

Tapani Liukkonen

METSÄTYÖKONEEN TYÖOHJELMA- JA KARTTAJÄRJESTELMÄ

Opinnäytetyö

Kajaanin ammattikorkeakoulu

Hallinnon ja kaupan ala

Tietojenkäsittelyn koulutusohjelma

Syksy2002



Kajaanin
ammattikorkeakoulu

OPINNÄYTETYÖ TIIVISTELMÄ

Ala Hallinnon ja kaupan ala	Koulutusohjelma Tietojenkäsittelyn koulutusohjelma
Tekijä(t) Tapani Liukkonen	
Työn nimi Metsätyökoneen työohjelma- ja karttajärjestelmä	
Vaihtoehtoiset ammattiopinnot	Ohjaaja(t) Synnöve Hämäläinen
Aika Syksy 2002	Sivumäärä 28 + 6 liitteet
<p>Tiivistelmä</p> <p>Työn tarkoituksena oli Gisforest Oy:n uuden karttakomponentin testaus ja komponenttipohjaisen työkonejärjestelmän perusrungon kehitys tuleville projekteille. Työn tilaaja oli Gisforest Oy.</p> <p>Työn teoriaosaan kuului vanhempiin vastaaviin järjestelmiin ja valmiisiin komponentteihin tutustuminen, lisäksi materiaaliin kuului komponenttien valmistajien ohjekirjat ja yleisesti Windows - ympäristössä tapahtuvaa komponenttiohjelmointia käsittelevä kirjallisuus. Käytännön osuuteen kuului järjestelmän ohjelmointi ja uuden karttakomponentin testaus käytännössä.</p> <p>Työn tuloksena syntyi ohjelmistorunko kevyille karttajärjestelmille joiden ominaisuuksiin kuuluu kyky käsitellä erilaisia karttamateriaaleja ja olennaisena osana kyky lisätä materiaaliin omia merkintöjä. Työn ohessa testattu karttakomponentti osoittautui nopeaksi ja resursseja tuhlaamattomaksi. Järjestelmä toteutettiin Microsoft Visual Basicilla. Karttakomponentin perusosana toimi MapX - komponentti ja tietokanta toteutettiin Microsoft Access - muodossa.</p>	
Luottamuksellisuus	Kyllä
Hakusanat	MapX, komponenttiohjelmointi, Visual Basic, SQL
Säilytyspaikka	Kajaanin Ammattikorkeakoulun kirjasto



**Kajaanin
ammattikorkeakoulu**

ABSTRACT OF THE FINAL YEAR PAPER

Faculty Administration and Business	Degree programme Data Processing
Author(s) Tapani Liukkonen	
Title Work Instruction and Mapping System for Forestry Machinery	
Alternative professional studies	Instructor(s) Synnöve Hämäläinen
Date Autumn 2002	Total number of pages 28 + 6 appendices
<p>Abstract</p> <p>The purpose of this final year project was to test the new mapping component made by Gisforest Oy and to develop a framework of component based work machine system for future projects. This project was commissioned by Gisforest Oy.</p> <p>The theoretical part of the project included familiarizing with the older, comparable systems and components manufactured by other companies; the material also contained manuals and instructions for these components. Literature concerning component based programming on the Windows environment was also studied. The practical part of the project contained the programming of the system and testing of the new mapping component.</p> <p>As a result of the project, a framework for efficient mapping systems with the ability to handle different kinds of mapping materials and basic drawing capabilities was created. In addition to the development of the system, the new mapping component was tested and it proved to be fast and efficient with resources. The system was implemented by using the Microsoft Visual Basic, the main body for the mapping capabilities was made using the MapX component and the database utilises Microsoft Access.</p>	
Confidentiality status	Yes
Keywords	MapX, component programming, Visual Basic, SQL
Deposited at	Kajaani Polytechnic Library

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

SISÄLLYSLUTTELO

<u>1</u>	<u>JOHDANTO</u>	1
<u>2</u>	<u>PAIKKATIETO-OHJELMISTO</u>	2
<u>3</u>	<u>TYÖKALUT</u>	4
<u>3.1</u>	<u>Ohjelmointikieli</u>	4
<u>3.2</u>	<u>Tietokanta</u>	5
<u>3.3</u>	<u>MapX</u>	5
<u>3.4</u>	<u>InstallShield Express 3.5</u>	6
<u>4</u>	<u>KOMONENTTIOHJELMOINTI</u>	7
<u>5</u>	<u>TYÖN TAVOITTEET JA MÄÄRITYKSET</u>	9
<u>6</u>	<u>JÄRJESTELMÄN SUUNNITTELU JA TOTEUTUS</u>	11
<u>6.1</u>	<u>Suunnittelu</u>	11
<u>6.2</u>	<u>Toteutus</u>	12
<u>6.3</u>	<u>Testaus</u>	14
<u>7</u>	<u>OHJELMAN TOIMINTA</u>	16
<u>7.1</u>	<u>Työkalupakki</u>	16
<u>7.2</u>	<u>Valikot</u>	18
<u>7.3</u>	<u>Piirtotyökalut</u>	19
<u>8</u>	<u>YHTEENVETO</u>	21

LÄHTEET

LIITTEET

KÄSITELUETTELO

ActiveX	Microsoft Windows:ssa käytettävä komponenttitekniikka
ADO	ActiveX Data Objects, Microsoftin tietokannan käsittelyyn tarkoitettu rajapinta
debug	Ohjelman virheiden jäljitys
DGPS	Differential Global Positioning System, satelliitteihin ja maamajakkaan perustuva paikanmäärittäjäjärjestelmä
DLL	Dynamic Link Library, ajonaikana tarvittaessa käyttöön ladattava komponentti
GIS	Geographical Information System, paikkatietojärjestelmä
GPS	Global Positioning System, satelliittipaikannusjärjestelmä
IDE	Integrated Development Environment, kehitysympäristö
MapX	MapInfo Corporationin karttakomponentti
OCX	ActiveX Control, tietyn tehtävän suorittava komponentti, jota voi käyttää eri projekteissa.
Projektio	Matemaattinen malli, joka muuntaa kohteiden sijainnit kolmiulotteiselta pallopinnalta sijainneiksi kaksiulotteiselle tasolle, esimerkiksi paperikartalle
Relaatio	Kahden taulun välinen suhde, jonka perusteella tauluja voidaan liittää yhteen
SQL	Structured Query Language, relaatiotietokantojen hallinto- ja käsittelykieli

1 JOHDANTO

Tämän työn toimeksiantaja oli Gisforest Oy (liite 4). Gisforest Oy on vuonna 1990 perustettu metsäteollisuuden metsätietojärjestelmiin erikoistunut ohjelmistotalo.

Opinnäytetyöni kohteena olivat komponenttipohjaisen karttajärjestelmän ohjelmointi Windows-ympäristössä sekä uuden karttamoottorin testaus. Järjestelmän työnimenä toimi "Metsätyökoneen työohjelma- ja karttajärjestelmä". Nimensä mukaisesti ohjelma pyörittää erilaisia työohjeita sekä kartta-aineistoja, mahdollistaen omien merkintöjen ja huomioiden kirjaamisen kartalle. Jatkossa ohjelma toimii testijärjestelmänä ja runkona mahdollisille siihen perustuville tuotteille, joissa tarvitaan muokattavuutta ja vakiokomponenttien käyttöä nopeuttamaan järjestelmän kehitystä asiakkaan tarpeita vastaavaksi räätälöidyksi tuotteeksi.

Järjestelmä on toteutettu käyttäen MapInfo Corporationin MapX-karttakomponenttia ja Gisforestin omia komponentteja. Käyttöliittymä ja omat komponentit on tehty käyttäen ohjelmointikielenä Microsoft Visual Basicia. Tietokantana toimii Microsoft Access -kanta.[1,2]

2 PAIKKATIIETO-OHJELMISTO

Paikkatieto-ohjelmisto GIS (Geographical Information System) on tietojärjestelmä, joka käsittelee paikkaan sidottua tietoa. Tietoihin kuuluvat olennaisesti sijaintitiedot ja ominaisuustiedot.

Sijaintitieto tarkoittaa koordinaatti-, geometria- tai topologiatietoa. Ominaisuustieto puolestaan voi olla yksilöivää, paikantavaa, ajoittavaa tai kuvailevaa. Ohjelmissa paikkatieto esitetään usein karttana, mistä käyttäjä voi valita alueita tai kohteita tarkasteltavaksi. Useissa ohjelmissa on myös mahdollista lisätä tai muokata tietoa.

Karttoja on kahta tyyppiä, maantieteellisiä ja ei-maantieteellisiä. Ei-maantieteellinen kartta sisältää kohteita, joiden sijaintia maapallolla ei ole määritelty. Esimerkkeinä tämän kaltaisista kohteista käyvät rakennusten pohjapiirrokset. Vastaavasti maantieteellisissä kartoissa kohteilla on tietty sijainti maapallolla.

Ei-maantieteellisillä kartoilla on koordinaattijärjestelmä, mutta ei projektiota. Maantieteellisissä kartoissa on koordinaattijärjestelmä, tyypillisesti pituus- ja leveyspiireihin perustuva, mutta myös projektio.

Koordinaattijärjestelmällä tarkoitetaan parametreja, joiden avulla kohteen sijainti paikallistetaan. Erilaisia koordinaattijärjestelmiä on käytössä useita eri puolilla maapalloa. Kaikki koordinaattijärjestelmät vääristävät jotain kartan ominaisuuksista, kuten pinta-alojen

tai etäisyyksien suhteita. Syynä tähän on karttojen siirto kolmiulotteiselta pallolta näytön tai paperikartan tasolle. Näitä vääristymiä pyritään pienentämään projektiolla.[3,4]

Projektioita käyttämällä voidaan vähentää karttakohteiden vääristymistä tasopinnalle siirtämisen yhteydessä. Projektioita on useita erilaisia, joista jokainen on suunniteltu pitämään jokin kartan ominaisuuksista vääristymättömänä. Projektiota käytetään usein synonyymina puhuttaessa koordinaattijärjestelmästä, vaikka projektiio on vain yksi koordinaattijärjestelmän parametreista.[3,4,5,6]

Nykyään paikkatieto-ohjelmistoihin on usein liitetty GPS-laitteen käyttömahdollisuus. GPS (Global Positioning System) on Yhdysvaltain puolustusvoimien ylläpitämä satelliittipaikannusjärjestelmä. Järjestelmän osia ovat 24 maata kiertävää satelliittia ja käyttäjällä oleva GPS-laitte. Siviilikäytössä olevien laitteiden tarkkuus on noin 15 metriä olosuhteista riippuen.

DGPS (Differential-GPS) korjauksen avulla laitteen tarkkuudeksi saadaan noin 1 – 5 metriä. DGPS:ssä GPS-laitte saa satelliittien lähettämän signaalin lisäksi signaalin esimerkiksi radiomajakasta. Koska radiomajakakan tarkka sijainti tunnetaan, voidaan tämän signaalin tietojen avulla laskea sijainti tarkemmin kuin pelkän satelliittisignaalin avulla. DGPS-palvelua tarjoaa Suomessa muun muassa YLE.[7]

GPS-laitteen avulla voidaan seurata esimerkiksi ajoneuvon liikettä reaaliajassa kartalla tai siirtää ohjelmaan GPS-laitteen muistista reitti- ja aluetietoja.

3 TYÖKALUT

Työssä käytetyistä ohjelmointikielistä, komponenteista ja hyväksikäytetyistä ohjelmista tai muista ratkaisuista esitellään tärkeimmät. Visual Basicin yksittäisiä peruskomponentteja ei esitellä erikseen, koska niitä tulee pieneenkin projektiin helposti useita ja niiden tuntemus työn kannalta ei ole ratkaisevaa.

3.1 Ohjelmointikieli

Visual Basic on Microsoftin kehittämä basic-murre, joka sisältää olio-ominaisuuksia ja läheisen integraation Windows-käyttöjärjestelmän toimintojen ja kirjastojen kanssa. Visual Studio-paketti, johon Visual Basic kuuluu, sisältää graafisen kehitysympäristön (IDE), jossa yhdistyvät koodieditori, debuggeri ja kääntäjä. Ohjelman voi siis kirjoittaa, testata ja kääntää samassa ympäristössä ilman apuohjelmien käyttöä. Lisäksi kehitysympäristö tarjoaa aputoimintoja, kuten koodia kirjoittaessa ilmestyvän apuikkunan, jossa kerrotaan sillä hetkellä kirjoitettavana olevan komennon mahdollisten parametrien määrä ja tyypit. Testauksen aikana ohjelmaan voidaan asettaa vahteja ja pysähtymiskohtia, joiden avulla voidaan tarkkailla muuttujien ja olioiden tiloja ohjelman suorituksen aikana, joko reaaliajassa tai pysähtyneessä tilassa.

3.2 Tietokanta

Microsoft Access 2000:een perustuva tietokanta on relaatiotietokanta. Relaatiotietokanta on järjestetty kokoelma toisiinsa liittyviä tietoja. Relaatiomallien pohjana on E. F. Codd:n vuonna 1970 julkaisema "A Rational Model of Data for Large Shared Data Banks" (Communications of the ACM, Vol. 13, No. 6, June 1970).

Tietokanta muodostuu tauluista, joista kukin sisältää kokoelman yhteen aiheeseen liittyvää tietoa. Kaikki tieto tietokannassa on tallennettu tauluihin. Jokaisessa taulussa tulisi olla perusavain, jonka avulla tietueet yksilöidään. Perusavain muodostuu yhdestä tai useammasta avainkentästä, joiden perusteella taulun tiedot voidaan yksiselitteisesti tunnistaa.

Taulujen vaakasuorista riveistä käytetään nimitystä tietue, vaihtoehtoinen nimi on rivi. Tietue pitää sisällään tiedot yksittäisestä tallennetun tiedon kohteesta. Taulujen pystysuorat rivit ovat sarakkeita. Sarakkeet sisältävät yhdenlaista tietoa, esimerkiksi "nimi"-sarake sisältäisi nimiä, jotka kaikki olisivat tietotyyppiä char.

Yhteyksien avulla muodostetaan yhteys eri taulujen välille samaa tietoa sisältävien kenttien avulla. Viiteavain sisältää samaa tietoa kuin yhteyden toisessa päässä oleva perusavain. Viiteavain toimii linkkinä taulujen välillä.

3.3 MapX

MapX on MapInfo Corporationin kehittämä ja valmistama ActiveX-karttakontrolli (OCX). Kontrollin avulla esitetään sovelluksissa muodoltaan erilaisia karttoja. Karttoihin voidaan lisätä uusia tasoja, joiden avulla voidaan toteuttaa esimerkiksi piirto-ominaisuuksia. Kartat voivat myös sisältää tietoa, kuten asukasmäärät, joista saadaan muodostettua graafiset esitykset vaikkapa asukastiheydestä eri alueilla. Näitä tietoja säilytetään MapInfon omassa karttatietokannassa. MapX:n avulla ohjelmiin saadaan helposti upotettua karttaominaisuudet ja työkalut näiden ominaisuuksien hyödyntämiseen.[1,2]

3.4 InstallShield Express 3.5

Tämä InstallShield Software Corporationin kehittämä ohjelma helpottaa asennuspakettien tekoa valmiista ohjelmistoprojekteista. Asennuspakettien avulla saadaan käyttäjälle helppo ja vaivaton tapa asentaa valmis tuote koneelle. InstallShield huolehtii rekisterimuutoksista ja tiedostojen kopioinnista paikoilleen säästäten käyttäjää vaivalloiselta käsityöltä ja ohjelmiston kehittäjän oman asennusohjelman teolta. Myöhemmin päivitykset saadaan asennettua yhtä vaivattomasti yhdellä työkalulla ohjelman huolehtiessa vanhojen tiedostojen ja merkintöjen poistosta ja korvaamisesta.

4 KOMPONENTTIOHJELMOINTI

Komponenttiohjelmoinnilla pyritään luomaan yleiskäyttöisiä ohjelman osia. Komponentti on eri sovelluksissa uudelleenkäytettävä ohjelmamoduuli, jossa on yhden tai useamman luokan toteutus. Komponentteja käytetään komponentin tunnetun rajapinnan kautta ja tämän rajapinnan pitäisi pysyä samana, jotta eri versiot ja ohjelmat olisivat yhteensopivia ilman laajempia muutoksia. Komponentin sisäistä toteutusta voidaan muuttaa vapaasti niin pitkään kuin tämä tunnettu rajapinta pysyy samana.

Komponenttipohjaisen ohjelmistokehityksen sanotaan usein nopeuttavan ohjelmien kehitystä, koska ohjelmat voidaan nopeasti koota testatuista vakiokomponenteista. Käytännössä tämänkaltaisia nopeasti käytettäviä komponentteja ovat lähinnä rajattua tehtävää suorittavat komponentit. Monimutkaisemmat komponentit joudutaan lähes aina joka tapauksessa räätälöimään sovelluskohtaisesti.

Käytännössä yleispätevien komponenttien kirjoittaminen on hankalaa. Etenkin, jos ohjelman kehitys joudutaan aloittamaan ilman tarkkoja määrittäyksiä. Tämä lisää komponentin ja koko ohjelman välistä riippuvuutta, ja lopulta komponentin siirto suoraan toiseen ohjelmaan on hankalaa ja aikaa vievää. Käytäntö on myös opettanut sen, että yleensä komponentista tulee hyvä vasta noin kolmanteen versioon mentäessä. Tämän jälkeen komponentin rajapinta on jo niin vakiintunut, ettei siirrossa ole enää muutoksista johtuvia ongelmia.

Komponentin tekoon vaikuttaa suoraan se, kuinka hyvin sen rajapinnat on suunniteltu. Kun suunnitelmat ovat tarkat ja useaan kertaan läpikäytyt, niin rajapinnasta voidaan heti alussa tehdä vakaa. Jatkossa komponenttia voidaan käyttää eri ohjelmissa ilman komponentin

rajapintojen eri versioiden aiheuttamia muutoksia muiden ohjelmien sisäisiin rakenteisiin. Suunnitteluun käytetyllä ajalla voidaan siis säästää ohjelmistojen kirjoittamiseen kuluva aikaa.

Ohjelmisto voidaan myös jakaa loogisiin osiin, joista muodostetaan erillisiä komponentteja. Tällä tavalla ohjelman teko voidaan hajauttaa useammalle eri henkilölle, joista jokaisella on oma vastuualue. Ryhmätyössä suunnittelun asema korostuu entisestään. Määritysten täytyy olla eri osien osalta siinä vaiheessa, että useamman ihmisen on mahdollista alkaa kehittämään erillisiä komponentteja, joista muodostuu loppujen lopuksi yksi toimiva ohjelma. Määritysten ollessa vajaita tai yhteen sopimattomia, joudutaan myöhemmin suorittamaan työlästä komponenttien yhteensovittamista ja useamman eri ohjelman osan korjausta.

Ylläpito ja päivitys voivat myös hyötyä ohjelman jakamisesta komponentteihin. Ohjelman koostuessa komponenteista voidaan virheiden ja virheilmoitusten perusteella päätellä, missä osassa ohjelmaa virhe on. Tämän jälkeen voidaan tutkia vain osaa ohjelmistokoodista sen sijaan, että käytäisiin koko ohjelmaa lävitse. Vian löytyminen on siis nopeampaa, ja korjauksen jälkeen riittää yhden komponentin päivitys asiakkaan koneilta. Etenkin mobiilisovellusten puolella, jossa päivitysten siirtotienä voidaan käyttää vaikkapa langattomia puhelinyhteyksiä, saavutetaan lisähyötyjä päivityspakettien pysyessä pieninä. Siirron vaatima siirtoaika ja asennukseen kuluva aika pienenevät, joten kustannukset pysyvät pienempinä.

Komponentin valmistumisen jälkeiseen hyötykäyttöön vaikuttavat suoraan dokumentaatiot. Ilman ajan tasalla pysyvää dokumentaatiota komponenttien käytöstä tulee hidasta ja hankalaa tai sitä ei käytetä ollenkaan, koska kukaan ei tiedä miten ja mitä komponentti tekee. Usein edes komponentin kehittäjä ei enää muista komponentin käyttötapaa ja -tarkoitusta, varsinkaan jos komponentin valmistumisesta on jo kulunut aikaa. Näiden seikkojen takia ohjelmistokehittäjien pitäisi muistaa, että komponentin dokumentointi on heidän vastuullaan. Syinä vajavasiin dokumentaatioihin on yleensä ajan puute tai dokumentoinnin vaivalloisuus. Tämän takia monet kehitysympäristöt tarjoavat työkalun, jonka avulla ohjelmistolistauksesta saadaan poimittua esimerkiksi kehittäjän kommentit, aliohjelmakutsut ja muuttujien määrittelyt automaattisesti muodostettavaan dokumentaatioon. Tällaisille työkaluille on myös mahdollista kirjoittaa lisäkommentointeja listaukseen, jotka työkalu kerää ja muodostaa niistä lisätietoja dokumentteihin. [8,9]

5 TYÖN TAVOITTEET JA MÄÄRITYKSET

Työn tavoitteeksi asetettiin komponenttipohjaisen karttajärjestelmän rungon ja esimerkkisovelluksen teko. Järjestelmän käyttöympäristöksi määriteltiin Windows-käyttöjärjestelmää käyttävät koneet, koska työkoneissa on yleisimmin valmiiksi asennettuna jokin Windows-versio. Käyttöympäristön valinnan mukaan ohjelmointikieleksi valittiin Visual Basic 5.0. Karttakomponentiksi otettiin MapInfo Corporationin MapX 4.5. Muita kolmansien osapuolten komponentteja pyritään välttämään, jottei tulisi ylimääräisiä ongelmia, sillä käytännössä on törmätty ongelmiin joidenkin valmistajien ohjelmistokomponenttien kanssa. Poikkeuksena käyttökiellessä on Farpointin ListPro-komponentin käyttö, jonka käyttäminen on sallittua sitä vaativissa kohdissa.

Tukikomponentteina toimivat Gisforestin omaa tuotantoa edustavat gfMapEng ja comLayout, jotka ohjaavat kartan toimintaa ja tulostusta. Paperille tai tiedostoon menevät tulosteet muotoillaan comLayout:lla erilaisiin raportteihin.

Karttojen pyöryksestä sovelluksessa huolehtii gfMapEng. Komponentti tekee yhteistyötä MapX-komponentin kanssa. Tämän komponentin avulla voidaan muun muassa käyttää eräitä MapX:n ominaisuuksista helpommin kuin käyttämällä niitä suoraan. Esimerkiksi koordinaattijärjestelmän vaihdon toteutus, joka suoraan käytettynä tapahtuu seuraavasti: ”NumericCoordSys.Set 8, 29, 7, 27, 0, , , 1, 3500000, 0”, vaatii yksitoista parametria. On huomattavasti helpompaa ja vähemmän virhealtista asettaa yksi muuttuja gfMapEng:stä seuraavalla tavalla: ”mobjEngine.CoordSys.SetNumeric gisLongLat”, joka tämän jälkeen asettaa oikean koordinaattijärjestelmän käyttöön.

Tietokannaksi valittiin Microsoft Access 2000–perusteinen ratkaisu, koska tuki sen käytölle on todennäköisesti valmiina suurimmassa osassa kohteena olevista Windows–koneista. Jos ajuria ei ole valmiina, niin sen hankkiminen ja asennus on ongelmaton. Access:n mdb–tiedostoa voidaan myös tarpeen vaatiessa käsitellä Access–ohjelmalla. Keveys oli myös yksi tekijöistä, joka vaikutti valintaan. Varsinaisella tietokantamoottorilla varustettu relaatiotietokanta vaatisi asennettuna enemmän tilaa ja vaatisi käyttöönsä muistia ja prosessoriaikaa huomattavasti enemmän.

Ohjelman käsittelemän aineiston siirtoon käytetään sähköpostia. Sähköpostin liitetiedostona tuleva aineisto tallennetaan 'c:\tyomaat'–hakemistoon, joka on ohjelman sovittu asennushakemisto. Lähtevä aineisto siirretään joko käsin sähköpostiohjelmassa tai työkone–järjestelmässä automaattisesti hyödyntäen Windowsin sisäistä ohjelmien välistä tiedonsiirtoa. Tällä tavoin saadaan lähetyksestä käyttäjälle vaivaton ja yksinkertainen.

Asennus- ja päivitysvaiheista pyritään tekemään mahdollisimman yksinkertaisia ja vähävaiheisia prosesseja. MapX–komponentin osalta tämä onkin varsin toivottavaa, sillä se sisältää kymmeniä dll- ja fonttikirjastoja. Näiden manuaalinen asentaminen nostaisi ohjelman käyttöönoton kynnyksestä huomattavasti.

Ohjelmiston käyttöliittymässä tavoitellaan helppoutta, yksinkertaisuutta ja muistettavuutta. Järjestelmää on tarkoitus käyttää työntekoon, joten sen käytön opetteluun ei saa kulua tarpeettomasti aikaa.

6 JÄRJESTELMÄN SUUNNITTELU JA TOTEUTUS

Suunnittelusta ja toteutuksesta käydään pääpiirteittäin lävitse kuinka ohjelman kehitys eteni työn aikana, ja miksi joihinkin ratkaisuihin on päädytty. Tietokannan käsittelyn osalta esitellään hiukan komponenttia, jonka avulla käsittelyn perusteet on suoritettu.

6.1 Suunnittelu

Ohjelman suunnittelun aloitti tietokannan ensimmäisen raakaversion rakenteen kehitys. Tämä ensimmäinen versio koki työn edistyessä lähes täydellisen muutoksen. Syynä tähän on se, että varsinaisesti kantaan tuleva tieto selvisi, kun käyttöliittymä oli saatu suhteellisen selville, ja oltiin varmoja siitä millaista tietoa sen kautta halutaan syöttää ja käsitellä. Relaatiomallia on pyritty seuraamaan, mutta siitä on poikettu tarpeen mukaan kannan pitämiseksi yksinkertaisena.

Käyttöliittymän suunnittelun lähtökohtana oli yksinkertaisuus. Ohjelman käytön hankaluus tai monimutkaisuus ei saa olla varsinaisen työtehtävän suorituksen hidasteena tai esteenä. Tämän vuoksi työkalunapeista on tehty kookkaita ja ikonien kuvista toimintoa kuvaavia. Samoin valikkorakenne on pidetty yksinkertaisena, jotta tärkeät asiat eivät katoaisi vaihtoehtojen sekamelskaan.

Näyttöjen ulkonäkö on yksinkertainen, ja toimintojen määrä on pyritty pitämään vähäisenä. Syöttökenttiin tuodaan arvot valmiiksi, jos ne ovat jostain olleet saatavilla. Tällä vähennetään käyttäjältä toistuvan tiedon syöttöä.

Ikkunoiden väliset suhteet olivat tärkeitä alusta saakka. Koska on mahdollista pitää yhtäaikaan auki useita eri ikkunoita, joissa on samaa tietoa, niin ilman synkronointia osa tiedosta voisi olla harhaanjohtavaa. Tämän estämiseksi tiedot päivitetään automaattisesti joka paikasta, jos tieto jossain muuttuu. Näin saadaan tieto pysymään ajan tasalla eri puolilla ohjelmaa.

6.2 Toteutus

Ohjelman teko aloitettiin toteuttamalla tietokanta. Tosin tämä vaihe tehtiin useampaan kertaan säilöttävän tiedon tarkentuessa, muun ohjelman kehityksen mukana. Tietokannan rakenteen eläminen toteutusvaiheen alussa ei haitannut muuta kehitystä, koska alussa ohjelma ei vielä millään tavalla käsitellyt tietokantaa. Toiminnallisuuden lisääntyessä tietokannan muutokset aiheuttivat muutamia muutoksia myös ohjelmakoodiin.

Käyttöliittymän osalta kehitys alkoi kartan upottamisella pääikkunaan. Tätä seurasivat kartan tulostukset paperille ja tiedostoon. Tällä tavoin kaiken perustana oleva toiminnallisuus saatiin heti ohjelmaan mukaan. Piirtotyökalut ja työkalupakki alkoivat seuraavaksi kehittyä rinnakkain. Näiden kahden käyttö vaikuttaa toisiinsa, joten helpointa oli toteuttaa niitä yhtä aikaa. Odotetusti työläämpi oli piirtotyökalut johtuen työkalujen luonteiden erosta. Työkalupakin työkaluista vain yksi vaatii erillisen syöttöikkunan avaamista, piirtotyökaluista lähes kaikki vaativat oman syöttöikkunansa. Esimerkkinä kuvionpiirtotyökalut, joiden syöttöruutu sisältää yksitoista (11) kenttää (liite 1). Näistä kahteen tulee data automaattisesti, eli kuvion pinta-ala ja piiri. Lisäksi kolme kenttää sisältää valintalistan, josta arvot voidaan valita, loput on syötettävä käsin.

Kartan liikuttamiseen ja muokkaukseen käytettävät gMapEng ja MapX 4.5 otettiin käyttöön näiden perustoimintojen valmistumisen jälkeen. Työkalupakin sisältämät perustyökalut, joilla hoidetaan kartan liikuttelu, syntyivät nopeasti. Piirtotyökalujen tekeminen oli enemmän aikaa

ja työtä vaativaa. Jokaisesta kuviosta välitetään piirtämisen yhteydessä tietokantaan kohteen ominaisuustietoja, kuten pituus tai pinta-ala, ja kohteen kartalla yksilöivä avain.

Näitä tietoja käytetään hyväksi muissa ohjelman ominaisuuksissa, kuten Info-työkalussa ja Työohjeen selaus-toiminnossa. Kohteen avaimen avulla voidaan esimerkiksi kohdistaa haluttaessa kartta oikeaan paikkaan, jos näkymä on jossain muualla.

Piirtotyökalujen syöttöikkunoiden teon yhteydessä tehtiin myös valikoista löytyvät syöttöikkunat. Jokainen syöttöikkuna, kuten piirtämisen yhteydessä esiin tulevat syöttöikkunatkin, pitävät takanaan ainakin yhden SELECT-, UPDATE- ja DELETE- lauseen.

Ennen kuin muita tietoja voidaan syöttää ohjelmaan, täytetään perustiedot (liite 2). Näitä tietoja ovat muun muassa yrittäjän tiedot ja puutavarakoodit. Valikosta löytyy syöttöikkuna myös ohjelman käsittelemille metsäkaupoille ja työmaille. Perusajatuksena on, että ensin luodaan kauppa tietoineen (liite 3) ja tälle lisätään tämän jälkeen työmaa tai useita työmaita. Kun työmaat on asetettu, voidaan niille lisätä kartan piirto-ominaisuuksilla työmaakuvioita ja niihin liittyvää muuta tietoa, kuten reittejä ja varastopaikkoja. Näiden tietojen syötössä on kuitenkin annettu käyttäjälle mahdollisuus jättää kaupat kokonaan pois, ja keskittyä pelkkien työmaiden tekoon. Kartalle voi piirtää kohteita ennen työmaita, sillä ne voidaan myös jälkikäteen liittää työmaihiin päivittämällä niiden tietoja.

Kauppojen ja työmaiden tietojen poistoon liittyy ohjelman ainoa monimutkaisempi SQL-lauseke, jossa on useita SELECT ja DELETE-käskyjä. Jos poistaa kaupan, niin kaupalle kuuluvat työmaatkin poistuvat, samoin kuin kaikki poistuvalla työmaalle kuuluvat tietokannassa olevat tiedot ja kartalla olevat merkinnät. Ilman tämän kaltaista toimintoa kantaan jäisi ylimääräistä tietoa, jonka joutuisi poistamaan useamman eri valikon ja toiminnon kautta. Tämän voisi toteuttaa myös tietokannan ominaisuuksilla, mutta tietojen syöttö saatiin tällä tavalla joustavammaksi.

Järjestelmä käyttää tietokantaa Microsoftin ADO-rajapinnan kautta. Tämä on vakiokomponentteja tietokantaratkaisujen toteuttamiseen Windows-pohjaisissa järjestelmissä. Yhteyden luonti tapahtuu keskitetysti yhdessä paikassa, ja myöhemmin tämä yhteys välitetään sitä tarvitseville komponenteille. Näin vältytään tietokantayhteyden jatkuvalta

uudelleenmäärittelyltä ja -luonnilta ohjelmalistauksen eri kohdissa. Muutoksien tekeminen on jatkossa helpompaa, koska muutoksia ei tarvitse tehdä eri puolille ohjelmaa.

SQL-lauseet, joita ohjelmassa käytetään, ovat yksinkertaisia ja yleisluontoisia. Lähinnä uusia tehdessä piti vaihdella kohteena olevien taulujen nimiä, ja tehdä tietorakenteita käsiteltävälle tiedolle.[10]

Tulostuskomponentti on käytössä vain pääohjelmalla, mutta suurimman osan tulosteista tuottavat muut komponentit. Tämän takia komponentit luovat väliaikaisia tulostustiedostoja, joiden syntymisestä pääohjelmalle välitetään tieto. Kun pääohjelma saa tiedon tulostamisesta, se lukee väliaikaistiedoston ja muotoilee sen sisällön käyttäen comLayout-komponenttia. Tällä tavoin on muodostettu jokainen ohjelman tuottama raportti. Raporttien ulkoasu on viimeistelemätön johtuen ohjelman testiluonteesta. Lopullinen ulkoasu on helppo muotoilla, kun käytössä on oikeaa tietoa.

6.3 Testaus

Kokonaisvaltaista testausta ohjelmalle ei ole suoritettu, koska se ei ole valmis ohjelmistotuote. Näin ollen olisi ollut turhaa testata asioita, jotka on tehty vain siksi että ohjelmasta saisi toimivan kehitysversion. Sen sijaan on keskitytty testaamaan erillisiä komponentteja ja toimintoja kehitystyön sivussa virheiden karsimiseksi. Vasta mahdollisen tuotteistuksen ja jatkokehityksen aikana voidaan ohjelman testaus suorittaa täysimittaisesti.

Suurin osa tässä vaiheessa suoritetusta testaamisesta on liittynyt syötetyn tiedon tarkastukseen ja käsittelyyn. Erilaisten lukujen ja tekstimuotoisten tietojen, kuten y-tunnus ja sähköposti-osoitteet, arvoalueiden ja oikeellisuuden tarkistuksia on testattu erilaisilla syötteillä, jotta voitaisiin varmistua siitä, ettei tietokantaan pääse virheellistä tietoa.

Monimutkaisin osa testauksesta oli muiden syötettyjen tietojen tarkistamisessa eli kartalle piirrettävissä kohteissa. Piirrettyjen kohteiden pitäisi osua kartalla oikeaan kohtaan mahdollisimman tarkasti, muuten syntyy virheellisiä karttamerkintöjä. Jos esimerkiksi karttamateriaalin ja omien kuvioden piirto tapahtuu eri koordinaattijärjestelmissä, tuloksena

voi olla metrien tai useiden kymmenien metrien sijaintivirhe. Tälläinen virhe voi äärimmillään johtaa siihen, että työkoneen kuljettaja hakkaa puita väärästä paikasta.

Jatkossa siirtotiedostojen luonnin ja lukemisen testaukseen tulee todennäköisesti kulumaan aikaa huomattavasti. Siirtotiedostossa sijaitsevat työmaiden kartat ja tiedot siellä suoritettavista toimenpiteistä. Näissä kahdessa vaiheessa täytyy huolehtia tiedon muuttumattomuudesta.

7 OHJELMAN TOIMINTA

Käyttöliittymän perustana toimivassa pääikkunassa suurimman tilan vie tarkoituksenmukaisesti karttaikkuna. Kartan tehtävänä on esittää sijaintia vastaava osa kartasta ja alueella mahdollisesti sijaitsevat työohjeet ja työmaakuviot. Ohjelman muut toiminnot liittyvät kartan ja tiedon käsittelyyn tai tiedon esittämiseen kartalta.

7.1 Työkalupakki

Kartan käsittelyyn liittyvät perustyökalut ovat kiinteässä työkalupakissa ikkunan yläosassa nopeaa ja jatkuvaa käyttöä silmälläpitäen (kuva 1).



Kuva 1. Työkalupakki

Työkalupakin toiminnot ovat:

Zoom +:

- lähestytään karttaa ja tarvittaessa vaihdetaan karttamateriaali tarkemmaksi.

Zoom -:

- loitonnutaan kartasta ja vaihdetaan tarvittaessa karttamateriaalia käsittämään suurempi alue

Käsi:

- vapaamuotoinen kartan siirto paikasta toiseen

Keskitys:

- kartan keskitys siihen pisteeseen jossa hiiren nappia painetaan

Info:

- työkalu, jolla saadaan kartalla olevista kohteista tiedot esille.

Mitta:

- mitalla mitataan matkoja kahden tai useamman pisteen välillä

GPS:

- kytkee GPS:n lukemisen päälle / pois

Hälytys:

- kytkee hälytyskohteiden seurannan päälle / pois

Kuittaus:

- kuittaa launneen hälytyksen

Tulostus ja tallennus

- nämä toiminnot joko tulostavat näkyvän kartan paperille tai muodostavat siitä kuvatiedoston

Kommentti-työkalut

- näillä kahdella työkalulla kartalle lisätään kommenttipiste tietoineen, luetaan pisteen tiedot tai poistetaan aiemmin luotu piste

7.2 Valikot

Osa harvemmin käytettävistä toiminnoista on sijoitettu valikoihin (kuva 2). Valikoiden rakenne on yksi tai maksimissaan kaksitasoinen.



Kuva 2. Valikot

Valikosta löytyvät toiminnot ovat:

Tiedosto

Tallenna kartta

- tällä muodostetaan näkyvästä kartasta kuvatiedosto, sama kuin työkalupakin tallennus

Lopeta

- sulkee ohjelman

Asetukset

Yleisasetukset

Kartta-asetukset

- näillä muokataan ohjelman käyttäytymistä eri tilanteissa

Perustiedot

Kaupat / Työmaat

- näiden kautta avautuu syöttöruudut kuudelle tietokannan tauluista, myös näiden tietojen muokkaus ja poisto ovat täällä.

Työohjeet

Selaa työohjeita..

- työohjeiden selaus, tiedot myös varasto- ja kuviotiedoista, jotka koskevat työmaata

Poista työohje

- työohjeiden poisto

Piirto

- avaa piirtotyökalut käyttöön

Info

- ohjelman nimi- ja versiotiedot

7.3 Piirtotyökalut

Piirtotyökaluissa sijaitsee erilaisten kuvioden, viivojen ja pisteiden piirtoon, poistoon ja muokkaukseen tarkoitettut työkalut (kuva 3).



Kuva 3. Piirtotyökalut

Neljää ensimmäistä työkalua käytetään erilaisten alueiden piirtämiseen. Alueet voivat kuvata esimerkiksi erityyppisiä metsänhoitotöitä, kaavoitusalueita, maa- tai metsätyppejä tai muita luokiteltuja aluetyppejä.

Kolme seuraavaa työkalua piirtävät viivoja. Näillä voidaan kuvata vaikkapa rajoja, reittejä, sähkölinjoja tai työmaiden reunoja.

Pistekohteiden piirtoon on tässä versiossa vain yksi työkalu, tässä tapauksessa varastopiste. Näitäkin työkaluja voisi olla useita erilaisia tarpeen mukaan, kuten rauhoitettu kohde, nähtävyys tai leiripaikka.

Kaksi viimeistä työkalua ovat piirrettyjen kuvioiden muokkaus ja poisto. Muokkaustyökalulla voidaan muokata aluekohteiden muotoa, kokoa ja sijaintia. Samoin viiva- ja pistekohteiden siirto tehdään tällä työkalulla. Poisto nimensä mukaisesti poistaa valitun kohteen tietoineen kartalta ja tietokannasta.

Piirtämisen jälkeen jokainen piirtotyökalu avaa syöttöikkunan, jonka kautta kohteet voidaan nimetä ja syöttää muita kohdetta koskevia tietoja. Nämä tiedot viedään tietokantaan, josta niitä tarpeen mukaan haetaan käsiteltäväksi.

8 YHTEENVETO

Työn ensisijaisena kohteena olleen työkoneohjelmiston rungon ja esimerkkisovelluksen teko eteni sujuvasti. Ohjelman perustoiminnallisuus on valmiina niiltä osin kuin se voidaan toteuttaa, tietämättä mihin ohjelmaa loppujen lopuksi tullaan käyttämään. Tässä vaiheessa kaikkia erikoisempia toimintoja ei ole liitettyä runkoon, etteivät kohdeprojektisidonnaiset asiat jäykistäisi ohjelman rakennetta liiaksi. Mahdollisissa jatkoprojekteissa runkoon pitää lisätä lähinnä uuden sovelluksen kohdealuetta vastaavat toiminnot ja lisäkomponentit.

Rungon testaus eteni rinnan lisättyjen toimintojen kanssa, mutta varsinainen testaus tapahtuu, kun ensimmäisen kerran aloitetaan rungon ympärille varsinaisen sovelluksen toteuttaminen oikeaan käyttöympäristöön.

Tärkeistä osista puuttuu lähinnä GPS-lukukomponentti, syynä komponentin keskeneräisyys. GPS:n avulla voitaisiin esimerkiksi toteuttaa ajoneuvon reitin seurantaa, tai muodostaa kuvioita kartalle kuljettujen reittien perusteella, riippuen ohjelmiston käyttötarkoituksesta.

Testattavana ollut uusi karttakomponentti edustaa kolmatta sukupolvea Gisforestin omaa tuotantoa edustavien karttakomponenttien sarjassa. Edeltäjiinsä verrattuna komponentin ominaisuudet ovat monipuolistuneet ja niiden käyttöönotto on yksinkertaistunut, komponentin tehokkuuden samaan aikaan parantuessa. Kartankäsittely on nopeutunut edellisiin versioihin verrattuna huomattavasti, josta selkeänä hyötynä on ohjelmien nopeutunut toiminta hitaammissa koneissa.

Mahdollisessa jatkokehityksessä ohjelman rakennetta pyritään yksinkertaistamaan entisestään. Esimerkiksi Access-tietokanta voitaisiin poistaa yksinkertaisimmissa tapauksissa kokonaan, sillä MapInfon karttamateriaalin säilytysformaatti on itsessään tietokanta. Tällä tavalla talletettuun tietoon voidaan tehdä samalla tavalla SQL-operaatioita kuin ulkoiseenkin tietokantaan.

GPS-komponentin lisäys tulee ajankohtaiseksi, kun sen ominaisuudet ovat kehittyneet tarpeelliselle tasolle. Puuttuvista ominaisuuksista voitaisiin mainita itse piirrettyjen reittien siirto laitteelle, joka on tällä hetkellä vielä täysin toteuttamatta. Itse muodostetun tiedon siirto GPS-laitteelle mahdollistaisi monipuolisten ohjelmien tekemisen. Pelkillä lukuominaisuuksilla varustettuna ohjelmat toimivat apuna seuratessa työn kulkua, mutta kirjoitusominaisuuksien avulla työskentelyä voitaisiin ohjailta uusilla reiteillä ja muilla tiedoilla, joita GPS-laitteeseen voidaan siirtää.

LÄHTEET

1. MapX Online help: "Introduction to MapX" , MapX:n mukana tuleva aputiedosto
2. MapX Online help: "What is MapX?" , MapX:n mukana tuleva aputiedosto
3. MapInfo Corporation, Troy New York, Karttakeskus Oy: MapInfo Professional 6.0 Käyttäjän opas, 2000
4. [Http://www.colorado.edu/geography/gcraft/notes/mapproj/mapproj_f.html](http://www.colorado.edu/geography/gcraft/notes/mapproj/mapproj_f.html)
(Luettu 14.11.2002)
5. [Http://www.colorado.edu/geography/gcraft/notes/coordsys/coordsys_f.html](http://www.colorado.edu/geography/gcraft/notes/coordsys/coordsys_f.html)
(Luettu 14.11.2002)
6. [Http://www.colorado.edu/geography/gcraft/notes/datum/datum_f.html](http://www.colorado.edu/geography/gcraft/notes/datum/datum_f.html)
(Luettu 14.11.2002)
7. Garmin ETrex / ETrex Summit käyttöopas, s. 54 , Garmin Etrex-laitteen mukana tuleva käyttöopas
8. Komponentti-ohjelmointi. Palaveri 28.9.2002. Kajaani
9. COM+ Component Services, KnowledgePool Tieturi, kurssimateriaalia
10. SQL Markku Laiho VAPK–kustannus, ATK-Instituutti
ISBN 951-861-204-861, 2. korjattu painos, 1992

LIITTEET

Liite 1: Työmaakuvion tietojen syöttö

Liite 2: Kaupan tietojen syöttö

Liite 3: Perustietojen syöttö

Liite 4: Toimeksianto

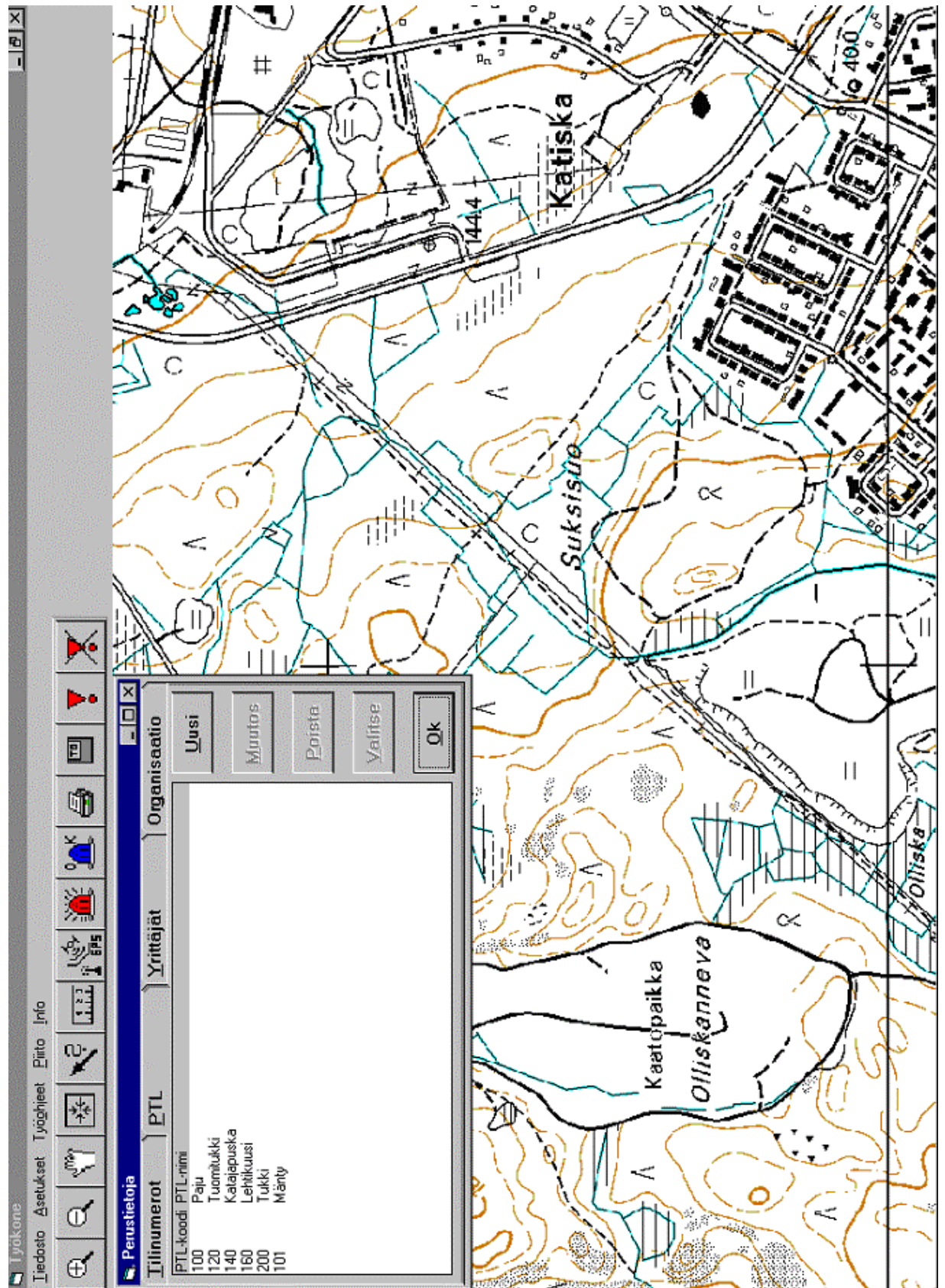
Liite 1: Työmaakuvion tietojen syöttö

The screenshot displays a GIS application window. The main area shows a topographic map with a construction site overlaid. The site is divided into several colored and hatched areas: a pink area, a green area labeled 'Sukksisäntö', a blue area, and a brown hatched area. A road labeled '1444' and a building labeled 'Katiska' are visible. The map also shows contour lines and other geographical features like 'Olliska' and 'Olliskanneva'. The toolbar on the left contains icons for zooming, panning, and other map navigation functions. The data entry form at the bottom contains the following information:

Kuvio	21
Kuvion N:o:	1
Työmaan N:o:	120
PTL:	1311
Tilinumero:	2000
Määrä 1:	2
Laji 1:	1200
Määrä 2:	2
Laji 2:	1821
Arvo:	1,468 ha
Pinta-ala:	0,604 m
Kuvion piiri:	

Buttons for 'Talleta' and 'Sulje' are located below the form. A legend on the right side of the map shows a hatched pattern next to the text 'Kaatopaikka Olliskanneva'.

Liite 2: Perustietojen syöttö



Liite 3: Perustietojen syöttö

The screenshot shows a GIS application interface. The background is a topographic map of Suksisaari and Katiska, featuring contour lines, roads, and buildings. Two data entry windows are overlaid on the map.

Top Window: Työmaat

Kaupat	Orig. tunnus	Asiakas	Puhelin	Puhelin 2	Sähköposti	Yhtälä
1	125231	iu	69 6986	698 698798		15749313
2	125231	vgl				1234567-

Bottom Window: Kaupan tiedot

Kauppanumero:

Org. tunnus:

Asiakkaan nimi:

Puhelin:

Puhelin 2:

Sähköposti:

Y-tunnus:

Buttons: Uusi, Muutos, Poista, Valitse, Ok, Tallenna, Sulje