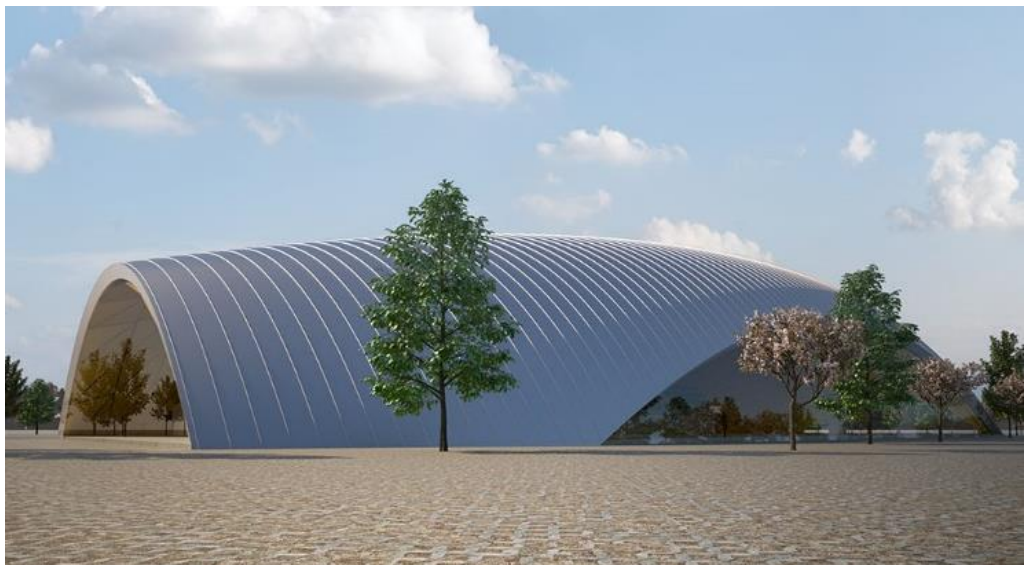


ARKKITEHDIN JA RAKENNESUUNNITTELIJAN VÄLISEN ALGORITMIAVUSTEISEN SUUNNITTELUPROSESSIN KEHITTÄMINEN LOPPURAPORTTI

Algoritmiavusteista suunnittelua voidaan hyödyntää monenlaisten rakenteiden kohdalla, mutta sen voi katsoa soveltuvan erityisesti vaativien muotojen sekä toistuvien rakenteiden suunnitteluun. Tämä johtuu pääasiassa siitä, että algoritmipohjaisten menetelmien avulla kyseisten rakenteiden suunnittelu on huomattavasti helpompaa perinteisiin menetelmiin verrattuna. Esimerkiksi hyvin haastavien muotojen kohdalla tieto- ja laskentamallien rakentaminen voi olla lähes mahdotonta perinteisin menetelmin. Algoritmiavusteisen suunnittelun parametrinen luonne tuo myös helpotusta toistuvien rakenteiden suunnitteluun. Suunniteltavien rakenteiden muodosta ja ominaisuuksista riippumatta algoritmiset menetelmät voivat tuoda merkittäviä etuja koko suunnitteluprosessin etenemiseen. Tässä kokeiluhankkeessa keskitytään ennen kaikkea tarkastelemaan suunnitteluprosessin eri vaiheita ja testaamaan algoritmisen suunnittelun soveltuvuutta niihin.

Suunnitteluprosessia tarkastellaan fiktiivisen suunnittelukohteen avulla. Tarkoituksena on testata ja analysoida algoritmisten menetelmien soveltuvuutta prosessin eri vaiheisiin aina arkkitehdin lähtötiedoista rakenneseosien ja liitosten toteutussuunnitteluun. Suunnitteluprosessiin osallistuu sekä arkkitehti että rakennesuunnittelija. Arkkitehti hyödyntää suunnittelussaan algoritmisia menetelmiä ja tarkoituksena on testata niiden tehokasta hyödyntämistä rakennesuunnittelun lähtötietona. Hankkeen tavoitteena on suunnitteluprosessin lisäksi tarkastella älykästä vuorovaikutusta arkkitehdin ja rakennesuunnittelijan välillä. Suunnitteluosapuolien välisessä tiedonsiirrossa hyödynnetään pilvipohjaista tiedonjakotapaa, joka voidaan myös linkittää älykkäästi algoritmiseen suunnitteluprosessiin. Kuvassa 1 esitetään arkkitehdin visualisointi suunniteltavasta kattorakenteesta.

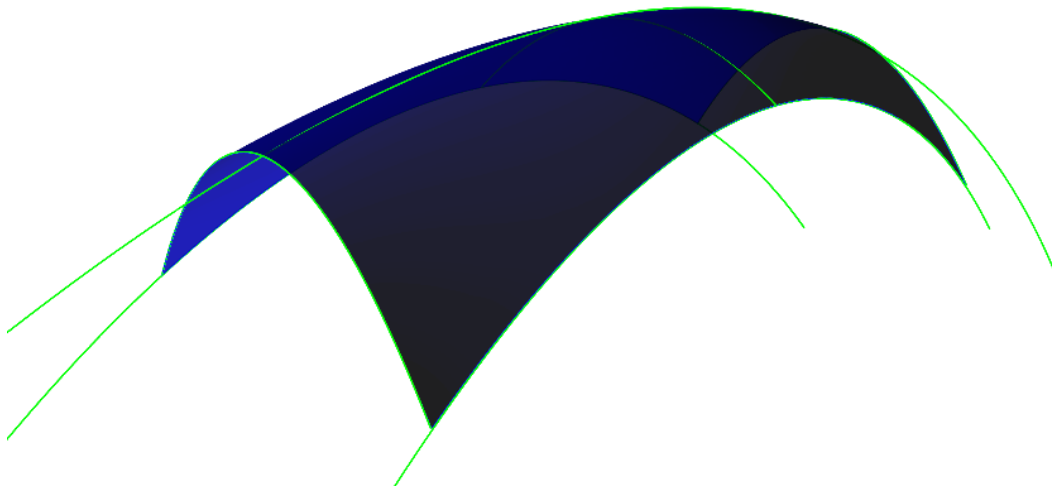


Kuva 1. Arkkitehdin visualisointi suunniteltavasta kattorakenteesta.

Suunnittelukohteen esittely

Tässä tutkimuksessa käydään läpi algoritmiavusteista suunnitteluprosessia kuvitteellisen suunnittelukohteen avulla. Suunnittelukohteeksi valikoitui uimahalli, jonka merkittävä ominaisuus on monimuotoinen teräsrakenteinen kattorakenne. Suunnittelun painopiste kohdistetaan erityisesti haastavan kattorakenteen suunnitteluun, jotta päästään näkemään algoritmiavusteisen suunnittelun todelliset hyödyt ja haasteet. Kattorakenne sisältää sekä vaativia muotoja että toistuvia rakenneosia, minkä johdosta sen koetaan olevan oivallinen algoritmiavusteisen suunnitteluprosessin testauskohde. Suunnittelussa ei pyritä täydelliseen, toteutettavaan rakenteeseen vaan tarkoituksena on testata kattorakenteen avulla tyyppillisiä suunnittelutehtäviä. Suunniteltava kattorakenne on haastavan muotonsa johdosta oivallinen esimerkki algoritmisten menetelmien mahdollisuuksista. Perinteisiä suunnittelumenetelmiä käytettäessä pelkästään geometrian luominen ja mahdollinen muokkaaminen olisivat hyvin työläitä tehtäviä.

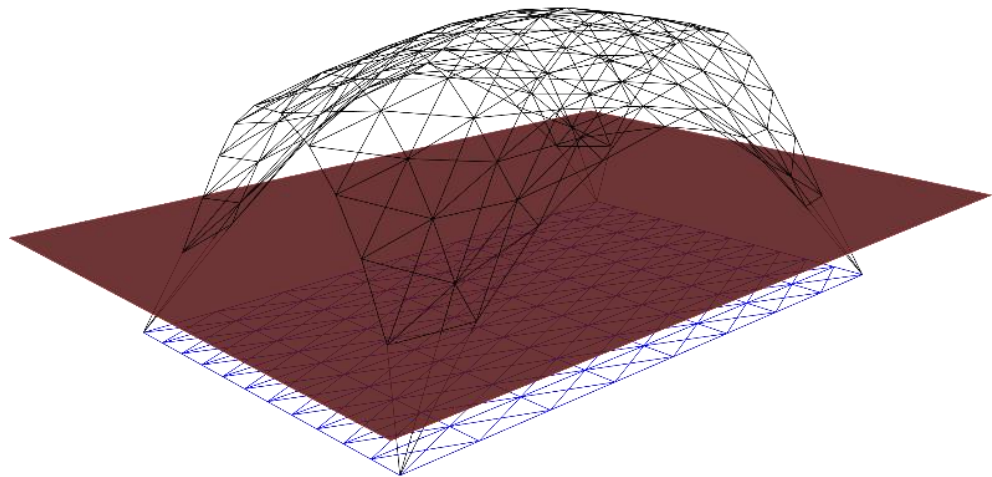
Arkkitehti muodostaa kattorakenteesta useita erilaisia vaihtoehtoja, joiden avulla suunnitteluprosessia voidaan tutkia ja kehittää. Katon perusmuoto koostuu kaarevasta pinnasta, joka generoidaan paraabelikäyrien avulla. Esimerkki katon muodon määrittävistä paraabelikäyristä ja niiden kautta kulkevasta ohjauspinnasta esitetään kuvassa 2. Ohjauspinta jaetaan pitkittäis- ja poikittaissuuntaisiin osiin ja muodostuneiden ruutukuvioiden lävitse luodaan vinosauvat, jolloin kattorakenne koostuu säännöllisistä kolmiopinnoista. Kattopinnan viivajako toteutetaan parametrisesti, minkä ansiosta viivageometrian tiheydellä muokkaaminen on hyvin helppoa ja nopeaa.



Kuva 2. Kattorakenteen muodon määrittävä ohjauspinta.

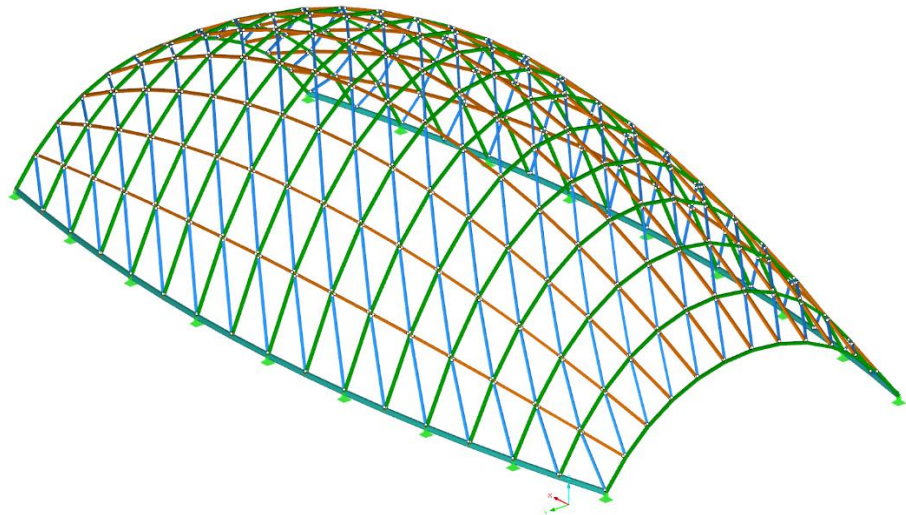
Katon perusmuoto generoidaan niin sanotun dynaamisen relaksaation avulla. Siinä viivageometrian muodostavat sauvat ovat jousia ja sauvojen välisiin solmupisteisiin kohdistuu pysty akselin suuntainen voima. Tällöin sauvojen muodostamasta rakennejärjestelmästä

muodostuu rakenteellisen toimivuuden kannalta optimaalinen. Monet seikat, kuten rakennuspaikka, rakennuksen toiminnot ja rakennettavuus, johtavat kuitenkin tyypillisesti toisenlaiseen perusmuotoon kuin jokin geometrialtaan rakennemielessä optimaalinen muoto. Tämän takia muodon generoimiseen käytettävien relaksaatiofunktioiden määrittäminen on tärkeä osa suunnittelua. Kun kattorakenteen geometria on määritetty, voidaan se leikata perustustason korkoasemassa. Kuvassa 3 esitetään esimerkki dynaamisen relaksaation avulla muodostetusta kattomuodosta.



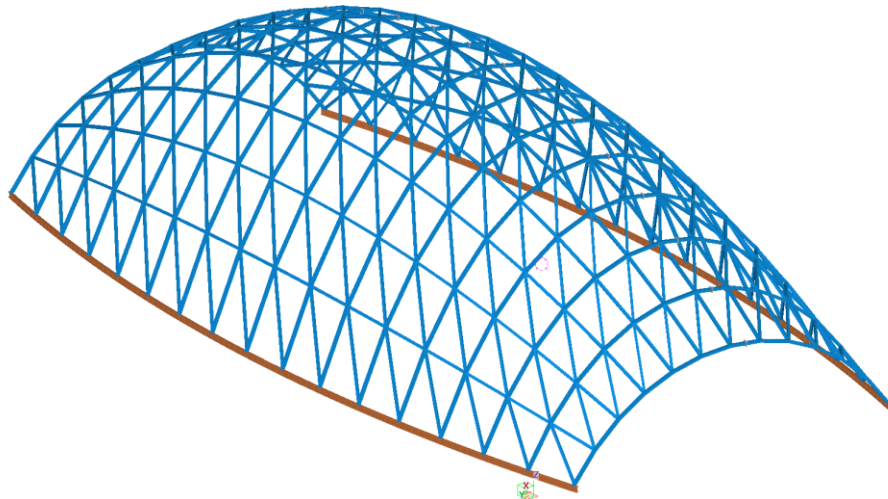
Kuva 3. Katon perusmuodon generoiminen dynaamisen relaksaation avulla.

Kun arkkitehti on luonut algoritmiavusteisesti kaarevan kattopinnan ja generoinut siihen viivageometrian, rakennesuunnittelija voi hyödyntää tätä geometriatietoa oman suunnittelunsa lähtötietona. Viivageometriaa voidaan helposti hyödyntää esimerkiksi laskentamallin luomiseen. Arkkitehdin geometriatieto järjestetään ja jäsennellään rakennesuunnittelun kannalta toimivaksi kokonaisuudeksi, jotta laskentamallista voidaan muodostaa rakenteellisesti toimiva kokonaisuus. Kuvassa 4 esitetään esimerkki erään kattorakenteen FEM-laskentamallista. Rakennejärjestelmä koostuu IPE-profiililla toteutetuista tukipalkeista sekä neliönmuotoisilla putkiprofiileilla muodostetusta ristikkorakenteesta. Kattorakenne tukeutuu molemmilla puolilla yhdeksään tukipisteeseen.



Kuva 4. Esimerkki kattorakenteen FEM-laskentamallista.

Tukipalkkien päälle tukeutuvat poikittaissuuntaiset kaaripalkit muodostuvat kahdesta kokoonpanosta. Teräskokoonpanot kiinnittyvät toisiinsa tukipalkkien puolivälissä jatkosliitoksen avulla. Kattorakennetta kannattelevat tukipalkit toteutetaan kaarevina rakenteina. Palkkien päitä jatketaan, jotta myös päätyjen kaaripalkit pystyvät tukeutumaan riittävästi niiden päälle. Rakennneosat sijoittuvat pääasiassa laskentamallin sauvojen kanssa samoihin sijainteihin, mutta joitakin pieniä muutoksia on mahdollista tehdä. Esimerkiksi vinosauvojen sijaintia voidaan muokata siten, että sauvat eivät törmää keskenään. Algoritmien generoima rakennemalli esitetään kuvassa 5. Rakennemallin muodostamiseen hyödynnetään tietomallinnuksessa yleisesti käytettävää Tekla Structuresia.



Kuva 5. Kattorakenteen rakennemalli.

Tässä kokeiluhankkeessa toteutetaan useita erilaisia variaatioita edellä esitetystä teräsrakenteisesta kattorakenteesta. Katon muoto ja muut ominaisuudet eivät kuitenkaan ole tämän työn kannalta oleellisia, koska tarkoituksena on ennen kaikkea kokeilla ja kehittää algoritmiavusteisen suunnitteluprosessin toimintaa.

Suunnitteluprosessin ominaisuudet

Algoritmiavusteinen suunnitteluprosessi koostuu tyypillisesti monista algoritmeja sisältävistä tiedostoista eli niin sanotuista algoritmimalleista sekä niiden avulla ohjattavista perinteisistä suunnittelumalleista. Tieto linkittyy algoritmien välityksellä toisiinsa siten, että koko suunnitteluprosessista voidaan muodostaa automaattisesti päivittyvä systeemi. Kaiken perustana toimivat visuaalisen ohjelmointialustan avulla muodostettavat algoritmit sekä niiden rajapinnat muihin käytettäviin ohjelmiin. Algoritmisen suunnittelun kohdalla onkin tärkeää kiinnittää huomiota siihen, miten suunnittelutietoa jaotellaan ja ryhmitellään. Tämä pätee sekä yhden tiedoston sisällä tapahtuvaan ryhmittelyyn että käytettävien mallien väliseen jaotteluun.

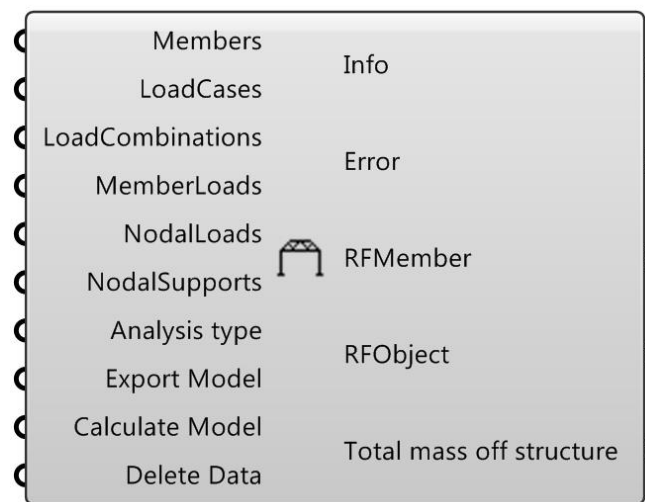
Algoritmeja luotaessa on hyvä pitää mielessä kaikki mahdolliset tapaukset, joihin niiden tulee pystyä reagoimaan. Tämä on usein mahdotonta ja siksi algoritmimalleja luotaessa on tärkeää kiinnittää huomiota mallien selkeyteen. Kun algoritmit nimetään, jaotellaan ja ryhmitellään selkeästi, voidaan niitä käyttää helpommin hyödyksi myös tulevaisuudessa. Myös koko prosessin selkeyteen tulee kiinnittää huomiota. Tämä tapahtuu pääasiassa jakamalla algoritmeja erillisiin algoritmimalleihin, joita voidaan käsitellä itsenäisinä kokonaisuuksina. Algoritmien avulla ohjataan tyypillisesti perinteisiä suunnitteluohjelmistoja. Tällöin tulee kiinnittää huomiota myös algoritmien ja suunnitteluohjelmistojen väliseen tiedonsiirtoon.

Käytettävät ohjelmat ja lisäosat

Tässä tutkimuksessa algoritmiavusteisen suunnittelun alustana toimii Rhinoceros 3D -ohjelman lisäosa Grasshopper. Grasshopper sisältää lukuisia tiedon käsittelyyn ja muokkaamiseen käytettäviä komponentteja. Tässä kokeiluhankkeessa käytetään pääasiassa vain valmiita ohjelmakomponentteja, jolloin suunnittelijalla ei tarvitse olla erityistä osaamista tekstimuotoisesta ohjelmoinnista. Komponenttivalikoimaa on mahdollista laajentaa myös useiden erilaisten lisäosien (engl. plug-in) avulla. Lisäosat toimivat usein esimerkiksi rajapintoina Grasshopperin ja muiden ohjelmistojen välillä. Lisäosa on tyypillisesti joukko Grasshopperin kankaalle sijoitettavia visuaalisia ohjelmakomponentteja, joiden toimintaperiaate ei juurikaan eroa alkuperäisistä komponenteista. Saatavilla olevien lisäosien sisältö ja laajuus vaihtelevat suuresti ja niitä ovatkin kehittäneet niin yksityishenkilöt kuin suuret ohjelmistoyrityksetkin. Pääasiassa lisäosat ovat ilmaisia ja ne ovat ladattavissa Food4Rhino-sivustolta. Tässä tutkimuksessa hyödynnetään monia Grasshopperin toimintaa laajentavia lisäosia ja niistä työn kannalta merkittävimmät ovat:

1. Grasshopper-RFEM Link.
2. Grasshopper-Tekla Live Link.
3. Speckle.

Grasshopper-RFEM Link (GH-RFEMLink) on A-Insinöörit Suunnittelu Oy:n kehittämä lisäosa, jonka avulla voidaan ohjata algoritmisesti Dlubal Software Inc:n RFEM-ohjelmistoa. RFEM on rakenteiden mitoitukseen ja analysointiin erikoistunut FEM-laskentaohjelma, joka soveltuu useiden erityyppisten rakenteiden suunnitteluun. GH-RFEMLink toimii rajapintana Grasshopperin ja RFEM:n välillä ja sen avulla on tällä hetkellä mahdollista rakentaa erilaisia laskentamalleja sekä mitoittaa sauvamaisia teräsrakenteita. GH-RFEMLink hyödyntää RFEM:n RF-COM -lisäosia, joka on sen avoin ohjelmoitava rajapinta. Lisäosan avulla on mahdollista luoda Grasshopperissa FEM-mallin sisältämät sauvat, niihin kohdistuvat piste- ja viivakuormat sekä rakenneosien ulkoiset tuennat. Lisäosan avulla voidaan määrittää myös muut laskentamallin kannalta olennaiset ominaisuudet, kuten esimerkiksi kuormaluokat ja kuormitusyhdistelyt. Laskentamallin luomiseen käytettävä komponentti on esitetty kuvassa 6.



Kuva 6. RFEM-mallin luomiseen käytettävä komponentti.

Kuten kuvasta 6 nähdään, voidaan lisäosan avulla käynnistää FEM-mallin laskenta. Kun laskenta on suoritettu, voidaan linkin välityksellä siirtää rakenteisiin kohdistuvat kuormitukset takaisin Grasshopperiin. Tämän lisäksi RFEM-linkin avulla voidaan ohjata RFEM:n RF-STEEL EC3 -lisäosaa ja siten suorittaa teräsosien Eurokoodin (EN 1993-1) mukainen mitoitus.

Grasshopper-Tekla Live Link on Trimblen kehittämä lisäosa Grasshopperin ja Tekla Structuresin välille. Linkki mahdollistaa tietomallin luomisen Teklaan Grasshopperia hyödyntämällä. Linkki sisältää lähes kaikki tietomallin luomiseen ja muokkaamiseen tarvittavat komponentit. Tällä hetkellä lisäosan ainoat rajoitteet ovat pultti- ja hitsauskomponenttien puuttuminen, mutta niidenkin luominen onnistuu osana liitoskomponentteja. Grasshopper-Tekla Live Linkin avulla on mahdollista esimerkiksi luoda erilaiset teräs-, betoni- ja puuosat sekä määrittää niiden attribuuttitiedot. Linkki mahdollistaa Grid-verkkojen, erilaisten liitoskomponenttien sekä betoniraudoitteiden luomisen. Lisäksi lisäosan

avulla voidaan linkittää Teklassa luotuja osia ja komponentteja Grasshopperiin ja muuntaa Tekla-objekteja Grasshopperin ymmärtämäksi yksinkertaiseksi geometriatiedoksi. (Grasshopper-Tekla Live Link) Grasshopper-Tekla Live Linkin komponenttivalikko on esitetty kuvassa 7.



Kuva 7. Grasshopper-Tekla Live Link komponenttivalikko.

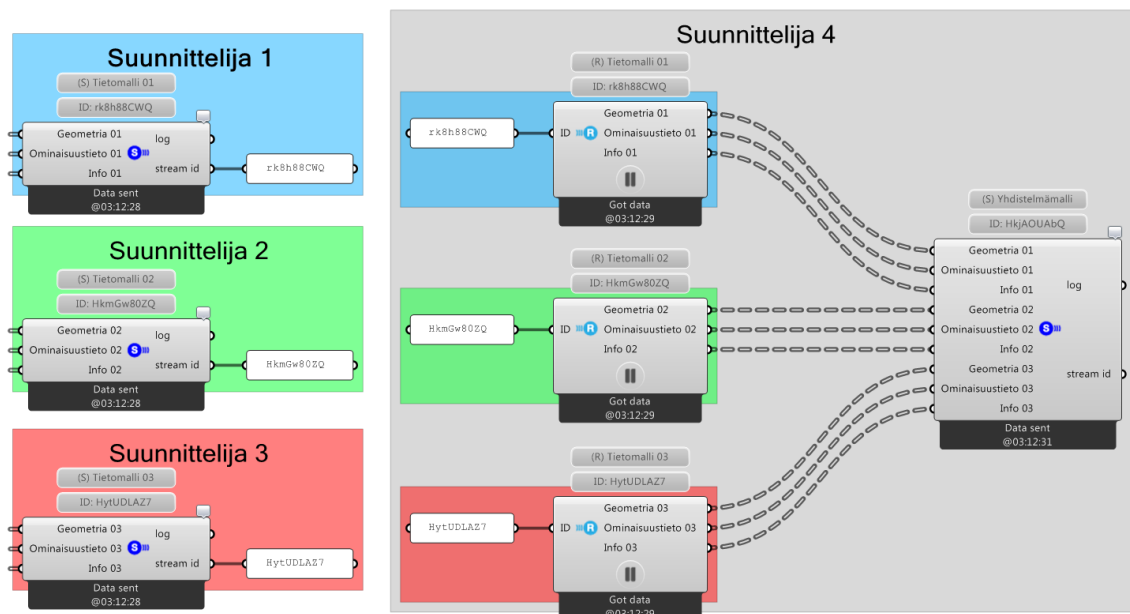
Listan viimeinen lisäosa on algoritmimallien pilvipohjaiseen jakamiseen tarkoitettu Speckle. Se on UCL The Bartlett School of Architecturen tutkimuksen pohjalta syntynyt tiedonjakoalusta, joka mahdollistaa algoritmiaivusteiselle suunnitteluprosessille ominaisen suunnittelutiedon vaiheittaisen kehittymisen. Sen avulla pienemmistä osakokonaisuuksista voidaan helposti muokata ja yhdistellä tarkoituksenmukaisia suunnittelukokonaisuuksia. Speckle on melko uusi lisäosa, mutta sitä kehitetään avoimen lähdekoodin menetelmällä, minkä vuoksi kehitys on ollut nopeaa ja käyttäjien tarpeisiin joustavasti mukautuvaa. Sen tarkoituksena on mahdollistaa algoritmipohjainen suunnitteluprosessi, jossa suunnittelutieto linkittyy älykkäästi sekä suunnitteluosapuolien että suunnittelumallien välillä. (Speckle Works)

Speckle-lisäosaa voidaan hyödyntää sekä Rhinon että Grasshopperin käyttöliittymässä. Molemmissa käyttöliittymissä peruseriaate on sama: haluttua tietoa voidaan joko lähettää pilvipalveluun tai vastaanottaa sitä sieltä. Käyttäjän tulee rekisteröityä ja kirjautua omilla tunnuksillaan Speckleen, jotta hän voi käyttää palvelua. Specklellä on tällä hetkellä oma serveri, jota voi hyödyntää tiedon varastointiin, mutta myös itsemääritetyn serverin käyttäminen on mahdollista. Tässä tutkimuksessa käytetään ainoastaan Specklen omaa serveriä. Pilvipalveluun ladattava suunnittelutieto on salattua ja sen voi määrittää olevan joko yksityistä tai julkista. Yksityiseen suunnittelutietoon voi lisätä oikeudet myös haluamilleen henkilöille, esimerkiksi kaikille projektiin osallistuville suunnitteluosapuolille. Speckle mahdollistaa näin niin sanotun multi-user työskentelyn, jossa useat eri osapuolet voivat muokata ja linkittää suunnittelutietoa keskenään reaaliaikaisesti. (Speckle Works)

Kun suunnittelija haluaa esimerkiksi jakaa geometriatietoa useisiin eri malleihin ja yhdistää nämä geometriat yhdeksi yhdistelmämalliksi, onnistuu se helposti Specklen avulla. Käytännössä haluttu suunnitteludata varastoidaan ensin Specklen määrittämän ID-tunnuksen ja itse määritetyn nimen avulla halutulle serverille. Tämän jälkeen haluttu suunnittelutieto voidaan hakea nimen tai tunnuksen avulla serveriltä ja hyödyntää sitä normaaliin tapaan. Kun alkuperäinen suunnittelutieto muuttuu, päivittyvät tiedot automaattisesti myös kaikkiin niihin malleihin, joissa tietoa hyödynnetään. Specklen komponentit toimivat dynaamisesti siten, että kun tiedon lähettävään komponenttiin lisätään uusi syöttö-

portti, muodostuu myös tiedon vastaanottaviin komponentteihin automaattisesti uusi tulostusportti samalla nimellä kuin syöttöportti. Tämä lisää tiedonsiirron selkeyttä ja helpottaa suunnitteluprosessin etenemistä erillisten algoritmimallien välillä.

Kuvassa 8 esitetään esimerkki Specklen toiminnasta. Siinä kuvataan kuvitteellinen tilanne, jossa kolme eri suunnittelijaa luovat kolme erilaista tietomallia siten, että suunnittelutieto päivittyy Specklen pilvipalveluun. Tämän jälkeen neljäs osapuoli kokoaa tietomallit yhteen ja jakaa muodostuneen yhdistelmämallin myös Specklen avulla valitulle serverille. Kaikki nämä neljä suunnittelijaa voivat työskennellä samanaikaisesti omien tiedostojensa parissa ja tieto linkittyy toisiinsa Specklen kautta automaattisesti.



Kuva 8. Esimerkki suunnitteluosapuolien välisestä pilvipohjaisesta tiedonsiirrosta Specklen avulla.

Edellä esitettyjen kolmen lisäosan lisäksi hankkeessa käytetään ja kokeillaan myös lukuisia muita sovelluksia. Monet lisäosat koetaan tutkimuksessa hyödyllisiksi, mutta niiden vaikuttavuus ei ole työn kannalta merkittävä.

Tiedonsiirto

Koska algoritmiaivusteinen suunnitteluprosessi perustuu toisiinsa linkitettyyn suunnittelutietoon, on tärkeää kiinnittää huomiota linkityksien oikeanlaiseen muodostamiseen. Erilaiset suunnittelukokonaisuudet linkittyvät toisiinsa aina jonkinlaisen rajapinnan kautta. Kahden erillisen tiedoston välinen tiedonsiirto voi tapahtua joko kahden algoritmimallin tai algoritmimallin ja jonkin perinteisen suunnittelutiedon, esimerkiksi BIM-mallin, välillä. Algoritmimalli linkittyy käytettävään suunnitteluohjelmistoon tyypillisesti siten, että algoritmeihin tehtävät muutokset päivittyvät reaaliaikaisesti linkitettyyn suunnittelutietoon. Algoritmien ja ulkopuolisten ohjelmistojen välinen rajapinta kannattaakin

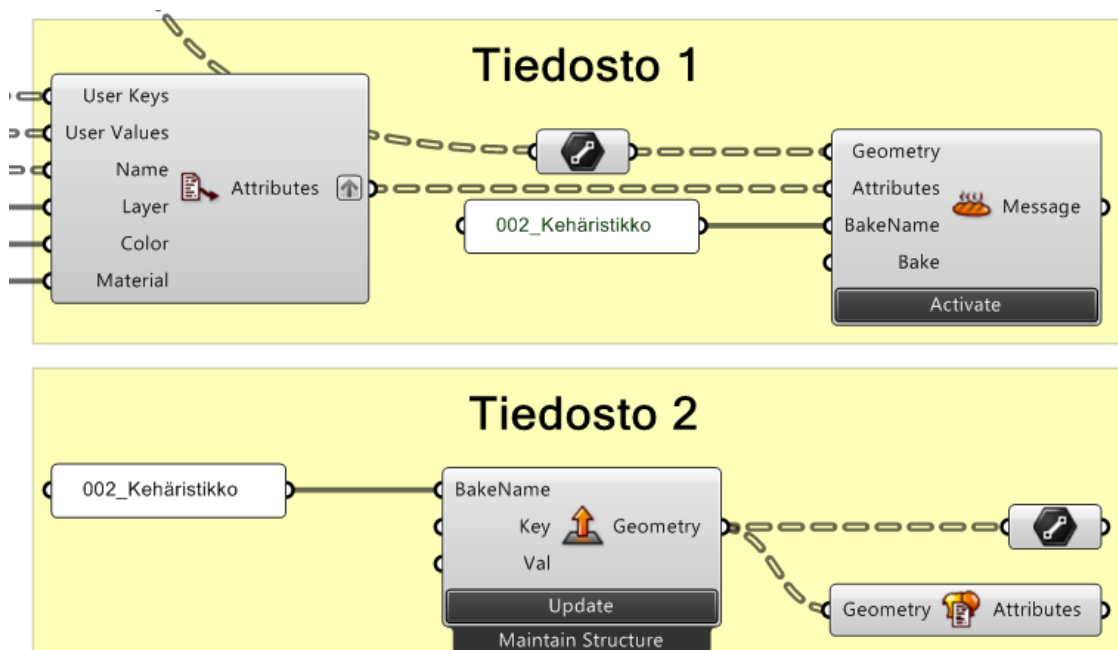
usein sulkea erityisesti suuria tiedostoja muokatessa. Tällöin algoritmeihin tehtävät muutokset eivät päivitä linkitettyä tiedostoa automaattisesti ja muokkaaminen säilyy nopeana. Rajapinta voidaan tyypillisesti sulkea helposti asettamalla yksi tarvittava arvo todesta epätodeksi.

Täysin algoritmisessa suunnitteluprosessissa kaikki tarvittava suunnittelutieto sijaitsee algoritmeissa eikä niillä ohjattavaa perinteistä suunnittelumallia ole välttämätöntä tallentaa ollenkaan. Esimerkiksi algoritmien avulla ohjattava FEM-malli voidaan luoda ja laskea sekä laskennan tulokset palauttaa algoritmimalliin kokonaan algoritmien avulla. Tällöin tiedostokooltaan suurta FEM-mallia ei ole välttämätöntä tallentaa ollenkaan, joskin se on suunnittelun raportoinnin kannalta suositeltavaa. Prosessin kannalta on usein luontevaa, että osa suunnittelusta tehdään algoritmiaivusteisesti ja osa perinteisiä menetelmiä käyttäen. Tällöin tulee kiinnittää erityistä huomiota ulkopuolisia ohjelmia ohjaavien lisäosien toimintaan. Useimmissa lisäosissa voidaan esimerkiksi valita poistaako komponentti aiemmin luodut tiedot vai ei. Jos algoritmien ohjaamaa mallia muokataan perinteisin keinoin, tulee tämä huomioida rajapinnan asetuksia määrittäessä. Esimerkiksi Tekla Live Linkin kautta mallia muokatessa rajapinta säilyttää aina kaikki BIM-mallissa luodut osat. Tästä syystä algoritmien avulla generoitavia osia ei tyypillisesti kannata tallentaa Tekla-malliin ennen kuin niiden algoritminen muokkaaminen on täysin valmis.

Kuten edellä esitetään, algoritmimallit linkittyvät usein johonkin perinteiseen suunnittelumalliin tai -tiedostoon jonkin Grasshopperin lisäosan avulla. Toinen oleellinen tiedonsiirtorajapinta sijaitsee kahden algoritmimallin välillä. Koska suunnittelussa hyödynnettäviä algoritmeja on hyvä jakaa kokonaisuuden selkeyttämiseksi pienempiin osiin, tulee myös algoritmimallien väliseen tiedonsiirtoon kiinnittää huomiota. Kahden algoritmimallin linkittäminen voidaan toteuttaa usealla tavalla, mutta tässä tutkimuksessa keskitytään tarkastelemaan kahta erilaista tapaa: Rhinoceros-ohjelman kautta tapahtuvaa sekä pilvipohjaisesti toteutettavaa tiedonsiirtoa. Koska Grasshopper on Rhinon lisäosa, yksi luotettava tapa siirtää tietoa kahden algoritmimallin välillä on muuntaa tietosisältö pysyväksi (engl. bake) Rhinossa. Tämä tapahtuu käytännössä siten, että Grasshopperin avulla luotavien algoritmien tulosteet, esimerkiksi geometriatieto, tallennetaan Rhino-tiedostoon ja tätä tiedostoa käytetään seuraavan algoritmimallin lähtötietona. Rhino-tiedostoon voidaan liittää myös erilaista metadataa ja ominaisuustietoa, jota voidaan hyödyntää seuraavan algoritmimallin luomisessa.

Rhinon kautta tapahtuvaan tiedonsiirtoon on kehitetty useita erilaisia lisäosia, joista yksi usein käytetty on Elefront. Sen perustehtävä on luokitella suunnittelutietoa Grasshopperin puolella siten, että tiedot siirtyvät myös Rhino-tiedostoon. Elefrontin avulla geometriatietoon voidaan liittää erilaisia ominaisuustietoja, kuten esimerkiksi nimi, tunnus, väri ja materiaali. Tämän jälkeen haluttu suunnittelutieto voidaan tallentaa pysyvästi Rhino-tiedostoon niin, että myös kaikki ennalta määritetyt ominaisuustiedot tallentuvat. Tallennettavalle suunnittelutiedolle tulee antaa nimi, jonka avulla siihen voidaan myöhemmin vii-

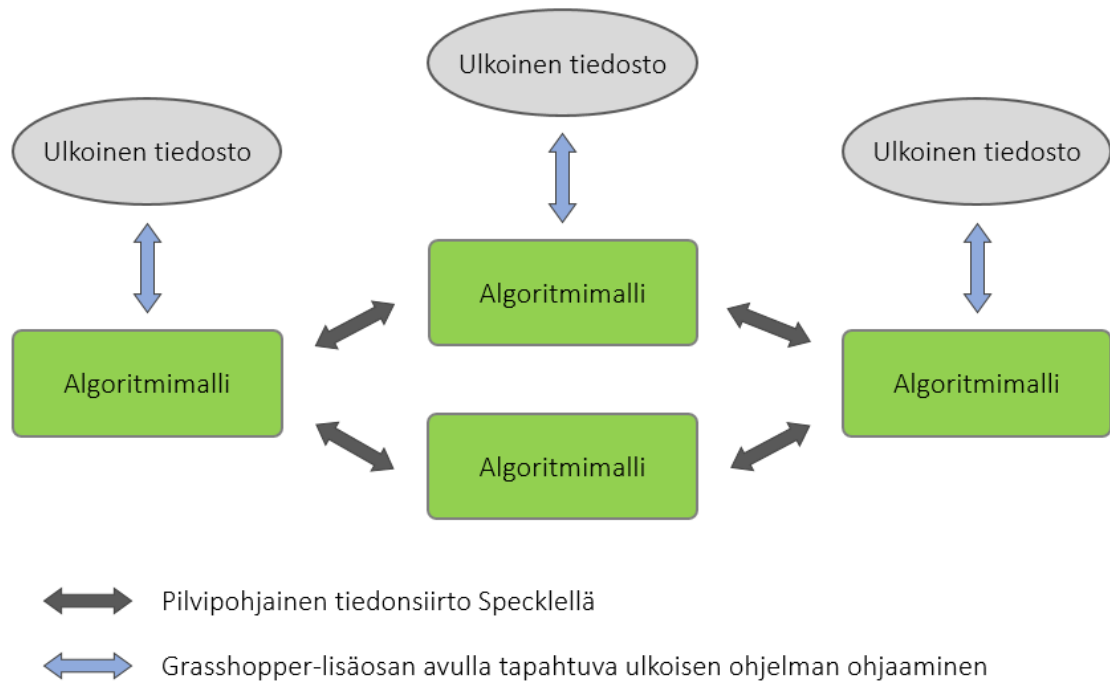
tata. Kun suunnittelutieto on tallennettu Elefrontin avulla Rhinoon ja Rhino-tiedosto tallennettu, voidaan tätä tiedostoa käyttää tiedonsiirtoon. Rhino-tiedostosta on mahdollista saada kaikki tarvittava suunnittelutieto ilman, että niiden generoimia algoritmeja tarvitsee jakaa. Suunnittelija voi avata Rhino-tiedoston ja viitata siihen tallennettuun suunnittelutietoon aiemmin määritetyn nimen tai tunnuksen mukaan ja saada näin suunnittelutiedon käyttöön myös Grasshopperissa. Kuvassa 9 esitetään esimerkki siitä, miten Elefrontin avulla halutulle geometriatiedolle syötetään ensin ominaisuustiedot ja tämän jälkeen suunnittelutieto muunnetaan pysyväksi Rhinossa ja sille annetaan tunnuksena toimiva nimi. Tämän jälkeen suunnittelutietoon voidaan viitata aiemmin määritetyn nimen avulla.



Kuva 9. Esimerkki algoritmimallien välisestä tiedonsiirrosta Elefrontin avulla.

Toinen tässä työssä tutkittava ja testattava algoritmimallien välinen tiedonlinkitystapa on pilvipohjainen tiedonsiirto. Siihen kokeillaan kolmea erilaista Grasshopperin lisäosaa: Flux, Konstru ja Speckle. Näistä kolmesta päädytään lopulta käyttämään Speckleä. Specklen avulla tiedonsiirto voidaan toteuttaa suoraan Grasshopper-tiedostosta toiseen ilman erillistä siirtotiedostoa. Specklen avulla on helppo siirtää kaikenlaista tietoa ilman lisäosan ylimääräisiä sisäisiä luokituksia. Kun suunnittelutieto siirtyy erillisen serverin kautta, säästytään manuaaliselta siirtotiedostojen lähettämiseltä. Pilvipohjaisen tiedonsiirron avulla algoritmiaivusteinen suunnitteluprosessi toimii huomattavasti automatisoidummin ja helpommin kuin Rhino-tiedostojen kautta tapahtuvassa tiedonsiirrossa. Speckleä käytettäessä viimeisin algoritmien luoma suunnittelutieto tallentuu aina käytettävälle serverille. Näin ollen, vaikka suunnittelun lähtötietona toimiva algoritmimalli suljetaan, pystytään sen pilvipohjaisesti varastoimaan tietosisältöä yhä muokkaamaan. Toisaalta kun algoritmimallia jälleen muokataan, päivittyvät muutokset reaaliaikaisesti pilvipalveluun ja sieltä tietoa hyödyntäviin algoritmimalleihin.

Tutkimuksessa kokeillaan erilaisia tiedonsiirtotapoja ja -ohjelmia, joiden avulla algoritmiaivusteinen suunnitteluprosessi voidaan toteuttaa. Työssä päädytään siihen tulokseen, että pilvipohjainen tiedonsiirto on käytännöllisin tapa jakaa tietoa eri algoritmimallien välillä. Algoritmimallit pyritään jaottelemaan siten, että yksi algoritmimalli on yhteydessä korkeintaan yhteen ulkoiseen ohjelmaan. Kuvassa 10 esitetään periaatekuva tässä tutkimuksessa käytettävästä algoritmiaivusteisen suunnitteluprosessin tiedonsiirrosta.



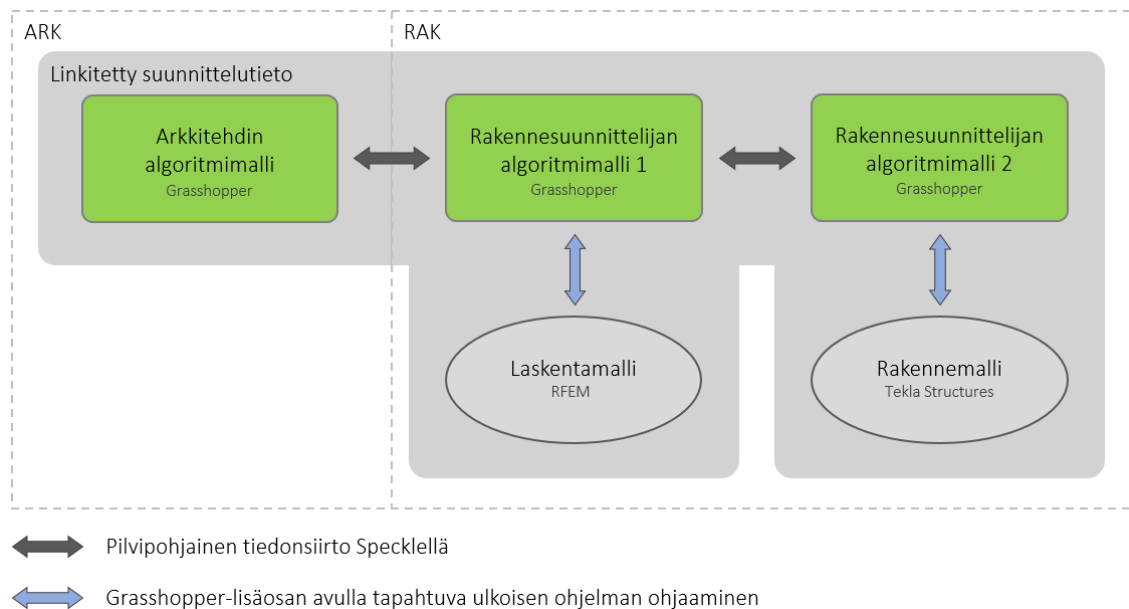
Kuva 10. Periaatekuva tutkimuksessa tapahtuvasta tiedonsiirrosta.

Erillisten mallien ja tiedostojen välisten rajapintojen lisäksi kiinnitetään erityistä huomiota algoritmimallien sisäiseen tiedonsiirtoon. Vaikka algoritmimallit pyritään säilyttämään maltillisen kokoisina, on niiden sisältöä hyvä jakaa vielä pienempiin osakokonaisuuksiin. Kun osakokonaisuudet nimetään, ryhmitellään ja visualisoidaan selkeästi, säilyvät algoritmimallit käyttökelpoisina myös tulevaisuudessa.

Teräsrakenteiden algoritmiaivusteinen suunnittelu

Tässä tutkimuksessa keskitytään tutkimaan erityisesti teräsrakenteiden algoritmiaivusteista suunnittelua. Rakenteiden mitoituksessa hyödynnetään RFEM-laskentaohjelmaa ja tietomallintamisessa Tekla Structures -ohjelmaa. Rakennesuunnittelun lähtötietona toimivat arkkitehdin muodostamat algoritmimallit, joista saadaan erityisesti rakenteiden päägeometria. Arkkitehdin ja rakennesuunnittelijan välinen tiedonsiirto toteutetaan Speckle-lisäosan avulla, jolloin koko arkkitehti- ja rakennesuunnittelun sisältävä prosessi toimii algoritmiaivusteisesti. Näin ollen kaikki suunnittelutieto on linkitetty yhteen systeemiin koko prosessin ajan ja mahdollisten muutosten generoiminen onnistuu algoritmien avulla.

Suunnitteluprosessi etenee arkkitehdin päägeometriasta rakenteellisesti toimiviin rakeneosiin ja liitoksiin. Aluksi arkkitehti luo ohjauspinnan kattorakenteen kokonaismuodolle. Tämän jälkeen rakennesuunnittelija luo arkkitehdin geometriatiedon pohjalta laskentamallin. Laskentamalli luodaan RFEM-laskentaohjelmaan GH-RFEMLink-lisäosaa hyödyntämällä. Laskentamallin avulla saadaan määritettyä rakenteiden sisäiset kuormitukset sekä kuormat rakenteellisesti kestävät teräsprofiilit. Kun rakenteiden mitoitus on suoritettu ja kestävyys varmistettu, voidaan saatujen tietojen pohjalta luoda rakennemalli. Rakennemalli luodaan Tekla Live Link -lisäosaa käyttämällä Tekla Structures -ohjelmaan. Rakennemallin luomisessa hyödynnetään sekä arkkitehdiltä saatua geometriatietoa että laskentamallista saatavia poikkileikkaustietoja. Kun rakennesuunnittelija on varmistunut rakenteiden kestävyydestä, palauttaa hän päivittyneen suunnittelutiedon arkkitehdille alakaton sekä vesikaton suunnittelua varten. Kuvassa 11 esitetään suunnitteluprosessin osakokonaisuudet sekä niissä hyödynnettävät ohjelmistot.



Kuva 11. *Teräsrakenteiden suunnittelussa hyödynnettävä algoritmiavusteinen suunnitteluprosessi.*

Kuten kuvasta 11 nähdään, suunnitteluprosessissa kaikki suunnittelutieto on linkitettyinä toisiinsa, jolloin manuaalista tiedonsiirtoa ei tarvita. Suunnitteluprosessi on täysin algoritmien avulla, jolloin kaikki suunnittelutieto luodaan ja muokataan algoritmien avulla. Ulkopuoliset ohjelmistot, RFEM ja Tekla, toimivat täten ainoastaan suunnittelutiedon havainnollistamisen sekä analysoimisen työkaluna. Algoritmisessa suunnittelussa tietokoneen rooli muuttuu tiedon esittäjästä tiedon generaattoriksi.

Vahvuudet ja hyödyntämismahdollisuudet

Tässä kokeiluhankkeessa kehitetty algoritmiavusteinen suunnitteluprosessi toimii kailta osiltaan hyvin. Tutkimuksen perusteella algoritmisilla menetelmillä on monia hyödyntämismahdollisuuksia ja niillä on monia etuja perinteisiin menetelmiin verrattuna. Algoritmiavusteisen suunnittelun suurimmat vahvuudet ovat erityisesti geometrisesti vaativien sekä toistuvien rakenteiden suunnittelussa. Tässä tutkimuksessa hyödynnettävän kattorakenteen muotoa voidaan pitää hyvänä esimerkkinä haastavasta geometriasta. Vaikka kantava rakenne muodostuukin pääasiassa suorista rakenneosista, on kaarevan kokonaismuodon luominen hyvin haastavaa perinteisin menetelmin. Lisäksi kattorakenne koostuu toistuvista rakenneosista, joiden hallitsemiseen ja muokkaamiseen algoritmit soveltuvat hyvin. Koska kattorakenteen viivageometria luodaan jakamalla kattopinta kolmionmuotoisiin pintoihin kahden avoimen numeroparametrin avulla, voidaan rakenneosien tiheysjakoa muokata helposti näitä kahta parametria muuttamalla. Näin ollen katon kaarevan muodon mukaan luotavien rakenneosien jakoa voidaan muuttaa algoritmien avulla reaaliaikaisesti, kun perinteisin menetelmin samojen muutosten tekeminen olisi hyvin työlästä ja aikaa vievää.

Haastavien muotojen ja toistuvien rakenteiden lisäksi algoritmisia menetelmiä hyödynnetään erityisesti suunnittelutiedon linkittämisessä. Perinteisessä suunnitteluprosessissa erilaisten suunnittelutietoa sisältävien ohjelmien ja tiedostojen välinen tiedonsiirto on hyvin manuaalista ja työlästä. Kun kaikki suunnittelutieto luodaan algoritmien avulla, voidaan suunnittelutiedon luovat algoritmit linkittää toisiinsa ja näin välttää mahdollisilta tiedonsiirto-ongelmilta. Tällöin kaikki perinteiset suunnitteluohjelmat, jotka voidaan linkittää algoritmeihin, voidaan linkittää myös toisiinsa. Tässä tutkimuksessa käytettäviä suunnitteluohjelmia ovat Tekla Structures, RFEM, Mathcad ja Excel. Suunnittelutiedon integraatio vähentää päällekkäistä työtä ja ristiriitaisuuksien syntymistä. Koska kaikkia suunnitteluohjelmia käytetään samojen rakenteiden suunnitteluun, on niissä paljon samaa suunnitteludataa. Algoritmien avulla riittää, että tämä suunnittelutieto luodaan kerran ja sen jälkeen se voidaan linkittää helposti kaikkiin tarvittaviin malleihin. Kun esimerkiksi arkkitehti on muodostanut kattorakenteen viivageometrian, voidaan sitä hyödyntää sekä laskenta- että rakennemallin luomisessa. Kun geometriatieto muuttuu, päivittyvät muutokset automaattisesti molempiin malleihin. Tällöin suunnitelmat pysyvät ajantasaisina sekä ristiriidattomina keskenään.

Suunnittelun nopeutumisen ja tehostumisen lisäksi algoritmisilla suunnittelumenetelmillä voidaan saavuttaa suuria kustannussäästöjä. Tämä perustuu siihen, että algoritmit ovat hyödyllinen työkalu erityisesti hankkeen alkuvaiheessa tapahtuvaan alustavaan suunnitteluun. Koska rakennushankkeen kokonaiskustannukset määräytyvät suurelta osin aivan hankkeen alkuvaiheessa, voidaan luonnossuunnittelussa tehtävillä ratkaisulla vaikuttaa merkittävästi hankkeen kokonaiskustannuksiin. Algoritmien luoma automaatio ja niiden parametrinen luonne mahdollistavat erilaisten vaihtoehtojen nopean vertailun. Suunnitte-

lun alkuvaiheessa ei myöskään tarvitse vielä kiinnittää huomiota pienimpiin yksityiskoh-
tiin, jolloin rakenteet luovien algoritmien logiikan määrittäminen on helpompaa. Algorit-
mien avulla voidaan luoda perinteisiä menetelmiä nopeammin hyvin erilaisia rakenteita
ja niiden muodostamia kokonaisuuksia. Kun arkkitehdin geometriatieto voidaan vielä
helposti linkittää laskentaohjelmaan, päästään tarkastelemaan nopeasti erilaisten raken-
nejärjestelmien toimivuutta. Kun rakenteisiin linkitetään vielä kustannustietoa, voidaan
eri rakennejärjestelmien kustannuksia vertailla nopeasti.

Ongelmat ja puutteet

Vaikka tässä työssä toteutettava algoritmien suunnitteluprosessi toimii kaikilta osiltaan
hyvin, törmätään tutkimusta tehdessä myös moniin ongelmiin ja puutteisiin. Suurin koko
prosessia koskeva ongelma on algoritmisen tiedonsiirron vaikeus. Jotta suunnittelukoko-
naisuus pysyisi selkeänä ja jotta siinä käytettäviä algoritmeja voidaan hyödyntää myös
tulevaisuudessa, tulee suunnittelukokonaisuutta jakaa erillisiin algoritmimalleihin. Tämä
tarkoittaa käytännössä erillisten Grasshopper-tiedostojen käyttöä. Lisäksi kun suunnitte-
luun osallistuu useita eri suunnitteluosapuolia, on erillisten tiedostojen käyttäminen suo-
tavaa. Algoritmeja voidaan helposti kopioida tiedostosta toiseen, mutta tällöin niiden vä-
linen älykäs linkitys katoaa. Algoritmimallien väliseen tiedonsiirtoon on erilaisia tapoja
ja lisäosia, mutta monet niistä koetaan prosessin kannalta huonoiksi.

Vaikka algoritmiavusteisen suunnittelun vahvuus on yksinkertaisessa ja tiiviissä muo-
dossa säilyvä suunnittelutieto, saattaa suuri tietomäärä silti muodostua ongelmaksi. Al-
goritmimallit ovat tiedostokooltaan huomattavasti perinteisiä suunnittelumalleja pienem-
piä, mutta erityisesti tiedonsiirto ja suunnittelumallien generoiminen vaativat silti paljon
laskentatehoa. Algoritmit kannattaa jakaa prosessin selkeyttämiseksi erillisiin algoritmi-
malleihin, jotka sitten voidaan linkittää toisiinsa esimerkiksi Speckle-lisäosan avulla. Tut-
kimusta tehdessä huomataan, että Speckle-komponenttien kautta tapahtuvassa tiedonsiir-
rossa saavutetaan melko nopeasti tiedonsiirtomaksimi. Tämä tarkoittaa sitä, että tietoa
lähettävä Speckle-komponentti ei pysty lähettämään kovinkaan suuria tietomääriä.
Grasshopper-tiedostoon voidaan asettaa samanaikaisesti useita tietoa lähettäviä kom-
ponentteja, minkä avulla tämä ongelma voidaan ratkaista. Tämä kuitenkin vähentää suun-
nittelun selkeyttä huomattavasti. Speckle on vielä melko uusi lisäosa, joten tällaisiin on-
gelmiin saattaa tulla parannuksia tulevaisuudessa.

Algoritmiavusteisessa suunnittelussa perinteiset suunnittelumallit, kuten esimerkiksi ark-
kitehti-, laskenta- ja rakennemalli luodaan algoritmien avulla. Kokeilussa huomataan, että
haastetta aiheuttavat erityisesti algoritmien ja perinteisen suunnittelun rajapinta. Jos esi-
merkiksi algoritmien avulla luotava tietomalli tallennetaan ja algoritmeja muokataan tä-
män jälkeen, generoivat algoritmit uudet osat vanhojen päälle. Tällöin tietomalli sisältää
kaksinkertaiset, päällekkäiset rakenneosat, mikä on tietysti väärin. Tästä syystä aina kun
suunnittelua jatketaan suunnitteluohjelmiston sulkemisen jälkeen, avataan tyhjä tiedosto,

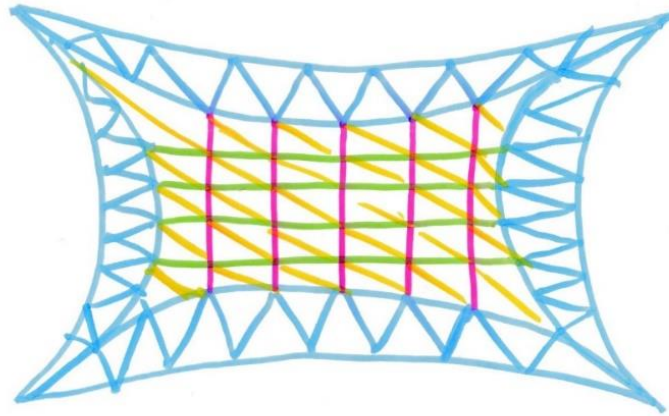
mihin rakenteet generoidaan uudestaan algoritmien avulla. Kattorakenteen kohdalla laskenta- ja rakennemallin tietomäärät säilyvät maltillisina ja näin ollen ne rakentuvat tyhjiin tiedostoon nopeasti. Jos tietomallien koko kuitenkin kasvaa huomattavasti suuremmaksi, saattaa mallin generoimiseen kuluva aika olla huomattavasti suurempi. Erityisesti Teklaan luotavan rakennemallin kohdalla tämä saattaa aiheuttaa ongelmia. Tällöin rakennemalli kannattaa luoda useiden algoritmimallien avulla ja tallentaa tietomalli normaaliin tapaan. Kun rakennemallia muokataan, tulee mallista poistaa muokattavat rakenneosat ja luoda ne uudestaan algoritmien avulla. Jos muokattavat rakenteet vaikuttavat rakennemallin muihin osiin, saattaa rakennemalli rakentua väärin ja luoda malliin ristiriitaisuuksia.

Rakennusten suunnittelussa pyritään käyttämään mahdollisimman paljon tyyppiratkaisuja, jotta rakenteet ovat helposti toteutettavissa ja jotta niiden kustannukset säilyvät maltillisina. Rakennukset voidaan kuitenkin muodostaa hyvin monenlaisista rakenneosista ja liitoksista, minkä vuoksi ne eroavat aina jonkin verran toisistaan. Tämä aiheuttaa haasteita algoritmiaivusteiselle suunnitteluprosessille. Algoritmien avulla luodaan logiikka siitä, miten erilaiset suunnitteluratkaisut muodostuvat. Jotta algoritmit toimisivat hyvin, tulee niitä luotaessa ottaa huomioon kaikki mahdolliset rakenteisiin kohdistuvat muutokset. Tämä on erityisesti suurien suunnittelukokonaisuuksien kohdalla käytännössä mahdotonta.

Algoritmien avulla voidaan luoda esimerkiksi Tekla-malliin kaikki tarvittavat rakenneosat ja liitokset, mutta niiden muodostaminen saattaa olla huomattavasti hitaampaa perinteisiin menetelmiin verrattuna. Jos suunniteltavat rakenteet muodostavat esimerkiksi monimutkaisen kokonaisuuden ilman matemaattisia säännönmukaisuuksia, on hyvin vaikeaa luoda algoritmeja, jotka pystyvät reagoimaan rakennejärjestelmään kohdistuviin muutoksiin. Tällöin algoritmiaivusteisella suunnittelulla ei saavuteta minkäänlaista lisäarvoa perinteisiin menetelmiin verrattuna. Algoritmien avulla voidaan luoda myös logiikkaa, joka pystyy reagoimaan monimutkaisiin kokonaisuuksiin. Tämä on kuitenkin hyvin työlästä, ja jos algoritmeja ei pystytä enää tulevaisuudessa hyödyntämään, voidaan niiden käyttöä pitää tässä tapauksessa turhana.

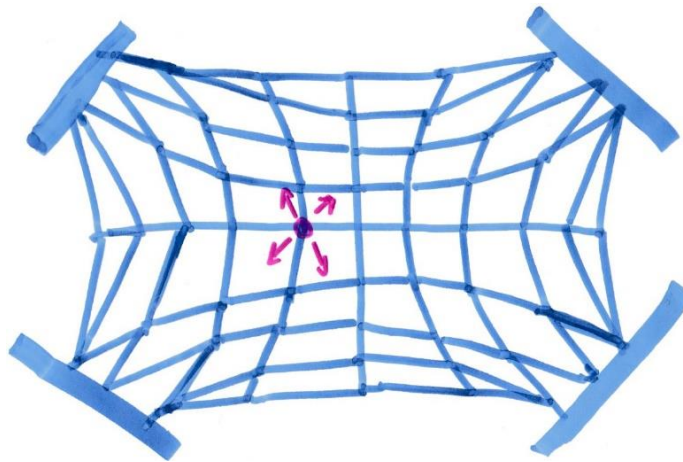
Jatkotutkimuskohteet

Tässä kokeiluhankkeessa algoritmiaivusteista suunnitteluprosessia tutkitaan teräsrakenteisen kattorakenteen avulla. Kattorakenteen kantavan rungon suunnittelua voitaisiin jatkokehittää esimerkiksi rakennettavuuden sekä rakenteisiin kohdistuvien voimien ja rasituksen näkökulmasta. Kattorakenteen perusmuodon pintaan pitäisi tällöin sovittaa tietyn rakennekonseptin mukainen malli. Ensimmäinen, niin sanottu perinteinen, tapa on muodostaa primääri-, sekundääri-, tertiäärirakenne jne. Rakentamisen logistinen vaatimus pilkkoo rakennuksen osiin. Primäärirakenteina toimivat esimerkiksi nivelöidyt reunakaaret. Nämä yhdistetään lakialueen ylittävillä poikittaisilla kaarilla. Rakennemallia on havainnollistettu kuvassa 12.



Kuva 12. Rakenneosien luokitteluun perustuva rakennekonsepti.

Toinen vaihtoehto on suunnitella avaruusristikko, jossa kattokuoren muoto on lukittu toiminnallisista ja esteettisistä syistä johtuen. Rakenneosien tiheysjako määritellään avaruusristikolle iteraation lähtökohdaksi. Solmupisteiden paikkoja haetaan dynaamisen relaksaation avulla siirtelemällä solmupisteitä kuoren pinnassa. Tässä kehitystyön kohteena olisi ohjausfunktioiden kehittäminen relaksaatiomallia varten. Rakennekonseptia on havainnollistettu kuvassa 13.



Kuva 13. Solmupisteiden dynaamiseen relaksaatioon perustuva rakennekonsepti.

Tässä kokeiluhankkeessa teräsrakenteiden poikkileikkaukset optimoidaan RFEM-laskentaohjelmaa hyväksi käyttäen. Jatkossa tämä optimointi voidaan mahdollisesti laajentaa myös kokonaisten rakennejärjestelmien optimointiin. Optimoinnin toteutus on tällä hetkellä vielä melko hidasta, mutta sitä on mahdollista kehittää. Kun optimointi kohdistetaan kokonaisiin rakennejärjestelmiin siten, että niiden geometria voi muuttua, tekee se optimoinnista selvästi haastavampaa. Jos kokonaisten rakennejärjestelmien älykäs optimointi saadaan toteutettua hyvin, saatetaan sillä saavuttaa hyvin merkittäviä kustannussäästöjä. Jos rakennejärjestelmiä aletaan optimoida rakennesuunnittelussa niiden geometriaa

muokkaamalla, saattaa se vaikuttaa koko suunnitteluprosessin toteutukseen. Koska arkkitehti päättää tyypillisesti päägeometriasta heti suunnitteluprosessin alkuvaiheessa, vaatii kokonaisvaltainen optimointi täysin uudenlaista yhteistyötä. Optimoinnilla voidaan kuitenkin saavuttaa hyvin merkittäviä etuja, minkä vuoksi se on tärkeä jatkotutkimuskohde.

Erityisesti koko suunnitteluprosessin kannalta oleellinen tutkimuskohde on prosessissa tapahtuva tiedonsiirto. Pilvipohjainen tiedonsiirto koetaan tutkimuksessa oivalliseksi menetelmäksi, mutta myös kehitettäviä asioita löytyy. Erityisesti suurten tietomäärien siirtoon ja varastointiin tulee kiinnittää huomiota. Jos suunnittelutietoa ja erityisesti muodostettuja algoritmeja voidaan varastoida jollakin järkevällä tavalla, voidaan niitä käyttää tehokkaasti myös tulevaisuudessa. Jos suunnittelussa käytettävät algoritmit luodaan systemaattisesti ja raportoidaan selkeästi, voidaan niitä hyödyntää muissakin projekteissa. Kaikista algoritmeista voisi olla mahdollista rakentaa laaja datapankki, josta niitä voitaisiin käydä älykkäästi hakemassa samankaltaisen suunnittelutehtävän sattuessa eteen. Vaikka tässä tutkimuksessa rakennesuunnittelija saa lähtötietonsa arkkitehdilta pilvipalvelun kautta, voidaan tätä menetelmää yhä jatkokehittää. Koska algoritmien avulla tapahtuva reaaliaikainen, pilvipohjainen tiedonsiirto koetaan hyväksi menetelmäksi, kannattaa sen mahdollisuuksia kokeilla myös muiden rakennushankkeeseen osallistuvien tahojen välillä.

Algoritmiavusteinen suunnittelu etenee algoritmeihin ohjelmoitujen sääntöjen mukaan. Nykyään suunnittelussa voidaan hyödyntää myös tietokoneen tekoälyä. Tämä onkin hyvin merkittävä jatkotutkimuskohde. Tekoälyä on mahdollista hyödyntää suunnittelussa hyvinkin laajasti. Tekoälyä voidaan hyödyntää hyvin eritasoisissa ongelmissa, mutta parhaimmillaan se voi oppia tunnistamaan rakenteisiin kohdistuvat kuormitukset ja luomaan automaattisesti niihin soveltuvat rakenneosat. Esimerkiksi liitossuunnittelussa algoritmit luovat täsmällisesti sellaisen liitoksen kuin suunnittelija määrittää. Kun liitossuunnittelussa hyödynnetään tekoälyä ja koneoppimista, suunnittelee tietokone liitoksen oman mielensä mukaan sille annetun opetusdatan puitteissa. Tämä tarkoittaa sitä, että jos tekoälyn halutaan osata muodostavan tyypillisesti käytettyjä rakenteita ja liitoksia tulee niille syöttää riittävän paljon monipuolista opetusdataa. Tekoälyn hyödyntäminen on hyvin monimutkainen ja haastava kokonaisuus ja se on yksi merkittävimmistä algoritmiavusteisen suunnittelun jatkotutkimuskohteista.

Hankkeen toteutunut viestintä

Kokeiluhankkeen kulusta ja siinä saavutetuista tuloksista on viestitetty erilaisissa tilaisuuksissa ja tiedotuskanavissa. KIRA-digi -organisaation kautta tapahtuneiden esittely- ja tulostilaisuuksien lisäksi kokeiluhankettamme on esitelty esimerkiksi Teräsrakenneyhdistyksen T&K -päivillä Hämeenlinnassa 15.8. Lisäksi hankkeesta on tiedotettu A-Insiöörien nettisivuilla sekä sosiaalisen median kanavissa. Yksi esimerkki tulosten toteutuksesta viestinnästä löytyy täältä: <https://www.youtube.com/watch?v=aYfP-hpfRrU>

LÄHTEET

Grasshopper-Tekla Live Link, Tekla Structures Support, verkkosivu. Saatavissa (viitattu 20.6.2018): https://teklastructures.support.tekla.com/not-version-specific/en/ext_grasshopperteklalink

Speckle Works, verkkosivu. Saatavissa (viitattu 20.6.2018): <https://speckle.works/>.