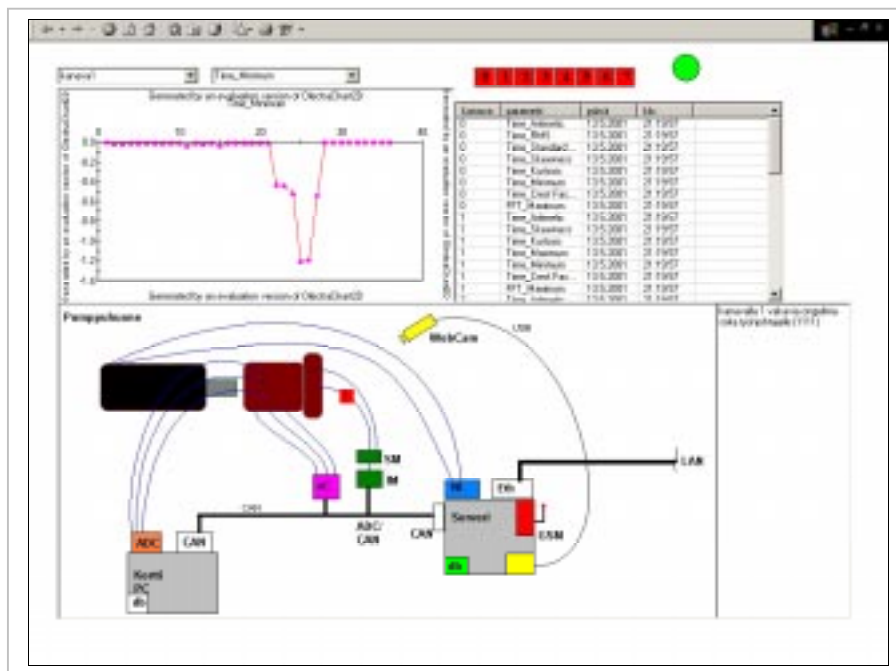


# Tutkimusraportti BVAL73-011151

## Selainpohjainen käyttöliittymä diagnostiikkajärjestelmälle

Jyrki Tervo, Jouni Alhainen, Eero Vaajoensuu



Espoo 5.11.2001

A Työraportti	
B Julkinen tutkimusraportti	X
Tutkimusraportti luottamuksellinen saakka	

Raportin nimi Selainpohjainen käyttöliittymä diagnostiikkajärjestelmälle	
Toimeksiantaja/rahoittaja ja tilaus pvm/nro TEKES, Teollisuus, VTT	Raportin numero BVAL73-011151
Projektin nimi Koneiden ja laitteiden kunnan ja käyttöolosuhteiden monitorointi ja diagnostiikka	Suoritteen numero V9SU00892
Laatija(t) Jyrki Tervo, Jouni Alhainen, Eero Vaajoensuu	Sivujen/ liitesivujen lukumäärä 13 / -
Avainsanat Käyttöliittymä, diagnostiikka, internet, selain	
Tiivistelmä KÄKI-teknologiaohjelmaan sisältyneen projektin Koneiden ja laitteiden kunnan ja käyttöolosuhteiden monitorointi ja diagnostiikka osatehtävänä oli suunnitella kunnossapitojärjestelmän käyttöliittymä, jolla mahdollistetaan valvottavien kohteiden seuranta tehokkaasti ja joustavasti.  Tässä raportissa demonstroitu käyttöliittymä on tietokantoihin nojautuva järjestelmä, jolla kunnossapitojärjestelmän laajentaminen sekä ylläpito on joustavaa. Mittaustiedot luetaan tietokannasta ja analysoinnin tulokset talletetaan tietokantaan. Järjestelmään mahdollisesti liitettävä asiantuntijajärjestelmä tukeutuu myös tietokantoihin, eli hakee tarvitsemansa tiedot tietokannasta ja tallettaa johtopäätökset sinne. Erityistä huomiota on kiinnitetty havainnollisuuteen hyödyntämällä multimedian mahdollisuuksia. Käyttöliittymän kehitystyön pohjana on ollut internet-selain, mikä antaa uusia mahdollisuuksia käyttöliittymän käytettävyydelle perinteisiin käyttöliittymiin nähden.	
Allekirjoitukset, Espoo 5.11.2001	
Kenneth Holmberg Tutkimuspäällikkö	Jyrki Tervo Erikoistutkija
Tarkastanut	
Jakelu (asiakkaat ja VTT):	
<i>VTT:n nimen käyttäminen mainonnassa tai tämän raportin osittainen julkaiseminen on sallittu vain VTT:ltä saadun kirjallisen luvan perusteella.</i>	

## Alkusanat

Tämä raportti liittyy osaltaan KÄKI-teknologiaohjelman menetelmäprojektiin M9 – Koneiden ja laitteiden kunnon ja käyttöolosuhteiden monitorointi ja diagnostiikka. Projektin ovat rahoittaneet TEKES, Metso Paper, Rautaruukki Raahen Steel, Lillbacka, Metso Minerals, Outokumpu PoriCopper, sekä VTT. Raportissa on esitetty lyhyesti hankkeessa demonstroitu diagnostiikkajärjestelmän selainpohjaisen käyttöliittymän toiminta. Kiitämme hankkeeseen osallistuneita yrityksiä ja projektin rahoittajia työtä kohtaan osoitetusta mielenkiinnosta, asiantuntevista kommentteista ja taloudellisesta tuesta.

Espoo, 5.11.2001

Tekijät

## Sisällysluettelo

<b>1</b>	<b>Johdanto .....</b>	<b>4</b>
<b>2</b>	<b>Tavoitteet.....</b>	<b>4</b>
<b>3</b>	<b>Menetelmät ja ohjelmointityökalut .....</b>	<b>6</b>
<b>4</b>	<b>Käyttöliittymä ja sen demonstraatio .....</b>	<b>6</b>
<b>5</b>	<b>Huolto-ohjeet .....</b>	<b>9</b>
5.1	Access tietokanta .....	9
5.2	Asetukset taulukko .....	10
5.3	Huolto-ohjelman toiminta .....	11
<b>6</b>	<b>Mahdollisuudet .....</b>	<b>12</b>
<b>7</b>	<b>Yhteenveto .....</b>	<b>12</b>

# 1 Johdanto

KÄKI-teknologiaohjelmaan sisältyneen projektin Koneiden ja laitteiden kunnan ja käyttöolosuhteiden monitorointi ja diagnostiikka - tavoitteena oli kehittää koneiden ja laitteiden kunnan diagnostisointiin ratkaisumalleja joilla mahdollistetaan valvottavien kohteiden kunnanvalvonta ja sen tuloksena huoltokohteen ja huoltoajankohdan määrittäminen. Toisena tavoitteena oli kehittää ratkaisumalleja jotka mahdollistavat perinteisten kunnanvalvontamenetelmien ohella myös olemassa olevan anturoinnin antaman informaation hyödyntämisen. Menetelmäprojektin perimmäisenä tavoitteena oli koota yhteen ja dokumentoida helposti hyödynnettävien työkalujen muodostama paketti, josta voidaan hyödyntää yritysten sovelluskohteen kannalta tarkoituksenmukaisia osia.

Käyttöliittymä toimii helppokäyttöisenä rajapintana kunnanvalvonnan sovellutuksiin ja järjestelmän tuottamiin tietoihin.

Käyttöliittymän kehittämisessä on lähdetty liikkeelle tietokantoihin nojautuvasta järjestelmästä, jolloin järjestelmän laajentaminen sekä ylläpito on joustavaa. Mittaustiedot luetaan tietokannasta, analysoinnin tulokset talletetaan tietokantaan. Järjestelmään mahdollisesti liitettävä asiantuntijajärjestelmä tukeutuu myös tietokantoihin, eli hakee tarvitsemansa tiedot tietokannasta ja tallettaa johtopäätökset sinne. Erityistä huomiota on kiinnitetty havainnollisuuteen hyödyntämällä multimedian mahdollisuuksia. Käyttöliittymän kehitystyö perustuu internet-selaimen, mikä antaa uusia mahdollisuuksia käyttöliittymän käytettävyydelle.

# 2 Tavoitteet

Käyttöliittymä muodostaa käyttäjän ja koneen välisen rajapinnan. Tämän vuoksi tavoitteeksi on selkeästi asetettu käyttäjätasoisuus. Termi on monisisältöinen ja teollisuusympäristössä sille voidaan asettaa tehokkuuteenkin liittyviä määritelmiä. Seuraavat tavoitteet ovat yleisesti esitettyjä [1]:

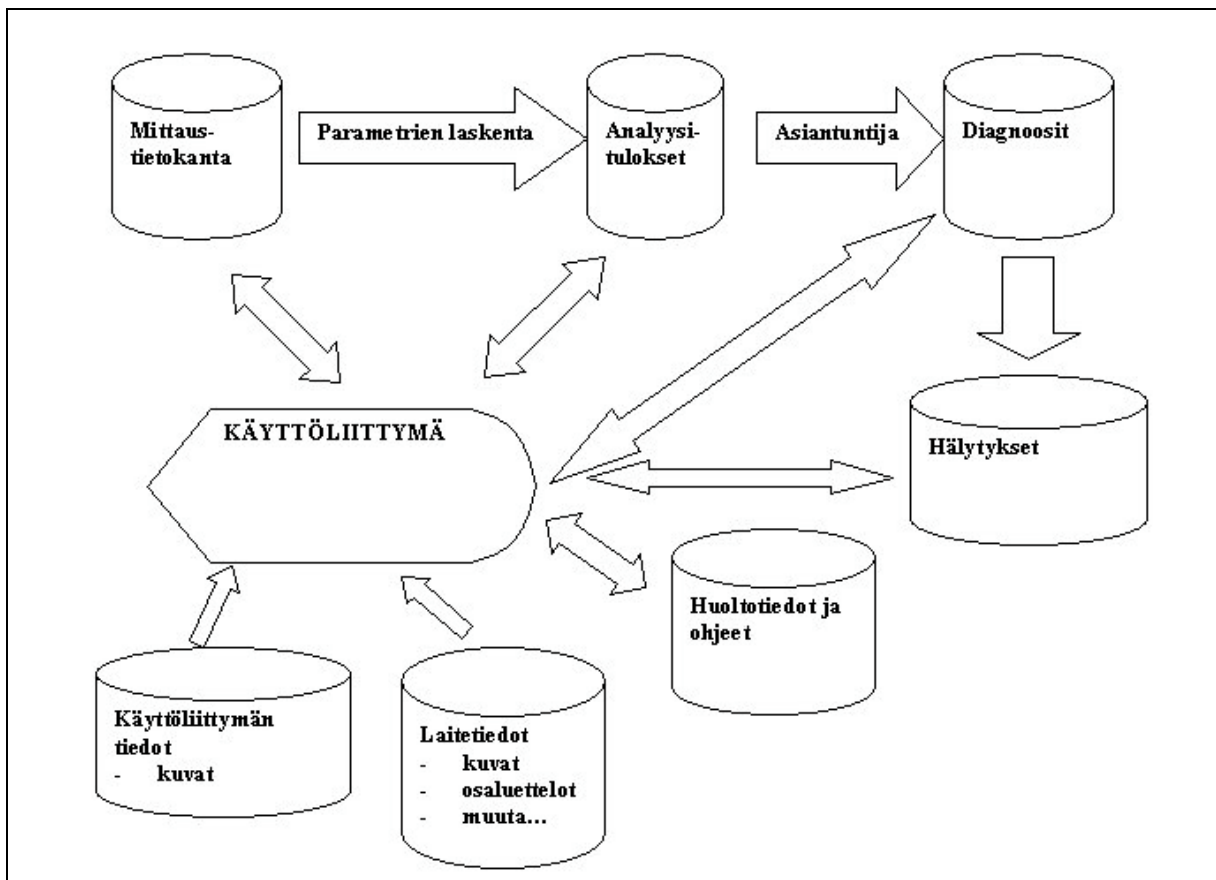
- Oppimisen helppous, eli kuinka nopeasti tyypillinen käyttäjä oppii käyttämään järjestelmää?
- Nopeus, eli tehtävien suoritus ei saa viedä liikaa aikaa.
- Käyttäjän tekemien virheiden määrä – eli millaisiin virheisiin ja kuinka usein järjestelmä antaa mahdollisuuden?
- Muistamisen helppous taas liittyy oppimiseen. Kuinka kauan käyttäjät muistavat järjestelmän toiminnan?
- Käyttäjän tyytyväisyys, eli tuntee käyttäjä turhautuneisuutta vai tyytyväisyyttä?

Tehokkuus nousee esiin lähes kaikissa edellä mainituissa tavoitteissa. Operaattorien koulutus on kallista, joten oppimisen helppous on erittäin tärkeää. Oppimista voidaan kuvata myös sanalla käytettävyyden – lisäksi termi pitää sisällään myös käyttäjän tyytyväisyyden. Nopeus, suorituskyky ja virheiden määrä vaikuttaa järjestelmän elinaikaisiin kokonaiskustannuksiin. Käyttäjän kokema tyydytys ei todennäköisesti ole taloudellisessa mielessä kovinkaan tärkeä,

vaikkakaan sitä ei saisi laiminlyödä. Listaan voidaan lisätä myös ylläpidettävyys – kuinka paljon tehokasta työaika joudutaan käyttämään järjestelmän ylläpitoon?

Käyttöliittymälle asetetut tärkeimmät tavoitteet olivat hyvä käytettävyys ja ylläpidettävyys. Käytettävyyden varmistamiseksi on käyttöliittymän oltava riittävän visuaalinen ja sisältää runsaasti grafiikkaa. Käyttöliittymässä on oltava mahdollista päästä käsiksi hyvinkin yksityiskohtaiseen mittaus- ja muuhun kunnossapitotietoon, mikä kertoo järjestelmän piiriin kuuluvien laitteiden kunnosta ja sen kehityksestä. Tästä syystä käyttöliittymässä on voitava siirtyä ylemmältä karkealta tasolta alemmalle ja yksityiskohtaisemmalle tasolle. Käyttöliittymän on oltava helposti ”tavoitettavissa” esimerkiksi eri puolilla tehdasta – myös suoraan asiantuntijoiden työpisteistä. Lisäksi käyttöliittymään on voitava liittää järjestelmiin liittyviä lisätietoja, kuten esimerkiksi koneiden ja laitteiden piirustuksia.

Edellä mainituista syistä johtuen päädyttiin siihen, että käyttöliittymän tulee olla web-sivusto, mitä voidaan käyttää web-selaimella. Pyrkimyksenä oli myös hyödyntää tietokantoja, jolloin käyttöliittymä voi poimia nimet, linkit, osoitteet, tekstit, hälytystiedot, graffit jne. Tietokannasta, jolloin esimerkiksi uusien mittausten ja/tai valvottavien kohteiden lisääminen järjestelmään olisi yksinkertaista. Lohkokaavio käyttöliittymän mahdollisesta ympäristöstä on esitetty kuvassa 1. Käyttöliittymän kautta voidaan siis lukea tietokantoja, joita voivat olla mittaustiedot, mittaustiedoista lasketut analyysitulokset, asiantuntijan tai asiantuntijajärjestelmän tekemät diagnoosit sekä niistä generoidut hälytykset. Tärkeä osa on myös tiedot tehtävistä huolloista sekä tiedot laitteista ja koneista.



Kuva 1. Lohkokaavio käyttöliittymän ympäristöstä.

Lisäksi pohdittiin mahdollisuutta ohjelmoida käyttöliittymää varten oma selain, mutta ajatuksesta luovuttiin koska haluttiin käyttää käytössä oleva työaika varsinaisen käyttöliittymädemostraaation suunnitteluun ja ohjelmointiin.

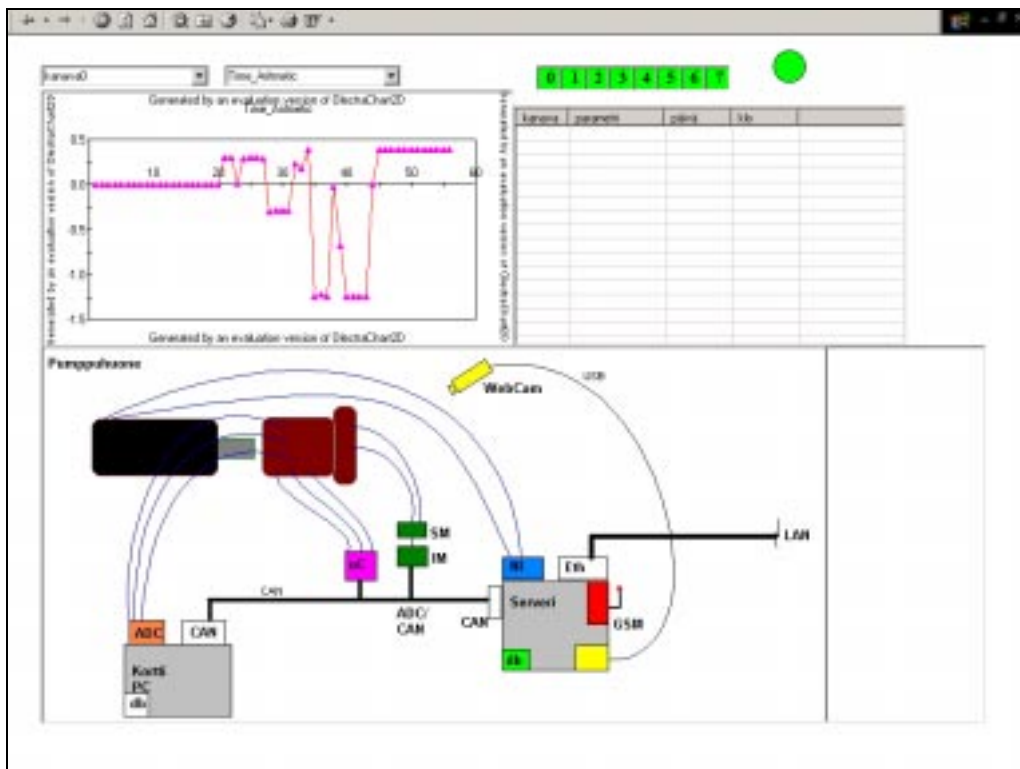
Koska hankkeessa toteutettava käyttöliittymä on demonstraatioluonteinen, päätettiin käyttöliittymä toteuttaa VTT Valmistustekniikan pumppukoelaitteistolle.

### 3 Menetelmät ja ohjelmointityökalut

Käyttöliittymän perusta eli tietokannat tehtiin Access-ohjelmistolla [2, 3]. Muut käytetyt ohjelmointityökalut olivat Microsoftin FrontPage [4, 5], Visual Basic 6.0 [6, 7, 8], Java Script [9], ASP [10], sekä Olectra Chart 6.02 ja True WebChart 6.02 [11, 12]. Visual Basic 6.0:lla ohjelmoitiin tarvittavat ActiveX-kontrollit. TrueWebChartia käytettiin grafiikan toteuttamiseen ActiveX-kontrollissa. FrontPagella tehtiin html-koodi, joka mahdollistaa web-selaimen käytön tulosten katselemisen. Access-tietokantaa käytetään mittaustulosten ja niistä laskettujen parametrien tallentamiseen.

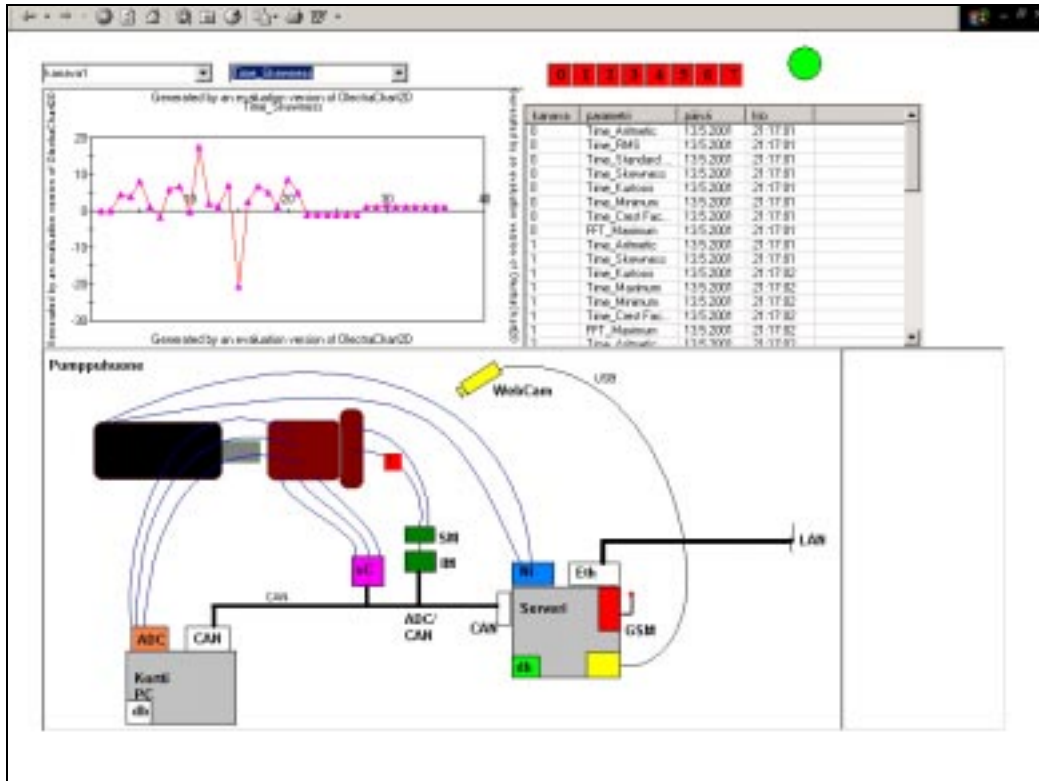
### 4 Käyttöliittymä ja sen demonstraatio

Mitatut tiedot ja lasketut arvot on talletettu Microsoftin access-tietokantaan. Tietokannan tietojen visualisointia toteutettiin ActiveX-kontrollina, joka sisällytettiin html-koodiin. Kuvaa-jaan käytettiin Olectran True WebChart OCX-komponenttia. Tällä hetkellä demo toimii ainoastaan Microsoftin Internet Explorer selaimella. Perustilassa käyttäjä voi käydä katsomassa eri kohteissa tehtyjä mittausten tuloksia ja laskettuja arvoja.



Kuva 2. Käyttöliittymän näkymä perustilassa.

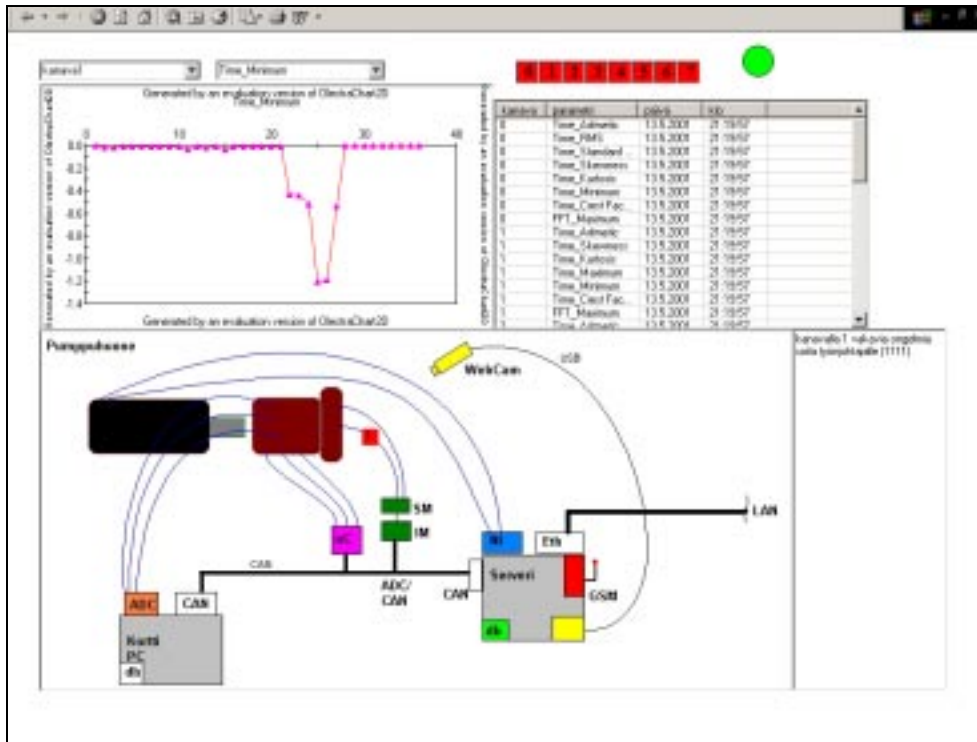
Ohjelma käy tietokantaa jatkuvasti läpi. Kunkin kanavan jokaiselle tarkkailtavalle parametrille on määritetty tietokannassa hälytysrajat, joiden ylittäminen tai alittaminen aiheuttaa hälytyksen. Niiden mittauskanavien numerot, joiden hälytysrajat rikotaan, muuttuvat punaiseksi. Esimerkkitapauksessa kaikilla kanavilla esiintyy ongelmia. Listalta voi lukea millä mittauskanavalla esiintyy ongelmia ja mitkä parametrit ovat hälytysrajojen ulkopuolella. Klikkaamalla hiirellä tiettyä riviä listalta saadaan ko. mittaukset näkyviin kuvaajaan. Samalla kuvaajan alla olevassa mittauksen 'layout' kuvassa vilkkuu punainen numeroitu neliö osoittaen missä ongelman aiheuttava mittauskanava sijaitsee (kuva 3).



Kuva 3. Käyttöliittymän näkymä vikatilanteessa.

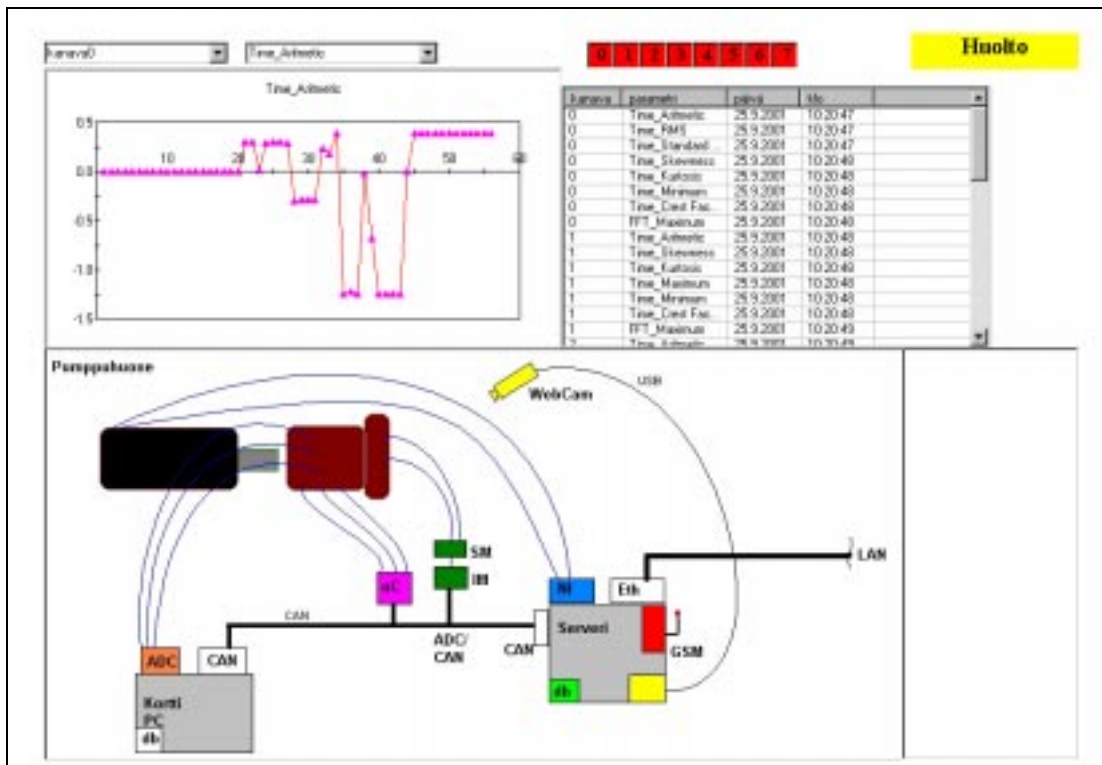
Klikkaamalla hiirellä layout-kuvassa näkyvää vikatilanteen aiheuttamaa kanavaa saadaan näkyviin mahdollisia toimintaohjeita (kuva 4).





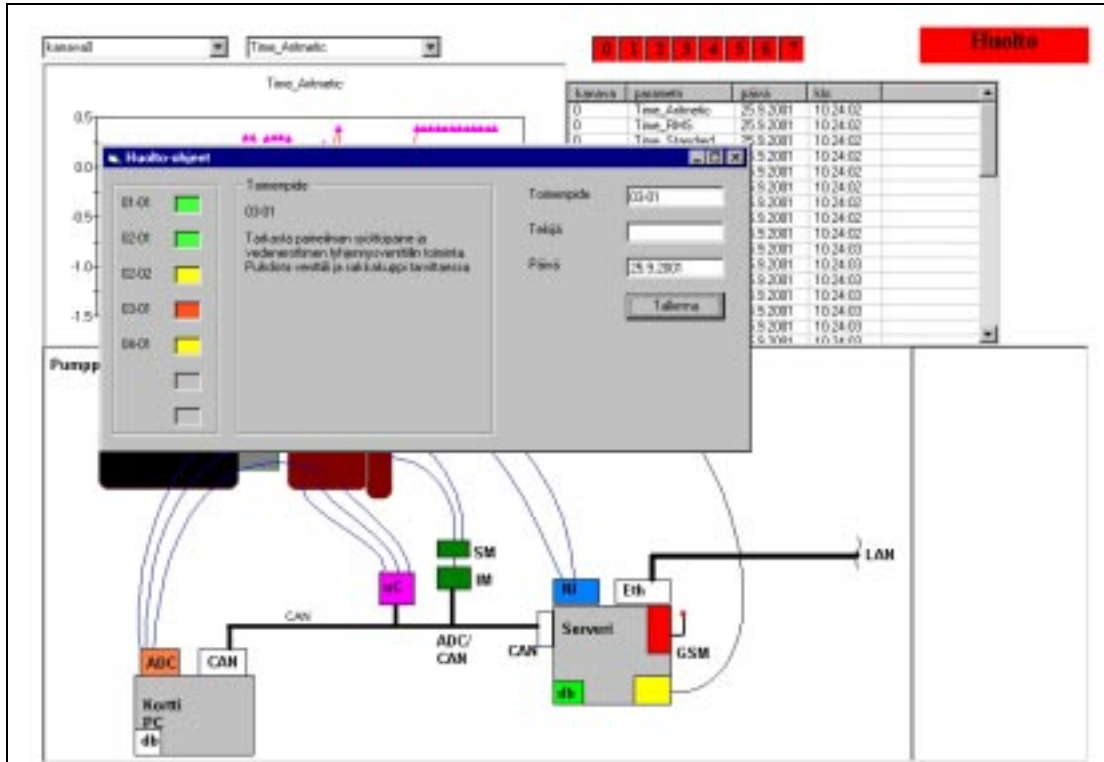
Kuva 4. Käyttöliittymän näkymä vikatilanteessa toimintaohjeella.

Huoltovälin täytyessä tulee näytön oikeaan yläkulmaan painike, jonka pohjaväri kuvaa kii-reellisemmän tarvittavan huoltotoimenpiteen statusksen.



Kuva 5. Näyttö huoltotarpeesta varoituksella.

Painettaessa varoituspainiketta saadaan uusi ikkuna, jossa esitetään huoltotarpeen omaavat kohteet ja niihin liittyvät toimenpideohjeet. Ikkunassa voidaan myös kuitata huoltotoimenpiteet tehdyksi.



Kuva 6. Näyttö huoltotarpeesta sisältäen toimenpideohjeet.

## 5 Huolto-ohjeet

Huolto-ohjelmalla mallinnetaan järjestelmää, jossa työstökoneen tai muun laitteiston huollot eivät ole sidottuja määrättyyn viikkorytmiin (esim. viikko huollot, kuukausi huollot jne.), vaan että laitteiston huollot suoritetaan todellisten ajotuntien/ajokilometrien mukaan. Ohjelma toimii siten, että se antaa värikoodiin perustuvan hälytyksen, kun laitteistolle on tarvetta suorittaa huolto. Värikoodit ovat seuraavat: vihreä = huolto tulossa, keltainen = huolla, punainen = huolto myöhässä. Huollon tarve ohjelmassa perustuu ajosysäyksiin (10s), koska tämä yksikkö on demo-ohjelman toiminnan kannalta sopivan pitkä aikayksikkö, ja tietokannassa määriteltäisiin huoltoväleihin.

### 5.1 Access tietokanta

Ohjelma käyttää Access-tietokantaa huoltotoimenpiteiden ja asetustietojen lukemiseen sekä tehtyjen toimenpiteiden tallentamiseen. Tietokanta sisältää kolme taulukkoa, jotka ovat asetukset, toimenpiteet ja tehdyt toimenpiteet. Seuraavassa kuvaukset näistä taulukoista.

Huolto-ohjeet taulukko näkyy taulukossa 1. Taulukkoon on syötetty joitakin keksittyjä toimenpiteitä sekä kullekin toimenpiteelle huoltoväli.

Taulukko 1. Esimerkki huolto-ohjeista

<b>Tunnus</b>	<b>Toimenpide</b>	<b>huoltoväli</b>	<b>ajotunnit</b>
01-01	Vaihda hydraulikan öljyt	150	0
02-01	Vaihda ilman suodatin	150	0
02-02	Tarkista ilman paine	600	0
02-03	Tarkista pneumatiikka letkujen kunto ja vaihda tarvittaessa	7000	0
03-01	Puhdista leikkaus pöytä	150	0

Tunnussarakkeessa on toimenpiteen tunnus. Se on muotoa xx-xx. Ensimmäisessä osassa ilmoitetaan kohteen numero esim. kohde 1 ilmoitetaan 01 jne. Jälkimmäisessä osassa ilmoitetaan toimenpiteen numero. Näin ollen esim. 23-11 merkitsee, että kyseessä on kohteen 23 toimenpide 11. Eri toimenpiteillä on aina oltava eri tunnus, eli tunnussarakkeessa ei saa olla kahta samaa tunnusta. Toimenpidesarakkeessa ilmoitetaan ko. toimenpiteen suoritusohjeet. Huoltovälisarakeessa ilmoitetaan ajosysäyksinä (1 sysäys = 10s) ko. huollon suoritusväli. Esimerkiksi, jos huolto tulee suorittaa 150 sysäyksen (1500 sekunnin) välein, sarakkeeseen merkitään 150. Ajotunnit sarake päivittyy automaattisesti huolto-ohjelman toimesta. Sarakkeessa oleva luku ilmaisee kuinka kauan kone on käynyt viimeisestä ko. toimenpiteelle tehdystä huollosta.

## 5.2 Asetukset taulukko

Asetukset taulukkoon on tallennettu tieto hälytysrajoista. Hälytysrajoja on kolme vihreä, keltainen ja punainen. Taulukossa 2 rajoiksi on asetettu 0,8, 0,95 ja 1,3. Tämä tarkoittaa sitä, että kun ajotunnit on 80% huoltovälistä huolto-ohjelma antaa vihreän hälytyksen ja vastaavasti kun ajotunnit on 95% huoltovälistä saadaan keltainen hälytys jne.

Taulukko 2. Esimerkki varoituskoodien asetusarvoista.

<b>Vihreä raja</b>	<b>Keltainen raja</b>	<b>Punainen raja</b>
0,8	0,95	1,3

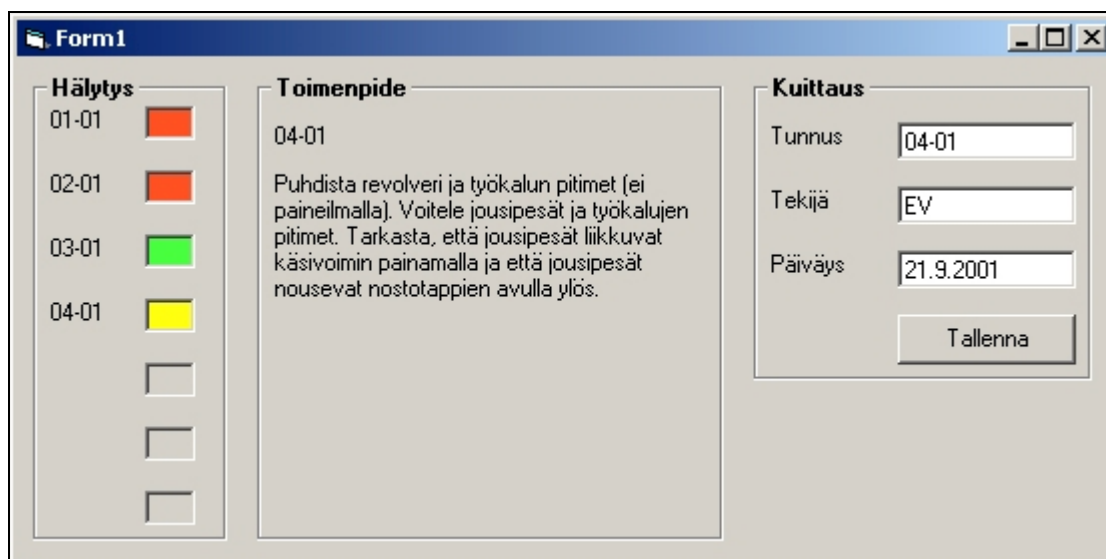
Taulukko 3. Esimerkki tehdyt toimenpiteet - taulukosta.

<b>Numero</b>	<b>Tunnus</b>	<b>Tekijä</b>	<b>Päiväys</b>
1	01-01	EV	22.8.2001
2	02-01	EV	22.8.2001
3	02-02	JT	5.9.2001

Taulukossa 3 olevassa tehdyt toimenpiteet taulukossa numerosarake ilmaisee ko. huollon järjestysnumeron. Sarake päivittyy automaattisesti. Tunnussarake ilmaisee tehdyn toimenpiteen. Tekijäsarake ilmaisee kuka huollon on suorittanut. Sarakkeeseen voi syöttää työntekijän kononimen tai pelkästään nimikirjaimet. Päiväyssarakkeessa on toimenpiteen suorituspäivä.

## 5.3 Huolto-ohjelman toiminta

Huolto-ohjelma on kirjoitettu Visual Basic 6.0:lla. Kuvassa 7 on huolto-ohjelman ohjelmaikkuna. Ikkuna on jaettu kolmeen kehykseen, jotka ovat Hälytys, Toimenpide ja Kuittaus. Hälytyskehykseen ilmestyy huoltotoimenpiteen tunnus ja värikoodi silloin kun ko. toimenpide tulee suorittaa. Toimenpide kehyksessä voidaan lukea mitä huoltotoimenpide edellyttää tehtäväksi. Kuittaus kehyksessä voidaan kuitata toimenpide tehdyksi.



Kuva 7. Huolto-ohjelman ohjelmaikkuna

Ohjelma toimii 10 sekunnin syklillä. Tämä tarkoittaa sitä, että ohjelma suorittaa 10 sekunnin välein seuraavien 6 toimenpiteen joukon. Toimenpiteet ovat suoritus järjestyksessä.

1. Ohjelma poistaa aiemmat hälytykset hälytyskehyksestä (mikäli niitä on)
2. Ohjelma laskee huolto-ohjeet taulukkoon syötetyt rivit (siis eri toimenpiteiden määrän)
3. Ohjelma siirtyy huolto-ohjeet taulukon ensimmäiselle riville
4. Ohjelma kasvattaa huolto-ohjeet taulukon ko. rivin ajotunnit sarakkeen arvoa yhdellä.
5. Ohjelma vertaa huolto-ohjeet taulukon ko. rivin huoltoväli sarakkeen arvoa ajotunnit sarakkeen arvoon ja antaa hälytyksen, mikäli asetukset taulukossa annetut hälytysrajat täyttyvät. Hälytys kirjoitetaan hälytys kehyksen ylimmälle vapaalle riville.
6. Ohjelma siirtyy huolto-ohjeet taulukon seuraavalle riville ja suoritta kohdat 4 ja 5, tai jos seuraavaa riviä ei ole siirtyy odotustilaan (10 s)

Hälytyksen ilmestyttyä hälytyskehykseen saa tunnusta klikkaamalla esille toimenpiteen huolto-ohjeen. Huolto-ohje ilmestyy toimenpidekehykseen. Kehyksen ensimmäisellä rivillä on toimenpiteen tunnus. Jos toimenpide halutaan kuitata suoritetuksi voidaan tunnusta klikata, jolloin kuittauskehykseen ilmestyy ko. tunnus ja päivämäärä.

Ohjelmaikkunan oikeassa laidassa on kuittauskehys. Tässä kehyksessä voidaan kuitata toimenpide suoritetuksi syöttämällä toimenpiteen tunnus, tekijän nimi ja päiväys niille varattuihin ruutuihin ja painamalla tallenna painiketta. Kuittauksen yhteydessä ohjelma tallettaa tehdyn toimenpiteen tehdyt toimenpiteet taulukkoon, poistaa ko. hälytyksen hälytyskehyksestä ja nollaa toimenpiteet taulukosta ko. toimenpiteen ajotunnit sarakkeen.

## 6 Mahdollisuudet

Toteutetussa demonstraatioissa ei ehditty havainnollistaa kaikkia piirteitä, jotka ovat mahdollista toteuttaa. Näistä tärkein lienee vikailmoitusten ja hälytysten toimittaminen vastuuhenkilöille. Vikailmoitukset voidaan lähettää määrämuotoisina esimerkiksi sähköpostilla tai tekstiviestinä GSM-puhelimeen. Viestin lähetys käyttöliittymästä voi tapahtua automaattisesti tai sitten valvomohenkilöstön toimesta.

Käyttöliittymän vikailmoitukset voivat olla myös huomattavasti monipuolisempia kuin mitä on demonstroitu. Käyttöliittymään voidaan toteuttaa linkki, mitä kautta voi päästä esimerkiksi tietokantaan johon on kerätty vikakohteen ongelmahistoria (mitä on rikkoutunut, mitä on korjattu ja miten). Vastaavasti tietokantaan on mahdollista tallettaa hyvinkin yksityiskohtaisia huolto-ohjeita, koneiden piirustuksia ynnä muuta sellaista tietoa, mikä auttaa kunnossapitohenkilökuntaa.

Ohjelmalla voi olla eri käyttäjätasoja. Käytännössä tarvitaankin vähintään kaksi eri tasoa – järjestelmävastaavat ja käyttäjät. Lisäksi saattaa olla tarvetta määritellä käyttäjillekin erilaisia oikeuksia. Osalla voi olla esimerkiksi oikeudet katsoa pelkästään mittauksia ja selailta hälytyksiä, osalla taas täytyy olla oikeus esimerkiksi kuitata ongelmat ja mahdolliset huollot.

## 7 Yhteenveto

Hankkeessa osoitettiin selainpohjaisen käyttöliittymän olevan mahdollista toteuttaa suhteellisen yksinkertaisin työkaluin. Käyttöliittymän tekninen toteutus ei lopultakaan näytä olevan suurin vaikeus, vaan selkeän vision luonnosteleminen siitä mitä käyttöliittymältä odotetaan - mitä siltä halutaan ja miltä sen tulee näyttää. Vaikka käyttöliittymän funktioon ei kuulu koneiden diagnostisointi tai mikään muukaan kunnonvalvonnan perustehtävä, on se silti tärkeä osa nykyaikaista kunnonvalvonnan järjestelmää. Toteutettuna tietokantoihin nojaava selainpohjainen käyttöliittymä muodostaa joustavan ja havainnollisen rungon, missä diagnostiikka-järjestelmään liitettäviä moduuleita voidaan yhdistää.

## Lähdeluettelo

- [1] Shneiderman, B., *Designing the User Interface – Strategies for effective Human-Computer Interaction*. Addison-Wesley Longman Inc., Third Edition, 1998.
- [2] Microsoft Access 97. Microsoft Corporation, 1996.
- [3] Prague, G., Irwin, M., *Access 97 Bible*. IDG Books Worldwide Inc., Foster City, California, USA, 1997.
- [4] Microsoft Frontpage 2000. Microsoft Corporation, 1999.
- [5] Buyens, J., *Running Microsoft Frontpage 2000 – käyttäjän käsikirja*. Oy Edita Ab/IT Press, Jyväskylä, 2000.
- [6] Microsoft Visual Basic 6.0. Microsoft Corporation, 1998.
- [7] Rahmel, D., *Visual Basic 6 tietokantaohjelmointi*. IT Press, Jyväskylä 1999.
- [8] Rahmel D., *Building Web Database Applications with Visual Studio 6*. McGraw-Hill, California, USA, 2000.

- [9] Moncur, M., Javascript. IT Press, Jyväskylä 2000.
- [10] Wille, C., Koller, C., Active Server Pages Trainer – Tietokantaohjelmointi Internetissä. Oy Edita Ab/IT Press, Helsinki, 2000.
- [11] Olectra Chart 6.0.2. APEX Software Corporation, 1999.
- [12] True Web Chart 6.0.2. Apex Software Corporation, 1999.