

Janne Salasmaa

MagiCADilla ja NavisWorksilla tehtävän suunnit- telu- ja mallinnustyön tehostaminen apuohjel- mien avulla

Metropolia Ammattikorkeakoulu

Insinööri (YAMK)

Talotekniikka

Opinnäytetyö

11.4.2018

Tekijä Otsikko Sivumäärä Aika	Janne Salasmaa MagiCADilla ja NavisWorksilla tehtävän suunnittelu- ja mallinnustyön tehostaminen apuohjelmien avulla 37 sivua 11.4.2018
Tutkinto	Insinööri (YAMK)
Tutkinto-ohjelma	Talotekniikka
Ammatillinen pääaine	LVI-tekniikka
Ohjaajat	Lehtori Katri Onnela Teknologijahtaja Tero Järvinen
<p>Opinnäytetyössä laadittiin työssään tietomallinnusta käyttävän talotekniikkasuunnittelijan päivittäistä työtä helpottavia ja tehostavia apuohjelmia. Työ tehtiin Granlund Oy:lle ja siinä keskityttiin apuohjelmiin, jotka liittyvät ohjelmistoihin MagiCAD for AutoCAD ja NavisWorks, eli Granlundin tämän hetken tärkeimpiin tietomallinnustyökaluihin.</p> <p>Granlundin talotekniikkasuunnittelijat käyttävät käytännössä työssään samanaikaisesti MagiCADia ja NavisWorksia. Siksi opinnäytetyön tavoitteeksi asetettiin näiden ohjelmien yhteistoiminnan parantaminen ja tehostaminen apuohjelmien kautta.</p> <p>Laadittuja apuohjelmia ovat</p> <ul style="list-style-type: none"> - "Näytä osa Naviksessa". Tämän avulla MagiCADissa valittu osa löydetään helposti NavisWorks-yhdistelmämallista. - "Näytä osa AutoCADissa". Tämän avulla NavisWorks-yhdistelmämallissa valittu osa löydetään helposti asiaankuuluvasta CAD-piirustuksesta. - "CUTNAVIS" ja "CUTNAVISXY". Nämä ovat apuohjelmia, joilla leikataan erilaisia alueita näkyviin yhdistelmämallista. Alueen määrittäminen tapahtuu MagiCADissa. <p>Apuohjelmat hyödyntävät ohjelmointirajapintoja NavisWorks API, AutoCAD .NET ja MagiCAD COM Interface. Käytännön ohjelmointityö tehtiin Microsoft Visual Studiolla, ohjelmointikielellä C#.</p> <p>Apuohjelmat on otettu käyttöön koko Granlund-konsernissa. Apuohjelmia käyttää päivittäin arviolta yli 100 henkilöä. Ohjelmien käyttö nopeuttaa ja sujuvoittaa työskentelyä. "Epäviralisen" palautteen perusteella on havaittavissa, että ohjelmia yleisesti pidetään tärkeinä ja hyödyllisinä.</p>	
Avainsanat	LVI-suunnittelu, tietomallinnus, MagiCAD, NavisWorks

Author Title	Janne Salasmaa Optimizing the Design and Modeling Workflow Done with MagiCAD and NavisWorks with Add-in Software
Number of Pages Date	37 pages 11 April 2018
Degree	Master of Engineering
Degree Programme	Building Services Engineering
Professional Major	Building Services Engineering
Instructors	Katri Onnela, Senior Lecturer Tero Järvinen, Technology Director
<p>In the final year project, several add-in programs for enhancing the tools of building services designer were created. The project concentrated specifically to speeding up the workflow of a designer who mainly uses MagiCAD and NavisWorks simultaneously as tools, the method that is currently considered the best way to produce building services design documentation for complex buildings. However, this simultaneous use will cause some waste of time, unless add-in software is used. Therefore, following software was created:</p> <ul style="list-style-type: none"> - “Show part in Navis” to quickly and easily find a part, selected in MagiCAD, from the 3D-model in NavisWorks. - “Show part in Autocad” to quickly and easily find a part, selected in NavisWorks, from the corresponding CAD-drawing. - “CUTNAVIS” and “CUTNAVISXY” to cut sections and areas to be visible in NavisWorks but defining the geometry of the section in MagiCAD. <p>The developed add-in software has already been used for some time, by hundreds of people. Only unofficial feedback has been gathered so far, but judging from that, the software is commonly considered necessary and useful.</p>	
Keywords	building services design, BIM, MagiCAD, NavisWorks

Sisällys

Lyhenteet

1	Johdanto	1
2	Tausta	4
3	Tavoitteen asettelu ja laaditut apuohjelmat	6
3.1	"Näytä osa Naviksessa"	6
3.2	"Näytä osa Autocadissa"	8
3.3	"CUTNAVIS"	10
3.4	"CUTNAVISXY"	13
4	Työn toteutus	18
4.1	Ohjelmointirajapinnat	18
4.2	Ohjelmointityökalut	19
4.3	Ohjelmien rakenneperiaatteita	19
4.4	Ohjelmien toteutus	21
4.4.1	"Näytä osa Naviksessa"	22
4.4.2	"Näytä osa Autocadissa"	24
4.4.3	CUTNAVIS	25
4.4.4	CUTNAVISXY	27
4.5	Ohjelmien päivitys	27
5	Mahdollisia kehityskohteita	28
5.1	NWCOUT	28
5.2	Historiatiedot	29
5.3	Asennusten järjestely yhdistelmämallissa	29
5.4	Leikkaustulosteiden teko yhdistelmämallista	32
6	Yhteenveto	35
	Lähteet	37

Lyhenteet

API	Application programming interface, ohjelmointirajapinta.
COM	Component Object Model. Eräs tapa mm. tiedonsiirtoon eri ohjelmien välillä.
DWG	AutoCADin tiedostomuoto.
IFC	Industry Foundation Classes, rakennusten tietomallinnuksessa yleisimmin käytetty eri osapuolten välinen tiedonsiirtomuoto.
NWC	NavisWorksin tiedostomuoto.

1 Johdanto

Granlund on lähes 800 henkilöä työllistävä talotekniikkasuunnittelun, kiinteistö-, energia- ja ympäristöasioiden konsultoinnin sekä ohjelmistojen asiantuntijakonserni (1). Granlundin suurin toimiala on talotekniikkasuunnittelu, josta puolestaan suurinta osaa hallitsee LVI-suunnittelu ja hyvänä kakkosena on sähkösuunnittelu.

Ennen rakennushankkeen toteuttamista ja usein myös osittain toteutuksen kanssa ajallisesti rinnakkain laaditaan taloteknisistä järjestelmistä toteutussuunnitelmat. Yleensä nämä suunnitelmat laaditaan tietomallintamalla. Tietomallinnus tapahtuu nykyään käytännössä henkilökohtaisella CAD-työasemalla ("pöytäkone" tai "kannettava") ja sen seurauksena tieto suunnitelluista järjestelmistä saadaan tallennettua tietokoneohjelmien "ymmärtämään" muotoon. Tämä mahdollistaa paitsi perinteisten paperikuvien tulostamisen, myös mm. verkostojen virtauslaskelmien teon ja verkostojen 3D-mallien tarkastelun.

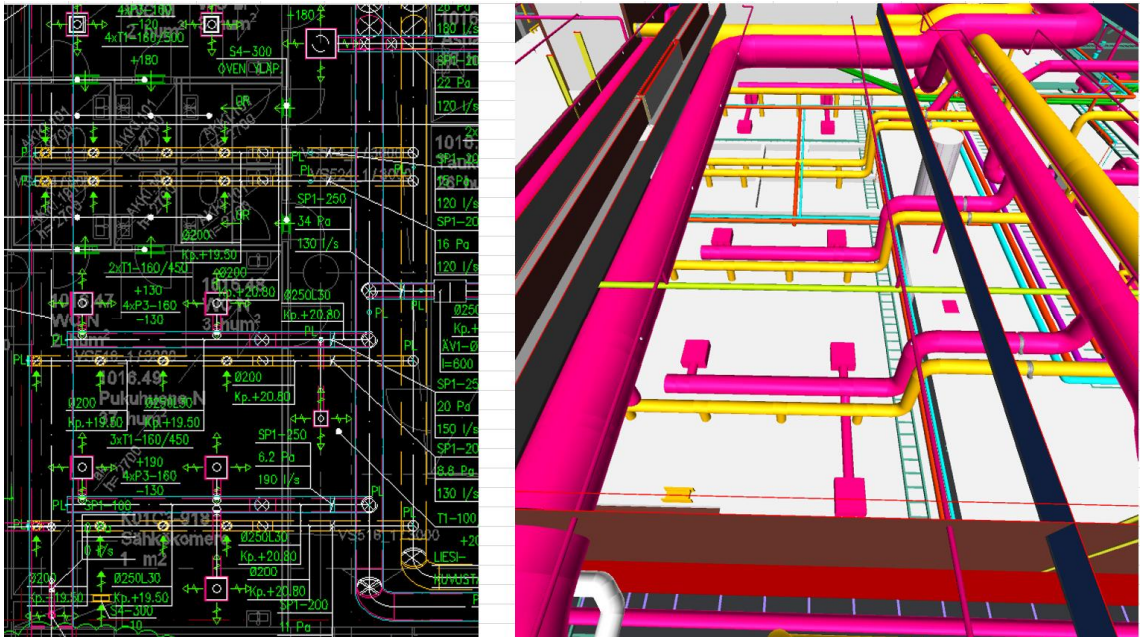
Granlundin suunnittelukohteet ovat usein suuria ja niissä on lukuisia taloteknisiä järjestelmiä. Tällaisten kohteiden tietomallinnuksessa on toistaiseksi todettu parhaaksi käyttää kahden ohjelman yhdistelmää. Nämä ohjelmat ovat MagiCAD for AutoCAD ja NavisWorks. MagiCAD for AutoCAD on CAD-suunnitteluohjelmisto AutoCADiin Progran Oy:n tekemä laajennusohjelmisto. AutoCAD on Autodesk-nimisen suuryrityksen tuote. NavisWorks on niin ikään nykyään Autodeskin omistama tuote.

MagiCADia käyttäen LVIS-suunnittelija/mallintaja tuottaa esimerkiksi ilmanvaihtosuunnitelman. NavisWorksin avulla hän voi suunnittelun/mallintamisen yhteydessä säännöllisesti seurata ja tarkastaa, miten tämä MagiCADilla luotu 3D-malli sopii yhteen rakennuksen muun tietomallin kanssa (rakenteet, muu LVI, sähköasennukset jne). Pääreitit ja verkostojen periaatteelliset sijoitukset toisiinsa nähden tietysti sovitaan ennen mallinnuksen alkamista, laatimalla eri tekniikanalojen suunnittelijoiden yhteistyönä mm. leikkauspiirustukset oleellisimmilta osin. Useimmiten rakennuksen geometria on sen verran monimutkainen ja verkostot pitää saada mahtumaan niin pieneen tilaan, että mallinnusvaiheessa tulee monta ns. tenkkapoota, vaikka olisi kuinka hyvin laaditut leikkaukset lähtökohtana.

Edellä mainittuun kahden ohjelman yhtäaikaiseen käyttöön liittyen on kaikilla Granlundilla talotekniikan tietomallinnukseen osallistuvilla henkilöillä työpisteensä varustuksena vähintään kaksi näyttöä. Verkostojen tietomallinnusta tehtäessä yleisin tilanne on se, että yhdessä näytössä on MagiCAD/pohjakuvanäkymä esim. jonkin kerroksen ilmanvaihtosuunnitelmasta ja toisessa näytössä NavisWorks/3D-näkymä yhdistelmämallista siitä kohtaa rakennusta jota MagiCADissa parhaillaan työstetään. sivulla 3 Kuva 1 alla näyttää esimerkiksi tilanteen, jossa on IV-pohjapiirustus ja vastaava kohta yhdistelmämallia, jossa ovat näkyvissä myös betonirakenteet, alakatot, vesi- ja viemäri- sekä sähkö- ja sprinklerasennukset.

MagiCADissa tehtävän mallinnuksen edistyminen ei päivity aivan reaaliajassa NavisWorksiin, vaan mallintaja tekee aina järkeväksi katsomanaan ajakohtana "NWCOUT"-komennon, joka luo aktiivista DWG:tä vastaavan NWC-tiedoston, joka on puolestaan asianomaiseen kohtaan NavisWorks sin yhdistelmämallia linkitetty. NWC on NavisWorks sin oma tiedostomuoto.

Periaatteessa NavisWorksiin on mahdollista linkittää myös suoraan DWG-tiedostoja, mutta MagiCADin tapauksessa tämä ei ole käyttökelpoinen tapa. Nimittäin jos MagiCADissa on 2D-esitys päällä (vrt. Kuva 1), kuten useimmiten on, linkitetty DWG näkyy NavisWorks sissa 2D:nä, eli se on 3D-yhdistelmämallissa käyttökelvoton. Tämä puute johtuu siitä että Progman ei ole laatinut MagiCADille asianmukaista ns. Object Enableria (2).

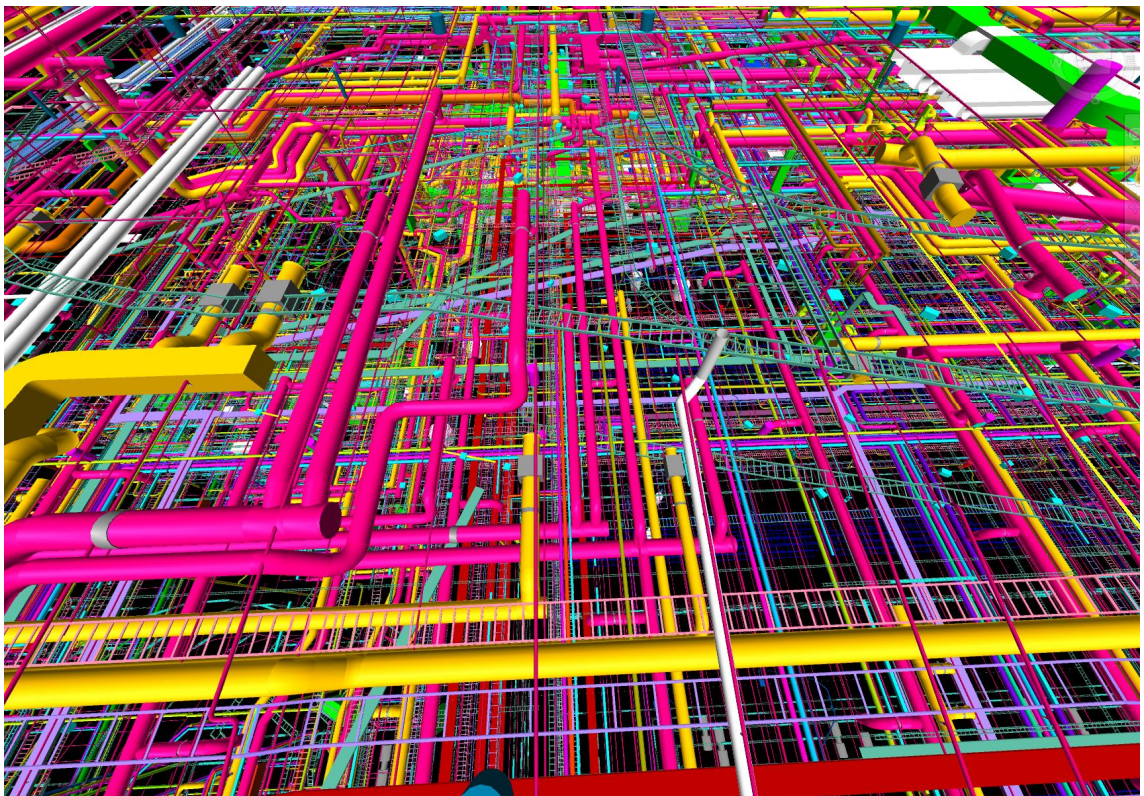


Kuva 1. IV-pohjapiirustus ja vastaava kohta yhdistelmämallia

Edellä mainitussa perustyöskentelyssä MagiCADilla ja NavisWorksilla siis käsitellään samaa asiaa eri näkökulmista. Näin ollen käyttäjän kannalta olisi edullisinta, jos näillä kahdella ohjelmalla olisi tiivis yhteys keskenään ja niitä voisi ohjata kuin yhtä kokonaisuutta. MagiCADilla ja NavisWorksilla ei kuitenkaan ole tällaista yhteyttä toisiinsa. Näin ollen, kun työntekijä esim. vaihtaa aluetta, jota MagiCADissa työstää, joutuu hän myös hakemaan saman alueen näkyviin NavisWorks-mallista. Tämä voi olla yllättävän paljon aikaa vievää erityisesti sellaisen henkilön osalta, joka ei ole kovin kokenut NavisWorksin käyttäjä ja/tai ei tunne mallinnettavaa rakennusta ”kuin omia taskujaan”. Edellä mainituista syistä tämän opinnäytetyön tavoitteeksi asetettiin ohjelmien käytettävyyden ja sitä kautta työn tuottavuuden parantaminen MagiCADiin ja NavisWorksiin kytkeytyvien apuohjelmien avulla.

2 Tausta

Vaatimukset rakennusten sisäolosuhteille ovat kasvaneet lähimenneisyyden aikana huomasti. Ennen vanhaan, sään ollessa tyyni ja kuiva, on ollut aivan normaalia esim. pyyntiretkillä yöpyä pelkkään lumihankeen ilman minkäänlaista rakennettua suojaa (3, s.374). Nykyään sen sijaan ollaan tilanteessa, jossa rakennukset on varustettava runsain taloteknisin järjestelmin (Kuva 2), jotta rakennuksen olosuhteet olisivat sellaiset, että niihin tyytymättömien käyttäjien osuus kaikista käyttäjistä on riittävän pieni.



Kuva 2. Näkymä erään suuren rakennuksen (keskeneräisestä) taloteknisestä tietomallista.

Sen lisäksi, että kunkin järjestelmän on toimittava suunnitellulla tavalla, sen on myös mahdollista muiden asennusten sekaan ja oltava huollettavissa, eli eri suunnittelualueiden suunnitelmien on oltava yhteen sovitettuja. Koska myös suunnittelun aikataulut ovat useimmiten kiireisiksi puristettuja, tietomallintaminen on käytännössä ainoa tapa jolla voidaan saada suurten ja monimutkaisten rakennusten suunnitelmat riittävän laadukkaiksi ja ristiriidattomiksi aikataulun puitteissa.

Tämän opinnäytetyön tavoitteena on saada aikaan hyödyttömän ajankäytön vähene- mistä. Nykypäivänä kiirettä riittää muutenkin, ei ole mielekästä, että ihminen käyttää toistuvasti aikaansa sellaisen asian tekemiseen, joka voidaan sopivalla tietoteknisellä ratkaisulla tehdä automaattisesti.

Talotekniikan suunnittelu- ja mallinnustyötä tekevällä henkilöllä on tietomalliyhteensovi- tuksen lisäksi tehtävänä myös laajempi, koko elämään liittyvä yhteensovitus. Seu- raavat, keskenään joskus ristiriidassa olevat asiat olisi saatava hallittua:

- Vastuu työn tuloksesta. Työntekijä saa työnantajalta palkkaa ja siksi hänellä on moraalinen velvollisuus tehdä työssään parhaansa.
- Vastuu maapallon tulevaisuudesta. Talotekniset suunnitelmat pitäisi saada laa- dittua myös niin, että kyseisten järjestelmien energiankäyttö on mahdollisimman vähäistä ja myös niiden rakentamiseen tarvittavan raaka-aineen määrä olisi saatava minimoitua.
- Vastuu oman ammattikunnan tulevaisuudesta. Työntekijän on myös muistetta- va pysyä ns. hereillä, etteivät asiat pitkällä aikavälillä mene siihen suuntaan, et- tä työntekijöiden asema suhteessa työnantajaan heikkenee. Esimerkiksi työ- sähköpostien lukeminen loman aikana on tavallaan karhunpalvelus ja ”ennak- kopuukotus selkään” tuleville sukupolville, koska mitä useampi sähköposteja lomillaan lukee, sen normaalimmaksi käytäntö muodostuu ja ennen pitkää se onkin muuttunut kuin varkain jo vaatimukseksi.
- Vastuu yksityiselämästä. Suunnittelutyön laatu kärsii väistämättä jos työ pilko- taan liian pieniin osiin. Toisaalta aikataulut ovat pääsääntöisesti kireitä. Tunnol- linen työntekijä haluaa kuitenkin tehdä hyvää jälkeä aikataulun puitteissa mikä usein johtaa ylityöpaineisiin. Toisaalta samaan aikaan henkilöä tarvittaisiin myös muualla, esim. kotona, varsinkin jos hänellä on pieniä lapsia. Ja vaikka ei olisikaan, useimmilla ihmisillä on runsaasti kaikenlaista tehtävää joka on työteh- tävien kanssa kilpailemassa käytettävissä olevasta ajasta.

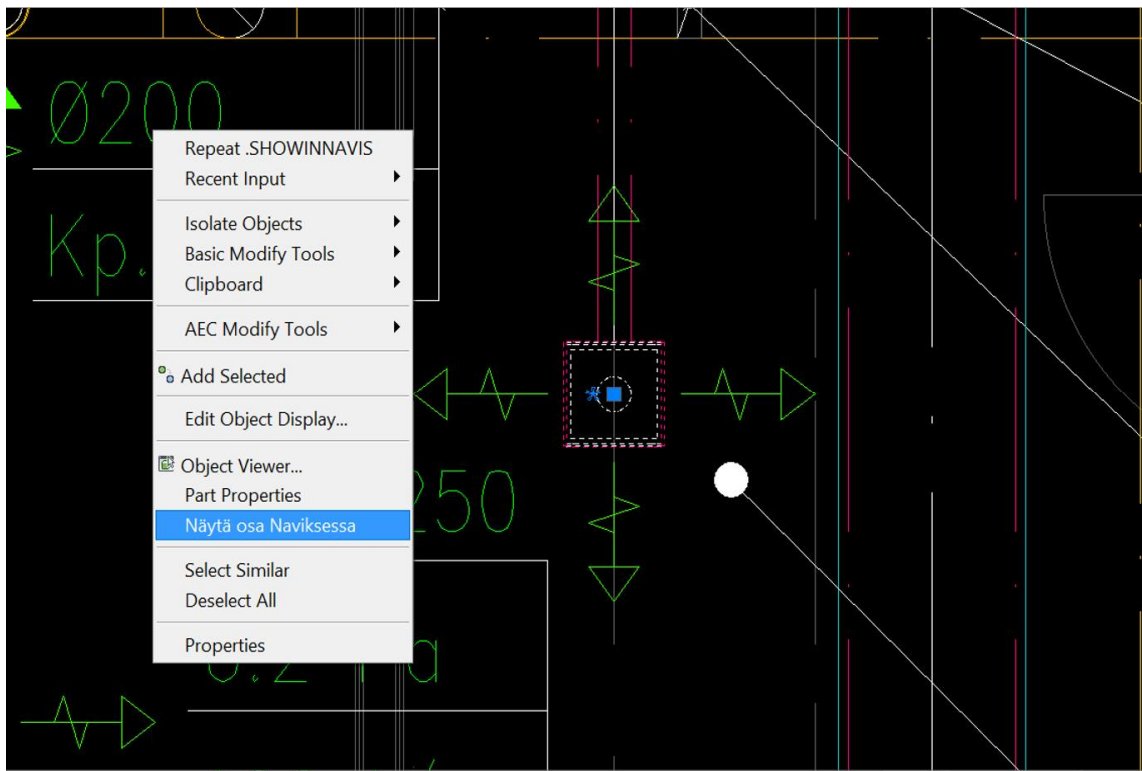
Kukin ratkaisee omalta osaltaan, minkälaisen painoarvon antaa kullekin edellä mainitulle asialle. Joka tapauksessa edellä mainitunkin valossa selvää on, että työkaluista ai- heutuva ajanhukka kannattaa ehdottomasti pyrkiä minimoimaan.

3 Tavoitteen asettelu ja laaditut apuohjelmat

Seuraavassa on lyhyt kuvaus, minkälaisen apuohjelmien laadinta asetettiin opinnäytetyön tavoitteeksi.

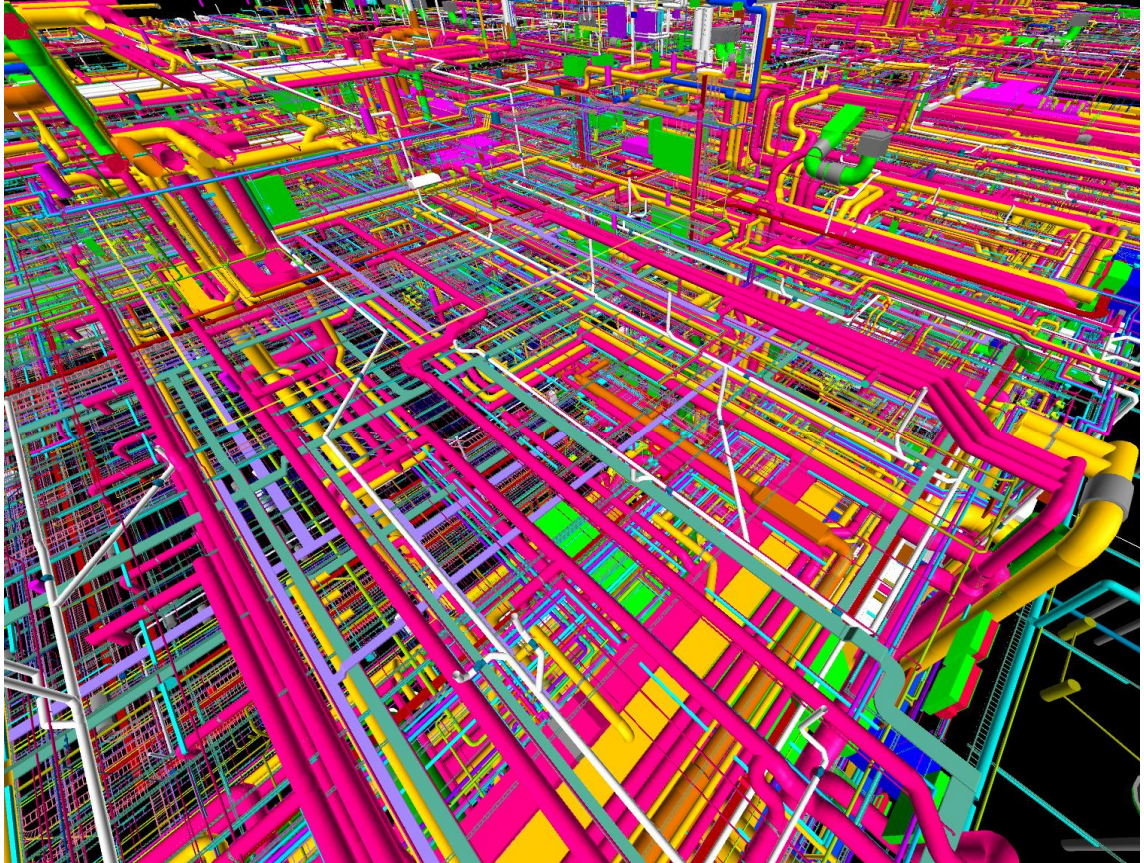
3.1 ”Näytä osa Naviksessa”

”Näytä osa Naviksessa” on AutoCADiin ladattava lisätoiminto, jolla käyttäjä voi siirtyä NavisWorksin 3D-mallissa kohtaan, joka on MagiCADissa valittu, esim. venttiilin tai valaisimen luo. Kuva 3 esittää, miltä toiminto käytännössä näyttää tietokoneen ruudulla (valitaan hiirellä napsauttamalla jokin MagiCAD-osa ja hiiren oikealla napsauttaessa avautuvasta valikosta valitaan ”Näytä osa Naviksessa”).



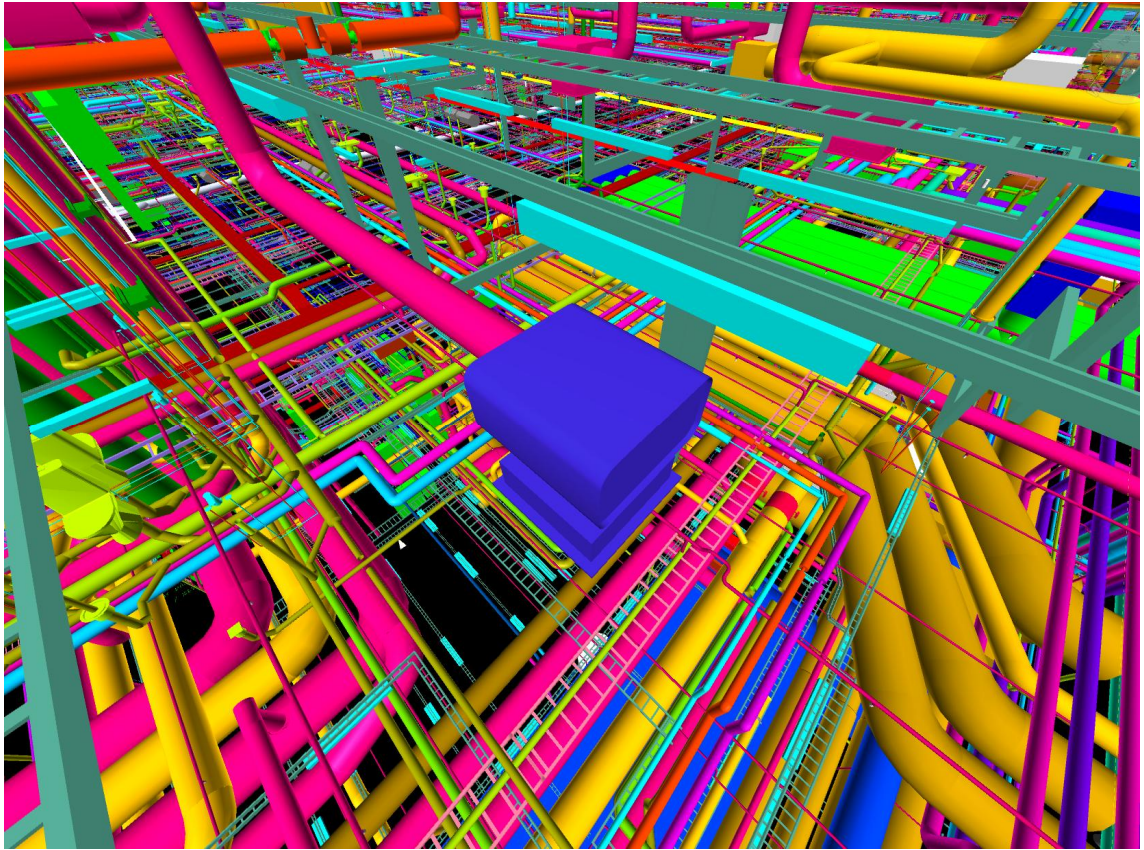
Kuva 3. ”Näytä osa Naviksessa”-toiminto Magicadissa.

Kuva 4 puolestaan näyttää esimerkin näkymästä tietokoneen sillä ruudulla, jossa on NavisWorks auki, ennen "Näytä osa Naviksessa"-toiminnon suorittamista. Oikean kohdan luo navigointi manuaalisesti olisi aina jonkinasteinen ylimääräinen työ ja ajanhukka.



Kuva 4. Esimerkki Navisworksin näkymästä ennen "Näytä osa Naviksessa" toiminnon suorittamista.

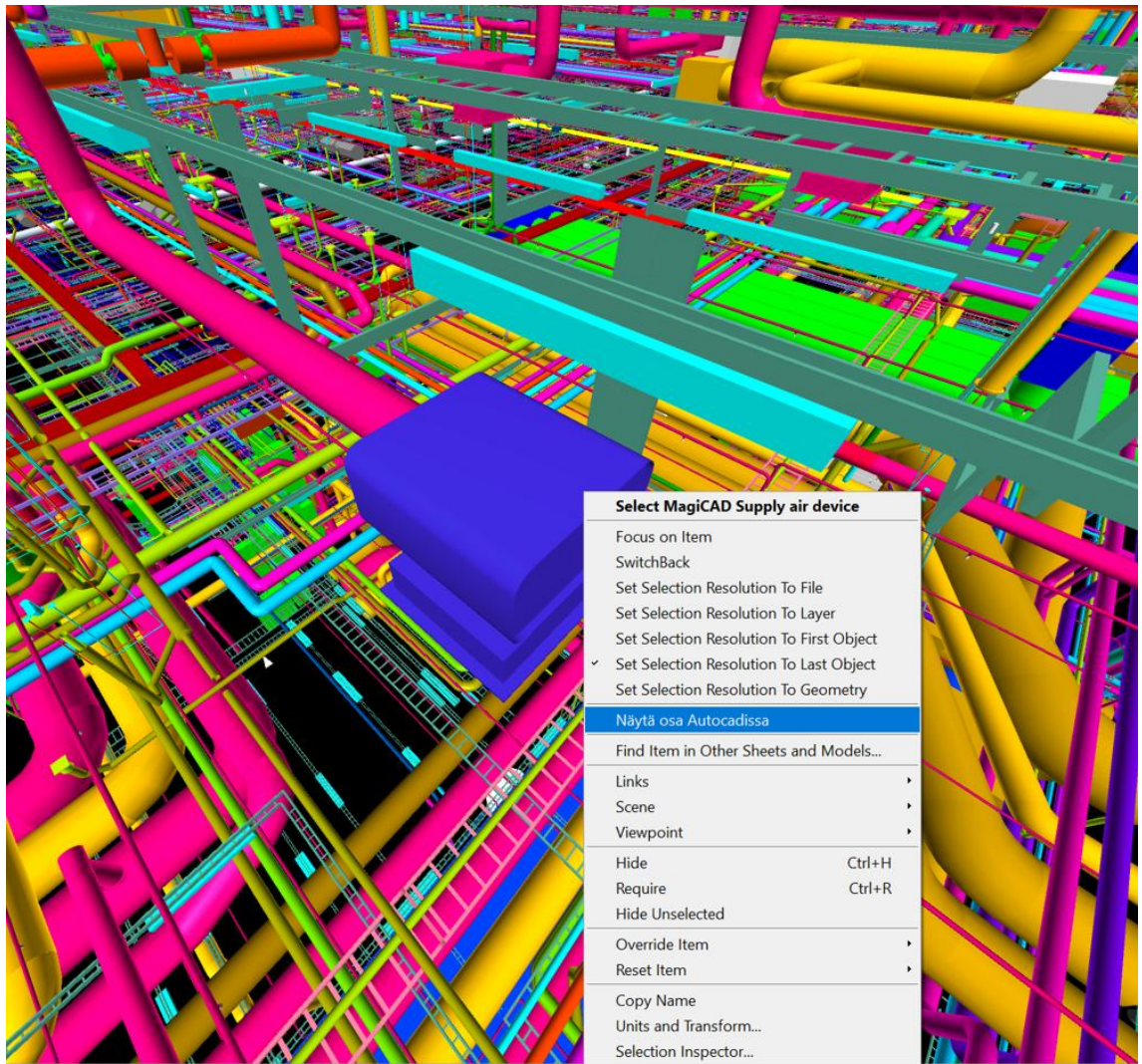
Kuva 5 näyttää, miten näkymä tietokoneen NavisWorks-ruudulla välittömästi muuttuu "Näytä osa Naviksessa"-toiminnon suorittamisen jälkeen. Magicadissa valittu osa näkyy keskellä sinisenä, "neula löytyi heinäsuovasta".



Kuva 5. NavisWorks:n näkymä "Näytä osa Naviksessa" toiminnon suorittamisen jälkeen.

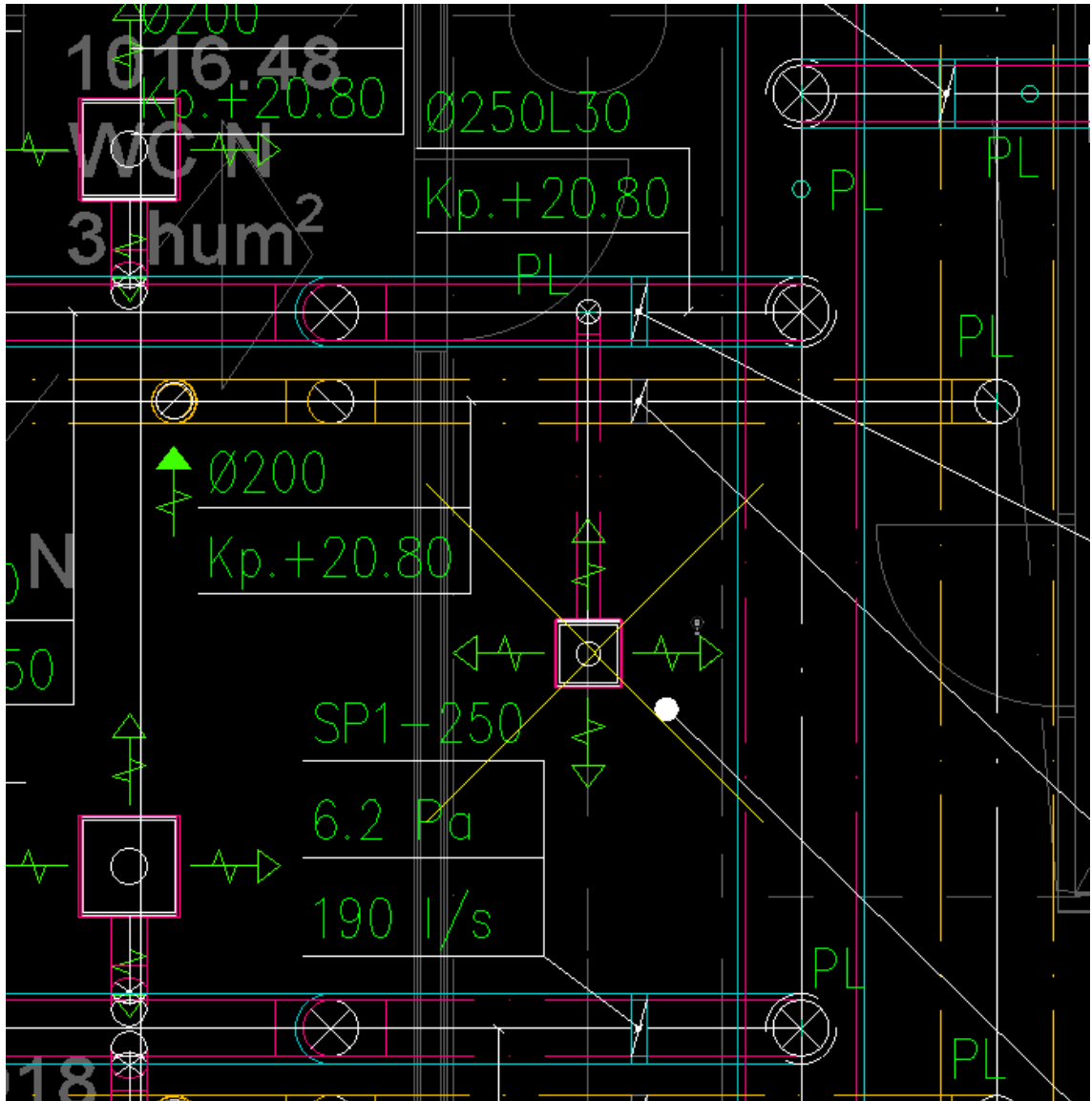
3.2 "Näytä osa Autocadissa"

"Näytä osa Autocadissa" on NavisWorksiin ladattava lisätoiminto ja toimii vastaavasti kuin "Näytä osa Naviksessa", mutta juuri päinvastaiseen suuntaan. Toiminto myös avaa AutoCADiin kyseisen piirustuksen/DWG-tiedoston, jos se ei ole valmiiksi auki. Jos koko AutoCAD ei ole valmiiksi auki, toiminto käynnistää ensin sen. Kuva 6 esittää, millä toiminto näyttää tietokoneen ruudulla/NavisWorksissa.



Kuva 6. "Näytä osa Autocadissa"-toiminto Navisworksissa.

Kuva 7 näyttää "Näytä osa Autocadissa"-toiminnon seurauksen: AutoCAD zoomaa NavisWorksissa valittuun osaan ja piirtää vielä selvyyden vuoksi rastin kyseisen osan kohdalle.

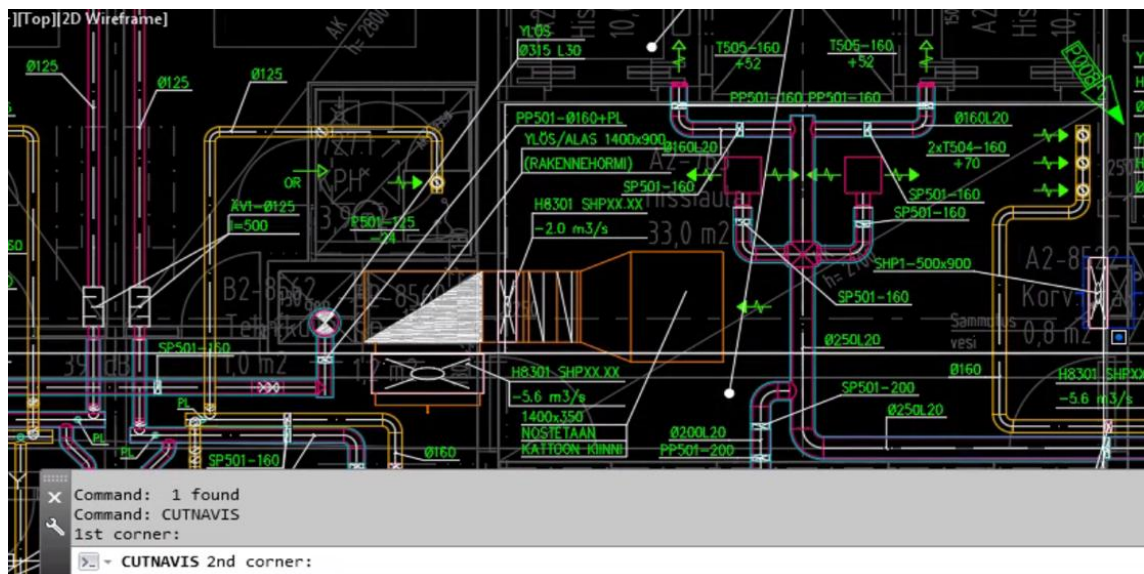


Kuva 7. "Näytä osa Autocadissa"-toiminto AutoCADissa

3.3 "CUTNAVIS"

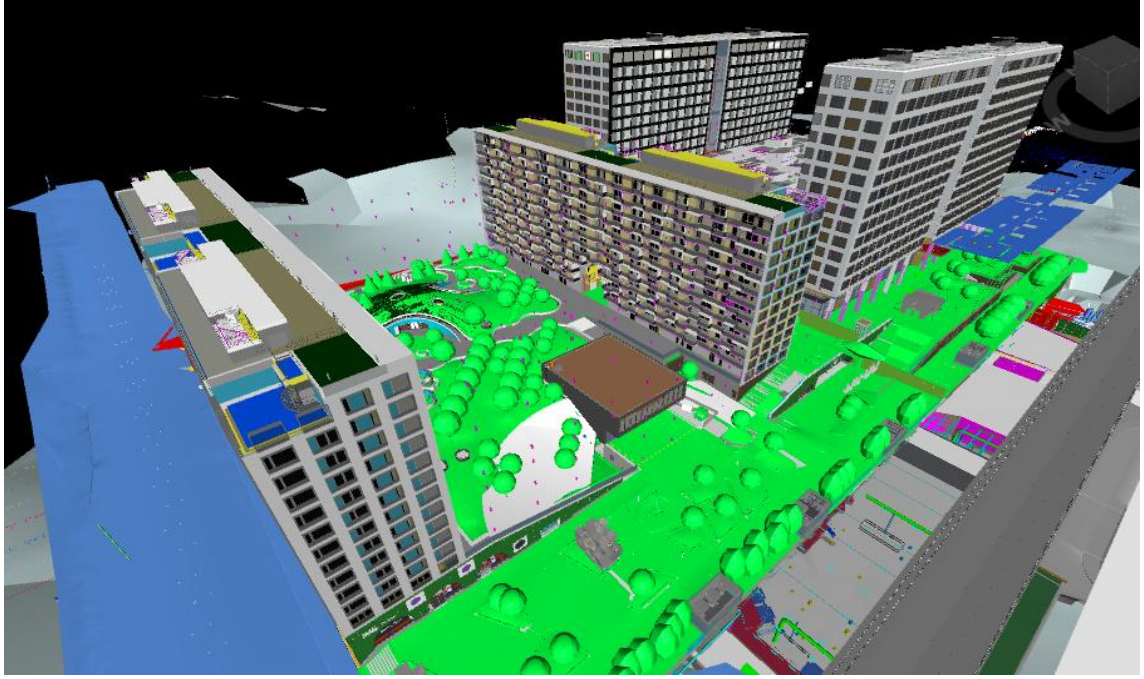
NavisWorksissa eräs oleellinen ominaisuus on se, että 3D-mallista voidaan leikata vain haluttu siivu näkyviin käyttäen maksimissaan kuutta leikkaustasoa samaan aikaan. AutoCAD-lisätyökalu/komento "CUTNAVIS" toimii siten, että se pyytää näyttämään ensin kaksi nurkkapistettä. Yleensä komentoa suoritettaessa ruudulla on näkyvillä LVI-

tai sähkösuunnitelma, jonka taustalla on arkkitehtipohja, joten käyttäjä napsauttaa hiirellä esim. jonkin huoneen kahta vastakkaista nurkkaa. Kuva 8 näyttää esimerkin toiminnon suorittamisesta AutoCADissa. Ohjelma hakee lisäksi aktiiviseen DWG-kuvaan liittyvästä MagiCAD-projektista kyseisen kerrostason lattian absoluuttisen korkeusase- man sekä kerroksen korkeuden. Edellä mainittujen tietojen perusteella ohjelma leikkaa sillä hetkellä auki olevan NavisWorks-mallin ko. kohdasta.



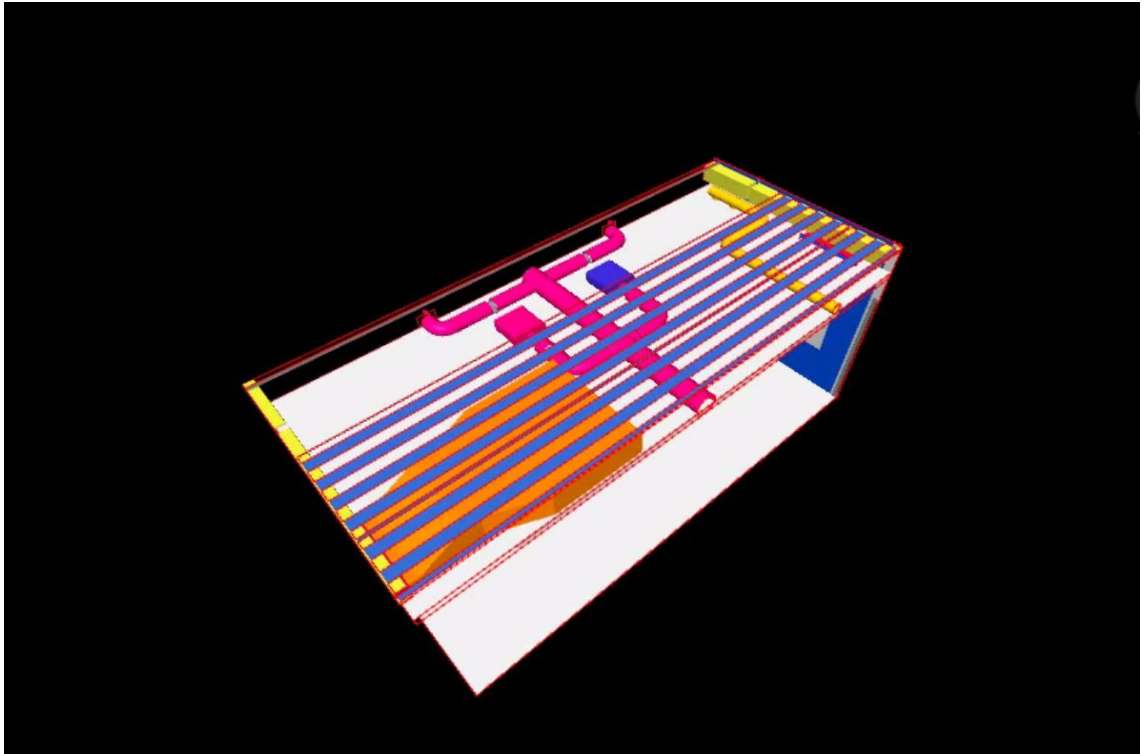
Kuva 8. Komento "CUTNAVIS" AutoCADissa.

Kuva 9 on esimerkki Navisworksin näkymästä ennen "CUTNAVIS"-toiminnon suorittamista. Tässä tapauksessa koko rakennus on näkyvässä. Hyvin usein halutaan esim. tarkastella jonkin tietyn tilan katossa sijaitsevia talotekniikka-asennuksia. Näkymän säätäminen sopivaksi tällaista tarkastelua varten "käsini" olisi aina jonkinasteinen ylimääräinen työ ja ajan haaskaamista.



Kuva 9. Esimerkki NavisWorksin näkymästä ennen "CUTNAVIS" toiminnon suorittamista.

Kuva 10 esittää NavisWorks:n näkymää "CUTNAVIS" toiminnon suorittamisen jälkeen. Katossa näkyvät siniset soirot ovat ontelolaattaa. Oletusarvoisesti ohjelma tekee yläpuolen leikkauksen 400mm seuraavan kerroksen lattiatason alapuolelle. Tällä pyritään siihen, että kyseistä välipohjalaattaa ei näkyisi, mutta käsiteltävän kerroksen katossa sijaitseva talotekniikka tulisi mahdollisimman selvästi näkyviin. Jos välipohjalaatta on oletusarvoa paksumpi, se jää osittain näkyviin.

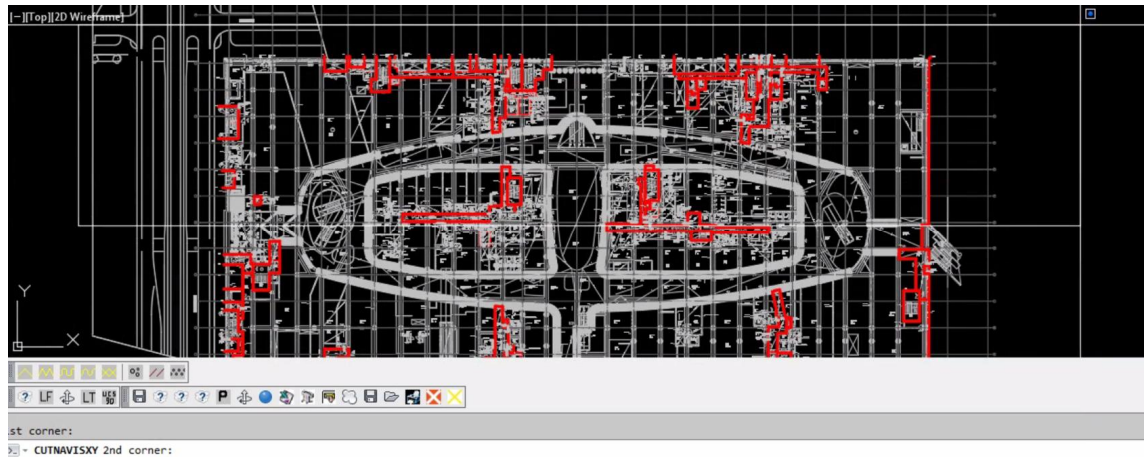


Kuva 10. NavisWorks:n näkymä "CUTNAVIS" toiminnon suorittamisen jälkeen.

3.4 "CUTNAVISXY"

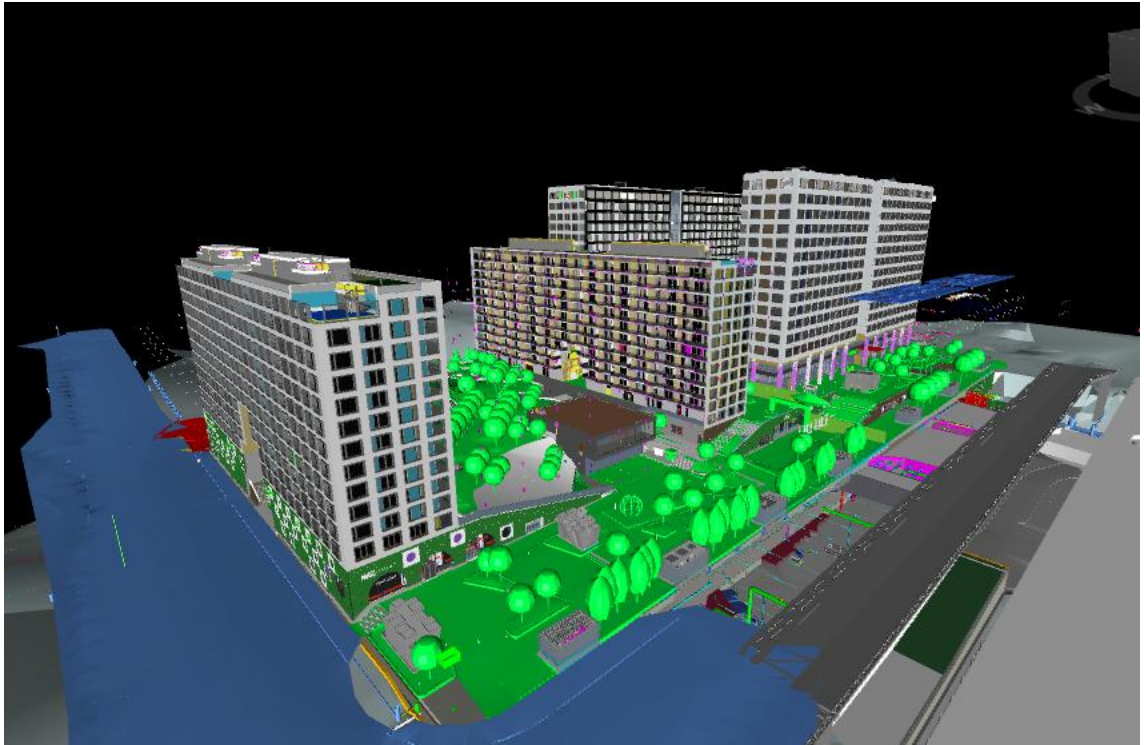
"CUTNAVISXY"-toiminto on muuten sama kuin edellä selostettu "CUTNAVIS", mutta tässä mallia leikataan vain neljällä leikkaustasolla X ja Y / sivusuunnista, eli kerroskorkeutta ei oteta huomioon. Toiminto sopii hyvin esim. NavisWorks-mallin leikkaamiseen joltakin moduulilinjalta koko rakennuksen osalta tai esim. jonkin tekniikkakuilun näyttämiseen helposti kaikkien kerrosten osalta.

Kuva 11 esittää "CUTNAVISXY"-toimintoa Autocadissa. Tässä tapauksessa aktiivinen piirustus on arkkitehtipohja. Toiminto ei tarvitse tietoa kerroskorkeudesta eikä niin ollen MagiCADIakaan. Tässä esimerkissä toiminto on suoritettu niin, että leikkaustoiminto suoritettiin pohjapiirustuksen "yläpuoliskolle" (tämä ilmenee kuvassa siten, että "yläpuolisko" ympäröi suorakaide).



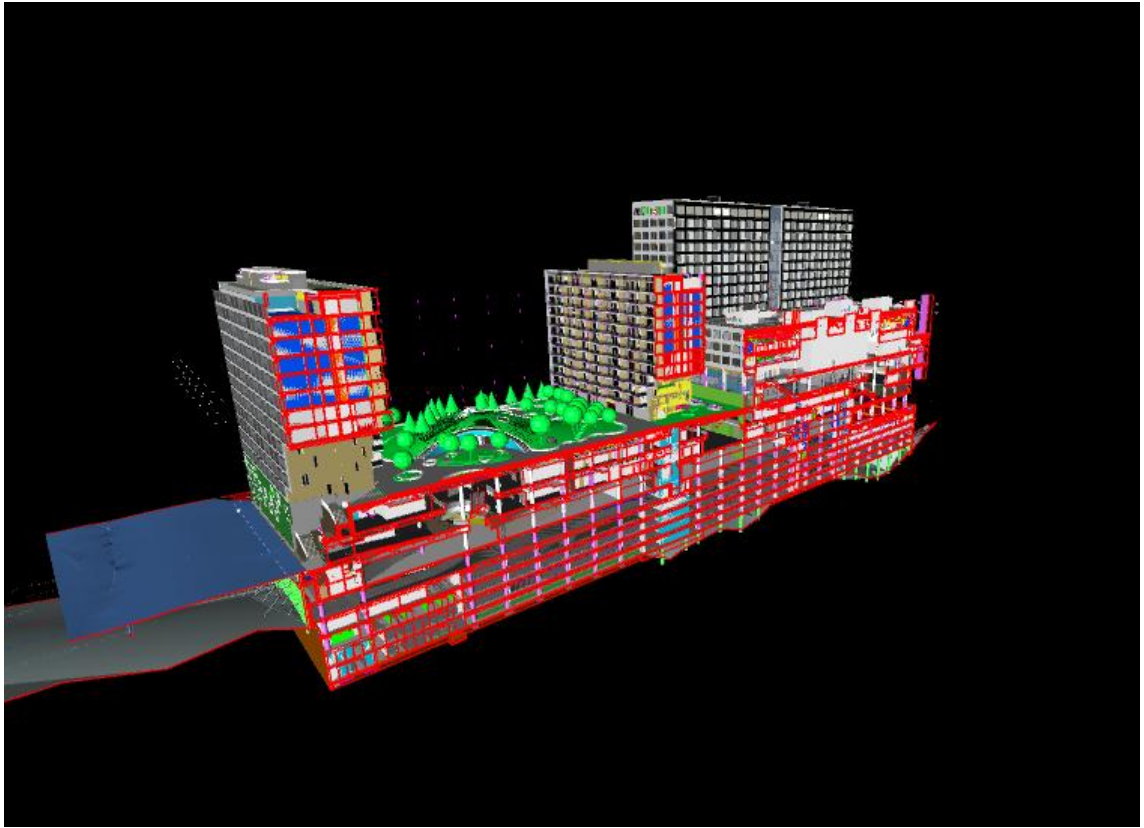
Kuva 11. "CUTNAVISXY"-toiminto Autocadissa.

Kuva 12 on esimerkki NavisWorks:n näkymästä ennen "CUTNAVISXY"-toiminnon suorittamista. Koko rakennus on näkyvässä. Leikkauspintojen luonti NavisWorks:n toiminnolla "käsien" haluttuun kohtaan pohjapiirustusta olisi aina jonkin verran aikaa vievä ylimääräinen työ.



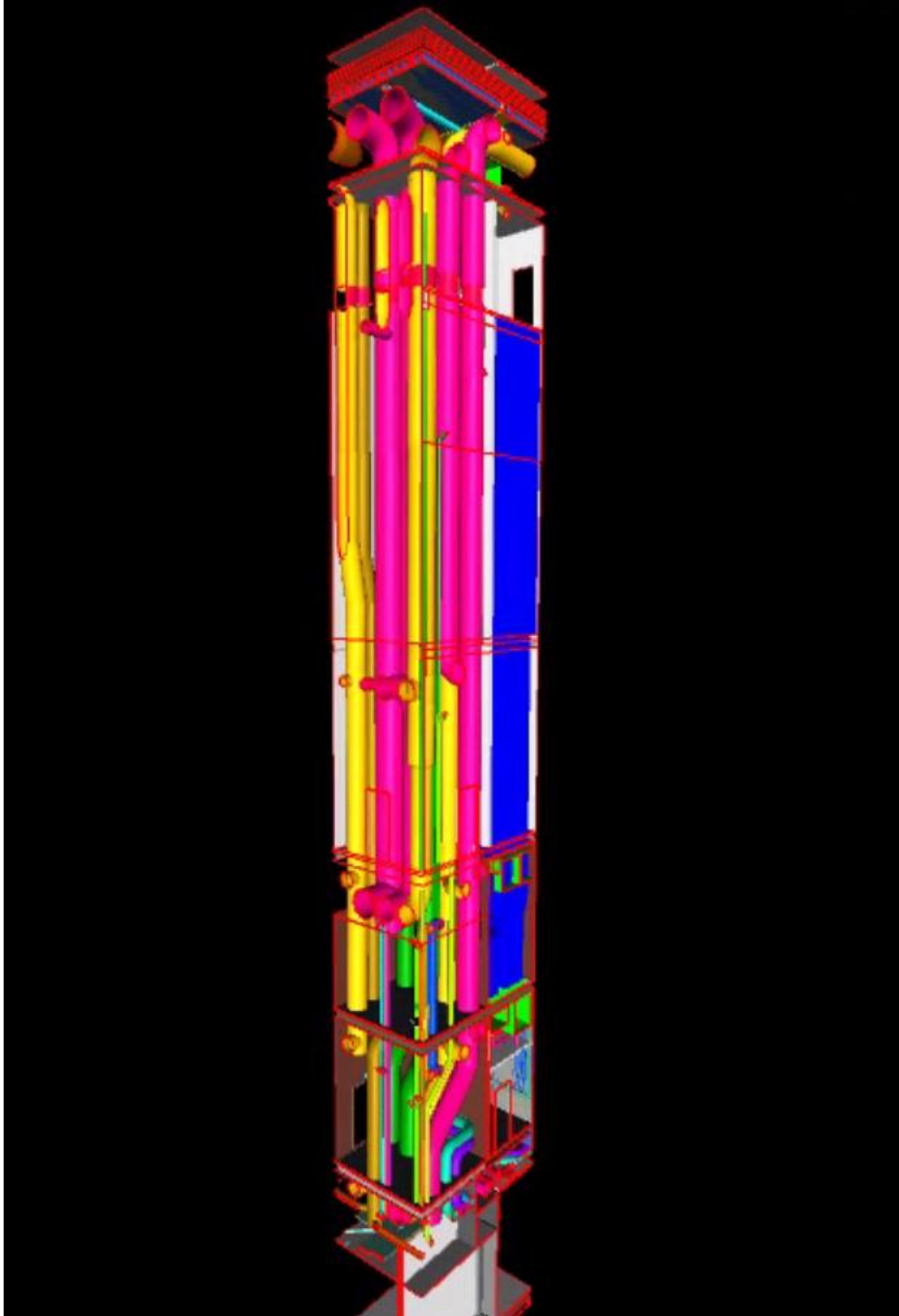
Kuva 12. Esimerkki Navisworks:n näkymästä ennen "CUTNAVISXY"-toiminnon suorittamista.

Kuva 13 on esimerkki, miten NavisWorks:n näkymä muuttuu "CUTNAVISXY"-toiminnon suorittamisen jälkeen. Toiminto luo aina neljä pystysuuntaista leikkauspintaa, mutta tässä tapauksessa kolme pintaa sijaitsee rakennuksen ulkopuolella (vrt. kuva 11), mistä seuraa, että näytöllä nähdään vain rakennuksen keskikohdalla oleva leikkaus.



Kuva 13. NavisWorks:n näkymä "CUTNAVISXY"-toiminnon suorittamisen jälkeen.

Kuva 14 esittää NavisWorks-näkymää, kun "CUTNAVISXY"-toiminnolla mallista on leikattu esiin LVI-kuilu. Tällaiset tarkastelut ovat usein hyödyllisiä, mutta NavisWorksin toiminnoilla näiden neljän leikkauspinnan luonti "käsini" oikeaan kohtaan mallia on hidasta ja niin ollen turhaa ajanhukkaa.



Kuva 14. "CUTNAVISXY"-toiminnolla mallista esiin leikattu LVI-kuilu.

4 Työn toteutus

Edellisessä luvussa esitettiin, mitkä olivat laadittavien apuohjelmien toiminnallisuuden tavoitteet. Sen selvittämiseksi, onko kyseisten kuvausten mukaisia ohjelmia ylipäätään mahdollista toteuttaa järkevällä työmäärällä, oli ensin selvittävä muutamia perusasioita.

4.1 Ohjelmointirajapinnat

Ohjelmointirajapinta (engl. Application programming interface, API) on määritelmä, jonka mukaan eri ohjelmat voivat tehdä pyyntöjä ja vaihtaa tietoja eli keskustella keskenään (4). Työn tavoitteeksi asetettujen ohjelmien toteuttamismahdollisuuksia tarkasteltaessa oli ensimmäiseksi selvittävä, onko kaikilla asiaan liittyvillä ohjelmistoilla (AutoCAD, MagiCAD, NavisWorks) ylipäätään olemassa jonkinlainen ohjelmointirajapinta.

NavisWorks

NavisWorksissa on v. 2011 lähtien ollut käytettävissä .NET API. Jo tätä ennen NavisWorks sisälsi COM API:n, joka edelleen on joidenkin toimintojen osalta ainoa tie saada toimenpide suoritettua. Valmistaja kuitenkin suosittelee uusien sovellusten käyttävän ensisijaisesti .NET APIa (5) ja niin tein myös tähän työssä laadittujen sovellusten kanssa. .NET API:n kautta päästään käyttämään myös kaikkia COM API:n toimintoja.

AutoCAD

AutoCAD on varustettu erittäin runsailla ja monipuolisilla ohjelmointi-, räätälöinti- ja laajennusmahdollisuuksilla. Lukemattomat tahot ovatkin aikojen saatossa laatineet AutoCADin eri ohjelmointirajapintojen kanssa kommunikoivia sovelluksia. Jotkut näistä sovelluksista ovat suuria kokonaisuuksia, näistä hyvänä esimerkkinä MagiCAD.

AutoCADin eri ohjelmointirajapintoja ovat (6)

- ObjectARX
- .NET API
- Visual LISP

- ActiveX

Tämän työn yhteydessä päädyin käyttämään .NET APIa tietäen sen käyttömahdollisuuksiltaan hyvin laajaksi, jolloin se varmasti ”riittäisi” näiden yksinkertaisten ohjelmien luontiin. NavisWorksin osalta käytännössä ainoa järkevä vaihtoehto on .NETin käyttö, jolloin on luontevaa saman projektin yhteydessä käyttää myös AutoCADin puolella .NETiä.

MagiCAD

MagiCAD sisältää COM-ohjelmointirajapinnan, jonka toiminnot ovat melko rajalliset. Tämän työn puitteissa ei MagiCADin kanssa kovin suurta ohjelmallista tiedonvaihtotarvetta ollutkaan, joten COM-liitännän kanssa pärjättiin kyllä. MagiCADin COM-liitännän käyttöä varten tarvitaan aiheesta laadittu dokumentaatio, joka ei tule MagiCADin perusasennuksen mukana.

4.2 Ohjelmointityökalut

Viimeistään siinä vaiheessa, kun varmistui, että työssä pitää käsitellä .NET-asioita, oli selvää, että ohjelmien kirjoittamisessa tulisin käyttämään Microsoft Visual Studiota. Visual Studio on maailman käytetyimpiä ohjelmakehitysympäristöjä. Siinä on runsaasti ominaisuuksia, jotka mm. helpottavat ja nopeuttavat ohjelmakoodin kirjoittamista.

Pääasialliset ohjelmointikielet kehitettäessä .NET ohjelmia Windows-ympäristöön ovat C# ja Visual Basic .NET. Tämän työn yhteydessä laaditut ohjelmat tein C# -kielellä.

4.3 Ohjelmien rakenneperiaatteita

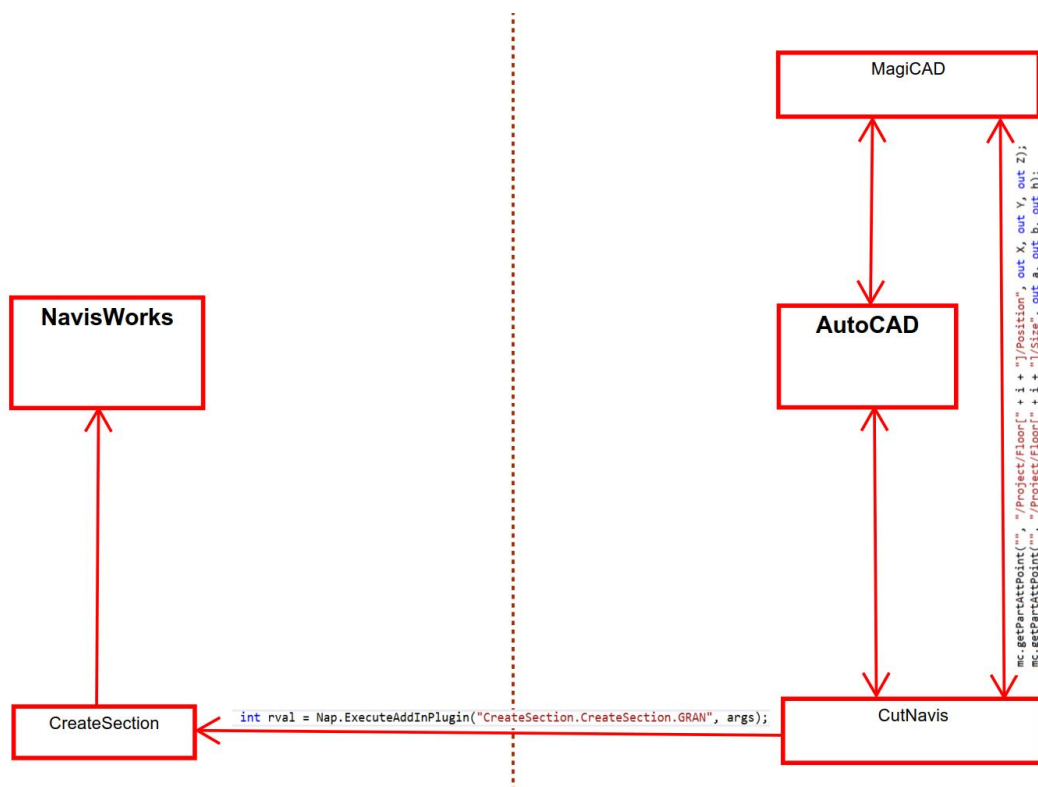
Tietokoneohjelmien laatimisessa, kuten monessa muussakin tehtävässä, kannattaa noudattaa seuraavia pääperiaatteita

- Asiat kannattaa pilkkoa selkeisiin ja helposti hallittaviin kokonaisuuksiin.

- Pilkkominen kannattaa tehdä mahdollisuuksien mukaan niin, että palanen on myöhemmin hyödynnettävissä myös muuhun tarkoitukseen kuin juuri nyt työn alla olevaan.

Kuvissa 15 ja 16 on pari esimerkkiä pilkkomisesta tässä tapauksessa.

CUTNAVIS

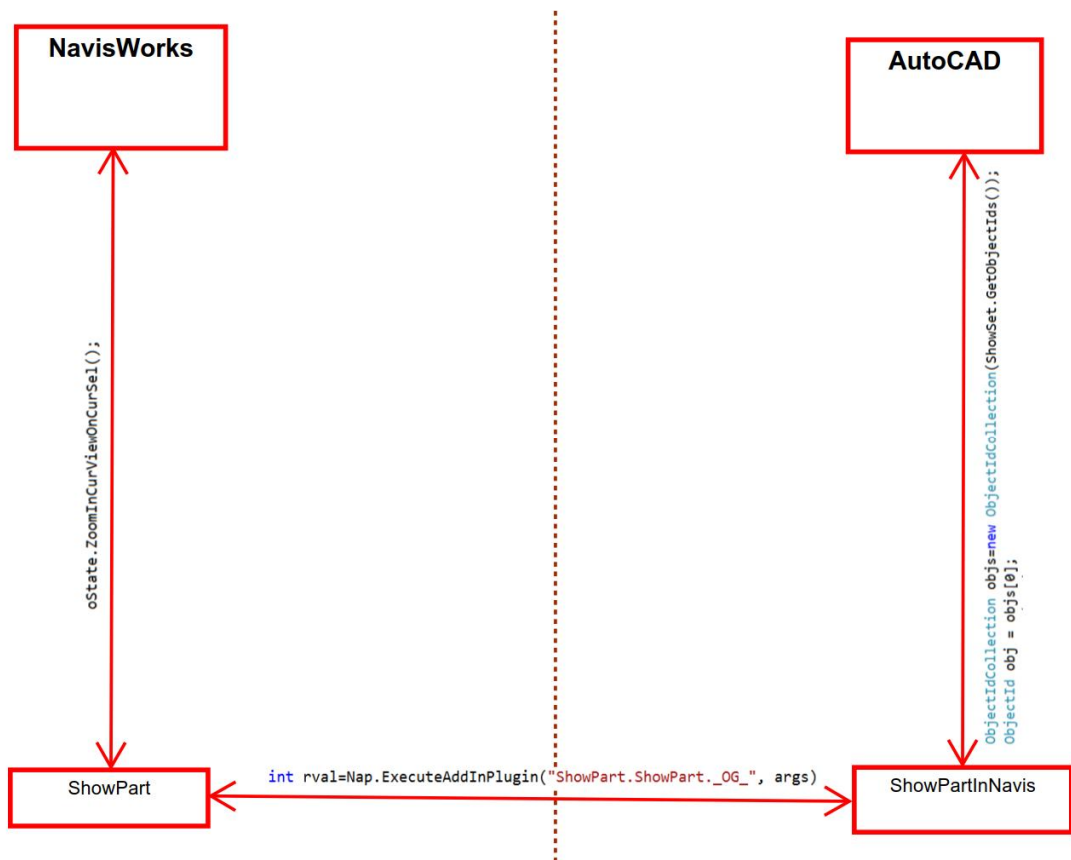


Kuva 15. CUTNAVIS-toiminnon periaatteellinen rakenne.

Kuvan 15 esimerkissä CutNavis – apuohjelma huolehtii lähtötietojen saannista AutoCADista ja MagiCADista. CutNavis muodostaa näistä lähtöarvoista sitten parametrit, joiden kera kutsuu NavisWorksin muistiavaruudessa majoilevaa erillistä CreateSection-apuohjelmaa. CreateSection tekee saamiensa parametrien perusteella leikkaukset NavisWorksiin.

CreateSection ei siis ”tiedä”, mikä ohjelma sitä kutsuu, eikä tarvitse sitä tietoa mihinkään. Näin ollen CreateSection-apuohjelmaa voivat hyödyntää tarvittaessa muutkin ohjelmat kuin CutNavis.

”Näytä osa Naviksessa”



Kuva 16. ”Näytä osa Naviksessa”-toiminnon periaatteellinen rakenne.

Myös kuvan 16 esimerkissä toiminto on rakennettu kahden lisäohjelman kautta, toinen ladataan AutoCADiin ja toinen NavisWorksiin. NavisWorks-apuohjelmaa ShowPart voi tarvittaessa hyödyntää mikä tahansa muukin ohjelma kuin ShowPartInNavis.

4.4 Ohjelmien toteutus

Kaikkien tässä yhteydessä tehtyjen ohjelmien suoritettava muoto on .DLL-tiedosto. AutoCADin osalta nämä ladataan käyttöön NETLOAD-komennolla. Granlundin ympäristössä tämä on käynnistyskriptillä määritelty tapahtumaan automaattisesti. DLL-tiedostojen lataamisen jälkeen ko. apuohjelmien komennot ovat käytettävissä kuin mikä tahansa AutoCAD-komennot.

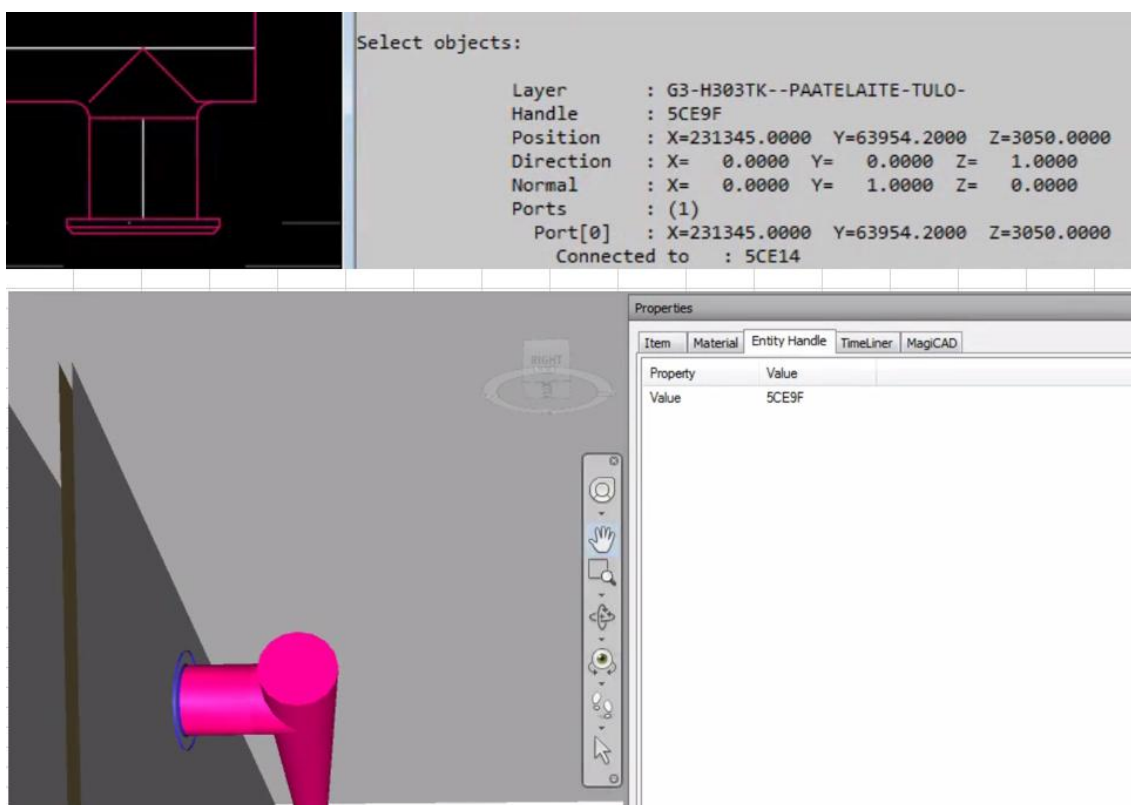
NavisWorksin osalta apuohjelmat (NavisWorksin dokumentaatioissa näiden nimitys on ”Plugin”) tallennetaan ohjelmahakemiston ”Plugins”-alihakemistoon, jotta NavisWorks

osaisi automaattisesti ladata ne käytettäviksi. Kukin apuohjelma tallennetaan .DLL-tiedoston nimeä vastaavaan alihakemistoon. Esimerkki apuohjelman sijainnista on siis C:\Program Files\Autodesk\Navisworks Simulate 2018\Plugins\ShowPart\ShowPart.dll

4.4.1 ”Näytä osa Naviksessa”

Jotta tavoitteenasettelun mukainen toiminto olisi mahdollinen, on oltava joku tapa löytää AutoCADissa valittu osa, esim. tuloilmalaite, NavisWorksista. Tämän vuoksi oli tarkasteltava, minkälaista tietoa talotekniset osat NavisWorksissa sisältävät ja riittävätkö ko. tiedot osan varmaan identifiointiin.

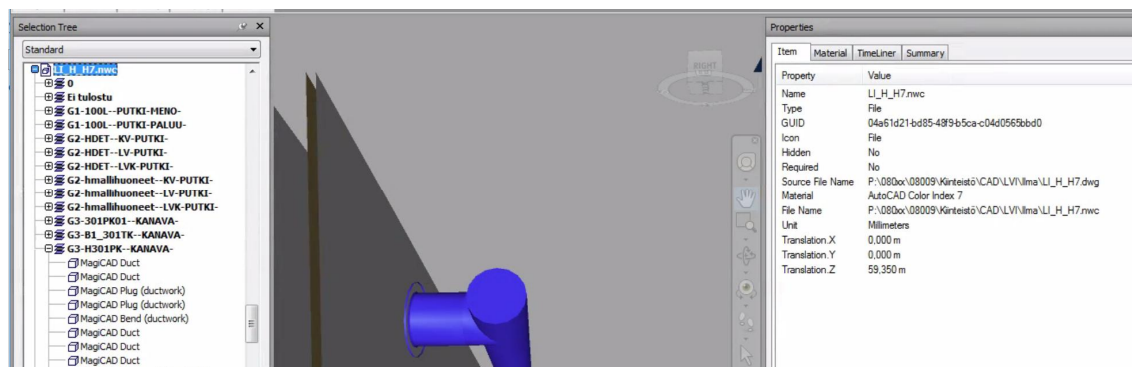
AutoCADissa jokainen osa sisältää yksilöllisen ”Handle”-tunnisteen. Tämä tieto siirtyy myös NavisWorksiin (kuva 17).



Kuva 17. Handle-tunniste AutoCADissa ja NavisWorksissa.

Handle-tunniste on kuitenkin uniikki vain kyseisen DWG:n sisällä. Useimmiten NavisWorksin yhdistelmämallissa olevat osat ovat lähtöisin lukuisista eri DWG:eistä. Tämän vuoksi tarvittiin myös keino selvittää, mistä DWG:stä mikäkin NavisWorksissa oleva

osa on lähtöisin. Tämä tieto onkin saatavilla NavisWorks:n puumaista tietorakennetta hyödyntäen (kuva 18).



Kuva 18. DWG-tiedoston tunniste, "Source File Name".

Ominaisuustieto "Source File Name" kertoo DWG-tiedoston nimen polkuineen eli yksilöi sen tarkasti.

"Näytä osa Naviksessa"-ohjelman suoritus etenee seuraavasti

1. Kun ko. DLL on Autocadiin ladattu, voidaan osan ollessa valittuna hiiren oikealla avautuvasta valikosta valita "Näytä osa Naviksessa".
2. Ohjelma tarkistaa ensin, että valittuna on vain yksi osa (koko homman idea on löytää yhden valitun osan vastapari NavisWorksistä).
3. Ohjelma on rajattu toimimaan vain MagiCAD-osien kanssa, joten se ensin tarkistaa onko valittu osa MagiCAD-osa.
4. Ohjelma ottaa yhteyden auki olevaan NavisWorks-ohjelmaan. Jotta ohjelma toimisi, NavisWorks:n ja asiaankuuluvan mallin on oltava auki. Tämä sen vuoksi, että AutoCADista/MagiCADista ei ole saatavilla tietoa, mikä NavisWorks-tiedosto olisi kulloinkin "Näytä osa Naviksessa" -komentoa suoritettaessa se oikea tiedosto.
5. Ohjelma kutsuu NavisWorks-pluginia "ShowPart" ja lähettää sille parametreina valitun osan Handle-tunnisteen sekä aktiivisen DWG:n nimen polkuineen.

6. ShowPart-plugin etsii ensin NavisWorksin yhdistelmämallista parametrina saamaansa tiedostonimeä vastaavan osamallin.
7. ShowPart-plugin hakee löytämästään osamallista parametrina saamaansa Handle-tunnusta vastaavan osan. Osan löydyttyä ohjelma asettaa sen valituksi.
8. NavisWorksin (COM) APIsta löytyy toiminto, jolla saadaan zoomattua suoraan osaan, joka on sillä hetkellä valittuna. Tämä onkin tähän tarkoitukseen erittäin hyvin sopiva toiminto, ja oikea osa tulee välittömästi keskelle näyttöä. Zoomauksen aikaansaava koodi on seuraavanlainen:

```
ComApi.InwOpState10 oState = ComApiBridge.ComApiBridge.State;
oState.ZoomInCurViewOnCurSel();
```

4.4.2 "Näytä osa Autocadissa"

Myös "Näytä osa Autocadissa"-toiminnossa käytetään osien identifiointiin samoja tietoja kuin edellä mainitussa, eli DWG-tiedostonimi polkuineen ja Handle-tunniste. Ohjelman suoritus etenee seuraavasti:

1. Komento "Näytä osa Autocadissa" näkyy hiiren oikealla napsauttaessa, koska ohjelmakoodissa on seuraava määrittely.

```
[PluginAttribute("Näytä osa Autocadissa", "_OG_")]
[AddInPluginAttribute(AddInLocation.CurrentSelectionContextMenu)]
```

2. Ensin ohjelma tarkistaa, onko valittuna vain yksi osa.
3. Ohjelma kulkee valitusta osasta lähtien tietorakennepuun juureen ja hakee sieltä "Source File Name"-ominaisuuden.
4. Ohjelma tarkistaa, onko "Source File Name" .DWG -päätteinen, ja onko ko. tiedostoa olemassa.
5. Ohjelma hakee valitun osan Handle-tiedon.

6. Ohjelma ottaa yhteyden AutoCADiin. Tässä yhteydessä ohjelma käynnistää AutoCADin jos se ei ole valmiiksi auki.
7. Ohjelma hakee AutoCADissa auki olevien dokumenttien joukosta valittua osaa vastaavaa DWG:tä. Jos sitä ei löydy, ohjelma avaa sen ja kysyy käyttäjältä avataanko se "Read-Only" -tilassa.
8. Ohjelma tarkistaa, onko valitun osan Handle-tunnusta vastaavaa osaa Autocadissa olemassa.
9. Jotta osaan zoomaus onnistuisi, täytyy AutoCADin olla ns. Model Spacessa, ei Paper Spacessa. Tämä on varmistettu seuraavalla koodilla:

```
TheDoc.SetVariable("TILEMODE", 1);
```

10. Ohjelma zoomaa Handle-tiedon perusteella osaan. Handle-tietoa kannattaa käyttää siksi, että sekä AutoCADissa että NavisWorksissa koordinaatistoa voidaan siirtää mitä ihmeellisimmillä tavoilla. Handle-tietoa käytettäessä apuohjelman ei tarvitse huolehtia siitä, mihin kohtaan koordinaatistoa ollaan zoomaamassa. Handle-tiedon perusteella saadaan zoomattua osaan seuraavasti:

```
TheDoc.SendCommand("_ZOOM " + "O " + "(HANDENT " + "\"" + HandleToZoom + "\"" + " )");
```

11. Koska em. komento zoomaa usein turhan lähelle, ohjelma vielä säätää zoomausta sopivammaksi ja piirtää sitten osan päälle ruksin. Ruksi piirretään AutoCAD-layerille "MAGI_GARBAGE". Tämä on MagiCADin käyttämä erikois-layer, johon piirretään nimenomaan tällaiset väliaikaiset apumerkinnot, ja merkinnot saadaan helposti poistettua MagiCADin toiminnolla.

4.4.3 CUTNAVIS

CUTNAVIS-ohjelman ei ole tarpeen tunnistaa, mikä objekti/osa tai tiedosto NavisWorksissa vastaa mitään AutoCADissa/MagiCADissa. Lähtöoletta on, että käyttäjä on valmiiksi avannut sen NavisWorks-mallin, johon hän haluaa leikkauksia luoda tämän

toiminnon avulla. Ohjelma tekee leikkaukset sillä hetkellä auki olevaan malliin tutkittua, vastaako malli kyseistä DWG:tä. Ohjelman suoritus etenee seuraavasti:

1. Autocadissa annetaan komento CUTNAVIS. Yleisesti AutoCADin käyttöliittymää on helppo räätälöidä ja komentojen kätevämpää suorittamista varten niille tehdään usein oma painike ja / tai kohta esim. ctrl + hiiren oikealla aukeavaan pikavalikkoon-
2. Ohjelma pyytää näyttämään kaksi kulmapistettä ja ottaa muistiin ko. koordinaatitipisteet.
3. Ohjelma ottaa yhteyden MagiCADin COM-liitäntään, tämä tapahtuu seuraavalla koodilla:

```
MagiCAD mc = (Magi-
CAD) System.Runtime.InteropServices.Marshal.GetActiveObject("MagiComSrv.M
agiCAD");
```

4. Ohjelma pyytää MagiCADilta tiedon sen hetkistä aktiivista DWG:tä vastaavan kerroksen lattian +korosta sekä kerroskorkeudesta. Nämä ovat ne z- ja h- tiedot, jotka syötetään MagiCADin projektitietoihin (kuva 19). Käytäntönä on pidetty, että h tarkoittaa etäisyyttä lattiapinnasta lattiapintaan.

Name	x	y	z	a	b	h
Katto	0	0	92650	500000	300000	2950
H15	0	0	88150	500000	300000	4500
H14	0	0	84550	500000	300000	3600
H13	0	0	80950	500000	300000	2600

Kuva 19. Kerrostiedot MagiCAD-projektissa.

5. Nyt ohjelmalla on tieto laatikkomaisesta tilavuudesta (em. kulmapisteet, lattia-korko sekä kerroskorko vähennettynä oletusarvoisella välipohjan paksuudella). Ohjelma ottaa yhteyden auki olevaan NavisWorksiin ja kutsuu CreateSection-pluginia lähettäen sille parametreina em. geometriatiedot. Lisäparametrina annetaan leikkaustasojen lukumäärä, joka tässä tapauksessa on 6. Tämän parametrin käytöllä saadaan CreateSectionin käyttömahdollisuudet monipuolisemmiksi.

6. CreateSection sisältää funktion MakeClipPlane. Tätä funktiota kutsutaan tässä tapauksessa kuusi kertaa ja joka kerralla NavisWorksin mallia leikataan eri suunnasta. Lopputuloksena on laatikkomainen leikattu siivu mallista, AutoCADissa valitusta alueesta riippuen esim. yksi huone.

4.4.4 CUTNAVISXY

CUTNAVISXY-ohjelman rakenne on hyvin pitkälti sama kuin CUTNAVIS-ohjelman. Molemmat myös latautuvat AutoCADiin yhden ja saman DLL:n kautta, vaikka ovatkin siis kaksi eri komentoa. CUTNAVISXY ei ota yhteyttä MagiCADiin ja kutsuu NavisWorks Pluginia "CreateSection" lähettämällä leikkausten lukumäärä-parametrina 4 (leikkaukset vain sivusuunnista). CUTNAVISXY-komentoa voi tarvittaessa käyttää pelkän AutoCADin ja NavisWorksin kanssa, MagiCADia ei sen puolesta tarvitse olla edes asennettuna.

4.5 Ohjelmien päivitys

AutoCAD- ja NavisWorks-päivitysten yhteydessä pitää myös yllä mainitut apuohjelmat päivittää, mutta tämä tarkoittaa käytännössä vain ohjelmakoodin avaamista Visual Studioon, AutoCAD- ja NavisWorks-tiedostoviittausten päivitystä ja uusien DLL:ien "tulostamista". "Target Framework"-asetuksen voi joutua muuttamaan, jos AutoCAD tai NavisWorks ovat versiopäivityksensä yhteydessä siirtyneet uudemman .NET Frameworkin käyttöön.

5 Mahdollisia kehityskohteita

Tätä kirjoittaessa, alkuvuodesta 2018, yhdistelmä MagiCAD for AutoCAD + NavisWorks on edelleen paras vaihtoehto suurten kohteiden talotekniikan tietomallinnukseen. Tuloaan tekee kuitenkin nimenomaan rakennusten tietomallinnusta varten kehitetty ohjelmisto nimeltä Revit (sekin Autodeskin tuote). Revit on siis vaihtoehto AutoCADin käytölle, mutta myös Revitin kanssa voidaan ja kannattaa käyttää MagiCADia, josta on tehty erikseen Revitissä toimiva versio (7).

Revit + Magicad yhdistelmä sisältää hyviä ominaisuuksia (7, s. 17, 18, 29, 31), joita on toivottu myös Autocad + MagCADiin, mutta luultavasti ei tulla niihin toteuttamaan. Todennäköisesti Revit + MagiCADin käyttö tulee lisääntymään ja AutoCAD + MagiCAD jää pois jollain aikavälillä. Kuitenkin mm. siksi, että useitakin vuosia kestävässä projektissa ei voida vaihtaa suunnitteluohjelmistoa kesken kaiken, todennäköisesti vielä viidenkin vuoden päästä AutoCAD + MagiCADia käytetään paljon. Siksi on edelleen mielekästä miettiä, millä tavoin MagiCAD for AutoCAD + NavisWorks -yhdistelmän kanssa työskentelyä voisi vielä tehostaa.

5.1 NWCOUT

Kuten aiemmin mainitsin, nykyinen käytäntö edellyttää, että AutoCADissa tehdään silloin tällöin "NWCOUT"-komento, jotta päivittynyt MagiCAD-malli saadaan päivitettyä NavisWorksiin. Komennon suoritus vie isojen tiedostojen osalta jopa minuutteja aikaa. Tämän on usein hukkaan heitettyä aikaa, koska AutoCADilla ei voi tehdä mitään muuta komennon suorituksen aikana.

Asiaan ei ole ihan helppoa ratkaisua, sekään ei auta että NWCOUT-komennon antaa-kin AutoCADiin ladattu apuohjelma siten että NWCOUT komennon lähetys tapahtuu ohjelmakoodista joka on siirretty eri säikeeseen (ohjelmia voi jakaa useisiin säikeisiin, joita tietokone prosessoi rinnakkain) kuin AutoCAD-ohjelman pääsäie. NWCOUT jumittaa tässäkin tapauksessa pääsäikeen suorituksensa ajaksi.

Yksi ratkaisu olisi, että NWCOUT-komento tehtäisiinkin kokonaan eri prosessista (erillinen auki oleva AutoCAD-ohjelma), kuin se millä parhaillaan työskennellään. Tätä toista AutoCADia kannattaisi myös ajaa kokonaan erillisellä palvelimella, jolloin se ei veisi resursseja koneella, jolla käyttäjä tekee töitä. Työskentelyohjelmasta annettaisiin komento, joka käynnistää apuohjelman, joka ohjaa tämän toisen AutoCADin avaamaan työstettävän DWG:n (tai lataamaan sen viitekuvaksi). Sitten apuohjelma käskisi tämän toisen AutoCADin suorittaa NWCOUT-komennon. Tässä olisi se haitta, että näin toimittaessa DWG pitäisi aina tallentaa ennen kuin halutaan tehdä päivitetty NWC. On parempi että käyttäjä saa halutessaan olla tallentamatta DWG:tä. Näin ollen sillä hetkellä työmuistissa (ei levyllä) oleva DWG pitäisi saada kopioitua ”työ-AutoCADista” ”NWC-orja-AutoCADin” muistiavaruuteen.

5.2 Historiatiedot

Joskus työskennellessä varsinkin suurten tietomallinnuskohteiden kanssa, joiden LVI-mallintamiseen on voinut osallistua kymmeniä henkilöitä, tulee mieleen että tietomallin olisi hyvä sisältää hyvä myös historiatiedot. Kaikilla osilla, jokaisella putkenpätkällä olisi tieto, kuka ja milloin sen on mallintanut ja miten ja milloin sitä on muokattu ja kenen toimesta.

Historiatietoja tallentava ja hallitseva apuohjelma olisi varmaankin melko yksinkertaisesti toteutettavissa. Tietomallinnuksessa syntyvä data rikastuisi ja saisi ikään kuin uuden ulottuvuuden historiatiedoista. Tämän tiedon hyödyntämistapoja on helppo keksiä. Yleensäkin, jos jäsennellyn datan kerääminen on mahdollista ilman ylimääräistä vaivannäköä, sitä kannattaa harrastaa, koska tietomassoille voi olla myöhemmin käyttöä (vrt. kohta 6).

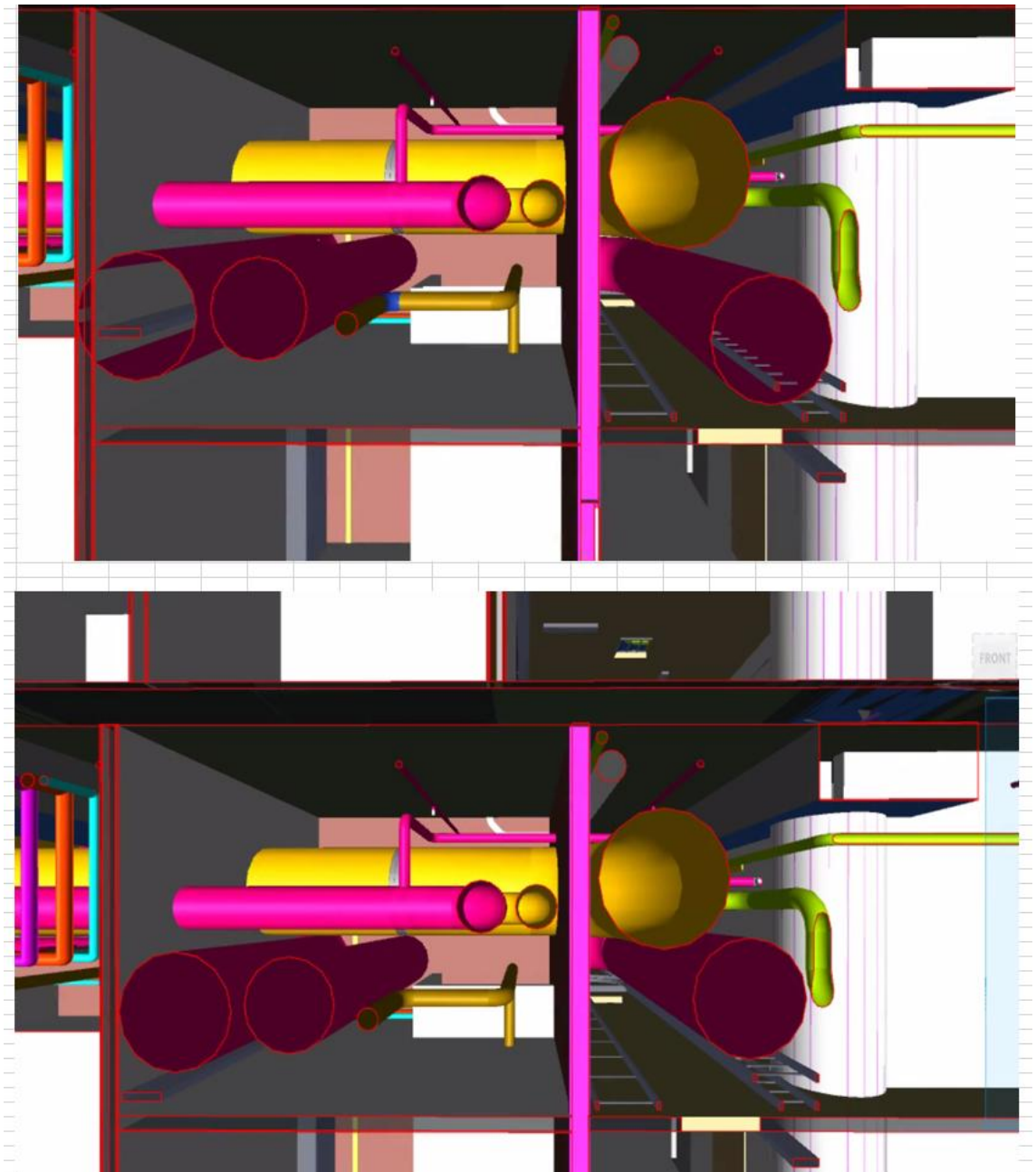
Kun kerätään tietoa, joka voidaan kohdistaa tarkalleen yksittäiseen henkilöön, on tietysti toimittava hyvin varovasti ja harkiten. Tällaisen tiedon keruu voi aiheuttaa negatiivisia reaktioita.

5.3 Asennusten järjestely yhdistelmämallissa

Yhdistelmämallit eivät ole heti syntyessään täysin ristiriidattomia. Ihanteellinen käytäntö on, että eri suunnittelualojen suunnittelijat pitävät tiiviisti yhteyttä jonkin tietyn alueen

mallintamisen aikana, ja varsinkin informoivat toisiaan siirtotarpeista, jos joutuvat mallintamaan toisen "reviirille". Kuitenkin yleinen käytäntö on, että eri osapuolten toimitettua suunnittelumallinsa pidetään palavereja, joissa sovitaan, miten asennukset järjestellään kohdissa, joissa ne näyttävät risteävän keskenään. Palavereissa sovitut asiat täytyy suunnittelijoiden myöhemmin päivittää malleihinsa, jos ja kun eivät ehdi sitä heti palaverissa tekemään.

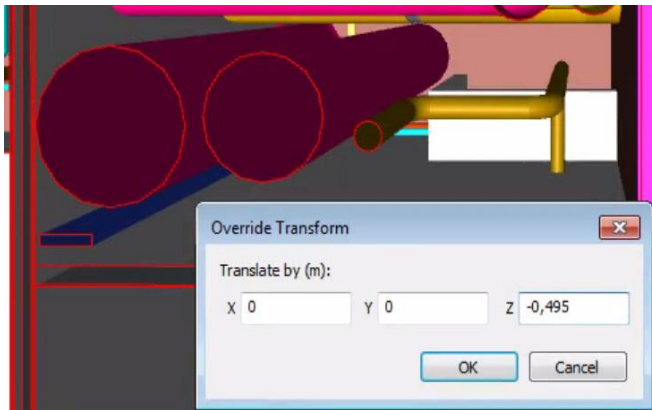
NavisWorksissa osia voidaan myös siirrellä hiirellä (kuva 20), joskaan tätä toimintoa ei ole toistaiseksi paljon käytetty.



Kuva 20. Osien siirto NavisWorksissa. Yllä alkuperäinen tilanne, alla korjattu risteilyjä.

NavisWorksissa tehdyt siirrot eivät tietenkään automaattisesti välity suunnitelmadokumentteihin.

Kun osaa NavisWorksissa siirretään, tallentuu osalle tieto (kuva 21), minkä verran sitä on siirretty (tai vaikka käännetty).

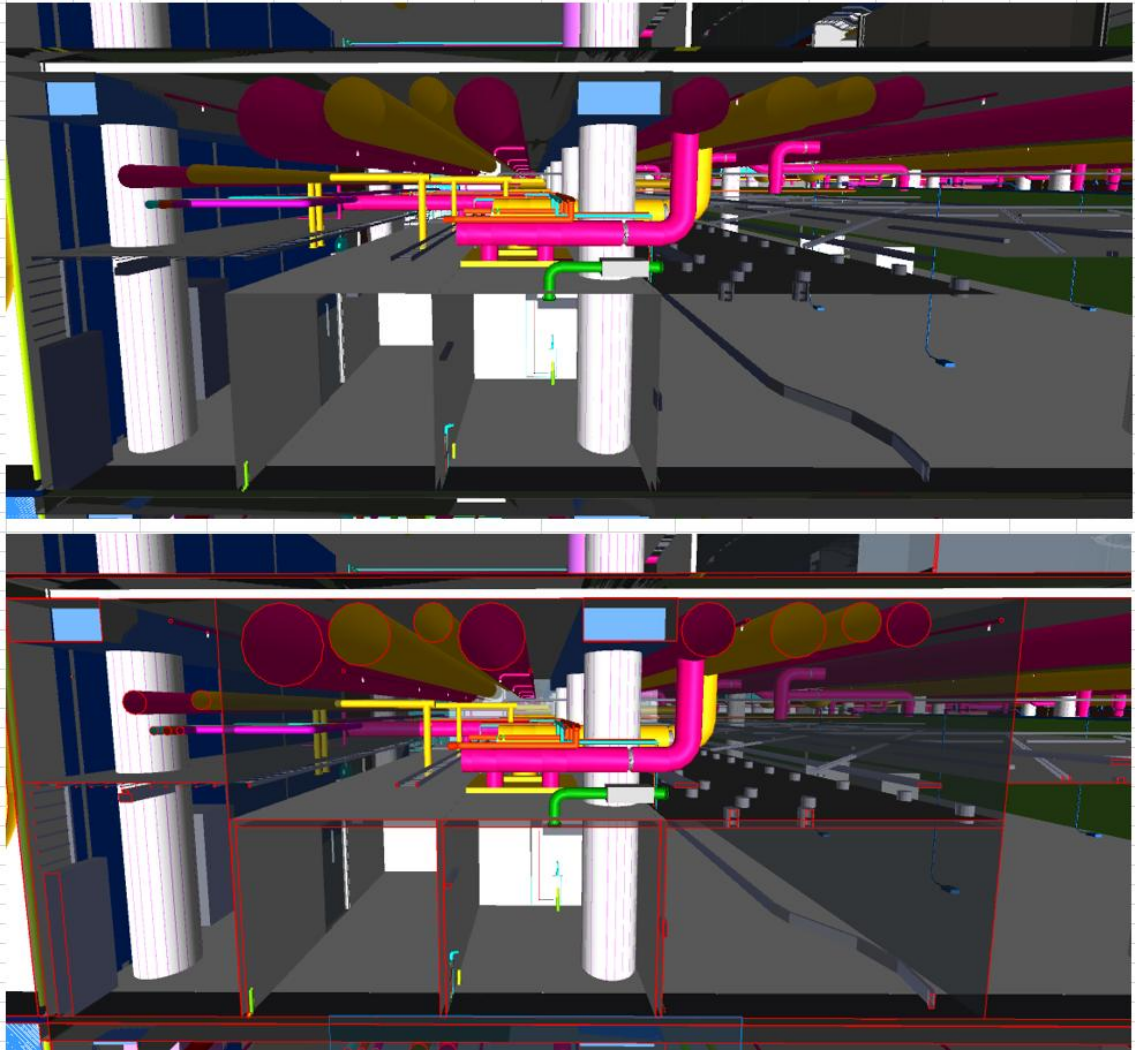


Kuva 21. Alaspäin siirretylle sähköhyllylle tallentunut tieto siirrosta

NavisWorksin APla voisi hyödyntää tässä yhteydessä esim. siten, että luotaisiin apuohjelma, joka huolehtii näiden siirtotietojen koonnasta ja "delegoinnista" eri suunnitteluosapuolille esim. siten, että suoraan suunnitteluohjelmaan saataisiin tuotua jonkinlainen merkintä, mitä pitää siirtää ja mihin. Jossain tapauksissa voisi olla mahdollista hoitaa osan siirto suunnitteluohjelmassa myös suoraan ohjelmallisesti, NavisWorksissa tehdyn siirron mukaisesti.

5.4 Leikkaustulosteiden teko yhdistelmämallista

Leikkausten ulkonäkö tietokoneen näytöllä on huomattavasti parantunut NavisWorksin versiossa 2018, koska vihdoinkin leikkauspintaan tulee korostusmerkinnät (vrt. kuva 22, samasta kohdasta otettu leikkaus versioilla 2016 ja 2018).



Kuva 22. Leikkaukset NavisWorksissa. Yllä versio 2016, alla versio 2018.

Tietomalleja osataan hyödyntää työmaillakin yhä enemmän, mutta usein on kuitenkin vielä tarvetta ”virallisille” leikkauspiirustuksille eli perinteisille paperitulosteille, joissa merkinnät ovat Suomen Rakentamismääräyskokoelmassa esitettyjen piirrosmerkintäohjeiden mukaisia. Tilanteessa, jossa yhdistelmämalli on jo tehty ja halutaan jostakin kohtaa leikkaus selkeyttämään sitä, miten ko. kohdan asennukset on määrä tehdä, ei ole järkevä vaihtoehto laatia kokonaan erillistä leikkauspiirustusta. Samat asiathan joudutaan päivittämään myös malliin (mikäli leikkaus poikkeaa sen hetkisestä mallista).

MagiCADissa on toiminto, jolla saadaan leikkaukset laadittua 3D-mallin pohjalta automaattisesti. Suuria tiedostoja käsitellessä toiminto on kuitenkin hidas. Talotekniikan lisäksi leikkaukseen on yleensä saatava myös välipohjat, seinät ym. rakenteet jollain seuraavista tavoista:

- Taustalle otetaan arkkitehdin laatima leikkaus ko. kohdasta. Usein tarvittavasta kohdasta ei ole valmiina leikkausta, ja jos arkkitehti suostuu lisäleikkauksen tekemään, sitä odotellessa kuluu aikaa.
- TATE-suunnittelija piirtää itse leikkaukseen minimissään kuuluvat välipohjat ym. Tämä on periaatteessa päällekkäistä ja ylimääräistä työtä johon käytetty aika on pois oleellisemmilta työtehtäviltä.
- MagiCADin automaattinen leikkaustoiminto toimii myös rakenteiden kanssa, jos ARK- ja/tai RAK-malli on saatavilla 3D DWG:nä. Yleensä ARK/RAK-mallien julkaisu tapahtuu IFC-muodossa, 3D-DWG-malleja eivät ARK/RAK-suunnittelijat yleensä säännöllisesti jakele tai päivitä. IFC-mallin kääntämiseen DWG:ksi on olemassa työkaluja (käännös vie isojen mallien kanssa tuntikausia aikaa). Joka tapauksessa MagiCADin leikkaustoiminnon käyttö vie runsaasti aikaa jos mukana on vähänkin isompi RAK/ARK 3D DWG ja mahdollisesti lopputulos ei kuitenkaan ole toivotun kaltainen.

NavisWorksin avulla saadaan rakennuksesta mistä kohtaa tahansa leikkaus välittömästi. Malliin on linkitetty kaikkien suunnittelutahojen uusin malli. Puuttuu vain toiminto, jolla saadaan "paperitulosteen" kaltainen leikkauspiirustus kuvan 22 esitystavan sijaan. Tämä on toiminto, jonka kehittäminen NavisWorksin API:n kautta voisi olla hyödyllistä.

6 Yhteenveto

Työn tavoitteeksi oli asetettu päivittäistä työtä helpottavien apuohjelmien laatiminen. Vaatimusten mukaisten ohjelmien toteutus onnistui, ne toimivat suunnitellulla tavalla ja ovat käytössä koko Granlund-konsernissa, joten työtä voidaan pitää siltä osin onnistuneena. Ohjelmat ovat olleet käytössä jo monta kuukautta ja päivittäin niitä käyttää arviolta yli 100 henkilöä.

Tunnetun Mooren lain (8) mukaan transistorien lukumäärä mikropiirissä kaksinkertaistuu noin kahden vuoden välin. Tämä tarkoittaa myös että tietokoneiden suoritusteho on kasvanut valtavasti esim. tämän vuosituhatosen aikana (=suunnilleen aika jonka MagiCAD on ollut olemassa). Jatkovana ongelmana on kuitenkin, että tietokoneella tehtävän työn tuottavuutta laskee ohjelmien hidastelu. Varsinkin suurten tietomallinnuskohdeiden kanssa tämä tulee esiin, vaikka mallit jaetaan pienempiin osiin ja mm. arkkitehtipohjat tehdään tietoteknisesti mahdollisimman keveiksi siivoamalla niistä sellaiset elementit pois, joita ei LVI-suunnitelman yhteyteen tarvita.

Miten tämä on mahdollista? Miksi tilanne on aina yhtä tukala ja ohjelmien hidastelu ei lopu?

Vähemmän tunnetun Wirthin lain (9) mukaan ohjelmistojen laitteistovaatimukset kasvavat nopeammin kuin tietokonelaitteistojen suorituskyky kasvaa. Eri ohjelmien välillä on varmasti paljon eroja tässä suhteessa, ja jotkut varmasti päivitysten myötä jopa nopeutuvat, mutta esimerkiksi Autocad vaikuttaa jokaisen päivityksen jälkeen pöhöttyneemmältä ja tönkömmältä kuin aiemmin, vaikka laitteistokin päivitetään säännöllisesti.

Kuten monelle muullekin asialle, ohjelmistoille pitäisi saada jonkinlainen laatu- tai ympäristöluokitusjärjestelmä. Nopeudelle ja eri ominaisuuksien hyödyllisyyden ja hyödyttömyyden arviointiin pitäisi olla jotkut yleiset kriteerit ja rیمانalitusten myynti pitäisi kieltää.

Tietomallien hyödyntämisessä aivan oma ulottuvuutensa on se, minkälaista hyötyä voi olla saavutettavissa ohjelmistoilla, jotka saavat lähtötietoa lukuisten rakennusten tieto-

malleista ja oppivat siitä. Tekoälyjärjestelmien kehitys on hyvässä vauhdissa (10). Rakennusten tietomallit lienevät rakenteensa ja runsautensa ansiosta erinomaista opetusmateriaalia oppiville tekoälyjärjestelmille ja tämä materiaali lisääntyy jatkuvasti, joten onkin varsin todennäköistä että tekoälyjärjestelmien käyttö rakennusten tietomallien yhteydessä tulee merkittäväksi tekijäksi lähitulevaisuudessa.

Lähteet

- 1 Granlund / Yhtiöstä. 2017. Verkkoaineisto, Granlund Oy.
<http://www.granlund.fi/yhtiosta>. Päivitetty 2017. Luettu 22.1.2018.
- 2 Autodesk Knowledge Network. 2016. Verkkoaineisto.
<https://knowledge.autodesk.com/support/navisworks-products/learn-explore/caas/CloudHelp/cloudhelp/2017/ENU/Navisworks-Manage/files/GUID-BA7FE308-D8D8-4723-9804-E70274178C1A-htm.html> . Päivitetty 16.5.2016. Luettu 24.1.2018.
- 3 Martti Levon / Kustaa Vilkuna. 1946. Puurakennustaito. Helsinki: Oy Suomen Kirja
- 4 Wikipedia. 2015. Verkkoaineisto.
<https://fi.wikipedia.org/wiki/Ohjelmointirajapinta>. Päivitetty 27.11.2015. Luettu 23.1.2018.
- 5 Autodesk Developer Network. 2017. Verkkoaineisto, Autodesk, Inc.
<http://usa.autodesk.com/adsk/servlet/index?id=15024694&siteID=123112>. Päivitetty 2017. Luettu 23.1.2018.
- 6 Autodesk Developer Network. 2017. Verkkoaineisto, Autodesk, Inc.
<http://usa.autodesk.com/adsk/servlet/index?id=1911627&siteID=123112>. Päivitetty 2017. Luettu 23.1.2018.
- 7 Siloaho, Sami. 2017. MagiCAD for Revit tietomallipohjaisessa LVI-suunnittelussa. Insinööriyö. Metropolia Ammattikorkeakoulu.
- 8 Wikipedia. 2017. Verkkoaineisto. https://fi.wikipedia.org/wiki/Mooren_laki Päivitetty 13.1.2017. Luettu 25.1.2018.
- 9 Wikipedia. 2017. Verkkoaineisto. https://en.wikipedia.org/wiki/Wirth%27s_law Päivitetty 7.1.2018. Luettu 25.1.2018.
- 10 Honkela, Timo. 2017. Rauhankone. Helsinki: Gaudeamus