



SAVONIA

■ OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN JA LIIKENTEEN ALA

RAKENNESUUNNITTELUN TIETOMALLIVAATIMUSTEN MUKAISEN KOMPONENTIN KEHITTÄMINEN

Tekla Structures -ohjelmiston komponenttityökaluilla

Koulutusala Tekniikan ja liikenteen ala			
Koulutusohjelma Rakennustekniikan koulutusohjelma			
Työn tekijä(t) Kimmo Kuusela			
Työn nimi Rakennesuunnittelun tietomallivaatimusten mukaisen komponentin kehittäminen Tekla Structures -ohjelmiston komponenttityökaluilla			
Päiväys	25.4.2014	Sivumäärä/Liitteet	34/40
Ohjaaja(t) lehtori Viljo Kuusela, pt. tuntiopettaja Matti Ylikärppä			
Toimeksiantaja/Yhteistyökumppani(t) Insinööritoimisto ConnAri Oy			
<p>Tiivistelmä</p> <p>Opinnäytetyön tarkoituksena oli perehtyä <i>Tekla Structuresin</i> komponenttityökaluihin sekä tehdä niillä parametrisoituja komponentteja yrityksen käyttöön. Komponentit ovat ohjelmistoissa käytettäviä rakenteita tai rakenteiden osia, esimerkiksi sandwich-elementti tai pilarin ja palkin välinen liitos. Yritykselle räätälöidyillä komponenteilla helpotetaan suunnittelijoiden päivittäisiä työtehtäviä sekä lisätään työn tehokkuutta.</p> <p>Työn lähtötietona oli, että <i>Tekla Structuresiin</i> voidaan tehdä komponentteja kahdella eri tavalla. Ensimmäinen tapa on käyttää hyväksi <i>Tekla Structuresista</i> löytyvää Custom Component Editor -työkalua ja toinen tapa Tekla Open API -rajapintaa hyödyntäen. Ohjeet työn toteutukseen löytyivät pääasiassa Teklan tarjoamista oppaista, mutta paljon apua saatiin myös Teklan työntekijöiltä. Myös aiemmin tehdyt opinnäytetyöt auttoivat pääsemään alkuun. Teoriaosuus työhön on koottu tietomallinnusta käsitteivistä kirjoista ja RT-korteista sekä Internet-sivustoista.</p> <p>Työn tuloksena saatiin Custom Component Editorilla tehtyjä komponentteja Insinööritoimisto ConnAri Oy:n käyttöön. Komponenttien tekemisen ohessa, saatiin laadittua myös ohjeet, joiden avulla yrityksen työntekijä voi alkaa itse tekemään haluamiaan komponentteja. Ohjeet sisältävät myös tietoa haasteista, joita voi tulla eteen työskentelyn aikana ja sen kuinka niistä selvittää.</p>			
Avainsanat Tekla Structures, Custom Component Editor, Open API, Tietomallinnus			

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme In Construction Engineering			
Author(s) Kimmo Kuusela			
Title of Thesis Developing Component that Meets the Requirements Set by BIM in Structural Engineering with Component Tools of Tekla Structures			
Date	25 April 2014	Pages/Appendices	34/40
Supervisor(s) Mr Viljo Kuusela, Lecturer Mr Matti Ylikärppä, Lecturer			
Client Organisation /Partners Insinööritoimisto ConnAri Oy			
<p>Abstract</p> <p>The objective of this thesis was to get acquainted with the component tools of Tekla Structures and with those tools create parameterized components for the use of the company. Components are structures or parts of a structure which are used in the software, such as a sandwich panel or the joint between a column and a beam. Components that are specifically made to serve the need of the company can ease the daily work and increase the efficiency of the work.</p> <p>It was known that components could be made by two different ways to the Tekla Structures. The first way is to use the Custom Component Editor which can be found inside the software and the second way is to build the component by using the Tekla Open API interface. Most of the instructions on how to make a component were found from the guide books offered by Tekla but a large amount of help was also received from the employees of Tekla. Previously made theses were quite helpful at the beginning as well. The theoretical part of the thesis was gathered from books and RT cards as well as from Internet pages that are related to the subject of building information modeling.</p> <p>As a result of this thesis components were made with the Custom Component Editor for the use of Insinööritoimisto ConnAri Oy. The component making process also produced instructions which the employees of the company can utilize when they want to learn how to make components themselves. The instructions also included challenges that might appear during the process and also how to overcome them.</p>			
Keywords Tekla Structures, Custom Component Editor, Open API, Building Information Modeling			

ESIPUHE

Opinnäytetyö on toteutettu Insinööritoimisto ConnAri Oy:lle alkuvuodesta 2014. Kiitokset mielenkiintoisesta aiheesta menevät yrityksen työntekijöille Ari Tanskaselle ja Juhani Lyytiselle. Aihetta tutkaillessani olen saanut aivan uuden käsityksen tietomallintamisen mahdollisuuksista ja oppinut paljon Tekla Structures -ohjelmiston komponenttityökaluista. Komponenttien tekeminen sisälsi runsaasti haasteita, joiden päihittämisessä auttoivat suuresti Teklan työntekijöiden ammattitaito ja ohjeistaminen opinnäytetyöprojektin edetessä. Kiitokset Teklan suunnalta tulleesta avustuksesta menevät Mikko Raikaalle, Jari Pataselle sekä Sebastian Lindholmille.

Aiheen laajuus käsitti alun perin vain Custom Component Editorin, mutta jälkeinpäin tarkasteltuna oli hyvä, että ohjaava opettajani Viljo Kuusela ohjeisti tutkimaan myös Tekla Open API -rajapintaa, vaikkakin se lisäsi työtaakkaa suuresti. Opinnäytetyön laajuuden kasvaessa olin pakotettu opiskelemaan ohjelmointia, jonka edes alkeellinen hallitseminen on auttanut omalta osaltaan ymmärtämään paremmin myös Tekla Structures -ohjelmistoa ja sitä kuinka se toimii. Kiitokset menevät myös kieliasun tarkastaneelle Jarna Aromaa-Laamaselle ja mielenkiintoisille sähköpostikeskusteluille. Jatkossa pyrin laittamaan pisteen luettelon loppuun omatoimisesti.

Komponenttien kehittäminen jatkuu myös opinnäytetyön ulkopuolella. Tällä hetkellä työkaluna toimii Custom Component Editor, mutta aika näyttää tuleeko jatkossa rinnalle myös Tekla Open API -rajapinta ja ohjelmoinnin tuomat haasteet.

Kuopiossa 24.4.2014

Kimmo Kuusela

SISÄLTÖ

1	JOHDANTO	6
1.1	Työn tausta ja tavoitteet	6
1.2	Toimeksiantaja	7
2	TIETOMALLINNUS	8
2.1	Yleisesti.....	8
2.2	Miten tietomallinnus eroaa perinteisestä suunnittelusta?.....	9
2.3	Tietomallintamisen tason vaatimukset rakennesuunnittelijalle eri suunnitteluvaiheissa.....	10
2.4	Tietomallinnuksen yhdenmukaistamiseen tähtäävät kehitysprojektit.....	14
2.4.1	TELU2012.....	14
2.4.2	YTV2012.....	16
2.4.3	BEC2012.....	18
3	TYÖKALUT KOMPONENTTIEN TEKEMISEEN.....	20
3.1	Tekla Structures.....	21
3.2	Custom Component Editor	22
3.3	Tekla Open API.....	23
4	TIETOMALLIVAATIMUSTEN MUKAISET KOMPONENTIT	25
4.1	Kehitysprojektit tietomallinnettujen komponenttien tukena	25
4.2	Esimerkkikohteessa tarvittavat komponentit	27
4.3	Tehdyt komponentit.....	28
4.3.1	Sokkelielementti.....	28
4.3.2	Sokkelielementin alapään kiinnitys	29
4.3.3	Sokkelielementin nurkkaliitos	29
5	TULOKSET	30
6	POHDINTA.....	31
6.1	Komponenttien ja ohjeiden tekeminen	31
6.2	Komponenttityökalujen tarjoamat haasteet.....	32
6.3	Opinnäytetyön aikaansaamat hyödyt itselle ja tilaajalle.....	32
	LÄHTEET	33
	LIITE 1 ESIMERKKI KOMPONENTTI CUSTOM COMPONENT EDITORILLA.....	35
	LIITE 2 ESIMERKKI TEKLA OPEN APIN KÄYTTÄMISESTÄ	71

1 JOHDANTO

1.1 Työn tausta ja tavoitteet

Tietomallinnus on valtaamassa rakennusalaan Suomessa, kun yhä useammat alan ammattilaiset alkavat hahmottaa, kuinka suuret hyödyt sillä voidaan saavuttaa rakennusprojektin jokaiseen vaiheeseen. Joissain rakennuskohteiden vaatimuksissakin on jo määritelty, että projekti tulee toteuttaa tietomallintamalla. Suomessa tietomallinnus on otettu paremmin vastaan kuin missään muualla maailmassa, mikä selittyy sillä, että Suomi on teknologisesti edistynyt valtio, jolla on pieni ja nopeasti uusiin ideoihin reagoiva rakennusala. Edellä mainitut asiat yhdistettynä avoimiin standardeihin sekä eräänlaiseen luottamukseen rakennusalaan kohtaan, antavat hyvän pohjan tietomallinnuksen yleistykselle ja kehittymiselle Suomessa. (wspgroup.com b.) Tietomallit luovat havainnollisemman esityksen rakennuskohteesta ja mahdollistavat rakennusprojektin kokonaisvaltaisen hallinnan yhdellä kaiken sisältävällä mallilla. Vanha sanonta liittyen hyvin suunniteltuun, voidaan havaita käytännössä projekteissa, jotka toteutetaan vain rakentamista itseään lukuun ottamatta täysin sähköisesti tietomallin avulla. Parhaassa tapauksessa pitkälle suunniteltu tietomalli voi tuoda suuret säästöt, kun siinä huomioidaan koko rakennuksen elinkaari. Tällöin suunnittelutyöhön uhrattu hieman suurempi työmäärä ei paina enää missään.



Kuva 1. Tietomallinnus maailmalla (wspgroup.com a)

Opinnäytetyö käsittelee Tekla Structures -suunnitteluohjelmiston tarjoamia vaihtoehtoja komponenttien luomiseen. Työn ohella on myös tarkoitus luoda toimivia parametrisoituja komponentteja yrityksen käyttöön. Komponenteista pyritään tekemään sen tasoisia, että ne noudattavat Yleiset tietomallin vaatimukset -RT-korttien edellyttämiä mallinnuksen tasoja. Samalla pyritään tekemään myös yksityiskohtaiset ohjeet, joita noudattamalla tulisi oppia ainakin alkeet Teklan Custom Component Editorista. Tekla Open API vaatii jonkinlaista käsitystä ohjelmoinnista, joka ei kuulu ainakaan toistaiseksi rakennesuunnittelijoiden arkeen, joten sen käsittely jää tässä työssä taka-alalle. Aihe kuitenkin kiinnostaa sen verran, että siihen tullaan perehtymään hieman opinnäytetyössä.

1.2 Toimeksiantaja



Kuva 2. ConnArin logo (connari.fi)

Opinnäytetyön tilaajana toimii kuopiolainen Insinööritoimisto ConnAri Oy. Nykyinen toimitusjohtaja Ari Tanskanen perusti yrityksen vuonna 2007. Suunnittelutoimisto työllistää täysipäiväisesti kaksi rakennusalan ammattilaista ja kesäisin yrityksellä on myös tapana palkata harjoittelija ammattikorkeakoulusta. ConnArin markkina-alueena toimii pääasiassa Suomi, mutta projekteja on ollut myös Venäjällä sekä Pohjoismaissa. Mallinnukseen yritys käyttää pääasiassa Tekla Structures -ohjelmistoa ja mitoitukseen Robot Structural Analysis -ohjelmistoa. Suunnittelukohteista suurin osa on teollisuuden puolelta. (connari.fi)

2 TIETOMALLINNUS

2.1 Yleisesti

Tietomallinnus on rakennetun ympäristön elinkaaren tietojen hallintaa digitaalisessa muodossa. Tietomallinnus mahdollistaa rakennuskohteen tietojen hallinnan suunnittelusta rakentamiseen sekä käyttöön ja ylläpitoon. Se antaa mahdollisuuden tarkastella eri vaihtoehtoja rakentamiskohteen koko elinkaaren ajalta. (Laitinen 2012-05-30, 4.)

Tietomallintamista voidaan tehdä siihen soveltuvilla ohjelmistoilla eri laajuudella sen mukaan, millainen kohde on suuruudeltaan, luonteeltaan ja toteutustavaltaan. Tilaaja voi myös tietenkin antaa omat vaatimuksensa mallinnuksen tarkkuudelle. Mallintaminen voidaan rajoittaa vain osalle hanketta tai vain joihinkin hankkeen osapuoliin, mutta sillä voidaan myös ottaa hallintaan kohteen koko elinkaaren aikaiset tiedot. Parhaimmillaan tietomalli sisältää rakennushankkeen kaikkien osapuolten tietojen hallinnan sähköisessä muodossa koko elinkaaren ajalta. (Tietomallinnettava rakennushanke. Ohjeita rakennuttajalle. RT 10-10992, 3.)

Tietomallintaminen voidaan jakaa **neljään** luokkaan sen laajuuden perusteella: **suppea tietomallintaminen, integroitu tietomallintaminen, suunnittelun ja toteutuksen yhdistävä tietomallintaminen** sekä **elinkaarihankkeet**. Suppeassa tietomallintamisessa vain yksi hankkeen osapuoli käyttää tietomallintamista hyväkseen, tällöin tietomallintamista saatu hyöty ei tavoita koko hanketta. Integroidussa tietomallintamisessa eri suunnitteluosapuolet tekevät yhteistyötä tietomallin ja hyväksikäyttäen, esimerkkinä rakennesuunnittelijat ja talotekniikansuunnittelijat. Suunnittelun ja toteutuksen yhdistävä tietomallintaminen tukee rakentajan määrätietojen hallintaa, kustannuslaskentaa ja hankkeen aikataulujen hallintaa. 3D-suunnittelua, johon on lisätty aikatekijä, kutsutaan nimellä 4D-suunnittelu ja 4D-suunnittelua, johon on lisätty vielä kustannustekijä, kutsutaan nimellä 5D-suunnittelu. Elinkaarihankkeet tähtäävät rakennuskohteen koko elinkaaren hallintaan ja tietomallin mahdollisimman laajaan käyttöön kaikissa suunnittelun, rakentamisen, käytön ja ylläpidon vaiheissa (6D-suunnittelu?). (Tietomallinnettava rakennushanke. Ohjeita rakennuttajalle. RT 10-10992, 3.)

Tietokoneilla tehtävä suunnittelu voidaan jakaa kolmeen tasoon siihen käytettävien ohjelmistojen perusteella. Tasolla yksi suunnittelu hoidetaan perinteisesti 2D-pohjaisesti. Tasolla kaksi suunnittelussa käytetään jo 3D-mallinnusta, mutta koska mallit eivät sisällä muuta kuin rakennuksen ja sen osien muodon, eivät ne sovellu kuin suunnitelmien havainnollistamiseen ja visualisointeihin. Tasolla kaksi suunnitteluasiakirjat ovat ensimmäisen tason mukaisesti vain 2D-piirustuksia. Siirryttäessä tasolle kolme voidaan alkaa käyttää termiä tietomallintaminen. Mallintamisessa käytetään 3D-suunnittelua, mutta toisin kuin tasolla kaksi, mallit sisältävät rakennuksen muodon lisäksi myös muuta oleellista tietoa. Näiden tietojen avulla saadaan tarvittaessa selville esimerkiksi rakennuksen kustannukset, energiankulutus, ilmanvaihto, valaistus ja akustiikka. Tietomallit tallennetaan joko ohjelmistojen omissa tallennusmuodoissa tai järjestelmäriippumattomassa IFC-tallennusmuodossa. (Tietomallinnettava rakennushanke. Ohjeita rakennuttajalle. RT 10-10992, 2.)

2.2 Miten tietomallinnus eroaa perinteisestä suunnittelusta?

Monelle asiaan perehtymättömälle voi tulla helposti sellainen käsitys, että kaikki tietokoneella tehtävät mallintaminen luokitellaan tietomallinnukseksi, mutta näin ei tietenkään ole. 3D-mallit jotka sisältävät ulkokuoren, mutta eivät pidä minkäänlaista tietoa sisällään, eivät ole tietomalleja. Ne soveltuvat hyvin vain visualisointiin. Myöskään mallin osat, joiden sijaintia tai mittasuhteita ei voi yksinkertaisesti muuttaa, koska ne eivät pidä sisällään parametrejä, eivät kuulu yllä mainittuun kategoriaan. Esimerkkinä voisi mainita palkin, jonka ominaisuuksia muutettaessa, joutuu sen mallintamaan alusta alkaen tai ainakin tekemään suuren työn sitä muokatessa. Tietomallissa palkin muuttaminen onnistuu parametrien vuoksi erittäin helposti. Parametrit myös mahdollistavat sen, että palkin muuttamisen myötä, myös siihen liitoksissa olevat osat muuttuvat automaattisesti. Esimerkiksi pilarin kokoa muuttamalla myös sen aluslevy muuttaa kokoaan. Tietomalleja eivät ole myöskään ne mallit, joiden jokaiseen "näkymään" joutuu tekemään muutokseen erikseen, kun yhden "näkymän" osaa muutetaan. Tällöin yhden "näkymän" sisältävä tieto ei siis päivity automaattisesti muihin "näkymiin", mikä voi hyvin helposti johtaa virheisiin suunnitteluprosessin aikana varsinkin silloin, jos malliin joudutaan tekemään muutoksia/korjauksia. (Eastman, Teicholz, Sacks ja Liston 2008, 15 - 16.)

Rakennushankkeessa, jossa tieto siirtyy yhä 2D-muodossa, hankkeeseen liittyvän dokumentoidun tiedon arvo ja merkitys kasvavat suunnittelun ja rakentamisen edetessä. Kun siirrytään hankevaiheesta toiseen, osa tietovarannosta häviää, koska sitä joudutaan muokkaamaan uuteen muotoon hankevastuun siirtyessä toisalle. Tietojen häviämistä tapahtuu esimerkiksi, kun hankevastuu siirtyy suunnittelusta rakentamiseen. Suurin tietovarannon menetys tapahtuu rakennuksen käyttöönotto vaiheessa, kun suunnittelun ja rakentamisen tuottama tieto siirretään toteutussuunnitelmista kiinteistönhallinnan tietokantoihin ja huoltokirjoihin. Kaikkea tietovarantoa ei tarvita tietenkään kohteen jokapäiväisessä käytössä, mutta tiedon kysyntä kasvaa, kun kohteeseen suoritetaan korjauksia ja kunnostuksia. Tietomallinnuksella toteutetun kohteen kaikki tieto on digitaalisessa muodossa, jonka vuoksi tietoa voidaan hyödyntää tarvittaessa helposti koko rakennuksen elinkaaren ajan. Taulukossa 1. on perinteisen suunnittelun ja integroidun tietomallinnuksen vertailua. (Tietomallinnettava rakennushanke. Ohjeita rakennuttajalle. RT 10-10992, 2.)

Taulukko 1. Perinteisen ja tietomallipohjaisen suunnittelun eroja (Tietomallinnettava rakennushanke. Ohjeita rakennuttajalle. RT 10-10992, 2)

Perinteinen suunnittelu	Mallintava integroitu suunnittelu
Tiimit <ul style="list-style-type: none"> • hajautuneita • perinteiseen hankeosapuolijakoon keskittyneitä • tottuneet toimittamaan ainoastaan pyydettyä tietoa (toimivat "pienimmän vaivan" periaattella) • painottavat toiminnassa osapuolten välistä työnjakoa • hierarkisesti johdettuja • perustuvat eriytyneeseen ammattitaitoon 	<ul style="list-style-type: none"> • aikaisempien hankkeiden perusteella muodostuneita • muodostetaan hankkeen keskeisistä toimijoista ja osajista • toimittavat tarvittavaa tietoa oikea-aikaisesti • painottavat toiminnassa yhteistyökykyä • parhaiden käytössä olevien kykyjen perusteella avoimesti johdettuja • perustuvat eri osapuolten osaamisen yhdistämiseen
Prosessit <ul style="list-style-type: none"> • vaiheittaisia, erillisiä, eriytyneitä • tietoa "hamstrataan" ja se kootaan viime hetkellä • tietoa jaetaan vain pyytäjälle (maksajalle) ja vain pyydettyäessä 	<ul style="list-style-type: none"> • samanaikaisia, päällekkäisiä, monitasoisia • tiedon kokoamisessa ennakoidaan tulevia hankevaiheita • tietoa jaetaan avoimesti ja vapaasti tiedon tarvisijoille
Riskit <ul style="list-style-type: none"> • kohdistuvat yksittäisiin osapuoliin 	<ul style="list-style-type: none"> • kohdistuvat kaikkiin osapuoliin ja jaetaan tarkoituksenmukaisesti • pienenevät hankkeen hallittavuuden parantuaessa
Hyödyt ja voitot <ul style="list-style-type: none"> • mahdollisimman paljon hyötyä itselle mahdollisimman pienellä vaivalla • toiminta perustuu yleensä nopeimmin saavutettaviin voittoihin 	<ul style="list-style-type: none"> • mallintaminen hyödyttää kaikkia hankeosapuolia • suurimmat hyödyt koituvat hankkeen omistajalle ja keskeisille toimijoille • toiminta on arvoperusteista
Kommunikaatio <ul style="list-style-type: none"> • piirustusperusteinen 	<ul style="list-style-type: none"> • tietomallipohjainen (3-, 4-, 5-ulotteinen)
Sopimukset <ul style="list-style-type: none"> • kannustavat yksipuoliseen toimintaan • pyrkivät kohdentamaan riskejä 	<ul style="list-style-type: none"> • kannustavat yhteistyöhön • edistävät ja tukevat avointa tiedonjakoa

2.3 Tietomallintamisen tason vaatimukset rakennesuunnittelijalle eri suunnitteluvaiheissa

Tietomallintamista ei ole järkevää toteuttaa heti hankkeen alkuvaiheessa suurella tarkkuudella, koska suunnitelmat tulevat luultavammin muuttumaan vielä hankkeen edetessä. Tarjouskilpailuvaiheessa ei ole vielä edes varmaa, jääkö tarjottu projekti oman yrityksen toteutettavaksi. Jokaiseen suunnitteluvaiheeseen on määrätty vähimmäisvaatimukset tietomallintamisen tasolle Yleisissä tietomallivaatimuksissa (YTV2012).

Vaatimusmalli

Suunnittelu saa alkunsa vaatimusmallista. Se voi olla taulukkomuotoinen esitys, piirustus, tekstiasiakirja, tietomalli tai näiden yhdistelmä. Siinä esitetään rakennesuunnittelulle asetettuja tavoitteita ja vaatimuksia esimerkiksi käytettävät määräykset ja ohjeet, tilaajan antamat lähtötiedot ja veloitteet sekä muut mahdolliset vaatimukset ja lähtötiedot (muuntojoustavuus, vapaat korkeudet yms.). (Yleiset tietomallivaatimukset 2012. Osa 5. Rakennesuunnittelu. RT 10-11070, 3.)

Ehdotussuunnittelu

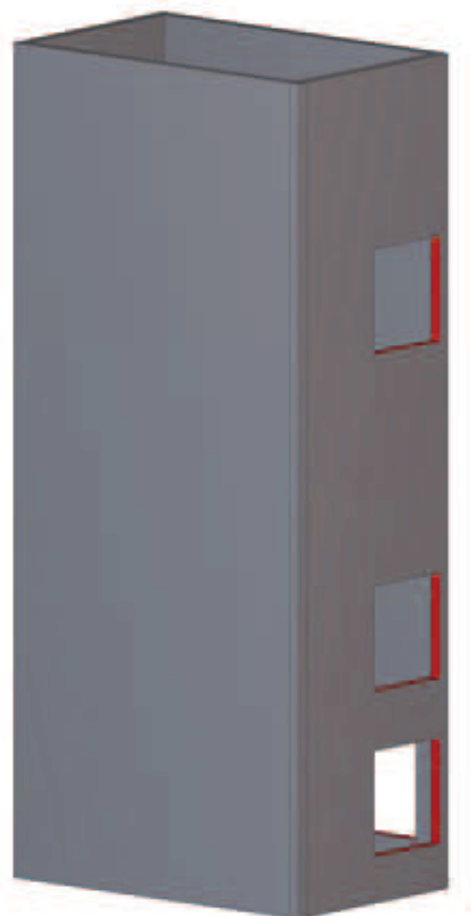
Ehdotussuunnitteluvaiheessa rakennesuunnittelija ottaa arkkitehdin esittämät vaihtoehdot tarkasteluun ja arvioi niiden toteuttamiskelpoisuutta. Varsinaista mallinnusvaatimusta rakennesuunnittelijalla ei ole vielä tässä vaiheessa, mutta projektikohtaisesti voidaan sopia, että rakennesuunnittelija mallintaa esimerkiksi erilaisia runkovaihtoehtoja kustannusten selvittämiseksi. Mallinnuksen tarkkuus vastaa yleissuunnitteluvaihetta. (Yleiset tietomallivaatimukset 2012. Osa 5. Rakennesuunnittelu. RT 10-11070, 4.)

Yleissuunnittelu

Rakennesuunnittelija varmistaa yleissuunnitteluvaiheessa tietomallin avulla rakennejärjestelmän mitoituksen, vaatimukset ja vaikutukset muiden suunnittelijoiden työhön. Rakennemallin tulee täyttää YTV:n osassa 5 määritellyt vaatimukset (YTV2012 Osa 5. Rakennesuunnittelu). Mallia tulee voida käyttää alustavassa suunnitelmien yhteensovittamisessa eli muun muassa törmäystarkasteluissa. Yhteensovittaminen on yleensä joko pääsuunnittelijan tai tietomallikoordinaattorin vastuulla, mutta työn hoitaminen voidaan siirtää myös jollekin toiselle osapuolelle. Yhteensovittamisessa tehdyillä tarkastuksilla varmistetaan mallien olevan mallinnusvaatimusten mukaisia. Samalla varmistetaan myös suunnitelmien laatua ja määrätietojen luotettavuutta. (Yleiset tietomallivaatimukset 2012. Osa 1. Yleinen osuus. RT 10-11066, 8.)

Yleissuunnitteluvaiheessa projektin tietomallista tulisi saada seuraavat hyödyt (Yleiset tietomallivaatimukset 2012. Osa 5. Rakennesuunnittelu. RT 10-11070, 4):

- visuaalinen tarkastelu
- lähtötietona määrä- ja kustannuslaskennoille
- alustava runkoaikataulutus
- suunnitelmien yhteensovittaminen
- lähtötietona rakennesuunnittelijan lujuuslaskentamallille
- lähtötietona seuraavalle suunnitteluvaiheelle.



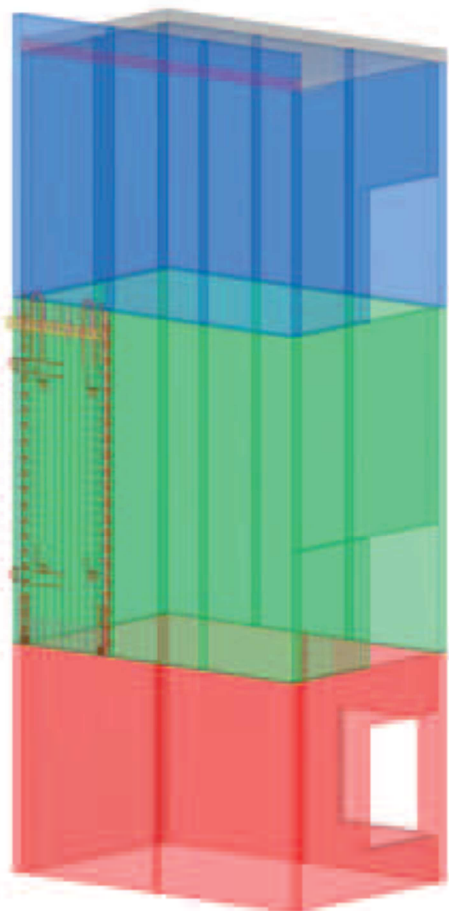
Kuva 3. Porrashuoneen mallitarkkuus yleissuunnitteluvaiheessa (Yleiset tietomallivaatimukset 2012. Osa 5. Rakennesuunnittelu. RT 10-11070, 4)

Hankintoja palveleva suunnittelu

Hankintoja palvelevassa suunnittelussa kohteen rakennemalli kehitetään hankintakyselyjen edellyttämälle tasolle ja laaditaan tarjouspyyntöasiakirjat hankintakyselyjä varten. Mallista tulisi saada irti kantavien ja ei-kantavien betonirakenteiden koko, laajuus, määrät sekä tarkka sijainti. Tällä tarkkuudella tehtävä suunnittelu palvelee esimerkiksi talotekniikan suunnittelijoiden tekemiä LVI-suunnitelmia. (Yleiset tietomallivaatimukset 2012. Osa 5. Rakennesuunnittelu. RT 10-11070, 5.)

Hankintoja palvelevassa suunnitteluvaiheessa projektin tietomallista tulisi saada seuraavat hyödyt (Yleiset tietomallivaatimukset 2012. Osa 5. Rakennesuunnittelu. RT 10-11070, 5):

- suunnitelmien havainnollistaminen
- määrälaskenta
- suunnitelmien yhteensovittaminen
- työturvallisuuden ja rakennusalueen käytön suunnittelu
- rakentamisaikataulun suunnittelu ja havainnollistaminen
- asennus- ja työjärjestystensuunnittelu
- lähtötieto toteutussuunnittelulle.



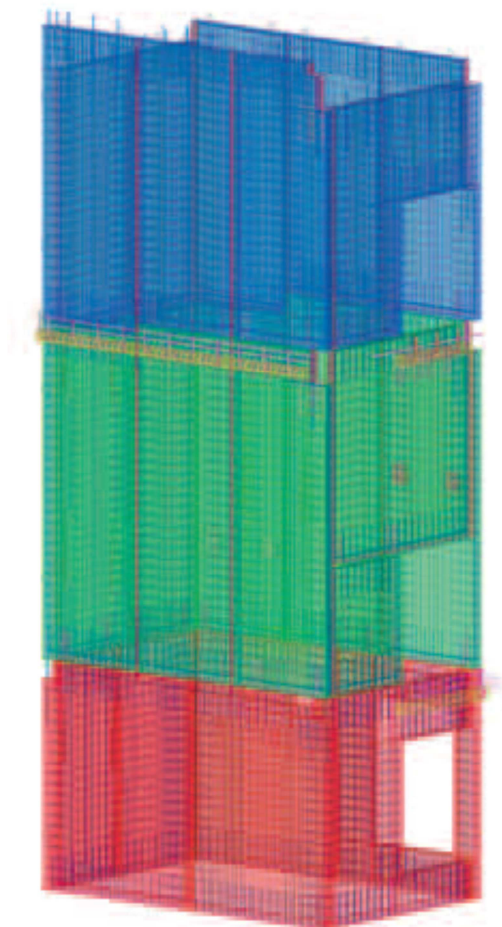
Kuva 4. Porrashuoneen mallitarkkuus ja mallielementti hankintoja palvelevassa suunnitteluvaiheessa. (Yleiset tietomallivaatimukset 2012. Osa 5. Rakennesuunnittelu. RT 10-11070, 5)

Toteutussuunnittelu

Rakennemallin sisältö ja tarkkuus määräytyy toteutussuunnitteluvaiheessa rakennesuunnittelijan tehtävien laajuuden mukaan. Jos rakennesuunnittelijan työtehtäviin sisältyy esimerkiksi elementtisuunnittelu, tulee kaikki elementit mallintaa samaan tarkkuuteen kuin hankintoja palvelevan suunnitteluvaiheen mallielementit. Mikäli elementtisuunnittelu ei kuulu rakennesuunnittelijan työtehtäviin, jatkaa hän mallin kehittämistä muiden rakenteiden osalta. (Yleiset tietomallivaatimukset 2012. Osa 5. Rakennesuunnittelu. RT 10-11070, 8.) Rakennesuunnittelijan tietomallin tulisi vastata arkkitehtimallia ja sen tulisi täyttää sille YTV:n osassa 5 (rakennesuunnittelu) määrätyt vaatimukset. Mallia tulee voida hyödyntää määrälaskennassa, suunnitelmien yhteensovittamisessa ja toteutusaikataulun laatimisessa. (Yleiset tietomallivaatimukset 2012. Osa 1. Yleinen osuus. RT 10-11066, 9.)

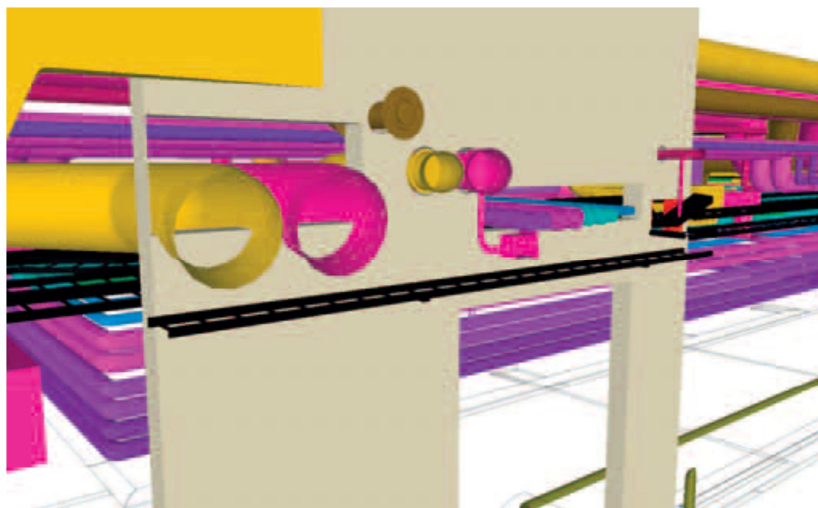
Toteutussuunnitteluvaiheessa projektin tietomallista tulisi saada seuraavat hyödyt (Yleiset tietomallivaatimukset 2012. Osa 5. Rakennesuunnittelu. RT 10-11070, 8):

- suunnitelmien havainnollistaminen
- määrälaskenta
- suunnitelmien yhteensovittaminen
- työturvallisuuden ja rakennusalueen käytön suunnittelu
- rakentamisaikataulun ja toteutumatilanteen esittäminen ja havainnollistaminen
- asennus- ja työjärjestysten suunnittelu.



Kuva 5. Porrashuoneen mallitarkkuus toteutussuunnitteluvaiheessa (Yleiset tietomallivaatimukset 2012. Osa 5. Rakennesuunnittelu. RT 10-11070, 8)

Toteutussuunnitteluvaiheessa tulisi myös tehdä niin sanottu yhdistelmämalli, jolla havainnollistetaan suunnitelmia ja tarkastellaan suunnitelmien yhteensopivuutta. Tässä vaiheessa tehdään esimerkiksi seuraavia tarkasteluja: TATE-järjestelmien törmäystarkastelut, järjestelmien ja rakenteiden törmäystarkastelut, järjestelmille varattujen tilojen riittävyyden verifiointi sekä reikä- ja varaussuunnittelu. (Yleiset tietomallivaatimukset 2012. Osa 1. Yleinen osuus. RT 10-11066, 9.)



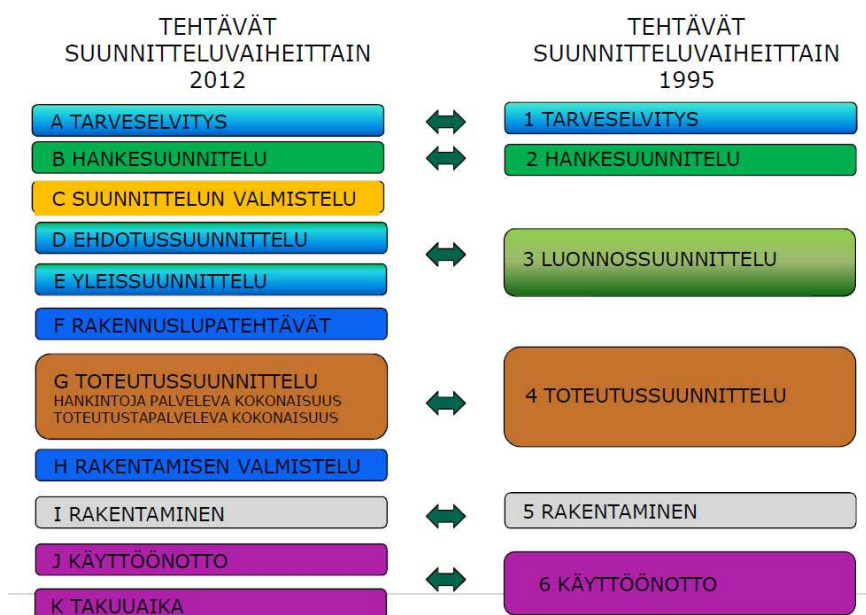
Kuva 6. Rakenne- ja TATE-mallin törmäystarkastelu (Yleiset tietomallivaatimukset 2012. Osa 6. Laadunvarmistus, 5)

2.4 Tietomallinnuksen yhdenmukaistamiseen tähtäävät kehitysprojektit

Jotta tietomallinnuksesta saadaan kaikki sen potentiaali jatkossa irti, tarvitaan yhteiset normit, joiden mukaan voidaan suunnitella ja mallintaa. Eri suunnitteluosapuolten mallien yhdistämisessä on järjetön, ellei jopa mahdoton työ, jos tietomallinnuksessa ei ole sovellettu yhteisiä "pelisääntöjä". Seuraavissa kappaleissa on esitelty suunnittelua ja mallinnusta edistäviä kehitysprojekteja.

2.4.1 TELU2012

Tehtäväluettelot 2012 -projektin tarkoituksena oli päivittää vuoden 1995 rakennuttamisen ja suunnittelun tehtäväluettelot nykypäivään. Uusissa tehtäväluetteloissa tehtävät on ryhmitelty suunnittelualoittain ja kokonaisuuksiin, jotka helpottavat suunnittelun hankintaa. Tämän uudelleen järjestelyn myötä hankekohtaisten konsulttisopimusten laadinta on helpottunut, hankkeita pystytään rinnastamaan toisiinsa entistä helpommin ja tehtäväluettelot soveltuvat kaikkiin toteutusmuotoihin. Hankkeen tehtävien jakaantuessa tehtäväkokonaisuuksiin, voivat ne eri hankintamuodoissa ajoittua ja sijoittua eri tavoin. (Uudistamisen periaatteet ja tehtäväluetteloiden käyttö, 6.) Kuviossa 1 näkyy, kuinka tehtäväkokonaisuudet ovat muuttuneet verrattuna aikaisempaan.



Kuvio 1. Tehtäväkokonaisuuksien päivittyminen (TATE12 Vihdoin valmiina?, 2)

TELU 2012 sisältää seuraavaa (Uudistamisen periaatteet ja tehtäväluetteloiden käyttö, 2 - 3):

Tehtäväluettelot:

- Hankkeen johtamisen ja rakennuttamisen tehtäväluettelo HJR12
- Pääsuunnittelun tehtäväluettelo PS12
- Arkkitehtisuunnittelun tehtäväluettelo ARK12
- Geosuunnittelun tehtäväluettelo GEO12
- Rakennesuunnittelun tehtäväluettelo RAK12
- Taloteknisen suunnittelun tehtäväluettelo TATE12

Täydentävät RT-kortit

- Tehtäväluettelon käyttöohje KO12
- Hanketietokortti HT12

Erityisalojen tulossa olevat tehtäväluettelot

- Akustiikkasuunnittelun tehtäväluettelo AKU12
- Sisustussuunnittelun tehtäväluettelo SIS12
- Valaistussuunnittelun tehtäväluettelo VAL12
- Maisemasuunnittelun tehtäväluettelo
- Elinkaarisuunnittelijan tehtäväluettelo.

Tehtäväluetteloiden tarkoitus on määritellä talonrakennusta koskevien suunnittelutehtävien sisältö ja laajuus. Niitä käytetään lisäksi suunnittelukokonaisuuden hallinnassa sekä osana suunnittelun laadunvarmistusta. Luetteloiden tarkoitus ei ole kuitenkaan määritellä suunnittelurajoja, vaan eri suunnittelutehtävien suorittajat ovat hankekohtaisesti valittavissa. Tehtäväluettelot ovat tarkoitettu käytettäväksi eri toteutus- ja urakkamuodoissa perinteisistä kokonaisurakoista projektinjohtototeutukseen ja suunnittelun sisältäviin urakoihin. Taulukossa 2 on esitetty hankkeen johtamisen ja suunnittelun tehtäväluettelorakenne. (Tehtäväluettelot. Käyttöohje KO12. RT 10-11105, 1 - 2.)

Taulukko 2. Tehtäväluektorakenne (Tehtävälueellot. Käyttöohje KO12. RT 10-11105, 2)

	JOHTAMINEN		RAKENUSSUUNNITTELU				MUUT SUUNNITTELU- JA ASIAANTUNTIJATEHTÄVÄT				
	Hankkeen johtamisen ja rakennuttamisen tehtävälueello	Pääsuunnittelun tehtävälueello	Arkkitehtisuunnittelun tehtävälueello	Taloteknisen suunnittelun tehtävälueello	Rakennesuunnittelun tehtävälueello	Geoteknisen suunnittelun tehtävälueello	Sisustus suunnittelun tehtävälueello	Akustiikka suunnittelun tehtävälueello	Valaistus suunnittelun tehtävälueello	Elinkaariaasian tuntiija Palotekninen asiantuntija	
	HJR12	PS12	ARK12	TATE12	RAK12	GEO12	SIS12	AKU12	VAL12		
TEHTÄVÄKOKONAISUUS	A	Tarveselvitys									
	B	Hankesuunnittelu									
	C	Suunnittelun valmistelu									
	D	Ehdotussuunnittelu									
	E	Yleissuunnittelu									
	F	Rakennus lupatehtävät									
	G	Toteutus suunnittelu									
	H	Rakentamisen valmistelu									
	I	Rakentaminen									
	J	Käyttöönotto									
	K	Takuu aika									

2.4.2 YTV2012

Yleiset tietomallivaatimukset 2012 sisältää ohjeet tietomallipohjaiseen suunnitteluun. Alun perin Senaatti-kiinteistöjen vuonna 2007 julkaisemat ohjeet päivitettiin vastaamaan nykypäivää vuoden 2011 aikana COBIM-hankkeen muodossa. Hanketta oli toteuttamassa muun muassa Senaatti-kiinteistöt, Suomen johtavia suunnittelutoimistoja, kiinteistön omistajia ja urakoitsijoita. (Yleiset tietomallivaatimukset 2012. Esittely. RT 10-11080, 1.)

COBIM (Common BIM requirements) nimellä kulkenut projekti tietomallinnusvaatimusten määrittämiseksi sai siis lopulta nimekseen YTV2012. Yleisten tietomallivaatimusten tarkoituksena on luoda Suomeen yleisesti käytössä olevat standardit tietomallipohjaiseen suunnitteluun. Jotta tietomallinnuksesta saadaan paras hyöty irti, tarvitaan yleiset normit, joiden mukaisesti suunnittelua tehdään. YTV2012 antaa hyvät edellytykset tilaajan, urakoitsijan ja eri suunnittelijoiden väliseen yhteistyöhön. (Kautto 2012-1, 40.) YTV kattaa uudis- ja korjausrakentamiskohteet sekä rakennusten käytön ja ylläpidon (Yleiset tietomallivaatimukset 2012. Esittely. RT 10-11080, 2).

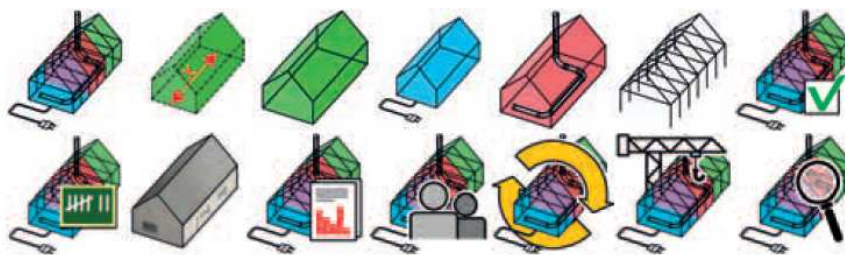
Yleisesti YTV:ssä mallinnukselle asetettuja tavoitteita ovat

- tukea hankkeen päätöksentekoprosesseja
- sitouttaa osapuolet hankkeen tavoitteisiin mallin avulla
- havainnollistaa suunnitteluratkaisuja
- auttaa suunnittelua ja suunnitelmien yhteensovittamista
- nostaa ja varmistaa rakennusprosessin ja lopputuotteen laatua
- tehostaa rakentamisaikaisia prosesseja
- parantaa turvallisuutta rakentamisen ja rakennuksen elinkaaren aikana
- tukea hankkeen kustannus- ja elinkaarianalyyssejä
- tukea hankkeen tietojen siirtämistä käytönaikaiseen tiedonhallintaan.

(Yleiset tietomallivaatimukset 2012. Esittely. RT 10-11080, 2.)

YTV on 14 osainen ja sisältää seuraavaa (Yleiset tietomallivaatimukset 2012. Esittely. RT 10-11080, 1):

- Osa 1. **Yleinen osuus**
- Osa 2. **Lähtötilanteen mallinnus**
- Osa 3. **Arkkitehtisuunnittelu**
- Osa 4. **Talotekninen suunnittelu**
- Osa 5. **Rakennesuunnittelu**
- Osa 6. **Laadunvarmistus**
- Osa 7. **Määrälaskenta**
- Osa 8. **Mallien käyttö havainnollistamisessa**
- Osa 9. **Mallien käyttö talotekniikan analyyseissä**
- Osa 10. **Energia-analyysit**
- Osa 11. **Tietomallipohjaisen projektin johtaminen**
- Osa 12. **Tietomallien hyödyntäminen rakennuksen käytön ja ylläpidon aikana**
- Osa 13. **Tietomallien hyödyntäminen rakentamisessa**
- Osa 14. **Tietomallien hyödyntäminen rakennusvalvonnassa.**



Kuva 7. YTV:n sisältö kuvina (Yleiset tietomallivaatimukset 2012. Esittely. RT 10-11080, 1)

Osat 1 ja 6 (yleinen osuus ja laadunvarmistus) tulisi kaikkien tietomallihankkeeseen osallistuvien osapuolten käydä vähintäänkin läpi oman alansa vaatimusten lisäksi. Yleisessä osuudessa kuvataan projekteissa käytettävän tietomallinnuksen perusteet, vaatimukset ja käsitteet. Siinä myös määritellään yleiset tavoitteet tietomallintamiselle. Laadunvarmistuksessa perehdytään menetelmään, jolla muissa osioissa kuvattu tietosisältö saadaan käyttökelpoiseen ja vaatimukset täyttävään muotoon. Tietomallin käyttökelpoisuus määräytyy pitkälti sen mukaan, kuinka tarkoituksenmukainen ja luotettava se on. (Yleiset tietomallivaatimukset 2012. Esittely. RT 10-11080, 1 - 3.)

2.4.3 BEC2012

BEC-projektin tarkoituksena on ollut yhtenäistää elementtirakenteiden suunnittelu-/mallinnuskäytäntöjä ja samalla kehittää mallintavan suunnittelun työkaluja. BECin mukaisesti mallinnetut kohteet helpottavat niiden hyödyntämistä suunnittelutoimistojen ulkopuolella. BEC-ohjeistuksen mukaisesti mallinnettu kohteen malli, piirustukset ja määräluettelot ovat elementtiteollisuuden toiveiden mukaisia, ja tämän vuoksi palvelevat teollisuuden tarpeita paremmin. (Salmela 2013-1.) BEC-projekti jatkuu myös vuonna 2014. Siinä on osallisena elementtiteollisuuden lisäksi suuria suunnittelutoimistoja kuten Ramboll Finland sekä Pöyry. (Elementtisuunnittelu.fi c.)

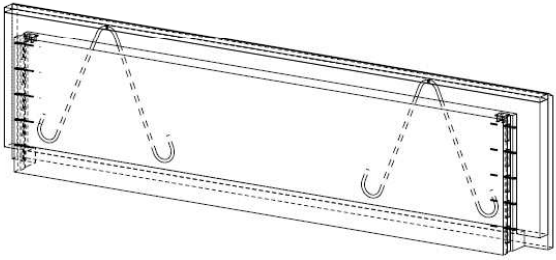
Elementtisuunnittelun sivuilla (Elementtisuunnittelu.fi c) kerrotaan, että BEC-projektissa on tähän mennessä luotu seuraavaa:

- Mallinnusohje
 - Tekla Structures -suomiympäristön päivitys yhteensopivaksi mallinnusohjeen kanssa
- Tyypielementtipiirustukset ja mallielementtien tietomallit
- Esimerkkiluettelot ja luettelo-ohje
 - Tekla Structures -suomiympäristön raporttipohjat
 - Seinien ja laattojen määrälaskentaohjeet
- Mallinnustyökalut
 - HI- ja I palkki- komponentit
 - Sokkelipalkkikomponentti
 - Nostolenkkityökalut
 - Ontelolaatan kololaattatyökalut
 - TT-laattaliitokset
 - HTT-laattakomponentti
 - Ontelolaatan rei'itysten tarkastustyökalu
 - Perustus/pilarikenkäliitos
 - Seinäkenkäliitos
 - Kahden seinän välinen liitos.

BEC2012-projektissa tuotetun elementtisuunnittelun mallinnusohjeen tarkoitus on määritellä YTV2012 rakennesuunnittelu-osan ohjeita tarkemmat vaatimukset betonielementtien mallinnukseen. Mallinnusohjeet siis täydentävät toisiaan elementtisuunnittelua suorittaessa. Elementtisuunnittelun mallinnusohjeiden tuomaa lisätarkkuutta tarvitaan, kun halutaan lukea tarkkaa ja tuotevalmistukseen soveltuvaa tietoa suoraan tuotannonohjausjärjestelmään. (Kautto 2012-1.)

RAUDOITELUETTELO															
RAUDOITTEET		D	L	AL	PAINO	TAVUTUSMITAT [mm]							KOMMENTTI		
TYYPIN NRO	LKM	LAATU	[mm]	[mm]	YHTI [kg]	a	b	c	d	e	u	v	x	TD	
A	1	5	A500HW	16	5400	42.7	5400								
A	2	6	A500HW	10	5400	20.0	5400								
D	3	27	B600KX	7	1370	11.2	500	410	500					42	
U	4	18	A500HW	8	3000	21.4	106	1270	180	1270	180				
D	5	10	A500HW	8	1130	4.5	500	180	500					50	
D	6	4	A500HW	12	2300	8.2	700	1270	400					80	
B	7	2	B600KX	11	6360	9.5	5872	520						90	
B	8	2	B600KX	11	1870	2.8	1382	520						90	
D	9	27	B600KX	7	970	7.9	300	410	300					42	
RAUDOITTEIDEN KOKONAISPAINO [kg]:													128.0		

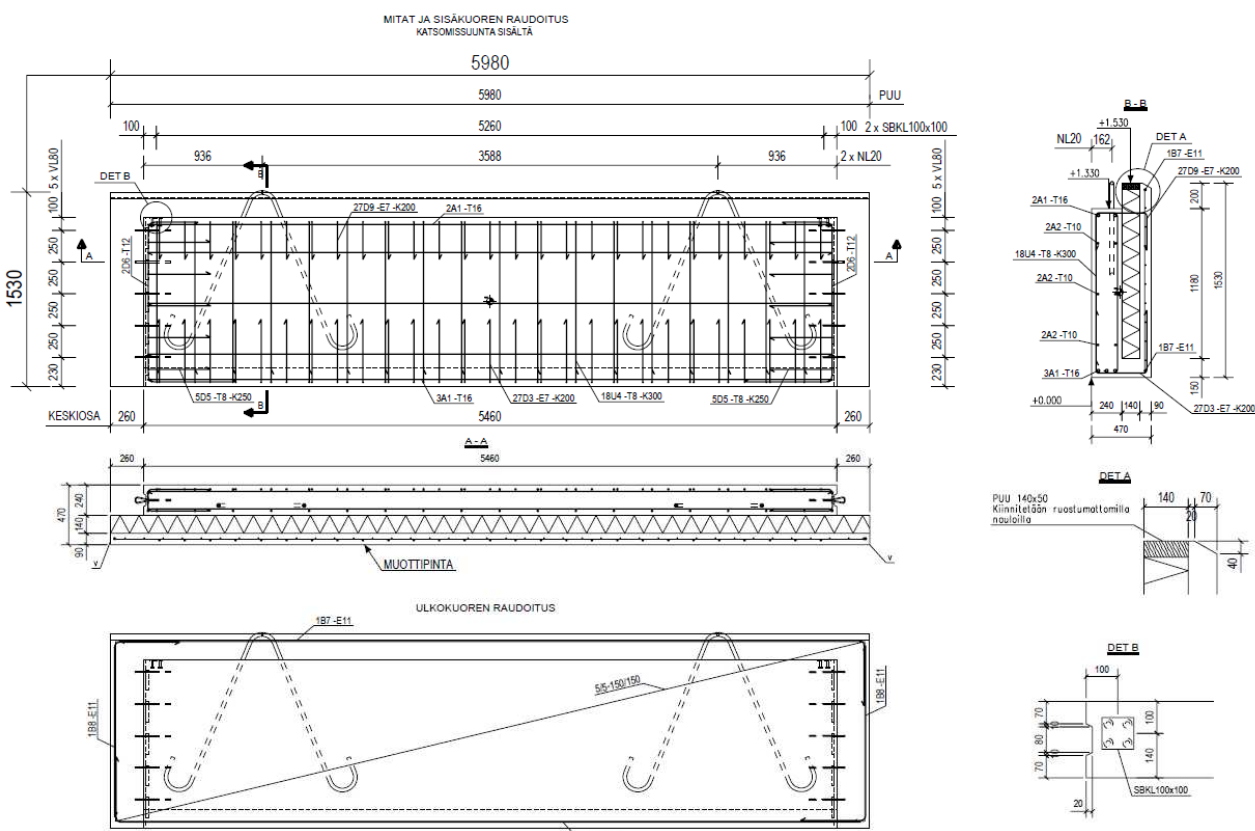
SUUNNITELUN LAHTOTIEDOT					
Rastitusluokka	XC2	Sisäkuori			
Rastitusluokka ulkokuori	XC34, XF1	Ulkokuori			
Suunnitella käyttökä	100v	Sisäkuori			
Suunn. käyttöikä ulkokuori	50v	Ulkokuori			
TUOTETIEDOT					
Betonipete 1	Nimellisavo 30mm	Sallittu mittapoikkeama 10mm			
Toleranssiluokka	Luokka N (2011)				
Pintakäsittely 1	Terdohierro				
Viistee 1	Kynätyöntösty merkityihin neuroihin				
VALUTARVIKELUETTELO					
PIIR. NUMERO	LKM	MATERIAALI	PINTA-ALA [m ²]	MAÄRA	YKS
AS-1	1	C30/37 C30/37 SÄÄNKESTÄVÄ		1.85 0.82	m ² m ²
				ELEMENTTI PAINO: 6.67 t	
MAÄRA TARVIKKEET					
2	kg	NL20			
2	kg	SBKL100x100			
10	kg	VL80			
5.98	m	PUU 50x140 PUU			
7.95	m ²	ERISTE SPU 140mm			
16.6	kg	ak-S150 B600KX			
25.8	kg	A500HW ø8			
20.0	kg	A500HW ø10			
8.2	kg	A500HW ø12			
42.7	kg	A500HW ø16			
19.1	kg	B600KX ø7			
12.3	kg	B600KX ø11			



MALLIPIIRUSTUS 28.2.2013

KOKO		PIIRITTEIDEN SISÄLTÖ		MITTAMAUK	
ELEMENTTIPIIRUSTUS		ELEMENTTIPIIRUSTUS		1:10	
AS-1, SOKKELELEMENTTI		AS-1, SOKKELELEMENTTI		1:25	
				1:30	
PIIRI	SIUN	TARK.	PIVI		
PIIRI	SIUN	TARK.	PIVI		
TYÖNUMERO		ALANUMERO		PIIR. NRO	
RAK		RAK		AS-1	
				MUTOS	

MITAT JA SISÄKUOREN RAUDOITUS
KATSOISSUUNTA SISÄLTÄ



ULKOKUOREN RAUDOITUS

KOKO		TYÖNUMERO		ALANUMERO		PIIR. NRO	
AS-1		RAK		RAK		AS-1	
						MUTOS	

Kuva 8. BECin mukainen kantavan sokkelelementin piirustus Tekla Structures -ohjelmistolla tehtynä (Elementti-suunnittelu.fi b)

3 TYÖKALUT KOMPONENTTIEN TEKEMISEEN

Tekla Structuresiin (lyhemmin TS) onnistuu parametrisoitujen komponenttien tekeminen pääsääntöisesti kahdella toisistaan suuresti eroavalla tavalla. Ensimmäinen tapa luoda parametrisoitu eli älykäs komponentti on käyttää hyväkseen TS:sta löytyvää Custom Component Editoria (lyhemmin CC Editor). CC Editorilla luodaan älykkäitä komponentteja enemmän visuaaliselta pohjalta eli komponentti on koko ajan nähtävissä, kun sitä muokataan ja parametrisoidaan. Toinen tapa komponentin tekemiseen on Tekla Open API -rajapinnan hyödyntäminen. Open API:n käyttämiseen tarvitaan Microsoftin Visual Studio ja mieluiten C#-koodikielen perusteiden hallinta, sillä komponentin rakentaminen tapahtuu puhtaasti ohjelmoimalla. Ohjelmoimalla luotua komponenttia kutsutaan nimellä Plugin ja CC Editorilla luotua komponenttia nimellä Custom Component, kummatkin komponentit otetaan käyttöön TS:n Component Catalogista.

Open API ja CC Editor ovat käytöltään hyvin erilaisia, joten se kumpaa tapaa käyttää komponentin tekemiseen, kannattaa harkita tapauskohtaisesti. Custom Componentit soveltuvat yksinkertaiseen ja nopeaan työskentelyyn. Kun CC Editorin käytön oppii, on helppoa tehdä yksinkertaisia parametrisoituja komponentteja vaikka projektin edetessä. Nämä komponentit voivat hyvinkin olla pelkästään projektikohtaisia ja niiden käyttöliittymän ei välttämättä tarvitse avautua muille kuin tekijälleen. Open APIlla komponenttien luominen on hieman hitaampaa, koska kaikki mallinnus ja parametrisointi tehdään koodin sisällä. Open API soveltuukin paremmin vaativampien ja monimutkaisempien komponenttien tekemiseen, joiden käyttäminen ei rajoitu vain yhteen projektiin.

3.1 Tekla Structures

Tekla Structures on Teklan kehittämä tietomallinnusohjelmisto. Sillä voidaan mallintaa kaikenlaisia rakenteita materiaalista riippumatta ja sisällyttää useita materiaaleja saman mallin alle. TS:sta on saatavana erilaisia ohjelmistokokoonpanoja rakentamisen eri toimialojen tarpeisiin.

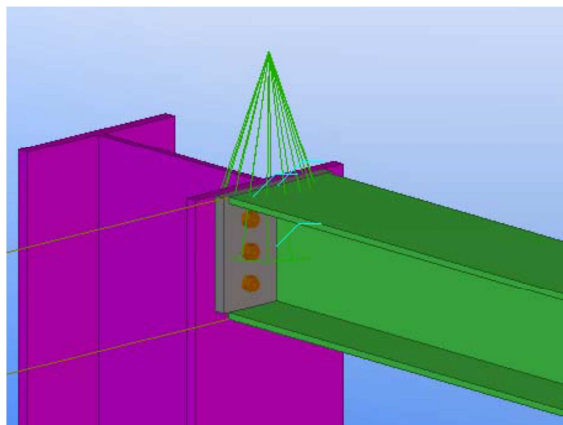
Tekla Structuresin ohjelmistokokoonpanot ovat tällä hetkellä seuraavat (Tekla.fi b):

- Tekla Structures **Full**
 - on kaiken kattava ohjelmistokokoonpano kaikkeen rakennesuunnitteluun ja rakentamisen hallintaan. Sillä voidaan luoda 3D-malleja teräs- ja betonirakenteista sekä tuottaa ja seurata tietoja luonnosvaiheesta valmistus-, pystytys- ja työmaanhjausvaiheisiin.
- Tekla Structures **Precast Concrete Detailing**
 - on betonielementtien suunnitteluun ja valmistukseen soveltuva kokoonpano. Sillä on käytännössä samat ominaisuudet kuin TS Full -kokoonpanolla, mutta ilman teräsrakenteiden yksityiskohtaista mallinnusta (esim. liitokset).
- Tekla Structures **Steel Detailing**
 - on teräsrakenteiden suunnitteluun soveltuva kokoonpano, joka sisältää teräsrakenteiden liitos ja konepajasuunnittelussa tarvittavat toiminnot. Se verrattavissa TS Full kokoonpanoon, mutta ilman betonirakenteiden yksityiskohtaista mallinnusta.
- Tekla Structures **Cast in Place**
 - on teräsbetonin ja paikallavalun suunnitteluun soveltuva kokoonpano. Se sisältää paikallavalun detaljointitoiminnot. Sillä voidaan luoda yksityiskohtaisia 3D-malleja betonirakenteista sekä tuottaa malleihin perustuvia tietoja valmistusta ja valua varten.
- Tekla Structures **Engineering**
 - on kokoonpano, joka mahdollistaa synkronoidun suunnittelun projektiosapuolten kesken.
- Tekla Structures **Primary**
 - sisältää saman kuin TS Full -ohjelmistokokoonpano, mutta mallin kokoa on rajoitettu. Se voi sisältää enintään 2500 osaa (pois lukien pultit ja mutterit) ja 5000 raudoitetta tai raudoiteryhmää.
- Tekla Structures **Viewer**
 - mahdollistaa mallinnetun rakenteen katselun ja raportoinnin. Se soveltuu rakenteen esittelyyn ja konsultointiin rakennusprojektin kaikissa vaiheissa.

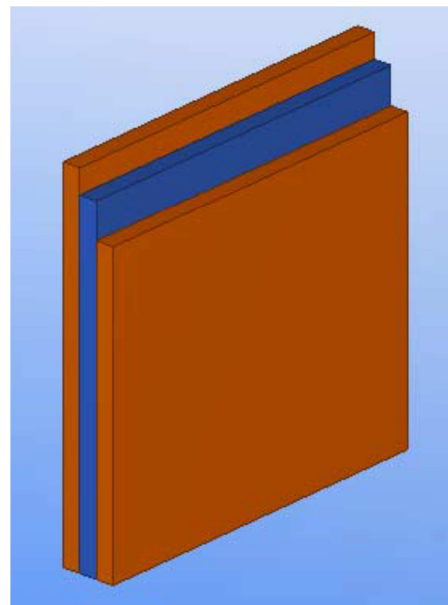
Tekla Structuresista on saatavilla nykyisin myös Full-versioita vastaava opiskelijaversio. Sen voi ladata ilmaiseksi neljän kuukauden kokeilujaksoksi Teklan sivuilta. (Tekla.fi a.) Opinnäytetyössä osa työskentelystä on tehty opiskelijaversiolla.

3.2 Custom Component Editor

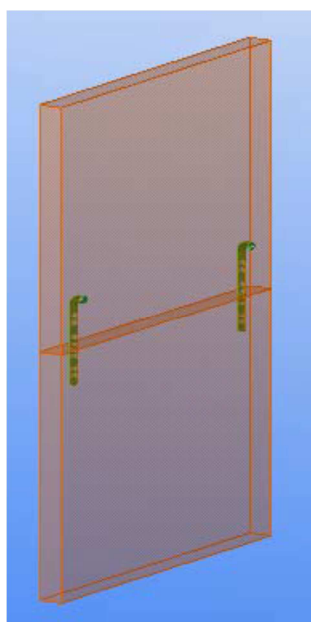
Teklan CC Editor mahdollistaa parametrien eli älyn lisäämisen luotuihin komponentteihin. Custom Component -tyyppejä löytyy neljä erilaista: **connections**, **parts**, **details** ja **seams**. Jokaiselle komponentille määritellään luomisvaiheessa vähintään tyyppi ja nimi. Komponentti voidaan luoda alusta asti mallintamalla tai räjäyttämällä jo olemassa oleva komponentti. Kaikkia osia sen ei tarvitse sisältää heti luomisvaiheessa, vaan siihen voidaan tehdä lisäyksiä vielä Custom Component Editorin puolella. (Extranet.tekla.com c.)



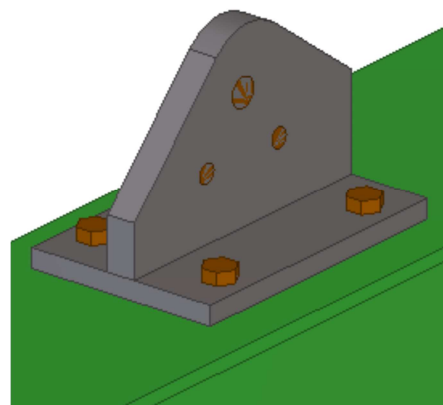
Kuva 9. Connection (Extranet.tekla.com c)



Kuva 10. Part (Extranet.tekla.com c)



Kuva 11. Seam (Extranet.tekla.com c)



Kuva 12. Detail (Extranet.tekla.com c)

Tekla Structures tekee komponenttiin sen luomisen yhteydessä Dialog Boxin eli käyttöliittymän, jota kautta komponenttia voi muokata. Dialogiin lisätään tietoja editorin puolella. Dialogin muokkaus onnistuu, joko suoraan INP-tiedostoa muuttamalla esimerkiksi MS:n NotePad-työkalulla tai Custom Component Dialog Editor -työkalulla, joka löytyy TS:n sisältä. INP-tiedostoa muokatessa ongelmana on, että aina kun tiedostoa muuttaa, joutuu avaamaan mallin uudestaan, jotta näkee miten valikot, kuvat ja tekstit asettuvat dialogiin. Dialog Box Editoria käytettäessä, dialogi on koko ajan näkyvässä, eikä muokkauksessa tarvitse kirjoittaa koodia.

CC Editorin käytön perusteiden oppiminen on suhteellisen helppoa. Työ vaikeutuu sitä mukaa, kun komponentista pyrkii tekemään monimutkaisemman ja lisäämään siihen paljon älyä. Tärkeintä on pitää komponentti mahdollisimman yksinkertaisena, mutta kuitenkin toimivana. Osa ominaisuuksista kannattaa lisätä alikomponenttina, jolloin itse komponentti säilyy parametriensa puolesta hieman yksinkertaisempuna ja toimii paremmin. Jos koko rakenne pyritään tekemään yhdellä komponentilla, tulee siihen ensinnäkin helposti suuri määrä toisiinsa viittaavia parametrejä, joiden hallinta on haastavaa tekijälle ja vielä haastavampaa muille. Tämä on ongelma, mikäli komponenttiin halutaan tehdä muutoksia tulevaisuudessa. Toinen ongelma on komponentin hitaus. Monimutkaiset parametrit voivat vaatia useita minutteja generointiaikaa sisäisten laskutoimitusten vuoksi. Kolmas ongelmakohta on komponentin sisäiset viittaukset eli ohjelma ei osaa päättää, mikä parametreistä oli ensin. Tämä aiheuttaa loop-tilanteita komponentissa sitä käytettäessä. (Patanen 2014-01-29.)

3.3 Tekla Open API

Open API (suomeksi "avoin sovellusten ohjelmointi rajapinta") helpottaa käyttäjän kommunikointia Tekla Structuresin ja muiden ohjelmien välillä. Tekla Open API, joka tunnetaan myös nimellä .NET API, tarjoaa rajapinnan, jolla kolmannen osapuolen sovellukset voivat olla vuorovaikutuksessa Tekla Structuresin mallin ja piirustusten osiin. Open API avaa Tekla Structuresin käyttäjille mahdollisuuden kehittää ohjelmaa tarkemmin omia tarpeitaan vastaavaksi. Open APIlla voidaan kehittää Tekla Structuresiin sovelluksia (applications) sekä laajennuksia (plug-ins). Applicationin ja pluginin keskeinen ero on niiden käyttämisessä. Application käynnistetään TS:n ulkopuolella, sitä ei siis voi suorittaa suoraan TS:ssa, ellei sitä ole sisällytetty makroon. Pluginit käynnistetään TS:n sisällä joko Component Catalogista, jos kyseessä mallissa käytettävä plugin, tai työkaluvalikosta, jos kyseessä piirustuksessa käytettävä plugin. Applicationit ovat yksinkertaisempia, joten niiden luominen ja testaaminen on huomattavasti helpompaa kuin pluginien. (Extranet.tekla.com b.)

Open API:n avulla voidaan siis luoda TS:n malliin plugin-komponentteja eli plugareita. Plugareita on pääasiassa kahta eri tyyppiä: PluginBase- ja ConnectionBase-komponentit. PluginBase on yleinen komponentti (esimerkiksi sandwich-elementti), jonka ei tarvitse olla sidoksissa muihin osiin, vaan se voidaan laittaa mihin paikkaan mallissa vain. ConnectionBase on joko liitos (esimerkiksi pilari-palkki-liitos) tai detalji (esimerkiksi palkin nostolenkki). (Extranet.tekla.com a.) Ohjelmoimalla tehdyt plugarit ovat toiminnaltaan samanlaisia kuin Custom Component Editorilla tehdyt komponentit.

Tekla Open API:a voidaan käyttää joko C#- tai VB-kielellä. Koodin kirjoittaminen onnistuu esimerkiksi Microsoftin Visual Studiolla. Open API:n käyttäminen vaatii siis jonkinlaisen pohjan ohjelmoinnin maailmasta, ilman sitä on sen käytön opettelu varsin haastavaa. Komponenttien luomiseen Open APIlla suositellaan C#-koodikieltä, sillä sitä käytetään yleisemmin, joten sen käyttöön löytyy apua paremmin. Ohjeita Tekla Open API:n käyttöön löytyy esimerkiksi Teklan Extranetistä sekä Teklan Open API -keskustelupalstoilta. (Extranet.tekla.com b.)

4 TIETOMALLIVAATIMUSTEN MUKAISET KOMPONENTIT

4.1 Kehitysprojektit tietomallinnettujen komponenttien tukena

Mikäli komponenteissa pyritään noudattamaan kehitysprojektien (YTV2012 ja BEC2012) asettamia vaatimuksia, tulisi niiden sisältää runsaasti erilaista tietoa. Osa tiedosta on kuitenkin liikaa alun suunnitteluvaiheissa, joissa tietomallintamista ei tarvitse toteuttaa suurella tarkkuudella. Komponenteissa olisi hyvä olla mahdollisuus jättää mallintamatta ylimääräistä tietoa suunnittelun alkuvaiheissa.

Rakennesuunnittelijalle tulee ensi kertaa mallinnusvaatimuksia vasta yleissuunnitteluvaiheessa, tällöin rakennesuunnittelija mitoittaa rakenteet ja mallintaa ne kohteen rakennemalliin. Lähtötietona rakennesuunnittelijan mitoitukselle toimii arkkitehdin suunnittelema ja mallintama tietomalli. Yleissuunnitteluvaiheessa esimerkiksi sandwich-tyylisen elementin ei tarvitse sisältää muuta kuin ulko- ja sisäkuoren, eristeen sekä näiden osien paksuudet ja elementin sijoittumisen mallissa. Tämän tarkempaa mallintamista ei ole syytä toteuttaa vielä tässä vaiheessa, sillä rakennemallia ei tarvitse käyttää muualla kuin alustavassa suunnitelmien yhteensovittamisessa esimerkiksi talotekniikan suunnitelmien kanssa. (Yleiset tietomallivaatimukset 2012. Osa 1. Yleinen osuus. RT 10-11066, 8.)

Hankintoja palvelevassa suunnitteluvaiheessa, rakennemallista tulisi saada irti kantavien ja eikantavien betonirakenteiden koko, laajuus, määrät sekä tarkka sijainti. Tässä vaiheessa riittää siis yleissuunnitteluvaiheessa käytetty tarkkuus kaikkialla muualla paitsi mallielementeissä. Mallielementit tulee tässä vaiheessa mallintaa lopulliseen tarkkuuteensa, joka vastaa toteutussuunnittelun tarkkuustasoa. Sandwich-tyylisen mallielementin tulisi siis tässä vaiheessa sisältää raudoitukset ja nostolenkit sekä muut lopullisessa elementissäkin olevat tiedot. (Yleiset tietomallivaatimukset 2012. Osa 5. Rakennesuunnittelu. RT 10-11070, 5.)

Toteutussuunnitteluvaiheessa rakennemalli täydennetään lopulliseen tarkkuuteensa eli tässä vaiheessa rakenteet mallinnetaan sillä tarkkuudella, millä ne toteutetaan käytännössä. Mallintamisessa hyödynnetään talotekniikan suunnitelmia, jotta nähdään esimerkiksi, missä kohdissa rakenteissa on reikävarauksia. Sandwich-tyylisen elementin tulee sisältää tässä vaiheessa siis kaikki tarvittava tieto, jota tarvitaan sen tekemiseen. (Yleiset tietomallivaatimukset 2012. Osa 5. Rakennesuunnittelu. RT 10-11070, 8.)

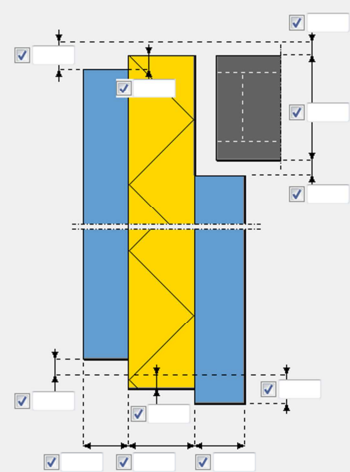
Toteutussuunnitteluvaiheessa rakenteihin lisätään mallissa kaikki tarvittava tieto, joko mallintamalla tai tekstitietona. Sandwich-tyyliseen elementtiin lisätään tässä vaiheessa tarkka geometria perusviisteitä lukuun ottamatta, tieto perusviisteistä täytyy kuitenkin lisätä elementtiin tekstitietona. Ansaat voidaan joko mallintaa tai lisätä nekin tekstitietona. (Elementtisuunnittelu.fi a.) Tärkeintä on, että kaikki tarvittava tieto löytyy elementistä itsestään, eikä piirustusvaiheessa tarvitse lisätä enää erikseen mitään. Tekstitiedon määrä riippuu mallinnettavasta rakenteesta. Kuvassa 13 on Tekla Structures -ohjelmistosta otetut kuvaruutukaappaukset sandwich-elementti komponentin dialogista, niistä näkee hyvin, mitä kaikkea tietoa kyseisen komponentin tulisi ainakin pitää sisällään.

Tekla Structures Bearing_sandwich (1)

Save Load standard Save as standard Help...

modify connection type

Lähtötiedot Tuotetiedot Position



Elementin korkeus

Ulkokuoren betoni

Eriste

Sisäkuoren betoni

Elementtitunnus

Alottava numero

Sisäkuoren nimi

Eristeen nimi

Ulkokuoren nimi

Suunnittelukäyttöikä

Rakenneluokka

Rasitusluokka

Paloluokka

Elementtityyppi

Nimi

Tuoteryhmä

Luokka

Lohko

Alalohko

Kerros

OK Apply Modify Get F / Γ Cancel

Tekla Structures Bearing_sandwich (1)

Save Load standard Save as standard Help...

modify connection type

Lähtötiedot Tuotetiedot Position

Betonipeite 1 (mm)

Betonipeite 2 (mm)

Maksimiraakoko (mm)

Mittatoleranssi (mm)

Toleranssiluokka

Pintakäsittelyluokka

Pintakäsittely 1

Pintakäsittely 2

Viisteet 1

Viisteet 2

Muotistanostolujuus

Kuljetus- ja asennuslujuus

Pakkasenkestolujuus

Suojahuokossuhde

Kuvaus

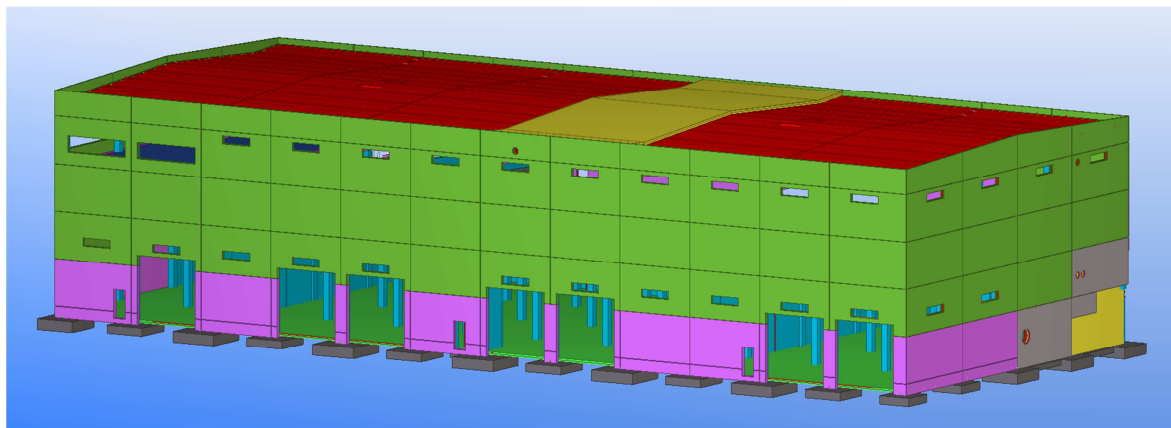
Tuotekoodi

OK Apply Modify Get F / Γ Cancel

Kuva 13. Kantavan sandwich-elementin käyttöliittymä (Kuvaruutukaappaus Tekla Structures -ohjelmistosta)

4.2 Esimerkkikohteessa tarvittavat komponentit

ConnArin projekteista suurin osa sijoittuu teollisuusrakentamiseen. Komponentit, joita esimerkiksi hallirakennuksissa tarvitaan, ovat pitkälti samanlaisia projektista toiseen. Opinnäytetyötä tehdessä pyritään toteuttamaan komponentteja, joille yrityksen työntekijöillä on eniten kysyntää. Esimerkkikohteena käytetään betonirakenteista teollisuushallia.



Kuva 14. Vastaanotto- ja käsittelyhalli (Insinööritoimisto ConnAri Oy)

Ulkokuori

Esimerkkikohteessa oli käytetty seinien mallintamisessa muun muassa kuorielementtejä, maanpainesieniä, ei-kantavia sokkelielementtejä ja ei kantavia sandwich-elementtejä. Sandwich-elementti esimerkiksi tarvitsee suuren joukon komponentteja nostamista, kiinnittymistä ja raudoitusta ajatellen. Sandwich-elementti oli itsessään komponentti, joka sisälsi sisä- ja ulkokuoren sekä eristeen. Sen tietoja pystyttiin muokkaamaan helposti valikon kautta. Rauditus, nostolenkit ja elementin kiinnittyminen viereisiin elementteihin tai esimerkiksi pilariin, toteutettiin kuhunkin työhön soveltuvalla komponentilla. Joissain tapauksissa elementteihin tuli myös aukotuksia ikkunoiden tai ovien vuoksi, tällöin tarvittiin komponentti myös niiden luomiseen. Rakennuksen kulmat vaativat myös erillisen komponentin sandwich-elementin leikkaamiseen ja sovittamiseen. Leikkaus onnistuisi kyllä tavan työkalujakin käyttämällä, mutta kohteessa jossa on paljon kulmia, on se turhan työlästä.

Kantavat rakenteet

Rakennuksen runko koostui suorakulmaisista pilareista ja pilarianturoista, palkeista, ontelolaatoista sekä katon TT-laatoista. Pilari esimerkiksi tarvitsee komponentteja nostamista varten, anturaan kiinnittymiseen, raudoitukseen sekä palkkien kannattamiseen. Pilari itse voidaan toteuttaa joko komponenttina tai TS:n työkaluilla. Esimerkkikohteessa pilarin kiinnittyminen anturaan toteutettiin siihen soveltuvalla komponentilla, samoin tehtiin nostolenkkien sekä konsolien kanssa. Anturan, pilarin ja konsolien raudoittamiseen tarvittiin kaikkiin erilainen komponentti. Kohteessa käytettiin kahdenlaisia palkkeja. Leukapalkkeja käytiin yhdessä ontelolaattojen kanssa ja harjapalkkeja katolla yhdessä TT-laattojen kanssa. Leukapalkit oli mallinnettu TS:n omalla työkalulla, mutta harjapalkit jouduttu tekemään komponentilla. TT-laatat tarvitsivat kahdenlaisia komponentteja toisiinsa kiinnittymiseen, riippuen siitä, tapahtuiko kiinnitys pitkältä vai lyhyeltä sivulta.

Muut

Teollisuusrakennukseen tulee usein mallintaa myös suuri määrä muita täydentäviä osia kuten tikkaat, kaiteita ja portaita. Näiden mallinnus on erittäin työlästä ilman kunnollista komponenttia. TS:n sisältä löytyykin hyvät välineet edellisten luomiseen. Esimerkimmallista löytyi edellisten lisäksi myös oviaukonkynnyksen mallintamiseen soveltuva komponentti.

Komponentteja yhteensä

Laskujen mukaan esimerkikohteen kaltaiseen hallirakennukseen tarvitaan TS:n perustyökalujen lisäksi noin 25–30 erilaista komponenttia ja mikäli mukaan otetaan vielä raudoitus, nousee erilaisten komponenttien määrä noin kymmenellä. Komponenttien määrä vaihtelee suuresti riippuen kohteesta ja siitä, kuinka tarkasti pyritään mallintamaan, joten edellä mainitut luvut eivät välttämättä ole vertailukelpoisia muiden kohteiden kanssa. Epätarkkuutta aiheuttaa myös se, että osa komponenteista oli räjäytetty osiin, kun ne oli saatu lopulliseen muotoonsa. Räjäyttämällä varmistetaan, että komponentti ei vahingossa muokkaa itseään.

4.3 Tehdyt komponentit

Esimerkkikohteessa olleet ongelmat puuttuvien tai puutteellisten komponenttien kanssa ovat toistuneet muissakin projekteissa. Kokonaan on puuttunut sokkelielementin alapään kiinnitykseen tarvittava komponentti. Pieniä lisäyksiä on kaivattu sokkelielementin mallintavaan komponenttiin. Sokkelielementin kulmaliitos on myös jouduttu toteuttamaan TS:n työkaluilla, joka on osaltaan lisännyt työtaakkaa. Opinnäytetyön ohessa tehdään kolme komponenttia yrityksen käyttöön. Komponentit luodaan Custom Component Editoria käyttämällä.

4.3.1 Sokkelielementti

Tekla Structuresista löytyy itsessään sokkelielementti komponentti, mutta siitä puuttuu joitain toimintoja, joita kaivataan. Tarkoituksena on kehittää TS:n komponenttia, pidemmälle luomalla kokonaan uusi yrityksen käyttöön räätälöity komponentti.

Sokkelielementti komponentille määritellään parametrit CC Editoria käyttäen. Yrityksen toiveissa oli, että komponentti sisältäisi seuraavat osat: sisäkuori, keskiosa, eriste, ulkokuori, ulkokuoren yläpään viiste, reunapuu yläpähän, eristekolat, siteitä kuorien välille sekä raudoitus.

Ulkokuoren korkeuden tulisi olla erikseen muutettavissa, muuten elementillä saisi olla sama korkeus kaikkialla. Sisä- ja ulkokuoren paksuuden tulisi olla muokattavissa, kuten myös eristeen paksuuden. Reunapuun leveys on aina sama kuin eristeellä, mutta paksuuden tulee olla muokattavissa, lisäksi tulee voida valita, laitetaanko puuta yläpähän vai ei. Eristekolojen sijainti, määrä ja koko tulee olla muokattavissa. Siteiden sijaintia ja määrää tulee myös voida muuttaa. Raudoitus tulisi voida määrittää sisä- ja ulkokuoreen erikseen. Kaikkien materiaalien tulee olla muokattavissa.

Komponentin rakentaminen aloitetaan tyhjästä. Ensiksi mallinnetaan sisäkuori, eriste, keskiosa ja ulkokuori. Tämän jälkeen osista luodaan komponentti ja jatketaan muokkausta editorin puolella. Reunapuun mallinnetaan normaalisti elementin yläpään. Eristekolot ja siteet toteutetaan mallintamalla sekä Array-komponentilla. Rauditus liitetään sisä- ja ulkokuoreen alikomponenttina.

4.3.2 Sokkelielementin alapään kiinnitys

Tekla Structuresista ei löydy sokkelielementin alapään kiinnityskomponenttia, yläpään sellainen on kylläkin mahdollista saada. Alapään kiinnityskomponentti on hyvin tarpeellinen osa ja sille on suurta kysyntää yrityksen sisällä. Aikaisemmin yläpään kiinnityskomponentti on jouduttu räjäyttämään ja muokkaamaan siitä sopiva alapään. Tämä on ollut hidasta ja se on jouduttu toteuttamaan jokaisessa projektissa aina uudelleen.

Tarkoituksena on räjäyttää sokkelielementin yläpään kiinnityskomponentti ja muokata siitä sopiva alapään. Parametrien lisäys tapahtuu CC Editorissa. Komponentin tulisi sisältää seuraavat osat: mutteri, aluslevy, pultti ja sisäkierrehylsy.

Mutterin materiaalin tulisi olla muokattavissa. Aluslevyn profiilia, paksuutta ja materiaalia tulisi voida muokata. Pultin ylitys mutterista sekä sen materiaali tulee olla muokattavissa. Pilariin kiinnittyvän sisäkierrehylsyn tulisi olla muokattavissa erilliseksi tai se tulisi voida jättää kokonaan pois ja laittaa tilalle pultin kiinnittyminen suoraan pilariin. Sen mukaan miten pilari valetaan, voi alapään kiinnitys olla kahdenlainen. Pultti lisätään joko suoraan pilariin, jolloin muottipinta ei voi olla tappia vasten, tai kiinnitetään valussa sisäkierrehylsy pilariin, jolloin ei ole väliä millä puolella muottipinta on.

Sokkelielementin alapään kiinnityskomponentti luodaan räjäyttämällä sokkelielementin yläpään kiinnityskomponentti ja muokkaamalla sitä omiin tarpeisiin sopivaksi. Mallin puolella tehdään valmiiksi mutteri, aluslevy ja pultti sekä sisäkierrehylsy, joka toimii alikomponenttina lopullisessa komponentissa. Seuraavaksi osista luodaan komponentti ja aloitetaan parametrusointi CC Editorin puolella. Ohjeissa on toteutettu yksinkertaistettu versio alapään kiinnityskomponentista.

4.3.3 Sokkelielementin nurkkaliitos

Viimeisenä komponenttina toteutetaan sokkelielementin nurkkaliitos. Komponentti on melko yksinkertainen ja sen osat koostuvat vain leikkauksista. Komponenttiin täytyy lisätä jonkin verran älyä, jotta se esimerkiksi tunnistaa, kuinka paksu sokkelielementin on kyseessä. Liitos tehdään toimivaksi vain 90-asteen nurkissa.

5 TULOKSET

Opinnäytetyön tarkoituksena oli tutkia työkaluja, joilla Tekla Structures -ohjelmistoon voidaan luoda parametrisoituja komponentteja. Tarkoitus oli myös tehdä komponentteja opinnäytetyön tilanneen yrityksen käyttöön sekä luoda ohjeet komponenttien tekemiseen Custom Component Editorilla.

Opinnäytetyössä selviää, että komponentit voidaan tehdä kahdella toisistaan hyvin paljon eroavalla tavalla. Ensimmäisellä tavalla komponentit tehdään Tekla Structures -ohjelmistosta löytyvällä Custom Component Editorilla. Toinen tapa tehdä komponentteja on käyttää hyväksi Tekla Open API -rajapintaa ja Microsoft Visual Studiota. Ensimmäinen tapa soveltuu yksinkertaisiin ja toinen tapa monimutkaisiin komponentteihin.

Yrityksen käyttöön tehtiin kolme komponenttia Custom Component Editorilla. Tehdyt komponentit ovat sokkelielementti ja kaksi hieman toisistaan eroavaa kiinnikettä. Alun perin oli tarkoitus tehdä myös nurkkaliitoksen luova komponentti, mutta siirtyminen uudempaan Tekla Structures -versioon poisti kysynnän tälle kyseiselle komponentille.

Opinnäytetyön tuloksena saatiin myös liitteenä olevat ohjeet, joiden avulla voidaan oppia perusteet Custom Component Editorista. Lisäksi saatiin myös pienet ohjeet yksinkertaisen applikaation tekemiselle Microsoft Visual Studiolla Tekla Open APIa hyväksi käyttäen.

6 POHDINTA

6.1 Komponenttien ja ohjeiden tekeminen

Komponenttien tekeminen oli alussa uuvuttavaa työtä. Komponentteja voidaan parametrisoida monella eri tapaa ja monta kertaa työtä tehdessäni huomasin jälkeenkäin, että saman asian olisi voinut tehdä huomattavasti yksinkertaisemmin ja toimivammin. Sokkelielementti komponentin tein useaan otteeseen, ensimmäiset kerrat menivät täysin opetellessa työkalun käyttöä. Komponentit kannattaa todellakin suunnitella perusteellisesti ennen kuin niitä alkaa rakentaa. Suunnitteluvaiheessa tulisi myös ottaa huomioon, mihin tasoihin osien handle-pisteet kiinnitetään ja mitä ongelmia tiettyihin osiin kiinnittämisen seurauksena voi tulla. Perusteellinen suunnittelu vähentää komponentin tekemisen aikana tulevia korjaus- ja muutostarpeita ja näin ollen nopeuttaa sen luomisprosessia.

Kaikkiin komponentteihin tuli joitain muutoksia, joita ei osannut vielä alun suunnitteluvaiheissa ottaa huomioon. Sokkelielementin raudoitus jätettiin kokonaan lisäämättä komponenttiin, sillä se olisi tehnyt komponentista huomattavasti monimutkaisemman. Kuorien raudoitukseen löytyy kelpo komponentti, jolla raudoitus hoituu jälkeenkäinkin nopeasti. Sokkelielementin alapään kiinnityskomponentista tehtiin kaksi erilaista versiota, ensimmäisessä on sisäkierrähelys ja toisessa suoraan pilariin kiinnittyvä pultti. Alun perin tarkoitus oli luoda kumpikin samaan komponenttiin, mutta lopussa huomattiin, että syntyi ongelmia kokoonpanojen kanssa. Sisäkierrähelys sai kyllä helposti kiinnitettyä pilarin kokoonpanoon, mutta kuinka pultin liittää siihen niin, että se on osana pilarin kokoonpanoa vain silloin, kun sisäkierrähelys ei laiteta. Sokkelielementin nurkkaliitosta ei loppujen lopuksi toteutettu ollenkaan, sillä päivitys uudempaan Tekla Structures 20.0 -versioon toi mukanaan uuden työkalun, jolla osien pintojen venyttäminen ja lyhentäminen käy vaivattomasti.

Komponenttien tekeminen onnistui loppujen lopuksi hyvin. Tässä vaiheessa ei tietenkään vielä tiedä, kuinka paljon ongelmia niiden käyttämisessä tulee vielä olemaan, koska testaamista ei ole suoritettu perusteellisesti. Komponentteja parannetaan sitä mukaan, kun niissä havaitaan kehitettävää/puutteita toiminnallisuudessa.

Opinnäytetyön ohessa luodut ohjeet komponenttien tekemiseen Custom Component Editorilla oli alun perin tarkoitus tehdä sokkelielementin alapään kiinnityskomponentin pohjalta, mutta koska komponentin parametrisoinnin loppuvaiheessa Variables-tila näytti niin sekavalta, päätin tehdä kokonaan uuden esimerkki komponentin. Ohjeissa esitellään tarkasti, kuinka tehdä yksinkertainen kiinnityskomponentti. Varsinaista käyttötarkoitusta ohjeiden komponentille ei sellaisenaan ole, mutta ohjeiden pohjalta kukin pystyy toteuttamaan omia monimutkaisempia komponentteja, joista voidaan saada käytännön hyötyäkin. Tekla Open APIlla ei tehty komponentteja, vaan pelkästään yksinkertainen sovellus, jonka tekemiseen on pienet ohjeet myös. Jotta Open APIlla olisi onnistunut toteuttamaan samantasoisia komponentteja kuin Custom Component Editorilla, olisi työtunteja joutunut käyttämään runsaasti enemmän sen opiskeluun.

6.2 Komponenttityökalujen tarjoamat haasteet

Olen itse saanut hyvän pohjan Tekla Structures -ohjelmistolla työskentelyyn viime vuoden harjoittelun muodossa, ilman sitä olisi työn vaikeusaste lisääntynyt huomattavasti. Mikäli aikoo lähteä kehittämään komponenttia Custom Component Editorilla, suosittelisin opettelemaan ainakin perusteet TS:sta. Jos taas aikoo kehittää komponentit Tekla Open APIa käyttäen, mitä Tekla tuntuu suosittavan CC Editorin sijaan, olisi hyvä käydä jonkinlainen koulutus C#-koodikielestä, sillä itseisopiskeluna sen opettelussa on oma työnsä. Open APIa varten on myös hyvä opetella perusteita Tekla Structurista, jotta saa ymmärrystä siihen, miten se toimii ja mitä kaikkea sillä voidaan ylipäätään tehdä.

CC Editorin kanssa haasteita tuotti Variables-aulukon Formula-kohtaan kirjoitettavat kaavat. Niissä sai todella olla tarkkana, sillä yksikin välilyönti väärässä paikkaa saattoi tarkoittaa sitä, että kaava ei toiminut enää ollenkaan. Ongelmia syntyi myös handle-pisteiden vääriin tasoihin kiinnittämisestä. Kun alussa ei vielä tiennyt komponentin rakentamisen vaiheita, oli vaikea miettiä komponenttia alusta loppuun ja sen vuoksi kiinnityksiä joutui monesti muuntelemaan jälkeenpäin. Tällainen korjaaminen heikentää helposti Variables-aulukon luettavuutta, joka on hyvin keskeinen asia, kun tehdään haastavia ja runsaasti parametrejä sisältäviä komponentteja.

Tekla Open API:n suhteen suurin haaste oli täysi ymmärtämättömyys ohjelmointia ja C#-koodikieltä kohtaan. Kevään aikana suoritettujen tuntien ohjelmoinnin perusteiden parissa auttoivat ymmärtämään Open APIa paremmin, mutta paljon opiskeltavaa jäi vielä jäljelle, koodikieli kun on vasta pieni osa kokonaisuutta.

Tekla Open API:n hallinta vaatii enemmän opiskelua kuin Custom Component Editor. Mikäli aikoo opetella vain toisen tavan, on syytä huomioida käyttötarkoitus. Jos tarkoituksena on oppia luomaan vain pieniä ja yksinkertaisia komponentteja, joiden tekemiseen menee joitain tunteja, on valinta silloin Custom Component Editor. Jos taas tarkoituksena on luoda erittäin monimutkaisia komponentteja ja kenties muitakin laajennuksia Tekla Structures -ohjelmistoon, on valinta silloin Tekla Open API.

6.3 Opinnäytetyön aikaansaamat hyödyt itselle ja tilaajalle

Opinnäytetyöprosessi on laajentanut näkemystäni tietomallintamisesta ja sen hyödyntämisestä rakennusalalla. Olen oppinut käyttämään komponenttien parametrisointiin käytettävää Custom Component Editoria mielestäni varsin monipuolisesti, mutta paljon opittavaa on vielä. Opinnäytetyön aikana eksyin myös opettelemaan ohjelmointia, mikä auttoi ymmärtämään paremmin käytettävää Tekla Structures -ohjelmistoa sekä tietenkin Tekla Open API -rajapintaa. Vaikkakin Open API:n käsittely jäi opinnäytetyössä vähälle, sain hyvän pohjan jatko-opiskelulle sen parissa.

Yritys ottaa hiljalleen komponenttini käyttöönsä ja tilauksia muillekin komponenteille on jo tullut. Jää nähtäväksi, kuinka hyvin tekemäni ohjeet toimivat opettamisessa käytännössä. Ohjeiden tueksi on tarkoitus järjestää myös muuta koulutusta.

LÄHTEET

- Connari.fi [verkkoaineisto]. [viitattu 17.3.2014] Saatavissa: <http://www.connari.fi>
- EASTMAN, Chuck, TEICHOLZ, Paul, SACKS, Rafael ja LISTON, Kathleen. 2008. BIM Handbook. Hoboken: John Wiley & Sons, Inc.
- Elementtisuunnittelu.fi a [verkkoaineisto]. [viitattu 2014-03-25] Saatavissa: <http://www.elementtisuunnittelu.fi/fi/suunnitteluprosessi/mallintava-suunnittelu>. BEC2012 Elementtisuunnittelun mallinnusohje_103
- Elementtisuunnittelu.fi b [verkkoaineisto]. [viitattu 2014-03-25] Saatavissa: <http://www.elementtisuunnittelu.fi/fi/suunnitteluprosessi/mallipiirustukset>. AS-01 Sokkeli.
- Elementtisuunnittelu.fi c [verkkoaineisto]. [viitattu 2014-03-25] Saatavissa: <http://www.elementtisuunnittelu.fi/fi/uutiset/2013/11/25/bec-projektissa-tuotettu-runsaasti-uusia-tyokaluja>
- Extranet.tekla.com a [verkkoaineisto]. [viitattu 2014-03-17] Saatavissa: <https://extranet.tekla.com>. Tekla Open API: Creating Model Plugins.
- Extranet.tekla.com b [verkkoaineisto]. [viitattu 2014-03-17] Saatavissa: <https://extranet.tekla.com>. Tekla Open API Developer's Guide.
- Extranet.tekla.com c [verkkoaineisto]. [viitattu 2014-03-17] Saatavissa: <https://extranet.tekla.com>. Tekla Structures Custom Components.
- KAUTTO, Tero 2012-1. BEC2012-Elementtisuunnittelun mallinnusohje. YTV2012 osa5: Rakennesuunnittelu. Betoni. [viitattu 2014-03-17]. Saatavissa: <http://www.betoni.com/betoni-lehti/arkisto/2012-1>
- LAITINEN, Jarmo 2012-05-30. Professori. Lean ja BIM yhdessä tuottavat lisäarvoa [luentomateriaali]. Saatavissa: http://www.ril.fi/media/files/tietomallit/ril_laitinen.pdf
- NSSOY. TATE2012 VIHDOIN VALMIINA? [verkkoaineisto]. [viitattu 17.3.2014]. Saatavissa: <http://nssoy.fi/uploads/TATE12vihdoinkinvalmiina.pdf>
- PATANEN, Jari 2014-01-29. Opinnäytetyö Teklan komponenteista [sähköpostiviesti]. Vastaanottaja Kimmo Kuusela.
- Rakli. Uudistamisen periaatteet ja tehtäväluetteloiden käyttö [verkkoaineisto]. [viitattu 2014-02-12]. Saatavissa: <http://www.rakli.fi/media/rakennuttaminen/telu-periaatteet-12022014.pdf>
- SALMELA, Pasi 2013-1. BEC-luettelot tietomallista. [viitattu 2014-03-25] Saatavissa: <http://www.betoni.com/betoni-lehti/arkisto/2013-1>
- TEHTÄVÄLUETTELOT. KÄYTTÖOHJE KO12. RT 10-11105. [online]. Helsinki: Rakennustieto [viitattu 2014-03-17]. Saatavissa: <https://www.rakennustieto.fi>
- Tekla.fi a [verkkoaineisto]. [viitattu 2014-03-17] Saatavissa: <http://www.tekla.com/fi/ratkaisut/opiskelijat>
- Tekla.fi b [verkkoaineisto]. [viitattu 2014-03-17] Saatavissa: <http://www.tekla.com/fi/tekla-structures-ohjelmistokokoonpanot>
- TIETOMALLINNETTAVA RAKENNUSHANKE. OHJEITA RAKENNUTTAJALLE. RT 10-10992. [online]. Helsinki: Rakennustieto [viitattu 2014-03-17]. Saatavissa: <https://www.rakennustieto.fi>
- Wspgroup.com a [verkkoaineisto]. [viitattu 2014-03-20] Saatavissa: <http://www.wspgroup.com/en/wsp-group-bim/BIM-around-the-world/>
- Wspgroup.com b [verkkoaineisto]. [viitattu 2014-03-20] Saatavissa: <http://www.wspgroup.com/en/wsp-group-bim/BIM-around-the-world/Finland/>

YLEISET TIETOMALLIVAATIMUKSET 2012. ESITTELY. RT 10-11080. [online]. Helsinki. Rakennustieto [viitattu 2014-03-17]. Saatavissa: <https://www.rakennustieto.fi>

YLEISET TIETOMALLIVAATIMUKSET 2012. OSA 1. YLEINEN OSUUS. RT 10-11066. [online]. Helsinki: Rakennustieto [viitattu 2014-03-17]. Saatavissa: <https://www.rakennustieto.fi>

YLEISET TIETOMALLIVAATIMUKSET 2012. OSA 5. RAKENNESUUNNITTELU. RT 10-11070. [online]. Helsinki: Rakennustieto [viitattu 2014-03-17]. Saatavissa: <https://www.rakennustieto.fi>

YLEISET TIETOMALLIVAATIMUKSET 2012. OSA 6. LAADUNVARMISTUS. RT 10-11071. [online]. Helsinki: Rakennustieto [viitattu 2014-03-24]. Saatavissa: <https://www.rakennustieto.fi>

YLEISET TIETOMALLIVAATIMUKSET 2012. OSA 8. MALLIEN KÄYTTÖ HAVAINNOLLISTAMISESSA. RT 10-11073. [online]. Helsinki: Rakennustieto [viitattu 2014-03-25]. Saatavissa: <https://www.rakennustieto.fi>

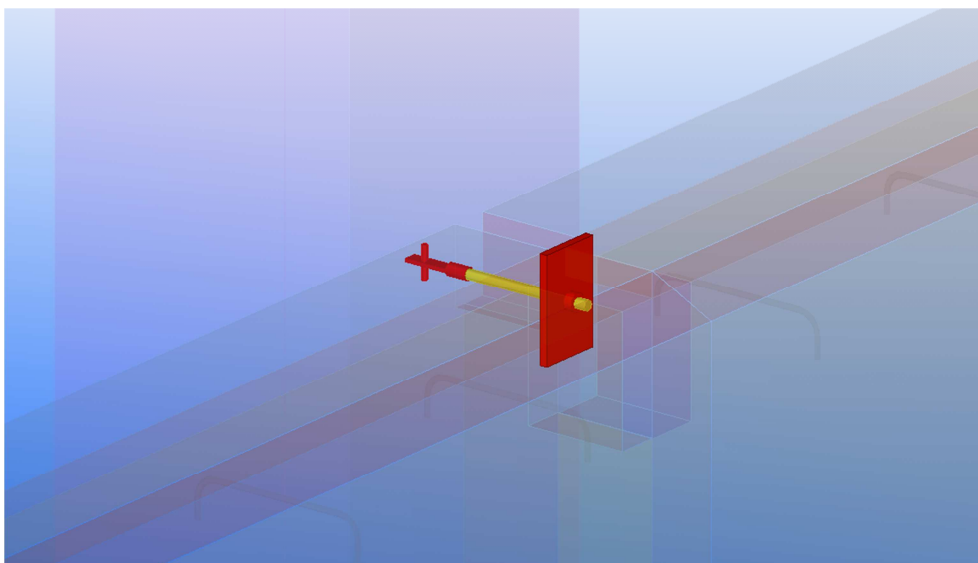
LIITE 1 ESIMERKKI KOMPONENTTI CUSTOM COMPONENT EDITORILLA

Väline:

Tekla Structures -ohjelmisto 20.0

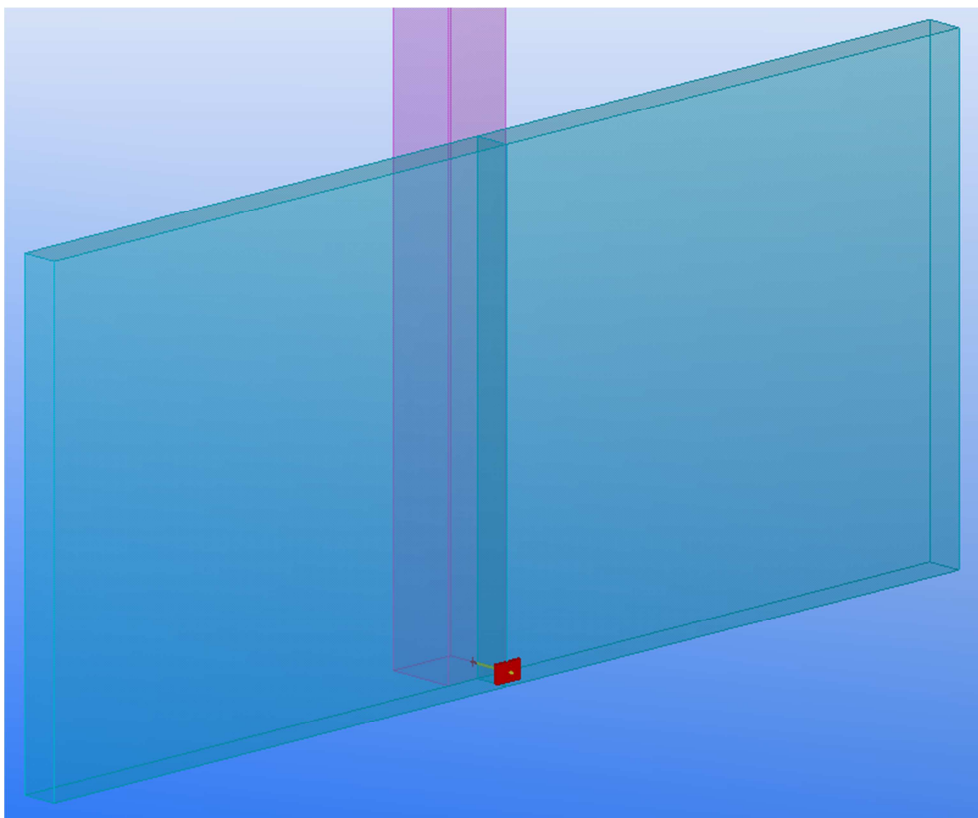
1. Komponentin luominen

Komponentti luodaan käyttäen hyväksi jo olemassa olevaa yläpään kiinnikettä PC_CO_EW_EW_GR_02 (kuva 1.). Sitä varten mallinnetaan kaksi sandwich-elementtiä ja pilari. Komponentti luodaan niiden liitoskohtaan, jonka jälkeen se räjäytetään osiksi.

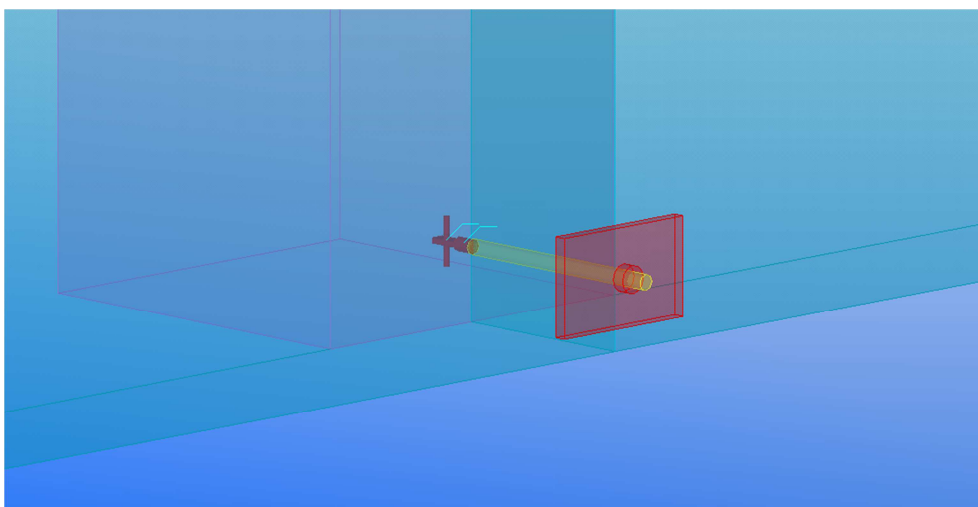


Kuva 1. Sandwich-tyyppisen elementin yläpään kiinnityskomponentti

Räjäytetyistä osista voidaan käyttää hyväksi sisäkierrehylsy, pultti, aluslevy ja mutteri. Mallinnetaan uutta komponenttia varten kaksi betonielementtiä ja yksi betonipilari. Siirretään edellä mainitut kiinnikkeen osat elementtien alapäähän ja muokataan niistä sopivia.

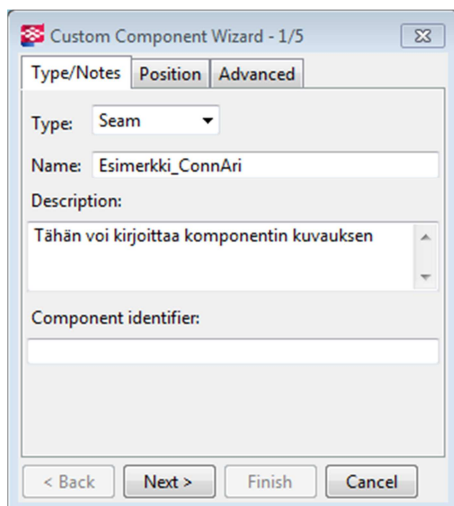


Kuva 2. Lähtötilanteen kokonaisuus



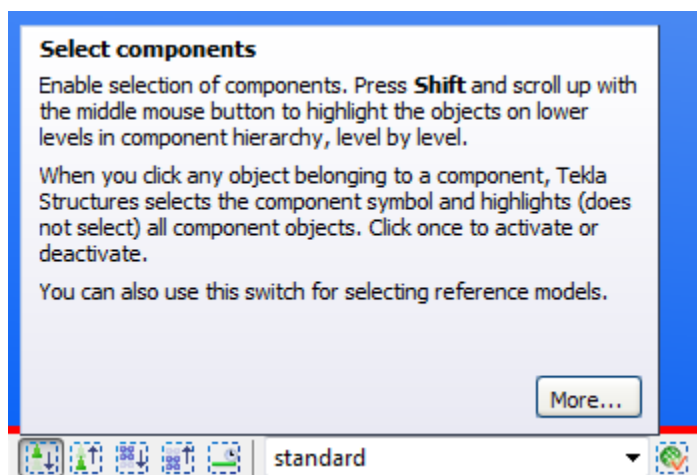
Kuva 3. Osat muokattuna ja siirrettynä alapäähän

Valitaan Detailing - Component - Define Custom Component... ja aletaan määrittää komponentin asetuksia ohjeiden mukaisesti.



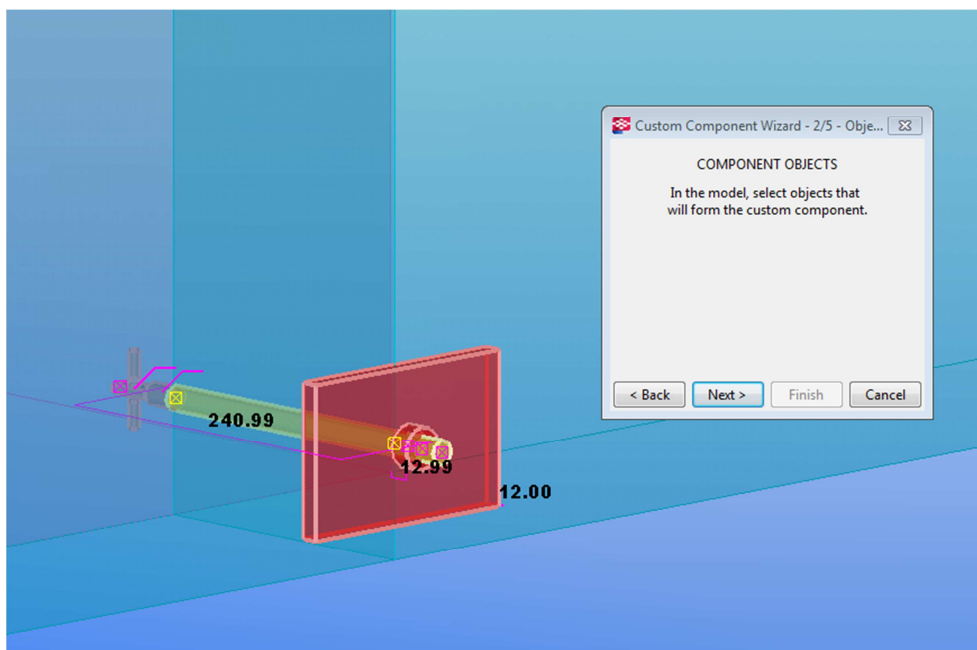
Kuva 4. Komponentin luomisen ensimmäinen vaihe

Ensimmäisenä määritetään komponentille vähintään tyyppi ja nimi, kuvaus on vapaaehtoinen. Nimen tulisi olla täysin uniikki eli samannimistä komponenttia ei saa löytyä ohjelmistosta.



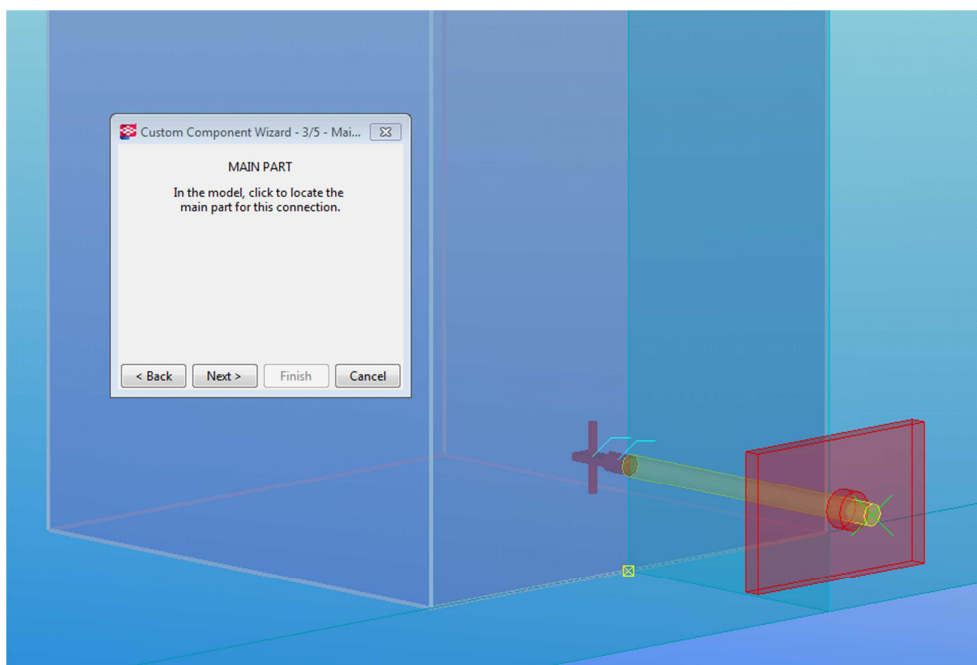
Kuva 5. Select components -painike

Seuraavassa vaiheessa määritellään komponentin osat. Koska sisäkierrehylsy tulee toimimaan alikomponenttina, on TS:n alakulmassa oltava valittuna Select components -painike (kuva 5.)



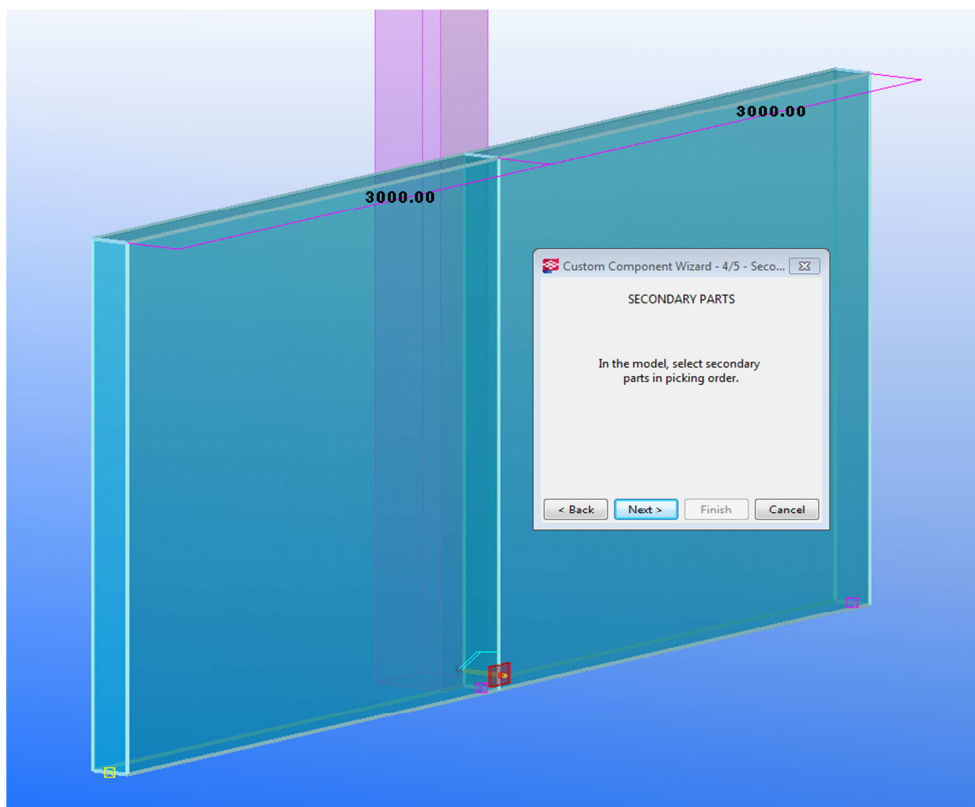
Kuva 6. Komponentin tekemisen toinen vaihe

Toisessa vaiheessa valitaan osat, jotka muodostavat komponentin eli sisäkierrehylsy, pultti, aluslevy ja mutteri. Komponenttia voidaan muokata ja lisätä siihen osia vielä editorin puolella, joten ei haittaa, vaikka kaikkia osia ei vielä tässä vaiheessa olisi mallinnettuna, esimerkiksi leikkaukset luodaan vasta editorissa.



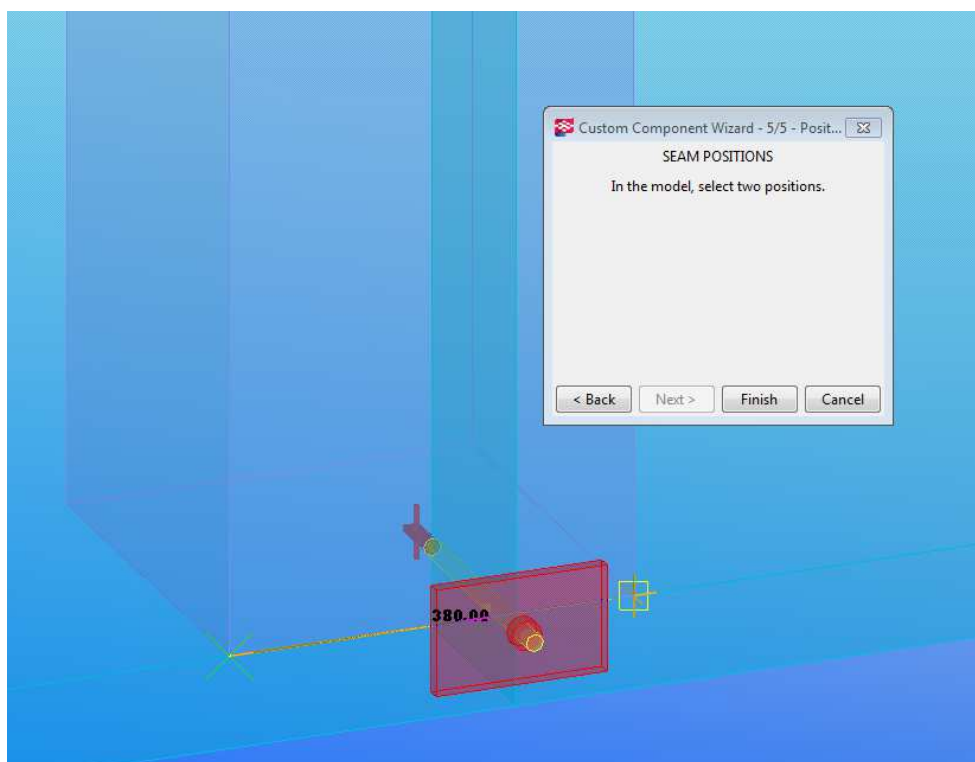
Kuva 7. Komponentin tekemisen kolmas vaihe

Kolmannessa vaiheessa valitaan komponentin pääosa, joka on tässä tapauksessa pilari. Pääosan määrittästä tarvitaan, kun komponenttia ollaan luomassa malliin.



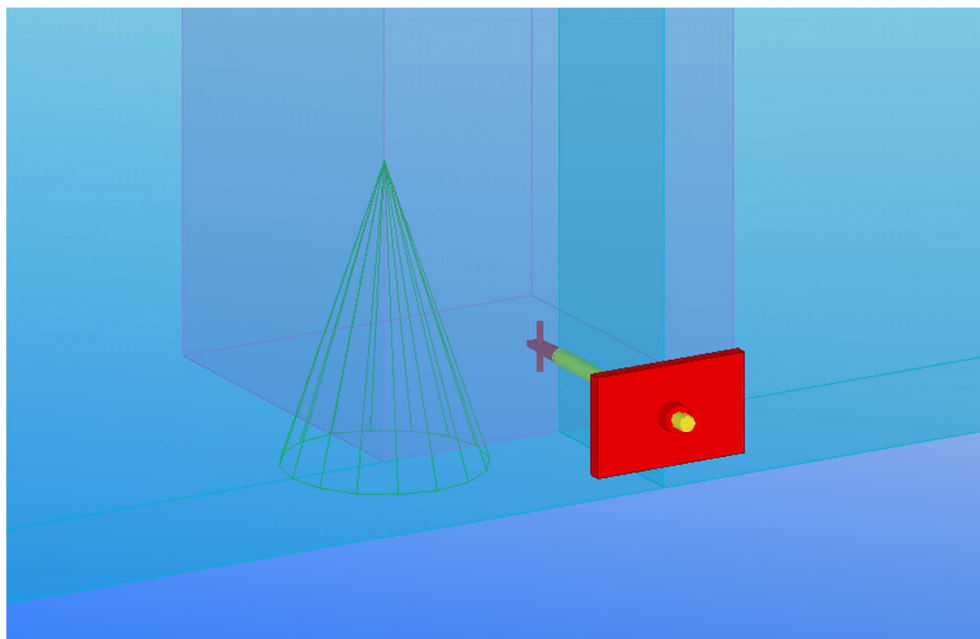
Kuva 8. Komponentin tekemisen neljäs vaihe

Neljännessä vaiheessa valitaan muut osat, joihin komponentti kiinnittyy eli valitaan seinäelementit. Elementtien valinta tehdään siinä järjestyksessä, kun ne halutaan valmiista komponentista luodessakin valita.



Kuva 9. Komponentin tekemisen viides vaihe

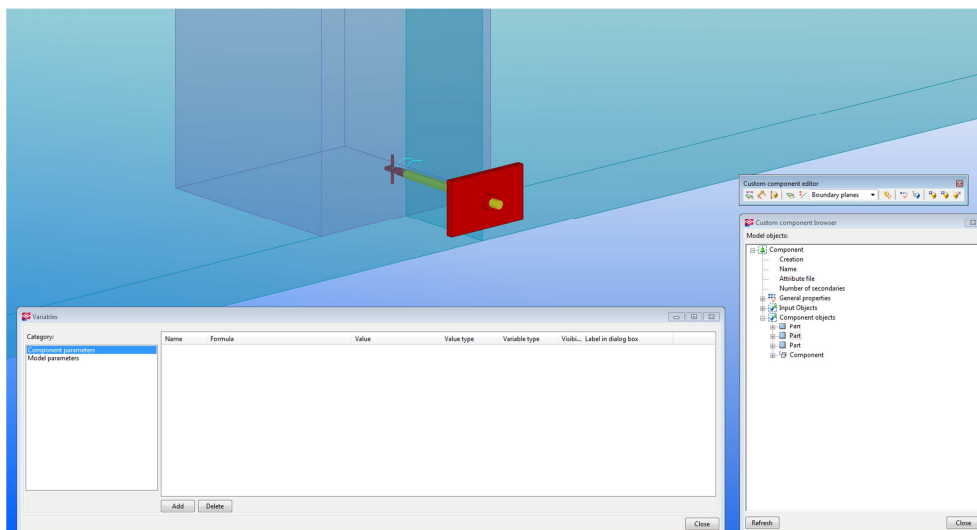
Viimeisenä valitaan kaksi pistettä, joiden suhteen komponentti muodostuu ja painetaan Finish.



Kuva 10. Alkeellinen komponentti

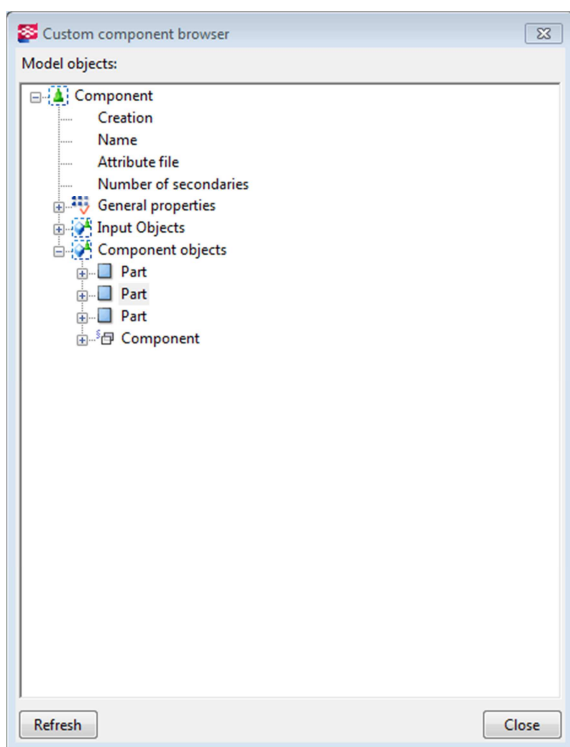
Meillä on nyt valmis komponentti, jota voidaan käyttää sellaisenaan samanlaisissa liitostapauksissa. Mikäli komponentti halutaan kuitenkin toimivaksi useissa eri tapauksissa, aloitetaan sen parametrisointi editorin puolella.

2. Editorin esittely



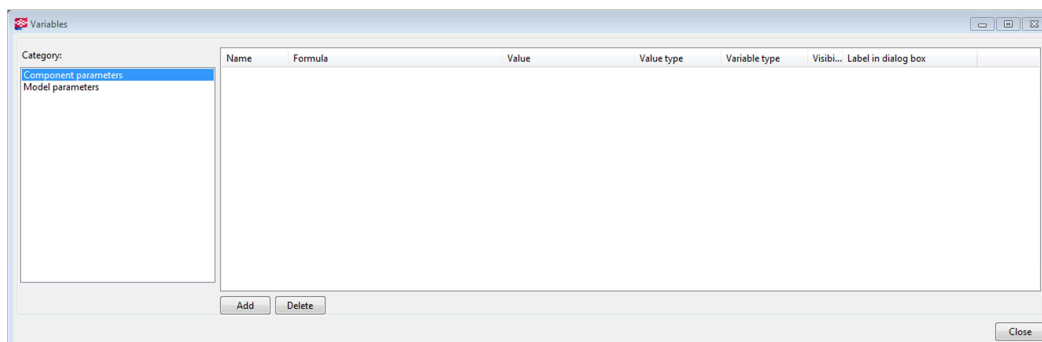
Kuva 11. Custom Component Editor

Editorissa aukeaa näkymiä sivulta ja ylhäältä sekä yksi 3D-näkymä. Ohjelmistossa on näkyvissä myös tavalliset näkymät mallin puolelta, mutta niitä ei tulisi käyttää editorin ollessa auki. Kaikki muutokset komponenttiin on tehtävä editorin näkymissä. Parametrisoinnissa käytetään kolmea ikkuna, jotka ovat Variables, Custom Component Editor ja Custom Component Browser. Variables ja Custom Component Browser löytyvät Custom Component Editor -ikkunan painikkeiden takaa.



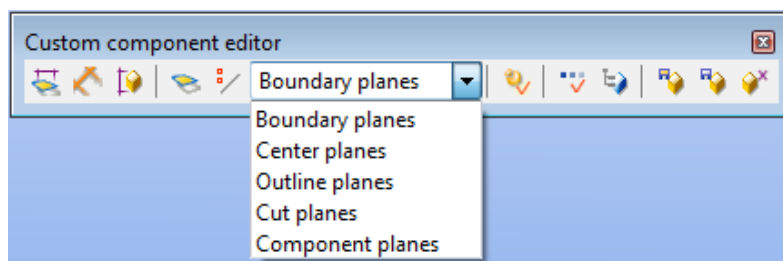
Kuva 12. Custom component browser

CC Browserista löytyvät kaikki komponentin muodostavat osat sekä osat, joihin komponentti kiinnittyy.



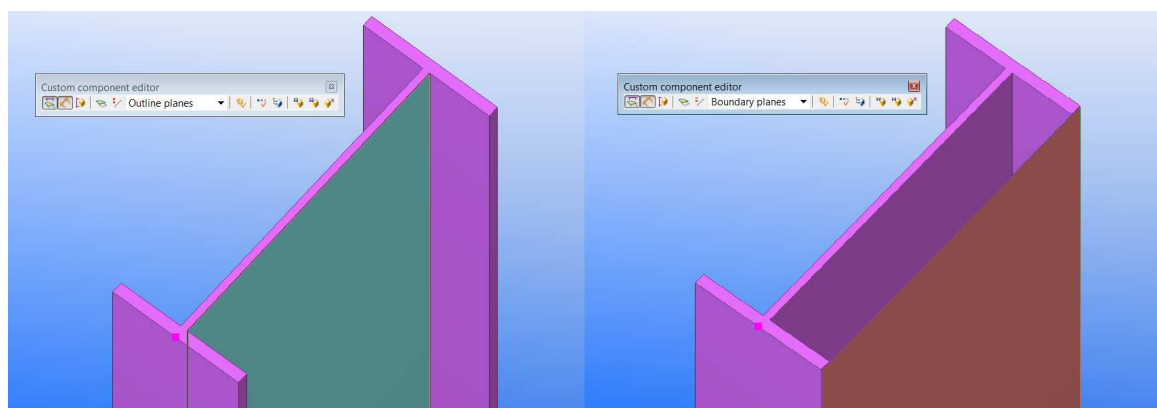
Kuva 13. Variables

Variables-ikkunaan päivittyvät tehdyt parametrizoinnit. D-alkuiset parametrit syntyvät automaattisesti, kun handle-pisteitä kiinnitetään tasoihin. P-alkuiset parametrit luodaan itse. Parametrizointi tarkoittaa yksinkertaisuudessaan sitä, että komponentin osien handle-pisteet eli "kahvat" kiinnitetään eri tasoihin. Tasoja on viisi erilaista ja ne löytyvät CC Editor -ikkunasta.



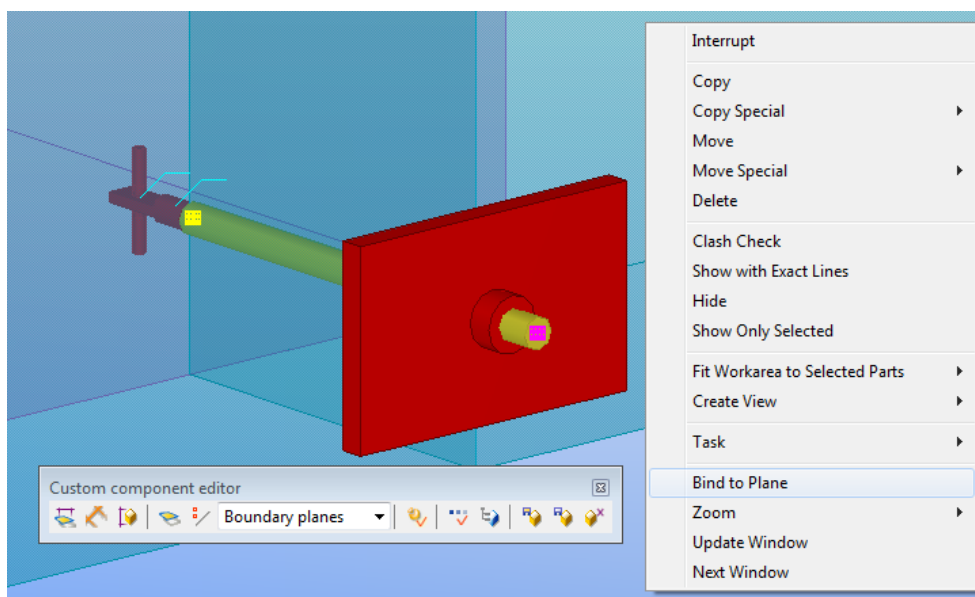
Kuva 14. Custom component editor

CC Editor -ikkunasta voidaan valita tasot, joihin handle-pisteet kiinnitetään. Boundary ja Outline planes ovat osan ulkopintoja, Center planes on osan keskikohta, Cut planes on osan leikkauspinta ja Component planes on komponenttipintoja. Boundary ja Outline planes eroavat toisistaan kuvan mukaisesti.



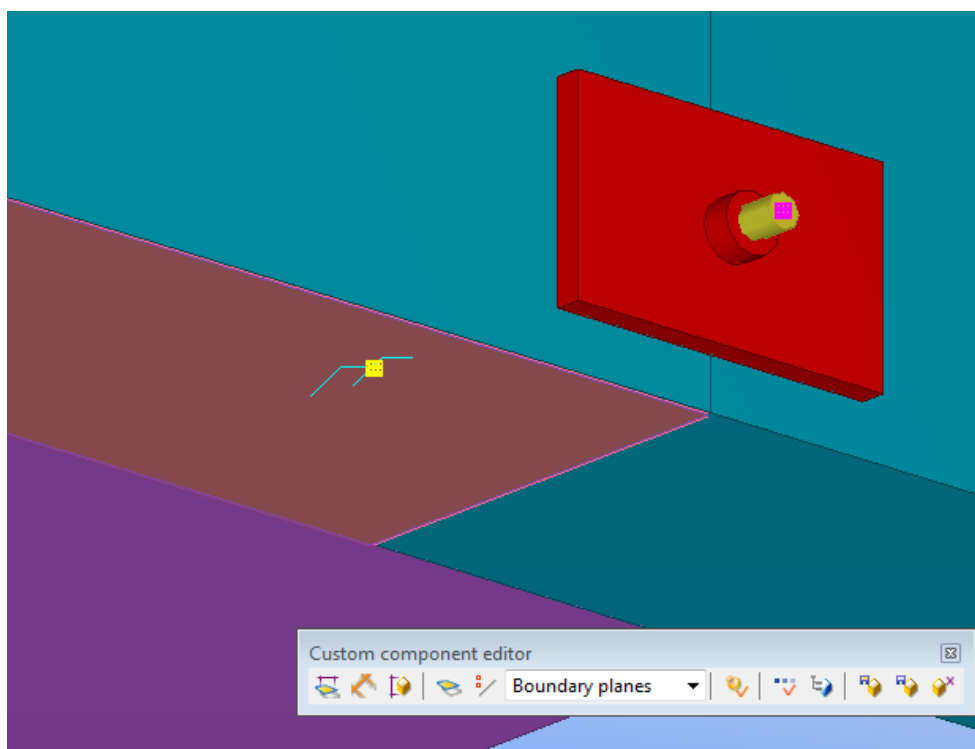
Kuva 15. Vasemmalla Outline planes ja oikealla Boundary planes

3. Pultin kiinnittäminen



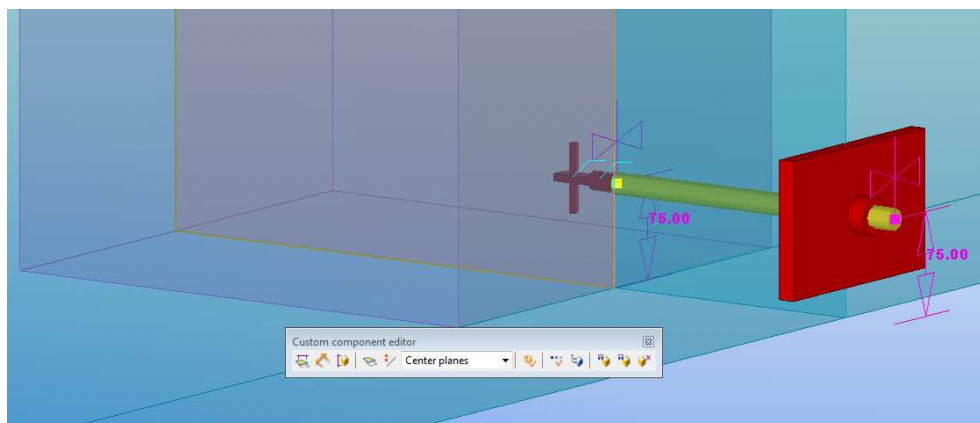
Kuva 16. Bind to Plane -komennon käyttäminen

Aloitetaan komponentin parametrisointi pultista. Hiiren oikean painikkeen alta löytyy Bind to Plane -komento, jolla tasoihin kiinnittäminen tapahtuu.



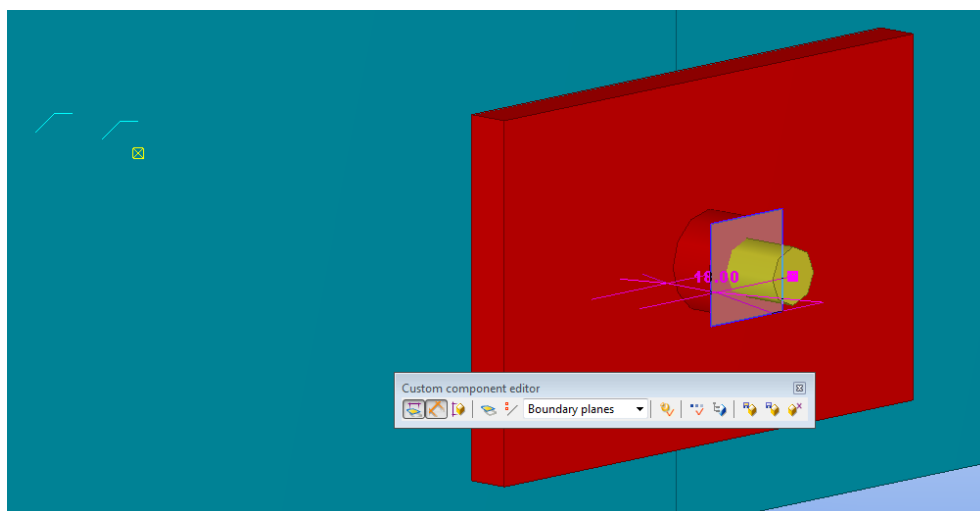
Kuva 17. Pultin kiinnittäminen z-suunnassa

Klikataan pulttia kerran, jonka jälkeen sen handle-pisteet tulevat näkyviin. Valitaan handle-pisteet aluevalinnalla Alt-näppäin pohjassa. Kiinnitetään pultti z-suunnassa elementin Boundary planes -tasoon, jolloin sen korkeus määräytyy tuon tason mukaisesti. Kiinnittäminen olisi onnistunut myös Component Planes -tasoon, silloin komponentin toiminta olisi kokonaisuudessaan hieman muuttunut.



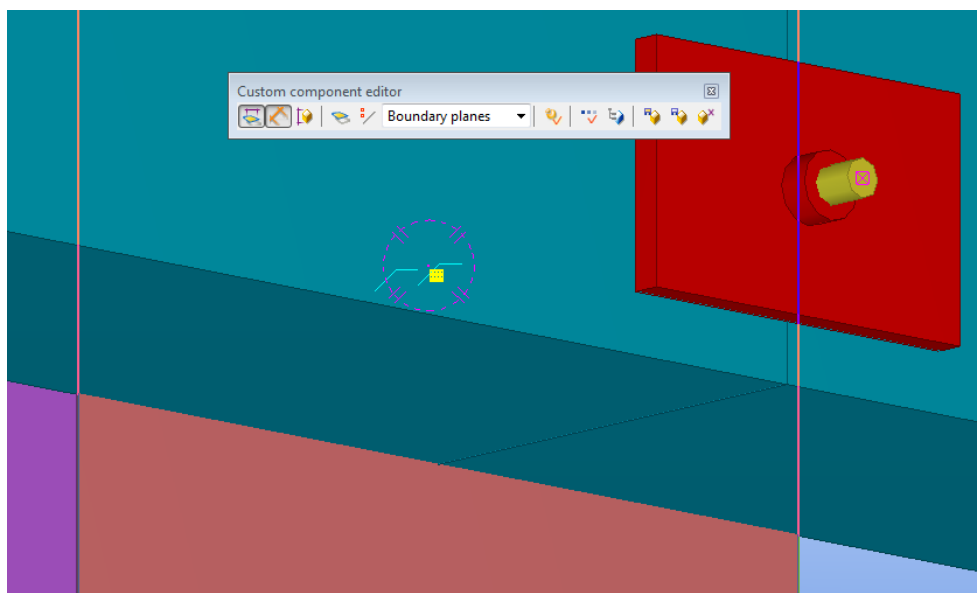
Kuva 18. Pultin kiinnittäminen x-suunnassa

Pultin x-suunta määritetään toimimaan pilarin keskikohdan mukaan eli pultti muodostuu aina keskele pilaria, huolimatta pilarin koosta. Joissain tapauksissa tasoihin kiinnittäminen ei aina onnistunut, jos osat olivat läpinäkyviä.



Kuva 19. Pultin kiinnitys y-suunnassa mutteriin

Koska halutaan, että pultin pituus muuttuu sen mukaan, kuinka paksu aluslevy, mutteri ja elementit ovat, kiinnitetään sen toinen pää mutterin Boundary planes -tason mukaan.



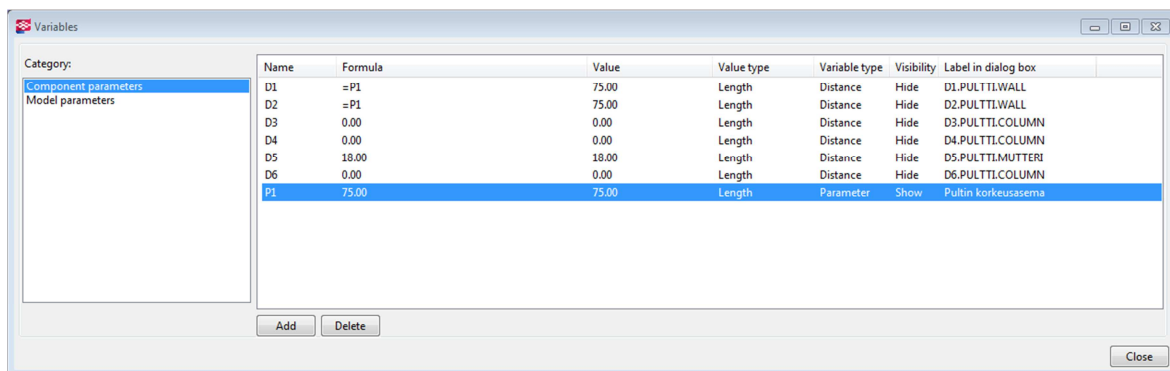
Kuva 20. Pultin kiinnitys y-suunnassa pilariin

Pultin toinen pää kiinnetään pilarin Boundary planes -tasoon, joten vaikka elementin eivät olisi aina samalla etäisyydellä pilarista, pultti osaa muuttaa pituuttaan.

Category:	Name	Formula	Value	Value type	Variable type	Visibility	Label in dialog box
Component parameters	D1	75.00	75.00	Length	Distance	Hide	D1.PULTTI.WALL
Model parameters	D2	75.00	75.00	Length	Distance	Hide	D2.PULTTI.WALL
	D3	0.00	0.00	Length	Distance	Hide	D3.PULTTI.COLUMN
	D4	0.00	0.00	Length	Distance	Hide	D4.PULTTI.COLUMN
	D5	18.00	18.00	Length	Distance	Hide	D5.PULTTI.MUTTERI
	D6	0.00	0.00	Length	Distance	Hide	D6.PULTTI.COLUMN

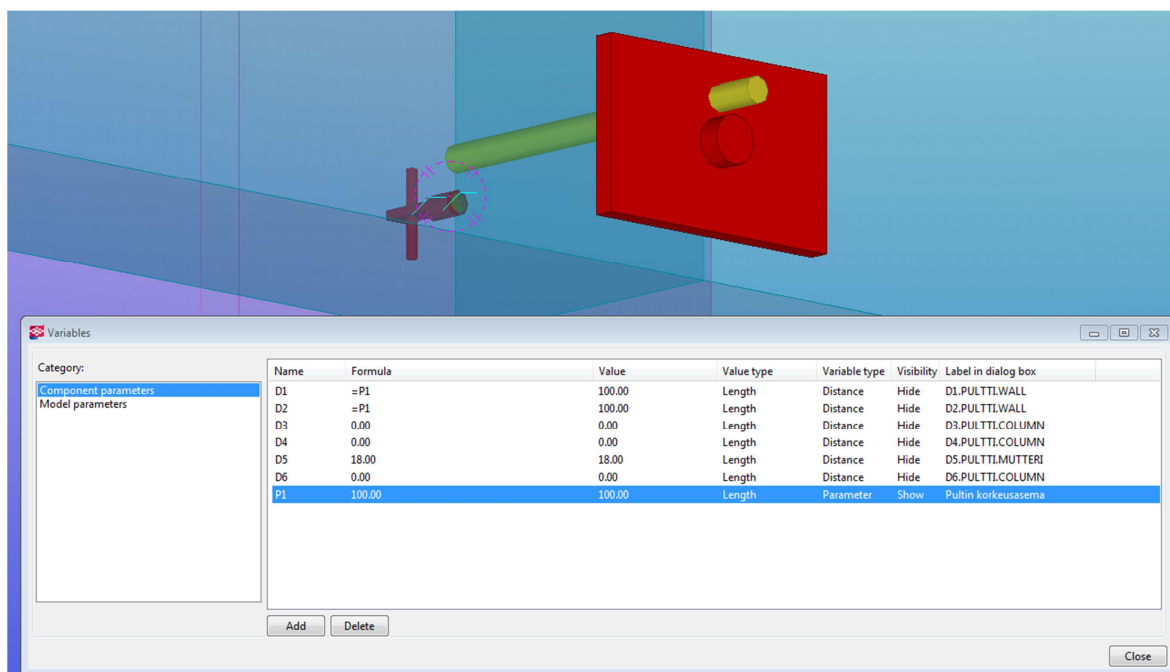
Kuva 20. Variables taulukko D1 - D6

Variables-valikon tulisi näyttää kutakuinkin tältä pultin handle-pisteiden kiinnittämisen jälkeen. Sinne on ilmestynyt D-alkuisia parametrejä kuusi kappaletta.



Kuva 21. D-parametrien yhdistäminen P-parametriin

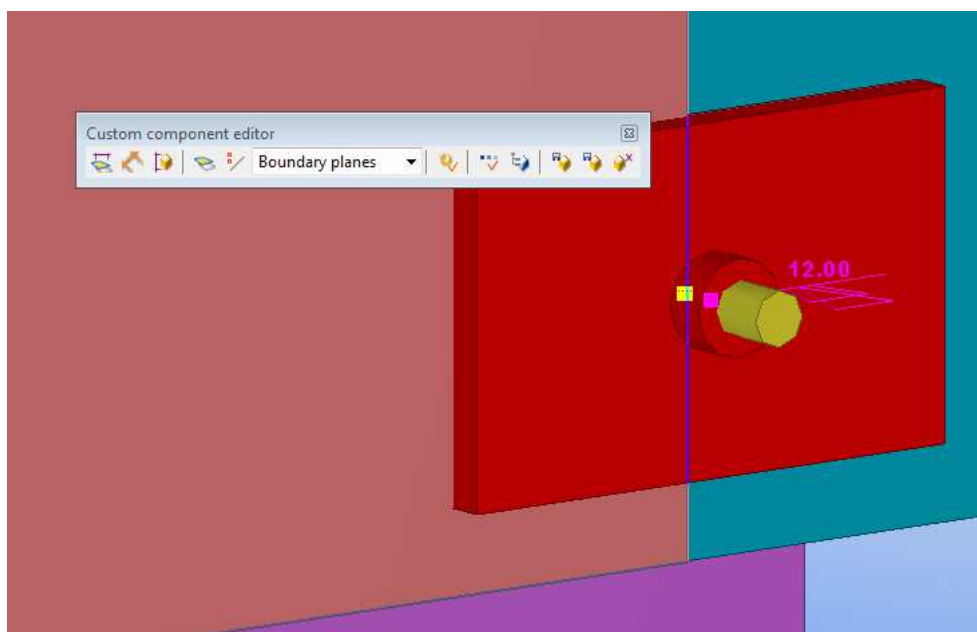
Pultin korkeutta halutaan pystyä säätämään helposti dialogin kautta, joten lisätään Add-painikkeella P-alkuinen parametri. Sille annetaan arvoksi Formula-kohtaa pultin korkeus elementin alapinnasta. Value type -kohtaan valitaan Length, koska kyseessä on pituustyypinen parametri. Label in dialog box -kohtaan kirjoitetaan kuvaus sille, mitä tapahtuu, kun Formula-kohdan arvoa muutetaan. Selvyyden vuoksi voi olla hyvä erotella D- ja P-alkuiset parametrit siten, että D-parametrit ovat aina piilotettuina ja P-parametrit näkyvinä. Visibility-kohtaa muuttamalla saadaan siis näkyviin arvoja komponentin dialogiin. Lisäksi on hyvä liittää P-parametrit ohjaamaan D-parametrejä eli kirjoitetaan nyt D1- ja D2-kohtiin "=P1", jolloin P-parametriä muuttamalla myös D1- ja D2-parametrit muuttuvat. Tällä tavalla saadaan pidettyä tietynlainen selkeys Variables-aulukossa.



Kuva 22. P1-parametrin testaus

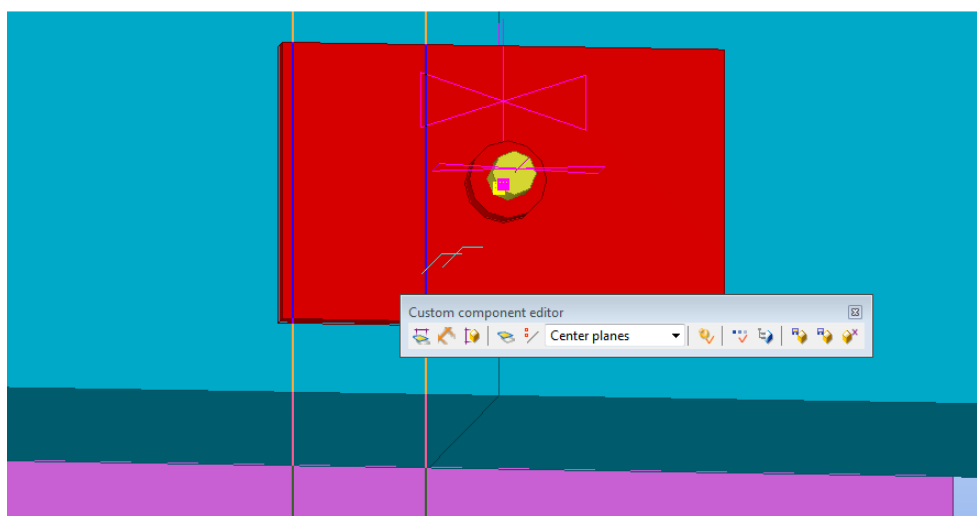
Kun P1 Formula-kohtaan muutetaan nyt arvoja, pultin korkeus muuttuu. Ensimmäinen parametrisointi on nyt luotu.

4. Aluslevyn kiinnittäminen



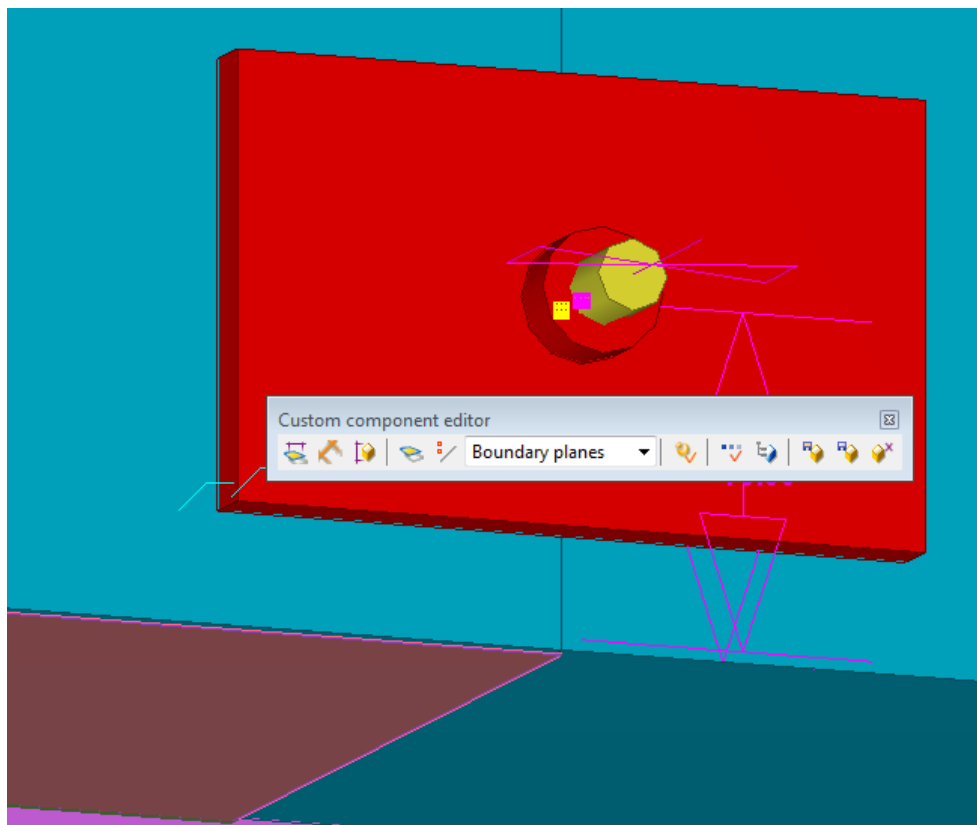
Kuva 23. Aluslevyn y-suunnan kiinnitys

Kiinnitetään aluslevyn alku- ja loppupiste vasemman elementin ulkokuoreen y-suunnassa.



Kuva 24. Aluslevyn x-suunnan kiinnitys

X-suunnan kiinnitys tapahtuu keskelle pilaria eli samaan paikkaan kuin pultin kiinnityskin.



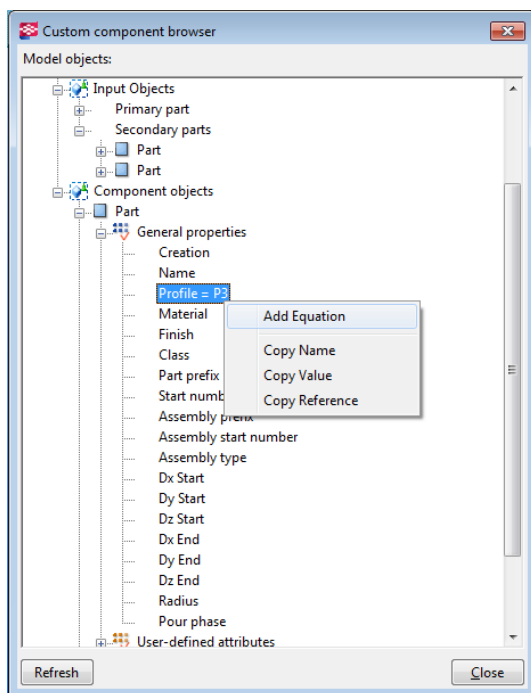
Kuva 25. Aluslevyn z-suunnan kiinnitys

Z-suunnassa aluslevy kiinnitetään myös samaan paikkaan kuin pultti eli vasemman elementin alapintaan.

Category:	Name	Formula	Value	Value type	Variable type	Visibility	Label in dialog box
Component parameters	D1	=P1	75.00	Length	Distance	Hide	D1.PULTTI.WALL
Model parameters	D2	=P1	75.00	Length	Distance	Hide	D2.PULTTI.WALL
	D3	0.00	0.00	Length	Distance	Hide	D3.PULTTI.COLUMN
	D4	0.00	0.00	Length	Distance	Hide	D4.PULTTI.COLUMN
	D5	18.00	18.00	Length	Distance	Hide	D5.PULTTI.MUTTERI
	D6	0.00	0.00	Length	Distance	Hide	D6.PULTTI.COLUMN
	D7	=P2	12.00	Length	Distance	Hide	D7.ALUSLEVY.WALL
	D8	0.00	0.00	Length	Distance	Hide	D8.ALUSLEVY.WALL
	D9	0.00	0.00	Length	Distance	Hide	D9.ALUSLEVY.COLUMN
	D10	0.00	0.00	Length	Distance	Hide	D10.ALUSLEVY.COLUMN
	D11	75.00	75.00	Length	Distance	Hide	D11.ALUSLEVY.WALL
	D12	75.00	75.00	Length	Distance	Hide	D12.ALUSLEVY.WALL
	P1	75.00	75.00	Length	Parameter	Show	Pultin korkeusasema
	P2	12.00	12.00	Length	Parameter	Show	Aluslevyn paksuus
	P3	100*160	100*160	Profile	Parameter	Show	Aluslevyn profiili

Kuva 26. Variables-taulukko aluslevyn kiinnityksen jälkeen

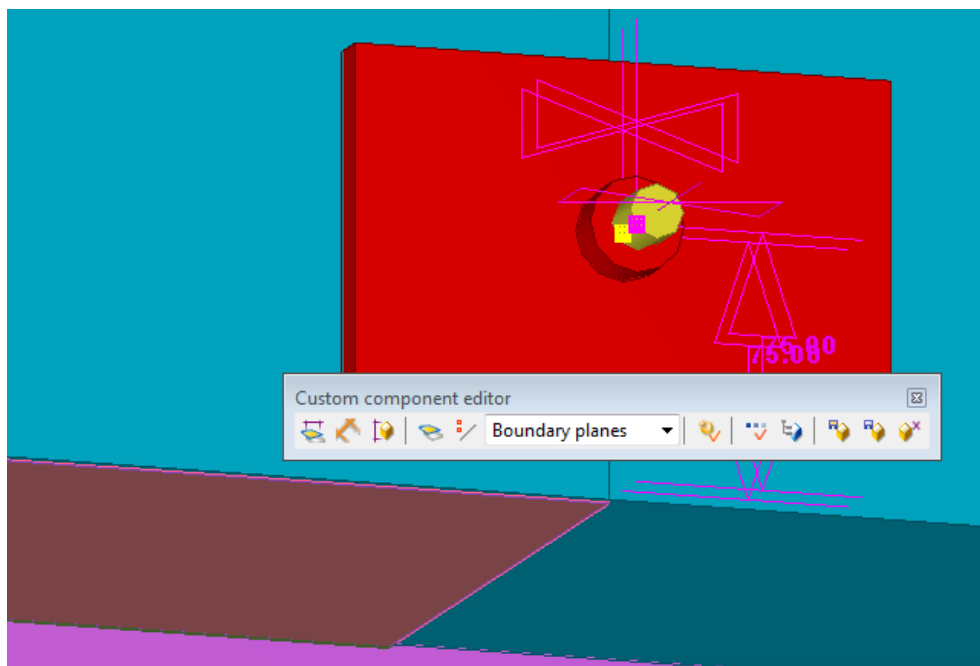
Aluslevyn paksuutta ja profiilia halutaan pystyä muuttamaan dialogin kautta, joten lisätään P2- ja P3-parametrit. P2-parametriin nimeksi annetaan "Aluslevyn paksuus" ja Formula-kohtaan kirjoitetaan sen arvo. Tyyppi on P1-parametrin mukaisesti Length. D7-parametriin kirjoitetaan taas viittaus P2-parametriin. P3-parametriin kirjoitetaan nimeksi "Aluslevyn profiili", tyyppiä vaihdetaan Profile ja annetaan sille arvoksi "100*160". Jotta aluslevyn profiili muuttuisi, täytyy tehdä pieni lisäys CC Browserin puolella.



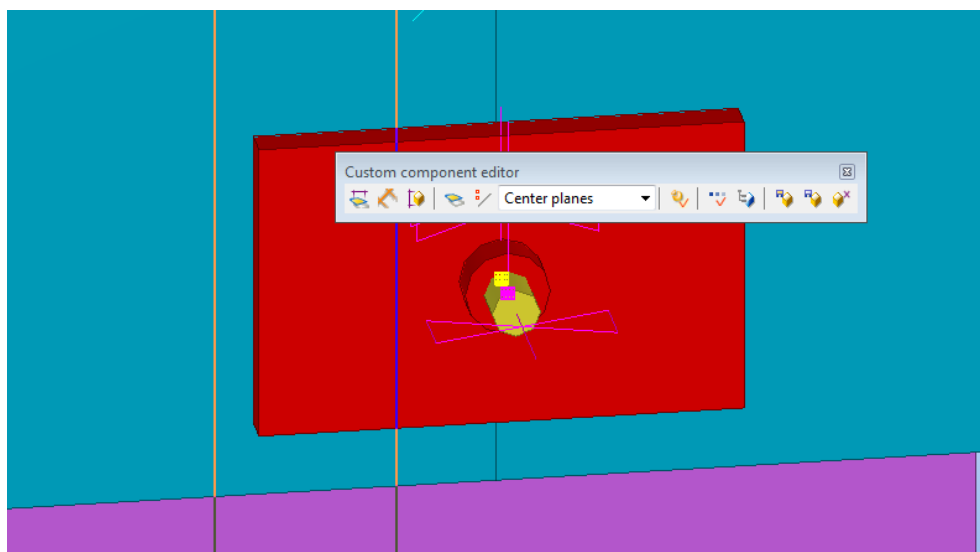
Kuva 27. Yhteyksen luominen Variables-valikkoon

Lisätään aluslevyn Profile-kohtaan CC Browserissa "= P3" Add Equation -toiminnolla. Nyt profilia voidaan muuttaa P3-parametrin kautta.

5. Mutterin kiinnittäminen

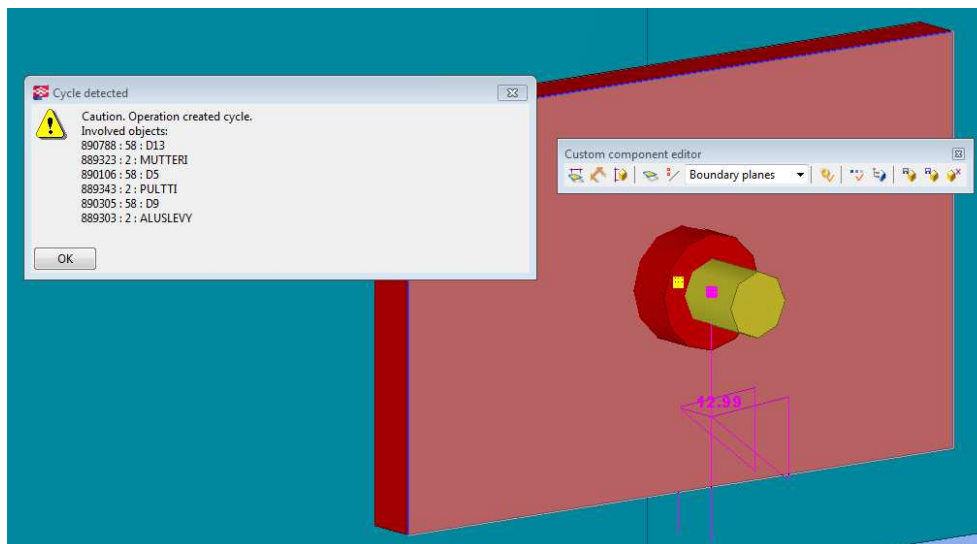


Kuva 28. Mutterin z-suunnan kiinnitys



Kuva 29. Mutterin x-suunnan kiinnitys

Mutteri kiinnitetään x- ja z-suunnassa samalla tavalla kuin pultti ja aluslevykin.



Kuva 30. Loop-tilanteen muodostuminen ja mutterin y-suunnan kiinnitys

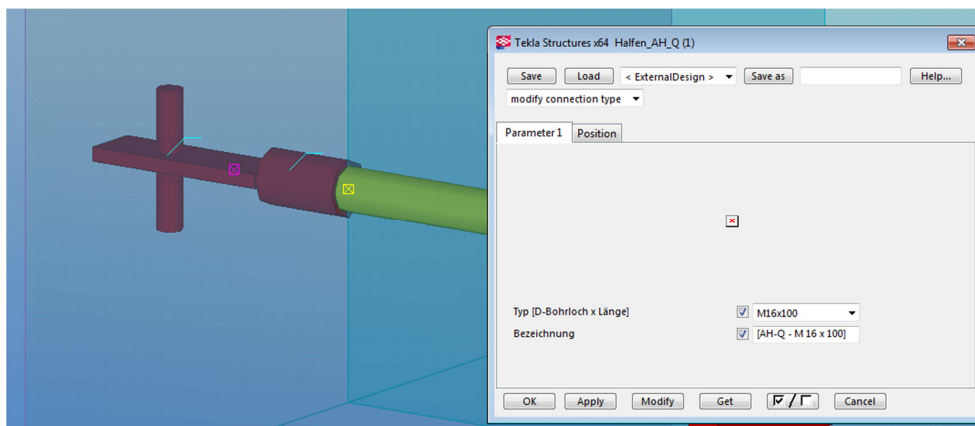
Mutteri kiinnitetään y-suunnassa toimimaan aluslevyn ulkopinnan mukaan. Kun tein aluksi aluslevyn ja mutterin parametrisointeja, kiinnitin ne toimimaan pultin keskikohdan tasojen mukaisesti. Tällöin ohjelmisto antoi kuitenkin varoituksen, että toiminta muodosti loopin eli osien kiinnityksissä tuli viitauksia toistensa suhteen. Mikäli parametrisoinnissa on kuitenkin noudattanut nykyisiä ohjeita, ei tätä varoitusta pitäisi nyt tulla.

Name	Formula	Value	Value type	Variable type	Visibility	Label in dialog box
D7	=P2	12.00	Length	Distance	Hide	D7.ALUSLEVY.WALL
D8	0.00	0.00	Length	Distance	Hide	D8.ALUSLEVY.WALL
D9	0.00	0.00	Length	Distance	Hide	D9.ALUSLEVY.COLUMN
D10	0.00	0.00	Length	Distance	Hide	D10.ALUSLEVY.COLUMN
D11	=P1	75.00	Length	Distance	Hide	D11.ALUSLEVY.WALL
D12	=P1	75.00	Length	Distance	Hide	D12.ALUSLEVY.WALL
D13	12.99	12.99	Length	Distance	Hide	D13.MUTTERI.ALUSLEVY
D14	0.00	0.00	Length	Distance	Hide	D14.MUTTERI.ALUSLEVY
D15	0.00	0.00	Length	Distance	Hide	D15.MUTTERI.COLUMN
D16	0.00	0.00	Length	Distance	Hide	D16.MUTTERI.COLUMN
D17	=P1	75.00	Length	Distance	Hide	D17.MUTTERI.WALL
D18	=P1	75.00	Length	Distance	Hide	D18.MUTTERI.WALL
P1	75.00	75.00	Length	Parameter	Show	Pultin korkeusasema
P2	12.00	12.00	Length	Parameter	Show	Aluslevyn paksuus
P3	100*160	100*160	Profile	Parameter	Show	Aluslevyn profiili

Kuva 31. Variables-tilauskko mutterin kiinnityksen jälkeen

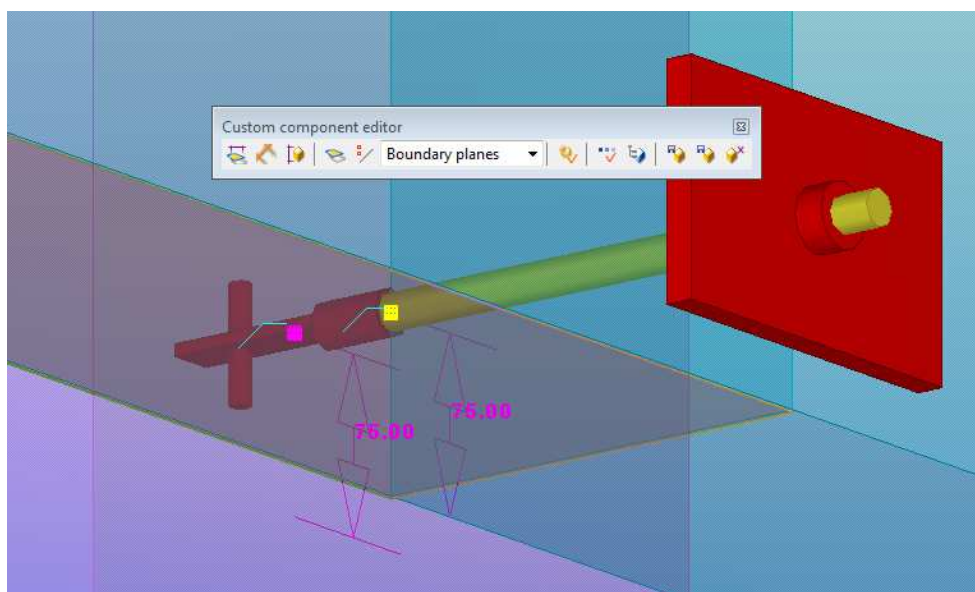
Variables-tilauskko tulisi näyttää nyt kutakuinkin tällaiselta, uutta parametreit D13-D18. Mutterin korkeus laitetaan muuttumaan parametrin P1 mukaisesti.

6. Sisäkierrehylsyn kiinnittäminen



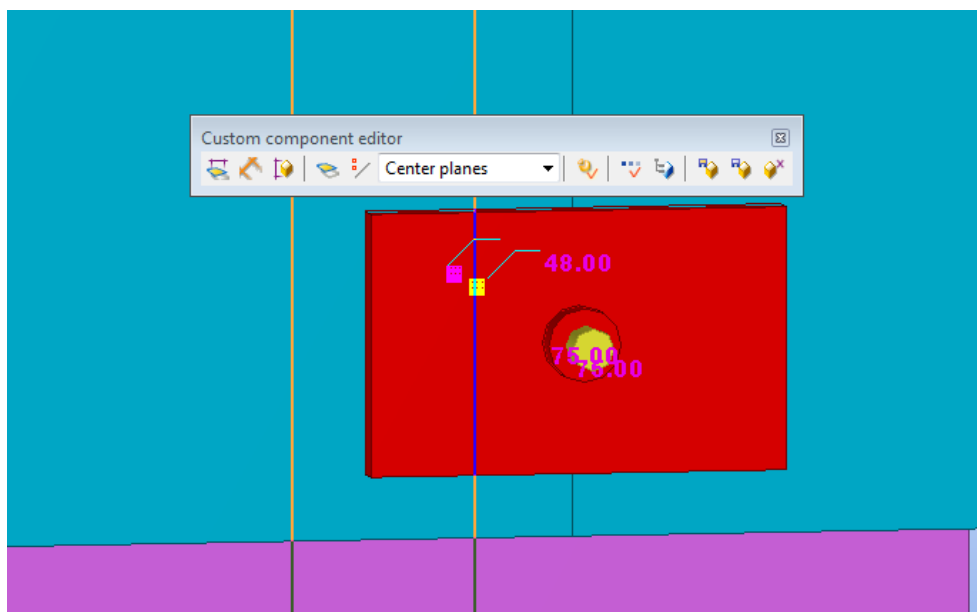
Kuva 32. Sisäkierrehylsyn koon muuttaminen ja kiinnitys pilarin kokoonpanoon

Muutetaan tässä vaiheessa sisäkierrehylsyn koko vastaamaan pultin halkaisijaa ellei sitä jo ole tehty. Sisäkierrehylsy kiinnittämisessä noudatetaan taas samaa kuin pultissa, aluslevyssä ja mutterissakin. Sisäkierrehylsyn tulisi kiinnittyä pilarin kokoonpanoon, jotta se tulostuisi piirustuksissa oikein. Tämän vuoksi tässä vaiheessa olisi hyvä liittää se pilariin. Liittäminen onnistuu, kun Select components – painike on valittuna.



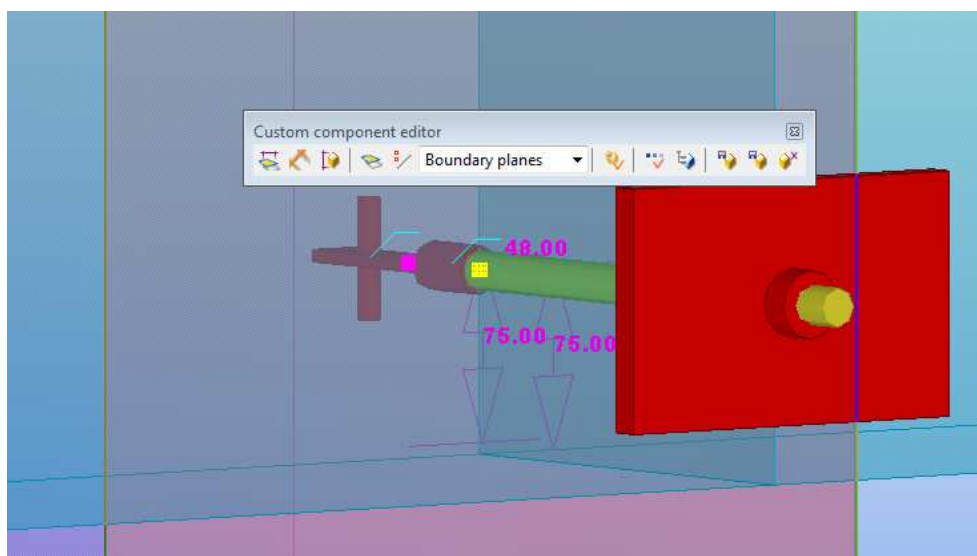
Kuva 33. Sisäkierrehylsyn z-suunnan kiinnitys

Z-suunnan kiinnitys tapahtuu vasemman elementin alapintaan.



Kuva 34. Sisäkierrehylsyn x-suunnan kiinnitys

X-suunnassa sisäkierrehylsy kiinnittyy pilarin keskikohtaan mukaisesti.



Kuva 35. Sisäkierrehylsyn y-suunnan kiinnitys

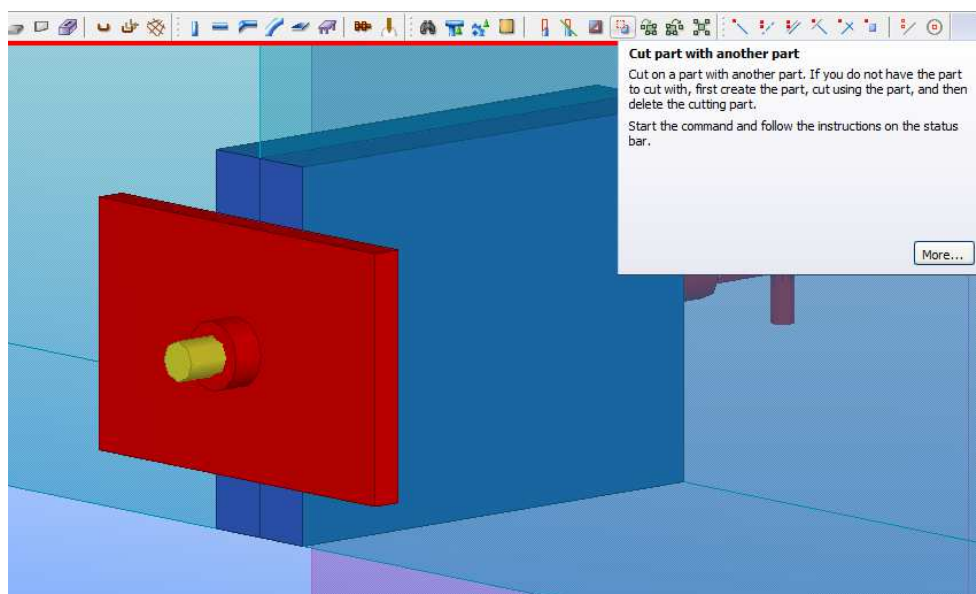
Y-suunnassa sisäkierrehylsy kiinnitetään pilarin ulkopinnan mukaan, jotta vaikka elementtien ja pilarin väli kasvaa, pystyy sisäkierrehylsy aina paikoillaan.

Name	Formula	Value	Value type	Variable type	Visibility	Label in dialog box
D13	12.99	12.99	Length	Distance	Hide	D13.MUTTERL.ALUSLEVY
D14	0.00	0.00	Length	Distance	Hide	D14.MUTTERL.ALUSLEVY
D15	0.00	0.00	Length	Distance	Hide	D15.MUTTERL.COLUMN
D16	0.00	0.00	Length	Distance	Hide	D16.MUTTERL.COLUMN
D17	=P1	75.00	Length	Distance	Hide	D17.MUTTERL.WALL
D18	=P1	75.00	Length	Distance	Hide	D18.MUTTERL.WALL
D19	=P1	75.00	Length	Distance	Hide	D19.Halfen_AH_Q.WALL
D20	=P1	75.00	Length	Distance	Hide	D20.Halfen_AH_Q.WALL
D21	48.00	48.00	Length	Distance	Hide	D21.Halfen_AH_Q.COLUMN
D22	0.00	0.00	Length	Distance	Hide	D22.Halfen_AH_Q.COLUMN
D23	0.00	0.00	Length	Distance	Hide	D23.Halfen_AH_Q.COLUMN
D24	0.00	0.00	Length	Distance	Hide	D24.Halfen_AH_Q.COLUMN
P1	75.00	75.00	Length	Parameter	Show	Pultin korkeusasema
P2	12.00	12.00	Length	Parameter	Show	Aluslevyn paksuus
P3	100*160	100*160	Profile	Parameter	Show	Aluslevyn profiili

Kuva 36. Variables-tila sisäkierrehylsyn kiinnityksen jälkeen

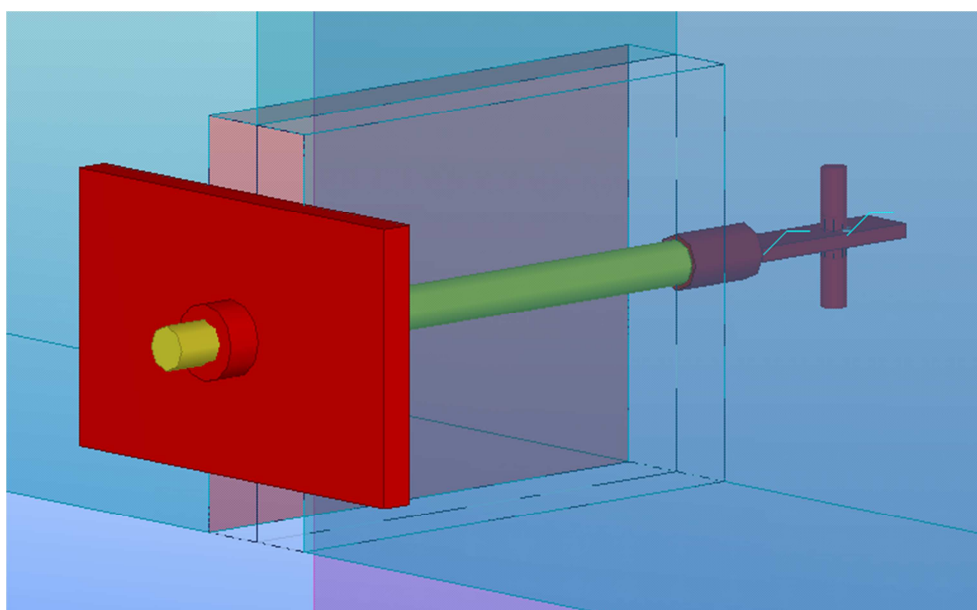
Variables-tilaan muutetaan sisäkierrehylsyn z-suunnan korkeuksien tilalle "=P1", tällöin kaikki komponentin osat liikkuvat korkeutensa suhteen saman parametrin kautta.

7. Leikkauksien lisääminen elementteihin



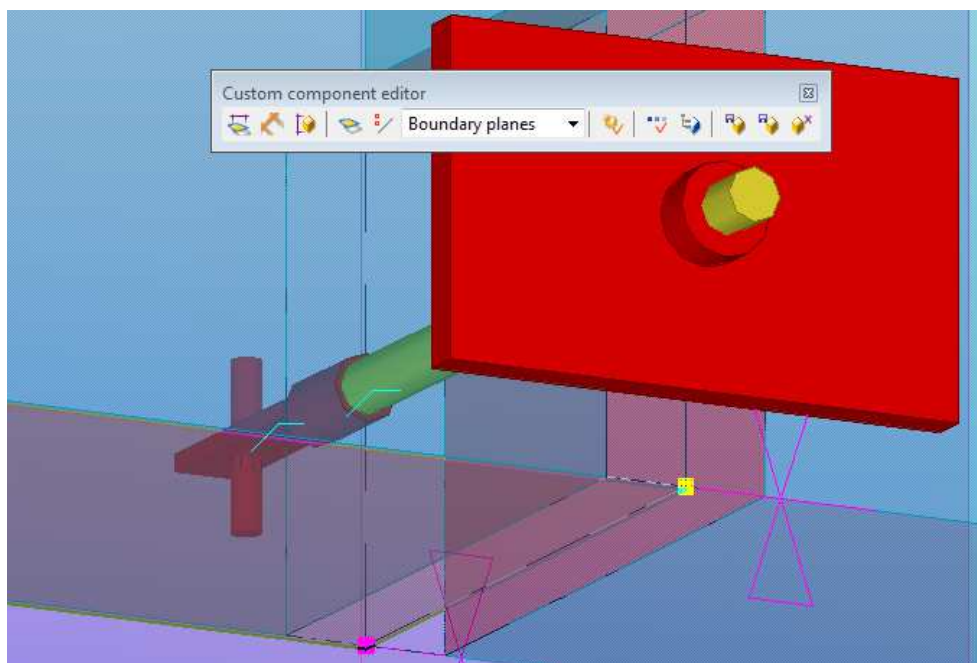
Kuva 37. Elementtien leikkaaminen

Luodaan elementteihin leikkaukset mallintamalla kaksi palkkia editorin puolella ja sitten Cut part with another part -toiminnolla leikataan elementtejä niillä.



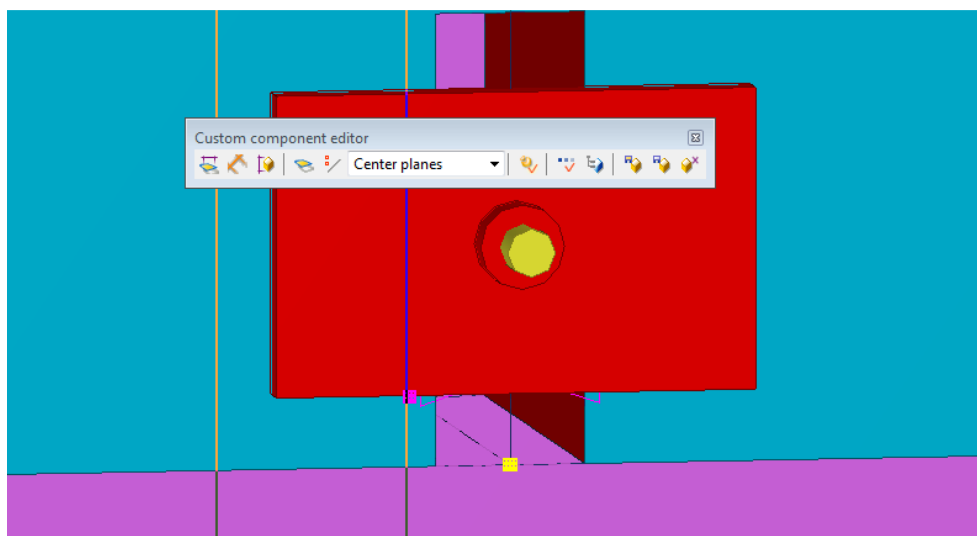
Kuva 38. Leikatut elementit

Poistetaan lopuksi palkit, jonka jälkeen vain leikkauspinnat jäävät näkyviin.



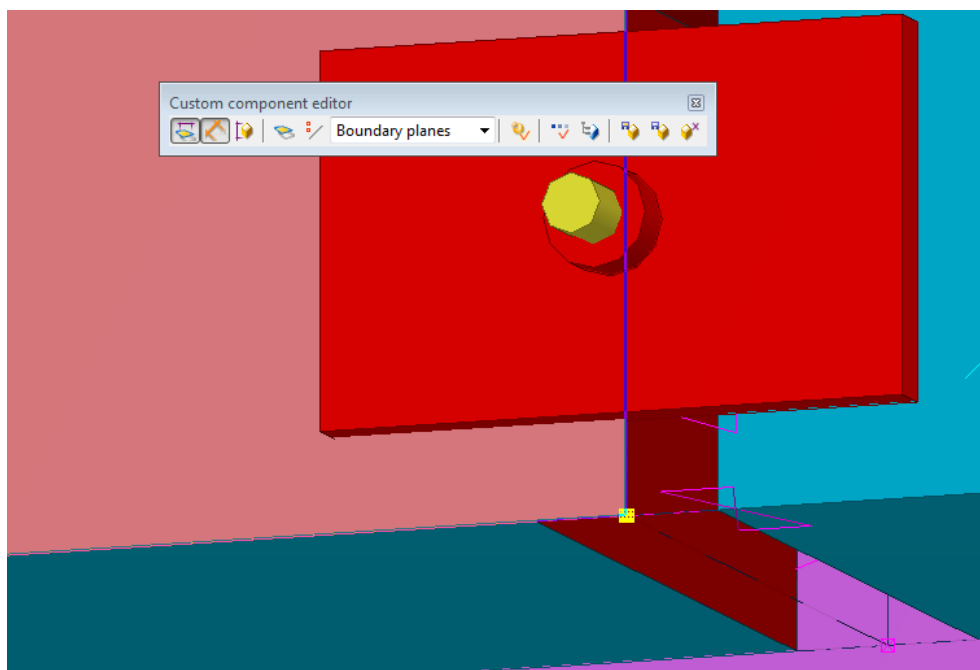
Kuva 39. Leikkauksien kiinnitys z-suunnassa

Kummankin puolen leikkaus kiinnitetään vasemman elementin alapintaan. Muutoksia Variables-taulukon parametreihin ei tarvitse tehdä, sillä leikkauksien sijainnin ei tarvitse muuttua.

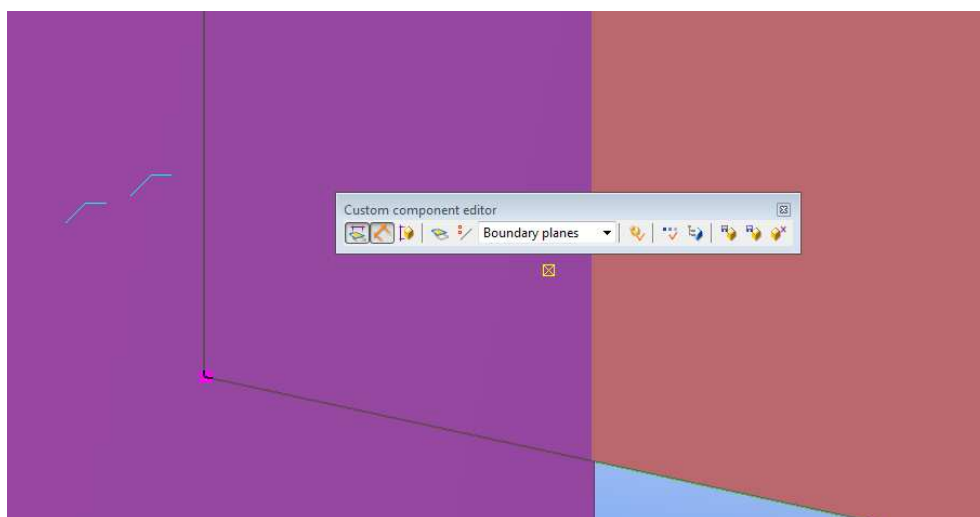


Kuva 40. Leikkauksien kiinnitys x-suunnassa

X-suunnassa kumpikin puoli kiinnitetään taas pilarin keskikohdan mukaisesti.



Kuva 41. Vasemman elementin leikkauksen kiinnitys ulkokuoreen



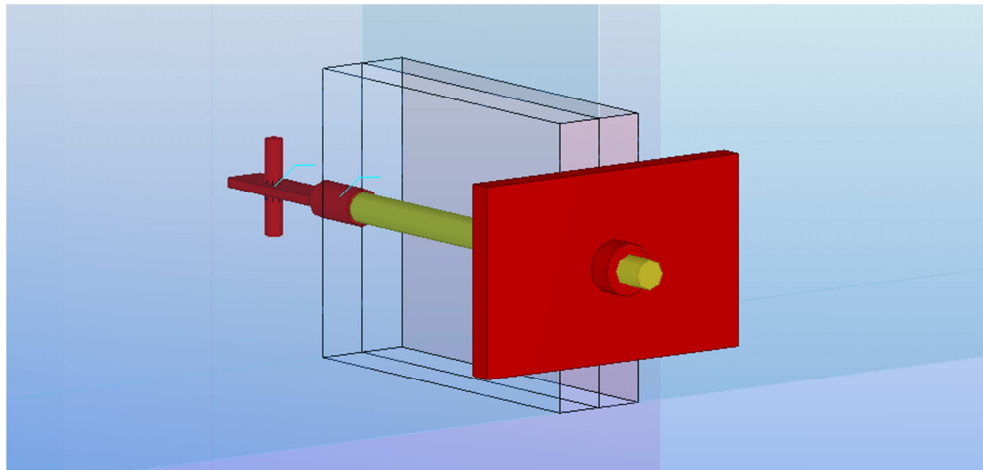
Kuva 42. Vasemman elementin leikkauksen kiinnitys sisäkuoreen

Koska leikkauksien tulee reagoida elementin paksuuden muutoksiin, kiinnitetään vasemman puolen leikkauksen alku- ja loppupisteet eri puolille elementtejä. Sama tehdään myös oikean puolen leikkaukselle.

Name	Formula	Value	Value type	Variable type	Visibility	Label in dialog box
D24	0.00	0.00	Length	Distance	Hide	D24.Halfen_AH_Q.COLUMN
D25	0.00	0.00	Length	Distance	Hide	D25.LEIKKAUS.WALL
D26	0.00	0.00	Length	Distance	Hide	D26.LEIKKAUS.WALL
D27	0.00	0.00	Length	Distance	Hide	D27.LEIKKAUS.WALL
D28	0.00	0.00	Length	Distance	Hide	D28.LEIKKAUS.WALL
D29	0.00	0.00	Length	Distance	Hide	D29.LEIKKAUS.COLUMN
D30	0.00	0.00	Length	Distance	Hide	D30.LEIKKAUS.COLUMN
D31	0.00	0.00	Length	Distance	Hide	D31.LEIKKAUS.COLUMN
D32	0.00	0.00	Length	Distance	Hide	D32.LEIKKAUS.COLUMN
D33	0.00	0.00	Length	Distance	Hide	D33.LEIKKAUS.WALL
D34	0.00	0.00	Length	Distance	Hide	D34.LEIKKAUS.WALL
D35	0.00	0.00	Length	Distance	Hide	D35.LEIKKAUS.WALL
D36	0.00	0.00	Length	Distance	Hide	D36.LEIKKAUS.WALL
P1	75.00	75.00	Length	Parameter	Show	Pultin korkeusasema
P2	12.00	12.00	Length	Parameter	Show	Aluslevyn paksuus
P3	100*160	100*160	Profile	Parameter	Show	Aluslevyn profiili

Kuva 43. Variables-tila leikkauksien kiinnityksen jälkeen

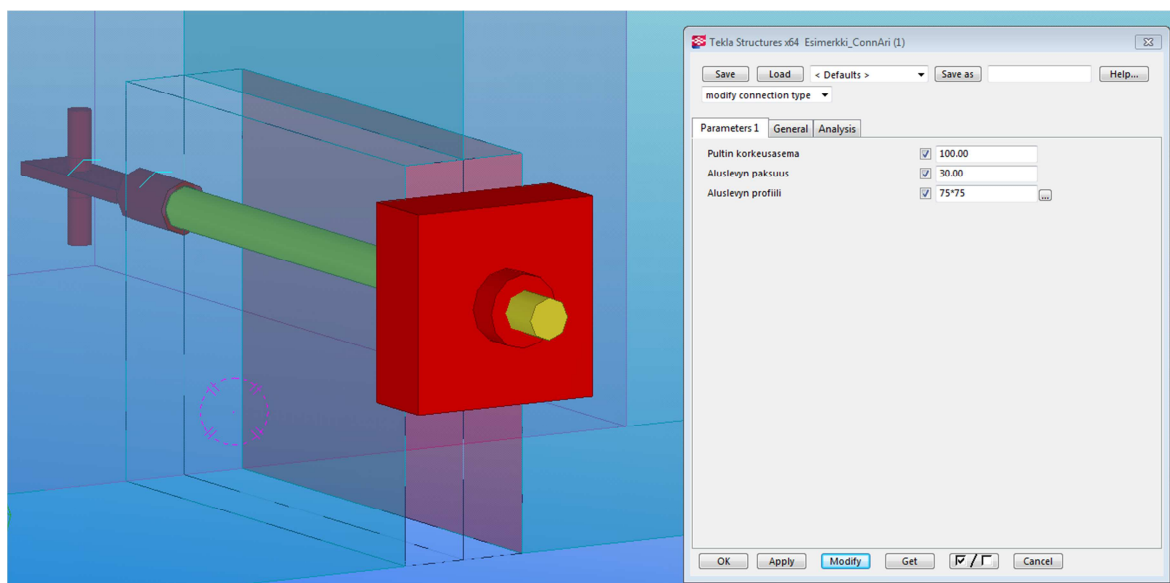
Variables-tilan tulisi näyttää nyt suunnilleen tältä. Uusia ovat parametrit D25-D36.



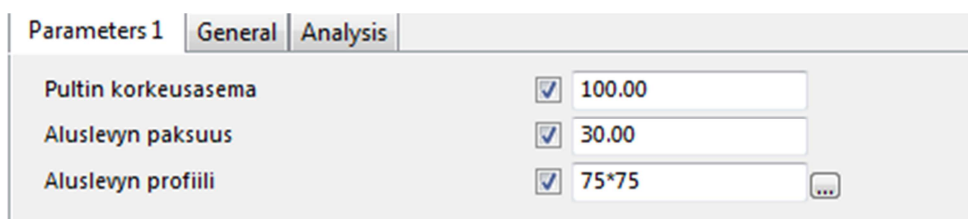
Kuva 44. Komponentti sisältäen kaikki osansa

Komponentti näyttää nyt kokonaisuudessaan tältä. Tallennetaan komponentti ja suljetaan editori.

8. Komponentin testaus

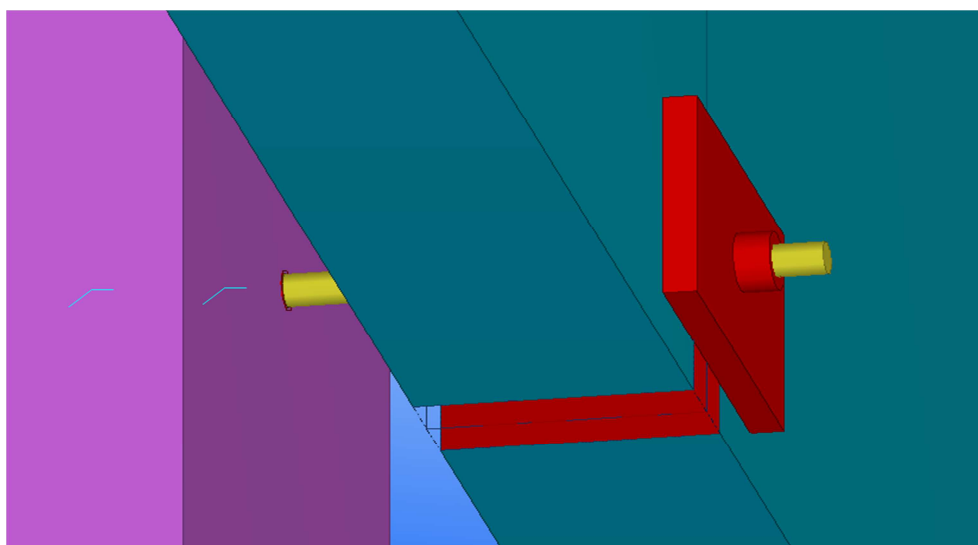


Kuva 45. Komponentin korkeuden sekä aluslevyn paksuuden ja profiilin muuttaminen



Kuva 46. Dialogin tekstikentät tarkennettuna

Testataan komponentin toimivuutta mallin puolella. Dialogissa tulisi nyt näkyä seuraavat kohdat, jotka laitettiin Variables-taulukon kautta näkyviksi.

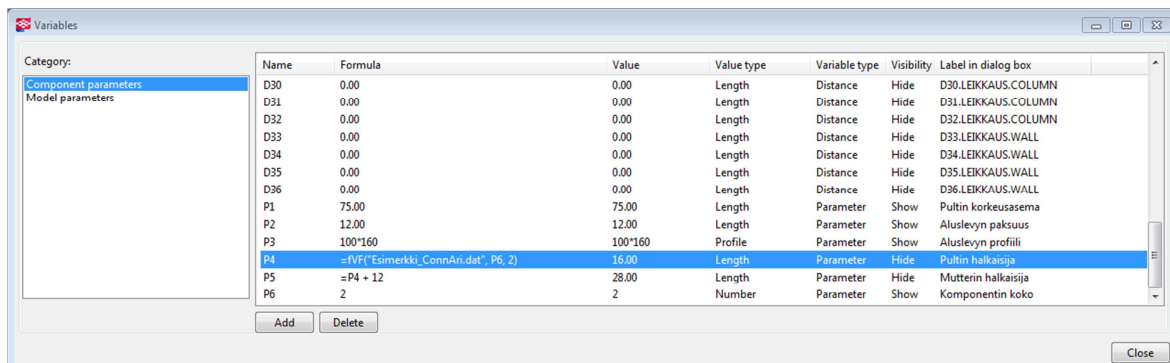


Kuva 47. Etäisyyttä pilariin kasvatettu ja elementtien paksuutta muutettu

Komponentin tulisi pysyä kasassa myös silloin, kun etäisyys pilariin ja elementtien paksuudet muuttuvat. Kaikki näyttäisi olevan kunnossa.

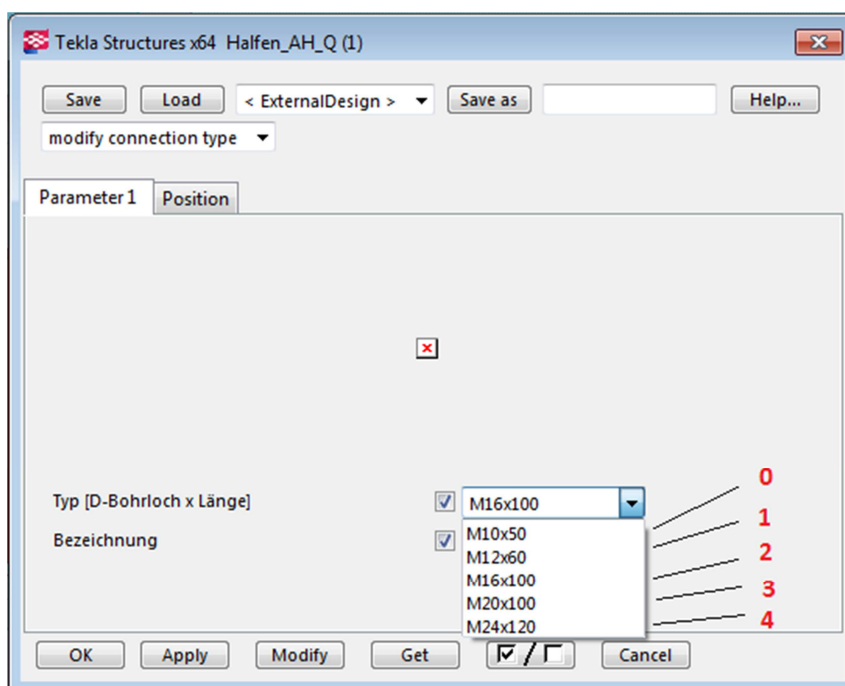
9. Lisäparametrisoinnit ja dialogin muokkaus

Siirrytään takaisin editorin puolelle ja jatketaan parametrisointia. Nyt luodaan muun muassa alasvetovalikko, jonka kautta voidaan helposti valita sisäkierrehylsyn koko.



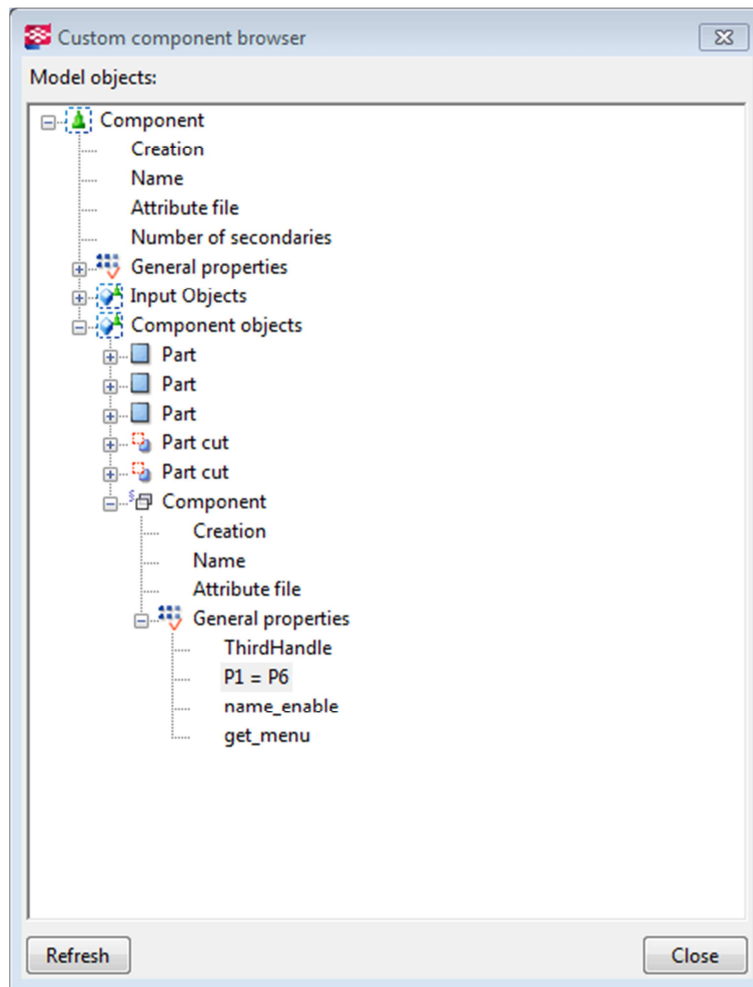
Kuva 48. .dat-tiedostoon viittaus

Lisätään Variables-taulukkoon kohdat P4 - P6. P4 kohdassa määritetään viittaus .dat-tiedostoon, josta haetaan pultin halkaisija. P5 kohdassa määritetään, että mutterin paksuus on aina 12 millimetriä suurempi kuin pultin halkaisija. P6 kohdassa määritetään komponentin koko. P4 ja P5 ovat tyyppiä Length, kun taas P6 tyyppiä Number.



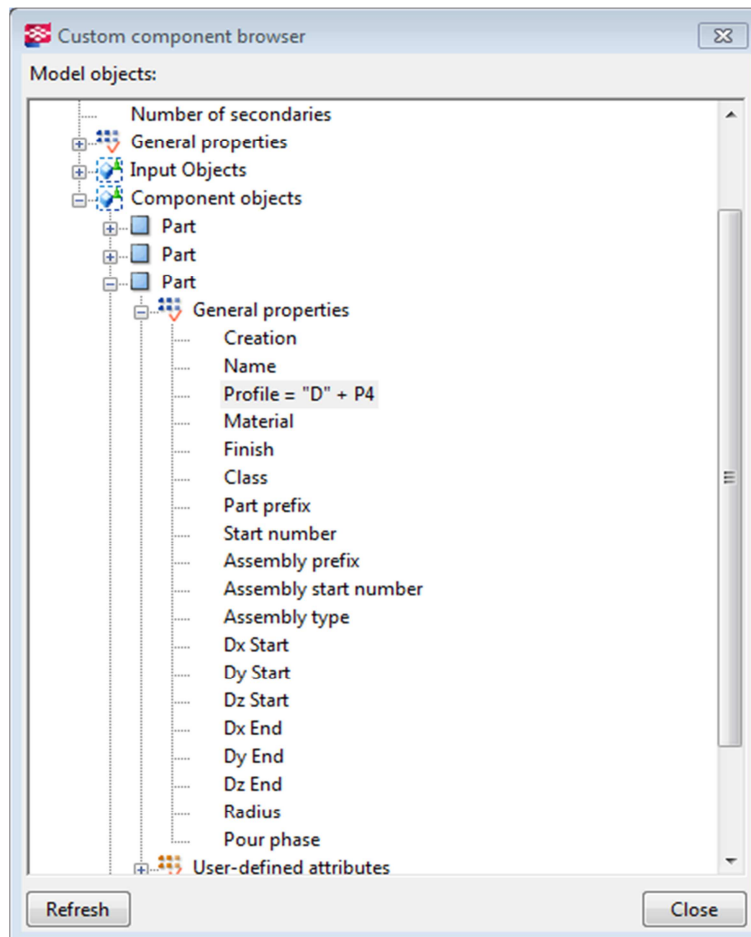
Kuva 49. Sisäkierrehylsyn dialogi

Komponentin kokoa voidaan ohjalla P6-parametrilla luvuilla 0 - 4 kuvan mukaisesti. Ensin on kuitenkin luotava yhteys CC Browserin kautta.



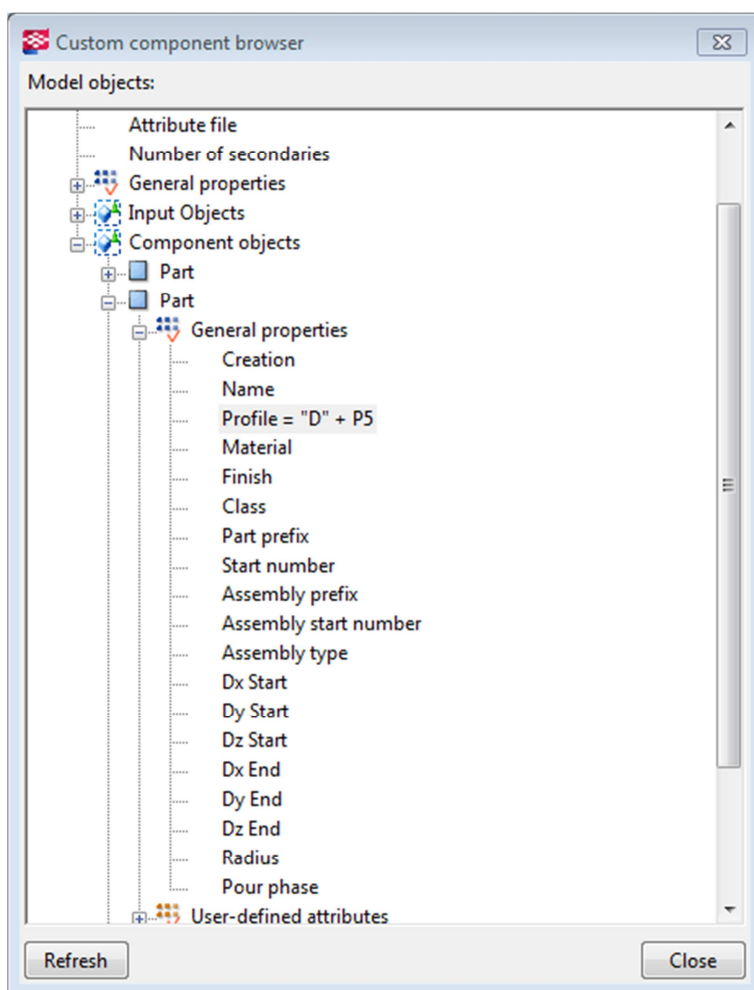
Kuva 50. Yhteyden luominen P6-parametriin

Sisäkierrehylsy komponenttiin lisätään CC Browser -valikon kautta yhteys Variables-valikon P6-parametriin. Kun P6 kohdan arvoja muutetaan nyt välillä 0 - 4, muuttuu alikomponenttina toimivan sisäkierrehylsyn kokokin.



Kuva 51. Yhteyden luominen P4-parametriin

CC Browser -valikon kautta lisätään yhteys pultin halkaisijaan, kirjoittamalla pultin Profile-kohtaan "D" + P4".



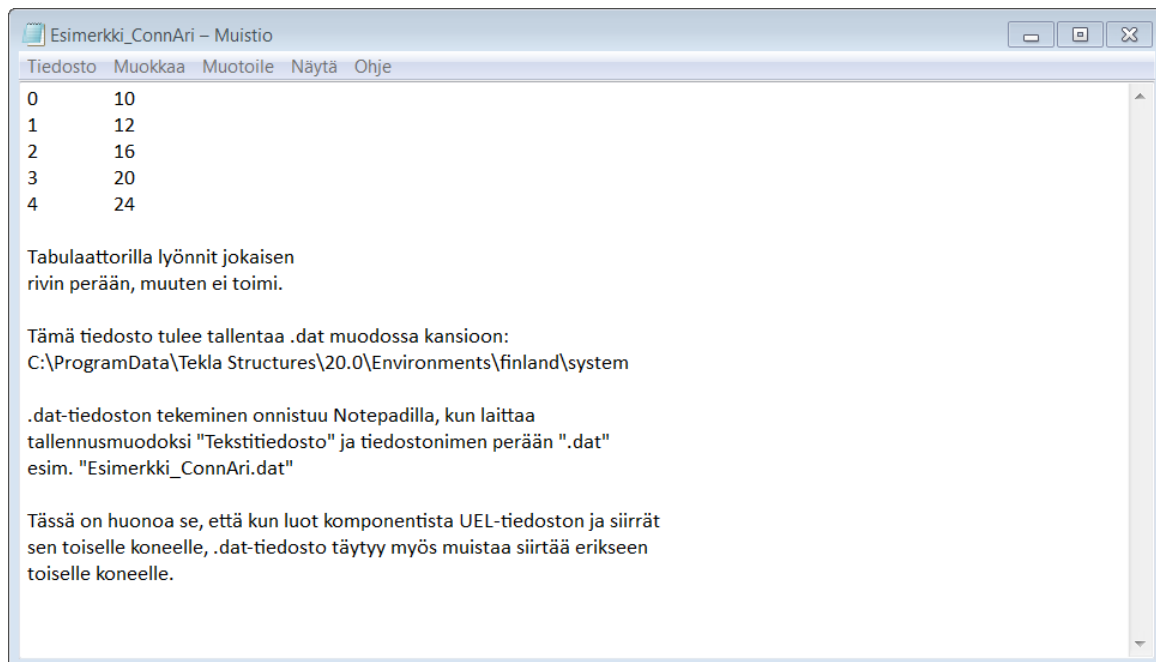
Kuva 52. Yhteyden luominen P5-parametriin

Mutterin halkaisijaan lisätään yhteys kirjoittamalla "= "D" + P5" mutterin Profile-kohtaan.

P4	=fVF("Esimerkki_ConnAri.dat", P6, 2)	16.00	Length	Parameter	Hide	Pultin halkaisija
----	--------------------------------------	-------	--------	-----------	------	-------------------

Kuva 53. P4-parametri

Jotta P4-parametri saadaan toimimaan kuvassa olevalla tavalla, täytyy luoda .dat-tiedosto tiettyyn kansioon (kuva 54.)



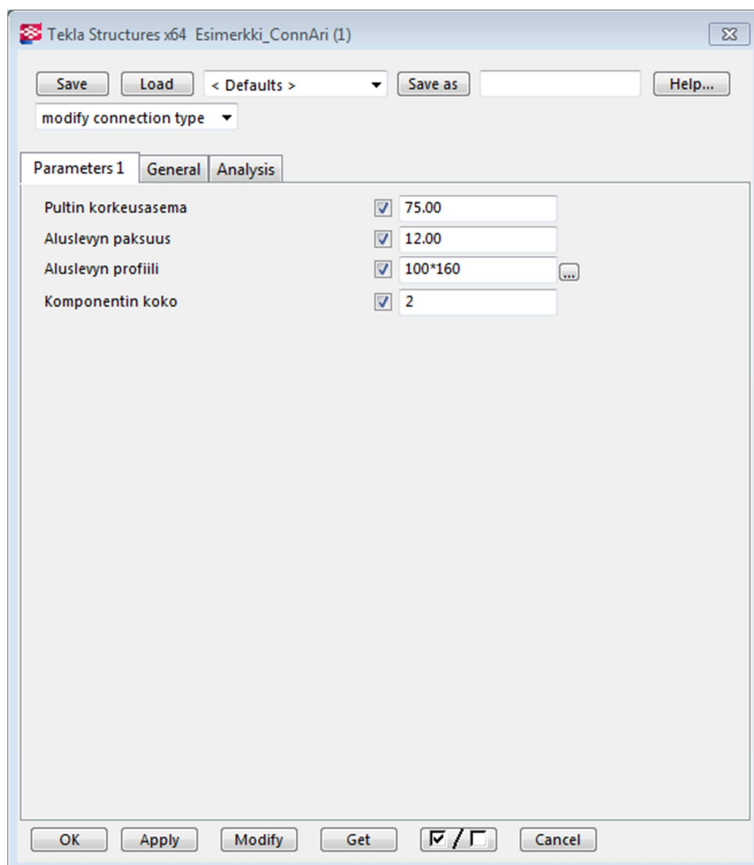
Kuva 54. .dat-tiedosto

P4-parametrin sulkujen sisäinen teksti sisältää kolme kohtaa (kuva 53.). Ensimmäisessä kohdassa ("Esimerkki_ConnAri.dat") ilmoitetaan lainausmerkkien sisällä .dat tiedoston nimi, josta tiedot haetaan. Toisessa kohdassa (P6) ilmoitetaan viittaus parametriin, jonka mukaisesti tiedot poimitaan tietystä riviltä .dat tiedostosta. Kolmannessa kohdassa (2) kerrotaan sarake, josta tieto otetaan .dat-tiedostossa. P4-parametri ottaa siis "Esimerkki_ConnAri.dat" -nimisestä tiedostosta riviltä kaksi (P6-parametrin arvo on tällä hetkellä kaksi) ja sarakkeesta kaksi tiedon 16. Kun P6-parametrin numeroa muutetaan välillä 0 - 4, poimii parametri P4 tietoja eri kohdista kuvan 54. mukaisesti.

```
P4      =if (P6==0) then 10 else if (P6==1) then 12 else if (P6==2) then 16 else if (P6==3) then 20 else if (P6==4) then 24 else 0 endif endif endif endif
```

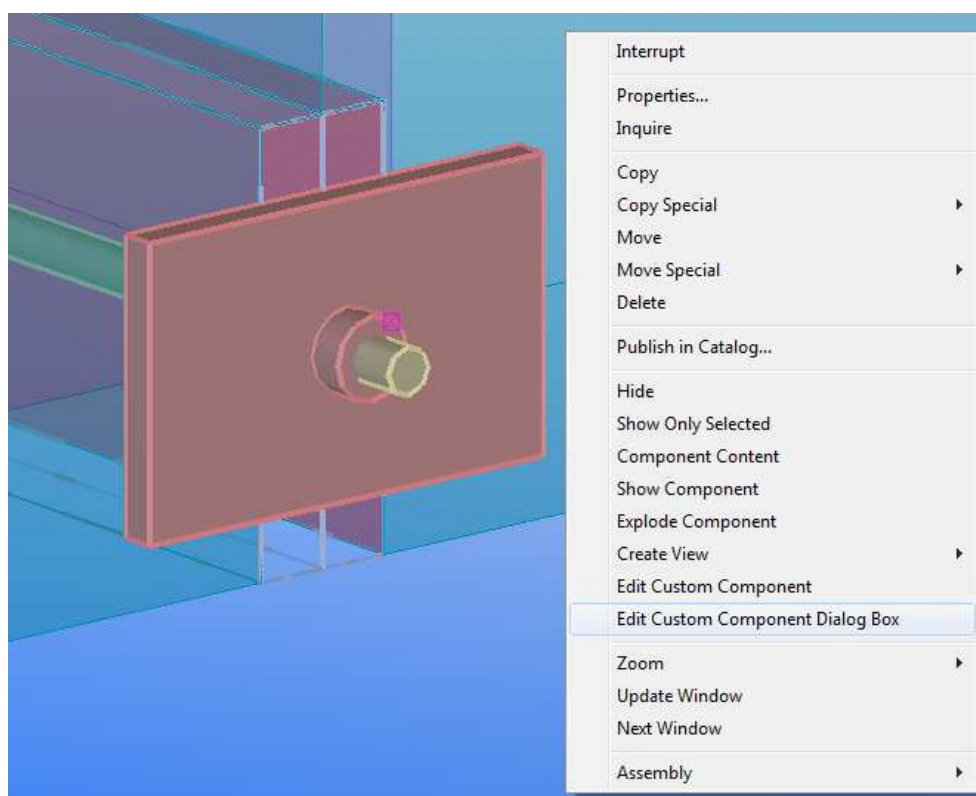
Kuva 55. .dat-tiedoston korvaaminen if-lauseella

.dat-tiedosto voidaan korvata myös if-lauseella, jossa määritetään, mitä jokainen numero (0 - 4) P6-parametrissä aiheuttaa P4-parametrille. Eli esimerkiksi lause "if (P6==0) then 10" tarkoittaa, että jos kohdan P6 arvo on nolla (0), niin kohdan P4 arvo on kymmenen (10).

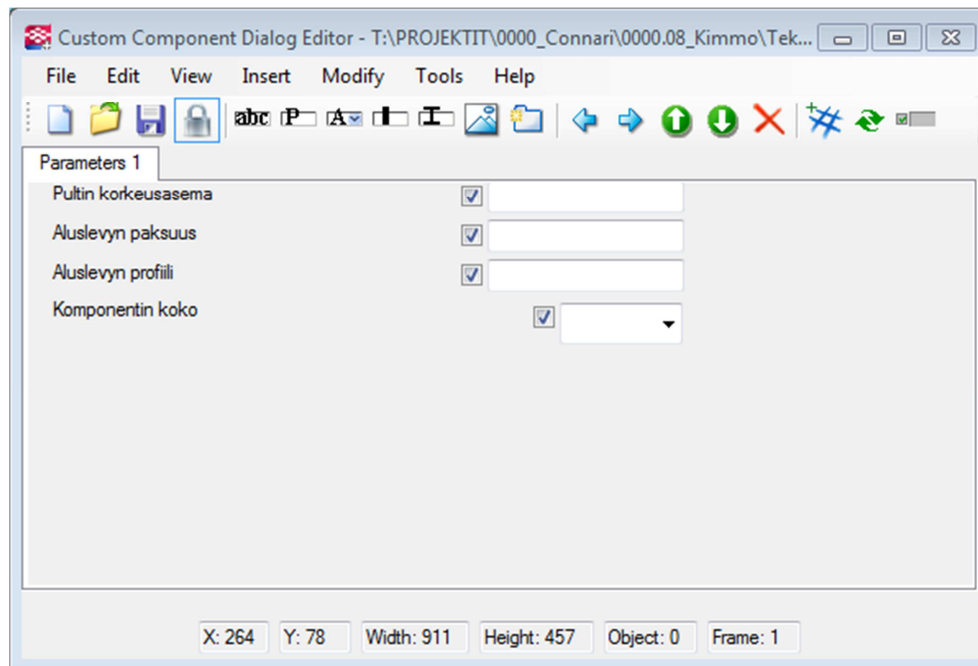


Kuva 56. Komponentin dialogi ennen muokkaamista

Tallennetaan komponentti ja siirrytään pois editorista mallin puolelle, jossa dialogin tulisi nyt näyttää tältä. Komponentin koko ilmoitetaan nyt vain numerolla, eikä se ole oikein selkeä tapa. Aukaistaan dialogin editori ja muokataan sitä hieman.

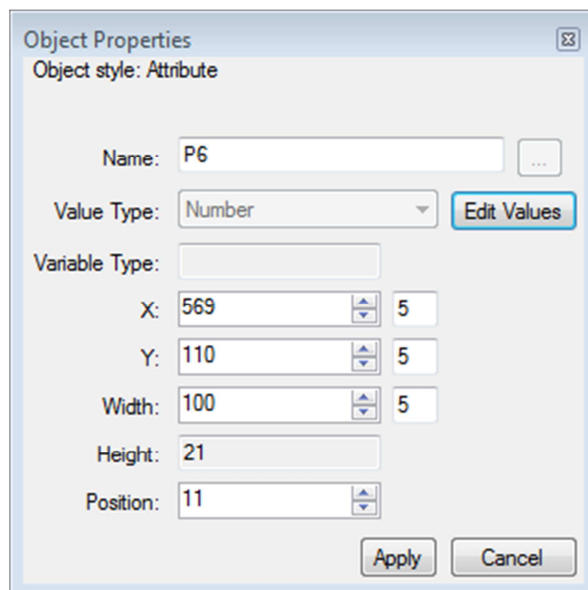


Kuva 57. Dialogin muokkaamistyökaluun siirtyminen



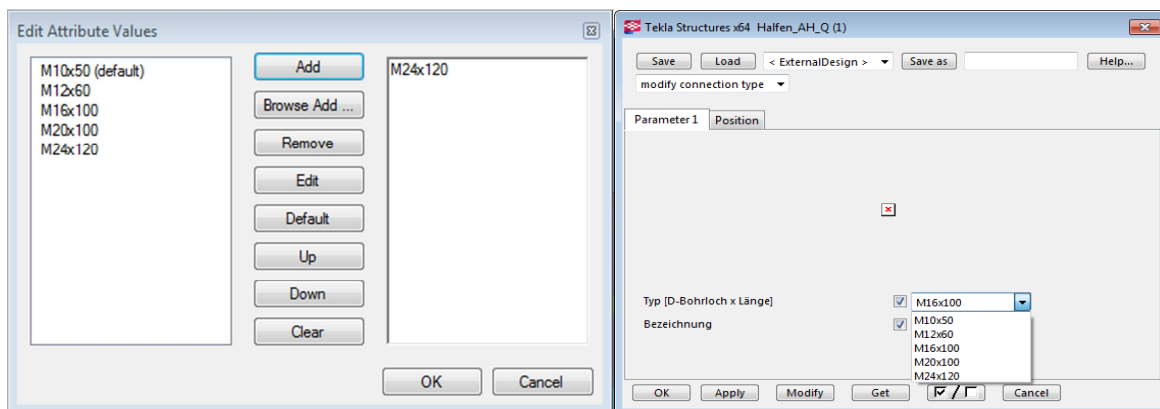
Kuva 58. Custom Component Dialog Editor

Poistetaan Komponentin koko -kohdasta tekstikenttä ja lisätään tilalle alasvetovalikko, joka voidaan ottaa käyttöön editorin yläosasta.



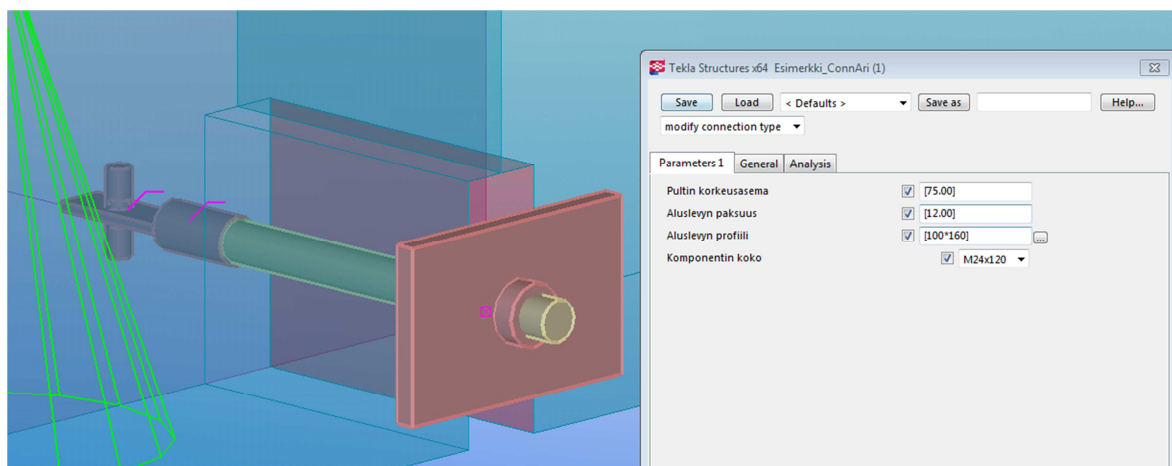
Kuva 59. Alasvetovalikon tiedot

Alasvetovalikon ja sen tilalla olleen tekstikentän nimen tulee olla sama (P6). Klikataan Edit Values -painiketta ja lisätään alasvetovalikkoon arvoja.



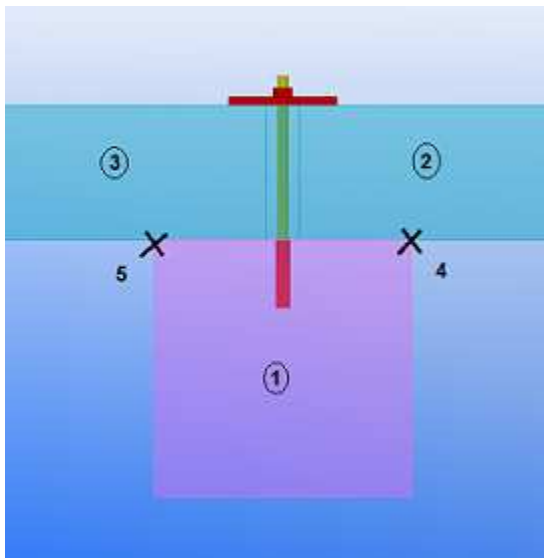
Kuvat 60 ja 61. Arvojen lisääminen alasetovalikkoon

Lisätään alasetovalikkoon samat arvot kuin sisäkierrehylsy komponentissa. Järjestys on oltava sama. Klikataan OK ja tarkistetaan vielä, ettei alasetovalikon nimi ole muuttunut, jonka jälkeen klikataan Apply. Tallennetaan dialogiin tehdyt muutokset ja suljetaan se. Seuraavaksi tallennetaan malli ja avataan Tekla Structures uudestaan, jolloin nähdään dialogiin tehdyt muutokset.



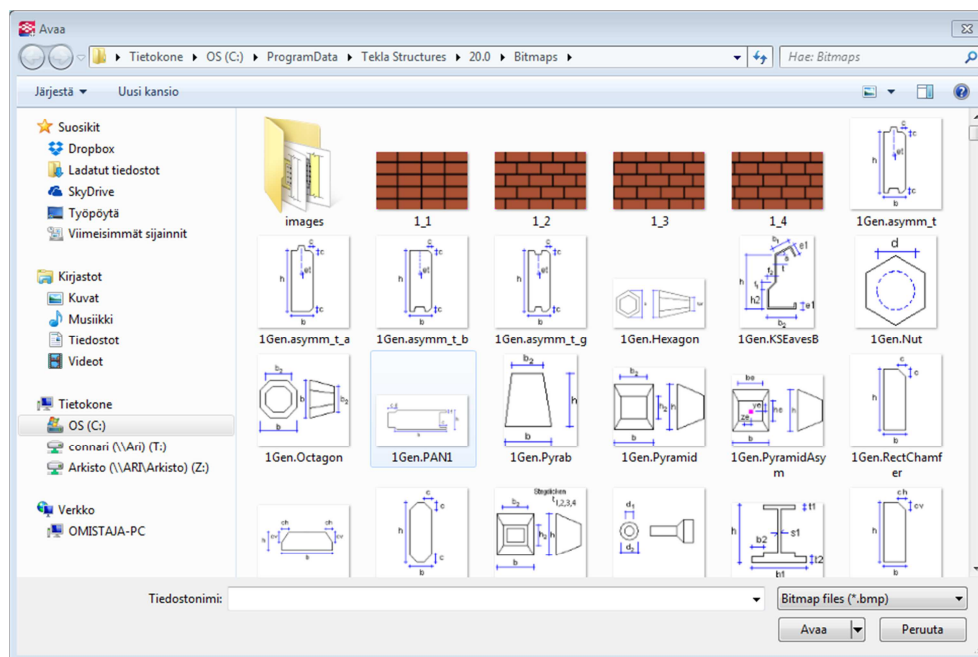
Kuva 62. Dialogin testaaminen

Testataan luotua parametrisoitua komponenttia. Komponentin kokoa muuttamalla, tulisi nyt muuttua myös pultin ja mutterin halkaisija. Komponentti toimii nyt niin kuin pitääkin, mutta sen käyttöliittymässä on vielä parannettavaa.



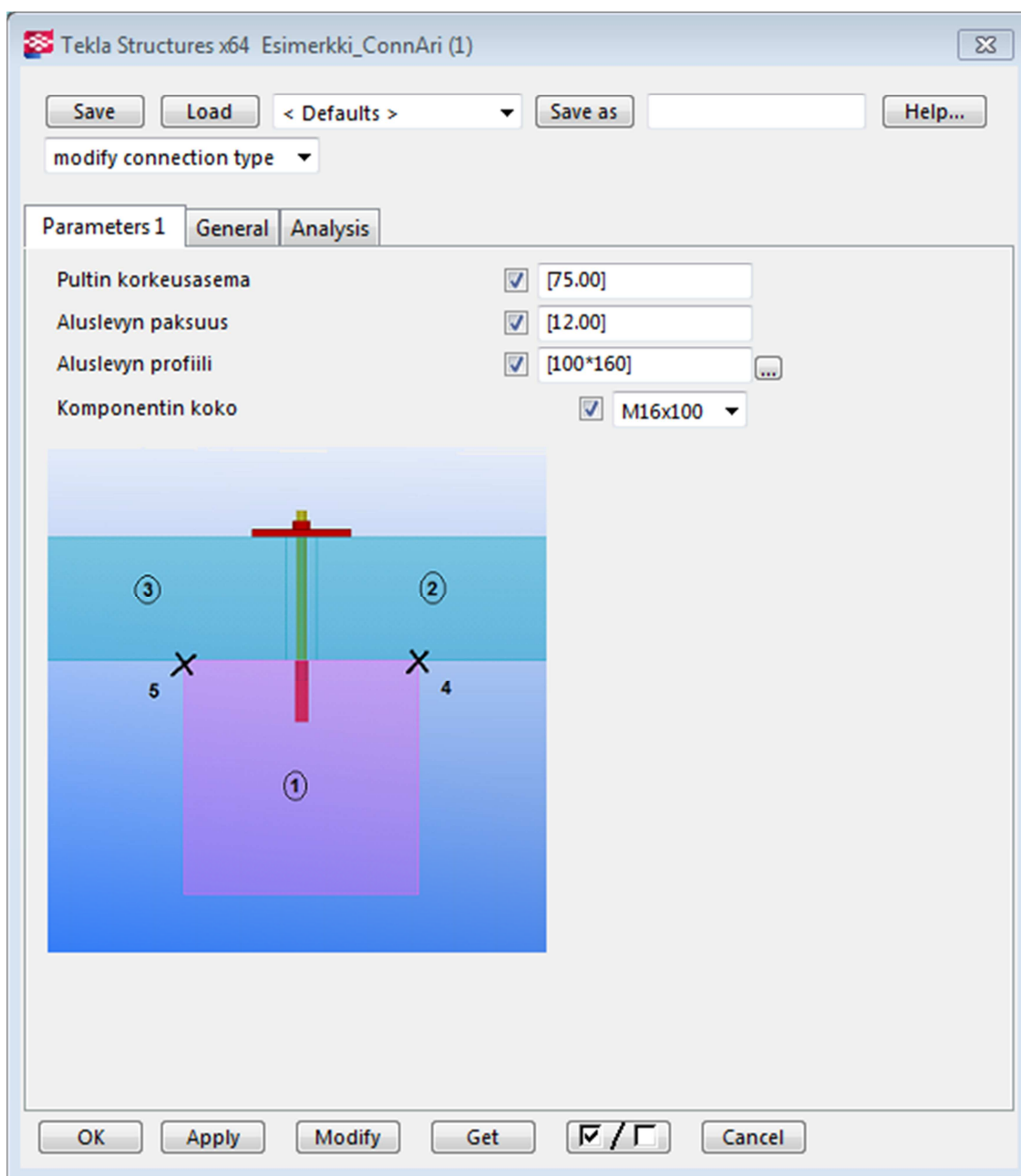
Kuva 63. Kuvaruutukaappaus

Otetaan kuvaruutukaappauksella ylhäältä päin kuva luodusta komponentista ja muokataan sitä hie-
man jollain ohjelmalla (esimerkiksi Paint). Kuvan tulee tallentaa .bmp-muodossa.



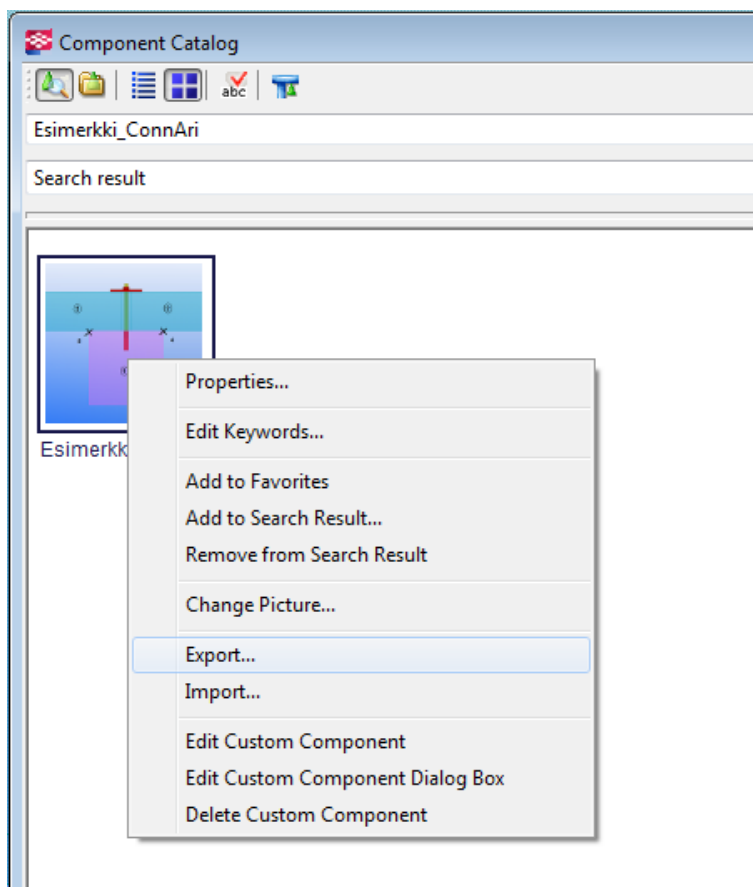
Kuva 64. Bitmaps-kansio

Siirretään kuva Bitmaps-kansioon, josta se voidaan ottaa käyttöön dialogissa. Avataan dialogin edi-
tori uudestaan ja lisätään kuva. Kuvan lisäykseen löytyy painike editorin yläosasta.



Kuva 65. Valmis dialogi

Komponentin valmis käyttöliittymä näyttää nyt tältä. Kuvasta nähdään, missä järjestyksessä osia pitäisi valita, jotta komponentti muodostuu oikein. Kun Custom Component Dialog Editorilla tehdään muutoksia, joutuu Tekla Structuresin aina käynnistämään uudelleen, ennen kuin muutokset tulevat näkyviin dialogissa.



Kuva 66. Pienen kuvan lisääminen ja eksportointi UEL-tiedostoksi

Komponenttiin voidaan vielä lisätä pikku kuva, joka näkyy catalogissa. Lisäys tapahtuu Change Picture... -toiminnolla. Viimeinen asia komponentin tekemisessä on UEL-tiedoston luominen, mikä tapahtuu Export... -toiminnolla. Komponentin tuominen muihin malleihin onnistuu nyt catalogissa Import...-toiminnolla.

10. Yhteenveto

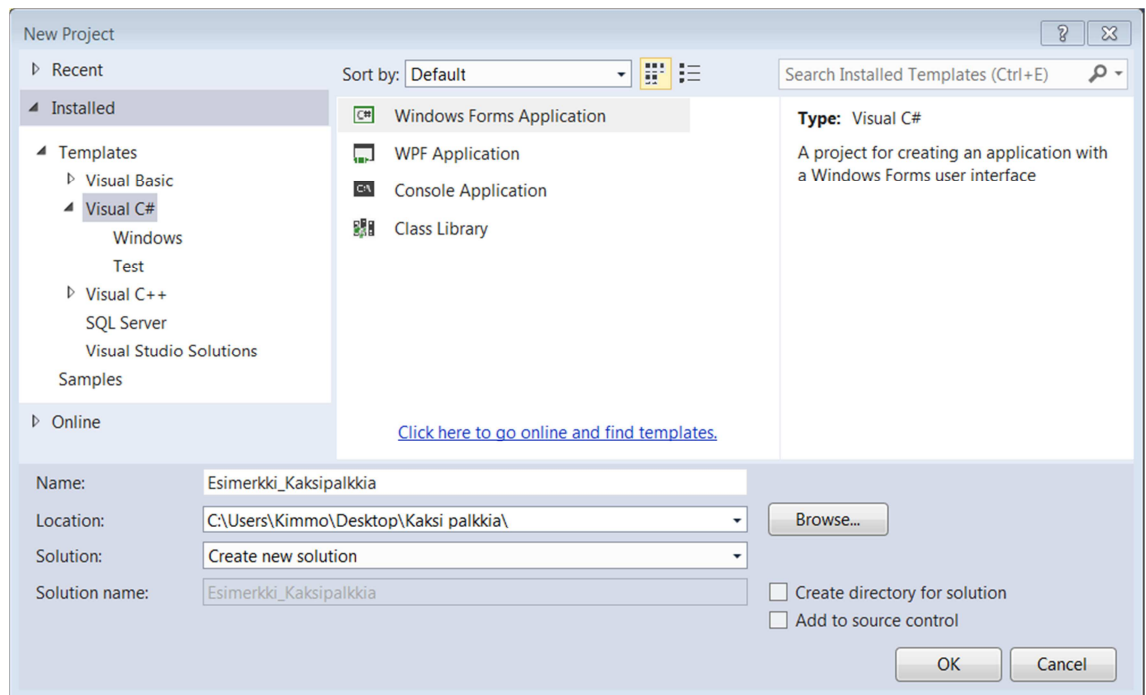
Näissä ohjeissa käsiteltiin yksinkertaisen kiinnityskomponentin luominen. Yrityksen käyttöön luomani komponentit sisälsivät sen verran paljon tietoa, että ohjeiden tekeminen niiden pohjalta olisi ollut erittäin turhauttavaa. Pidän tärkeämpänä esitellä näissä ohjeissa, mitä kaikkea Custom Component Editorilla voidaan tehdä. Monimutkaisempien komponenttien tekeminen onnistuu ohjeiden avulla, joutuu käyttämään vain hieman omaa ajattelua mukana. Esimerkiksi materiaalitietojen lisääminen dialogiin onnistuu aivan yhtä helposti, luomalla yhteyksiä Variables-taulukon ja Custom Component Browser -ikkunan välille.

LIITE 2 ESIMERKKI TEKLA OPEN APIN KÄYTTÄMISESTÄ

Välineet:

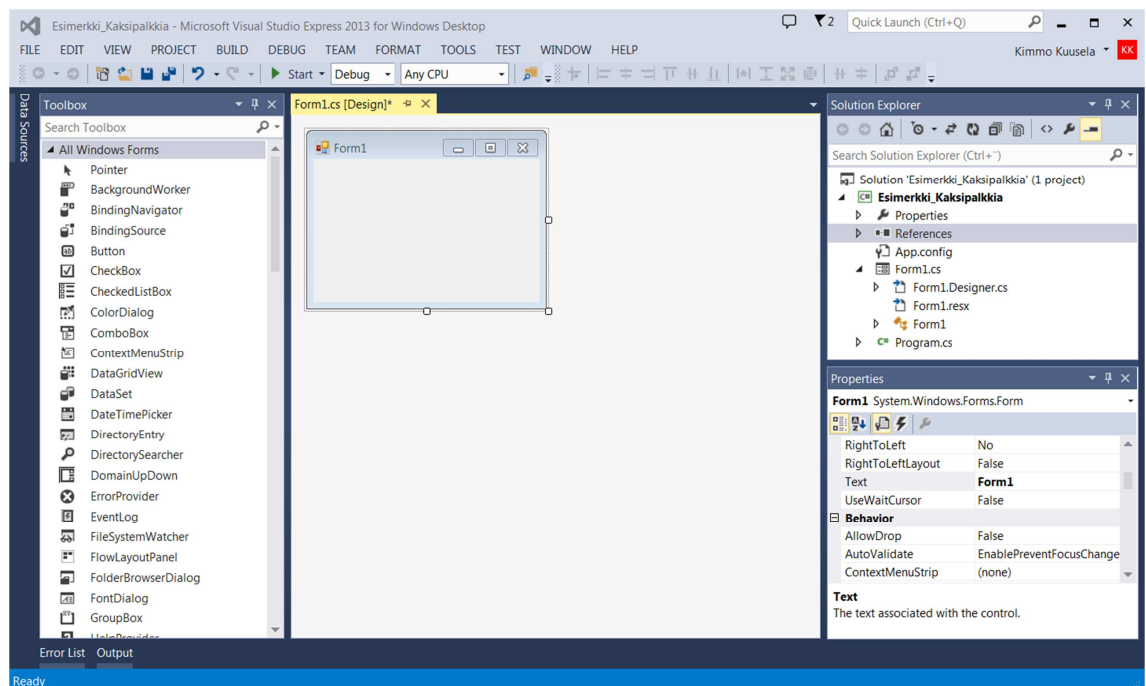
Tekla Structures 19.1

Microsoft Visual Studio Express 2013

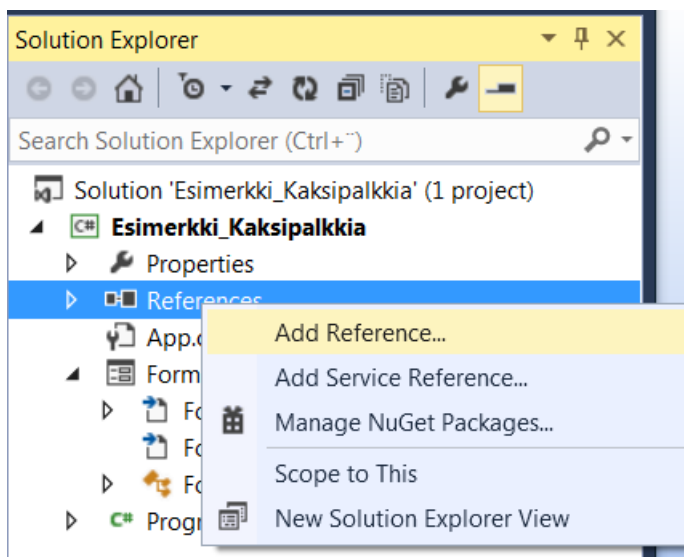


Kuva 1. Uusi projekti

Aloitetaan tekemällä C#-koodikielen pohjautuva projekti Visual Studiolla.



Kuva 2. Lähtötilanne

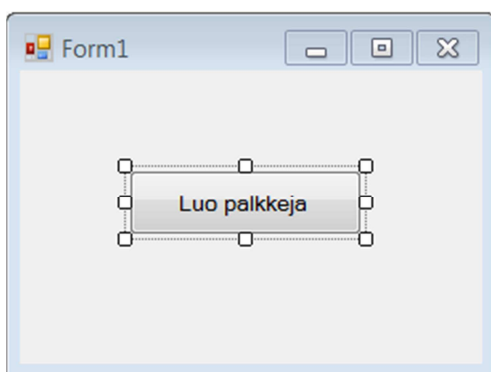


Kuva 3. Referenssien eli viittauksen lisäys



Kuva 4. Referenssien sijainti

Lisätään referenssit projektiin esimerkiksi Solution Explorer -ikkunan kautta. Tässä esimerkissä tarvitaan seuraavia referenssejä: Tekla.Structures.dll ja Tekla.Structures.Model.dll.



Kuva 5. Painike

Lisätään lomakkeeseen painike (Button) Toolboxin kautta. Painikkeen nimeä voidaan muuttaa Properties -valikon kautta kohdassa Text. Klikataan painiketta kahdesti ja aletaan kirjoittaa koodia sen taakse. Vihreällä oleva teksti on vain koodiin kirjoitettua ohjetta, eikä vaikuta applicationin toimintaan.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Tekla.Structures.Model; //Sisältää mm. Model, Beam ja Position
using Tekla.Structures.Model.UI; //Sisältää Picker
```



```

namespace Esimerkki_Kaksipalkkia
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            // Luodaan ensimmäisenä yhteys avoimeen malliin
            Model omaMalli = new Model();

            // Luodaan palkki ja picker, jolla voidaan
            // valita mallista sijainti palkille
            Beam omaPalkki = new Beam();
            Picker omaPicker = new Picker();

            // Annetaan seuraavaksi palkille tiedot kuten nimi,
            // materiaali, profiili yms.
            omaPalkki.Name = "Palkki";
            omaPalkki.Profile.ProfileString = "HEA300";
            omaPalkki.Material.MaterialString = "S355JR";
            omaPalkki.Class = "3";

            // Määritetään seuraavaksi palkin sijainti mallissa
            omaPalkki.StartPoint = omaPicker.PickPoint();
            omaPalkki.EndPoint = omaPicker.PickPoint();

            // Määritetään miten palkki muodostuu
            // suhteessa alku- ja loppupisteisiin
            omaPalkki.Position.Depth = Position.DepthEnum.BEHIND;
            omaPalkki.Position.Plane = Position.PlaneEnum.MIDDLE;
            omaPalkki.Position.Rotation = Position.RotationEnum.TOP;

            // Lisätään palkki malliin
            omaPalkki.Insert();

            // Päivitetään muutokset malliin
            omaMalli.CommitChanges();

            // Luodaan toinen palkki
            Beam omaPalkki1 = new Beam();
            Picker omaPicker1 = new Picker();

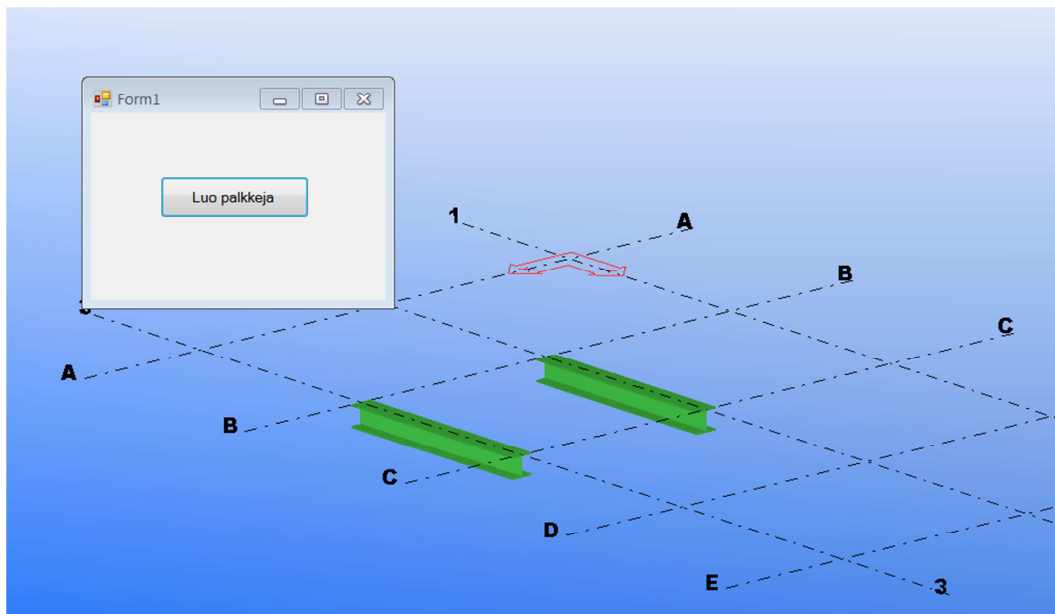
            // Määritetään toisenkin palkin sijainti mallissa
            omaPalkki1.StartPoint = omaPicker1.PickPoint();
            omaPalkki1.EndPoint = omaPicker1.PickPoint();

            // Kopioidaan ensimmäisen palkin tiedot toiselle palkille
            omaPalkki1.Name = "Palkki";
            omaPalkki1.Profile.ProfileString = "HEA300";
            omaPalkki1.Material.MaterialString = "S355JR";
            omaPalkki1.Class = "3";
            omaPalkki1.Position.Depth = Position.DepthEnum.BEHIND;
            omaPalkki1.Position.Plane = Position.PlaneEnum.MIDDLE;
            omaPalkki1.Position.Rotation = Position.RotationEnum.TOP;
            omaPalkki1.Insert();

            // Päivitetään muutokset malliin, muuten näkyvä joudutaan päivittämään
            // manuaalisesti ennen kuin muutokset tulevat näkyviin
            omaMalli.CommitChanges();
        }
    }
}

```

Testataan seuraavaksi applicationin toimintaa Tekla Structuresissa. Tätä varten TS tulee aukaista. Nyt kun TS ja Visual Studio ovat kumpikin auki, painetaan Visual Studiassa F5, jolloin ohjelman tulisi käynnistyä.



Kuva 6. Palkkien luominen Tekla Structures -ohjelmistoon

Kun Luo palkkeja -painiketta klikataan, ohjelmisto kysyy käyttäjältä kahteen otteeseen kaksi pistettä, joiden mukaan se luo kaksi palkkia malliin. Palkkien tiedot on määritelty koodissa.