumaan ennen kesätöitä, joten tämä oli käytännössä ratkaistu heti alkuun. Se tapahtuu IF Else -ehtolauseilla ja tämän avulla iLogic muuttaa muutaman parametrin arvon mitkä oli valmiiksi liitetty mallissa oleviin piirteisiin. Normaalisti tämä olisi pitänyt parametrivalikosta mennä muuttamaan useammasta parametrusta ja se oli aikaa vievää. Muutos onnistuu nyt kuvassa 13 näkyvässä yhdellä parametrilla missä on valinnat: ”Vasen” ja ”Oikea”.



Kuva 13. Esimerkkejä kyselyikkunan välilehdistä ja valinnoista.

Kätisyyden muuttaminen oli vaikein tehtävä. Nevalan kanssa yhteistyössä teimme useammat toiminnot mallipuuhan Inventorin perustyökaluilla kiskojen kätisyyden vaihtamiseksi ja iLogic ei tuntunut toimivan yhden kriittisimmän kanssa näistä toiminnoista. Toiminto on ”Direct Edit” ja sen aktivoiminen sekä deaktivoiminen ilmeisesti korruptoi parametrin jatkuvasti ja tämä kyseinen parametri piti poistaa. Tämä on mahdollisesti jokin vanhan Inventor-version ongelma. Parametria kokeiltiin linkittää toiseen toimintoon ilman, että iLogic oli liitetty siihen ja se ei toiminut

myös näinkään. Direct Edit -toiminnolla tässä tapauksessa kierretään toinen sivutaivutettu elementinpuolikas 180 astetta ja sen jälkeen yhdistetään toisilla toiminnoilla aiemmin keskeltä ”katkaistuun” puoleen takaisin.



Kuva 14. Kiskoelementin taivutuksen suunta ja PEN-puolet vaihdettuna.

Tämän toimintaan saattamiseksi jouduimme ilman toimintoihin linkitettyjä parametreja tekemään muutaman erilaisen säännön iLogicissa. Säännöt käynnistyvät toistensa sisältä RunRule-koodileikkeellä. Näiden sääntöjen avulla aktivointi ja deaktivointi toimii. Jos kaikki ehdot olivat yhdessä säännössä, kätisyyden vaihto ei toiminut. Kätisyyden vaihdossa piti ottaa koodeissa myös huomioon tiettyjen parametrit pitävän olevan ”peilikuvana”, koska tietyt kiskot ovat käytännössä väärinpäin. Tämä kuitenkin oli aivan toteutettavissa ehtolauseilla.

5.1.1 Katkaisu- ja taivutusmitat

Kiskoelementtien valmistusta varten VEBA-suunnittelijat joutuvat Excelin avulla laskemaan jokaisen kiskoelementin kiskojen katkaisu- ja taivutusmitan. Tämä on turhauttava ja aikaa vievä prosessi, jos kiskoelementtejä on paljon. Suunnittelija joutuu kaikki kiskot laskemaan uusiksi mittojen muuttuessa ja kaikki tiedot pitää

syöttää manuaalisesti. Kiskoelementeissä voi olla myös lyhennyksiä tietyissä kiskoissa ja se täytyy ottaa myös huomioon.

Tämän välttämiseksi Nevala oli alustavasti kehitellyt tapaa missä parametrit linkitetään piirustuksissa oleviin tekstimuotoisiin symboleihin ja näissä näkyisi mitat suoraan parametreista piirustuksissa. Ajansäästö olisi iso tämän toimiessa. Aluksi mitat tulivat luonnosten referenssimittojen parametreista. Tämänkaltaisen ratkaisu ei kiskoelementtien kohdalla toiminut, koska kiskojen suunnat myös muuttuvat mikä katkaisee mittaviivat luonnoksissa.

Mittoja varten sain tutkittavakseni myös vanhan Excelin millä elementtejä on laskettu, mutta siinä oli laskukaavoissa virheitä ja siitä ei ollut lopussa apua. Uusi Excel oli myös kehitelty aiemmin, mutta sitä ei ollut otettu käyttöön tietojeni mukaan syystä tai toisesta. Tutkiessani uutta Excel-tiedostoa, huomasin myös siinä olevan kaavavirheitä ja se johti kiskojen uusien testikappaleiden ottoon. Kiskoista otettiin mitat ennen ja jälkeen taivutuksen. Päädyimme ratkaisuun, että laskemme kiskojen mittaerot neutraalista linjasta, mikä kulkee kiskoelementin keskeltä elementin päätyihin ja lisäisimme kaavoihin mahdolliset mittakorjaukset taivutuksista testikappaleiden perusteella.

Parametrit tein kiskokohtaisesti ja niitä varten Nevala teki taustalla kuvassa 15 näkyviä tekstimuotoisia symboleita. Kaikille elementeille on omat kaavat ja ohjauksellisten variaatioiden vuoksi iLogicin apu oli kaavoissa normaaliin parametrien rajoitteiden vuoksi erittäin hyödyllinen. Tuplataivutettu kiskoelementti oli haastavin näissä kaavoissa, koska elementille oli käytännössä 4 eri variaatiota mitkä piti ottaa huomioon kaavoissa.

PE ja PEN eri puolilla vaihtavat hieman etäisyyttä toisiin kiskoihin nähden ja käti-syyden vaihto muuttaa mitat ja edellä mainitut peilikuviksi, jolloin myös kaavan pitää olla peilikuva. Kaavoihin piti myös lisätä eri lyhennykset sivuilla oleviin kiskoihin. iLogicissa tehdyn helpon ehtolausekoodin avulla pystyin tekemään laskukaavoista dynaamiset.

Katkaisupituudet kerros 1: Taivutustiedot kerros 3

Kisko 1=2186 mm	Kisko 1 = 548,5 mm, 742,5 mm
Kisko 2=2113 mm	Kisko 2 = 628,5 mm, 752,5 mm
Kisko 3=1982 mm	Kisko 3 = 653,5 mm, 752,5 mm
Kisko 4=2666 mm	Kisko 4 = 749,5 mm, 723,5 mm
Kisko 5=2554 mm	Kisko 5 = 768,5 mm, 712,5 mm
Kisko 6=2868 mm	Kisko 6 = 838,5 mm, 622,5 mm
Kisko 7=2898 mm	Kisko 7 = 886,5 mm, 752,5 mm
Kisko 8=3199 mm	Kisko 8 = 1036 mm, 730 mm

Kuva 15. Esimerkkejä tekstisymboleista.

Pituuksia eri suunnittelijat kävivät testaamassa jälkeinpäin ja kaavoihin tehtiin korjauksia saaduista mitoista. Huomasimme taivutusten lopputuloksista, että mitat eivät pitäneet paikkansa myöskään korjausten kanssa. Saadut taivutusten jälkeiset mitat olivat aina eri kerta toisensa jälkeen. Ongelman syyksi esiintyi eri työntehtäjät taivuttamassa ja vanha taivutuskone.

Taivutuskoneella tehtyjen testien mukaan, kone ei anna samaa mitta taivutuksen jälkeen kappaleelle ja taivuttajiin itsessään kaavoilla ei voi tietenkään vaikuttaa. Päädyimme ottamaan mitoista tehdyt pituuskorjaukset pois ja siirryimme käyttämään taivutusten sisämittaa kaavoissa. Tämänkaltaista satunnaisesti muuttuvaa virhettä ei Inventorin avulla voi korjata, vaan se vaatii vähintään taivutuskoneen päivitystä.

5.1.2 Automaattinen piirustus

Suunnittelijoiden työtä nopeuttamaan ja helpottamaan ehdotin, että käytämme iLogicia tekemään Inventor API:n pohjalta automaattiset työ kuvat ja sen avulla myös lisäämään tehdyt symbolit pituuksille. Suunnittelijan ei tarvitse itse etsiä tiettyjä sopivia symboleita isosta listasta vaan koodi lisää määritetyt symbolit oikeille kiskoelementeille. Muita mittoja symbolien lisäksi koodi ei tee liiallisen koodin paisunnan ja myös kiskoelementtien mallien yleisen haasteellisuuden vuoksi. Pintoja pitäisi nimetä mallissa koodia varten, että iLogic osaa laittaa mittaviivat

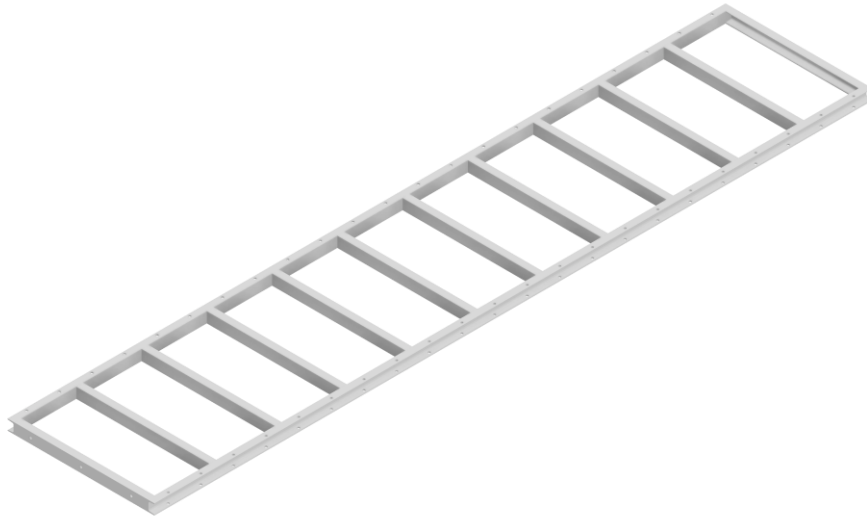
oikeille paikoille. Koska monia kiskoja poistetaan tietyissä tapauksissa elementeistä, ei se ollut ajankäytöllisesti järkevää.

Koodi tekee peruskuvannot ja lisää elementtien kerrostiedot oikeille paikoille sekä kiskojen järjestysnumerot kuvannon alle. Symbolit lisätään myös piirustukseen samassa koodissa vierekkäin sopivaan paikkaan. Koodi ottaa huomioon tekovaiheessa kerrosten lukumäärän ja jättää laittamatta tietyt symbolit. Automaattisen piirustuksen saa luotua suoraan kiskoelementtejä varten tehdystä käyttöliittymästä.

5.2 Marine-alusta

Kuvassa 16 Marine käytössä olevalle varioitavalle runkosokkelille täytyi kehittää myös iLogic-konfiguraattori. Nevala oli tätä kehittänyt aiemmin jonkin verran ja hänen keskeneräinen tuotoksensa oli ollut hieman testikäytössä VEDA-puolen suunnittelijalla. Marine-alustaan oli upotettu Excel-tiedosto parametrien hallintaa ja valintoja helpottamaan, mutta Excelin kautta sitä ei ollut kovin mukava käyttää, oli hidasta ja epäselkeää. Marine-alustaan päätimme kehittää yhtä helppokäyttöisen käyttöliittymän kuin kiskoelementteihin.

Marine-alusta oli hyvin yksinkertainen malli, puhtaasti Multi-Body -tyyliin toteutettu ja mallissa ei vielä ollut kaikkia vaadittuja toimintoja. Palkkeja piti yksitellen ja käyttäjän määrittämänä saada poistettua alustan välistä. Maadoitusreiän sijaintia piti onnistua myös säätämään kenttien välillä.



Kuva 16. Marine-alusta.

Tehdyssä kuvan 17 konfiguraattorin käyttöliittymässä pystyy myös alustan syvyyttä muokkaamaan ja niin kutsuttujen kenttien lukumäärää. Syvyyttä muokattaessa yli rajan tiettyyn mittaan, ohjelmoitu koodi lisää sivuille ylimääräisen pultin reiän. Alustassa piti olla myös eri koot UNP-teräspalkkien kokoja varten ja niistä on nyt iLogicissa standardikoot listattuna käyttäjän valittavaksi. UNP-koon valinta vaihtaa myös materiaalin, joka on kopioitu Steel-materiaalista useammaksi ”UNP & koko”-nimelle piirustusta varten.

Itse käyttöliittymä on yksinkertainen ja ei sisällä kuvia. Jokaiselle kentälle, joita alustassa on 11 kpl on omat valintansa ja käyttöliittymän toimintaa on myös ohjelmoitu. Boolean-tyyppisiä parametreja on hyödynnetty sulkemaan tietyt parametrit, kun esimerkiksi valitaan tietyt kentät vain aktiiviseksi. Tämän viimeisen valitun kentän ”Ei”-arvo sulkee sen jälkeiset kentät ja ne saa vain yksitellen päälle numerojärjestyksessä jälkikäteen virhetilanteiden välttämiseksi mallissa olevissa piirteissä.

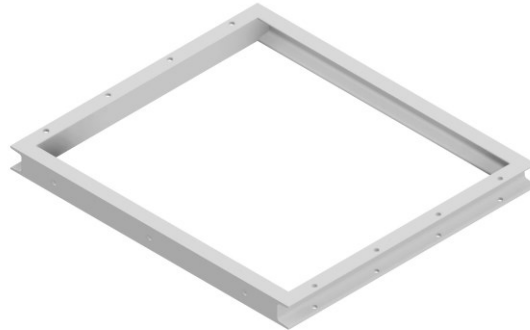
The image shows three sequential screenshots of the VEO Marine hull design software interface, specifically the 'Suunnitteluautomaatti' (Design Automation) window. The interface is in Finnish and is titled 'VEO'.

- Left Screenshot:** The 'Kentät' (Fields) tab is active. It displays a list of 11 hull stations (Kenttä 1 to Kenttä 11). Each station has a 'Käytössä' (In Use) checkbox and a 'Leveys' (Width) dropdown menu, all set to 'Kyllä' (Yes) and '400 mm' respectively. At the bottom, there is a 'Save New .ipt and .jdw' button and a 'Done' button.
- Middle Screenshot:** The 'Rungon tiedot' (Hull Data) tab is active. It shows various parameters for the hull, including 'Rungon tyyppi' (Hull type) set to 'Suora' (Straight), 'Rungon kiinnitys Vasen' (Left hull attachment) set to 'Vapaa' (Free), and 'Rungon kiinnitys Oikea' (Right hull attachment) set to 'Kierte' (Screw). It also includes fields for 'Rungon lyhenne kiinnitykseen Vasen' (Left hull attachment abbreviation) and 'Rungon lyhenne kiinnitykseen Oikea' (Right hull attachment abbreviation), both set to '2 mm'. The 'Maadotuspiste' (Datum point) is set to 'Kenttä 11', and 'Kiinnitys' (Attachment) is set to 'Vapaa'. The 'Maadotusliiri' (Datum offset) is set to '0 mm'. The 'Vasen päätypalkki auki' (Left end beam open) is set to 'Sisään' (In), and 'Oikea päätypalkki auki' (Right end beam open) is set to 'Sisään' (In). The 'Takapalkki auki' (Rear beam open) is set to 'Ulo' (Out). The 'Rungon tiedot' (Hull data) section shows 'Syvyys' (Depth) set to '900 mm' and 'Pituus' (Length) set to '18400 mm'. At the bottom, there is a 'Save New .ipt and .jdw' button and a 'Done' button.
- Right Screenshot:** The 'Profiilin tiedot' (Profile Data) tab is active. It shows the 'Profiili' (Profile) set to 'UNP 65'. The 'Korkeus' (Height) is set to '85 mm', 'Leveys' (Width) is set to '42 mm', 'Sisäpöyryntys' (Internal curvature) is set to '7.5 mm', 'Ulkopöyryntys' (External curvature) is set to '4 mm', 'Sedämä' (Sediment) is set to '5.5 mm', and 'Ulkonurkka' (External corner) is set to '7.5 mm'. Below this, there are 10 'Valpalkki' (Ribs) sections, each with a dropdown menu set to 'Kyllä' (Yes). At the bottom, there is a 'Save New .ipt and .jdw' button and a 'Done' button.

Kuva 17. Marine-alustan kyselyikkuna välilehtineen.

Suunnittelijat myös toivoivat mallille kykyä Save As -toiminnossa piirustuksen mukaan saamiselle. Mallille on tehty koodi, joka normaalisti tallentaa mallin uudelle nimelle, avaa vanhan mallin nimellä olemassa olevan piirustuksen ja yksinkertaisesti vaihtaa mallin referenssin uuteen tehtyyn kopioon mallista sekä tallentaa piirustuksen samaan sijaintiin malliin sopivalla nimellä. Etuna tässä prosessissa on, että suunnittelijan ei tarvitse mallia heti laittaa Vaultiin säilytettäväksi tai käyttää Vaultin Copy Design -toimintoa.

Mallin käyttöönotossa ilmeni alussa hieman ongelmia luonnoksen rajoitteiden rikoutumissa ja tietyissä piirteissä olevissa päällekkäisyyksissä. Ongelmat on niiden löytymisen jälkeen korjattu mallissa välittömästi. Marine-alusta on ollut tämän jälkeen erittäin suuressa käytössä, ja ongelmia ei ole alussa olleiden lastentautien jälkeen oikeastaan ilmestynyt myöhemmin.

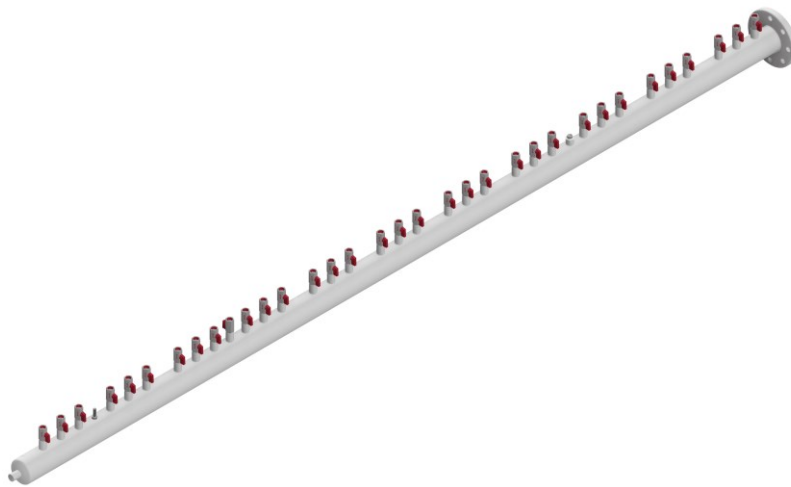


Kuva 18. Alusta kahdella kentällä ja välipalkki poistettuna.

5.3 Jäähdytysputki

Kuvan 19 jäähdytysputki oli ensimmäinen malli minkä olin nähnyt vilaukselta ennen kesätöitä ja on kehityksen kohteista ollut laajin yksittäinen malli. Malli on myös Multi-Body -tyyliin toteutettu ja Marine-alustan tavoin myös siihen oli upotettu Excel-taulukko. Jäähdytysputken kehitys vain ei ollut edennyt tästä eteenpäin hetkeen. Excel-taulukosta otin tietenkin vain tarpeelliset parametrit, mutta malliin oli luotu paljon parametreja ja niiden selvittely vei aikaa. Näiden parametrien lisäksi piti luoda erittäin mittavasti lisää parametreja käyttöliittymän valinnoiksi.

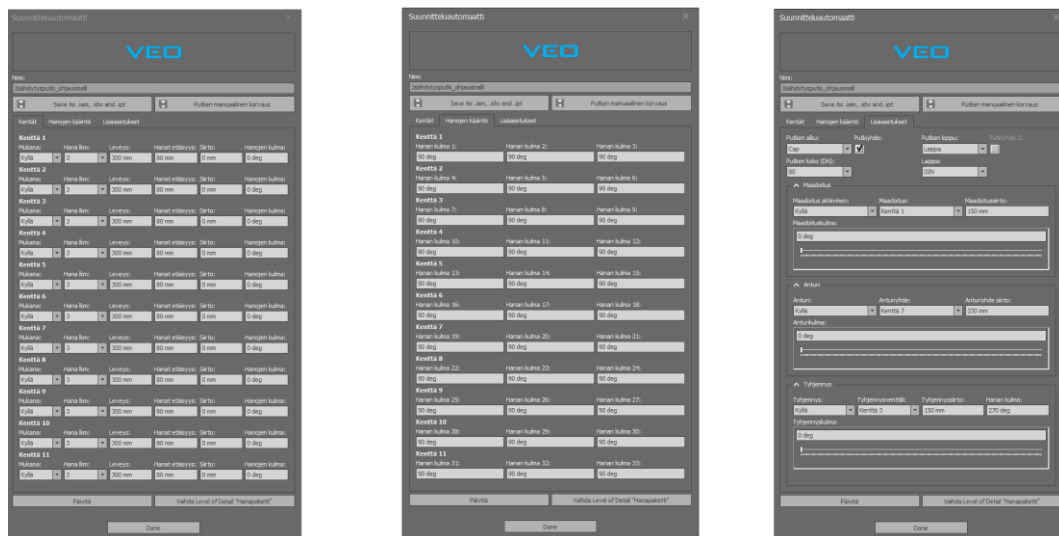
Jäähdytysputkea tultaisiin käyttämään keskuksissa samalla "Kenttä"-tyylillä kuin Marine-alustaakin ja tämä antoi tilaisuuden kopioida jo luotua koodia uudelleenkäyttöä varten. Käyttöliittymän sulkutoimintoa ei tarvinnut myöskään ohjelmoida uudelleen, mutta parametrit sitä varten piti luoda. Putkessa oli alun perin vain 7 kenttää ja sitä päätettiin laajentaa myös 11 kappaleeseen. Kyselyikkunan käyttöliittymästä paisui valintojen määrän vuoksi erittäin iso, koska putkessa oli todella paljon "yhteitä" hanojen ja putken väliin. Hanoissa piti olla myös kääntyvyyttä hallinnoivia valintoja.



Kuva 19. Jäähdytysputken kokoonpano.

Puolessavälissä jäähdytysputken kehitystyötä päätimme luoda osamallista kuitenkin kokoonpanon. Tämä muutos tehtiin sen vuoksi, että pystyisimme hallitsemaan myös itse hanakomponenttien läsnäoloa ja kääntöä. Multi-Body -jäähdytysputken valinnoissa säädetään vain putken ympäri menevää kiertoa. Muutosta varten aiemmin luotu kyselyikkuna kopioitiin kokoonpanoon ja myös valintaparametrit otettiin kokoonpanoon export-toiminnolla.

iLogic tarjoaa helpon tavan hallita osamallin parametrejä, ja tätä varten ei tarvinnut muuta toimenpidettä kuin linkittää kokoonpanossa olevat parametrit osamallin vastaaviin koodileikkeiden avulla. Kokoonpanosta hallinta alaspäin on tietenkin Top-Down -menetelmää ja ei ole kovin suotavaa pitää 2 erilaista kyselyikkunaa samalle ”mallille” ja täten kaikki uudet hallintaparametrit tehtiin kokoonpanossa olevaan kyselyikkunaan. Tämä voi aiheuttaa ristiriidan kuvan 20 käyttöliittymässä, joka ei päivitä alhaalta ylös tehtyjä valintoja ja voi näyttää väärän valinnan kokoonpanotasolla.



Kuva 20. Jäähdytysputken kyselyikkuna välilehtineen.

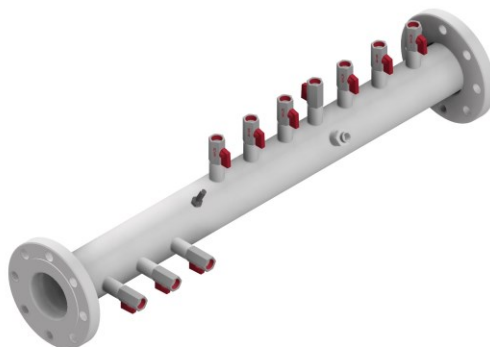
Hanojen käytöstä pois ottaminen ei kokoonpanossa riittänyt vaan myös niiden rajoitteille piti tehdä koodiin samat toimenpiteet kuin itse hanoille. Tämän jälkeen ongelmaksi muodostui, että miten itse jäähdytysputki tallennetaan uudelle nimelle. Marine-alustan tallennuskoodin pystyi uudelleenkäyttämään, mutta se ei tallenna itse osia uusiksi varianteiksi. Tämä vaati hieman tutkimista, jota tein myös vapaa-ajalla.

Tätä ongelmaa tutkin Inventor API:n avulla ja löysin tavan käynnistää Inventorin Save & Replace -toiminnon iLogicilla. API:ssa niin kutsuttu Command Manager hallinnoi Inventorin yläpalkista olevia toimintoja ja komento niille on käytännössä sama eri nimellä tietenkin itse koodissa. Tämän komennon lisäsin For Each -koodiin, joka käy osien esiintymät läpi kokoonpanossa tiettyä nimeä etsien ja valitsee automaattisesti määritetyn osan. Nimi tietenkin on hyvä vakioida kokoonpanon selaimessa, että koodit toimivat vielä uuden osan nimellä. Tätä kutsutaan selainnoodin vakiinnuttamiseksi [13].

Koodin lisäys tallennuskoodin perään riitti ratkaisemaan tämän ongelman. Nyt kokoonpano, piirustus ja osa tallentuvat uudelle nimelle. Tämä lisätty koodi ei toki toimi, jos tallentaa Vault-projektitiedostossa määritetyn polun ulkopuolelle. Toisen ongelma muodostui Inventorin toiminnosta nimeltä Level of Detail, joka tulee

käyttöön heti kun jokin osa otetaan pois käytöstä kokoonpanossa. Hanojen poisto ei toimi koodin avulla, kun ei ole oikea Level of Detail päällä. Ongelma tulee ratkeamaan käytännössä itsestään, koska uusimmissa Inventor-versioissa Level of Detail -ominaisuutta ei ole. Toistaiseksi käyttäjä joutuu itse sen vaihtamaan manuaalisesti.

Jäähdytysputken koodissa on myös DN-standardikoot putkelle ja laipalle on myös DIN- sekä JIS-tyyppiset arvot listattuna ehdoissa eri projektitarpeille. Putken päädyn voi muuttaa tulpatuksi, laipalla, levyllä, avonaiseksi tai ”putkiyhteellä” varustetuksi. Kuvassa 21 on edellä mainitusta pieni esimerkki kahdesta laipasta. Käyttöliittymässä on välilehdille jaettu valintaparametrit järjestyksessä sekä selkeiksi kokonaisuuksiksi.

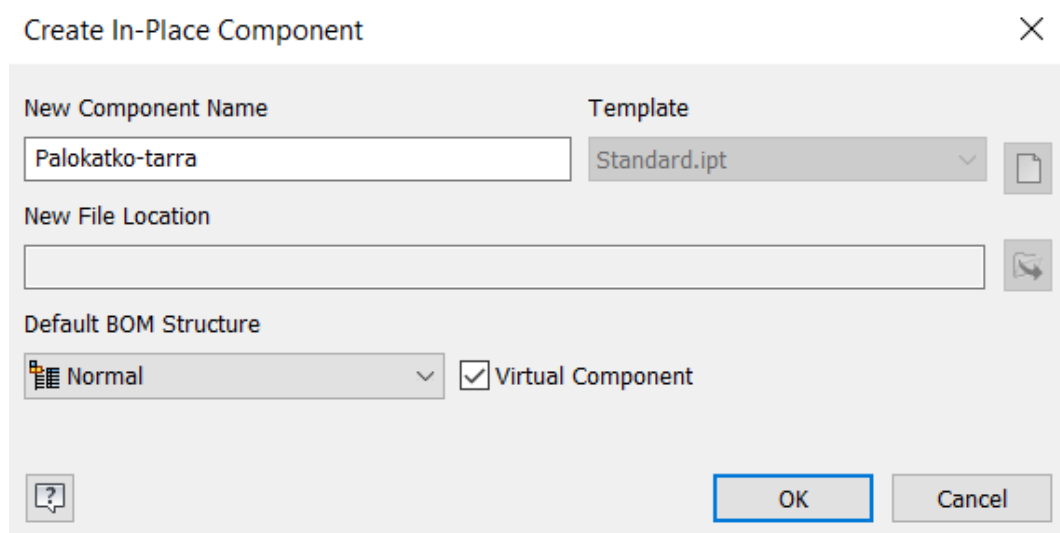


Kuva 21. Jäähdytysputki varioituna kahdella laipalla.

Jäähdytysputken käyttöliittymää parantelin vielä käyttöönoton jälkeen ja siihen lisäsin tietyille kääntyville osille liukusäätimet. Nevala omasta puolestaan lisäsi mahdollisuuden putken toisen puolen päätyyn samat vaihtuvat osat mitkä lisäsin tämän jälkeen valintoihin. Käyttöönottaessa oli samanlaista piirteiden päällekkäisyyksiä kuin Marine-alustassa, mutta ne on nykyiseen versioon korjattu.

5.4 Virtuaaliset komponentit

Aina ei ole tarvetta kokoonpanoihin laittaa kaikkia osia tai niiden laittaminen ei välttämättä tuo lopputuloksen kannalta mitään lisäarvoa. Tätä varten Inventorissa on ominaisuus luoda Virtual Component -kokoonpanotiedostossa. Normaalisti tämä tehdään kokoonpanossa "Create"-toiminnosta, kuten kuvassa 22 ja sitten kaikki sen iProperties-tiedot syötetään manuaalisesti luonnin jälkeen. Tapa on hie-
man hidas useamman komponentin kanssa, mutta Inventorin API tarjoaa myös ta-
van lisätä komponentin ohjelmallisesti. iLogic pystyy myös luodun komponentin
iProperties-tiedot lisäämään koodileikkeiden avulla ja BOM-luettelossa kom-
ponentin lukumäärän yliajamaan.



Kuva 22. Normaali tapa luoda virtuaalinen komponentti.

Tämä oli suunnittelijoiden toive ja sen toteuttaminen oli oikeastaan erittäin nopea toimenpide. Virtuaalikomponenttien lisäämistä varten sain VEBA-suunnittelijalta tiedot komponenttien nimistä ja kuvauksista. Samalta suunnittelijalta sain myös kehitysehdotuksia ja kaikki nämä ehdotukset myös toteutettiin. Kehitysehdotuk-
sena oli mm. virtuaalikomponenttien kuvaukset myös toisella kielellä eri projek-
teja varten ja toinen ehdotus oli komponenttien sijaitsevan omassa kokoonpa-
notiedostossa. Tämä mahdollistaa piirustuksiin kyvyn tuoda niistä oman osaluet-
telon.

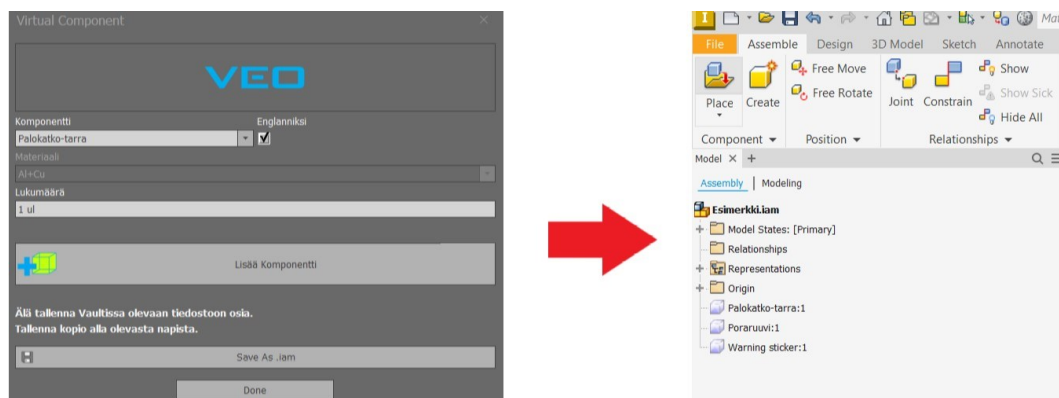
```

If Komponentti = "Kieltotarra" And Englanniksi = False Then
Component = "Warning sticker"
Kuvaus = "Käveleminen kielletty"
Else If Komponentti = "Kieltotarra" And Englanniksi = True Then
Component = "Warning sticker"
Kuvaus = "Walking forbidden"

```

Kuva 23. Esimerkki koodin ehtolauseista.

Virtuaalisia komponentteja varten loin oman kyselyikkunan missä valitaan haluttu komponentti ja sen määrä. Koodi vaihtaa määritetyt komponentit kuvauksineen tietyissä parametreissa ja nämä parametrit on linkitetty iProperties-tietoja hallinoiviin koodileikkeisiin. Kun käyttäjä laittaa ruksin kuvassa 24 näkyvään komponenttilistan viereiseen parametriin, saa suunnittelija englanninkieliset-versiot komponenttien iProperties-tiedoista.



Kuva 24. Komponenttien kyselyikkuna ja lopputulos selaimessa.

Virtuaaliset komponentit tehdyn kehitystyön ansiosta onnistuu nyt lisäämään nopeasti osaluetteloon ja niiden iProperties-tiedot ei ole suunnittelijan muistin varassa. Lisäetuna on myös lisäys johdonmukaisesti osaluetteloon ilman mahdollisia käyttäjän tekemiä kirjoitusvirheitä. Koska myös lukumäärän pystyy koodilla yliajamaan BOM-tiedoissa, ei ole tarvetta laittaa useampaa samanlaista komponenttia kokoonpanon selaimeen. Virtuaalisista komponenteista on tehty kaksi eri versiota suunnittelijoiden käyttöön. Suunnittelijat saavat VEBA-tarvikkeille oman kokoonpanotiedoston ja myös Kaapelinreitti-komponenteille kuten PPU40-putki.

6 KEHITYSTYÖN VAIKUTUS

6.1 Käyttäjien palaute

Käyttäjien palautteita varten opinnäytetyötä varten tein Microsoft Forms avulla kyselyn ja kyselyyn vastasi 8 henkilöä mekaniikkasuunnittelusta anonymisti. Kyselyssä oli 9 kpl kysymyksiä ja myös vapaan kommentoinnin mahdollisuus. Kyselyn tulokset olivat erittäin positiivisia ja toi esille arvokasta tietoa muiden suunnittelijoiden halusta oppia iLogicia sekä myös muita mielenkiintoisia kehitysehdotuksia. Kyselyn kysymykset osittain painottuivat käyttäjäkokemuksiin ja mitä kehitystyöstä yleisesti ollaan mieltä.

Haluaisitko nähdä lisää iLogic-malleja ja suunnitteluautomaatiota yleisesti?

[Lisätietoja](#)

● Kyllä	8
● En osaa sanoa	0
● Ei	0



Kuva 25. Kysymys #6.

Ylhäällä kuvassa 25 on ensimmäisen läpikäytävän kysymyksen tulokset ja ne ovat erittäin selkeät kaikkien vastanneiden kesken. Tämä tulos tuo esille laajaa halua kehittää asioita. iLogicia ja suunnitteluautomaatiota halutaan nähdä lisää jatkossa yksimielisesti kuten lähtökohtaisesti mielestäni pitääkin. Kehityksen pitää olla jatkuva ja sitä ei saa pysäyttää. Koska suunnitteluautomaatiota on tehty lähtökohtaisesti erittäin vähän isossa mittakaavassa, tämä näyttää tien mihin pitää edetä kehityksessä. Myös itse tehdystä kehitystyöstä ovat suunnittelijat tyytyväisiä.

Mitä mieltä olet tehdystä kehitystyöstä?

[Lisätietoja](#)

● Mahtavaa	7
● Ihan okei	1
● En osaa sanoa	0
● Ei kiitos	0
● Mikä kehitystyö?	0



Kuva 26. Kysymys #8.

yllä olevan kuvan 26 kysymyksen tulos antaa myös lähes yksimielisen mielipiteen, että tehty kehitystyö on ollut oikeanlaista. Asioita varmasti helpottaa, jos tehty kehitystyö on oikeanlaista myös suunnittelijalle ja heitä on konsultoitu kehitystyön aikana. Kaikkia ei varmasti voi aina täydellisesti miellyttää kehitystyötä tehdessä, mutta sen kokonaishyöty voi näkyä vasta myöhemmässä vaiheessa käyttäjälle ja lisäksi jos kehitystä tehdään enemmän suunnitteluautomaation lisäämiseksi.

Ovatko iLogic-mallit käyttäjäystävällisiä?

[Lisätietoja](#)

● Kyllä	5
● En osaa sanoa	3
● Ei	0



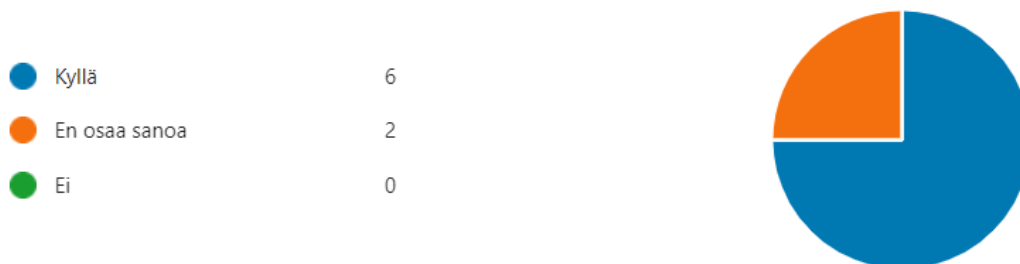
Kuva 27. Kysymys #3.

Käyttäjäystävällisyys on hieman epävarma kuvan 27 tuloksissa, mutta pääsääntöisesti suunnittelijat näyttäisivät olevan sitä mieltä, että iLogic-mallit ovat käyttäjäystävällisiä. Käyttäjäystävällisyyden lisäämiseksi voi aina tehdä lisää kehitystä ja käyttäjien ehdotukset pitää ottaa vastaan. Käyttäjäystävällisyys on isossa osassa kehitystyön läpiajossa, ja käyttäjät ottavat vastaan kehitystä helpommin, jos

kynnys uuden opetteluun ei ole iso. Helppokäyttöisyys pitää olla kunnossa jatkosakin ja se on kaikkien etu.

Onko iLogic-ohjausmalleissa tarpeeksi valintoja muokkausta varten?

[Lisätietoja](#)



Kuva 28. Kysymys #5.

iLogic-mallien kyselyikkunat voi kasvaa erittäin isoiksi ja valintojen määrä myös vaikuttaa käyttäjäystävällisyyteen. Kyselyikkunoissa valintoja voi olla myös liikaa ja se voi hidastaa niiden käyttäjää. Tärkeintä on löytää oikea tasapaino ja varmistaa, että päävalinnat löytyvät kyselyikkunoista helposti ja valintoja lisätään sitä mukaa kun niitä pyydetään tai toivotaan. Kuvan 28 tuloksien mukaan ei tämän asian suhteen ole oikeastaan nähdäkseeni toistaiseksi suurempia ongelmia.

Haluaisitko oppia iLogicia?

[Lisätietoja](#)



Kuva 29. Kysymys #7.

Erittäin positiivista on, että suunnittelijat haluavat oppia lisää ja tähän varmasti on vaikuttanut iLogicin hyötyjen näyttäminen kehitystyön aikana konkreettisesti. iLogic on erittäin hyödyllinen työkalu kaikkien suunnittelijoiden käsissä, jos sitä ehtii opetella. iLogicin oppimisessa voi kuitenkin mennä hieman aikaa. Kaikki

tietenkään ei kuvan 29 kysymyksen mukaan halua oppia, mutta ei se ole oikeastaan pakollistakaan.

6.2 Tehostus

Kyselyssä oli myös kysymyksiä liittyen suunnittelijoiden mielipiteisiin, miten ovat iLogic-kehitystuotokset tehostaneet työntekoa ja arvioita niiden avulla tehdystä ajansäästöstä. Tehostukseen liittyvät kyselytulokset ovat teoreettisia ja niitä ei ole mitenkään mitattu tätä kyselyä tarkemmin. Tuloksista ilmenee hieman jakaumaa ja erilaisia mielipiteitä.

iLogic-pohjaiset älykkäät ohjausmallit ovat tehostaneet työntekoasi?

[Lisätietoja](#)

● Vahvasti samaa mieltä	6
● Osittain samaa mieltä	0
● En osaa sanoa	1
● Osittain eri mieltä	1
● Vahvasti eri mieltä	0



Kuva 30. Kysymys #1

VEO:lla ei aiemmin ole ollut ohjausmalleja käytössä ja muutos on iso alusta alkaen mallintamiseen nähden. Kuvan 30 tuloksessa kaikki ei ole sitä mieltä, että ohjausmallit olisi tehostaneet työntekoa. Tähän tietenkin voi vaikuttaa mikä tahansa syy ja kehitystyötä ei ole kaikissa VEO:n tuoteperheissä tehty tasapuolisesti. Kehitystyö on käytännössä suurimmaksi osaksi koskenut VEBA-kiskosiltoja. Prosentuaalisesti isompi osa vastaajista on sitä mieltä, että ohjausmallit ovat heidän työntekoansa tehostaneet ja se on oikeanlainen suuntaa antava merkki.

Kuinka ohjausmallit ovat projektien läpimenoaikaan vaikuttanut?

[Lisätietoja](#)

● Merkittävästi nopeutunut	5
● Nopeutunut	2
● Ei ole	1
● Hidastunut	0
● Merkittävästi hidastunut	0



Kuva 31. Kysymys #2.

Läpimenoaika on varmasti nopeutunut osittain ja kuvan 31 tulos antaa saman mielipiteen yhtä vastaajaa lukuun ottamatta. Koska ohjausmallit ovat työntekoa nopeuttaneet jää suunnittelijalle aikaa tehdä muita tehtäviä liittyen tehtävään projektiin. Uskon ohjausmallien auttavan vähentämään ainakin projektien myöhästymisiä deadlinejen yli.

Koetko iLogic-suunnitteluautomaation vähentäneen toistuvia ja puuduttavia tehtäviä?

[Lisätietoja](#)

● Paljon	6
● Hieman	2
● En osaa sanoa	0
● En usko	0
● Ei ole	0



Kuva 32. Kysymys #4.

Kuvan 32 kysymyksen mukaan suunnittelijat ovat sitä mieltä, että toistuvat sekä puuduttavat tehtävät ovat vähentyneet suunnitteluautomaation myötä. Kaikki mahdollinen pitää automatisoida mitä pystyy, koska näkemäni mukaan aikaa käytetään edelleen asioihin mitä pystytään helposti automatisoimaan. Säästöt ovat isot, kun kaikki turhat toistuvat työt voidaan eliminoida. Osa näistä tehtävistä on

myös puuduttavia ja niiden poistuttua työtyytyväisyys myös varmasti kasvaisi. Suunta on tuloksen mukaan oikea.

Arvio omasta säästyneestä ajasta kuukausitasolla nykyisten kehitystyön tuotoksilla?

[Lisätietoja](#)



Kuva 33. Kysymys #9.

Kehitystyön tuotokset ovat onnistuneet tuloksien mukaan tunti määrällisesti paljon säästämään aikaa suunnittelijoilta. Kuvan 33 tulokset jakautuvat kolmeen osaan, mutta näistä voi päätellä kuukausitasolla olevan noin 20–30 h tasolla suunnittelijoiden mielestä. Ajan säästö voi olla suunnittelijan kohtaisesti hyvinkin erilainen riippuen kuinka paljon kyseinen suunnittelija käyttää esimerkiksi ohjaukselle tai muita iLogiciin liittyvää. Jos näiden teoreettisten tuloksien perusteella laskee vuosiksi säästöt eteenpäin, luku ei ole pieni.

1	anonymous	noin 2h / projekti
2	anonymous	Karkeasti 40 min/ elementti

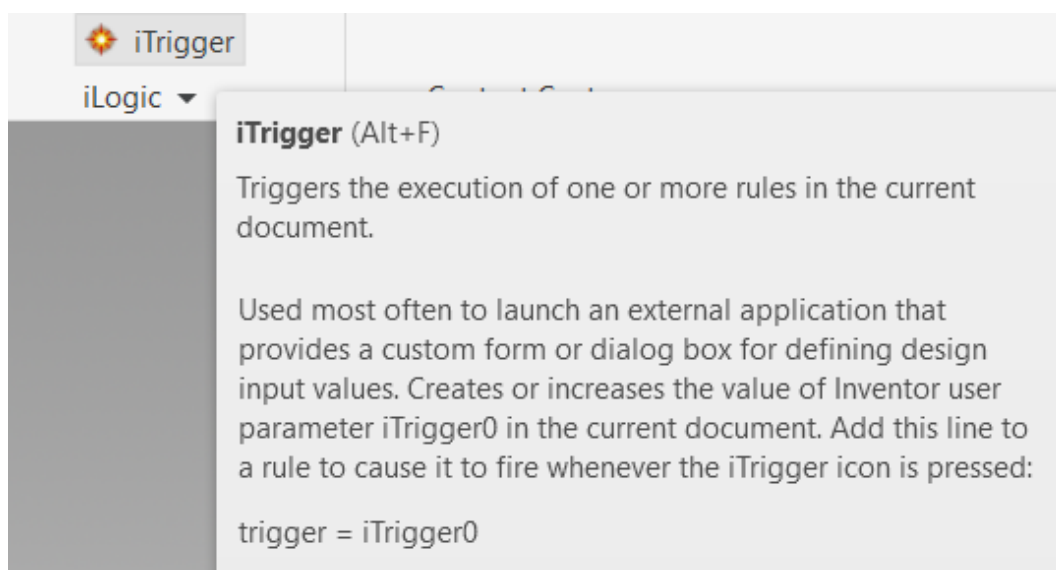
Kuva 34. Vapaita kommentteja elementtien pituuksien laskennasta.

Kuvan 34 esitetyissä vapaissa kommentteissa halusin myös tietää kuinka paljon aikaa suunnittelijat ovat käyttäneet esimerkiksi kiskoelementtien Excel-laskentatiedostoon ja sain 2 vastausta näiden käyttöön liittyen. Kiskosiltoja voi olla useampi projekteissa ja silloissa voi olla niiden lisäksi useampi elementti. Näitä elementtejä varten on aikaa Excelin kanssa kulunut käyttäjien palautteiden perusteella paljon. Suunnittelija voi olla pahimmillaan useamman tunnin mittojen muuttuessa

laskemassa mittoja uusiksi. Katkaisu- ja taivutusmitat piirustuksissa ovat jatkossa toivottavasti poistaneet tämän Excel-käytännön.

7 KÄYTTÄJÄYSTÄVÄLLISYYS

Kehitystyön kohteissa on pyritty ottamaan käyttäjäystävällisyys huomioon ajatuksella, että olisivat mahdollisimman helposti käyttöönotettavissa. iLogicin kyselyikkunoiden avaus alun perin tapahtui Event Triggers -toiminnon avulla ja tutkin tämän lisäksi toista tapaa avata ikkunat helpommin. Tutkimuksissa selvisi, että Inventor tarjoaa iLogicia varten kuvan 35 toiminnon nimeltä iTrigger. iTrigger käytännössä tekee parametrivalikkoon yksiköttömän parametrin, minkä voi linkittää koodiin komennolla: `trigger = iTrigger0`. iTriggerin voi myös laittaa Inventorin asetuksista valittuun pikanäppäimeen, mikä on VEO:n käyttäjille asetettu tyhjänä olevaan ALT+F-yhdistelmään. ALT+F painamalla parametriin tulee numero aina lisää ja se aktivoi avaamista varten tehdyn koodin.



Kuva 35. iTrigger.

Kolmas tapa avata kyselyikkunat, on navigoida mallipuun iLogic-välilehdelle. Tämä tapa on hieman vaarallinen, koska käyttäjä voi vahingossa poistaa tehdyn kyselyikkunan tai jopa tehdyn koodin. Tämän vuoksi sitä ei voi suositella muille kuin ohjelmoijalle. Event Triggers oli lopuksi myös hieman ongelmallinen, koska se availi kyselyikkunoita satunnaisesti useamman kerralla kokoonpanoissa ja tämän vuoksi

siitä luovuttiin. iTriggeriin on käyttäjät tottunut jo ja sen on todettu olevan helpoin tapa myös kokoonpanoissa osista käydä avaamassa kyselyikkunat.

Kyselyikkunoihin on myös kiskoelementtien tapauksissa lisätty kuvia helpottamaan suunnittelijaa parametrien muokkauksessa ja valinnoissa. Kiskoelementtien muokkaus voi olla hieman haasteellista niiden variaatioiden lukumäärän vuoksi niin malleissa kuin niiden kyselyikkunoissa. Parametrien muuttaminen pelkästään nimen avulla voi olla varsinkin uuden suunnittelijan näkökulmasta hieman epäselvä tilanne. PEN-puolen vaihto myös kiskoelementtien kyselyikkunoissa vaihtaa nykyisen puolen kuvan siihen sopivaksi ja tuplataivutetun kiskoelementin tapauksessa kätisyyttä hallinnoivan parametrin vieressä oleva kuva tekee tämän myös. Muihin kyselyikkunoihin ei ole lisätty kuvia ajan- ja tilanpuutteen vuoksi.



Kuva 36. Jäähdytysputken ohjeistus.

Kyselyikkunoiden käyttöä varten on tehty myös useampaa ohjeistusta käyttöönottoa helpottamaan kuten kuvassa 36 näkyvän jäähdytysputken ohjausmallille tehty ohje. Mallien tekijälle käyttö voi olla helppoa ja suoraviivaista, kun hän tietää miten ne toimivat, mutta ohjeistus on tarpeellista muille suunnittelijoille oikeanlaista käyttöä varten. Tietyn kyselyikkunan tunnistamista varten on niiden yläreunaan lisätty kytketyn mallin nimi, jos on suunnittelijalla useampi samanlainen malli

kokoonpanossa käytössä. Tämä ehkäisee väärän mallin parametrien muuttamista virheellisesti.

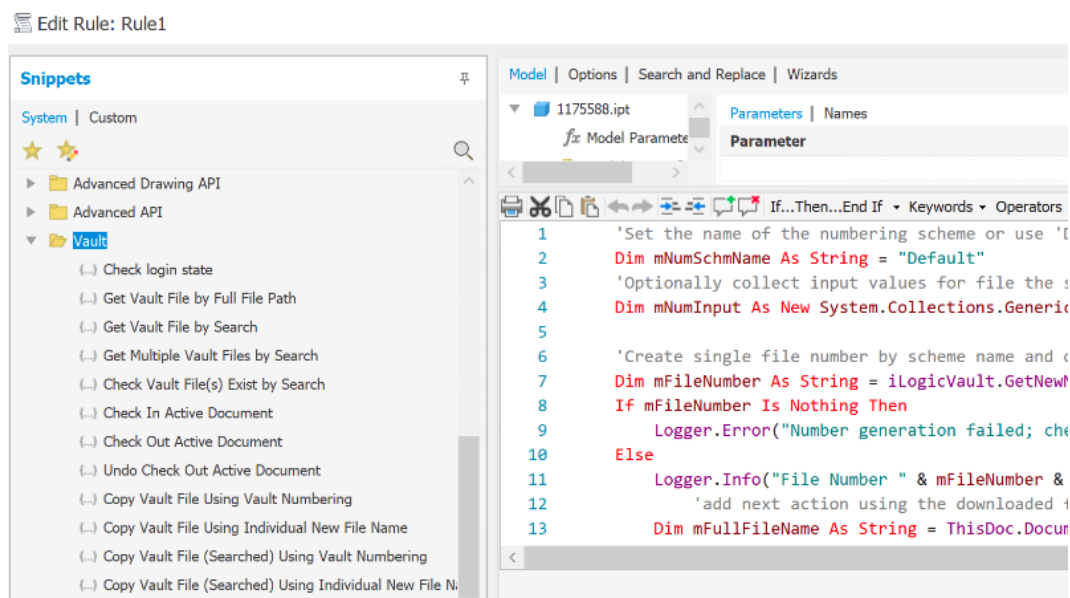
Ohjeistuksia on tehty useampia ja ne sisältävät ohjeistetun prosessin, miten malli haetaan ja tallennetaan uudelle nimelle omaa käyttöä varten Vaultista. Ohjeistuksissa on lyhyesti mainittu mitä mahdollisesti iLogic-mallilla pystyy tekemään ja kuinka sitä käytetään, mitä pitää laittaa käyttöönottaessa päälle ja mitä ei saa tehdä. Esimerkiksi jäähdytysputken tapauksessa on mainittu, että tietty Level of Detail pitää laittaa päälle ennen käyttöä ”suppress”-toiminnon vuoksi. Isompaa ohjeistusta on tehty myös, miten iLogicin kyselyikkunoita tehdään. Tässä ohjeistuksessa on käyty läpi mitä toiminnallisuuksia iLogicin Forms -editorissa on ja miten esimerkiksi iLogicilla saa luotua pikanäppäimen koodiin kyselyikkunoita varten.

Pelkällä tekstillä on erittäin vaikeata hahmottaa mitä käyttäjän pitäisi tehdä, joten ohjeistuksista on pyritty tekemään kuvalliset nuolilla varustettuina, että ohjeistukset eivät ole lukijalle epäselkeitä ja ovat mukavia lukea. Ohjeistuksia on pyritty päivittämään aina kun sille on ollut tarve ja myös pyydettyäessä. Ohjeistukset on päätetty tehdä myös kaikista tulevista kehitystyön projekteista iLogic-malleille, koska kaikki suunnittelijat eivät malleja käytä päivittäin tai käyttävät todella harvoin. Tällaisissa tapauksissa ohjeet toimivat myös muistin virkistämisenä, että ei käyttäjän tarvitse lähteä kyselemään muilta työntekijöiltä apua.

8 JATKOKEHITYS

8.1 iLogic & Vault

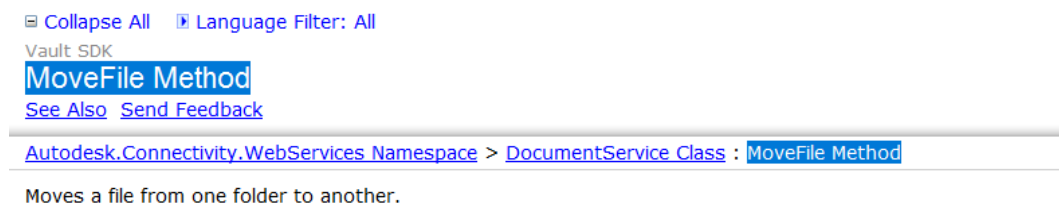
Jatkokehitystä ajatellen tulevaisuudessa iLogic saa 2024-version mukana uusia koodileikkeitä, mitkä mahdollistavat iLogicin toiminnan Vaultin kanssa. Kuvassa 37 näkyvät koodileikkeet mahdollistavat tiedostojen haun suoraan Inventoriin koodin avulla ja tehdä niistä automaattisesti kopion. Koodileikkeet mahdollistavat myös muita toimintoja kuten hakua tietyin kriteerein suoraan parametriin listaksi, jolloin käyttäjä voi tuoda tietyn mallin konfiguroitavaan kokoonpanoon. Koodileikkeet mahdollistavat myös automaattiset Check In- ja Check Out -toimintojen käytön.



Kuva 37. Inventor Professional 2024 Ilogic Vault -koodileikkeet.

Koodileikkeistä lupaavimmat liittyen tiedostojen kopiointiin tietääkseni ei siirrä kopioitua tiedostoa oikeaan kansioon vaan kopioi sen alkuperäisen osan sijaintiin. Tämän kiertämiseksi on mahdollisesti Vault SDK -ohjeistuksessa ratkaisu ja se on kuvan 38 MoveFile Method. MoveFile Method vaikuttaa lupaavalta, koska se on lyhyt koodileike ja siihen ei tarvitse kuin määrittää nykyinen tiedosto, mistä se siirretään ja mihin.

Teoriassa tämä metodi olisi lupaava ratkaisu sillä ehdolla, että nämä Vault SDK -ohjeistuksen koodit toimivat iLogicissa. Tämä vaatii lähempänä ajankohtana lisätutkimuksia ja kokeilua VEO:lla, koska opiskelijat eivät saa Vaultia käyttöönsä Autodeskillä. Vault SDK -ohjeistus ei ole saatavilla verkossa vaan sen saa käyttöönsä vain Vaultin tiedostoista asentamalla.



Kuva 38. MoveFile Method. [14]

8.2 Kiskoelementtien jatkokehitys

Talven aikana olen ideoinut Nevalan kanssa, miten kiskoelementtejä voisi kehittää iLogicin avulla isommaksi kokonaisuudeksi. Ideana olisi, että kiskoelementtejä ei kasattaisi tulevaisuudessa käsin Inventorissa kiskosilloiksi kaikista osista turhaa toistuvaa manuaalista työtä käyttäen. Tätä varten iLogicin avulla kehitettäisiin ”Kiskosilta”-konfiguraattori, millä iLogicin avulla uusien Vault -koodileikkeiden myötä tuotaisiin viimeisin kiskoelementin tiedosto, kaikki siihen tarvittavat osat iLogicin Add Component -koodileikkeellä ja ne rajoitettaisiin myös koodin avulla oikeisiin paikkoihin.

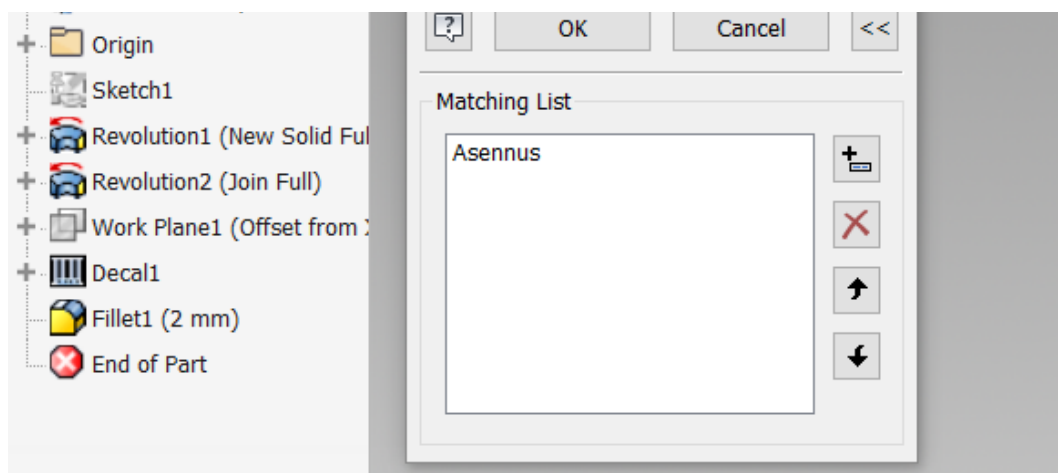
Toiminnan sekä jälkikäteen käyttäjän säädettävyyden varmistamiseksi, olisivat osia varten kaikki koodi ja parametrit tyhjässä kokoonpanotiedostossa valmiina. Toimintaa säättävät parametrit olisivat valmiiksi luodussa käyttöliittymässä deaktivoituna ja muuttuisivat aktiiviseksi riippuen tuodusta elementistä koodissa määritetysti. Valmiissa käyttöliittymässä olisi mahdollisesti myös napit jokaiselle kiskoelementille mikä avaisi sen tietyn kiskoelementin valmiin käyttöliittymän. Tämän avulla ei tarvitse tehdä uuteen pääkäyttöliittymään lisää valintoja ja voi käyttää jo olemassa olevia kyselyikkunoita.

Toiminta olisi helpompaa, jos uusia tiedostoja ei tarvitse tehdä jokaisesta osasta, vaan käytettäisiin valmiita kokoonpanoja mihin on tarvittavat variaatiot luotu valmiina käyttöön ja käyttöliittymän säädön varassa. Kaikkiin tarvittaviin osiin olisi suotavaa myös tehdä valmiit rajoitteet ohjelmointia helpottamaan. Käyn läpi seuraavissa osioissa, millä toiminnoilla nämä saavutettaisiin.

8.2.1 iMate

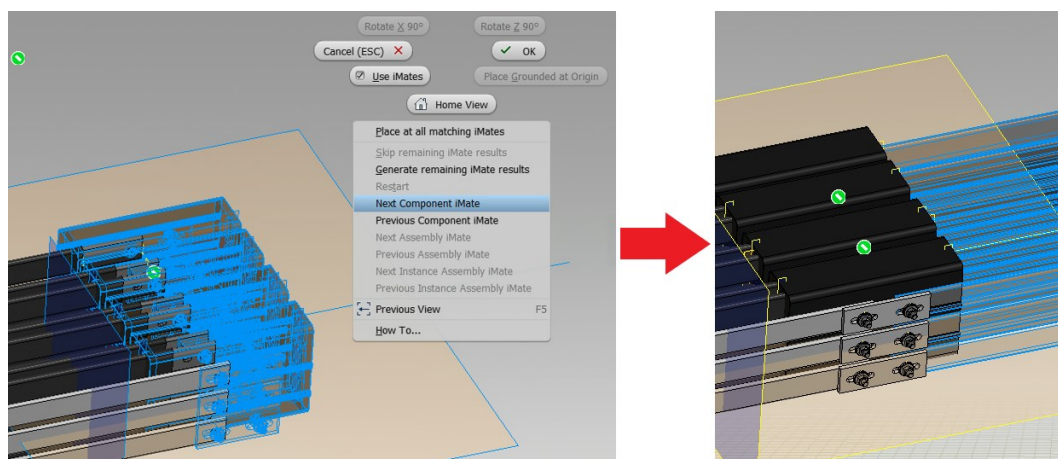
Inventorissa on pitkän ajan ollut toiminto mitä tietääkseni ei VEO:lla ole käytössä ja se on iMate. iMaten avulla malleihin voi luoda valmiiksi määritettyjä rajoitteita esimerkiksi asennusliitosta varten kokoonpanoissa. Näitä rajoitteita voi yhdistää yhdeksi yksittäiseksi ”Composite”-rajoitteeksi, mikä helpottaa myös liittämään malleja toisiinsa ja sitä voi ajatella iMate-kokoonpanona. Composite-rajoite sisältää kaikki liitosta varten tarvittavat iMatet ja siihen tarvitaan vastaava liitos, missä on samat iMate rajoitteet compositeksi yhdistettynä.

iMateihin voi myös määrittää kuvan 39 ”Matching list”-toiminnon kautta yhteensopivia iMateja. Tästä edellä mainitusta ominaisuudessa voi olla hyötyä eri mallien asennustavoissa. Toiminto mahdollistaa ainoastaan asennusjärjestyksen eikä sillä pysty estämään liitosta toiseen samankaltaiseen iMateen. iMaten luonti on erittäin helppoa ja sen luonti on käytännössä samanlainen prosessi kuin laittaisi normaalia rajoitetta paikalleen tiettyyn osaan.



Kuva 39. iMate Properties Matching list.

Mallia tuodessa kokoonpanoon aktivoimalla valikosta nappi ”Interactively place with iMates”, Inventor ehdottaa heti oikeata asennuspaikkaa ja vaivatta liittää oikealle paikalleen. iMaten voi valita myös seuraavaan sopivaan paikkaan oikean hiiren painikkeesta valitsemalla ja painamalla ”Next Component iMate” kuvan 40 mukaisesti. Samalla tavalla voi myös valita, että liitä kaikkiin sopiviin iMateihin.

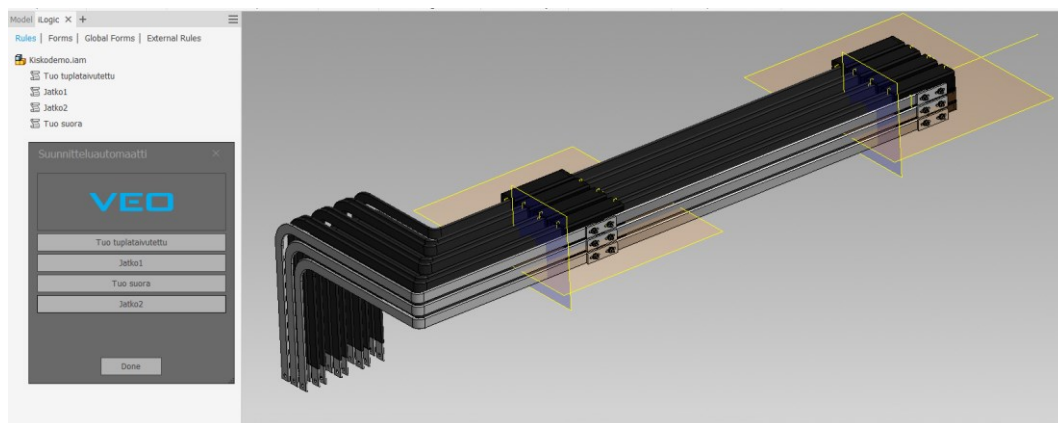


Kuva 40. Elementtien ja jatkojen lisääminen iMatella interaktiivisesti.

iMaten-toiminnon hyödyntämisellä on käsittämätön mahdollisuus suunnittelijoiden ajankäytön tehostamiselle ja voi eliminoida toistuvaa työtä suuresti. iLogic voi myös hyödyntää iMateja suoraan koodileikkeillä. Koodileikkeisiin tarvitsee vain määrittää minkä tietyn mallin iMate asettuu toiseen tietyn malliin määritettyyn

iMateen. Koodileikettä on erittäin helppo hyödyntää juuri tämän takia ja säästää paljon aikaa ohjelmoinnissa, jos malleissa on valmiiksi iMatet lisättynä.

Nopean testin mukaan erittäin pienen kuvassa 41 näkyvän kiskosillan pohjan tekeminen kahdella elementillä ja kahdella jatkolla meni noin minuutti interaktiivisen iMate-toiminnolla ja iLogicin sekä iMatejen yhdistävän koodin avulla pääsi alle 4 sekunnin lukemiin. iLogic ja iMaten yhdistelevällä menetelmällä voi aikaan tieteenkin vaikuttaa jo tietokoneen yleinen suorituskky mallien rajoittamiseen sekä tuomiseen kokoonpanoon. iMatet palvelisi erinomaisesti ”Kiskosilta”-konfiguraattoria helppokäyttöisyydellään ja niistä on myös hyötyä muuhun käyttöön kuten päivittäisissä kokoonpanojen rakentamisissa.



Kuva 41. iLogic & iMate yhteistyössä.

Käytännössä ei ole mitään järkevää syytä miksi näitä ei jatkossa käytettäisi. VEO:lla on monia tuhansia malleja missä ei ole iMateja ja niitä jatkuvasti kasataan kokoonpanoihin vanhoilla rajoittamistavoilla käyttöön asiakasprojekteissa. iMatet ei tietenkään ole uusi ominaisuus vaan se on ollut Inventorissa erittäin kauan, mutta sen käyttämättömyyteen varmasti liittyy tieto sen olemassaolosta ja mahdollisuuksista. iMatejen etuna on myös se, että Inventorin content centerissä eli paikasta mistä tuodaan vakiokomponentit kuten pultit, on iMatet valmiiksi jo luotu kyseisiin komponentteihin.

8.2.2 Automaattinen uudelleen nimeäminen ja kopiointi

Testieni mukaan iLogicilla onnistuu automaattinen nimeäminen osaa kokoonpanoon tuodessa. Osan nimeäminen käytännössä koodin avulla toimii siten, että osa avataan taustalla kun se on tuotu kokoonpanoon ja iLogic tallentaa sen heti uudelle nimelle ja uudeksi yksittäiseksi osaksi iLogic koodileikkeellä: `ThisDoc.Document.SaveAs(NewFileNameAndExtensions, False)`. Kun osa avataan taustalla käyttäjälle näkymättömästi, säästetään hieman aikaa inventorin tekemässä prosessoinnissa.

Aiemmin mainittu `SaveAs`-koodi myös korvaa testieni mukaan nykyisen osamallin avonaisessa kokoonpanossa, kun koodin viimeinen Boolean-tieto on epätosi eli `False`. Kiskoelementtien automaattinen nimeäminen olisi helppoa, koska niiden nimeämistyyli projekteille seuraa tiettyä kaavaa: `projektinro.tunnus_elementti_nro`. Automaattisella nimeämisellä pystyy säästämään monta minuuttia, kun ei kiskoelementtiä tarvitse erikseen manuaalisesti tallentaa projektin kansioon ja etsiä Vaultista.

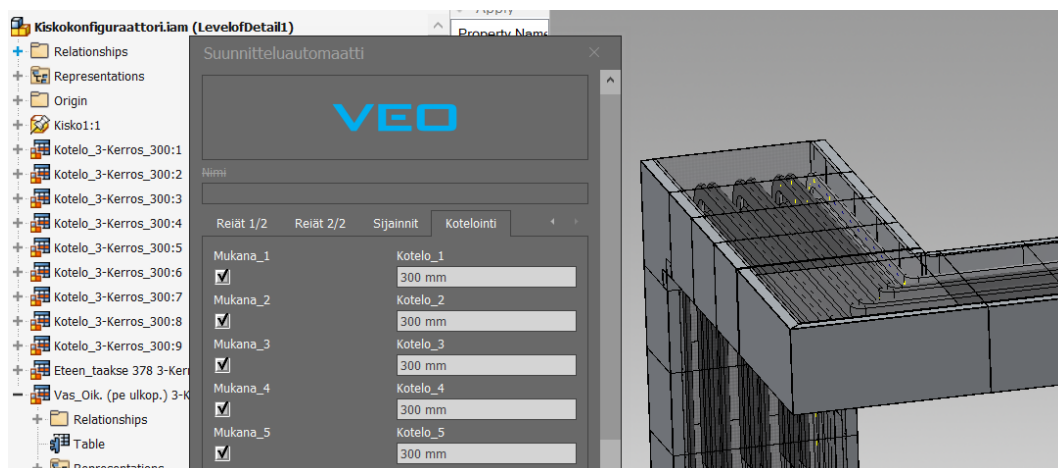
Yhdistämällä aiemmin esitetyssä kuvassa 41 näkyvä iMate kokoonpano iLogicilla ja tämän automaattisen nimeämisen, aikaa kului kuitenkin vain 8 sekuntia kokoonpanon rakentamiseen sekä molempien elementtien nimeämiseen. Tähän tallennuskoodiin voi lisätä sulkeiden sisään ennen nimeä `"ThisDoc.Path"`-lauseen, jolloin osa tallentuu nykyisen kokoonpanon sijaintiin. Tallennuskoodia voi hyödyntää myös muihin kiskosilloille tarvittaviin osiin helposti.

8.2.3 iPart & iAssembly

iPart ja iAssembly ovat myös tehokkuudeltaan omaa luokkaansa, koska valmiiseen taulukkoon voi luoda paljon eri variaatioita malleista. iPart & iAssemblyissa on etuna se, että yksittäisiä malleja ei tarvitse kokoonpanossa nimetä uusiksi käyttöä varten. Jokainen kokoonpanoon tuotu iPart tai iAssembly voi olla taulukon erillisellä tai samalla rivillä ilman, että niitä olisi nimetty uusiksi itsenäisiksi tiedostoiksi.

iPartit ja iAssemblyt vähentävät projektikohtaisten mallien luontia, koska luodut variaatiot ovat edellä mainittujen ominaisuuksien vuoksi aina uudelleen käytettävissä. Taulukot ovat Excel-pohjaisia ja variaatioiden määrää voi laajentaa erittäin helppokäyttöisesti kopioimalla rivejä Excel-toiminnoilla. iAssemblyista ei tosin kannata luoda liian monimutkaisia, jolloin uusien rivien luonti vaikeutuu ja hidastuu.

iLogic ja ”Kiskosilta” -konfiguraattoria silmällä pitäen ovat myös iAssembly ja iPart tärkeässä asemassa. iLogic pystyy valmiiden koodileikkeiden avulla vaihtamaan näiden rivejä ja tätä varten voi linkittää koodin parametrit käyttäjän hallinointia varten suoraan käyttöliittymästä kuvan 42 mukaisesti. Rivejä pitää vain tehdä erittäin paljon, jos on esimerkiksi kiskoelementtien kotelo minkä mitta voi olla tilanteesta riippuva. iPart- ja iAssembly-malleihin voi myös laittaa iMatet, ohjelmoinnin sekä manuaalisen käytön varalle asennusta helpottamaan ja nopeuttamaan.



Kuva 42. iAssembly-rivin vaihto iLogicilla.

8.3 Kehityksen kohteiden parantelu

Kiskoelementtien automaattiseen piirustukseen on suunnittelijoiden toimesta toivottu dynaamisesti vaihtuvaa tekstiä PE- ja PEN-tiedoille. Inventor 2021 ei tarjoa mahdollisuutta linkittää normaalia tekstimuotoista parametria piirustukseen. Tämä ominaisuus löytyy ainakin 2024 vuoden Inventorista, mutta asiaa hieman

tutkittuani, Inventorissa on toinen keino millä sen rajoituksen pystyy kiertämään 2021-versiossa.

iLogic pystyy hallinnoimaan custom iProperties -kenttien tietoja ja näihin voisi kyseiset tiedot lisätä sekä muuttumaan dynaamisesti varioitavan kiskoelementin parametrialintojen mukaan. Kuvassa 43 esitetyt Custom iProperties -kenttien tekstimuotoisen tiedon voi kaikesta huolimatta piirustukseen lisätä. Tämä vaatisi vain Sketch-symbolien päivittämistä ja jokaiseen kiskoelementtiin custom iProperties -kenttien koodin lisäämistä.

Automaattisissa piirustuksissa käytettävät katkaisu- ja taivutuspituudet mahdollisesti pitää myös mahdollisesti jälkikäteen tarkistella uudestaan. Kun taivutuskonetta saadaan päivitettyä, mitat voi päivityksen jälkeen muuttua liikaa ja silloin tehdyt kaavat eivät enää ole päteviä. Mahdollisen päivityksen jälkeen on mahdollista, että mitat ovat aina taivutuksen jälkeen tasan ja ne pystyy lyödä kaavoihin lukkoon.

Name: Modify

Type: Delete

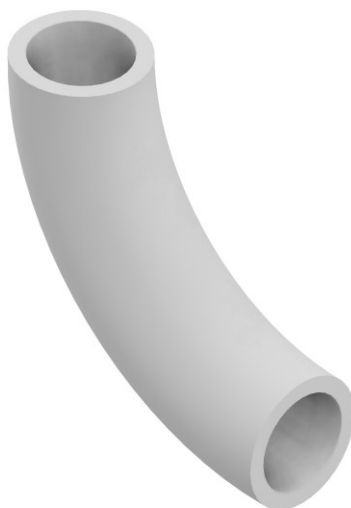
Value:

Name	Value	Type
Kisko1	PEN	Text
Kisko8	PE	Text

Kuva 43. Custom iProperties

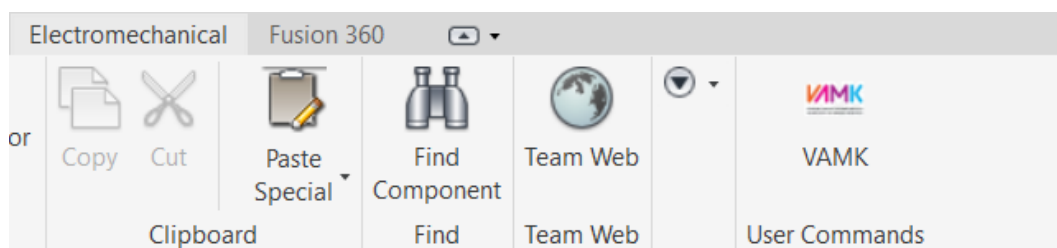
Jäähdytysputkeen voisi lisätä valinnan mahdollisuuden kumpaankin päähän putkea kuvan 44 mutkalle. Tämän avulla voisi liittää useamman putken eri asennoissa, mutta toistaiseksi tämänkaltaiselle ei ole toivetta suunnittelijoiden puolelta tullut. Tämän ominaisuuden lisääminen olisi todella helposti toteutettavissa ja palvelisi mahdollisia tulevaisuuden asiakastarpeita. Jäähdytysputken kyselyikkunaan voisi

myös lisätä osatiedoston nimen. Suunnittelija näkee helposti, että se on varmasti korvattu alkuperäisesti tiedostosta muokkausta varten.



Kuva 44. Mutka jäähdytysputkelle.

Inventor toi 2023 vuoden versiossa ominaisuuden lisätä ilogic-sääntöjä ja kyselyikkunoita suoraan ylärivin työkalupalkkiin kuten kuvassa 45. Ensimmäisenä näkisin helpon tavan lisätä tehdyt kyselyikkunat suoraan työkalupalkista eli Inventorin Ribbonista, jolloin ei tarvitse välttämättä suunnittelijoiden asetettua pikanäppäintä iTriggerille käyttää. Nykyisellään Inventorin vanhoissa versioissa vain lisäosien avulla. Nappeja voi käyttää myös muunlaisiin koodeihin, jos sellaisia pääte-tään kehittää suunnittelijoita varten.



Kuva 45. Esimerkki napin lisäämisestä.

9 ITSEARVIOINTI JA YHTEENVETO

9.1 Itsearviointi

Mielestäni opinnäytetyöstä tuli oikein hyvä ja laajempi kuin alun perin ajattelin. Sain sisällytettyä työhön laajasti ja monipuolisesti tehtyjä kehityksiä. Opinnäytetyön teko myös antoi minulle mahdollisuuden hieman yksityiskohtaisemmin tarkastella tehtyjä kehityksiä ja miten niitä voisi parannella. Opinnäytetyössä oli myös VEO:n puolella kiinnostusta jatkokehitysten kannalta ja uskoakseni sain onnistuneesti esiteltyä erittäin mielenkiintoisia sekä tarpeellisia ratkaisuja tulevaisuuden kannalta VEO Oy:llä.

Uuden 3D CAD -ohjelmiston opettelu on ollut nopeatahtista ja siihen on saanut Inventor-asiantuntija Rami Nevalalta todella hyvää koulutusta, mutta iLogicin opettelu piti tehdä itsenäisesti ja oma-aloitteisesti. iLogicin harjoittelu myös vapaa-ajalla on kasvattanut tietämystäni ja ohjelmointiosaamista. Ohjelmoinnissa it sessään olen huomannut, että minun pitäisi kommentoida koodia huomattavasti nykyistä enemmän. Minun pitäisi samalla myös siirtyä käyttämään ”Select Case”-menetelmää ”If Else”-lauseiden sijasta, koska se on kevyempi ja lyhyempi tapa kirjoittaa samanlaisia ehtoja koodiin.

Opinnäytetyötä tehdessäni näin vain enemmän mahdollisuuksia iLogicin hyödyntämiselle, mutta myös sitä tukevien mallinnusperiaatteiden ja tapojen käytössä. Koulussa suorittamani 3D-kurssit pitivät sisällään, miten ohjausmalleja luodaan sekä niiden periaatteita ja niistä tiedoista oli hyötyä myös tämän kehityksen ohjausmalleille. Koen iLogic-osaamisen tukevan varmasti tulevaisuuden uraani Inventorin käytön merkeissä ja suunnitteluautomaation parissa.

9.2 Yhteenveto

iLogicin hyöty tehokkuuden kasvattamisessa ja toistuvien töiden vähentämisessä on verraton 3D-suunnittelun parissa. Aiemmin tehty nollasta mallintaminen projekteille pitäisi olla historiaa nykypäivän teknologioiden avulla. Pienellä

suunnitteluautomaatiolla on saavutettu turhaa ja paljon aikaa vaativaa Excelin käytön eliminointia, variaatioiden teon helpottaminen nopeasti sekä käyttäjäystävällisesti. iLogic on mahdollisesti osittain onnistunut suunnittelussa projektien läpimenoaikaa pienentämään palautteiden perusteella. Mallien helppo säädettävyys suunnittelijalle on merkkiasemassa räätälöinnissä asiakkaiden tilojen sekä tarpeiden mukaan.

Mahdollisuuksia on myös luotu turhien komponenttien lisäämisen vähentämiseksi kokoonpanoihin ja tehty valmista virtuaalista kirjastoa vakiokomponenteille osaluetteloon laittamista varten ripeästi. Lisätutkimukset jatkokehitystä varten on myös tehty ja tuotu esille opinnäytetyössä. Jatkokehitystä varten on tunnistettu myös muita tehostamista edistäviä asioita kuten iMate. iMate tulisi ottaa käyttöön laajasti ja varsinkin standardinimikkeille VEO:lla. Ilman iMatejen läsnäoloa niistä ei mielestäni saisi tehdä standardinimikkeitä tulevaisuutta ajatellen.

Opinnäytetyössä tuotiin myös esille miksi mallinnustavoissa parametrien käyttö sekä nimeäminen on tärkeitä ja myös ne tulisi ottaa käyttöön oikeanlaisissa mallinnustavoissa mallien helppoa jälkikäteen muokkausta varten. Mallien tekeminen huonosti ei hyödytä ketään ja jälkikäteen korjailu on aikaa vievää. Kaikki hyvät mallinnustavat myös edistävät ja helpottavat mahdollista jälkeenkäytettävää suunnitteluautomaatiota.

LÄHTEET

1. Sippola H. Energiakeskittymän ytimessä – VEO:n tarina. Vaasa: VEO; 2017. s. 19.
2. VEO. Tietoa meistä. [Internet]. [Viitattu 29.12.2023]. Saatavilla: <https://veo.fi/fi/tietoa-meista/yritys/>
3. VEO. Liiketoiminta-alueet. [Internet]. [Viitattu 28.12.2023] Saatavilla: https://veo.fi/wp-content/uploads/2022/03/VEO_etu-sivu_hero_1920x1080px.jpg
4. Hietikko E. 8. uudistettu painos. Solidworks 2020. Helsinki: BoD - Books on Demand; 2020. s. 23-25.
5. Nevala R. Konsultti. [Haastattelu]. 29.12.2023.
6. Munford P, Normand P. Mastering Autodesk Inventor 2016 and Autodesk Inventor LT. Indianapolis: Sybex; 2016. s. 223, 410.
7. Tuhola E, Viitanen K. 1. painos. 3D-mallintaminen suunnittelun apuvälineenä. Tampere: Tammertekniikka; 2008. s. 33.
8. Martio A. 1. painos. Tuotekonfigurointi ja tuotetiedon hallinta. Espoo; Amartekno Oy; 2015. s. 20-36, 122, 206, 255, 263.
9. Autodesk. Inventorin tärkeimmät ominaisuudet. [Internet]. [Viitattu 28.12.2023]. Saatavilla: <https://www.autodesk.fi/products/inventor/features>
10. Autodesk. [herzinj]. [Internet]. 24.04.2018 [Viitattu 28.12.2023]. Saatavilla: <https://blogs.autodesk.com/inventor/illogic-autodesk-inventors-automation-engine/>
11. Autodesk. Key features of Vault. [Internet]. [Viitattu 27.12.2023]. Saatavilla: <https://www.autodesk.com/products/vault/features>

12. VEO Oy. Inventor/Projektien haasteita. [Palaveri]. 06.10.2023
13. Brown C. iLogic Normalize Browser Nodes, With Code. [Internet]. 29.6.2020 [Viitattu 07.01.2024]. Saatavilla: <https://clintbrown.co.uk/2020/07/29/illogic-normalise-browser-nodes-with-code/>
14. Vault SDK. Developer Documentation 2021. [Dokumentaatio]. [Viitattu 01.01.2024].

LIITTEET

LIITE 1. Henkilöhaastattelu: Konsultti Rami Nevala 12/2023

Voitko kertoa lyhyesti sinun urataustasi ja kerro samalla 3D CAD osaamisestasi?

Valmistuin vuonna 1996 Kone- ja Metallitekniikan Teknikoksi, silloin käytössä oli AutoCad 12. Alotin mekaniikkasuunnittelijana eri yrityksissä. Pari vuotta myöhemmin tulivat AutoCad 13 ja AutoCad 14 joilla aloitin 3D-Mallinnuksen. Versioon 14 tuli lisäpaketti Genius, joka mahdollisti komponenttikirjaston käytön ja muokkaamisen. AutoCad Mechanical Desktop ja SolidWorks 99+ muuttivat mallinnuksen parametriseksi, jonka jälkeen Inventor Versio 2 Beta julkaistiin Suomessa.

Siirryin Autodeskin maahantuojaan palvelukseen 2011 jonka aikana kehitin Inventoria yhteistyössä eri sovelluskehittäjien kanssa. Koulutin ja konsultoin. 2006 aloitin yrittäjänä ja käyttöön tuli Alibre Design. Monien vaiheiden jälkeen olen vastaanottanut eri insinööritoimistojen mekaniikkatiimien perustamisesta ja toimintojen kehittamisestä. Nykyinen työnantaja on VEO Oy, jossa kehitän erilaisia suunnitteluratkaisuja tarpeiden mukaan.

Mikä tekee Inventorista erinomaisen sen kilpailijoihin verrattuna?

Inventorin ohutlevyominaisuudet, teräs- ja putkisuunnittelutyökalut, työkuvat, ohjausmallien kautta hallittavat kokonaisuudet sekä ohjelman helppokäyttöinen räätälöinti eri yritysten tarpeisiin tekee Inventorista tehokkaan ja kevyen ohjelman. Projektitiedostojen kautta dokumenttien hallinta on tehokkaampaa verrattuna kilpailijoihin. Relatiiviset tiedostopolut ovat tärkeä tekijä tiedostojen käsitelystä.

Mitä voit kertoa parametrisesta mallintamisesta ja sen hyödyistä?

Osilla ja kokoonpanoilla on aina keskinäisiä parametreja sidoksissa toisiinsa. Osamallinnuksessa luonnoksia ja piirteitä voidaan ohjata helpommin parametrien ja funktioiden kautta. Nämä parametrit ohjaavat myös kokoonpanoja, joiden kautta muokattavuus helpottuu. Parametrien kautta voidaan luoda iLogic – ohjelmoinnin avulla tehokkaampia ratkaisuja eri suunnitteluvaiheisiin.

Mikä on Multi-Body malli ja mitä etuja se tarjoaa mallintamiseen?

Multi-Body on yksittäinen malli sisältäen itsenäisiä tilavuusmalleja, luonnoksia, työpiirteitä, mittaparametreja ym. Voimme luoda ohjausmalleja, joiden kautta hallitaan suuriakin kokonaisuuksia, sillä suunnittelu on jatkuvasti kehittyvä sekä muuttuva työvaihe. Jos jotain muuttuu, sillä on aina vaikutus kokonaisuuteen. Vertaan Multi-Body ympäristöä usein muovailuvahan käyttöön. Multi-Body ohjaa osien lisäksi myös kokoonpanoja vieden tietoa työkuviin asti.

Kerro Top-Down menetelmästä ja miten paljon sitä lähtökohtaisesti käytetään yritysmaailmassa?

Kokonaisuuden hallinta alkaa ympäristön luonnostelusta, josta jatketaan matkaa alaspäin kohti tarkempia yksityiskohtia. Ruutupaperi on oiva esimerkki, kun lähdetään luomaan uutta kokonaisuutta. Sen käyttö on jatkuvasti lisääntynyt, kun pyritään luomaan tuotteita tai tuoteperheitä. Ohjelman käytön tehostamisen kautta tämä ominaisuus kehittyy ja lyhentää prosessia suunnittelusta valmistukseen asti.

Kerro mikä on ohjausmalli ja miksi niitä tulisi hyödyntää?

Ohjausmallin kautta hallitaan kokonaisuutta, joka ohjaa kauttaaltaan koko suunnitteluprosessia. Aiemmin käytettiin nimitystä Skeleton sen luurankotyyppisen tiedon hallinnan vuoksi. Suunnittelussa harvoin tiedetään tarkkaa lopputulosta, joten ohjausmallien kautta eri piirteitä ja tietoja voidaan sitoa yhteen. Tärkeintä on kuitenkin kokonaisuuden hallinta jo suunnittelun alkuvaiheessa. Ohjausmallin avulla

säästetään aikaa ja minimoidaan virheet, kun dokumenttien väliset sidokset otetaan tehokkaaseen käyttöön.

Miten paljon konfiguraattoreita käytetään ja mitä etua ne tuovat suunnitteluun?

Konfiguraattoreiden käyttö on lisääntynyt viimeisen 15 vuoden aikana huomattavasti, varsinkin silloin kun iLogic on tullut tehokkaammin mukaan. Jatkuva kilpailu ja asiakaskohtaisten ratkaisujen tarjonnan lisääminen auttaa kehittämään toimintaa. Konfiguraattoreiden avulla vähennetään yksittäisten mallien ja versioiden määrää, kun kokonaisuutta ohjataan konfiguraattorin avulla. Tuotteet ja laitteet yhtenäistetään konfiguraattoreiden avulla, koko prosessi suunnittelusta lopputuotteeksi tehostuu tuoden kokonaisvaltaista hyötyä.

Miten kehitystyötä yrityksissä yleensä tehdään suunnittelun tehostamisessa?

Ensimmäisenä tulee mieleen sana: asenne. Rutiinit ja urautuminen ovat suurin haitta kehittymiselle. Yrityskohtaisissa toimintamalleissa on suuria eroja, niihin vaikuttavat eniten halu kehittyä mutta myös periaatteet. Yksittäisten henkilöiden toiminnalla on vahva vaikutus. Jatkuva koulutus ja konsultointi auttaa avaamaan silmiä kehityksen tarpeellisuuteen.

Olen kouluttanut ja kehittänyt satoja yrityksiä Suomessa, ja sen myötä on ollut havaittavissa lukuisia toimintamalleja. Mitä suurempi yritys, niissä suunnittelua kehitetään enemmän sillä työntekijöiden määrän lisääntyessä yksittäiset tarpeet ja ideat huomioidaan eri tavoin.

Minkälaiset ovat yleensä lähtökohdat ja mitä muuta pitää ottaa huomioon?

Lähtökohdan määrittämiseen tarvitaan halu, idea tai tarve. Usein voidaan aloittaa ihan tyhjästä, kun on selkeä päämäärä. Yleensä pitkäjänteinen työ vaatii

kehittämään toimintaa, kun esim. Tuotannolliset tai taloudelliset syyt antavat sen sykäyksen. Aina täytyy ottaa huomioon läpileikkaus toimintamallista.

Kuinka iso tarve yrityksillä on suunnitteluautomaatiolle?

Tarve on suurempi kuin mitä tällä hetkellä on käytössä, yritykset eivät välttämättä tiedosta sitä. Pitäisi enemmän nk. "viheltää peli poikki" ja katsoa kokonaisuutta. Usein ulkopuolinen konsultointi auttaa kehittämään toimintaa. Myös tiedon omaaminen on pullonkaula tarpeen määrittelyyn. On hankala kehittää toimintaa, jos ei tiedä mahdollisuuksista.

Mitkä ovat olleet merkittävimmät kehitysaskeleet tai trendit vuosien saatossa 3D CAD-suunnittelussa?

Parametrinen mallinnus, dokumenttien hallinta, alakohtaisten työkalujen lisääminen ja erityisesti lujuuslaskennan lisääminen osaksi ohjelmaa ovat lisänneet tehokkuutta. Varsinkin prosessisuunnittelun tehostaminen eri ohjelmissa on parantanut kokonaisvaltaista projektien läpivientä.

Miten muuten suunnittelua tulisi kehittää ja tehostaa mielestäsi VEO:lla jonkin tietyn ominaisuuden avulla Inventorissa tai yleisesti mallinnustavoissa?

Yhteisten pelisääntöjen luonti ja toimintatapojen yhtenäistäminen ovat pohja kokonaisuudelle. Kiire ei saa olla koskaan tekosyy omien toimintatapojen kehittämiseksi. Ohjausmallien ja varsinkin erilaisten konfiguraattoreiden käyttöä täytyisi huomattavasti lisätä. Tärkein asia on kuitenkin jatkuva koulutus ja kommunikointi.

Suunnittelijoiden persoonalliset ominaisuudet näkyvät usein toimintatavoissa ja useammin pitäisi kyseenalaistaa omaa toimintamallia. Yhteistyötä suunnittelun ja tuotannon kesken täytyisi lisätä huomattavasti, lisääntyvä kommunikointi auttaa

eteenpäin. Mutta asenne täytyy olla kohdillaan. Palaute voi olla positiivista tai negatiivista, aina voimme oppia ja kehittyä.

LIITE 2. Mekaniikkasuunnittelun henkilöstökyselyn kysymykset.

1. iLogic-pohjaiset älykkäät ohjausmallit ovat tehostaneet työntekoasi?
2. Kuinka ohjausmallit ovat projektien läpimenoaikaan vaikuttanut?
3. Ovatko iLogic-mallit käyttäjäystävällisiä?
4. Koetko iLogic-suunnitteluautomaation vähentäneen toistuvia ja puuduttavia tehtäviä?
5. Onko iLogic-ohjausmalleissa tarpeeksi valintoja muokkausta varten?
6. Haluaisitko nähdä lisää iLogic-malleja ja suunnitteluautomaatiota yleisesti?
7. Haluaisitko oppia iLogicia?
8. Mitä mieltä olet tehdystä kehitystyöstä?.
9. Arvio omasta säästyneestä ajasta kuukausitasolla nykyisten kehitystyön tuotoksilla?
10. Vapaa sana (Jos olet käyttänyt kiskoelementtien laskemiseen tarkoitettua exceliä, haluaisin tietää kuinka paljon siihen on kulunut aikaa yms.)

LIITE 3. Henkilöstökyselyn vapaat kommentit.

ID	Vastaukset
1	Noin 2 h / projekti
2	Karkeasti 40 min/ elementti
3	Putkien hanat pitäisi päivittää kaikki samoin päin. Ja oikea hana jahka se saadaan lukittua.. Muutaman kerran tarvittava tiedosto on hukkunut ja sama osa on jouduttu tekemään uudelleen. Eiköhän näistä ajan kanssa hyviä tuotteita tule.
4	En ole käyttänyt, vastasin vain tiedossani oleviin asioihin. Hyvää työtä Mikael ja hyvä kysely! -Janne
5	iLogic ohjausmalleja saisi lisäillä lisää eri vaiheisiin, ja fixailla vanhojen vajavaisuuksia. Esim iLogic palaveri, vaikka kerran kuussa, jossa kerätään palautteita virallisesti yms. Olisi hyvä myös, jos iLogicia osaisi muutkin kuin vain yksi tai kaksi suunnittelijaa, pitäisi löytää aikaa sen opetteluun.
6	Erinomaista toimintaa. Kaikki toistuvat ja suoraviivaiset suunnitteluprosessit tulisi automatisoida, jotta suunnittelua voidaan tehostaa ja käyttää resurssit merkittävien yksityiskohtien parissa -> Laatu paranee ja virheiden määrä pienenee. Skriptien testaukseen, vikasietoisuuteen ja dokumentaatioon täytyy panostaa. Skriptit vaativat myös ylläpitoa. Suunnitteluautomaateille voisi perustaa jonkinlaisen kanavan (tai Git-versiohallinnan), jossa voidaan raportoida poikkeamista ja toivoa ominaisuuksia/antaa automaatioehdotuksia.