

MicroStation V8i-käyttöympäristö

Juha Orre

Tekniikan koulutusalan opinnäytetyö
Kone- ja tuotantotekniikka
Insinööri (AMK)

KEMI 2013

TIIVISTELMÄ

KEMI-TORNION AMMATTIKORKEAKOULU, Tekniikan yksikkö

Koulutusohjelma:	Kone- ja tuotantotekniikka
Opinnäytetyön tekijä:	Juha Orre
Opinnäytetyön nimi:	MicroStation V8i-käyttöympäristö
Sivuja (joista liitesivuja):	54 (20)
Päiväys:	6.11.2013
Opinnäytetyön ohjaaja:	Ins. (YAMK) Ari Pikkarainen
<p>Opinnäytetyön toimeksiantajana toimi Kemi-Tornion ammattikorkeakoulun tekniikan yksikkö. Työn aihe oli MicroStation V8i -suunnitteluohjelmiston opetuskäyttöön tarkoitettu käyttöympäristö, jonka avulla oppilaat saisivat paremman kuvan työelämässä käytettävistä ympäristöistä ja joka helpottaisi työskentelyä harjoitustöiden parissa. Tehtävänä oli suunnitella ja toteuttaa käyttöön asennettava ohjelmistoversio aiheesta.</p> <p>Kouluissa opetusta annetaan yleensä vain ohjelmistojen perusversioihin aika- ja resurssiseikkojen vuoksi. Myös ohjelmistojen räätälöiminen on vähäistä ja jää useimmiten itseopiskeluksi asian harrastajien keskuudessa. Dokumentoitu toimiva käyttöympäristö toimisi näin ollen myös opetusvälineenä jatkossa tuleville kursseille.</p> <p>Työssä käytettiin hyväksi pääasiassa aikaisempaa ohjelmointitaitoa ja -kokemusta, joka yhdistettiin tehdassuunnitteluympäristöön. Valmiiden mallien puuttuessa ja asiakirjallisuuden vähäisyyden vuoksi tietoa saatiin suurimmaksi osaksi ohjelman valmistajan yhteisöfoorumilta. Toteutus tapahtui suunnittelu-, toteutus- ja testausvaiheissa, jotka toistuivat työn edetessä ja yksittäisten kokonaisuuksien valmistuttua. Ohjelmoinnissa käytettiin MicroStationin omaa VBA-ohjelmointialustaa. Toimintoja testattiin erilaisissa käyttöympäristöissä ja kokemuksia keräämällä ohjelmisto kehittyi vaihe vaiheelta.</p> <p>Tuloksena saatiin verkossa toimiva tietokantapohjainen käyttöympäristöversio, jossa oppilaille on oma käyttöympäristö ja opettajalle erillinen admin-ympäristö. Ohjelmistoon tehtiin hallintatyökalut, joiden avulla menu- ja tehtävärakenteita sekä solukirjastoja ja tietokantaa ylläpidetään. Konsepti on sovellettavissa jatkossa myös työelämään ja muunnettavissa moniin eri käyttötarkoituksiin.</p>	
Asiasanat: MicroStation, V8i, käyttöympäristö, suunnittelu, ohjelmointi, VBA.	

ABSTRACT

KEMI-TORNIO UNIVERSITY OF APPLIED SCIENCES, Technology

Degree programme:	Mechanical and Production Engineering
Author:	Juha Orre
Thesis title:	MicroStation V8i User Interface
Pages (of which appendixes):	54 (20)
Date:	6 November 2013
Thesis instructor(s):	Ari Pikkarainen (M. Eng.)
<p>The title of this final project is MicroStation V8i User Interface and it was commissioned by Kemi-Tornio University of Applied Sciences. The aim was to build up a software solution for educational purposes which could give the students better understanding of real life's work environments.</p> <p>Typically, schools do not have time to teach more than the basic versions of design programs. However, the skill of using these programs and different interfaces is very important, and some kind of teaching aid was needed.</p> <p>The project was carried out by the repetitive circle of design, programming and testing. The great advantage in this work was programming skills and previous experience of VBA language. Information was gathered by studying the Be Communities forum and a couple of MicroStation books. There was not too much material for this project, but by testing a step-by-step the solution was found.</p> <p>The result of this study was a network based user interface with two different user levels, one for the students and the other for the administrator. With the admin-tools the database and menu structures are updatable without programming skills. The solution is also applicable to other MicroStation environments.</p>	
<p>Keywords: MicroStation, V8i, user interface , design, programming, VBA.</p>	

SISÄLLYS

TIIVISTELMÄ	2
ABSTRACT	3
SISÄLLYS	4
1 JOHDANTO	6
2 KEMI-TORNION AMMATTIKORKEAKOULU	8
2.1 Tekniikan koulutusala	9
2.2 CAD-koulutus	10
3 MICROSTATION JA VERSIOHISTORIA.....	12
3.1 Historiaa	12
3.2 Versiohistoria	14
4 KÄYTTÖLIITTYMÄN MUOKKAUS.....	17
4.1 Dgnlib	17
4.2 Workspace - Configuration	17
4.3 Ucf.....	19
4.4 Workspace - Customize.....	21
5 VBA-OHJELMOINTI JA TIETOKANNAT	22
5.1 VBA.....	22
5.2 Ohjelmointi ja makrot	23
5.3 Tietokannat	28
6 UUSI KÄYTTÖYMPÄRISTÖ.....	31
6.1 Tekniset ratkaisut	31
6.2 Ohjelmiston toiminta ja tiedonkulku	33
6.3 Asennus	34
6.4 Kehitystestaus.....	34
6.5 Käyttöönottotestaus ja -kokemuksia.....	34
7 KÄYTTÖOHJEET	36
7.1 Käyttäjän käyttöohjeet.....	36
7.1.1 Ohjelman käyttö	36
7.1.2 Kotipaketti.....	40
7.2 Pääkäyttäjän ohjeet.....	40
7.2.1 Solujen hallinta solukirjastoissa.....	40
7.2.2 Solukirjastojen hallinta.....	41

7.2.3	Soluryhmien hallinta	45
7.2.4	Kotipaketin luonti	50
8	POHDINTA	52
	LÄHTEET	53
	LIITTEET	54

1 JOHDANTO

MicroStation on monipuolinen ja laajalti käytössä oleva suunnittelutyökalu. Perusversiona kuitenkin se, kuten myös moni muu ohjelmisto, on hieman kankea kiireisessä työelämässä. Sen vuoksi suunnittelumaailman moniin eri tarpeisiin on kehitetty valmiita käyttöympäristöjä, joita voidaan lisensoidusti ostaa ja asentaa käyttäjien työasemiin. Näin koneen suunnittelijat, putkistosuunnittelijat, rakennesuunnittelijat ja maanmittaritkin saavat käyttöönsä valmiit ja tehokkaat työkalut työtänsä varten. Monet suunnittelu-toimistot ovat myös kehittäneet itse erilaisia käyttöympäristöjä omiin tarpeisiinsa. Käyttöympäristö (eng. User Interface) tarkoittaa käsitteenä ohjelmiston muokkaamista omiin tarpeisiin, esimerkiksi tekemällä omia menuvalikoita, joilla voidaan valita helposti solukirjastoja ja niistä yksittäisiä osia piirustukseen.

Ongelmia kuitenkin tuottavat alati muuttuvat ohjelmistoversiot, räätälöityjen pakettien kalleus ja käyttöympäristöjen sekä solukirjastojen hajanaisuus. Usein valmiiksi tehtyjen osapiirrosten hyödyntäminen jää tekemättä ja sama osa piirretään tai jopa suunnitellaan moneen kertaan. Tämä ei ole pelkästään MicroStationia koskeva asia, vaan sama ilmiö toistuu useissa insinööritoimistoissa ja useissa eri ohjelmistoissa. Kuitenkin ilman eri tarkoituksiin soveltuvia käyttöympäristöjä, valmiine kirjastoineen, työstä ei tule tuottavaa eikä mielekäästä.

Kouluissa opetus rajoittuu luonnollisista syistä MicroStationin perusympäristöön ja tulevat insinöörit eivät välttämättä saa riittävän hyvää kuvaa työelämässä käytettävistä ohjelmistoista ja niihin asennettavista käyttöympäristöistä. Tähän asiaan Kemi-Tornion ammattikorkeakoulu haluaa saada muutoksen, ja tämä opinnäytetyö on osa tekniikan yksikön MicroStation-opetuksen kehittämistä. Opetukseen halutaan luoda käyttöympäristö, joka vastaa paremmin työelämän mukaista ympäristöä verrattuna perusversioon, ja jonka myötä oppilaiden työskentely ohjelmalla helpottuu. Koulu saa samalla arvokasta tietoa ohjelman käyttöliittymän muokkauksesta ja voi jakaa tätä tietoa eteenpäin tulevassa opetuksessa.

Lähtökohdaksi työssä otettiin helppokäyttöisyys ja helppo päivitettävyyys. Ympäristön muuttaminen eri tarkoitukseen tulisi onnistua ilman ohjelmointitaitoa ja päivitysten tulisi olla kaikkien käytössä yhdellä muutoksella ilman ohjelmistoasennuksia. Uudet menut ja työkaluvalikot, kuin myös vanhojen muuttaminen, tapahtuisivat pääkäyttäjän toimes-

ta keskitetysti. Kaikki resurssit, kuten solukirjastot, makrot ja tietokanta, tulisivat olla kaikkien saavutettavissa ja yhteisessä käytössä, ajantasaisuus olisi siis reaaliaikainen. Käyttöympäristö voitaisiin ottaa pois käytöstä perusopetuksen aikana ja helposti taas käyttöön sitä tarvittaessa. Oppilastyönä voitaisiin tehdä solukirjastoja, jotka kytetään käyttöympäristöön, joko valmiisiin tai myöhemmin tehtäviin menurakenteisiin.

Vaatimuksia oli paljon ja valmista mallia ei ollut, mutta selkeä lähtökohta oli koulun verkon ja palvelinten hyödyntäminen. Oli siis selvää, että suunnittelutyö tulisi viemään suurimman osan ajasta. Kenttä oli avoin ja mielenkiintoinen.

2 KEMI-TORNION AMMATTIKORKEAKOULU

Kemi-Tornion ammattikorkeakoulu aloitti toimintansa osana valtakunnallista kokeilua vuonna 1992, jolloin Kemin ja Tornion kaupungit päättivät perustaa vuonna 1994 toteutuneen kuntayhtymän amk:n ylläpitämistä varten. Koulutusaloina aloittivat tekniikan ala, terveysala, sosiaaliala ja kaupan ala. Kulttuuriala käynnistyi muutaman vuoden kuluksi ja myöhemmin kuntayhtymään liitettiin myös toisen asteen ammatillisia oppilaitoksia. Opetusyksiköt pohjautuivat silloisten opistotasoisien oppilaitosten ja niiden henkilökunnan varaan, ja amk:n ensimmäisenä rehtorina toimi Kyösti Kurtakko (1992 – 1995). (Kemi-Tornion amk:n www-sivut 2013, hakupäivä 30.9.2013.)

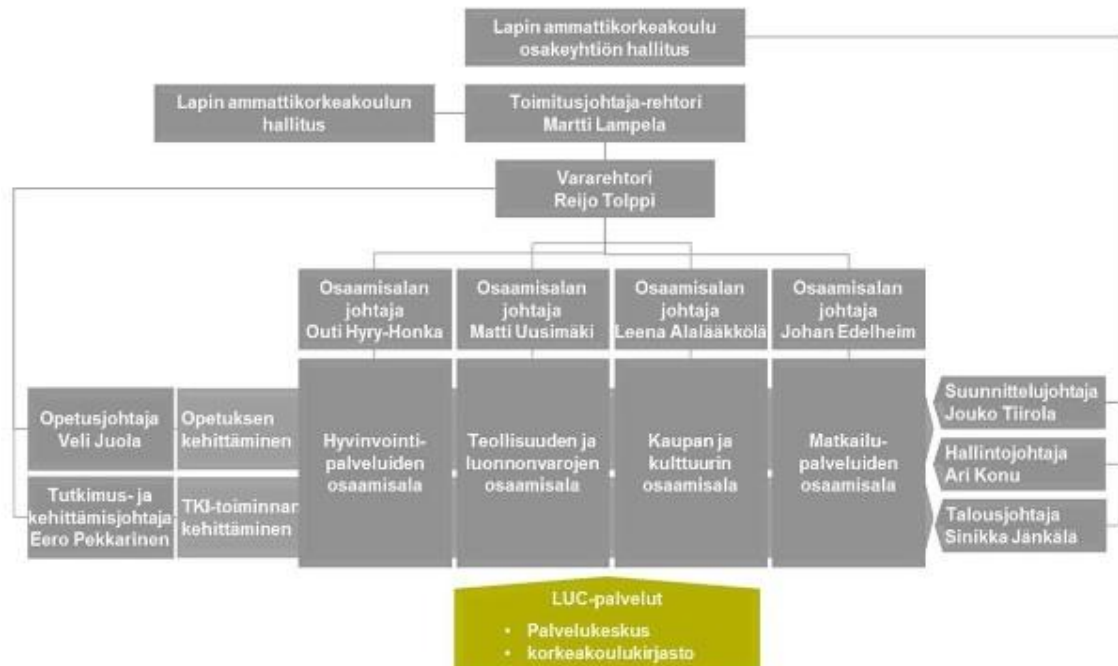
Ammattikorkeakoulu vakinaistettiin vuonna 1997, ja kuntayhtymään kuuluvat toisen asteen oppilaitokset koottiin Kemi-Tornion ammattiopistoksi vuonna 2000. Tuolloin Kemi-Tornion ammattikorkeakoulun kuntayhtymän opiskelijamäärä oli vähän yli 4000 henkeä ja henkilökuntaan kuului n. 400 henkeä, vuotuisen liikevaihdon ollessa n. 26,7 miljoonaa euroa. Vuonna 2007 aloitti toimintansa Kemi-Tornionlaakson koulutuskuntayhtymä Lappia, johon fuusioituivat Kemi-Tornion ammattikorkeakoulun kuntayhtymä ja Länsi-Lapin koulutuskuntayhtymä. Lappian kotipaikkana on Tornio ja kuntayhtymän jäsenkuntia ovat Kemi, Tornio, Keminmaa, Kolari, Muonio, Pello, Simo, Tervola ja Ylitornio. (Koulutuskuntayhtymä Lappian www-sivut 2013, hakupäivä 30.9.2013.)

Kemi-Tornion ammattikorkeakoulun koulutusaloja ovat kulttuuriala (Torniossa), liike-talous (Kemissä ja Torniossa), tietojenkäsittely (Torniossa), sosiaali- ja terveysala (Kemissä) sekä tekniikka (Kemissä). Vuonna 2011 liikevaihto oli 24,8 miljoonaa euroa, henkilökuntaa 264 henkeä ja opiskelijamäärä yhteensä 2949 henkeä. Koulutusta tarjotaan sekä nuoriso- että aikuiskoulutuksena ja myös vieraskieliset koulutusohjelmat kuuluvat tarjontaan. Ammattikorkeakoulun tutkimus-, kehittämis- ja innovaatiotoiminta (TKI) tukee alueen kehittämistä ja vastaa yritysten todellisiin kehittämistarpeisiin. Se on laajuudeltaan ammattikorkeakoulujen valtakunnallista kärkitasoa ja tarjoaa myös opiskelijoille kontaktin todelliseen yritys-elämään. (Kemi-Tornio amk:n www-sivut 2013, hakupäivä 30.9.2013.)

Kuluva vuosi tulee olemaan viimeinen Kemi-Tornion ammattikorkeakoulun nykymuotoisessa toiminnassa, sillä sen ja Rovaniemen ammattikorkeakoulun toiminnat yhdistetään 1.1.2014 alkaen Lapin ammattikorkeakouluksi. Lapin ammattikorkeakoulun halli-

tus on hyväksynyt toimilupahakemuksen 26.9.2013. Korkeakoulua ylläpitää Lapin ammattikorkeakoulu Oy, jonka osakkaat ovat Rovaniemen kaupunki, Kemin kaupunki, Tornion kaupunki, Kemi-Tornionlaakson koulutuskuntayhtymä Lappia, Rovaniemen koulutuskuntayhtymä ja Lapin yliopisto. Yhtiön kotipaikka on Rovaniemi ja toimitusjohtaja-rehtorina aloittaa TK Martti Lampela. Toiminnan alkaessa liikevaihto on n. 43 miljoonaa euroa, henkilöstömäärä n. 560 ja oppilaita yhteensä 5618. (Lapin amk:n www-sivut 2013, hakupäivä 30.9.2013.)

Lapin amk:n osaamisaloja ovat hyvinvointipalvelujen osaamisala. (Kemi-Tornio ja Rovaniemi), kaupan ja kulttuurin osaamisala (Rovaniemi ja Tornio), matkailupalvelujen osaamisala (Rovaniemi) ja teollisuuden ja luonnonvarojen osaamisala (Kemi ja Rovaniemi) (kuvio 1). (Lapin amk:n www-sivut 2013, hakupäivä 7.10.2013.)



Kuvio 1. Lapin amk:n organisaatiokaavio. (Lapin amk:n www-sivut 2013, hakupäivä 7.10.2013.)

2.1 Tekniikan koulutusala

Tekniikan alalla on nuorille ja aikuisopiskelijoille tarjolla insinööri (AMK) -tutkintoon tähtäävä koulutusohjelma kone- ja tuotantotekniikan sekä sähkötekniikan koulutusohjelmissa, ja näiden lisäksi tuotantotalouden koulutusohjelmassa aikuisopiskelijoille.

Insinööri (ylempi AMK) -tutkintonimikkeen voi suorittaa teknologiaosaamisen johtamisen koulutusohjelmassa. Koulutuksessa pyritään vastaamaan tuleviin työelämän tarpeisiin, ja esimerkiksi kaivosalaan liittyvä koulutus on tullut osaksi koulutusohjelmaa. Tutkintoon johtavan koulutuksen lisäksi opetusta tarjotaan myös avoimessa ammattikorkeakoulussa, jonka opintopisteet voidaan jatkossa hyväksilukea mahdollisessa tutkintoon johtavassa koulutuksessa. (Kemi-Tornio amk:n www-sivut 2013, hakupäivä 30.9.2013.)

Tekniikan opetustilat sijaitsevat Kemissä osoitteessa Tietokatu 1 (kuva 1). Rakennuksen saneeraus- ja perusparannustyöt valmistuivat vuonna 2011 ja uudistuneet tilat vihittiin käyttöön 12.12.2011. Samalla rakennus nimettiin Kosmos-taloksi. Uusiin tiloihin muutti tekniikan lisäksi myös sosiaali- ja syksyllä 2013. (Kemi-Tornio amk:n opiskelijoiden www-sivut 2013, hakupäivä 30.9.2013.)



Kuva 1. Kosmos-rakennus.

Uusien tilojen myötä myös kalusto uusiutui ja esimerkiksi CAD-koulutus sai tehokkaammat koneet ja asianmukaiset, hyvin ilmastoidut luokkatilat.

2.2 CAD-koulutus

CAD (Computer-Aided Design) -koulutuksessa pyritään seuraamaan yritysmaailman tarpeita ja kohdentamaan CAD-ohjelmistot kulloinkin yleisimmin käytettäviin ohjelmiin

ja versioihin. Kuluvana lukuvuotena koulutusta annetaan Inventor, AutoCAD ja MicroStation -ohjelmiin sekä 2D ja 3D -suunnittelussa että CAD-perusteissa yleisellä tasolla. CAD-luokka on tehokkaassa käytössä, sillä päivisin opetusta saavat nuorispuolen opiskelijat ja iltaisin ja viikonloppuisin aikuisopiskelijat. Myös muissa ohjelmistoissa pyritään seuraamaan työelämän tarpeita, joten käyttöjärjestelmänä on nyt Windows 7 ja toimisto-ohjelmistona Office 2010.

Kemi-Tornion ammattikorkeakoulu tarjoaa CAD-koulutusta myös yrityksille, joko oppilaitoksen tiloissa tai yrityksessä paikan päällä. Alueen yrityksiin ollaan myös jatkuvassa yhteydessä ja kuunnellaan mitä tarpeita yritysmaailmassa kulloinkin on. Eräs edellä mainituista tarpeista on myös MicroStationin käyttöympäristö V8i-versiolle, joka tässä opinnäytetyössä rakennetaan opetuskäyttöä varten. Erilaisia käyttöympäristöjä on olemassa vanhemmille versioille, mutta versiomuutokset, myös muissa ohjelmissa, antavat aihetta jatkuvalla kehitystyölle. Uuden käyttöympäristön avulla oppilaat saavat paremman käsityksen MicroStationin mahdollisuuksista suunnittelutyössä ja työvälineinä.

3 MICROSTATION JA VERSIOHISTORIA

Bentley Systems Inc:n MicroStation on maailman johtavin mallinnusympäristö arkkitehtuurissa, suunnittelussa, rakentamisessa ja kaikissa infrastruktuurin toiminnoissa, kuten teillä ja rautateillä, silloissa, tehdassuunnittelussa, kaivoksissa, vesistö rakentamisessa, maanmittauksessa jne. Sitä voidaan käyttää niin suunnitteluovelluksena kuin teknologia-alustanakin ja sillä tuotetaan esimerkiksi 2D-suunnittelua ja 3D-mallinnusta, 3D-pdf -tiedostoja, animaatioita ja simulaatioita. Dataa tulostetaan MicroStationin oman dgn-formaatin lisäksi monenlaisissa CAD-formaateissa, mikä tekee ohjelmistosta monipuolisen ja skaalautuvan. (Bentley Communities www-sivut 2013, hakupäivä 7.10.2013)

3.1 Historiaa

MicroStationin tarina sai alkunsa 1980-luvun alkupuolella, jolloin Keith Bentley työskenteli DuPont:n tekniikan osastolla Delawaren Wilmingtonissa. Yritys käytti Integraphin ohjelmistoja prosessilaitoksiensa sähkökaavioiden tuottamiseen. Ohjelmistot toimivat tuolloin VAX-keskustietokoneilla ja niitä käytettiin päätelaitteilla, jotka olivat yhteydessä keskustietokoneeseen. Kallis päätelaitekohtainen lisensointi rajoitti kapasiteetin nostoa työskentelyssä ja tämä oli ongelma yritykselle. (Weisberg 2008, 10-1.)

Keith Bentley uskoi, että on olemassa edullisempi vaihtoehto ja kehitti omalla ajallaan PseudoStationiksi kutsumansa sovelluksen, joka mahdollisti Integraphin CAD-sovelluksen käytön. Sovellus toimi DEC VT-100 terminaalipäätteellä (kuva 2) ja oli varsin tehokas ja edullinen vaihtoehto olemassa olevien piirustusten muokkauksessa.



Kuva 2. DEC VT-100. (vt100.net www-sivu, hakupäivä 7.10.2013.)

Kun Keith Bentley vuonna 1983 siirtyi Integraphilta työskentelemään veljensä Barryn kanssa Dynamic Solution -yritykseen, hän neuvotteli itselleen kaupalliset oikeudet Integraphin PseudoStation-käyttäjien tekniseen tukeen. Los Angelesissa oli tuolloin paljon Integraphin asennuksia, ja veljekset löysivät vastaanottavaisen käyttäjäkunnan PseudoStationilleen. Keith perusti Bentley Systems Incorporated (BSI) -yrityksen sovelluksen kehitystyötä varten ja PseudoStationin varsinainen myynti alkoi heinäkuussa 1984. (Weisberg 2008, 10-1.)

Keith Bentley tuli vakuuttuneeksi siitä, että sen mitä Integraph teki VAX-keskustietokoneella, pystyisi hän tekemään IBM:n PC/AT -mikrotietokoneella. Ohjelmiston uusi versio sai nimekseen MicroStation ja se esiteltiin Compaq 286 -tietokoneelle keväällä 1985 ja myyntiin se tuli 1986. Pian Keith ja Barry myivät osuutensa Dynamic Solutionista ja muuttivat Philadelphiaan päätyen Extoniin, jossa BSI:n pääkonttori sijaitsee. Scott Bentley liittyi yritykseen huolehtimaan käyttöpäästä ja sai myöhemmin tuekseen neljännen veljen Ray Bentleyyn. (Weisberg 2008, 10-1.)

Näin Bentley Systems Incorporation yrityksenä ja MicroStation tuotteena olivat syntyneet. Keith ja Barry Bentley omistivat yrityksestä puolet ja Integraph toisen puolen. Vuonna 1989 MicroStation-versioita tehtiin monille käyttöjärjestelmille kuten PC:lle DOS-pohjaisena, Apple Macintoshille MAC-versiona ja UNIX-työasemille MicroStation 32 ja MicroStation GIS -versioina. PC- ja UNIX-versiot toimivat tuolloin näppäimistöltä, joten hiirellä käytettävä MAC-versio oli monenkin asiantuntijan mielestä tulevaisuuden alusta. (Weisberg 2008, 10-2.)

Tärkeä etappi BSI:n taipaleella oli sen ja Integraphin ajautuminen erilleen vuonna 1992. Siihen asti Bentley oli saanut keskittyä pelkästään ohjelmiston kehittämiseen Integraphin hoitaessa myynnin ja markkinoinnin. Arvokasta tietämystä tälle saralle toi mukanaan vuonna 1994 yritykseen liittynyt viides veli Greg Bentley, joka oli rakentanut finanssialan analyysiohjelmiston Devon Systemsille. Tästä alkoi BSI:n taival itsenäisenä yrityksenä. Oman kehitystyön lisäksi tärkeä elementti MicroStationin kehityksessä oli strategisten suhteiden luominen itsenäisten ohjelmistoyritysten kanssa, jonka ansiosta ohjelmisto laajeni ja kehittyi nykyiseen formaattiinsa. Tällaisia yhteyksiä olivat esimerkiksi Jacobus Technology, WorkPlace Systems, GEOPAK ja BRICS. (Weisberg 2008, 10-3 – 10-9.)

MicroStation-ohjelmiston kannalta eräitä merkittäviä etappeja olivat SELECT-lisenssointi sekä MicroStation J- ja V8 -versiot. SELECT-lisenssointi antoi asiakkaille mahdollisuuden lyhyeen, esimerkiksi kolmen kuukauden käyttöön ja myös mahdollisuuden käyttää aina viimeisintä MicroStation-versiota. MicroStation J toimi Java-ohjelmointikielellä, joka objektorientoituneena sovelluksena vähensi käyttöjärjestelmäriippuvuutta. MicroStation V8 puolestaan ei enää tukenut UNIX- ja Apple-versioita lainkaan. Ohjelmointikieleksi tulivat C ja C++, ja VBA otettiin kehittäjien ohjelmointikieleksi. V8-versio toi MicroStationin myös ajan tasalle teknisten vaatimusten kanssa, joita olivat mm. tiedostokoon kasvattaminen 32 Mb:stä 4Gb:iin, maksimi solun koon nostaminen 64 Kb:stä virtuaalisesti rajoittamattomaan, sallimalla rajoittamattoman referenssimäärän jne. Ajantasaistus tuli tarpeeseen, sillä aiemmat versiot olivat jääneet jälkeen tietotekniikan kehityksestä ja käyttäjien tarpeista. (Weisberg 2008, 10-11 – 12, 10-21 – 22.)

Lähtiessään Integraphilta 1983 Keith poikkesi Huntswillessä ja tarjosi PseudoStationia Integraphille 5000 dollarin hintaan (Weisberg 2008, 10-1). Sanojensa mukaan se, ettei hän myynyt ohjelmistoa, oli ”yksi onnekkaiden sattumien sarjassa” (Weisberg 2008, 10-1). Vuonna 2012 Bentley Systems Incorporation toimi yli 45 maassa 3000 partnerin voimin ja yrityksellä oli 500 miljoonan dollarin vuosittaiset tulot (Bentley www-sivut 2013, hakupäivä 15.10.2013).

3.2 Versiohistoria

Luettelomaisessa versiohistoriassa on esitelty julkaisuajankohta ja muutamia tärkeimpiä versiouudistuksia ja -muutoksia.

- Microstation Version 1 – 1986

Ensimmäinen PC-koneelle laadittu MicroStation-sovellus.

- MicroStation 2.0 – helmikuu 1987

DGN-tiedostot esiteltiin.

- MicroStation 3.0 – joulukuu 1988

Menut esiteltiin. Tähän asti komennot olivat olleet näppäimistöperäisiä.

Sovellus julkaistiin useammille käyttöjärjestelmille, kuten esimerkiksi

MicroStation Mac (v3.4) 1989.

- MicroStation V4 – joulukuu 1990
Motif-graafinen käyttöliittymä ja MDL-ohjelmointi. Uusia ominaisuuksia olivat esimerkiksi jaetut solut, nimetyt tasot, online help, referenssien clipping ja masking, true associative -mitoitus ja tulostuksen esikatselu. Nexus-päivitys sisälsi ensimmäisen DWG-kääntäjän ja kytkennän Windows 3.1:een.
- Bentley CSP / SELECT alkoi tammikuussa 1993
Bentley CSP, alkujaan MicroStation CSP (Comprehensive Support Program), oli ensimmäinen Bentley Systems Inc. -yrityksen itsenäisesti markkinoima ja myymä sovellus.
- MicroStation V5 (v5.0) – lokakuu 1993
Mm. kuvarealistinen renderöinti, binäärinen rasterointituki ja kustomoidut viivatyypit. V5 for Intel NT oli ensimmäinen natiivi Windows-sovellus.
- MicroStation 95 (v5.5) – marraskuu 1995
AccuDraw ja SmartLine -toiminnot esiteltiin. Uusia ominaisuuksia olivat myös mm. MicroStation BASIC, ankkuroitavat työkalulaatikot (tool boxes) ja solun valitsija (cell selector).
- MicroStation SE (v5.7) – marraskuu 1997
Ensimmäiset värilliset ikonit esiteltiin. SE oli viimeinen multi-platform -julkaisu ja se tuki 13 erilaista alustaa.
- MicroStation/J (v7.x) – joulukuu 1998
Tämä oli ensimmäinen versio, johon oli integroitu Java. Versio esitteli mm. Solids modelling -toiminnon (SmartSolid/SmartSurface), QuickVisionGL-toiminnon ja parannetun mitoituksen. Versio tarjosi ensimmäisenä samanaikaisen lisensoinnin SELECTserver:n kautta. MicroStation/J v7.1 tarjosi Windows 2000 -tuen vuonna 1999.
- MicroStation V8 (v8.0) – lokakuu 2001
V8-versio sisälsi monia uusia ja päivitettyjä ominaisuuksia. Esimerkiksi DWG:n lukemiseen tai kirjoittamiseen ei enää tarvittu kääntäjää. Undo-toiminto, tasot ja referenssit muuttuivat rajoittamattomiksi. AccuSnap, teksti- ja mitoitustyylit sekä VBA esiteltiin.
- MicroStation V8.1 (v8.1) helmikuu 2003

Uusia ominaisuuksia olivat mm. digitaalinen allekirjoitus ja digitaalinen tiedostosuojausoikeus. Parannuksia oli tehty 3D Raster Manageriin.

- MicroStation V8 2004 (Edition (v8.5) – huhtikuu 2004

Adoben PDF-kirjastot tulivat MicroStationiin, joka mahdollisti suoran pdf-tulostuksen ja 3D PDF:n.

- MicroStation V8 XM Edition (v8.9) – toukokuu 2006

Uusi versio toimi myös Windows Vistassa ja Microsoft SharePointissa ja tuki AutoCAD 2007/2008 DWG:tä.

- MicroStation V8i (v8.11) – marraskuu 2008

Tämä versio oli ensimmäinen uuden sukupolven MicroStation. Siinä oli AutoCAD 2009 DWG-tuki, import-toiminto Rhino 3DM-formaatille ja vienti OBJ-formaatille. Uusia toimintoja olivat myös Dynamic Views, 3D-mallien visualisointi näyttötyyleillä ja leikkauspinnat. Muita uudistuksia olivat mm. integroitu Print Organizer, Web Map Server -tuki ja GPS-laitetuki.

- MicroStation V8i (SELECTseries 1) – marraskuu 2009

Versio tarjosi Autodesk® RealDWG™ -tuen, objektituen AutoCAD sovellusten datalle ja ESRI SHP -tiedostojen referenssimahdollisuuden. Muita uudistuksia olivat mm. 3D-tulostustuki ja täysi integraatio Luxology Rendering -moottorille.

- MicroStation V8i (SELECTseries 2) – kesäkuu 2010

Tähän versioon oli integroitu Pointool Vortex:n pistepilvituki ja Citrix-valmius. Animaatioiden visuaalisuutta oli parannettu ja tuki Autodesk® RealDWG™ 2010 -kirjastoille ja Autodesk® FBX tiedostoille lisätty.

- MicroStation V8i (SELECTseries 3) – toukokuu 2012

3D-mallien ja 2D-suunnitelmien interaktiivisuutta oli parannettu kuin myös 3D PDF:n ja 3D tulostuksen ominaisuuksia. Simuloinnin ja suunnittelun ominaisuuksia oli parannettu esimerkiksi auringon valotuksen laskimella ja auringon varjogeneraattorilla. Formaattitukina tulivat Autodesk® RealDWG™ 2012, LandXML sekä Bentley Civil Terrain Models JT- ja IFC-tiedostoformaateissa.

(Bentley Communities www-sivut 2013, hakupäivä 8.10.2013)

4 KÄYTTÖLIITTYMÄN MUOKKAUS

Käyttöliittymän muokkaus sinällään on varsin yksinkertainen asia. Kopioidaan PageLayout.dgnlib-tiedosto uuteen sijaintiin, merkitään uusi sijainti Workspace-toiminnon Configuration-muuttujaan ja tehdään käyttöliittymä Workspace toiminnon Customize-työkalulla. Tämä on kuitenkin konekohtainen toiminto, joten toiminnot pitäisi suorittaa jokaisella koneella joka kerta kun halutaan toisenlainen käyttöliittymä. Tämä johtaisi ennen pitkää käyttöliittymäkaokseen ja ohjelmien uudelleenasetuksiin. Jotta käyttöliittymästä kasvaisi käyttöympäristö, täytyy edellä mainittuja toimintoja automatisoida ja tutkia hieman tarkemmin.

4.1 Dgnlib

Dgnlib-tiedostot on luotu parantamaan CAD-standardien noudattamista projekteissa. MicroStation DGN Library -tiedostot esiteltiin Microstation V8 -versiossa ja ne ovat erilaisilla asetuksilla varustettuja normaaleja piirustustiedostoja, ilman mitään graafista informaatiota. Asetuksina näissä voivat olla esimerkiksi tekstityylit, tasot, mitoitusyyliit ja multi-line -asetukset. Kun tiedosto avataan MicroStationissa, asetukset tulevat käyttöön. Tiedostopäätte on tavallisen dgn sijasta dgnlib, joka tarkoittaa dgn-kirjastoa. (Ecado [www-sivut 2000 - 2013](http://www-ivut.2000-2013), hakupäivä 1.10.2013.)

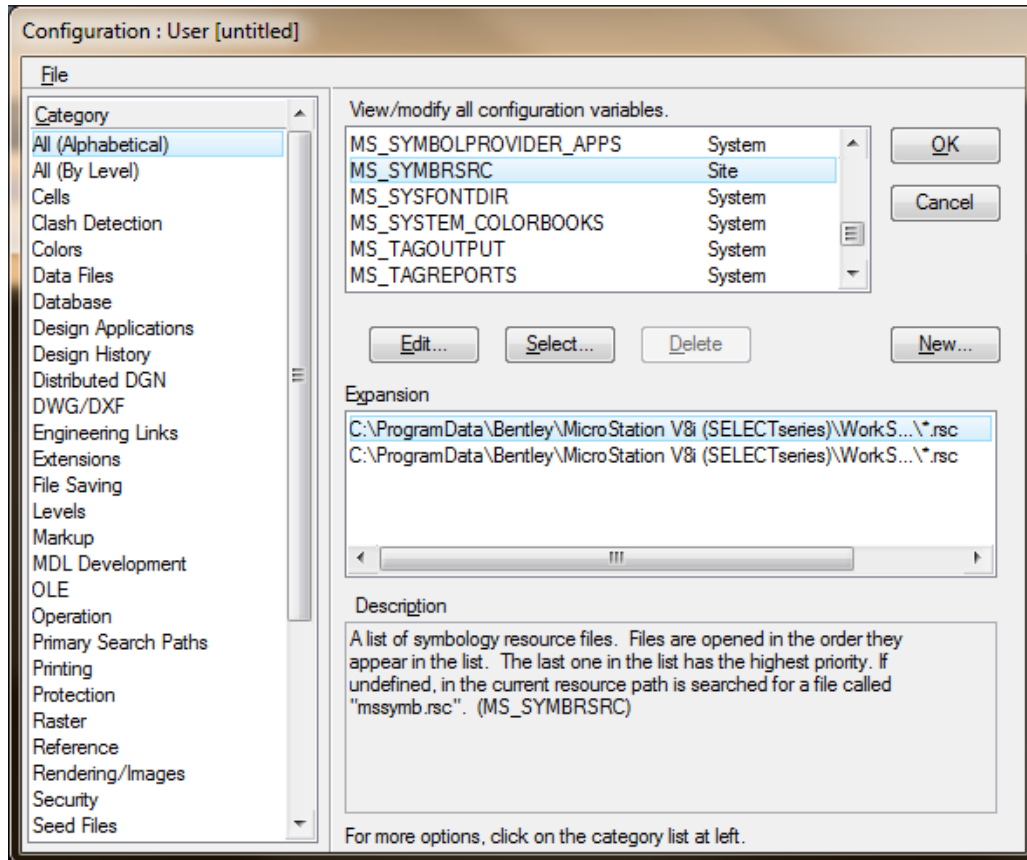
Dgnlib-tiedostoja on useita erilaisia ja ne asentuvat MicroStationin asennuksen mukana. Tiedostot sijaitsevat oppilaitoksen tietokoneissa osoitteessa `c:\Users\Public\Bentley\Workspace\Users\` ja kotikoneilla osoitteessa `c:\Program Data\Bentley\MicroStation V8i (Select Series)\Workspace\System\dgnlib\`.

Koska käyttöliittymäsovelluksessa on tarkoitus muokata lähinnä ulkoasua, valitaan tiedosto PageLayout.dgnlib ja kopioidaan se muokattavaksi yhteiseen sijaintiin palvelimelle.

4.2 Workspace - Configuration

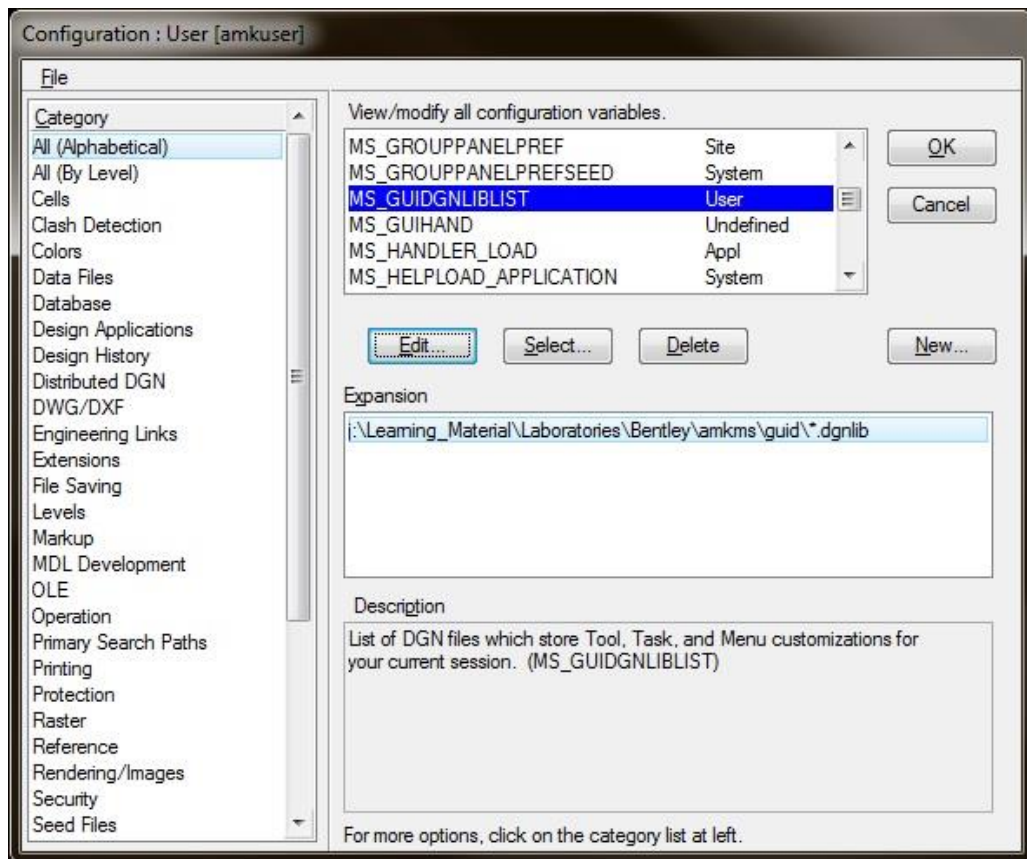
Workspace-menuvalikon Configuration-toiminnolla voidaan lisätä, muokata ja poistaa konfiguraatiomuuttujia (Configuration Variables), joiden avulla ohjelmalle ilmoitetaan kirjastojen sijainnit. Esimerkiksi Open Line Style Library -ikkunaan ilmestyvien kirjastojen sijainnit.

totiedostojen sijainti on määrittänyt muuttujassa MS_SYMBRSCR. (Krishnan, Taylor 2011, 267.) (Kuva 3)



Kuva 3. Muuttujat alkuperäisinä.

Kun halutaan muokata työkaluja, tehtäviä ja menuvalikoita, muokataan MS_GUIDGNLIBLIST-muuttujaa (Bentley Communities www-sivut 2013, hakupäivä 1.4.2013). Muokkaus tapahtuu valitsemalla haluttu muuttuja ja klikkaamalla Edit-painiketta. Muuttujan väri muuttuu listassa siniseksi ja System-teksti muuttuu User-tekstiksi merkiksi siitä, että muuttuja on käyttäjän editoima (kuva 4). Muuttujan arvoksi merkitään se kansio polku, mihin PageLayout.dgnlib-tiedosto on kopioitu.



Kuva 4. Muuttuja käyttäjän muokkaamana.

Muuttujan arvo voidaan siis kirjoittaa Configuration-ikkunaan käsin, mutta arvoja voidaan ohjata myös ucf-tiedostoista.

4.3 Ucf

Ucf-tiedosto (User Configuration File) on tiedosto, jossa voidaan määrätä MicroStationin käyttäjäkohtaisia asetuksia. Oletustiedosto on untitled.ucf (kuva 5), joka voidaan kopioida halutun nimiseksi ja muokata tekstieditorilla.

```

#-----
#
# untitled.ucf - Copy this file as a starting point for new workspaces.
#
# $SRCSfile: untitled.ucf,v $
# $Revision: 1.1 $
#   $Date: 2008/07/30 19:58:57 $
#
# $Copyright: (c) 2008 Bentley Systems, Incorporated. All rights reserved. $
#
#-----

_USTN_USERDESCR   = Untitled Workspace
_USTN_PROJECTNAME = untitled

%if defined (_NAVIGATOR)
_USTN_USERINTNAME = default
%endif
MS_PROTECTION_ENABLE = 1

```

Kuva 5. Ucf-tiedosto alkuperäisenä.

Oletustiedostossa ei ole määritelty muuttujia, vaan MicroStation avautuu perustilassa. Tässä käyttöliittymäsovelluksessa tiedoston nimi on amkuser.ucf ja siihen tehdään seuraavan kuvan (kuva 6) mukaiset muutokset.

```

%if defined (_NAVIGATOR)
_USTN_USERINTNAME = default
%endif
MS_PROTECTION_ENABLE = 1
MS_GUIDGNLIBLIST=j:\Learning_Material\Laboratories\Bentley\amkms\guid\*.dgnlib
MS_VBSEARCHDIRECTORIES=j:\Learning_Material\Laboratories\Bentley\amkms\macros\*.mvba
MS_VBAAUTOLOADPROJECTS=j:\Learning_Material\Laboratories\Bentley\amkms\macros\test1.mvba
MS_DBASE=j:\Learning_Material\Laboratories\Bentley\amkms\db\
MS_CELLSELECTOR=j:\Learning_Material\Laboratories\Bentley\amkms\cells\
MS_DEF=

```

Kuva 6. Ucf-tiedosto käsiteltynä.

MS_DEF-muuttujaan voidaan määritellä piirustusten oletuspolku, joka tässä on jätetty tyhjäksi. Näin konfiguraatiomuuttujia voidaan ohjata käyttäjätiedoston kautta valitsemalla oikea käyttäjä, avausohjeet on esitetty luvussa 7.

4.4 Workspace - Customize

Customize-ikkuna on paikka, missä voidaan rakentaa käyttäjälle omat menuvalikot ja tehtäväpaletit. Oletustiedostoa ei voi muokata, vaan kaikki Customize-ikkunan valinnat ovat oletuksena harmaina. Edellisissä kappaleissa esitettyjen toimenpiteiden ansiosta ikkunassa voidaan kuitenkin valita konfiguraatiomuuttujaan merkitty Page-Layout.dgnlib-tiedosto, joka mahdollistaa omien valikoiden rakentamisen. Valikoiden rakentaminen on esitetty luvussa 7.

Käyttöliittymän automatisointia käyttöympäristöksi ei kuitenkaan voida tehdä pelkillä MicroStationin työkaluilla, vaan siihen tarvitaan VBA-ohjelmointia ja tietokantaa.

5 VBA-OHJELMOINTI JA TIETOKANNAT

Perusasioita, kuten useiden toimintojen yhdistämisiä, voidaan suorittaa nauhoittamalla makroja, mutta automatisointia ja todellista interaktiivisuutta makroilla ei saada aikaan. Tätä varten tarvitaan VBA-ohjelmointia, jolla tehtyjen sql-lauseiden avulla tietokannan kanssa keskustellaan. VBA on ollut osana MicroStationia versiosta 8 lähtien. MicroStationin on myös voinut kytkeä tietokantoihin aiemminkin, mutta tässä sovelluksessa yhteyden määrittämiseen käytetään palvelimella sijaitsevaa VBA-ohjelmaa, joten yhteyden muodostus on automaattinen.

5.1 VBA

VBA tulee sanoista Visual Basic for Applications. VB tarkoittaa Visual Basic -ohjelmointikieltä ja A sovelluskohtaisia objekteja ja käyttäjän käyttöliittymiä. Visual Basicin voidaan ajatella käsittävän 2/3 ohjelmoinnista ja Application-osuuden 1/3. Hallitsemalla siis esimerkiksi MicroStationin VBA:n, hallitsee jo 2/3 Microsoft Excelin VBA:sta. (Winters 2006, 2.)

VBA löytyy varsin monen ohjelmiston taustalta, kuten esimerkiksi Word, Excel, MS Project, PowerPoint, Inventor, MicroStation jne. Se on tarkoitettu sovelluskohtaisiin ohjelmistojen räätälöinteihin ja toimintojen automatisointeihin. Sillä voidaan kytkeä sovellus tietokantoihin, tehdä uusia käyttöliittymiä ja datan syöttöikkunoita, mahdollisuudet ovat erittäin moninaiset. Esimerkiksi Excel VBA:ssa voidaan syöttää kooditasolta tietoa lomakkeille, jolloin saadaan omilla käyttöliittymillä varustettuja erittäin tehokkaita ohjelmistoja vähällä vaivalla. Excel-funktioita voidaan käyttää myös toiseen suuntaan kooditasolla, mikä lisää tehokkuutta ja tämä sama asia pätee myös MicroStationissa.

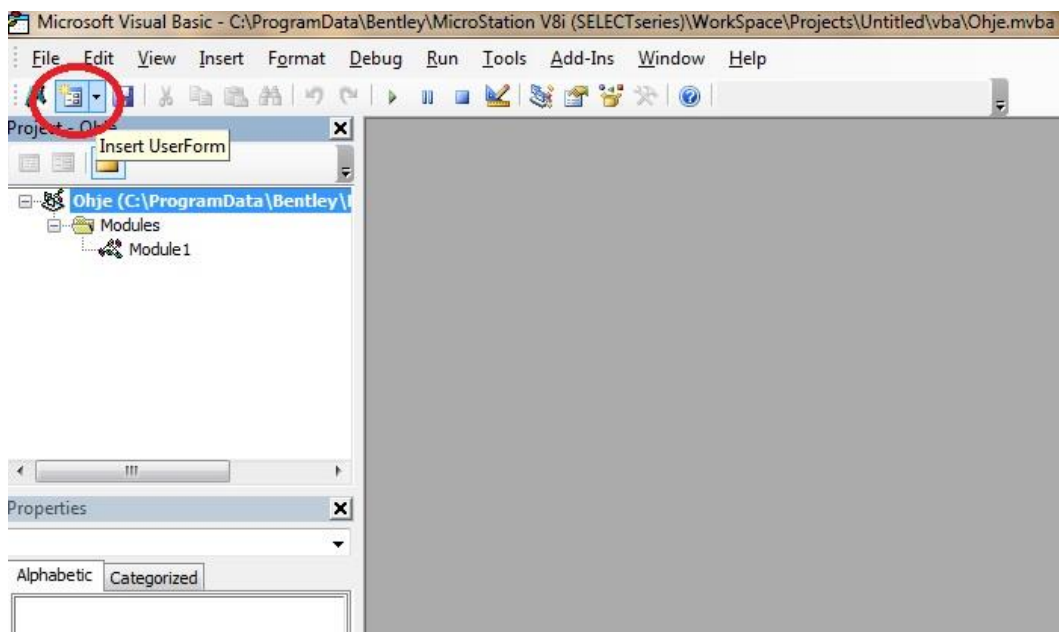
VBA pohjautuu siis Visual Basic -ohjelmointikieleen, josta viimeinen versio VB6 korvautui v. 2003 Visual Studio -ohjelmointialustalla. Tuolloin myös Visual Basicin rakenne ja ohjelmointitapa muuttuivat radikaalisti ja vanha VB jäi ainoastaan VBA-sovellusten käytettäväksi. Tämän vuoksi nykyisin ohjelmointikoulutuksessa ei enää saakaan tuota 2/3:aa VBA-opista suoraan, vaan perehtymiseen täytyy käyttää enemmän aikaa. Mutta ohjelmoijien mielipide on, että kun osaa yhden kielen, ei toisen oppiminen ole enää vaikeaa.

5.2 Ohjelmointi ja makrot

Kun puhutaan makroista, ajatellaan usein vain perinteisiä, nauhoittamalla tehtyjä makroja. Kuitenkin jokainen VBA:lla tehty proseduuri (aliohjelma tai metodi) on sinällään erillinen makro. VBA-ohjelma onkin makrojen kokoelma, jossa makroja suoritetaan käyttöliittymästä päin ja eri makrot voivat kytkeytyä toisiinsa jonkin isomman toimintakokonaisuuden aikaansaamiseksi.

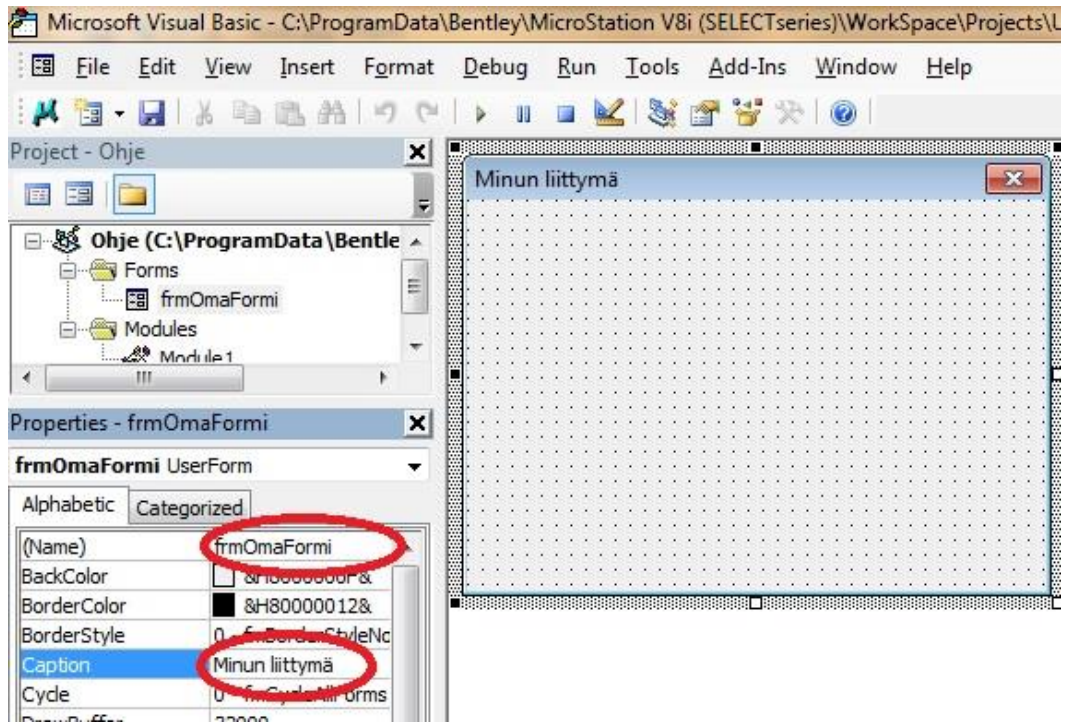
VBA:n peruseriaatteita voidaan ymmärtää parhaiten esimerkkien kautta. Seuraavassa rakennetaan pieni käyttäjän käyttöliittymä, lisätään siihen otsikko- ja tekstinsyöttökenttä sekä painonappi ja tulostuskenttä.

VBA:ta tuotetaan Visual Basic Editorilla, joka avataan Utilities-valikon Macro -> Visual Basic Editor -kohdasta. Avautuvasta editorista valitaan Insert UserForm -käsky (kuva 7).



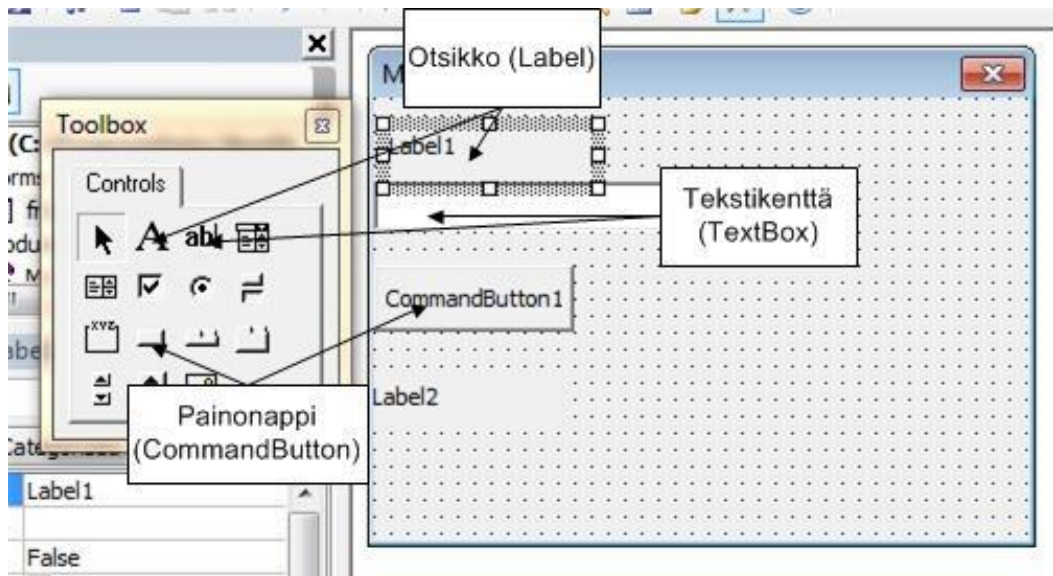
Kuva 7. UserFormin lisäys.

Properties-ikkunassa määritellään kaikille objekteille ominaisuudet. Annetaan nyt ikkunalle nimi (Name) ”frmOmaFormi” ja näkyvä otsikko (Caption) ”Minun liittymä” (kuva 8).



Kuva 8. Ikkunan ominaisuudet.

Klikataan hiirellä UserFormia ja avautuvasta Toolbox-ikkunasta vedetään hiirellä formille kaksi otsikkokenttää (Label), yksi tekstikenttä (TextBox) ja yksi painonappi (CommandButton) (kuva 9). Näitä objekteja sanotaan kontrolliksi (Controls).



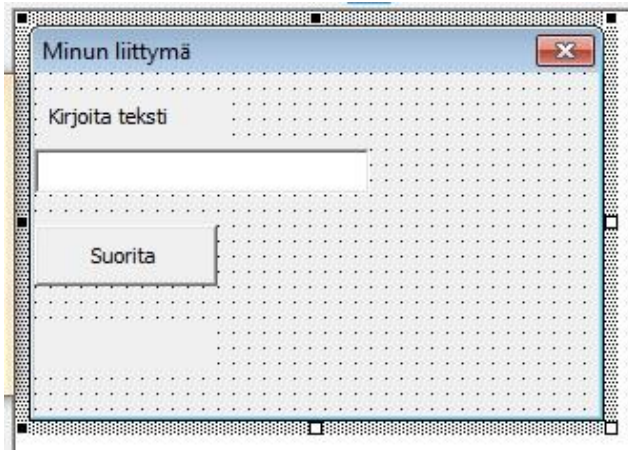
Kuva 9. Kontrollit.

Kirjoitetaan jokaiselle kontrollille ominaisuudet Property-ikkunassa seuraavan taulukon mukaisesti (taulukko 1).

Taulukko 1. Kontrollien ominaisuudet.

Property	Label1	Label2	TextBox1	CommandButton1
Name	lblOtsikko	lblTulostus	txtTeksti	cmdSuorita
Caption	Kirjoita teksti			Suorita

Käyttöliittymä näyttää nyt seuraavalta (kuva 10).



Kuva 10. Käyttöliittymät.

Kun tuplaklikataan painonappia, niin päästään koodin kirjoitustilaan. Kirjoitetaan seuraavan kuvan mukainen proseduurin (kuva 11).

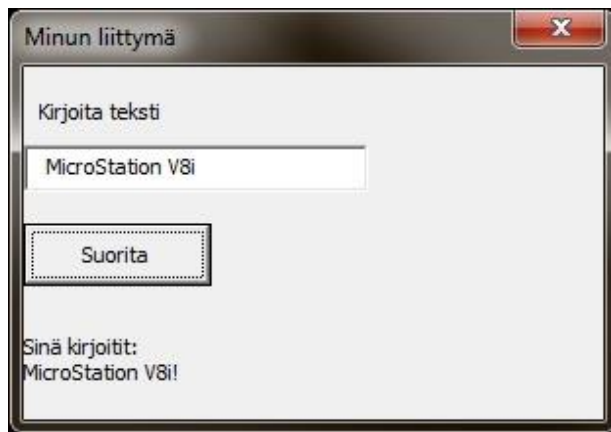
```

ogramData\Bentley\MicroStation V8i (SELECTseries)\WorkSpace\Projects\Untitled\vba\Ohje.mvba - [frmOmaFormi (
t  Format  Debug  Run  Tools  Add-Ins  Window  Help
Ln 3, Col 8
Run Sub/UserForm
Private Sub cmdSuorita_Click()
    lblTulostus.Caption = "Sinä kirjoitit: " & txtTeksti.Text & "!"
End Sub

```

Kuva 11. Koodiesimerkki.

Nyt voidaan klikata Run Sub/UserForm -painonappia (kuva 11) ja suorittaa ohjelma. Kun tekstikenttään kirjoitetaan jokin teksti ja painetaan nappia, teksti ilmestyy tulostuskenttään (kuva 12).



Kuva 12. Toimintojen testaus.

Edellä oleva esimerkki on puhtaasti Visual Basicia, sillä siinä ei ole käytetty lainkaan MicroStationin objekteja. Seuraavassa kuvassa on esimerkki koodista, jossa liitetään Ruuviliitos-solukirjasto solunvalintaikkunaan ja avataan se (kuva 13). Siinä on käytetty mm. AttachCellLibrary-ominaisuutta, joka on pelkästään MicroStationissa toimiva.

```

Private Sub Ohje()
AttachCellLibrary "c:\Ruuviliitos.cel", msdConversionModePrompt
If IsCellLibraryAttached Then
' List the names of all of the cells that are
' eligible to be the active cell.
Dim oCIE As CellInformationEnumerator
Dim OCellInfo As CellInformation
' Do not include shared cells, but include everything
' in the MS_CELLLIST
Set oCIE = GetCellInformationEnumerator(True, True)

Do While oCIE.MoveNext
Dim message As String
message = ""
Set OCellInfo = oCIE.Current
With OCellInfo
If .Is3D Then message = message & "3D "
message = message & .Name
message = message & .Description
End With
Debug.Print message
Loop
End If
'Avataan soluikkuna
Dim startPoint As Point3d
Dim point As Point3d, point2 As Point3d
Dim lngTemp As Long

' Start a command
CadInputQueue.SendCommand "DIALOG CELLMANTENANCE"

CommandState.StartDefaultCommand
End Sub

```

Kuva 13. Esimerkki MicroStationin objektista.

Koodia voidaan tuottaa myös nauhoittamalla makro perinteisesti, eli:

1. Painetaan makron nauhoituspainiketta.
2. Suoritetaan jokin toiminto MicroStationissa.
3. Painetaan makron lopetuspainiketta.

Makron suorituskoodi ilmestyy Visual Basic Editorin Module1-moduuliin. Seuraavassa kuvassa on esimerkki, jossa on nauhoitettu viivan piirtäminen Place SmartLine -komennolla (kuva 14). Kun koodi ajetaan Run-komennolla, se piirtää viivan automaattisesti piirustukseen. Tällä tavalla voidaan suorittaa automaattisesti MicroStationin omia käytäntöjä ja lisätä saatu koodi omaan proseduriin.

```
Sub Macro2 ()
    Dim startPoint As Point3d
    Dim point As Point3d, point2 As Point3d
    Dim lngTemp As Long

    ' Start a command
    CadInputQueue.SendCommand "PLACE SMARTLINE"

    ' Coordinates are in master units
    startPoint.X = -756.971015323925
    startPoint.Y = 1679.84145042147
    startPoint.Z = 0#

    ' Send a data point to the current command
    point.X = startPoint.X
    point.Y = startPoint.Y
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    point.X = startPoint.X + 790.615502393222
    point.Y = startPoint.Y + 258.522093698069
    point.Z = startPoint.Z
    CadInputQueue.SendDataPoint point, 1

    ' Send a reset to the current command
    CadInputQueue.SendReset

    CommandState.StartDefaultCommand
End Sub
```

Kuva 14. Esimerkki automaattikoodista.

5.3 Tietokannat

Tietokanta (database) on tietovarasto, joka yksinkertaisimmillaan voi olla vaikka kynällä kirjoitettu osoitekirja. Useimmiten tietokannat mielletään kuitenkin sähköisiksi tietovarastoiksi, joita ylläpitävät tietokantaohjelmistot ja -palvelimet. Tällaisia ohjelmistoja ovat esimerkiksi Access ja MySQL, jotka sopivat pienille tietomäärille ja vähäisemmälle käyttäjämäärälle sekä Microsoft SQL ja Oracle, jotka suoriutuvat raskaasta raskautuksesta.

Tietokanta koostuu erillisistä asioista, joilla on yhteys toisiinsa. Tiedot varastoidaan tauluihin (table), jotka koostuvat kentistä (field) muodostuvista tietueista. Jokaiselle kentälle määrätään tietotyyppi, joka voi olla esimerkiksi varchar (tekstiä), integer (kokonaisluku), date (päivämäärä) jne., tällä tavoin tietueet voidaan jäsentää oikein ja listata esimerkiksi päivämääräjärjestyksessä. Tietueista muodostuu tauluun rivi, jonka sarakkeita kentät ovat. Jokaisella taululla on ns. pääavain (Primary Key), joka on yksilöllinen (unique index), automaattisesti muodostuva kokonaisluku. Tämä takaa sen, että taulussa voi olla esimerkiksi useita samannimisiä artikkeleita ja ne voidaan kuitenkin erottaa toisistaan.

Seuraavassa esitetään yksinkertainen, Access:lla tehty, tietokantataulu ensin rakennusnäkyssä ja sitten muutaman tietolisäyksen jälkeen (kuvat 15 ja 16).



Kentän nimi	Tietotyyppi
intHenkilöID	Laskuri
strSukunimi	Teksti
strEtunimi	Teksti
strKatuOsoite	Teksti
intPostinro	Luku
strPostitoimipaikka	Teksti

Kuva 15. Rakennusnäky.

intHenkiloID	strSukunimi	strEtunimi	strKatuOsoi	intPostinro	strPostitoir
1	Pitkänen	Juha	Almenikintie 3	95440	Tornio
2	Aaltonen	Kalevi	Aaltokuva 2	95800	Kurkoo
3	Penttilä	Pentti	Katutie 44	96420	Tohmala
*	Uusi				

Kuva 16. Datanäkymä.

Relaatiotietokannalla tarkoitetaan hajautettua tietokantaa, jossa pyritään välttämään päällekkäisiä tallennuksia, eli sama tieto esiintyy vain kerran ja yhdessä taulussa. Esimerkiksi tehdään äänilevytietokanta, johon edellisen esimerkin henkilöt kytketään (kuvat 17 ja 18).

Kentän nimi	Tietotyyppi
intLevyID	Laskuri
strLevyNimi	Teksti
strArtisti	Teksti
dtJulkaisuPvm	Pvm./klo

Kuva 17. Rakennenäkymä.

intLevyID	strLevyNimi	strArtisti	dtJulkaisuPvm
1	Boogie Woogie	Mr Mojo	3.6.2013
2	Kappelin portaat	Suntion pojat	5.6.1998
3	Latotanssit	Various Artists	4.7.1988
*	Uusi		

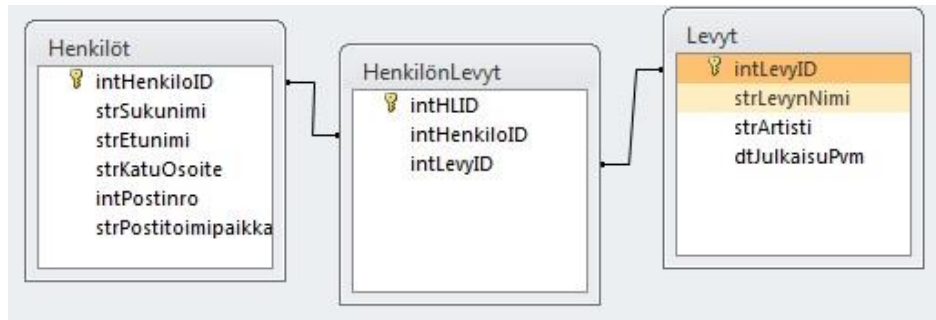
Kuva 18. Datanäkymä.

Sitten tehdään pääavaimiin perustuva kytkentätaulu (kuva 19).

Kentän nimi	Tietotyyppi
intHLID	Laskuri
intHenkiloID	Luku
intLevyID	Luku

Kuva 19. Yhdistelmätaulu.

Nyt kytketään taulut toisiinsa relaatioilla (kuva 20).



Kuva 20. Relaatiot.

Henkilöinen äänilevystön tallentaminen tapahtuu siis pelkästään yksilöllisten id-tunnusten perusteella ja kaikki asiat ovat vain kertaalleen tallennettuja (kuva 21). Ilman relaatiota sama äänilevy pitäisi tallentaa jokaisen omistajan kohdalle erikseen, ja datamäärä paisuisi ennen pitkää kohtuuttomaksi.

intHLID	intHenkiloID	intLevyID	L
1	1	1	1
2	1	2	2
3	2	3	3
4	2	1	1
5	1	3	3
*	Uusi		

Kuva 21. Yhdistelmätaulun datanäkymä.

Henkilöiden äänilevyt haetaan sql-kyselyllä käyttämällä join-lauseita, joilla yhdellä kyselyllä voidaan hakea tietoa useammasta taulusta. Esimerkiksi "select * from henkilönlevyt HL inner join henkilöt H on LH.intHenkiloID = H.intHenkiloID inner join levyt L on LH.intLevyID = L.intLevyID order by H.strSukunimi". Hakuihin voidaan käyttää myös ohjelmistojen ohjattuja raportointityökaluja, jotka muodostavat hakulauseet automaattisesti.

6 UUSI KÄYTTÖYMPÄRISTÖ

Edellisissä luvuissa on esitetty perusasiat, joten nyt käyttöliittymä, ohjelmointi ja tietokannat voidaan yhdistää uudeksi käyttöympäristöksi. Käyttöympäristö on pyritty suunnittelemaan mahdollisimman helppokäyttöiseksi, interaktiiviseksi ja laajennettavaksi, joten tämä on luonut haasteita teknisille ratkaisuille. Kuinka muutokset ja lisäykset saadaan reaaliaikaiseksi? Kuinka muutokset voisi tehdä vain yhteen paikkaan? Kuinka konekohtaisilta päivityksiltä vältetään?

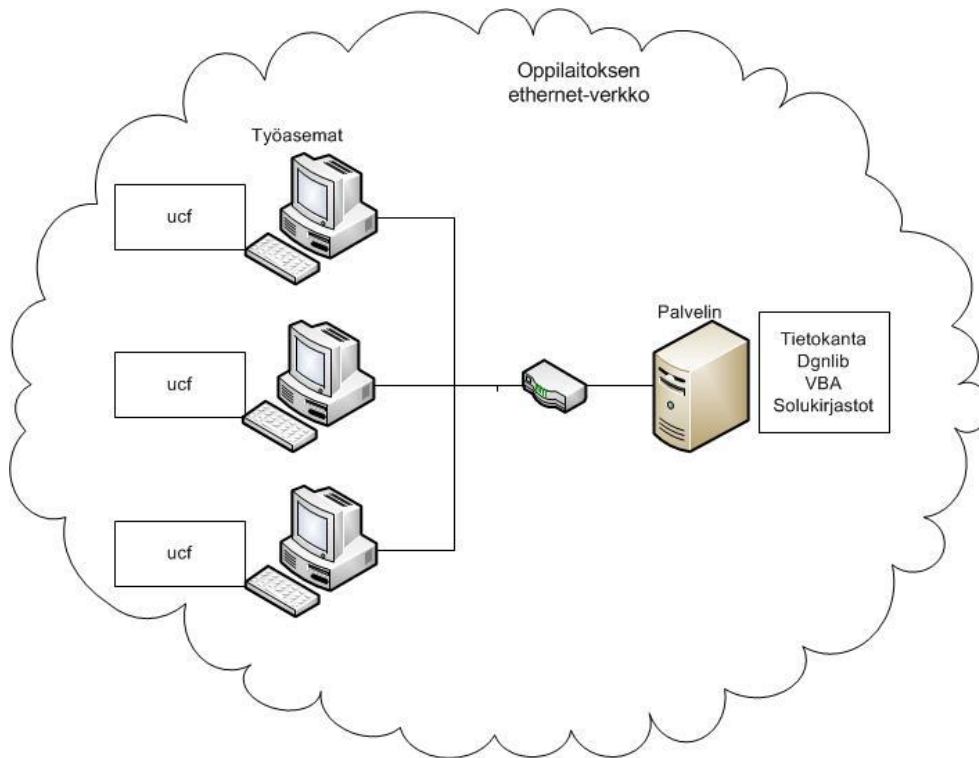
6.1 Tekniset ratkaisut

Reaaliaikaisuuden ja päivitettävyyden vuoksi käyttöympäristöä alettiin suunnitella heti alussa verkkoratkaisun pohjalta. Perinteinen yhden koneen käyttöliittymä -periaate olisi pitemmän päälle kestävä, muutokset pitäisi tehdä aina kaikkiin koneisiin ja kaikki se aika olisi poissa opetus- ja oppimistyöltä. Perusopetuksessa on hyvä käyttää normaaliasetuksia perusvalikoiden kanssa ja kun perusasiat on opittu, jatkaa laajennettuihin käyttöliittymiin.

Lähtökohtaisesti käyttäjien koneille tulisi asentaa mahdollisimman vähän käyttöympäristömekanismeja ja taas verkkoon yhteisiin resursseihin mahdollisimman paljon. Suunnittelussa päädyttiin ratkaisuun, jossa käyttäjän koneelle asennetaan vain yksi tiedosto, jossa kerrotaan missä resurssit sijaitsevat (kuvio 2). Tämäkin tiedosto voidaan asentaa verkosta suoritettavalla bat-tiedostolla, joten käyttäjän ei tarvitse osata teknologiaa lainkaan. Asennetun tiedoston käskyt suoritetaan käyttäjävalinnalla, joten kun amkuserkäyttäjä valitaan piirustusta avattaessa, ympäristö aktivoituu. Muilla käyttäjävalinnoilla menurakenteet säilyvät perustilassa.

Käyttöjärjestelmä rajattiin Windows 7 -versioon, joka on tämän hetken yleisin järjestelmä niin kouluissa kuin yrityksissäkin. Rajaus mahdollisti puolestaan Access-tietokannan käytön, sillä käyttöjärjestelmässä on oletuksena mukana tarvittavat ajurit, eikä Access-ohjelmistoja tarvitse asentaa mihinkään koneeseen, ei edes palvelimelle.

Verkkoratkaisussa kaikki palvelimelle tehtävät muutokset näkyvät käyttäjille reaaliaikaisesti kuhan MicroStation-ohjelma käynnistetään uudelleen.



Kuvio 2. Verkkoratkaisu.

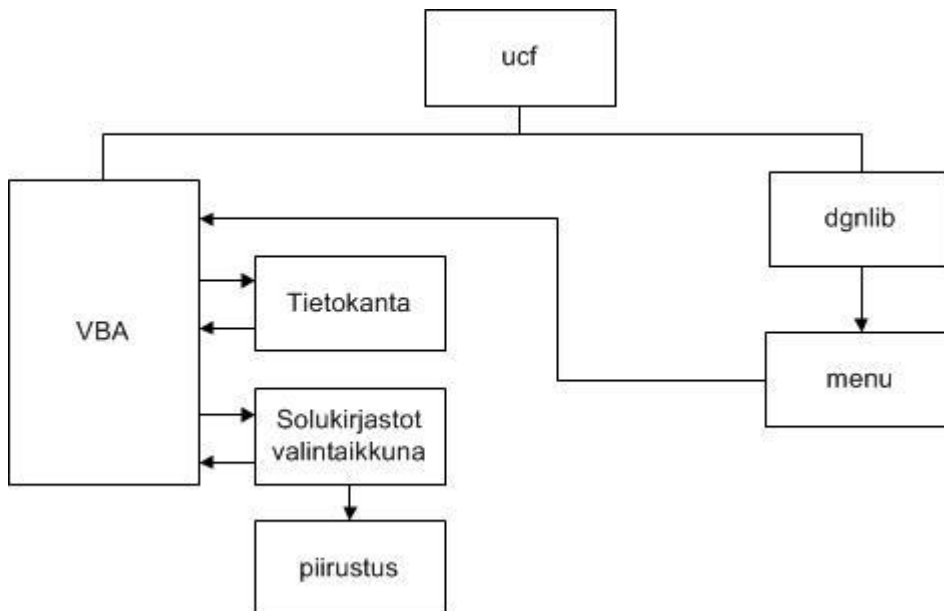
Kotona tehtäviä harjoitustöitä varten laadittiin ns. kotipaketti, joka asennetaan tarkoitusta varten luodulla bat-tiedostolla. Kotipaketissa on palvelimen kansiorakennetta vastaava rakenne, joka muodostuu automaattisesti koneen c-asemalle. Paketin asennuksessa kopioituu kotikoneelle myös amkuserHome.ucf-tiedosto, jossa on kerrottu, missä tarvittavat resurssit sijaitsevat.

Pääkäyttäjälle on tehty oma käyttöliittymä, joka avautuu valittaessa amkadmin-käyttäjä. Käyttöliittymällä voidaan hallita palvelimella sijaitsevaa tietokantaa, joka ohjaa solukirjastojen käyttöä. Pääkäyttäjä myös hallinnoi solukirjastoja ja tarvitsee palvelimen hakemistoihin luku- ja kirjoitusoikeudet, kun taas tavallisella käyttäjällä ei saa olla kuin lukuoikeus. Näin solukirjastot pysyvät järjestyksessä ja solujen oikeellisuus ja käyttökelpoisuus kontrolloidaan ennen järjestelmään lisäämistä.

6.2 Ohjelmiston toiminta ja tiedonkulku

Tiedonkulku on esitetty kuviossa 3.

1. Amkuser.ucf kutsuu muokatun PageLayout.dgnlib-tiedoston ja tarvittavan VBA-makro-tiedoston.
2. PageLayout.dgnlib rakentaa menu- ja tehtävävalikot.
3. Valittaessa solukirjastoryhmä menu- tai tehtävävalikosta, menukomento lähettää VBA:lle proseduurikutsun ja parametrin.
4. VBA avaa tietokantayhteyden ja etsii kannasta kaikki solukirjastot, joissa ryhmäID on sama kuin lähetetty parametri.
5. Mikäli ID löytyy, VBA avaa solukirjaston valinta -käyttäjäikkunan, listaa siihen ryhmän solukirjastot ja sulkee tietokantayhteyden.
6. Kun solukirjasto valitaan ja painetaan Avaa-painiketta, VBA kutsuu MicroStationin solunvalintaikkunan ja liittää palvelimella sijaitsevan saman nimisen solukirjaston ikkunaan.
7. Käyttäjä voi lisätä jonkin kirjaston solun piirustukseen.



Kuvio 3. Tiedonkulku.

6.3 Asennus

Asennukset palvelimelle suoritettiin syksyn 2013 aikana. Verkkoympäristö ja tarvittavat käyttöoikeudet on luotu keväällä 2013. Käyttäjäkoneiden ucf-tiedostot voidaan asentaa oppilastyönä ohjatusti ja verkossa olevien tiedostojen muokkaukset viimeisteltiin testauksen aikana.

6.4 Kehitystestaus

Ohjelmiston toimintoja ja niiden yhteensopivuutta on testattu koko kehityskaaren ajan ensin yhdellä koneella, sitten kahdella yhtäaikaisella ja lopuksi yhden luokan rasiustestillä. Testauksien jälkeen on suoritettu tarvittavat korjaukset ja muutokset. Kotikonepaketin testi on myös suoritettu useammalla koneella.

Admin-testauksessa päähuomio kiinnitettiin verkkoympäristön ja käyttöohjeiden toimivuuteen.

6.5 Käyttöönottestaus ja -kokemuksia

Ohjelman käyttöönottestaus ja toimivuus suoritettiin käyttöohjeita seuraamalla, jolloin saatiin myös kokemukset sekä käyttäjien että pääkäyttäjän ohjeista.

Käyttäjätestaajiksi valittiin joukko kolmannen vuosikurssin oppilaita, joilla kokemusta MicroStationista oli jo jonkin verran. Käyttöliittymä on tarkoitettu hieman pidemmälle ehtineille opiskelijoille, joten ryhmä oli tarkoitukseen sopiva. Kommentit ja palaute olivat pääosin positiivisia, mikä osoitti ohjelman logiikan ja käyttöohjeiden toimivuuden. Muutama parannusehdotuskin saatiin, joista yksi ohjeiden ymmärrettävyyteen liittyvä ehdotus toteutettiin. Sen sijaan ohjelmistoon liittyvät muutosehdotukset todettiin liian suuriksi muutoksiksi, joten esimerkiksi oppilaiden omat objektit valikkorakenteeseen jäivät odottamaan mahdollista jatkokehitystä.

Pääkäyttäjän ohjeet ja ohjelman hallintatoiminnot käytiin läpi opettajan kanssa. Kaksi tuntia kestäneen istunnon aikana luotiin ja poistettiin solukirjastoja ja soluryhmiä, sekä käsiteltiin päämenuvalikon hallinta. Istuntoa nopeutti opettajan aikaisempi tutustuminen ohjeisiin sekä pitkäaikainen kokemus MicroStationin käytöstä ja versioista. Testauksen

aikana ohjelmiston logiikka ja tiedonkulku konkretisoituvat teoriasta käytännöksi. Käyttöohjeet todettiin toimiviksi, mutta opettaja halusi tehdä itselleen henkilökohtaisen, muistiinpanotyypin lyhytversion. Jokainen käyttäjä kokee käyttöohjeet eri tavalla ja omat muistiinpanot nopeuttavat toimintojen muistamista. Myös uusia huomioita MicroStationin makrojen latautumisesta pantiin merkille, nämä eivät kuitenkaan aiheuttaneet muutoksia käyttöohjeisiin, ja ratkaisu löytyi jo saman päivän aikana. Testauksessa käytiin läpi myös palvelimella sijaitseva kansiorakenne ja kotikone-paketin sisältö. Alkuperäisestä ohjelmistosta otettiin lopuksi varmuuskopio, joka voidaan palauttaa mahdollisten tiedosto- tai palvelinvikojen sattuessa.

Käyttöönotto oppilaiden kanssa aloitetaan tammikuun 2014 aikana. Siihen mennessä opettaja valmistaa toisen valikkorakenteen solukirjastoineen, joita oppilaat pääsevät harjoitustöissään täydentämään. Ohjelmisto ja käyttöohjeet todettiin onnistuneesti valmistuneiksi.

7 KÄYTTÖOHJEET

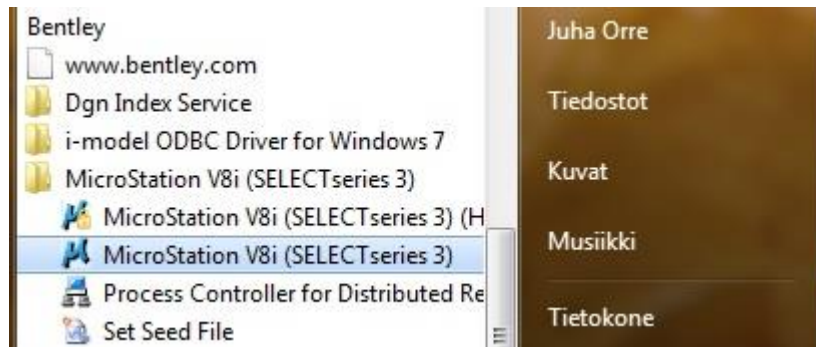
Käyttöohjeet tarvitaan erikseen opiskelijalle eli käyttäjälle ja opettajalle eli pääkäyttäjälle. Pääkäyttäjällä on oikeus hallita solukirjastoja, sekä muuttaa tarvittaessa kirjastojen ryhmitystä. Tavallinen käyttäjä voi ainoastaan käyttää kirjastoja piirustuksissaan.

7.1 Käyttäjän käyttöohjeet

Ohjelmaa voidaan käyttää sekä oppilaitoksen verkossa että kotikoneella desk top -sovelluksena. Jälkimmäistä varten on rakennettu ns. kotipaketti, joka asennetaan käyttäjän kotikoneelle. Tulostettava versio ohjeista löytyy liitteestä (liite 1).

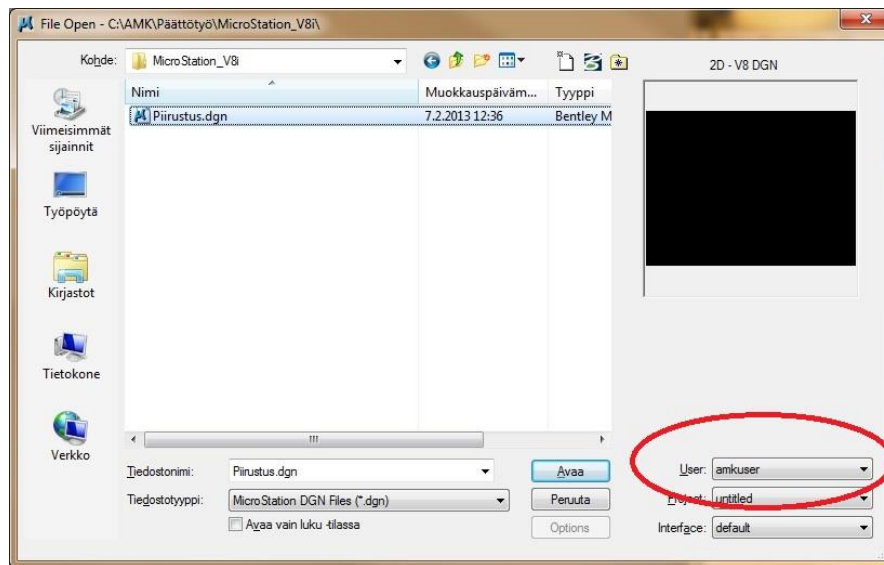
7.1.1 Ohjelman käyttö

1. Käynnistä MicroStation V8i työpöydän tai käynnistysvalikon pikakuvakkeesta (kuva 22).



Kuva 22. Ohjelman käynnistys.

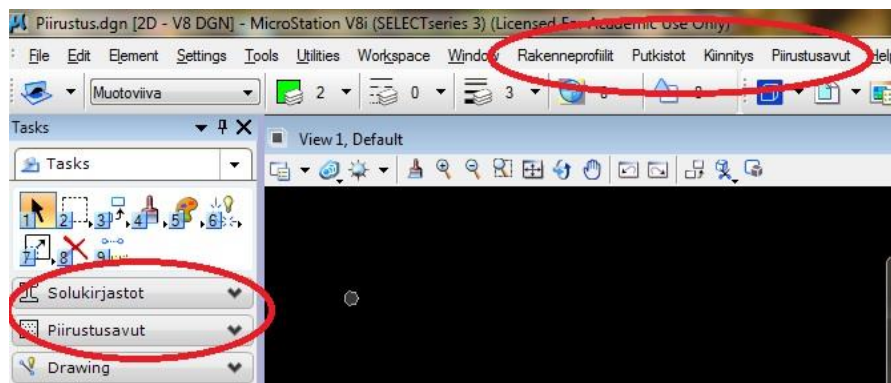
2. Valitse aloitusikkunan User-alasvetovalikosta käyttäjä amkuser (kuva 23).



Kuva 23. Käyttäjän valinta.

Tämän jälkeen avaa normaalilla tavalla haluamasi piirustus tai aloita uusi.

3. Nyt käytettävissäsi ovat uudet solukirjastot sekä menu- että tehtävävalikoissa (kuva 24).



Kuva 24. Valikot.

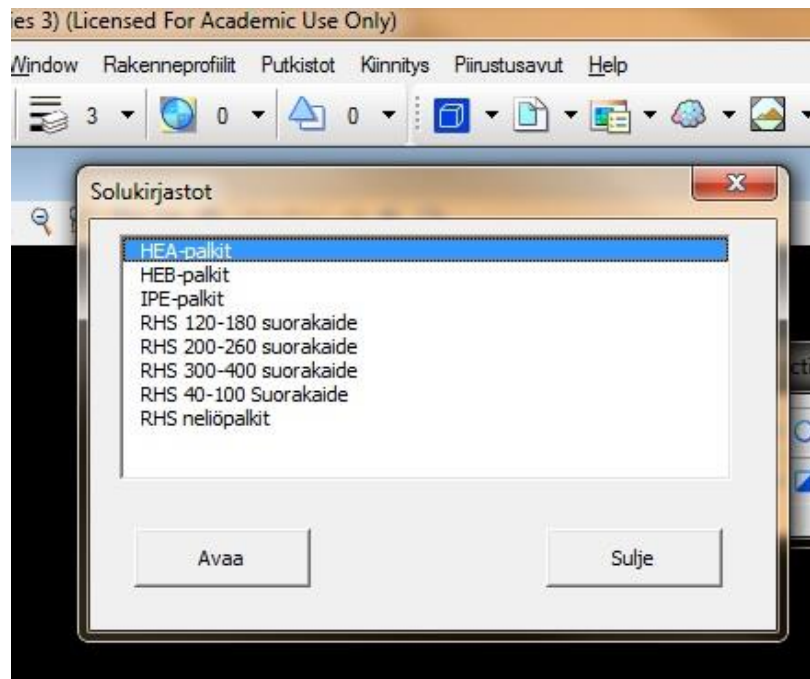
Kirjastoja käytetään yksinkertaisesti valitsemalla valikosta tarvittava kirjasto, ohjelma hoitaa solujen haun automaattisesti. Seuraavassa haetaan esimerkiksi HEA-palkit piirustukseen.

Ensin valitaan Rakenneprofiili-valikosta kirjasto Palkit (kuva 25).



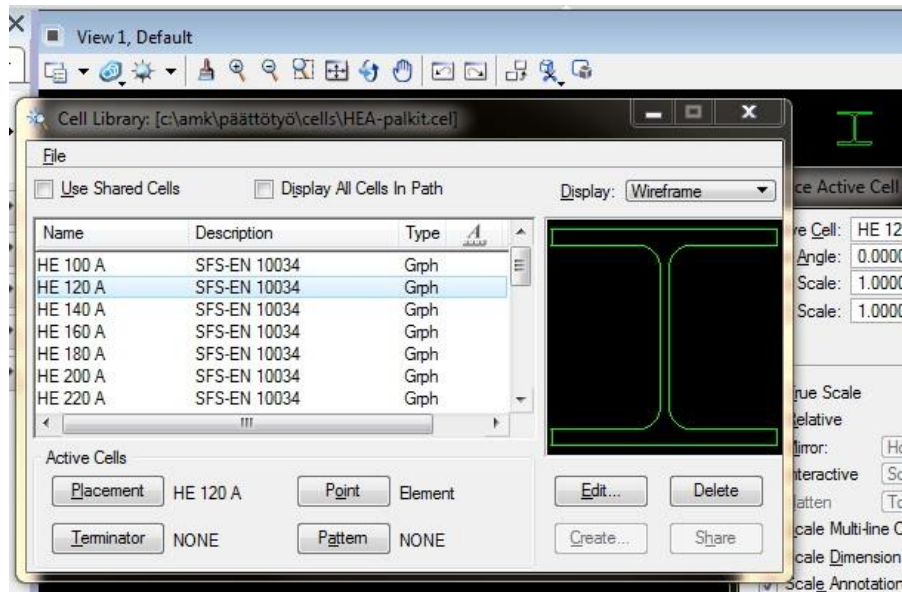
Kuva 25. Solukirjaston valinta.

Avautuvasta ikkunasta klikataan HEA-palkit aktiiviseksi ja painetaan Avaa-painiketta (kuva 26).



Kuva 26. Solukirjaston valinta.

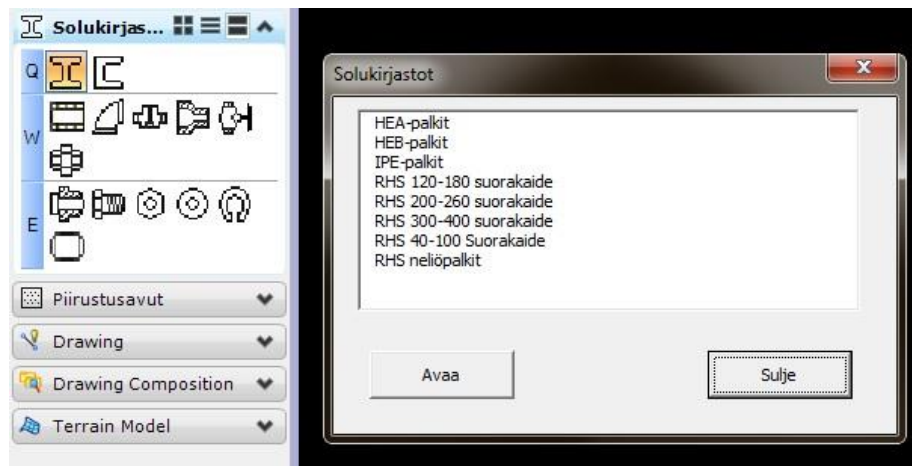
Ohjelma kytkee automaattisesti HEA-palkki -kirjaston MicroStationin omaan solunvalintatyökaluun, josta haluttu palkki lisätään normaalisti piirustukseen tuplaklikkaamalla (kuva 27).



Kuva 27. Solun lisääminen piirustukseen.

Solunvalintaikkunaa ei tarvitse välttämättä sulkea, ohjelma vaihtaa uuden ryhmän ikkunaan automaattisesti.

4. Vasemman laidan tehtävävalikko toimii samalla tavalla kuin ylälaidan menuvalikko (kuva 28).



Kuva 28. Tehtävävalikko.

7.1.2 Kotipaketti

Oppilaitoksen verkossa osoitteessa j:\Learning_Material\Laboratories\Bentley\amkms\ sijaitsee Kotikone-niminen kansio, joka kopioidaan esimerkiksi tikulle kotipaketin asennusta varten. Paketti asennetaan suorittamalla kotikoneelle.bat-tiedosto.

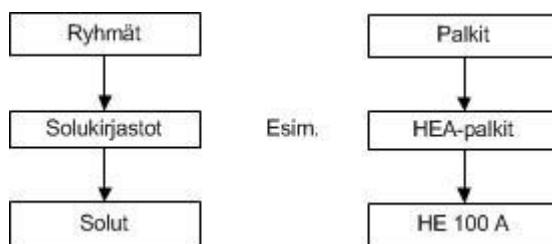
1. Kopioi Kotikone-kansio tikulle.
2. Kytke tikku kotikoneeseen ja suorita (tuplaklikkaa) kotikoneelle.bat-tiedosto.

Nyt koneellesi on luotu c-asemalle amkms-niminen kansio, jossa ovat kaikki tarvittavat tiedostot. Älä muuta tai poista näitä tiedostoja, tällöin ohjelma lakkaa toimimasta. Osoitteeseen c:\ProgramData\Bentley\MicroStation V8i (SELECTseries)\Workspace \users\ on kopioitunut amkuserHome.ucf-tiedosto, älä tee siihenkään muutoksia. Mikäli ohjelmistoon tulee muutoksia tai lisäyksiä, saat koululta uuden paketin. Sen asennus korvaa vanhat tiedostot ja saat uudet resurssit heti käyttöösi.

Nyt voit seurata käyttöohjetta, mutta aloitusikkunassa valitse amkuserHome-niminen käyttäjä.

7.2 Pääkäyttäjän ohjeet

Pääkäyttäjä voi hallita ryhmiä, solukirjastoja ja yksittäisiä soluja. Rakenne on hierarkkinen kuten esimerkiksi Windows-käyttöjärjestelmän kansiorakenteessa (kuvio 4). Tulostettava versio ohjeista löytyy liitteestä (liite 2).



Kuvio 4. Hierarkia.

7.2.1 Solujen hallinta solukirjastoissa

Solukirjastot sijaitsevat oppilaitoksen palvelimella osoitteessa j:\Learning_Material\Laboratories\Bentley\amkms\cells. Pääkäyttäjällä täytyy olla kirjoitusoikeudet kyseiseen kansioon. Solukirjastoissa on tiedostopäätte .cel, esimerkiksi HEA-palkit.cel.

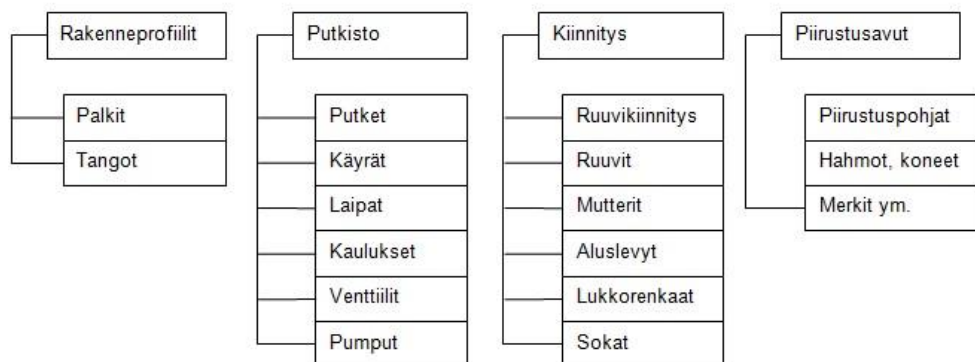
Solun lisääminen tapahtuu yksinkertaisesti lisäämällä uusi solu haluttuun kirjastoon, jonka jälkeen solu on kaikilla käytettävissä. Tietokantamuutoksia ei tarvita. Tärkeää on vain tehdä muutokset palvelimella olevaan kansioon. Samalla tavalla tehdään solujen poistaminen ja muuttaminen.

7.2.2 Solukirjastojen hallinta

Solukirjastot sijaitsevat fyysisesti osoitteessa j:\Learning_Material\Laboratories\Bentley\amkms\cells. MicroStation käyttää Access-tietokantaa solukirjastojen näyttämiseen ja etsimiseen. Tietokanta sijaitsee samalla palvelimella osoitteessa j:\Learning_Material\Laboratories\Bentley\amkms\db.

Tietokantaa hallitaan MicroStationiin rakennetuilla työkaluilla, jotka tulevat näkyviin valittaessa aloitusikkunassa amkadmin-käyttäjää. Solukirjastot puolestaan kopioidaan manuaalisesti edellä mainitun palvelimen kansioon. Tärkeää on, että tietokannassa ja kansiossa ovat samannimiset solukirjastot, esimerkiksi palvelimen kansiossa HEA-palkit.cel ja tietokannassa nimi HEA-palkit. Tällöin ohjelma löytää oikean solukirjaston sitä haettaessa.

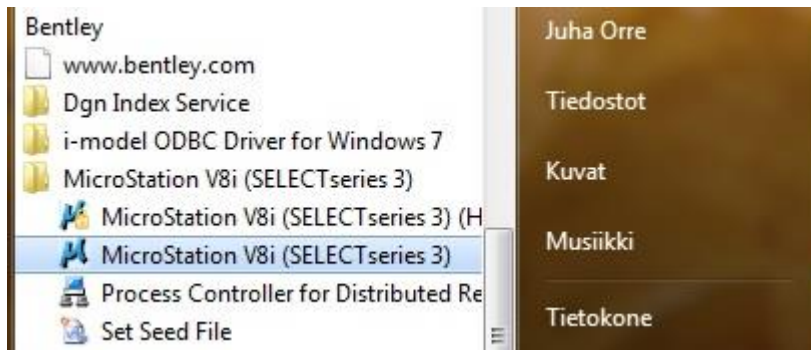
Käyttöympäristöön on rakennettu valmiiksi valikkorakenne, jota voidaan jatkossa muokata ja laajentaa (kuvio 5). Rakenne auttaa oppilaita ymmärtämään erilaisten rakenteiden, materiaalien ja kiinnitysten käyttöä ja ryhmittelyä. Rakenne on ns. lähtötilanne, joka tulee muuttumaan käytön tuoman kokemuksen mukaan. Jotkin solukirjastot ovat vielä tyhjiä, näitä voidaan täydentää esimerkiksi oppilaiden harjoitustöinä.



Kuvio 5. Alkuperäinen valikkorakenne.

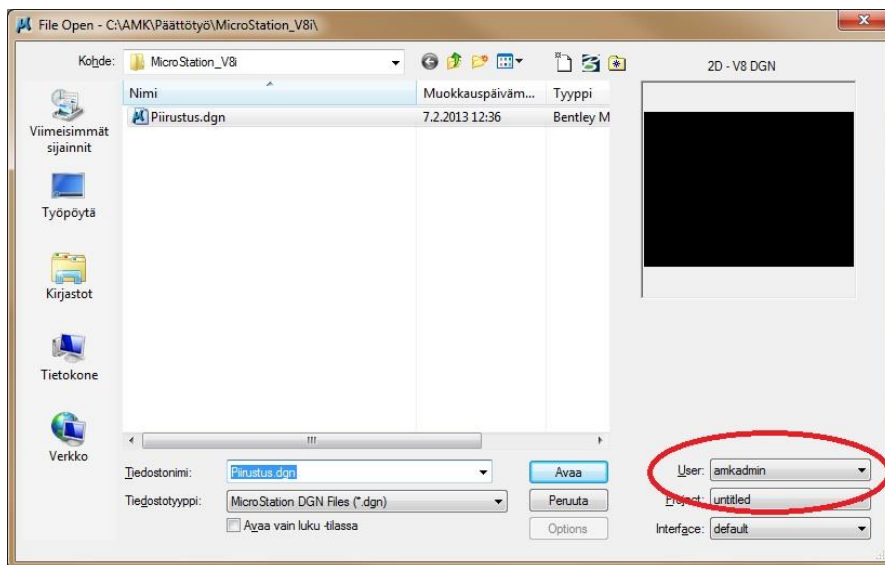
7.2.2.1 Solukirjaston lisäys

1. Käynnistä MicroStation V8i työpöydän tai käynnistysvalikon pikakuvakkeesta (kuva 29).



Kuva 29. Ohjelman käynnistys.

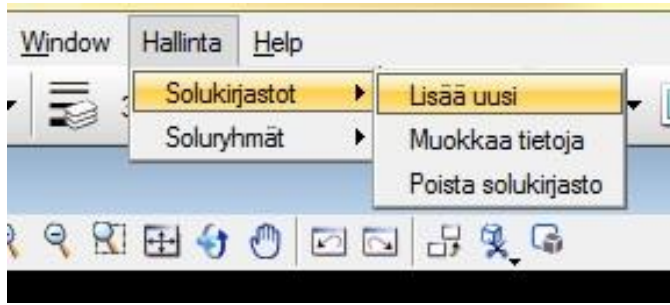
2. Valitse aloitusikkunan User-alasvetovalikosta käyttäjä amkadmin (kuva 30).



Kuva 30. Käyttäjän valinta.

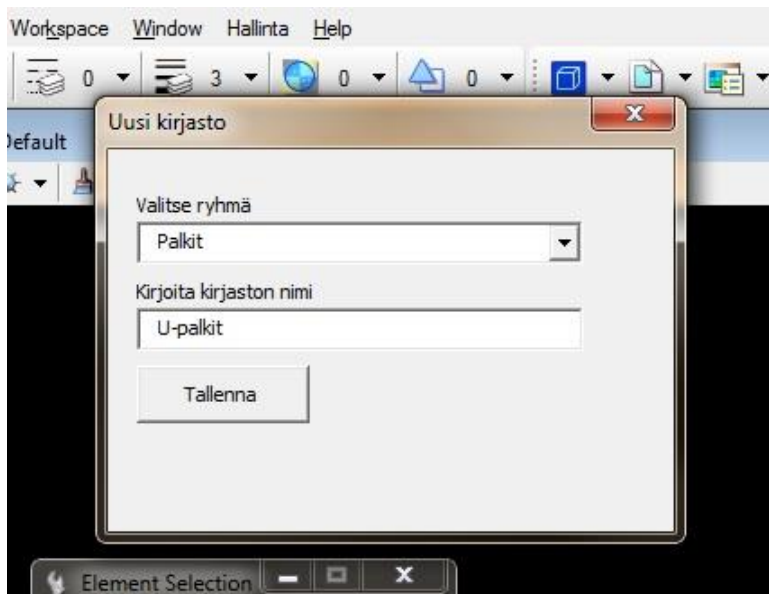
Tämän jälkeen avaa normaalilla tavalla mikä tahansa piirustus tai aloita uusi.

3. Kun lisää uuden solukirjaston järjestelmään, kopioi tai tee aluksi solukirjasto j:\Learning_Material\Laboratories\Bentley\amkms\cells -kansioon, esimerkiksi U-palkit.cel. Tämän jälkeen avaa MicroStationin Hallinta-menusta solukirjastojen lisäystoiminto kuvan mukaisesti (kuva 31).



Kuva 31. Solukirjaston lisäys.

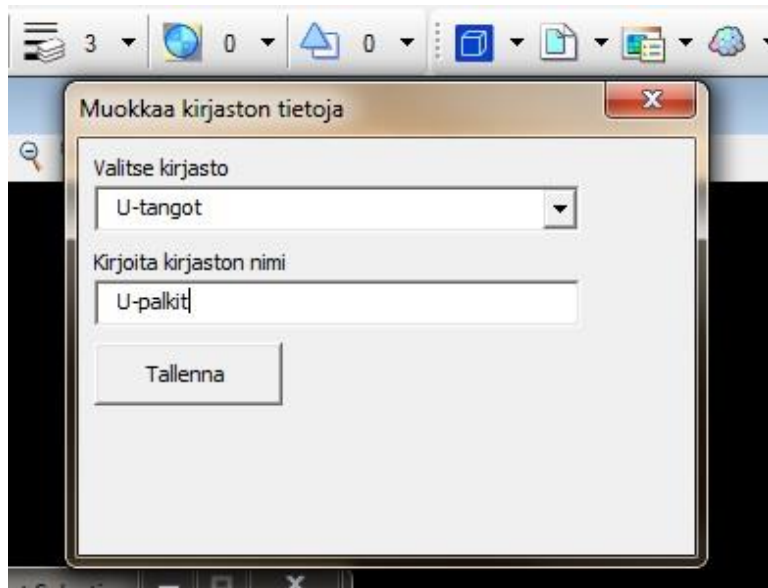
Sitten valitse avautuvan ikkunan Valitse ryhmä -alasvetovalikosta ryhmä, johon uusi solukirjasto liitetään. Kirjaston nimeksi annetaan täsmälleen sama nimi kuin solukirjastolla on, mutta ilman .cel päätettä (kuva 32). Polkua ei tarvitse muuttaa. Paina lopuksi Tallenna-painiketta.



Kuva 32. Solukirjaston lisäys.

7.2.2.2 Solukirjaston muokkaus

Hallintatyökalun Muokkaa tietoja -toiminto on solukirjaston nimen muuttamista varten. Alasvetovalikosta valitaan haluttu kirjasto, kirjoitetaan uusi nimi ja tallennetaan muutos (kuva 33).

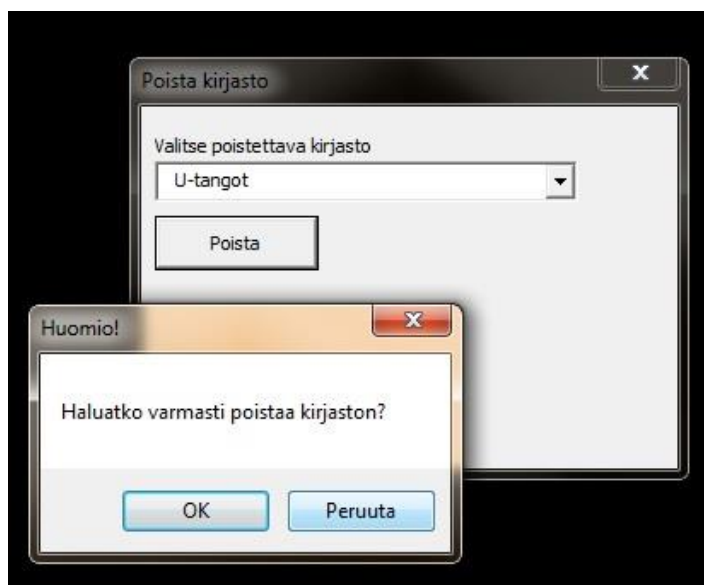


Kuva 33. Solukirjaston muuttaminen.

Nimimuutokset täytyy tehdä myös PageLayout.dgnlib ja kyseisen solukirjaston cel-tiedostoon.

7.2.2.3 Solukirjaston poistaminen

Jos kirjasto halutaan poistaa, valitaan poistoikkunassa haluttu kirjasto alasvetovalikosta, ja painetaan Poista-painiketta. Tämän jälkeen ohjelma vielä varmistaa asian kaiken varalta (kuva 34).



Kuva 34. Solukirjaston poistaminen.

Nyt solukirjasto voidaan poistaa palvelimen kansioista, joskaan se ei ole välttämätöntä. Kun tietokannassa ei ole viittausta kirjastoon, ei se myöskään ole valittavissa piirustuksiin.

7.2.2.4 Polun muuttaminen

Ohjelma käyttää solukirjastojen etsimiseen amkuser.ucf-tiedostoon merkittyä tiedostopolkua. Mikäli siis verkkoympäristö muuttuu ja esimerkiksi verkkoasemien tunnukset tai polut vaihtuvat, niin tiedosto avataan tekstinkäsittelyohjelmalla (esimerkiksi Notepad) ja kaikki polut muutetaan uusiksi (kuva 35). Muutos tehdään vain ns. ”pääpolun” osalta. Esimerkin amkms-kansiossa sijaitsevat kaikki ohjelman vaatimat kansiot, kuten Cells, Macros, DB jne. Muutokset tehdään myös amkadmin.ucf-tiedostoon.

```
%if defined (_NAVIGATOR)
  _USTN_USERINTNAME = default
%endif
MS_PROTECTION_ENABLE = 1
MS_GUIDGNLIBLIST=j:\Learning_Material\Laboratories\Bentley\amkms\guid*.dgnlib
MS_VBASEARCHDIRECTORIES=j:\Learning_Material\Laboratories\Bentley\amkms\macros\*.mvba
MS_VBAAUTOLOADPROJECTS=j:\Learning_Material\Laboratories\Bentley\amkms\macros\test1.mvba
MS_DBASE=j:\Learning_Material\Laboratories\Bentley\amkms\db\
MS_CELLSELECTOR=j:\Learning_Material\Laboratories\Bentley\amkms\cells\
MS_DEF=
```

Kuva 35. Polun muutos.

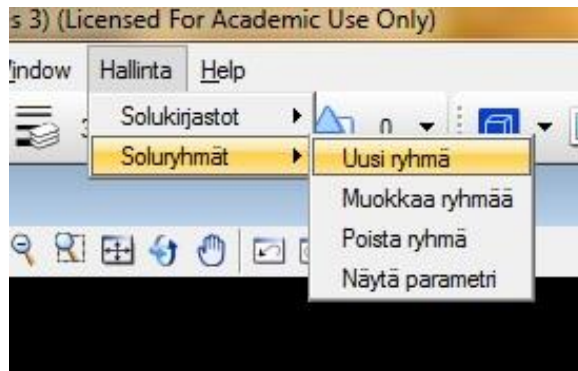
Tiedosto sijaitsee oppilaitoksen tietokoneissa osoitteessa c:\Users\Public\Bentley\Workspace\Users\ ja kotikoneilla osoitteessa c:\Program Data\Bentley\MicroStation V8i (Select Series)\Workspace\Users\. Mikäli tiedosto ei löydy edellä mainituista osoitteista, kannattaa käyttää resurssin hallinnan etsi-toimintoa. Muutos täytyy tehdä kaikkiin tietokoneisiin, joten helpoin tapa on muuttaa tiedot palvelimella sijaitsevaan amkusers.ucf-tiedostoon ja kopioida se ohjatusti oppilastyönä tarkoitusta varten tehdyllä bat-tiedostolla verkon yli. Komento täytyy suorittaa jokaisella koneella erikseen.

7.2.3 Soluryhmien hallinta

Hallintatyökalulla voidaan lisätä ja poistaa ylävalikossa ja tehtävävalikossa olevia solukirjastoryhmiä.

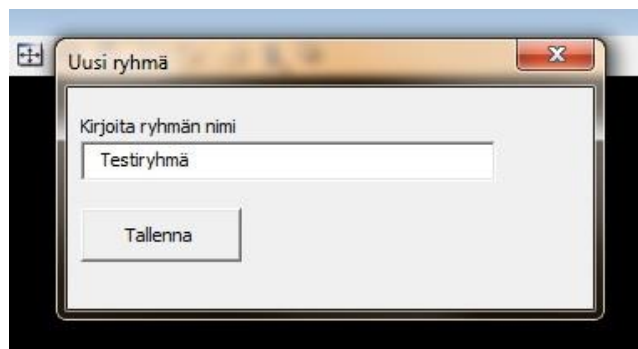
7.2.3.1 Soluryhmien lisäys

1. Käynnistä MicroStation V8i työpöydän tai käynnistysvalikon pikakuvakkeesta, valitse käyttäjäksi amkadmin ja avaa piirustus tai tee uusi.
2. Kun lisäät uuden ryhmän, valitse Hallinta-valikosta Soluryhmät ja Uusi ryhmä (kuva 36).



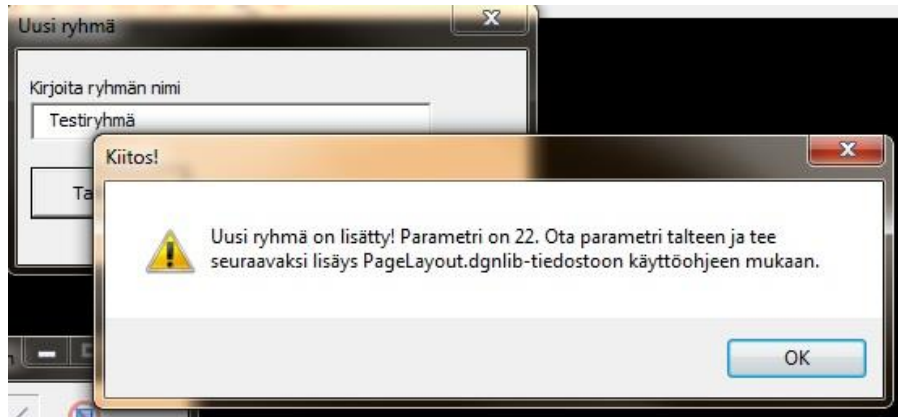
Kuva 36. Soluryhmän lisäys.

Kirjoita ryhmän nimi ja paina Tallenna-painiketta (kuva 37).



Kuva 37. Ryhmän nimi.

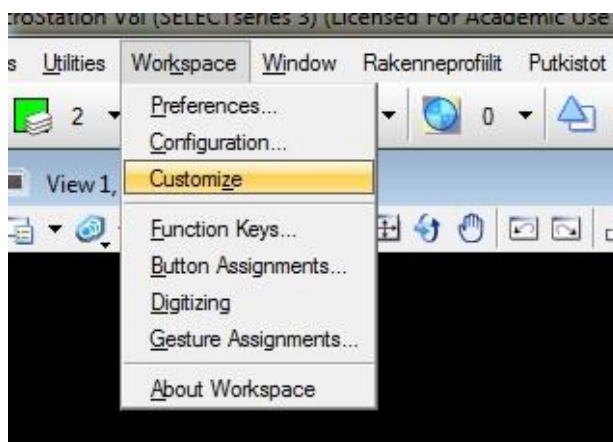
Ohjelma ilmoittaa tietokantaan lisätyn soluryhmän parametrin, ota se talteen sillä tarvitset sitä jatkossa (kuva 38). Voit myös tarkistaa parametrin myöhemmin tehtävään varatulla työkalulla.



Kuva 38. Soluryhmän parametri ohjelman suorittamista varten.

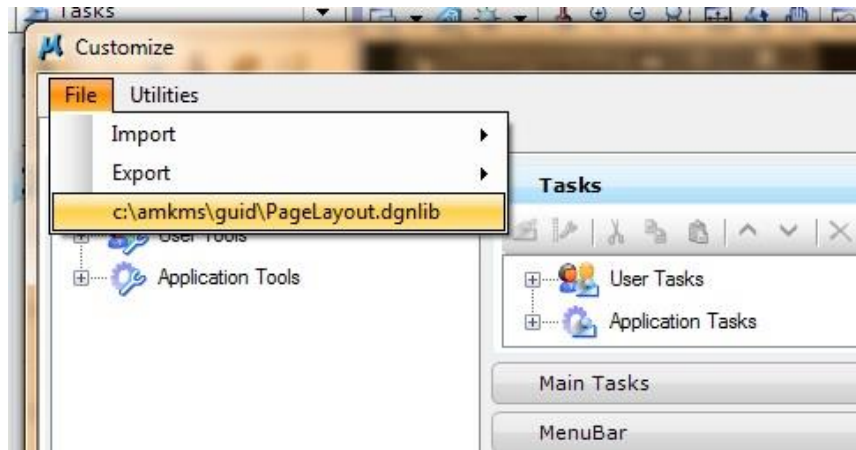
3. Sulje MicroStation, avaa uusi työpöydän tai käynnistysvalikon kuvakkeesta ja valitse käyttäjäksi amkuser. Seuraavaksi tehdään menumuutos MicroStationin käyttöympäristöön.

Valitse ylävalikon Workspace-valikosta Customize-toiminto (kuva 39).



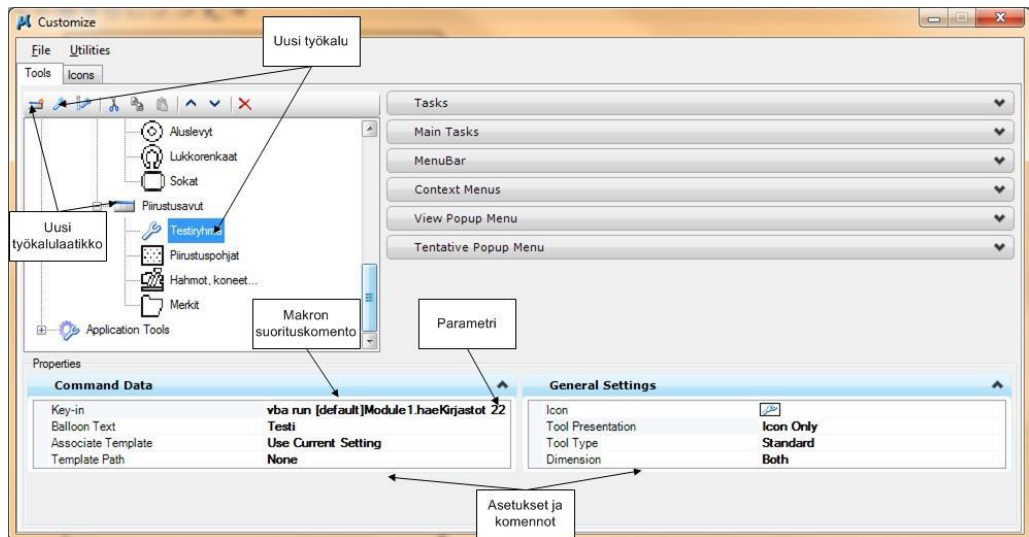
Kuva 39. Käyttöympäristön muokkaus.

Valitse avautuvan Customize-ikkunan File-valikosta PageLayout.dgnlib-tiedosto. Mikäli olet oppilaitoksen verkossa, tiedosto avautuu suoraan palvelimen j:\Learning_Material\Laboratories\Bentley\amkms\guid -kansioista. Kotikoneella tiedosto avautuu kuvan mukaisesta (kuva 40) osoitteesta ja se täytyy muokkauksen jälkeen kopioida palvelimen kansioon.



Kuva 40. PageLayout-tiedoston valinta.

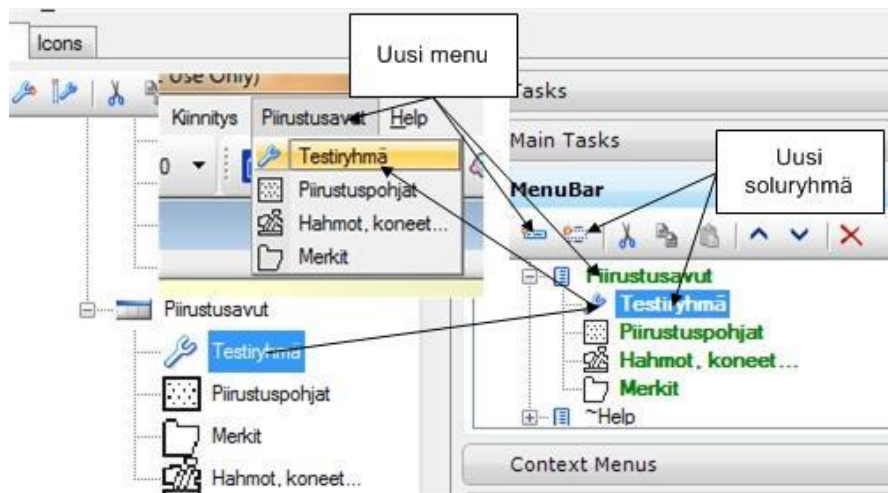
Uusi soluryhmä lisätään Tools-välilehdellä New Tool -toiminnolla (kuva 41). Hallinnan helpottamiseksi voidaan myös tehdä uusia työkalulaatikoita New Tool Box -toiminnolla. Key In -kohtaan kirjoitetaan, tai kopioidaan edellisestä, makron suorituskomento. Parametrin paikalle vaihdetaan tallennuksen yhteydessä saatu parametri. Muut asetukset voidaan tehdä kuvan mukaisesti. Icons-välilehdellä voidaan tuoda Import-käskyllä uusia ikoneita sovellukseen ja ne kytketään Tools-välilehdellä solukirjastoon.



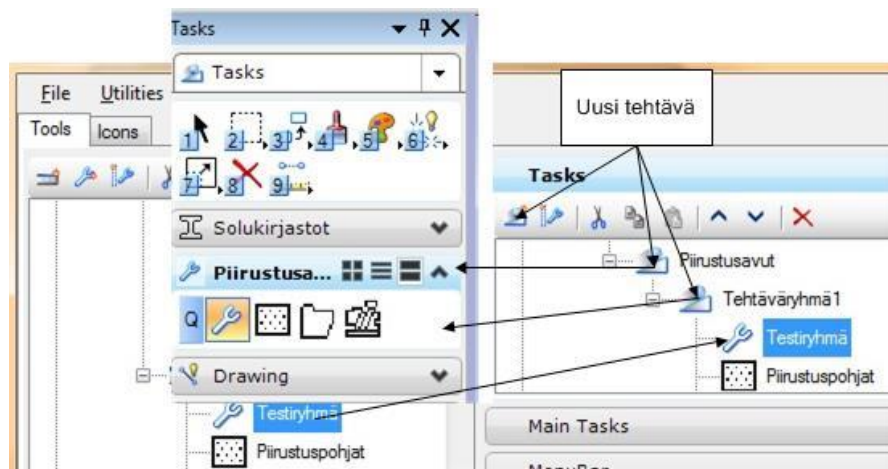
Kuva 41. Uuden tehtävän luonti solukirjaston avaamiseksi.

Tämän jälkeen uusi solukirjasto raahataan hiirellä Tasks ja MenuBar -toimintoihin. Kyseisissä toiminnoissa voidaan tehdä myös erikseen tehtäviä ja

menuvalikoita, mutta tässä esitetysti toimintoja ei tarvitse muuttaa kuin pelkästään Tools -välilehdellä. Ennen raahaamista rakennetaan tehtävä- ja menuhierarkia seuraavien kuvien mukaisesti (kuvat 42 ja 43). Kuvissa on esitetty myös luontitoiminnot ja mihin ne vaikuttavat käyttöympäristössä.



Kuva 42. Ylämenun luonti.



Kuva 43. Tehtävän luonti vasempaan valikkoon.

Customize-ikkunassa on myös paljon muita toimintoja, mutta tämä käyttöympäristösovel-
vellus tulee toimeen edellä esitetyillä toimenpiteillä.

7.2.3.2 Soluryhmien muokkaus ja poisto

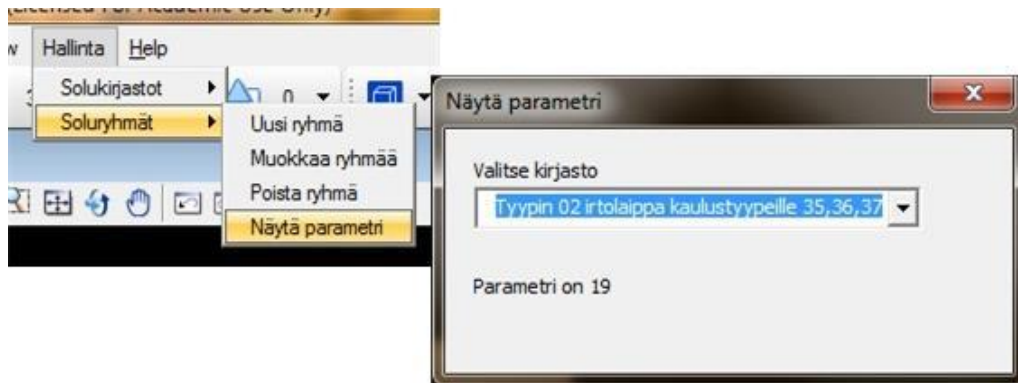
Muokkaus, eli nimen muuttaminen, ja poisto suoritetaan edellisessä kohdassa esitetyillä työkaluilla ja vastaavassa järjestyksessä.

1. Käynnistä MicroStation V8i työpöydän tai käynnistysvalikon pikakuvakkeesta, valitse käyttäjäksi amkadmin ja avaa piirustus tai tee uusi.
2. Valitse hallinta-valikosta kohta Soluryhmät ja Muokkaa ryhmää -toiminto.
3. Valitse ryhmä alasetoivalikosta ja kirjoita uusi nimi sille osoitettuun tekstikenttään ja tallenna muutos.
4. Sulje MicroStation, käynnistä uusi, valitse käyttäjäksi amkuser ja avaa piirustus tai tee uusi.
5. Tee käyttöympäristömuutokset kappaleen 7.2.3.1 kohdassa 3 esitetyillä työkaluilla.

Poistaminen suoritetaan vastaavalla tavalla tietokannasta ja käyttöympäristöstä.

7.2.3.3 Parametrin haku

Mikäli parametri unohtui tai sitä ei ennätetty ottaa lisäsvaiheessa talteen, se voidaan katsoa Hallinta-työkalun Näytä parametri -toiminnolla (kuva 44).



Kuva 44. Parametrin näyttäminen.

7.2.4 Kotipaketin luonti

Kotipaketti on luotu valmiiksi j:\Learning_Material\Laboratories\Bentley\amkms\kotikone-kansioon. Tämän kansion oppilaat voivat kopioida tikulleen ja asentaa omalle koneelleen.

Kotikone-kansio sisältää:

1. Kaikki .cel-päätteiset solukirjastot
2. kotikoneelle.bat-tiedoston

3. mstokem.mdb-tietokannan
4. amkuserHome.ucf-tiedoston
5. Test1.mvba-makrotiedoston
6. PageLayout.dgnlib-tiedoston

Kun kotikone.bat suoritetaan, se rakentaa hakemistot ja kopioi tiedostot käyttäjän koneelle. Huomioi, että aina kun muutoksia tehdään verkossa oleviin kirjastoihin tai käyttöympäristöön, muuttuneet tiedostot täytyy kopioida kotipakettiin ja jakaa oppilaille. Uudelleen asentaminen korvaa vanhat tiedostot.

8 POHDINTA

Tehtäväkenttä oli siis avoin ja osoittautui todella mielenkiintoiseksi. Alkuperäinen ajatus VBA-koodilla ohjatusta menurakenteesta osoittautui jo alkutaipaleella käyttökelvottomaksi. Samoin toimintojen seed-tiedoston kautta ohjaaminen oli mahdotonta ja myös VBA:lla ohjatut Configuration Variables -muuttujat olivat testatessa huonosti käytännön sopivia. Riittävän monen vuokaavion, yrityksen ja erehdyksen kautta käyttöympäristö kuitenkin alkoi hahmottua ja realisoitua. Viimeiset parannukset ja korjaukset tehtiin pääkäyttäjän työn helpottamiseksi työn lopussa. Tietokannan avaus ja solukirjastojen sijainti muutettiin ucf:stä käsin toimivaksi alkuperäisen VBA-koodin ja tietokantaohjauksen sijaan. On siis selvää, että kehittämiskohteita sovelluksesta löytyy edelleen. Eräs idea jatkokehittäjille olisi tehdä hallintatyökalu solukirjastojen tallentamiseen ja lataamiseen, jolloin ohjelmakoodi voisi lukea kirjaston nimen tiedostosta ja merkitä sen tietokantaan. Tällä tavalla taas yksi mahdollinen käyttäjän virhe saataisiin eliminoitua.

Alkuperäiset tavoitteet ja määrittelyt täyttyivät kuitenkin kaikilta osin. Käyttöympäristössä voidaan valita helposti haluttu solukirjasto ja lisätä osa piirustukseen, päivitys onnistuu keskitetysti, reaaliaikaisuus toimii jne. Konsepti voidaan kopioida helposti myös yrityskäyttöön, konekohtaiset räätälöinnit voidaan siirtää syrjään ja aloittaa keskitettyjen resurssien käyttö ilman lisäosainvestointeja. Suunnittelutyön yhteydessä kehitettiin myös kotipaketti, joka alun perin ei kuulunut tehtävänantoon. Sen avulla käyttöympäristön siirtäminen konekohtaiseksi onnistuu vaivattomasti, mutta toivottavasti resurssija hyödynnetään pääasiallisesti verkkoympäristössä alkuperäisen ajatuksen mukaisesti.

Opetuskäytössä oppilaat tulevat saamaan todellisemman rajapinnan työelämään, ja myös ymmärryksen MicroStationin skaalautuvuudesta ja räätälöitävyydestä. Niin ikään vaativimpien harjoitustöiden tekeminen helpottuu, kun kaikkia ruuveja ja laippoja ei tarvitse piirtää alusta pitäen, vaan ne voidaan valita M- ja DN-koolla suoraan solukirjastosta. Perusympäristö saadaan helposti käyttöön valitsemalla oletuskäyttäjä tiedoston avauksessa, perusasiat on joka tapauksessa aina opittava.

Käyttöympäristö on testattu ja asennettu koulun verkkoympäristöön. Toivottavasti siitä saadaan tulevaisuudessa irti se hyöty, mitä lähdettiin hakemaan.

LÄHTEET

- Bentley www-sivut 2013, hakupäivä 15.10.2013. <www.bentley.com>.
- Bentley Communities www-sivut 2013, hakupäivä 7.10.2013. <communities.bentley.com>.
- Ecado www-sivut 2000 - 2013, hakupäivä 1.10.2013. <www.ecado.co.uk>.
- Kemi-Tornion amk:n www-sivut 2013, hakupäivä 30.9.2013. <www.tokem.fi>.
- Kemi-Tornio amk:n opiskelijoiden www-sivut 2013, hakupäivä 30.9.2013. <edu.tokem.fi>.
- Koulutuskuntayhtymä Lappian www-sivut 2013, hakupäivä 30.9.2013. <www.lappia.fi>.
- Krishnan, G.V. & Taylor, James E. 2011. Harnessing MicroStation V8i. Delmar: Cengage Learning.
- Lapin amk:n www-sivut 2013, hakupäivä 30.9.2013. <www.lapinamk.fi>.
- Weisberg, David.E. The Engineering Design Revolution: The People, Companies and Computer Systems That Changed Forever the Practise of Engineering, hakupäivä 15.10.2013. <www.cadhistory.net>.
- Winters, Jerry 2006. Learning MicroStation VBA. Exton: Bentley Institute Press. vt100.net -www-sivu, hakupäivä 7.10.2013. <www.vt100.net>.

LIITTEET

- Liite 1. Käyttäjän ohjeet
- Liite 2. Pääkäyttäjän ohjeet