



# **Teräsristikko tasokehän rautalankamallista 3d-malliksi sovellusohjelman avulla**

Ville Korkiamäki

Opinnäytetyö  
Huhtikuu 2012  
Rakennustekniikan koulutusohjelma  
Talonrakennustekniikka

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Rakennustekniikan koulutusohjelman  
Talorakennustekniikka

VILLE KORKIAMÄKI:

Teräsristikko tasokehän rautalankamallista 3d-malliksi sovellusohjelman avulla

Opinnäytetyö 27 sivua  
Huhtikuu 2012

---

Tämän opinnäytetyön tarkoituksena oli tuottaa tilaajalle ohjelma, jonka avulla voidaan nopeuttaa teräsristikon mallintamista ja vähentää mallintamisessa syntyviä virheitä. Tavallisesti ristikko on mallinnettu erikseen laskentaohjelmaan ja mallinnusohjelmaan. Tässä opinnäytetyössä on pyritty karsimaan tätä päällekkäisyyttä ohjelmoinnin avulla. Opinnäytetyönä tehty ohjelma lukee XML-tiedoston, johon on laskentaohjelmasta tulostettu kaikki ristikon mallintamiseen tarvittava tieto, ja luo tietojen pohjalta ristikon Tekla Structures 3d-malliin. Opinnäytetyöohjelma toteutettiin Visual Basic .Net -ohjelmointikielellä.

Toteutuneesta ohjelmasta tuli asetettujen tavoitteiden mukainen. Ohjelmalla pystytään luomaan ristikko kahdella tarkkuudella. Kun ristikkoa luodaan yksinkertaistetulla geometrialla, ohjelma mallintaa ristikon osat laskentamallin mukaisia solmupistetietoja noudattaen. Tämä tarkkuus on riittävä ristikon painon suuruusluokan arvioimiseen ja näin tarjouslaskennan tekoon. Kun ristikko luodaan tarkalla geometrialla, ohjelma pyytää käyttäjää määrittämään laskentaohjelmasta saaduille solmupisteille uudet koordinaattipisteet 3d-mallista. Tämän jälkeen ohjelma lukee laskentaohjelmasta saadut solmupisteiden liitostiedot ja mallintaa kuhunkin solmupisteeseen liittyvät sauvaosat siten, että liitoksesta tulee halutun kaltainen. Tarkalla geometrialla mallinnettaessa ohjelma viimeistelee ristikon, leikkaamalla ja sovittamalla osat siten, ettei päällekkäisyyksiä pääse syntymään. Lopuksi ohjelma vielä liittyy luodut osat yhdeksi kokoonpanoksi siten, että ristikon alapaarteesta tulee ristikon pääosa. Ristikon luomisen lisäksi opinnäytetyöohjelmalla pystytään muokkaamaan ristikon sauva- ja solmupistetietoja ja tallentamaan nämä uudelleen XML-tiedostoksi myöhempää käyttöä varten.

## ABSTRACT

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Construction Engineering  
Building Construction

VILLE KORKIAMÄKI:  
Steel Truss from Calculation Software into a 3d-Model

Bachelor's thesis 27 pages  
April 2012

---

The meaning of this thesis was to produce a program which makes it easier and faster to create a 3d-model of a steel truss structure. Another meaning for this program is to reduce the errors made in modeling. Usually you need to create two different truss models, one for the calculation software, and other for the modeling software. With the program, made for this thesis, we have tried to eliminate this unnecessary work. The program reads the truss data that has been stored into a XML-file by the calculation software. After this the program creates a truss structure into the Tekla Structures 3d-model, based on the information that was found in the XML-file. The program was created with Visual Basic .NET 2010 programming language.

The finished program came out to be as it was originally designed. You can create a truss with two different definitions. When you are creating a truss with simplified geometry the program will create the parts of a truss with the same geometry that was used in the calculation software. This accuracy is close enough to estimate the weight of the structure and so you can calculate an offer based on that knowledge. When you are creating a truss with exact geometry you need to give new node coordinates from the 3d-model to the program. After this the program reads the node data from the XML-file and creates the truss parts based on that knowledge. With the program made for this thesis you can also modify the truss data and save it as new XML-file.

---

Key words: modeling, steel truss, programming, Visual Basic .Net

## SISÄLLYS

1	JOHDANTO.....	5
1.1	Tausta.....	5
1.2	Tavoitteet .....	6
1.3	Rajaukset.....	6
1.4	Aineisto ja menetelmät .....	6
2	OHJELMAN SUUNNITTELU.....	8
2.1	Ohjelman toiminta .....	8
2.2	Käytetyt ohjelmistot ja tekniikat.....	9
2.2.1	Visual Basic .NET.....	9
2.2.2	Tekla Structures .....	9
2.2.3	XML.....	10
3	OHJELMAN RAKENNE JA TOTEUTUS .....	11
3.1	Ohjelman toimintaperiaate.....	11
3.2	XML-tiedosto.....	12
3.3	Käyttöliittymä .....	13
3.4	Ristikon luominen.....	15
3.5	Tiedon säilyttäminen ja muokkaaminen ohjelmassa .....	17
3.6	Koodaaminen .....	18
3.7	Testaaminen .....	18
4	ESIMERKKI OHJELMAN KÄYTÖSTÄ .....	19
4.1	Ristikko yksinkertaisen geometrian mukaan .....	19
4.2	Ristikko tarkan geometrian mukaan .....	22
5	JATKOKEHITYS .....	25
5.1	Ohjelman automaation parantaminen .....	25
5.2	Tekla Structures plug-in.....	25
5.3	Ohjelman testaaminen ja laajentaminen .....	25
6	YHTEENVETO .....	26
	LÄHTEET.....	27

# 1 JOHDANTO

## 1.1 Tausta

Tämä opinnäytetyö tehtiin suunnittelutoimisto SS-Teracon Oy:n tilauksesta. Teollisuushallien ja julkisten rakennusten rakennesuunnitteluun erikoistunut yritys on perustettu Tampereella vuonna 1992. Perustamisen jälkeen yritys on avannut toimipisteet myös Turkuun ja Vaasaan. Nykyään SS-Teracon suunnittelee teräs- ja betonirakenteita Suomen lisäksi muuhun Skandinaviaan ja Venäjälle. Opinnäytetyön valvojana SS-teraconilta toimi Olli Hirvonen.

Rakennusala elää jatkuvan muutoksen keskellä. Tietotekniikan yleistymisen myötä toimistoihin tulvii uusia ohjelmistotyökaluja, jotka kaikki pyrkivät olemaan edeltäjiään tehokkaampia, yksinkertaisempia, nopeampia ja laaja-alaisempia. Myös suunnittelutoimistojen kesken on menossa kehityksen kilpajuoksu, jossa jokaisen toimiston on tehtävä parhaansa pysyäkseen ajanhermoilla ja kilpailukykyisenä kilpailijoihinsa nähden.

Saatavilla olevat suunnitteluohjelmistot eivät useinkaan taivu sellaisenaan yksittäisen suunnittelutoimiston erityistarpeisiin ja -tapoihin. Tästä syystä isoimmilla suunnittelu-toimistoilla onkin omat ohjelmointigurunsä ja kehitystiimensä, jotka pyrkivät valjastamaan suunnitteluohjelmistojen koko kapasiteetin yrityksen käyttöön. Ohjelmoinnin avulla suunnittelutyöstä pyritään tekemään mahdollisimman selkeää ja tehokasta. Lisäksi ohjelmoinnilla pyritään pienentämään inhimillisten virheiden riskiä.

Perinteisesti ristikkorakenteiden mallinnus on suunnittelutoimistoissa tehty kahteen kertaan. Ensiksi pilari-ristikkokehä on mallinnettu rautalankamallina laskentaohjelmaan, jossa on suoritettu rakenteen mitoitus. Tämän jälkeen ristikko on siirretty mittoineen ja profiilitietoinen 3d-mallinnusohjelmaan. Mallinnusohjelman avulla ristikosta ja muista rakenteista pystytään tehokkaasti tuottamaan tarvittavat piirustukset ja luettelot. Tämä edellä kuvattu toimintatapa on melko hidas ja virhealtis. Tässä opinnäytetyössä on etsitty ohjelmoinnin kautta ratkaisua, jolla saataisiin siirrettyä laskentaohjelmassa tehty ristikkorakenteiden mallinnustyö suoraan mallinnusohjelmaan.

## 1.2 Tavoitteet

Opinnäytetyön tavoitteena on tuottaa ohjelma, joka siirtää laskentaohjelmaan mallinnetun teräsristikon mallinnusohjelmaan. Näin ohjelma nopeuttaa teräsristikon mallintamista sekä vähentää ristikon käsinmallinnuksessa syntyviä virheitä. Ohjelman suunnittelussa pyritään huomioimaan ohjelman tuleva kehitystyö ja laajentaminen. Lisäksi opinnäytetyön tarkoituksena on tutustua Tekla Structures -ohjelman rajapintaohjelmointiin ja näin kartuttaa tilaajan ohjelmointitietoa ja -taitoa.

Opinnäytetyöhön kuuluu myös ohjelman toteuttamiseen tarvittavan siirtotiedoston laatiminen. Tiedostolle pitää valita tiedostomuoto ja hahmotella sen rakenne. Siirtotiedoston tulee sisältää kaikki ristikko-osien mallinnukseen tarvittava tieto sekä tulevaisuuden kehitystyötä silmällä pitäen myös muuta helposti laskentaohjelmasta saatavilla olevaa tietoa.

## 1.3 Rajaukset

Ohjelma suunnitellaan yksinkertaisille lape- ja harjaristikoille, joissa on enintään kaksi yläpaarretta ja yksi alapaarre. Ristikon liitokset pilareihin voivat olla joko nivellisiä tai jäykkiä. Ristikko-osat voivat olla joko neliö- tai suorakaideputkia ja yhteen solmupisteeseen voi osien lisäksi liittyä enimmillään kaksi diagonaalisauvaa ja yksi vertikaalisauva. Näillä edellä mainituilla rajauksilla pystytään toteuttamaan ohjelma, jonka toimintakenttä kattaa suurimman osan suunnittelutoimistoissa suunniteltavista teräsristikoista.

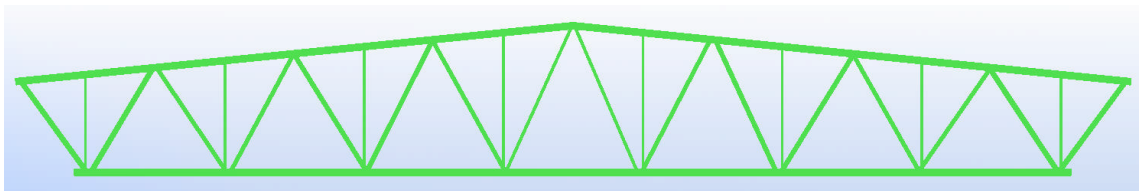
## 1.4 Aineisto ja menetelmät

Ohjelmointi toteutettiin käyttämällä Microsoftin kehittämää Visual Basic .NET -ohjelmointikieltä. Ohjelmointikieleen perehtyminen tapahtui aluksi Michael Halvorsonin kirjoittaman Visual Basic 2008 Tehokas hallinta -kirjan avulla. Tämä lisäksi tietoa haettiin internetistä. Parhaaksi tiedon lähteeksi osoittautui videopalvelu Youtube, jonne on ladattu roppakaupalla yksityiskohtaisia esimerkkejä siitä, miten erilaisia asioita saadaan helposti toteutettua Visual Basic .NET -kielellä.

Tekla Structures -mallinnusohjelmaan ja sen ohjelmointiin tutustuminen tapahtui Teklan oman Extranet-palvelun kautta. Extranetistä löytyy hyviä oppaita ja esimerkkejä siitä kuinka ohjelmallisesti pystytään ottamaan yhteys Tekla Structures -mallinnusohjelmaan ja suorittamaan siellä erilaisia toimintoja. Tämän lisäksi Teklan Extranetistä löytyy käyttäjäfoorumi, jonka Open Api -osiossa mallinnusohjelman käyttäjät ympäri maailmaa ovat esittäneet ohjelmointiin liittyviä kysymyksiä ja vastauksia.

## 2 OHJELMAN SUUNNITTELU

Ohjelmaa suunniteltiin yhdessä opinnäytetyön valvojan kanssa. Ensimmäisessä palaverissa määriteltiin ohjelman tavoitteet ja rajaukset. Suunnittelu pohjautui laskentaohjelmista saatavilla olevaan tietoon sekä laadittuun esimerkkiristikoon (kuva 1). Esimerkkiristikon avulla käytiin läpi ristikkomallintamisen vaiheita, ja mietittiin sen pohjalta ohjelman toimintaa. Ohjelmoinnin ohessa pidetyissä palavereissa keskusteltiin ohjelmoinnissa esiin nousseista ongelmista ja ideoista. Ohjelman suunnittelussa pyrittiin siihen, että toteutettu ohjelma olisi selkeä, helppokäyttöinen ja luotettava apuväline ristikkoiden mallintamisessa.



KUVA 1. Esimerkkiristikko

### 2.1 Ohjelman toiminta

Ohjelma toteutetaan siirtotiedoston avulla, johon tulostetaan laskentaohjelmasta ristikon sauva- ja solmupistetiedot. Siirtotiedoston tiedostomuodoksi valittiin XML-formaatti, koska se on yleiskäyttöinen ja laajasti hyväksytty tiedostomuoto. XML-tiedosto on myös helposti luettavissa ja kirjoitettavissa ohjelmallisesti.

Varsinainen opinnäytetyöohjelma toimii siten, että se lukee siirtotiedostosta ristikko-osien profiili- sekä paikkatiedot ja luo näiden tietojen pohjalta osat 3d-malliin. Ristikko voidaan luoda kahdella tarkkuudella. Laskentavaiheessa ristikko luodaan ilman yksityiskohtaisempaa sauvojen sijoittelua. Tällöin ristikon sauvat mallinnetaan alkuperäisen tasokehämallin mukaisia keskilinjoihin noudattaen. Tämä ristikon mallintamisen tarkkuus on riittävä ristikon painon suuruusluokan arvioimiseen ja näin tarjouslaskennan tekoon. Tuotantokuvientekovaiheessa ristikon sauvojen liitokset ja liitoksetäisyydet mallinnetaan tarkasti. Tällöin tasokehämallista tulostettujen ristikko-osien profiili- ja paikkatietojen lisäksi tarvitaan jokaisen solmukohtaan liitostiedot, jotka myös ovat tallennettuna siirto-



tiedostoon. Ohjelma lukee liitostiedot ja mallintaa kuhunkin solmupisteeseen liittyvät sauvaosat liitostiedoissa kuvatulla tavalla. Ristikon luomisen lisäksi opinnäytetyöohjelmalla on pystyttävä muokkaamaan ristikon sauva- ja solmupistetietoja ja tallentamaan nämä uudelleen siirtotiedostoksi myöhempää käyttöä varten.

## **2.2 Käytetyt ohjelmistot ja tekniikat**

### **2.2.1 Visual Basic .NET**

Visual Basic .NET(VB.NET) on yleiskäyttöinen ohjelmointikieli, jonka Microsoft kehitti alunperin Basic-ohjelmointikielen pohjalta 1990-luvun alussa. Vuonna 2002 kieltä kehitettiin radikaalisti, jolloin se liitettiin osaksi Microsoftin .NET-perhettä. VB.NET-kielen viimeisin versio on huhtikuussa 2010 julkaistu Visual Basic 2010. VB.NET-ohjelmoinnissa käytetään yleisesti Microsoftin Visual Studio -ohjelmistoa, joka on myös osa Microsoftin .NET-konseptia. VB.NET muistuttaa nykyään melko paljon C-sukuisia kieliä. VB.NET-kielellä ohjelmoidut ohjelmat toimivat ainoastaan Windows-käyttöjärjestelmissä ja ohjelmien ajamiseen tarvitaan lisäksi .NET frameworkin viimeisin versio. (Wikipedia, 2012.)

### **2.2.2 Tekla Structures**

Tekla Structures on Tekla Oyj:n kehittämä rakennuksen tietomallinnusohjelmisto, jota voidaan käyttää apuna mm. rakennesuunnittelussa ja projektin hallinnassa. Tekla Structures on lanseerattu vuonna 2004, ja se pohjautuu alun perin vuonna 1992 julkaistuun teräsrakenteiden suunnitteluohjelmaan nimeltä Xsteel. Tekla Structures -ohjelmistolla kehitettyjä rakennemalleja voidaan rakennusprojektissa hyödyntää koko rakennusprojektin ajan, luonnosvaiheen suunnittelusta aina toteutumien tarkkailuun asti. Ohjelmistolla pystytään luomaan erilaisia kolmiulotteisia teräs-, betoni- sekä elementtirakenteita ja luomaan näistä vaivattomasti erilaisia luetteloita ja piirustuksia. (Tekla.com, 2012.)

Tekla Structures sisältää Teklan kehittämän ohjelmointirajapinnan nimeltä Open API (Application Programming Inteface). Rajapinnan ansioista ulkopuoliset toimijat voivat

kehittää ohjelmaan erilaisia lisäsovelluksia ja -toimintoja. Rajapinta on toteutettu Microsoftin .NET-tekniikan avulla. Rajapinnan kautta voidaan siirtää dataa mallinnusohjelman ja minkä tahansa COM- tai .NET-yhteensopivan ohjelman välillä. (Tekla.com, 2012.)

### 2.2.3 XML

XML (Extensible Markup Language) on merkintäkieli, joka asettaa dokumentin tiedon merkitsemiselle joukon sääntöjä siten, että tiedon pystyy ymmärtämään niin ihminen kuin tietokonekin. XML-kieltä voidaan käyttää tiedonvälitykseen järjestelmien ja ohjelmien välillä sekä dokumenttien tallentamiseen. Kielen on kehittänyt World Wide Web Consortium, ja kielen ensimmäinen versio otettiin käyttöön vuonna 1998. XML-dokumentin rakenne seuraa aina tiettyjä lainalaisuuksia. Tiedosto alkaa aina prologilla, jossa kerrotaan käytettävä XML-versio sekä muita lähtötietoja. Tämän jälkeen tulee koodiosuus. Koodiosuus koostuu elementeistä, joille on määritelty alku- ja loppumerkki. Oikein kirjoitetussa XML-tiedostossa on yksi juurielementti, jonka sisään kaikki muut elementit tietoisena tulevat. Elementeille voidaan antaa arvoja, ja niitä voidaan järjestellä rajattomasti sisäkkäin ja peräkkäin. (Wikipedia:XML.)

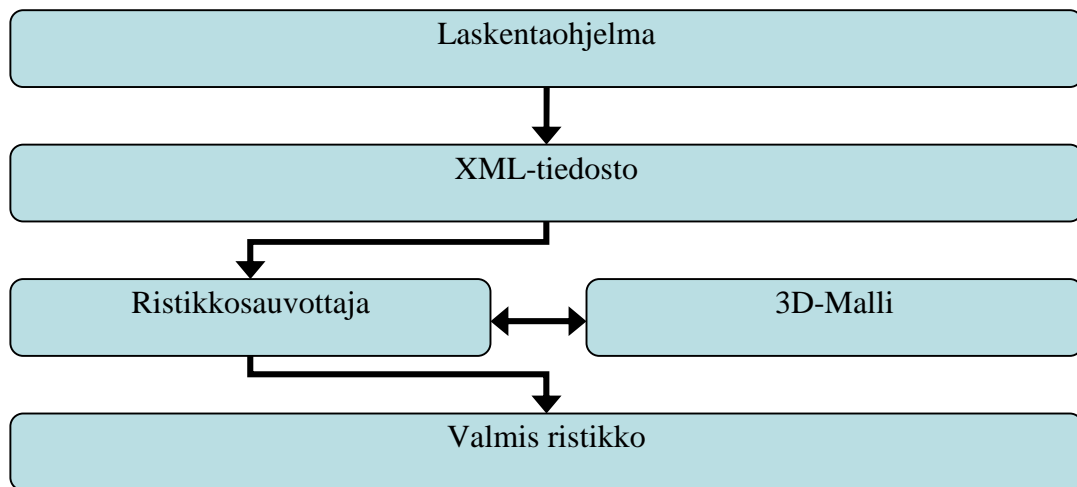
Esimerkki XML-koodista:

```
<?xml version="1.0" encoding="UTF-8"?>
<JuuriElementti>
<Kysymys>Montako insinööriä tarvitaan vaihtamaan hehkulamppu?</Kysymys>
<Vastaus>Laskelmien mukaan 0,8374 kappaletta</Vastaus>
</JuuriElementti>
```

### 3 OHJELMAN RAKENNE JA TOTEUTUS

#### 3.1 Ohjelman toimintaperiaate

Ristikkosauvottaja-ohjelma toimii siten, että se lukee laskentaohjelmasta tulostetun XML-tiedoston ja käsittelee luettua dataa. Tämän jälkeen ohjelma keskustelee 3d-mallin kanssa ja luo valmiin ristikon oikeine profiili- ja paikkatietoineen 3d-malliin. Mallin-  
nusmenetelmä on esitetty kuviossa 1.



KUVIO 1. Ohjelman toimintaperiaate

### 3.2 XML-tiedosto

XML-formaatin mukaiseen siirtotiedostoon tallennetaan kaikki laskentaohjelmasta saatavilla oleva tieto.

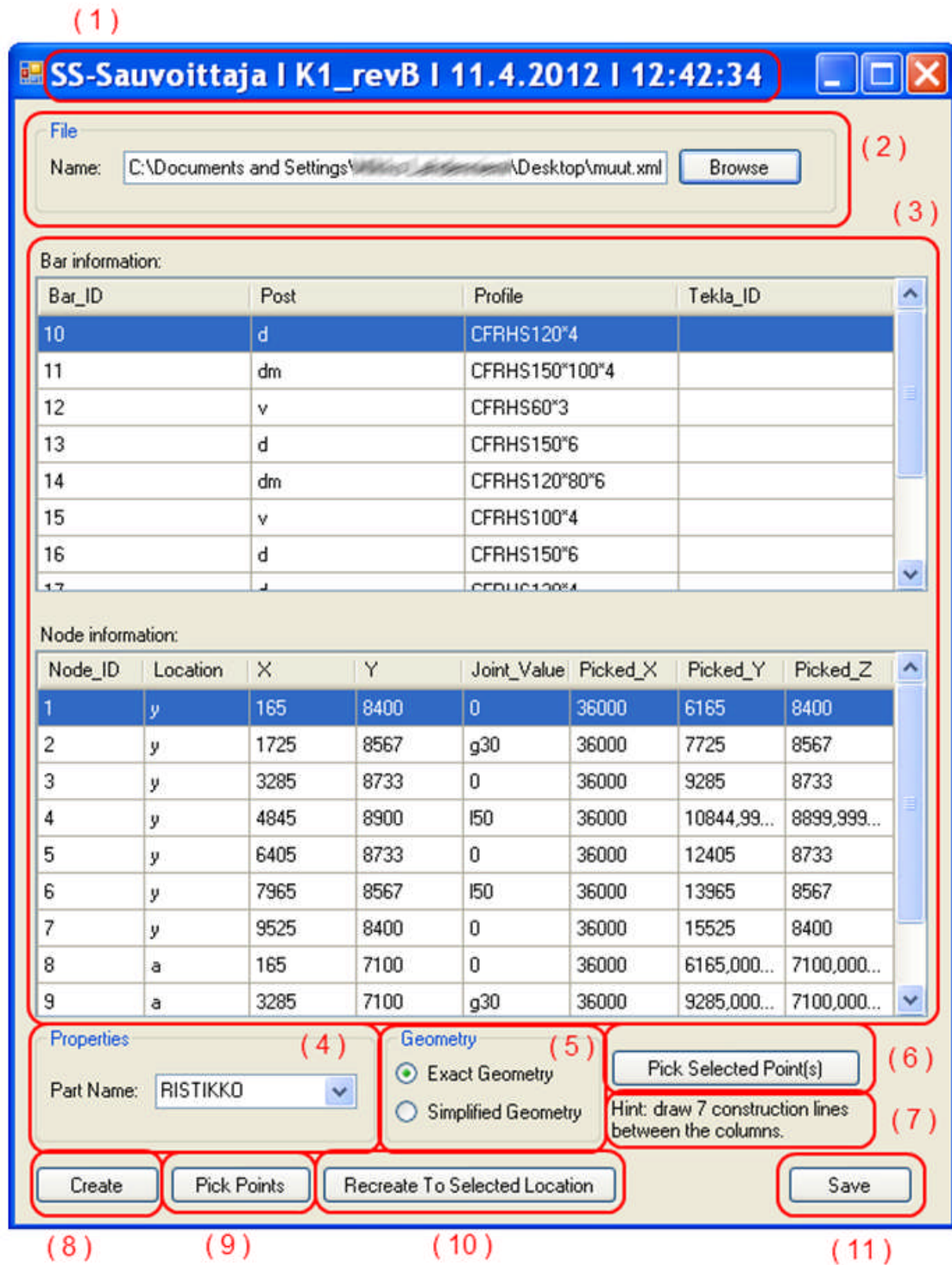
Esimerkki siirtotiedoston toteutuneesta tiedostorakenteesta:

```
<?xml version="1.0" encoding="UTF-8" ?>
<dataroot>
  <Information>
    <Type>R1</Type>
    <Date>30.3.2012</Date>
    <Time>08:21</Time>
  </Information>
  <Node>
    <Node_ID>1</Node_ID>
    <Location>y</Location>
    <X>0</X>
    <Y>1500</Y>
    ...
  </Node>
  <Node>
    ...
  </Node>
  ...
  <Bar>
    <Bar_ID>1</Bar_ID>
    <Post>d</Post>
    <Profile>CFRHS100*5</Profile>
    ...
  </Bar>
  <Bar>
    ...
  </Bar>
  ...
</dataroot>
```

XML-tiedoston ensimmäisellä rivillä on jokaisesta XML-tiedostosta löytyvä prologi, joka sisältää käytettävän XML-version sekä merkistön, jolla tiedosto on kirjoitettu. Seuraavaksi XML-tiedostossa tulee juurielementti, jonka sisälle kaikki muut dokumentin elementit tietoinen kirjoitetaan. Juurielementin sisälle on esimerkissä ensimmäiseksi sijoitettu Information-elementti, jonka sisään on kirjoitettu omiin elementteihinsä ristikon laskentatyyppi sekä XML-tiedoston luontipäivä ja kellonaika. Tämän jälkeen juurielementin sisään on kirjoitettu jokaista ristikon sauvaosaa vastaava sauvaelementti sauvatietoinen sekä jokaista solmupistettä vastaava solmupiste-elementti solmupistetietoinen. Esimerkissä on esitetty vain osa XML-tiedoston todellisesta sisällöstä. XML-tiedosto päättyy aina juurielementin lopetuskäskyyn.

### 3.3 Käyttöliittymä

Ohjelman käyttöliittymä (kuva 2) on ohjelman käytettävyyden kannalta äärimmäisen tärkeässä asemassa. Tästä syystä käyttöliittymästä on pyritty tekemään mahdollisimman selkeä ja informatiivinen.



KUVA 2. Opinnäytetyöohjelman käyttöliittymä

XML-tiedoston lukeminen opinnäytetyöohjelmaan onnistuu kolmella tavalla. Ensimmäinen tapa lukea tiedosto on klikata browse-painiketta (kuva 2, kohta 2), jolloin Windowsin tuttu open file dialog -ikkuna aukeaa, ja tiedosto on etsittävässä tietokoneelta. Toinen tapa lukea XML-tiedosto on kirjoittaa tiedoston tiedostopolku suoraan ohjelman tekstikenttään ja hyväksyä polku painamalla näppäimistön enter-painiketta. Kolmas tapa lukea XML-tiedosto on raahata hiiren kursorilla XML-tiedosto lomakkeen päälle ja tiputtaa tiedosto siihen. Tällöin ohjelma hakee automaattisesti tiedoston tiedostopolun tekstikenttään ja lukee tiedoston ohjelmaan. Kun XML-tiedosto on luettu ohjelmaan, lomakkeen otsikkoriville tulee ohjelman nimen lisäksi näkyviin ristikon laskentatyyppi sekä XML-tiedoston luontipäivämäärä ja -kellonaika (kuva 2, kohta 1).

Kun XML-tiedosto on luettu tietokantaan, näytetään käyttäjälle oleellinen tieto taulukoituna Datagridview-objektin avulla (kuva 2, kohta 3). Sauvatiedoille ja solmupistetiedoille on molemmille oma taulukkonsa, josta tiedot ovat luettavissa. Tarpeen tullen taulukoiden tietoja voidaan myös muokata. Tietoa voidaan myös järjestellä sarakkeiden arvojen mukaan nousevaan tai laskevaan järjestykseen.

Ohjelman käyttöliittymän properties-osiossa (kuva 2, kohta 4) käyttäjä voi määrittellä minkä nimen mallinnusohjelma antaa ristikko-osille. Syöttöikkuna sisältää valmiiksi yleisimmin käytetyt ristikko-osien nimet, mutta käyttäjä voi halutessaan määrittää osille myös jonkin muun nimen kirjoittamalla sen suoraan tekstikenttään. Ohjelman geometria-osiossa (kuva 2, kohta 5) määrittellään halutaanko ristikko mallintaa yksinkertaistulla vai tarkalla geometrialla.

Ohjelman käyttöliittymän Pick Selected Point(s)-painikkeella (kuva 2, kohta 6) voidaan solmupistetaulukosta valitulle solmupisteelle antaa uusi koordinaattipiste 3d-mallista. Kun painiketta painetaan, ohjelma jää odottamaan, että käyttäjä klikkaa 3d-mallista koordinaattipistettä. Käyttäjän antama koordinaattipiste lisätään solmupisteen tietoihin. Painikkeen avulla voidaan myös antaa useamman solmupisteen koordinaatit samanaikaisesti. Kun painiketta painettaessa taulukosta on valittuna useampi solmupisterivi, ohjelma jää odottamaan, että 3d-mallista klikkaillaan yhtä monta koordinaattipistettä, kuin solmupisterivejä oli aktiivisena. Klikkailujärjestys on tällöin järjestysnumeroltaan pienimmästä solmupisteestä kohti suurinta.

Luotaessa ristikkoa tarkalla geometrialla, pitää malliin piirtää apuviivoja solmupisteiden klikkailun helpottamiseksi. Ohjelma ilmoittaa tekstikentässä tarvittavien apuviivojen määrän (kuva 2, kohta 7). Apuviivojen käyttöä on esitelty tarkemmin luvussa 4.2.

Ohjelman käyttöliittymän Create-painikkeella (kuva 2, kohta 8) ristikko luodaan 3d-malliin. Kun ristikkoa luodaan yksinkertaistetulla geometrialla, täytyy käyttäjän antaa ristikolle paikka ja suunta 3d-mallista. Tämä tapahtuu klikkaamalla kahta pistettä 3d-mallissa create-painikkeen klikkaamisen jälkeen.

Ohjelman käyttöliittymän Pick Points-painikkeella (kuva 2, kohta 9) ohjelmalle syötetään jokaista solmupistettä vastaavat koordinaattipisteet 3d-mallista. Painikkeen painamisen jälkeen pisteet klikkaillaan mallista numerojärjestyksessä pienimmästä suurimpaan. Solmupisteet on numeroitu siten, että ensimmäisenä ovat yläpaarteen solmupisteet vasemmalta oikealle ja sen jälkeen alapaarteen solmupisteet vasemmalta oikealle.

Ohjelman käyttöliittymän Recreate-painikkeella (kuva 2, kohta 10) voidaan kertaalleen luotu ristikko luoda uudestaan 3d-malliin käyttäen samaa solmupistesijoittelua kun aikaisemmin. Painikkeen painamisen jälkeen käyttäjä klikkaa 3d-mallista pisteen, johon haluaa ristikon ensimmäisen solmupisteen sijoittuvan.

Ohjelman käyttöliittymän Save-painikkeella (kuva 2, kohta 11) voidaan tallentaa ohjelman tietokannassa olevat tiedot uudelleen XML-tiedostoksi. Tämä on hyödyllistä silloin kun uudet, tarkan geometrian, solmupisteet on klikkailtu mallista ja halutaan välttyä tekemästä samaa työtä toiseen kertaan. Lisäksi save-painikkeella saadaan tallennettua mahdolliset muutokset, jotka käyttäjä on tehnyt sauva- ja solmupistetaulukoihin.

### **3.4 Ristikon luominen**

Ristikon luominen 3d-malliin tapahtuu ohjelmallisesti Tekla.Structures.Model-kirjaston sisältämän Beam-luokan avulla. Ohjelmassa määritellään uusi beam-luokan mukainen palkki-muuttuja, jolle määritellään tarvittavat ominaisuudet. Tällaisia ominaisuuksia ovat esimerkiksi alku- ja loppupiste sekä profiili- ja nimitiedot. Tämän jälkeen Beam-luokan insert-toiminnolla lisätään kyseisillä ominaisuuksilla varustettu palkki 3d-malliin.

Esimerkkikoodi ristikko-osan luomisesta:

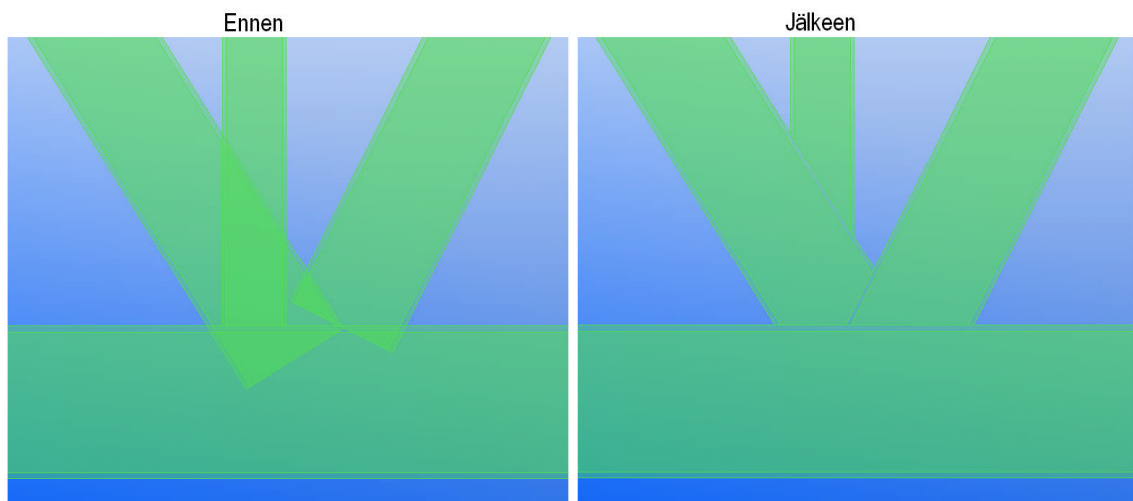
```
Dim MyBeam As New Tekla.Structures.Model.Beam
MyBeam.Name = "ristikko"
MyBeam.Profile.Profilestring = "CFRHS90*40"
MyBeam.StartPoint.X = 1500
MyBeam.StartPoint.Y = 1500
MyBeam.StartPoint.Z = 1000
MyBeam.EndPoint.X = 4000
MyBeam.EndPoint.Y = 1500
MyBeam.EndPoint.Z = 2000
MyBeam.Insert()
```

*'Esitellään beam-luokan mukainen muuttuja  
'Määritetään muuttujan arvot*

*'Lisätään osa 3d-malliin*

Luotaessa ristikkoa yksinkertaisen geometrian mukaan, ristikko-osien alku- ja loppupisteiksi määritellään suoraan laskentaohjelmasta saatujen solmupisteiden koordinaatit. Tarkan geometrian laskennassa palkin sijainti määritellään erilaisten laskurutiinien kautta, joita ohjelmassa on kymmeniä. Palkin sijaintiin ohjelmassa voidaan vaikuttaa alku- ja loppupisteen lisäksi myös erilaisilla asema- ja etäisyysominaisuuksilla. XML-tiedostossa on määritelty onko uumasauvojen liitos paarteella vapaaväliliitos vai limityслиitos. Samassa yhteydessä kerrotaan kuinka suuri mahdollinen vapaaväli on millimetreissä tai kuinka suuri mahdollinen limitys on prosentteissa.

Kun ohjelma on luonut kaikki ristikko-osat tarkan geometrian mallinnuksessa, ohjelma viimeistelee osat leikkaamalla ja sovittamalla ne osien leikkauspintojen suuntaisesti (kuva 3). Ohjelma tunnistaa kumpi päällekkäin tulevista osista leikataan toisella osalla XML-tiedoston sauvatietoihin tallennetun On Top Of Bar -ominaisuuden avulla.

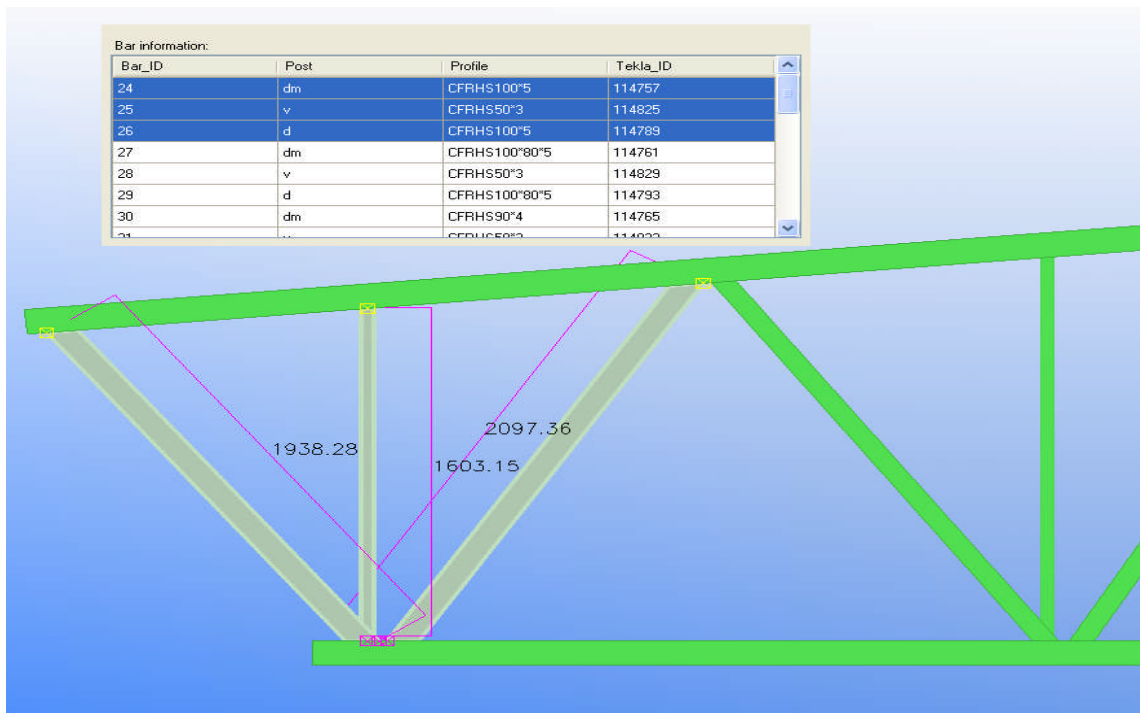


KUVA 3. Ristikon sauvaosat ennen liitoksen viimeistelyä sekä viimeistelyn jälkeen



Lopuksi ohjelma käyttää mallinnusohjelman hitsaustyökalua osien yhteenliittämiseksi. Ristikosta tulee tällöin yhtenäinen kokoonpano, jonka pääosaksi määritellään aina ristikon alapaarre.

Kun ristikko-osa on siirretty 3d-malliin, voidaan mallilta kysyä minkä ID-numeron malli osalle antoi. Tämän ID-numeron avulla voitiin taulukoihin lisätä ominaisuus, jolla taulukoista valittu osa tai solmupiste tulee aktiiviseksi 3d-mallissa (kuva 4). Tämä helpottaa osien ja solmupisteiden ominaisuuksien tunnistamista ja muokkaamista.



KUVA 4. Luodut osat voidaan paikantaa mallista valitsemalla halutut osat taulukosta

### 3.5 Tiedon säilyttäminen ja muokkaaminen ohjelmassa

Tietojen säilyttämiseen ohjelmassa päädyttiin käyttämään paikallista tietokantaa (local database). Paikallinen tietokanta on nimensä mukaan ohjelman sisäänrakennettu tiedontallentamisen apuväline, jonka kapasiteetti on periaatteessa rajaton. Tietokannan sisältämää tietoa pystytään ohjelmassa muokkaamaan ohjelman pääsivulla olevien DataGridView-tilaukko-objektien avulla. Kaikki taulukoissa tehdyt muokkaukset tallentuvat suoraan myös tietokantaan. Samoin ohjelman tekemät muutokset tietokannassa näkyvät käyttäjälle näytettävissä taulukoissa.

Tietokannan käyttö ohjelmassa tuntuu jälkeinpäin ajateltuna hiukan ylimitoitetulta ohjelman luonteeseen nähden. Tietokanta tarjosi kuitenkin joitakin yksittäisiä ominaisuuksia, jotka helpottivat koodaamista. Eräs tällainen ominaisuus oli esimerkiksi tietokannan tulostaminen XML-tiedostoksi, johon riitti käytännössä vain yksi komentorivi.

### **3.6 Koodaaminen**

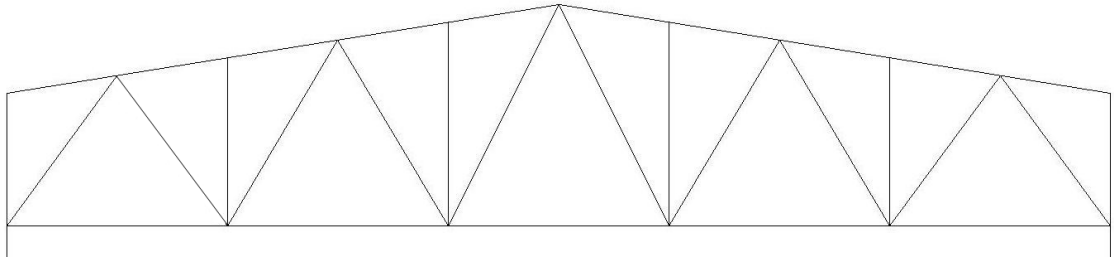
Ohjelmoinnissa päätyökaluna toimi Microsoftin ilmainen Visual Basic 2010 express -kehitysympäristö. Kehitysympäristön käyttäjäystävällisyys ja graafinen käyttöliittymä helpottivat ohjelman perustoimintojen koodaamista. Ohjelmaa koodattiin yhteensä, kommentteineen kaikkineen, reilut 2500 riviä. Tämän lisäksi ohjelmassa on lomakkeen ulkoasuun ja muihin toimintoihin liittyvää kehitysympäristön automaattisesti tuottamaa koodia. Ohjelmoinnin yhteydessä ohjelmakoodia kommentoitiin paljon, jotta koodin sisäistäminen jälkeinpäin olisi helppoa ja vaivatonta. Ohjelmoinnin ja ohjelmakoodin lukemisen helpottamiseksi ohjelmakoodia pyrittiin myös jakamaan pieniksi aliohjelmiksi. Aliohjelma on pätkä ohjelmakoodia, joka suorittaa jonkin tietyn tehtävän ohjelmakokonaisuudesta.

### **3.7 Testaaminen**

Ohjelmakoodia testattiin esimerkkiristikolla koodaamisen edistyessä. Virheiden etsintää helpottivat kehitysympäristön virheenetsintätyökalut, joiden avulla ohjelmaa ja sen muuttujia pystyttiin tutkimaan yksityiskohtaisesti ajon aikana. Ohjelman rungon valmistuttua ohjelmaa testattiin, myös neljällä muulla ristikolla tulosten oikeellisuuden varmistamiseksi. Ohjelmaa pyrittiin testaamaan mahdollisimman monipuolisesti myös erilaisilla syöttötiedoilla ja tapahtumasarjoilla. Opinnäytetyöohjelman laaja-alaisempi testaaminen erilaisilla ristikoilla jää ajan puutteen vuoksi myöhempään ajankohtaan.

## 4 ESIMERKKI OHJELMAN KÄYTÖSTÄ

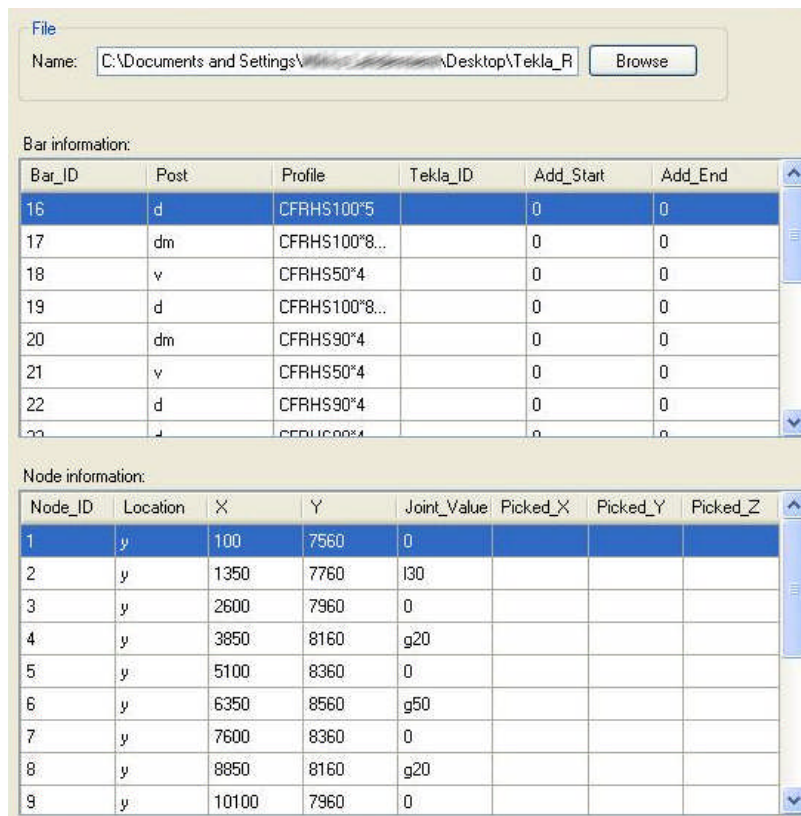
Tässä luvussa esitellään ohjelman käyttöä ja toimintoja. Esimerkit on toteutettu jäykkäliitoksisella harjaristikolla (kuva 5).



KUVA 5. Esimerkeissä käytetyn ristikon rautalankamalli

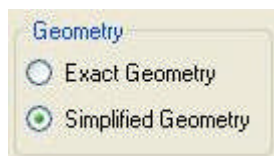
### 4.1 Ristikko yksinkertaisen geometrian mukaan

Opinnäytetyöohjelma tarvitsee toimiakseen Tekla Structures 3d-mallin. Tästä syystä opinnäytetyöohjelma ei suostu käynnistymään, ellei 3d-malli ole auki mallinnusohjelmassa. Opinnäytetyöohjelman käynnistyttyä ensimmäinen toimenpide on hakea haluttu XML-tiedosto ohjelman käyttöön. Tämä tapahtuu esimerkiksi raahaamalla haluttu tiedosto opinnäytetyöohjelman päälle. Jos ohjelmalle syötetty tiedosto on oikein rakennettu XML-tiedosto, ohjelma lukee tiedoston sisältämän datan tietokantaan ja näyttää tärkeimmät tiedot sauva- ja solmupistetaulukoissa (kuva 6).



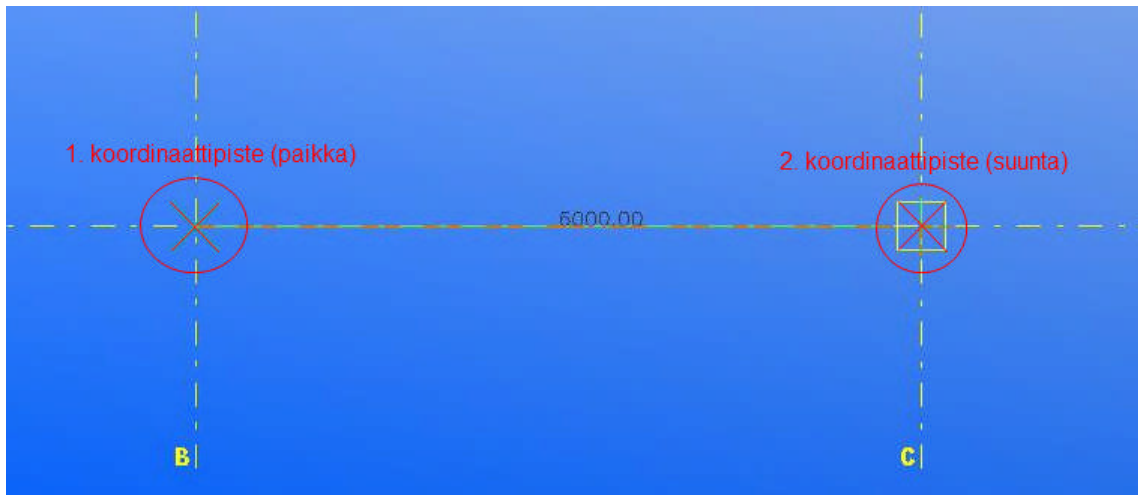
KUVA 6. Opinnäytetyöohjelma XML-tiedoston lukemisen jälkeen

Luotaessa yksinkertaistettua ristikkoja on geometriavalintapainike asetettava asentoon Simplified Geometry (kuva 7).



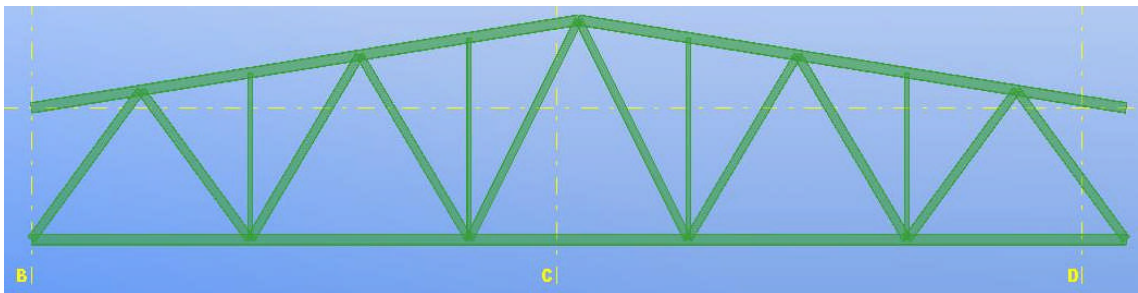
KUVA 7. Yksinkertaistetun geometrian valinta

Tämän jälkeen ristikko on valmiina luotavaksi 3d-malliin. Create-painikkeen painamisen jälkeen ohjelmalle on annettava 3d-mallista ristikon paikka ja suunta kahdella koordinaattipisteellä (kuva 8). Koordinaattipisteistä annettaessa ei tarvitse huolehtia 3d-mallin koordinaatisto, sillä ohjelma muuttaa koordinaatiston automaattisesti globaaliksi pisteiden antamisen ajaksi. Pisteiden syöttämisen jälkeen ohjelma palauttaa aikaisemmin käytössä olleen koordinaatiston takaisin käyttöön.



KUVA 8. Koordinaattipisteiden antaminen 3d-mallista

Koordinaattipisteiden antamisen jälkeen ohjelma luo XML-tiedoston mukaiset osat 3d-malliin (kuva 9). Osat luodaan laskentaohjelmasta saatujen solmupistekoordinaattien mukaan käyttäjän määrittelemään paikkaan ja suuntaan 3d-mallissa. Yksinkertaisella geometrialla luodun ristikon tarkkuus tulee paremmin esiin kuvassa 10.



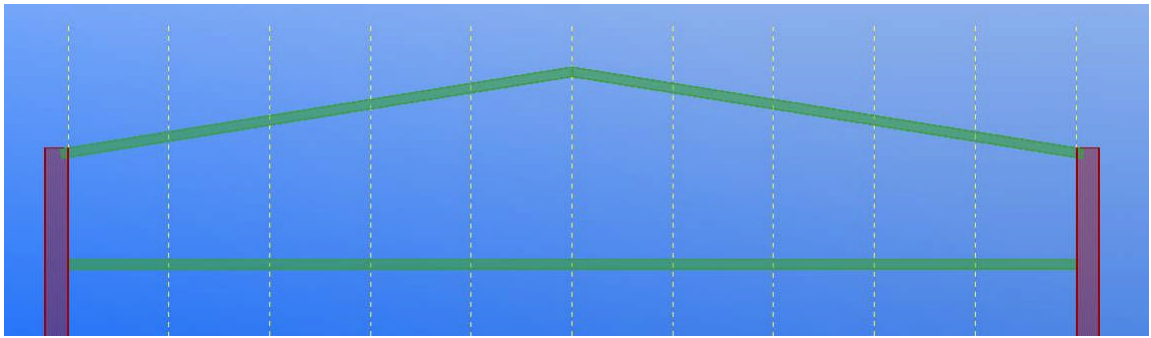
KUVA 9. Yksinkertaisella geometrialla luotu ristikko



KUVA 10. Yksinkertaisella geometrialla luodun ristikon uumasauvojen liittyminen alapaarteeseen

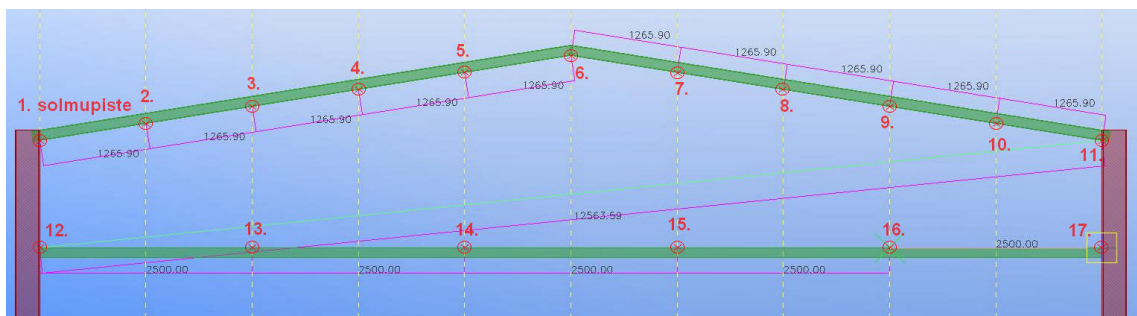
#### 4.2 Ristikko tarkan geometrian mukaan

Luotaessa ristikköä tarkan geometrian mukaan, täytyy ristikon solmupisteet määrittää uudelleen 3d-mallista. Kun opinnäytetyöohjelman geometriavalintapainike on asetettu Exact Geometry -asentoon, aktivoituu ohjelman käyttöliittymän Pick Points -painike. Pick Points -painikkeen painamisen jälkeen ohjelmalle on klikkailtava 3d-mallista ristikon toteutuvat solmupisteet. Jotta solmupisteet olisi helppo klikkailla 3d-mallista, täytyy malliin tehdä muutamia esivalmisteluja (kuva 11). Esivalmisteluna malliin luodaan väliaikaiset ylä- ja alapaarteet haluttuun korkoon halutuilla poikkileikkausprofiileilla varustettuina. Tämän jälkeen ristikko jaetaan apuviivojen avulla osiin toteutuvien solmupisteiden kohdilta, kuten kuvassa 11 on tehty. Ohjelma kertoo tarvittavien apuviivojen määrän Pick Selected Point(s) -painikkeen alapuolella, kun geometria-asetukseksi on valittu Exact Geometry.



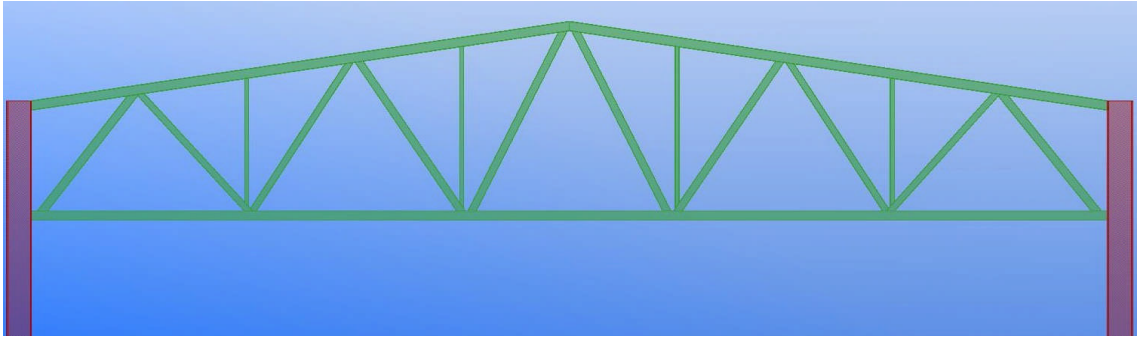
KUVA 11. Esivalmistelut

Esivalmistelujen jälkeen painamalla Pick Points -painiketta päästään antamaan ohjelmalle toteutuvia solmupisteitä 3d-mallista. Solmupisteet on klikkailtava numerojärjestyksestä siten, että ensimmäisenä ovat yläpaarteen solmupisteet vasemmalta oikealle ja sen jälkeen alapaarteen solmupisteet vasemmalta oikealle. Kuvassa 12 on esitetty esimerkkiristikon solmupisteiden klikkailujärjestys.

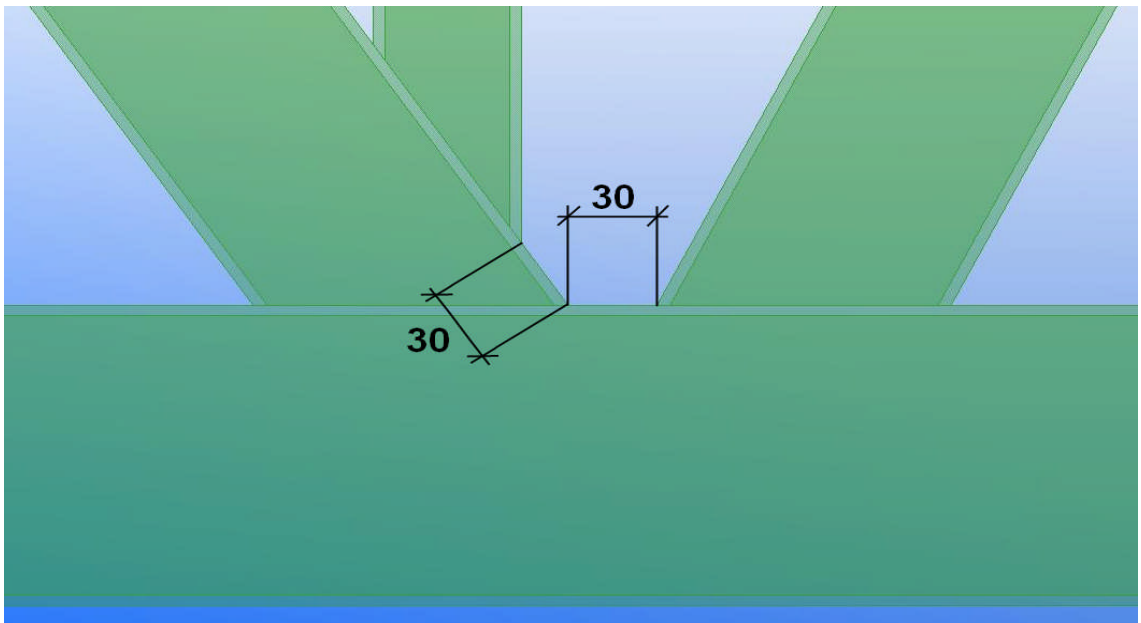


KUVA 12. Solmupisteiden klikkailujärjestys

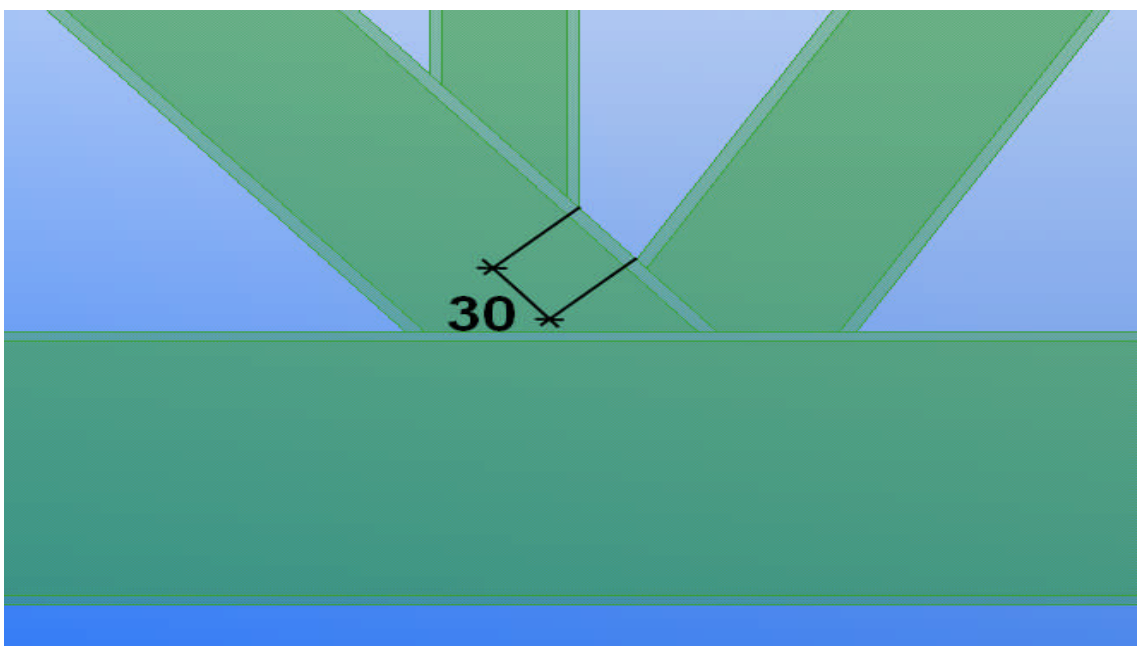
Kun solmupisteiden klikkailu on suoritettu, opinnäytetyöohjelma sijoittaa annetut koordinaattipisteet solmupistetaulukkuun. Kun jokaiselle solmupisteelle on annettu uusi koordinaattipiste, create-painike aktivoituu ja ristikko voidaan luoda 3d-malliin. Ennen ristikon luomista kannattaa 3d-mallista poistaa väliaikaiset ylä- ja alapaarteet. Luotaessa ristikkoa tarkan geometrian mukaan opinnäytetyöohjelma laskee jokaisen solmuun liittyvän osan sijainnin solmupisteeseen nähden siten, että XML-tiedostossa annetut solmupisteen liitostiedot toteutuvat. Toteutunut ristikko on esitetty kuvassa 13. Kuvassa 14 ja kuvassa 15 näkyy tarkemmin miten ohjelma mallintaa yksittäisten solmupisteiden liitokset.



KUVA 13. Tarkalla geometrialla luotu ristikko



KUVA 14. Paarteen ja uumasauvojen liitos. Liitokseen oli määritelty 30 mm vapaaväli.



KUVA 15. Alapaarteen ja uumasauvojen liitos, liitokseen oli määritelty 50 % limititys



## 5 JATKOKEHITYS

Jatkokehitysideoita syntyi ohjelmoinnin aikana runsaasti ja niitä pyrittiin mahdollisuuksien mukaan toteuttamaan jo opinnäytetyöprosessin aikana. Seuraavaksi käsitellään joitakin ideoita, jotka jäivät vielä toteuttamatta.

### 5.1 Ohjelman automaation parantaminen

Ohjelman automatisointia voidaan parantaa erityisesti mallinnettaessa ristikkoo tarkalla geometrialla. Pitkälle kehitetty ohjelma voisi nappia painamalla mallintaa väliaikaiset paarteet, piirtää tarvittavat apuviivat sekä hakea paarteiden ja apuviivojen leikkauskohdat suoraan mallinnusohjelmasta.

### 5.2 Tekla Structures plug-in

Opinnäytetyöohjelma voidaan sisällyttää Tekla Structures -mallinnusohjelmaan muuntamalla se plug-in-työkaluksi. Plug-in-työkalun avulla mallinnettuun ristikkoon pystyttäisiin helposti tekemään muutoksia, sen sijaan että ristikko täytyy aina tuhota ja luoda uudelleen. Lisäksi Plug-in-työkalu olisi aina helposti löydettävissä mallinnusohjelman muiden työkalujen joukosta.

### 5.3 Ohjelman testaaminen ja laajentaminen

Opinnäytetyöohjelmaa ehdittiin opinnäytetyön puitteissa testata viidellä eri ristikolla. Ohjelman laaja-alaisempi testaaminen on hyvä suorittaa, ennen ohjelman varsinaista käyttöönottoa.

Ohjelman käyttöala on opinnäytetyön puitteissa rajattu koskemaan vain ristikoiden perustapauksia. Tulevaisuudessa ohjelman käyttöalaa voidaan laajentaa koskemaan esimerkiksi ristikoita joissa on poikkileikkaukseltaan pyöreitä teräsputkia, ja ristikoita joissa on useampia ylä- ja alapaarteita.

## 6 YHTEENVETO

Opinnäytetyön tarkoituksena oli tuottaa tilaajalle ohjelma, jonka avulla voidaan nopeuttaa teräsristikon mallintamista ja vähentää siinä syntyviä virheitä. Valmistunut ohjelma täyttää sille asetetut tavoitteet ja on laajemman testauksen jälkeen valmis käyttöönotettavaksi. Ohjelmalla pystytään luomaan, laskentamallista tulostetun siirtotiedoston avulla, teräsristikko 3d-malliin. Ohjelman avulla mallinnetussa ristikossa on varmasti samat teräsprofiilit joilla ristikko on laskentaohjelmassa mitoitettu. Lisäksi ohjelma mallintaa ristikon liitokset automaattisesti ja nopeuttaa näin suunnittelijoiden työtä.

Ohjelman tekeminen oli työläs prosessi, joka vaati paljon aikaa ja kärsivällisyyttä. Samaan aikaan ohjelman tekeminen oli kuitenkin myös erittäin mielenkiintoista ja palkitsevaa ja onkin hyvin todennäköistä, että opinnäytetyön laatijan ohjelmointikokemus tulee vielä karttumaan tulevien vuosien saatossa.

## LÄHTEET

Tekla.com: Merkkipaalut. Luettu 15.4.2012

<http://www.tekla.com/fi/about-us/history/milestones/Pages/Default.aspx>

Tekla.com: Peruskäsitteitä. Luettu 15.4.2012

<http://www.tekla.com/fi/solutions/building-construction/Pages/basic-concepts.aspx#tekla-open-api>

Tekla.com: Tekla BIM. Luettu 15.4.2012

<http://www.tekla.com/fi/products/tekla-structures/Pages/Default.aspx>

Wikipedia: Visual Basic .Net. Luettu 15.4.2012.

[http://en.wikipedia.org/wiki/Visual\\_Basic\\_.NET](http://en.wikipedia.org/wiki/Visual_Basic_.NET)

Wikipedia: XML. Luettu 15.4.2012

<http://en.wikipedia.org/wiki/XML>