

Opinnäytetyö (AMK)

Konetekniikka

Valmistussuunnittelun kehittäminen

2024

Onni Nissinen

RUNKO- JA VARUSTELUSUUNNITTELUN GEOMETRIATIEDOSTOT


TURKU AMK
TURKU UNIVERSITY OF
APPLIED SCIENCES

OPINNÄYTETYÖ (AMK)

TURUN AMMATTIKORKEAKOULU

Konetekniikka|Meritekniikka

2024 | 56 sivua

Onni Nissinen

Runko- ja varustelusuunnittelun geometriatiedostot

Opinnäytetyön tarkoituksena oli kehittää Marine Alutech Oy Ab suunnitteluosaston menetelmiä geometriatiedostojen siirtämisessä ja muokkaamisessa PDM järjestelmään ja tuotantoon.

Työn julkinen osuus tiivistettynä kuvaa suunnitteluprosessia ja tavoitteet ovat selventää lukijalle mahdollisimman selkeästi, millä tavalla valmistussuunnittelussa voidaan kehittää ja tehostaa geometria tiedostojen siirtämistä Visual Basic koodaamisen avulla. Tiedostojen siirtoon tarvitaan myös olennaisena osana kappaleiden merkitseminen tuotannossa. Tutkimalla ohjelmiston ominaisuuksia ja asetuksia, sieltä löytyi ratkaisu koodaamisen hyödyntämiseen.

Tiedonsiirto ja tiedostojen muodostaminen koodauksen avulla toimi hyvin sekä ilman epävarmuustekijöitä. Geometriatasojen luonti automaattisesti pystyttiin toteuttamaan yhdessä VBA-koodin kanssa. PDM-järjestelmän integrointitapa oli myös mahdollista useiden tiedostojen siirrossa. Kappaleiden tunnistetiedon lisääminen geometriatiedoston sisälle automaattisesti oli haastavaa useista menetelmistä huolimatta.

Asiasanat:

Solidworks, 3D-mallinnus, makro, valmistussuunnittelu

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Mechanical Engineering | Naval Architecture

2024 | 56 pages

Onni Nissinen

Hull- and outfitting's geometric files

The purpose of the thesis was to develop the methods of Marine Alutech oy Ab design department in transferring and editing geometry files to production.

The public part of the work summarizes the design process and attempts to explain to the reader as clearly as possible, how in manufacturing design the transfer of geometry files can be developed and enhanced using Visual Basic coding. Marking the parts in production is also an essential point of file transfer. Examining the features of the software was a solution for utilizing coding.

Data transfer and creating files using coding worked well and without uncertainties. The automatic creation of geometry layers could be implemented together with the VBA code. The integration method of the PDM system was also possible in the transfer of several files. Adding the identification information of the pieces inside the geometry file automatically was challenging despite several methods.

Keywords:

Solidworks, 3D-modeling, Macro, Manufacturing Design

Sisältö

1 Johdanto	8
1.1 Yleistä	8
1.2 Työn tavoite ja rakenne	9
1.3 Toimeksiantaja	10
2 Suunnitteluosaston rakenne yleisesti	11
2.1 Runkosuunnittelu	11
2.2 Kone ja varustelusuunnittelu	12
2.3 Sähkösuunnittelu	12
2.4 Sisustussuunnittelu	13
3 Suunnittelun tuottamat geometriatiedostot	14
4 Solidworks mallinnusohjelmisto	15
4.1 Ohjelmiston käyttötarkoitus yrityksessä	16
4.2 Geometria tiedostojen luonti	16
4.3 Mapping asetustiedoston hyödyntäminen	18
4.4 Solidworks kirjaston hyödyntäminen	19
5 Visual Basic historia	21
6 Solidworks VBA-Makro	22
6.1 Yleistä	22
6.2 Codestack ohjelmistosivusto	23
6.3 Geometriatiedostojen teko VBA Makron avulla	24
6.3.1 Katkaisulistan nimeäminen VBA-koodauksen avulla	26
6.3.2 Geometrian muuttaminen tietylle tasolle	29
6.3.3 Ohutlevyasetusten hyödyntäminen	30

6.3.4 Makron tiedoston siirtonopeudet	32
6.4 Parametrit tiedostonimeen makron avulla	34
6.5 Kappaleen tunnisteen muuttaminen geometriaksi	35
6.5.1 Solidworks 3D-tila	35
6.5.2 Progecad parametrinen tekstialue	38
6.5.3 AlphaCam tunnistetieto	40
7 CISS Office Ohjelmisto	42
7.1 Yleistä	42
7.2 Geometria tiedostojen käyttö PDM rakenteessa	45
7.3 Geometriatiedostojen tuonti PDM-rakenteeseen	48
7.4 Tiedostojen haku	49
7.5 Kehitysideat CISSBASE ohjelmistoon	50
8 Yhteenveto	51

Liitteet

Liite 1. Macro to rename SOLIDWORKS bodies of cut-list items using the predefined template

Liite 2. Export flat patterns from SOLIDWORKS part or assembly components

Liite 3. Mapping Entities When Saving Drawings as .DXF or .DWG Files

Kuvat

Kuva 1. Progecad DWG-tiedosto	14
Kuva 2. Polviot	17
Kuva 3. Mapping	19
Kuva 4. Kirjaston linkitetty komponentti (Piirustus)	20
Kuva 5. Makron luominen	22
Kuva 6. Katkaisulistan nimeäminen automaattisesti	26
Kuva 7. VBA koodi	27
Kuva 8. Cutlist taulukko	28
Kuva 9. 2D Geometria	29
Kuva 10. Solidworks asetukset	30
Kuva 11. VBA koodi "Sheetmetal Options"	31
Kuva 12. Kaaren polviot	32
Kuva 13. Parametrinen tiedostonimi	34
Kuva 14. Property valikko 3D-tila	36
Kuva 15. Tunnistetieto mallissa	37
Kuva 16. Progecad tunnistetieto	39
Kuva 17. ALPHACAM asetukset	40
Kuva 18. ALPHACAM nestaus tilan näyttö	41
Kuva 19. Geometritiedostot rakenteessa	45
Kuva 20. DWG-tiedoston kappaleryhmä	47
Kuva 21. Tiedoston tuonti	48
Kuva 22. Hakutoiminto CISSBASE	49

Taulukot

Taulukko 1. Tiedostonsiirtonopeus	33
Taulukko 2. Tiedostonsiirtonopeus Grafiikka	33
Taulukko 3. CISSBASE Ohjelmistot	44

Käytetyt lyhenteet ja sanasto

AlphaCam	Jyrsintäkoneen postproessori ohjelmisto
CISSBASE	Tuotetiedon hallintaohjelmisto
Codestack	VBA-koodi julkaisusivusto
Cutlist	Profiili-/ohutlevy katkaisulista
DXF/DWG	Geometriatiedosto
DNV GL	Kansainvälinen merenkulun luokituslaitos
ERP	Enterprise Resource Planning
Mapping	Tasojen muokkaaminen
Multibody	Kappaletiedoston sisäinen malli
PDM	Product Data Management
Parasolid	Kevennetty tiedosto mallista
Postproessori	Työstöradan ohjelmointi
Progecad	2D/3D ohjelmisto
PS	Port side
SB	Starboard side
Solidworks	3D suunnitteluohjelmisto

1 Johdanto

1.1 Yleistä

Marine Alutech Oy Ab suunnitteluosaston toiminta ja sen ympäristö ovat otollinen kehityskohde. Kasvavan yrityksen ja lisääntyvien tilausten määrä kasvattavat myös suunnitteluosaston ja sen rakenteiden tarkempaa tarkastelua ja kehittämistä. Meriteollisuuden osana yrityksen tuotekategoriassa työveneiden osuus ja niiden valmistaminen ovat suurimmassa osassa tämän yrityksen toimintaa. Tuotteiden suunnittelu perustuu referensseihin ja aiempiin hyväksi todettuihin suunnitelmiin yleisellä suunnittelun tasolla, esimerkiksi rungon valmistuksessa.

Työveneiden rakentaminen perustuu hyvin yksilöityihin ratkaisuihin ja tietyn kategorisen mallin rakentamiseen. Työveneiden rakentaminen ja niiden rungon valmistaminen ovat monesti yksilöityjä ja yksityiskohdat ovat uniikkeja, mutta monesti niiden hyödyntäminen kaikissa venemalleissa voi olla jopa mahdotonta. Usein rungon tukirakenteet, esimerkiksi polviot, jäykisteet ja muut yleiset tukirakenteet ovat jokainen erilaisia lähes kaikissa veneen kaarissa.

Työveneen on tarkoitus olla usein hyvin nopea ja kevyt, tämän takia materiaalina on alumiini tai komposiitti ja veneen runkomuoto on tarkoitettu nopeaan siirtymiseen kiireisellä aikataululla. Muihin tarkoituksiin valmistettavien hitaampien alusten kuormankantokyky on kuitenkin olla huomattavasti suurempi. Runko ja sen valmistaminen on hyvin tarkkaan säädelty luokituslaitosten, esimerkiksi kansainvälisen luokituslaitoksen DNV:n toimesta.

Propulsiossa eli työntövoiman tuottamisessa käytetään vesisuihkuvetolaitteita tai sitten perinteisiä potkureita akselivetoisena. Myös tämän kokoluokan veneissä on yleisesti keulapotkuri. Moottoreina aluksen kokoon suhteutettuna käytetään suuria keskinopeita diesel koneita, joiden valmistajina ovat tunnettuja esimerkiksi MAN ja Scania.

1.2 Työn tavoite ja rakenne

Opinnäytetyössä kehitetään suunnittelumenetelmää geometriatiedostojen hallitsemisessa ja niiden tuottamisessa tuotantoon Solidworks ohjelmalla, sekä osana sitä Progecad ohjelmistolla. Tavoitteena on automatisoida geometriatiedostojen siirtoa suoraan 3D-tilasta, jolloin voidaan ohittaa yhden vaiheen prosessissa ja piirustustilan luominen joissakin tapauksissa kokonaan. Tämä vähentää esimerkiksi geometriatiedoston muokkaamista, siirtämistä ja päivittämistä manuaalisesti.

Solidworks ohjelmistossa ei ole itsessään mahdollisuutta siirtää automaattisesti yhtä aikaa monia geometriatiedostoja suoraan 3D- tilasta omiksi tiedostoikseen, vaan tämän voi tehdä ainoastaan piirustustilan kautta.

Uudessa menetelmässä käytetään geometrian siirtoon VBA- koodausta (Visual Basic for Applications), jossa mallinnusohjelmiston VBA-moduuliin tallennetaan tekstitiedostoja. Näiden koodien avulla voidaan toistaa Solidworks ohjelmiston toimintoja.

Tiedostojen siirtoon tarvitaan myös olennaisena osana kappaleiden merkitseminen suunnittelussa ja tunnistaminen tuotannossa. Tutkitaan suunnittelun ja tuotannon ohjelmistojen ominaisuuksia, mikä voisi olla paras ratkaisu tähän.

Tarkastellaan myös menetelmää, miten voidaan muokata tiedostotunnistetta teknisillä parametreilla. Nämä voisivat olla hyödyllisiä geometrian esimerkiksi tiedostojen lajittelussa. Tämä tapahtuu myös VBA-koodin avulla.

1.3 Toimeksiantaja

Marine Alutech Oy Ab sijaitsee Salon Teijossa. Toiminta on saanut alkunsa vuonna 1985. Yhtiössä valmistetaan ja suunnitellaan erilaisia alumiiniveneitä. Yhtiön asiakkaisiin kuuluu maailmanlaajuisesti esimerkiksi rajavartiolaitokset ja erilaiset pelastus- ja satamaviranomaiset.

Kotimaassa yhtiö valmistaa ja huoltaa esimerkiksi puolustusvoimille taistelu-, ja kuljetusveneitä. yhtiön asiakkaita ovat myös vartio-, luotsi-, SAR-, ja työveneet merenkulku- ja satamaviranomaisille. Viimeisimpänä tilauksena Jurmo-luokan kuljetusveneet.

Yhtiöllä on myös huoltokeskus Hangon Lappohjassa. Veneen suunnittelun ja rakentamisen lisäksi Marine Alutech tarjoaa myös valmistetuille aluksille huoltosopimuksia asiakkailleen.

Marine Alutechin suunnittelu perustuu yleisesti viranomais- ja luokituslaitosten säännöstöihin ja täyttää turvallisuusvaatimukset erilaisissa meriympäristöissä maailmanlaajuisesti.

(www.marinealutech.com)

2 Suunnitteluosaston rakenne yleisesti

2.1 Runkosuunnittelu

Runkosuunnittelun tavoitteena on luoda asiakkaan lähtökohdista käyttötarkoitukseen sopiva ja oikean kokoinen runkomuoto. Rungon muodossa otetaan huomioon nopeuden lisäksi siihen asennettavien komponenttien sijaintia kokoa sekä painojakautumaa niiden välillä. Propulsioon valinta vaikuttaa esimerkiksi perän ja pohjan muotoiluun hyvin paljon.

Materiaalina alumiini asettaa myös erilaisia vaatimuksia rungon jäykkyyden ja kestävyys suunnittelussa. Esimerkiksi rungon hitsaaminen ja materiaalin eläminen täytyy ottaa huomioon jo suunnittelun alkuvaiheessa.

Usein suunnittelun pohjana rungossa käytetään valmista referenssiä, jota muokataan tarpeen mukaan sopivaksi ja tiettyyn käyttötarkoitukseen. Tarkemmassa runkosuunnittelussa täytyy ottaa myös huomioon luokituslaitosten säännöt ja laskelmat. Runkosuunnittelussa tuotetaan valmistus-, osavalmistus-, luokituslaitoksen hyväksyntää varten erilaisia rakenteisiin liittyviä piirustuksia ja dokumentteja sekä myös asiakkaan hyväksyntää varten useita valmistusdokumentteja. Erilaiset kaaviot luokituslaitosta varten, kuten esimerkiksi läpivienteihin liittyvät piirustukset rungossa ja kannessa sekä kuoressa.

Prosessissa tarvitaan monta suunnittelukierrosta, ennen kuin lopulliset dokumentit ja kuvat ovat valmiita. Myös eri suunnittelun vaiheissa voidaan muokata tai muuttaa rakennetta suunnittelun edetessä. Usein tämä tulee eteen, kun aletaan tekemään tarkempaa yksityiskohtaista suunnittelua.

Yhdessä runkosuunnittelun kanssa kulkee koko ajan paino- sekä vakavuuslaskenta, jonka tavoitteena on pysyä tietyissä raameissa asiakkaan sekä luokituslaitoksen sääntöjen kanssa.

2.2 Kone ja varustelusuunnittelu

Tämän osaston tavoitteena on valita yhdessä asiakkaan kanssa parhaat mahdolliset prosesseihin ja laitteisiin liittyvät komponentit.

Näistä lähtökohdista lasketaan myös aluksen optimaalinen ja tehokas propulsio-laitteisto ja konetehto, jolla päästään asiakkaan vaatimustasolle, esimerkiksi nopeuden suhteen.

Tietyn mallisen aluksen täytyy myös palvella erilaisia asioita, esimerkiksi puolustusvoimien, luotsaus- tai pelastusaluksen ominaisuuksiin tarkoituksen mukaisesti. Tämän takia aluksia varustellaan hyvinkin eri näkökohdista ja kaikkien laitteistojen täytyy olla tarkkaan määriteltyjä sekä spesifejä tietyn tyyppisessä tehtävässä.

Luokituksen mukaisesti kaikkiin aluksiin varustellaan myös merenkulkuun liittyvät antennit, tutkat, pelastuslaitteet ja navigointijärjestelmät.

Osastolle kuuluu myös kaikkien systeemien prosessikaavioiden laadinta, sekä luokituslaitoksen hyväksyntään tulevat kaaviot.

2.3 Sähkösuunnittelu

Osana varustelua myös sähkösuunnittelu on mukana varustesuunnittelussa asiakkaiden, sekä laitetoimittajien kanssa yhteistyössä. Osaston tarkoituksena on laskea aluksen sähkökuormitusta erilaisissa olosuhteissa myös asiakkaan vaatimusten mukaisesti.

Osastolla suunnitellaan alukseen sähkökaaviot sekä kaapeloinnit. Näiden järjestelmien pohjalta voidaan myös osaltaan tehdä eri osastoihin liittyvät sähköjärjestelmien hankinnat ja määritellä vaatimukset yksittäisille laitteistoille.

Nykyaikaisissa veneissä on paljon erilaisia automaatiojärjestelmiä, jotka liittyvät esimerkiksi luokituksen vaatimuksiin ja merenkulun kansainväliseen turvallisuuteen.

Sähkösuunnittelun tehtävänä on myös automaatio- ja navigointijärjestelmien hankinta, sekä suunnittelu yhteensopivaksi aluksen muiden järjestelmien kanssa.

Suunnittelu on yhteistyötä muun suunnitteluorganisaation kanssa, sekä tuotannossa projektien eteenpäin viemisessä.

2.4 Sisustussuunnittelu

Veneen sisäpuolinen runko kokonaisuudessaan sisustetaan erilaisilla materiaaleilla. Osastokohtaisilla materiaaleilla on tarkoitus eristää ja suojata rungon materiaaleja paloturvallisuuden tai ääni- ja lämmöneristyksen osalta. Myös erilaiset ympäristöt vaikuttavat esimerkiksi lämmöneristykseen rungossa sekä ohjaamossa.

Sisustussuunnittelussa tuotetaan projektiin liittyvät luokituksen määräyksiin perustuvat dokumentit, esimerkiksi paloluokitukseen liittyvät materiaalit, paksuudet ja rungon palosuojaukset.

Oleskelutiloihin suunnitellaan asiakkaan vaatimusten pohjalta sisustuselementit ja kalusteet, kaapistot ja säilytystilat. Yleiset sosiaalityilat. Ohjaamoon liittyvät sisustusmateriaalit, eristykset ja kalusteet.

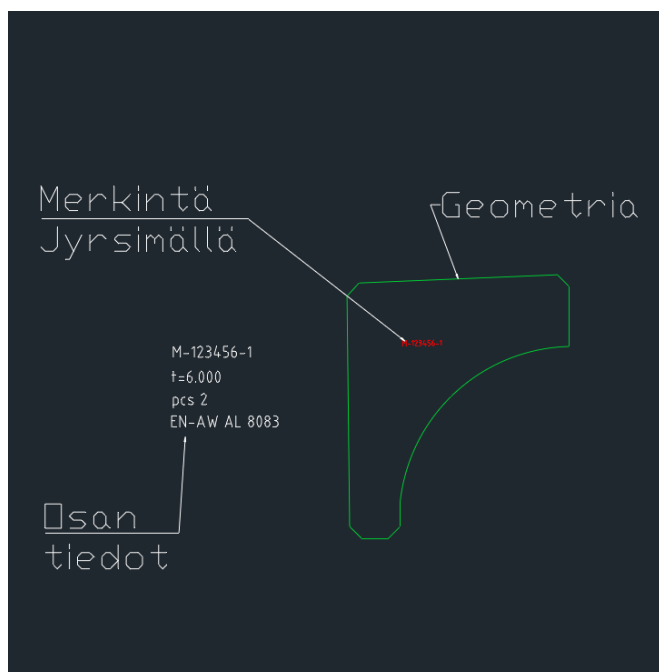
Veneiden erilaiset käyttötarkoitukset asettavat myös vaatimuksia turvalliseen operointiin esimerkiksi rajaturvallisuudessa tai kriisitilanteessa. Näissä tilanteissa esimerkiksi ohjaamon suojaus on myös yksi osa sisustusratkaisua.

3 Suunnittelun tuottamat geometriatiedostot

Osastot, joissa suunnitellaan Solidworks ohjelmistolla erilaisia ohutlevy kappaleita, syntyy geometriatiedostoja, joiden pääasiallisena tehtävänä on tuottaa leikkausgeometriaa, reikiä, muotoja sekä teknistä tietoa geometria tai objekti muodossa. Tiedostot muutetaan myöhemmin postprosessorilla työstökoneen ymmärtämään muotoon.

Yleisesti geometria tiedostot ovat muodossa DWG tai DXF. DWG on uudempi tiedosto formaatti, johon voidaan sisällyttää enemmän tietoa kuin vanhempaan DXF formaattiin, mutta kuitenkin yksinkertaisessa muodossaan kumpikin muoto käy.

Opinnäytetyössä pyritään menetelmään, jossa yksittäinen kappale identifioidaan omaksi tiedostokseen ja nimetään mallin tai cutlistan mukaan. Työn tavoitteena on myös siirtää yhä suurempi osa tuotetiedosta piirustusdokumenttiin. Tässä kuvassa esitetään myös esimerkiksi materiaaliin ja tilaukseen liittyvää tietoa, jota pyritään minimoimaan uudessa menetelmässä. (Kuva 1).



Kuva 1. Progecad DWG-tiedosto

4 Solidworks mallinnusohjelmisto

Ohjelmalla voidaan tehdä kolmenlaisia perustiedostoja; Osia, kokoonpanoja sekä valmistuspiirustuksia. Nämä ovat toisiinsa sidoksissa siten että, muutettaessa osaa kokoonpanossa myös osatiedosto muuttuu sekä sen piirustus muuttuu, tai päinvastoin. Osien tai kokoonpanojen mittoja voidaan myös linkittää siten että toista osaa muutettaessa muut kappaleet seuraavat muutetun kappaleen mittoja. Mittoja on myös mahdollista linkittää excel- taulukkolaskentaohjelmaan. On myös mahdollista tehdä samasta osasta monta eri "versiota" eli niin sanottuja konfiguraatioita, joilla pyritään helpottamaan suunnittelijan työtä. Esimerkiksi, voidaan luoda yksi malli ikkunalasista, johon luodaan useita eri konfiguraatioita. Tässä tapauksessa erikokoisia ikkunoita, joiden leveys ja korkeusmitat ovat erilaisia. Tämän jälkeen yhtä ikkunamallia voidaan käyttää kokoonpanossa, jossa tarvitaan erikokoisia ikkunoita.

Mallintaminen perustuu piirteisiin, ehkä suurimpana erona 2D CAD ohjelmistoihin, käyttäjän luoma kappale tai kokonaisuus on muokattavissa erittäin vapaasti, ilman että työ olisi aloitettava uudelleen alusta.

Mallinnus aloitetaan yleensä piirtämällä kaksiulotteinen kuva kappaleesta X, Y-tasolle (tai vaihtoehtoisesti XZ/YZ- tai muulle aputasolle), jonka jälkeen syntyneestä "sketsistä" tehdään pyörähdyskappale tai se pursotetaan kolmanteen suuntaan. On myös mahdollista luonnostella 3- ulotteinen hahmotelma halutusta kappaleesta, jos tämä vaikuttaa järkevämmältä, esimerkiksi jos kyseessä on jokin taivutettu putkirakenne.

Kaksi erilaista kokoonpanojen suunnittelutapaa on kuitenkin selvästi eroteltavissa; ns. Bottom - up ja top - down. Nimensä mukaisesti ensimmäisessä valmistetaan ensin laitteen osat ja näistä tehdään isompi kokonaisuus eli kokoonpano. Jälkimmäisessä mallinnetaan uusia osia jopa täysin tyhjiin kokoonpanoon. Menetelmää voi vaihtaa työn edetessä.

David E. Weisberg Solidworks (PDF) cadhistory.net.7.3.2022 (englanniksi)

4.1 Ohjelmiston käyttötarkoitus yrityksessä

Solidworks on yrityksen suunnittelussa pääohjelmisto, jolla tehdään uusia 3D-malleja, yhdistellään erilaisia kokonaisuuksia ja kokoonpanoja.

Rungon pintamalli luodaan muilla ohjelmilla ja vain rakenteet/varustelu Solidworks ohjelmalla.

Ohjelmistolla tehdään suurin osa kappaleiden valmistus- sekä työohjeistuksesta tuotantoon.

Ohjelma on myös linkitetty PDM-alustaan (CISBASE), jossa hallitaan 3D-tiedostoja ja niihin liittyviä muita dokumentteja ja tiedostomuotoja, esimerkiksi geometriatiedostoja sekä PDF-tiedostomuotoja.

4.2 Geometria tiedostojen luonti

Solidworks ohjelmisto osaa tuottaa geometria tiedostoja suoraan 3D tilasta, tai sen jälkeen, kun mallista siirretään ohutlevyosaksi muutettu kappale piirustukseen.

Ensiksi kuitenkin 3D-mallista kannattaa luoda ohutlevy kappale, jos halutaan myöhemmin, että se osaa tehdä taivutetusta osasta levityskuvan leikkaamista varten.

Ohjelmistossa on oletuksena valmiina myös parametriset arvot erilaisille taivutusarvoille. Näitä arvoja voidaan myös muuttaa tarvittaessa yrityksen omille työstökoneille, esimerkiksi excel - taulukko pohjaisten muutosparametrien avulla.

Myös piirustukseen saa oletuksena esimerkiksi taivutuskulman ja säteen.

Erilaisten asetusten avulla voimme määrittää, mitä tietoja tai geometriaa siirtyy joko piirustus pohjaan tai suoraan geometriatiedostoon.

Toimintamenetelmänä suunnittelussa on käyttää mallinnusohjelmistoa multibody tiedostoina, esimerkiksi rungon jäykisteiden ja esimerkiksi polvioiden mallintamisessa. Näin saadaan tiedostosta kevyempi ja kokoonpanoja on runkomallissa paljon vähäisempi määrä. Näin myös suurempien kokoonpanojen kanssa työskentely on helpompaa.

Suurien kokoonpanojen kanssa kannattaa käyttää myös parasolid-mallia, josta on karsittu piirrepuu. Parasolid koostuu erikokoisista objekteista, joiden mallintamiseen tarvittavat Featuret ja niiden sisällä olevat parametrit ovat karsittu pois. Ohutlevymalleja pystyy kuitenkin luomaan myös parasolid tiedostosta uudelleen.

4.3 Mapping asetustiedoston hyödyntäminen

Solidworks ohjelmistoon on rakennettu toiminto, jolla voidaan hyödyntää eri DXF/DWG tiedoston tasojen siirtämistä oikeilla väreillä tai muuten hyväksi havaitsemilla asetuksilla tuotannon käytettäväksi. Tätä sanotaan nimellä Mapping. Tuotannossa on jo valmiiksi määrättyjä asetuksia postprosessorille, jolloin niitä ei tarvitse muuttaa erikseen käsin DXF/DWG tiedostoista käsin. Tällä tavalla voidaan automaattisesti lisätä geometria tiedostoon erilaisia värejä, tekstiä tai jopa pelkkiä piirteitä 3D ohjelmistosta eri tasoille.

Tämä ominaisuus parantaa merkittävästi tiedostojen tuottamista automaattisesti ja vähentää virheiden määrää. Asetusten muokkaaminen ja niiden kokeileminen käytännössä auttaa ymmärtämään vasta, kuinka hyödyllinen tämä ominaisuus on.

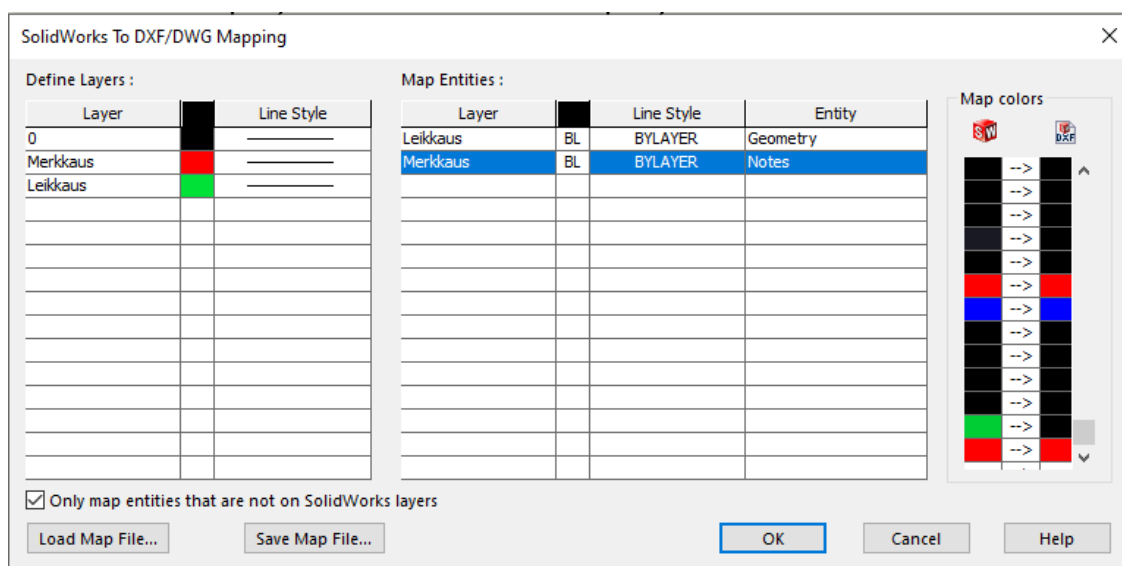
Mapping asetukset tallennetaan tiedostoon, jota voidaan muokata, jakaa tai päivittää tarvittaessa. Värien tarkoitus on erotella postprosessorille määriteltävät työstöradat. Tässä tapauksessa merkkkaus tarkoittaa myös työstöä jyrsimällä tietyn kokoisella fontilla, joka sekin voidaan määrittää erikseen. Leikkaus tarkoittaa yleensä kappaleen leikkaamista koko materiaalin läpi. Kuitenkin jyrksinnässä voidaan käyttää myös taso-/taskujyrshintä, sekä erilaisia pyöristyksiä.

Solidworks ohjelmiston olioita voi olla monta kappaletta yhtä geometriatasoa nähden, tai vastaavasti geometriatasoja voi olla enemmän eri väreillä yhtä oliota nähden.

Esimerkkiasetukset (Kuva 3).

Luomme 3 tasoa geometriaan, jotka tulevat DWG tiedostoon.

TASO	VÄRI	SOLIDWORKS OLIOT
0	Musta	Kaikki muut oliot
Merkkaus	Punainen	Merkintä
Leikkaus	Vihreä	Geometria



Kuva 3. Mapping

4.4 Solidworks kirjaston hyödyntäminen

Kirjastossa on valmiiksi erilaisia työkaluja, muotteja ja komponentteja. Kirjastoon voi tehdä myös itse erilaisia tekstejä, jotka ovat suoraan linkitettyinä 3D mallin parametreihin, esimerkiksi juuri kappaleiden tiedostonimen merkkkaus, materiaali ja kappalemäärä voivat olla yksi tapa nimetä ohutlevykappaleita piirustustilassa.

Näitä voidaan hyödyntää, kun kappaleista on olemassa oikeat tiedot esimerkiksi Description kentässä Property asetuksissa ja ne ovat nimetty oikein.

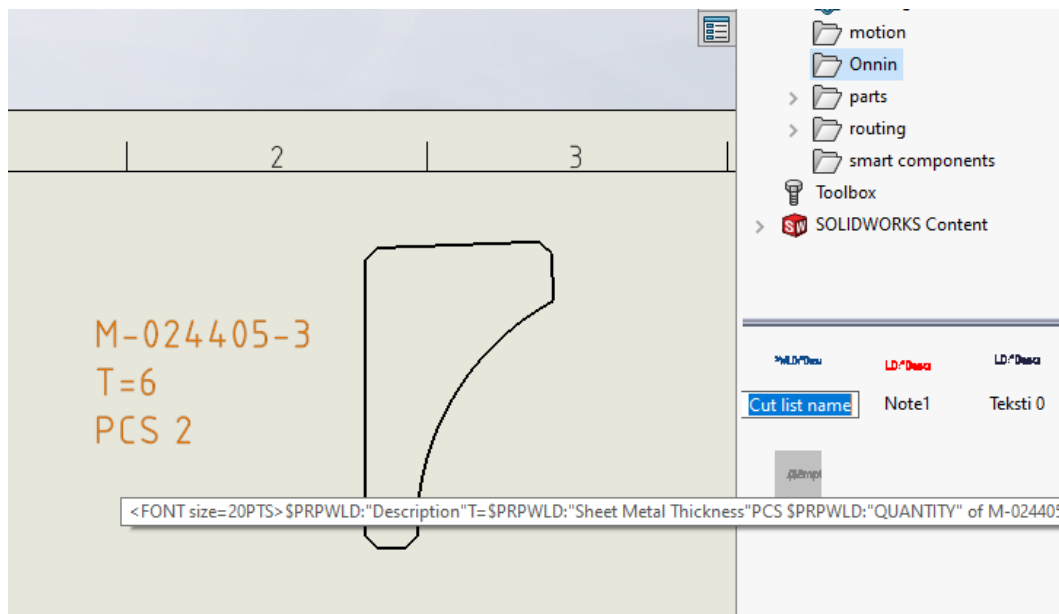
Tämä menetelmä on todettu hyväksi ja sillä voidaan yksilöidä erilaisia kappaleita toisistaan hyvin tehokkaasti ennen geometriaksi muuttamista (Kuva 4).

Piirustukseen on tuotu polvio, kirjastosta voidaan siirtää hiirellä aiemmin tallennettu objekti kappaleen päälle, jolloin objekti hakee mallin asetuksista seuraavat:

Cut-listan nimi (M-024405-3)

Materiaalin paksuus (T=6)

Kappalemäärä (PCS 2)



Kuva 4. Kirjaston linkitetty komponentti (Piirustus)

5 Visual Basic historia

Visual Basic on Microsoftin kehittämä BASIC-sukuinen yleiskäyttöinen ohjelmointikieli. Visual Basic pohjautuu 1980-luvulla julkaistuu QuickBASIC-kielen. Visual Basic saavutti 1990-luvulla laajan suosion, joka on jatkunut 2000-luvun aikana. Kieltä käyttävät sekä aloittelijat että ammattilaiset ja se soveltuu sekä pienten että laajojenkin ohjelmien laadintaan.

Visual Basic -kielen ensimmäinen versio esiteltiin vuonna 1991. Versio 1.0 oli saatavissa sekä Windowsille että MS-DOS-käyttöjärjestelmään. DOS-version kehitys kuitenkin loppui ja seuraavat versiot (2.0–6.0) toimivat enää vain Windowsissa. Vuonna 2002 tapahtui kielessä merkittävä uudistus. Tuolloin julkaistiin Visual Basic .Net (VB.NET), joka on myös kielen nykyinen versio. Se kuuluu Microsoftin .NET-perheeseen.

VB.NET-uudistuksesta huolimatta vuonna 1998 julkaistu Visual Basic 6.0 on edelleen laajalti käytössä. Yhtenä syynä vanhan version käyttöön on se, että VB.NET poikkeaa huomattavasti aiemmista versioista. Vanhojen ohjelmien muunnos VB.NET-ympäristöön on suhteellisen työlästä. 6.0-versiolla laaditut ohjelmat myös toimivat paremmin vanhoissa käyttöjärjestelmissä.

Visual Basic .Net on ohjelmointikielenä jonkin verran lähempänä C-sukuisia kieliä kuin aiemmat Visual Basicin versiot. Tämä johtuu pitkälti .NET-Frameworkin käytöstä. Ominaisuuksiltaan VB.NET vastaa hyvin pitkälle C#-kieltä. Ensimmäisissä versioissa mahdollisuudet olio-ohjelmointiin olivat hyvin rajalliset, mutta VB.NET:in myötä kielestä on tullut täysiveroinen oliokieli.

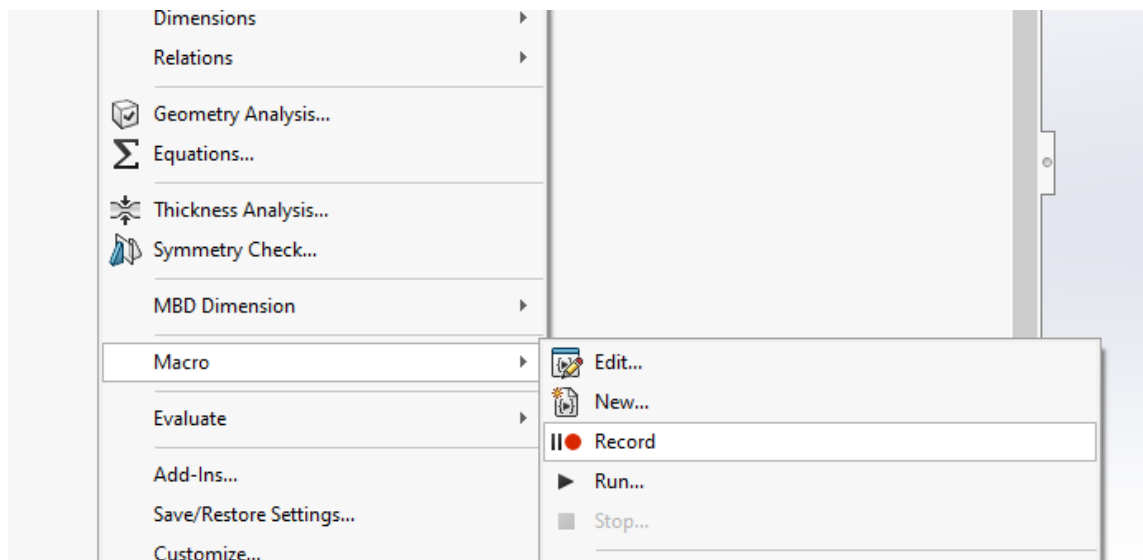
(https://fi.wikipedia.org/wiki/Visual_Basic)

6 Solidworks VBA-Makro

6.1 Yleistä

Solidworks on tuonut ohjelmistoon jo vuosia sitten VBA-koodaus ominaisuuden, joka tarjoaa käyttäjälle mahdollisuuden luoda useita erilaisia automaattisia käskykomentoja, tai tuoda ja tallentaa uusia komentojonoja, jotka helpottavat huomattavasti esimerkiksi paljon toistoa vaativia toimintoja. Makrojen toiminnot voivat liittyä joko mallinnustilaan tai piirustus pohjaan, tai molempiin yhtä aikaa. Käyttäjä voi siis luoda koodauksen avulla täysin samoja toimintoja, kuin hän käyttäisi niitä yksitellen käyttäen erilaisia featureja tai erikoistoimintoja, joita käytetään esimerkiksi ohutlevymallien tekemiseen tai muokkaamiseen.

Moduulissa on myös VBA-koodauksen tapaan mahdollisuus niin sanotusti nauhoittaa toimintojen tekemistä, jolloin VBA-generaattori osaa luoda tästä toiminnosta koodin, jota voidaan myös muokata jälkepäin. (Kuva 5).



Kuva 5. Makron luominen

6.2 Codestack ohjelmistosivusto

Sivusto tarjoaa avoimen lähdekoodin kautta valmiita makropaketteja esimerkiksi Solidworks mallinnusohjelmiston käyttäjille. SOLIDWORKS-työkalumakron julkaisuja on esitelty seuraavassa.

Codestack sivustolla on julkaistu kaksi pääluokkaa makroja:

Koulutusmakrot:

Ne näyttävät tietyn AP:n käytön, mutta makro itsessään ei ratkaise mitään ongelmaa. Se on tarkoitettu SOLIDWORKS API -kehittäjien käytettäväksi. (API Application Programming Interface)

Apuohjelman makro:

Vaikka sitä voidaan käyttää myös opetustarkoituksiin, tämä makro ratkaisee tietyn ongelman ja sitä voivat käyttää kuka tahansa SOLIDWORKS API -kehittämistaidoista riippumatta. Tällaiset makrot julkaistaan SOLIDWORKS-osiossa.

Jotta artikkeli julistettaisiin SOLIDWORKS-työkaluksi, sivulle on määritettävä 2 lisämääritettä: layoutin tulee olla yhtä suuri kuin sw-tool ja ryhmän tulee olla yhtä suuri kuin tämän makron luokka. Tällä hetkellä tuettuja ryhmiä ovat esimerkiksi:

- Malli
- Materiaalit
- Kehys
- Kehittäjät
- Mukautetut ominaisuudet
- Osa
- Kokoonpano
- Piirustus
- Turvallisuus
- Luonnos

- Esitys
- Geometria
- Tuonti ja vienti
- Liiketutkimus
- Vaihtoehdot
- Leikkaa-lista

<https://www.codestack.net/solidworks-tools/>

Esimerkkien avulla kuka tahansa henkilö voi olla kehittämässä sivustoa ja tuoda artikkelin ja omia kehitysehdotuksia esimerkiksi Solidworks ohjelmiston ominaisuuksien kehittämiseen ja ohjelmiston käyttämiseksi entistä tehokkaammalla tavalla.

Solidworks käyttäjät voivat antaa myös parannusehdotuksia jo valmiiksi julkaistuihin makropaketteihin, jolloin myös erilaisissa tilanteissa esiin tulevat virheet voidaan myöhemmin korjata.

6.3 Geometriatiedostojen teko VBA Makron avulla

Ohutlevyjen geometrian muokkaamiseen ja siirtämiseen voimme käyttää valmiiksi luotuja tekstitiedostoja, joilla pystytään luomaan yhdestä multibody kappaleesta geometria tiedostoja. Tämä sama VBA-koodi pystyy tekemään saman myös kokoonpano tilassa. (Liite 2).

Uudessa menetelmässä muunnamme jokaisen ohutlevyosan geometriaksi DXF-kansioon, johon tallentuu kaikki osat yhdestä mallitiedostosta. Makron ajaminen edellyttää PDM-ohjelman kautta avatun mallin varaamista omalle koneelle. Kansiorakenne voidaan määrittää suoraan makroon, ja samalla voidaan luoda sille tiedostopääte.

VBA-Koodin ominaisuuksiin kuuluu myös tiedostomuodon valinta. Joka yksinkertaisesti määritetään DXF – tai DWG muotoiseksi.

Makron käyttäminen suoraan 3D-tilasta nopeuttaa geometrian tekemistä, mutta tuotannon kannalta oleellista tuotetietoa esimerkiksi tekstiä ei voi sisällyttää mukaan tiedostoon. Geometriaan voidaan kuitenkin lisätä erilaisia merkintäviivoja ohutlevystä, kuten esimerkiksi särmäys.

Särmäyksen mitoitus levyn reunasta on tärkeää, koska nykyaikaisilla koneilla särmäys tapahtuu vasteiden avulla. Ohjelmistoilla voidaan myös luoda valmiita särmäysohjelmia. Näillä voidaan suoraan määrittää vasteiden mitoitus särmäyskohdasta.

Viivojen merkinnässä on tuotannolla myös mahdollisuus tarkistaa geometrian ja piirustuksen välillä. Joissakin tapauksissa voidaan käyttää suoraan merkintäviivaa mitoitukseen tai suoraan taivutuksen tekemiseen.

Alapuolisia merkintöjä ei tarvitse käyttää, koska ahiota ei käännetä yleensä jysintäalustalla. Kuitenkin erikoistapauksissa tämäkin on mahdollista. Yleisesti suuremmissa kokonaisuuksissa voidaan käyttää myös merkintää esimerkiksi kappaleiden paikoituksessa, hitsausta tai muuta kappaleiden kokoamistapaa ajatellen.

Yleisenä käytäntönä suunnittelussa nykyään käytetään piirustusohjelman kautta siirtoa, jolloin levitysmalli luodaan kappaletta siirrettäessä piirustukseen.

Käyttämällä piirustusohjelmia sekä luomalla geometria esimerkiksi polviolle, dokumentoinnissa geometriatiedostoon lisätään tuotetietoa, kuten kappaleen nimitys, kappalemäärä ja materiaali. Näiden tietojen käyttäminen geometriana oleellista tuotannon kannalta.

Seuraavassa menetelmässä voimme luoda katkaisulistan nimet tiedostopäätteestä automaattisesti makron avulla.

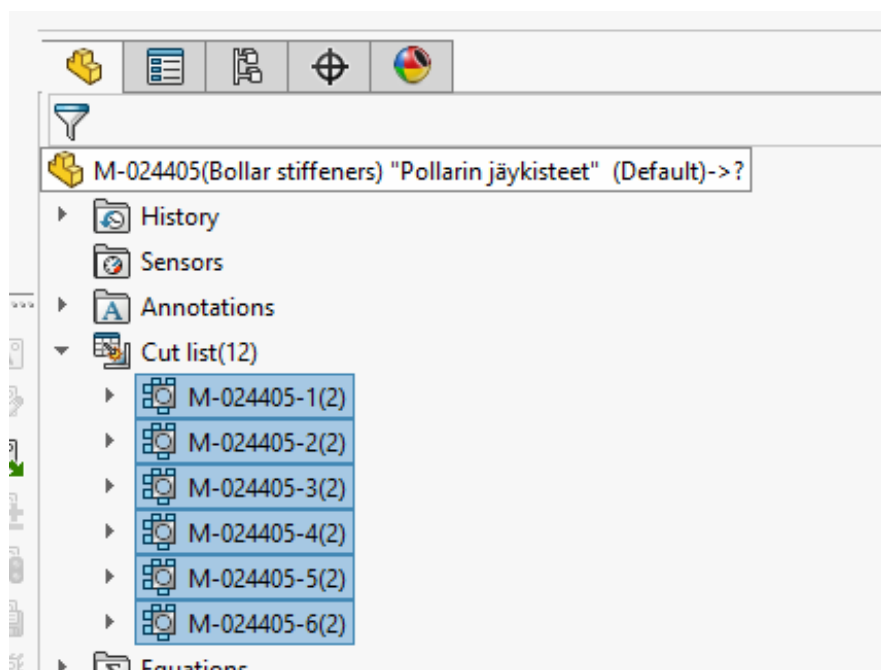
Makron käyttäminen osien numeroinnissa

6.3.1 Katkaisulistan nimeäminen VBA-koodauksen avulla

VBA koodauksen käyttäminen luotujen ohutlevyosien listauksessa, jolla voidaan tietyin ehdoin nimetä katkaisulistan kansiot uudelleen, esimerkiksi käyttäen juuri eriteltyä tapaa nimetä jokainen ohutlevymalli, jolloin osan perään tulee liite:

M-000001-1, -2, -3....

() Suluisissa, ohjelmisto osaa laskea identtiset tai peilatut kappaleet. (kuva 6).



Kuva 6. Katkaisulistan nimeäminen automaattisesti

Tekstiedosto (Liite 1) voidaan tallentaa suoraan uudeksi makroksi VBA-kooditilaan, tekstiedoston alkuun kirjoitetaan myös automaattisesti julkaisijan Copyright-leima. (Kuva 7).

```

Microsoft Visual Basic for Applications - Cut_list_names - [Cut_list_names1 (Code)]
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Project - Cut_list_names
Properties - Cut_list_names1
Cut_list_names1 Module
Alphabetic Categorized
(Name) Cut_list_names1

'*****
'Copyright (C) 2023 Xarial Pty Limited
'Reference: https://www.codestack.net/solidworks-api/document/cut-lists/rename-cut-list-items/
'License: https://www.codestack.net/license/
'*****

Const NAME_TEMPLATE = "<$PRP:PartNo>"
Const INDEX_FORMAT As String = "-0"
Const ALWAYS_ADD_INDEX As Boolean = True

Dim swApp As SldWorks.SldWorks

Sub main()

try_
  On Error GoTo catch_

  Set swApp = Application.SldWorks

  Dim swModel As SldWorks.ModelDoc2

  Set swModel = swApp.ActiveDoc

  If Not swModel Is Nothing Then

    Dim vCutLists As Variant
    vCutLists = GetCutLists(swModel)
  
```

Kuva 7. VBA koodi

Cutlistan nimeäminen on yleisesti tärkeää, koska sillä voidaan yksilöidä ohutlevy osat. Näitä samoja parametrejä käytetään myös piirustuksien tekemisessä ja kappaleiden yksilöimisessä sekä kappalemäärän varmistamisessa. Ohjelma osaa ottaa huomioon myös peilatut kappaleet erikseen, ja laskea ne mukaan listaukseen ja piirustuksen taulukkoon

Tallennettuun taulukkoon voidaan määrittää erilaisia parametreja ja tässä esimerkissä on määritelty esimerkiksi katkaisulistan nimet, jotka tulevat näkyviin DESCRIPTION kohtaan. Myös tuotannon kannalta oleelliset tiedot, kuten kappalemäärä, materiaali ja paksuus ovat tärkeää tietoa.



Nimike-kenttä on esimerkin vuoksi tyhjä. Tähän kohtaan lukee automaattisesti yleensä myös projektin eli aluksen tyyppi, sekä 3D-mallin selitys. Nämä asetukset tulevat suoraan PDM ohjelman Property asetuksista.

Piirustusohjelmat ovat yleisesti vakioita, joihin ei tule mitään manuaalista käsin kirjoitusta taulukkokenttiin tai parametriin tietoihin.

PDM-ohjelmiston kautta dokumentit päivittyvät automaattisesti. Myös viittaukset eri kokoonpanoon tai osan omaan valmistuspiirustukseen voidaan liittää erilaisiin POM-luetteloihin.

Nimiketaulukko päivittyy myös PDM-ohjelmiston kautta ja siihen merkitään yleisesti projektin nimi, eli veneen malli tai valmistusnumero. (kuva 9).

Weldment Cut List (M-000002)				
ITEM NO.	QTY.	DESCRIPTION	Sheet Metal Thickness	MATERIAL
1	1	M-000001-1	3	3.3547 (EN-AW 5083)
2	1	M-000001-2	6	3.0515 (EN-AW 3103)
3	1	M-000001-3	20	3.3545 (EN-AW 5086)

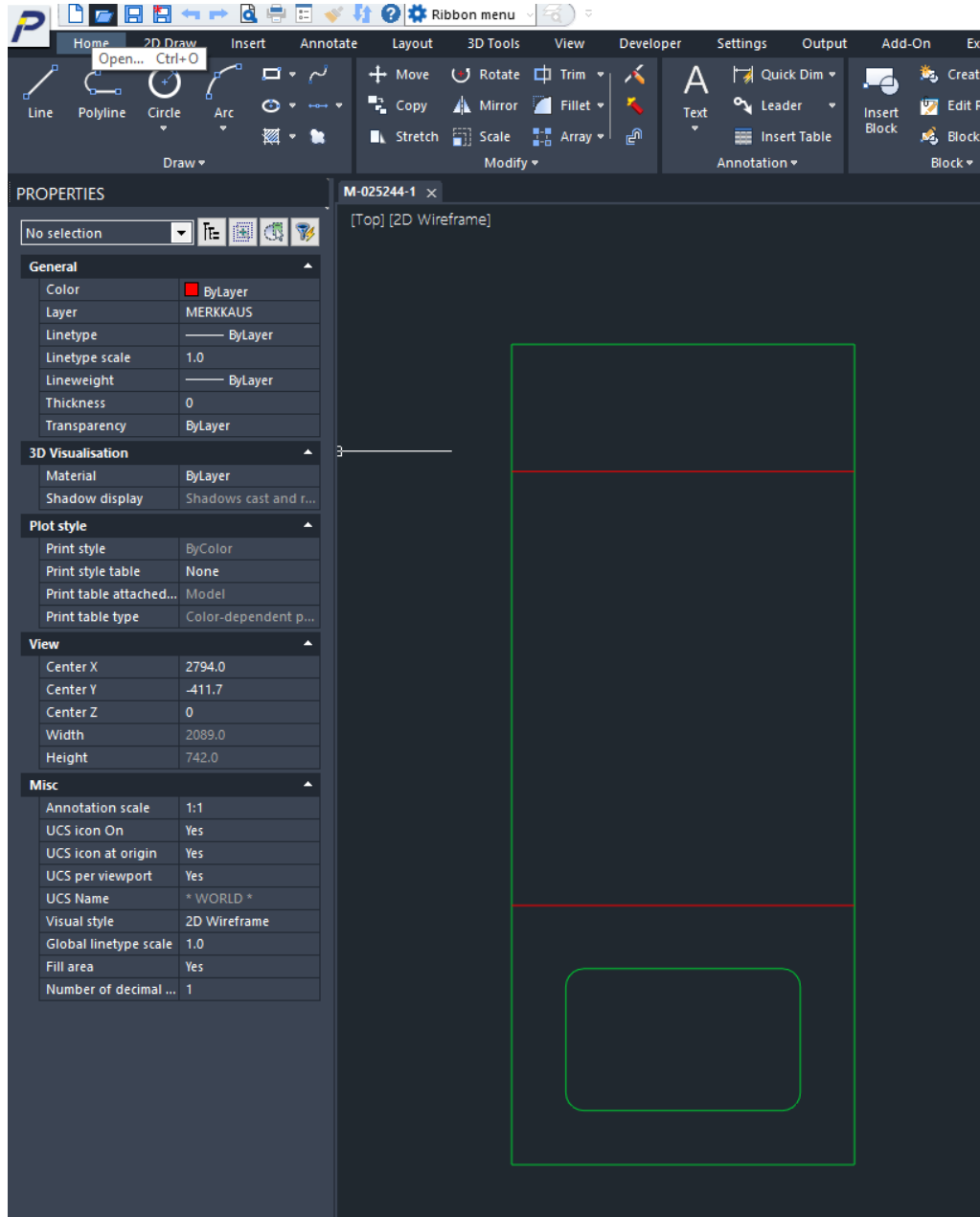
THE INFORMATION HEREON IS THE PROPERTY OF MARINE ALUTECH OY AB WITHOUT WRITTEN PERMISSION, ANY COPYING, TRANSMITTAL TO OTHERS, AND ANY USE EXCEPT THAT FOR WHICH IT IS LOANED IS PROHIBITED				
Filename DRW-00319.SLDDRW	STAND. 	HYV.	SUHDE 1:1@A3	PVM
		SUUN RWE	NIMIKE/TUOTE New drawing	
Telakkatie, 25570 Teijo, Finland Tel. 358 2 7288100 Fax. 358 2 7288129 e-mail. info@marinealutech.com		MASSA 129 kg	SIVU 1(1)	PIIR.NO. DRW-00319 Edition
6		7		8

Kuva 8. Cutlist taulukko

6.3.2 Geometrian muuttaminen tietylle tasolle

VBA makro osaa hyödyntää myös aiemmin tehtyä Mapping-tiedostoa. (Kuva 9).

- Leikkaus taso muutetaan vihreälle värille
- Merkkkaus taso muutetaan punaiselle värille (Taivutusviivat)



Kuva 9. 2D Geometria

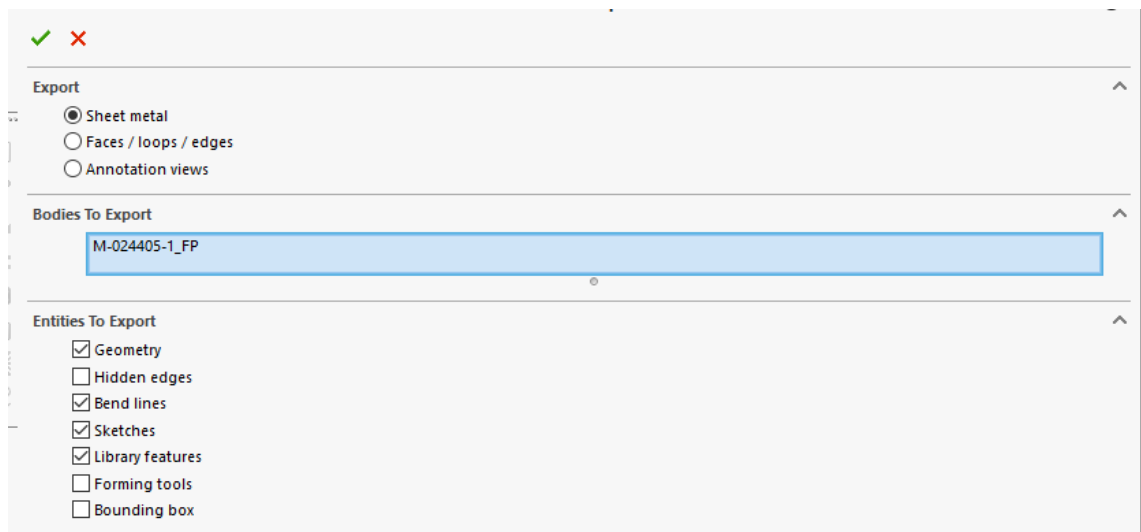
6.3.3 Ohutlevyasetusten hyödyntäminen

VBA-Koodiin (Liite 2) on sisäänkirjoitettu mahdollisuus ottaa huomioon myös ohjelmiston asetuksia muodostaessaan geometriatiedostoa, esimerkiksi ohutlevyasetuksia.

VBA-koodin alussa "const FLAT PATTERN OPTIONS As Integer" kohtaan syötetään kaikki asetukset, joiden halutaan aktivoituvan, esimerkiksi:

- Kirjastoon luodut mallit
- Taivutusviivat
- Ohutlevygeometria
- Luonnokset

Manuaalisesti geometriatiedostoa luotaessa asetukset näyttää seuraavalta. (Kuva 10).



Kuva 10. Solidworks asetukset

VBA-koodin alussa käyttäjä voi lisätä tai poistaa ohutlevyasetuksia kohdassa 'Const FLAT_PATTERN_OPTIONS', joka lisää asetuksen geometrian luontivaiheessa. (kuva 11).

Uuden tallentamisen jälkeen voidaan myös huomata yksi makron ominaisuus, jolla voidaan ohittaa edellinen samanniminen tiedosto. Tämä on hyvä ottaa huomioon suurempien tiedostomäärien tekemisessä. Myös tämän asetuksen muokkaaminen tapahtuu VBA-koodin alussa.

'Const SKIP_EXISTING_FILES As Boolean = False

```
*****
'Copyright (C) 2023 Xarial Pty Limited
'Reference: https://www.codestack.net/solidworks-api/document/sheet-metal/export-all-flat-patterns/
'License: https://www.codestack.net/license/
*****

Enum SheetMetalOptions_e
    ExportFlatPatternGeometry = 1
    IncludeHiddenEdges = 2
    ExportBendLines = 4
    IncludeSketches = 8
    MergeCoplanarFaces = 16
    ExportLibraryFeatures = 32
    ExportFormingTools = 64
    ExportBoundingBox = 2048
End Enum

Const SKIP_EXISTING_FILES As Boolean = False

Const OUT_NAME_TEMPLATE As String = "DXFs\<$CLPRP:Description>.dxf"

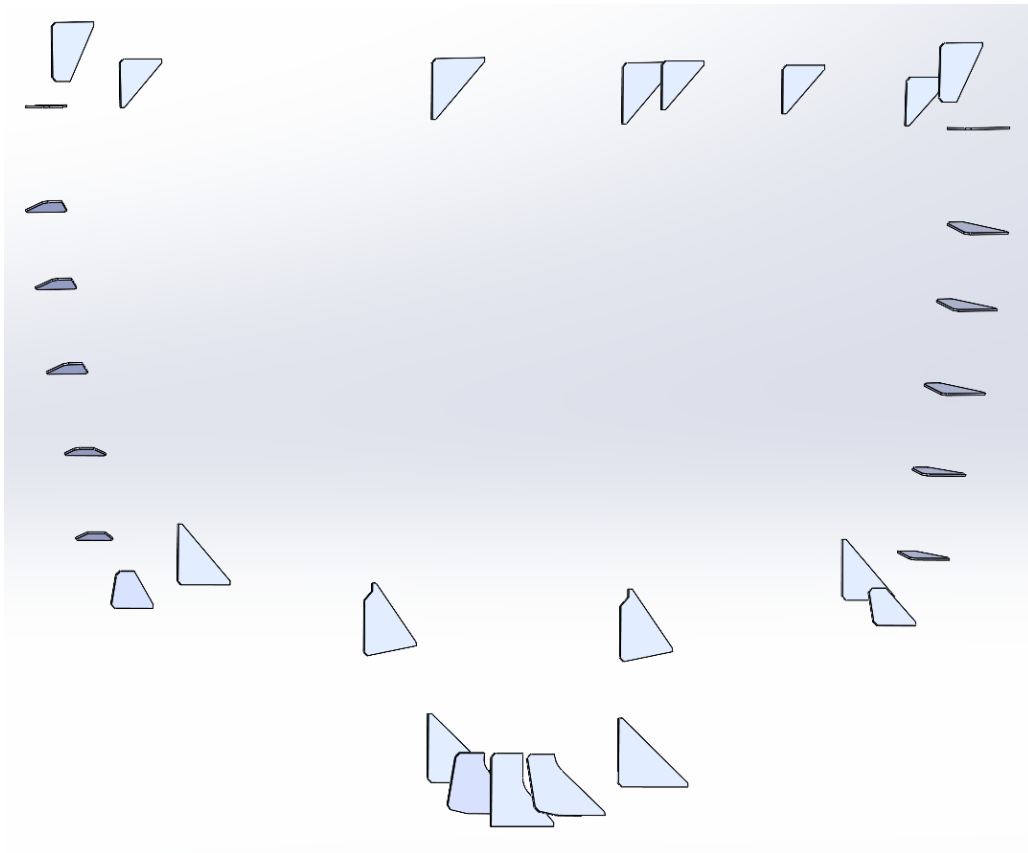
Const FLAT_PATTERN_OPTIONS As Integer = SheetMetalOptions_e.ExportBendLines + SheetMetalOptions_e.ExportFlatPatternGeometry
Dim swApp As SldWorks.SldWorks
```

Kuva 11. VBA koodi "Sheetmetal Options"

6.3.4 Makron tiedostonsiirtonopeudet

Tässä havainnollistetaan geometriatiedostojen luonti multibody mallista, johon on mallinnettu 24 polviota yhdestä kaaresta.

Usein lähes jokainen polvio on peilattu PS puolelta SB puolelle tiettyjä poikkeuksia lukuun ottamatta. Kuten kuvasta näemme polvioiden muoto vaihtelee jokaisessa pitkittäisjäykisteiden risteämiskohdassa (Kuva 12).



Kuva 12. Kaaren polviot

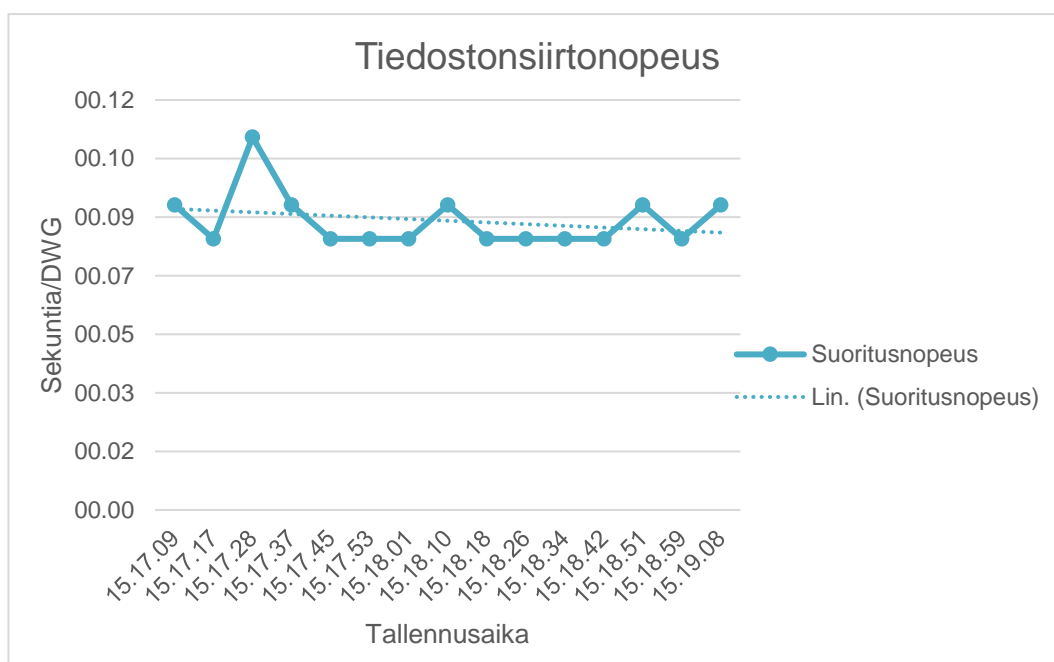
Makron tiedoston luonti ja ajo on toteutettu koneen C-aseamalla, koska verkon yli ajettaessa nopeudet voivat vaihdella paljon. (Taulukko 1).

Taulukko 1. Tiedostonsiirtonopeus

15:17:00	Polvio	Suoritusnopeus	Tallennusaika
	B1.0	00:09	15:17:09
	B1.1	00:08	15:17:17
	B1.2	00:11	15:17:28
	B1.3	00:09	15:17:37
	B1.4	00:08	15:17:45
	B1.5	00:08	15:17:53
	B1.6	00:08	15:18:01
	B1.7	00:09	15:18:10
	B1.8	00:08	15:18:18
	B1.9	00:08	15:18:26
	B1.10	00:08	15:18:34
	B1.11	00:08	15:18:42
	B1.12	00:09	15:18:51
	B1.13	00:08	15:18:59
	B1.14	00:09	15:19:08

Tiedostojen välisessä luonnissa ole juurikaan aikaeroja. Yksittäisen samanlaisen geometrian muuntaminen peräkkäin. (Taulukko 2.)

Taulukko 2. Tiedostonsiirtonopeus



6.4 Parametrit tiedostonimeen makron avulla




VBA Koodi tekee tiedostot automaattisesti järjestyksenä tiedostonimi tai cutlistan numerosarja. Geometriatiedostot tulevat joko omaan kansioon, jonka nimenä voidaan pitää DXF tai jokin muu. Aiemmin halutulla tavalla tiedostojen paikka voidaan myös määrittää esimerkiksi materiaalin paksuuden mukaan, tai esimerkiksi PDM-Ohjelmiston luoman M-koodin mukaan, riippuen sen käyttötarkoituksesta.

Solidworks mallit ja piirustukset ovat tietyssä kansiossa omalla koneella, aina kun ne tallennetaan varattuina PDM-ohjelmistossa, näin se osaa hakea ja tallentaa ne omaan järjestelmään.

Kun tiedostoja ei varata paikallisesti PDM ohjelmistossa, ne tallentuvat väliaikaiseen kansioon TEMP.

Esimerkki, jossa VBA-koodia (Liite 2) on muutettu niin, että se osaa hakea tiettyjä parametrejä suoraan tiedostonimeen. (Kuva 13).

- Cut-listan nimi
- Kappalemäärä
- Materiaalin paksuus
- Alumiinimateriaali

Nimi	Tila	Muokauspäivä
 M-000001-1_PCS 1_3mm 3.3547 (EN-AW 5083)		16.10.2023 8.20
 M-000001-2_PCS 1_6mm 3.0515 (EN-AW 3103)		16.10.2023 8.20
 M-000001-3_PCS 1_20mm 3.3545 (EN-AW 5086)		16.10.2023 8.20

Kuva 13. Parametrinen tiedostonimi

6.5 Kappaleen tunnisteiden muuttaminen geometriaksi

Tuotannossa ohutlevykappaleet ovat yksilöity merkitsemällä, eli jyrkimällä kappaleen pintaan tunniste, tämä on yleensä tehty suoraan DWG-tiedostoon käsin kirjoittamalla.

Metalliteollisuudessa yleinen menetelmä on myös esimerkiksi plasma tai laser leikkauksessa muste- tai lasermerkkaus.

Yksittäiset kappaleet, esimerkiksi juuri veneen jäykisteet ovat jokainen erilaisia. Tämän vuoksi niiden yksilöinti on erittäin tärkeää. Myös kokoonpanopiirustuksissa jokainen kappale saa oman paikkansa, jolloin kokoamisvaiheessa se voidaan paikantaa helpommin.

3D-Mallin polviot suunnitellaan yleensä multibody menetelmällä, jolloin esimerkiksi runkoon liittyvät kaarien polviot mallinnetaan yhtä aikaa. Jolloin niiden kokoaminen samasta mallista on myös helpompaa.

Seuraavassa osiossa tutkitaan kappaleiden yksilöimistä tekstin avulla.

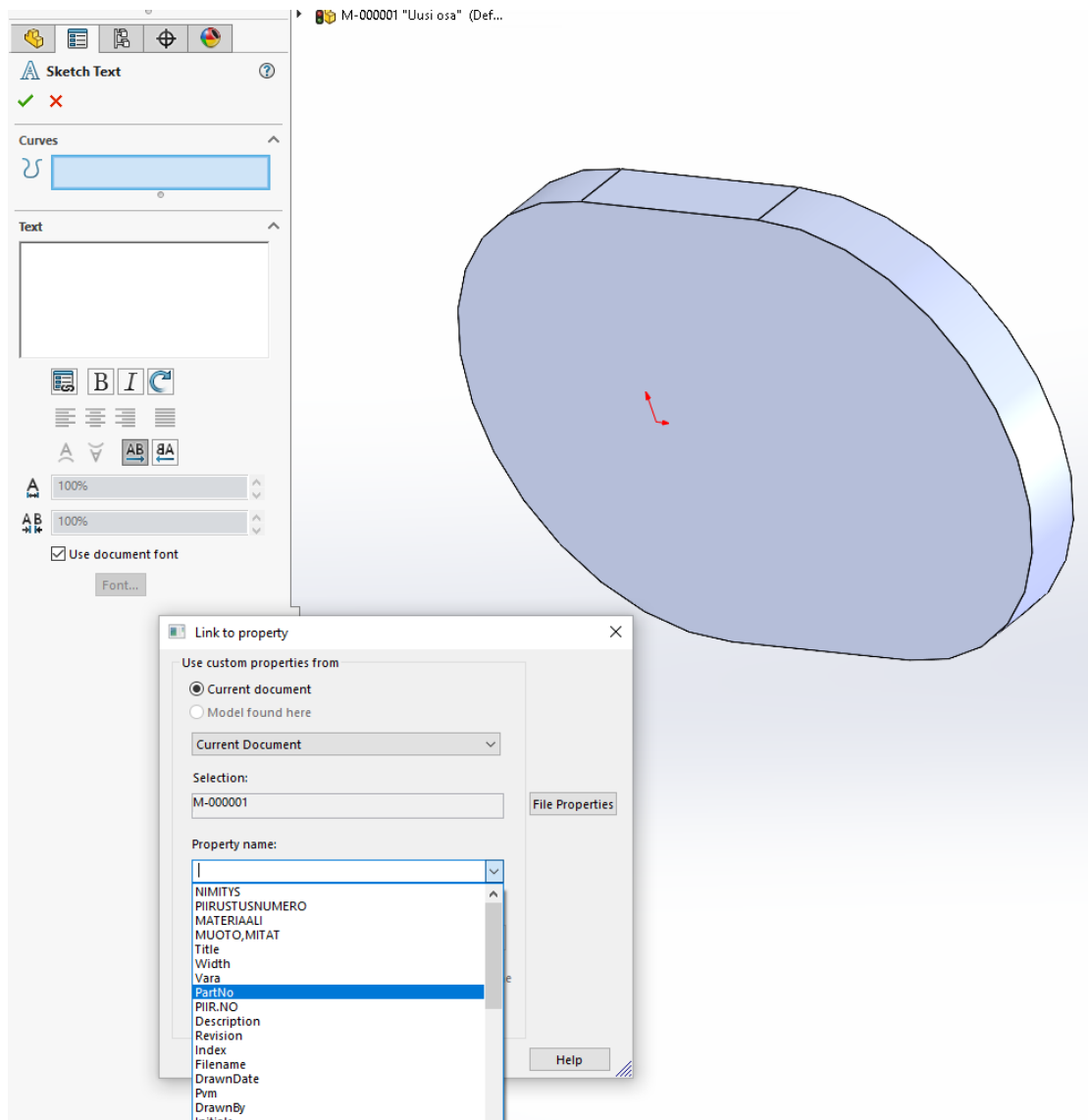
6.5.1 Solidworks 3D-tila

Yksittäisen geometriatiedoston menetelmässä tutkitaan myös tekstigeometrian tuomista erilaisilla vaihtoehdoilla.

Käytännössä 3D tilassa voi luonnostella tekstiä, jonka Property ominaisuutta voi käyttää päätasolla, ei esimerkiksi Cutlist Property valikkoa. Tällä hetkellä tämä ominaisuus ei toimi multibody tasolla, niin kuin haluaisimme tässä menetelmässä.

Ominaisuuden lisääminen suoraan 3D-tilaan voisi helpottaa juuri kappaleiden yksilöintiä ja merkitsemistä esimerkiksi jyrsimällä tai lasermerkkauksella.

Menetelmässä voidaan lisätä tuotetietoa luonnokseksi suoraan File Properties asetusten kautta. (Kuva 14).



Kuva 14. Property valikko 3D-tila

Tunnistetiedon tuominen 3D tilassa, esimerkiksi konfiguroimalla teksti kappalenimeksi tai tässä tapauksessa tiedostonimellä. (Kuva 15).



Kuva 15. Tunnistetieto mallissa

3D-Tilan hyödyntäminen käyttämällä PROPERTY asetuksia FLAT PATTERN konfigurointi välilehden kautta, jossa konfiguraation nimi yksilöi jokaisen ohutlevymallin ja tätä kautta tapahtuu osan nimeäminen. Tässä menetelmässä on se hyöty, että normaalitilassa konfiguraatio supressoi tekstin pois, jolloin mallista tulee kevyempi.

3D-tekstin luonnissa ei ole Custom Property ominaisuutta, niin kuin se löytyy esimerkiksi piirustusdokumentin NOTE-asetuksista.

Solidworks ohjelmiston kehitystyö etenee myös näiden parametrien osalta tulevaisuudessa ja parametrinen tekstitieto tulee olemaan tulevaisuudessa myös osa ohjelmiston uusia ominaisuuksia.

Suurien 3D-mallien yhteydessä tämän menetelmän käyttäminen lisää malliin myös enemmän luonnoksia, joka voi hidastaa mallin käytettävyyttä ja latausaikaa. Luonnosten näkyminen ja niiden piilottaminen on tärkeää, että ne eivät tule näkyviin kaikissa muissa kokoonpanoissa.

Suurten mallien lataaminen on myös haastavaa esimerkiksi PDM-ohjelmiston kanssa.

Solidworks ohjelmiston ominaisuuksia ei ole myöskään luotu käytettäväksi hyvin kompleksisten ja laajojen kokonaisuuksien hallitsemiseen, tämän vuoksi 3D-mallien kokoa ja erityisesti sen toimivuutta pitää osata myös arvioida uusien kehitysmenetelmien luomisessa.

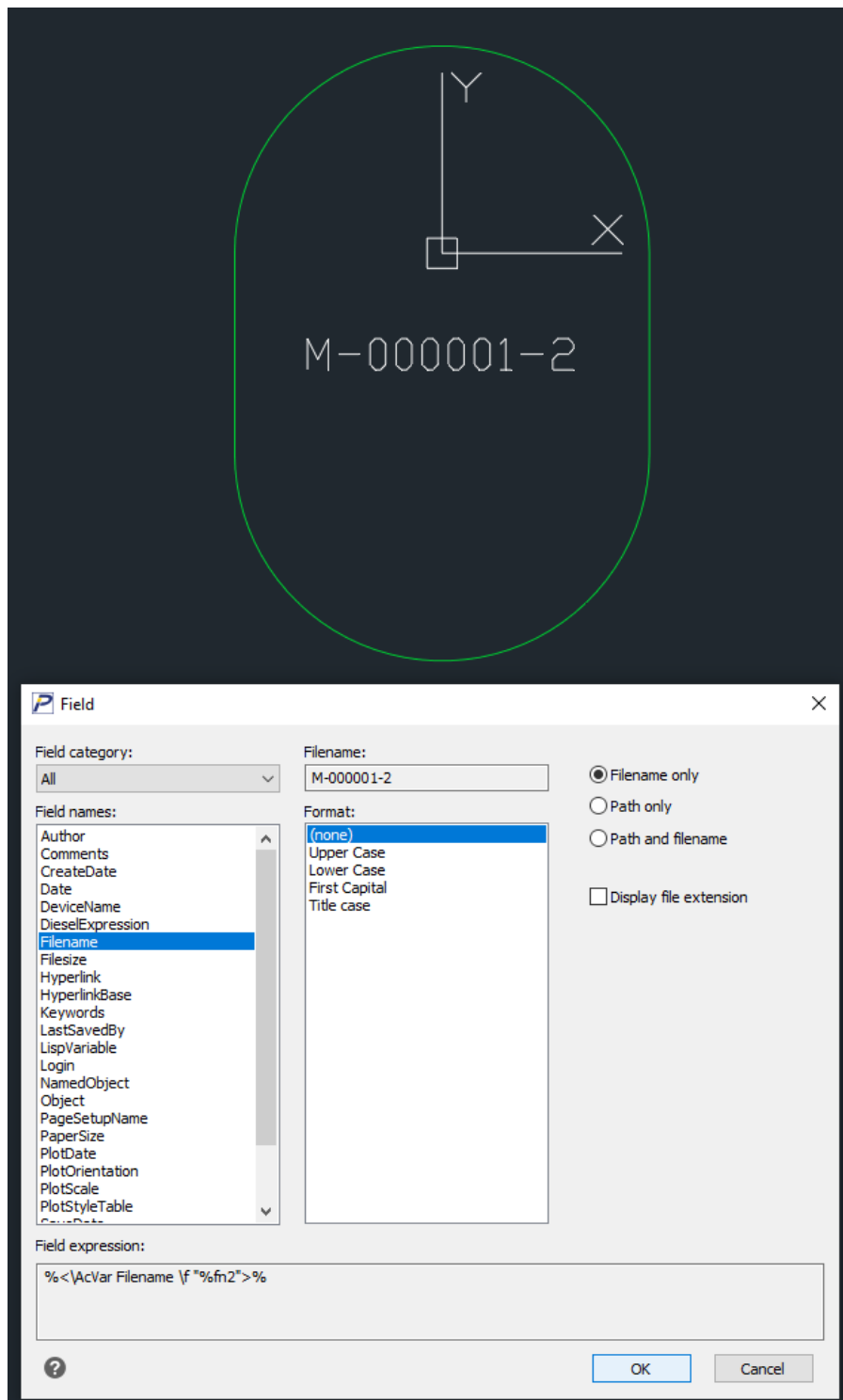
6.5.2 Progecad parametrinen tekstialue

Tunnistetiedon lisääminen geometria tiedostoon, jolla pystyy tuottamaan teksti geometriaa automaattisesti kappaleen sisälle. Progecad ymmärtää 3D-tiedoston sisäisiä objekteja, mutta tällä menetelmällä voidaan tuoda geometrian sisälle kentän, jolla esimerkiksi tiedostonimi voidaan muuttaa suoraan geometriaksi.

Jos tiedostonimi kappalekohtaisessa tuonnissa nimetään cut-listan mukaan, voisimme käyttää suoraan tiedostonimeä esimerkiksi yhtenä parametriseena tekstigeometriana cad-ohjelmiston sisällä.

Yleisesti tätä parametrista aluetta käytetään jo piirustustilassa, mutta näin saadaan tunnistetiedon myös työstörataan, jolloin kappaleen yksilöinti on mahdollista tätä kautta.

Progecad ymmärtää generoida parametrisiä tietoja, esimerkiksi tiedostonimen tai jonkin muun konfiguroitavan parametrin automaattisesti. (Kuva 16).

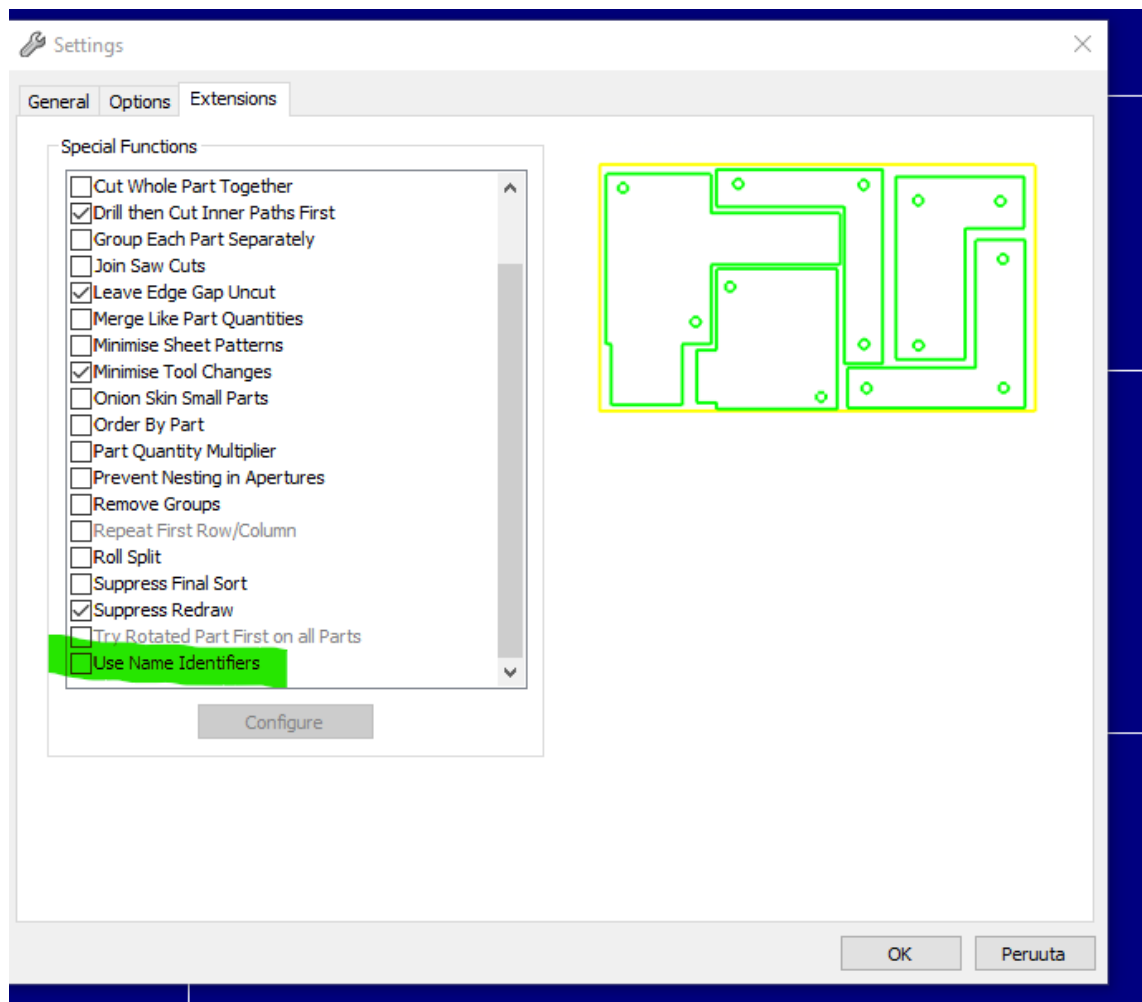


Kuva 16. Progecad tunnistetieto

6.5.3 AlphaCam tunnistetieto

Jyrsintäkoneen postprosessorin eli AlphaCam nestautietoja voidaan myös käyttää tunnistetiedon lisäämisessä. Yleisesti nestauksen aikana syntyy erilaisia työn kiertoja, joita voidaan lisätä esimerkiksi työstökarttaan ja nestausdokumenttiin. Yhtenä vaihtoehtona on lisätä tunnistetieto tässä vaiheessa tuotanto prosessia.

Postprosessorin asetuksia muuttamalla kappalekohtainen työstöjärjestys esitetään nestauksen yleisnäkymässä. Asetuksia muuttamalla voidaan nämä järjestysnumerot vaihtaa kappalenumeroiksi. (kuva 17).

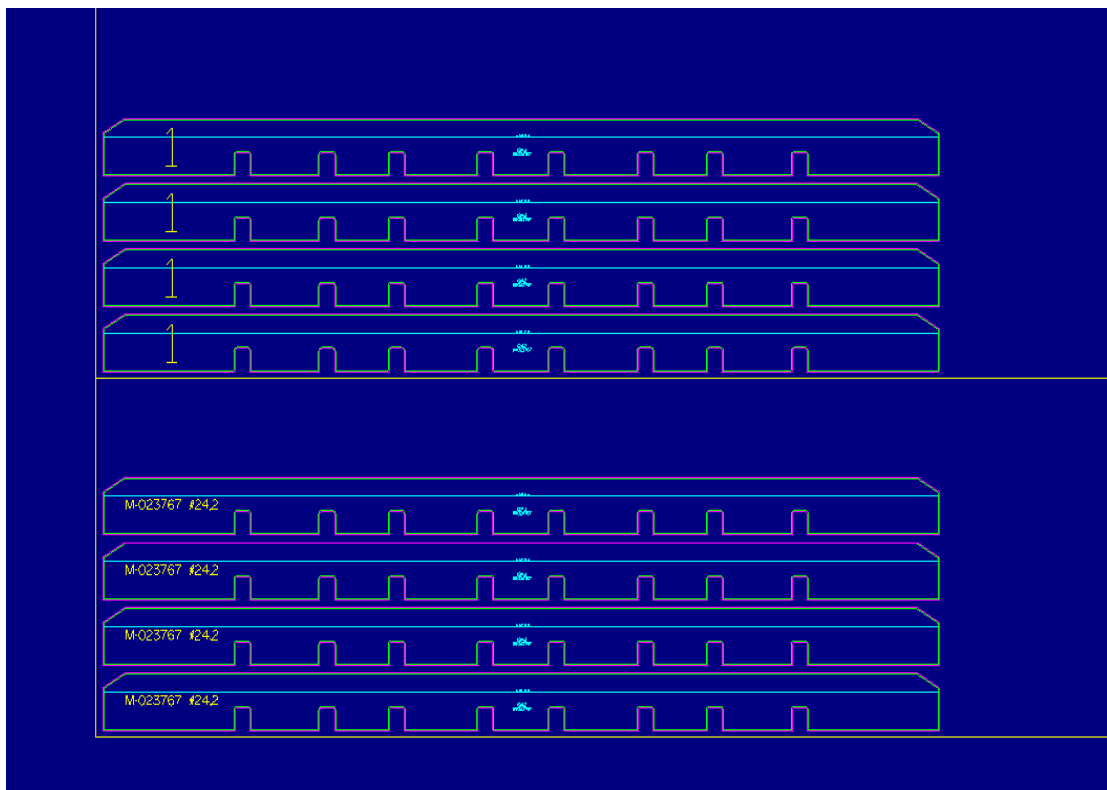


Kuva 17. ALPHACAM asetukset

ALPHACAM nestaus tilassa näkyy asetuksen vaikutukset merkintätapaan. (Kuva 18).

Osoitetaan normaali työnkierto ja järjestys ylhäällä olevissa kappaleissa.

Alemmissa jäykisteissä käytetään "Use Name Identifiers" asetusta, jolloin postproessori osaa käyttää kappaleen tunnistenumeroa geometriana. Tämä menetelmä rajaa pois manuaalista tietojen syöttämistä kappaleisiin. Asetuksen käyttäminen ja esimerkiksi oikeiden kirjasinkokojen muuttaminen asetuksista ei ole mahdollista, vaan se täytyy tehdä jokaiselle tekstinosalle erikseen käsin. Myös hyvin suurien ja kaarevien jäykisteiden merkintätapa tällä menetelmällä on ongelmallista, koska automatiikan takia tietokone osaa rajata alueen vain raamina, joka syntyy geometrian maksimimitoista.



Kuva 18. ALPHACAM nestaus tilan näyttö

7 CISS Office Ohjelmisto

7.1 Yleistä

CISS Officen tietokanta (CISSBASE) tarjoaa keskitetyn tiedon tallennuspaikan kaikille Ciss-ohjelmistotuotteillemme ja täten vaivattoman tiedon jakamisen kaikille osapuolille ajasta ja paikasta riippumatta. Kaikki CISS-ohjelmistoperheen tuotteet ovat samassa tietokannassa, joten kaikki sovellukset toimivat saumattomasti yhdessä. Hankkiessasi minkä tahansa CISS-ohjelmamme, käytössäsi ovat myös kaikki CISS Office -ohjelman ominaisuudet. Eri toimialojen raporttien luominen ja tiedostojen hallinta on helppoa, kun kaikki tieto on yhdessä ja samassa tietokannassa.

CISS Officella hallitset kaikki organisaation dokumentit yhdellä ja samalla ohjelmalla – Office-tiedostot, sähköpostit, kuvat, videot, tekniset dokumentit (esim. CAD-tiedostot) jne. Kaikkiin dokumentteihin liittyvät informaatiot, metatiedot, eri versiot, työnkulut, käyttäjien oikeudet jne. ovat hallittavissa CISS Officella.

CISS Office varmistaa automaattisesti, että vain uusin versio tiedostosta on saatavilla. Esimerkiksi mekaniikkasuunnitteluyritys voi säästää vuosittain merkittävästi kustannuksissa, jotka aiheutuvat väärän tiedostoversion käytöstä. Virheiden määrä pienenee, kun käytössä on aina viimeisin versio CAD-piirustuksesta. Ilmoitus uudesta keskeneräisestä tiedostoversiosta välittyy kaikille käyttäjille heti.

Metatietojen avulla pystytään säästämään merkittävä määrä aikaa tiedostojen käsittelyssä. Virheiden määrä pienenee, kun metatietoa hyödynnetään dokumentin tietojen lisäämisessä ja päivityksessä. Esimerkiksi yrityksen nimi ja yhteystiedot (metatietoja), voidaan vakiodia päivittymään aina samaan paikkaan Word-tiedostossa ja CAD-piirustuksen otsikkotaulussa.

CISS Office -tietojärjestelmän sisällä dokumentit voivat kommunikoida käyttäjien kanssa. Esimerkiksi dokumentille voidaan määrittää muistutus, jolla on tarkka

ajankohta, jolloin se ilmoittaa valitulle henkilölle sopimuksen uusimisen tarpeellisuudesta. Pilvessä tai omalla palvelimella oleva älykäs objekti tekee asioita käskystäsi ja silloinkin, kun et itse ole paikalla.

Jokaiselle CISS Officessa olevalla tiedostolla on oikeudet ja turvaluokitus. Käyttäjälle voidaan määrittää oikeus dokumentin käsittelyyn (luku, tulostus, muokkaus ja poisto). Turvaluokka määrittelee, onko käyttäjällä oikeus nähdä turvaluokitettuja dokumentteja. Turvaluokat ovat: erittäin salainen (I), salainen (II), luottamuksellinen (III), käyttö rajoitettu (IV) ja julkinen (ei luokiteltu).

CISS Officella saa voidaan seurata jokaisen dokumentin tilaa (työnkulkua) visuaalisesti. Jokaisessa vaiheessa voidaan valita tietty henkilö, joka tarkastaa tai hyväksyy kyseisen vaiheen. Dokumentin minimityönkulku on kesken – valmis. Esimerkiksi piirustuksilla työnkulku voi olla suunnittelu – tarkastus – hyväksyntä.

Haun voi tehdä millä tahansa yksilöivällä termillä. Tiedoston metatietojen lisäksi voit hakea tekstiä myös tiedoston sisältä. Ohjelma hallitsee kaikki tiedostomuodot – Office-tiedostot, valokuvat, videot, CAD-ohjelmien käyttämät tiedostot jne.

CISS Officen käytön oppii nopeasti, joten ohjelma soveltuu erinomaisesti myös pienille yrityksille, joilla ei ole omaa IT-tukiosastoa. Yrityksestä saa tukipalvelut, ylläpitosopimuksen ja käyttöönottokoulutuksen kaikkiin ohjelmistoihin.

www.ciss.fi/tiedonhallinta/ciss-office

CISSBASE tarjoaa käyttäjilleen erilaisia ohjelmistopaketteja, joita voidaan hyödyntää suunnittelun eri vaiheissa. (Taulukko 3).



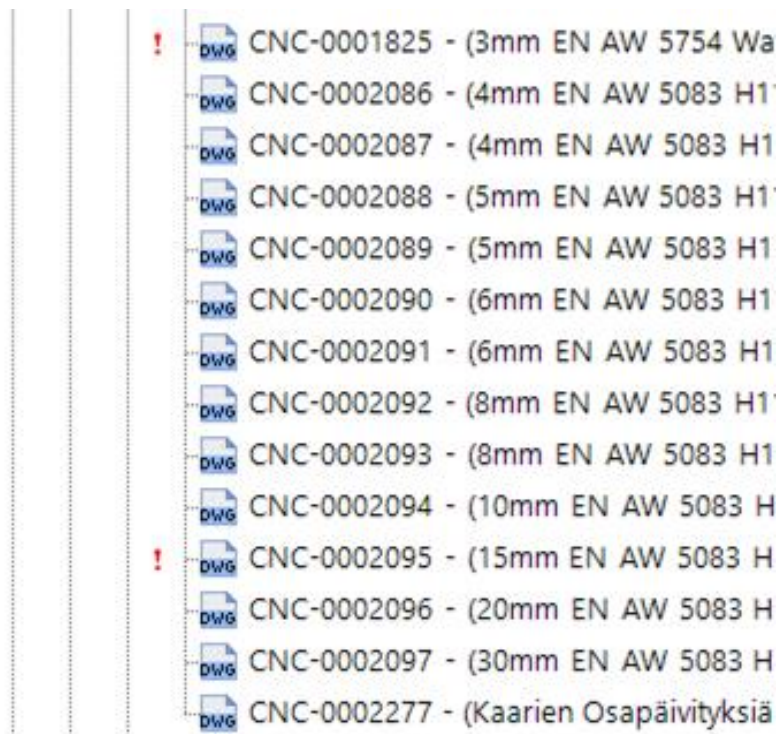
Taulukko 3. CISSPACE Ohjelmistot

7.2 Geometria tiedostojen käyttö PDM rakenteessa

Leikkausgeometrioiden tallennus vaihtelee eri projektien välillä ja niiden yhtenäistämiseen tarvitaan suunnitelma, joka olisi toimiva omassa tuotannossa, sekä myös alihankinnan käyttämistä ajatellen.

Projektiesimerkki aikaisemmasta geometriatiedostojen rakenteesta alihankintaan. CISSBASE tiedostorakenne runkosuunnittelussa.

Rakenteesta huomataan, kuinka voidaan yksilöidä kappaleet tietyllä materiaalin paksuudella, mutta tarkemmin yksittäisen kappaleen geometrian löytämiseen tarvitaan isompi työmäärä. (Kuva 19).



Kuva 19. Geometritiedostot rakenteessa

Multibody-mallien takia yksittäistä geometriatiedostoa on vaikea nimetä samalla revisiotunnuksella kuin piirustus, siksi yksittäisestä polviosta tai muusta runkojäykisteestä ei ole erillistä valmistuspiirustusta.

Geometriaryhmät luodaan materiaalipaksuuden mukaan tiedostoihin ja revisioidaan uudella numerolla esimerkiksi [1]. Uudelleen revisioitu numero muuttaa tiedoston keskeneräiseksi, jolloin esimerkiksi tuotanto-osastoa indikoidaan, milloin uusi geometria on lisätty kansioon. (Kuva 21).

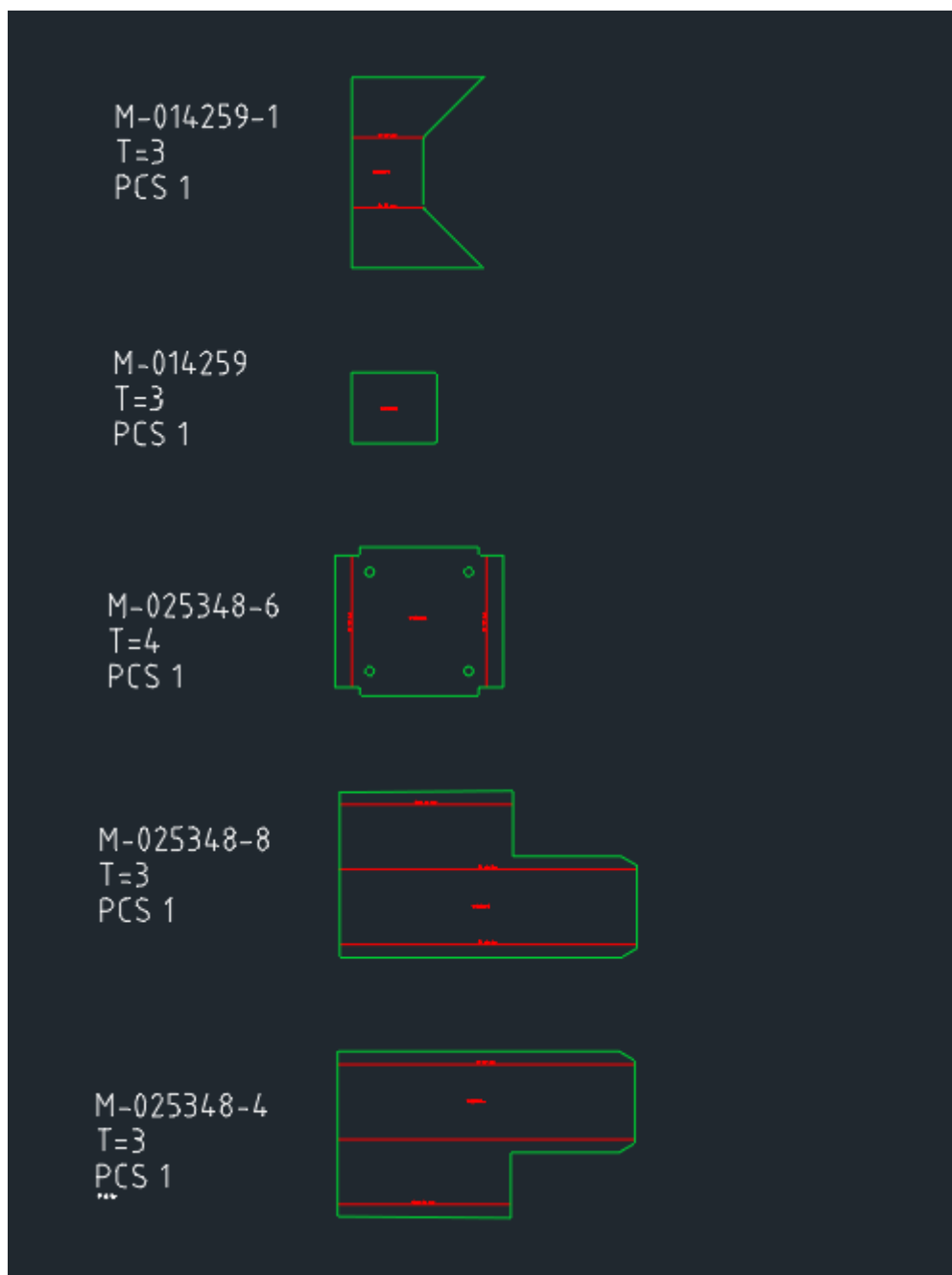
Geometriatiedoston revisiointi on järkevää siinä silloin, kun dokumentti seuraa tunnusta. Sillä voidaan osoittaa esimerkiksi uusien muutosten tai tiedoston tallennus kaikille. PDM-järjestelmän ominaisuus on, että kun tiedosto merkitään valmiiksi, sitä voidaan muokata vain, tekemällä uusi revisio.

Jokaisella kappaleryhmällä on myös erikseen oma CNC-leikkuutiedosto, jonka suunnittelija tekee aluksi suunnittelurakenteeseen, tämän jälkeen piirustuksesta voidaan tallentaa DWG.

Piirustus pohjan kautta tuotaessa on oltava varma, että jokainen geometria on oikeassa mittasuhteessa 1:1, tämä ei aina toteudu. Tämän vuoksi SW:ssä jokainen yhden sivun kappalegeometrioiden joukko täytyy tehdä samalla mittakaavalla, jotta siirto onnistuu oikein.

Kappaleryhmien tallennus samaan DWG-tiedostoon ja suurien kokonaisuuksien hallinta vaatii isoa työmäärää manuaalisesti.

Esimerkissä kappaleryhmä (Kuva 20).



Kuva 20. DWG-tiedoston kappaleryhmä

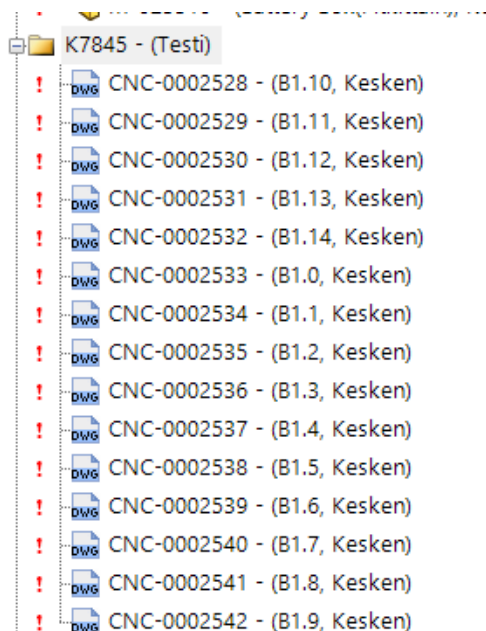
7.3 Geometriatiedostojen tuonti PDM-rakenteeseen

Uudessa menetelmässä luodut tiedostot voidaan tuoda ryhmänä PDM-kantaan "raahaamalla". Ohjelmisto tekee jokaiselle objektille oman tunnuksen.

PDM-Järjestelmä luo aina uuden tunnisteiden dokumentin tai objektin tuonnin yhteydessä. Tuotannon kannalta DWG-tiedostoissa tunnistetietona riittää katkaisulistat numero.

Alkuperäinen Cutlist numero säilyy tunnuksen perässä (Nimityksenä). Menetelmän ideana oli kuitenkin tehdä dokumentille ainut tunniste, joka olisi alkuperäinen Cutlist numero, tämä ei kuitenkaan ole mahdollista ainakaan vielä ohjelmistokehittäjän tietojen mukaan. (Kuva 21).

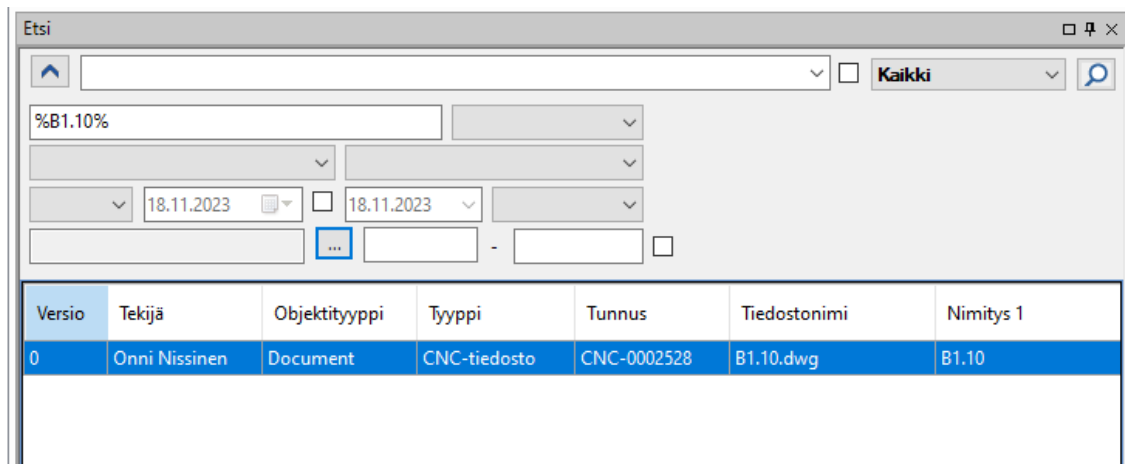
Suunnittelussa yleisesti piirustusdokumentaatiolla olisi hyvä ohjata yhden geometrian nimikettä itsenäisesti, ei yleisesti kokoonpanon useita kappaleita kerrallaan. Joissakin kokoonpanoissa, joissa on pelkästään leikattavaa tai suoria kappaleita, voidaan kuitenkin luoda dokumentteja kappaleryhmälle. Näissä tapauksissa kappaleille ei erikseen muodostu valmistuspiirustuksia.



Kuva 21. Tiedoston tuonti

7.4 Tiedostojen haku

Hakukoneella geometriatiedoston hakeminen tapahtuu etsi-kentän kautta. Yksittäisen kappaleen geometria voidaan löytää nopeasti, kunhan haku tehdään prosentti merkein, hakuohjelma listaa tarvittaessa alapuolelle siihen liittyvän samannimisen 3D-mallin sekä piirustusdokumentin. Tässä tapauksessa testaamme vain geometriatiedoston hakua luodulla nimikkeellä. (Kuva 22).



Kuva 22. Hakutoiminto CISSBASE

CISSBASE hakutoiminnolla tulisi olla mahdollisuus päästä suoraan yksittäisen ohutlevygeometriaan käsiksi. PDM-ohjelmiston hakuohjelma on erittäin tehokas työkalu oikein käytettynä. Tarvittaessa myös yksittäisen osan päivittäminen ja muutokset onnistuvat helposti.

7.5 Kehitysideat CISSBASE ohjelmistoon

Yksi kehitystyön alla oleva idea on, että ohjelmisto osaa hakea automaattisesti työkansiosta M-koodin alla olevien saman alkuisten nimien mukaiset DWG tiedostot, kuten järjestelmä jo nyt hyödyntää tiedostonimen tunnistusta.

Ohjelmiston rakenteessa on käytössä ominaisuus, joka osaa esimerkiksi hakea uuden tiedostomuodon rakenteeseen, mistä on tallennettu samanniminen, esimerkiksi piirustus. Tällä tavalla PDF tai DWG tallennetaan asennus- tai valmistusdokumentin alle, jolloin sitä on helppo tarkastella.

Ohjelmistokehittäjän ja tämän opinnäytetyön tuloksena syntyi myös uusi menetelmä, jossa rakenteessa siirretään myös yhtä aikaa kaikki DWG-tiedostot, joissa on sama tunnus alussa kuin päätiedostolla. Näin voidaan toteuttaa yhden kappaleen geometriatiedostojen luontia tiedostonimen tunnistamisella.

Toiminto on sen vuoksi tärkeä, että suunnittelussa ei tarvitse siirtää yksittäisiä tiedostoja suunnittelupuun rakenteeseen, vaan toiminto valikoi automaattisesti oikeat geometriat työkansion alta, mihin Solidworks asetuksissa tallennus tapahtuu.

Nämä uudet ominaisuudet helpottavat suunnittelutyötä tulevaisuudessa ja lisäävät rakenteen tekemisessä automatiikkaa, sekä vähentävät myös useiden tiedostojen nimeämistä manuaalisesti.

Opinnäytetyön tuloksena ohjelmistokehitystä voidaan viedä siihen suuntaan, että PDM-järjestelmä osaa hyödyntää automaattista M-koodin tunnistusta myös useiden kappaletiedostojen osalta. Tämä uusi menetelmä on erittäin hyödyllinen ja helpottaa geometriatiedostojen hallintaa.

8 Yhteenveto

Tuotannon ja yrityksen suunnitteluosaston välillä on katsottu olevan välttämätön ja saumaton yhteistyö läpi tuotteen tai projektin koko elinkaaren ajan. Omista lähtökohdista olen ollut erilaisten teknologiayritysten palveluksessa enemmän tuotannon puolella ja sitäkin vähemmän suunnitteluosastolla. Tämä uusi näkökulma suunnittelutyön myötä on ollut huomata, miten monenlaisia ja yksilöllisiä yrityksen omiin tarpeisiin luodut rakenteet ovat.

Erittäin merkityksellistä voi olla esimerkiksi erittäin hyvin suunniteltu tuotannonohjausjärjestelmä tai jokin muu integroitu tapa tuotannon ja suunnittelun välillä, unohtamatta osto-osastoa ja tämän organisointia osaksi tätä ketjua. Tämä on erityisen huomioitavaa silloin, kun yritys käyttää apunaan esimerkiksi alihankintaa ja alihankintasopimuksia, tämän takia esimerkiksi hyvä dokumentointi on erittäin tärkeää. Toisaalta alihankintaan liittyy myös paljon sellaista, jonka vuoksi hyvin tiivis yhteistyö voi olla hankalaa, esimerkiksi liikesalaisuuksien takia. Dokumentointi tässä mielessä pitää olla erittäin selkeää ja mahdollisimman yksinkertaista.

Suunnittelun dokumentointi yleisesti on järkevää pitää myös yrityksen sisällä kansainvälisenä pääkielenä jo senkin takia, että samoja dokumentteja tai muita tiedostoja voidaan käyttää myöhemmin esimerkiksi alihankinnassa. Myös geometriatiedostojen yksinkertaistaminen ja niiden yksilöinti on havaittu tehokkaaksi menetelmäksi monessa metallialan yrityksessä.

Tuotantoon siirrettävien tiedostojen hallinta automaattisen siirron avulla lisää tehokkuutta ja laadunhallintaa. Tämän vuoksi dataan ei tarvitse tehdä suuria muutoksia käsin, jolloin virheiden määrää voidaan minimoida.

Tiedostoihin ei tarvitse lisätä tuotannon kannalta oleellista valmistustietoa, vaan tuotannossa täytyy olla revisioitu ja ajantasainen dokumentointi, eli piirustus.

Menetelmäkehityksen ideana on myös yksilöidä jokainen ohutlevyosa ja tehdä siitä näin ollen tunnistettava, myös tiedostotasolla. Myöhemmin käytettävä, muokattava, tai poistettava geometria on helposti saatavilla ja päivitettävissä.

Opinnäytetyön tarkoitus on tuoda yrityksen runkosuunnittelun osalta erilaista tiedostojensiirtomenetelmää makrojen avulla, jolla päästään erittäin merkittäviin tuloksiin tehokkuudessa ja etenkin suorien ja yksikertaisten kappaleiden siirtämisessä valmistusprosessiin nopealla menetelmällä.

Postprosessori on yleensä tarkoitettu käsittelemään kappaleen sijoittelua etenkin nestauksen osalta tiedosto/yksittäinen geometria menetelmällä, jolloin tämäkin tukee tätä menetelmäkehittämistä. Alihankinnassa omat postprosessori ohjelmat luovat tilanteen, jossa yksittäinen geometria täytyy olla mahdollisimman yksinkertainen ja valmistukseen tai tuotetietoon perustuva lisätieto geometriatiedoston sisällä voi jopa hankaloittaa osien sijoittelua tai yleisesti nestausta.

Uusien projektien myötä voimme hyödyntää VBA-koodaamista, jos tuotannossa siirrytään yhden geometrian menetelmään. Nykyisellään tämä kehitys sopii eniten yksinkertaisten kappaleiden, kuten polvioiden tai jäykisteiden siirtämiseen tuotantoon. Laajemmin menetelmän kannatus tuotannossa vaatii myös siellä parempaa piirustusdokumentaation käyttämistä tulevaisuudessa.

PDM-rakenteen puolella tiedostojen muodostaminen useiden tiedostojen siirtämisessä kehitty koko ajan. Näin ollen myös suora geometriatiedostojen siirto ja automatisointi tulee olemaan tulevaisuudessa helpompaa. Koko ajan kasvava ja nopeutuva massamuistimäärä edistää myös kehitystä. Tuotantoon siirrettävässä datassa voimme käyttää myös toista vaihtoehtoa, jossa geometriatiedostojen arkistointi ilman PDM-järjestelmää voisi olla myös joissakin tapauksissa yksi tapa.

Mapping tiedoston käyttö ei ole yleisesti tiedossa suunnittelun sisällä, mutta geometriatiedoston muokkaaminen käsin vähenee asetuksen ansiosta merkittävästi. Asetuksien ylläpitäminen ja yhdenmukaistaminen osaston sisällä voisi olla yksi vaihtoehto, esimerkiksi niin, että piirustusohjelmien tapaan myös nämä tiedostoasetukset voisivat olla kaikkien saatavilla.

Jatkosuunnitelmissa olisi myös edelleen kehittää yksittäisen kappaleen tunnistamiseen tarvittavaa merkintätapaa ja sen automatisointia, niin

ohjelmistovalmistajien kehitystyön tuloksena, kuin myös omassa suunnittelutyön kehittämisessä. Solidworks ohjelmiston kehittäminen, esimerkiksi mittatiedon, kuten särmäys mittojen ja asetusten osalta 3D-mallista geometriatiedostoon suoraan tallentamista esimerkiksi tekstigeometriaksi tai objektiksi. Tämä ominaisuus on käytössä nykyisin vain piirustus pohjan kautta tallennettaessa.

Myös alumiinin työstämiseen, leikkaamiseen ja merkintään tarkoitettuja menetelmiä on monia, joiden kehittäminen tämän työn rinnalla voisi olla yksi kehityskohde kappaleen yksilöimisessä.

Laajenevan tuotannon tarpeeseen PDM-ohjelmiston rinnalle tarkoitettuja tuotannonsuunnittelualustoja pitää myös kehittää eteenpäin. Tämä helpottaa esimerkiksi suurempien kokonaisuuksien aikataulutusta ja tuotannon kyvykkyyden tutkimista tulevaisuudessa.

Osavalmistuksessa suunnittelun tuottamat geometriatiedostot ja yleiset dokumentit voidaan tuoda osaksi edellä mainittuja tuotannonohjaus tai suunnittelujärjestelmiä ja sen myötä valmistusaikataulua. Tuotteiden seuraaminen ja laajemman kokonaisuuden hahmotus ja tiedon tuottaminen ovat näin ollen paremmin nähtävissä myös yrityksen muissa elimissä.

Kiitokset:

Lopuksi haluan kiittää tämän opinnäytetyömahdollisuuden tekemisestä ja kiitokset hyvästä yhteistyöstä:

Marine Alutech Oy Ab:

Toimitusjohtaja Niko Haro

Runkosuunnittelija Esko Loukkola

IT-tuki Joonas Panu

Tuotannon alumiiniosaston vetäjä Juha Iivarinen

Muut yhteistyökumppanit:

PLM Group Oy

Niklas Johansson (PLM Group Customer Support)

CISSBASE ohjelmistokehitys yhteistyöhenkilö

Jari Maunu

CISS Software Oy

LÄHTEET

Kirjallisuuslähteet:

TkT Esa Hietikko, 2016 Solidworks - Tietokoneavusteinen suunnittelu. 7. painos. Books on Demand, Helsinki, Suomi.

Internet:

Hexagon AB 2023. Luettu 07.09.2023.

<https://hexagon.com/products/product-groups/computer-aided-manufacturing-cad-cam-software/alphacam>

CISS Office. CISS Software 2023. Luettu 15.10.2023.

<https://www.ciss.fi/tiedonhallinta/ciss-office/>

Export flat patterns from SOLIDWORKS part or assembly components. 2023 Xarial Pty Limited. Luettu 07.08.2023.

<https://www.codestack.net/solidworks-api/document/sheet-metal/export-all-flat-patterns/>

Macro Library for SOLIDWORKS Automation. 2023 Xarial Pty Limited. Luettu 15.9.2023

<https://www.codestack.net/solidworks-tools/>

2020 SolidWorks. Mapping Entities When Saving Drawings as .DXF or .DWG Files. 2023 Xarial Pty Limited. Luettu 23.08.2023.

https://help.solidworks.com/2020/english/SolidWorks/slddxf/t_map_entities_on_export_to_dwg.htm?format=P&value=

Marine Alutech.com. Luettu 03.09.2023.

<https://www.marinealutech.com/about-us/>

Macro to rename SOLIDWORKS bodies of cut-list items using the predefined template. 2023 Xarial Pty Limited. Luettu 12.08.2023

<https://www.codestack.net/solidworks-api/document/cut-lists/rename-cut-list-bodies/>

Solidworks. David E. Weisberg Solidworks (PDF) cadhistory.net. 7.3.2022 (englanniksi). Luettu 04.09.2023.

<https://fi.wikipedia.org/w/index.php?title=SolidWorks&veaction=edit§ion=2>

Visual Basic. Isanovic S. VBA-ohjelmointi Office 2000:ssa, s. 5. Schildts Kustannus Oy, 2000. Luettu 11.09.2023.

https://fi.wikipedia.org/wiki/Visual_Basic

1 Macro to rename SOLIDWORKS bodies of cut-list items using the predefined template.

```

*****
'Copyright(C) 2023 Xarial Pty Limited
'Reference: https://www.codestack.net/solidworks-api/document/cut-lists/rename-cut-list-bodies/
'License: https://www.codestack.net/license/
*****

Const NAME_TEMPLATE As String = "<PartNo>"

Dim swApp As SldWorks.SldWorks

Sub main()

    Set swApp = Application.SldWorks

    Dim swPart As SldWorks.PartDoc

    Set swPart = swApp.ActiveDoc

    ProcessCutLists swPart

End Sub

Sub ProcessCutLists(model As SldWorks.ModelDoc2)

    Dim swFeat As SldWorks.Feature

    Set swFeat = model.FirstFeature

    Do While Not swFeat Is Nothing

        Dim swBodyFolder As SldWorks.BodyFolder

        If swFeat.GetTypeName2() = "SolidBodyFolder" Then
            Set swBodyFolder = swFeat.GetSpecificFeature2
            swBodyFolder.UpdateCutList
        ElseIf swFeat.GetTypeName2() = "CutListFolder" Then
            Set swBodyFolder = swFeat.GetSpecificFeature2

            Dim name As String
            name = ComposeName(NAME_TEMPLATE, swFeat)

            RenameBodies swBodyFolder.GetBodies(), name

        End If

        Set swFeat = swFeat.GetNextFeature

    Loop

End Sub

Sub RenameBodies(bodies As Variant, bodyName As String)

    If Not IsEmpty(bodies) Then

        Dim i As Integer

        For i = 0 To UBound(bodies)
            Dim swBody As SldWorks.Body2

```

```

        Set swBody = bodies(i)

        swBody.name = bodyName & If(i > 0, "-" & CStr(i), "")
    Next

End If

End Sub

Function ComposeName(template As String, cutListFeat As SldWorks.Feature) As String

    Dim regEx As Object
    Set regEx = CreateObject("VBScript.RegExp")

    regEx.Global = True
    regEx.IgnoreCase = True
    regEx.Pattern = "<[^>]*>"

    Dim regExMatches As Object
    Set regExMatches = regEx.Execute(template)

    Dim i As Integer

    Dim outName As String
    outName = template

    For i = regExMatches.Count - 1 To 0 Step -1

        Dim regExMatch As Object
        Set regExMatch = regExMatches.Item(i)

        Dim prpName As String
        prpName = Mid(regExMatch.Value, 2, Len(regExMatch.Value) - 2)

        outName = Left(outName, regExMatch.FirstIndex) & GetPropertyValue(cutListFeat.CustomPropertyManager,
prpName) & Right(outName, Len(outName) - (regExMatch.FirstIndex + regExMatch.Length))

    Next

    ComposeName = outName

End Function

Function GetPropertyValue(custPrpMgr As SldWorks.CustomPropertyManager, prpName As String) As String

    Dim resVal As String

    custPrpMgr.Get2 prpName, "", resVal

    GetPropertyValue = resVal

End Function

```

Export flat patterns from SOLIDWORKS part or assembly components

```

*****
'Copyright(C) 2023 Xarial Pty Limited
'Reference: https://www.codestack.net/solidworks-api/document/sheet-metal/export-all-flat-patterns/
'License: https://www.codestack.net/license/
*****

Enum SheetMetalOptions_e
ExportFlatPatternGeometry = 1
IncludeHiddenEdges = 2
ExportBendLines = 4
IncludeSketches = 8
MergeCoplanarFaces = 16
ExportLibraryFeatures = 32
ExportFormingTools = 64
ExportBoundingBox = 2048
End Enum

Const SKIP_EXISTING_FILES As Boolean = False

Const OUT_NAME_TEMPLATE As String =
"DXFs\<_FileName_>\<_FeatureName_>\<_ConfName_>\<_CLPRP:Description>.dxf"

Const FLAT_PATTERN_OPTIONS As Integer = SheetMetalOptions_e.ExportBendLines +
SheetMetalOptions_e.ExportFlatPatternGeometry

Dim swApp As SldWorks.SldWorks

Sub main()

Set swApp = Application.SldWorks

try_:
On Error GoTo catch_

Dim swModel As SldWorks.ModelDoc2
Set swModel = swApp.ActiveDoc

If swModel Is Nothing Then
Err.Raise vbError, "", "Please open assembly or part document"
End If

If swModel.GetType() = swDocumentTypes_e.swDocASSEMBLY Then

Dim swAssy As SldWorks.AssemblyDoc

Set swAssy = swModel

swAssy.ResolveAllLightWeightComponents True

Dim vComps As Variant
vComps = GetDistinctSheetMetalComponents(swAssy)

Dim i As Integer

For i = 0 To UBound(vComps)

Dim swComp As SldWorks.Component2
Set swComp = vComps(i)

ProcessSheetMetalModel swAssy, swComp.GetModelDoc2(), swComp.ReferencedConfiguration

Next

ElseIf swModel.GetType() = swDocumentTypes_e.swDocPART Then

Dim swPart As SldWorks.PartDoc

```

```

Set swPart = swApp.ActiveDoc

ProcessSheetMetalModel swPart, swPart, swPart.ConfigurationManager.ActiveConfiguration.Name

Else
Err.Raise vbError, "", "Only assembly and part documents are supported"
End If

swApp.SendMsgToUser2 "Operation completed", swMessageBoxIcon_e.swMbInformation,
swMessageBoxBtn_e.swMbOk

GoTo finally_

catch_:
swApp.SendMsgToUser2 Err.Description, swMessageBoxIcon_e.swMbStop, swMessageBoxBtn_e.swMbOk
finally_:

End Sub

Function GetDistinctSheetMetalComponents(assy As SldWorks.AssemblyDoc) As Variant

Dim vComps As Variant
vComps = assy.GetComponents(False)

Dim i As Integer

Dim swSheetMetalComps() As SldWorks.Component2

For i = 0 To UBound(vComps)

Dim swComp As SldWorks.Component2
Set swComp = vComps(i)

If False = swComp.IsSuppressed() Then

If Not ContainsComponent(swSheetMetalComps, swComp) Then

If IsSheetMetalComponent(swComp) Then
If (Not swSheetMetalComps) = -1 Then
ReDim swSheetMetalComps(0)
Else
ReDim Preserve swSheetMetalComps(UBound(swSheetMetalComps) + 1)
End If

Set swSheetMetalComps(UBound(swSheetMetalComps)) = swComp
End If

End If

End If

Next

If (Not swSheetMetalComps) = -1 Then
GetDistinctSheetMetalComponents = Empty
Else
GetDistinctSheetMetalComponents = swSheetMetalComps
End If

End Function

Function IsSheetMetalComponent(comp As SldWorks.Component2) As Boolean

Dim vBodies As Variant
vBodies = comp.GetBodies3(swBodyType_e.swSolidBody, Empty)

If Not IsEmpty(vBodies) Then

Dim i As Integer

For i = 0 To UBound(vBodies)
Dim swBody As SldWorks.Body2
Set swBody = vBodies(i)

```

```

If False <> swBody.IsSheetMetal() Then
IsSheetMetalComponent = True
Exit Function
End If

Next
End If

IsSheetMetalComponent = False

End Function

Function ContainsComponent(comps As Variant, swComp As SldWorks.Component2) As Boolean

Dim i As Integer

For i = 0 To UBound(comps)
Dim swThisComp As SldWorks.Component2
Set swThisComp = comps(i)

If swThisComp.GetPathName() = swComp.GetPathName() And swThisComp.ReferencedConfiguration =
swComp.ReferencedConfiguration Then
ContainsComponent = True
Exit Function
End If
Next

ContainsComponent = False

End Function

Function ComposeOutFileName(template As String, rootModel As SldWorks.ModelDoc2, sheetMetalModel As
SldWorks.ModelDoc2, conf As String, flatPatternFeat As SldWorks.Feature, cutListFeat As SldWorks.Feature) As String

Dim regEx As Object
Set regEx = CreateObject("VBScript.RegExp")

regEx.Global = True
regEx.IgnoreCase = True
regEx.Pattern = "<[^>]*>"

Dim regExMatches As Object
Set regExMatches = regEx.Execute(template)

Dim i As Integer

Dim outFileName As String
outFileName = template

For i = regExMatches.Count - 1 To 0 Step -1

Dim regExMatch As Object
Set regExMatch = regExMatches.Item(i)

Dim tokenName As String
tokenName = Mid(regExMatch.Value, 2, Len(regExMatch.Value) - 2)

outFileName = Left(outFileName, regExMatch.FirstIndex) & ResolveToken(tokenName, rootModel, sheetMetalModel,
conf, flatPatternFeat, cutListFeat) & Right(outFileName, Len(outFileName) - (regExMatch.FirstIndex +
regExMatch.Length))
Next

ComposeOutFileName = ReplaceInvalidPathSymbols(GetFullPath(rootModel, outFileName))

End Function

Function ReplaceInvalidPathSymbols(path As String) As String

Const REPLACE_SYMB As String = "_"

Dim res As String
res = Right(path, Len(path) - Len("X:\"))

```

```

Dim drive As String
drive = Left(path, Len("X:\"))

Dim invalidSymbols As Variant
invalidSymbols = Array("/", ":", "**", "?", "*****", "<", ">", "|")

Dim i As Integer
For i = 0 To UBound(invalidSymbols)
Dim invalidSymb As String
invalidSymb = CStr(invalidSymbols(i))
res = Replace(res, invalidSymb, REPLACE_SYMB)
Next

ReplaceInvalidPathSymbols = drive + res

End Function

Function ResolveToken(token As String, rootModel As SldWorks.ModelDoc2, sheetMetalModel As SldWorks.ModelDoc2,
conf As String, flatPatternFeat As SldWorks.Feature, cutListFeat As SldWorks.Feature) As String

Const FILE_NAME_TOKEN As String = "_FileName_"
Const ASSM_FILE_NAME_TOKEN As String = "_AssmFileName_"
Const FEAT_NAME_TOKEN As String = "_FeatureName_"
Const CONF_NAME_TOKEN As String = "_ConfName_"

Const PRP_TOKEN As String = "$PRP:"
Const CUT_LIST_PRP_TOKEN As String = "$CLPRP:"
Const ASM_PRP_TOKEN As String = "$ASSMPRP:"

Select Case LCase(token)
Case LCase(FILE_NAME_TOKEN)
ResolveToken = GetFileNameWithoutExtension(sheetMetalModel.GetPathName)
Case LCase(FEAT_NAME_TOKEN)
ResolveToken = flatPatternFeat.Name
Case LCase(CONF_NAME_TOKEN)
ResolveToken = conf
Case LCase(ASSM_FILE_NAME_TOKEN)
If rootModel.GetPathName() = "" Then
Err.Raise vbError, "", "Assembly must be saved to use " & ASSM_FILE_NAME_TOKEN
End If
ResolveToken = GetFileNameWithoutExtension(rootModel.GetPathName())
Case Else

Dim prpName As String

If Left(token, Len(PRP_TOKEN)) = PRP_TOKEN Then
prpName = Right(token, Len(token) - Len(PRP_TOKEN))
ResolveToken = GetModelPropertyValue(sheetMetalModel, conf, prpName)
Elseif Left(token, Len(ASM_PRP_TOKEN)) = ASM_PRP_TOKEN Then
prpName = Right(token, Len(token) - Len(ASM_PRP_TOKEN))
ResolveToken = GetModelPropertyValue(rootModel, rootModel.ConfigurationManager.ActiveConfiguration.Name,
prpName)
Elseif Left(token, Len(CUT_LIST_PRP_TOKEN)) = CUT_LIST_PRP_TOKEN Then
prpName = Right(token, Len(token) - Len(CUT_LIST_PRP_TOKEN))
ResolveToken = GetProperty(cutListFeat.CustomPropertyManager, prpName)
Else
Err.Raise vbError, "", "Unrecognized token: " & token
End If

End Select

End Function

Function GetModelPropertyValue(model As SldWorks.ModelDoc2, confName As String, prpName As String) As String

Dim prpVal As String
Dim swCustPrpMgr As SldWorks.CustomPropertyManager

Set swCustPrpMgr = model.Extension.CustomPropertyManager(confName)
prpVal = GetPropertyValue(swCustPrpMgr, prpName)

If prpVal = "" Then

```

```

Set swCustPrpMgr = model.Extension.CustomPropertyManager("")
prpVal = GetPropertyValue(swCustPrpMgr, prpName)
End If

GetModelPropertyValue = prpVal

End Function

Function GetPropertyValue(custPrpMgr As SldWorks.CustomPropertyManager, prpName As String) As String
Dim resVal As String
custPrpMgr.Get2 prpName, "", resVal
GetPropertyValue = resVal
End Function

Function GetFileNameWithoutExtension(path As String) As String
GetFileNameWithoutExtension = Mid(path, InStrRev(path, "\") + 1, InStrRev(path, ".") - InStrRev(path, "\") - 1)
End Function

Function GetCutListFeatures(model As SldWorks.ModelDoc2) As Variant
GetCutListFeatures = GetFeaturesByType(model, "CutListFolder")
End Function

Function GetFlatPatternFeatures(model As SldWorks.ModelDoc2) As Variant
GetFlatPatternFeatures = GetFeaturesByType(model, "FlatPattern")
End Function

Sub ProcessSheetMetalModel(rootModel As SldWorks.ModelDoc2, sheetMetalModel As SldWorks.ModelDoc2, conf As String)

Dim vCutListFeats As Variant
vCutListFeats = GetCutListFeatures(sheetMetalModel)

If Not IsEmpty(vCutListFeats) Then

Dim vFlatPatternFeats As Variant
vFlatPatternFeats = GetFlatPatternFeatures(sheetMetalModel)

If Not IsEmpty(vFlatPatternFeats) Then

Dim swProcessedCutListsFeats() As SldWorks.Feature

Dim i As Integer

For i = 0 To UBound(vFlatPatternFeats)

Dim swFlatPatternFeat As SldWorks.Feature
Dim swFlatPattern As SldWorks.FlatPatternFeatureData

Set swFlatPatternFeat = vFlatPatternFeats(i)

Set swFlatPattern = swFlatPatternFeat.GetDefinition

Dim swFixedEnt As SldWorks.Entity

Set swFixedEnt = swFlatPattern.FixedFace2

Dim swBody As SldWorks.Body2

If TypeOf swFixedEnt Is SldWorks.Face2 Then
Dim swFixedFace As SldWorks.Face2
Set swFixedFace = swFixedEnt
Set swBody = swFixedFace.GetBody
Elseif TypeOf swFixedEnt Is SldWorks.Edge Then
Dim swFixedEdge As SldWorks.Edge
Set swFixedEdge = swFixedEnt
Set swBody = swFixedEdge.GetBody
Elseif TypeOf swFixedEnt Is SldWorks.Vertex Then
Dim swFixedVert As SldWorks.Vertex
Set swFixedVert = swFixedEnt
Set swBody = swFixedVert.GetBody
End If

Dim swCutListFeat As SldWorks.Feature

```

```

Set swCutListFeat = FindCutListFeature(vCutListFeats, swBody)

If Not swCutListFeat Is Nothing Then

Dim isUnique As Boolean

If (Not swProcessedCutListsFeats) = -1 Then
isUnique = True
Elseif Not ContainsSwObject(swProcessedCutListsFeats, swCutListFeat) Then
isUnique = True
Else
isUnique = False
End If

If isUnique Then

If (Not swProcessedCutListsFeats) = -1 Then
ReDim swProcessedCutListsFeats(0)
Else
ReDim Preserve swProcessedCutListsFeats(UBound(swProcessedCutListsFeats) + 1)
End If

Set swProcessedCutListsFeats(UBound(swProcessedCutListsFeats)) = swCutListFeat

Dim outFileName As String
outFileName = ComposeOutFileName(OUT_NAME_TEMPLATE, rootModel, sheetMetalModel, conf, swFlatPatternFeat,
swCutListFeat)

If Not SKIP_EXISTING_FILES Or Not FileExists(outFileName) Then
ExportFlatPattern sheetMetalModel, swFlatPatternFeat, outFileName, FLAT_PATTERN_OPTIONS, conf
End If
End If

Else
Err.Raise vbError, "", "Failed to find cut-list for flat pattern " & swFlatPatternFeat.Name
End If

Next

Else
Err.Raise vbError, "", "No flat pattern features found"
End If

Else
Err.Raise vbError, "", "No cut-list items found"
End If

End Sub

Function FileExists(filePath As String) As Boolean
FileExists = Dir(filePath) <> ""
End Function

Function FindCutListFeature(vCutListFeats As Variant, body As SldWorks.Body2) As SldWorks.Feature

Dim i As Integer

For i = 0 To UBound(vCutListFeats)

Dim swCutListFeat As SldWorks.Feature
Set swCutListFeat = vCutListFeats(i)

Dim swBodyFolder As SldWorks.BodyFolder
Set swBodyFolder = swCutListFeat.GetSpecificFeature2

Dim vBodies As Variant

vBodies = swBodyFolder.GetBodies

If ContainsSwObject(vBodies, body) Then
Set FindCutListFeature = swCutListFeat
End If

```



```

Next
End Function

Function ContainsSwObject(vArr As Variant, obj As Object) As Boolean

If Not IsEmpty(vArr) Then

Dim i As Integer

For i = 0 To UBound(vArr)

Dim swObj As Object
Set swObj = vArr(i)

If swApp.IsSame(swObj, obj) = swObjectEquality.swObjectSame Then
ContainsSwObject = True
Exit Function
End If
Next

End If

ContainsSwObject = False

End Function

Function GetFeaturesByType(model As SldWorks.ModelDoc2, typeName As String) As Variant

Dim swFeats() As SldWorks.Feature

Dim swFeat As SldWorks.Feature

Set swFeat = model.FirstFeature

Do While Not swFeat Is Nothing

If typeName = "CutListFolder" And swFeat.GetTypeName2() = "SolidBodyFolder" Then
Dim swBodyFolder As SldWorks.BodyFolder
Set swBodyFolder = swFeat.GetSpecificFeature2
swBodyFolder.UpdateCutList
End If

ProcessFeature swFeat, swFeats, typeName

Set swFeat = swFeat.GetNextFeature

Loop

If (Not swFeats) = -1 Then
GetFeaturesByType = Empty
Else
GetFeaturesByType = swFeats
End If

End Function

Sub ProcessFeature(thisFeat As SldWorks.Feature, featsArr() As SldWorks.Feature, typeName As String)

If thisFeat.GetTypeName2() = typeName Then

If (Not featsArr) = -1 Then
ReDim featsArr(0)
Set featsArr(0) = thisFeat
Else
Dim i As Integer

For i = 0 To UBound(featsArr)
If swApp.IsSame(featsArr(i), thisFeat) = swObjectEquality.swObjectSame Then
Exit Sub
End If
Next

```

```

ReDim Preserve featsArr(UBound(featsArr) + 1)
Set featsArr(UBound(featsArr)) = thisFeat
End If

End If

Dim swSubFeat As SldWorks.Feature
Set swSubFeat = thisFeat.GetFirstSubFeature

While Not swSubFeat Is Nothing
ProcessFeature swSubFeat, featsArr, typeName
Set swSubFeat = swSubFeat.GetNextSubFeature
Wend

End Sub

Sub ExportFlatPattern(part As SldWorks.PartDoc, flatPattern As SldWorks.Feature, outFilePath As String, opts As SheetMetalOptions_e, conf As String)

Dim swModel As SldWorks.ModelDoc2
Set swModel = part

Dim error As ErrObject
Dim hide As Boolean

try_:

On Error GoTo catch_

If False = swModel.Visible Then
hide = True
swModel.Visible = True
End If

swApp.ActivateDoc3 swModel.GetPathName(), False, swRebuildOnActivation_e.swDontRebuildActiveDoc, 0

swModel.FeatureManager.EnableFeatureTree = False
swModel.FeatureManager.EnableFeatureTreeWindow = False
swModel.ActiveView.EnableGraphicsUpdate = False

Dim curConf As String

curConf = swModel.ConfigurationManager.ActiveConfiguration.Name

If curConf <> conf Then
If False = swModel.ShowConfiguration2(conf) Then
Err.Raise vbError, "", "Failed to activate configuration"
End If
End If

Dim outDir As String
outDir = Left(outFilePath, InStrRev(outFilePath, "\"))

CreateDirectories outDir

Dim modelPath As String

modelPath = part.GetPathName

If modelPath = "" Then
Err.Raise vbError, "", "Part document must be saved"
End If

If False <> flatPattern.Select2(False, -1) Then
If False = part.ExportToDWG2(outFilePath, modelPath, swExportToDWG_e.swExportToDWG_ExportSheetMetal, True, Empty, False, False, opts, Empty) Then
Err.Raise vbError, "", "Failed to export flat pattern"
End If
Else
Err.Raise vbError, "", "Failed to select flat-pattern"
End If

swModel.ShowConfiguration2 curConf

```

```

GoTo finally_

catch_:
Set error = Err
finally_:

swModel.FeatureManager.EnableFeatureTree = True
swModel.FeatureManager.EnableFeatureTreeWindow = True
swModel.ActiveView.EnableGraphicsUpdate = True

If hide Then
swApp.CloseDoc swModel.GetTitle
End If

If Not error Is Nothing Then
Err.Raise error.Number, error.Source, error.Description, error.HelpFile, error.HelpContext
End If

End Sub

Sub CreateDirectories(path As String)

Dim fso As Object
Set fso = CreateObject("Scripting.FileSystemObject")

If fso.FolderExists(path) Then
Exit Sub
End If

CreateDirectories fso.GetParentFolderName(path)

fso.CreateFolder path

End Sub

Function GetFullPath(model As SldWorks.ModelDoc2, path As String)

GetFullPath = path

If IsPathRelative(path) Then

If Left(path, 1) <> "\" Then
path = "\" & path
End If

Dim modelPath As String
Dim modelDir As String

modelPath = model.GetPathName

modelDir = Left(modelPath, InStrRev(modelPath, "\") - 1)

GetFullPath = modelDir & path

End If

End Function

Function IsPathRelative(path As String)
IsPathRelative = Mid(path, 2, 1) <> ":" And Not IsPathUnc(path)
End Function

Function IsPathUnc(path As String)
IsPathUnc = Left(path, 2) = "\\
End Function

```



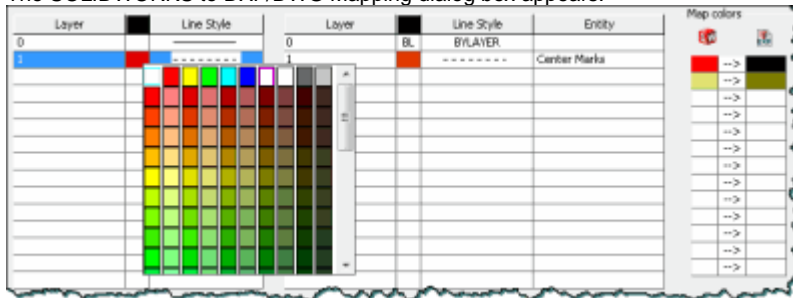
9 Mapping Entities When Saving Drawings as .DXF or .DWG Files

You can map SOLIDWORKS drawing entities and colors when you save drawings as .dxf or .dwg files to configure these entities and colors in the target files.

You can map SOLIDWORKS entities to layers, colors, or line styles you specify in the SOLIDWORKS to DXF/DWG Mapping dialog box. You can also map exported colors.

1. From a SOLIDWORKS drawing, click **File > Save As** and select .dwg or .dxf as the file type.
2. Click **Options**.
3. For **Custom map SOLIDWORKS to DXF/DWG**, select **Enable** and clear **Don't show mapping on each save**.
4. Click **OK**.
5. Type a name for the file, select a file location, and click **Save**.

The SOLIDWORKS to DXF/DWG Mapping dialog box appears.



6. In the SOLIDWORKS to DXF/DWG Mapping dialog box, specify the mappings:
 - Click **Load Map File** to upload mappings from a file.
 - Manually specify layers, map entities, or color mappings.
7. To save DXF/DWG export map file settings, click **Save Map File**.
8. Click **OK**.

10 Saving and Loading Map Entities

When exporting drawings as .dxf or .dwg files, you can save .dxf or .dwg entity property mappings in a map file and load them the next time you save a SOLIDWORKS drawing to either file type.

To save or load entity property mappings when exporting SOLIDWORKS drawings to .dwg or .dxf files:

1. Follow the main procedure for [mapping entities](#) through step 5.
2. Save or load entity and color mappings:
 - From the SOLIDWORKS to DXF/DWG Mapping dialog box, specify layers, and map colors and entities, and click **Save Map File** to save property mappings.
 - Click **Load Map File** to load previously saved property mappings.
3. Browse to a folder and enter a file name.
 - Click **Save** to save property mappings to a map file.
 - Click **Open** to load a map file with saved property mappings.
4. Click **OK** to close the SOLIDWORKS to DXF/DWG Mapping dialog box and export the file.

11 Defining Layers

When exporting drawings as .dxf or .dwg files, you can define layers in the SOLIDWORKS to DXF/DWG Mapping dialog box.

To add or edit a layer:

1. Follow the main procedure for [mapping entities](#) through step 5.
2. Under **Define Layers**, select the row for the layer and do the following:
 - a. Click in the **Layer** column and type a number or name for the new layer.
 - b. Click in the **Color** column for the layer and select an item from the grid.
 - c. Click in the **Line Style** column for the layer and select a line style.
3. Repeat Step 2, as required, to specify other layers.
4. Specify map entities or colors as required.
5. To save DXF/DWG export map file settings, click **Save Map File**.
6. Click **OK**.

12 Mapping Entities

When exporting drawings as .dxf or .dwg files, you can map entities in the SOLIDWORKS to DXF/DWG Mapping dialog box.

To map entities:

1. Follow the main procedure for [mapping entities](#) through step 5.
2. Under **Map Entities**, select the row for the entity and do the following:
 - a. Click in the **Layer** column and select a layer.

You must define the layers before you can map the entities.



- b. Click in the **Color** column for the layer and select an item from the grid.

- c. Click in the **Line Style** column for the layer and select a line style.
 - d. Click in the **Entity** column and select an entity to map.
3. Repeat Step 2, as required, to map other entities.
4. Specify map colors as required.
5. To save DXF/DWG export map file settings, click **Save Map File**.
6. To apply the mapping file settings only to those entities whose layers are not defined as SOLIDWORKS drawing file layers, click **Only map entities that are not on SOLIDWORKS layers**.
7. Click **OK**.

13 Mapping Colors

When exporting drawings as .dxf or .dwg files, you can define the color mapping from SOLIDWORKS to the target file in the SOLIDWORKS to DXF/DWG Mapping dialog box.

To map colors when saving drawings as .dxf or .dwg files:

1. Follow the main procedure for [mapping entities](#) through step 5.
2. Under **Map Colors**, specify a color mapping:
 - a. Click the next empty row in the **SOLIDWORKS**  column and select a color from the grid for the source file.
 - b. Click in the **DXF/DWG**  column and select mapped color for the target file.
3. Repeat Step 2, as required, to map other colors.
4. To save DXF/DWG export map file settings, click **Save Map File**.
5. Click **OK**.